HySEA cloud computing platform: Models, schemes and HPC

Manuel J. Castro Díaz
EDANYA Group. University of Málaga. Spain.

V EWM Summer School. May 2013.

Outline

Edanya Team

- Team leader: Carlos Parés Madroñal.
- Marc de la Asunción.
- Manuel J. Castro Díaz.
- José María Gallardo Molina.
- José Manuel González Vida.
- Jorge Macías Sánchez.
- Tomás Morales de Luna.
- María de la Luz Muñoz Ruiz.
- Sergio Ortega Acosta.
- Carlos Sánchez Linares

Goals

Edanya: Main Goals

Development of robust, reliable and low computational cost numerical tools for the simulation of geophysical flows and the prediction of emergency situations such as river flooding or oil spills, tsunamis, debris avalanches ...

Goals

Edanya: Main Goals

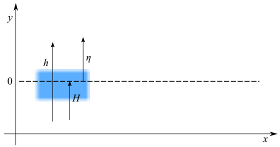
Development of robust, reliable and low computational cost numerical tools for the simulation of geophysical flows and the prediction of emergency situations such as river flooding or oil spills, tsunamis, debris avalanches . . .

Ingredients

- Depth averaged models: Shallow-water type systems.
- Numerical schemes: High order path-conservative finite volume schemes.
- HySEA: High Performance Cloud Computing software to simulate geophysical flows.

One-layer shallow water system

$$
\left\{
\begin{aligned}
&\frac{\partial h}{\partial t} + \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} = 0 \\
&\frac{\partial q_x}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q_x^2}{h} + \frac{g}{2}h^2\right) + \frac{\partial}{\partial y}\left(\frac{q_x q_y}{h}\right) = gh\frac{\partial H}{\partial x} + S_{f_1}(w) \\
&\frac{\partial q_y}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q_x q_y}{h}\right) + \frac{\partial}{\partial y}\left(\frac{q_y^2}{h} + \frac{g}{2}h^2\right) = gh\frac{\partial H}{\partial y} + S_{f_2}(w),
\end{aligned}
\right.
$$



- $w = (h, q_x, q_y)^T$,
- h is the water depth,
- $\boldsymbol{q} = (q_x, q_y)$, is the mass flow,
- $\boldsymbol{u} = (u_x, u_y) = \dfrac{\boldsymbol{q}}{h}$ is the mean velocity,
- H is the bathymetry,
- g is the acceleration of gravity,
- $S_{f_i}(w)$, $i = 1, 2$, parametrize the friction terms.

One-layer shallow water system

$$
\begin{cases}
\dfrac{\partial h}{\partial t} + \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} = 0 \\[2mm]
\dfrac{\partial q_x}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_x^2}{h} + \dfrac{g}{2}h^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_x q_y}{h}\right) = gh\dfrac{\partial H}{\partial x} + S_{f_1}(w) \\[2mm]
\dfrac{\partial q_y}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_x q_y}{h}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_y^2}{h} + \dfrac{g}{2}h^2\right) = gh\dfrac{\partial H}{\partial y} + S_{f_2}(w),
\end{cases}
$$

- Quadratic:

$$S_{f_1}(w) = -c u_x \|\boldsymbol{u}\| \quad S_{f_2}(w) = -c u_y \|\boldsymbol{u}\|;$$

- Chèzy:

$$S_{f_1}(w) = -\frac{g}{C^2} u_x \|\boldsymbol{u}\| \quad S_{f_2}(w) = -\frac{g}{C^2} u_y \|\boldsymbol{u}\|;$$

- Manning:

$$S_{f_1}(w) = -gh\frac{n^2}{h^{4/3}} u_x \|\boldsymbol{u}\| \quad S_{f_2}(w) = -gh\frac{n^2}{h^{4/3}} u_y \|\boldsymbol{u}\|.$$

One-layer shallow water system

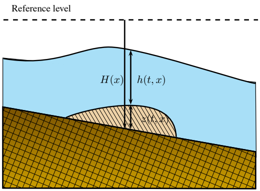
$$
\begin{cases}
\dfrac{\partial h}{\partial t} + \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} = 0 \\
\dfrac{\partial q_x}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_x^2}{h} + \dfrac{g}{2}h^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_x q_y}{h}\right) = gh\dfrac{\partial H}{\partial x} + S_{f_1}(w) \\
\dfrac{\partial q_y}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_x q_y}{h}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_y^2}{h} + \dfrac{g}{2}h^2\right) = gh\dfrac{\partial H}{\partial y} + S_{f_2}(w),
\end{cases}
$$

J.F. Gerbeau, B. Perthame.

Derivation of viscous Saint-Venant system for laminar shallow water; Numerical validation.
Discrete and continuous dynamical systems- Series B 1 (1):89-102, 2001.

Shallow water-Exner equations

$$
\begin{cases}
\dfrac{\partial h}{\partial t} + \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} = 0 \\[2ex]
\dfrac{\partial q_x}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_x^2}{h} + \dfrac{g}{2}h^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_x q_y}{h}\right) = -gh\dfrac{\partial z_b}{\partial x} + gh\dfrac{\partial \tilde{H}}{\partial x} + S_{f_1}(w) \\[2ex]
\dfrac{\partial q_y}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_x q_y}{h}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_y^2}{h} + \dfrac{g}{2}h^2\right) = -gh\dfrac{\partial z_b}{\partial y} + gh\dfrac{\partial \tilde{H}}{\partial y} + S_{f_2}(w), \\[2ex]
\dfrac{\partial z_b}{\partial t} + \xi\left(\dfrac{\partial q_{b,x}}{\partial x}(w) + \dfrac{\partial q_{b,y}}{\partial y}(w)\right) = 0
\end{cases}
$$



- $w = (h, q_x, q_y, z_b)^T$;
- h is the water depth, z_b the sediment layer depth;
- $\boldsymbol{q} = (q_x, q_y)$, is the mass flow;
- $\tilde{H}$ is the non-erodible bathymetry;
- $\xi = 1/(1-\rho_0)$ and ρ_0 is the porosity of the sediment;
- $(q_{b,x}(w), q_{b,y}(w))$ the solid transport discharge.

Shallow water-Exner equations

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} = 0 \\ \dfrac{\partial q_x}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_x^2}{h} + \dfrac{g}{2}h^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_x q_y}{h}\right) = -gh\dfrac{\partial z_b}{\partial x} + gh\dfrac{\partial \tilde{H}}{\partial x} + S_{f_1}(w) \\ \dfrac{\partial q_y}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_x q_y}{h}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_y^2}{h} + \dfrac{g}{2}h^2\right) = -gh\dfrac{\partial z_b}{\partial y} + gh\dfrac{\partial \tilde{H}}{\partial y} + S_{f_2}(w), \\ \dfrac{\partial z_b}{\partial t} + \xi\left(\dfrac{\partial q_{b,x}}{\partial x}(w) + \dfrac{\partial q_{b,y}}{\partial y}(w)\right) = 0 \end{cases}$$

- Fowler-Grass:

$$q_{b,x}(w) = A_g \frac{z_b}{\overline{z}_0} u_x \|\boldsymbol{u}\|^2 \quad q_{b,y}(w) = A_g \frac{z_b}{\overline{z}_0} u_y \|\boldsymbol{u}\|^2,$$

where $\overline{z}_0$ represents the mean value of the thickness of the sediment layer.

Shallow water-Exner equations

$$
\begin{cases}
\dfrac{\partial h}{\partial t} + \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} = 0 \\
\dfrac{\partial q_x}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_x^2}{h} + \dfrac{g}{2}h^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_x q_y}{h}\right) = -gh\dfrac{\partial z_b}{\partial x} + gh\dfrac{\partial \tilde{H}}{\partial x} + S_{f_1}(w) \\
\dfrac{\partial q_y}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_x q_y}{h}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_y^2}{h} + \dfrac{g}{2}h^2\right) = -gh\dfrac{\partial z_b}{\partial y} + gh\dfrac{\partial \tilde{H}}{\partial y} + S_{f_2}(w), \\
\dfrac{\partial z_b}{\partial t} + \xi\left(\dfrac{\partial q_{b,x}}{\partial x}(w) + \dfrac{\partial q_{b,y}}{\partial y}(w)\right) = 0
\end{cases}
$$

- Fowler- Meyer-Peter &Müller:

$$
q_{b,x} = 8\frac{z_b}{z_0}\sqrt{(\rho_s/\rho_f - 1)gd_i^3}\,\frac{u_x}{\|\boldsymbol{u}\|}\max\left(\tau_* - \tau_{*,c}, 0\right)^{3/2}
$$

$$
q_{b,y} = 8\frac{z_b}{z_0}\sqrt{(\rho_s/\rho_f - 1)gd_i^3}\,\frac{u_y}{\|\boldsymbol{u}\|}\max\left(\tau_* - \tau_{*,c}, 0\right)^{3/2}
$$

$$
\tau_* = \frac{\rho_f n^2 \|\boldsymbol{u}\|^2}{(\rho_s - \rho_f)d_i h^{1/3}}.
$$

with ρ_s is the density of the sediment, ρ_f the density of the fluid, d_i is the mean sediment grain diameter, n is the Manning coefficient and $\tau_{*,c}$ is the non-dimensional critical shear stress ($\tau_{*,c} \approx 0.047$).

Shallow water-Exner equations

$$
\begin{cases}
\dfrac{\partial h}{\partial t} + \dfrac{\partial q_x}{\partial x} + \dfrac{\partial q_y}{\partial y} = 0 \\[2mm]
\dfrac{\partial q_x}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_x^2}{h} + \dfrac{g}{2}h^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_x q_y}{h}\right) = -gh\dfrac{\partial z_b}{\partial x} + gh\dfrac{\partial \tilde{H}}{\partial x} + S_{f_1}(w) \\[2mm]
\dfrac{\partial q_y}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_x q_y}{h}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_y^2}{h} + \dfrac{g}{2}h^2\right) = -gh\dfrac{\partial z_b}{\partial y} + gh\dfrac{\partial \tilde{H}}{\partial y} + S_{f_2}(w), \\[2mm]
\dfrac{\partial z_b}{\partial t} + \xi\left(\dfrac{\partial q_{b,x}}{\partial x}(w) + \dfrac{\partial q_{b,y}}{\partial y}(w)\right) = 0
\end{cases}
$$

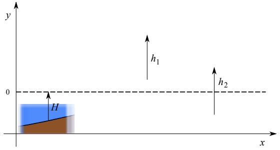
M. Castro Díaz, E. Fernández-Nieto, A. Ferreiro.
Sediment transport models in shallow water equations and numerical approach by high order finite volume methods, Computers & Fluids 37:299–316, 2008.

A. C. Fowler, N. Kopteva, C. Oakley.
The formation of river channels, SIAM Journal on Applied Mathematics 67:1016–1040, 2007.

S. Cordier, M. Le, T. Morales de Luna.
Bedload transport in shallow water models: Why splitting (may) fail, how hyperbolicity (can) help. Advances in Water Resources 34: 980–989, 2011.

Two-layer shallow-water model

$$\begin{cases}
\dfrac{\partial h_1}{\partial t} + \dfrac{\partial q_{1,x}}{\partial x} + \dfrac{\partial q_{1,y}}{\partial y} = 0 \\
\dfrac{\partial q_{1,x}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{1,x}^2}{h_1} + \dfrac{g}{2}h_1^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{1,x}q_{1,y}}{h_1}\right) = -gh_1\dfrac{\partial h_2}{\partial x} + gh_1\dfrac{\partial H}{\partial x} + S_{i_1}(w) \\
\dfrac{\partial q_{1,y}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{1,x}q_{1,y}}{h_1}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{1,y}^2}{h_1} + \dfrac{g}{2}h_1^2\right) = -gh_1\dfrac{\partial h_2}{\partial y} + gh_1\dfrac{\partial H}{\partial y} + S_{i_2}(w) \\
\dfrac{\partial h_2}{\partial t} + \dfrac{\partial q_{2,x}}{\partial x} + \dfrac{\partial q_{2,y}}{\partial y} = 0 \\
\dfrac{\partial q_{2,x}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{2,x}^2}{h_2} + \dfrac{g}{2}h_2^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{2,x}q_{2,y}}{h_2}\right) = -grh_2\dfrac{\partial h_1}{\partial x} + gh_2\dfrac{\partial H}{\partial x} + S_{i_3}(w) + S_{f_1}(w) \\
\dfrac{\partial q_{2,y}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{2,x}q_{2,y}}{h_2}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{2,y}^2}{h_2} + \dfrac{g}{2}h_2^2\right) = -grh_2\dfrac{\partial h_1}{\partial y} + gh_2\dfrac{\partial H}{\partial y} + S_{i_4}(w) + S_{f_2}(w),
\end{cases}$$



- $w = (h_1, q_{1,x}, q_{1,y}, h_2, q_{2,x}, q_{2,y})^T$;
- h_i, is the layer depth, $i = 1, 2$;
- $\boldsymbol{q}_i = (q_{i,x}, q_{i,y})$, is the mass flow at each layer, $i = 1, 2$;
- $r = \rho_1/\rho_2$ is the ratio of the constant densities of the layers $(\rho_1 < \rho_2)$;
- $S_{i_l}(w)$, $l = 1, \cdots, 4$, parametrize the friction terms between the two layers;
- $S_{f_l}(w)$, $l = 1, 2$ parametrize the friction terms between the second layer and the bottom topography.

Two-layer shallow-water model

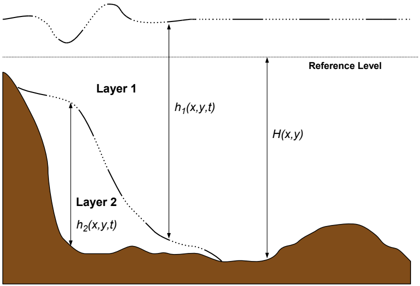
$$\begin{cases}
\dfrac{\partial h_1}{\partial t} + \dfrac{\partial q_{1,x}}{\partial x} + \dfrac{\partial q_{1,y}}{\partial y} = 0 \\
\dfrac{\partial q_{1,x}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{1,x}^2}{h_1} + \dfrac{g}{2}h_1^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{1,x}q_{1,y}}{h_1}\right) = -gh_1\dfrac{\partial h_2}{\partial x} + gh_1\dfrac{\partial H}{\partial x} + S_{i_1}(w) \\
\dfrac{\partial q_{1,y}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{1,x}q_{1,y}}{h_1}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{1,y}^2}{h_1} + \dfrac{g}{2}h_1^2\right) = -gh_1\dfrac{\partial h_2}{\partial y} + gh_1\dfrac{\partial H}{\partial y} + S_{i_2}(w) \\
\dfrac{\partial h_2}{\partial t} + \dfrac{\partial q_{2,x}}{\partial x} + \dfrac{\partial q_{2,y}}{\partial y} = 0 \\
\dfrac{\partial q_{2,x}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{2,x}^2}{h_2} + \dfrac{g}{2}h_2^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{2,x}q_{2,y}}{h_2}\right) = -grh_2\dfrac{\partial h_1}{\partial x} + gh_2\dfrac{\partial H}{\partial x} + S_{i_3}(w) + S_{f_1}(w) \\
\dfrac{\partial q_{2,y}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{2,x}q_{2,y}}{h_2}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{2,y}^2}{h_2} + \dfrac{g}{2}h_2^2\right) = -grh_2\dfrac{\partial h_1}{\partial y} + gh_2\dfrac{\partial H}{\partial y} + S_{i_4}(w) + S_{f_2}(w),
\end{cases}$$

- $$S_{i_1} = c(u_{2,x} - u_{1,x})\|\boldsymbol{u}_1 - \boldsymbol{u}_2\|,\; S_{i_2} = c(u_{2,y} - u_{1,y})\|\boldsymbol{u}_1 - \boldsymbol{u}_2\|,\; S_{i_3} = -rS_{i_1},\; S_{i_4} = -rS_{i_2}.$$
- $$S_{i_1} = c\frac{h_1h_2}{rh_1 + h_2}(u_{2,x} - u_{1,x})\|\boldsymbol{u}_1 - \boldsymbol{u}_2\|,\; S_{i_2} = c\frac{h_1h_2}{rh_1 + h_2}(u_{2,y} - u_{1,y})\|\boldsymbol{u}_1 - \boldsymbol{u}_2\|,\; S_{i_3} = -rS_{i_1},\; S_{i_4} = -rS_{i_2}.$$

2D (Simplified) Two-layer Savage-Hutter shallow-water model

Hypothesis

- We consider an stratified media composed by a non viscous and homogeneous fluid with constant density ρ_1 (water) and a granular material with density ρ_s and porosity ψ_0. We suppose the mean density of the granular material is given by: $\rho_2 = (1 - \psi_0)\rho_s + \psi_0\rho_1$.
- Fluid and granular material are immiscible.



2D (Simplified) Two-layer Savage-Hutter shallow-water model

$$
\begin{cases}
\dfrac{\partial h_1}{\partial t} + \dfrac{\partial q_{1,x}}{\partial x} + \dfrac{\partial q_{1,y}}{\partial y} = 0 \\[2mm]
\dfrac{\partial q_{1,x}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{1,x}^2}{h_1} + \dfrac{g}{2}h_1^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{1,x}q_{1,y}}{h_1}\right) = -gh_1\dfrac{\partial h_2}{\partial x} + gh_1\dfrac{\partial H}{\partial x} + S_{i_1}(w) \\[2mm]
\dfrac{\partial q_{1,y}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{1,x}q_{1,y}}{h_1}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{1,y}^2}{h_1} + \dfrac{g}{2}h_1^2\right) = -gh_1\dfrac{\partial h_2}{\partial y} + gh_1\dfrac{\partial H}{\partial y} + S_{i_2}(w) \\[2mm]
\dfrac{\partial h_2}{\partial t} + \dfrac{\partial q_{2,x}}{\partial x} + \dfrac{\partial q_{2,y}}{\partial y} = 0 \\[2mm]
\dfrac{\partial q_{2,x}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{2,x}^2}{h_2} + \dfrac{g}{2}h_2^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{2,x}q_{2,y}}{h_2}\right) = -grh_2\dfrac{\partial h_1}{\partial x} + gh_2\dfrac{\partial H}{\partial x} + S_{i_3}(w) + \tau_x(w) \\[2mm]
\dfrac{\partial q_{2,y}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{2,x}q_{2,y}}{h_2}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{2,y}^2}{h_2} + \dfrac{g}{2}h_2^2\right) = -grh_2\dfrac{\partial h_1}{\partial y} + gh_2\dfrac{\partial H}{\partial y} + S_{i_4}(w) + \tau_y(w),
\end{cases}
$$

- h_i, is the layer depth, $i = 1, 2$,
- $\boldsymbol{q}_i = (q_{i,x}, q_{i,y})$, is the mass flow at each layer, $i = 1, 2$,
- $w = (h_1, q_{1,x}, q_{1,y}, h_2, q_{2,x}, q_{2,y})^T$,
- $S_{i_l}(w)$, $l = 1, \ldots, 4$, parametrize the friction between the layers,
- $\tau(w) = (\tau_x(w), \tau_y(w))$ parametrize the Coulomb friction term.

2D (Simplified) Two-layer Savage-Hutter shallow-water model

$$
\begin{cases}
\dfrac{\partial h_1}{\partial t} + \dfrac{\partial q_{1,x}}{\partial x} + \dfrac{\partial q_{1,y}}{\partial y} = 0 \\
\dfrac{\partial q_{1,x}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{1,x}^2}{h_1} + \dfrac{g}{2}h_1^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{1,x}q_{1,y}}{h_1}\right) = -gh_1\dfrac{\partial h_2}{\partial x} + gh_1\dfrac{\partial H}{\partial x} + S_{i_1}(w) \\
\dfrac{\partial q_{1,y}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{1,x}q_{1,y}}{h_1}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{1,y}^2}{h_1} + \dfrac{g}{2}h_1^2\right) = -gh_1\dfrac{\partial h_2}{\partial y} + gh_1\dfrac{\partial H}{\partial y} + S_{i_2}(w) \\
\dfrac{\partial h_2}{\partial t} + \dfrac{\partial q_{2,x}}{\partial x} + \dfrac{\partial q_{2,y}}{\partial y} = 0 \\
\dfrac{\partial q_{2,x}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{2,x}^2}{h_2} + \dfrac{g}{2}h_2^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{2,x}q_{2,y}}{h_2}\right) = -grh_2\dfrac{\partial h_1}{\partial x} + gh_2\dfrac{\partial H}{\partial x} + S_{i_3}(w) + \tau_x(w) \\
\dfrac{\partial q_{2,y}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{2,x}q_{2,y}}{h_2}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{2,y}^2}{h_2} + \dfrac{g}{2}h_2^2\right) = -grh_2\dfrac{\partial h_1}{\partial y} + gh_2\dfrac{\partial H}{\partial y} + S_{i_4}(w) + \tau_y(w),
\end{cases}
$$

$$
\tau(w) = \begin{cases}
\left(gh_2(1-r)\frac{u_{2,x}}{\|\boldsymbol{u}\|}\tan(\delta_0), gh_2(1-r)\frac{u_{2,y}}{\|\boldsymbol{u}\|}\tan(\delta_0)\right) & \text{if } \tau > \sigma_c \\
\boldsymbol{u}_2 = \boldsymbol{0} & \text{otherwise}
\end{cases}
$$

2D (Simplified) Two-layer Savage-Hutter shallow-water model

$$\begin{cases}
\dfrac{\partial h_1}{\partial t} + \dfrac{\partial q_{1,x}}{\partial x} + \dfrac{\partial q_{1,y}}{\partial y} = 0 \\
\dfrac{\partial q_{1,x}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{1,x}^2}{h_1} + \dfrac{g}{2}h_1^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{1,x}q_{1,y}}{h_1}\right) = -gh_1\dfrac{\partial h_2}{\partial x} + gh_1\dfrac{\partial H}{\partial x} + S_{i_1}(w) \\
\dfrac{\partial q_{1,y}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{1,x}q_{1,y}}{h_1}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{1,y}^2}{h_1} + \dfrac{g}{2}h_1^2\right) = -gh_1\dfrac{\partial h_2}{\partial y} + gh_1\dfrac{\partial H}{\partial y} + S_{i_2}(w) \\
\dfrac{\partial h_2}{\partial t} + \dfrac{\partial q_{2,x}}{\partial x} + \dfrac{\partial q_{2,y}}{\partial y} = 0 \\
\dfrac{\partial q_{2,x}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{2,x}^2}{h_2} + \dfrac{g}{2}h_2^2\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{2,x}q_{2,y}}{h_2}\right) = -grh_2\dfrac{\partial h_1}{\partial x} + gh_2\dfrac{\partial H}{\partial x} + S_{i_3}(w) + \tau_x(w) \\
\dfrac{\partial q_{2,y}}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q_{2,x}q_{2,y}}{h_2}\right) + \dfrac{\partial}{\partial y}\left(\dfrac{q_{2,y}^2}{h_2} + \dfrac{g}{2}h_2^2\right) = -grh_2\dfrac{\partial h_1}{\partial y} + gh_2\dfrac{\partial H}{\partial y} + S_{i_4}(w) + \tau_y(w),
\end{cases}$$

E. Fernández Nieto, F. Bouchut, D. Bresch, M.J. Castro, A. Mangeney.

A new Savage-Hutter type model for submarine avalanches and generated tsunami. J. Comp. Phys., 227: 7720-7754, 2008.

F. Bouchut, M. Westdickenberg.

Gravity driven shallow water models for arbitrary topography. Comm. in Math. Sci. 2: 359-389, 2004

General framework

General formulation

$$\frac{\partial w}{\partial t} + \frac{\partial F_1}{\partial x}(w) + \frac{\partial F_2}{\partial y}(w) + B_1(w)\frac{\partial w}{\partial x} + B_2(w)\frac{\partial w}{\partial y} = S_1(w)\frac{\partial H}{\partial x} + S_2(w)\frac{\partial H}{\partial y} + S_F(w),$$

General framework

General formulation

$$\frac{\partial w}{\partial t} + \frac{\partial F_1}{\partial x}(w) + \frac{\partial F_2}{\partial y}(w) + B_1(w)\frac{\partial w}{\partial x} + B_2(w)\frac{\partial w}{\partial y} = S_1(w)\frac{\partial H}{\partial x} + S_2(w)\frac{\partial H}{\partial y} + S_F(w),$$

- $w(\boldsymbol{x}, t) : D \times (0, T) \mapsto \omega \subset \mathbb{R}^n$, the vector of unknowns,
- D bounded domain of $\mathbb{R}^2$; ω convex subset of $\mathbb{R}^n$,
- $F_i : \omega \mapsto \mathbb{R}^n$, $i = 1, 2$, regular and locally bounded functions,
- $B_i : \Omega \mapsto \mathcal{M}_{N\times N}(\mathbb{R})$, $i = 1, 2$ regular and locally bounded matrix-valued functions,
- $S_i, S_F : \omega \mapsto \mathbb{R}^n$, $i = 1, 2$, regular and locally bounded functions,
- $H : D \subset \mathbb{R}^2 : \ \mapsto \mathbb{R}$ known function.

General framework

General formulation

$$\frac{\partial w}{\partial t} + \frac{\partial F_1}{\partial x}(w) + \frac{\partial F_2}{\partial y}(w) + B_1(w)\frac{\partial w}{\partial x} + B_2(w)\frac{\partial w}{\partial y} = S_1(w)\frac{\partial H}{\partial x} + S_2(w)\frac{\partial H}{\partial y} + S_F(w),$$

Nonconservative products

The nonconservative products $B_i(w)\frac{\partial w}{\partial \alpha}$, $S_i(w)\frac{\partial w}{\partial \alpha}$, $i = 1, 2$, $\alpha = x, y$ do not make sense in general within the framework of distributions. Here, we follow the theory developed by [Dal Maso, LeFloch and Murat] to give a sense to these products as Borel measures. This theory is based on the choice of a family of paths.

General Formulation: Nonconservative Form

General formulation

$$\frac{\partial w}{\partial t} + \frac{\partial F_1}{\partial x}(w) + \frac{\partial F_2}{\partial y}(w) + B_1(w)\frac{\partial w}{\partial x} + B_2(w)\frac{\partial w}{\partial y} = S_1(w)\frac{\partial H}{\partial x} + S_2(w)\frac{\partial H}{\partial y} + S_F(w)$$

General Formulation: Nonconservative Form

General formulation

$$\frac{\partial w}{\partial t} + \frac{\partial F_1}{\partial x}(w) + \frac{\partial F_2}{\partial y}(w) + B_1(w)\frac{\partial w}{\partial x} + B_2(w)\frac{\partial w}{\partial y} = S_1(w)\frac{\partial H}{\partial x} + S_2(w)\frac{\partial H}{\partial y} + S_F(w)$$

[LeFloch] Introducing the trivial equation $\partial_t H = 0$ and taking H as a new unknown $W := [w\ H]^\top \in \mathbb{R}^N$, $N = n+1$, $\mathcal{J}_k(w) = \dfrac{\partial F_k}{\partial w}$ and

$$\mathcal{A}_k = \left[\begin{array}{c|c} \mathcal{J}_k(w) + B_k(w) & -S_k(w) \\ \hline 0 & 0 \end{array}\right], \; k = 1,2 \; \in \mathcal{M}_{N\times N},$$

General Formulation: Nonconservative Form

General formulation

$$\frac{\partial w}{\partial t} + \frac{\partial F_1}{\partial x}(w) + \frac{\partial F_2}{\partial y}(w) + B_1(w)\frac{\partial w}{\partial x} + B_2(w)\frac{\partial w}{\partial y} = S_1(w)\frac{\partial H}{\partial x} + S_2(w)\frac{\partial H}{\partial y} + S_F(w)$$

[LeFloch] Introducing the trivial equation $\partial_t H = 0$ and taking H as a new unknown $W := [w\ H]^\top \in \mathbb{R}^N$, $N = n+1$, $\mathcal{J}_k(w) = \dfrac{\partial F_k}{\partial w}$ and

$$\mathcal{A}_k = \left[\begin{array}{c|c} \mathcal{J}_k(w) + B_k(w) & -S_k(w) \\ \hline 0 & 0 \end{array}\right], \; k = 1,2 \; \in \mathcal{M}_{N\times N},$$

the system can be written as follows:

$$\frac{\partial W}{\partial t} + \mathcal{A}_1(W)\frac{\partial W}{\partial x} + \mathcal{A}_2(W)\frac{\partial W}{\partial y} = S_F(W).$$

General Formulation: Nonconservative form

Given a unitary vector $\boldsymbol{\eta} = (\eta_1, \eta_2) \in \mathbb{R}^2$:

$$\mathcal{A}(W, \boldsymbol{\eta}) = \mathcal{A}_1(W)\eta_1 + \mathcal{A}_2(W)\eta_2.$$

We assume that the system is strictly hyperbolic, i.e.,

General Formulation: Nonconservative form

Given a unitary vector $\boldsymbol{\eta} = (\eta_1, \eta_2) \in \mathbb{R}^2$:

$$\mathcal{A}(W, \boldsymbol{\eta}) = \mathcal{A}_1(W)\eta_1 + \mathcal{A}_2(W)\eta_2.$$

We assume that the system is strictly hyperbolic, i.e.,

- $\forall W \in \Omega \subset \mathbb{R}^N$ and $\forall\ \boldsymbol{\eta} \in S^1$, the matrix $\mathcal{A}(W, \boldsymbol{\eta})$ has N real eigenvalues: $\lambda_1(W, \boldsymbol{\eta}) < \ldots < \lambda_N(W, \boldsymbol{\eta})$, being $R_j(W, \boldsymbol{\eta}), j = 1, \ldots, N$ the associated eigenvectors.

General Formulation: Nonconservative form

Given a unitary vector $\boldsymbol{\eta} = (\eta_1, \eta_2) \in \mathbb{R}^2$:

$$\mathcal{A}(W, \boldsymbol{\eta}) = \mathcal{A}_1(W)\eta_1 + \mathcal{A}_2(W)\eta_2.$$

We assume that the system is strictly hyperbolic, i.e.,

- $\forall W \in \Omega \subset \mathbb{R}^N$ and $\forall\ \boldsymbol{\eta} \in S^1$, the matrix $\mathcal{A}(W, \boldsymbol{\eta})$ has N real eigenvalues: $\lambda_1(W, \boldsymbol{\eta}) < \ldots < \lambda_N(W, \boldsymbol{\eta})$, being $R_j(W, \boldsymbol{\eta}), j = 1, \ldots, N$ the associated eigenvectors.
- $\mathcal{A}(W, \boldsymbol{\eta})$ is diagonalizable.

$$\mathcal{A}(W, \boldsymbol{\eta}) = \mathcal{K}(W, \boldsymbol{\eta})\mathcal{L}(W, \boldsymbol{\eta})\mathcal{K}^{-1}(W, \boldsymbol{\eta}),$$

$\mathcal{L}(W, \boldsymbol{\eta})$ diagonal matrix whose coefficients are the eigenvalues of $\mathcal{A}(W, \boldsymbol{\eta})$,

$\mathcal{K}(W, \boldsymbol{\eta})$ matrix whose columns are the eigenvectors.

General Formulation: Nonconservative form

A family of paths (Dal Maso, LeFloch, Murat) in $\Omega \subset \mathbb{R}^N$ is a locally Lipschitz map

$$\Phi: \, [0,1] \times \Omega \times \Omega \times \mathcal{S}^1 \to \Omega,$$

where $\mathcal{S}^1 \subset \mathbb{R}^2$ denotes the unit sphere, that satisfies some regularity conditions and

1. $\Phi(0; W_L, W_R, \boldsymbol{\eta}) = W_L$ and $\Phi(1; W_L, W_R, \boldsymbol{\eta}) = W_R$, for any $W_L, W_R \in \Omega$, $\boldsymbol{\eta} \in \mathcal{S}^1$.
2. $\Phi(s; W_L, W_R, \boldsymbol{\eta}) = \Phi(1-s; W_R, W_L, -\boldsymbol{\eta})$, for any $W_L, W_R \in \Omega$, $s \in [0,1]$, $\boldsymbol{\eta} \in \mathcal{S}^1$.
3. $\Phi(s; W, W, \boldsymbol{\eta}) = W$, for any $s \in [0,1]$, $\boldsymbol{\eta} \in \mathcal{S}^1$.

General Formulation: Nonconservative form

A family of paths (Dal Maso, LeFloch, Murat) in $\Omega \subset \mathbb{R}^N$ is a locally Lipschitz map

$$\Phi\colon [0,1] \times \Omega \times \Omega \times \mathcal{S}^1 \to \Omega,$$

where $\mathcal{S}^1 \subset \mathbb{R}^2$ denotes the unit sphere, that satisfies some regularity conditions and

1. $\Phi(0; W_L, W_R, \boldsymbol{\eta}) = W_L$ and $\Phi(1; W_L, W_R, \boldsymbol{\eta}) = W_R$, for any $W_L, W_R \in \Omega$, $\boldsymbol{\eta} \in \mathcal{S}^1$.
2. $\Phi(s; W_L, W_R, \boldsymbol{\eta}) = \Phi(1-s; W_R, W_L, -\boldsymbol{\eta})$, for any $W_L, W_R \in \Omega$, $s \in [0,1]$, $\boldsymbol{\eta} \in \mathcal{S}^1$.
3. $\Phi(s; W, W, \boldsymbol{\eta}) = W$, for any $s \in [0,1]$, $\boldsymbol{\eta} \in \mathcal{S}^1$.

A piecewise regular function W is a weak solution if and only if the two following conditions are satisfied:

(i) W is a classical solution where it is smooth.

(ii) At every point of a discontinuity W satisfies the jump condition

$$\int_0^1 \left(\sigma\mathcal{I} - \mathcal{A}(\Phi(s; W^-, W^+, \boldsymbol{\eta}), \boldsymbol{\eta})\right) \frac{\partial \Phi}{\partial s}(s; W^-, W^+, \boldsymbol{\eta})\, ds = 0, \tag{1}$$

where $\mathcal{I}$ is the identity matrix; σ, the speed of propagation of the discontinuity; $\boldsymbol{\eta}$ a unit vector normal to the discontinuity at the considered point; and W^-, W^+, the lateral limits of the solution at the discontinuity.

General Formulation: Nonconservative form

A family of paths (Dal Maso, LeFloch, Murat) in $\Omega \subset \mathbb{R}^N$ is a locally Lipschitz map

$$\Phi\colon [0,1] \times \Omega \times \Omega \times \mathcal{S}^1 \to \Omega,$$

where $\mathcal{S}^1 \subset \mathbb{R}^2$ denotes the unit sphere, that satisfies some regularity conditions and

1. $\Phi(0; W_L, W_R, \boldsymbol{\eta}) = W_L$ and $\Phi(1; W_L, W_R, \boldsymbol{\eta}) = W_R$, for any $W_L, W_R \in \Omega$, $\boldsymbol{\eta} \in \mathcal{S}^1$.
2. $\Phi(s; W_L, W_R, \boldsymbol{\eta}) = \Phi(1-s; W_R, W_L, -\boldsymbol{\eta})$, for any $W_L, W_R \in \Omega$, $s \in [0,1]$, $\boldsymbol{\eta} \in \mathcal{S}^1$.
3. $\Phi(s; W, W, \boldsymbol{\eta}) = W$, for any $s \in [0,1]$, $\boldsymbol{\eta} \in \mathcal{S}^1$.

- The choice of the family of paths is important because it determines the speed of propagation of discontinuities.
- It has to be based on the physical background of the problem: limit of viscous profiles ..., but in practice can be very difficult.
- From the mathematical point of view, some hypotheses concerning the relation of the paths with the integral curves of the characteristic fields can be imposed.

High order numerical schemes

Goal

Develop high order finite volume schemes for problems that can be written under the form:

$$\frac{\partial W}{\partial t} + \mathcal{A}_1(W)\frac{\partial W}{\partial x} + \mathcal{A}_2(W)\frac{\partial W}{\partial y} = 0,$$

High order numerical schemes

Goal

Develop high order finite volume schemes for problems that can be written under the form:

$$\frac{\partial W}{\partial t} + \mathcal{A}_1(W)\frac{\partial W}{\partial x} + \mathcal{A}_2(W)\frac{\partial W}{\partial y} = 0,$$

Properties:

- Shock capturing property: the solutions of this kind of systems may generate discontinuities in finite time, even for very regular initial conditions. The appearance and subsequent propagation of these shocks has to be correctly simulated.
- High accuracy in areas where the solution is regular.
- Well-balanced property: in cases where regular non-trivial stationary solutions exist, the schemes must be able to solve these solutions (or some of them) exactly or at least with an enhanced accuracy.

High order numerical schemes

Goal

Develop high order finite volume schemes for problems that can be written under the form:

$$\frac{\partial W}{\partial t} + \mathcal{A}_1(W)\frac{\partial W}{\partial x} + \mathcal{A}_2(W)\frac{\partial W}{\partial y} = 0,$$

Basic idea

- Split the computational domain into subsets of simple geometry called cells or finite volumes.
- Define a reconstruction operator of the unknowns W on each cell.
- Combine with your favourite first order finite volume scheme for nonconservative systems (path-conservative schemes).

Notation

- D is decomposed into subsets (closed polygons) called cells of finite volumes, $V_i \subset \mathbb{R}^2$;
- $\mathcal{N}_i$ is the set of indexes j such that V_j is a neighbor of V_i;
- E_{ij} is the common edge to two neighbor cells V_i and V_j, and $|E_{ij}|$ represents its length;
- $\boldsymbol{\eta}_{ij} = (\eta_{ij,x}, \eta_{ij,y})$ is the normal unit vector of the edge E_{ij} pointing towards the cell V_j;
- Δx is the maximum of the diameters of the cells;
- W_i^n will represent the constant approximation of the averaged solution in the cell V_i at time t^n provided by the numerical scheme:

$$W_i^n \cong \frac{1}{|V_i|} \int_{V_i} W(\mathbf{x}, t^n) d\mathbf{x}.$$

High order reconstruction operators

- P_i: Reconstruction operator over cell V_i such as, given the averages of W at a given stencil of cell V_i, P_i provides high order point values approximation of W at the cell V_i.

High order reconstruction operators

- P_i: Reconstruction operator over cell V_i such as, given the averages of W at a given stencil of cell V_i, P_i provides high order point values approximation of W at the cell V_i.
  - P_i depends on a family of values $\{W_j\}_{j\in\mathcal{B}_i}$, where $\mathcal{B}_i = \{\text{Set of indexes of neighbor cells of } V_i\}$.

High order reconstruction operators

- P_i: Reconstruction operator over cell V_i such as, given the averages of W at a given stencil of cell V_i, P_i provides high order point values approximation of W at the cell V_i.
  - P_i depends on a family of values $\{W_j\}_{j\in\mathcal{B}_i}$, where $\mathcal{B}_i = \{\text{Set of indexes of neighbor cells of } V_i\}$.
  - If P_i depends on time we note P_i^t

High order reconstruction operators

- P_i: Reconstruction operator over cell V_i such as, given the averages of W at a given stencil of cell V_i, P_i provides high order point values approximation of W at the cell V_i.
    - P_i depends on a family of values $\{W_j\}_{j\in\mathcal{B}_i}$, where $\mathcal{B}_i = \{\text{Set of indexes of neighbor cells of } V_i\}$.
    - If P_i depends on time we note P_i^t
- If $\mathbf{s} \in E_{ij}$, $W_{ij}^-(\mathbf{s},t)$, $W_{ij}^+(\mathbf{s},t)$, are the limit of P_i^t (P_j^t) when $\boldsymbol{x}$ tends to $\mathbf{s}$ through V_i (V_j):

$$\lim_{\substack{\boldsymbol{x}\to\mathbf{s}\\ \boldsymbol{x}\cdot\boldsymbol{\eta}_{ij}<k_{ij}}} P_i^t(\boldsymbol{x}) = W_{ij}^-(\mathbf{s},t), \qquad \lim_{\substack{\boldsymbol{x}\to\mathbf{s}\\ \boldsymbol{x}\cdot\eta_{ij}>k_{ij}}} P_j^t(\boldsymbol{x}) = W_{ij}^+(\boldsymbol{s},t).$$

High order reconstruction operators

H1 It is conservative, i.e.

$$W_i = \frac{1}{|V_i|}\int_{V_i} P_i(x)dx$$

.

H2 It is of order p at the edges of the cell.

H3 Order q inside V_i, i.e.,

$$P_i(\boldsymbol{x}) = W(\boldsymbol{x}) + O(\Delta x^q),\ \forall \boldsymbol{x} \in \ int(V_i)$$

.

H4 ∇P_i approximation of order m of the gradient of ∇W inside V_i.

High order state reconstruction operators: examples

H. J. Schroll and F. Svensson.
A Bihyperbolic Finite Volume Method for Quadrilateral Meshes. SIAM: J. Sci. Comput., 26(2):237–260, 2006.

M. Dumbser, M. Käser.
Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. J. Comput. Phys. 221, 693-723,2007.

José M. Gallardo, Sergio Ortega, Marc de la Asunción and José Miguel Mantas
Two-dimensional compact third-order polynomial reconstructions. Solving nonconservative hyperbolic systems using GPUs. J. Sci. Comput., 2011, DOI: 10.1007/s10915-011-9470-x.

Third order compact polynomial reconstruction operator

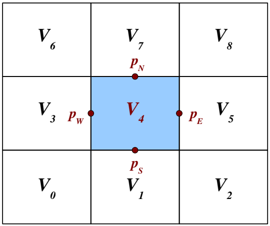
Properties:

- Third order of accuracy on the whole cell $\Rightarrow$ third order scheme.
- Compact stencil (nine cells) $\Rightarrow$ robustness.
- Low computational cost.
- Easy to implement.

The method can be applied on non-uniform quadrilateral meshes.

Idea:

- Reconstruction on the central cell: V_4.
- A second order polynomial $P_R(\mathbf{x})$ is associated to each point p_R, for $R \in \{N, S, E, W\}$.
- The polynomials $P_R(\mathbf{x})$ are then combined using a WENO procedure.
- The result is a third order conservative reconstruction operator on the cell V_4.



Third order compact polynomial reconstructions: procedure

Let $F(\mathbf{x})$ be the function to be reconstructed on the central cell V.

For $R \in \{N, S, E, W\}$:

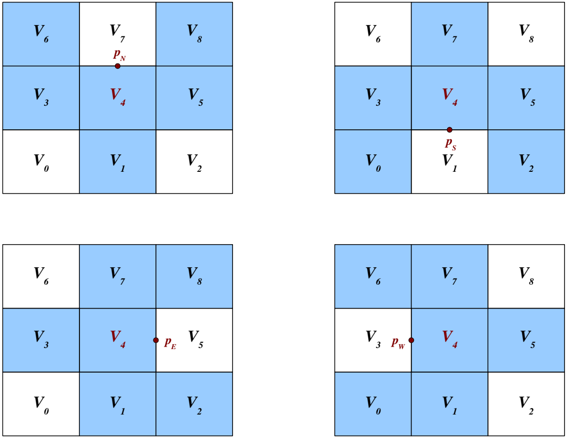
- Consider a second-order Taylor expansion of $F(\mathbf{x})$ around the point p_R.
- Approximate the coefficients (i.e., the derivatives of $F(\mathbf{x})$) in such a way that the corresponding polynomial $P_R(\mathbf{x})$ provides a third order approximation to $F(\mathbf{x})$ on V and it is conservative.

Remarks:

- For general non-uniform quadrilateral meshes, the approximations to the derivatives can be obtained by solving linear systems that only depend on the points p_R and the mesh; thus, they can be calculated and stored in a pre-processing step.
- The choice of appropriate stencils is fundamental to assure that each linear system possesses an unique solution.

Third order compact polynomial reconstructions: procedure

Stencils used to approximate the second order derivatives:



Third order compact polynomial reconstructions: procedure

- For $R \in \{N, S, E, W\}$, build the smoothness indicators

$$\beta_R = \iint_V \Delta x \Delta y \left([\partial_y P_R(\mathbf{x})]^2 + [\partial_y P_R(\mathbf{x})]^2 \right) d\mathbf{x}.$$

- If some $\beta_R >$ tol $= \Delta^2$ then redefine $P_R(\mathbf{x})$ to be a linear function providing a second order conservative approximation to $F(\mathbf{x})$ on V.
- Apply the WENO idea: define the nonlinear weights ω_R as

$$\omega_R = \frac{\alpha_R}{\alpha_N + \alpha_S + \alpha_E + \alpha_W}, \qquad \alpha_R = \frac{\lambda_R}{(\varepsilon + \beta_R)^r}.$$

  The linear weights λ_R are given the value $1/4$, as they are not chosen to improve the accuracy. Typically, $\varepsilon = 10^{-5}$ and $r = 4$.
- Finally, define the reconstruction operator $P(\mathbf{x})$ as

$$P(\mathbf{x}) = \omega_N \, P_N(\mathbf{x}) + \omega_S \, P_S(\mathbf{x}) + \omega_E \, P_E(\mathbf{x}) + \omega_W \, P_W(\mathbf{x})$$

High Order 2D scheme: conservative systems

Let us consider the conservative system

$$\frac{\partial W}{\partial t} + \frac{\partial F_1}{\partial x}(W) + \frac{\partial F_2}{\partial y}(W) = 0,$$

and let $\overline{W}_i(t)$ denotes the cell average over the cell V_i at time t.

The equation satisfies by $\overline{W}_i(t)$ is the following

$$\overline{W}_i'(t) = -\frac{1}{|V_i|}\left(\sum_{j\in\mathcal{N}_i}\int_{E_{ij}} F(W(\gamma,t))\cdot\boldsymbol{\eta}_{ij}\,d\gamma\right). \qquad (1)$$

High Order 2D scheme: conservative systems

A first order method and a reconstruction operator is used to approach the values of fluxes at the edges:

$$W_i'(t) = -\frac{1}{|V_i|}\left(\sum_{j\in\mathcal{N}_i}\int_{E_{ij}} G(W_{ij}^-(\gamma,t), W_{ij}^+(\gamma,t), \boldsymbol{\eta}_{ij})\, d\gamma\right), \qquad (1)$$

being $W_i(t)$ the approximation to $\overline{W}_i(t)$ and $W_{ij}^{\pm}(\gamma,t)$ the reconstruction at $\gamma \in E_{ij}$.

High Order 2D scheme: conservative systems

Using the divergence theorem, the previous semi-discrete numerical scheme can be rewritten as follows:

$$
\begin{aligned}
W_i'(t) &= -\frac{1}{|V_i|}\sum_{j\in\mathcal{N}_i}\int_{E_{ij}}\Big(G(W_{ij}^-(\gamma,t),W_{ij}^+(\gamma,t),\boldsymbol{\eta}_{ij}) - F(W_{ij}^-(\gamma,t))\cdot\boldsymbol{\eta}_{ij}\Big)\,d\gamma \\
&\quad -\frac{1}{|V_i|}\sum_{j\in\mathcal{N}_i}\int_{E_{ij}}\Big(F(W_{ij}^-(\gamma,t))\cdot\boldsymbol{\eta}_{ij}\Big)\,d\gamma \\
&= -\frac{1}{|V_i|}\sum_{j\in\mathcal{N}_i}\int_{E_{ij}}\Big(G(W_{ij}^-(\gamma,t),W_{ij}^+(\gamma,t),\boldsymbol{\eta}_{ij}) - F(W_{ij}^-(\gamma,t))\cdot\boldsymbol{\eta}_{ij}\Big)\,d\gamma \\
&\quad -\frac{1}{|V_i|}\int_{V_i}\nabla\cdot(F\circ P_i^t)(\mathbf{x}))\,d\mathbf{x} \\
&= -\frac{1}{|V_i|}\sum_{j\in\mathcal{N}_i}\int_{E_{ij}}\Big(G(W_{ij}^-(\gamma,t),W_{ij}^+(\gamma,t),\boldsymbol{\eta}_{ij}) - F(W_{ij}^-(\gamma,t))\cdot\boldsymbol{\eta}_{ij}\Big)\,d\gamma \\
&\quad -\frac{1}{|V_i|}\int_{V_i}\left(\mathcal{J}_1(P_i^t(\mathbf{x}))\frac{\partial P_i^t}{\partial x}(\mathbf{x}) + \mathcal{J}_2(P_i^t(\mathbf{x}))\frac{\partial P_i^t}{\partial y}(\mathbf{x})\right)dx,
\end{aligned}
$$

High Order 2D scheme for nonconservative systems

$$W_i'(t) = -\frac{1}{|V_i|}\left[\sum_{j\in\mathcal{N}_i}\int_{E_{ij}}\mathcal{D}_{ij}^{-}(W_{ij}^{-}(\gamma,t),W_{ij}^{+}(\gamma,t),\boldsymbol{\eta}_{ij})d\gamma \right.$$
$$\left. + \int_{V_i}\left(\mathcal{A}_1(P_i^t(\boldsymbol{x}))\frac{\partial P_i^t}{\partial x}(\boldsymbol{x}) + \mathcal{A}_2(P_i^t(\boldsymbol{x}))\frac{\partial P_i^t}{\partial y}(\boldsymbol{x})\right)d\boldsymbol{x}\right]$$

where $\mathcal{D}_{ij}^{\pm}(W_{ij}^{-}(\gamma,t),W_{ij}^{+}(\gamma,t),\boldsymbol{\eta}_{ij})$ is a path-conservative first order finite volume scheme.

In order to obtain a fully discrete numerical scheme:

- Use a quadrature formulae to approximate the integrals.
- Use a high order TVD Runge-Kutta numerical scheme for time integration.

M. Castro, E.D. Fernández, A. Ferreiro, J.A. García and C. Parés.
High order extensions of Roe schemes for two dimensional nonconservative hyperbolic systems. J. Sci. Comput., 39: 67-114, 2009.

Path-conservative schemes

We consider Path-conservative schemes in the sense introduced by Parés, 2006

- $$\mathcal{D}^{\pm}\big(W_L, W_R, \boldsymbol{\eta}\big),$$
  where $\mathcal{D}^-$ and $\mathcal{D}^+$ are two Lipschitz continuous functions from $\Omega \times \Omega \times \mathcal{S}^1$ to Ω satisfying:
  $$\mathcal{D}^{\pm}(W, W, \boldsymbol{\eta}) = 0, \quad \forall W \in \Omega, \forall \boldsymbol{\eta} \in \mathcal{S}^1, \tag{1}$$
  and
- for every $W_L, W_R \in \Omega$, and $\boldsymbol{\eta} \in \mathcal{S}^1$
  $$\mathcal{D}^-(W_L, W_R, \boldsymbol{\eta}) + \mathcal{D}^+(W_L, W_R, \boldsymbol{\eta}) = \int_0^1 \mathcal{A}\big(\Phi(s; W_L, W_R, \boldsymbol{\eta}), \boldsymbol{\eta}\big) \frac{\partial \Phi}{\partial s}(s; W_L, W_R, \boldsymbol{\eta})\, ds.$$

Path Conservative schemes: Convergence

- In Castro, LeFloch, Muñoz and Parés, 2008 and Parés-Muñoz, 2009 has been proved that the numerical solutions provided by finite difference/volumes path-conservative numerical scheme converge to functions which solve a perturbed system in which an error source-term appear on the right hand side (which is a measure supported on the discontinuities). This problem is common to any numerical scheme that introduces numerical diffusion.
- In order to overcome this difficulty, one can use viscosity-free methods based on Random Choice, as Glimm's method or control de entropy production of the numerical scheme: LeFloch, Coquel, Berthon, Entropy stable path-conservative schemes (SINUM 2012).
- In Muñoz-Parés, 2010 is shown that in certain situations this error vanishes for finite difference/volumes methods: this is the case of systems of balance laws. Moreover, in Castro et all, 2012 (M2AN) is shown that given any jump conditions related to a particular choice of paths then, it gives a third order approximation of the physically correct ones.
- In N. Chalmers and E. Lorin, 2012 is shown that if the numerical viscosity matrix commutes with $\mathcal{A}$, then shock curves associated to different numerical schemes are closed among them.

1D Model problem

Lets consider the system

$$w_t + F(w)_x + B(w) \cdot w_x = S(w)H_x, \tag{2}$$

- $w(x,t)$ takes values on an open convex set $\mathcal{O} \subset \mathbb{R}^N$,
- F is a regular function from $\mathcal{O}$ to $\mathbb{R}^N$,
- B is a regular matrix function from $\mathcal{O}$ to $\mathcal{M}_{N\times N}(\mathbb{R})$,
- S is a function from $\mathcal{O}$ to $\mathbb{R}^N$, and
- H is a function from $\mathbb{R}$ to $\mathbb{R}$.

By adding to (2) the equation $H_t = 0$, the system (2) can be rewritten under the form

$$W_t + \mathcal{A}(W) \cdot W_x = 0, \tag{3}$$

W is the augmented vector

$$W = \begin{bmatrix} w \\ H \end{bmatrix} \in \Omega = \mathcal{O} \times \mathbb{R} \subset \mathbb{R}^{N+1}$$

1D Model problem

Lets consider the system

$$w_t + F(w)_x + B(w) \cdot w_x = S(w) H_x, \tag{2}$$

- $w(x,t)$ takes values on an open convex set $\mathcal{O} \subset \mathbb{R}^N$,
- F is a regular function from $\mathcal{O}$ to $\mathbb{R}^N$,
- B is a regular matrix function from $\mathcal{O}$ to $\mathcal{M}_{N\times N}(\mathbb{R})$,
- S is a function from $\mathcal{O}$ to $\mathbb{R}^N$, and
- H is a function from $\mathbb{R}$ to $\mathbb{R}$.

By adding to (2) the equation $H_t = 0$, the system (2) can be rewritten under the form

$$W_t + \mathcal{A}(W) \cdot W_x = 0, \tag{3}$$

$\mathcal{A}(W)$ is the matrix whose block structure is given by:

$$\mathcal{A}(W) = \left[\begin{array}{c|c} A(w) & -S(w) \\ \hline 0 & 0 \end{array}\right],$$

$$A(w) = J(w) + B(w), \quad \text{being } J(w) = \frac{\partial F}{\partial w}(w).$$

Roe linearization I

Here, we consider numerical schemes defined using a generalized Roe matrix for (3) as defined by Toumi 1992:
Given a family of paths $\Phi = [\Phi_w, \Phi_H]^T$, a function $\mathcal{A}_\Phi : \Omega \times \Omega \mapsto \mathcal{M}_{(N+1)\times(N+1)}(\mathbb{R})$ is called a Roe linearization if it verifies the following properties:

- for any $W_L, W_R \in \Omega$, $\mathcal{A}_\Phi(W_L, W_R)$ has $N+1$ distinct real eigenvalues,
- for every $W \in \Omega$,
$$\mathcal{A}_\Phi(W, W) = \mathcal{A}(W); \tag{4}$$
- for any $W_L, W_R \in \Omega$,

$$\mathcal{A}_\Phi(W_L, W_R) \cdot (W_R - W_L) = \int_0^1 \mathcal{A}(\Phi(s; W_L, W_R)) \frac{\partial \Phi}{\partial s}(s; W_L, W_R)\, ds. \tag{5}$$

Roe-based schemes I

The following Roe linearizations $\mathcal{A}_\Phi(W_L, W_R)$ for system (2) are considered (Parés-Castro 2004):

$$\mathcal{A}_\Phi(W_L, W_R) = \left[\begin{array}{c|c} A_\Phi(w_L, w_R) & -S_\Phi(w_L, w_R) \\ \hline 0 & 0 \end{array}\right],$$

where

$$A_\Phi(w_L, w_R) = J(w_L, w_R) + B_\Phi(w_L, w_R).$$

Here, $J(w_L, w_R)$ is a Roe matrix of the Jacobian of the flux F in the usual sense:

$$J(w_L, w_R) \cdot (w_R - w_L) = F(w_R) - F(w_L);$$

$$B_\Phi(w_L, w_R) \cdot (w_R - w_L) = \int_0^1 B(\Phi_w(s; W_L, W_R)) \frac{\partial \Phi_w}{\partial s}(s; W_L, W_R)\, ds;$$

$$S_\Phi(w_L, w_R)(H_R - H_L) = \int_0^1 S(\Phi_w(s; W_L, W_R)) \frac{\partial \Phi_H}{\partial s}(s; W_L, W_R)\, ds.$$

It can be easily shown that, the resulting matrix is a Roe linearization provided it has $N+1$ different real eigenvalues.

Roe-based schemes II

Once the Roe linearization has been chosen, $\mathcal{D}^{\pm}_{i+1/2}$ can be defined:

$$\mathcal{D}^{\pm}_{i+1/2} = \widehat{\mathcal{A}}^{\pm}_{\Phi}(W_i^n, W_{i+1}^n) \cdot (W_{i+1}^n - W_i^n),$$

being

$$\mathcal{A}_{\Phi}(W_L, W_R) = \widehat{\mathcal{A}}^{+}_{\Phi}(W_L, W_R) + \widehat{\mathcal{A}}^{-}_{\Phi}(W_L, W_R)$$

is any decomposition of the Roe linearization of the form:

$$\widehat{\mathcal{A}}^{\pm}_{\Phi}(W_L, W_R) = \frac{1}{2}\left(\mathcal{A}_{\Phi}(W_L, W_R) \pm \mathcal{Q}_{\Phi}(W_L, W_R)\right),$$

where $\mathcal{Q}_{\Phi}(W_L, W_R)$ can be interpreted as a numerical viscosity matrix.

Roe-based schemes III

$\mathcal{D}^{\pm}_{i+1/2}$ can be re-written in terms of w, F, B, S and H as follows

$$D^{\pm}_{i+1/2} = \frac{1}{2}\Big(F(w_{i+1}) - F(w_i) + B_{i+1/2}(w_{i+1} - w_i) - S_{i+1/2}(H_{i+1} - H_i) \pm Q_{i+1/2}(w_{i+1} - w_i - A^{-1}_{i+1/2}S_{i+1/2}(H_{i+1} - H_i))\Big), \quad (6)$$

being

- $B_{i+1/2} = B_\Phi(W_i, W_{i+1})$,
- $S_{i+1/2} = S_\Phi(W_i, W_{i+1})$,
- $A_{i+1/2} = A_\Phi(W_i, W_{i+1})$ and
- $Q_{i+1/2} = Q_\Phi(W_i, W_{i+1})$ a numerical viscosity matrix obtained from $\mathcal{Q}_\Phi(W_i, W_{i+1})$

Roe-based schemes III

$\mathcal{D}^{\pm}_{i+1/2}$ can be re-written in terms of w, F, B, S and H as follows

$$\begin{aligned} D^{\pm}_{i+1/2} = \quad & \frac{1}{2}\Big(F(w_{i+1}) - F(w_i) + B_{i+1/2}(w_{i+1} - w_i) - S_{i+1/2}(H_{i+1} - H_i) \\ & \pm Q_{i+1/2}(w_{i+1} - w_i - A^{-1}_{i+1/2}S_{i+1/2}(H_{i+1} - H_i))\Big), \end{aligned} \tag{6}$$

Conservative systems

If the system is conservative and

$$\mathcal{F}_{i+1/2} = \frac{F(w_i) + F(w_{i+1})}{2} - \frac{1}{2}Q_{i+1/2}(w_{i+1} - w_i)$$

is a conservative flux, where $Q_{i+1/2}$ is defined in terms of $J_{i+1/2}$, then

$$D^{-}_{i+1/2} = \mathcal{F}_{i+1/2} - F(w_i) \quad D^{+}_{i+1/2} = F(w_{i+1}) - \mathcal{F}_{i+1/2}.$$

Roe-based schemes III

$\mathcal{D}^{\pm}_{i+1/2}$ can be re-written in terms of w, F, B, S and H as follows

$$D^{\pm}_{i+1/2} = \frac{1}{2}\Big(F(w_{i+1}) - F(w_i) + B_{i+1/2}(w_{i+1} - w_i) - S_{i+1/2}(H_{i+1} - H_i) \\ \pm Q_{i+1/2}(w_{i+1} - w_i - A^{-1}_{i+1/2}S_{i+1/2}(H_{i+1} - H_i))\Big)\,, \tag{6}$$

Different numerical schemes can be obtained for different definitions of $Q_{i+1/2}$

Roe-based schemes IV

- Roe scheme corresponds to the choice

$$Q_\Phi(W_L, W_R) = |A_\Phi(W_L, W_R)|,$$

- Lax-Friedrichs scheme:

$$Q_\Phi(W_L, W_R) = \frac{\Delta x}{\Delta t} Id,$$

  being Id the identity matrix.

- Lax-Wendroff scheme:

$$Q_\Phi(W_L, W_R) = \frac{\Delta t}{\Delta x} A_\Phi^2(W_L, W_R),$$

- FORCE and GFORCE schemes are presented in the bibliography as a convex combination of Lax-Friedrichs and Lax-Wendroff scheme:

$$Q_\Phi(W_L, W_R) = (1-\omega)\frac{\Delta x}{\Delta t} Id + \omega \frac{\Delta t}{\Delta x} A_\Phi^2(W_L, W_R),$$

  with $\omega = 0.5$ and $\omega = \frac{1}{1+\alpha}$, respectively, being α the CFL parameter.

Roe-based schemes V

We propose a class of finite volume methods defined by

$$Q_{i+1/2} = f(A_{i+1/2}),$$

where $f : \mathbb{R} \mapsto \mathbb{R}$ and $A_{i+1/2}$ a Roe matrix.

$Q_{i+1/2}$ has the same eigenvectors than $A_{i+1/2}$ and if $\lambda_{i+1/2,l}$ is an eigenvalue of $A_{i+1/2}$, then $f(\lambda_{i+1/2,l})$ is an eigenvalue of $Q_{i+1/2}$.

Some properties of f

- $f(x) \geq 0$ and smooth.
- $f(A_{i+1/2})$ should be easy to evaluate: no spectral decomposition of $A_{i+1/2}$ must be performed,
- L^∞ linear stability:
$$\nu \frac{\Delta x}{\Delta t} \geq f(\lambda_{i+1/2,l}) \geq |\lambda_{i+1/2,l}|,$$
for all $\lambda_{i+1/2,l}$ eigenvalue of $A_{i+1/2}$.
- $f(x)$ should be as close as possible to $|x|$.

PVM methods

One possible choice is to set

$$f(x) = P_l(x),$$

being $P_l(x)$ a polynomial of degree l,

$$P_l(x) = \sum_{j=0}^{l} \alpha_j^{i+1/2} x^j.$$

In this case

$$Q_{i+1/2} = P_l(A_{i+1/2}).$$

That is, $Q_{i+1/2}$ is a Polynomial Viscosity Matrix (PVM). See Castro-Fernández 2012.

Some well-known solvers as Lax-Friedrichs, Rusanov, FORCE/GFORCE, HLL, Roe, Lax-Wendroff, ... can be recovered as PVM methods

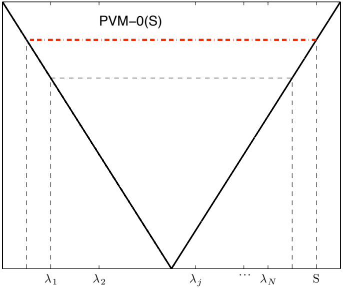
We use the following notation: PVM-$l(S_0, \cdots, S_k)$, where the parameters $S_0, \cdots, S_k$ will be related to the approximations of some wave speeds.

PVM-0(S_0) methods: Rusanov, Lax-Friedrichs and modified Lax-Friedrichs schemes

$$P_0(x) = S_0,$$

$$S_0 \in \{S_{Rus}, S_{LF}, S_{LF}^{mod}\}, \quad S_{Rus} = \max_j |\lambda_{j,i+1/2}|, \; S_{LF} = \frac{\Delta x}{\Delta t} \text{ and } S_{LF}^{mod} = \alpha \frac{\Delta x}{\Delta t}.$$

- Rusanov scheme corresponds to the choice $S_0 = S_{Rus}$,
- Lax-Friedrichs with $S_0 = S_{LF}$
- modified Lax-Friedrichs with $S_0 = S_{LF}^{mod}$.



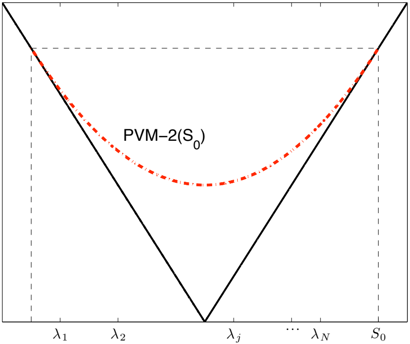
PVM-2(S_0) methods or FORCE type methods

$$P_2(x) = \alpha_0 + \alpha_2 x^2, \text{ such as } P_2(S_0) = S_0,\; P_2'(S_0) = 1,\; S_0 \in \{S_{Rus}, S_{LF}, S_{LF}^{mod}\}.$$

Remarks

- If $S_0 = S_{LF}$ then we obtain FORCE method.
- GFORCE scheme can be obtained by imposing

$$P_2(S_{LF}^{mod}) = S_{LF}^{mod}, \quad P_2'(S_{LF}^{mod}) = \frac{2\alpha}{1+\alpha},$$

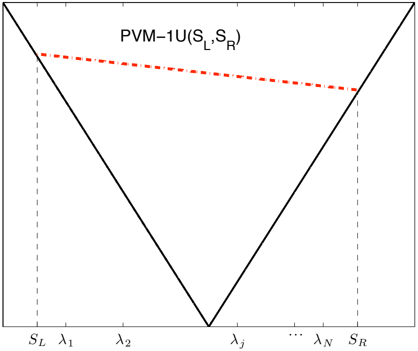


PVM-1U(S_L, S_R) or HLL method

Let S_L (respectively S_R) be an approximation of the minimum (respectively maximum) wave speed, we consider de polynomial

$$P_1(x) = \alpha_0 + \alpha_1\, x \quad \text{such as } P_1(S_L) = |S_L|,\ P_1(S_R) = |S_R|.$$

$$Q_{i+1/2} = \alpha_0 Id + \alpha_1 A_{i+1/2}$$



PVM-1U(S_L, S_R) or HLL method

The usual HLL scheme coincides with PVM-1U(S_L, S_R) in the case of conservative systems.
Let us suppose that the system is conservative. Then, the conservative flux associated to PVM-1U(S_L,S_R) is $\mathcal{F}_{i+1/2} = D^-_{i+1/2} + F(w_i)$. Taking into account that

$$\alpha_0 = \frac{S_R|S_L| - S_L|S_R|}{S_R - S_L}, \quad \alpha_1 = \frac{|S_R| - |S_L|}{S_R - S_L},$$

then

$$\begin{aligned}\mathcal{F}_{i+1/2} &= \frac{F(w_i)(S_R + |S_R| - S_L - |S_L|) + F(w_{i+1})(S_R - |S_R| - S_L + |S_L|)}{2S_R - 2S_L} \\ &\quad - \frac{(S_R|S_L| - S_L|S_R|)(w_{i+1} - w_i)}{2S_R - 2S_L} \\ &= \frac{S_R^+ F(w_i) - S_L^- F(w_{i+1}) + (S_R^+ S_L^-)(w_{i+1} - w_i)}{S_R^+ - S_L^-}\end{aligned}$$

which is a compact definition of the HLL flux, being $S_R^+ = \max(S_R, 0)$ and $S_L^- = \min(S_L, 0)$.

PVM-2U(S_L, S_R) method

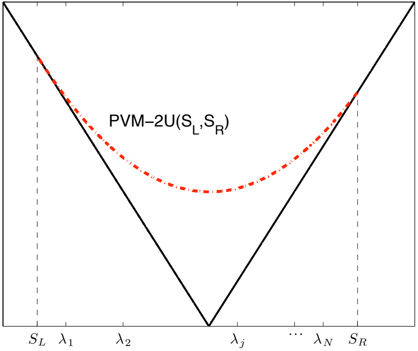
$$P_2(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2,$$

such as

$$P_2(S_m) = |S_m|, \; P_2(S_M) = |S_M|, \; P_2'(S_M) = \text{sgn}(S_M),$$

where

$$S_M = \begin{cases} S_L & \text{if } |S_L| \geq |S_R|, \\ S_R & \text{if } |S_L| < |S_R|. \end{cases} \quad S_m = \begin{cases} S_R & \text{if } |S_L| \geq |S_R|, \\ S_L & \text{if } |S_L| < |S_R|. \end{cases}$$



PVM-2U(S_L, S_R, S_{int}) method or IFCP method

$$P_2(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2,$$

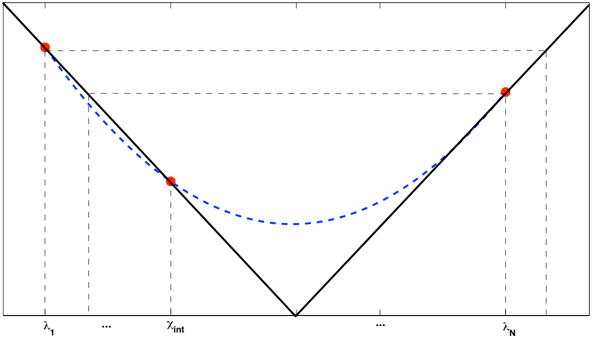
$$\begin{pmatrix} 1 & S_L & S_L^2 \\ 1 & S_R & S_R^2 \\ 1 & S_{int} & S_{int}^2 \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} |S_L| \\ |S_R| \\ |S_{int}| \end{pmatrix},$$

S_L (respectively S_R) is an approximation of the minimum (respectively maximum) wave speed and

$$S_{int} = \mathcal{S}_{ext} \max(|S_2|, \ldots, |S_{N-1}|),$$

$$\mathcal{S}_{ext} = \begin{cases} \text{sgn}(S_L + S_R), & \text{if } (S_L + S_R) \neq 0, \\ 1, & \text{otherwise.} \end{cases}$$

PVM-2U(S_L, S_R, S_{int}) method or IFCP method



PVM-(N-1)($\lambda_1, \cdots, \lambda_N$) or Roe method

$$P_{N-1}(\lambda_j) = |\lambda_j|, \quad j = 1, \cdots, N.$$

$$Q_\Phi(w_L, w_R) = |A_\Phi(w_L, w_R)| = \sum_{j=0}^{N-1} \alpha_j A_\Phi^j(w_L, w_R),$$

where $\alpha_j, j = 0, \cdots, N-1$ are the solution of the following linear system:

$$\begin{pmatrix} 1 & \lambda_1 & \dots & \lambda_1^{N-1} \\ 1 & \lambda_2 & \dots & \lambda_2^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_N & \dots & \lambda_N^{N-1} \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{N-1} \end{pmatrix} = \begin{pmatrix} |\lambda_1| \\ |\lambda_2| \\ \vdots \\ |\lambda_N| \end{pmatrix},$$

$\lambda_1, \cdots, \lambda_N$ are the eigenvalues of the matrix $A_{i+1/2}$.

PVM-Chebyshev method

- Chebyshev polynomials provide optimal approximation to the absolute value function in $[-1, 1]$:

$$|x| = \frac{2}{\pi} + \sum_{k=1}^{\infty} \frac{4}{\pi} \frac{(-1)^{k+1}}{(2k-1)(2k+1)} T_{2k}(x), \quad x \in [-1, 1],$$

  where the Chebyshev polynomials of even degree $T_{2k}(x)$ are recursively defined as

$$T_0(x) = 1, \quad T_2(x) = 2x^2 - 1, \quad T_{2k}(x) = 2T_2(x)T_{2k-2}(x) - T_{2k-4}(x).$$

- The polynomial

$$\tau_{2p}(x) = \frac{2}{\pi} + \sum_{k=1}^{p} \frac{4}{\pi} \frac{(-1)^{k+1}}{(2k-1)(2k+1)} T_{2k}(x), \quad x \in [-1, 1],$$

  verifies the equality

$$\| |x| - \tau_{2p}(x) \|_{\infty} = \frac{2}{\pi} \frac{1}{2p+1},$$

  which is optimal in $L^{\infty}(-1, 1)$ (see Bernstein 1913).

-

$$Q_{i+1/2} = P_{2p}(A_{i+1/2}) = |\lambda_{\max}| \tau_{2p}\left(\frac{1}{|\lambda_{\max}|} A_{i+1/2} \right) \approx |A_{i+1/2}|,$$

  where $\lambda_{\max}$ is the eigenvalue of $A_{i+1/2}$ with maximum absolute value.

PVM-Chebyshev method

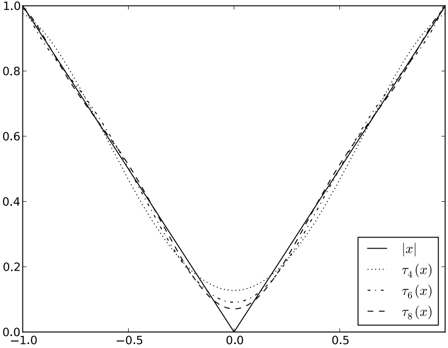


Figure : The Chebyshev approximations $\tau_{2p}(x)$ for $p = 2, 3, 4$.

- Notice that $\tau_{2p}(x)$ do not strictly satisfy the stability condition $\tau_{2p}(x) \geq |x|$. This drawback can be avoide by substituting $\tau_{2p}(x)$ by $\tau^{\epsilon}_{2p}(x)$,such that $\tau^{\epsilon}_{2p}(x) \geq |x|$.

References

References

I. Toumi.

A weak formulation for Roe approximate Riemann solver. J. Comput. Phys. 102:360-373, 1992.

C. Parés.

Numerical methods for nonconservative hyperbolic systems: a theoretical framework. SIAM J. Numer. Anal., 44(1):300-321, 2006.

M. Castro, J.M. Gallardo and C. Parés.

High order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products. Applications to shallow water systems. Math. Comp. 75: 1103-1134, 2006.

M. Castro, J.M. Gallardo, J.A. López and C. Parés.

Well-balanced high order extensions of Godunov's method for semilinear balance laws. SIAM J. Num. Anal., 46(2): 1012-1039, 2008.

M. Castro, E.D. Fernández, A. Ferreiro, J.A. García and C. Parés.

High order extensions of Roe schemes for two dimensional nonconservative hyperbolic systems. J. Sci. Comput., 39: 67-114, 2009.

References

References

P. Degond, P.F. Peyrard, G. Russo, Ph. Villedieu.

Polynomial upwind schemes for hyperbolic systems. C. R. Acad. Sci. Paris 1 328, 479-483, 1999.

M.J. Castro, A. Pardo, C. Parés, E.F. Toro.

On some fast well-balanced first order solvers for nonconservative systems. Mathematics of Computation 79: 1427-1472, 2010.

E. D. Fernández-Nieto, M.J. Castro, C. Parés.

On an Intermediate Field Capturing Riemann Solver Based on a Parabolic Viscosity Matrix for the Two-Layer Shallow Water System Journal of Scientific Computing 48 (1): 117-140, 2011.

M. Castro, E.D. Fernández.

A class of computationally fast first order finite volume solvers: PVM methods. SIAM Journal on Scientific Computing 34 (4): 2173–2196, 2012.

M.J. Castro, J.M. Gallardo, A.Marquina.

A new class of incomplete Riemann solvers based on uniform rational approximations to the absolute value function. Submitted to JCP, 2012. Submitted to JSC, 2013.

What is a GPU?

GPU (Graphics Processing Unit)

GPUs are specialized circuits designed to rapidly manipulate the building of images in a frame buffer intended for output to a display. Driven by the insatiable market demand for realtime, high-definition 3D graphics, the programmable Graphic Processor Unit or GPU has evolved into a highly parallel, multithreaded, many-core processor with tremendous computational power and very high memory bandwidth

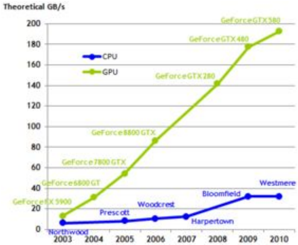
GPGPU (General-purpose computation on GPU)

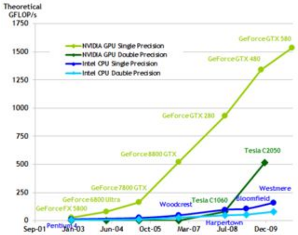
It is the technique of using a GPU to perform computations in applications traditionally handled by the CPU. It is made possible by the addition of programmable stages and higher precision arithmetic to the rendering pipelines, which allows software developers to use stream processing on non-graphics data.
There are some parallel computing architectures that allows this: Brook++, OPENCL, Cg, **CUDA**.

CPU vs GPU architectures

- GPUs are specialized for compute-intensive highly parallel computations.
- GPU are especially well-suited to address problems that present a high data-parallel paradigm.
- The same program is executed for each data element in GPUs, so the flow control of the code is simple.
- The code it is executed on many data elements in GPUS, so the memory access latency can be hidden with calculations instead of big data caches.

Why GPUs?





- GPUs
  - GeForce FX 5900: 450 MHz, 256 MB
  - GForce 8800 GTX: 128 Cores, 575 MHz, 728 MB
  - GeForce GTX 580: 512 Cores, 1544 MHz, 1546 MB

Code Example

CPU Code

```
int N = ... //global variable

void MatAdd(float A[N][N], float B[N][N],
float C[N][N])
{
for (int i = 0; i<N, i++)
{
    for (int j = 0; j<N; j++)
    {
    C[i][j] = A[i][j] + B[i][j];
    }
}
}

int main()
{
...
float A[N][N], B[N][N], C[N][N];
...
MatAdd(A, B, C); // C = A + B
}
```

GPU Code

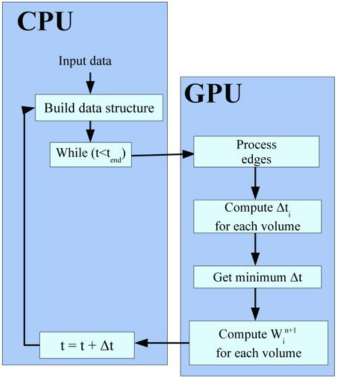
```
int N = ...

// Kernel definition
__global__ void MatAdd(float A[N][N], float B[N][N],
float C[N][N])
{
int i = threadIdx.x;
int j = threadIdx.y;
if(i<N && j<N)
    C[i][j] = A[i][j] + B[i][j];
}

int main()
{
...
// Kernel invocation with one block of N * N threads
int numBlocks = 1;
dim3 threadsPerBlock(N, N);
MatAdd<<<numBlocks, threadsPerBlock>>>(A, B, C);
}
```

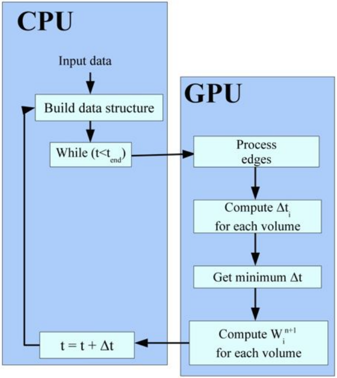
GPU Code Example

1. Input data in CPU.
2. Build data structure as arrays in CPU.
3. Loop in time.



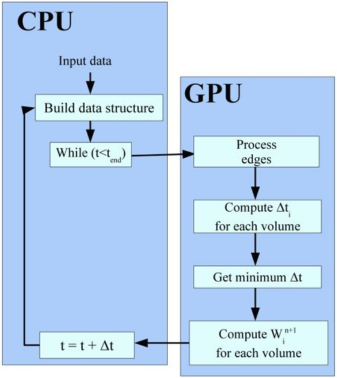
GPU Code Example

1. Input data in CPU.
2. Build data structure as arrays in CPU.
3. Loop in time.
    1. Copy CPU arrays in GPU arrays. Data is stored in GPU in *float4* textures.



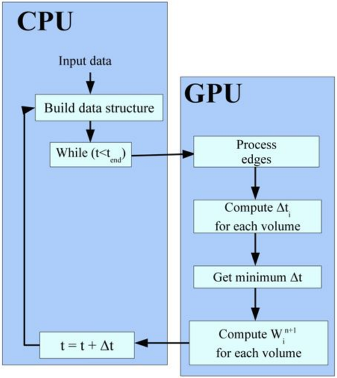
GPU Code Example

1. Input data in CPU.
2. Build data structure as arrays in CPU.
3. Loop in time.
    1. Copy CPU arrays in GPU arrays.
       Data is stored in GPU in *float4* **textures**.
    2. Calculate and copy the contributions $D^{+}_{i-\frac{1}{2}}$ and $D^{-}_{i+\frac{1}{2}}$ to *float4* arrays.



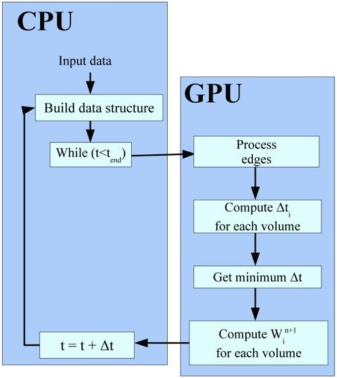
GPU Code Example

1. Input data in CPU.
2. Build data structure as arrays in CPU.
3. Loop in time.
   1. Copy CPU arrays in GPU arrays.
      Data is stored in GPU in *float4* **textures**.
   2. Calculate and copy the contributions $D^+_{i-\frac{1}{2}}$ and $D^-_{i+\frac{1}{2}}$ to *float*4 arrays.
   3. Compute Δt_i for each volume.



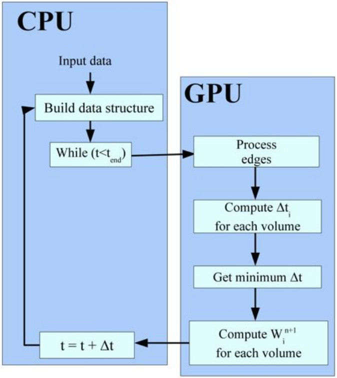
GPU Code Example

1. Input data in CPU.
2. Build data structure as arrays in CPU.
3. Loop in time.
   1. Copy CPU arrays in GPU arrays.
      Data is stored in GPU in *float4* **textures**.
   2. Calculate and copy the contributions $D^{+}_{i-\frac{1}{2}}$ and $D^{-}_{i+\frac{1}{2}}$ to *float*4 arrays.
   3. Compute Δt_i for each volume.
   4. Get the minimum of Δt_i.



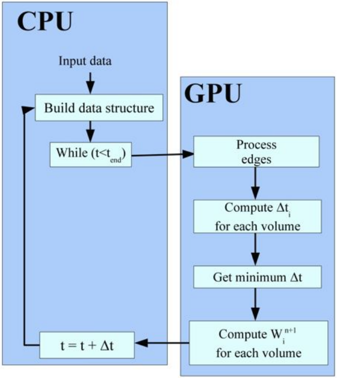
GPU Code Example

1. Input data in CPU.
2. Build data structure as arrays in CPU.
3. Loop in time.
    1. Copy CPU arrays in GPU arrays.
       Data is stored in GPU in *float4* **textures**.
    2. Calculate and copy the contributions $D^{+}_{i-\frac{1}{2}}$ and $D^{-}_{i+\frac{1}{2}}$ to $float4$ arrays.
    3. Compute Δt_i for each volume.
    4. Get the minimum of Δt_i.
    5. Calculate $W_i^{n+1} = W_i^n - \frac{\Delta t}{\Delta x}(D^{+}_{i-\frac{1}{2}} + D^{-}_{i+\frac{1}{2}})$.



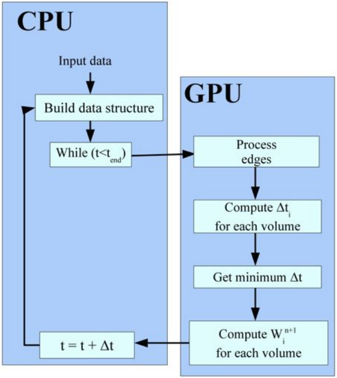
GPU Code Example

1. Input data in CPU.
2. Build data structure as arrays in CPU.
3. Loop in time.
    1. Copy CPU arrays in GPU arrays.
       Data is stored in GPU in *float4* **textures**.
    2. Calculate and copy the contributions $D^+_{i-\frac{1}{2}}$ and $D^-_{i+\frac{1}{2}}$ to $\mathit{float}4$ arrays.
    3. Compute Δt_i for each volume.
    4. Get the minimum of Δt_i.
    5. Calculate $W_i^{n+1} = W_i^n - \frac{\Delta t}{\Delta x}(D^+_{i-\frac{1}{2}} + D^-_{i+\frac{1}{2}})$.
    6. Next step in time.



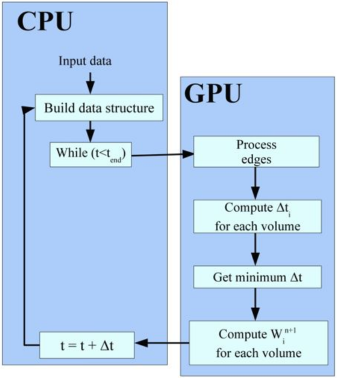
GPU Code Example

1. Input data in CPU.
2. Build data structure as arrays in CPU.
3. Loop in time.
   1. Copy CPU arrays in GPU arrays.
      Data is stored in GPU in *float4* **textures**.
   2. Calculate and copy the contributions $D^{+}_{i-\frac{1}{2}}$ and $D^{-}_{i+\frac{1}{2}}$ to $float4$ arrays.
   3. Compute Δt_i for each volume.
   4. Get the minimum of Δt_i.
   5. Calculate $W_i^{n+1} = W_i^n - \frac{\Delta t}{\Delta x}(D^{+}_{i-\frac{1}{2}} + D^{-}_{i+\frac{1}{2}})$.
   6. Next step in time.



GPU Code Example

1. Input data in CPU.
2. Build data structure as arrays in CPU.
3. Loop in time.
   1. Copy CPU arrays in GPU arrays.
      Data is stored in GPU in *float4* **textures**.
   2. Calculate and copy the contributions $D^+_{i-\frac{1}{2}}$ and $D^-_{i+\frac{1}{2}}$ to *float*4 arrays.
   3. Compute Δt_i for each volume.
   4. Get the minimum of Δt_i.
   5. Calculate $W_i^{n+1} = W_i^n - \frac{\Delta t}{\Delta x}(D^+_{i-\frac{1}{2}} + D^-_{i+\frac{1}{2}})$.
   6. Next step in time.



References

References

NVIDIA.
NVIDIA CUDA C Programming Guide. Version 4.0. 2011

NVIDIA. CUDA Zone.
http://www.nvidia.com/object/cuda_home.html. Accessed June 2009.

An example: First order PVM scheme for the two-layer shallow-water system

For each PVM method:

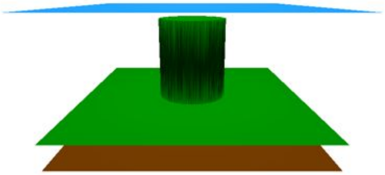
- **Two implementations in CPU** using C++ and the Eigen library: one for one thread and another for 4 threads using OpenMP. Both programs use double precision.
- **One implementation in GPU** using CUDA and single precision.
- Computer: Intel Core i7 920, 4 GB RAM.
- Graphics card: GeForce GTX 570.

An example: First order PVM scheme for the two-layer shallow-water system

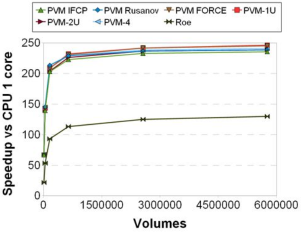
- Initial state:

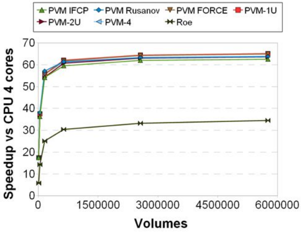
$$H(x,y) = 5$$
$$W_i^0(x,y) = \left(h_1(x,y),\ 0,\ 0,\ h_2(x,y),\ 0,\ 0\right)^T$$
$$h_1(x,y) = \begin{cases} 4 & \text{if } \sqrt{x^2+y^2} > 1.5 \\ 0.5 & \text{otherwise} \end{cases}$$
$$h_2(x,y) = 5 - h_1(x,y)$$

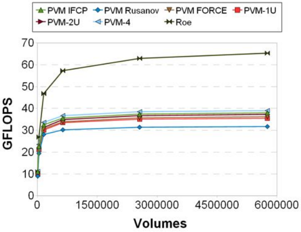
- Wall boundary conditions
- Time interval [0, 0.1]
- $\gamma = 0.9$, $r = 0.998$
- Different mesh sizes, up to 5760000 volumes



An example (Two-layer shallow-water system)

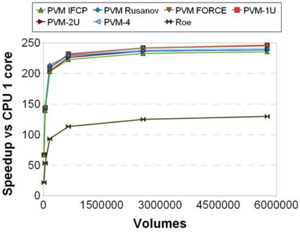


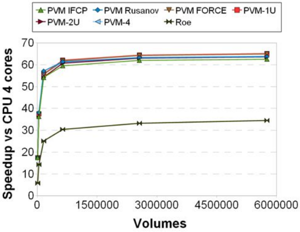


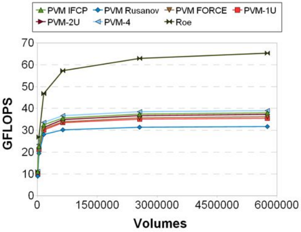


- PVM methods have been 1.9 times faster than Roe method on a GTX 570.
- Roe method has reached more GFLOPS than PVM methods.

An example (Two-layer shallow-water system)



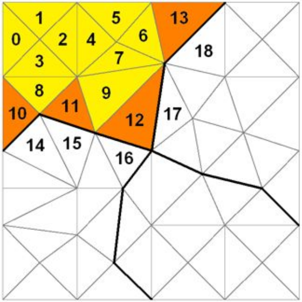




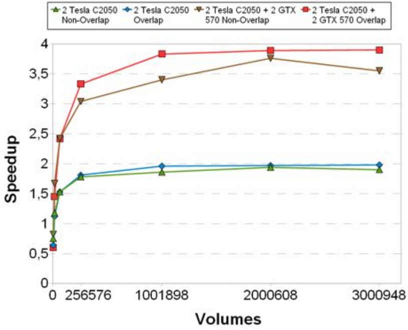
- PVM methods have been 1.9 times faster than Roe method on a GTX 570.
- Roe method has reached more GFLOPS than PVM methods.

Multi-GPU Implementation

- The triangular mesh is divided into submeshes. Each submesh is assigned to a CPU process, which uses a GPU to perform its computations.
- Volume indexation:
  1. Non-communication volumes.
  2. Comm. volumes of the submesh.
  3. Comm. volumes of adjacent submeshes.
- The communication volumes of the submesh must be reordered, overlapping the sendings.



Multi-GPU Implementation



- Two implementations:
  - One with blocking MPI sends and receives.
  - One that overlaps MPI communication with CPU-GPU memory transfers and kernel computation.

References

References

M. Lastra, J. M. Mantas, C. Ureña, M. J. Castro, J. A. García- Rodríguez,

Simulation of shallow-water systems using graphics processing units, Mathematics and Computers in Simulation 80 (3),598-618, 2009.

M. de la Asunción, J. M. Mantas, M. J. Castro,

Simulation of one-layer shallow water systems on multicore and CUDA architectures, The Journal of Supercomputing, 2010. Online.

M. J. Castro, S. Ortega, M. de la Asunción, J. M. Mantas, J. M. Gallardo,

GPU computing for shallow water flow simulation based on finite volume schemes, Comptes Rendus Mécanique 339 (2-3),165-184, 2011. Special issue on High Performance Computing.

M. de la Asunción, J.M. Mantas, M.J. Castro, E.D. Fernández-Nieto.

An MPI-CUDA implementation of an improved Roe method for two-layer shallow water systems on non-structured meshes. submitted to Journal of Parallel and Distributed computing, 72 (9): 1065-1072, 2012.

M de la Asunción, MJ Castro, ED Fernández-Nieto, JM Mantas, S. Ortega, J. Manuel González-Vida.

Efficient GPU implementation of a two waves TVD-WAF method for the two-dimensional one layer shallow water system on structured meshes. Accepted on Computers & Fluids, 2012.

HySEA: interdisciplinary platform

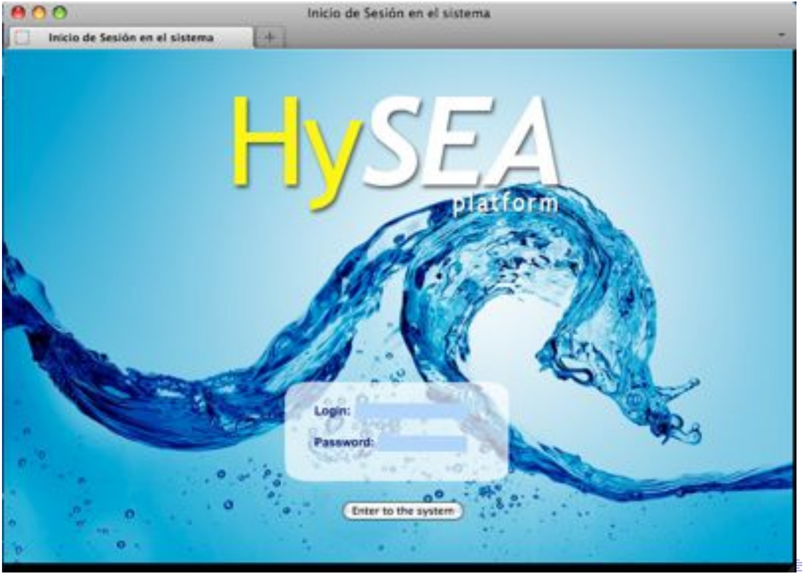
- Models based on geophysical flows with applications in:
  - Physical Oceanography
  - Marine Geology
  - Tsunami Research
  - Civil Engineering

HySEA: how to make more accessible our codes?

- Hardware:
  - No need to have a large computational infrastructure.
  - Codes have to be accessible from any architecture.
- Software:
  - No need to install specific libraries, compilers, etc.
  - Models have to be independent from the operative system.
  - Easy update process for code changes.
  - Updated documentation.

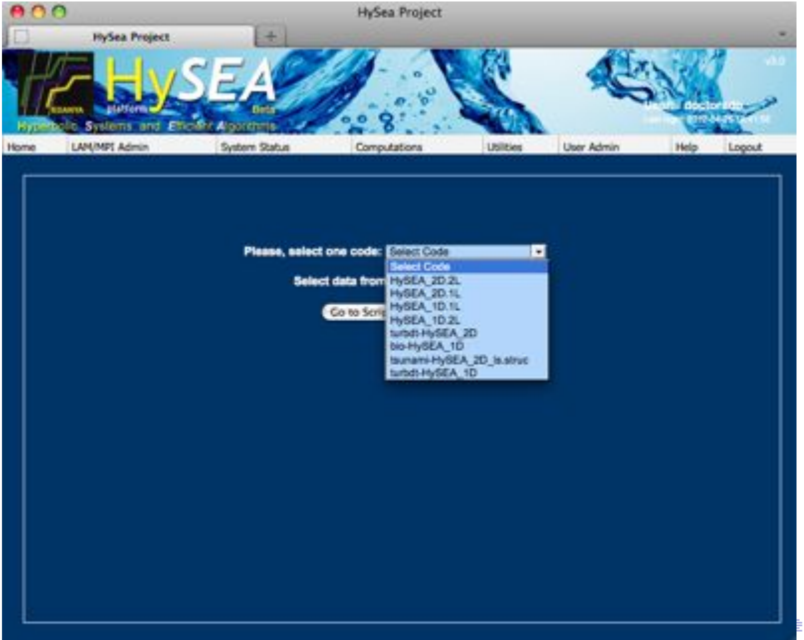
HySEA: making accessible our codes.

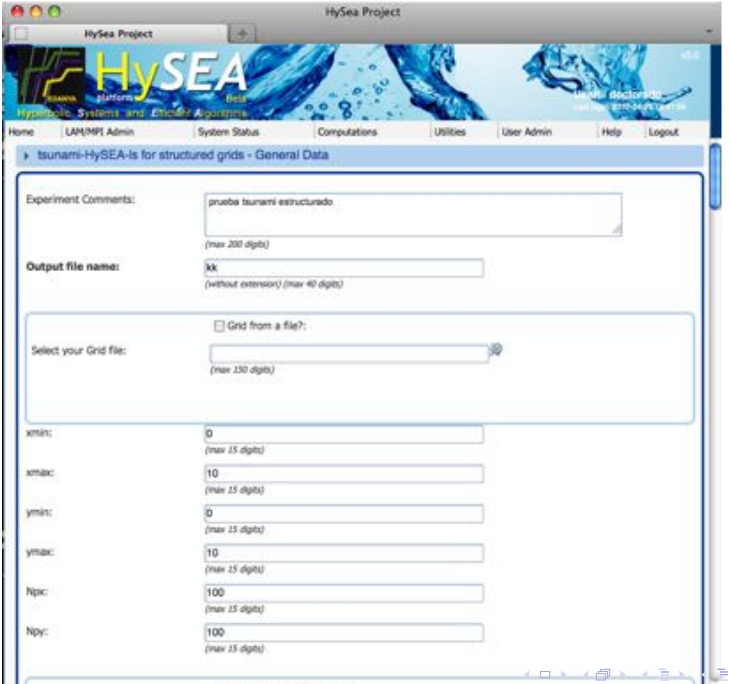
- Hardware:
  - We have a supercomputer (CPU's and GPU's): cires.uma.es
  - Numerical Methods Laboratory of the University of Málaga.
- Software:
  - The interface between the researcher and Cires is a web browser.

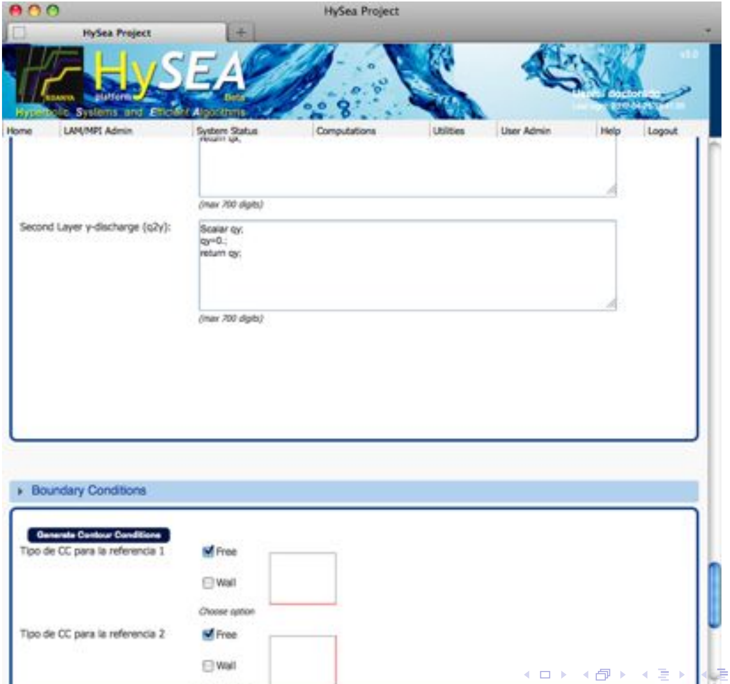

Inicio de Sesión en el sistema
Inicio de Sesión en el sistema
HySEA
platform
Login:
Password:
Enter to the system


HySea Project
HySea Project
HySEA
platform
Beta
Hyperbolic Systems and Efficient Algorithms
Home
LAM/MPI Admin
System Status
Computations
Utilities
User Admin
Help
Logout
Hi, mosaico!.
Please, select one option from the above menu.
"Hay algo que Dios ha hecho mal. A todo le puso limites menos a la tontería."
(Konrad Adenauer)


HySea Project
HySea Project
HySEA
platform
Beta
Hyperbolic Systems and Efficient Algorithms
Home
LAM/MPI Admin
System Status
Computations
Utilities
User Admin
Help
Logout
Hi, mosaicol.
Please, select one option from the above menu.
"Hay algo que Dios ha hecho mal. A todo le puso límites menos a la tontería."
(Konrad Adenauer)


HySea Project
HySea Project
HySEA
Hyperbolic Systems and Efficient Algorithms
Home
LAM/MPI Admin
System Status
Computations
Utilities
User Admin
Help
Logout
Please, select one code:
Select Code
Select Code
HySEA_2D.2L
HySEA_2D.1L
HySEA_1D.1L
HySEA_1D.2L
turbdt-HySEA_2D
bio-HySEA_1D
tsunami-HySEA_2D_ls.struc
turbdt-HySEA_1D
Select data from
Go to Scri


HySea Project
HySea Project
HySEA
platform
Beta
Hyperbolic Systems and Efficient Algorithms
Home
LAM/MPI Admin
System Status
Computations
Utilities
User Admin
Help
Logout
tsunami-HySEA-ls for structured grids - General Data
Experiment Comments:
prueba tsunami estructurado
(max 200 digits)
Output file name:
kk
(without extension) (max 40 digits)
Grid from a file?:
Select your Grid file:
(max 150 digits)
xmin:
0
(max 15 digits)
xmax:
10
(max 15 digits)
ymin:
0
(max 15 digits)
ymax:
10
(max 15 digits)
Npx:
100
(max 15 digits)
Npy:
100
(max 15 digits)


HySea Project
HySea Project
HySEA
Home
LAM/MPI Admin
System Status
Computations
Utilities
User Admin
Help
Logout
(max 700 digits)
Second Layer y-discharge (q2y):
Scalar qy;
qy=0.;
return qy;
(max 700 digits)
Boundary Conditions
Generate Contour Conditions
Tipo de CC para la referencia 1
Free
Wall
Choose option
Tipo de CC para la referencia 2
Free
Wall

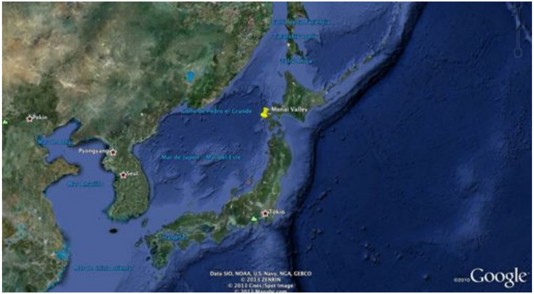

HySea Project
HySea Project
HySEA
platform
Beta
Hyperbolic Systems and Efficient Algorithms
Home
LAM/MPI Admin
System Status
Computations
Utilities
User Admin
Help
Logout
Please, select your execution scripts:
Don't forget to run the parallel machine before start your execution.
Graphical view of your nodes status
2012-04-25 18:47
Load/min
N. CPUs
c0-15
c0-2
c0-6
c0-13
Select the startup node for this job:
Run code

Dambreak problem: Limonero Dam (close to Málaga (Spain))

- Resolution 5 m $\times$ 5 m.
- Number of cells: 1052224.
- Real simulated time: 20 min.
- Used scheme: Second order HLL or PVM-1U(S_L,S_R).
- Positivity of the water height is ensured.
- Graphics card: GeForce GTX 570. Speedup: 230 (1 Intel Core i7 920).

Dambreak problem: Limonero Dam (close to Málaga (Spain))

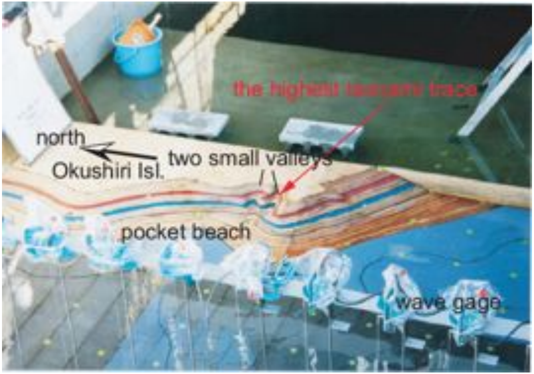
Monai valley: Location



Monai valley: Laboratory Experiment

Matsuyama and Tanaka (2001)

- Maximum run-up height at Monai zone in Okushiri Island (1993)
- Scale1:400
- 3.4 m wide, 205 m long flume
- Maximum depth 6 m.

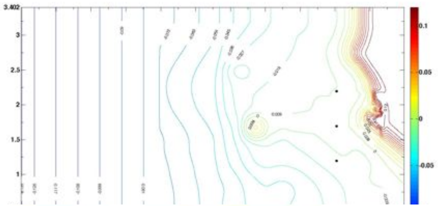


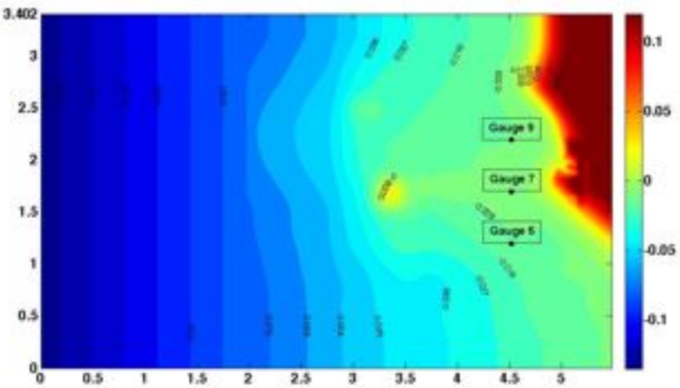
Monai valley: Benchmark problem

- Synolakis et al. (2007). NOAA Technical Memorandum. Standards, criteria, and procedures for NOAA evaluation of tsunami numerical models
- Liu et al. (2008). Third Int. Workshop on Long-Waves Runup models.
- http://isec.nacse.org/workshop/2004 cornell/bmark2.html (Inundation Science & Engineering Cooperative)
- http://nctr.pmel.noaa.gov/benchmark/Laboratory/Laboratory MonaiValley/ (NOAA Center for Tsunami Research)

Monai valley: Computational domain

- Rectangle 5.488 m $\times$ 3.402 m
- Minimum depth: -0.125 m (land)
- Maximum depth: 0.13535 m
- Grid size: 0.014 m

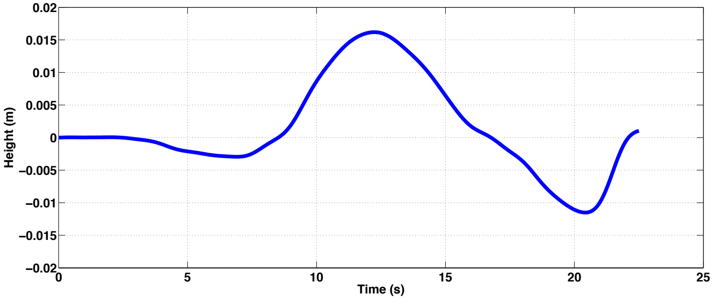




Monai valley: Input wave

Boundary conditions

- West: input wave
  - Time domain: 0-22.5 sec
  - Time step: 0.05 sec
- North, East and South: solid wall

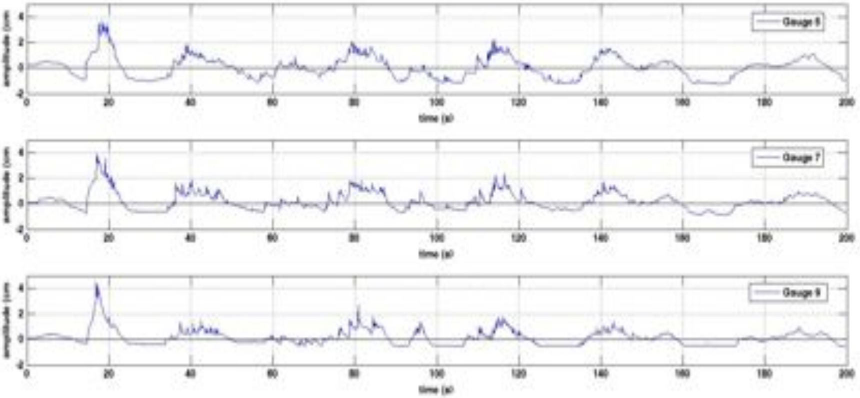


Monai valley: Numerical Experiments

Spatial resolution

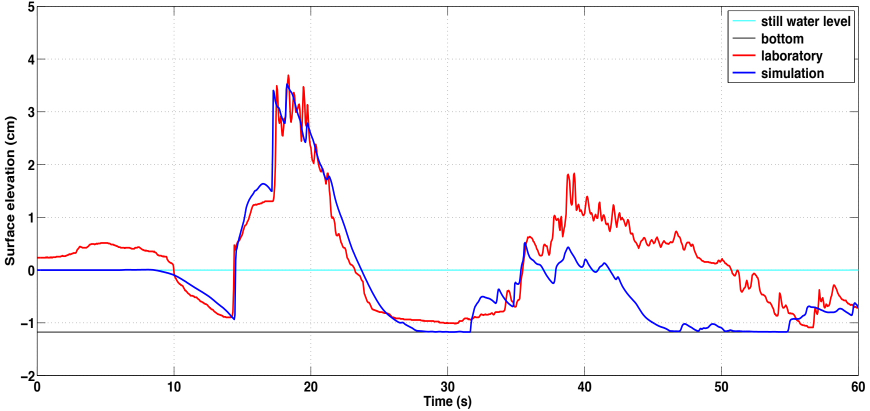
Grid	$\Delta x = \Delta y$ (cm)	# volumes	Comput. Time (s-(min))
393×244	0.014	95,892	91.54618 (1.52)
785×487	0.007	382,295	538.0129 (8.96)
1569×973	0.0035	1,526,637	3,693.026 (61.55)
3137×1945	0.00175	6,101,465	28,255.71 (470.92)

Monai valley: Time series at laboratory gauges



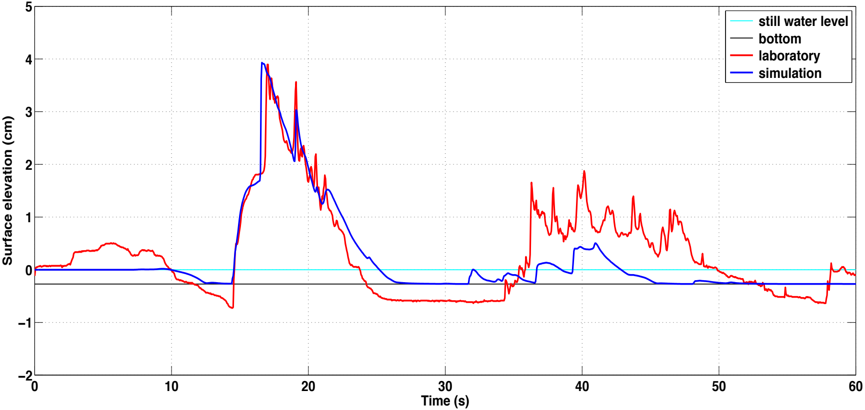
Monai valley: Physical model - Numerical simulation comparison

Gauge 5



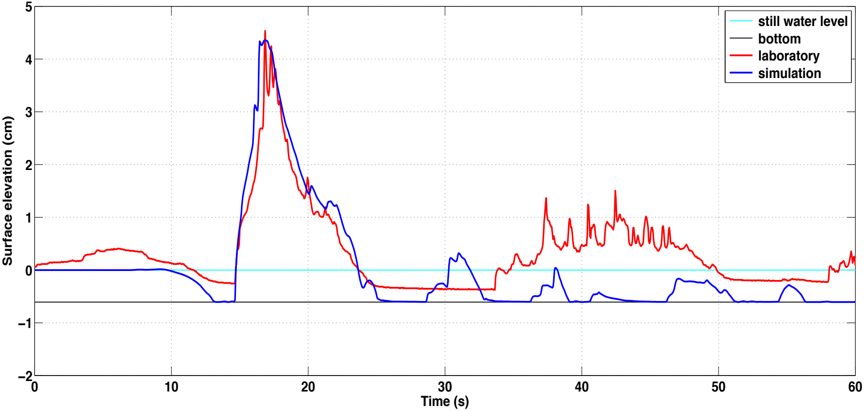
Monai valley: Physical model - Numerical simulation comparison

Gauge 7



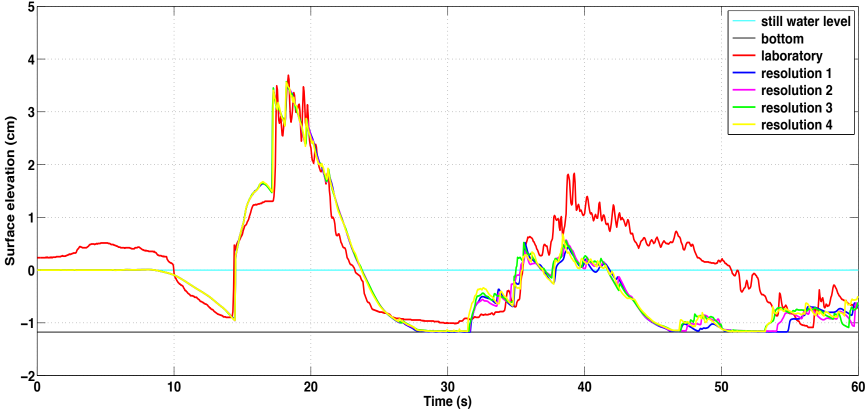
Monai valley: Physical model - Numerical simulation comparison

Gauge 9

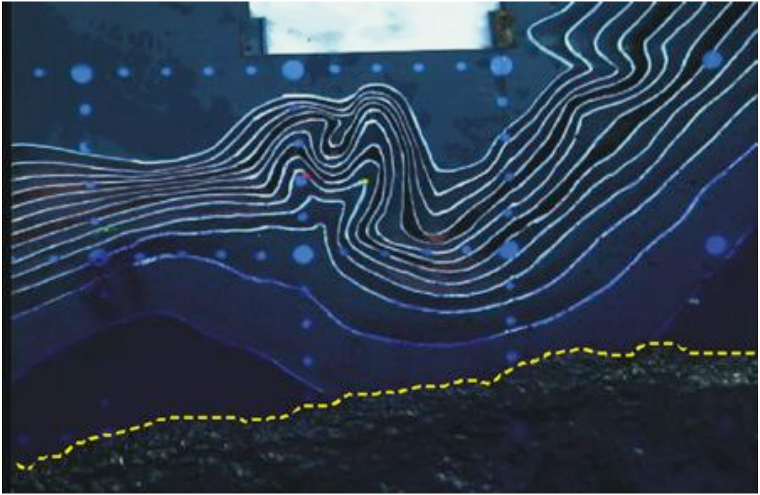


Monai valley: checking convergence

Gauge 5

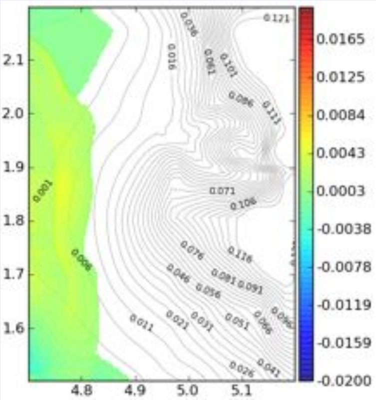


Monai valley: Video



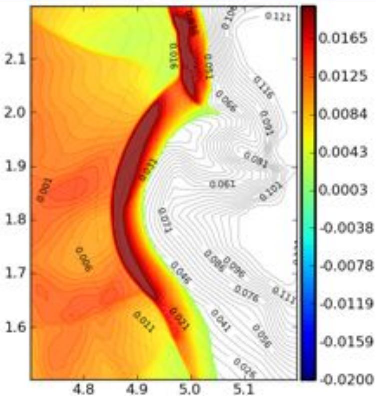
Monai valley: Frontal wave location

Physical model (Frame 10) - Numerical simulation (Time 15 s)



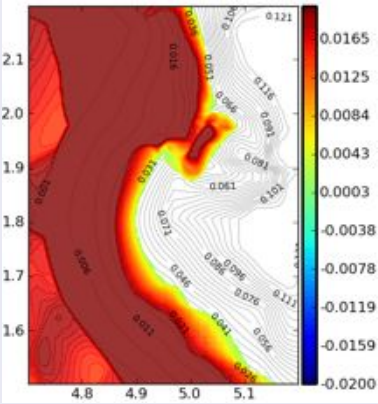
Monai valley: Frontal wave location

Physical model (Frame 25) - Numerical simulation (Time 15.5 s)



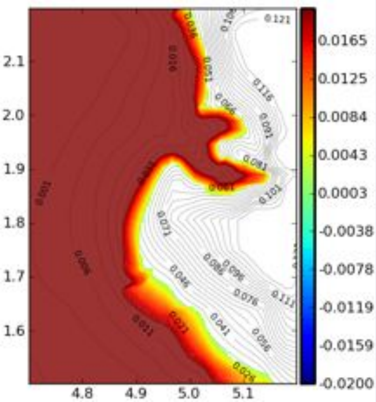
Monai valley: Frontal wave location

Physical model (Frame 40) - Numerical simulation (Time 16 s)



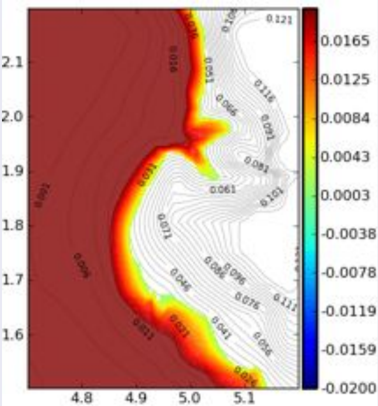
Monai valley: Frontal wave location

Physical model (Frame 55) - Numerical simulation (Time 16.5 s)



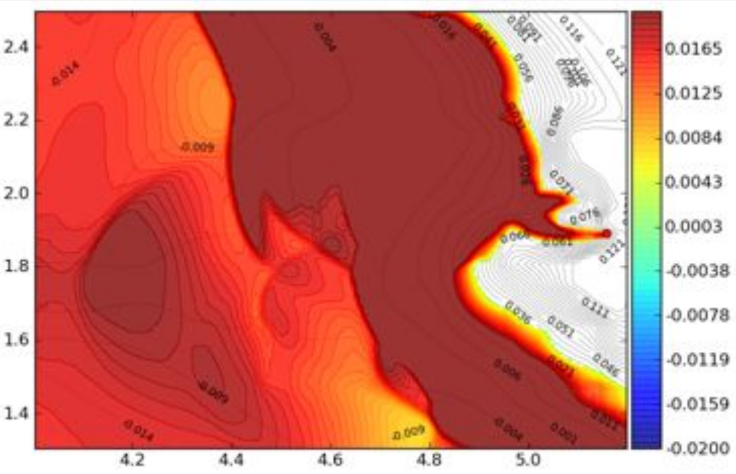
Monai valley: Frontal wave location

Physical model (Frame 70) - Numerical simulation (Time 17 s)

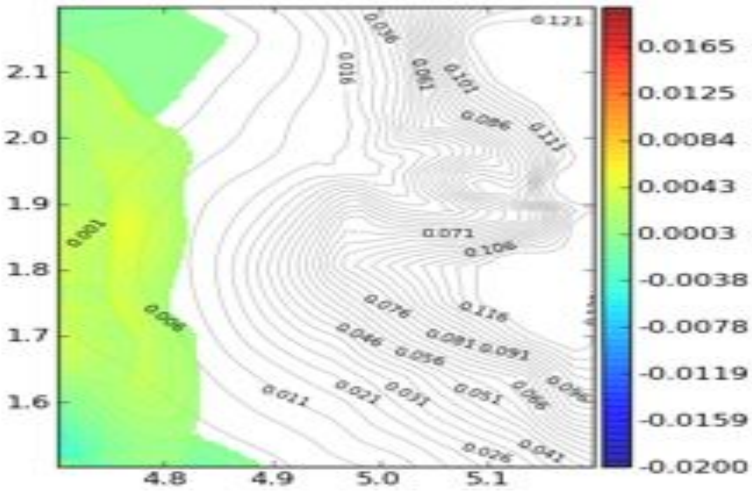


Monai valley: Maximal run-up height

Time 16.3 s - At point (5.15592356688,1.88961190965) - Max heigth: 0.0891 (0.08958)



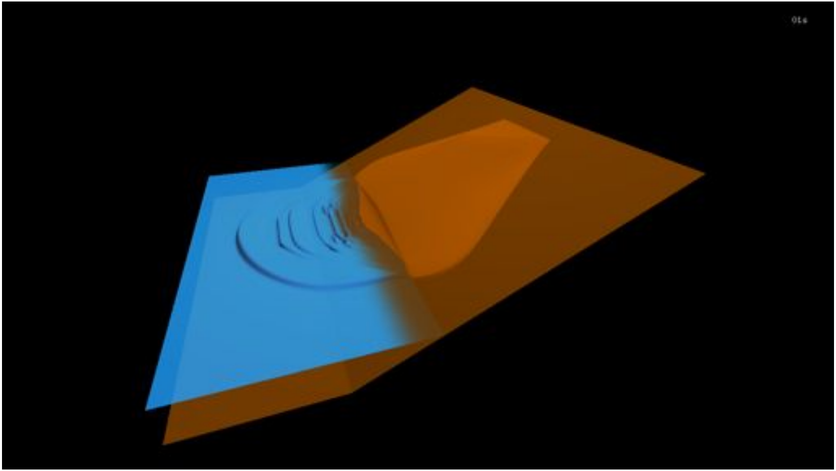
Monai valley: Video



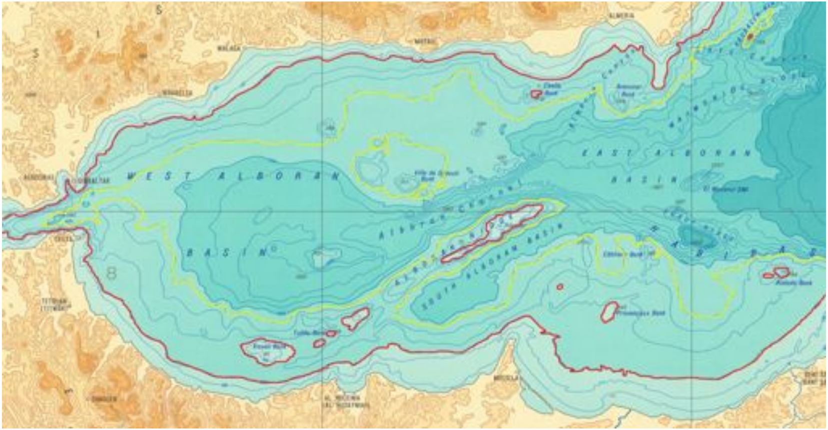
Tsunami generated by a landslide

- Number of cells: 640000.
- $r = 0.4$, Coulomb friction angle 25^o
- Real simulated time: 8 sec.
- Used scheme: PVM-2U(S_L,S_R,S_{int}) or IFCP
- Positivity of the water height is ensured.
- Graphics card: GeForce GTX 570. Speedup: 190 (1 Intel Core i7 920).

Tsunami generated by a landslide

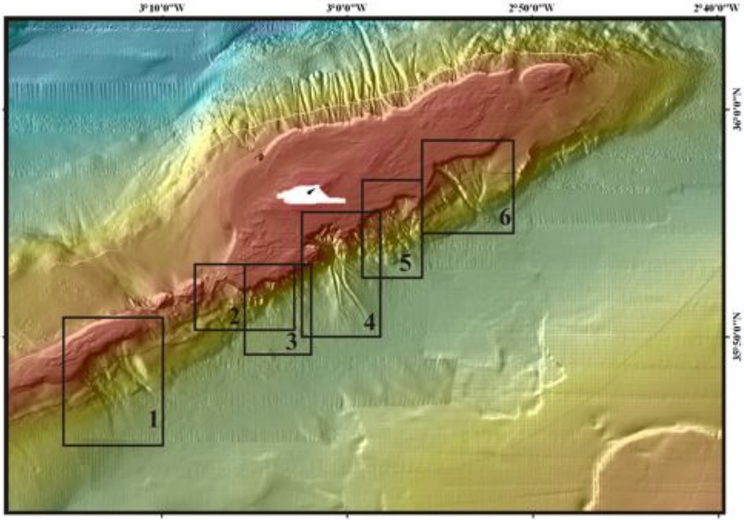


Tsunami at the Alboran Sea I

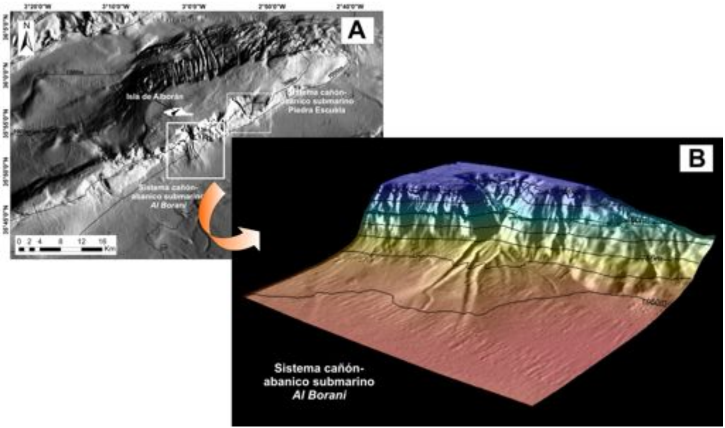


Tsunami at the Alboran Sea I

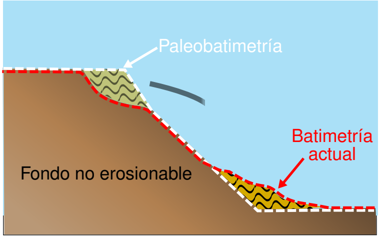
Tsunami at the Alboran Sea I

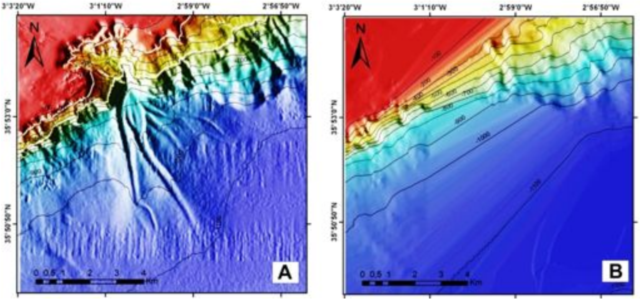


Tsunami at the Alboran Sea I

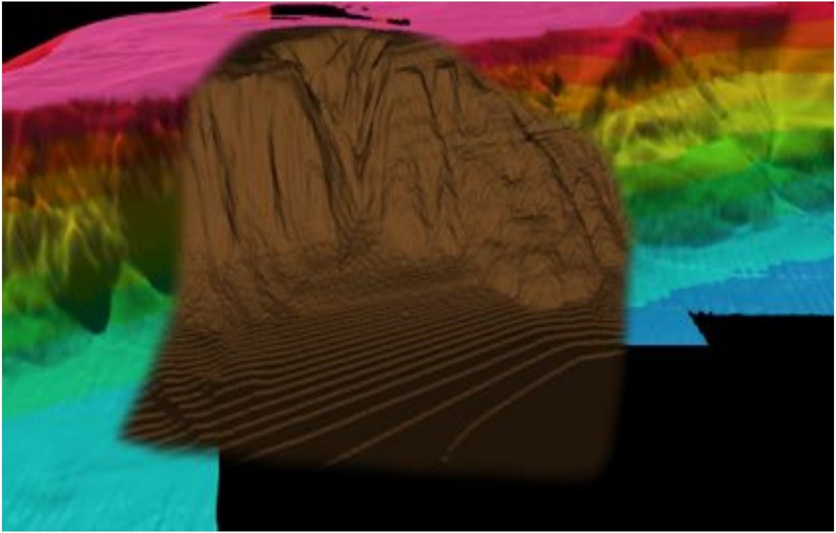


Tsunami at the Alboran Sea II: Bathymetry reconstruction

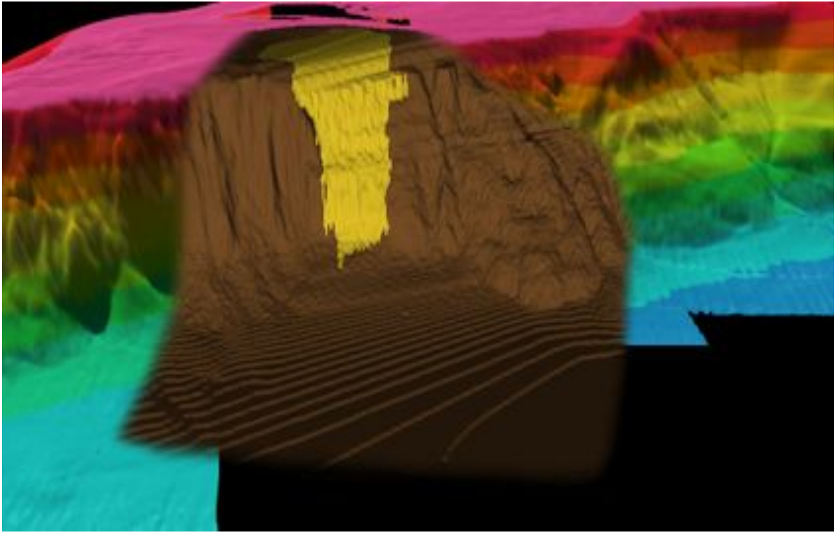




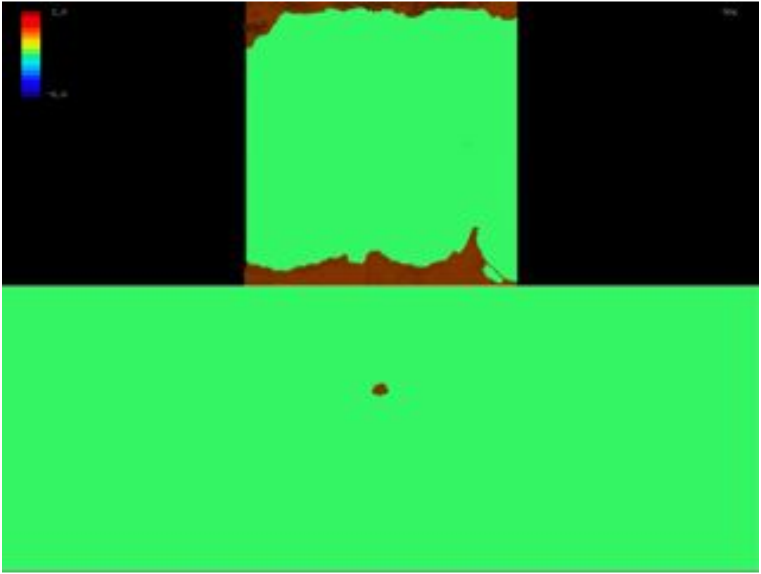
Tsunami at the Alboran Sea III: initial condition



Tsunami at the Alboran Sea III: initial condition



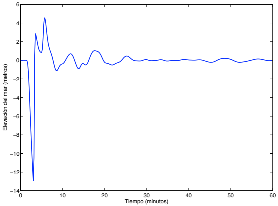
Tsunami at the Alboran Sea IV: simulation

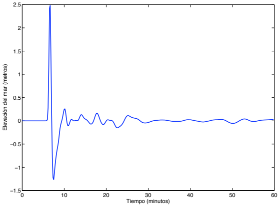


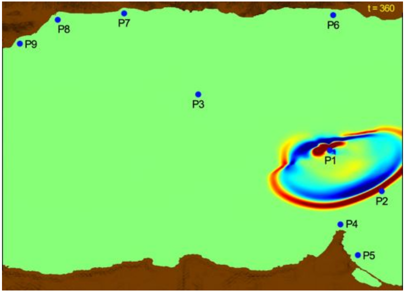
Tsunami at the Alboran Sea IV:simulation

Tsunami at the Alboran Sea V: simulation

P1 (generation zone) - P2 (SE - open sea)

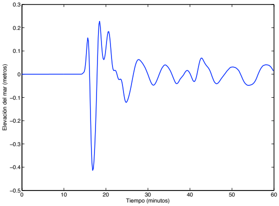


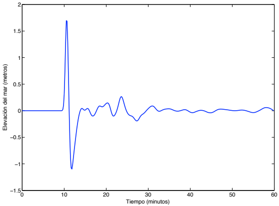


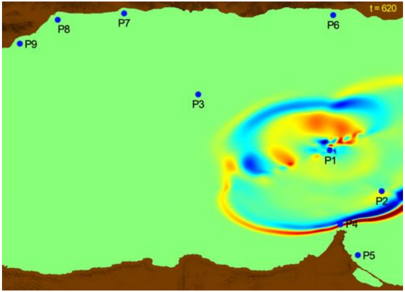


Tsunami at the Alboran Sea V: simulation

Puntos P3 (NO - open sea) - P4 (Tres Forcas)

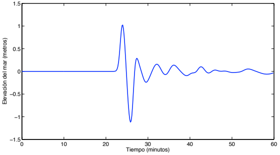


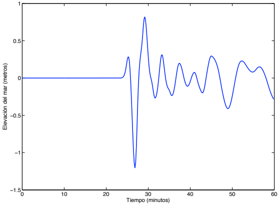


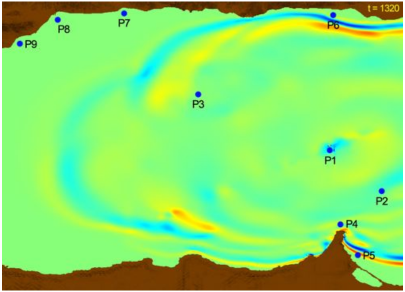


Tsunami at the Alboran Sea V: simulation

P5 (Melilla) - P2 (Adra)

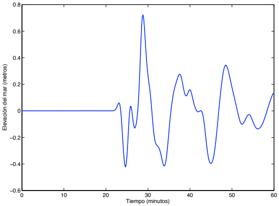


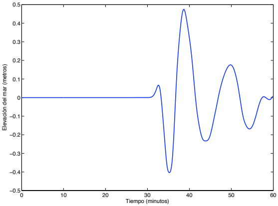


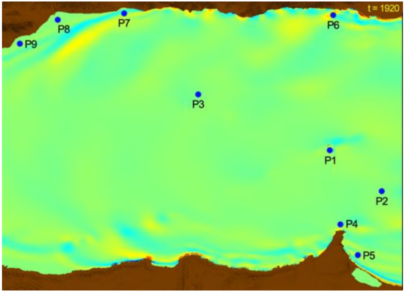


Tsunami at the Alboran Sea V: simulation

P7 (Torre del Mar) - P8 (Málaga)

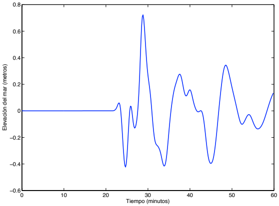


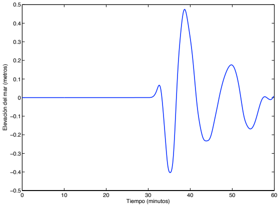


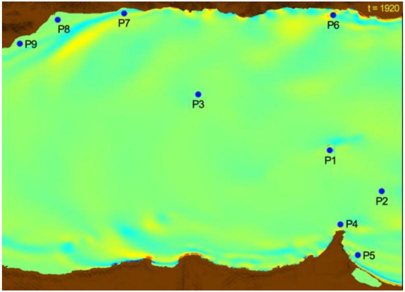


Tsunami at the Alboran Sea V: simulation

P7 (Torre del Mar) - P8 (Málaga)







Lituya Bay description

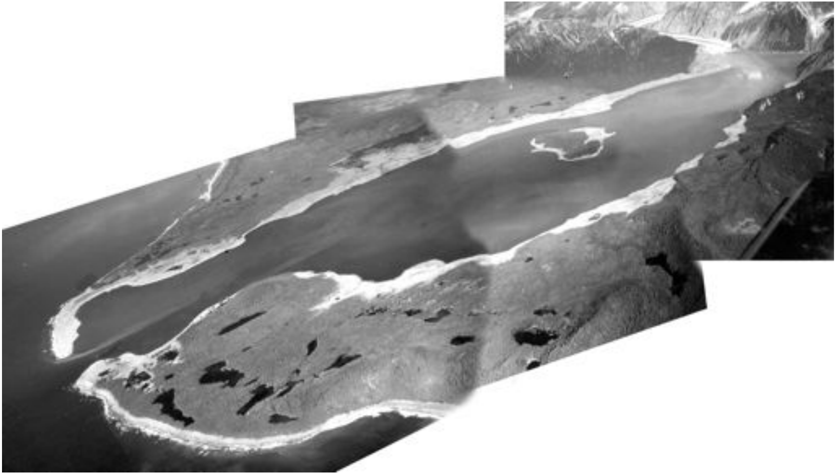
- Lituya bay is T-shaped, about 12 kilometres long and from 1.2 to 3 kilometres wide (except at the entrance, which is only 300m wide).
- Two glaciated inlets at the head, Gilbert Inlet and Crillon Inlet.
- Narrow and shallow entrance.
- High mountains with large slopes near the head of the bay.
- High tides in this area (about 3 m height).

Lituya Bay: Earthquake & Landslide

- At 10:16pm (local time) on July 10, 1958. M_w 8.3 earthquake.
- Southwest sides and bottoms of Gilbert and Crillon inlets moved northwestward and relative to the northeast shore at he head of the bay, on the opposite side of the Fairweather fault.
- Shaking lasted about 4 minutes. Estimated total movements of 6.4m horizontally and 1 m vertically.
- About 2 minutes after the beginning of the earthquake the landslide was triggered.
- Slide volume estimated by Miller, 1960 in $30.6 \times 10^6\ m^3$.

Picture from Weiss et. al., 2009

Lituya Bay: Effects of the 1958 tsunami



Lituya Bay: Tsunami source: reconstruction of the slide

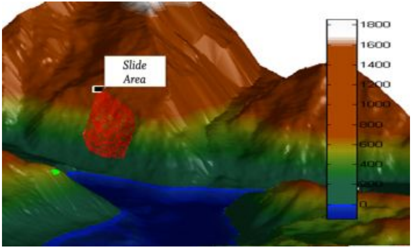
Landslide setup

- Determine the slide perimeter.
- Surface centroid located at 610 m elevation.
- Volume of reconstructed slide: 30.625×10^6 m^3.

Lituya Bay: Tsunami source: reconstruction of the slide

Landslide setup

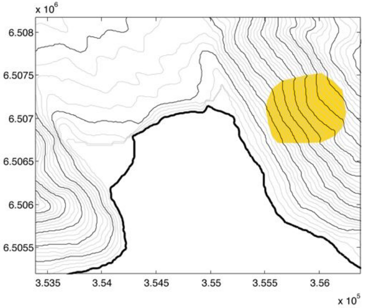
- Determine the slide perimeter.
- Surface centroid located at 610 m elevation.
- Volume of reconstructed slide: 30.625×10^6 m^3.

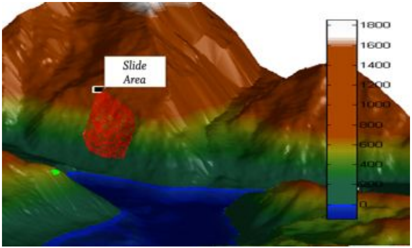


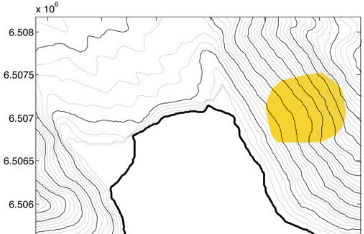
Lituya Bay: Tsunami source: reconstruction of the slide

Landslide setup

- Determine the slide perimeter.
- Surface centroid located at 610 m elevation.
- Volume of reconstructed slide: 30.625×10^6 m^3.



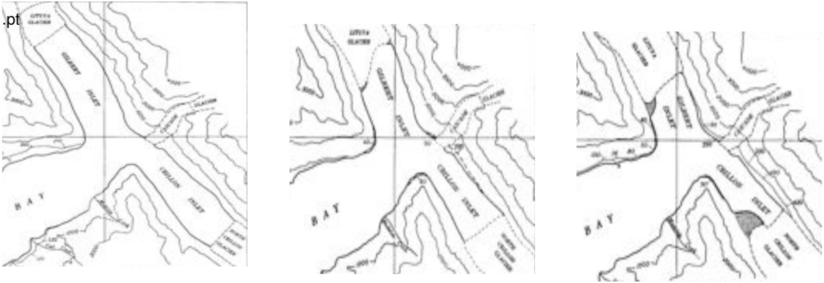




Lituya Bay: Topo-bathymetry reconstruction

Gilbert Inlet Glacier front reconstruction

- Prior to the 1958 low deltas of gravel had built out into Gilbert Inlet at the southwest and northeast margins of the Lituya Glacier front.
- After the 1958 tsunami the delta on the northeast side of Gilbert Inlet completely disappeared, and the delta on the southwest side was much smaller.
- According to [Miller, 1960], the glacier surface for several hundred feet from the front was severely crevassed.



Head of Lituya Bay, showing changes in shoreline and glacier fronts over time from [Miller, 1960].

Lituya Bay: Topo-bathymetry reconstruction

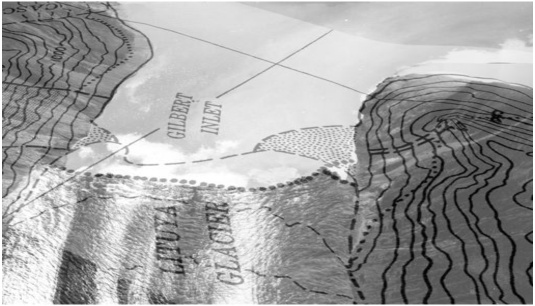
Gilbert Inlet Glacier front reconstruction

- Prior to the 1958 low deltas of gravel had built out into Gilbert Inlet at the southwest and northeast margins of the Lituya Glacier front.
- After the 1958 tsunami the delta on the northeast side of Gilbert Inlet completely disappeared, and the delta on the southwest side was much smaller.
- According to [Miller, 1960], the glacier surface for several hundred feet from the front was severely crevassed.

Lituya Bay: Topo-bathymetry reconstruction

Gilbert Inlet Glacier front reconstruction

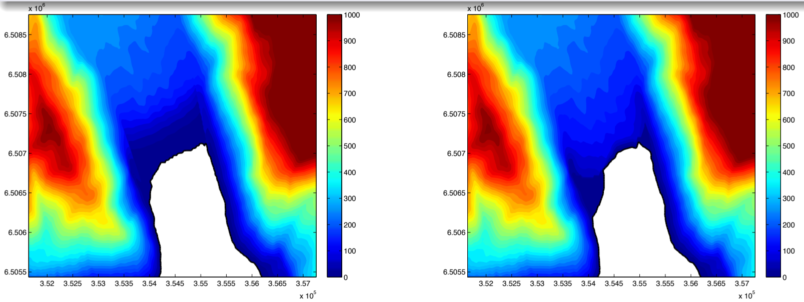
- Prior to the 1958 low deltas of gravel had built out into Gilbert Inlet at the southwest and northeast margins of the Lituya Glacier front.
- After the 1958 tsunami the delta on the northeast side of Gilbert Inlet completely disappeared, and the delta on the southwest side was much smaller.
- According to [Miller, 1960], the glacier surface for several hundred feet from the front was severely crevassed.



Lituya Bay: Topo-bathymetry reconstruction

Gilbert Inlet Glacier front reconstruction

- Prior to the 1958 low deltas of gravel had built out into Gilbert Inlet at the southwest and northeast margins of the Lituya Glacier front.
- After the 1958 tsunami the delta on the northeast side of Gilbert Inlet completely disappeared, and the delta on the southwest side was much smaller.
- According to [Miller, 1960], the glacier surface for several hundred feet from the front was severely crevassed.

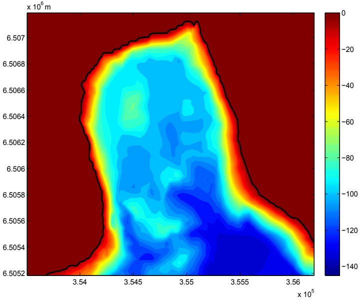


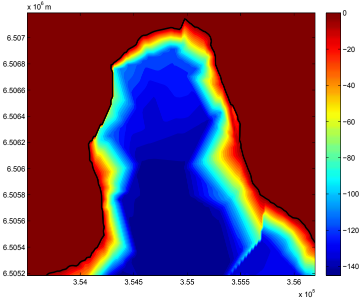
Gilbert Inlet shoreline and glacier front before and after the1958 tsunami.

Lituya Bay: reconstruction of the bathymetry

Bathymetric data (National Ocean Service (USA))

- Hydrographic Survey ID: H08492 (Digital sounding), 1959.
- Hydrographic Survey ID: H04608, 1926.



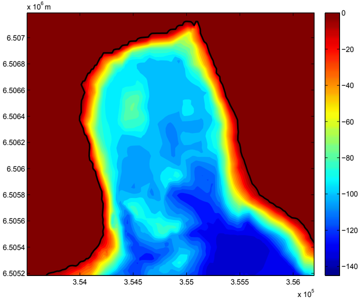


Gilbert Inlet bathymetries and shorelines before and after 1958.

Lituya Bay: reconstruction of the bathymetry

Bathymetric data (National Ocean Service (USA)

- Hydrographic Survey ID: H08492 (Digital sounding), 1959.
- Hydrographic Survey ID: H04608, 1926.



Gilbert Inlet bathymetries and shorelines before and after 1958.

Lituya Bay: Model setup

Model setup

- Initially water is at rest.
- Initial water level is set to 1.52 m (datum approximate mean sea level).
- A 4 m $\times$ 7.5 m rectangular grid with $3,650 \times 1,271 = 4,639,150$ cells has been designed in order to perform the simulation.
- A high efficient GPU based implementation of the model has been developed to be able to compute high accuracy simulations in reasonable computational time.
- The numerical simulation presented here lasts for 10 min.
- $14,516$ time iterations were required to obtain the 10 minutes simulation, requiring about $1,296.5$ seconds in nVidia GTX 580 card. This means that over $51.9 * 10^6$ cell are processed per second.

Lituya Bay: Parameter sensitivity analysis

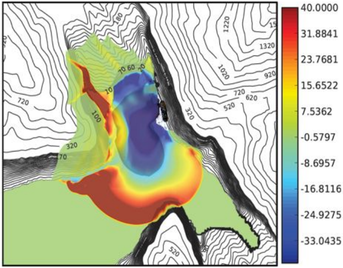
Parameters to be considered

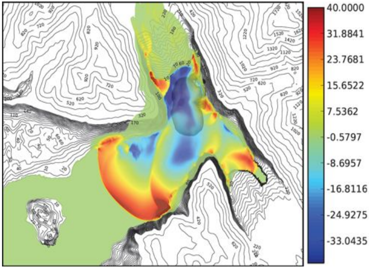
- Coulomb friction angle $\alpha \in [10, 16]$ degrees).
- Ratio of densities between water and the mean density of the slide $r \in [0.3, 0.5]$.
- Friction between layers $m_f \in [0.001, 0.1]$.

Lituya Bay: Parameter sensitivity analysis

Criteria selected in order to get the optimal parameters:

- the runup on the spur southwest of Gilbert Inlet had to be the closest to the optimal 524 m,
- the wave moving southwards towards the main stem of Lituya bay had a peak close to 208 m in the vicinity of Mudslide Creek,
- the simulated wave had run over through the Cenotaph Island, opening a narrow channel through the trees [Miller, 1960],
- the trimline maximum distance of 1,100 m from high-tide shoreline at Fish Lake had to be reached.

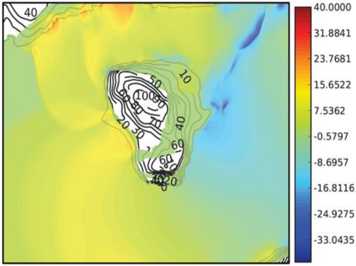


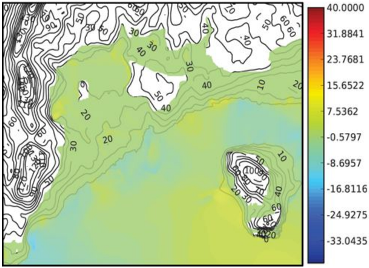


Lituya Bay: Parameter sensitivity analysis

Criteria selected in order to get the optimal parameters:

- the runup on the spur southwest of Gilbert Inlet had to be the closest to the optimal 524 m,
- the wave moving southwards towards the main stem of Lituya bay had a peak close to 208 m in the vicinity of Mudslide Creek,
- the simulated wave had run over through the Cenotaph Island, opening a narrow channel through the trees [Miller, 1960],
- the trimline maximum distance of 1,100 m from high-tide shoreline at Fish Lake had to be reached.





Parameter sensitivity analysis

Optimal parameters found:

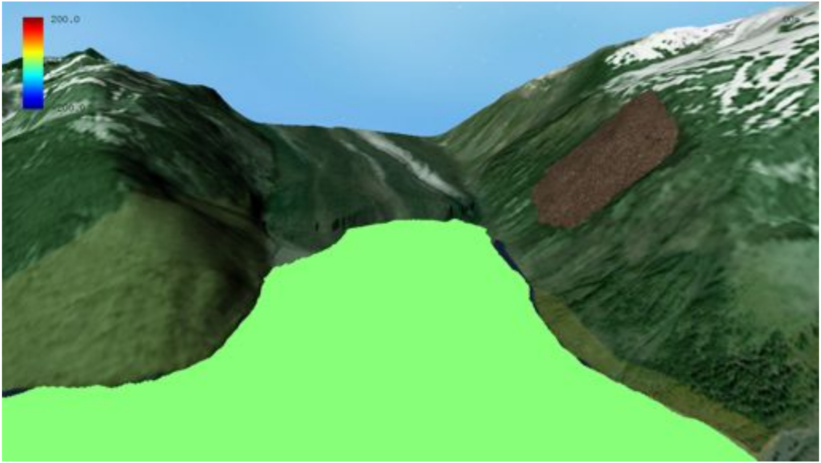
$$\begin{aligned} \alpha &= 13 \text{ degrees}, \\ r &= 0.44, \\ m_f &= 0.08. \end{aligned}$$

Lituya Bay: Settings

- Simulated time: 10 min.
- Mesh resolution: 3650x1271 = 4639150 volumes (4m $\times$ 7.5m).
- CFL: 0.9
- Ratio of densities: $r = 0.44$
- Coulomb angle of repose: 13°.
- Coef. Friction between layers: 0.08.
- Coef. Friction water-bottom: $2 \cdot 10^{-3}$.
- Velocity of impact sediment/water 110 m/s.

Lituya Bay: Model results

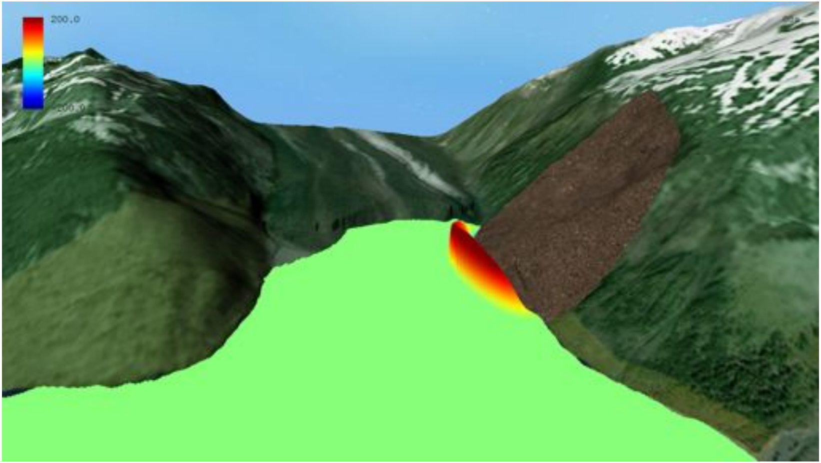
Giant wave generation



$t = 0$ s.

Lituya Bay: Model results

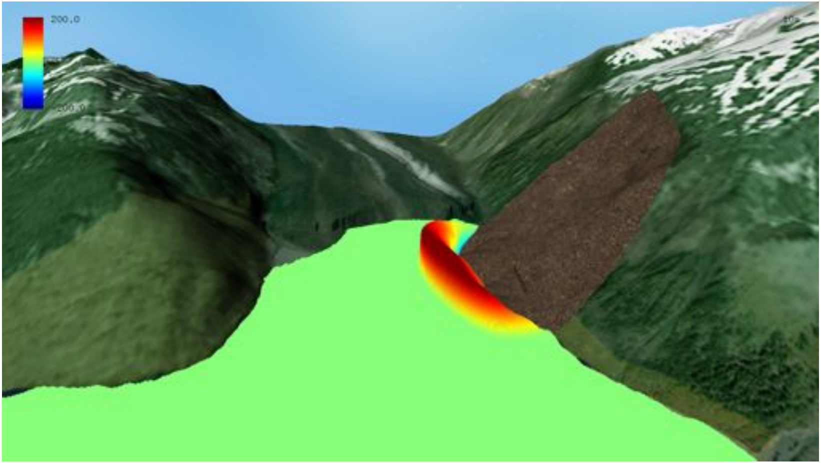
Giant wave generation



$t = 8$ s. The giant wave reaches 272.4m height

Lituya Bay: Model results

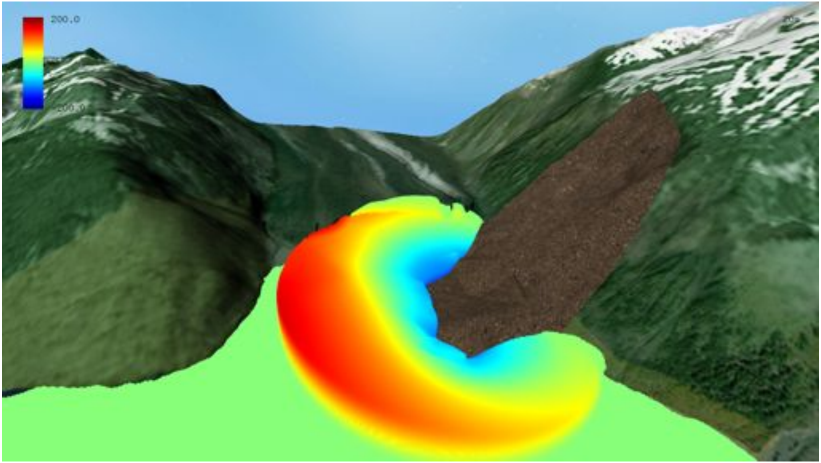
Giant wave generation



$t = 10$ s. The giant wave reaches 251.1m height

Lituya Bay: Model results

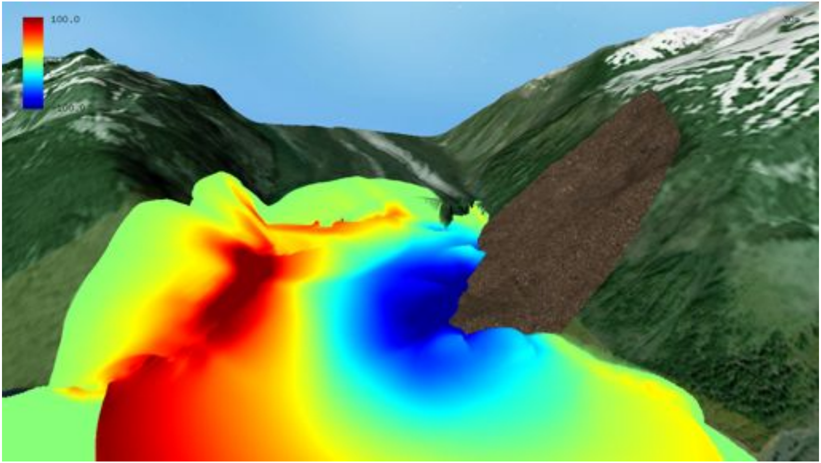
Giant wave generation



$t = 20$ s. Maximum wave height reaches 161.5m height

Lituya Bay: Model results

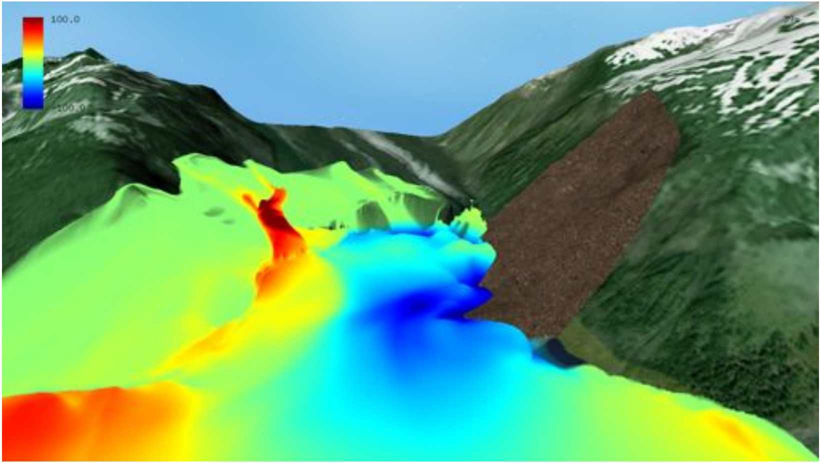
Giant wave generation



$t = 30$ s. Giant wave hits the spur southwest of Gilbert Inlet

Lituya Bay: Model results

Giant wave generation



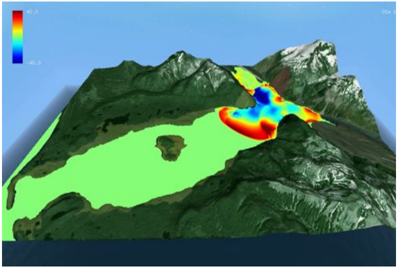
$t = 39$ s. Maximum runup: 523.9 m

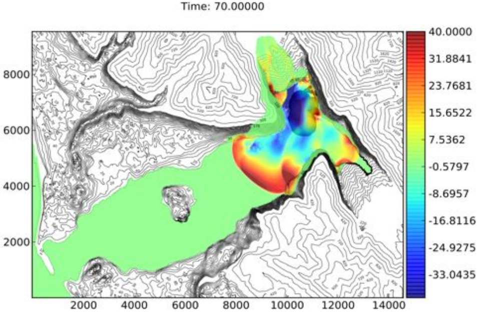
Lituya Bay: Model results

Wave evolution

Lituya Bay: Model results

Wave evolution. Some snapshots

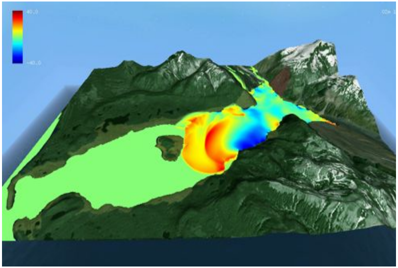


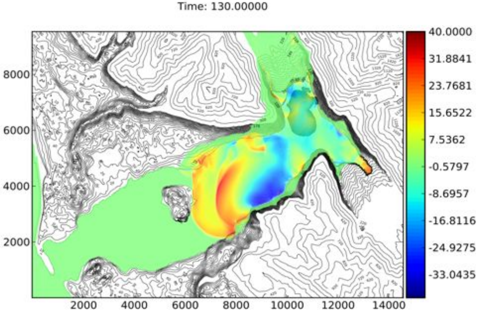


$t = 1$ m 10 s. Waves amplitude.

Lituya Bay: Model results

Wave evolution. Some snapshots

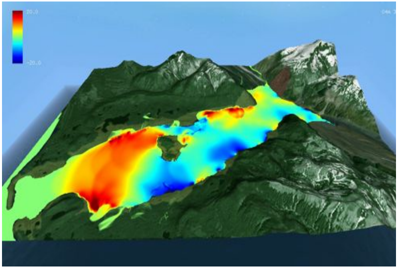


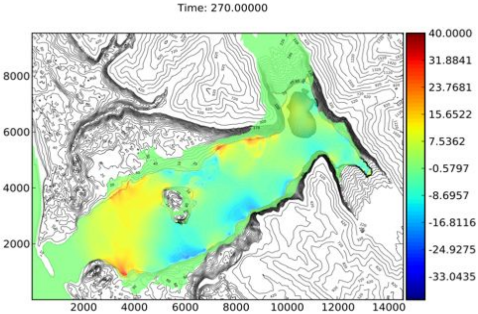


$t = 2$ m 10 s. Waves amplitude.

Lituya Bay: Model results

Wave evolution. Some snapshots

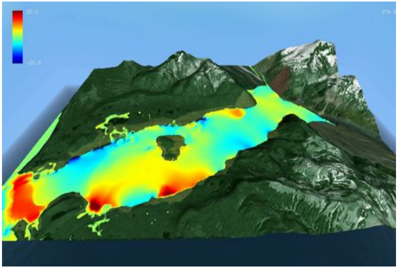


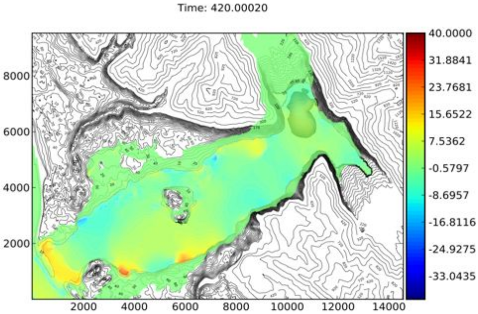


$t = 4$ m 30 s. Waves amplitude.

Lituya Bay: Model results

Wave evolution. Some snapshots

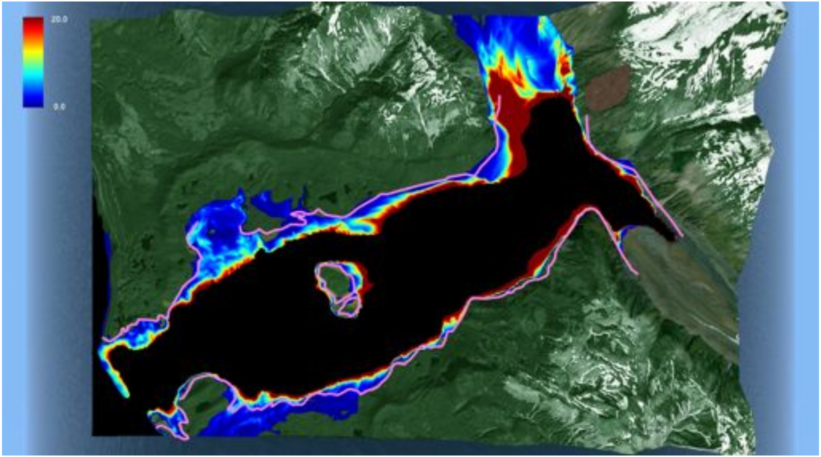




$t = 7$ m. Waves amplitude.

Lituya Bay: Model results

Inundation area and trimline comparison



Total inundation areas and trimline

Thanks

Webpage:

http://edanya.uma.es

Youtube Channel:

http://youtube.com/grupoedanya