Numerical simulation of tsunamis

Frédéric Dias, Denys Dutykh, Raphaël Poncet

CMLA, ENS Cachan & CEA–DAM

GDT ENS Cachan Bretagne, 3 décembre 2008

Outline

Free surface flows

Tsunamis : gravity waves created by earthquakes or submarine landslides
$\longrightarrow$ free-surface flow

Full fluid equations for free surface flows :

- two-phase Navier-Stokes equations (air/water)
- free surface potential Euler equations

- limited to individual waves propagating over a few meters
- computationally very intensive (3-dimensional,...)

$\longrightarrow$ necessity of finding simplified equations

Long wave approximation I

Characteristic lengths (during propagation) :

- wavelength λ : 100 km
- water depth h_0 : 4 km
- wave amplitude a : 0.5 m

Dimensionless parameters :

- nonlinearity parameter $\varepsilon = \dfrac{a}{h_0} \simeq 10^{-4}$.
- dispersion parameter $\mu^2 = \left(\dfrac{h_o}{\lambda}\right)^2 \simeq 10^{-4}$.

N.B. linear dispersion relation (constant depth h)

$$\omega(k) = \frac{2\pi}{\lambda(k)} = \sqrt{g|\vec{k}|\tanh(|\vec{k}|h)}.$$

Long wave approximation II

Asymptotic regimes for free surface potential equations :

- retain only terms in $O(\varepsilon + \mu^2)$
  $\longrightarrow$ Boussinesq equations
- retain only terms in $O(\varepsilon)$
  $\longrightarrow$ nonlinear shallow-water equations
- retain only terms in $O(1)$
  $\longrightarrow$ linear shallow water equations

Outline

Nonlinear shallow water equations

- integrated along the vertical coordinate
  $\longrightarrow$ 2-dimensional model
- the free surface is assumed to be a function $z = \eta(x, y, t)$
- dispersive effects neglected
- hyperbolic system of conservation laws

Nonlinear shallow water equations

$\eta(x, y, t)$: free surface amplitude, $h(x, y, t)$: bathymetry, $H = h + \eta$.

Mass conservation :

$$\partial_t H + \nabla \cdot (H\vec{u}) = 0,$$

Momentum conservation :

$$\partial_t(H\vec{u}) + \nabla \cdot (H\vec{u} \otimes \vec{u}) + \nabla(g\frac{H^2}{2}) = gH\nabla h + S(H, \vec{u}),$$

where S is a source term modeling for instance :

- Coriolis force : $S(H, \vec{u}) = \Omega \times \vec{u}$
- bottom friction, using semi-empirical engineering formulas :
  - Chézy law $S(H, \vec{u}) = -C_f g \vec{u} |\vec{u}|$
  - Manning-Strickler law, Darcy-Weisbach law ...

Systems of conservation laws

Rewrite equations as :

$$\partial_t w + \nabla \cdot (\mathcal{F}(w)) = \mathcal{S}(w),$$

where

- w : vector of conservative variables, $w = (H, H\vec{u})$
- $\mathcal{F}$: advection flux, $\mathcal{F}(H, \vec{u}) = (H\vec{u}, H\vec{u} \otimes \vec{u} + g\frac{H^2}{2}\mathrm{Id})$
- $\mathcal{S}$: source terms, $\mathcal{S}(H, \vec{u}) = (0, gH\nabla h + S(H, \vec{u}))$

Finite volumes framework

Integrate on a control volume :

$$\frac{d}{dt}\int_{\Omega} w\, d\Omega - \int_{\partial K} \mathcal{F}(w)\cdot \vec{n}\, d\sigma = \int_K \mathcal{S}(w)\, d\Omega.$$

Introduce cell averages (cell centered finite volumes) :

$$w_K(t) = \int_{\Omega} w(t,.)\, d\Omega.$$

Question : express the normal fluxes $(\mathcal{F}\cdot\vec{n})|_{\partial K}$ in terms of $\{w_K\}_{K\in\Omega}$
$\longrightarrow$ numerical fluxes

Finite volume framework – numerical fluxes

FVCF : finite volumes with characteristic fluxes
([Ghidaglia, Kumbaro, Le Coq '96])
Flux across the triangle edge shared by triangles K and L is :

$$\Phi(w_K, w_L, \vec{n}_{KL}) = \frac{\mathcal{F}_n(w_K) + \mathcal{F}_n(w_L)}{2} - U(\mu, \vec{n}_{KL})\frac{\mathcal{F}_n(w_K) - \mathcal{F}_n(w_L)}{2}$$

where μ is a mean state :

$$\mu = \frac{vol(K)w_K + vol(L)w_L}{vol(K) + vol(L)}$$

and U is the sign matrix :

$$U(w, \vec{n}) = sign(\mathbb{A}_n) = Rsign(\Lambda)R^{-1}, \qquad \mathbb{A}_n = \frac{\partial \mathcal{F} \cdot n(w)}{\partial w}$$

Remark : in our case, U can be computed analytically

Finite volume framework – numerical fluxes

Other numerical flux implemented : HLL (Harten, Lax, Van Leer '83)

Finite volume framework – 2$^{\text{nd}}$ order spatial discretisation

MUSCL type 2$^{\text{nd}}$ order discretisation

Search w in the class of affine-by-cell functions :

$$w_K(\vec{x}, t) = w_K + (\nabla w)|_K(\vec{x} - \vec{x_0}),$$

where x_0 is the barycenter of K

Gradient $(\nabla w)_K$ is reconstructed from $\{w_K\}_{K \in \Omega}$

- least square method

Need a slope limiter (finite volumes for NL hyperbolic systems)

- Barth-Jespersen limiter

Numerical approximation – additional difficulties

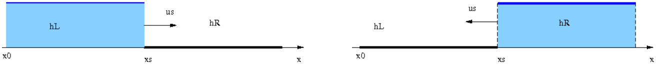
2 additional difficulties (crucial for tsunami applications) :

- Runup and rundown : at $H = 0$, the system loses its hyperbolicity
- source term $gH\nabla h$: numerical instabilities arise for steep bathymetry gradients if (some) static solutions are not discretely preserved
  $\longrightarrow$ « well-balanced » schemes

Numerical approximation – additional difficulties

runup/rundown treatment :

- Specific Riemann problem for dry/wet interface :



$$\frac{dx_s}{dt} := u_s = u_L + 2\sqrt{gh_L} \qquad \frac{dx_s}{dt} := u_s = u_R - 2\sqrt{gh_R}$$

source term treatment

- « well-balanced » scheme : modified H variable in fluxes computation (scheme stays conservative) ([Audusse '05])

Time integration

Strong stability preserving Runge-Kutta schemes SSP–RK
[Gottlieb & Shu '98] :

- explicit in time
- wide stability region
- nonlinearly stable and optimal for CFL

More precisely, we use SSP–RK4 (3) ([Spiteri & Ruuth '02]), 3$^{\text{rd}}$ 4-stage scheme with $\text{CFL} = 2$:

$$
\begin{aligned}
u^1 &= u^n + \tfrac{1}{2} dtL(u^n), \\
u^2 &= u^1 + \tfrac{1}{2} dtL(u^1), \\
u^3 &= \tfrac{2}{3} u^n + \tfrac{1}{3} u^2 + \tfrac{1}{6} dtL(u^n), \\
u^{n+1} &= u^3 + \tfrac{1}{2} dtL(u^3).
\end{aligned}
$$

Outline

Introduction

VOLNA : research code for the numerical simulation of water waves, used for prototyping operational codes

Domains of applications

- tsunami simulations (seismic, landslides)
- storm waves simulations (surges)

Needs

- computational domains of $\simeq$ 10 to 100 wavelengths
- precise, efficient
- robust
- tradeoff between precision/efficiency
- handle realistic scenarios

VOLNA

VOLNA code

preprocessing $\longrightarrow$ solver $\longrightarrow$ postprocessing

Python, Gmsh, librairies | C++ | VTK, Paraview

Features

- solves the nonlinear shallow water equations
- order 2 in space, 3 in time
- semi-automated preprocessing for data acquisition
- arbitrary varying in time bathymetry & boundary conditions
- unstructured meshes :
  - complex geometries
  - static adaptivity (refinement near shorelines, steep bathy. gradients,. . .)

    → precision where needed

    → can partially alleviate numerical difficulties

Preprocessing

Relies on open source libraries, and the `Python` scripting language

- data acquisition
  - IO : arbitrary geospatial data handling → GDAL/OGR
  - geographic to local coordinates projection → Proj.4
  - scattered data interpolation : natural neighbor interpolation → Pavel Sakov, `http://www.marine.csiro.au/~sak007/`
- complex geometries :
  - boolean operations on plane surfaces → GEOS
  - meshing → Gmsh

Outline

Numerical results

- validation against 4 test cases, 2 analytical, 2 experimental
- statically refined meshes (in shallow waters) are systematically used to reduce computational time.

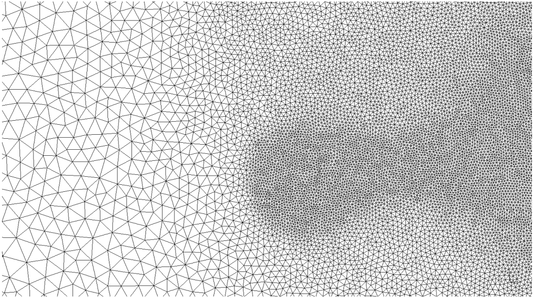


FIG.: mesh for Catalina 2 test case

- no bottom friction (e.g. no free parameter)

Catalina 1 – analytical

- runup on a plane sloping beach
- comparison with analytical solution
  [Carrier & Greenspan '58], [Carrier, Wu & Yeh '03]
- steep test case (strong bathymetry gradient) :

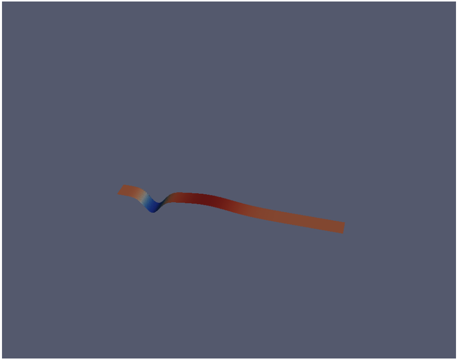


FIG.: initial data

Catalina 1 – analytical

- runup on a plane sloping beach
- comparison with analytical solution
  [Carrier & Greenspan '58], [Carrier, Wu & Yeh '03]
- steep test case (strong bathymetry gradient) :

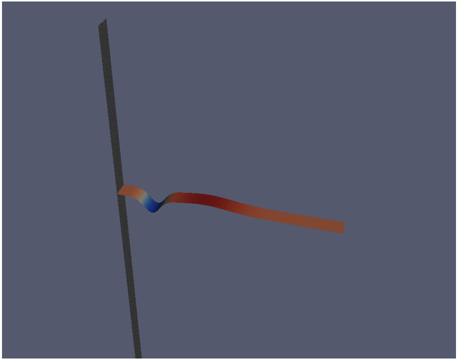
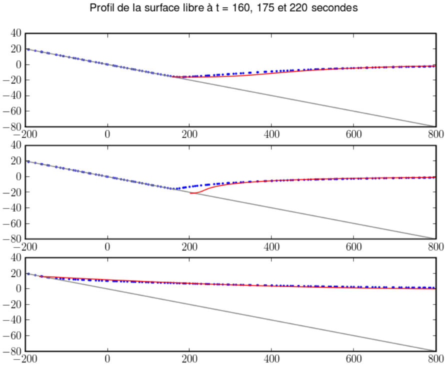


FIG.: initial data + bathymetry

Catalina 1 – analytical

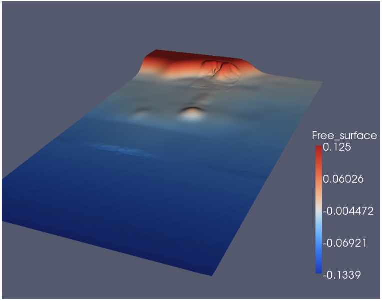


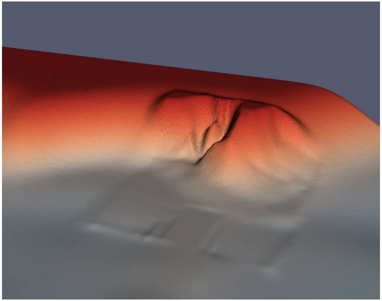
- globally good agreement
- discrepancies near the shoreline (VOLNA results comparable to other shallow water codes) ; may be due to
  - numerics
  - the fact that the analytical solution is not exact

Catalina 2 – experimental

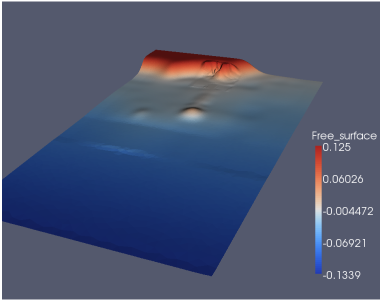
- 205 m long wave tank experiment
- reproduces at 1/400th scale the Okushiri tsunami (Japan, 1993)
- Complex 3D bathymetry

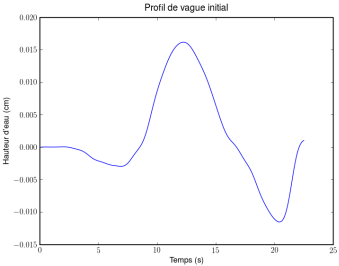
Catalina 2 – experimental



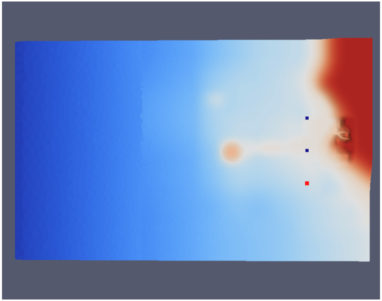


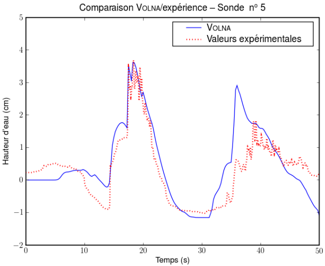
Catalina 2 – experimental



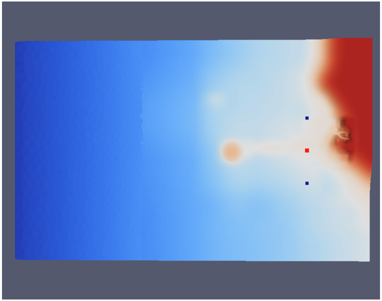


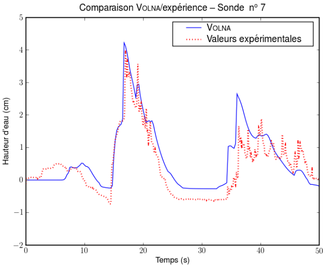
Catalina 2 – experimental



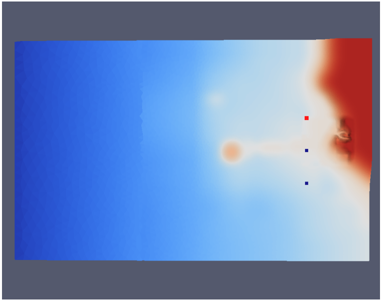


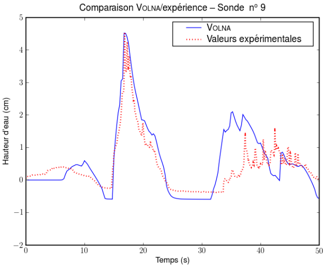
Catalina 2 – experimental





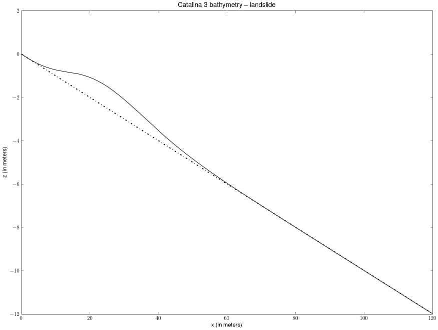
Catalina 2 – experimental





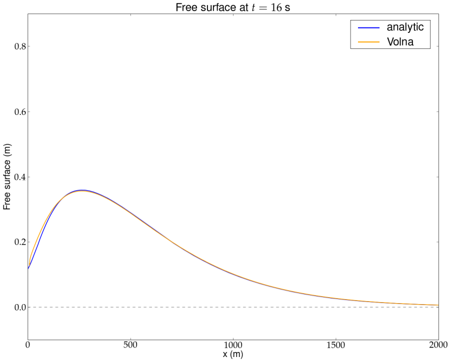
Catalina 3 – analytical landslide

- Comparison with analytical solution of the linearized shallow-water equations with a moving bottom (forcing term)



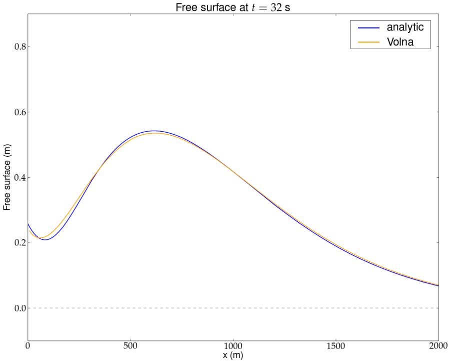
Catalina 3 – analytical landslide

- Comparison with analytical solution of the linearized shallow-water equations with a moving bottom (forcing term)



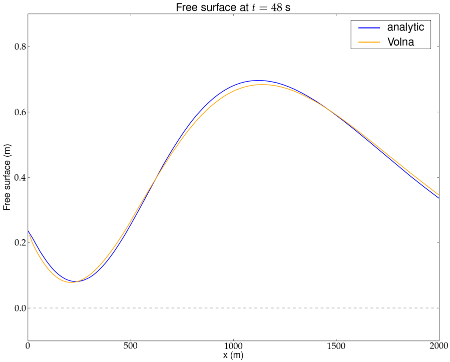
Catalina 3 – analytical landslide

- Comparison with analytical solution of the linearized shallow-water equations with a moving bottom (forcing term)



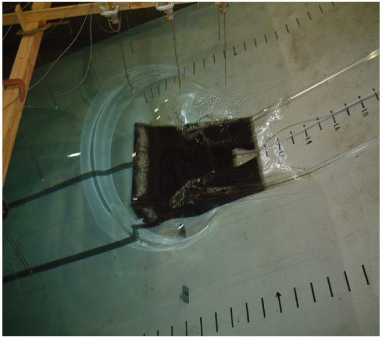
Catalina 3 – analytical landslide

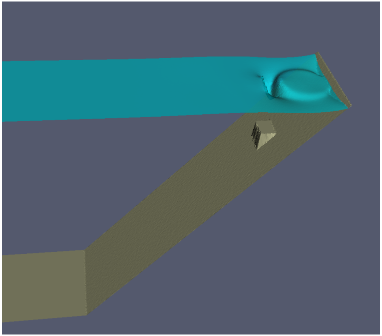
- Comparison with analytical solution of the linearized shallow-water equations with a moving bottom (forcing term)



Catalina 4 – wave tank landslide

- comparison with wave tank experiments ([Synolakis & Raichlen '03])
- difficult test case for NSWE equations

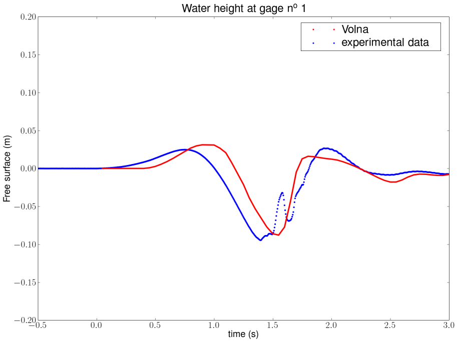




Catalina 4 – wave tank landslide

Preliminary results

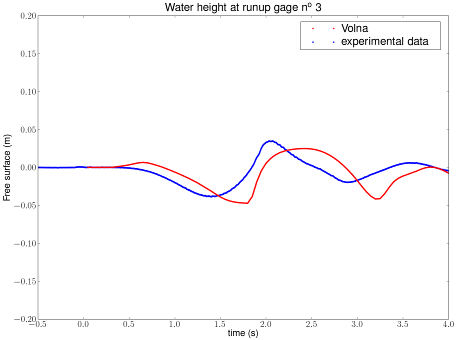
- reasonably good agreement except directly above the wedge (not shown)
- wave breaking, resolution too coarse, limits of NSWE ?



Catalina 4 – wave tank landslide

Preliminary results

- reasonably good agreement except directly above the wedge (not shown)
- wave breaking, resolution too coarse, limits of NSWE ?



Conclusion

- NSWE solved by FVM method with specific treatment of runup and source terms realizes a good trade-off between accuracy, robustness and efficiency for tsunami simulations
- static mesh refinement helps to further alleviate numerical incertainties and save computational time.

Outline

Dispersive effects

- inclusion of dispersive effects (Boussinesq equations) in a finite volumes framework

Tsunamis simulations

- Java tsunami (2006)
  - Initial data : Okada solution (elastodynamics)
  - bathymetry : GEBCO dataset

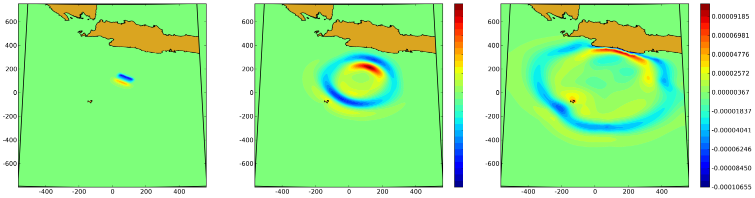
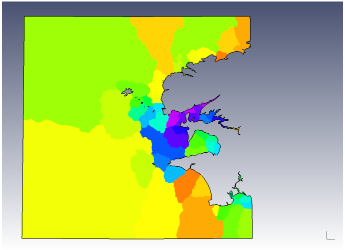


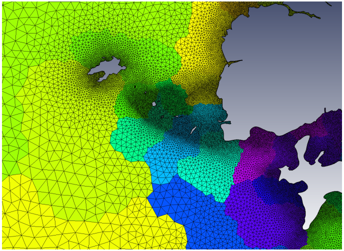
FIG.: Free surface (km) at $t = 0$, 15 and 30 minutes

- underwater landslide tsunami scenario in the Saint-Laurent river

Parallelization

- domain decomposition : split the mesh in sub meshes of equal size while minimizing boundary length
  → PARMETIS library





- inter-processes communications
  → MPI (graph topology) + « ghost cells »

Free surface potential equations