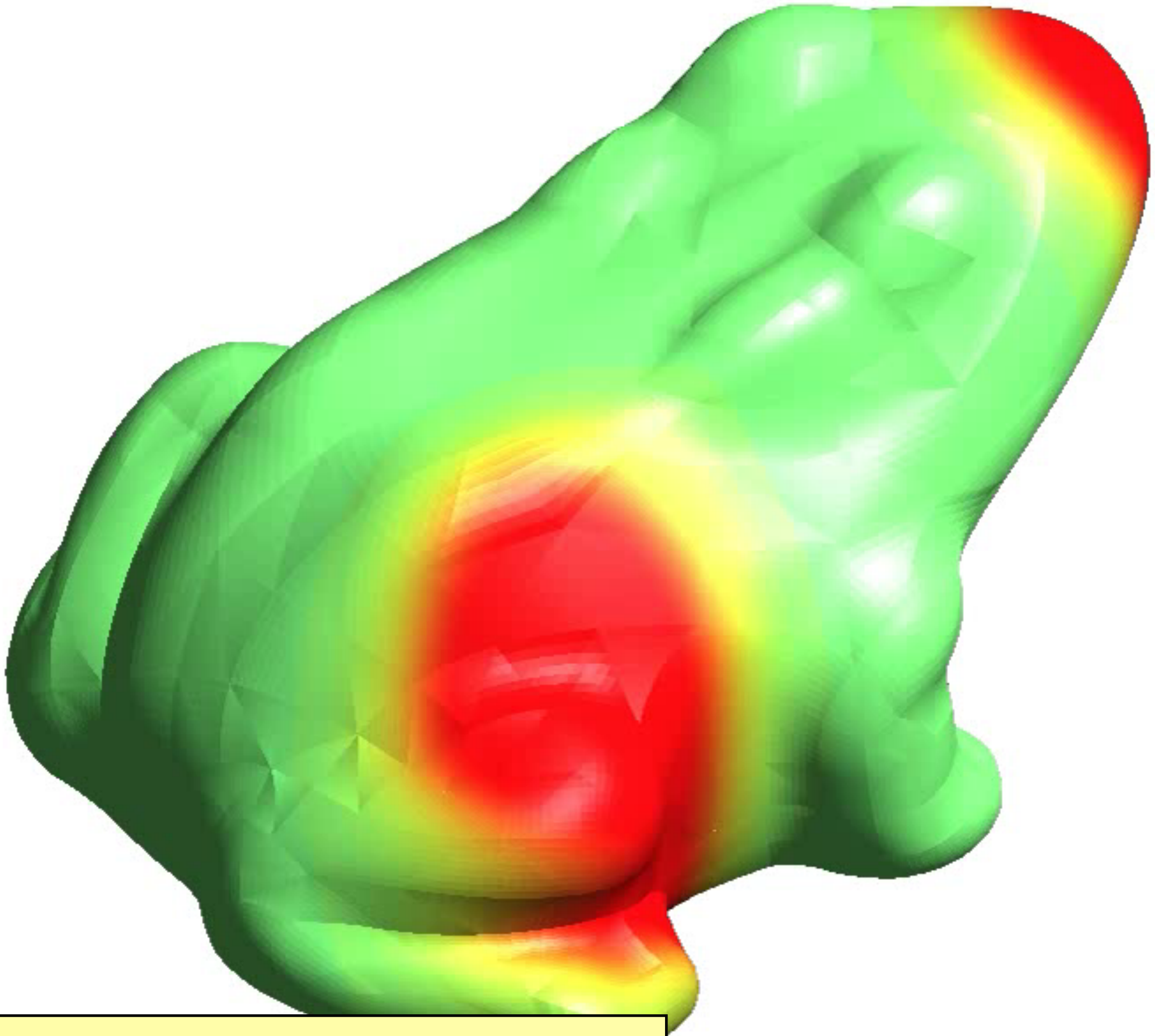
Second generation Louvain-la-neuve Ice-ocean Model

Paul-Emile Bernard, Sébastien Blaise,
Sylvain Bouillon, Richard Comblen, Anouk de Brauwere,
Benjamin de Brye, Thomas De Maet, Eric Deleersnijder,
Thierry Fichefet, Olivier Gourgue, Emmanuel Hanert,
Tuomas Kärnä, Jonathan Lambrechts, Vincent Legat,
Sébastien Legrand, Daniel Le Roux, Olivier Lietaer,
Samuel Melchior, Alice Pestiaux, Jean-François
Remacle, Karim Slaoui, Sébastien Schellen, Bruno Seny,
Christopher Thomas, Laurent White

http://www.climate.be/slim

SLIM

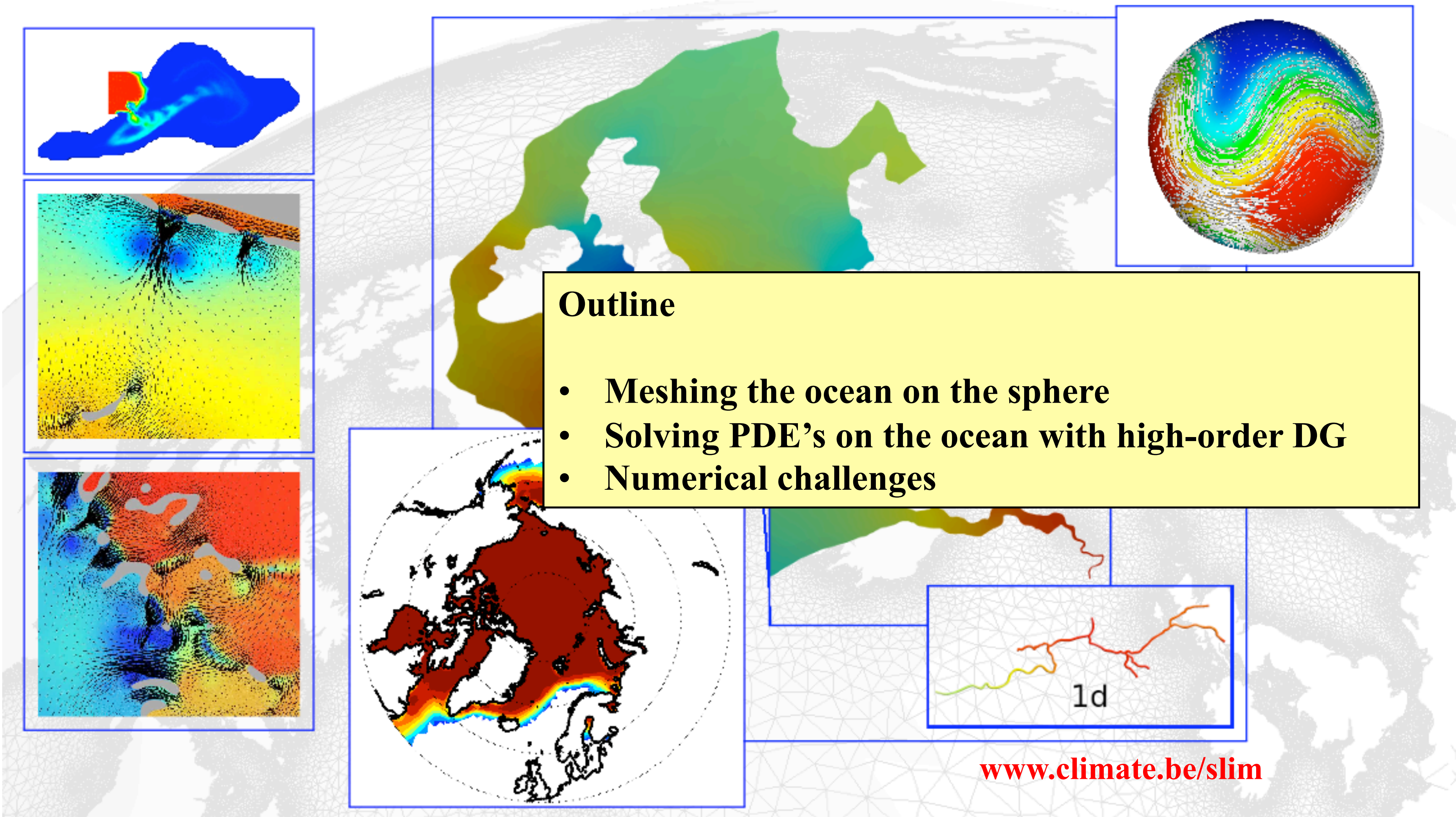
Gravity waves on a froggy planet



Building a general method for irregular manifolds

- **The method is independent of the manifold**
- **It must be easy to implement**
- **It must be robust to handle such a funny benchmark**

Slim : a multi-scale model for the ocean, coaslines and rivers

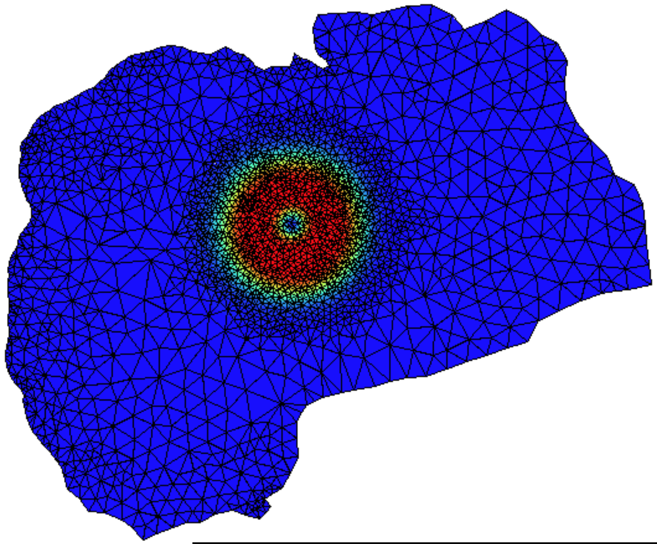


Outline

- **Meshing the ocean on the sphere**
- **Solving PDE's on the ocean with high-order DG**
- **Numerical challenges**

www.climate.be/slim

Are adaptive unstructured-grid models coming of age ?

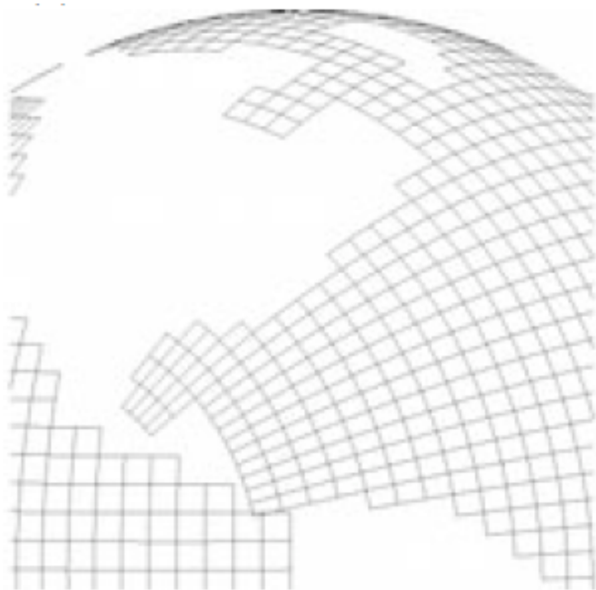


Reduced-gravity simulation of a baroclinic eddy in the Gulf of Mexico.

This simulation is several orders of magnitude cheaper than a constant resolution one of the same accuracy ! (Bernard, 2007)

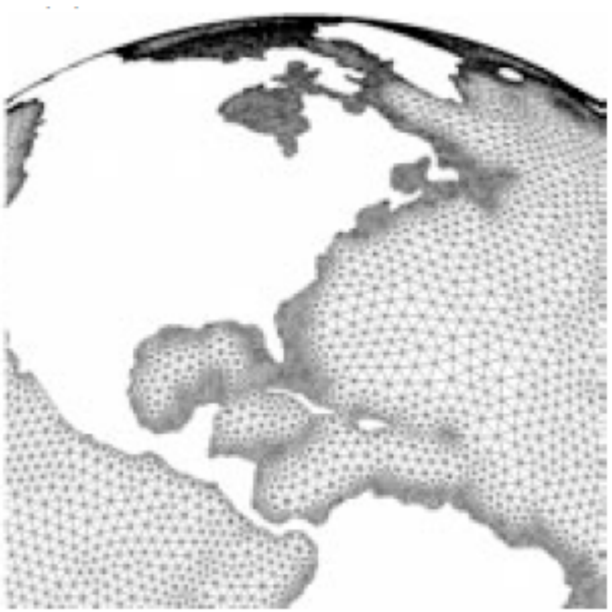
- **Numerical models of marine systems should be able to explicitly represent the broadest possible range of scales.**
- **Increasing the resolution everywhere is not the best option as this often results in a very inefficient use of the computational resources.**
- **The idea is to increase the resolution where and when it is needed !**

Structured grid …



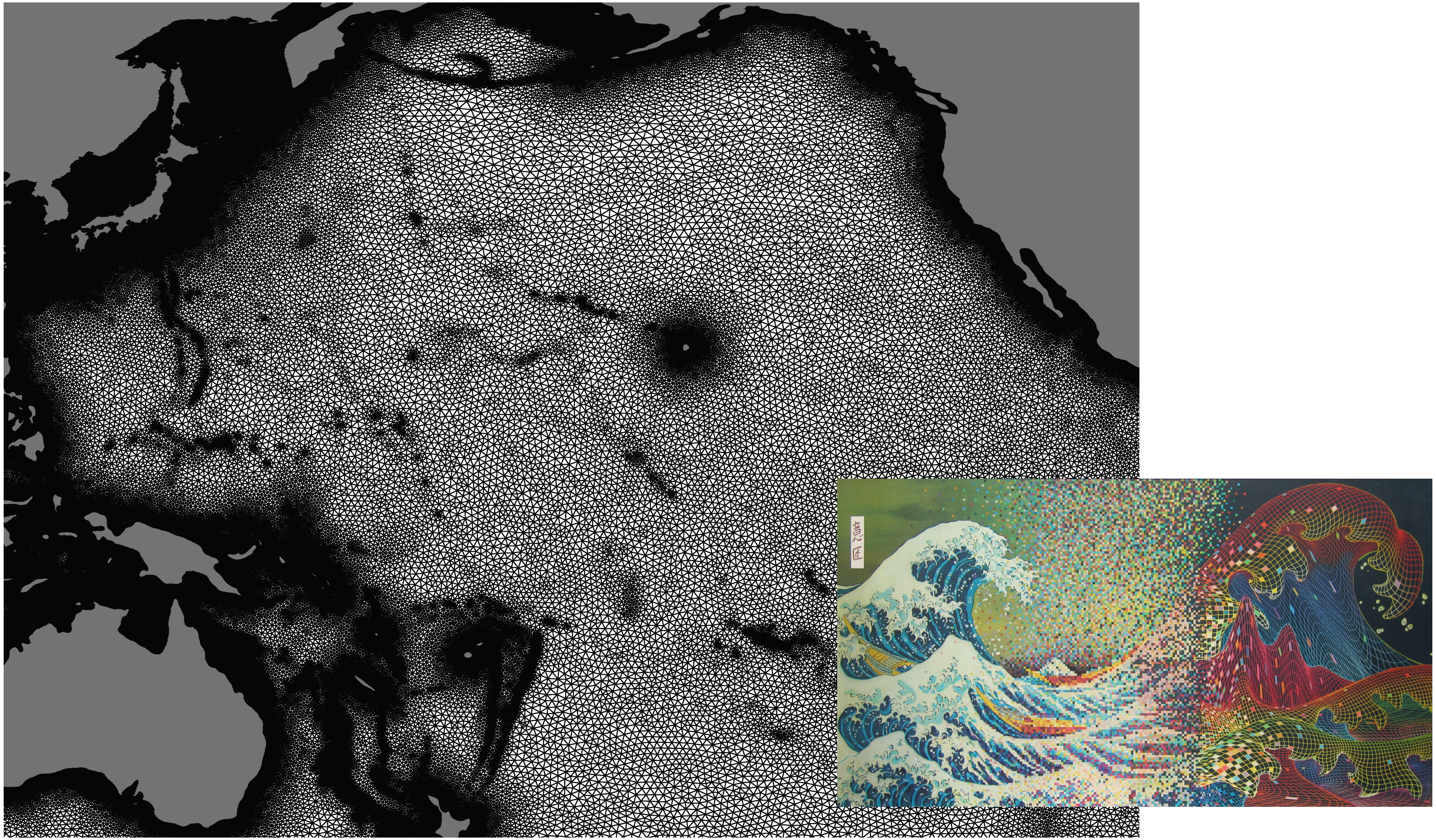
- Finite differences are easy to implement
- Programming is easy
- Well known in the world of oceanography
- Bad representation of the coastlines
- Difficult to enhance locally the resolution
- Poles singularity

…versus unstructured grid



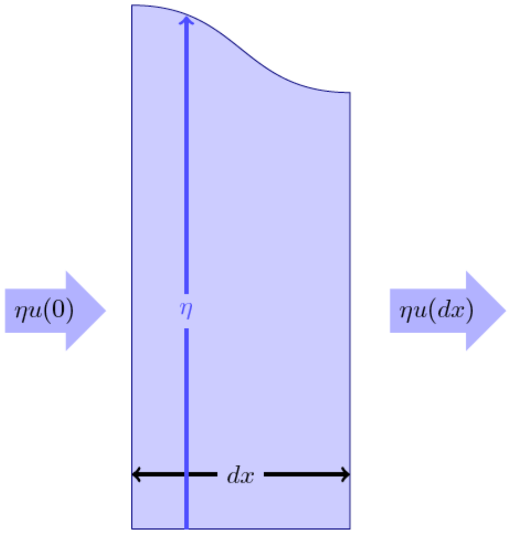
- Numerical methods are more complicated
- Programming is more complicated
- Not well known in the world of oceanography
- Accurate representation of the coastlines
- Enhancing the resolution is flexible
- No singular points

Now, let us simulate the Fukushima's tsunami…



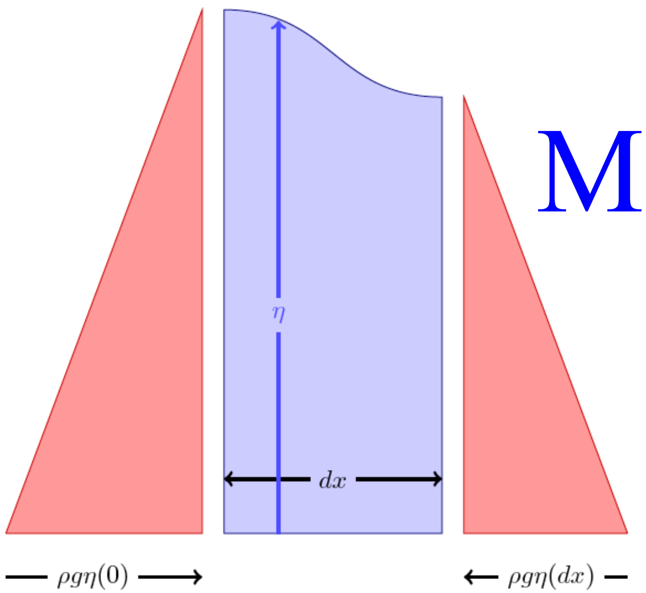
Mass balance

$$\frac{\partial \eta}{\partial t} + \eta_0 \frac{\partial u}{\partial x} = 0$$



$$\begin{aligned}
dx\left(\rho \frac{\partial \eta}{\partial t}\right) &= \rho\eta u(0) - \rho\eta u(dx) \\
\frac{\partial \eta}{\partial t} + \eta_0\left(\frac{u(0) - u(dx)}{dx}\right) &= 0 \\
&\downarrow \\
\frac{\partial \eta}{\partial t} + \eta_0 \frac{\partial u}{\partial x} &= 0
\end{aligned}$$

$$
\begin{aligned}
dx\left(\rho\eta_0\frac{\partial u}{\partial t}\right) &= \rho g\frac{\eta^2(0)}{2} - \rho g\frac{\eta^2(dx)}{2} \\
\eta_0\frac{\partial \eta}{\partial u} + g\eta_0\frac{\partial \eta}{\partial x} &= 0 \\
&\Big\downarrow \\
\frac{\partial u}{\partial t} + g\frac{\partial \eta}{\partial x} &= 0
\end{aligned}
$$



Momentum balance

$$\frac{\partial u}{\partial t} + g\frac{\partial \eta}{\partial x} = 0$$

$$\left\{ \begin{array}{lcl} \dfrac{1}{\eta_0}\dfrac{\partial \eta}{\partial t} + \boxed{\dfrac{\partial u}{\partial x}} & = & 0 \\ \boxed{\dfrac{\partial u}{\partial t}} + g\dfrac{\partial \eta}{\partial x} & = & 0 \end{array} \right.$$

Linear
Shallow Water
Equations

$$\frac{\partial^2 \eta}{\partial t^2} = g\eta_0 \frac{\partial^2 \eta}{\partial x^2}$$

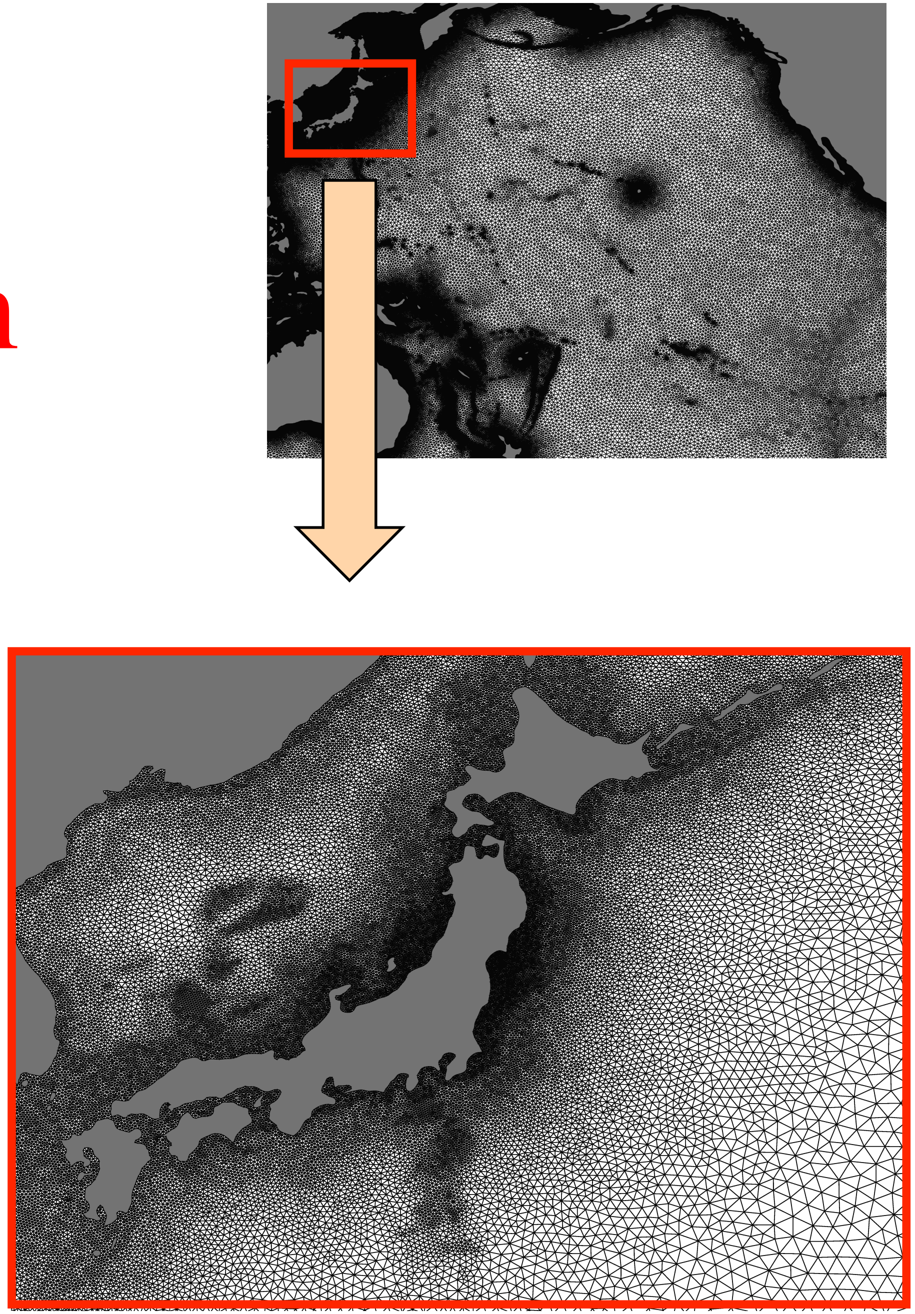
Wave Equation

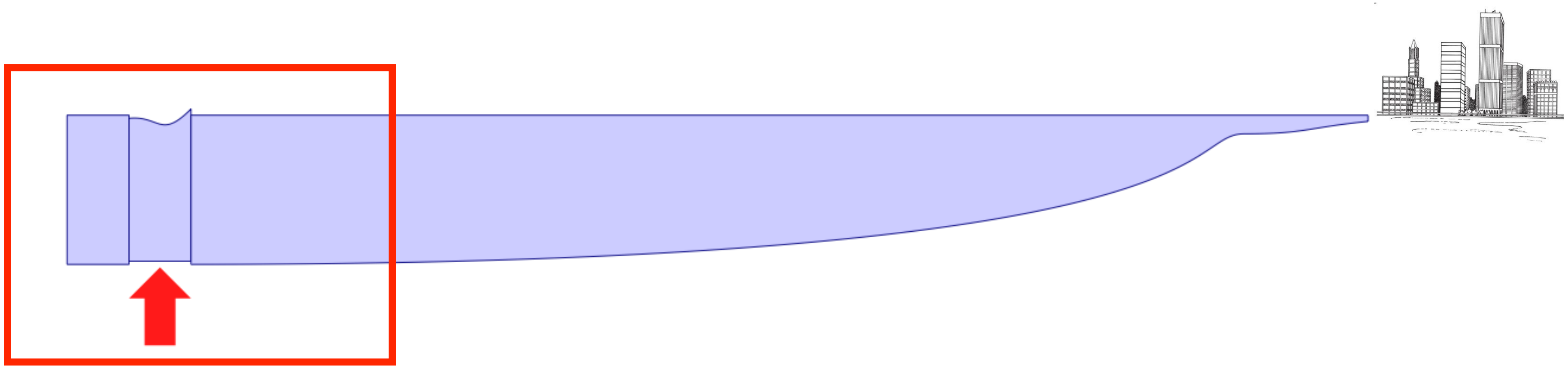
$$c = \sqrt{g\eta_0} \approx 200 \ [m/s]$$

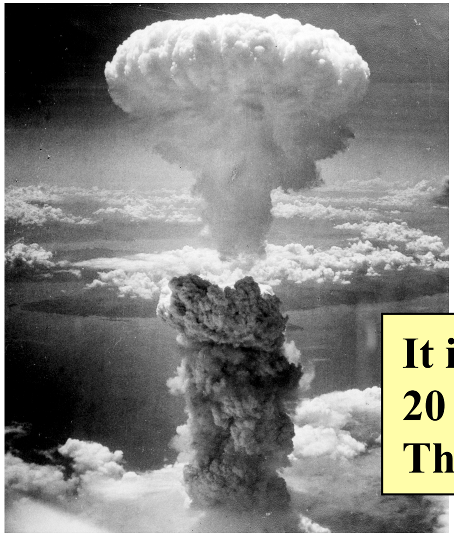
Gravity 9,81 m/s
Average depth of Pacific 4000 m

Waves are (very) fast !

And now,
we can zoom
on Japan !

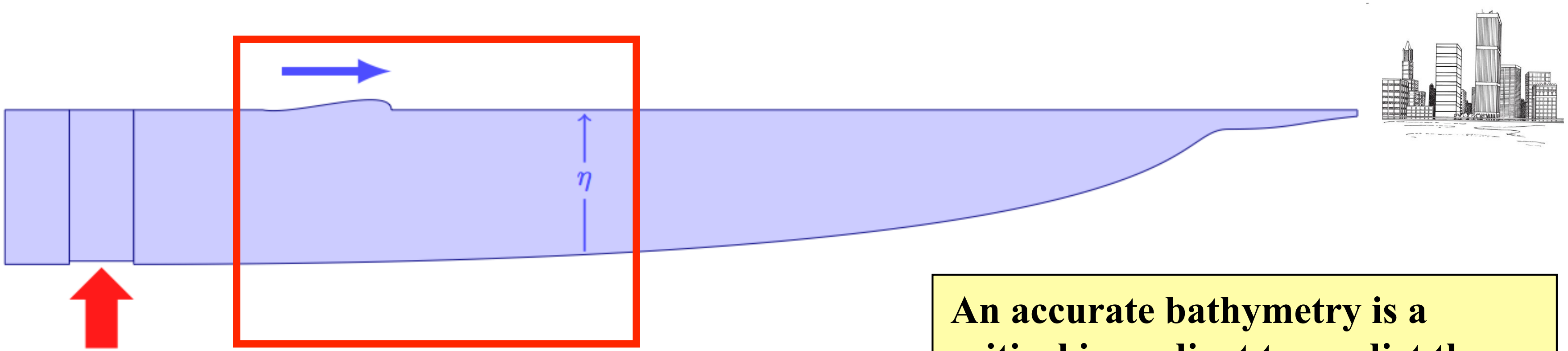




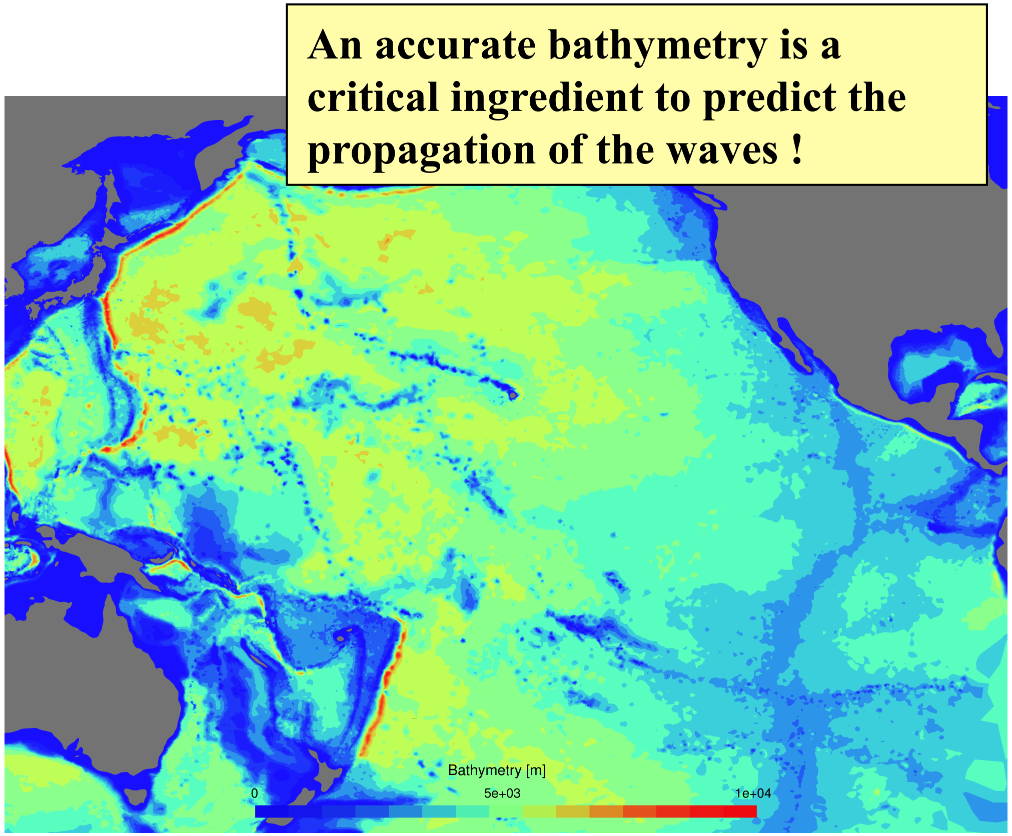


It is a huge energy !
20 x Energy of Hiroshima's bomb !
This initial condition must be provided to the model

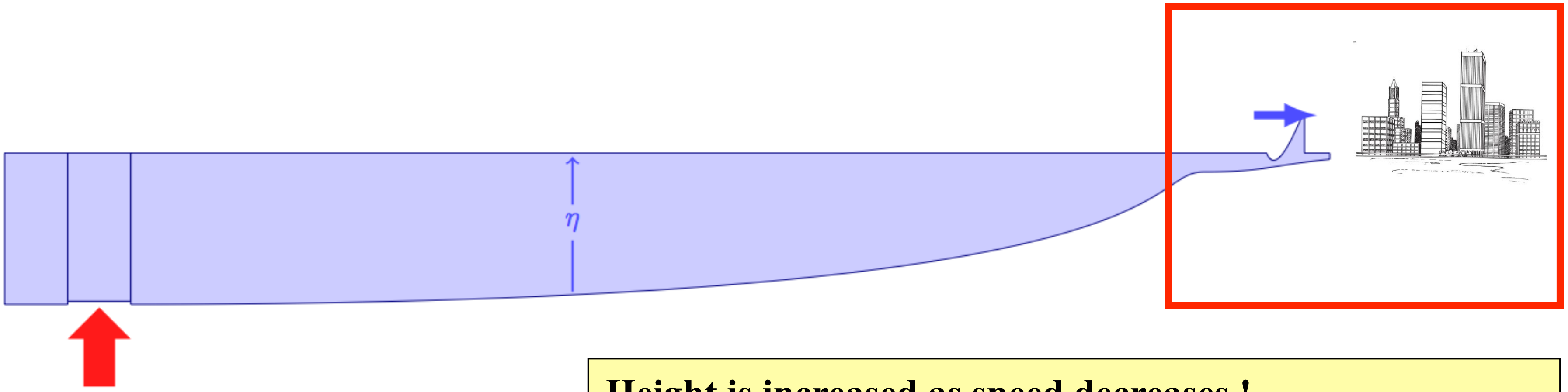
The earthquake motion displaces a column water



$$c = \sqrt{g\eta_0} \approx 200\ [m/s]$$



Small waves
travel fast
as function of the bathymetry

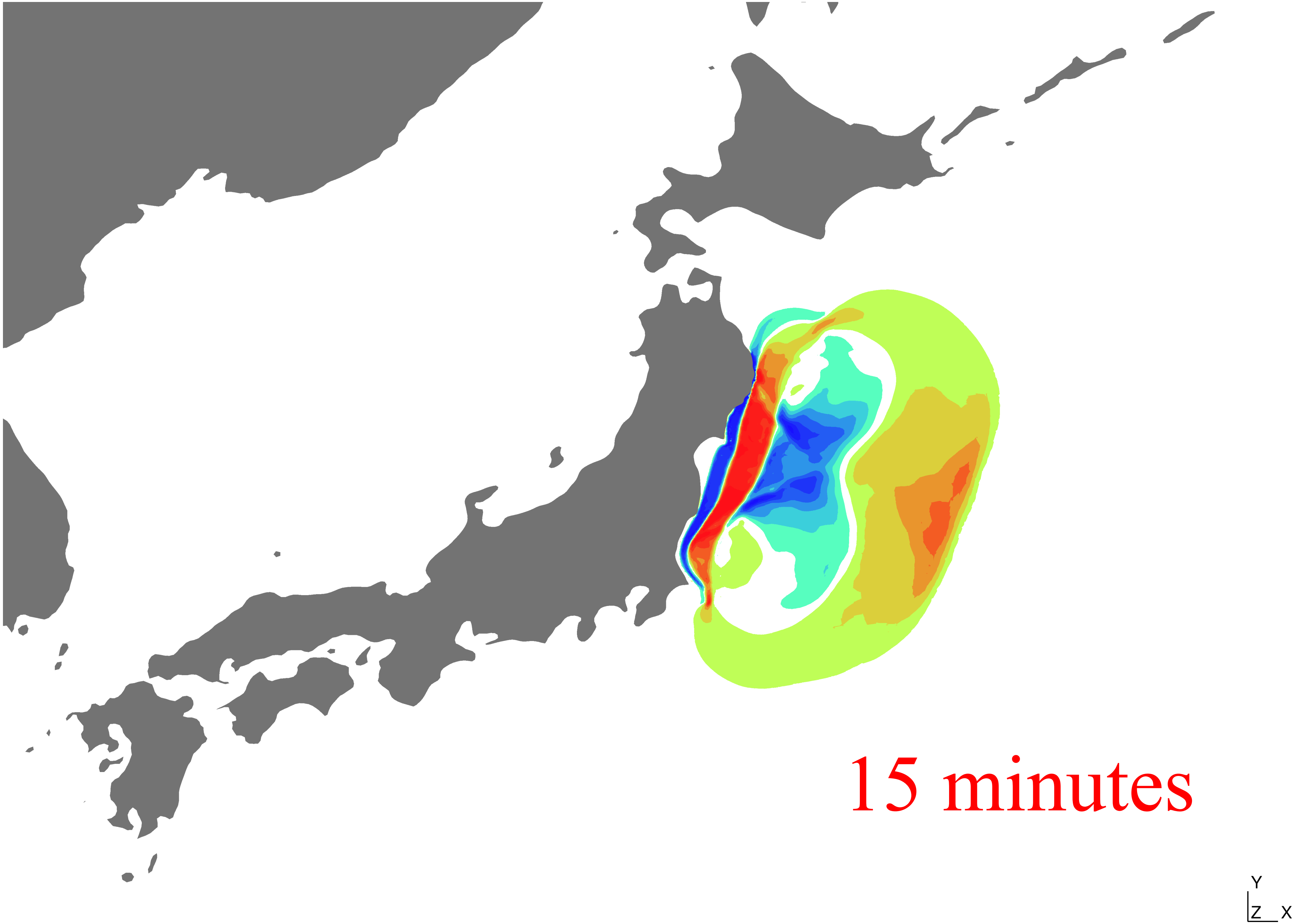


Height is increased as speed decreases !
An accurate shoreline description is required.

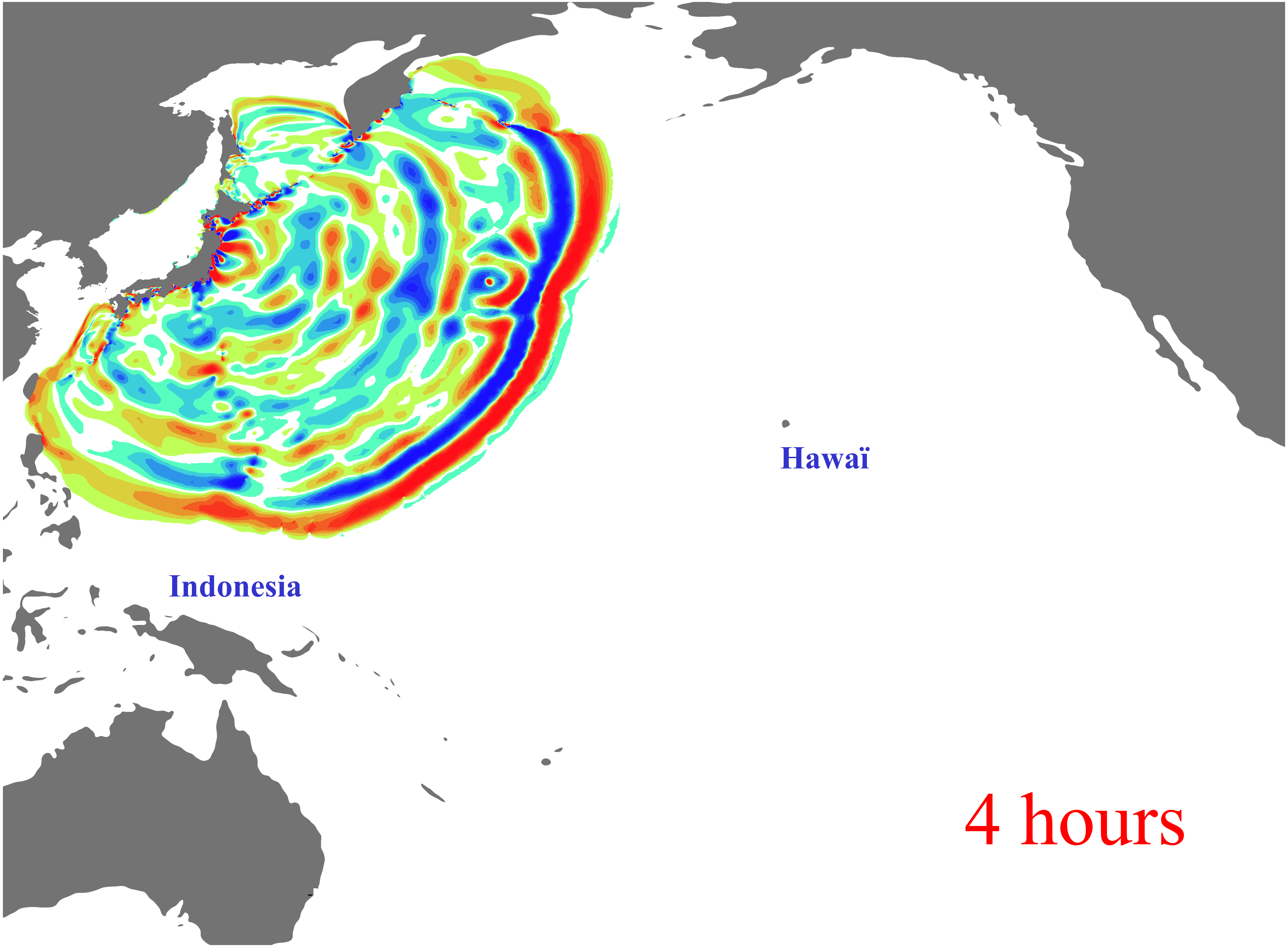
Waves compression forces waves to gain height !

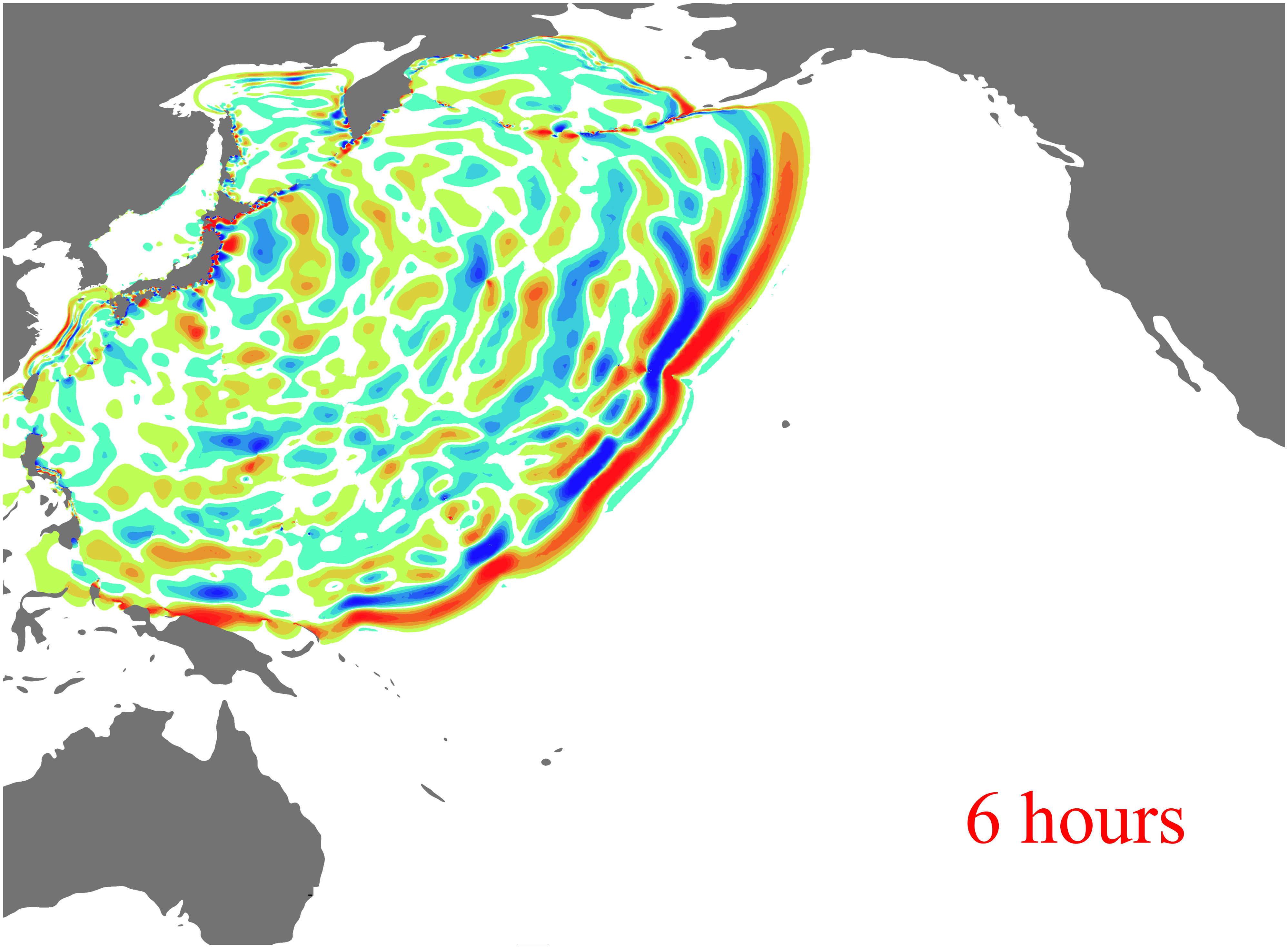
Simulating tsunamis is easy

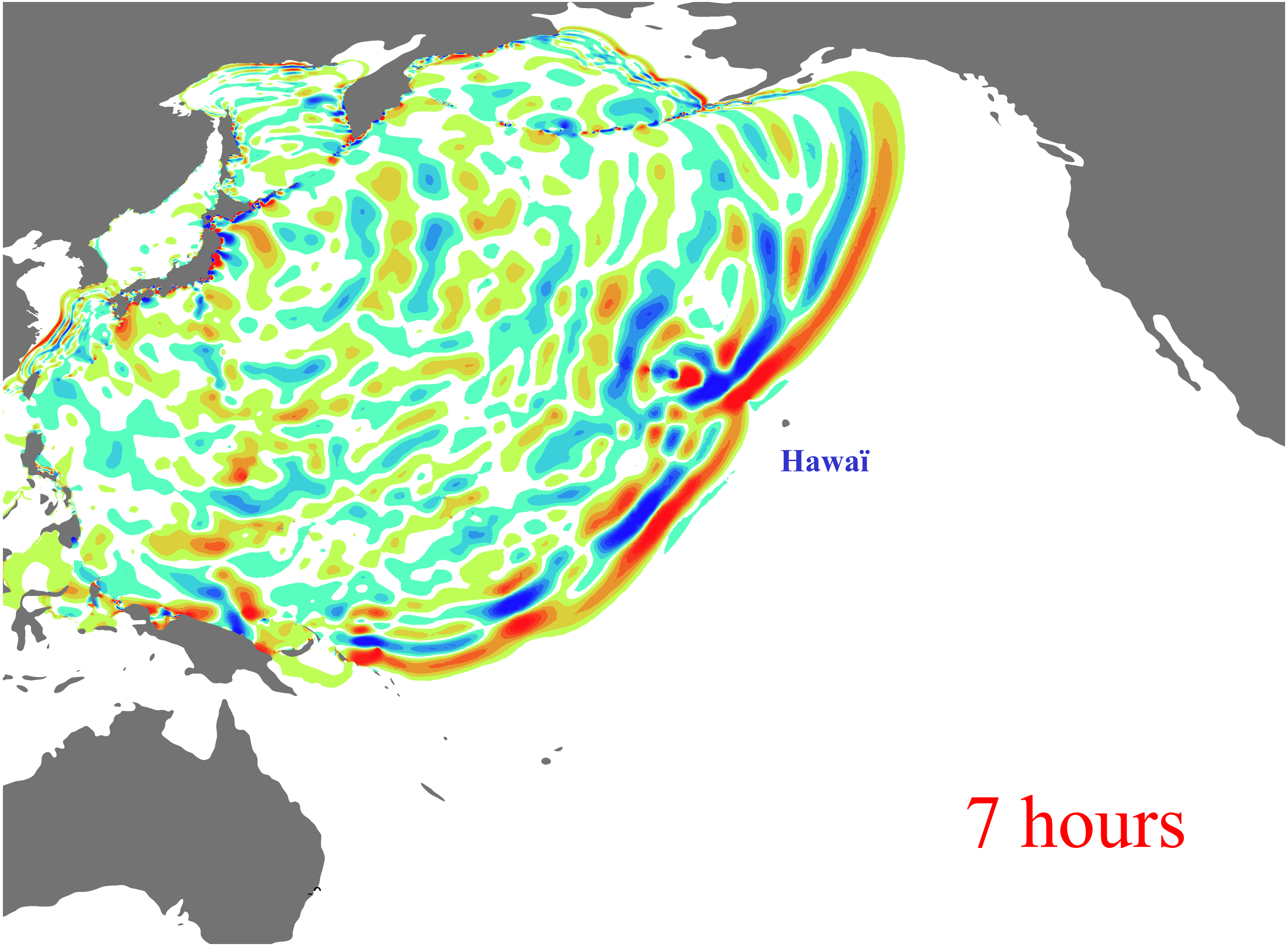
Simulation performed on May 17th by Jonathan Lambrechts and Benjamin de Brye

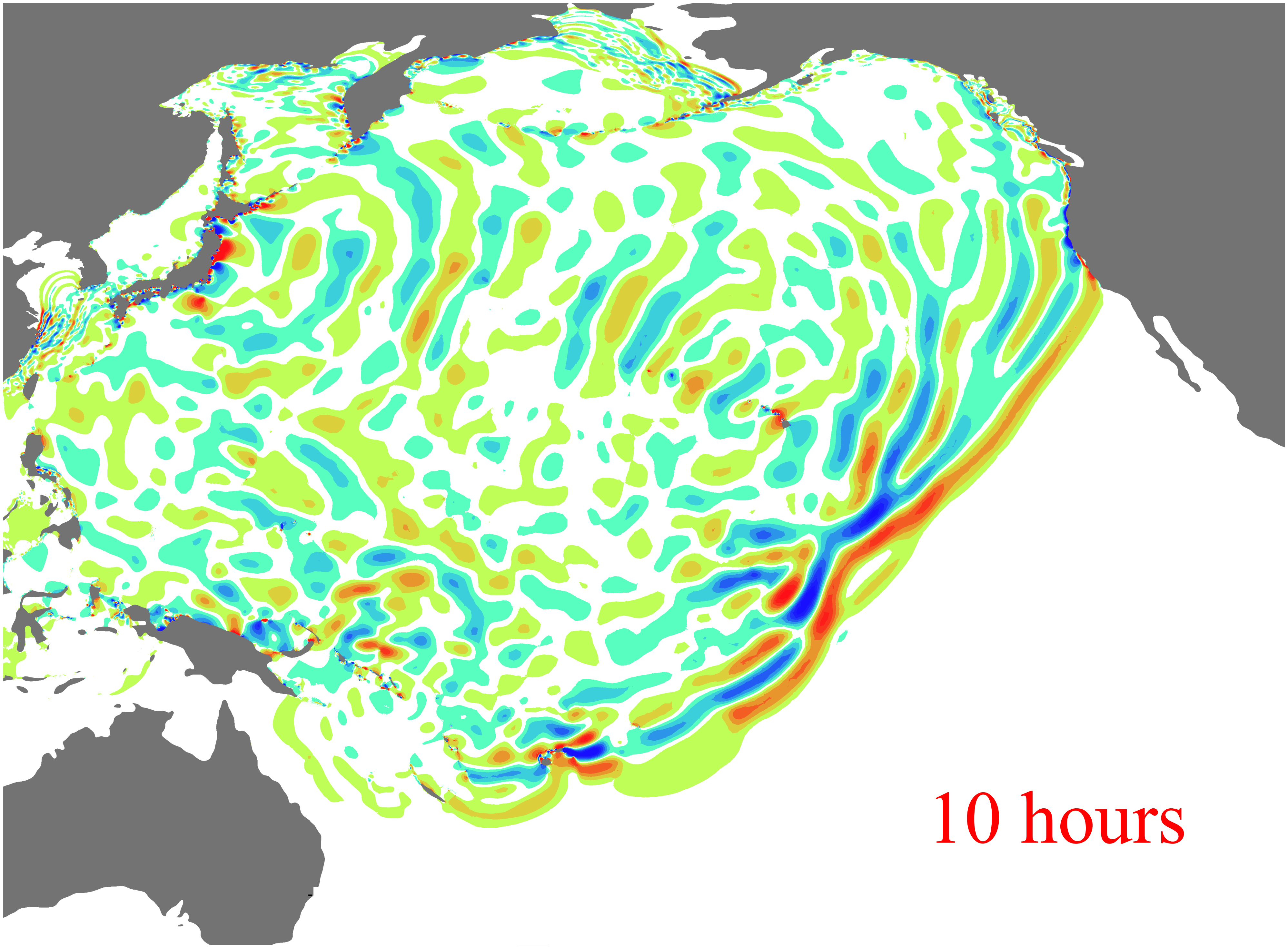

15 minutes
Y
Z
X


1 hour


Hawaï
Indonesia
4 hours


6 hours


Hawaï
7 hours


10 hours

A lot of physical processes inside the Shallow Water Equations

$$\boxed{\frac{\partial \eta}{\partial t}} + \boxed{\nabla \cdot \Big((h+\eta)\, \boldsymbol{u} \Big)} = 0,$$

$$\boxed{\frac{\partial \boldsymbol{u}}{\partial t}} + \boldsymbol{u} \cdot (\nabla \boldsymbol{u}) + \boxed{f\boldsymbol{k} \times \boldsymbol{u} + \boxed{g\nabla\eta}} = \boxed{\frac{1}{H}\nabla \cdot \Big(H\nu(\nabla \boldsymbol{u}) \Big)} + \frac{\boldsymbol{\tau}^s + \boldsymbol{\tau}^b}{\rho H}.$$

Waves equation
Equal-order discretization !

$$\boxed{P_1 - P_1}$$

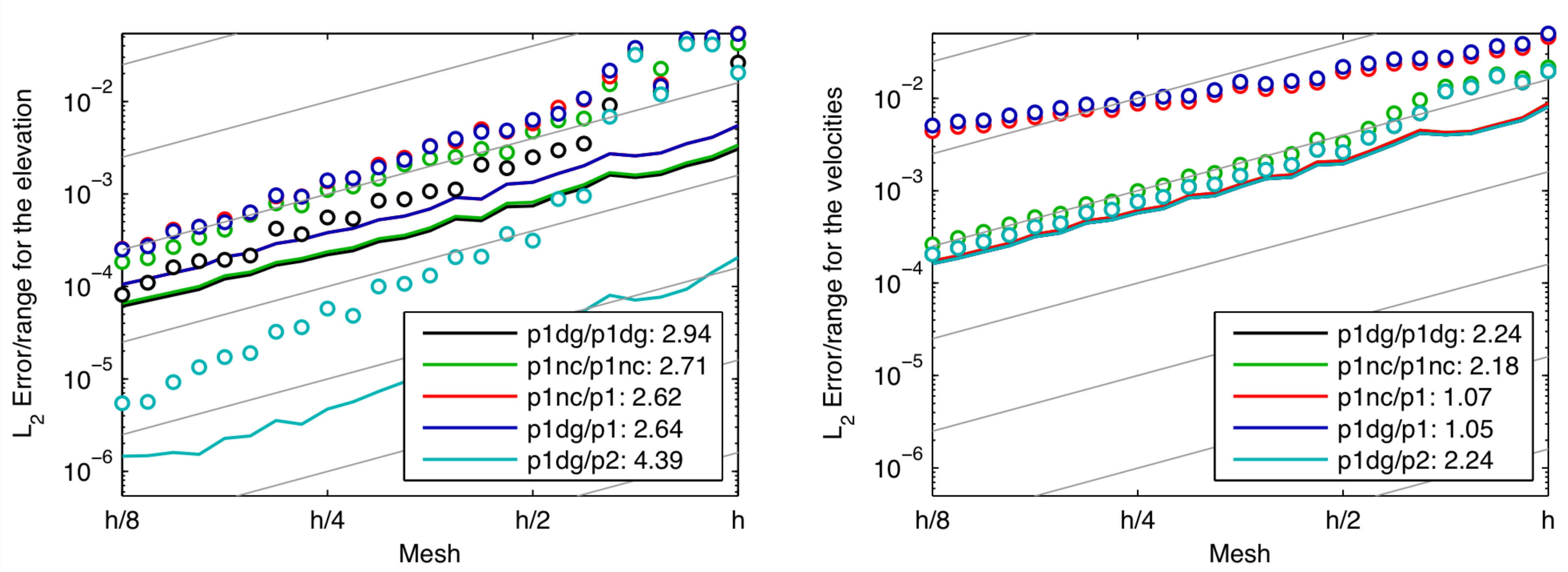
Geostrophy equilibrium
Exactly satisfied ?

$$\boxed{P_1^{DG} - P_2^{DG}}$$

Stokes problem:
LBB condition occurs !

$$\boxed{P_2 - P_1}$$

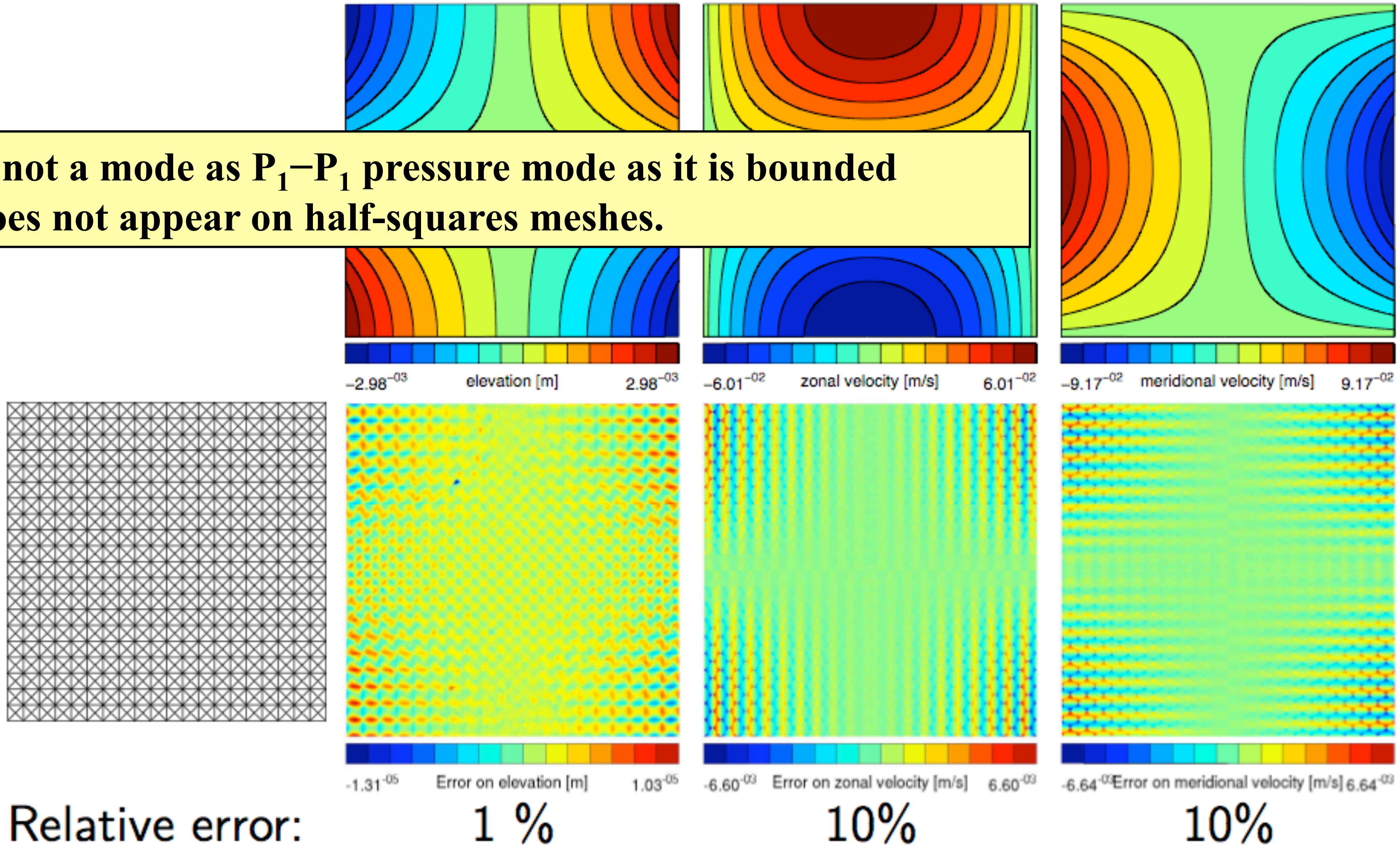
P_1^{NC}-P_1 inviscid computations look pretty nice …



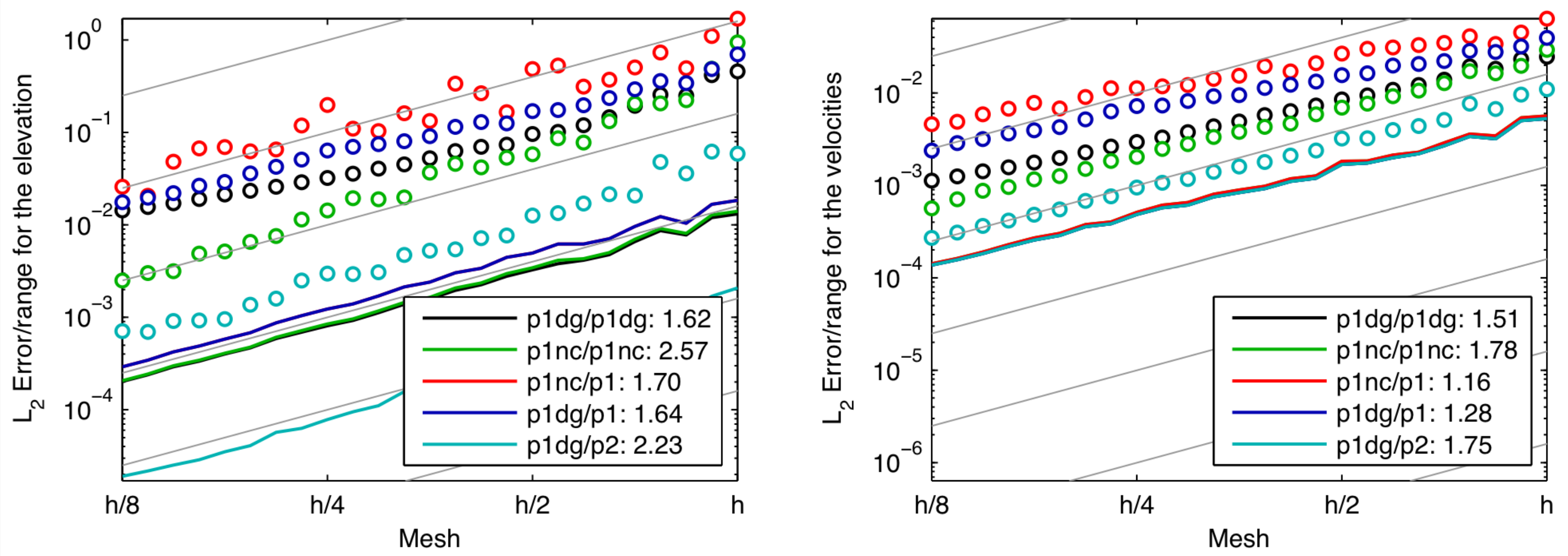
… but exhibit only a first-order convergence!

Structured noise is observed !

- it is not a mode as $P_1 - P_1$ pressure mode as it is bounded
- it does not appear on half-squares meshes.

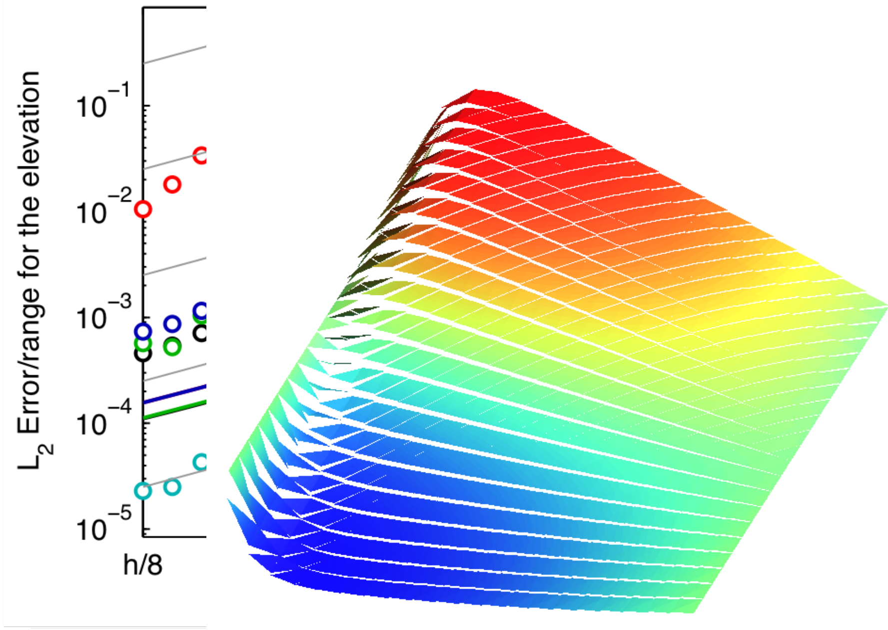


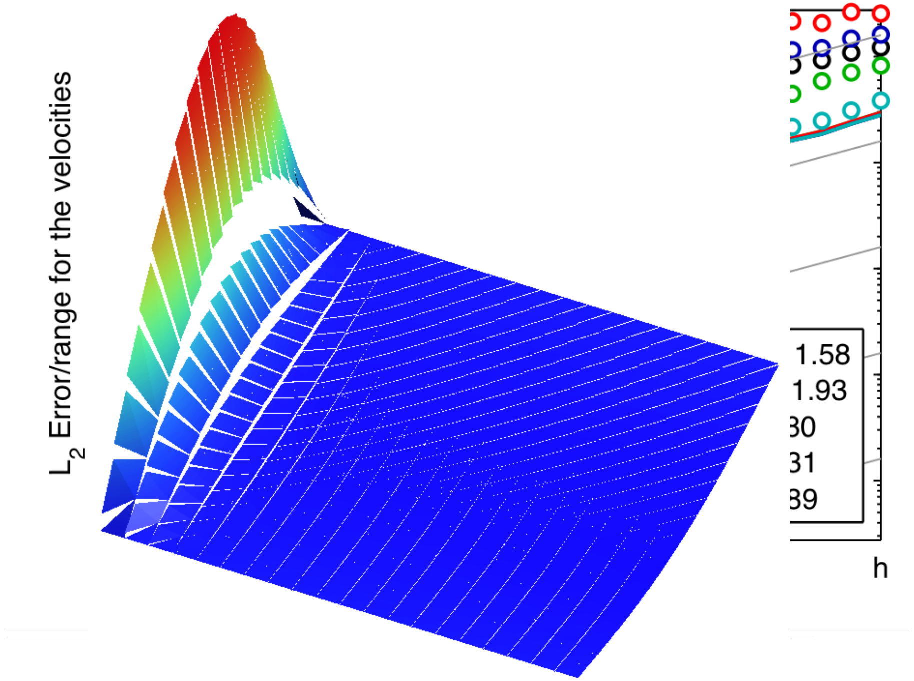
P_1^{DG}-P_2 wins the accuracy award!



- **Second-order convergence for all benchmarks.**
- **Higher order quadrature rules are required.**
- **Consistency requires to use P_2 tracers !**
- **Efficient iterative solution strategy ?**

Coriolis issue for P_1^{DG}-P_1^{DG}





- **Half an order of accuracy is lost with Coriolis**
- **Coriolis term has no corresponding interface term**
- **Only normal velocity jumps are removed by the Riemann solver**
- **Tangent velocity jumps amplified by Coriolis term and not damped**

The Galerkin Discontinuous Method

Finite Volumes

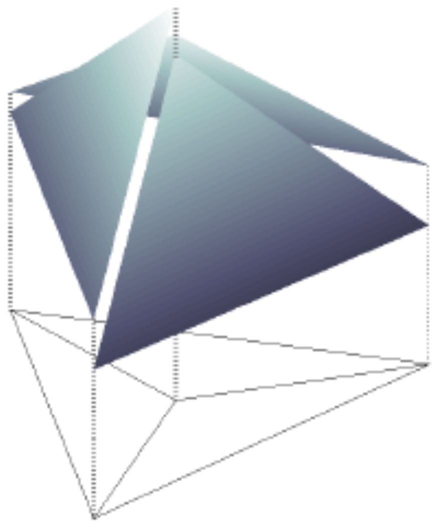
- **Natural treatment of wave-like terms**
- **Low order on unstructured meshes**

Continous Finite Elements

- **Optimal for second-order terms**
- **High order interpolation spaces**

Best of both approaches !

- **Wave terms handled in the finite volume spirit**
- **Second-order terms accurately handled with IP formulation**
- **High order interpolation spaces**



The Galerkin Discontinuous Method

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}, \eta) = \boldsymbol{f}(\boldsymbol{u}, \eta)$$

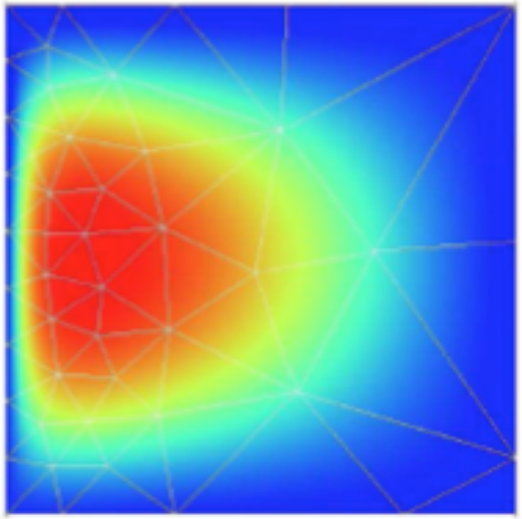
The classical DG weak formulation reads:

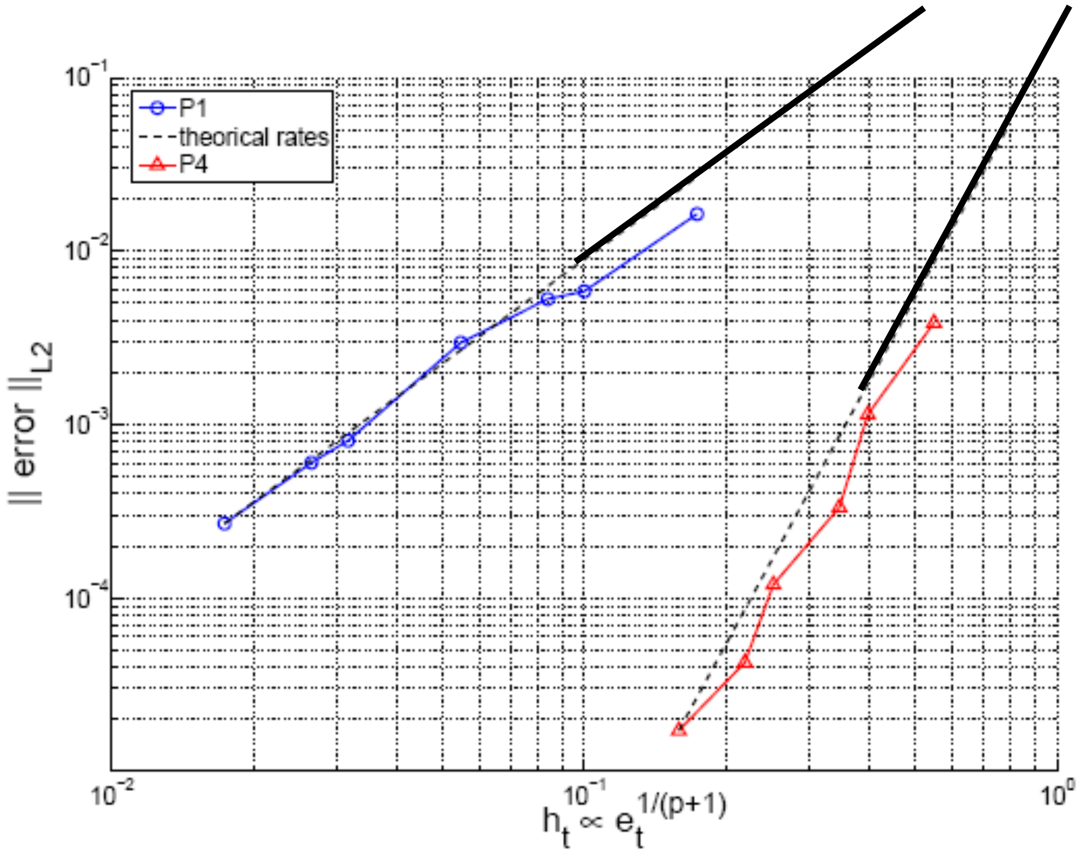
$$\frac{\partial}{\partial t} < \boldsymbol{u}^h \cdot \hat{\boldsymbol{u}}^h >_{\Omega_e} \quad - \quad < \boldsymbol{\sigma}(\boldsymbol{u}^h, \eta^h) \cdot \nabla \hat{\boldsymbol{u}}^h >_{\Omega_e}$$

$$+ \quad \ll \boldsymbol{\sigma}^*(\boldsymbol{u}^h, \eta^h) \cdot \boldsymbol{n} \cdot \hat{\boldsymbol{u}}^h \gg_{\partial\Omega_e} \quad = \quad \mathbf{0}$$

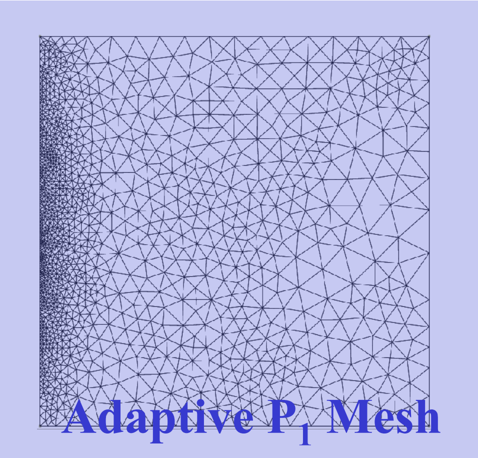
- **Bloc-diagonal global matrices**
- **Transfer between elements through the flux on the edges**
- **A weak collocated formulation can be also derived**
- **Upwinding by the flux evaluation (Riemann's solver)**

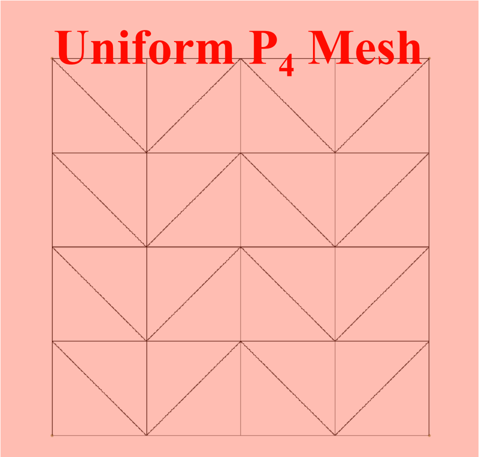
Theoretical rates of convergence are obtained for the analytical Stommel problem

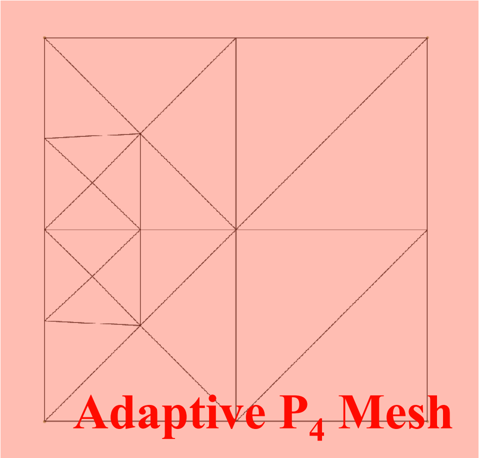


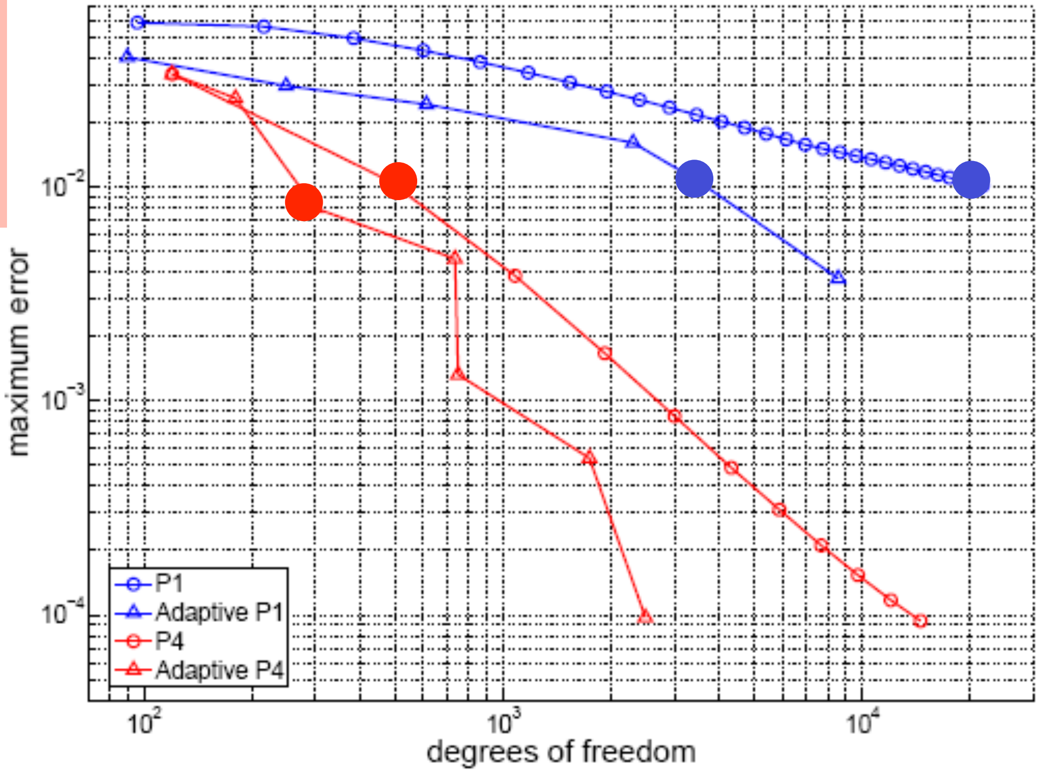


How does it converge ?









P_n^{DG}-P_n^{DG} is currently used because it is fast

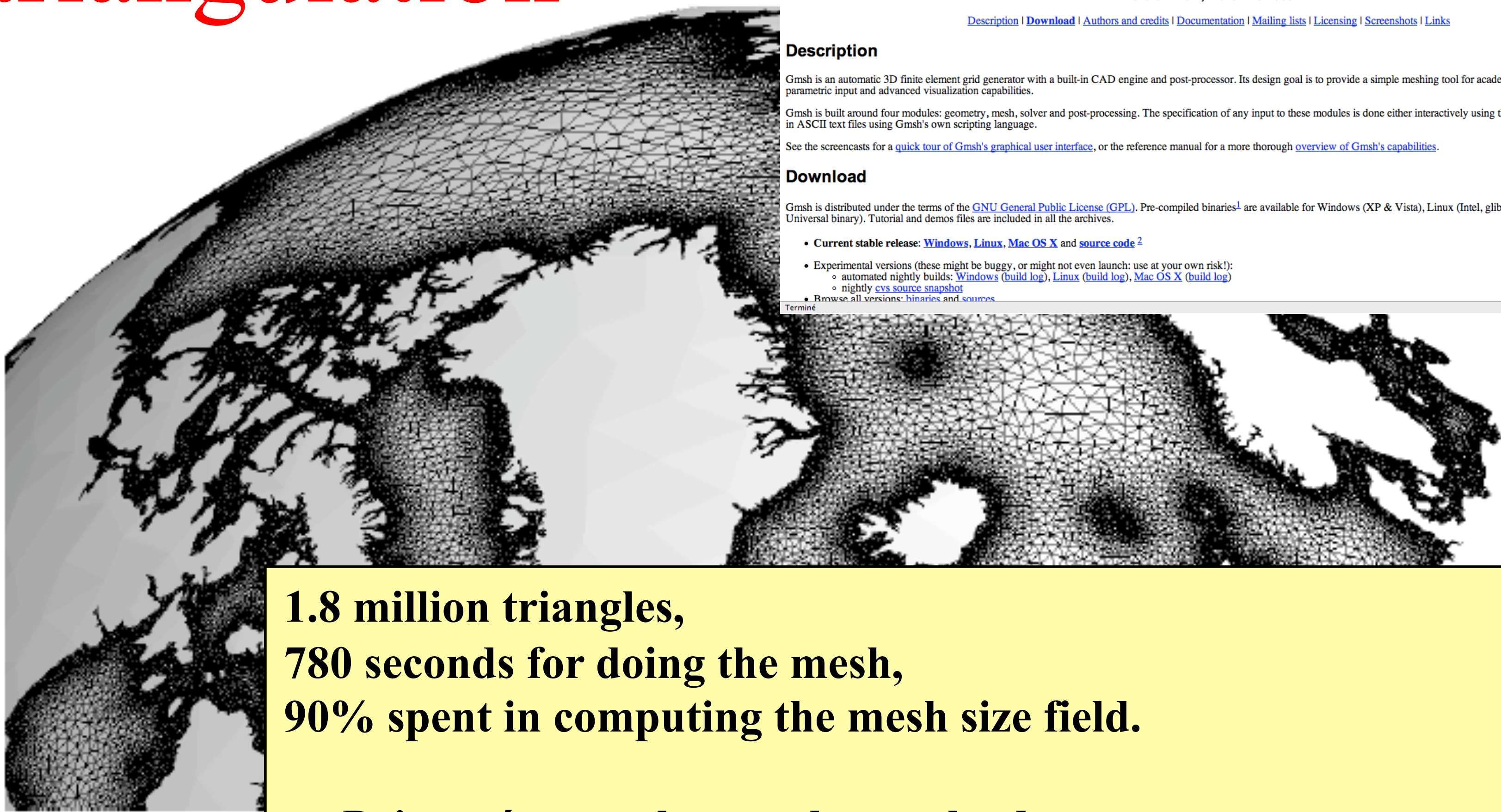
Implicit time marching

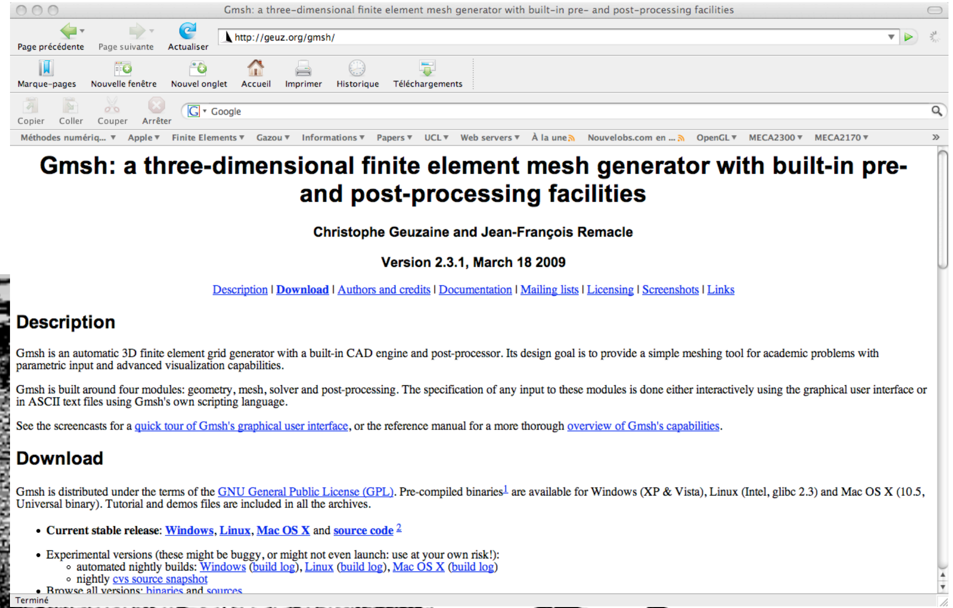
- **Implicit scheme needs linear solver**
- **DG + ILU(0) GMRES solution strategy is efficient**

Explicit time marching

- **Finite volume limiters can be applied**
- **Conservative wetting and drying procedures are available**

Delaunay based triangulation





Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities

Christophe Geuzaine and Jean-François Remacle

Version 2.3.1, March 18 2009

Description | Download | Authors and credits | Documentation | Mailing lists | Licensing | Screenshots | Links

Description

Gmsh is an automatic 3D finite element grid generator with a built-in CAD engine and post-processor. Its design goal is to provide a simple meshing tool for academic problems with parametric input and advanced visualization capabilities.

Gmsh is built around four modules: geometry, mesh, solver and post-processing. The specification of any input to these modules is done either interactively using the graphical user interface or in ASCII text files using Gmsh's own scripting language.

See the screencasts for a quick tour of Gmsh's graphical user interface, or the reference manual for a more thorough overview of Gmsh's capabilities.

Download

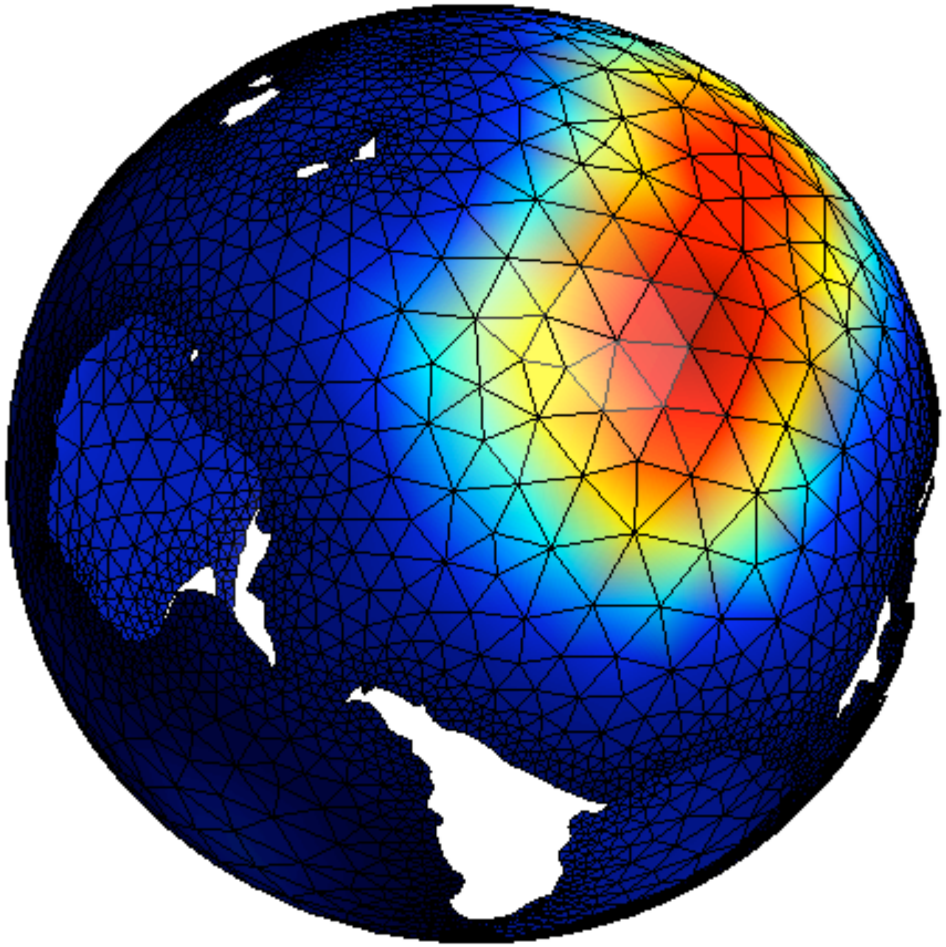
Gmsh is distributed under the terms of the GNU General Public License (GPL). Pre-compiled binaries[1] are available for Windows (XP & Vista), Linux (Intel, glibc 2.3) and Mac OS X (10.5, Universal binary). Tutorial and demos files are included in all the archives.

- Current stable release: Windows, Linux, Mac OS X and source code [2]
- Experimental versions (these might be buggy, or might not even launch: use at your own risk!):
  - automated nightly builds: Windows (build log), Linux (build log), Mac OS X (build log)
  - nightly cvs source snapshot

1.8 million triangles,
780 seconds for doing the mesh,
90% spent in computing the mesh size field.

- Poincaré waves have to be resolved
- Mesh size smaller along coastlines
- Geometry of the coastlines has to be represented

In a natural way, finite elements do not require a global system of coordinates!



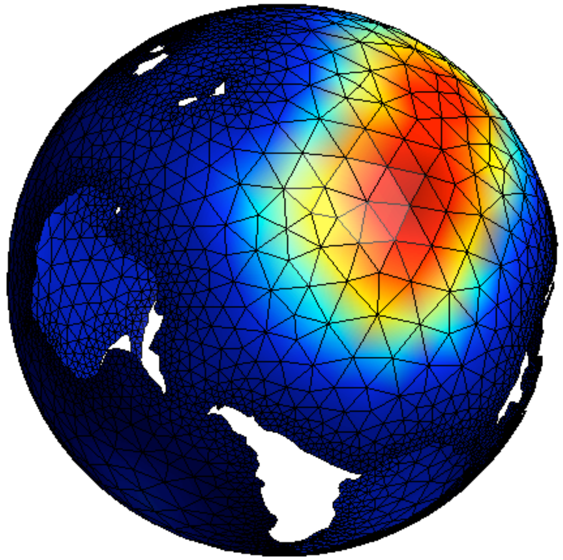
Solving the Laplace's equation on the sphere is trivial

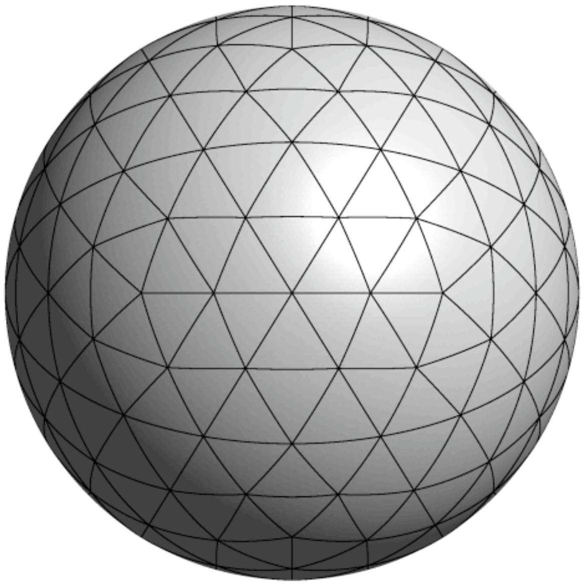
- **No need of spherical coordinates**
- **No poles singularity**
- **As simple as the planar problem**

The code :-)

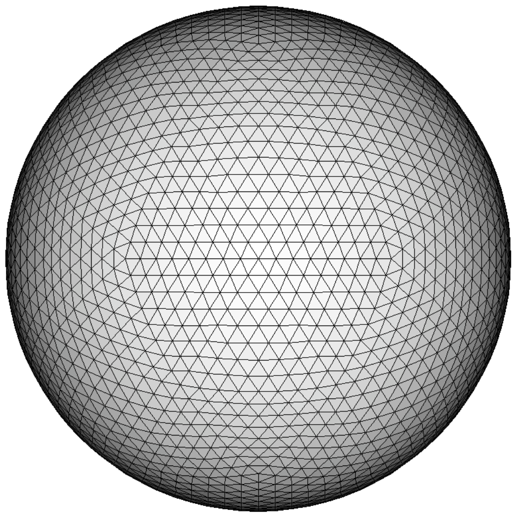
```
dxdxi = x2-x1; dxdet = x3-x1;
dydxi = y2-y1; dydet = y3-y1;
dzdxi = z2-z1; dzdet = z3-z1;

lgt11 = sqrt(dxdxi * dxdxi + dydxi * dydxi + dzdxi * dzdxi);
lgt22 = sqrt(dxdet * dxdet + dydet * dydet + dzdet * dzdet);
cos12 =      (dxdxi * dxdet + dydxi * dydet + dzdxi * dzdet) / (lgt11*lgt22);
sin12 = sqrt(1.0-cos12*cos12);
ajac  = sin12 * (lgt11*lgt22);
dxdxi = lgt11;          dydxi = 0.0;
dxdet = lgt22 * cos12; dydet = lgt22 * sin12;

dxidx =   dydet / ajac;
dxidy = - dxdet / ajac;
detdx = - dydxi / ajac;
detdy =   dxdxi / ajac;

for (int iInteg=0; iInteg < myElement.getNumberInteg(); iInteg++) {
  myElement.setShapes(iInteg);
  phi = myElement.getPhi();
  dphidxi = myElement.getDphidxi();
  dphidet = myElement.getDphideta();
  weight  = myElement.getWeight();

  for (int i=0;i<nn;i++) {
     dphidx[i] = (dphidxi[i] * dxidx + dphidet[i] * detdx);
     dphidy[i] = (dphidxi[i] * dxidy + dphidet[i] * detdy);   }

  for (int i=0;i<nn;i++) {
     for (int j=0;j<nn;j++) {
             Aloc(i,j) += ajac * weight[iInteg] * (dphidx[i]*dphidx[j] + dphidy[i]*dphidy
    [j]);}
     Bloc(i) += ajac * weight[iInteg]* source * phi[i];}}
}
```

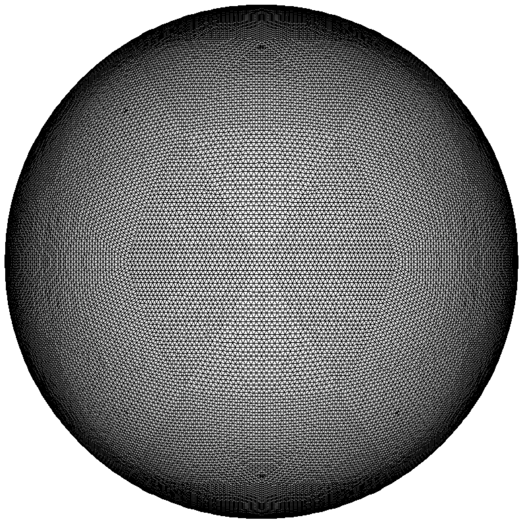
$$\nabla_{\xi\eta} u + f = 0$$





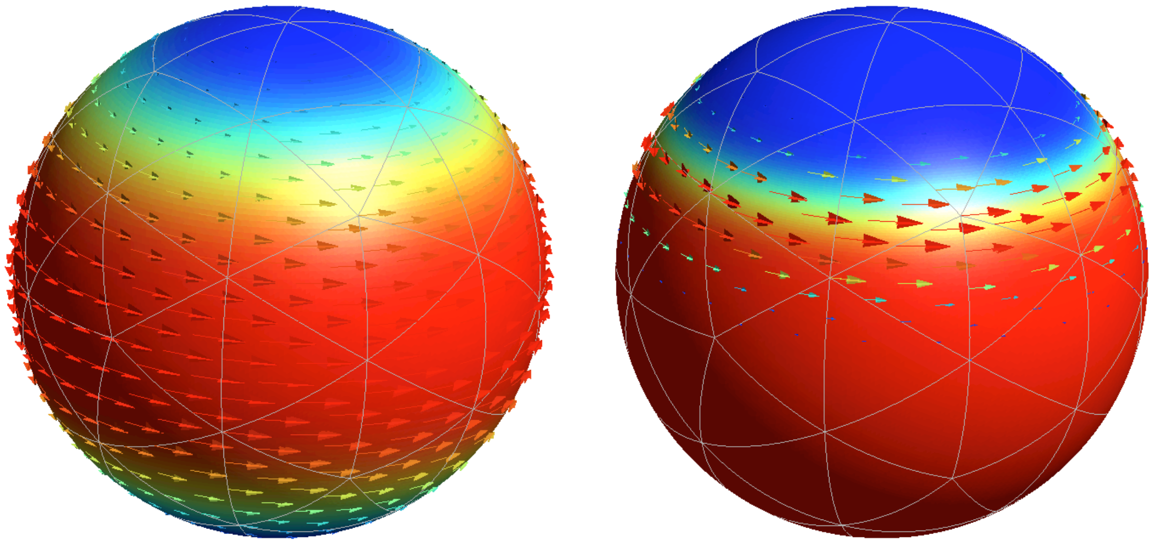
High-order versus low-order meshes





Williamson test case 2 and 3

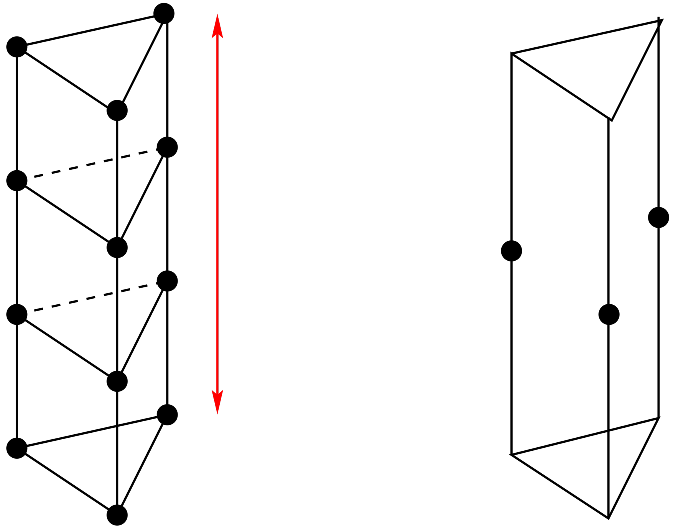
[Williamson et al. 1992]

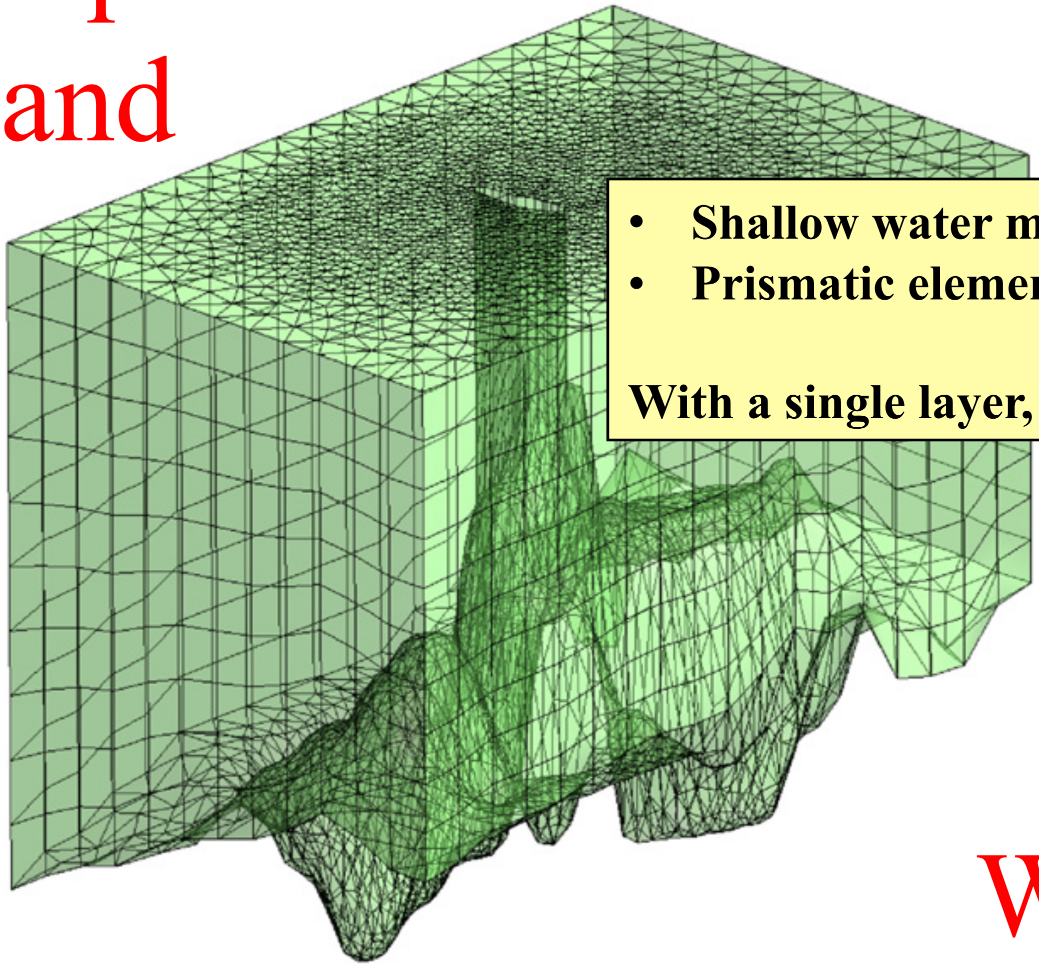


Analytical steady-state solution as a balance between non linear transport terms, pressure term and Coriolis force

The expected convergence rate is reached …

The hydrostatic Boussinesq equations and





- **Shallow water model is the depth-integrated 3D model**
- **Prismatic elements appear as a natural choice**

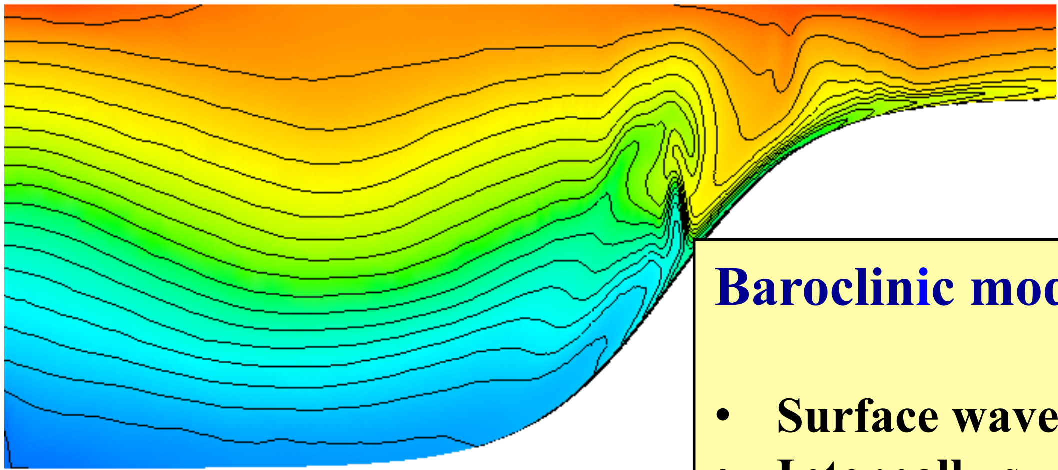
With a single layer, we solve the shallow water model !

… the Shallow Water Equations

3D : baroclinic effects take place!

Barotropic model

- **Surface waves and advection**
- **Subcritical for large scale problems**



Baroclinic model

- **Surface waves, advection and internal waves**
- **Internally supercritical flows are common**
- **Internal waves breaks can occur**
- **Density current fronts are supercritical**
- **Specific limiters are needed**

Internal waves couple…

- **Tracers are advected by the vertical velocity**
- **Vertical velocity is deduced from the horizontal velocity**
- **Pressure gradient is a source term for the horizontal momentum**
- **Pressure gradient is deduced from the density gradient**
- **Density gradient is linked to the tracers by an equation of state**

Interface terms must take into account this physics
at least for subcritical flows !

$$w \rightarrow S, T$$
$$\mathbf{u} \rightarrow w$$
$$p \rightarrow \mathbf{u}$$
$$\rho \rightarrow p$$
$$S, T \rightarrow \rho$$

…momenttum,
mass and tracers.

Lax-Friedrichs flux is the key ingredient …

$$\{F\} + \lambda_{\max}[u]$$

Deriving a Riemann solver would be quite difficult because the equations are not in a convervative form.

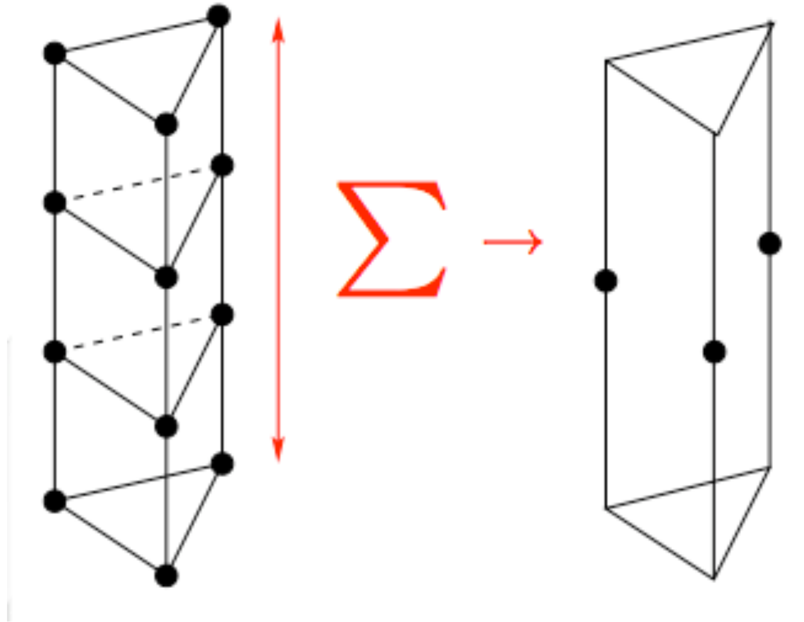
- **We add to the centered scheme a jump penalty term proportional to estimated maximum internal wave speed.**
- **Those terms are added only in prognostic equations related to baroclinic effects: momentum and tracer equations.**
- **The continuity equation does not have such interface terms.**

Semi-implicit (IMEX) Runge-Kutta schemes

- **The time step can easily be changed.**
- **High order versions are available.**
- **The linear system for 3d momentum has a block structure corresponding to the columns of dof 's.**

	Implicit	Explicit	Constrained
2d	Coriolis	Bottom friction	
	waves	horizontal diffusion	
		advection	
3d	Vertical processes	Bottom friction	waves
	Coriolis	horizontal diffusion	
		advection	

Implicit mode splitting procedure



- The 3D dof 's of a whole vertical line are aggregated into a single 2D dof.
- It can be viewed as a restriction on the functional space: the 2D mode corresponds to a single layer.

$$\int_{-h}^{\eta} \frac{\partial \boldsymbol{u}}{\partial t} + f\boldsymbol{k} \times \boldsymbol{u} + \ldots dz = (h+\eta)\left(\frac{\partial \boldsymbol{U}}{\partial t} + f\boldsymbol{k} \times \boldsymbol{U} + \ldots\right)$$

Elevation can be viewed as the 2d counterpart of vertical velocity

$$\boxed{\frac{\partial w}{\partial z} + \nabla_h \cdot \boldsymbol{u} = 0}$$

By integrating the equation over the vertical

$$w|_\eta - w|_{-h} + \int_{-h(x)}^{\eta(x)} \nabla_h \cdot \boldsymbol{u}\ dz = 0$$

$$\frac{\partial \eta}{\partial t} + \underbrace{\boldsymbol{u}|_\eta \nabla_h \eta - \boldsymbol{u}|_{-h} \nabla_h(-h) + \int_{-h(x)}^{\eta(x)} \nabla_h \cdot \boldsymbol{u}\, dz}_{\nabla_h \cdot \int_{-h}^{\eta} \boldsymbol{u}\ \mathrm{d}z} = 0$$

$$\boxed{\frac{\partial \eta}{\partial t} + \nabla_h \cdot ((h + \eta)\boldsymbol{U}) = 0}$$

Implicit mode splitting procedure

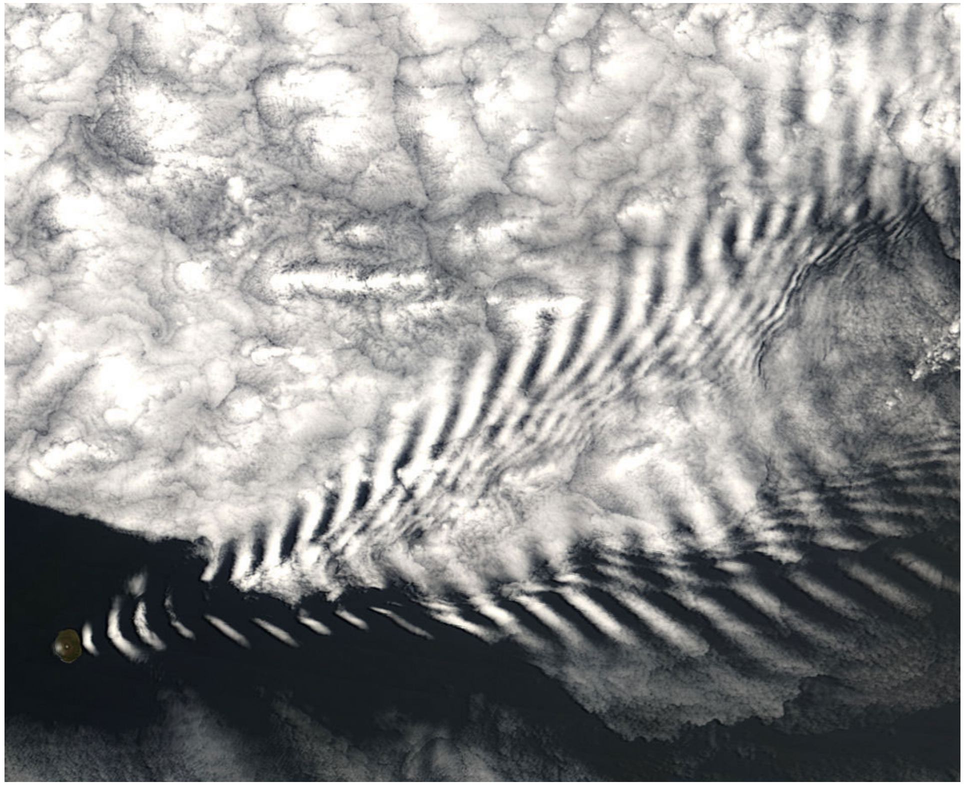
$$f(u_i) = 0$$
$$\sum w_i u_i = U$$

Lagrange multipliers ensure compatibility.

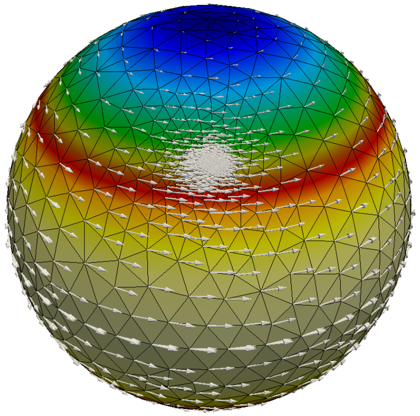
- **Requiring compatibily add too much equations.**
- **Incorporating Lagrange multipliers allows us to weakly impose the compatbility between the 2D and 3D velocity fields**

$$f(u_i) + w_i \lambda = 0$$
$$\sum w_i u_i = U$$

Internal waves in the lee of a moderately tall seamount

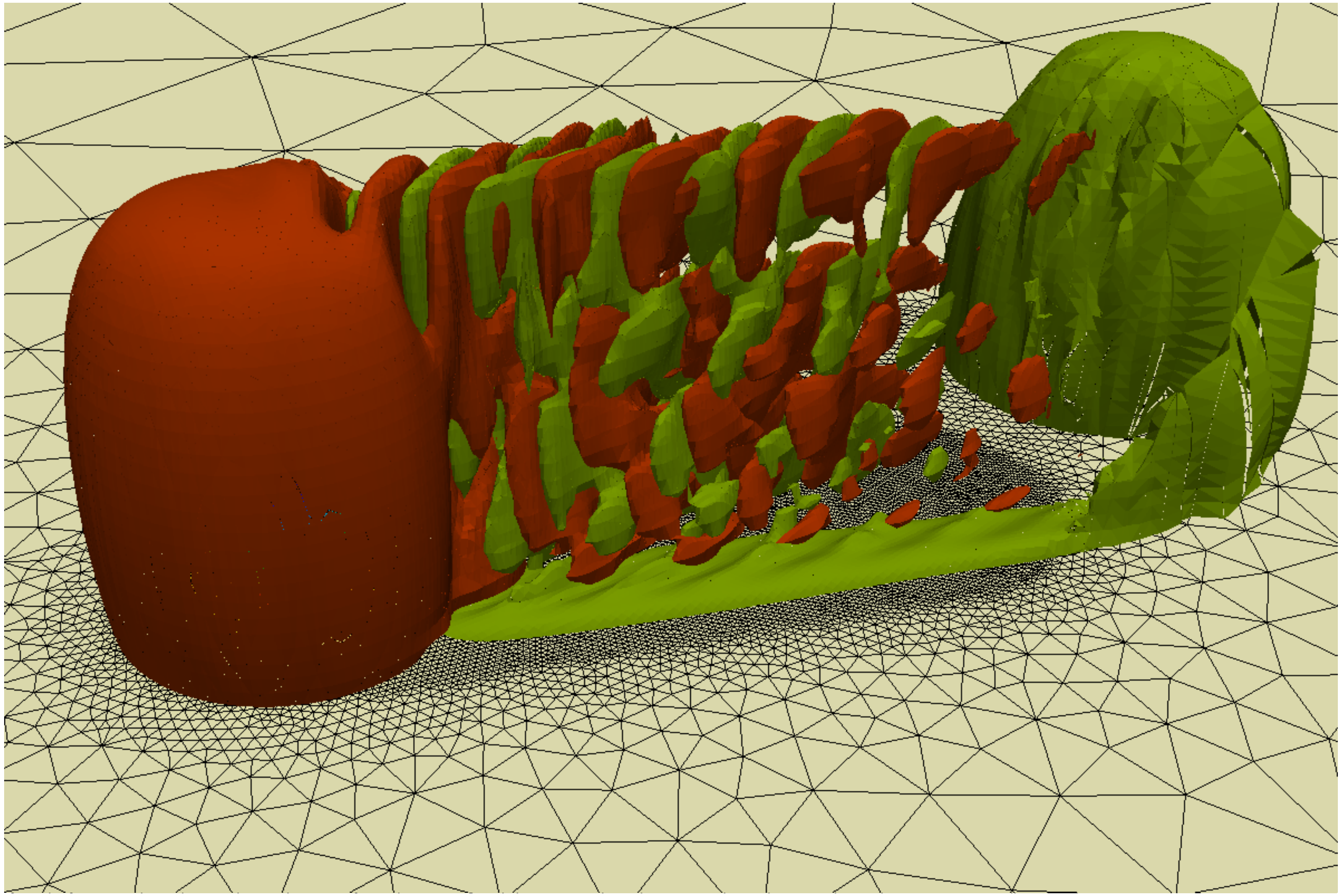


Cloud waves in the lee of Amsterdam island (NASA image from J. Schmalz)

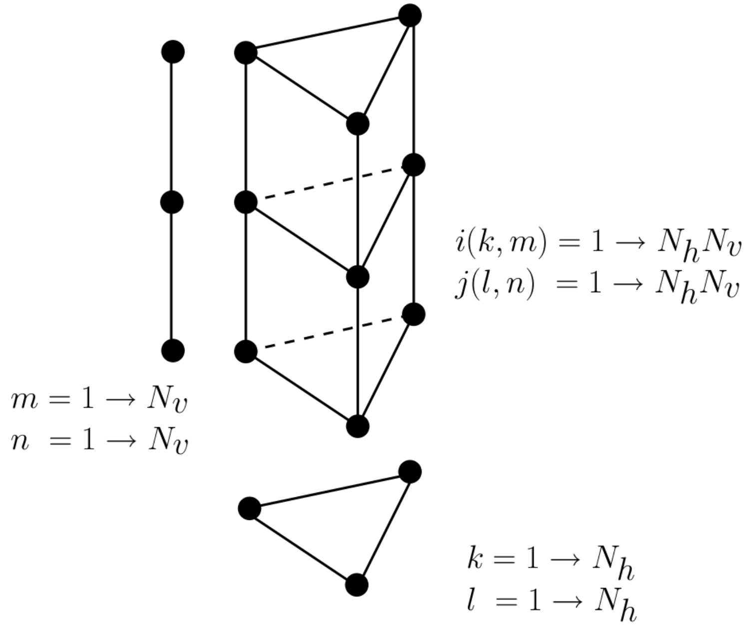


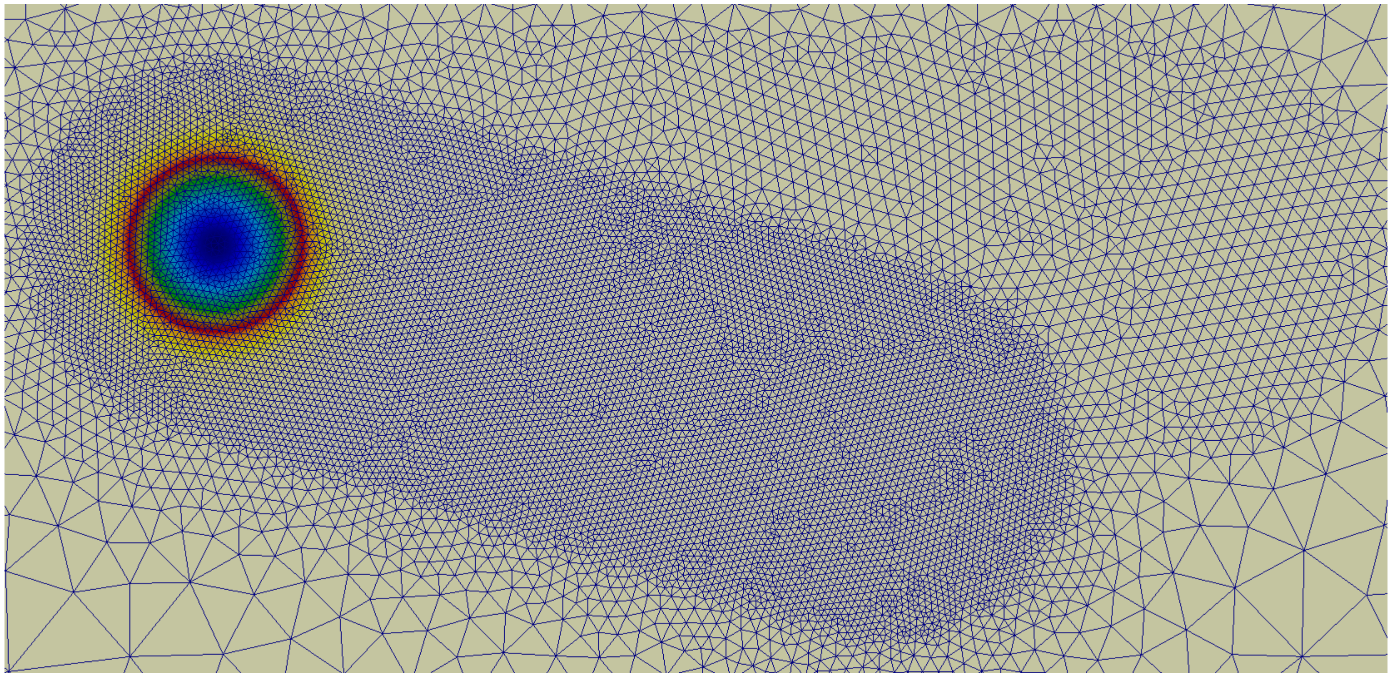
The computation starts with a global zonal geostrophic equilibrium ignoring the seamount as in Williamson testcase 5

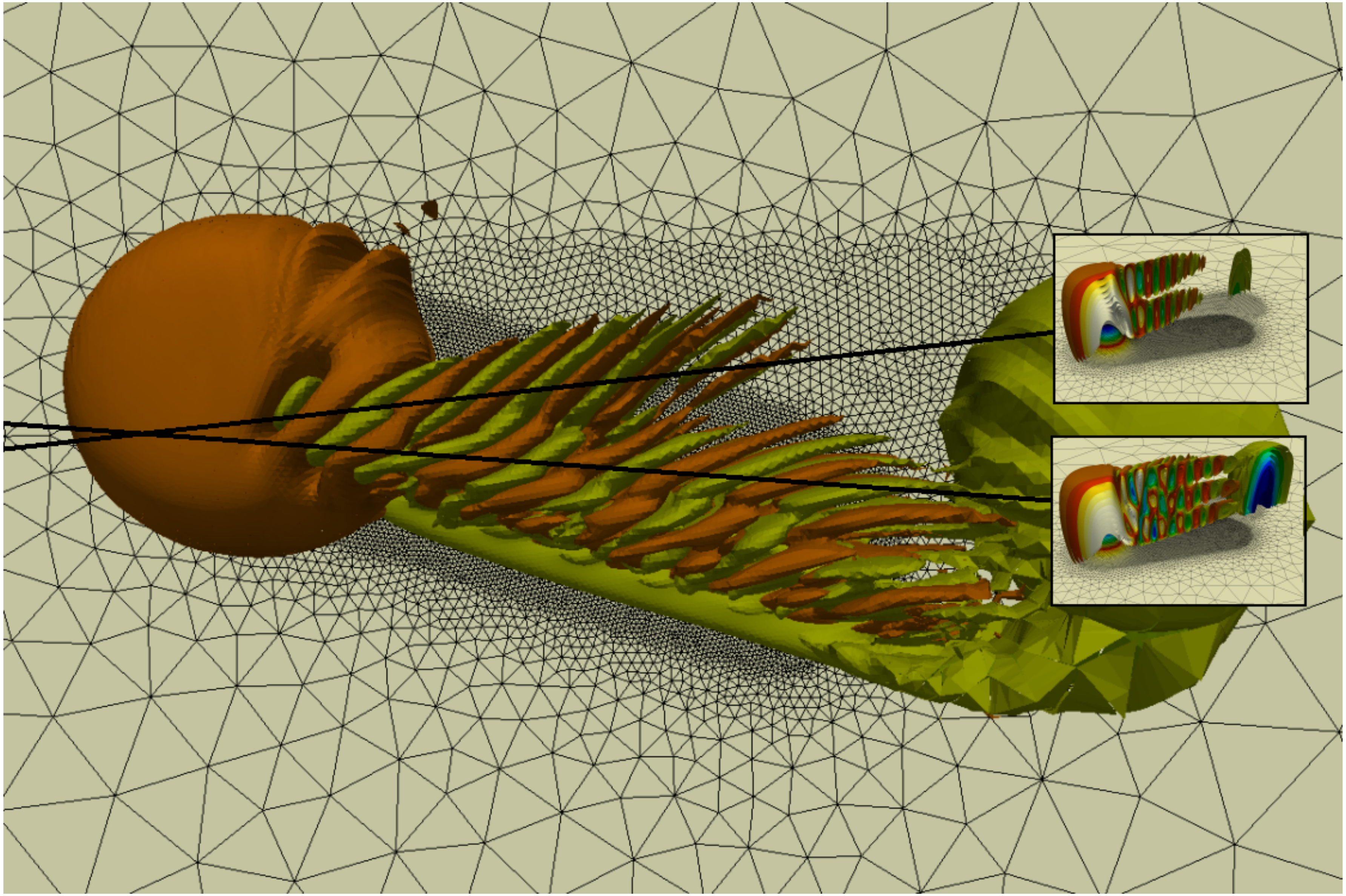
7 days evolution of density deviation field



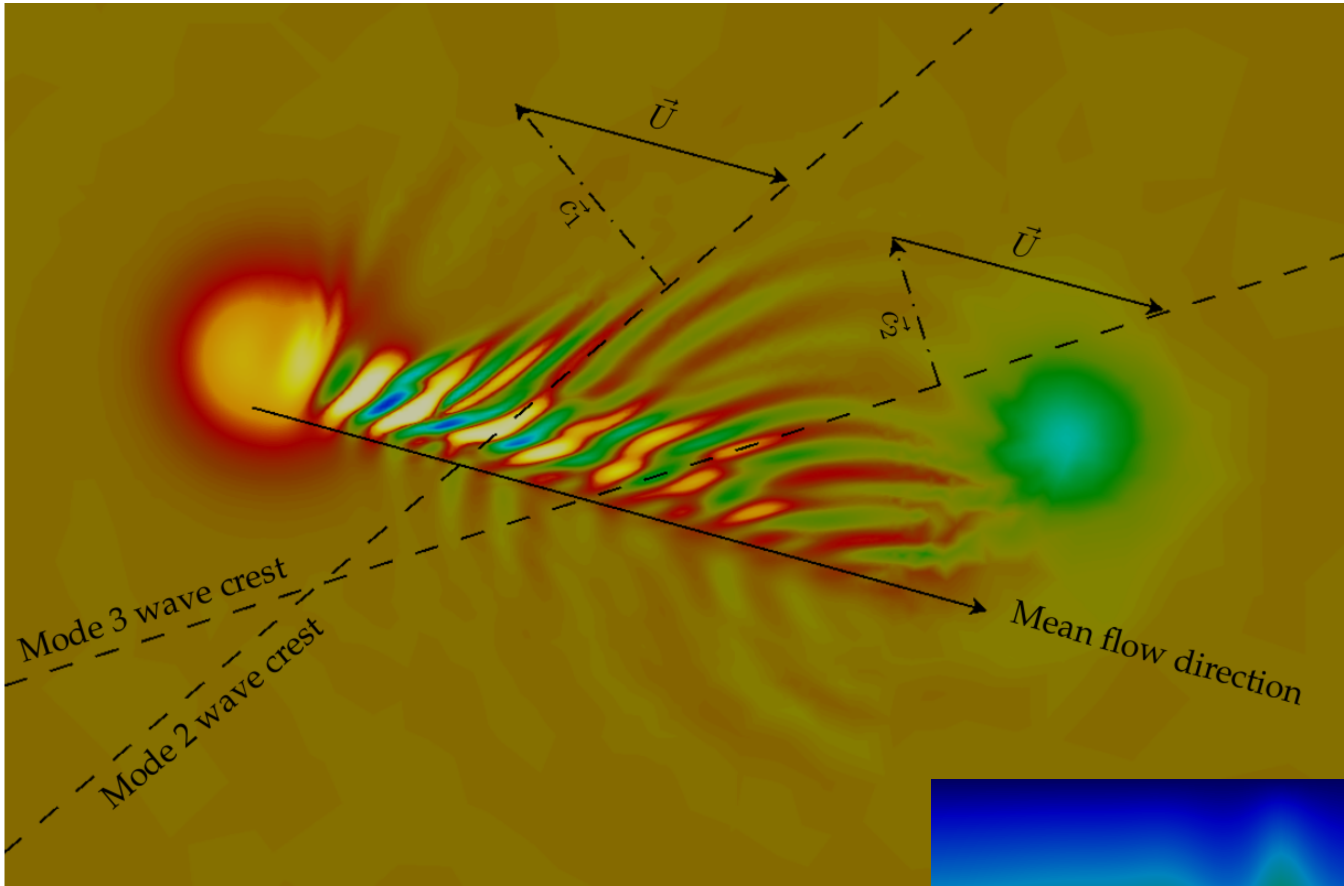
Mesh of 23562 triangles extruded into 25 σ layers

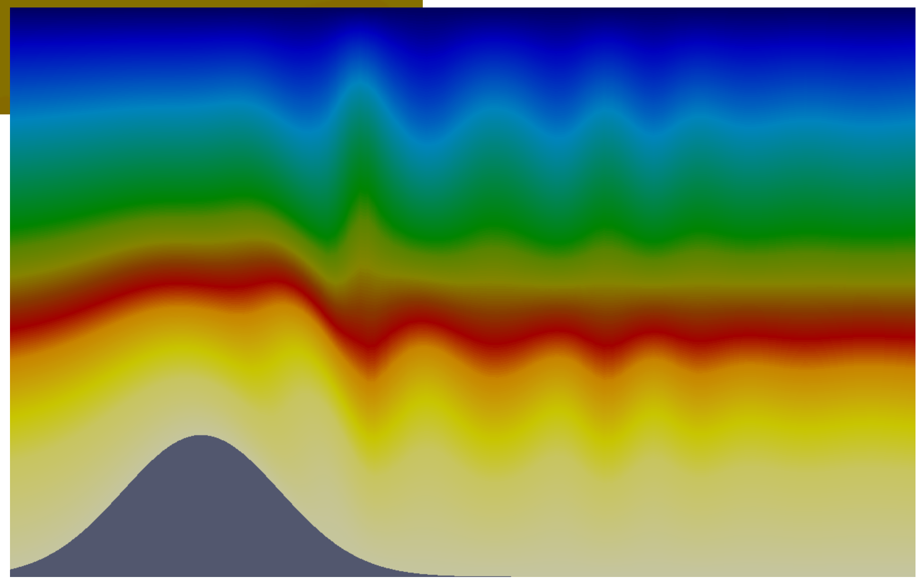






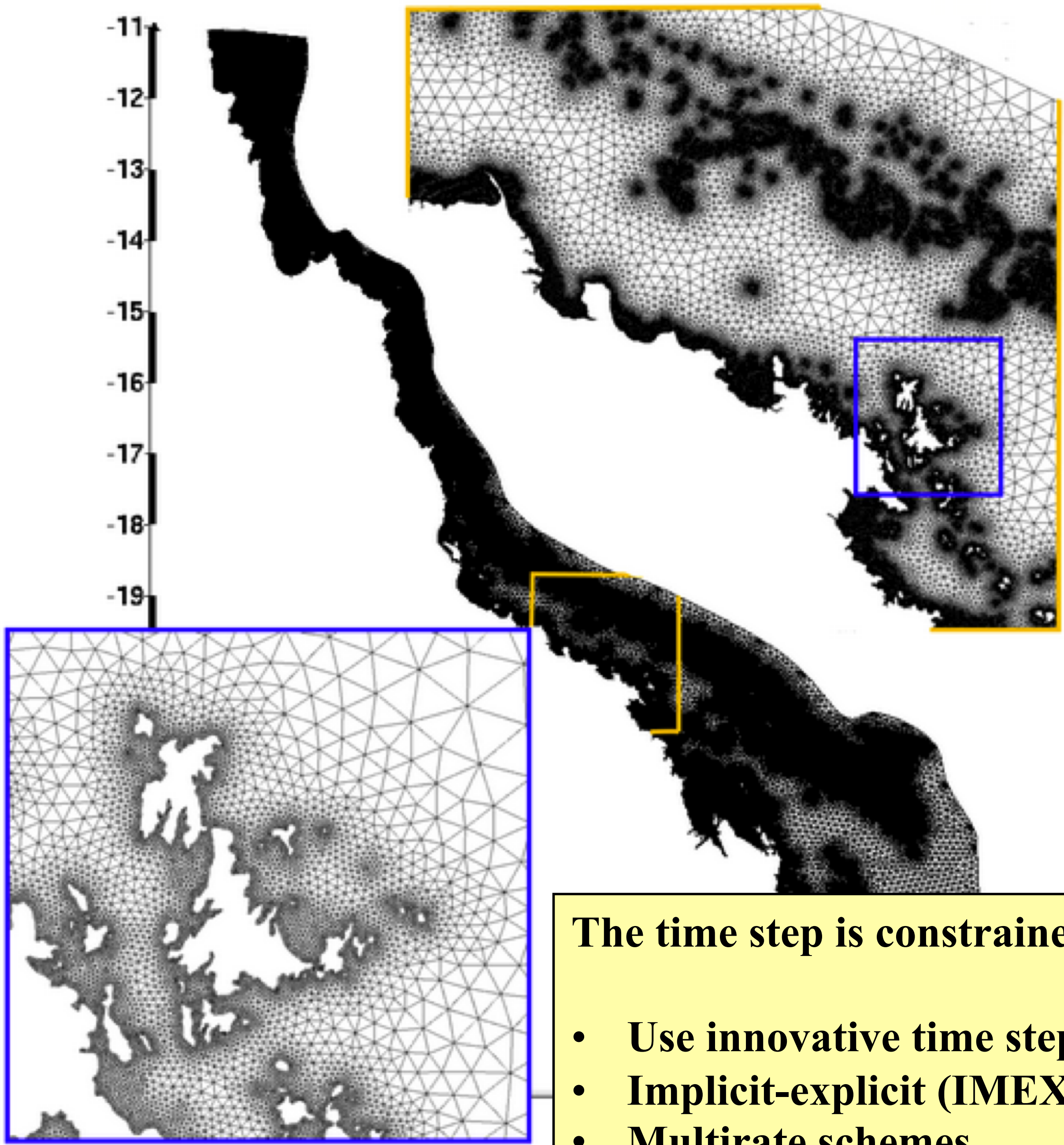
Two well separated modes at day 7





Cut in the density field at day 7

The time stepping issue



- ***890,000 triangles***
- ***Smallest element : 7 m***
- ***Largest element : 3,300 m***
- ***99.9 % > 60m***

The time step is constrained by the smallest element .

- **Use innovative time stepping procedures**
- **Implicit-explicit (IMEX) schemes**
- **Multirate schemes**

Reduce cost by 1000 !
Use high performance computers !

10 Gflops
2 processors

1.759 Pflops
224,162 processors

- **Exploit single precision BLAS/LAPACK for the efficient implementation of the explicit and implicit discontinuous Galerkin methods.**
- **Implement new time-integration procedures adapting the time step to the physical processes.**
- **Introduce multi-level methods for the implicit linear and non-linear solvers with multigrid methods as a preconditioner for stiff, non-linear and non-positive-definite systems.**

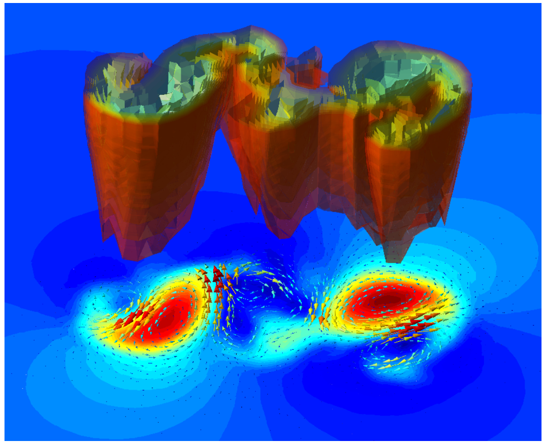
Each route could reduce the computational cost by one order of magnitude.

2D conclusions

- **DG is the most compelling solution**

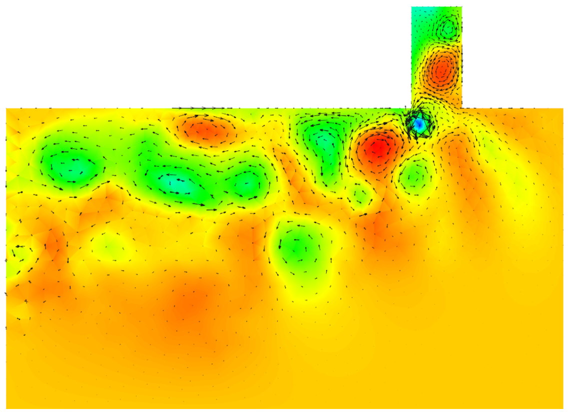
- **Both implicit and explicit procedures are needed**
  - **Implicit for long term simulations**
  - **Explicit allows to use simple limiters**

- **P_2 on curved meshes would be faster and more accurate with the same number of dofs**
  - **Efficient limiters for P_2 are not obvious to derive**

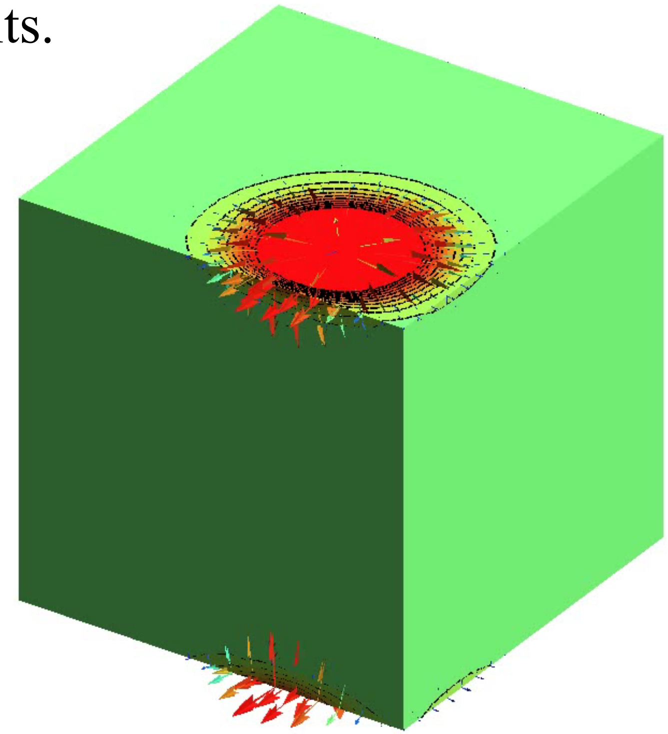


3D conclusions
The long way to realistic models

- **An accurate DG discretization on the sphere withe a flexible implicit mode splitting has been developed**
- **It should work with limiters for supercritical flows**
- **Mode splitting may not be the best solution**
  Multigrid implicit scheme, aware of the physics, is also attractive



- As far as the laws of mathematics refer to reality, they are not certain, and as far as they are certain, they do not refer to reality.
  Albert Einstein
- Everything is vague to a degree you do not realize
  till you have tried to make it precise.
  Bertrand Russell
- In these matters the only certainty is that nothing is certain.
  Pliny the Elder
- However beautiful the strategy,
  you should occasionally look at the results.
  Sir Winston Churchill



Quotes by (other) famous simulators