TUTORIAL 05:
NUMERICAL ASPECT II: STABILITY AND PGF

STEP 1: Logging onto the HPC cluster

- From a terminal/konsole:

```
ssh -X login@scp.chpc.ac.za
```

- Request one node with the alias command **qsubi1**

```
qsubi1
```

NRF SAEON · NICIS CHPC · IRD Institut de Recherche pour le Développement FRANCE · CNRS · LEGOS · CROCO Sud Coastal and Regional Ocean COmmunity model · SHOM · LOCEAN · Universidad de Concepción · DGEO Universidad de Concepción · Agencia Nacional de Investigación y Desarrollo Gobierno de Chile

OBJECTIVES

- Analyse the temperature equation, again !
- Will the ocean temperature be warm enough to swim tomorrow?
- Look at the consistency of a numerical scheme
- Analyse the stability of a numerical scheme
- Uncover the CFL condition

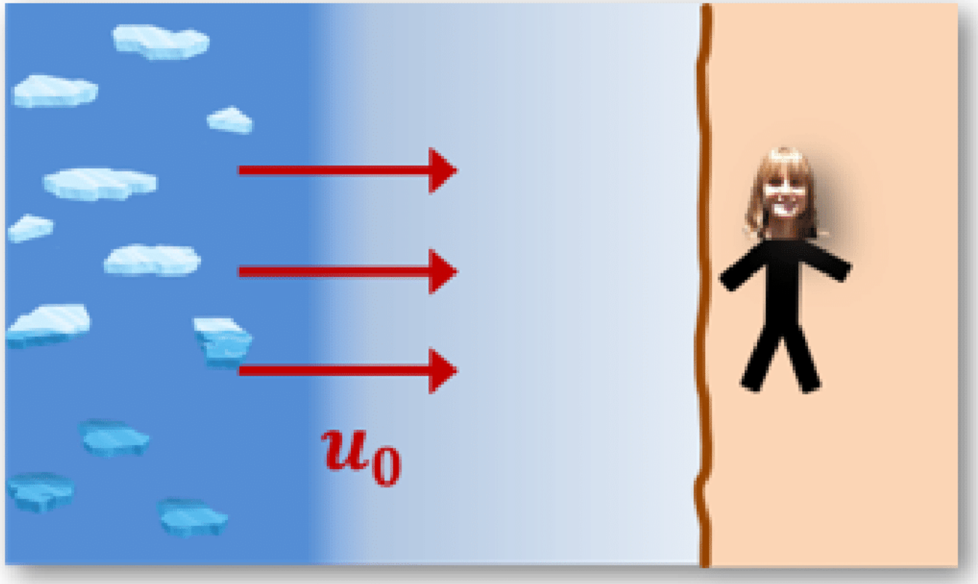
Consistency and Stability: Introduction (1/3)

→ From CROCO 3D temperature equation:

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

➥ We simplify the processes at work by studying a simple case study, where:

- there is no surface forcing (adiabatic).
- there is a constant current directed toward the shore u_0 (homogeneous in y).
- there is no variation of temperature with depth (barotropic case), i.e. we can cross-out the vertical turbulent diffusion term.
- there is no horizontal diffusion.



→ From the 3D temperature, we need to solve the 1D advection equation:

$$\frac{\partial T}{\partial t} + u_0\frac{\partial T}{\partial x} = 0 \quad x \in [0, L], \quad t \in [0, T] \quad (1)$$

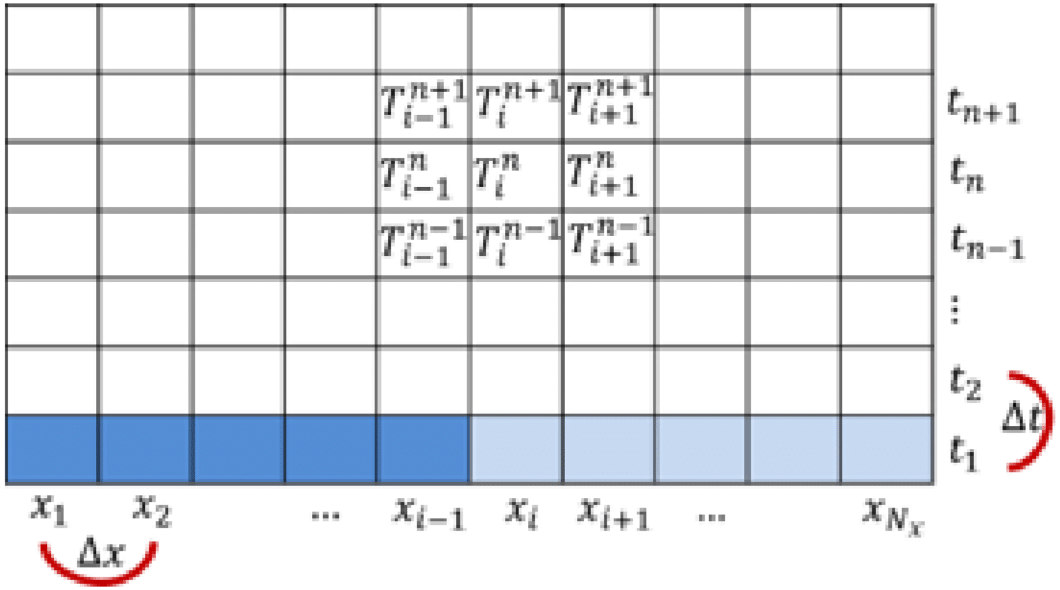
➥ There are only first-order derivatives in time and space.

➥ The initial conditions that portray this temperature front are known. The constant parameter u_0 (the current advecting the cold condition toward the coast) must be given.

Consistency and Stability: Introduction (2/3)

→ Same as in **#TUTORIAL03**, we work on a discretized model grid. We replace the continuous domain $[0, L] \times [0, T]$ by a set of **equally spaced mesh points**, such that:

$$x_i = i\Delta x, i = 1, \dots, N_x \quad \text{and} \quad t_n = n\Delta t, n = 1, \dots, N_t$$

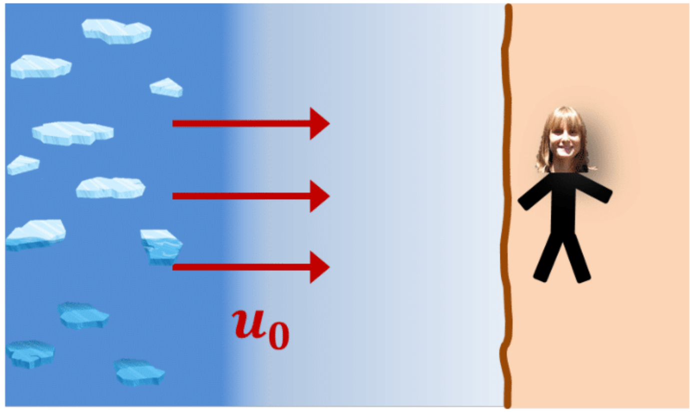


→ We need to find a **consistent** approximation for the equation derivatives: $\frac{\partial T}{\partial t}$ and $\frac{\partial T}{\partial x}$ on our model grid. This means that the error between the discretized and the real solution must approach 0.

Consistency and Stability: Introduction (3/3)

- We have the grid of our model (horizontal ~~and vertical~~)
- Lets solve this equation (1D-advective equation) :

$$\frac{\partial T}{\partial t} + u_0 \frac{\partial T}{\partial x} = 0 \quad x \in [O, L], \qquad t \in [0, T] \quad (1)$$



- We know ***T*** at **time t** at all x positions,
  - We want to compute ***T*** at **time t+dt**
- Lets find a good **numerical scheme** to solve this problem

> We need to find a **consistent** approximation for
> the **derivatives** of the equation : $\frac{\partial T}{\partial t}$ and $\frac{\partial T}{\partial x}$

Consistency of a numerical scheme (1/5)

→ In order to quantify the error we make by solving any equation on a spatial and temporal discretised grid, we use the Taylor series expansion of a continuous function f at a point x_0 close to a reference point x:

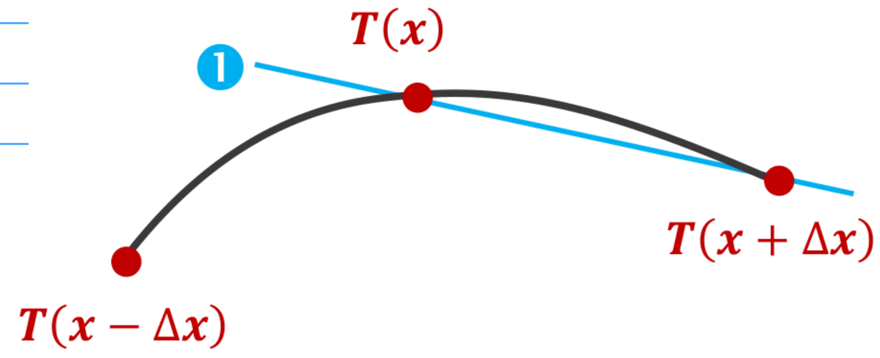
$$f(x_0) = f(x) + \frac{f'(x)}{1!}(x_0 - x) + \frac{f''(x)}{2!}(x_0 - x)^2 + \cdots + \frac{f^n(x)}{n!}(x_0 - x)^n + R(x)$$

➥ If x is close to x_0, such that $x_0 = x + \Delta x$, we can write:

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

→ Let discretize $\frac{\partial T}{\partial x}$. There are 3 different numerical schemes:

❶ The Downstream Scheme $\frac{\partial T}{\partial x} \simeq \frac{T(x+\Delta x) - T(x)}{\Delta x}$



Consistency of a numerical scheme (1/5)

→ In order to quantify the error we make by solving any equation on a spatial and temporal discretised grid, we use the Taylor series expansion of a continuous function f at a point x_0 close to a reference point x:

$$f(x_0) = f(x) + \frac{f'(x)}{1!}(x_0 - x) + \frac{f''(x)}{2!}(x_0 - x)^2 + \cdots + \frac{f^n(x)}{n!}(x_0 - x)^n + R(x)$$

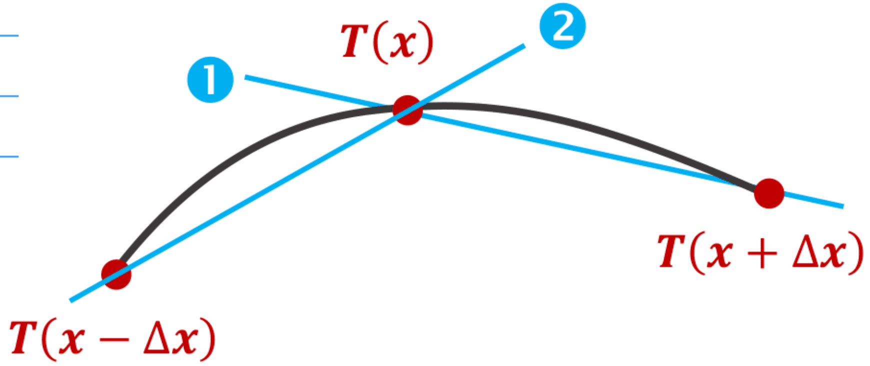
➥ If x is close to x_0, such that $x_0 = x + \Delta x$, we can write:

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

→ Let discretize $\frac{\partial T}{\partial x}$. There are 3 different numerical schemes:

❶ The Downstream Scheme $\frac{\partial T}{\partial x} \simeq \frac{T(x+\Delta x) - T(x)}{\Delta x}$

❷ The Upstream Scheme: $\frac{\partial T}{\partial x} \simeq \frac{T(x) - T(x-\Delta x)}{\Delta x}$



Consistency of a numerical scheme (1/5)

→ In order to quantify the error we make by solving any equation on a spatial and temporal discretised grid, we use the Taylor series expansion of a continuous function f at a point x_0 close to a reference point x:

$$f(x_0) = f(x) + \frac{f'(x)}{1!}(x_0 - x) + \frac{f''(x)}{2!}(x_0 - x)^2 + \cdots + \frac{f^n(x)}{n!}(x_0 - x)^n + R(x)$$

➥ If x is close to x_0, such that $x_0 = x + \Delta x$, we can write:

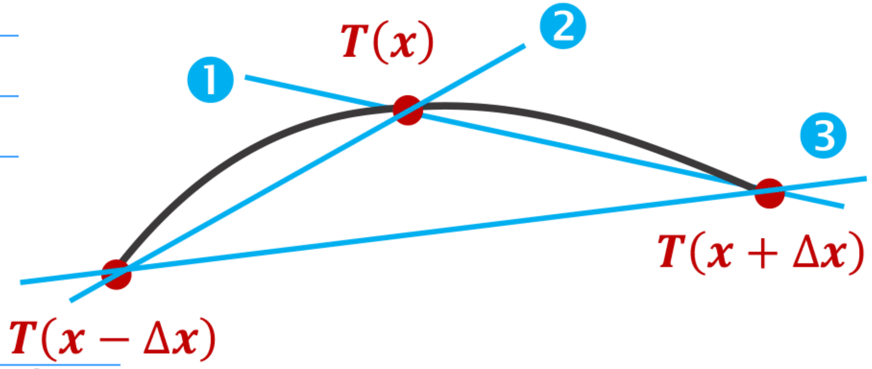
$$f(x + \Delta x) = f(x) + \frac{f'(x)}{1!}\Delta x + \frac{f''(x)}{2!}\Delta x^2 + \cdots + \frac{f^n(x)}{n!}\Delta x^n + R(x)$$

→ Let discretize $\frac{\partial T}{\partial x}$. There are 3 different numerical schemes:

❶ The Downstream Scheme $\frac{\partial T}{\partial x} \simeq \frac{T(x+\Delta x) - T(x)}{\Delta x}$

❷ The Upstream Scheme: $\frac{\partial T}{\partial x} \simeq \frac{T(x) - T(x-\Delta x)}{\Delta x}$

❸ The Centered Scheme $\frac{\partial T}{\partial x} \simeq \frac{T(x+\Delta x) - T(x-\Delta x)}{2\Delta x}$



Consistency of a numerical scheme (2/5)

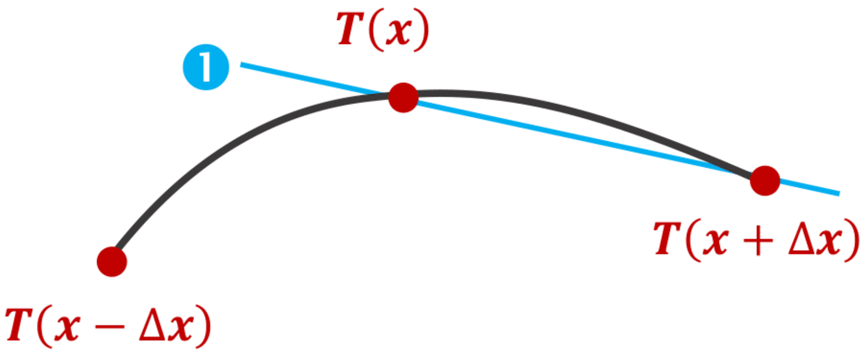
- Estimation of the error we make when we choose the Downstream Error

$$\frac{\partial T}{\partial x} = \frac{T(x+\Delta x) - T(x)}{\Delta x} + \text{Error}$$

$$f(x + \Delta x \quad T(x+\Delta x) = T(x) + \frac{T'(x)}{1!}\Delta x + \frac{T''(x)}{2!}\Delta x^2 + \cdots \quad + R(x)$$

$$\frac{\partial T}{\partial x} = \frac{T(x) + \frac{T'(x)}{1!}\Delta x + \frac{T''(x)}{2!}\Delta x^2 - T(x)}{\Delta x}$$

$$\frac{\partial T}{\partial x} = T'(x) + \frac{T''(x)\Delta x}{2!}$$



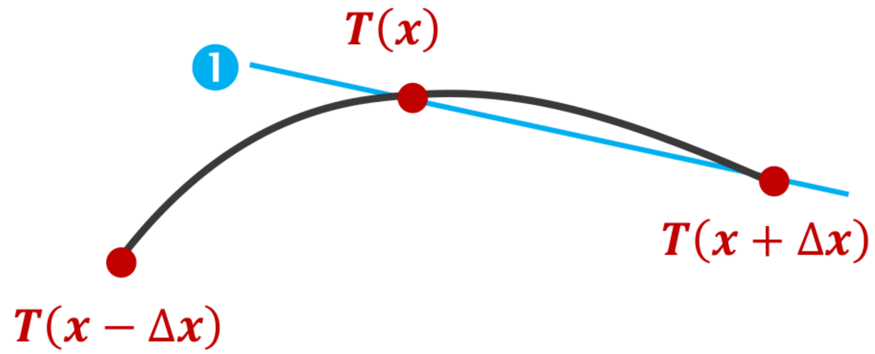
Consistency of a numerical scheme (2/5)

- Estimation of the error we make when we choose the Downstream Error

$$\frac{\partial T}{\partial x} = \frac{T(x+\Delta x) - T(x)}{\Delta x} + \text{Error}$$

$$\text{Error} = \frac{\Delta x}{2!} T''(x)$$

1st order error



Consistency of a numerical scheme (3/5)

- Estimation of the error we make when we choose the Upstream Scheme

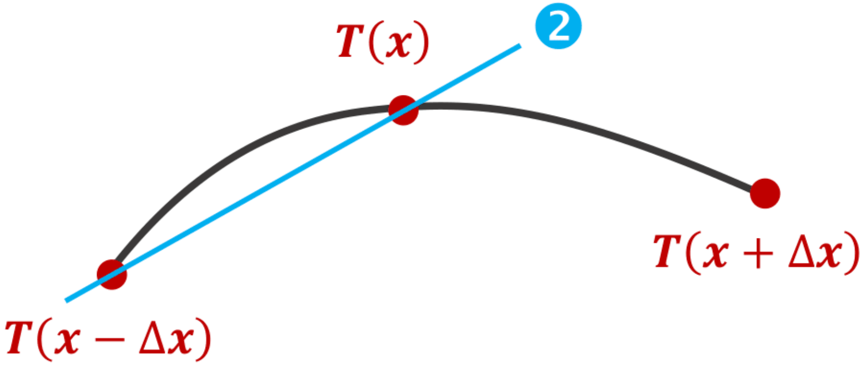
$$\frac{\partial T}{\partial x} = \frac{T(x) - T(x-\Delta x)}{\Delta x} + \text{Error !}$$

$$T(x - \Delta x) = T(x) - \frac{T'(x)}{1!}\Delta x + \frac{T''(x)}{2!}\Delta x^2 + \cdots$$

$$\frac{\partial T}{\partial x} = \frac{T(x) - \left(T(x) - \frac{T'(x)}{1!}\Delta x + \frac{T''(x)\Delta x^2}{2!}\right)}{\Delta x}$$

$$\frac{\partial T}{\partial x} = T'(x) - \frac{T''(x)}{2!}\Delta x$$

1st order error



Consistency of a numerical scheme (4/5)

- Estimation of the error we make when we choose the Centered Scheme

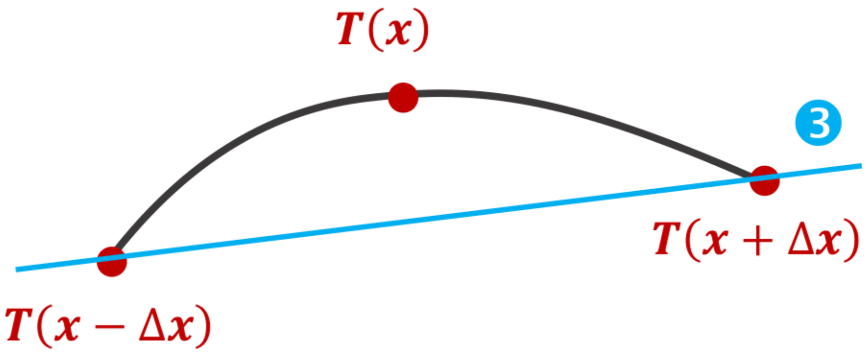
$$\frac{\partial T}{\partial x} = \frac{T(x+\Delta x) - T(x-\Delta x)}{2\Delta x} + \text{Error}$$

$$T(x+\Delta x) = T(x) + \frac{T'(x)}{1!}\Delta x + \frac{T''(x)}{2!}\Delta x^2 + \frac{T'''(x)}{3!}\Delta x^3 + \cdots$$

$$T(x-\Delta x) = T(x) - \frac{T'(x)}{1!}\Delta x + \frac{T''(x)}{2!}\Delta x^2 - \frac{T'''(x)}{3!}\Delta x^3 + \cdots$$

$$\text{Error} = \frac{\Delta x^2 \, T'''(x)}{3!}$$

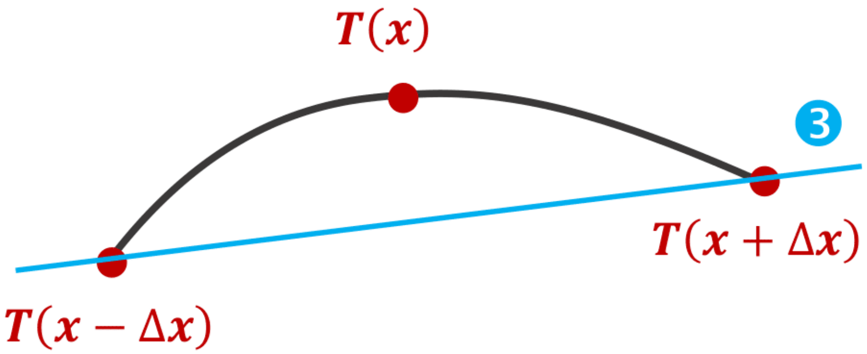
2nd order error



Consistency of a numerical scheme (5/5)

➥ With the centered scheme, the first-order derivative is better resolved than with the first order schemes.

⇨ The centered scheme is better than upstream and downstream schemes, because the **truncation error** is smaller. To improve it, you can increase your resolution (Δx ↘) or use higher-order schemes.



Stability of a numerical scheme (1/14)

Most important characteristic of a **numerical scheme**:

✓ **Consistence :** condition in space ✓

To improve the truncation error:

High order scheme

Increase the resolution (Δx smaller)

✓ **Stability** : condition in time

Does the error amplify during time?

if yes → numerical explosion / Blow Up

if no → stability

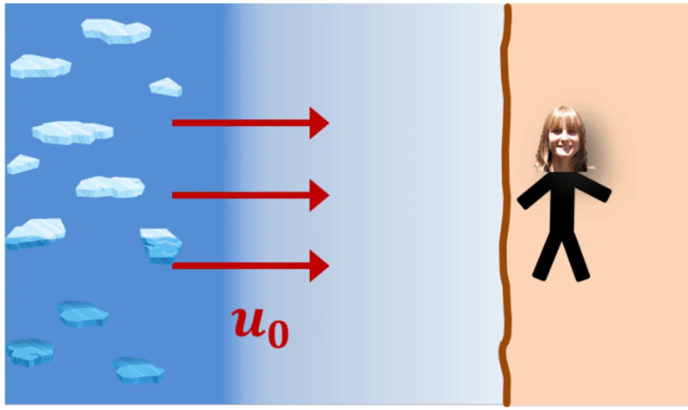
→ **Consistence + Stability → Convergence** of the discretized solution toward the **real solution**, $\forall$ **t** (Lax Theorem)

Stability of a numerical scheme (2/14)

❶ We will test the stability of a **downstream** scheme for both: $\frac{\partial T}{\partial t}$ and $\frac{\partial T}{\partial x}$, such that:

$$\frac{\partial T}{\partial t} \approx \frac{T(t+\Delta t) - T(t)}{\Delta t} = \frac{T_i^{n+1} - T_i^n}{\Delta t}$$

$$\frac{\partial T}{\partial x} \approx \frac{T(x+\Delta x) - T(x)}{\Delta x} = \frac{T_{i+1}^{n} - T_i^n}{\Delta x}$$



→ We inject this formulation into the 1D-advection equation. This leads to:

$$\frac{\partial T}{\partial t} + u_0 \frac{\partial T}{\partial x} = 0$$

$$\rightarrow \frac{T_i^{n+1} - T_i^n}{\Delta t} + u_0 \frac{T_{i+1}^n - T_i^n}{\Delta x} = 0$$

$$\rightarrow T_i^{n+1} = T_i^n - \frac{\Delta t}{\Delta x} u_0 \left(T_{i+1}^n - T_i^n\right)$$

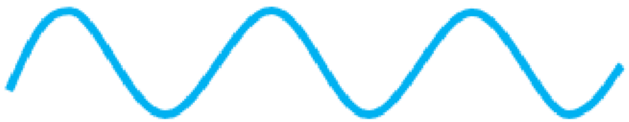
➥ This gives T at time $t + \Delta t$ as a function of T at time t. This is an **explicit method**. It is easy to solve

Stability of a numerical scheme (3/14)

→ We will perform a **von Neumann** stability analysis of our explicit solution.

➥ For this we use wave-like structure for $T(x)$ using complex form: $T_n = \hat{T}_n e^{ikx}$

- e^{ikx} is a wavy pattern that repeats indefinitely (k provides information about its zonal extension).
- $\hat{T}_n$ is the amplitude of the wavy pattern



→ We rewrite our explicit solution using this new notation.

$$T_i^{n+1} = T_i^n - \frac{\Delta t}{\Delta x} u_0 \left(T_{i+1}^n - T_i^n\right)$$

Von Neumann Stability

→ C (Courant Number)

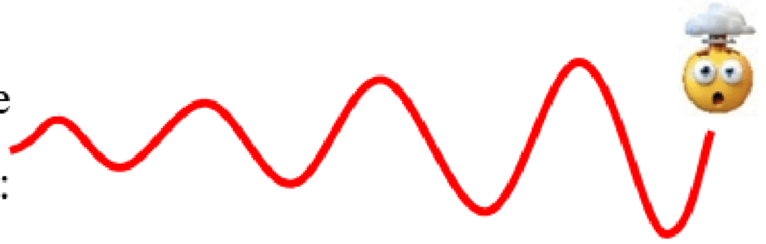
$$\hat{T}^{n+1} e^{ikx} = \hat{T}^n e^{ikx} - \frac{u_0 \Delta T}{\Delta x} \hat{T}^n \left(e^{ik(x+\Delta x)} - e^{ikx}\right)$$

Stability of a numerical scheme (4/14)

$$\hat{T}^{n+1} e^{ikx} = \hat{T}^{n} e^{ikx} - C \, \hat{T}^{n} \left(e^{i(kx+\Delta x)} - e^{ikx}\right)$$

→ We now define the amplification A, such that: $A = \dfrac{\hat{T}_{n+1}}{\hat{T}_n}$

➥ We want $A < 1$, because we do not want the amplitude of the oscillation to increase over time, otherwise the solution would explode:



$$\hat{T}_{n+1} = A\,\hat{T}_n = A^2\,\hat{T}_{n-1} = \cdots = A^n\,\hat{T}_0$$

$$A = \frac{\hat{T}_{n+1}}{\hat{T}_n} = 1 - C\left(e^{ik\Delta x} - 1\right) = 1 - C(\cos(k\Delta x) - i\,sin(k\Delta x) - 1) = 1 + C(1 - \cos(k\Delta x)) - i\,C\,sin(k\Delta x)$$

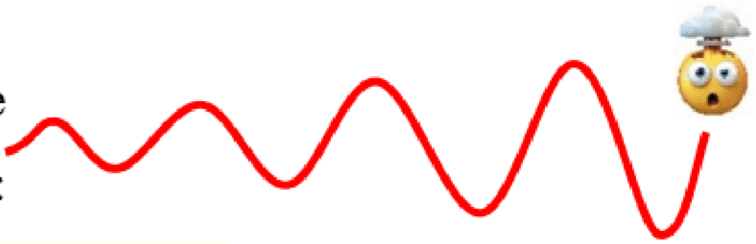
real part *imaginary part*

$\|A\|^2 =$ *real part*^{2} + *imaginary part*^{2}

$\|A\|^2 =$

Stability of a numerical scheme (4/14)

➥ We want $A < 1$, because we do not want the amplitude of the oscillation to increase over time, otherwise the solution would explode:



$$A = 1 + C(1 - \cos(k\Delta x)) - i\, C\, sin(k\Delta x)$$

$$||A||^2 = \textit{real part}^2 + \textit{imaginary part}^2$$

$$||A||^2 = [1 + C(1 - \cos(k\Delta x))]^2 + [C\, sin(k\Delta x)]^2$$

$$||A||^2 = 1 + C^2(1 - \cos(k\Delta x))^2 + 2C(1 - \cos(k\Delta x)) + C^2\, sin^2(k\Delta x)$$

$$||A||^2 = 1 + C^2(1 + \boxed{\cos^2(k\Delta x)} - 2\cos(k\Delta x)) + 2C(1 - \cos(k\Delta x)) + C^2\boxed{sin^2(k\Delta x)}$$

$$||A||^2 = 1 + C^2(2 - 2\cos(k\Delta x)) + 2C(1 - \cos(k\Delta x))$$

$$||A||^2 = 1 + 2C^2(1 - \cos(k\Delta x)) + 2C(1 - \cos(k\Delta x))$$

$$||A||^2 = 1 + (1 - \cos(k\Delta x)) \times 2C \times (1 + C)$$

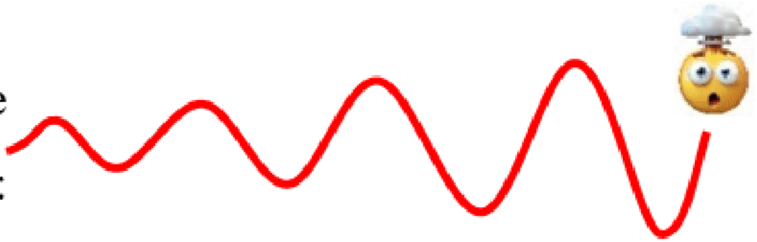
Stability of a numerical scheme (5/14)

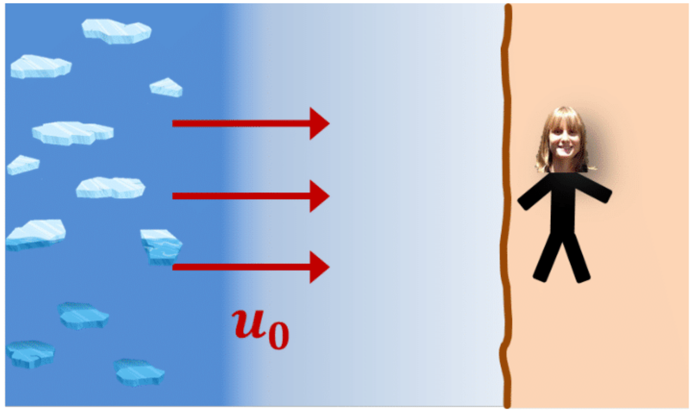
❶ We will test the stability of a **downstream** scheme for both: $\frac{\partial T}{\partial t}$ and $\frac{\partial T}{\partial x}$, such that:

$$\frac{\partial T}{\partial t} \approx \frac{T(t+\Delta t) - T(t)}{\Delta t} = \frac{T_i^{n+1} - T_i^n}{\Delta t} \qquad \frac{\partial T}{\partial x} \approx \frac{T(x+\Delta x) - T(x)}{\Delta x} = \frac{T_{i+1}^n - T_i^n}{\Delta x}$$

→ We now define the amplification A, such that: $A = \frac{\hat{T}_{n+1}}{\hat{T}_n}$

➥ We want $A < 1$, because we do not want the amplitude of the oscillation to increase over time, otherwise the solution would explode:





$$||A||^2 = 1 + (1 - \cos(k\Delta x)) \times 2C \times (1 + C)$$

> 0 $\quad >0$ $\quad >0$

$||A||^2 > 1 \implies$ **Inconditionnaly unstable scheme**

```
=========================================
=                                       =
=   STEP2D:   ABNORMAL JOB END          =
=                BLOW UP                =
=                                       =
=========================================
```

Stability of a numerical scheme (6/14)

❷ We will use the downstream scheme in space, and the upstream scheme in time. This is the upwind scheme:

$$\frac{\partial T}{\partial t} \approx \frac{T(t+\Delta t) - T(t)}{\Delta t} = \frac{T_i^{n+1} - T_i^n}{\Delta t} \qquad \frac{\partial T}{\partial x} \approx \frac{T(x) - T(x-\Delta x)}{\Delta x} = \frac{T_i^n - T_{i-1}^n}{\Delta x}$$

→ We inject this formulation into the 1D-advection equation. This leads to:

$$\frac{\partial T}{\partial t} + u_0 \frac{\partial T}{\partial x} = 0 \qquad \rightarrow \frac{T_i^{n+1} - T_i^n}{\Delta t} + u_0 \frac{T_i^n - T_{i-1}^n}{\Delta x} = 0$$

$$\rightarrow T_i^{n+1} = T_i^n - \frac{\Delta t}{\Delta x} u_0 \left(T_i^n - T_{i-1}^n\right)$$

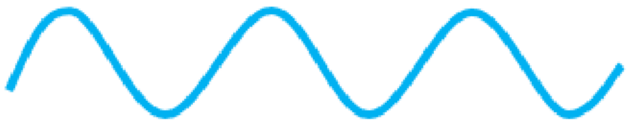
➥ This gives T at time $t + \Delta t$ as a function of T at time t. This is an **explicit method**. It is easy to solve

Stability of a numerical scheme (7/14)

→ We will perform a **von Neumann** stability analysis of our explicit solution.

➥ For this we use wave-like structure for $T(x)$ using complex form: $T_n = \hat{T}_n e^{ikx}$

- e^{ikx} is a wavy pattern that repeats indefinitely (k provides information about its zonal extension).
- $\hat{T}_n$ is the amplitude of the wavy pattern



→ We rewrite our explicit solution using this new notation.

$$T_i^{n+1} = T_i^n - \frac{\Delta t}{\Delta x} u_0 \left(T_i^n - T_{i-1}^n\right)$$

Von Neumann Stability

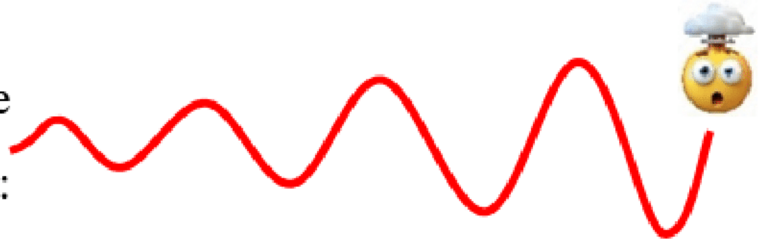
C (Courant Number)

$$\hat{T}^{n+1} e^{ikx} = \hat{T}^n e^{ikx} - \frac{u_0 \Delta T}{\Delta x} \hat{T}^n \left(e^{ikx} - e^{ik(x-\Delta x)}\right)$$

Stability of a numerical scheme (8/14)

$$\hat{T}^{n+1} e^{ikx} = \hat{T}^{n} e^{ikx} - C \left(e^{ikx} - e^{i(kx-\Delta x)} \right)$$

➥ We want $A < 1$, because we do not want the amplitude of the oscillation to increase over time, otherwise the solution would explode:



$$\hat{T}_{n+1} = A\,\hat{T}_n = A^2\,\hat{T}_{n-1} = \cdots = A^n\,\hat{T}_0$$

$$A = \frac{\hat{T}_{n+1}}{\hat{T}_n} = 1 - C\left(1 - e^{-ik\Delta x}\right) = 1 - C\left(1 - \left(\cos(k\Delta x) - i\,sin(k\Delta x)\right)\right) = 1 - C(1 - \cos(k\Delta x)) - iC\,sin(k\Delta x)$$

real part *imaginary part*

$\|A\|^2 =$ *real part*^{2} + *imaginary part*^{2}

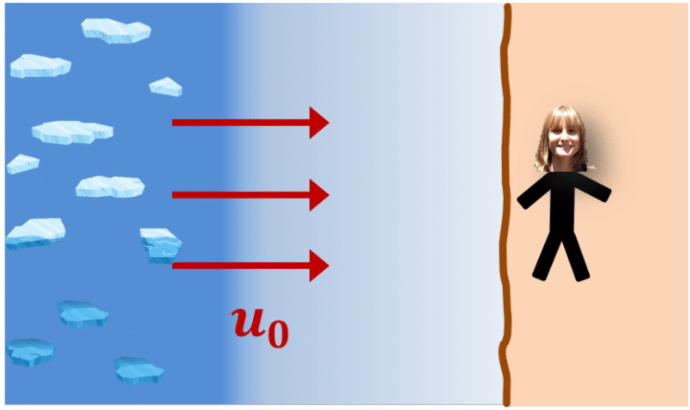
$\|A\|^2 =$

Stability of a numerical scheme (9/14)

In the case of the "Upwind" scheme? $$\frac{T_i^{n+1} - T_i^n}{\Delta t} + u_0 \frac{T_i^n - T_{i-1}^n}{\Delta x} = 0$$

Amplification: $|A| = 1 + 2C(1 - C)(\cos k\delta x - 1)$

$|A|<1$ if $C<1$ → conditionnaly stable if CFL<1



Courant-Friedrichs-Lewy (CFL) stability criterion :

$$C = \frac{u_0 \Delta t}{\Delta x} \leq 1$$

But numerical attenuation /diffusion ....

Stability of a numerical scheme (10/14)

- **Leapfrog / Centered**

$T_i^{n+1} = T_i^{n-1} - C\,(T_{i+1}^{n} - T_{i-1}^{n})$; $C = u_0\,dt / dx$

Conditionally stable: CFL condition C < 1
but **dispersive** (computational modes)

1D Advection equation:

$$\frac{\partial T}{\partial t} + u_0 \frac{\partial T}{\partial x} = 0$$

should be non-dispersive : the phase speed *ω/k* and group speed δω/δk are equal and constant (u_o)

- **Downstream (Euler) / Centered**

$T_i^{n+1} = T_i^{n} - C\,(T_{i+1}^{n} - T_{i-1}^{n})$

Unconditionally unstable

- **Upstream**

$T_i^{n+1} = T_i^{n} - C\,(T_i^{n} - T_{i-1}^{n})$, $C > 0$

$T_i^{n+1} = T_i^{n} - C\,(T_{i+1}^{n} - T_i^{n})$, $C < 0$

Conditionally stable,
not dispersive but **diffusive**

(monotone linear scheme)

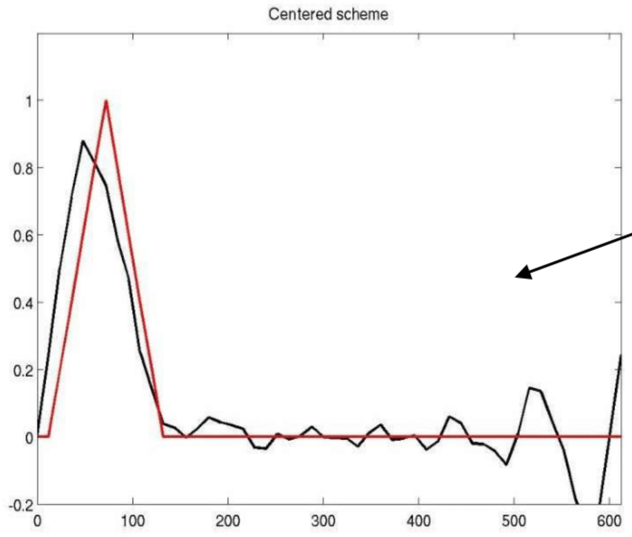
2nd order approx to the modified equation:

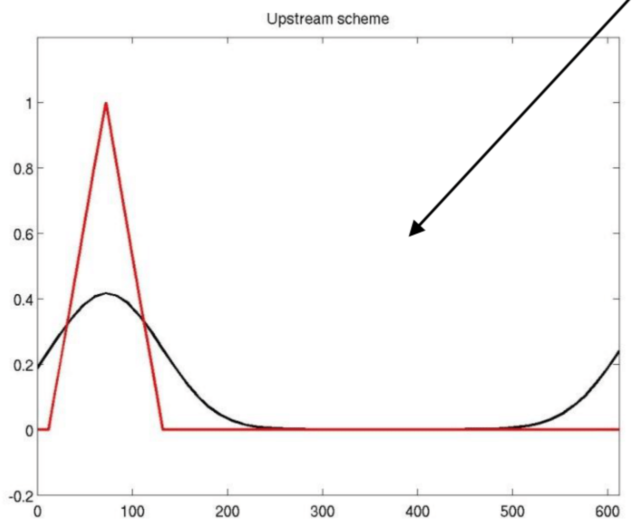
$$\partial_t\theta + c\partial_x\theta - \frac{c\Delta x}{2}\left(1 - \frac{c\Delta t}{\Delta x}\right)\partial_{xx}\theta = 0.$$

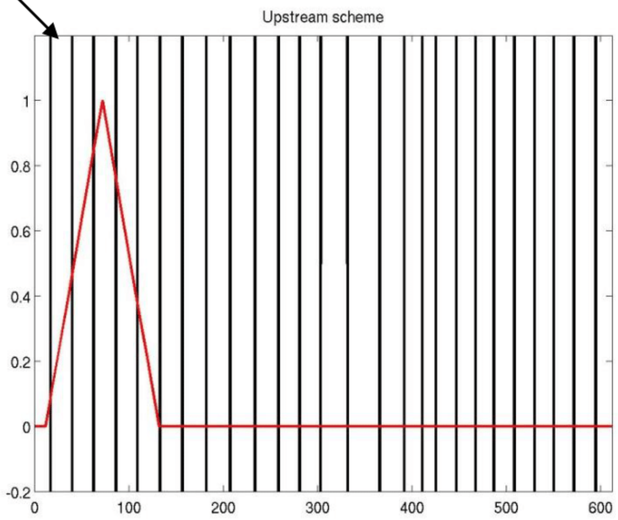
Stability of a numerical scheme (11/14)

A numerical scheme can be:

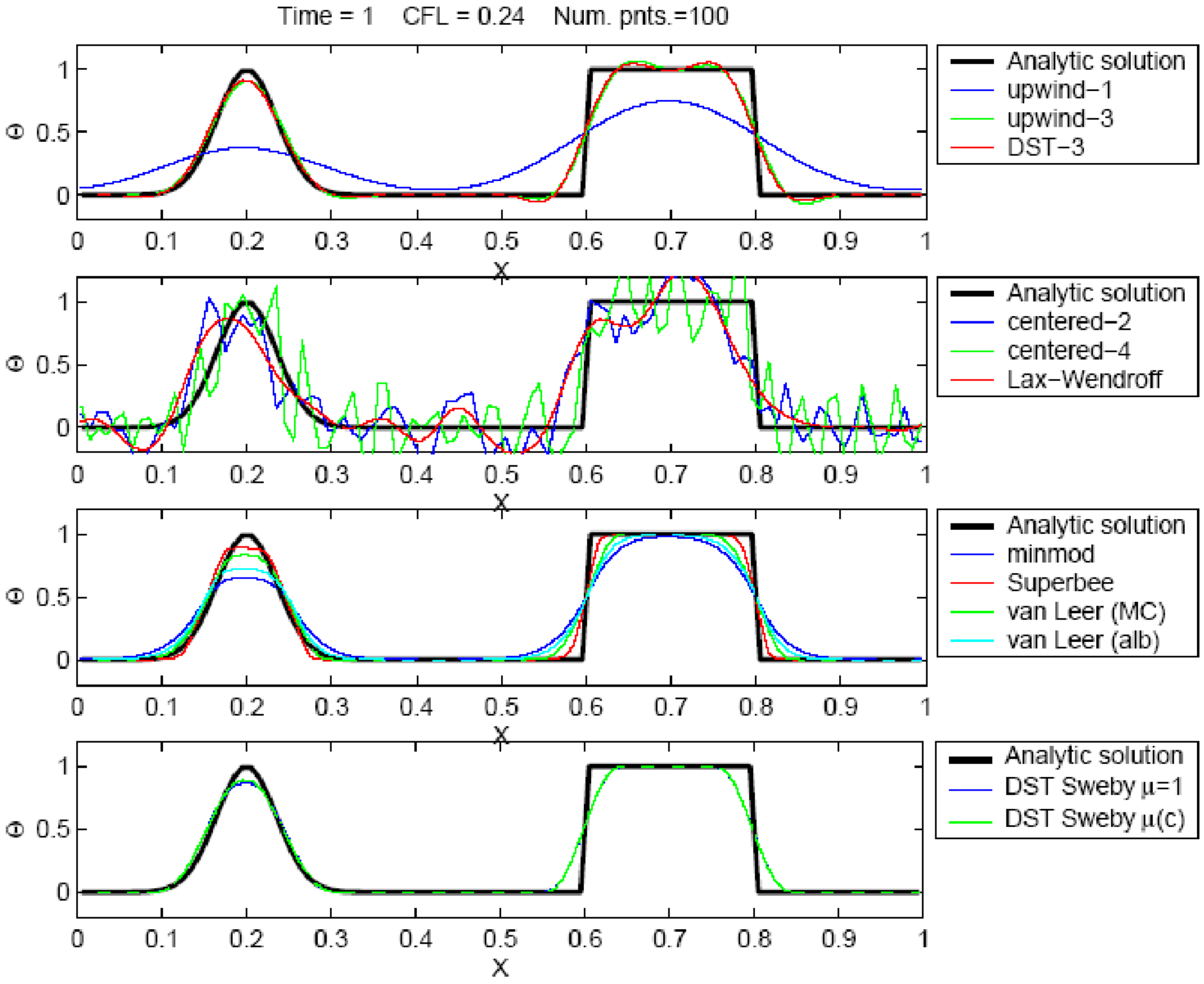
- **Dispersive**: ripples, overshoot and extrema (centered)
- **Diffusive** (upstream)
- **Unstable** (Euler/centered)







Stability of a numerical scheme (12/14)



Stability of a numerical scheme (13/14)

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

Stability of a numerical scheme (14/14)

Most important characteristic of a **numerical scheme**:

✓ **Consistence :** condition in space ✓

To improve the truncation error:

High order scheme

Increase the resolution (Δx smaller)

✓ **Stability** : condition in time

Does the error amplify during time?

if yes → numerical explosion / Blow Up

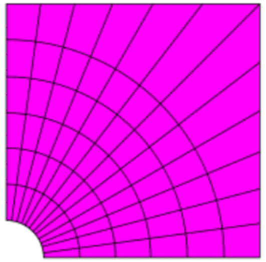
if no → stability

→ **Consistence + Stability → Convergence** of the discretized solution toward the **real solution**, ∀ **t** (Lax Theorem)

Horizontal discretization (1/3)

➢ **Structured grids**

The grid cells have the same number of sides.

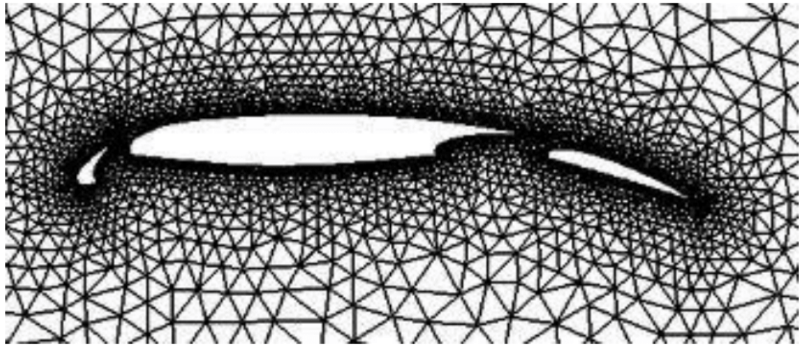


CROCO

➢ **Unstructured grids**

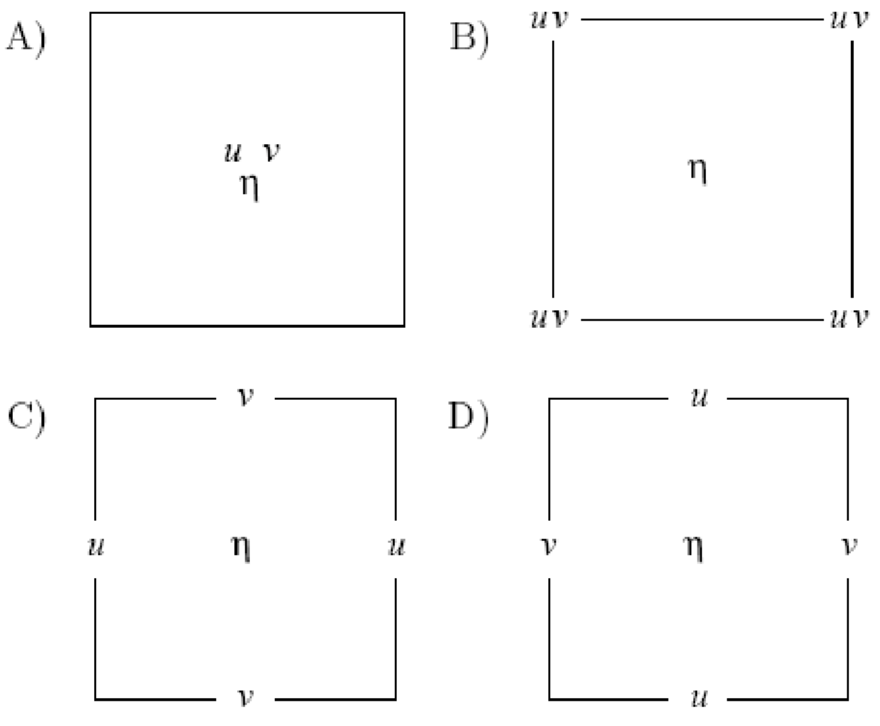
The domain is tiled using more general geometrical shapes (triangles, …) pieced together to optimally fit details of the geometry.

✓Good for tidal modeling, engineering applications.



✓Problems:
geostrophic balance accuracy,
wave scattering by non-uniform grids,
conservation and positivity properties, …

Horizontal discretization (2/3)



- A staggered difference is 4 times more accurate than non-staggered and improves the dispersion relation because of reduced use of averaging operators

Linear shallow water equation:

- A grid:

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

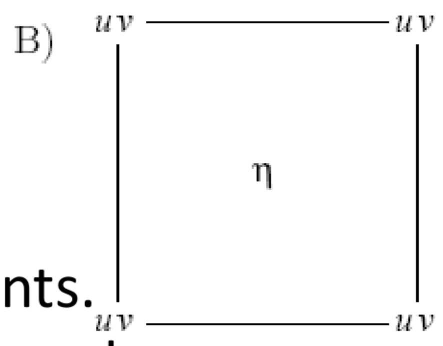
- B grid:

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

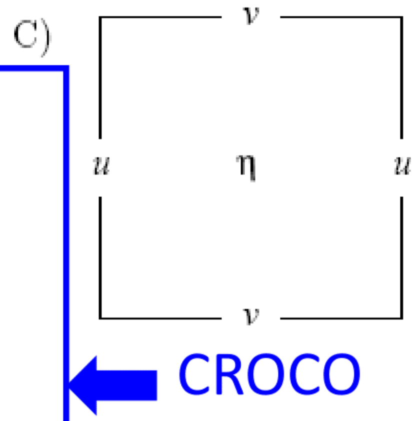
- C grid:

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

Horizontal discretization (3/3)



- 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 checkboard 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 checkboard mode) and inertia-gravity waves due to averaging the Coriolis force.

- Combinations can also be used (A + C)

Pressure Gradient Force (1/6)

The sigma coordinates represent with good accuracy the bottom and the surface layers. BUT the sigma coordinate system is also associated with errors in the estimation or the Pressure Gradient Force.

- In the momentum conservation equations, we find a term associated with the horizontal gradients of the pressure field:

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

➥ This horizontal gradient must be computed at constant z. It can be written:

$$-\frac{1}{\rho_0}\frac{\partial P}{\partial x}\bigg|_z$$

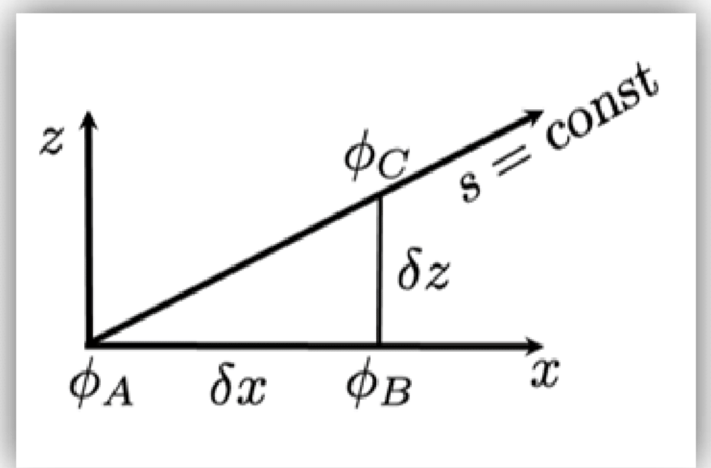
- We want to transform the horizontal derivative of P between z and s coordinates.

Pressure Gradient Force (2/6)

➥ This horizontal gradient must be computed at constant z. It can be written:

$$-\frac{1}{\rho_0}\frac{\partial P}{\partial x}\bigg|_z$$

➢ We want to transform the horizontal derivative of P between z and s coordinates. With a little bit of geometry, we can show that:



$$\frac{\partial \phi}{\partial x}\bigg|_s = \frac{\phi_C - \phi_A}{\delta x} \qquad \delta x,\ \delta z \to 0$$

$$= \frac{\phi_C - \phi_B}{\delta z}\left(\frac{\delta z}{\delta x}\right) + \frac{\phi_B - \phi_A}{\delta x}$$

$$\frac{\partial \phi}{\partial x}\bigg|_s = \frac{\partial \phi}{\partial z}\left(\frac{\partial z}{\partial x}\bigg|_s\right) + \frac{\partial \phi}{\partial x}\bigg|_z$$

$$\frac{\partial \phi}{\partial x}\bigg|_z = \frac{\partial \phi}{\partial x}\bigg|_s - \frac{\partial \phi}{\partial z}\frac{\partial z}{\partial x}\bigg|_s$$

➥ It follows that:

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

Pressure Gradient Force (3/6)

- We obtained:

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

$$-\frac{1}{\rho_0}\frac{\partial P}{\partial x}\bigg|_z = -\frac{1}{\rho_0}\frac{\partial P}{\partial x}\bigg|_s + \frac{1}{\rho_0}\frac{\partial P}{\partial s}\frac{\partial s}{\partial z}\frac{\partial z}{\partial x}\bigg|_s$$

- With $\frac{\partial s}{\partial z} \sim \frac{1}{H}$, the horizontal pressure gradient is written as the difference between 2 terms:

$$-\frac{1}{\rho_0}\frac{\partial P}{\partial x}\bigg|_z = -\frac{1}{\rho_0}\frac{\partial P}{\partial x}\bigg|_s + \frac{1}{H}\frac{1}{\rho_0}\frac{\partial P}{\partial s}\frac{\partial z}{\partial x}\bigg|_s$$

PGF in z coordinate

① PGF along iso-sigma surfaces

② Correction term to eliminate the vertical gradient contained in the first term

- On sigma level can have important differences of depth on a short scale $\frac{\partial z}{\partial x}\Big|_s$

- On steep slopes (sharp topographic changes such as the continental slope), terms ① and ② are **both large**, with comparable amplitude. One small error in their estimation results in important errors in the PGF calculous. This is called the **Truncation error**.

Pressure Gradient Force (4/6)

- To control the amplitude of the truncation error, we need to respect this condition:

$$\varepsilon = \frac{\left| \left.\frac{\partial P}{\partial x}\right|_s - \frac{\partial P}{\partial z}\left.\frac{\partial z}{\partial x}\right|_s \right|}{\left| \left.\frac{\partial P}{\partial x}\right|_s \right| + \left| \frac{\partial P}{\partial z}\left.\frac{\partial z}{\partial x}\right|_s \right|} \ll 1$$

- If the truncation error on the PGF is important, it can result in **artificial "numerical" currents over the slopes.**

↪ To check if there is an error in your configuration, you can run a neutral simulation (no forcing, no currents). If you run the model, you should have no current in the outputs. BUT if the pressure gradient errors are substantial, you will observer geostrophic currents over the slopes.

- To reduce the pressure gradient error…

Pressure Gradient Force (5/6)

- Smoothing the topography using a nonlinear filter and a criterium: → $r = \Delta h / h < 0.2$

- Using a density formulation →

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

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

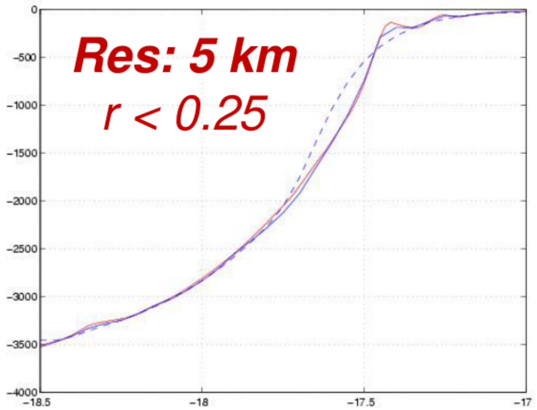
- Using high order schemes to reduce the truncation error (4th order, McCalpin, 1994)

- Gary, 1973: substracting a reference horizontal averaged value from density ($\rho' = \rho - \rho_a$) before computing pressure gradient

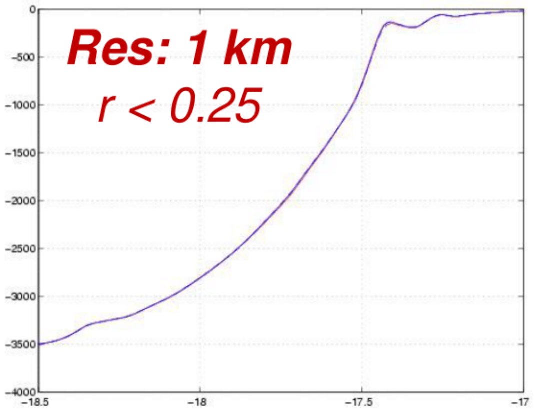
- Rewritting Equation of State: reduce passive compressibility effects on pressure gradient

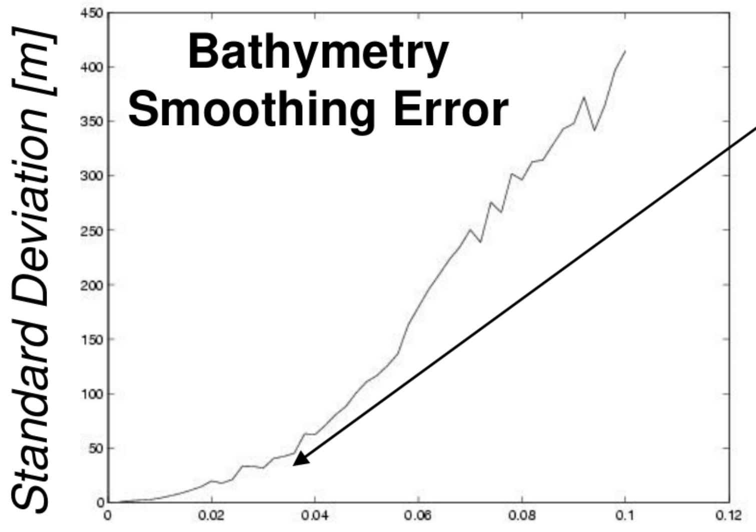
Pressure Gradient Force (6/6)

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



Senegal Bathymetry Profil





Grid Resolution [deg]

Convergence at ~ *4 km* resolution

STEP 5: Visualising model outputs

- Launch Matlab and edit the following file:

```
>> edit croco_diags.m
>> croco_diags
```

- Make your first plots:

```
>> plot_diags
```

- Visualise the outputs with croco_gui

```
>> croco_gui
```

- Enjoy!!!

STEP 6: Exiting

➤ Exit Matlab:

```
exit
```

➤ Give back the compute node:

```
exit
```

➤ Logoff the Lengau cluster:

```
exit
```