



HAL
open science

The VOLNA code for the numerical modelling of tsunami waves: generation, propagation and inundation

Denys Dutykh, Raphaël Poncet, Frédéric Dias

► To cite this version:

Denys Dutykh, Raphaël Poncet, Frédéric Dias. The VOLNA code for the numerical modelling of tsunami waves: generation, propagation and inundation. 2010. hal-00454591v2

HAL Id: hal-00454591

<https://hal.science/hal-00454591v2>

Preprint submitted on 23 Feb 2010 (v2), last revised 30 May 2011 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The VOLNA code for the numerical modelling of tsunami waves: generation, propagation and inundation

Denys Dutykh¹, Raphaël Poncet², and Frédéric Dias^{3*}

*Centre de Mathématiques et de Leurs Applications,
ENS Cachan and CNRS, UniverSud, 61 avenue du Président Wilson,
F-94235 Cachan Cedex, and LRC MESO, ENS Cachan, CEA DAM DIF*

¹ *Now at Université de Savoie, Laboratoire de Mathématiques LAMA - UMR 5127
Campus Scientifique, 73376 Le Bourget-du-Lac Cedex, France*

² *Now at CEA, DAM, DIF, F-91297 Arpaçon, France*

³ *On leave at University College Dublin, School of Mathematical Sciences, Belfield,
Dublin 4, Ireland*

Abstract

A novel tool for tsunami wave modelling is presented. This tool has the potential of being used for operational purposes: indeed, the numerical VOLNA code is able to handle the complete life-cycle of a tsunami (generation, propagation and run-up along the coast). The algorithm works on unstructured triangular meshes and, thus, can be run in arbitrary complex domains. It is often the case since natural coasts tend to be of fractal shape [SBG04]. This paper contains the detailed description of the finite volume scheme implemented in the code. We explain the numerical treatment of the wet/dry transition. This point is crucial for accurate run-up computation. Most existing tsunami codes use semi-empirical techniques at this stage, which are not always sufficient. The main reason is that people evacuation is decided on the base of inundation maps which are produced with this type of numerical tools. Finally we present several realistic test cases that partially validate our algorithm. Comparisons with analytical solutions and experimental data are performed. Finally the main conclusions are outlined and the perspectives for future research presented.

Key words: tsunami waves, shallow water equations, tsunami generation, run-up, finite volumes

* Corresponding author.

Email addresses: `Denys.Dutykh@univ-savoie.fr` (Denys Dutykh¹),
`raphael.poncet@gmail.com` (Raphaël Poncet²),
`Frederic.Dias@cmla.ens-cachan.fr` (Frédéric Dias³).

1 Introduction

After the Boxing Day tsunami [SB06] there is no need to explain the importance of research on tsunami waves. One of the primary objectives in this field consists in establishing and developing Tsunami Warning Systems (TWS) [Tat97,TGB⁺05]. This task is non trivial as explained by Synolakis [Syn05]:

For reference, the United States and Japan took more than 20 years to develop *validated numerical models* to predict tsunami evolution. And it took the US National Oceanic and Atmospheric Administration 30 years to fully develop its bottom-pressure recorders, which have been reliably detecting tsunamis for the past ten years.

After the Boxing Day tsunami, while developing their own national and regional capabilities, countries in the Indian Ocean and the Caribbean Sea have asked the PTWC (Pacific Tsunami Warning Center) to act as their interim warning center. India and Australia now have fully working national centers, while the National Oceanic and Atmospheric Administration of the U.S. has assisted both with instrumentation and the sophisticated forecast technology used in the Pacific. Europe however is trying to reinvent the early warning wheel. As a result, the Mediterranean remains the only world ocean or sea unprotected by any warning system.

The mathematical modelling and computation of propagating tsunami waves play an important rôle in TWS. Precision and robustness of the algorithm will affect performance and reliability of the whole system.

The importance of tsunami generation modelling is often underestimated by the scientific community. During several years the research of our group was focused on this topic and many interesting results were obtained [DD07c,Dut07], [DDK06,KDD07,DD09b,DD10]. We tried to incorporate some recent developments [DD07c] from this field into the *VOLNA* code.

It is difficult to find a topic in numerical analysis of hyperbolic PDEs which has been studied more than the numerical solution to the Nonlinear Shallow Water Equations (NSWE). The numerical scheme presented in this paper is not completely novel. The discretization methods used in *VOLNA* can be found in the modern literature on finite volumes methods [Kro97,BO04]. The main purpose here is to present a tool for tsunami wave modelling which covers the whole spectrum from generation to inundation. The emphasis is on the technical work which is typical of a numerical analyst and software developer. Tsunami practitioners can then concentrate on the physical aspects of tsunami propagation.

Nowadays, one is facing a somewhat strange situation. On one hand, there are only a few truly operational codes for tsunami wave modelling: MOST, NAMI, ComCot [Ima96, TG97, GOSI97, LWC98]. The numerical schemes used in these codes essentially correspond to the state of the art of the eighties. On the other hand, there is a plethora of NSWE codes developed in academic environments [Gla88, Cas90, Tor92, BDDV98, AC99, VC99, AGN05, BQ06, GNV00, GHS03], [ABB⁺04, KCY07, Geo06, GL06, ZCIM02, CIM⁺00, NPPN06, CFGR⁺05, WMC06], [Geo08, DKK07]. These codes use modern numerical methods but most of them run have been developed for rectangular domains and Cartesian meshes. They can be successfully used to test new numerical ideas, but these tools have little interest for tsunami operational research. This is why we had the idea to develop *VOLNA*. We tried to combine modern numerical techniques for hyperbolic systems with real world application-oriented design. The *VOLNA* code can be run efficiently in realistic environments. It was shown that natural coasts tend to have fractal forms [SBG04]. Hence, unstructured meshes are a natural choice in this type of situations.

The present article is organised as follows. In Section 2 the physical context of the study and some motivation for the choice of the mathematical model are presented. Section 3 contains a detailed description of the numerical method implemented in the *VOLNA* code. Then, we show in Section 4 some computations, which validate and illustrate the capabilities of *VOLNA*. Finally the main conclusions are outlined and the perspectives for future research presented in Section 5.

2 Physical context and mathematical model

In this study we focus on long wave propagation over realistic bathymetry. A sketch of the physical problem under investigation is given on Fig. 1. Let us explain the main assumptions and the domain of applicability of the *VOLNA* code.

Introduce some characteristic lengths. We denote by a_0 the typical wave amplitude, by h_0 the average depth and by ℓ the characteristic wave length. Several dimensionless numbers can be built from these three quantities, but traditionally one introduces the following two:

$$\varepsilon = \frac{a_0}{h_0}, \quad \mu^2 = \left(\frac{h_0}{\ell}\right)^2.$$

The first parameter ε measures the wave nonlinearity ($\varepsilon \ll 1$ means than nonlinearity is weak) while the second parameter μ^2 quantifies the importance of dispersive effects ($\mu \ll 1$ means than dispersion is weak). Taking a typical

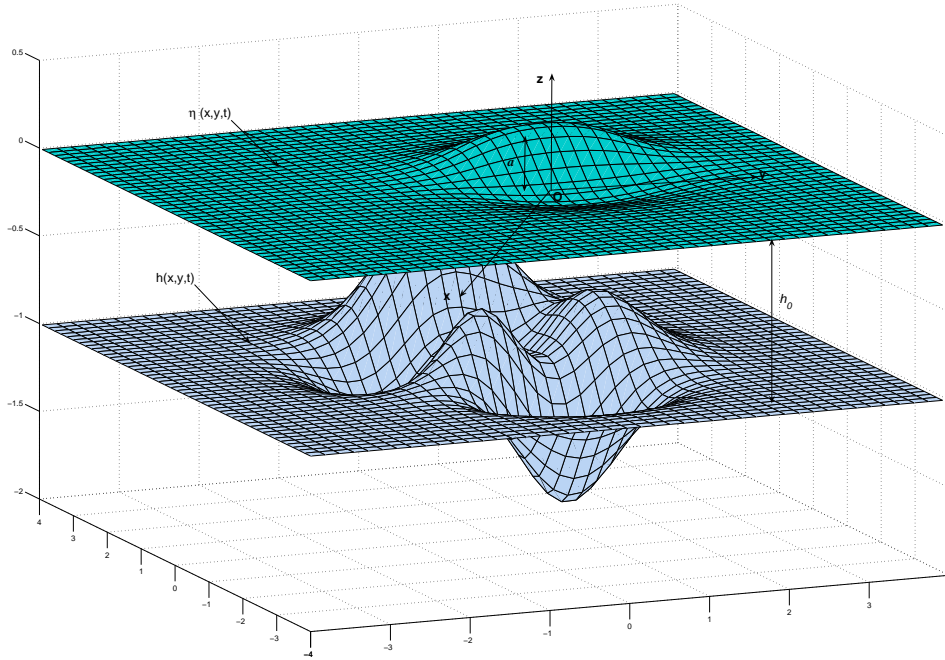


Fig. 1. Sketch of the fluid domain.

megatsunami offshore with roughly

$$a_0 \simeq 0.5 \text{ m}, \quad h_0 \simeq 4 \text{ km}, \quad \ell \simeq 100 \text{ km}$$

yields $\varepsilon = 1.25 \times 10^{-4}$ for the nonlinearity parameter and $\mu^2 = 1.6 \times 10^{-3}$ for the dispersion parameter. Both are weak. Using asymptotic expansions in the small parameters $\varepsilon \ll 1$ and $\mu^2 \ll 1$, one can derive Serre-type equations [Ser53,Per67,MBS03,DD07a,DM10]. Note that no operational codes based on the Serre equations exist. For example FUNWAVE is not a truly operational code. Neglecting the dispersive effects yields the classical Nonlinear Shallow Water Equations (NSWE):

$$H_t + \nabla \cdot (H\vec{u}) = 0, \tag{1}$$

$$(H\vec{u})_t + \nabla \cdot \left(H\vec{u} \otimes \vec{u} + \frac{g}{2} H^2 \right) = gH\nabla h, \tag{2}$$

where $H = h + \eta$ is the total water depth and $\vec{u} = (u, v)(\vec{x}, t)$ is the depth-averaged horizontal velocity. Traditionally, g denotes the acceleration due to the gravity and $h(\vec{x}, t)$ describes the bathymetry.

Remark 1 *The bathymetry $h(\vec{x}, t)$ is allowed to be time-dependent. It is important for the problem of tsunami generation by underwater earthquakes, submarine landslides, etc. The coupling with seismology is done through this function. Namely, various simplified earthquake models [DD07c,KDD07,DD09b],[DD10] provide the seabed displacements which are then transmitted to the ocean layer.*

In this study, the NSWE (1) and (2) are chosen to model tsunami generation, propagation and run-up. It is computationally advantageous to have a uniform model for all stages of tsunami life since many technical problems are thus avoided. The validity of the NSWE for tsunami generation was already examined in our previous study [KDD07], where an excellent performance of this model was shown for nondispersive long waves. In the present paper we show in Section 4 the ability of the NSWE to model the run-up process. For this purpose, comparisons with a laboratory experiment are performed. Thus, the chosen *complete* approach to tsunami wave modelling is very attractive from both the operational and research viewpoints.

The governing equations (1) and (2) have several nice mathematical properties. In particular, this system is strictly hyperbolic provided that $H > 0$. This property will be used extensively in the construction of the numerical scheme (see Section 3).

Let us discuss the eigensystem of the advective flux. First, we introduce conservative variables and rewrite the governing equations as a system of conservation laws:

$$\frac{\partial w}{\partial t} + \nabla \cdot \mathcal{F}(w) = \mathcal{S}(w), \quad (3)$$

where the following notation was introduced:

$$w(x, t) : \mathbb{R}^2 \times \mathbb{R}^+ \mapsto \mathbb{R}^3, \quad w = (w_1, w_2, w_3) = (H, Hu, Hv),$$

$$\mathcal{F}(w) = \begin{pmatrix} Hu & Hv \\ Hu^2 + \frac{g}{2}H^2 & Huv \\ Huv & Hv^2 + \frac{g}{2}H^2 \end{pmatrix} = \begin{pmatrix} w_2 & w_3 \\ \frac{w_2^2}{w_1} + \frac{g}{2}w_1^2 & \frac{w_2w_3}{w_1} \\ \frac{w_2w_3}{w_1} & \frac{w_3^2}{w_1} + \frac{g}{2}w_1^2 \end{pmatrix}, \quad \mathcal{S}(w) = \begin{pmatrix} 0 \\ gH \frac{\partial h}{\partial x} \\ gH \frac{\partial h}{\partial y} \end{pmatrix}.$$

After projecting the flux $\mathcal{F}(w)$ on a normal direction¹ $\vec{n} = (n_x, n_y)$, one can compute the Jacobian matrix \mathbb{A}_n . Its expression in physical variables has the following form:

$$\mathbb{A}_n = \frac{\partial(\mathcal{F}(w) \cdot \vec{n})}{\partial w} = \begin{pmatrix} 0 & n_x & n_y \\ -uu_n + gHn_x & u_n + un_x & un_y \\ -vu_n + gHn_y & vn_x & u_n + vn_y \end{pmatrix},$$

where $u_n = un_x + vn_y$ is the velocity vector projected on \vec{n} . The Jacobian matrix \mathbb{A}_n has three distinct eigenvalues:

$$\lambda_1 = u_n - c, \quad \lambda_2 = u_n, \quad \lambda_3 = u_n + c, \quad (4)$$

¹ In the finite volume method one often projects onto the face normal.

where $c = \sqrt{gH}$ is the speed of gravity waves in the limit of infinite wavelength. This quantity plays the same rôle as the sound speed in compressible fluid mechanics. It is now obvious that the system (1), (2) is strictly hyperbolic provided that $H > 0$. The eigenstructure of the Jacobian matrix \mathbb{A}_n is fundamental for constructing the numerical flux function (see Section 3.1) and thus, unwinding the discrete solution.

3 Discretization procedure

In this study we selected the most natural numerical method for this type of equations. Finite volume methods are a class of discretization schemes that have proven highly successful in solving numerically a wide class of systems of conservation laws. These systems often come from compressible fluid dynamics. In electromagnetism, for example, Discontinuous Galerkin methods have proven to be more efficient [CLS04]. When compared to other discretization methods such as finite elements or finite differences, the primary advantages of finite volume methods are robustness, applicability on very general unstructured meshes, and the intrinsic local conservation properties. Hence, with this type of discretization, mass, momentum and total energy are conserved exactly².

In order to solve numerically the system of balance laws (1), (2) one uses again the conservative form of governing equations (3). System (3) should be provided with an initial condition

$$w(\vec{x}, 0) = w_0(\vec{x}), \quad \vec{x} = (x, y) \in \Omega \quad (5)$$

and appropriate boundary conditions. The implementation of different boundary conditions will be discussed below (see Section 3.7).

3.1 First order scheme

The computational domain $\Omega \subset \mathbb{R}^2$ is triangulated into a set of non overlapping control volumes that completely cover the domain. Let \mathcal{T} denote a tessellation of the domain Ω with control volume K such that

$$\cup_{K \in \mathcal{T}} \bar{K} = \bar{\Omega}, \quad \bar{K} := K \cup \partial K.$$

For two distinct control volumes K and L in \mathcal{T} , the intersection is an edge with oriented normal \vec{n}_{KL} or else a vertex. We need to introduce the following

² This statement is true in the absence of source terms and appropriate boundary conditions.

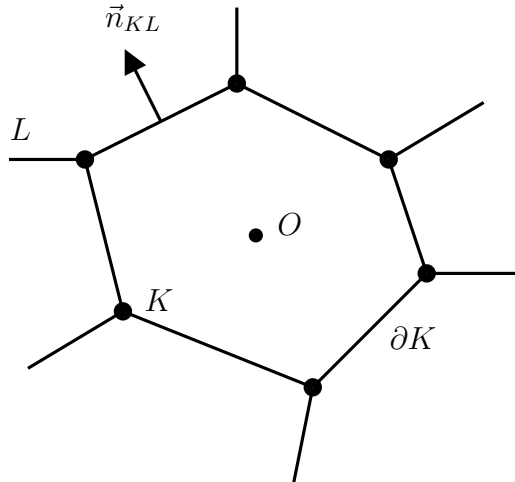


Fig. 2. An example of control volume K with barycenter O . The normal pointing from K to L is denoted by \vec{n}_{KL} .

notation for the neighbourhood of K :

$$\mathcal{N}(K) := \{L \in \mathcal{T} : \text{area}(K \cap L) \neq 0\},$$

a set of all control volumes L which share an edge in 2D or a face in 3D with the given volume K . In this study, we denote by $\text{vol}(\cdot)$ and $\text{area}(\cdot)$ the area and length respectively.

The choice of control volume tessellation is flexible in the finite volume method. In the present study we selected the cell-centered approach (see Fig. 3), which means that degrees of freedom are associated to cell barycenters. There exists an alternative vertex-centered method [BJ89,BO04] (see Fig. 4) which requires the construction of dual mesh even for first-order schemes. In the *cell-centered* finite volume scheme, the triangles themselves serve as control volumes with solution unknowns attributed to triangles barycenters. In the *vertex-centered* finite volume scheme, control volumes are formed as a geometric dual to the triangle complex and solution unknowns stored on vertex basis.

Remark 2 *Except for the construction of dual mesh in the vertex-centered approach, these two methods are almost equivalent in the interior of the computational domain Ω . However, the boundary conditions treatment is different and it is harder (or less natural in the authors' opinion) when data is stored at vertices. This is one more reason why we selected cell centers to store the solution's information.*

The first steps in Finite Volume (FV) methods are classical. One starts by integrating equation (3) on the control volume K (see Fig. 2 for illustration) and one applies Gauss-Ostrogradsky theorem for advective and diffusive fluxes. Then, in each control volume, an integral conservation law statement

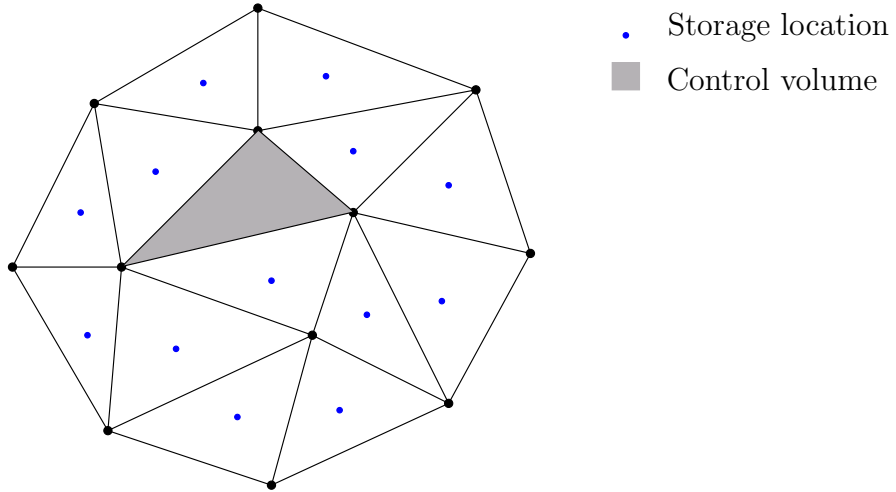


Fig. 3. Illustration for cell-centered finite volume method

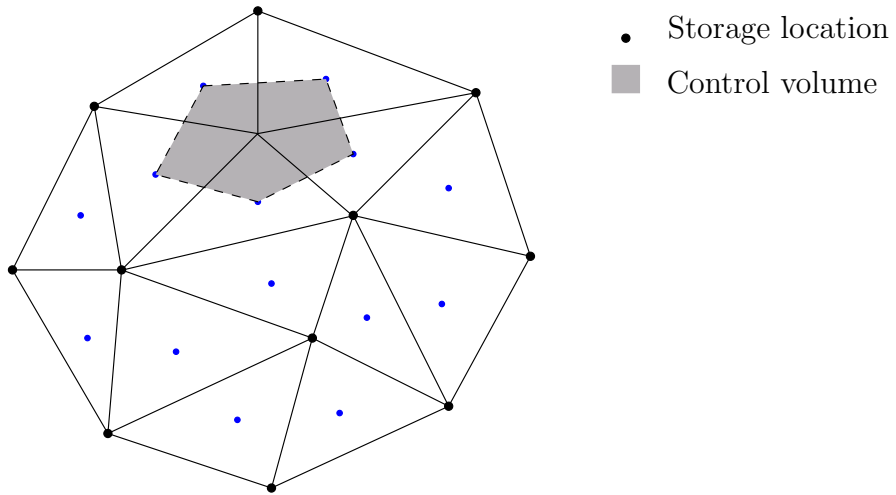


Fig. 4. Illustration for vertex-centered finite volume method

is imposed:

$$\frac{d}{dt} \int_K w \, d\Omega + \int_{\partial K} \mathcal{F}(w) \cdot \vec{n}_{KL} \, d\sigma = \int_K \mathcal{S}(w) \, d\Omega \quad (6)$$

Physically an integral conservation law states that the rate of change of the total amount of a quantity (for example: mass, momentum, total energy) with density w in a fixed control volume K is balanced by the flux \mathcal{F} of the quantity through the boundary ∂K and the production of this quantity \mathcal{S} inside the control volume.

The next step consists in introducing the control volume cell average for each $K \in \mathcal{T}$

$$w_K(t) := \frac{1}{\text{vol}(K)} \int_K w(\vec{x}, t) \, d\Omega .$$

After the averaging step, the finite volume method can be interpreted as pro-

ducing a system of evolution equations for cell averages, since

$$\frac{d}{dt} \int_K w(\vec{x}, t) d\Omega = \text{vol}(K) \frac{dw_K}{dt} .$$

Godunov was first [God59] to pursue and apply these ideas to the discretization of the gas dynamics equations.

However, the averaging process implies piecewise constant solution representation in each control volume with value equal to the cell average. The use of such a representation makes the numerical solution multivalued at control volume interfaces. Thereby the calculation of the fluxes $\int_{\partial K} (\mathcal{F}(w) \cdot \vec{n}_{KL}) d\sigma$ at these interfaces is ambiguous. A fundamental aspect of finite volume methods is the idea of substituting the true flux at interfaces by a numerical flux function

$$\left(\mathcal{F}(w) \cdot \vec{n} \right) \Big|_{\partial K \cap \partial L} \longleftarrow \Phi(w_K, w_L; \vec{n}_{KL}) : \mathbb{R}^3 \times \mathbb{R}^3 \mapsto \mathbb{R}^3 ,$$

a Lipschitz continuous function of the two interface states w_K and w_L . The key ingredient is the choice of the numerical flux function Φ . In general this function is calculated as an exact or even better approximate local solution of the Riemann problem posed at these interfaces. In the present study we implemented several numerical fluxes (HLL, HLLC, FVCF) described below.

Any numerical flux is assumed to satisfy the following properties:

Conservation. This property ensures that fluxes from adjacent control volumes sharing an interface exactly cancel when summed. This is achieved if the numerical flux function satisfies the identity

$$\Phi(w_K, w_L; \vec{n}_{KL}) = -\Phi(w_L, w_K; \vec{n}_{LK}).$$

Consistency. Consistency is obtained when the numerical flux with identical state arguments (in other words it means that the solution is continuous through an interface) reduces to the true flux of the same state, i.e.

$$\Phi(w, w; \vec{n}) = (\mathcal{F}(w) \cdot \vec{n})(w).$$

In the following paragraphs 3.1.1 – 3.1.3 we give several examples of numerical flux functions $\Phi(\vec{v}, w; \vec{n})$ which were implemented in the *VOLNA* code. These choices are justified by efficiency, clarity and personal preferences of the authors. However, we do not impose them and a final user can easily implement his favourite numerical flux function.

3.1.1 FVCF approach

First we describe the scheme called Finite Volumes with Characteristic Flux (FVCF) and proposed by Ghidaglia et al. in [Ghi95,GKC96,GKC01].

Consider a general system of conservation laws in 1D that can be written as follows:

$$\frac{\partial w}{\partial t} + \frac{\partial f(w)}{\partial x} = 0, \quad (7)$$

where $w \in \mathbb{R}^m$ and $f : \mathbb{R}^m \mapsto \mathbb{R}^m$. We denote by $A(w)$ the Jacobian matrix $\frac{\partial f(w)}{\partial w}$ and we deal with the case where (7) is *smoothly hyperbolic*, that is to say: for every w there exists a smooth basis $(r_1(w), \dots, r_m(w))$ of \mathbb{R}^m consisting of eigenvectors of $A(w)$. That is $\exists \lambda_k(w) \in \mathbb{R}$ such that $A(w)r_k(w) = \lambda_k(w)r_k(w)$. It is then possible to construct $(l_1(w), \dots, l_m(w))$ such that ${}^t A(w)l_k(w) = \lambda_k(w)l_k(w)$ and $l_k(w) \cdot r_p(w) = \delta_{k,p}$.

Let $\mathbb{R} = \cup_{j \in \mathbb{Z}} [x_{j-1/2}, x_{j+1/2}]$ be a one dimensional mesh. The goal is to discretize (7) by a finite volume method. We set $\Delta x_j \equiv x_{j+1/2} - x_{j-1/2}$, $\Delta t_n \equiv t_{n+1} - t_n$ (we also have $\mathbb{R}_+ = \cup_{n \in \mathbb{N}} [t_n, t_{n+1}]$) and

$$\tilde{w}_j^n \equiv \frac{1}{\Delta x_j} \int_{x_{j-1/2}}^{x_{j+1/2}} w(x, t_n) dx, \quad \tilde{f}_{j+1/2}^n \equiv \frac{1}{\Delta t_n} \int_{t_n}^{t_{n+1}} f(w(x_{j+1/2}, t)) dt.$$

With these notations, we deduce from (7) the *exact* relation:

$$\tilde{w}_j^{n+1} = \tilde{w}_j^n - \frac{\Delta t_n}{\Delta x_j} (\tilde{f}_{j+1/2}^n - \tilde{f}_{j-1/2}^n). \quad (8)$$

Since the $(\tilde{f}_{j+1/2}^n)_{j \in \mathbb{Z}}$ cannot be expressed in terms of the $(\tilde{w}_j^n)_{j \in \mathbb{Z}}$, one has to make an approximation. In order to keep a compact stencil, it is more efficient to use a three point scheme: the physical flux $\tilde{f}_{j+1/2}^n$ is approximated by a numerical flux $g_j^n(w_j^n, w_{j+1}^n)$. Let us show how this flux is constructed here. Since $A(w) \frac{\partial w}{\partial t} = \frac{\partial f(w)}{\partial t}$ we observe that according to (7)

$$\frac{\partial f(w)}{\partial t} + A(w) \frac{\partial f(w)}{\partial x} = 0. \quad (9)$$

This shows that the flux $f(w)$ is advected by $A(w)$ like w . The numerical flux $g_j^n(w_j^n, w_{j+1}^n)$ represents the flux at an interface. Using a mean value $\mu_{j+1/2}^n$ of w at this interface, we replace (9) by the linearization:

$$\frac{\partial f(w)}{\partial t} + A(\mu_{j+1/2}^n) \frac{\partial f(w)}{\partial x} = 0. \quad (10)$$

It follows that, defining the k -th characteristic flux component to be $f_k(w) \equiv$

$l_k(\mu_{j+1/2}^n) \cdot f(w)$, one has

$$\frac{\partial f_k(w)}{\partial t} + \lambda_k(\mu_{j+1/2}^n) \frac{\partial f_k(w)}{\partial x} = 0. \quad (11)$$

This linear equation can be solved explicitly now:

$$f_k(w)(x, t) = f_k(w)(x - \lambda_k(\mu_{j+1/2}^n)(t - t_n), t_n). \quad (12)$$

From this equation it is then natural to introduce the following definition.

Definition 1 For the conservative system (7), at the interface between the two cells $[x_{j-1/2}, x_{j+1/2}]$ and $[x_{j+1/2}, x_{j+3/2}]$, the characteristic flux g^{CF} is defined by the following formula for $k \in \{1, \dots, m\}$:

(we take $\mu_{j+1/2}^n \equiv (\Delta x_j w_j^n + \Delta x_{j+1} w_{j+1}^n) / (\Delta x_j + \Delta x_{j+1})$)

$$l_k(\mu_{j+1/2}^n) \cdot g_j^{CF,n}(w_j^n, w_{j+1}^n) = l_k(\mu_{j+1/2}^n) \cdot f(w_j^n), \text{ when } \lambda_k(\mu_{j+1/2}^n) > 0,$$

$$l_k(\mu_{j+1/2}^n) \cdot g_j^{CF,n}(w_j^n, w_{j+1}^n) = l_k(\mu_{j+1/2}^n) \cdot f(w_{j+1}^n), \text{ when } \lambda_k(\mu_{j+1/2}^n) < 0, \quad (13)$$

$$l_k(\mu_{j+1/2}^n) \cdot g_j^{CF,n}(w_j^n, w_{j+1}^n) = l_k(\mu_{j+1/2}^n) \cdot \left(\frac{f(w_{j+1}^n) + f(w_j^n)}{2} \right),$$

when $\lambda_k(\mu_{j+1/2}^n) = 0$.

Remark 3 At first glance, the derivation of (9) from (7) is only valid for continuous solutions since $A(w) \frac{\partial f(w)}{\partial x}$ is a non conservative product. In fact, equation (9) can be justified even in the case of shocks as proved in [Ghi98]. Let us briefly recall here the key point. Assuming that the solution undergoes a discontinuity along a family of disjoint curves, we can focus on one of these curves that we parameterize by the time variable t . Hence, locally, on each side of this curve, $w(x, t)$ is smooth and jumps across the curve $x = \Sigma(t)$. The Rankine-Hugoniot condition implies that $f(w(x, t)) - \sigma(t)w(x, t)$, where $\sigma(t) \equiv \frac{d\Sigma(t)}{dt}$, is smooth across the discontinuity curve and therefore $A(w) \frac{\partial f(w)}{\partial x}$ can be defined as $A(w) \frac{\partial f(w)}{\partial x} \equiv A(w) \frac{\partial (f(w) - \sigma w)}{\partial x} + \sigma \frac{\partial f(w)}{\partial x}$.

Proposition 1 Formula (13) can be written as follows: $g_j^{CF,n}(w_j^n, w_{j+1}^n) = g^{CF}(\mu_j^n; w_j^n, w_{j+1}^n)$ where

$$g^{CF}(\mu; v, w) \equiv \sum_{\lambda_k(\mu) < 0} (l_k(\mu) \cdot f(w)) r_k(\mu) + \sum_{\lambda_k(\mu) = 0} \left(l_k(\mu) \cdot \frac{f(v) + f(w)}{2} \right) r_k(\mu) + \sum_{\lambda_k(\mu) > 0} (l_k(\mu) \cdot f(v)) r_k(\mu). \quad (14)$$

Proof. This comes from the useful identity valid for all vectors Φ and μ in \mathbb{R}^m :

$$\Phi = \sum_{k=1}^{k=m} (l_k(\mu) \cdot \Phi) r_k(\mu).$$
 We also observe that (14) can be written under the

following condensed form:

$$g^{CF}(\mu; v, w) = \frac{f(v) + f(w)}{2} - U(\mu; v, w) \frac{f(w) - f(v)}{2}, \quad (15)$$

where $U(\mu; v, w)$ is the sign of the matrix $A(\mu)$ which is defined by

$$\text{sign}(A(\mu))\Phi = \sum_{k=1}^m \text{sign}(\lambda_k)(l_k(\mu) \cdot \Phi)r_k(\mu).$$

The form (15) refers to what we have called a numerical flux leading to a flux scheme [Ghi98]. ■

Remark 4 *Let us discuss the relation, in the conservative case, between the characteristic numerical flux g^{CF} and the numerical flux leading to Roe's scheme [Roe81]. The latter scheme relies on an algebraic property of the continuous flux $f(w)$ which is as follows. It is assumed that for all admissible states v and w , there exists a $m \times m$ matrix $A^{ROE}(v, w)$ such that $f(v) - f(w) = A^{ROE}(v, w)(v - w)$ (Roe's identity). Then the numerical flux leading to Roe's scheme is given by:*

$$g^{ROE}(v, w) = \frac{f(v) + f(w)}{2} - |A^{ROE}(v, w)| \frac{w - v}{2}. \quad (16)$$

But using Roe's identity, we obtain that

$$g^{ROE}(v, w) = \frac{f(v) + f(w)}{2} - \text{sign}(A^{ROE}(v, w)) \frac{f(w) - f(v)}{2}, \quad (17)$$

which is of the form (15): Roe's scheme is also a flux scheme. The characteristic flux proposed in this paper is more versatile than Roe's scheme in the sense that it does not rely on an algebraic property of the flux. Hence for complex systems (like those encountered in the context of two phase flows) this scheme appears like an efficient generalization of Roe's scheme. Moreover, as we shall see below, this scheme has a natural generalization to arbitrary non conservative systems. Finally, the fact that the numerical flux is a linear combination of the two fluxes induces a quite weak dependence on the state μ which appears in formula (14), see [CG00].

3.1.2 HLL numerical flux

Now we present another approximate Riemann solver which was proposed by Harten, Lax and van Leer [HLvL83]. Nowadays this method is known as the HLL scheme. While the exact solution to the Riemann problem contains a large amount of detail, the HLL solver assumes fewer intermediate waves. The simplified Riemann fan is illustrated on Figure 5. It consists of two waves separating three constant states.

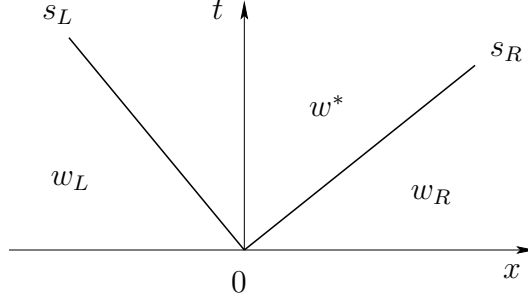


Fig. 5. Approximate Riemann fan corresponding to the HLL scheme.

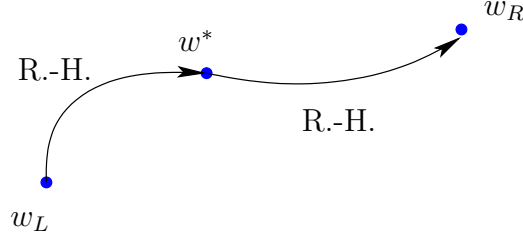


Fig. 6. Two states w_L and w_R connected by Rankine-Hugoniot curves represented in the phase space.

Consider the following Riemann problem:

$$\mathcal{R}(w_L, w_R) : \begin{cases} \frac{\partial w}{\partial t} + \frac{\partial F(w)}{\partial x} = 0, \\ w(x, 0) = \begin{cases} w_L, & x < 0, \\ w_R, & x > 0. \end{cases} \end{cases} \quad (18)$$

The intermediate state in the approximate Riemann fan will be denoted by w^* and two shock wave speeds are denoted by s_L and s_R respectively (see Figures 5 and 6 for illustration). In order to determine the unknown intermediate state, we write the Rankine-Hugoniot conditions twice:

$$\begin{cases} s_L(w^* - w_L) = F^* - F_L, \\ s_R(w_R - w^*) = F_R - F^*, \end{cases}$$

where $F_{L,R} := F(w_{L,R})$. It is straightforward to find the solution to this system:

$$w^* = \frac{s_R w_R - s_L w_L - (F_R - F_L)}{s_R - s_L},$$

$$F^* = F_L + s_L(w^* - w_L) = \frac{s_R F_L - s_L F_R + s_L s_R (w_R - w_L)}{s_R - s_L}. \quad (19)$$

Now we have all the elements to define the numerical flux of the HLL scheme:

$$\Phi_{HLL}(w_L, w_R) := \begin{cases} F_L, & s_L \geq 0, \\ F^*, & s_L < 0 \leq s_R, \\ F_R, & s_R < 0. \end{cases}$$

During the presentation of the HLL scheme we missed one important point: how to estimate the wave speeds s_L and s_R ? The answer is crucial for the overall performance of the scheme. With appropriate choices for the wave speeds s_L and s_R , the HLL scheme possesses very nice numerical properties. Namely, it satisfies an entropy inequality [Dav88], resolves isolated shocks exactly [HLvL83] and preserves positivity [EMRS91]. In our code we implemented the following choice for s_L and s_R which is motivated by analytical expressions for the Jacobian eigenvalues (4):

$$s_L = \min(u_L - c_L, u^* - c^*), \quad s_R = \min(u^* + c^*, u_R + c_R),$$

where $c_{L,R} := \sqrt{gH_{L,R}}$ is the gravity wave speed for the left and right states and

$$u^* = \frac{1}{2}(u_L + u_R) + c_L - c_R, \quad c^* = \frac{1}{2}(c_L + c_R) - \frac{1}{4}(u_R - u_L).$$

Numerical experiments show that this approximate Riemann solver is very robust with the above choice for the wave speeds [CIM⁺00,ZCIM02].

However, the HLL scheme has one important shortcoming: it cannot resolve isolated contact discontinuities. In the next section 3.1.3 we present another scheme which was designed to remedy this problem.

Remark 5 *Does the HLL scheme belong to the class of flux schemes or not? Recall that a finite volume scheme is called a flux scheme if its numerical flux can be written in the following form:*

$$\Phi = \frac{F(w_L) + F(w_R)}{2} - U(w_L, w_R) \frac{F(w_R) - F(w_L)}{2},$$

where $U(w_L, w_R)$ is some matrix. From the formula (19) for the flux F^* it follows that the answer depends on the existence of a matrix M such that

$$w_R - w_L = M(F_R - F_L) \quad \text{or} \quad F_R - F_L = M^{-1}(w_R - w_L).$$

Fortunately, the answer is positive for NSW E and the matrix M can be effectively constructed. It allows us to rewrite the HLL numerical flux in the

shorthand form $\Phi_{HLL} = U_R F_R + U_L F_L$, where

$$U_R = \begin{cases} 0, & s_L \geq 0, \\ \frac{s_L s_R}{s_R - s_L} M - \frac{s_L}{s_R - s_L} \mathbb{I}, & s_L < 0 \leq s_R, \\ \mathbb{I}, & s_R < 0, \end{cases} \quad U_L = \begin{cases} \mathbb{I}, & s_L \geq 0, \\ -\frac{s_L s_R}{s_R - s_L} M + \frac{s_R}{s_R - s_L} \mathbb{I}, & s_L < 0 \leq s_R, \\ 0, & s_R < 0, \end{cases}$$

with $\mathbb{I} := (\delta_{ij})$. In part, the robustness of the HLL scheme can be explained by this nice property.

3.1.3 HLLC flux

The HLL scheme presented briefly in the previous section was later improved by Toro, Spruce and Speares [TSS94]. Their modification concerns essentially the structure of the Riemann fan which is depicted on Figure 7. Namely, they introduced a contact discontinuity between two shock waves of the HLL scheme. That is why the novel scheme was called the HLLC scheme [FT95].

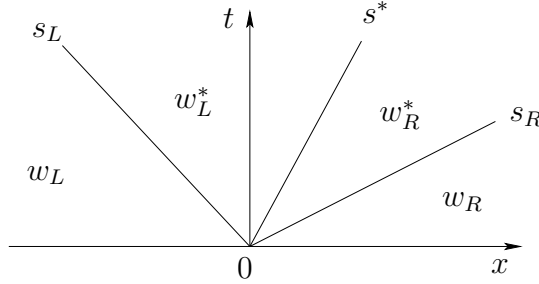


Fig. 7. Approximate Riemann fan corresponding to the HLLC scheme.

Here we do not provide details on the derivation of the HLLC scheme and refer to the original articles and others which can feel this gap [BCCC97, KCY07].

We consider the same Riemann problem (18). In the HLLC approximation, the solution to this Riemann problem consists of three waves with speeds s_L , s^* and s_R separating four constant states w_L , w_L^* , w_R^* and w_R . Wave speeds $s_{L,R}$ are estimated as in previous section 3.1.2, while s^* is given by this formula:

$$s^* = \frac{s_L H_R(u_R - s_R) - s_R H_L(u_L - s_L)}{H_R(u_R - s_R) - H_L(u_L - s_L)}.$$

The intermediate states $w_{L,R}^* = (H_{L,R}^*, (Hu)_{L,R}^*, (Hv)_{L,R}^*)$ are computed in this way:

$$\begin{aligned} H_{L,R}^* &= \frac{s_{L,R} - u_{L,R}}{s_{L,R} - s^*} H_{L,R}, \\ (Hu)_{L,R}^* &= \frac{s_{L,R} - u_{L,R}}{s_{L,R} - s^*} (Hu)_{L,R} + \frac{g}{2} H_{L,R}^2 \frac{(2s_{L,R} - s^* - u_{L,R})(s^* - u_{L,R})}{(s_{L,R} - s^*)^3}, \\ (Hv)_{L,R}^* &= \frac{s_{L,R} - u_{L,R}}{s_{L,R} - s^*} (Hv)_{L,R} + \frac{g}{2} H_{L,R}^2 \frac{(2s_{L,R} - s^* - u_{L,R})(s^* - u_{L,R})}{(s_{L,R} - s^*)^3}. \end{aligned}$$

Finally, the numerical flux of the HLLC scheme is defined as

$$\Phi_{HLLC}(w_L, w_R) := \begin{cases} F_L, & s_L \geq 0, \\ F_L^* := F_L + s_L(w_L^* - w_L), & s_L < 0 \leq s^*, \\ F_R^* := F_R + s_R(w_R^* - w_R), & s^* < 0 \leq s_R, \\ F_R, & s_R < 0. \end{cases}$$

Remark 6 *As in the previous scheme, one can check whether the HLLC scheme is a flux scheme or not. It turns out that it is.*

3.2 Semidiscrete scheme

After introducing the cell averages w_K and numerical fluxes into (6), the integral conservation law statement becomes

$$\frac{dw_K}{dt} + \sum_{L \in \mathcal{N}(K)} \frac{\text{area}(L \cap K)}{\text{vol}(K)} \Phi(w_K, w_L; \vec{n}_{KL}) = \frac{1}{\text{vol}(K)} \int_K \mathcal{S}(w) d\Omega.$$

We denote by \mathcal{S}_K the approximation of the quantity $\frac{1}{\text{vol}(K)} \int_K \mathcal{S}(w) d\Omega$. The source term discretization is discussed in Section 3.4. Thus, the following system of ordinary differential equations (ODE) is called a semi-discrete finite volume method:

$$\frac{dw_K}{dt} + \sum_{L \in \mathcal{N}(K)} \frac{\text{area}(L \cap K)}{\text{vol}(K)} \Phi(w_K, w_L; \vec{n}_{KL}) = \mathcal{S}_K, \quad \forall K \in \mathcal{T}. \quad (20)$$

The initial condition for this system is given by projecting (5) onto the space of piecewise constant functions

$$w_K(0) = \frac{1}{\text{vol } K} \int_K w_0(x) d\Omega.$$

This system of ODE should also be discretized. There is a variety of explicit and implicit time integration methods. Let w_K^n denote a numerical approximation of the cell average solution in the control volume K at time $t^n = n\Delta t$. The simplest time integration method is the forward Euler scheme

$$\frac{dw_K}{dt} \cong \frac{w_K^{n+1} - w_K^n}{\Delta t}.$$

When applied to (20) it produces the fully-discrete finite volume scheme:

$$\frac{w_K^{n+1} - w_K^n}{\Delta t} + \sum_{L \in \mathcal{N}(K)} \frac{\text{area}(L \cap K)}{\text{vol}(K)} \Phi(w_K^n, w_L^n; \vec{n}_{KL}) = \mathcal{S}_K^n, \quad \forall K \in \mathcal{T}. \quad (21)$$

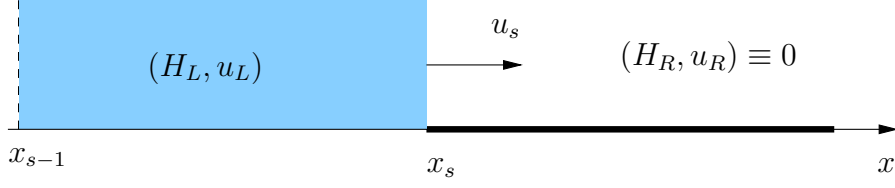


Fig. 8. Shoreline left Riemann problem.

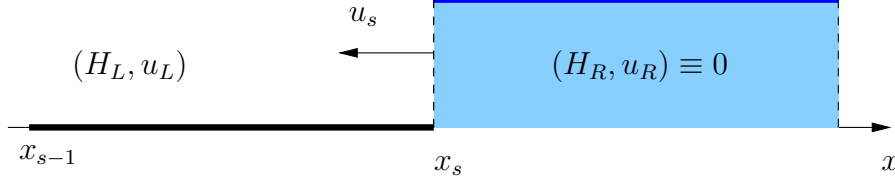


Fig. 9. Shoreline right Riemann problem.

The time discretization used in this study is detailed in Section 3.5.

3.3 Run-up algorithm

As already pointed out above, the NSWs are strictly hyperbolic if $H > 0$, i.e. when some water is present. The shoreline position is given by the following implicit relation $H(\vec{x}, t) = 0$. In these locations the system loses its strict hyperbolicity. Finally, in dry regions, $H < 0$, the system is non-hyperbolic, i.e. ill-posed. All these facts mean that there are some major theoretical difficulties in considering the inundation problem.

Very often some ad-hoc artificial techniques are implemented to circumvent run-up and run-down problems (“slot technique” of Madsen et al. [MSS97], algorithm of Hibberd-Peregrine [HP79], use of coordinate transformations [OHK97] and so on).

Analytically, shoreline boundary conditions have a very simple form:

$$H(\vec{x}_s(t), t) = 0, \quad \frac{d\vec{x}_s}{dt} = \vec{u}(\vec{x}_s(t), t),$$

where $\vec{x}_s(t)$ is the shoreline position.

The algorithm proposed by Brocchini et al. [BBMA01] was chosen for the *VOLNA* code. It is based on the shoreline Riemann problem (see Figures 8 and 9 for illustration) [Sto57]:

$$\mathcal{R}_{\text{left}}(w_L) : \begin{cases} \frac{\partial w}{\partial t} + \frac{\partial F(w)}{\partial x} = 0, \\ w(x, 0) = \begin{cases} w_L, & x < 0, \\ 0, & x > 0. \end{cases} \end{cases} \quad \mathcal{R}_{\text{right}}(w_R) : \begin{cases} \frac{\partial w}{\partial t} + \frac{\partial F(w)}{\partial x} = 0, \\ w(x, 0) = \begin{cases} 0, & x < 0, \\ w_R, & x > 0. \end{cases} \end{cases}$$

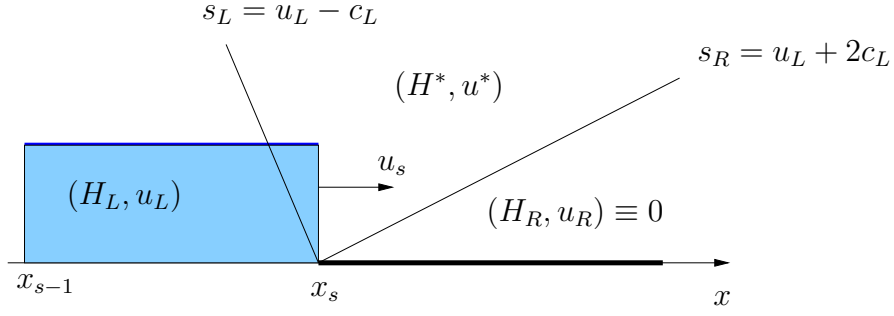


Fig. 10. Shoreline Riemann problem (left) and wave propagation speeds.

The main idea to solve the shoreline Riemann problem is to pass to the limit $w_L \rightarrow 0$ or $w_R \rightarrow 0$ in the solution to the classical Riemann problem (18). Technical details can be found in [BBMA01]. However, we do not need to know the complete solution. It is sufficient to extract the wave propagation speeds at the shoreline (see Figure 10). These analytically determined speeds are imposed in an approximate Riemann solver when a wet/dry transition is detected.

Consider two control volumes K and L which share a common face. We must find the numerical flux $\Phi(w_K, w_L, \vec{n}_{KL})$ across this face. Let us summarize the key points of the method:

Wet/wet interface: If $H_L > 0$ and $H_R > 0$, we apply in the usual way an approximate Riemann solver which gives the numerical flux Φ .

Dry/dry interface: If $H_L = H_R = 0$, we just return the zero flux $\Phi = 0$ since there is no flow between two dry cells.

Wet/dry interface: If $H_R = 0$ and $H_L > 0$, we have a situation corresponding to the left shoreline Riemann problem. Its solution yields the following choice of the wave speeds:

$$s_L := (\vec{u}_L \cdot \vec{n}_{KL}) - c_L, \quad s_R := (\vec{u}_L \cdot \vec{n}_{KL}) + 2c_L.$$

Then we apply the HLL or the HLLC scheme with the above values of s_L and s_R (see Figure 10 for illustration).

Dry/wet interface: If $H_R > 0$ and $H_L = 0$, we have a situation symmetric to the previous case. One must solve the right shoreline Riemann problem. It provides the following speeds:

$$s_L := (\vec{u}_R \cdot \vec{n}_{KL}) - 2c_R, \quad s_R := (\vec{u}_R \cdot \vec{n}_{KL}) + c_R.$$

Here again, the HLL or the HLLC scheme is applied.

We would like to underline the simplicity of this approach. In fact, there is no special treatment for the interface. This algorithm is run uniformly in the whole computational domain leading to an easy and robust implementation. We validate this method in sections 4.2 – 4.4.

3.4 Source terms discretization

In this section we discuss some issues related to the source term discretization and we explain a technique to remedy them.

Source terms of the form $gH\nabla h$ arise in the horizontal momentum conservation equation (2). Obviously, this term is equal to zero when the bottom is even $h = \text{const}$. However, it is not the case in the real world applications. The magnitude of this term is proportional to the bed slope and may take large values when abrupt changes are present in the bathymetry.

Another profound property of NSWSE is that the system (1), (2) admits non-trivial steady states. They can be determined from the following steady equations:

$$\begin{cases} \nabla \cdot (H\vec{u}) = 0, \\ \nabla \cdot \left(H\vec{u} \otimes \vec{u} + \frac{\rho}{2}H^2 \right) = gH\nabla h \end{cases}$$

It is not so trivial to find analytical solutions to these equations. However, an ideal numerical scheme should preserve them. Recall that for 1D flows it is possible to describe the whole family of steady states and this information can be used to design efficient source term discretizations [LR98, VC99]. Since realistic applications require a 2D solver, we will address this problem directly in 2D.

As explained above, it seems to be extremely difficult to construct a scheme which preserves exactly all steady state solutions. Thus, we have to simplify the problem. We will focus our attention on a simple class of steady solutions which are called in the literature “lake at rest”:

$$\vec{u} = 0, \quad \eta := H - h = \text{const}. \quad (22)$$

The last relations can be expressed in discrete variables:

$$\vec{u}_K = \vec{u}_L = 0, \quad H_K - h_K = H_L - h_L = \text{const}. \quad (23)$$

We briefly present the method chosen for our code and developed in [ABB⁺04], [AB05]. It is mainly based on the idea of the interface hydrostatic reconstruction.

The well-balanced algorithm takes as input the vector of conservative variables $\{w_K\}_{K \in \mathcal{T}}$, bathymetry data $\{h_K\}_{K \in \mathcal{T}}$ and is composed of the following steps:

- Assume that the control volumes K and L share a common face $f = K \cap L$. In this case, the interface bathymetry is defined as $h_{KL}^* := \min(h_K, h_L)$. This step is done only once at the initialization stage.

- The hydrostatic reconstructed interface water depth is given by

$$H_{KL}^* = (H_K - h_K + h_{KL}^*)_+, \quad \text{where } z_+ = \max(z, 0).$$

From the discrete interpretation (23) of the well-balanced condition (22), we define a new vector of the interface conservative variables:

$$w_{KL}^* := \begin{pmatrix} H_{KL}^* \\ H_{KL}^* \vec{u}_K \end{pmatrix}. \quad (24)$$

- From the balance of hydrostatic forces $\nabla \left(\frac{g}{2} H^2 \right) = gH \nabla h$, the adapted discretization of the source terms is introduced:

$$\mathcal{S}_K^*(w_K, w_{KL}^*, \vec{n}_{KL}) := \begin{pmatrix} 0 \\ \frac{g}{2} (H_{KL}^{*2} - H_K^2) \vec{n}_{KL} \end{pmatrix}$$

- The well-balanced scheme is obtained by replacing cell-centered values w_K by new interface values (24):

$$\frac{w_K^{n+1} - w_K^n}{\Delta t} + \sum_{L \in \mathcal{N}(K)} \frac{\text{area}(L \cap K)}{\text{vol}(K)} \Phi(w_{KL}^*, w_{LK}^*; \vec{n}_{KL}) = \mathcal{S}_K^*(w_K^n, w_{KL}^*, \vec{n}_{KL}), \quad \forall K \in \mathcal{T}.$$

It can be proven [AB05] that the hydrostatic reconstruction strategy preserves the “lake at rest” solutions and ensures the positivity property. We describe here only the first-order algorithm for the sake of simplicity. The extension to the second order can be found in the original papers and in [Aud04].

3.5 Time discretization

In the previous sections we considered the spatial discretization procedure with a finite-volume scheme. It is a common practice in solving time-dependent PDEs to first discretize the spatial variables. This approach is called method of lines:

$$w_t + \partial_x f(w) = S(w) \xrightarrow{\text{FV}} w_t = \mathcal{L}(w) \quad (25)$$

In order to obtain a fully discrete scheme, we must discretize the time evolution operator. In the present work we chose the so-called Strong Stability-Preserving (SSP) time discretization methods described in [Shu88, GST01, SR02]. Historically these methods were initially called Total Variation Diminishing (TVD) time discretizations.

The main idea behind SSP methods is to assume that the first order forward Euler method is strongly stable (see the definition below) under a certain norm

for the method of lines ODE (25). Then, we try to find a higher order scheme. Usually the relevant norm is the total variation³ norm:

$$\text{TV}(w^n) := \sum_j |w_j^n - w_{j-1}^n|$$

and TVD discretizations have the property $\text{TV}(w^{n+1}) \leq \text{TV}(w^n)$.

Remark 7 *Special approaches are needed for hyperbolic PDEs since they contain discontinuous solutions and the usual linear stability analysis is inadequate. Thus a stronger measure of stability is usually required:*

Definition 2 *A sequence $\{w^n\}$ is said to be **strongly stable** in a given norm $\|\cdot\|$ provided that $\|w^{n+1}\| \leq \|w^n\|$ for all $n \geq 0$.*

A general m -stage Runge-Kutta method for (25) can be written in the form

$$w^{(0)} = w^n, \tag{26}$$

$$w^{(i)} = \sum_{k=0}^{i-1} \left(\alpha_{i,k} w^{(k)} + \Delta t \beta_{i,k} \mathcal{L}(w^{(k)}) \right), \quad \alpha_{i,k} \geq 0, \quad i = 1, \dots, m, \tag{27}$$

$$w^{n+1} = w^{(m)}. \tag{28}$$

In [SO88] the following result is proved

Theorem 1 *If the forward Euler method is strongly stable under the CFL restriction $\Delta t \leq \Delta t_{FE}$*

$$\|w^n + \Delta t \mathcal{L}(w^n)\| \leq \|w^n\|,$$

then the Runge-Kutta method (26) – (28) with $\beta_{i,k} \geq 0$ is SSP, $\|w^{n+1}\| \leq \|w^n\|$, provided the following CFL restriction is fulfilled:

$$\Delta t \leq c \Delta t_{FE}, \quad c = \min_{i,k} \frac{\alpha_{i,k}}{\beta_{i,k}}.$$

Here we give a few examples of SSP schemes which are commonly used in applications:

- Optimal⁴ second order two-stage SSP-RK(2,2) scheme with CFL = 1:

$$\begin{aligned} w^{(1)} &= w^{(n)} + \Delta t \mathcal{L}(w^{(n)}), \\ w^{(n+1)} &= \frac{1}{2} w^{(n)} + \frac{1}{2} w^{(1)} + \frac{1}{2} \Delta t \mathcal{L}(w^{(1)}); \end{aligned}$$

³ The notion of total variation is used essentially for 1D discrete solutions.

⁴ Optimality in the sense of CFL condition.

- Optimal third order three-stage SSP-RK(3,3) scheme with CFL = 1:

$$\begin{aligned}
w^{(1)} &= w^{(n)} + \Delta t \mathcal{L}(w^{(n)}), \\
w^{(2)} &= \frac{3}{4}w^{(n)} + \frac{1}{4}w^{(1)} + \frac{1}{4}\Delta t \mathcal{L}(w^{(1)}), \\
W^{(n+1)} &= \frac{1}{3}w^{(n)} + \frac{2}{3}w^{(2)} + \frac{2}{3}\Delta t \mathcal{L}(w^{(2)});
\end{aligned}$$

- Third order four-stage SSP-RK(3,4) scheme with CFL = 2:

$$\begin{aligned}
w^{(1)} &= w^{(n)} + \frac{1}{2}\Delta t \mathcal{L}(w^{(n)}), \\
w^{(2)} &= w^{(1)} + \frac{1}{2}\Delta t \mathcal{L}(w^{(1)}), \\
w^{(3)} &= \frac{2}{3}w^{(n)} + \frac{1}{3}w^{(2)} + \frac{1}{6}\Delta t \mathcal{L}(w^{(n)}), \\
w^{(n+1)} &= w^{(3)} + \frac{1}{2}\Delta t \mathcal{L}(w^{(3)}).
\end{aligned}$$

The linear absolute stability region for the RK and SSP-RK schemes is the same. However the nonlinear absolute stability regions are quite different [CP92].

We tested these different schemes in our numerical code and we decided to adopt SSP-RK(3,4) due to its accuracy and wide stability region. In our opinion this scheme represents a very good trade-off between precision and robustness.

3.6 Second order extension

If we analyze the above scheme, we understand that in fact, we have only one degree of freedom per data storage location. Hence, it seems that we can expect to be first order accurate at most. In the numerical community first order schemes are generally considered to be too inaccurate for most quantitative calculations. Of course, we can always make the mesh spacing extremely small but it cannot be a solution since it makes the scheme inefficient. From the theoretical point of view the situation is even worse since an $\mathcal{O}(h^{\frac{1}{2}})$ L_1 -norm error bound for the monotone and E-flux schemes [Osh84] is known to be sharp [Pet91], although an $\mathcal{O}(h)$ solution error is routinely observed in numerical experiments. On the other hand, Godunov has shown [God59] that all linear schemes that preserve solution monotonicity are at most first order accurate. This rather negative result suggests that a higher order accurate scheme has to be essentially nonlinear in order to attain simultaneously a monotone resolution of discontinuities and high order accuracy in continuous regions.

A significant breakthrough in the generalization of finite volume methods to higher order accuracy is due to N.E. Kolgan [Kol72,Kol75] and van Leer [vL79]. They proposed a kind of post-treatment procedure currently known as solution *reconstruction* or MUSCL⁵ scheme. In the above papers the authors used linear reconstruction (it will be chosen in this study as well) but this method has already been extended to quadratic approximations in each cell [BF90].

3.6.1 Historical remark

In general, when we read numerical articles which use the MUSCL scheme, the authors often cite the paper by van Leer [vL79]. It is commonly believed in the scientific community that B. van Leer was first to propose the gradient reconstruction and slope limiting ideas. Unfortunately, because of political reasons, the works of N.E. Kolgan [Kol72,Kol75] remained unknown for a long time. We would like to underline the fact that the first publication of Kolgan came out seven years before van Leer's paper. Van Leer seems to be aware of this situation since in his recent review paper [vL06] one can find "A historical injustice" section:

"It has been pointed out to me by Dr. Vladimir Sabelnikov, formerly of TsAGI, the Central Aerodynamical National Laboratory near Moscow, that a scheme closely resembling MUSCL (including limiting) was developed in this laboratory by V. P. Kolgan (1972). Kolgan died young; his work apparently received little notice outside TsAGI."

3.6.2 TVD and MUSCL schemes

There is a property of scalar nonlinear conservation laws, which was probably observed for the first time by Peter Lax [Lax73]:

The total increasing and decreasing variations of a differentiable solution between any pair of characteristics are conserved.

In the presence of shock waves, information is lost and the total variation decreases. For compactly supported or periodic solutions, one can establish the following inequality

$$\int_{-\infty}^{+\infty} |dw(x, t_2)| \leq \int_{-\infty}^{+\infty} |dw(x, t_1)|, \quad t_2 \geq t_1. \quad (29)$$

⁵ MUSCL stands for Monotone Upstream-centered Scheme for Conservation Laws.

This motivated Harten [Har83] to introduce the notion of discrete total variation of numerical solution $u_h := \{u_j\}$

$$TV(w_h) := \sum_j |w_{j+1} - w_j|,$$

and the discrete counterpart to (29)

$$TV(w_h^{n+1}) \leq TV(w_h^n).$$

If this property is fulfilled, then a finite volume scheme is said to be total variation diminishing (TVD). The following theorem was proved in [Har83]:

Theorem 2 (i): *Monotone schemes are TVD*

(ii): *TVD schemes are monotonicity preserving, i.e. the number of solution extrema is preserved in time.*

Remark 8 *From the mathematical point of view it would be more correct to say “the total variation non-increasing (TVNI) scheme” but the “wrong” term TVD is generally accepted in the scientific literature.*

In one space dimension the construction of TVD schemes is not a problem anymore. Let us recall that in this study we are rather interested in two space dimensions (or even three in future work). In these cases the situation is considerably more complicated. Even if we consider the simplest case of structured cartesian meshes and apply a 1D TVD scheme on a dimension-by-dimension basis, a result of Goodman and Leveque shows [GV85] that TVD schemes in two or more space dimensions are only first order accurate. Motivated by this negative result, weaker conditions yielding solution monotonicity preservation should be developed.

In this article we will describe the construction and practical implementation of a second-order nonlinear scheme on unstructured (possibly highly distorted) meshes. The main idea is to find our solution as a piecewise affine function on each cell. This kind of linear reconstruction operators on simplicial control volumes often exploit the fact that the cell average is also a pointwise value of any valid (conservative) linear reconstruction evaluated at the gravity center of a simplex. This reduces the reconstruction problem to that of gradient estimation given cell averaged data. In this case, we express the reconstruction in the form

$$w_K(\vec{x}) = \bar{w}_K + (\nabla w)_K \cdot (\vec{x} - \vec{x}_0), \quad K \in \mathcal{T}, \quad (30)$$

where \bar{w}_K is the cell averaged value given by the finite volume method, $(\nabla w)_K$ is the solution gradient estimate (to be determined) on the cell K , $\vec{x} \in K$ and the point \vec{x}_0 is chosen to be the gravity center for the simplex K .

It is very important to note that with this type of representation (30) we remain absolutely conservative, i.e.

$$\frac{1}{\text{vol}(K)} \int_K w_K(\vec{x}) d\Omega \equiv \bar{w}_K$$

due to the choice of the point \vec{x}_0 . This point is crucial for finite volumes because of intrinsic conservative properties of this method.

In next sections we describe briefly two common techniques: Green-Gauss integration and least squares methods for solution gradient estimation on each cell. There are other available techniques. We can mention here an implicit gradient reconstruction method proposed in [MG96] and reused later in [AMS04], for example. We decided not to implement this approach in our research code since this procedure is computationally expensive⁶.

3.6.3 Green-Gauss gradient reconstruction

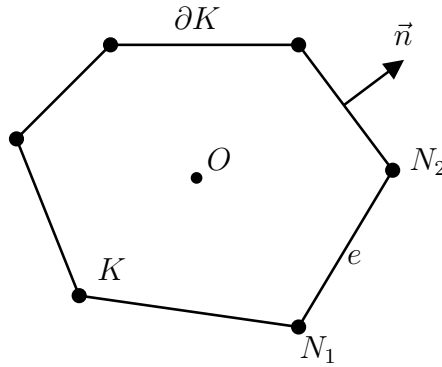


Fig. 11. Illustration for Green-Gauss gradient reconstruction. Control volume K with barycenter O and exterior normal \vec{n} .

This gradient reconstruction technique can be easily implemented on simplicial meshes. It is based on two very simple ideas: the mean value approximation and Green-Gauss-Ostrogradsky formula.

Consider a control volume K with barycenter O . The exterior normal to an edge $e \in \partial K$ is denoted by \vec{n}_e . This configuration is depicted on Fig. 11. In order to estimate the solution gradient on K (or in other words, to estimate its value at gravity center O) we make the following mean value approximation

$$(\nabla w)_K = (\nabla w)|_O \cong \frac{1}{\text{vol}(K)} \int_K \nabla w d\Omega,$$

⁶ In order to reconstruct the solution gradient we have to solve a linear system of equations. Recall that the gradient is estimated at each time step on each control volume. This factor slows down considerably explicit time discretizations.

and apply Green-Gauss-Ostrogradsky formula

$$(\nabla w)_K \cong \frac{1}{\text{vol}(K)} \int_{\partial K} w \otimes \vec{n} \, d\sigma = \frac{1}{\text{vol}(K)} \sum_{e \in \partial K} \int_e w \otimes \vec{n}_e \, d\sigma \cong \sum_{e \in \partial K} \frac{\text{area}(e)}{\text{vol}(K)} w|_{e/2} \otimes \vec{n}_e,$$

where $w|_{e/2}$ denote the solution value at the face (or edge in 2D) centroid. The face value needed to compute the reconstruction gradient can be obtained from a weighted average of the values at the vertices on the face [HC89]. In 2D it simply becomes

$$w|_{e/2} = \frac{w_{N_1} + w_{N_2}}{2}.$$

This approximation yields the following formula for gradient estimation:

$$(\nabla w)_K \cong \sum_{e \in \partial K} \frac{\text{area}(e)}{\text{vol}(K)} \frac{(w_{N_1} + w_{N_2})}{2} \otimes \vec{n}_e.$$

The gradient calculation is exact whenever the numerical solution varies linearly over the support of the reconstruction.

This procedure requires the knowledge of the solution values at the mesh nodes $\{N_i\}$. Recall that a cell centered finite volume scheme provides us with data located at cell gravity centers. Thus, an interpolation technique is needed. The quality of Green-Gauss gradient reconstruction greatly depends on the chosen interpolation method. The method chosen here is explained in Section 3.6.6.

3.6.4 Least-squares gradient reconstruction method

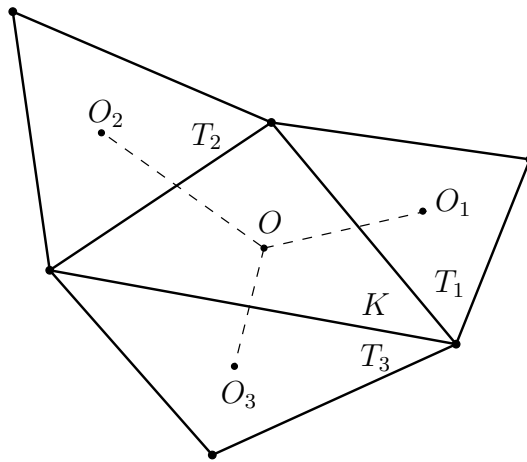


Fig. 12. Illustration for least-squares gradient reconstruction. A triangle control volume with three adjacent neighbors is depicted.

In this section we consider a triangle⁷ control volume K with three adjacent neighbors T_1 , T_2 and T_3 . Their barycenters are denoted by $O(\vec{x}_0)$, $O_1(\vec{x}_1)$, $O_2(\vec{x}_2)$ and $O_3(\vec{x}_3)$ respectively. In the following we denote by w_i the solution value at gravity centers O_i :

$$w_i := w(\vec{x}_i), \quad w_0 := w(\vec{x}_0).$$

Our purpose here is to estimate $\nabla w = (\partial_x w, \partial_y w)$ on the cell K . Using Taylor formula, we can write down the three following relations:

$$w_1 - w_0 = (\nabla w)_K \cdot (\vec{x}_1 - \vec{x}_0) + \mathcal{O}(h^2), \quad (31)$$

$$w_2 - w_0 = (\nabla w)_K \cdot (\vec{x}_2 - \vec{x}_0) + \mathcal{O}(h^2), \quad (32)$$

$$w_3 - w_0 = (\nabla w)_K \cdot (\vec{x}_3 - \vec{x}_0) + \mathcal{O}(h^2). \quad (33)$$

If we drop higher order terms $\mathcal{O}(h^2)$, these relations can be viewed as a linear system of three equations for two unknowns⁸ $(\partial_x w, \partial_y w)$. This situation is due to the fact that the number of edges incident to a simplex mesh in \mathbb{R}^d is greater or equal (in this case see Remark 9) to d thereby producing linear constraint equations (31) – (33) which will be solved analytically here in a least squares sense.

First of all, each constraint (31) – (33) is multiplied by a weight $\omega_i \in (0, 1)$ which will be chosen below to account for distorted meshes. In matrix form our non-square system becomes

$$\begin{pmatrix} \omega_1 \Delta x_1 & \omega_1 \Delta y_1 \\ \omega_2 \Delta x_2 & \omega_2 \Delta y_2 \\ \omega_3 \Delta x_3 & \omega_3 \Delta y_3 \end{pmatrix} (\nabla w)_K = \begin{pmatrix} \omega_1 (w_1 - w_0) \\ \omega_2 (w_2 - w_0) \\ \omega_3 (w_3 - w_0) \end{pmatrix},$$

where $\Delta x_i = x_i - x_0$, $\Delta y_i = y_i - y_0$. For further developments it is convenient to rewrite our constraints in abstract form

$$[\vec{L}_1, \vec{L}_2] \cdot (\nabla w)_K = \vec{f}. \quad (34)$$

We use a normal equation technique in order to solve symbolically this abstract form in a least squares sense. Multiplying on the left both sides of (34) by

⁷ Generalization to other simplicial control volumes is straightforward.

⁸ This simple estimation is done for the scalar case only $w = (w)$. For more general vector problems the numbers of equations and unknowns must be changed depending on the dimension of vector w .

$[\vec{L}_1 \vec{L}_2]^t$ yields

$$G(\nabla w)_K = \vec{b}, \quad G = (l_{ij})_{1 \leq i, j \leq 2} = \begin{pmatrix} (\vec{L}_1 \cdot \vec{L}_1) & (\vec{L}_1 \cdot \vec{L}_2) \\ (\vec{L}_2 \cdot \vec{L}_1) & (\vec{L}_2 \cdot \vec{L}_2) \end{pmatrix} \quad (35)$$

where G is the Gram matrix of vectors $\{\vec{L}_1, \vec{L}_2\}$ and $\vec{b} = \begin{pmatrix} (\vec{L}_1 \cdot \vec{f}) \\ (\vec{L}_2 \cdot \vec{f}) \end{pmatrix}$. The so-called normal equation (35) is easily solved by Cramer's rule to give the following result

$$(\nabla w)_K = \frac{1}{l_{11}l_{22} - l_{12}^2} \begin{pmatrix} l_{22}(\vec{L}_1 \cdot \vec{f}) - l_{12}(\vec{L}_2 \cdot \vec{f}) \\ l_{11}(\vec{L}_2 \cdot \vec{f}) - l_{12}(\vec{L}_1 \cdot \vec{f}) \end{pmatrix}.$$

The form of this solution suggests that the least squares linear reconstruction can be efficiently computed without the need for storing a non-square matrix.

Now we discuss the choice of weight coefficients $\{\omega_i\}_{i=1}^3$. The basic idea is to attribute bigger weights to cells barycenters closer to the node N under consideration. One of the possible choices consists in taking a harmonic mean of respective distances $r_i = \|\vec{x}_i - \vec{x}_N\|$. This purely metric argument takes the following mathematical form:

$$\omega_i = \frac{\|\vec{x}_i - \vec{x}_N\|^{-k}}{\sum_{j=1}^3 \|\vec{x}_j - \vec{x}_N\|^{-k}},$$

where k in practice is taken to be one or two (in our numerical code we choose $k = 1$).

Remark 9 *When a triangle shares an edge with the boundary $\partial\Omega$ (see Fig. 14 for illustration), the gradient reconstruction procedure becomes even simpler, since the number of constraints is equal to d and the linear system (31) – (33) becomes completely determined:*

$$\begin{aligned} w_1 - w_0 &= (\nabla w)_K \cdot (\vec{x}_1 - \vec{x}_0) + \mathcal{O}(h^2), \\ w_2 - w_0 &= (\nabla w)_K \cdot (\vec{x}_2 - \vec{x}_0) + \mathcal{O}(h^2), \end{aligned}$$

or in componentwise form it reads

$$\begin{pmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{pmatrix} (\nabla w)_K = \begin{pmatrix} w_1 - w_0 \\ w_2 - w_0 \end{pmatrix}.$$

The unique solution to this linear system is given again by Cramer's rule

$$(\nabla w)_K = \frac{\begin{pmatrix} (y_2 - y_0)(w_1 - w_0) - (y_1 - y_0)(w_2 - w_0) \\ (x_1 - x_0)(w_2 - w_0) - (x_2 - x_0)(w_1 - w_0) \end{pmatrix}}{(x_1 - x_0)(y_2 - y_0) - (x_2 - x_0)(y_1 - y_0)}.$$

3.6.5 Slope limiter

The idea of incorporating limiter functions to obtain non-oscillatory resolution of discontinuities and steep gradients dates back to Boris and Book [BB73]. When the limiter is identically equal to 1, we have the unlimited form of the linear interpolation. In the 1D case one can easily find in the literature about 15 different limiter functions such as CHARM, minmod, superbee, van Albada and many others. On unstructured meshes the situation is quite different. In the present study we decided to choose the Barth-Jespersen limiter proposed in [BJ89]. Here we do not discuss its construction and properties but just give the final formula. We need to introduce the following notation

$$w_K^{min} := \min_{L \in \mathcal{N}(K)} w_L, \quad w_K^{max} := \max_{L \in \mathcal{N}(K)} w_L.$$

The limited version of (30) is given by the following modified reconstruction operator

$$w_K(\vec{x}) = \bar{w}_K + \alpha_K (\nabla w)_K \cdot (\vec{x} - \vec{x}_0), \quad K \in \mathcal{T},$$

where it is assumed that $\alpha_K \in [0, 1]$. Obviously, the choice $\alpha_K = 0$ corresponds to the first order scheme while $\alpha_K = 1$ is the unlimited form. Barth and Jespersen [BJ89] propose the following choice of α_K :

$$\alpha_K^{BJ} := \min_{\forall f \in \partial K} \begin{cases} \frac{w_K^{max} - \bar{w}_K}{w_K(\vec{x}_f) - \bar{w}_K} & \text{if } w_K(\vec{x}_f) > w_K^{max}, \\ \frac{w_K^{min} - \bar{w}_K}{w_K(\vec{x}_f) - \bar{w}_K} & \text{if } w_K(\vec{x}_f) < w_K^{min}, \\ 1 & \text{otherwise.} \end{cases}$$

where \vec{x}_f denotes the face f centroid.

Although this limiter function does not fulfill all the requirements of finite volume maximum principle on unstructured meshes [BO04], it can be shown that it yields finite volume schemes possessing a global extremum diminishing property. Also this limiter produces the least amount of slope reduction which can be advantageous for accuracy. Note that in practical implementation minor modifications are required to prevent near zero division for almost constant solution data.

3.6.6 Solution interpolation to mesh nodes

We have seen above that several gradient reconstruction procedures (in particular gradient estimation on the faces) require the knowledge of the solution at mesh nodes (or vertices). This information is not directly given by the finite volume method since we chose the cell-centered approach.

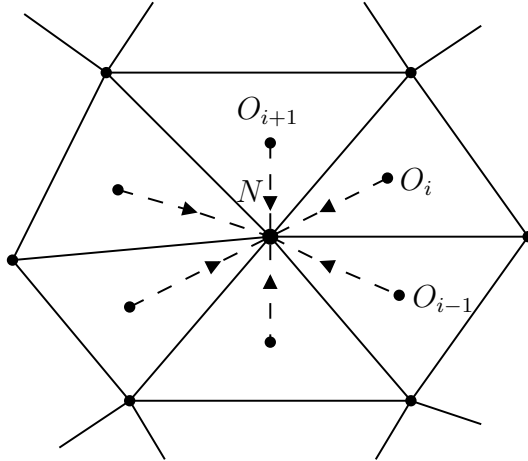


Fig. 13. Triangles with their barycenters O_i sharing the same vertex N .

Let us consider a node $N(x_n, y_n)$ of the tessellation \mathcal{T} and a control volume K_i with barycenter $O_i(x_i, y_i)$ having this node as a vertex (see Fig. 13 for illustration). The MUSCL procedure provides a solution gradient on each cell. Thus, using the Taylor formula or, equivalently, the representation (30) we can estimate the solution value at the node N

$$w_N = \bar{w}_{K_i} + (\nabla w)_{K_i} \cdot (\vec{x}_N - \vec{x}_i). \quad (36)$$

The problem is that we will have $d(N)$ different values of the solution in the same point depending on the control volume under consideration. Here $d(N)$ is the degree of vertex N in the sense of graph theory. One of the possible ways to overcome this contradiction is averaging. One interesting technique was proposed in [HC89] and further improved in [KMC03]. In our turn, we slightly modified this method. The algorithm implemented in our code is briefly described here.

First of all, let us look for the vertex value \bar{w}_N as a weighted sum of the values w_{N_i} computed by formula (36) from each surrounding cell

$$\bar{w}_N = \frac{\sum_{i=1}^{d(N)} \omega_i w_{N_i}}{\sum_{i=1}^{d(N)} \omega_i}.$$

The weighting factors $\{\omega_i\}_{i=1}^{d(N)}$ are made to satisfy the condition of zero

pseudo-Laplacian

$$L(x_n) \equiv \sum_{i=1}^{d(N)} \omega_i(x_i - x_n), \quad L(y_n) \equiv \sum_{i=1}^{d(N)} \omega_i(y_i - y_n). \quad (37)$$

These conditions have a very simple interpretation. They are imposed so that the method be exact for affine data over the stencil.

As in the original formulation by Holmes and Connell [HC89], the weighting factor ω_i is written as

$$\omega_i = 1 + \Delta\omega_i.$$

The weights $\{\omega_i\}$ are determined by solving an optimization problem in which the cost-function to be minimized is defined as

$$\frac{1}{2} \sum_{i=1}^{d(N)} (r_i \Delta\omega_i)^2 \rightarrow \min \quad (38)$$

with two constraints given by (37). It should be noted that the cost function is slightly different from the original formulation. The difference lies in the factor of

$$r_i^2 \equiv \|\vec{ON} - \vec{OO}_i\|^2$$

which was introduced in [KMC03]. This modification effectively allows larger values of weight $\Delta\omega_i$ for those cells closer to the node in question.

Employing the method of Lagrange multipliers, the original optimization problem, which was to minimize the cost function given by (38) with the constraints (37) is equivalent to minimizing the function \mathcal{L} defined by

$$\mathcal{L} = \frac{1}{2} \sum_{i=1}^{d(N)} (r_i \Delta\omega_i)^2 - \lambda \sum_{i=1}^{d(N)} \omega_i(x_i - x_n) - \mu \sum_{i=1}^{d(N)} \omega_i(y_i - y_n) \rightarrow \min$$

which leads to

$$\Delta\omega_i = \frac{\lambda(x_i - x_n) + \mu(y_i - y_n)}{r_i^2}.$$

The two Lagrangian multipliers, λ and μ , are obtained from

$$\lambda = \frac{r_y I_{xy} - r_x I_{yy}}{I_{xx} I_{yy} - I_{xy}^2}, \quad \mu = \frac{r_x I_{xy} - r_y I_{xx}}{I_{xx} I_{yy} - I_{xy}^2},$$

where

$$r_x = \sum_{i=1}^{d(N)} (x_i - x_n), \quad r_y = \sum_{i=1}^{d(N)} (y_i - y_n). \\ I_{xx} = \sum_{i=1}^{d(N)} \frac{(x_i - x_n)^2}{r_i^2}, \quad I_{yy} = \sum_{i=1}^{d(N)} \frac{(y_i - y_n)^2}{r_i^2}, \quad I_{xy} = \sum_{i=1}^{d(N)} \frac{(x_i - x_n)(y_i - y_n)}{r_i^2}.$$

The last step consists in renormalizing the weights $\{\omega_i\}_{i=1}^{d(N)}$ to the range $[0, 1]$.

Remark 10 *The above algorithm is not computationally expensive since the weights $\{\omega_i\}_{i=1}^{d(N)}$ depend only on the tessellation \mathcal{T} geometry. It means that they can be computed and stored before the main loop in time and reused during the computations later.*

Remark 11 *Even if we suggest to use the above method (since it gives slightly better results), we would like to give another idea (based on a purely metrics argument) of how to construct the weights $\{\omega_i\}_{i=1}^{d(N)}$. This approach is considerably simpler than solving an optimization problem and it was already used in the least squares gradient reconstruction method (see Section 3.6.4). In fact, in order to calculate ω_i one can simply take the harmonic mean of distances between the node N under the question and respective cell barycenter O_i (see Fig. 13):*

$$\omega_i := \frac{\|\vec{x}_N - \vec{x}_i\|^{-k}}{\sum_{j=1}^{d(N)} \|\vec{x}_N - \vec{x}_j\|^{-k}},$$

where k in practice is equal to 1 or 2.

This choice does not guarantee exact interpolation of globally linear functions.

3.7 Boundary conditions implementation

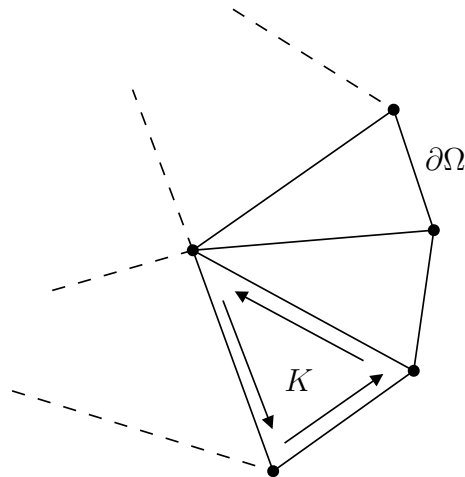


Fig. 14. Control volume sharing a face with boundary $\partial\Omega$.

So far we have not discussed the implementation of boundary conditions. The flavor of boundary conditions treatment for hyperbolic systems is given here and we refer to [GP05] for a general discussion. This is a very important topic since they actually determine the solution. Let us consider the space discretization of the system (7) by a cell centered finite volume method. For instance for the time explicit discretization we have the scheme (21). Of course this

formula is not valid when K meets the boundary of Ω (see Fig.14 for illustration). When this occurs, we have to find the numerical flux $\Phi(w_K^n, K, \partial\Omega)$. In practice, this flux is not given by the physical boundary conditions and moreover, in general, (7) is an ill-posed problem if we try to impose either v or $F(v) \cdot \vec{n}$ on $\partial\Omega$. This can be understood in a simple way by using the following linearization of this system:

$$\frac{\partial w}{\partial t} + \underline{A}_n \frac{\partial w}{\partial n} = 0, \quad (39)$$

where \vec{n} represents the direction of the external normal on $K \cap \partial\Omega$, \underline{A}_n is the advection matrix:

$$\underline{A}_n \equiv \left. \frac{(\partial F(w) \cdot \vec{n})}{\partial w} \right|_{w=\underline{w}}, \quad (40)$$

and \underline{w} is the state around which the linearization is performed. When (7) is hyperbolic, the matrix \underline{A}_n is diagonalizable on \mathbb{R} and by a change of coordinates, this system becomes an uncoupled set of m advection equations:

$$\frac{\partial \xi_k}{\partial t} + \lambda_k \frac{\partial \xi_k}{\partial n} = 0, \quad k = 1, \dots, m. \quad (41)$$

Here the λ_k are the eigenvalues of \underline{A}_n and according to their sign, waves are going either into the domain Ω ($\lambda_k < 0$) or out of the domain Ω ($\lambda_k > 0$). Hence we expect that it is only possible to impose p conditions on $K \cap \partial\Omega$ where $p \equiv \#\{k \in \{1, \dots, m\} \text{ such that } \lambda_k < 0\}$.

Let us consider now a control volume K which meets the boundary $\partial\Omega$. We take $\underline{w} = w_K^n$ and write the previous linearization. We denote by x the coordinate along the outer normal so that (39) reads:

$$\frac{\partial w}{\partial t} + \underline{A}_n \frac{\partial w}{\partial x} = 0, \quad (42)$$

which happens to be the linearization of the 1D (*i.e.* when $nd = 1$) system. First we label the eigenvalues $\lambda_k(\underline{w})$ of \underline{A}_n by increasing order:

$$\lambda_1(\underline{w}) \leq \lambda_2(\underline{w}) \leq \dots \leq \lambda_p(\underline{w}) < 0 \leq \lambda_{p+1}(\underline{w}) \dots \leq \lambda_m(\underline{w}). \quad (43)$$

- (i) The case $p = 0$. In this case information comes from inside Ω and therefore we take:

$$\Phi(w_K^n, K, \partial\Omega) = F(w_K^n) \cdot \vec{n}_K. \quad (44)$$

In the Computational Fluid Dynamics literature this is known as the ‘‘supersonic outflow’’ case.

- (ii) The case $p = m$. In this case information comes from outside Ω and therefore we take:

$$\Phi(w_K^n, K, \partial\Omega) = \Phi_{given}, \quad (45)$$

where Φ_{given} are the given physical boundary conditions. In the Computational Fluid Dynamics literature this is known as the “supersonic inflow” case.

- (iii) The case $1 \leq p \leq m - 1$. As already discussed, we need p scalar information coming from outside of Ω . Hence we assume that we have on physical ground p relations on the boundary:

$$g_l(w) = 0, \quad l = 1, \dots, p. \quad (46)$$

Remark 12 *The notation $g_l(w) = 0$ means that we have a relation between the components of w . However, in general, the function g_l is not given explicitly in terms of w . For example $g_l(w)$ could be the pressure which is not, in general, one of the components of w .*

Since we have to determine the m components of $\Phi(w_K^n, K, \partial\Omega)$, we need $m - p$ supplementary scalar conditions. Let us write them as

$$h_l(w) = 0, \quad l = p + 1, \dots, m. \quad (47)$$

In general (46) are referred to as “physical boundary conditions” while (47) are referred to as “numerical boundary conditions”.

Then we take:

$$\Phi(w_K^n, K, \partial\Omega) = F(w) \cdot \vec{n}_K, \quad (48)$$

where w is solution to (46)-(47) (see however Remark 15 and (54)).

Remark 13 *The system (46)-(47) for the m unknowns $w \in G$ is a $m \times m$ nonlinear system of equations. We are going to study its solvability, see Theorem 3.*

Let us first discuss the numerical boundary conditions (47). By analogy with what we did on an interface between two control volumes K and L , we take (recall that $\underline{w} = w_K^n$):

$$\tilde{l}_k(\underline{w}) \cdot (F(w) \cdot \vec{n}_K) = \tilde{l}_k(\underline{w}) \cdot (F(w_K^n) \cdot \vec{n}_K), \quad k = p + 1, \dots, m. \quad (49)$$

In other words, we set $h_k(w) \equiv \tilde{l}_k(w_K^n) \cdot (F(w) \cdot \vec{n}_K) - \tilde{l}_k(w_K^n) \cdot (F(w_K^n) \cdot \vec{n}_K)$. We have denoted by $(\tilde{l}_1(\underline{w}), \dots, \tilde{l}_m(\underline{w}))$ a set of left eigenvectors of \underline{A}_n : ${}^t\tilde{A}_n \tilde{l}_k(\underline{w}) = \lambda_k \tilde{l}_k(\underline{w})$ and by $(r_1(\underline{w}), \dots, r_m(\underline{w}))$ a set of right eigenvectors of \underline{A}_n : $\underline{A}_n r_k(\underline{w}) = \lambda_k r_k(\underline{w})$. Moreover the following normalization is taken: $\tilde{l}_k(\underline{w}) \cdot \tilde{r}_p(\underline{w}) = \delta_{k,p}$.

According to [GP05] we have the following result on the solvability of (46)-(47).

Theorem 3 *In the case $1 \leq p \leq m - 1$, assume that $\lambda_{p+1}(\underline{w}) > 0$, and*

$$\det_{1 \leq k, l \leq p} \left(\sum_{i=1}^m r_k^i(\underline{w}) \frac{\partial g_l}{\partial w_i}(\underline{w}) \right) \neq 0. \quad (50)$$

With the choice (49) the nonlinear system (46)-(47) has one and only one solution v , for $v - \underline{w}$ and $g_l(\underline{w})$ sufficiently small.

Remark 14 In this result we exclude the case where the boundary is characteristic i.e. the case where one of the λ_k is equal to 0. This case cannot be dealt with at this level of generality. On the other hand, wall boundary conditions belong to this category. They can be discussed and handled directly on the physical system under consideration. In this section we will show how to do it for the NSWE equations (see Paragraph 3.7.1). Moreover, the treatment of wall boundaries of compressible Euler equations and some two-phase systems [DDG08b,DDG10,DDG08a,Dut07] can be done in a similar way.

Remark 15 In practice, (46)-(47) are written in a parametric way. We have a set of m physical variables w (e.g. pressure, densities, velocities, ...) and we look for w satisfying:

$$g_l(w) = 0, \quad l = 1, \dots, p, \quad (51)$$

$$\tilde{l}_k(\underline{w}) \cdot \Phi = \tilde{l}_k(\underline{w}) \cdot (F(w_K^n) \cdot \vec{n}_K), \quad (52)$$

$$\Phi = F(w) \cdot \vec{n}_K, \quad (53)$$

and then we take:

$$\Phi(w_K^n, K, \partial\Omega) = \Phi. \quad (54)$$

The system (51)-(52)-(53) is then solved by Newton's method.

3.7.1 Impermeable boundary

Consider the case of a rigid wall boundary

$$\vec{u}(\vec{x}, t) \cdot \vec{n} = 0, \quad \vec{x} \in \partial\Omega, \quad (55)$$

and the hyperbolic system (1), (2). The flux Φ that we have to determine on the boundary $\partial\Omega$ has the following form if we take into account (55):

$$\Phi = (\mathcal{F} \cdot \vec{n})|_{\partial\Omega} = \begin{pmatrix} 0 \\ \frac{g}{2} H^2 n_x \\ \frac{g}{2} H^2 n_y \end{pmatrix}. \quad (56)$$

Thus, we have to determine $\frac{g}{2} H^2$ on the boundary $\partial\Omega$. For this purpose we employ a complementary numerical boundary condition as it was explained above:

$$l_3(w_K) \cdot \Phi = l_3(w_K) \cdot \mathcal{F}_n(w_K), \quad (57)$$

where l_3 is the left eigenvector corresponding to the positive eigenvalue $\lambda_3 = u_n + c = c > 0$. Solving equation (57) leads to the following value of the

unknown component:

$$\frac{g}{2}H^2\Big|_{\partial\Omega} = \left(cHu_n + \frac{g}{2}H^2\right)\Big|_K,$$

which determines completely the boundary flux (56).

3.7.2 Generating boundary

Now let us consider a boundary where the total water depth is prescribed:

$$H|_{\partial\Omega} = H_0(\vec{x}_s, t) > 0, \quad \vec{x}_s \in \partial\Omega.$$

Taking into account this information, the flux Φ to be determined has the following form:

$$\Phi = \left(\mathcal{F} \cdot \vec{n}\right)\Big|_{\partial\Omega} = \begin{pmatrix} H_0u_n \\ H_0uu_n + \frac{g}{2}H_0^2n_x \\ H_0vu_n + \frac{g}{2}H_0^2n_y \end{pmatrix}. \quad (58)$$

Hence, we have to find u and v on the generating boundary $\partial\Omega$. The normal velocity will be immediately deduced from this information $u_n := un_x + vn_y$.

Throughout this section we will assume that the flow is “subsonic”, i.e. $|\vec{u} \cdot \vec{n}| \leq c$. We could also consider the “supersonic” case, but physically this situation is rather exotic. Henceforth, we have one negative eigenvalue $\lambda_1 = u_n - c$, one positive $\lambda_3 = u_n + c$ and $\lambda_2 = u_n$ can be in principle of any sign. Thus we have to consider two cases: $u_n < 0$ and $u_n \geq 0$. In the first case we need a supplementary physical condition (on the tangential velocity to the boundary), in the second one we use a supplementary numerical condition:

$$l_2(w_K) \cdot \Phi = l_2(w_K) \cdot \mathcal{F}_n(w_K).$$

Both lead to the same conclusion: $u_\tau|_{\partial\Omega} = u_\tau|_K$, where $u_\tau := un_y - vn_x$ is the tangential velocity. Computations very similar to the previous section 3.7.1 lead to the following solution:

$$u_n|_{\partial\Omega} = \frac{Hcu_n + \frac{g}{2}(H^2 - H_0^2)}{H_0c}, \quad u|_{\partial\Omega} = u_n|_{\partial\Omega} n_x + u_\tau n_y, \quad v|_{\partial\Omega} = u_n|_{\partial\Omega} n_y - u_\tau n_x.$$

Substituting these expressions into (58) gives the boundary flux Φ .

4 Numerical results

The numerical tests presented here are of two kinds. The first one is a comparison with analytical solutions (or approximate analytical solutions) : sections 4.1, 4.2, 4.4. This allows us to test the correctness and precision of the numerical scheme. The second one is a comparison with results from laboratory experiments : sections 4.3. This allows us to test the capacity of the code to reproduce actual events, and in particular assess the validity of the nonlinear shallow-water equations for tsunami modeling.

4.1 Convergence test

We begin the presentation of numerical tests by the simplest one – convergence test. We would like to show the accuracy of the MUSCL scheme implementation. To do it, we solve numerically the following scalar linear advection equation

$$\frac{\partial w}{\partial t} + \vec{u}_0 \cdot \nabla w = 0, \quad \vec{u}_0 \in \mathbb{R}^2$$

with smooth⁹ initial conditions. Moreover, it has almost compact support in order to reduce the influence of boundary conditions. It is obvious that this equation will just translate the initial form in the direction \vec{u}_0 . So, we have an analytical solution which can be used to quantify the numerical method error. On the other hand, to measure the convergence rate, we constructed a sequence of refined meshes.

Fig. 15 shows the error of the numerical method in L_∞ norm as a function of the mesh characteristic size. The slope of these curves represents an approximation to the theoretical convergence rate. On this plot, the blue curve corresponds to the first order upwind scheme while the other two (red and black) correspond to the MUSCL scheme with least-squares (see Section 3.6.4) and Green-Gauss (see Section 3.6.3) gradient reconstruction procedures respectively. One can see that the blue curve slope is equal approximately to 0.97 which means first order convergence. The other two curves have almost the same slope equal to 1.90 indicating a second order convergence rate for the MUSCL scheme. We remark that in our implementation of the second-order scheme the least-squares reconstruction seems to give slightly more accurate results than the Green-Gauss procedure.

The next figure represents the measured CPU time in seconds again as a function of the mesh size. Obviously, this kind of data is extremely computer

⁹ We intentionally choose a smooth initial condition since the discontinuities can decrease the overall accuracy of the scheme.

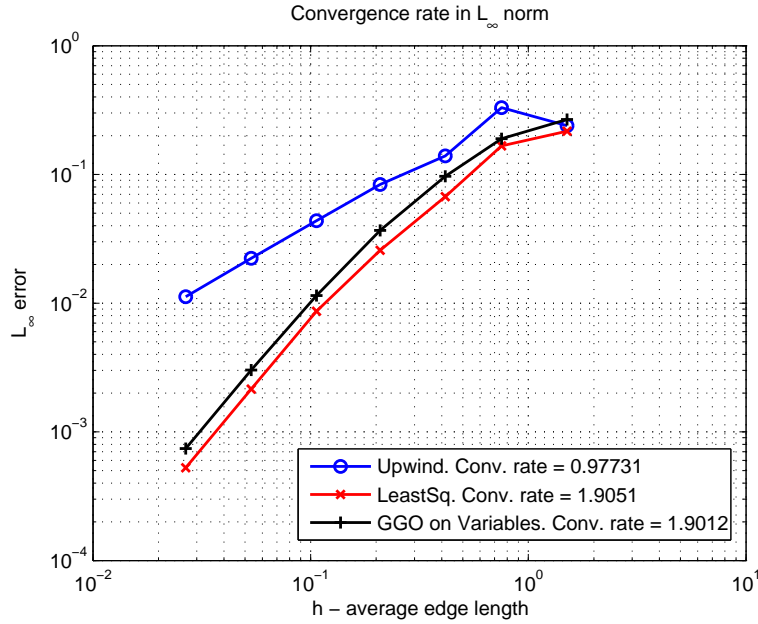


Fig. 15. Numerical method error in L_∞ norm.

dependent but the qualitative behaviour is the same on all systems. On Fig. 16 one can see that the “fastest” curve is the blue one (first order upwind scheme). Then we have two almost superimposed (black and red) curves referring to the second-order gradient reconstruction on variables. Here again one can notice that the least-squares method is slightly faster than the Green-Gauss procedure. On this figure we represented one more curve (the highest one) which corresponds to Green-Gauss gradient reconstruction on fluxes (it seems to be very natural in the context of the FVCF scheme explained in Section 3.1.1). Our numerical tests show that this method is quite expensive from the computational point of view and we decided not to choose it.

The next three sections deal with the validation of VOLNA again a set of benchmarks for tsunami modelling proposed at the Catalina 2004 workshop on long waves ; they are among the 6 benchmarks currently recommended by the United States National Oceanic and Atmospheric Administration (NOAA) for the evaluation of operational tsunami forecasting models [SBT⁺07]. In order to help the reproductibility and comparison of numerical results, all the following test cases make use of publicly available data¹⁰. Although, for each benchmark, we present results for one mesh only, we have checked that the simulations converge as the mesh resolution increases.

¹⁰ <http://www.cee.cornell.edu/longwave/index.cfm?page=benchmark>

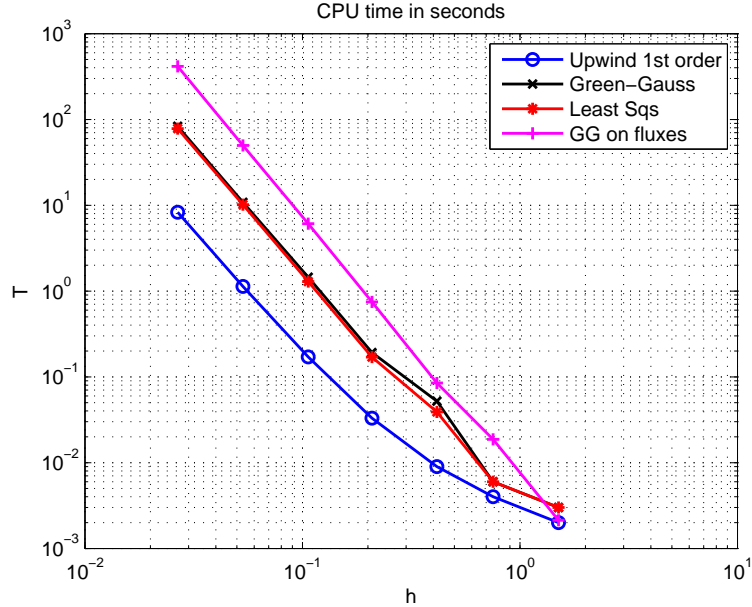


Fig. 16. CPU time for different finite volume schemes.

4.2 Tsunami run-up onto a plane beach

In this test case, we look at the runup of a tsunami wave over a plane sloping beach (of slope $\frac{1}{10}$). The initial depression wave propagates leftwards (see figure 26). The result of the simulation is compared to an analytical solution, obtained using the initial value problem technique of Carrier, Wu and Yeh [CWY03]. Note that the computational domain is approximately 50 km long, whereas the shoreline motion scale is 1 km ; hence, we choose to refine the mesh by a factor 10 near the initial shoreline. The results presented here correspond to a resolution of 8 meters in the direction of propagation. Moreover, the initial free surface amplitude is a few meters high. Thus, due to the difference of spatial scales between the bathymetry, the initial free surface and the computational domain dimensions, the source term $gH\nabla h$ is very steep (see figure 26), which renders the use of a well-balanced scheme mandatory. This test case is one-dimensional. Since our code is two-dimensional, we implement it using a two dimensional computational domain, with translation invariance in the transverse direction.

We can see on Figures 17 and 18 that the numerical results match pretty well the approximate analytical solution, especially near the shoreline location. This ensures the accuracy of the runup algorithm presented in section 3.3.

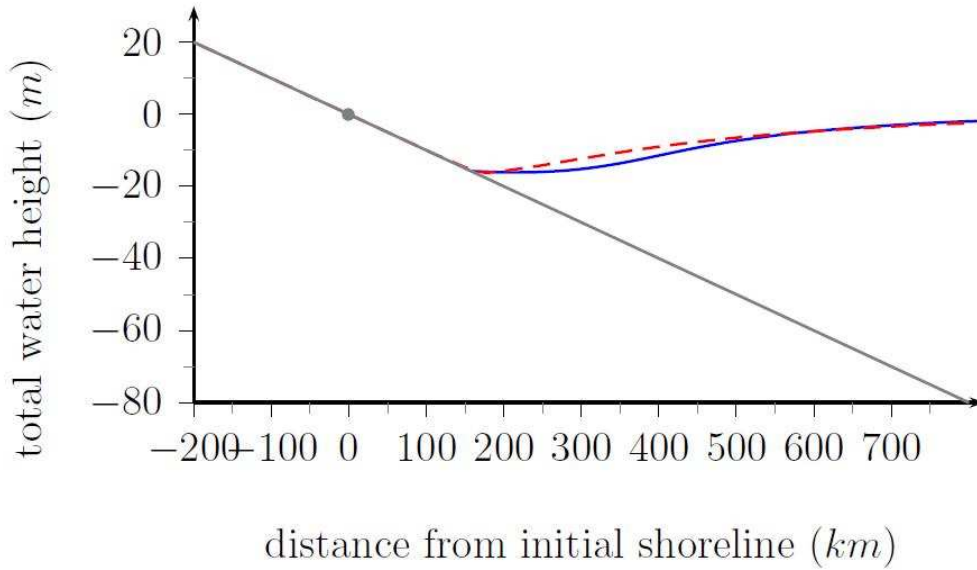


Fig. 17. Catalina 1 benchmark — comparison between analytical (solid) and numerical (dashed) values for free surface at time 160 seconds. The gray line represents the beach, and the gray point the initial shoreline location.

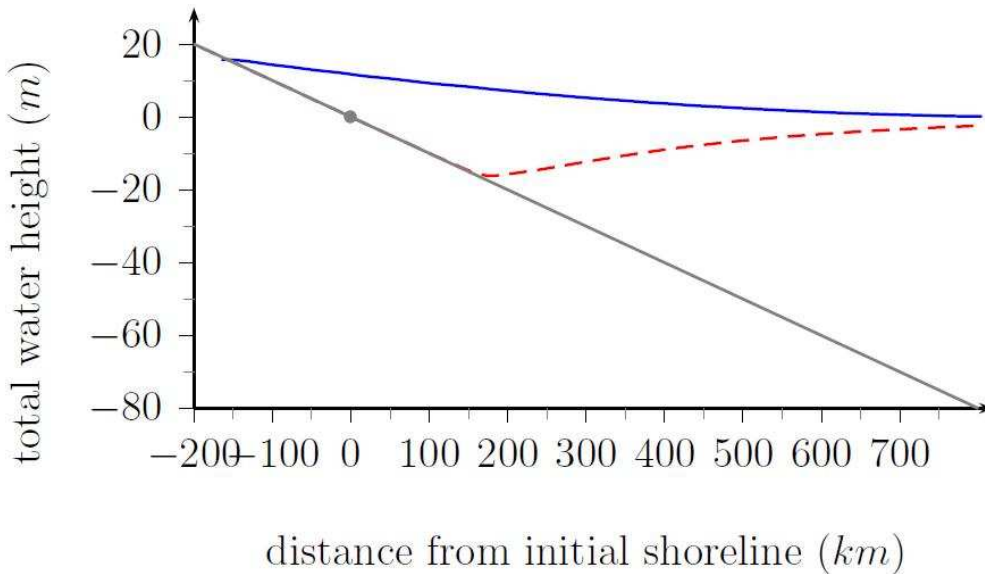


Fig. 18. Catalina 1 benchmark — comparison between analytical (solid) and numerical (dashed) values for free surface at time 220 seconds. The gray line represents the beach, and the gray point the initial shoreline location.

4.3 Tsunami run-up onto a complex 3-dimensional beach

This experiment reproduces at $\frac{1}{400}$ scale the Monai valley tsunami, which struck the Island of Okushiri (Hokkaido, Japan) in 1993, in a 205 meters

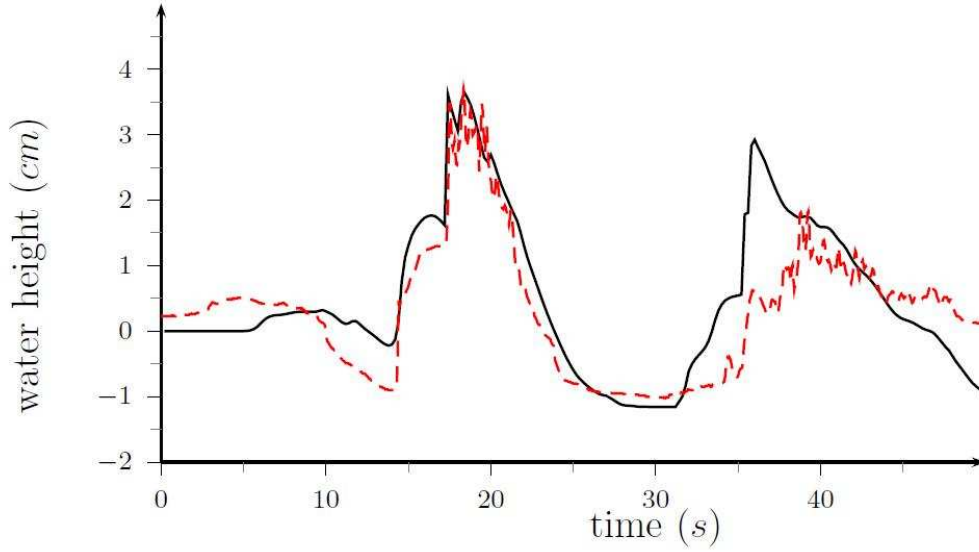


Fig. 19. Catalina 2 benchmark — comparison between numerical results and experimental data at gage 5.

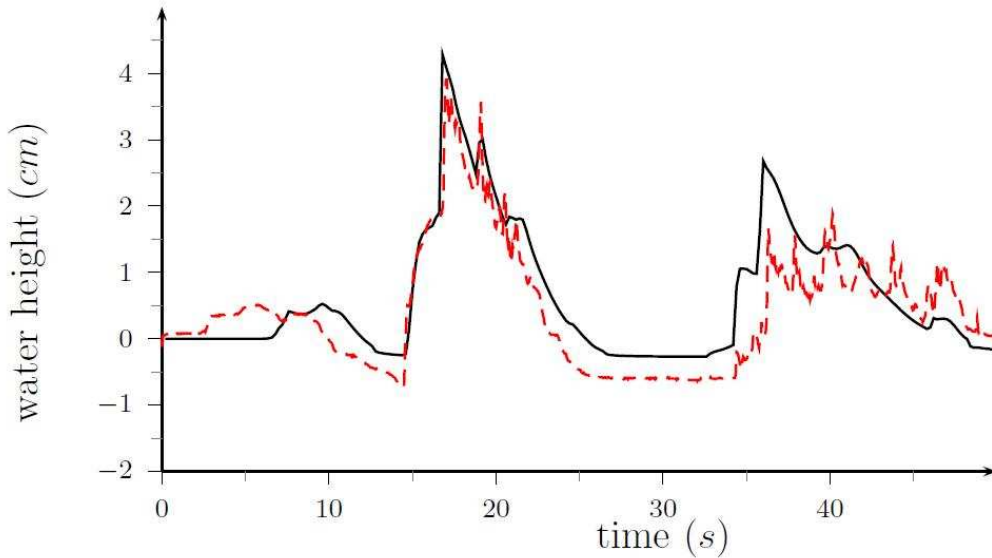


Fig. 20. Catalina 2 benchmark — comparison between numerical results and experimental data at gage 7.

long wave tank. The computational domain reproduces the last 5 meters of the wave tank. The initial incident wave offshore is given by experimental data, and fed as a time dependent boundary condition.

We compare the numerical results with the recorded data at three of the wave gages installed in the wave tank : gages number 5, 7 and 9, of respective coordinates $(4.521, 1.196)$, $(4.521, 1.696)$ and $(4.521, 2.196)$. This can be seen in Figures 19 – 21. The main wave (between times 15 and 25 seconds) is very

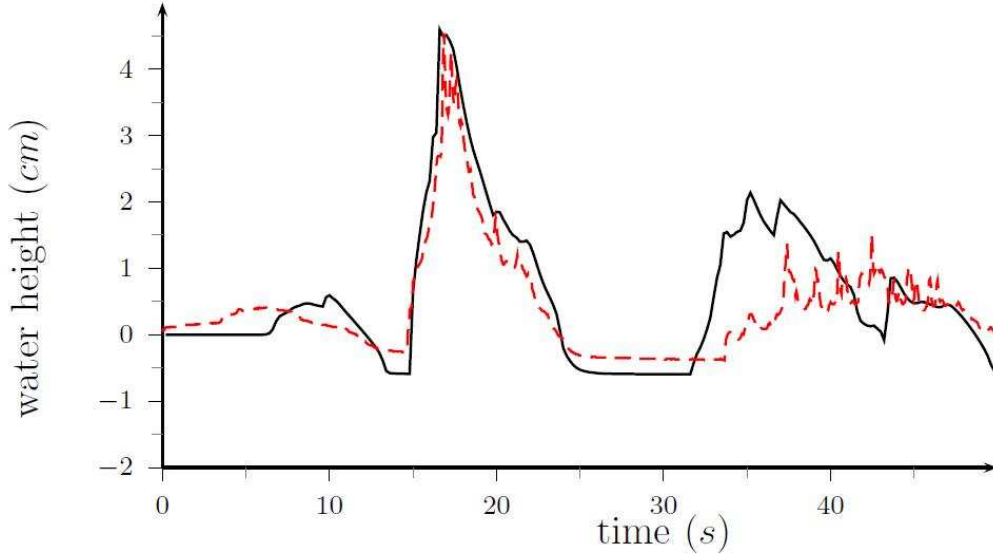


Fig. 21. Catalina 2 benchmark — comparison between numerical results and experimental data at gage 9.

accurately described, at all gages considered. Moreover, the maximal runup is adequately captured by the code. This value is extremely high, and occurred in Monai Valley (the corresponding canyon in the experimental setup, along with the free surface at the moment of maximum elevation are shown in figure 28). We obtain a maximal runup value of 8.05 centimeters, which corresponds to 32.2 meters put back at field scale. This is remarkably close to the measured value of 31.7 meters.

Hence the numerical model is able to reproduce the laboratory scenario accurately, even without bottom friction modelling (and thus without any free parameter). In this realistic test case, the ability to refine the mesh near the zones of interest is a very nice asset.

We performed another computation using the set-up described above. Namely, we solved equations (1), (2) completed by a conservation law for the wave energy. This system was recently proposed by Dutykh & Dias and we refer to [DD09a] for technical details and discussions. The total energy evolution is depicted on Figure 22. We represented two curves. The blue solid line corresponds to the solution of the augmented system of equations. The black broken line refers to the total energy, estimated from conservative variables:

$$E \approx \frac{1}{2}\rho H|\bar{u}|^2 + \frac{1}{2}\rho g\eta^2,$$

where ρ is the constant fluid density and $\eta = H - h$ is the free surface elevation with respect to the undisturbed water level. In complete accordance with results reported in [DD09a], the computed wave energy is not prone to numerical diffusion and has excellent monotonicity properties. Just at the end

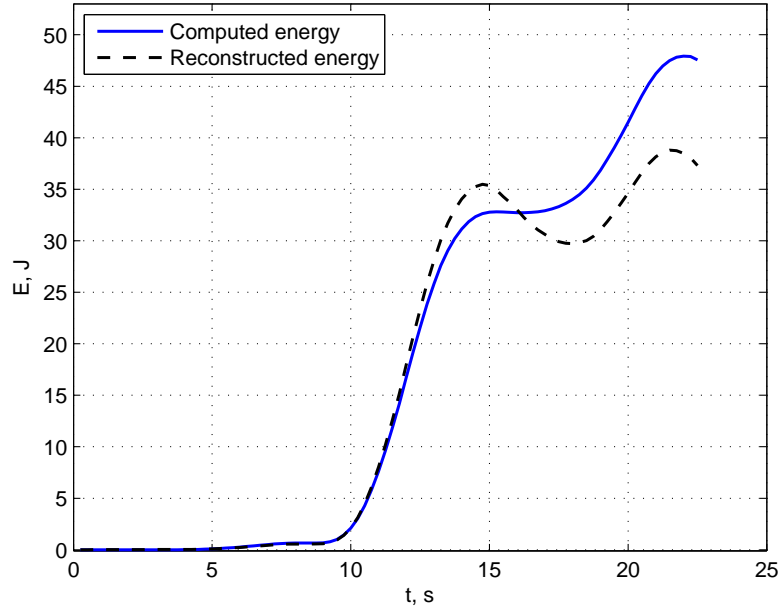


Fig. 22. Computed and reconstructed wave energy profiles for Catalina 2 benchmark problem.

of the simulation one can notice a little decrease in both curves. In fact, it is induced by energy losses due to the wave run-up on the beach.

4.4 Tsunami generation and runup due to a 2-dimensional landslide

In this test case, a translating Gaussian shaped mass, initially at the shoreline, translates rightwards and creates a wave (see figure 29). The seafloor can be described by the following equation :

$$h(x, t) = H(x) - h_0(x, t),$$

$$H(x) = x \tan(\beta), \quad \text{and} \quad h_0(x, t) = \delta \exp \left[- \left(2 \sqrt{\frac{x \mu^2}{\delta \tan(\beta)}} - \sqrt{\frac{g}{\delta}} \mu t \right)^2 \right],$$

x being the direction of propagation. Here, δ represent the maximum thickness of the sliding mass, μ the ratio between δ and the horizontal length of the mass, and β the beach slope. Notice that, at initial time, the submarine mass is partially underwater. Hence, this test case corresponds to a subaerial landslide. A sketch of this experiment can be seen in figure 29. This benchmark is one-dimensional, but is implemented using a two-dimensional computational

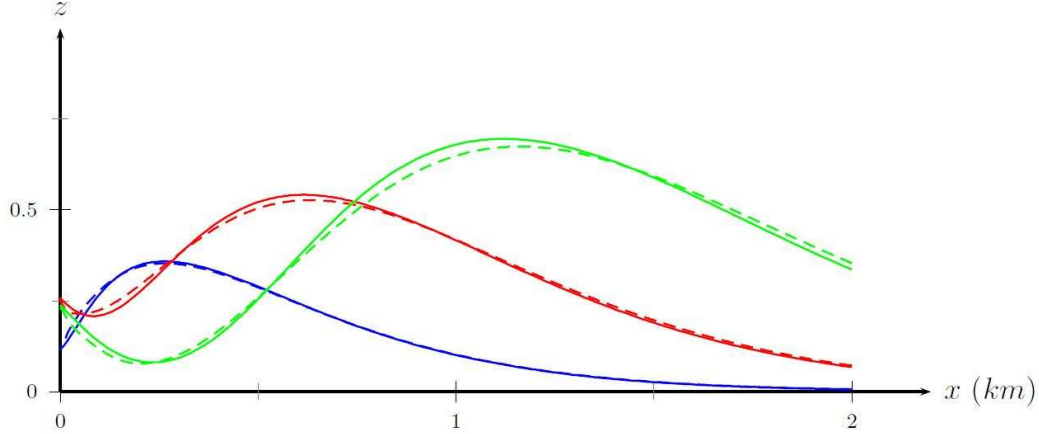


Fig. 23. Catalina 3 benchmark — free surface profiles at three different times — comparison between numerical results (dashed) and analytical formulas (solid lines), at times 16 (blue), 32 (red) and 48 (green) seconds.

domain, as in section 4.2. The numerical results we present are obtained as a one dimensional slice (which does not depend on the transverse variable).

The result of the numerical simulation is compared to an analytical solution, computed as an approximate solution of the linear shallow water equations with a forcing term [LLS03]. In figure 23, we can see the comparison between the analytical and numerical wave surface profiles at times 16, 32 and 48 seconds, for $\beta = 5.7^\circ$, $\delta = 1$ m, and $\mu = 0.01$ (fig. 23). These values ensure that we are in the linear regime of the shallow water equations (and thus that the analytical solution is a good approximation of the nonlinear equations solution). Hence, the comparison is meaningful. Good agreement is reached.

We also performed wave energy computation for this test-case. To our knowledge, the energy evolution has not been shown yet for a landslide generated wave. Computation results are presented on Figures 24 and 25 for two cases $\mu = 0.01$ and $\mu = 0.1$. In the latter case the linear shallow water equations (LSWE) are not valid even on small time scales. For more information, please refer to [LLS03].

Figure 24 shows the wave energy evolution with time. On Figure 25 we represented two trajectories in the energy phase-space (E_k, E_p) , where E_k is the kinetic energy and E_p is the potential one. We would like to point out the approximate energy repartition on the black curve ($\mu = 0.1$).

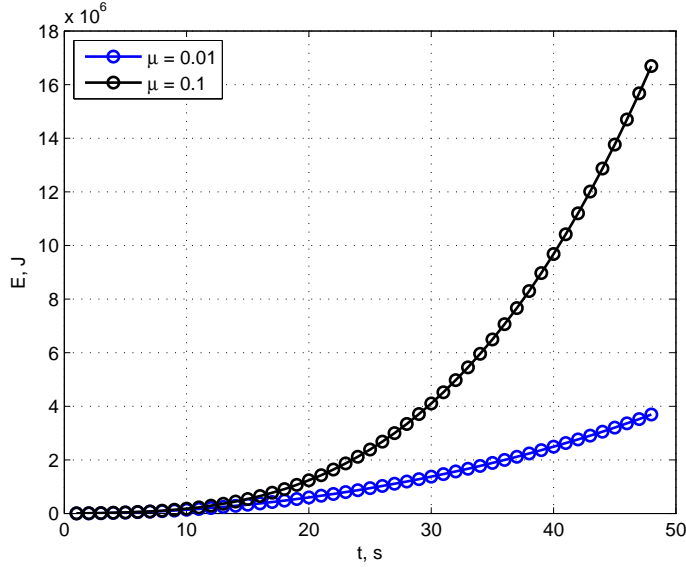


Fig. 24. Energy evolution with time for the 2-dimensional landslide test-case.

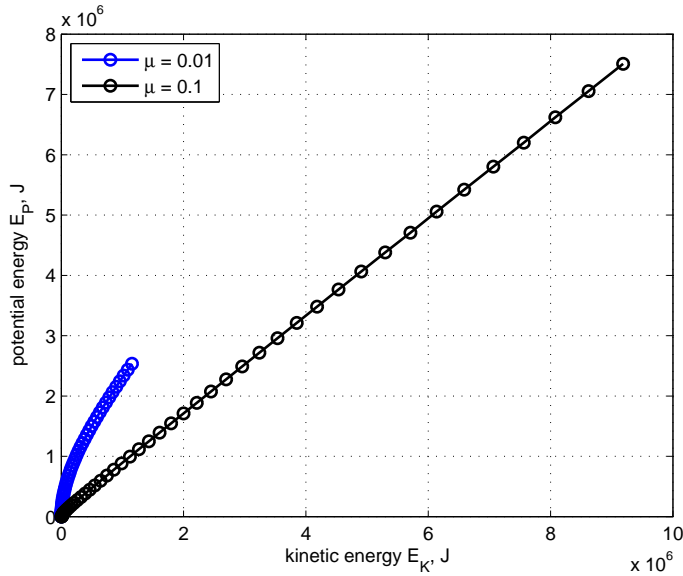


Fig. 25. Trajectories in the energy phase-space (E_k, E_p) for the 2-dimensional landslide test-case.

4.5 Summary

Using different analytical benchmarks, we have validated all components of our code : accuracy and order of convergence of the numerical scheme (section 4.1), run up algorithm and treatment of steep depth fields (section 4.2), and time varying bathymetry in a conservative shallow-water framework (section 4.4). Moreover, we have shown the capability of the code to model realistic events, using an experimental benchmark (section 4.3).

5 Conclusions and perspectives

In the present article we provided a detailed description of the *VOLNA* code, designed for complete tsunami wave modelling. Namely, we are able to simulate the whole life-cycle of a tsunami from generation to inundation. Special attention was paid to the run-up algorithm described in Section 3.3. The overall performance test and validation were done in Section 4.

The *VOLNA* code is operational and is able to run in complex and rapidly varying conditions. The use of unstructured meshes allows for taking into account the geometry of real coasts. Owing to the implementation of various types of boundary conditions, the code *VOLNA* can be coupled to other solvers and treat exclusively the zones where the NSWE are physically relevant.

Some new results were presented concerning the energy of tsunami waves [DD09a]. In particular, we show the wave energy evolution for the Catalina 2 test case (run-up on a complex 3D beach) and a landslide generated tsunami (Catalina 3 benchmark problem).

In the future we would like to add more physics to the *VOLNA* code: dissipative effects [DD07b,Dut09] (one could for example implement the dissipative terms from Bresch and Desjardins [BD03]) and dispersive effects.

Acknowledgements

This work has been partially supported by ANR HEXECO, Project n° BLAN07–1_192661 and by the 2008 Framework Program for Research, Technological development and Innovation of the Cyprus Research Promotion Foundation under the Project ΑΣΤΙ/0308(BE)/05. The third author acknowledges the support from the EU project TRANSFER (Tsunami Risk ANd Strategies For the European Region) of the sixth Framework Programme under contract no. 037058.

The first author acknowledges support from ANR MathOcéan (Project n° ANR-08-BLAN-0301-01) and from the program “Risques gravitaires, séismes” of Cluster Environnement and the research network VOR.

The authors thank Jean-Michel Ghidaglia for very helpful discussions and his support. Without him the *VOLNA* code could not have been developed in such a short term. The authors would also like to acknowledge Professor Costas Synolakis for valuable suggestions. His work on tsunami waves is an endless source of inspiration.

References

- [AB05] E. Audusse and M.-O. Bristeau. A well-balanced positivity preserving "second-order" scheme for shallow water flows on unstructured meshes. *J. Comput. Phys.*, 206:311–333, 2005.
- [ABB⁺04] E. Audusse, F. Bouchut, O.M. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM J. of Sc. Comp.*, 25:2050–2065, 2004.
- [AC99] K. Anastasiou and C. T. Chan. Solution of the 2d shallow water equations using the finite volume method on unstructured triangular meshes. *International Journal for Numerical Methods in Fluids*, 24:1225–1245, 1999.
- [AGN05] F. Alcrudo and P. Garcia-Navarro. A high-resolution Godunov-type scheme in finite volumes for the 2d shallow-water equations. *Int. J. Numer. Methods Fluids*, 16:489–505, 2005.
- [AMS04] F. Archambeau, N. Mehitoua, and M. Sakiz. Code Saturne: A finite volume code for the computation of turbulent incompressible flows - industrial applications. *International Journal On Finite Volumes*, 1:1–62, 2004.
- [Aud04] E. Audusse. *Modélisation hyperbolique et analyse numérique pour les écoulements en eaux peu profondes*. PhD thesis, Université Paris VI, 2004.
- [BB73] J.P. Boris and D.L. Book. Flux corrected transport: Shasta, a fluid transport algorithm that works. *J. Comp. Phys.*, 11:38–69, 1973.
- [BBMA01] M. Brocchini, R. Bernetti, A. Mancinelli, and G. Albertini. An efficient solver for nearshore flows based on the WAF method. *Coastal Engineering*, 43:105–129, 2001.
- [BCCC97] P. Batten, N. Clarke, Lambert C., and D.M. Causon. On the choice of wavespeeds for the HLLC Riemann solver. *SIAM J. Sci. Comput.*, 18(6):1553–1570, 1997.
- [BD03] D. Bresch and B. Desjardins. Existence of global weak solutions for a 2d viscous shallow water equations and convergence to the quasi-geostrophic model. *Communications in Mathematical Physics*, 238:211–223, 2003.
- [BDDV98] A. Bermudez, A. Dervieux, J.-A. Desideri, and M. E. Vazquez. Upwind schemes for the two-dimensional shallow water equations with variable depth using unstructured meshes. *Computer Methods in Applied Mechanics and Engineering*, 155:49–72, 1998.
- [BF90] T.J. Barth and P.O. Frederickson. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. *AIAA*, 90-0013, 1990.

- [BJ89] T.J. Barth and D.C. Jespersen. The design and application of upwind schemes on unstructured meshes. *AIAA*, 0366, 1989.
- [BO04] T.J. Barth and M. Ohlberger. *Encyclopedia of Computational Mechanics, Volume 1, Fundamentals*, chapter Finite Volume Methods: Foundation and Analysis. John Wiley and Sons, Ltd, 2004.
- [BQ06] F. Benkhaldoun and L. Quivy. A non homogeneous riemann solver for shallow water and two phase flows. *Flow, Turbulence and Combustion*, 76:391–402, 2006.
- [Cas90] V. Casulli. Semi-implicit finite difference methods for the two-dimensional shallow water equation. *Journal of Computational Physics*, 86:56–74, 1990.
- [CFGR⁺05] M.J. Castro, A.M. Ferreiro, J.A. Garcia-Rodriguez, J.M. Gonzalez-Vida, J. Macias, C. Pares, and M.E. Vazquez-Cendon. The numerical treatment of wet/dry fronts in shallow flows: Application to one-layer and two-layer systems. *Mathematical and Computer Modelling*, 42:419–439, 2005.
- [CG00] J. Cortes and J.-M. Ghidaglia. Upwinding at low cost for complex models and flux schemes. In *Trends in Numerical and Physical Modeling for Industrial Multiphase Flows*, 2000.
- [CIM⁺00] D.M. Causon, D.M. Ingram, C.G. Mingham, G. Yang, and R.V. Pearson. Calculation of shallow water flows using a cartesian cut cell approach. *Advances in Water Resources*, 23:545–562, 2000.
- [CLS04] B. Cockburn, F. Li, and C.-W. Shu. Locally divergence-free discontinuous Galerkin methods for the Maxwell equations. *Journal of Computational Physics*, 194:588–610, 2004.
- [CP92] J. H. E. Cartwright and O. Piro. The dynamics of Runge–Kutta methods. *Int. J. Bifurcation and Chaos*, 2:427–449, 1992.
- [CWY03] G. F. Carrier, T. T. Wu, and H. Yeh. Tsunami run-up and draw-down on a plane beach. *J. Fluid Mech.*, 475:79–99, 2003.
- [Dav88] S.F. Davis. Simplified second-order Godunov-type methods. *SIAM J. Sci. Statist. Comput.*, 9:445–473, 1988.
- [DD07a] D. Dutykh and F. Dias. Dissipative Boussinesq equations. *C. R. Mecanique*, 335:559–583, 2007.
- [DD07b] D. Dutykh and F. Dias. Viscous potential free-surface flows in a fluid layer of finite depth. *C. R. Acad. Sci. Paris, Ser. I*, 345:113–118, 2007.
- [DD07c] D. Dutykh and F. Dias. Water waves generated by a moving bottom. In Anjan Kundu, editor, *Tsunami and Nonlinear waves*. Springer Verlag (Geo Sc.), 2007.

- [DD09a] D. Dutykh and F. Dias. Energy of tsunami waves generated by bottom motion. *Proc. R. Soc. A*, 465:725–744, 2009.
- [DD09b] D. Dutykh and F. Dias. Tsunami generation by dynamic displacement of sea bed due to dip-slip faulting. *Mathematics and Computers in Simulation*, 80(4):837–848, 2009.
- [DD10] D. Dutykh and F. Dias. Influence of sedimentary layering on tsunami generation. *Computer Methods in Applied Mechanics and Engineering*, In Press, 2010.
- [DDG08a] F. Dias, D. Dutykh, and J.-M. Ghidaglia. A compressible two-fluid model for the finite volume simulation of violent aerated flows. Analytical properties and numerical results. Research report, CMLA, ENS de Cachan, 2008.
- [DDG08b] F. Dias, D. Dutykh, and J.-M. Ghidaglia. Simulation of free surface compressible flows via a two fluid model. In *Proceedings of OMAE2008 27th International Conference on Offshore Mechanics and Arctic Engineering, June 15-20, 2008, Estoril, Portugal*, 2008.
- [DDG10] F. Dias, D. Dutykh, and J.-M. Ghidaglia. A two-fluid model for violent aerated flows. *Computers and Fluids*, 39:283–293, 2010.
- [DDK06] D. Dutykh, F. Dias, and Y. Kervella. Linear theory of wave generation by a moving bottom. *C. R. Acad. Sci. Paris, Ser. I*, 343:499–504, 2006.
- [DKK07] A. I. Delis, M. Kazolea, and N. A. Kampanis. A robust high-resolution finite volume scheme for the simulation of long waves over complex domains. *Int. J. Numer. Meth. Fluids*, 56:419–452, 2007.
- [DM10] F. Dias and P. Milewski. On the fully-nonlinear shallow-water generalized Serre equations. *Physics Letters A*, 374(8):1049–1053, 2010.
- [Dut07] D. Dutykh. *Mathematical modelling of tsunami waves*. PhD thesis, École Normale Supérieure de Cachan, 2007.
- [Dut09] D. Dutykh. Visco-potential free-surface flows and long wave modelling. *Eur. J. Mech. B/Fluids*, 28:430–443, 2009.
- [EMRS91] B. Einfeldt, C.D. Munz, P.L. Roe, and B. Sjogreen. On Godunov-type methods near low densities. *J. Comput. Phys.*, 92:273–295, 1991.
- [FT95] L. Fraccarollo and E.F. Toro. Experimental and numerical assessment of the shallow water model for two-dimensional dam break type problems. *Journal of Hydraulic Research*, 33:843–864, 1995.
- [Geo06] D.L. George. *Finite Volume Methods and Adaptive Refinement for Tsunami Propagation and Inundation*. PhD thesis, Department of Applied Mathematics, University of Washington, Seattle, 2006.
- [Geo08] D.L. George. Augmented Riemann solvers for the shallow water equations over variable topography with steady states and inundation. *J. Comput. Phys.*, 227:3089–3113, 2008.

- [Ghi95] J.-M. Ghidaglia. Une approche volumes finis pour la résolution des systèmes hyperboliques de lois de conservation, note. Technical report, Département Transferts Thermiques et Aérodynamique, Direction des Etudes et Recherches, Electricité de France, HT-30/95/015/A, 1995.
- [Ghi98] J.-M. Ghidaglia. Flux schemes for solving nonlinear systems of conservation laws. In J.J. Chattot and M. Hafez, editors, *Proceedings of the meeting in honor of P.L. Roe*, Arcachon, July 1998.
- [GHS03] T. Gallouet, J.-M. Hérard, and N. Seguin. Some approximate Godunov schemes to compute shallow-water equations with topography. *Computers & Fluids*, 32:479–513, 2003.
- [GKC96] J.-M. Ghidaglia, A. Kumbaro, and G. Le Coq. Une méthode volumes-finis à flux caractéristiques pour la résolution numérique des systèmes hyperboliques de lois de conservation. *C. R. Acad. Sci. I*, 322:981–988, 1996.
- [GKC01] J.-M. Ghidaglia, A. Kumbaro, and G. Le Coq. On the numerical solution to two fluid models via cell centered finite volume method. *Eur. J. Mech. B/Fluids*, 20:841–867, 2001.
- [GL06] D.L. George and R.J. Leveque. Finite volume methods and adaptive refinement for global tsunami propagation and local inundation. *Sci. Tsunami Hazards*, 24(5):319, 2006.
- [Gla88] P. Glaister. Approximate Riemann solutions of the shallow water equations. *Journal of Hydraulic Research*, 26:293–300, 1988.
- [GNVC00] P. Garcia-Navarro and M. E. Vazquez-Cendon. On numerical treatment of the source terms in the shallow water equations. *Computers & Fluids*, 29:951–979, 2000.
- [God59] S.K. Godunov. A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sb.*, 47:271–290, 1959.
- [GOSI97] C. Goto, Y. Ogawa, N. Shuto, and F. Imamura. Numerical method of tsunami simulation with the leap-frog scheme. Technical report, UNESCO, 1997.
- [GP05] J.-M. Ghidaglia and F. Pascal. The normal flux method at the boundary for multidimensional finite volume approximations in cfd. *European Journal of Mechanics B/Fluids*, 24:1–17, 2005.
- [GST01] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43:89–112, 2001.
- [GV85] J.D. Goodman and R.J. Le Veque. On the accuracy of stable schemes for 2D conservation laws. *Math. Comp.*, 45(171):15–21, 1985.
- [Har83] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comp. Phys.*, 49:357–393, 1983.

- [HC89] D.G. Holmes and S.D. Connel. Solution of the 2d Navier-Stokes equations on unstructured adaptive grids. In *AIAA 9th Computational Fluid Dynamics Conference*, volume 89-1932-CP, June 1989.
- [HLvL83] A. Harten, P.D. Lax, and B. van Leer. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25:35–61, 1983.
- [HP79] S. Hibberd and D.H. Peregrine. Surf and run-up on a beach: a uniform bore. *J. Fluid Mech.*, 95:323–345, 1979.
- [Ima96] F. Imamura. *Long-wave runup models*, chapter Simulation of wave-packet propagation along sloping beach by TUNAMI-code, pages 231–241. World Scientific, 1996.
- [KCY07] D.-H. Kim, Y.-S. Cho, and Y.-K. Yi. Propagation and run-up of nearshore tsunamis with HLLC approximate Riemann solver. *Ocean Engineering*, 34:1164–1173, 2007.
- [KDD07] Y. Kervella, D. Dutykh, and F. Dias. Comparison between three-dimensional linear and nonlinear tsunami generation models. *Theor. Comput. Fluid Dyn.*, 21:245–269, 2007.
- [KMC03] S.-E. Kim, B. Makarov, and D. Caraeni. A multi-dimensional linear reconstruction scheme for arbitrary unstructured grids. Technical report, Fluent Inc., 2003.
- [Kol72] N.E. Kolgan. Application of the minimum-derivative principle in the construction of finite-difference schemes for numerical analysis of discontinuous solutions in gas dynamics. *Uchenye Zapiski TsAGI [Sci. Notes Central Inst. Aerodyn]*, 3(6):68–77, 1972.
- [Kol75] N.E. Kolgan. Finite-difference schemes for computation of three dimensional solutions of gas dynamics and calculation of a flow over a body under an angle of attack. *Uchenye Zapiski TsAGI [Sci. Notes Central Inst. Aerodyn]*, 6(2):1–6, 1975.
- [Kro97] D. Kroner. *Numerical Schemes for Conservation Laws*. Wiley, 1997.
- [Lax73] P.D. Lax. *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*. SIAM, Philadelphia, Penn., 1973.
- [LLS03] P.L.-F. Liu, P. Lynett, and C.E. Synolakis. Analytical solutions for forced long waves on a sloping beach. *J. Fluid Mech.*, 478:101–109, 2003.
- [LR98] A.-Y. Le Roux. Riemann solvers for some hyperbolic problems with a source term. *ESAIM: Proceedings*, 6:75–90, 1998.
- [LWC98] P.L.-F. Liu, S.-B. Woo, and Y.-K. Cho. Computer programs for tsunami propagation and inundation. Technical report, School of Civil and Environmental Engineering, Cornell University, 1998.

- [MBS03] P. A. Madsen, H. B. Bingham, and H. A. Schaffer. Boussinesq-type formulations for fully nonlinear and extremely dispersive water waves: derivation and analysis. *Proc. R. Soc. Lond. A*, 459:1075–1104, 2003.
- [MG96] S. Musaferija and D. Gosman. Finite-volume CFD procedure and adaptive error control strategy for grids of arbitrary topology. *J. Comp. Phys.*, 138:766–787, 1996.
- [MSS97] P.A. Madsen, H.A. Sorensen, and H.A. Schaffer. Surf zone dynamics simulated by a Boussinesq-type model. Part I. Model description and cross-shore motion of regular waves. *Coastal Engineering*, 32:255–287, 1997.
- [NPPN06] S. Noelle, N. Pankratz, G. Puppo, and J.R. Natvig. Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows. *J. Comput. Phys.*, 213:474–499, 2006.
- [OHK97] H.T. Ozkan-Haller and J.T. Kirby. A Fourier-Chebyshev collocation method for the shallow water equations including shoreline runup. *Applied Ocean Research*, 19:21–34, 1997.
- [Osh84] S. Osher. Riemann solvers, the entropy condition, and difference approximations. *SIAM J. Numer. Anal.*, 21(2):217–235, 1984.
- [Per67] D. H. Peregrine. Long waves on a beach. *J. Fluid Mech.*, 27:815–827, 1967.
- [Pet91] T. Peterson. A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM J. Numer. Anal.*, 28(1):133–140, 1991.
- [Roe81] P. L. Roe. Approximate riemann solvers, parameter vectors and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.
- [SB06] C.E. Synolakis and E.N. Bernard. Tsunami science before and beyond Boxing Day 2004. *Phil. Trans. R. Soc. A*, 364:2231–2265, 2006.
- [SBG04] B. Sapoval, A. Baldassarri, and A. Gabrielli. Self-stabilized fractality of seacoasts through damped erosion. *Phys. Rev. Lett.*, 93:098501 [4 pages], 2004.
- [SBT⁺07] C.E. Synolakis, E.N. Bernard, V.V. Titov, U. Kanoglu, and F.I. Gonzalez. Standards, criteria, and procedures for NOAA evaluation of tsunami numerical models. Technical report, NOAA/Pacific Marine Environmental Laboratory, 2007.
- [Ser53] F. Serre. Contribution à l’étude des écoulements permanents et variables dans les canaux. *La Houille blanche*, 8:374–388 & 830–872, 1953.
- [Shu88] C.-W. Shu. Total-variation-diminishing time discretizations. *SIAM J. Sci. Statist. Comput.*, 9:1073–1084, 1988.

- [SO88] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.
- [SR02] R. J. Spiteri and S. J. Ruuth. A new class of optimal high-order strong-stability-preserving time discretization methods. *SIAM Journal on Numerical Analysis*, 40:469–491, 2002.
- [Sto57] J.J. Stoker. *Water Waves: The mathematical theory with applications*. Interscience, New York, 1957.
- [Syn05] C. Synolakis. India must cooperate on tsunami warning system. *Nature*, 434:17–18, 2005.
- [Tat97] H. Tatehata. *Perspectives on Tsunami Hazard Reduction: Observations, Theory and Planning*, chapter The New Tsunami Warning System of the Japan Meteorological Agency, pages 175–188. Springer, 1997.
- [TG97] V.V. Titov and F.I. González. Implementation and testing of the method of splitting tsunami (MOST) model. Technical Report ERL PMEL-112, Pacific Marine Environmental Laboratory, NOAA, 1997.
- [TGB⁺05] V.V. Titov, F.I. Gonzalez, E. N. Bernard, M.C. Eble, H.O. Mofjeld, J.C. Newman, and A.J. Venturato. Real-time tsunami forecasting: Challenges and solutions. *Natural Hazards*, 35:41–58, 2005.
- [Tor92] E.F. Toro. Riemann problems and the WAF method for solving the two-dimensional shallow water equations. *Philosophical Transactions: Physical Sciences and Engineering*, 338:43–68, 1992.
- [TSS94] E.F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the HLL Riemann solver. *Shock Waves*, 4:25–34, 1994.
- [VC99] M.E. Vazquez-Cendon. Improved treatment of source terms in upwind schemes for the shallow water equations in channels with irregular geometry. *Journal of Computational Physics*, 148:497–526, 1999.
- [vL79] B. van Leer. Towards the ultimate conservative difference scheme v: a second order sequel to Godunov’ method. *J. Comput. Phys.*, 32:101–136, 1979.
- [vL06] B. van Leer. Upwind and high-resolution methods for compressible flow: From donor cell to residual-distribution schemes. *Communications in Computational Physics*, 1:192–206, 2006.
- [WMC06] Y. Wei, X.-Z. Mao, and K.F. Cheung. Well-balanced finite-volume model for long-wave runup. *Journal of Waterwave, Port, Coastal and Ocean Engineering*, 132:114–124, 2006.
- [ZCIM02] J.G. Zhou, D.M. Causon, D.M. Ingram, and C.G. Mingham. Numerical solutions of the shallow water equations with discontinuous bed topography. *Int. J. Numer. Meth. Fluids*, 38:769–788, 2002.

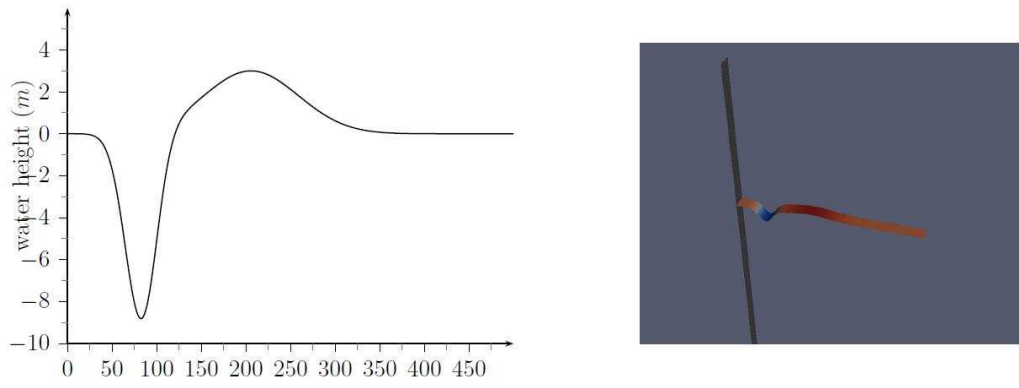


Fig. 26. Catalina 1 benchmark — initial free surface profile (left) ; bathymetry and initial free surface, at the same vertical scale (right).

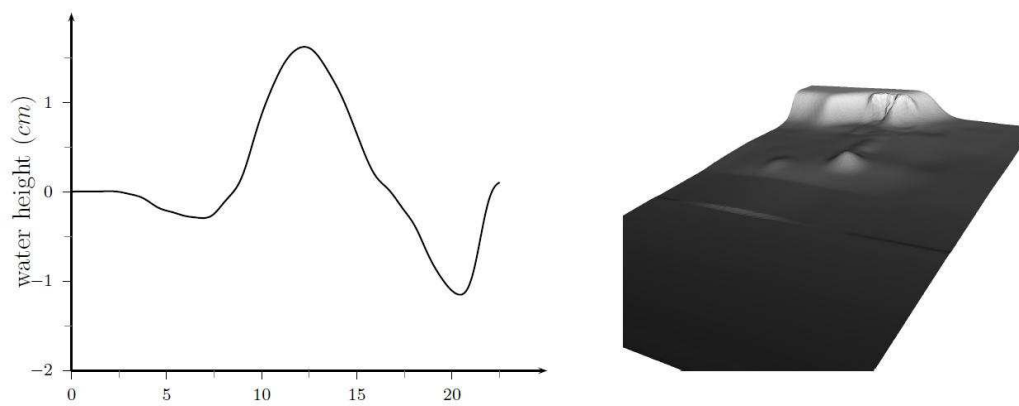


Fig. 27. Catalina 2 benchmark : initial free surface profile (left) ; bathymetry (right).

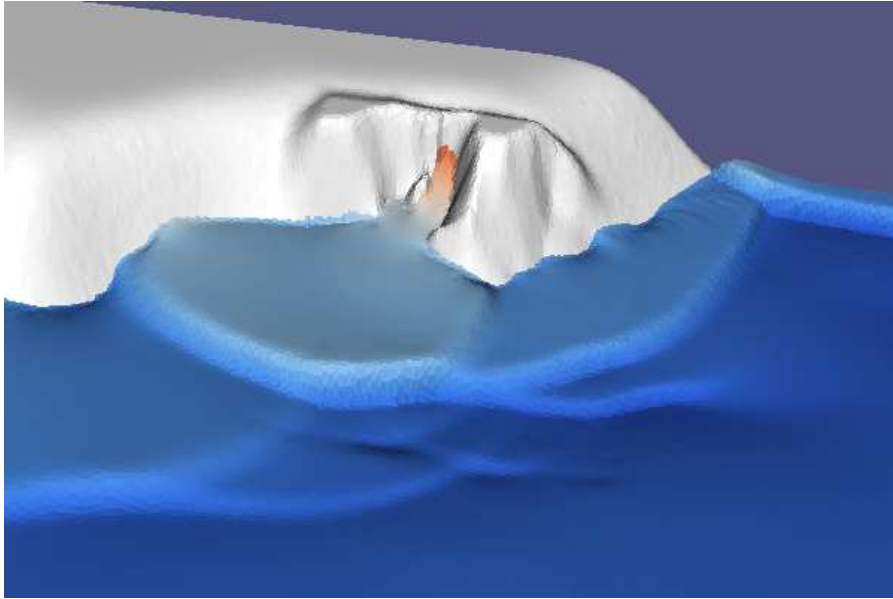


Fig. 28. Maximal runup in Monai valley. The vertical scale is magnified by a factor 3.

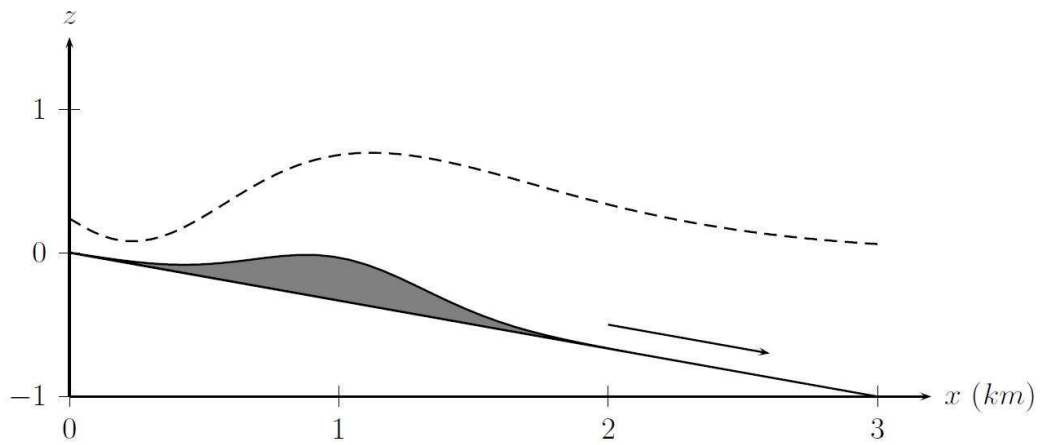


Fig. 29. Sketch of the analytical landslide test case. The submarine mass displacement (in gray) and free surface (dashed line) have been magnified by a factor 1000.