$r-$adaptation for Shallow Water flows: conservation, well balancedness, efficiency

Luca Arpaia and Mario Ricchiuto

early draft

Abstract

We investigate the potential of the so-called "relocation" mesh adaptation in terms of resolution and efficiency for the simulation of free surface flows in the near shore region. Our work is developed in three main steps. First, we consider several Arbitrary Lagrangian Eulerian (ALE) formulations of the shallow water equations on moving grids, and provide discrete analogs in the Finite Volume and Residual Distribution framework. The compliance to all the physical constraints, often in competition, is taken into account. We consider different formulations allowing to combine volume conservation (DGCL) and equilibrium (Well-Balancedness), and we clarify the relations between the so-called pre-balanced form of the equations (Rogers et al., *J.Comput.Phys.* 192, 2003), and the classical upwiniding of the bathymetry term gradients (Bermudez and Vazquez, *Computers and Fluids* 235, 1994). Moreover, we propose a simple remap of the bathymetry based on high accurate quadrature on the moving mesh which, while preserving an accurate representation of the initial data, also allows to retain mass conservation within an arbitrary accuracy. Second, the coupling of the resulting schemes with a mesh partial differential equation is studied. Since the flow solver is based on genuinely explicit time stepping, we investigate the efficiency of three coupling strategies in terms of cost overhead w.r.t. the flow solver. We analyze the role of the solution remap necessary to evaluate the error monitor controlling the adaptation, and propose simplified formulations allowing a reduction in computational cost. The resulting ALE algorithm is compared with the *rezoning* Eulerian approach with interpolation proposed e.g. in (Tang and Tang, *SINUM* 41, 2003). An alternative cost effective Eulerian approach, still allowing a full decoupling between adaptation and flow evolution steps is also proposed. Finally, a thorough numerical evaluation of the methods discussed is performed. Numerical results on propagation, and inundation problems shows that the best compromise between accuracy and CPU time is provided by a full ALE formulation. If a loose coupling with the mesh adaptation is sought, then the cheaper Eulerian approach proposed is shown to provide results quite close to the ALE.

1 Introduction

We investigate the potential benefits of r-adaptation techniques (or relocation adaptation) in the computation of propagation and interaction of free surface waves, including their runup on complex bathymetries. The main building blocks of our study are the following. First, we use the well known Shallow Water equations to model the hydrostatic free surface hydrodynamics in vicinity of the shore. The use of moving meshes will lead us to investigate various forms of the model equations in an Arbitrary-Lagrangian-Eulerian (ALE) setting. Particular attention is paid to the respect of all the physical constraints. Second, the equations are coupled with a Laplacian-based adaptive mesh deformation technique. The coupling between

flow evolution and mesh Partial Differential Equation (PDE) is discussed. Third, a thorough quantitative evaluation of the resulting algorithms on benchmarks involving wave propagation and inundation of complex bathymetries is performed.

The numerical approximation of Shallow Water flows is still a subject of intense research. For our purposes, the most interesting issue is the need of preserving, possibly to machine accuracy, the so-called lake at rest steady state. This property is known as C-property or *well balacedness* (WB). The initial work of [1] on the construction of well-balanced Finite-Volume approximations in one dimension, has been led throughout the years to many different results allowing the construction of unstructured mesh discretizations verifying the C-property via an appropriate coupling of the numerical flux and numerical source terms [2, 3], or based on different forms of the equations, as the well-balanced form of [4, 5, 6], or the so-called *pre-balanced* form of Rogers et al. [7, 8, 9]. These ideas have been also incorporated in Finite Element, Residual Distribution, and Discontinuous Galerkin methods (see e.g. [10, 11, 12, 13, 14] and references therein).

To enhance the resolution of complex wave patterns, and of the wetting/drying dynamics we study the use of mesh adaptation techniques based on nodes redistribution (or relocation). These are known as r-adaptation techniques. The reason for this choice is, on one hand the overhead represented by remeshing techniques [15, 16] w.r.t. a single time step of a fully explicit discretization of the shallow water model, on the other the potential shown in the past for these techniques in e.g. [17], and in the numerous works of Budd and collaborators (see e.g. the review [18] and references therein). Nodal movement is obtained by solving an appropriate Moving Mesh Partial Differential Equation (MMPDE). Originally, the equations for grid movement were developed using the visual analogy between potential solutions and curvilinear grids, thus solving, as for the flow potential, a Laplace equation for some reference/computational coordinates $\boldsymbol{X}$ on the actual/physical grid $\boldsymbol{x}$. In case of solution-driven adaptation a forcing term is added to take into account the regularity of the solution, and usually defined through a monitor function, see [19] for a thorough derivation. Equidistribution of the monitor function is nowday a standard way to achieve an optimal mesh. The central idea is to find a transformation $\boldsymbol{x} = M(\boldsymbol{X}, t)$ that equidistributes the monitor function on the reference domain. During the last decades theoretical arguments and experience lead to the design of quite general monitor functions which can ensure the adaptation to particular features of the solution; the arclenght-type monitor function of Winslow [20], based on solution gradients, is one of the most successful. Ceniceros and Hou [21], for example, used the temperature gradient in the computation of small scales blow up in Boussinesq convection. Budd, Callen and Walsh [22] compared arclength of the potential temperature with a different monitor function, based on potential vorticity, for the resolution of weather front formation. In more recent years, Huang [23] studied matrix valued monitor functions which can provide both control on mesh size and shape/orientation. A variational formulation of the equidistribution principle allows to take into account specific mesh quality measures such as orthogonality and smoothness, see here the pioneering work of Brackbill and Saltzman [24]. To further enhance the mesh quality, node insertion/deletion can be performed by appropriate local remeshing strategies. These, however, required much higher overheads and more complex data structures, compared to a single step of an explicit discretization of the flow PDEs, and are not considered here. For recent examples the interested reader may refer to [15, 16].

The coupling of the flow solver with the mesh at each time step is non-trivial, as the mesh equations depend on the solution on the (unknown) adapted mesh. In particular the Shallow Water equations and the MMPDE can be either solved *simultaneously* or *alternately*. The latter has been successfully implemented by [21], showing a significant reduction of stiffness problems even if it can lead to a lag in the mesh movement with respect to the physical features. Depending on the framework in which we evolve the PDE, two different *alternate* algorithms are tested at this point. If the PDE is written in Eulerian framework one get the *rezoning* method suggested in [17]. This approach, based on a sequence of mesh and flow iterations, uses the mesh solver as a black box, the flow equations being solved on a (different) fixed mesh at each time iteration. Its drawback is that, at each time iteration, the flow solver requires a remap/interpolation on the new mesh which may be quite expensive as it needs to guarantee the same properties as the flow solver itself (high order accuracy, non-oscillatory character/positivity preservation, C-property, mass conservation). At the opposite, once the grid has been adapted, one can evolve the flow with an Arbitrary-Lagrangian-Eulerian formulation of the equations, as suggested e.g. in [25]. In this case,

the properties of the flow solutions are only determined by the scheme. However, a proper ALE form of the numerical discretization has to be used. In particular, a well known requirement for ALE discretizations is the compatibility with a Geometric Conservation Law (GCL), which guarantees that no artificial volume (*viz* mass) is produced in the computational domain due to mesh motion. The discrete counterpart of this property is known as the DGCL (cf. [26], [27] for an overview). Ideally, in Shallow Water flows, we have to ensure the satisfaction of both a discrete analog of the GCL, and of the C-property, while still being able to conserve mass and momentum. A solution based on an ALE remap of the bathymetry has been suggested in [28]. However, unless such remap is very high order accurate, this quickly leads to a smoothing of the data, hence a re-initialization of the topography is required, spoiling mass conservation.

In this paper, we propose and evaluate simplified strategies allowing adaptive simulations of Shallow Water flows with wetting/drying fronts. In order to do this, we systematically review the forms of the equations which are best suited for the task of combining well-balancedness and DGCL on moving meshes; we use the resulting model equations to provide well-balanced high-order Finite Volume (FV) and Residual Distribution (RD) discretizations, clarifying the relations between the pre-balanced and well-balanced approaches; we provide a simple recipe to marry mass conservation and C-property on moving meshes using a re-interpolation of the nodal bathymetry based on accurate quadrature of the given bathymetric data; we define improved ad-hoc error estimators allowing to better track both smooth waves, and shorelines; finally, coupling strategies allowing cheaper and simpler interpolation algorithms in the adaptation phase, while retaining all the desired discrete properties, are evaluated in terms of CPU time for a given resolution, using standard benchmarks for near shore hydrodynamics.

The paper is organized as follows: section §2 presents the general setting, and it particular it recalls the main forms and properties of the Shallow Water equations, and of a simpler scalar model used to simplify part of the discussion. The well-balanced numerical approximation of the PDEs in ALE form with Finite Volume and Residual distribution schemes is discussed in section §3, with a discussion of the appropriate ALE form for balance laws. The moving mesh algorithm is presented in section §4, with some details concerning the management of wet-dry areas in shallow water simulations. In section §4.2 the interpolation strategy of [29] is presented as a conservative ALE remap. Three strategies to couple mesh movement and balance laws solution are presented in section section §5: the rezoning algorithm of Tang coupled with the SW equations, see Zhou [29] (EUL1), an improved version of the above algorithm (EUL2) and the ALE coupling. Finally, section §7 and section §8 presents a thorough study of the coupled algorithms in terms of accuracy, and CPU time for both simple academic problems and for some classical benchmarks involving the long wave runup on complex bathymetries. The paper is ended by a summary of the main results, and by an overlook at future developments.

2 Problem setting and model equations

2.1 Shallow Water equations and lake at rest

Our final objective is the simulation of the propagation and runup of free surface waves in the near shore region. A good model for the physics of these waves is given by the nonlinear Shallow Water equations, providing a depth averaged description of the flow, and reading

$$\frac{\partial \mathsf{u}}{\partial t} + \nabla \cdot \mathbf{f}(\mathsf{u}) + \mathsf{S}(\mathsf{u}, b) = 0\,, \qquad \boldsymbol{x} \in \Omega \tag{1}$$

where the conserved variables, flux function and source term are given by

$$\mathsf{u} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathbf{f}(\mathsf{u}) = \begin{bmatrix} hu & hv \\ hu^2 + \frac{1}{2}gh^2 & huv \\ huv & hv^2 + \frac{1}{2}gh^2 \end{bmatrix}, \quad \mathsf{S} = \begin{bmatrix} 0 \\ gh\dfrac{\partial b}{\partial x} \\ gh\dfrac{\partial b}{\partial y} \end{bmatrix} \tag{2}$$

with h the depth, $\boldsymbol{u} = (u,\, v)$ the depth averaged velocity, g the gravity acceleration, and $b = b(x, y)$ the bathymetry level. Equations (1)-(2) consitutes a non-homogeneous hyperbolic system of partial differential equations. In particular, given any vector $\boldsymbol{\xi} = (\xi_x, \xi_y) \in \mathbb{R}^2$ the flux Jacobian $\mathsf{K}(\boldsymbol{\xi}, \mathsf{u}) = \partial(\mathbf{f}(\mathsf{u}) \cdot \boldsymbol{\xi})/\partial \mathsf{u}$ admits a full set of real eigenvalues and linearly independent eigenvectors, namely

$$\mathsf{K}(\boldsymbol{\xi}, \mathsf{u}) = \begin{pmatrix} 0 & \xi_x & \xi_y \\ c^2\xi_x - u\boldsymbol{u}\cdot\boldsymbol{\xi} & \boldsymbol{u}\cdot\boldsymbol{\xi} + u\xi_x & u\xi_y \\ c^2\xi_y - v\boldsymbol{u}\cdot\boldsymbol{\xi} & v\xi_x & \boldsymbol{u}\cdot\boldsymbol{\xi} + v\xi_y \end{pmatrix} \tag{3}$$

with eigenvalues

$$\lambda_{1,3}(\mathsf{u}, \boldsymbol{\xi}) = \boldsymbol{u}\cdot\boldsymbol{\xi} \pm c\|\boldsymbol{\xi}\|, \quad \lambda_2(\mathsf{u}, \xi) = \boldsymbol{u}\cdot\boldsymbol{\xi} \tag{4}$$

with $c = \sqrt{gh}$. Later in the text we will also make use of the Jacobian at rest

$$\mathsf{A}(\xi, h) = \begin{pmatrix} 0 & \xi_x & \xi_y \\ gh\,\xi_x & 0 & 0 \\ gh\,\xi_y & 0 & 0 \end{pmatrix} \tag{5}$$

In the context of Shallow Water flows, an important role is played by the so-called "lake at rest" state which, denoting the free surface level $\eta = h + b$, and the discharge by $\boldsymbol{q} = h\boldsymbol{u}$, is the particular steady solution characterized by the two invariants :

$$\boldsymbol{q} = 0, \qquad h(x, y) + b(x, y) = \eta_0 = const \tag{6}$$

A numerical method approximating solutions of (1)-(2) is said to enjoy the *C-property* or also to be *well-balanced* if (6) is also an exact steady solution of the discrete equations. In other words, Well-balanced schemes provide a discrete analog of the relation

$$\nabla \cdot \mathbf{f} + \mathsf{S} = 0$$

allowing to preserve (6) exactly at the discrete level.

2.2 A scalar model

To illustrate some concepts and to better highlight certain numerical effects, we will also consider a simplified model mimicking the Shallow Water equations. This model reads

$$\frac{\partial u}{\partial t} + \nabla \cdot \boldsymbol{f}(u) + S(u, \boldsymbol{x}) = 0\,, \qquad \boldsymbol{x} \in \Omega \tag{7}$$

where, for a given flux $\boldsymbol{f}(u)$, the source term is defined as

$$S = \boldsymbol{a}(u) \cdot \nabla b$$

with $b = b(x, y)$ a given function, and with the flux Jacobian $\boldsymbol{a} = \partial \boldsymbol{f}(u)/\partial u$. The following definition of the fluxes will be used, $\boldsymbol{f}(u) = \boldsymbol{a}(x, y)u$, with $\nabla \cdot \boldsymbol{a} = 0$. Introducing the variable $\eta = u + b$, equation (7) admits a non-trivial steady state given by

$$u(x, y) + b(x, y) = \eta_0 = const \tag{8}$$

A numerical method approximating solutions of (7) is said to enjoy the *C-property* or also to be *well-balanced* if (8) is also an exact steady solution of the discrete equations. Well-balanced schemes provide a discrete analog of the relation $\nabla \cdot \boldsymbol{f} + S = 0$ allowing to preserve (8) exactly at the discrete level.

3 Numerical approximation on moving meshes

To embed adaptive mesh deformation in the numerical solution of (1)(2) an appropriate Arbitrary Lagrangian Eulerian (ALE) formulation will be used. The objective of the following sections is to recall some basic aspects related to ALE, to show how different forms of the PDE impact the possibility of combining the Geometric Conservation Law (DGCL) with the C-property. For more details concerning the ALE formalism, the interested reader can refer to e.g. [30].

3.1 ALE basics

We start by considering a field of displacements for the points of the domain Ω, from the reference position $\boldsymbol{X}$ to the actual one $\boldsymbol{x}(t)$ according to the motion law

$$\left.\frac{\partial \boldsymbol{x}(t)}{\partial t}\right|_X = \boldsymbol{\sigma}(\boldsymbol{x}, t), \tag{9}$$

The symbol $|_X$ denotes derivatives computed in a coordinate system following the trajectory of the domain points. Solving (9) with $\boldsymbol{x}(0) = \boldsymbol{X}$, gives back the actual configuration through the mapping

$$M(t) : \Omega_X \to \Omega(t), \qquad \boldsymbol{x} = M(\boldsymbol{X}, t) \tag{10}$$

An important role is played by the Jacobian matrix of the mapping

$$\mathcal{J}_M = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}, \quad J_M = \det \mathcal{J}_M \geq 0$$

The determinant of the Jacobian satisfies the fundamental relation

$$\left.\frac{\partial J_M}{\partial t}\right|_X = J_M \nabla \cdot \boldsymbol{\sigma} \tag{11}$$

which is known as the Geometric Conservation Law (GCL), and is a local variant of volume conservation, reading for a moving control volume $C(t)$

$$\left.\frac{\partial C(t)}{\partial t}\right|_X = \int_{C(t)} \nabla \cdot \boldsymbol{\sigma}\, d\boldsymbol{x} \tag{12}$$

Lastly, we note that for a given function of the spatial coordinates $b(\boldsymbol{x})$, we can write

$$\left.\frac{\partial b(\boldsymbol{x})}{\partial t}\right|_X = \left.\frac{\partial \boldsymbol{x}(t)}{\partial t}\right|_X \cdot \nabla b$$

so that using (9) we have

$$\left.\frac{\partial b}{\partial t}\right|_X - \boldsymbol{\sigma} \cdot \nabla b = 0 \tag{13}$$

This last equation rapresents the time variation of the function $b(\boldsymbol{x}(t))$ measured from an observer which is following the domain motion $\boldsymbol{x} = M(\boldsymbol{X}, t)$. Summing Eq. (13) (pre-multiplied by J_M) to Eq. (11) (pre-multiplied by b) leads to the ALE remap equation for the function b

$$\left.\frac{\partial (J_M b)}{\partial t}\right|_X - J_M \nabla \cdot (b\boldsymbol{\sigma}) = 0 \tag{14}$$

3.2 Notation for mesh, geometry and unkonwns

Consider a discretization of the spatial domain Ω composed by non overlapping triangular elements. We will denote the grid by $\mathcal{T}_h$, h_K being the local reference element length. K is the generic triangle and $|K|$ its area. The vector $\mathbf{x} = \{...\, x_i \,...\, y_i ...\}$ with $i \in \mathcal{T}_h$ denotes the set of the nodal coordinates of the mesh. For every node i of the triangulation, $\mathcal{D}_i$ denotes the subset of triangles containing i. With a little abuse in the notation $j \in \mathcal{D}_i$ is the set of nodes j sharing an edge with node i. We then denote by C_i the median dual cell obtained by joining the gravity centers of the triangles in $\mathcal{D}_i$ with the midpoints of the edges meeting in i

$$|C_i| = \sum_{K \in \mathcal{D}_i} \frac{|K|}{3}$$

In a Finite Volume context we define also the boundary of the median dual cell as the interface $\partial C_i = \sum_{j \in \mathcal{D}_i} \partial C_{ij}$. The interface belonging to nodes i, j, denoted as ∂C_{ij}, is the union of two segments connecting the baricenters of the adiacent triangles $K \ni i, j$ with the midpoint of the edge ij (cf. right picture in figure 1). C_{ij} is the area delimited by ∂C_{ij} and by the two segments joining i with the gravity centers of the elements $K \ni i, j$. Both C_{ij} and ∂C_{ij} can be splitted over the adiacent triangles. We define the normal and area associated to the interface ij

$$\boldsymbol{n}_{ij} = \frac{1}{2} \sum_{K \ni i,j} \boldsymbol{n}_{ij}^K \,, \quad |C_{ij}| = \sum_{K \ni i,j} |C_{ij}^K|$$

whith $|C_{ij}^K| = \frac{|K|}{6}$. We will evolve approximations of solution averages over the standard median dual cells that we will denote as u_i (u_i in the scalar case)

$$\mathsf{u}_i(t) = \frac{1}{|C_i|} \int_{C_i} \mathsf{u}(\boldsymbol{x}, t)\, d\boldsymbol{x}$$

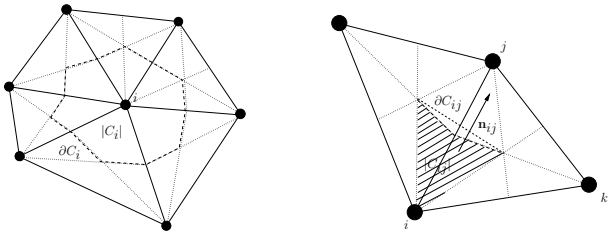

Figure 1: Left: Computational stencil for FV method. Right: cell and interface normals

For a Residual Distribution method, $\{\varphi_i\}_{i \in \mathcal{T}}$ is the standard P^1 continuous piecewise linear Lagrange interpolation kernel. Given an approximation of nodal values $\mathsf{u}_i(t) = \mathsf{u}(\boldsymbol{x}_i, t)$, we introduce the following continuous numerical approximation

$$\mathsf{u}_h(\boldsymbol{x}, t) = \sum_{i \in \mathcal{T}_h} \varphi_i(\boldsymbol{x}) \mathsf{u}_i(t)$$

The method evolves values of the unknowns at mesh nodes which, for simplicity, we shall still denote by the vector u_i, keeping the same notation of FV.

Note that, in the ALE case, the unknowns have both explicit and implicit dependence on time $\mathsf{u}_i(t) = \mathsf{u}(\boldsymbol{x}_i(t), t)$. A discrete evaluation, for example in t^n, will be denoted by $\mathsf{u}_i^n = \mathsf{u}(\boldsymbol{x}_i(t^n), t^n)$. Also geometrical quantities and physical data, will change in time according to the transformation. Keeping the same notation we have $C_i^n = C_i(t^n)$, $\partial C_i^n = \partial C_i(t^n)$ and $b^n = b(\boldsymbol{x}(t^n))$.

3.3 Combining DGCL, C-property and mass conservation

Consider for the moment the simple scalar model (7). Using the relations recalled in the previous section, we can derive the differential ALE formulation

$$\left.\frac{\partial(J_M u)}{\partial t}\right|_X + J_M \nabla \cdot \big(\boldsymbol{f}(u) - u\boldsymbol{\sigma}\big) + J_M S = 0 \tag{15}$$

or equivalently its integral form on a control volume $C(t)$ whose boundary moves with the points of the domain

$$\left.\frac{\partial}{\partial t}\right|_X \int_{C(t)} u \, d\boldsymbol{x} + \int_{C(t)} \nabla \cdot (\boldsymbol{f} - u\boldsymbol{\sigma}) \, d\boldsymbol{x} + \int_{C(t)} S \, d\boldsymbol{x} = 0$$

A numerical method approximating solutions of (15) on a moving mesh is said to verify a DGCL if for $S = 0$, the state $u = u_0 = const$ is an exact solution of the discrete equations. In other words, a numerical method verifies the DGCL if it also embeds an exact dicretization of the GCL (11) (or (12)) allowing, for $S = 0$, constant states to remain constant independently on an externally imposed mesh movement.

However, for a balance law S may be different from 0, and the relevant state to be preserved may not be $u = const$ but $\eta = const$ as seen before. Note in particular, that we may write (15) as

$$J_M \underbrace{\left(\left.\frac{\partial u}{\partial t}\right|_X - \boldsymbol{\sigma} \cdot \nabla u\right)}_{H_1} + u \underbrace{\left(\left.\frac{\partial J_M}{\partial t}\right|_X - \nabla \cdot \boldsymbol{\sigma}\right)}_{H_2} + J_M \underbrace{\Big(\nabla \cdot \boldsymbol{f} + S\Big)}_{H_3} = 0$$

Existing Eulerian discretization methods do embed integral (or even local) variants of $H_3 = 0$ (C-property) and ALE discretization can provide exact integral variants of $H_2 = 0$ (DGCL). However, unfortunately Eulerian methods are unable to embed exact integral (or local) forms of the advection equation H_1. So, in correspondence of steady equilibria, these methods will always have a truncation associated to the term H_1. On the other hand, at the continuous level we ca use (13) to deduce that

$$H_1 + \underbrace{\left.\frac{\partial b}{\partial t}\right|_X - \boldsymbol{\sigma} \cdot \nabla b}_{H_4} = \left.\frac{\partial \eta}{\partial t}\right|_X - \boldsymbol{\sigma} \cdot \nabla \eta$$

which is of course null when η is the invariant associated to the equilibrium $H_3 = 0$.

This suggests that, a better form of (7) for computations on moving meshes, is that obtained by summing Eq. (14) to Eq. (15). This leads to a Well-Balanced ALE form of the problem reading

$$\left.\frac{\partial(J_M \eta)}{\partial t}\right|_X + J_M \nabla \cdot \big(\boldsymbol{f}(u) - \eta\boldsymbol{\sigma}\big) + J_M S = 0 \tag{16}$$

In this case one can do much better job in the approximation of the lake at rest solution. In particular, we can write (16) as

$$J_M \underbrace{\left(\left.\frac{\partial \eta}{\partial t}\right|_X - \boldsymbol{\sigma} \cdot \nabla \eta\right)}_{H_1 + H_4} + \eta \underbrace{\left(\left.\frac{\partial J_M}{\partial t}\right|_X - \nabla \cdot \boldsymbol{\sigma}\right)}_{H_2} + J_M \underbrace{\Big(\nabla \cdot \boldsymbol{f} + S\Big)}_{H_3} = 0$$

If η is constant any Eulerian method will be able to embed the condition $H_1 + H_4 = 0$ while, choosing appropriate schemes verifying both the DGCL and the C-property, we will be able to satisfy all the compatibility requirements, and preserve steady equilibria independently on the mesh movement strategy.

For the shallow Water equations, one can proceed in a very similar fashion. A straightforward ALE formulation of (1) leads to the system

$$\left.\frac{\partial(J_M \mathsf{u})}{\partial t}\right|_X + J_M \nabla \cdot \left(\mathbf{f}(\mathsf{u}) - \mathsf{u}\boldsymbol{\sigma}\right) + J_M \mathsf{S} = 0$$

which is not well suited to preserve the lake at rest equilibrium (6). Proceeding as before, we may add to the continuity equation expression (14) to obtain the well-balanced form

$$\left.\frac{\partial(J_M \mathsf{w})}{\partial t}\right|_X + J_M \nabla \cdot \left(\mathbf{f}(\mathsf{u}) - \mathsf{w}\boldsymbol{\sigma}\right) + J_M \mathsf{S} = 0 \tag{17}$$

with u and $\mathbf{f}$ still given by (2), and with

$$\mathsf{w} = \begin{bmatrix} \eta \\ hu \\ hv \end{bmatrix}.$$

As we will see, this form easily allows to preserve exactly the steady state (6). As a particular case and for completeness, we recall that the pre-balanced form of the Shallow Water equations of [7] is obtained by introducing the modified flux and source functions

$$\tilde{\mathbf{f}}(\mathsf{w}; b) = \begin{bmatrix} hu & hv \\ hu^2 + \frac{1}{2} g \left(\eta^2 - 2\eta b\right) & huv \\ huv & hv^2 + \frac{1}{2} g \left(\eta^2 - 2\eta b\right) \end{bmatrix}, \quad \tilde{\mathsf{S}} = \begin{bmatrix} 0 \\ g\eta \dfrac{\partial b}{\partial x} \\ g\eta \dfrac{\partial b}{\partial y} \end{bmatrix} \tag{18}$$

These definitions lead to pre-balanced ALE form of the shallow-water system reading

$$\left.\frac{\partial(J_M \mathsf{w})}{\partial t}\right|_X + J_M \nabla \cdot \left(\tilde{\mathbf{f}}(\mathsf{w}; b) - \mathsf{w}\boldsymbol{\sigma}\right) + J_M \tilde{\mathsf{S}} = 0 \tag{19}$$

We recall that the this form satisfies the relation, (see [7])

$$\frac{\partial \tilde{\mathbf{f}}(\mathsf{w}; b)}{\partial \mathsf{w}} = \frac{\partial \mathbf{f}(\mathsf{u})}{\partial \mathsf{u}}$$

so the pre-balanced system has the same eigen-structure of the standard one.

3.3.1 Mass conservation

We now consider the additional constraint of conserving the total water mass in the domain. We integrate the mass conservation equation in well-balanced form in space and in time

$$\int_{\Omega(t)} \eta(\boldsymbol{x}(t), t)\, d\boldsymbol{x} - \int_{\Omega_X} \eta(\boldsymbol{X}, 0)\, d\boldsymbol{x} + \int \int_{\partial\Omega(t)} (h\boldsymbol{u} - \eta\boldsymbol{\sigma}) \cdot \boldsymbol{n}\, ds\, dt = 0$$

Let $H(t)$ be the total mass of water at time t, $H(t) = \int_{\Omega(t)} h\, d\boldsymbol{x}$, and define $B(t) = \int_{\Omega(t)} b\, d\boldsymbol{x}$, we can rewrite mass conservation statement separating the terms in h and the terms in b

$$H(t) - H(0) + \int \int_{\partial\Omega(t)} (h\boldsymbol{u} - h\boldsymbol{\sigma}) \cdot \boldsymbol{n}\, ds dt + B(t) - B(0) - \int \int_{\partial\Omega(t)} b\boldsymbol{\sigma} \cdot \boldsymbol{n}\, ds\, dt = 0 \tag{20}$$

which states that, modulo the boundary conditions, we have conservation over the full domain if the ALE remap equation (14) is satisfied, namely if

$$B(t) - B(0) - \int \int_{\partial\Omega(t)} b\boldsymbol{\sigma} \cdot \boldsymbol{n}\, ds\, dt = 0$$

So a scheme approximating (17) will be exactly mass conservative only if the bathymetry is evolved according to an integral form of the ALE remap (14). This is the strategy proposed in [28]. However, as pointed out in the same paper, this approach leads changes in the bathymetric altitudes which will depend on the scheme. For example, substantial smoothing of the bed slopes has been observed. To deal with this issues, in [28] the authors propose to regularly re-initialize the bathymetric data. This will however violate (14), and so a mass loss will be associated to each of these re-initialization steps. Here we propose an alternative solution, allowing to preserve mass down to almost machine accuracy. Assume for simplicity that the domain boundaries are not moving, or that $\boldsymbol{\sigma} \cdot \boldsymbol{n}$ is verified. We can write the mass error at time t as

$$E_{mass} = H(t) - H(0) + \int \int_{\partial\Omega} h\boldsymbol{u} \cdot \boldsymbol{n} \, ds \, dt = B(0) - B(t)$$

We now remark that the two quantities on the right hand side are in principle equal, as they are both approximations of the integral of $b(\boldsymbol{x})$ over the domain. If the domain boundaries are not moving, this quantity should remain constant in time. In practice however, these two integrals will be evaluated on a moving mesh. This means that, even if both the domain of integration and the data being integrated are constant, the quadrature points used will move, so the result will not be the same. To be more precise, with the numerical approximation, $B(t)$ will be splitted in integrals over the set of median dual cell areas. In case of standard FV or P^1 RD discretizations

$$B(t) = \sum_{i \in \mathcal{T}_h} \int_{C_i(t)} b \, d\boldsymbol{x} = \sum_{i \in \mathcal{T}_h} b_i(t) |C_i(t)| \tag{21}$$

with $b_i = b(\boldsymbol{x}_i(t))$. Our idea is to compute different nodal values $b_i \neq b(\boldsymbol{x}_i(t))$ such that the total mass error is reduced by simply increasing the accuracy which the elemental integrals are evaluated with. So we will set

$$\int_{C_i(t)} b(\boldsymbol{x}(t)) \, d\boldsymbol{x} \approx |C_i(t)| \sum_{f=1}^{N_q} \omega_q b(\boldsymbol{x}_q(t))$$

and we we will compute bathymetric nodal values

$$b_i = \sum_{f=1}^{N_q} \omega_q b(\boldsymbol{x}_q(t)) \tag{22}$$

here it is essential to underline that $b(\boldsymbol{x}_q)$, on the right hand side, is a given high accurate (analytical or reference one, interpolated on a fine mesh) representation of the bathymetry.

The above analysis is correct if the entire domain is wet. In presence of wetting drying, there is a major complication related to the fact that the volume containing water mass is moving, and its movement is a-priori independent on the mapping defined by $\boldsymbol{\sigma}$. Of course, in this case the flow equations are only solved in the wet region $\Omega_W(t)$. So mass conservation now reads

$$\int_{\Omega_W(t)} \eta(\boldsymbol{x}(t), t) \, d\boldsymbol{x} - \int_{\Omega_{W,X}} \eta(\boldsymbol{X}, 0) \, d\boldsymbol{x} + \int \int_{\partial\Omega_W(t)} (h\boldsymbol{u} - \eta\boldsymbol{\sigma}) \cdot \boldsymbol{n} \, ds \, dt = 0$$

Water depth and discharge are both null at the shoreline ∂I_W while the ALE flux is null at the domain boundaries. So we choose to define two boundary regions $\partial\Omega_W = \partial\Omega \cap \partial\Omega_W + \partial I_W$ and write

$$H(t) - H(0) + \int \int_{\partial\Omega \cap \partial\Omega_W} h\boldsymbol{u} \cdot \boldsymbol{n} \, ds \, dt + B_W(t) - B_W(0) - \int \int_{\partial I_W(t)} b\boldsymbol{\sigma} \cdot \boldsymbol{n} \, ds \, dt = 0$$

The mass error, due to the deformation of the computational domain, becomes

$$E_{mass} = -\left(B_W(t) - B_W(0) - \int \int_{\partial I_W(t)} b\boldsymbol{\sigma} \cdot \boldsymbol{n} \, ds \, dt \right)$$

As before, this quantity is not zero, as we do not use the ALE remap to evolve the (given) bathymetry. To link this residual error to the previous case, we add and remove the following quantity:

$$Q = B_D(t) - B_D(0) - \int \int_{\partial I_D(t)} b\boldsymbol{\sigma} \cdot \boldsymbol{n}\, ds\, dt$$

the sub-script $\cdot_D$ denoting integrals over the dry area. We finally obtain

$$E_{mass} = B(0) - B(t) + Q$$

The difference between the first two terms can be reduced as discussed before. The reminder Q is a geometrical term associated to the deformation in dry areas. Unfortunately, we are not able to guarantee any a-priori control on this term, since, as we will see later, grid adaptation w.r.t. the shoreline benefits from the possibility of exploiting points in the dry region. In this paper, this geometrical factor arising from deformation in dry areas will be accounted for by uniformly redistributing the mass excess/defect in the wet region,

3.4 Finite Volume discrete approximation

We consider the standard well balanced node-centered Finite Volume (FV) scheme based on Roe's linearized Riemann solver. see [1, 2, 31] and references therein for details. From the final form of the discretization based on the well balanced ALE form of the equations, we will show the equivalence between the scheme obtained using the well-balanced form (17), and the pre-balanced formulation (19) obtained with definitions (18).

The FV discrete evolution equations read

$$\begin{aligned} |C_i^{n+1}|\mathsf{w}_i^* =& |C_i^n|\mathsf{w}_i^n - \Delta t\, R_i(\mathsf{w}^n;\, b^n) \\ |C_i^{n+1}|\mathsf{w}_i^{n+1} =& |C_i^n|\mathsf{w}_i^n - \frac{\Delta t}{2} R_i(\mathsf{w}^n;\, b^n) - \frac{\Delta t}{2} R_i(\mathsf{w}^*;\, b^{n+1}) \end{aligned} \tag{23}$$

where we have

$$R_i(\mathsf{w};\, b) = \sum_{j \in \mathcal{D}_i} (\mathcal{F}_{ij} + \mathcal{S}_{ij}) \tag{24}$$

$\mathcal{F}_{ij}$ is a consistent numerical approximation of the flux along ∂C_{ij}, while $\mathcal{S}_{ij}$ is an approximation of the integral of the source term on C_{ij}. In this work we use the Roe-type numerical flux which, on moving meshes and for the well balanced formulations (16) and (17), reads

$$\mathcal{F}_{ij} = \mathcal{F}_{ij}(\breve{\mathsf{u}}_i,\, \breve{\mathsf{u}}_j;\, \breve{b}_i,\, \breve{b}_j) = \frac{\mathbf{f}(\breve{\mathsf{u}}_j) + \mathbf{f}(\breve{\mathsf{u}}_i)}{2} \cdot \boldsymbol{n}_{ij} - \upsilon_{ij} \frac{\breve{\mathsf{w}}_j + \breve{\mathsf{w}}_i}{2} - \frac{\left|\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3\right|}{2} (\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i) \tag{25}$$

with I_3 the 3×3 identity matrix, where, the $\breve{\cdot}$ values denote linearly reconstructed values of a quantity, while, using the notation of (3), $\mathsf{K}_{ij} = \mathsf{K}(\boldsymbol{n}_{ij}, \mathsf{u}_{ij}^*)$ is the flux Jacobian evaluated using a Roe linearization u_{ij}^*. The absolute value of a matrix is computed through eigenvalues decomposition $|\mathsf{A}| = \mathsf{R}|\Lambda|\mathsf{R}^{-1}$. Note that the same reconstruction is used for the components of u and for b so that a constant state $\eta = \eta_0$ is exactly approximated (see e.g. [32, 31, 12]). From (25), the scalar expressions are obtained by replacing $\mathbf{f}(\mathsf{u})$ by $\boldsymbol{f}(u)$, w by η, and K_{ij} by $\boldsymbol{a}\cdot\boldsymbol{n}_{ij}$. Concerning the ALE related aspects, following the closure proposed in [27], all the geometrical quantities needed to evaluate R_i are obtained on the mid-point averaged mesh $\mathcal{T}_h^{n+1/2} = (\mathcal{T}_h(t^n) + \mathcal{T}_h(t^{n+1}))/2$, while the projected interface velocities υ_{ij} are defined as

$$\upsilon_{ij} = \int_{\partial C_{ij}^{n+1/2}} \boldsymbol{\sigma}_h \cdot \boldsymbol{n}\, ds \tag{26}$$

with $\boldsymbol{\sigma}_h$ a piecewise linear continuous interpolation of the nodal discrete velocities

$$\boldsymbol{\sigma}_h = \sum_{i \in \mathcal{T}} \varphi_i \frac{\boldsymbol{x}_i^{n+1} - \boldsymbol{x}_i^n}{\Delta t} \tag{27}$$

Simple algebraic manipulations show that this definition satisfies the integral DGCL (see [33] for details)

$$|C_i^{n+1}| - |C_i^n| = \Delta t \sum_{j \in \mathcal{D}_i} v_{ij} = \Delta t \int\limits_{C_i^{n+1/2}} \nabla \cdot \boldsymbol{\sigma}_h \tag{28}$$

It is interesting to note the full analogy with the *interface velocity consistency condition* proposed in [34] where the DGCL closure is achieved within the approach of Wang, see [26].

Concerning the topography source term, following [2, 31], we distinguish two contributions, the first balancing the central part of the fluxes, and the second the upwind dissipation term :

$$\mathcal{S}_{ij} = \mathcal{S}_{ij}^c + \mathcal{S}_{ij}^* \tag{29}$$

Introducing the average values

$$u_{ij}^- = \frac{\breve{u}_i + u_i}{2}, \quad u_{ij} = \frac{\breve{u}_j + \breve{u}_i}{2}$$

and the projected Jacobians $a_{ij}^- = \boldsymbol{a}(u_{ij}^-) \cdot \boldsymbol{n}_{ij}$ and $a_{ij} = \boldsymbol{a}(u_{ij}) \cdot \boldsymbol{n}_{ij}$, in the scalar case, we have

$$\mathcal{S}_{ij}^c = a_{ij}^-(\breve{b}_i - b_i) + \frac{1}{2} a_{ij}(\breve{b}_j - \breve{b}_i) \tag{30}$$

For the Shallow Water equations, we can proceed analogously and introduce the averages

$$h_{ij}^- = \frac{\breve{h}_i + h_i}{2}, \quad h_{ij} = \frac{\breve{h}_j + \breve{h}_i}{2}$$

the bathymetry variation vectors

$$\Delta b_{ij}^- = \begin{pmatrix} \breve{b}_i - b_i \\ 0 \\ 0 \end{pmatrix}, \quad \Delta b_{ij} = \begin{pmatrix} \breve{b}_j - \breve{b}_i \\ 0 \\ 0 \end{pmatrix}$$

and the Jacobians at rest $\mathsf{A}_{ij}^- = \mathsf{A}(\boldsymbol{n}_{ij}, h_{ij}^-)$, and $\mathsf{A}_{ij} = \mathsf{A}(\boldsymbol{n}_{ij}, h_{ij})$ (cf. (5) for the notation). The centered component of the source can be now written as

$$\mathcal{S}_{ij}^c = \mathsf{A}_{ij}^- \Delta b_{ij}^- + \frac{1}{2} \mathsf{A}_{ij} \Delta b_{ij} \tag{31}$$

Concerning the upwind balancing term, the original definition given in [1, 2] leads to the following expression for the Shallow Water system

$$\mathcal{S}_{ij}^* = -\frac{\operatorname{sign}(\mathsf{K}_{ij} - v_{ij}\mathsf{I}_3)}{2} (\mathsf{A}_{ij} - v_{ij}\mathsf{I}_3) \Delta b_{ij} \tag{32}$$

with the matrix sign computed by standard eigenvalue decomposition. For (16) we get the simpler expression (cf. equation (30))

$$\mathcal{S}_{ij}^* = -\frac{|a_{ij} - v_{ij}|}{2} (\breve{b}_j - \breve{b}_i) \tag{33}$$

The bathymetric values b_i^{n+1} (in the previous paragraph time dependency has been dropped for clarity) are computed as explained in the previous section, cf. eq. (22). For the FV method the nodal values are computed using gaussian quadrature formula over each sub-triangle C_{ij}^K

$$b_i^{n+1} = \frac{1}{|C_i^{n+1}|} \sum_{j \in \mathcal{D}_i} \sum_{K \ni i,j} \frac{|K^{n+1}|}{6} \sum_{q=1}^{N_q} \omega_q b_q^{n+1}$$

given $b_q = b(\boldsymbol{x}_q)$ with $x_q = \sum_{j \in K} \boldsymbol{\lambda}_j^q x_j$ and $y_q = \sum_{j \in K} \boldsymbol{\lambda}_j^q y_j$. The baricentric coordinates of the quadrature points $\boldsymbol{\lambda}_j^q$ are defined over the sub-triangles C_{ij}^K. The one point quadrature with baricentric coordinate

in i corresponds to a constant approximation of the bathymetry function over the median dual cell (zero order, $r = 0$) and coincides with the standard choice $b_i^{n+1} = b(\boldsymbol{x}_i^{n+1})$. In the numerical experiments we will test the impact of first and second order accurate formulas (denoted respectively $r = 1, 2$), in order to arbitrarly decrease the mass error.

With these definitions we have now the following characterization.

Proposition 1. *The finite volume discrete equations* (23)-(24) *with definitions* (25), (26), (29), (31) *(or* (30) *in the scalar case), and* (32) *(or* (33) *in the scalar case) verify the DGCL for constant b, and the C-property both on moving and fixed meshes, provided that the same reconstruction procedure is used for* u *and b.*

Proof. For constant b, and constant u_0, the discrete equations reduce to (28), which proves the first part (DGCL)

$$|C_i^{n+1}|\mathsf{u}_i^{n+1} - |C_i^n|\mathsf{u}_0 = \Delta t \sum_{j\in\mathcal{D}_i} \upsilon_{ij}\mathsf{u}_0 \quad \Rightarrow \quad \mathsf{u}_i^{n+1} = \mathsf{u}_0$$

For the second part, we need to prove that the steady equilibrium η_0 is preserved. In the scalar case we have for $\eta = \eta_0$ constant, and under the hypotheses made

$$R_i = \sum_{j\in\mathcal{D}_i}(F_{ij} + \mathcal{S}_{ij}) = \sum_{j\in\mathcal{D}_i} \frac{\boldsymbol{f}(\breve{u}_j) - \boldsymbol{f}(\breve{u}_i)}{2}\cdot\boldsymbol{n}_{ij} + \sum_{j\in\mathcal{D}_i}(\boldsymbol{f}(\breve{u}_i) - \boldsymbol{f}(u_i))\cdot\boldsymbol{n}_{ij} - \sum_{j\in\mathcal{D}_i}\upsilon_{ij}\eta_0 + \sum_{j\in\mathcal{D}_i} a_{ij}^-(\breve{b}_i - b_i) + \sum_{j\in\mathcal{D}_i}\frac{1}{2}a_{ij}(\breve{b}_j - \breve{b}_i)$$

having already neglected the upwind terms

$$-\sum_{j\in\mathcal{D}_i}\frac{|a_{ij} - \upsilon_{ij}|}{2}(\breve{\eta}_j - \breve{\eta}_i)$$

which vanish identically. We then have used the properties of the Roe average

$$\begin{aligned} R_i &= \sum_{j\in\mathcal{D}_i}\frac{1}{2}a_{ij}(\breve{u}_j - \breve{u}_i) + \sum_{j\in\mathcal{D}_i} a_{ij}^-(\breve{u}_i - u_i) - \sum_{j\in\mathcal{D}_i}\upsilon_{ij}\eta_0 + \sum_{j\in\mathcal{D}_i} a_{ij}^-(\breve{b}_i - b_i) + \sum_{j\in\mathcal{D}_i}\frac{1}{2}a_{ij}(\breve{b}_j - \breve{b}_i) \\ &= \sum_{j\in\mathcal{D}_i}\frac{1}{2}a_{ij}(\breve{\eta}_j - \breve{\eta}_i) + \sum_{j\in\mathcal{D}_i} a_{ij}^-(\breve{\eta}_j - \eta_i) - \sum_{j\in\mathcal{D}_i}\upsilon_{ij}\eta_0 \end{aligned}$$

proving the second part (C-property) in the scalar case:

$$|C_i^{n+1}|\eta_i^{n+1} - |C_i^n|\eta_0 = \Delta t \sum_{j\in\mathcal{D}_i}\upsilon_{ij}\eta_0 \quad \Rightarrow \quad \eta_i^{n+1} = \eta_0$$

For the Shallow Water equations the proof rests on the fact that, on the lake at rest state, we have $\mathsf{K}_{ij} = \mathsf{A}_{ij}$. In particular, proceeding as before we have using the last observation

$$\begin{aligned} R_i = &\sum_{j\in\mathcal{D}_i}\frac{\mathbf{f}(\breve{\mathsf{u}}_j) - \mathbf{f}(\breve{\mathsf{u}}_i)}{2}\cdot\boldsymbol{n}_{ij} + \sum_{j\in\mathcal{D}_i}(\mathbf{f}(\breve{\mathsf{u}}_i) - \mathbf{f}(\mathsf{u}_i))\cdot\boldsymbol{n}_{ij} - \sum_j \upsilon_{ij}\mathsf{w}_0 \\ &+ \sum_{j\in\mathcal{D}_i}\mathsf{A}_{ij}^-\,\Delta b_{ij}^- + \frac{1}{2}\sum_{j\in\mathcal{D}_i}\mathsf{A}_{ij}\,\Delta b_{ij} - \frac{|\mathsf{A}_{ij} - \upsilon_{ij}\mathsf{I}_3|}{2}(\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i) - \frac{\operatorname{sign}(\mathsf{A}_{ij} - \upsilon_{ij}\mathsf{I}_3)}{2}(\mathsf{A}_{ij} - \upsilon_{ij}\mathsf{I}_3)\Delta b_{ij} \end{aligned}$$

Note now that $\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i + \Delta b_{ij} = \breve{\mathsf{w}}_j - \breve{\mathsf{w}}_i$ which vanishes by hypothesis, so that the last two terms cancel each other. The rest of the proof is almost identical to the scalar case, and uses the fact that, on the selected equilibrium, $(\mathbf{f}(\breve{\mathsf{u}}_j) - \mathbf{f}(\breve{\mathsf{u}}_i))\cdot\boldsymbol{n}_{ij} = \mathsf{A}_{ij}(\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i)$ and the constancy of $\mathsf{w} = \mathsf{w}_0$.

□

Before moving on, it is interesting to note that the use of the FV discrete equations obtained by using the pre-balanced form of the shallow equations (19) are almost identical to those presented above which are instead derived from (17). In particular, we have the following equivalence.

Proposition 2. *The pre-balanced upwind FV discretization obtained from the pre-balanced form of the Shallow Water equation* (19) *with Roe's numerical fluxes and a non-upwind source term approximation is equivalent to the scheme given by* (23)-(24) *with definitions* (25), (26), (29), (31) *and* (32), *setting in* (29)

$$\mathcal{S}^*_{ij} = -\frac{\text{sign}(\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3)}{2}(\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3)\Delta b_{ij} \tag{34}$$

Proof. The well balanced FV spatial discretization obtained from (17) wich incorporates (34) reads:

$$\begin{aligned} R_i = &\sum_{j\in\mathcal{D}_i} \frac{\mathbf{f}(\breve{\mathsf{u}}_j) - \mathbf{f}(\breve{\mathsf{u}}_i)}{2}\cdot\boldsymbol{n}_{ij} + \sum_{j\in\mathcal{D}_i}(\mathbf{f}(\breve{\mathsf{u}}_i) - \mathbf{f}(\mathsf{u}_i))\cdot\boldsymbol{n}_{ij} - \sum_{j\in\mathcal{D}_i}\upsilon_{ij}\frac{\breve{\mathsf{w}}_j + \breve{\mathsf{w}}_i}{2} \\ &+ \sum_{j\in\mathcal{D}_i}\mathsf{A}^-_{ij}\,\Delta b^-_{ij} + \frac{1}{2}\sum_{j\in\mathcal{D}_i}\mathsf{A}_{ij}\,\Delta b_{ij} - \frac{|\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3|}{2}(\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i) - \frac{\text{sign}(\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3)}{2}(\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3)\Delta b_{ij} \end{aligned} \tag{35}$$

The equivalence $\nabla\cdot\mathbf{f}(\mathsf{u}) + gh\nabla b = \nabla\cdot\tilde{\mathbf{f}}(\mathsf{w}; b) + g\eta\nabla b$ is written here at discrete level (cf. equation (18) for the notation)

$$(\mathbf{f}(\breve{\mathsf{u}}_j) - \mathbf{f}(\breve{\mathsf{u}}_i))\cdot\boldsymbol{n}_{ij} + \mathsf{A}_{ij}\Delta b_{ij} = \tilde{\mathbf{f}}(\breve{\mathsf{w}}_j;\, \breve{b}_j) - \tilde{\mathbf{f}}(\breve{\mathsf{w}}_i;\, \breve{b}_i) + \tilde{\mathsf{A}_{ij}}\Delta b_{ij} \tag{36}$$

with (cf. equation (5)) $\tilde{\mathsf{A}}_{ij} = \mathsf{A}(\boldsymbol{n}_{ij},\, \eta_{ij})$ where, in analogy with the notation used so far, $\eta_{ij} = (\breve{\eta}_j + \breve{\eta}_i)/2$. Similarly, we can also define $\tilde{\mathsf{A}}^-_{ij} = \mathsf{A}(\boldsymbol{n}_{ij},\, \eta^-_{ij})$ with $\eta^-_{ij} = (\breve{\eta}_i + \eta_i)/2$

$$(\mathbf{f}(\breve{\mathsf{u}}_i) - \mathbf{f}(\mathsf{u}_i))\cdot\boldsymbol{n}_{ij} + A^-_{ij}\Delta b^-_{ij} = \tilde{\mathbf{f}}(\breve{\mathsf{w}}_i;\, \breve{b}_i) - \tilde{\mathbf{f}}(\mathsf{w}_i;\, b_i) + \tilde{A}^-_{ij}\Delta b^-_{ij} \tag{37}$$

We now use the fact that definition (34) of $\mathcal{S}^*_{ij}$ is such that when added to dissipation of the numerical flux one gets

$$-\frac{|\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3|}{2}(\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i) + \mathcal{S}^*_{ij} = -\frac{|\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3|}{2}(\breve{\mathsf{w}}_j - \breve{\mathsf{w}}_i) \tag{38}$$

Finally, substituting (36),(37) and (38) in the well balanced FV spatial discretization (35) we have

$$R_i = \sum_{j\in\mathcal{D}_i}(\tilde{\mathcal{F}}_{ij} + \tilde{\mathcal{S}}^c_{ij}) \tag{39}$$

with (cf. equation (18) for the notation)

$$\tilde{\mathcal{F}}_{ij} = \frac{\tilde{\mathbf{f}}(\breve{\mathsf{w}}_j;\, \breve{b}_j) + \tilde{\mathbf{f}}(\breve{\mathsf{w}}_i;\, \breve{b}_i)}{2}\cdot\boldsymbol{n}_{ij} - \upsilon_{ij}\frac{\breve{\mathsf{w}}_j + \breve{\mathsf{w}}_i}{2} - \frac{|\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3|}{2}(\breve{\mathsf{w}}_j - \breve{\mathsf{w}}_i)$$

and

$$\tilde{\mathcal{S}}^c_{ij} = \tilde{\mathsf{A}}^-_{ij}\Delta b^-_{ij} + \frac{1}{2}\tilde{\mathsf{A}}_{ij}\Delta b_{ij}$$

which is exactly the pre-balanced FV discretization otained from (19) (cf. [7, 8, 9]).

□

The last proposition shows that the well balanced discretization of [1, 2] is equivalent to the use of the pre-balanced form of the equation for a particular choice of the upwind component of the source. The proposition also shows that another viable alternative would be for example

$$\mathcal{S}^*_{ij} = -\frac{|\mathsf{A}_{ij} - \upsilon_{ij}\mathsf{I}_3|}{2}\Delta b_{ij} \tag{40}$$

which also leads to a well-balanced discretization (cf. proof of proposition 1). In our implementation, we have used (34). We have combined this with a Green-Gauss reconstruction [31, 35, 36] to achieve second order (scheme referred to as FROMM scheme in the results section). If necessary, monotonicity is enforced using the van Albada limiter (resulting scheme referred to as MUSCL scheme), while for the treatment of the wet dry interfaces, including the semi-implicit treatment of friction, we have followed [37, 38, 39, 35].

3.5 Residual Distribution discrete approximation

We compare the finite volume results to those of the two-step explicit Residual Distribution (RD) method developed in [40, 33, 14]. On a moving mesh, the evolution equation obtained with the RD method is derived from the weak form of the well balanced equation (17). After some manipulations the RD method can be written in the compact prototype form

$$|C_i^{n+1}|\mathsf{w}_i^{n+1} = |C_i^{n+1}|\mathsf{w}_i^* - \Delta t \sum_{K\in\mathcal{D}_i} \Phi_i^K(\mathsf{u}_h^n, \mathsf{u}_h^*;\, b_h^n, b_h^{n+1}) \tag{41}$$

where the fluctuations Φ_i^K define a splitting of the average element residual Φ^K, namely

$$\sum_{j\in K} \Phi_j^K = \Phi^K = \frac{1}{\Delta t}\Big(\int_{K^{n+1}} \mathsf{w}_h^* - \int_{K^n} \mathsf{w}_h^n \Big) + \frac{1}{2}\boldsymbol{\Phi}^K(\mathsf{u}_h^n;\, b_h^n) + \frac{1}{2}\boldsymbol{\Phi}^K(\mathsf{u}_h^*;\, b_h^{n+1}) \tag{42}$$

with (cf. equations (17))

$$\sum_{j\in K} \boldsymbol{\Phi}_j^K = \boldsymbol{\Phi}^K = \int_{\partial K^{n+1/2}} \big(\mathbf{f}(\mathsf{u}_h) - \boldsymbol{\sigma}_h \mathsf{w}_h\big)\cdot \boldsymbol{n}\, ds + \int_{K^{n+1/2}} \mathsf{S}(\mathsf{u}_h,\, b_h)\, d\boldsymbol{x} \tag{43}$$

The values w_i^* are obtained from a first order predictor which is computed as

$$|C_i^{n+1}|\mathsf{w}_i^* = |C_i^{n+1}|\mathsf{w}_i^n - \Delta t \sum_{K\in\mathcal{D}_i} \tilde{\boldsymbol{\Phi}}_i^K(\mathsf{u}_h^n;\, b_h^n) \tag{44}$$

where now the fluctuations $\tilde{\boldsymbol{\Phi}}_i^K$ are a splitting of the following *geometrically non-conservative* average steady cell residual, see [33]

$$\sum_{j\in K} \tilde{\boldsymbol{\Phi}}_j^K = \tilde{\boldsymbol{\Phi}}^K = \int_{\partial K^{n+1/2}} \mathbf{f}(\mathsf{u}_h)\cdot \boldsymbol{n}\, ds - \int_{K^{n+1/2}} \boldsymbol{\sigma}_h \cdot \nabla \mathsf{w}_h\, d\boldsymbol{x} + \int_{K^{n+1/2}} \mathsf{S}(\mathsf{u}_h,\, b_h)\, d\boldsymbol{x} \tag{45}$$

Note that, as for the FV scheme and as prescribed in the ALE formulation proposed in [33], most geometrical quantities necessary for the computation of the residuals (42) and (45) are evaluated on the mid-point averaged mesh $\mathcal{T}_h^{n+1/2}$ which ensure

$$|K^{n+1}| - |K^n| = \Delta t \int_{\partial K^{n+1/2}} \boldsymbol{\sigma}_h \cdot \boldsymbol{n}\, ds = 0 \tag{46}$$

In particular, all boundary integrals are evaluated by means of a 2 points Gauss-Legendre formula, while volume integrals are evaluated *exactly* w.r.t. the assumed linear variation of the quantities involved.

For the volume integral associated with the bathymetry, we used the following L^2 type projection

$$b_i^{n+1} = \frac{1}{|C_i^{n+1}|} \sum_{K\in\mathcal{D}_i} |K^{n+1}| \sum_{q=1}^{N_q} \omega_q b_q^{n+1} \varphi_q$$

The three points quadrature with baricentric coordinates in the triangle's vertex corresponds to a piecewice linear approximation of the bathymetry function over the triangles (first order, $r = 1$) and coincides with the standard choice $b_i^{n+1} = b(\boldsymbol{x}_i^{n+1})$. In the numerical experiments we will test the impact of second and third order accurate formulas (denoted respectively $r = 2, 3$).

The key properties of the method introduced above are determined by the definition of the split residuals Φ_i^K, $\boldsymbol{\Phi}_i^K$ and $\tilde{\boldsymbol{\Phi}}_i^K$. Following [40, 33, 14], we assume for these quantities the following form :

$$\begin{aligned}
\tilde{\boldsymbol{\Phi}}_i^K &= \boldsymbol{\beta}_i \tilde{\boldsymbol{\Phi}}^K(\mathsf{u}_h^n,\, b_h^n) \\
\boldsymbol{\Phi}_i^K &= \boldsymbol{\beta}_i \boldsymbol{\Phi}^K(\mathsf{u}_h^n,\, b_h^n) \\
\Phi_i^K &= \sum_{j\in K} \frac{m_{ij}^{K^{n+1}} \mathsf{w}_j^* - m_{ij}^{K^n} \mathsf{w}_j^n}{\Delta t} + \frac{1}{2}\boldsymbol{\Phi}_i^K(\mathsf{u}_h^n,\, b_h^n) + \frac{1}{2}\boldsymbol{\Phi}_i^K(\mathsf{u}_h^*,\, b_h^{n+1})
\end{aligned} \tag{47}$$

where with $\boldsymbol{\beta}_j$ we have denoted a set of *distribution matrices (resp. distribution coefficients in the scalar case) which are uniformly bounded w.r.t the cell residuals* (45), (42). The mass matrix entries $m_{ij}^K = \int_K \varphi_i \varphi_j d\boldsymbol{x}$ satisfy the identities (see [40] for a discussion of RD mass matrices)

$$\sum_{j \in K} m_{ij}^K = \beta_i \left| K \right|, \quad \sum_{i \in K} m_{ij}^K = \frac{|K|}{3}$$

As shown in [40] the above definitions give a scheme which si formally second order accurate, independently on the definition of the bounded distribution matrices (or coefficients), and of the mass matrices. With these definitions, we can also easily show the following.

Proposition 3. *The explicit predictor corrector residual distribution prototype* (41), (44), (47) *verifies the DGCL for constant b, and the C-property both on moving and fixed meshes, provided that the same linear piecewise continuous approximation is used for* u *and b, and that all integrals involving these quantities are evaluated exactly w.r.t. this variation.*

Proof. To prove the first part, we check that for constant bathymetry (hence for $\mathsf{S} = 0$), the splitting terms on the right hand sides in (41) and (44) are identically zero for a given constant state u_0. This is immediately shown for the predictor step (44) as for a constant state we get trivially $\tilde{\boldsymbol{\Phi}}^K = 0$. For the corrector, one can immediately see that, on a given element, for constant $\mathsf{u} = \mathsf{u}_0$ in (42)

$$\Phi^K = \mathsf{u}_0 \frac{|K^{n+1}| - |K^n|}{\Delta t} - \mathsf{u}_0 \int\limits_{\partial K^{n+1/2}} \boldsymbol{\sigma}_h \cdot \boldsymbol{n} \, ds$$

which also is identically zero as shown in (46) and we have $\mathsf{u}_i^{n+1} = \mathsf{u}_0$.

To prove the second part of the proposition, we proceed in an identical manner, except that now we assume that we have a constant state $\mathsf{w} = \mathsf{w}_0$. As before, one checks that trivially $\tilde{\boldsymbol{\Phi}}^K = 0$. In the corrector step

$$\Phi^K = \mathsf{w}_0 \frac{|K^{n+1}| - |K^n|}{\Delta t} - \mathsf{w}_0 \int\limits_{\partial K^{n+1/2}} \boldsymbol{\sigma}_h \cdot \boldsymbol{n} \, ds + \int\limits_{\partial K^{n+1/2}} \mathbf{f}(\mathsf{u}_h) \cdot \boldsymbol{n} \, ds + \int\limits_{K^{n+1/2}} \mathsf{S}(\mathsf{u}_h, \, b_h) = 0$$

provided that the boundary integral of the hydrostatic term in the flux balances exactly the integral of the source

$$\int_{K^{n+1/2}} g h_h \nabla h_h d\boldsymbol{x} = - \int_{K^{n+1/2}} g h_h \nabla b_h d\boldsymbol{x}$$

This is true under the hypotheses of exact integration w.r.t. the linear variation of depth and bathymetry. Thus $\mathsf{w}_i^{n+1} = \mathsf{w}_0$.

□

Concerning the actual definition of the distribution matrices and mass matrices, here we have followed [14]. In particular, as we will see in the following, three variants of the methods are considered. A simple centered variant, which is obtained by $\beta_i = 1/3$, and with m_{ij}^K the entries of the standard P^1 Galerkin finite element mass matrix. A second order stabilized version of the scheme is obtained by adding a streamline dissipation term, which leads to a genuinely explicit analog of the SUPG scheme of Hughes and co-workers (see [41, 42, 40, 14] and references therein for details). Finally, as in [14], a nonlinear method is obtained by blending the SUPG with a nonlinear distribution obtained by applying a limiter to a Lax-Friedrich's (LxF). The resulting scheme is referred to as the LLxF-SUPG in the following. For further details concerning the definitions of the quantities involved, and for specifics of their applications to the Shallow Water equations, the interested reader may refer to [43, 12, 40, 14] and references therein.

4 Adaptive mesh deformation

We use in this paper the elliptic mesh deformation technique used for time dependent flows also in [44, 17, 45, 29, 25]. Given an error monitor function ω, the mapping $\boldsymbol{x} = M(\boldsymbol{X}, t)$ which equidistributes ω on the reference domain, is computed by solving the following differential problem

$$\nabla_{\boldsymbol{X}} \cdot \boldsymbol{\Sigma} + \boldsymbol{F} = 0 \quad \boldsymbol{X} \in \Omega_X \tag{48}$$

with the subscript $\cdot_{\boldsymbol{X}}$ denoting derivatives w.r.t. the reference coordinates $\boldsymbol{X}$, and where the tensor $\boldsymbol{\Sigma}$ is a function of the displacement w.r.t. the reference configuration

$$\boldsymbol{\Sigma} = \omega \nabla_{\boldsymbol{X}} \boldsymbol{\delta}\,, \quad \boldsymbol{\delta} = \boldsymbol{x} - \boldsymbol{X} \tag{49}$$

The force is set to

$$\boldsymbol{F} = \nabla_{\boldsymbol{X}} \omega \tag{50}$$

leading to the method originally proposed by Ceniceros and Hou in [21]. For other interpretations of the method, and analogies with elastic energy minimization problems, we refer the interested reader to [18].

In this work, when solving problem (48), we will assume that the reference domain is a closed polygon whose boundary $\partial\Omega_X$ is composed by the union of m segments. $\partial\Omega_X$ is mapped by M into the boundary $\partial\Omega_x$ and we further assume that it is invariant to the transformation. In particular we consider free-slip boundary conditions

$$\boldsymbol{\delta} \cdot \boldsymbol{n} = 0, \quad \boldsymbol{X} \in \partial\Omega_X \tag{51}$$

with $\boldsymbol{\delta} = 0$ at the polygon's vertices. A standard method to impose boundary condition is contained in [17] where it is introduced a second map $M^{\partial} : \partial\Omega_X \to \partial\Omega_x$ which correspond to the trace of (10) on the boundary. This mapping is then used as Dirichlet conditions to solve the trasformation for inner points. Alternatively as shown in [46] the variational formulation could be complemented by a constraint equation to take into account (51). We will however stick to form (48), written in terms of displacement, which is suited to express directly the boundary conditions. Lastly, the key ingredient is the definition of the monitor function ω controlling both the force and the stiffness in (48). A classical definition, given by Winslow [20], couples the mesh motion with the gradient of a given function u on the actual mesh : $\omega = \omega(\nabla_{\boldsymbol{x}} u)$. Here we have also tested the influence of the Hessian of the function, and as in [29], we have selected as initial target for adaptation the free surface η. Our definition of the smoothness monitor is

$$\omega = \sqrt{1 + \alpha \left(\max\left(||\nabla_{\boldsymbol{x}}\eta||^*, ||\nabla^2_{\boldsymbol{x}}\eta||^*\right)\right)^2} \tag{52}$$

where the $\|\cdot\|^*$ represent normalized L^2-norms computed as (the subscript $_{\boldsymbol{x}}$ is dropped for simplicity)

$$||\nabla\eta||^* = \min\left(1, \frac{||\nabla\eta||}{\beta \max||\nabla\eta||}\right), \quad ||\nabla^2\eta||^* = \min\left(1, \frac{||\nabla^2\eta||}{\gamma \max||\nabla^2\eta||}\right)$$

Note that in all of the above formulas, the derivatives of η are computed on the actual (moving) mesh, making problem (48) nonlinear. The coefficients α, β and γ are free parameters, allowing to optimize the mesh movement.

In practice, given an initial mesh in the reference domain, the weak form of (48) with boundary conditions (51) is discretized with a standard P^1 Galerkin finite element method. Due to the dependence of ω on the derivatives of η on the new mesh, the weak form defines a nonlinear system of algebraic equations which needs to be solved by means of some iterative procedure. The choice of this procedure and its coupling with the flow evolution equations plays a crucial role in determining the balance between the gain brought by the adaptation procedure, and its cost overhead w.r.t. the evolution of the flow quantities with the explicit schemes discussed in section §3. For this reason, we have chosen a simple explicit Newton-Jacobi iteration method, as in [45]. In particular, if κ_{ij} are the entries of the standard P^1 finite element stiffness matrix

obtained from (48), at each time step, the displacement $\boldsymbol{\delta}^k = \boldsymbol{x}^k - \boldsymbol{x}^n$ is computed from the following relaxed iteration

$$\hat{\boldsymbol{\delta}}_i^{k+1} = \boldsymbol{\delta}_i^k - \frac{1}{\kappa_{ii}^k}\left(\sum_{j\in\mathcal{D}_i}\kappa_{ij}^k\boldsymbol{\delta}_j^k + \boldsymbol{f}_i^k\right) \tag{53}$$

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^n + \mu_i\,\hat{\boldsymbol{\delta}}_i^{k+1} \tag{54}$$

with $\boldsymbol{f}_i^k = \sum_{j\in\mathcal{D}_i}\kappa_{ij}^k\,\boldsymbol{x}^n$. Note that the method obtained is similar to the one proposed in [45], but recast in terms of displacements so to embed more naturally the boundary conditions. As in the last references, to improve the control on the regularity of the mesh, we have introduced a relaxation phase in the iterations. In particular, the following definition of the relaxation parameter μ_i has been used (cf. also [17],[45])

$$\mu_i = \min\left(1, \max\left(\vartheta, \tau||\nabla\eta_i||\right)\right)$$

To avoid nodes'depletion in regions with small solution variations, a threshold for the stiffness is tuned by fixing ϑ, if $\vartheta \sim 0$ the stiffness in regions where $\nabla\eta \sim 0$ is strongly increased.

Finally, we recall that the entries of the stiffness matrix κ depend on the value of the monitor ω, and thus on the value of the solution on the new grid. As a consequence an essential element of this method is a sufficiently accurate projection step allowing to remap the discrete solution on the moving mesh. This projection step has to be chosen very carefully, as it impacts the overall accuracy, monotonicity, and cost of the computation. This issue will be extensively covered in section §4.2.

4.1 Mesh Adaptation to the shoreline and wetting/drying

The treatment of the wetting/drying phenomenon is crucial in many applications. In this work we need to clarify two issues. One is how wetting/drying is embedded in the numerical methods introduced in section §3, the second is how the adaptive mesh movement is modified in correspondence of wet/dry interfaces. The first aspect is discussed thoroughly in [37, 39] and [12] for the FV and RD methods respectively. The interested reader is referred to these references for all details. We limit ourselves to note that the treatment of these regions requires the introduction of two small quantities. The first is a threshold value C_H, such that a node is considered dry if $h_i \leq C_H$. The second, is a cut-off required to modify the mass fluxes and velocities close to dry cells. This value will be denoted here by C_U, and typically $C_H \ll C_U$. Concerning the second issue, we discuss two separate aspects: how to ensure the C-property close to a wet/dry interface, on a moving mesh; how to modify the definition of the monitor function so that a high resolution of the wet/dry interface is obtained with the adaptive mesh.

Firstly, to guarantee that the mesh movement does not spoil the preservation of the lake at rest state close to partially wet cells, an ad-hoc treatment is introduced. This procedure impacts the way in which the new water depth is computed from the free surface level obtained from the explicit updates of the well balanced ALE schemes of section §3. Given values of η_i^n, and η_i^{n+1}, obtained from the discretizations, and of b_i^n, and b_i^{n+1}, obtained using the quadrature approach, we proceed as follows

1. $\forall i$ compute the maximum wet water level $\eta_{0i} = \max_{K\in\mathcal{D}_i}\max_{\substack{j\in K\\ h_j>C_H}}\eta_j$
2. $\forall i$ set $\Delta_{b_i} = b_i^* - b_i^n$ with $b_i^* = \begin{cases} \eta_{0i} & \text{if } i \text{ is dry and } b_i^n + C_H > \eta_{0i} \\ b_i^n & \text{otherwise} \end{cases}$
3. Compute the new water depth as

$$h_i^{n+1} = \max(0, \eta_i^{n+1} - b_i^{n+1} + \Delta_{b_i}) \tag{55}$$

This correction, guarantees that when the mesh is moving, if a node is passing from the wet to the dry region (or vice-versa), the new values of the depth and of the free surface are the correct ones: namely $h_i^{n+1} = 0$ and $\eta_i^{n+1} = b_i^{n+1}$ in the dry areas; $h_i^{n+1} = \eta_{0i} - b_i^{n+1}$ and $\eta_i^{n+1} = \eta_{0i}$ in the wet areas. This guarantees that a constant flat free surface level is exactly preserved also near shorelines.

Concerning the tracking of the wet/dry interface in the mesh adaptation procedure, we need to provide an appropriate modification of the monitor function ω. There are in literature some examples of such front-tracking error functions. For example, in the context of phase change problems, Mackenzie and Mekwi [47] defined $\omega = \alpha/\sqrt{\beta|\boldsymbol{x} - \boldsymbol{x}_{interf}| + \gamma}$. This expression, however, requires the knowledge of the distance function from the interface, whose computation may be quite costly. Here we propose a simpler approach explicitly exploiting the knowledge that $h \to 0$ at the front. We have added a new term ϕ into the monitor function (52), through a proper weight δ

$$\omega = \sqrt{1 + \alpha \left(\max\left(||\nabla\eta||, ||\nabla^2\eta||\right)\right)^2 + \delta\phi^2} \tag{56}$$

with $\phi = \nabla f(\boldsymbol{x})$, and f is a function which is constant everywhere except in the narrow region where $C_H < h < C_U$:

$$\begin{cases} f(\boldsymbol{x}) = 0, & \text{if} \quad h(\boldsymbol{x}) \le C_H \\ f(\boldsymbol{x}) = \frac{h - C_H}{C_U - C_H}, & \text{if} \quad C_H < h(\boldsymbol{x}) < C_U \\ f(\boldsymbol{x}) = 1, & \text{if} \quad h(\boldsymbol{x}) \ge C_U \end{cases}$$

4.2 High order projections from ALE remaps

As already said, the Newton-Jacobi iterations (53),(54) require the projection of the solution values on the last updated mesh. The problem of computing updates for the solution values due only to the mesh movement can be elegantly solved by using remaps generated by the same schemes used to evolve the solution. Indeed, as a consequence of the DGCL property, the limit for $\Delta t \to 0$ of the schemes presented in sections §3.4 and §3.5 provide an instantaneous approximation to the conservative ALE remap equation, see (14) where b is replaced by a generic variable to be projected.

For example, for the FV scheme, taking the limit for $\Delta t \to 0$, and using equations (23),(25) we obtain the one step projection over the sub-grid $\mathcal{T}_h^{k+1}$

$$|C_i^{k+1}|\mathsf{w}_i^{k+1} = |C_i^k|\mathsf{w}_i^k - R_i(\mathsf{w}^k) \tag{57}$$

with $\mathsf{w}_i^k = \mathsf{w}^n(\boldsymbol{x}_i^k)$ the projected solution onto the sub-mesh generated at the previous k-th Newton iteration, and

$$R_i(\mathsf{w}) = \sum_{j \in \mathcal{D}_i} \left(-\Delta x_{ij} \frac{\breve{\mathsf{w}}_j + \breve{\mathsf{w}}_i}{2} - \frac{|\Delta x_{ij}|}{2} (\breve{\mathsf{w}}_j - \breve{\mathsf{w}}_i) \right)$$

This provides an approximation of the ALE remap for an instantaneous step in which the interface velocity is replaced by an interface displacement

$$\Delta x_{ij} = \int_{\partial C_{ij}^{k+1/2}} \Delta\boldsymbol{x}_h \cdot \boldsymbol{n}\, ds\,, \quad \text{with} \quad \Delta\boldsymbol{x}_h = \sum_{i \in \mathcal{T}} \varphi_i(\boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i^k)$$

The advantage of this approach is that it retains all the properties of the original method. A second order, non-oscillatory, well-balanced, mass conserving projection can be obtained by applying the limited high-resolution FV scheme. If the scalar, decoupled nature of the projection equations (all quantities independently are transported in the direction of the displacement) reduces the cost of these evaluations, it still means that the cost of one projection will be that of a single step of the FV scheme. As this may be repeated at every Newton-Jacobi iteration, this cost may lead to an important overhead.

For completeness we report the remap obtained with the RD scheme, which leads to a two step projection with

$$|C_i^{k+1}|\mathsf{w}_i^{k+1} = |C_i^{k+1}|\mathsf{w}_i^* - \sum_{K\in\mathcal{D}_i} \Phi_i^K(\mathsf{w}_h^k, \mathsf{w}_h^*;) \tag{58}$$

where the fluctuations Φ_i^K define a splitting of the average element residual

$$\sum_{j\in K} \Phi_j^K = \Phi^K(\mathsf{w}_h^k, \mathsf{w}_h^*) = \Big(\int_{K^{k+1}} \mathsf{w}_h^* - \int_{K^k} \mathsf{w}_h^k \Big) - \int_{\partial K^{k+1/2}} \frac{\mathsf{w}_h^k + \mathsf{w}_h^*}{2} \Delta \boldsymbol{x}_h \cdot \boldsymbol{n}\, ds$$

The values w_i^* are obtained from a first order predictions obtained in a similar fashion by taking the limit $\Delta t \to 0$ in the RD predictor step, see (44).

5 Adaptive algorithms

We have now all the basic blocks to perform adaptive mesh simulations. These boil down to the flow evolutions equations (section §3) and to a PDE based mesh movement solver (MMPDE, discussed in section §4). We propose hereafter 2 *alternate* techniques, which are extensively tested in the numerical results. A weakly coupled ALE method and a decoupled adaptation-evolution steps. Particular cases of these two implementations have already been considered in literature (see e.g. [17] and [25] for the ALE). However, their impact on the overall cost of the simulation, and on the quality of the results has never been assessed.

5.1 Moving Mesh ALE algorithm (ALE)

The balance law is written, by means of the ALE formulation, directly in a framework coincident with the moving domain. At every time step we get the solution on the adapted grid, *independently on the interpolation scheme* which is only needed now to evaluate the error monitor. The algorithms reads :

`Step 1.` Taken a triangular mesh $\mathcal{T}_h^n$, compute the vectors of nodal coordinates $\mathbf{x}^n$, and the initial solution w_h^n. Set the initial conditions for the MMPDE, $\eta_h^1 = \eta_h^n$ and $\mathbf{x}^1 = \mathbf{x}^n$.

`DO k=1,kmax`

`Step 2.` Compute the monitor function $\omega^k = \omega(\eta_h^k)$ and matrix $\kappa = \kappa\left(\omega^k\right)$. Move the mesh according to the Newton-Jacobi iteration (Eq. (53) and (54)). At each iteration we get $\mathbf{x}^{k+1}$.

`Step 3.` Compute the interpolated free surface η_h^{k+1} according to the scalar version of FV/RD projections, (57) or (58) with frozen flow speed.

`ENDDO`

`Step 4.` Let $\mathbf{x}^{n+1} = \mathbf{x}^{kmax+1}$ and $\mathcal{T}_h^{n+1} = \mathcal{T}_h^{kmax+1}$. Evolve the underlying balance law in ALE framework with the FV/RD-RK2 scheme, see Eq. (23) or Eq. (41), on the midpoint grid $\mathcal{T}_h^{n+1/2}$.

`Step 5.` Let $\mathcal{T}_h^n = \mathcal{T}_h^{n+1}$ and $\mathsf{w}_h^n = \mathsf{w}_h^{n+1}$.

`IF (t > T) EXIT`

`ELSE GO TO Step 1.`

We see that the interpolated solution is only used to evaluate the error function, so the interpolation step can be simplified a great deal without affecting the quality of the solution, as the numerical tests will confirm.

5.2 Moving Mesh Eulerian algorithm/rezoning (EUL1)

In this case, considered for example in [17, 45], the balance law is resolved numerically at every time step in a purely Eulerian framework, and on a fixed mesh. The latter is then adapted to the new solution and an accurate guess for the values of the last solution on the new mesh is provided by the projection scheme. The algorithm reads:

`Step 1.` Taken a triangular mesh $\mathcal{T}_h^n$, compute the vectors of nodal coordinates $\mathbf{x}^n$, and the initial solution w_h^n. Set the initial conditions for the MMPDE, $\mathsf{w}_h^1 = \mathsf{w}_h^n$ and $\mathbf{x}^1 = \mathbf{x}^n$.

`DO k=1,kmax`

`Step 2.` Compute the monitor function $\omega^k = \omega(\eta_h^k)$ and matrix $\kappa = \kappa\left(\omega^k\right)$. Move the mesh according to the Newton-Jacobi iteration (Eq. (53) and (54)). At each iteration we get $\mathbf{x}^{k+1}$.

`Step 3.` Compute the full interpolated solution w_h^{k+1} according to FV/RD projections, see (57) or (58).

`ENDDO`

`Step 4.` Let $\mathbf{x}^{n+1} = \mathbf{x}^{kmax+1}$ and $\mathcal{T}_h^{n+1} = \mathcal{T}_h^{kmax+1}$. Moreover let $\mathsf{w}_h^n = \mathsf{w}_h^{kmax+1}$, the interpolated solution over the new mesh. Evolve the underlying conservation law in Eulerian framework using the FV/RD-RK2 scheme, see Eq. (23) and Eq. (41) with $\boldsymbol{\sigma} = 0$, on the grid $\mathcal{T}_h^{n+1}$.

`Step 5.` Let $\mathcal{T}_h^n = \mathcal{T}_h^{n+1}$ and $\mathsf{w}_h^n = \mathsf{w}_h^{n+1}$.

`IF (t > T) EXIT`

`ELSE GO TO Step 1.`

Since this time the interpolated solution will act as the initial condition for the new time iteration, great care has to be put in its computation. The interpolation step does not have to spoil the accuracy property of the numerical scheme. As a consequence, costly projections obtained from high resolution non-linear schemes have to be used to ensure that the quality of the results is not spoiled.

5.3 Moving Mesh Eulerian algorithm (EUL2)

In the previous algorithm, a double role emerges for the interpolation step. Firstly we need an interpolated solution η_h^k at every Newton sub-step in order to evolve the mesh. Secondly we provide an interpolated solution w_h^{kmax+1} on the final updated mesh in order to give a proper initial condition for the flow solver. This leads to the idea of using a simplified version of the Eulerian algorithm in which simplified scalar projections are used for the free surface variable, as in the ALE algorithm. A full high resolution remap is used only after reaching $k = kmax$ in the adaptation loop, to perform the interpolation $|C_i^{n+1}|\mathsf{w}_i^{kmax+1} = |C_i^n|\mathsf{w}_i^n - R_i$.

6 Computational details

In the scalar case the time step was prescribed in order to have, where possible (for examples a Lax-Friedrichs or first order Godunov scheme), the satisfaction of a discrete maximum principle

$$\Delta t = \mathrm{CFL} \min_{i \in \mathcal{T}_h} \frac{|C_i^{n+1}|}{\sum_{K \in \mathcal{D}_i} 3\alpha^K}$$

with CFL = 0.8 and $\alpha^K \geq \max_{j \in K} |\frac{1}{2}\bar{\boldsymbol{a}}^K \cdot \boldsymbol{n}_j|$ and $\bar{\boldsymbol{a}}^K$ is the average flux jacobian over a cell. The following MMPDE parameters are used $\alpha = 10$, $\beta = \gamma = 0.15$. We did not perform a serious optimization relative to these parameters but we procedeed after a few trials. The relaxation parameter is taken constant $\mu = 1$. We used only $kmax = 5$ iterations of the Newton-Jacobi method which, it is important to remark, do not

ensure the convergence of the iterative method within each time step. In practice they are sufficient to get nodes refinement.

For systems, the same definition of discrete maximum principle is not clear to the authors. For the Shallow Water experiments, the time step has been fixed using the above definition with CFL = 1.20 and $\alpha^K = \frac{1}{2}\max_{j\in K}\left(|l_j|\left(||\boldsymbol{u}_j^n|| + \sqrt{gh_j^n}\right)\right)$. Threshold value $C_H = 10^{-5}$ while for $C_U = h_K^2/L_{ref}$ with L_{ref} a reference length. In SW simulations, the MMPDE parameters used in the monitor function (see section §4) are $\alpha = 20$, $\beta = \gamma = 0.10$ and $\delta = 3\alpha$, unless otherwise specified. To tune the relaxation parameter we used $\tau = 3$ and $\vartheta = 0.7$. Again, they have been chosen after a few trials. Finally the number of iterations is always $kmax = 5$.

7 Numerical Results

7.1 Scalar tests

7.1.1 Well balanced form and accuracy

To test the WB property on moving mesh for the two ALE formulations seen in section §3.3, we use the simple case of linear advection of a smooth sinusoidal hill

$$\begin{cases} \frac{\partial u}{\partial t} + \boldsymbol{a}\cdot\nabla u + \boldsymbol{a}\cdot\nabla b = 0, & \boldsymbol{a} = [0,1]\,,\ \boldsymbol{x}\in[0,1]\times[0,2],\ t\in[0,1] \\ u_0(\boldsymbol{x}) = 1 - b(\boldsymbol{x}) + \cos^2(2\pi r) & \text{if } r \le 0.25,\ r = \sqrt{(x-0.5)^2 + (y-0.5)^2} \\ u_0(\boldsymbol{x}) = 1 - b(\boldsymbol{x}) & \text{otherwise} \end{cases}$$

The pseudo-bathimetry is defined by $b(\boldsymbol{x}) = 0.8e^{\psi(x,y)}$ with $\psi = -5\,(y-0.9)^2 - 50\,(x-0.5)^2$. The following arbitrary mapping is used to move the mesh

$$\begin{cases} x(t) &= X + 0.1\sin(2\pi X)\sin(\pi Y)\sin(2\pi t) \\ y(t) &= Y + 0.2\sin(2\pi X)\sin(\pi Y)\sin(4\pi t) \end{cases} \tag{59}$$

We check the validity of the analysis of section §3.2 on this smooth case by performing a grid convergence study (halving the mesh sizes h_K in the reference domain), and by visually checking the preservation of the state $\eta = 1$. The computations are run with the RD scheme, but the FV results are almost identical. The results are summarized in figure 2. We can confirm that: when no perturbation is added, the well balanced ALE formulation (16) (ALE WB in the figures) preserves the constant state to machine accuracy (not shown in the figures), while the classical ALE form (15) (ALE NO WB in the figure) does not, as the left and middle pictures clearly show.

For the smooth perturbation (and pseudo-bathymetry) considered here we observe second order of accuracy for both the formulations. However the presence of spurious oscillations in the flat region increase substantially the absolute value of the error obtained with the unbalanced ALE form.

7.1.2 Coupling algorithm: efficiency

To test the efficiency of the different methods proposed in section §5, we use the classical rotation of a smooth sinusoidal hill, this time with a source term

$$\begin{cases} \frac{\partial u}{\partial t} + \boldsymbol{a}\cdot\nabla u + \boldsymbol{a}\cdot\nabla b = 0, & \boldsymbol{a} = [-2y, 2x]\,,\ \boldsymbol{x}\in[-1,1]\times[-1,1],\ t\in[0,\pi] \\ u_0(\boldsymbol{x}) = 1 - b(\boldsymbol{x}) + \cos^2(2\pi r) & \text{if } r \le 0.25,\ r = \sqrt{x^2 + (y+0.5)^2} \\ u_0(\boldsymbol{x}) = 1 - b(\boldsymbol{x}) & \text{otherwise} \end{cases}$$

with

$$b(x,y) = 0.8e^{\psi(x,y)}, \quad \psi = -5y^2 - 5x^2$$

We perform a grid convergence study, and investigate the dependence of the error on the CPU time. We perform the same test for both the RD and FV scheme. In figure 3 the convergence curves for the different combinations of moving mesh algorithms and interpolations schemes are reported. The interpolation step necessary to evolve the mesh has a positive impact on mesh delay, however the specific scheme used, weakly influences mesh configuration. For the ALE and EUL2 algorithms, we see that all the curves in blue color are almost overlapped. Using this result, in the following numerical test cases, the EUL2 and ALE algorithms will be used in their faster versions with inaccurate interpolation into the MMPDE. On the contrary for the EUL1 algorithm there is only one interpolation scheme which guarentees stable and second order accurate results, actually the one which we evolve the PDE with.

In figure 4 the performances of the different algorithms are compared in terms of error/time. With the RD method, the ALE algorithm shows the lowest CPU time, for a fixed level of error (roughly 80% faster then a fixed grid computation). The Eulerian algorithms are less efficient because the full two stage RK interpolation had to be implemented (60% gain for EUL2 and 35% for EUL1). For the FV scheme the efficiency between ALE and Eulerian algorithms is more similar (ALE and EUL2 80%, EUL1 70%) because this time, in the interpolation step, the second stage of RK seemed to be not necessary, as emerges also from the work of [17]. Finally, for both RD and FV, the EUL2 rapresents a slight improvement respect to the EUL1 algorithm.

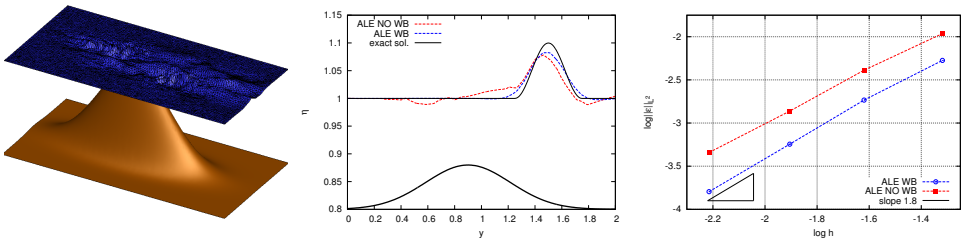

Figure 2: Linear Advection. Left: Lake at rest for the NO WB ALE formulation and failing in verifiyng Well Balanced. Middle: comparison beteween the numerical solution and exact one on the simmetry line $x = 0.5$. Right: convergence order for the L^2-norm of the error.

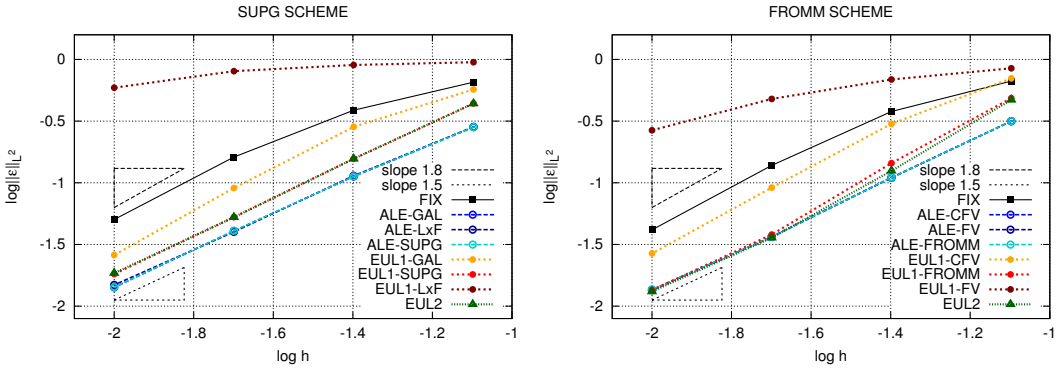

Figure 3: Rotation. Order of convergence: Left, RD scheme. Right, FV scheme.

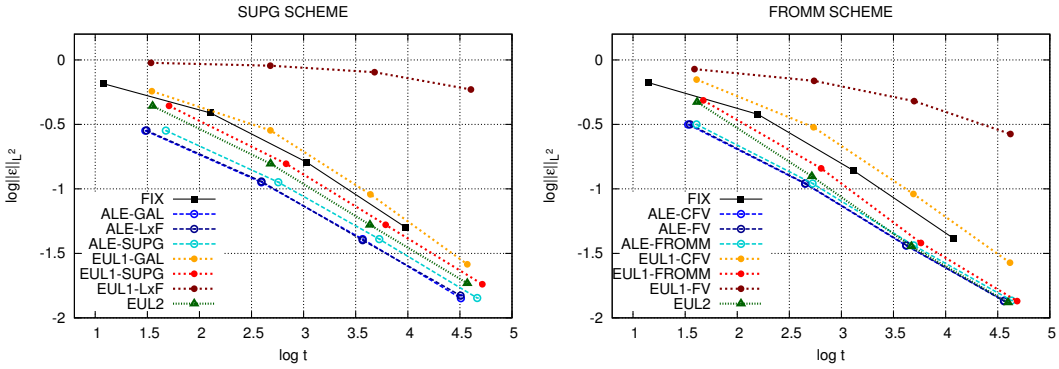

Figure 4: Rotation. Error vs CPU time: Left, RD scheme. Right, FV scheme.

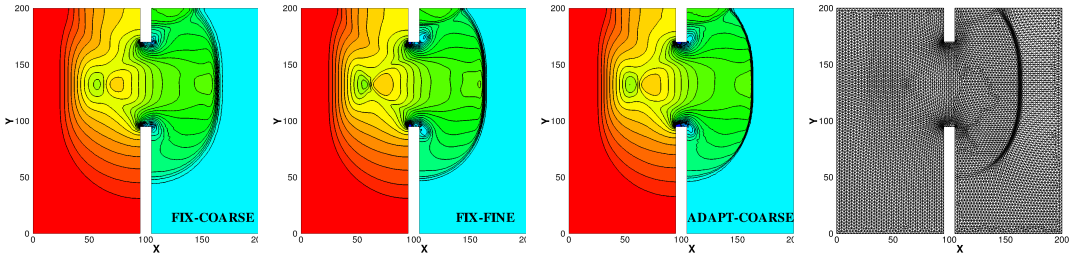

Figure 5: Asymmetric dam-break computed with RD scheme. 30 equispaced contour lines for h and adapted mesh.

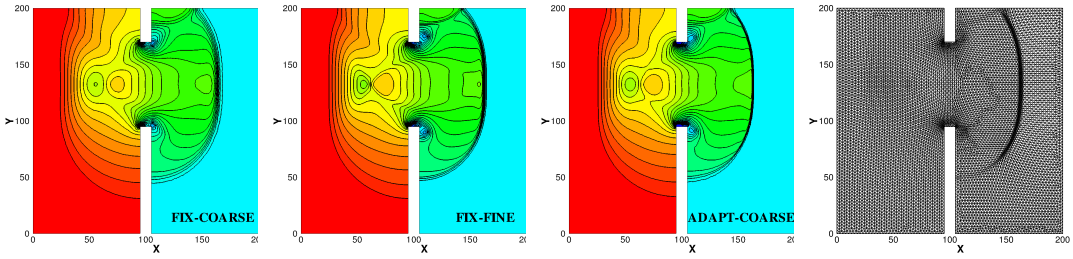

Figure 6: Asymmetric dam-break computed with FV scheme. 30 equispaced contour lines for h and adapted mesh.

8 Numerical results for Shallow Water equations

8.1 Asymmetric dam Break

This classical test benchmark, taken from [48], is used to test the adaptive algorithm when bores develop. The set-up consists in a square domain $[0 \times 200]^2\, m$ with a dam, placed at $x = 95\, m$, separating an upper and a lower bassin which contain water at different levels, respectively at $10\, m$ and $5\, m$. The sudden break of the dam leads to a depression wave advancing in the upper bassin and a bore advancing in the lower bassin. Two corners depression interact, forming a deep trough at the inlet of the dam.

The test is run with both the FV and RD scheme, on a coarse triangulation containing 14538 triangles and 7480 nodes, on a fine one, containing 77302 triangles and 39130 nodes, and on the coarse mesh with adaptive mesh deformation. The typical qualitative result obtained is provided in figures 5 and 6. The pictures show the potential of this adaptation procedure to provide with considerably fewer unknowns a much better resolution of both the smooth and the non-smooth flow features.

In figures 7,8 a comparison between the ALE algorithm and the EUL1 and EUL2 is shown. For both RD and FV, the ALE algorithm shows a well resolved bore and a correct computation of the trough with a significant saving in CPU time. As shown on table 1, the savings obtained with the ALE algorithm go up to 60% for RD, and 50% for FV. For the RD scheme, the cost of of a two-step interpolation, makes the EUL1 algorithm inefficient, thus the EUL2 a clear improvement. For FV both the interpolation based algorithms (EUL1 and EUL2) are not able of of providing a considerable improvement in the resolution of the peaks and the trough upstream the dam ($x \backsimeq 60$), probably due to execessive numerical diffusion in the interpolation. Some improvement is instead observed with the ALE algorithm, which also gives a much sharper capturing of the bore.

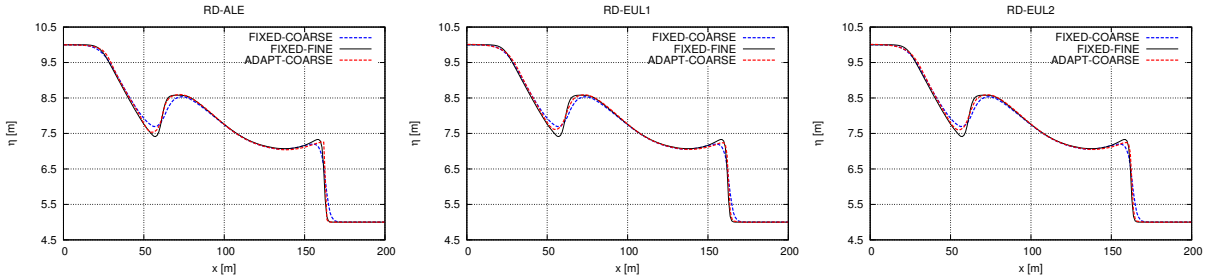

Figure 7: Asymmetric dam-break computed with RD scheme. Solution along the straight line at $y = 132.5$ for the different coupling. Left: ALE. Middle: EUL1. Right EUL2.

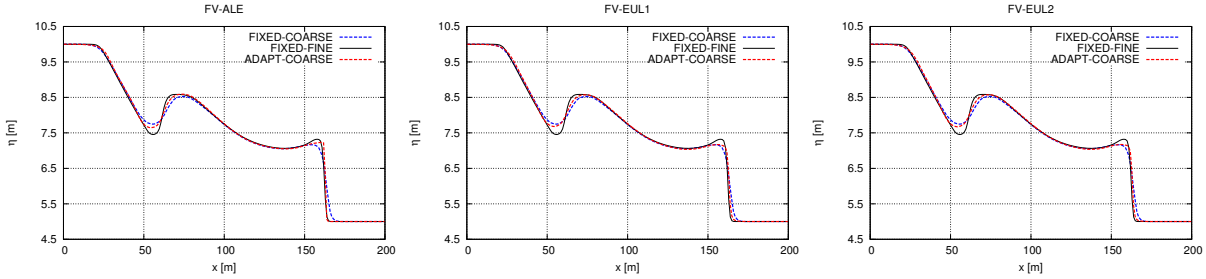

Figure 8: Asymmetric dam-break computed with FV scheme. Solution along the straight line at $y = 132.5$ for the different coupling. Left: ALE. Middle: EUL1. Right EUL2.

ALG.	MESH (Nodes)	RD [s]	FV [s]
FIX-COARSE	7480	11.34	11.97
FIX-FINE	39130	185.00	207.14
ADAPT-ALE	7480	77.48	100.16
ADAPT-EUL1	7480	169.63	150.52
ADAPT-EUL2	7480	98.30	111.15

Table 1: Asymmetric dam-break. CPU times.

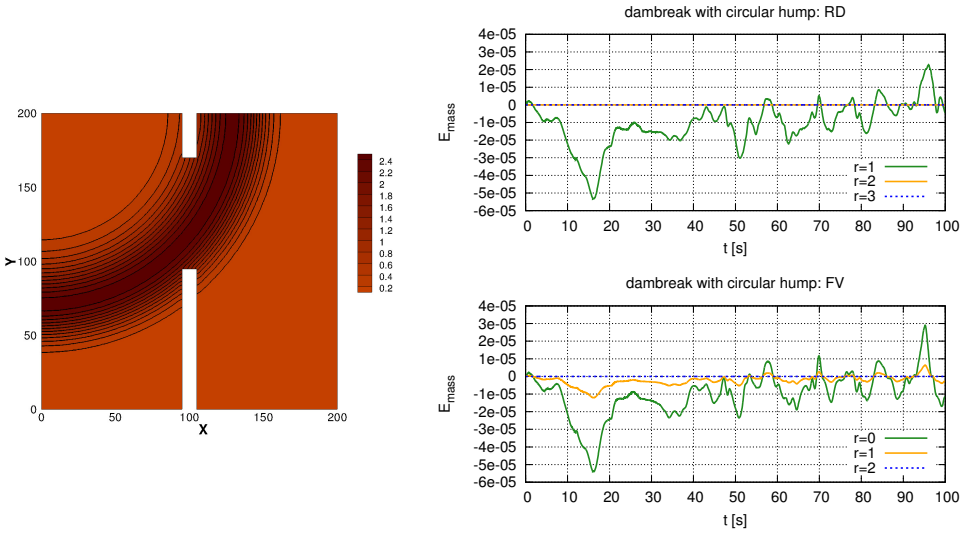

Figure 9: Dambreak with circular hump. Left: bathymetry. Right: dimensionless mass error for different quadrature formula of the bathymetry integral.

To check our mass conservation correction (cf. section § 3.3.1) we repeat this test adding a bathymetry shaped as a circular hump centered in $(x, y) = (0, 200)$, and defined by an exponential law in the radial direction (cf. left picture on figure 9). We report on the right pictures on figure 9 the mass error measured without any correction, and with corrections based on different quadrature formulas (for the definition of E_{mass}, see section § 3.3.1). We can clearly see that, even for this non-smooth case, we are able to preserve the total mass in the domain practically up to machine accuracy.

8.2 Small perturbation of a lake at rest

We consider the classical test of a small perturbation over an elliptic exponential hump (se e.g. [48, 14] for details concerning the test setup). This test allows to check the ability of the algorithms proposed to catch relatively smooth wave patterns, and to conserve mass, and the lake at rest state in the unperturbed regions. To run the test, we use a coarse triangulation, containing 12142 nodes and 23852 triangles, and we compute "reference" solutions on a finer mesh, containing 50631 nodes and 100376 triangles.

The qualitative behavior of the methods proposed can be seen in figures 10 and 11 (same contour lines drawn in all the pictures). We can see that the mesh follows quite well the propagation and transformation of the waves, providing, on the coarse mesh, a resolution very close to the reference one. No numerical artifacts are observed in the unperturbed region, as a consequence of the exact preservation of the lake at rest state. To perform a more quantitative analysis we report in table 2 the CPU times of all the schemes, and the water height along the line at y=0.5 on figures 12 and 13. For clarity, only the EUL2 method results are plotted in the latter figures, the EUL1 algorithm providing virtually identical solutions.

The cuts show how both the ALE and the rezoning algorithms provide solutions close to the reference one. The CPU time savings w.r.t. the reference are of the order of 70% for the ALE method, of 60% for the EUL2, and between 50% (for FV) and 40% (for RD) for the EUL1 algorithm.

Finally, figure 14 shows a study of mass conservation, providing additional proof that the corrections proposed allows to retain the physical mass in the domain virtually to machine accuracy.

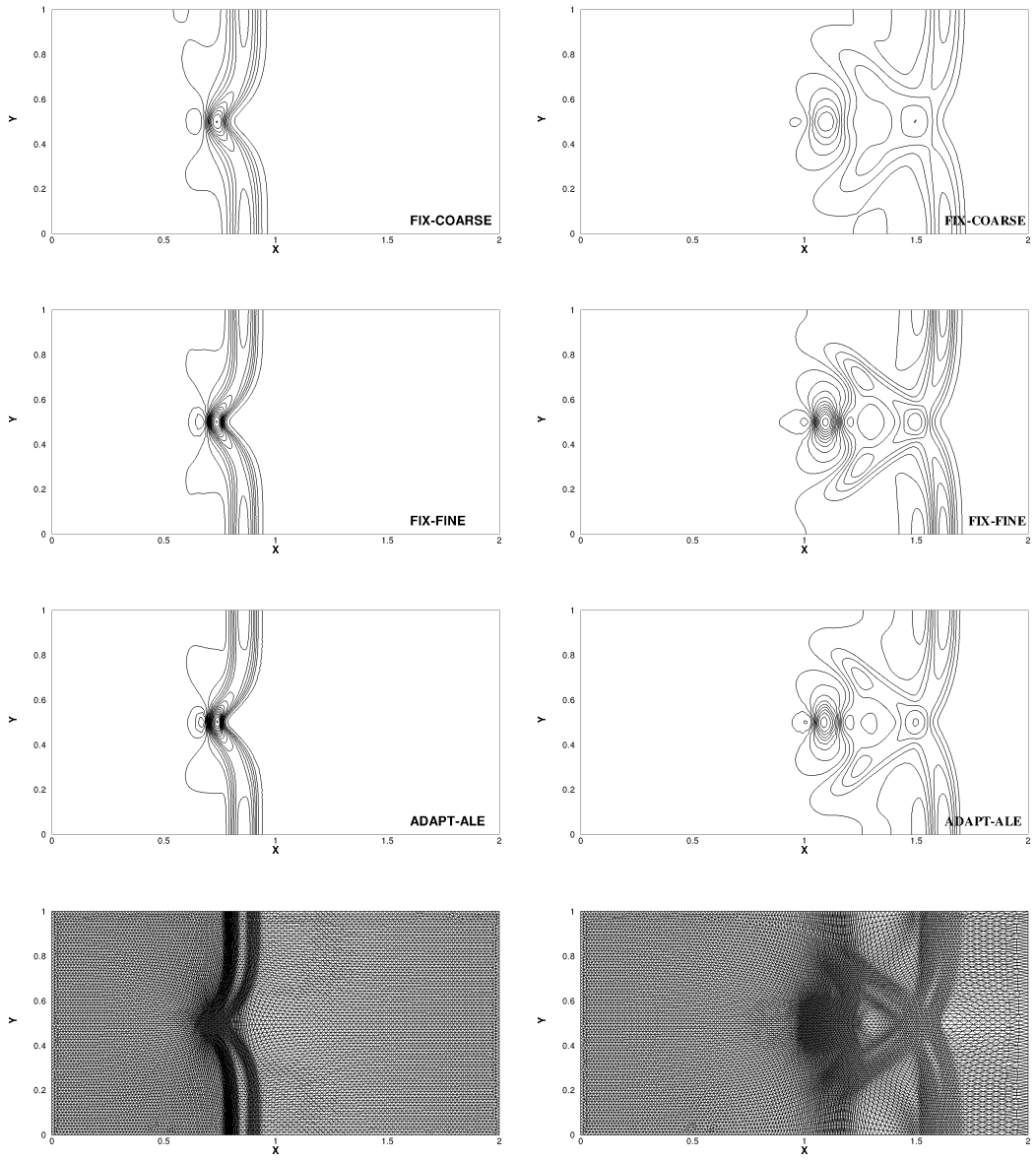

Figure 10: Small perturbation of a lake at rest (RD scheme). Solution isolines at $t = 0.24$, $t = 0.48$ are shown for fixed grid and adaptive computations. Top: fixed coarse grid. Middle: fixed fine grid. Bottom: adaptive ALE scheme.

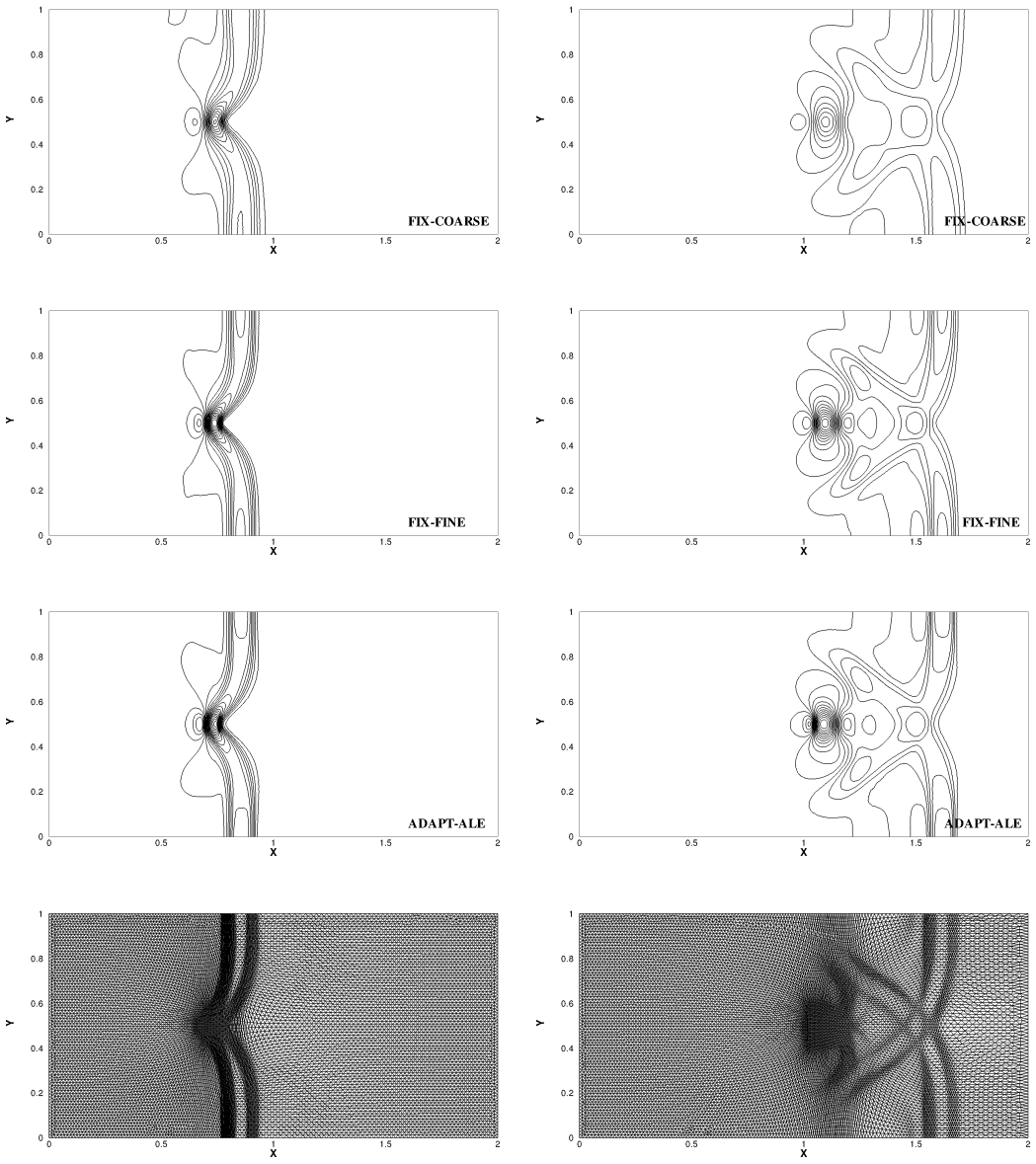

Figure 11: Small perturbation of a lake at rest (FV scheme). Solution isolines for $t = 0.24$, $t = 0.48$ are shown for fixed grid and adaptive computations. Top: fixed coarse grid. Middle: fixed fine grid. Bottom: adaptive ALE scheme.

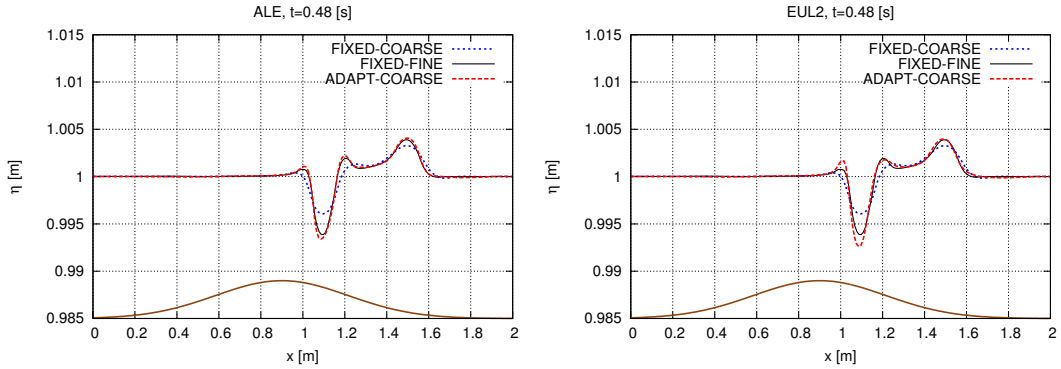

Figure 12: Small perturbation of a lake at rest (RD scheme). Solution at $t = 0.48$ along line $y = 0.5$. Left: ALE. Right: EUL2.

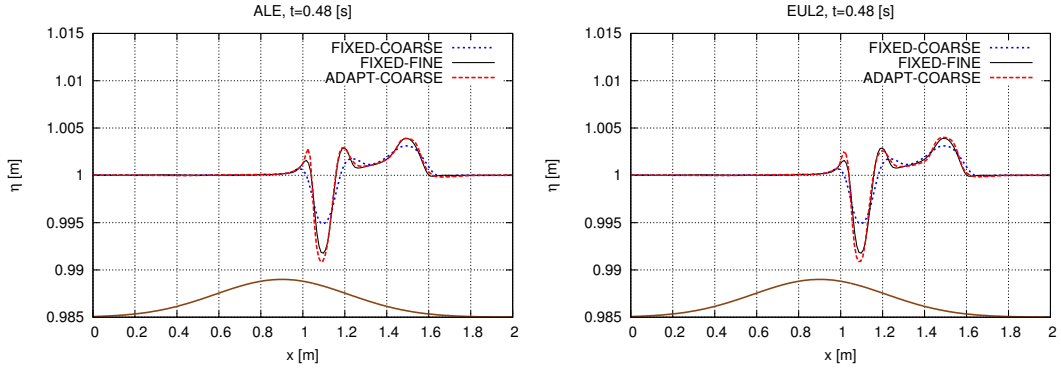

Figure 13: Small perturbation of a lake at rest (FV scheme). Solution at $t = 0.48$ along line $y = 0.5$. Left: ALE. Right: EUL2.

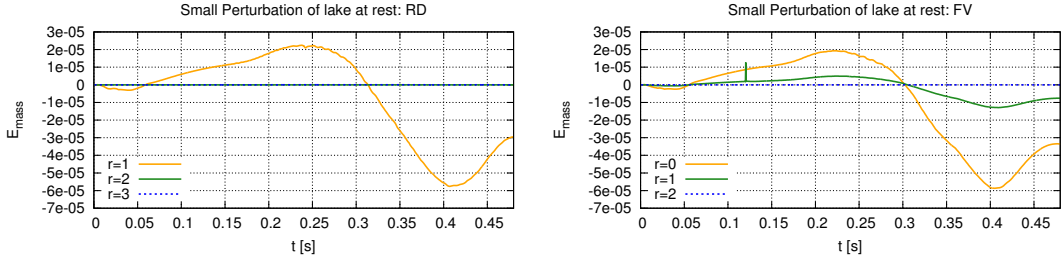

Figure 14: Small perturbation of a lake at rest. Dimensionless mass error for different quadrature formula of the bathymetry integral.

ALG.	MESH (Nodes)	RD [s]	FV [s]
FIX-COARSE	12142	73.60	79.06
FIX-FINE	50631	711.08	827.72
ADAPT-ALE	12142	204.96	254.77
ADAPT-EUL1	12142	416.33	392.99
ADAPT-EUL2	12142	282.28	319.12

Table 2: Small perturbation of a lake at rest. CPU times.

8.3 Runup on a conical island

This is another very classical benchmark aiming at reproducing some of the experiments of [49]. We refer to the above reference, and to [14, 12, 39] for the test setup. The parameter in the MMPDE are $\alpha = \delta = 20$, $\beta = \gamma = 0.2$. This benchmark will allow to test the ability of the algorithms proposed to track dry fronts, as well as the mass conservation correction. We have run the test on two meshes, both progressively refined in the region of interaction between the wave and the conical island. The coarse one, contains 10401 nodes, and 20580 triangles, with mesh sizes h_K going from 0.5 to 0.2 meters. The fine mesh contains 37982 nodes, and 75594 triangles, with mesh sizes going from 0.3 to 0.08 meters. The fine mesh results obtained with FV and RD are quite close (cf. figure 16), and similar to those typically shown in literature. They have been used as a reference for those obtained on the coarse mesh, with adaptive mesh deformation.

The qualitative behavior of the method is shown on figure 17. The pictures show the ability of the modified monitor function to track both the incoming and refracting waves, and the moving wet/dry interfaces. The gauge signals for the adaptive simulations are reported in figure 18 for the gauges g9 (upstream the island), g16 (lateral runup), and g22 (rear side runup). The results obtained on gauges 9 and 22 show that, for both FV and RD, the adaptive ALE algorithm provides results comparable to those obtained on the fine mesh. In particular, the interference between the two refracted waves that causes the peak and highest runup values on the back of the island, is well reproduced. This is also the case with the interpolation-based methods, which provide practically the same results (only EUL2 show in the plots). In the RD case, all the adaptive algorithms lead to a less impressive improvement in the lateral runup gauge 16.

CPU times are reported on table 3. We can see that the ALE adaptive computations allow still savings of the order of 71% w.r.t. the fine mesh computation. The percentages of CPU time reduction for the rezoning algorithms are close to 66% for the EUL2 method, and to 37% (for RD) and 44% (for FV) for the EUL1 algorithm. Lastly, the tables also report the % of the total cost represented by the moving mesh algorithm alone including the recomputation of geometrical quantities. These show that, while for the ALE the overhead w.r.t. a fixed mesh simulation is of 40%, the EUL2 and EUL1 algorithms counts for, respectively, 50% and 70% of the computation. This means that more time is spent adapting the mesh than in computing the flow. Clearly, this is a consequence of the costly projection steps on which the method relies.

Finally, figure 19 shows the study of mass conservation for this problem. The pictures prove how a high accuracy correction of the nodal bathymetric heights, combined with the redistribution of the spurious geometric mass generated by the motion of dry nodes, allows to reduce the mass error practically to zero.

8.4 Monai valley benchmark

This test involves the tsunami runup over a complex 3D bathymetry, and is a standard test for tsunami simulation models [50]. The experiment that it reproduces was carried out at Central Research Institute for Electric Power Industry (CRIEPI) in Abiko (Japan), and consisted of a 1/400 reproduction of the Hokkaido-Nansei-Oki tsunami of 1993 that struck Okushiri Island, with disastrous consequences especially in the region of the Monai village, on which the experiment itself focuses. For a full description of the

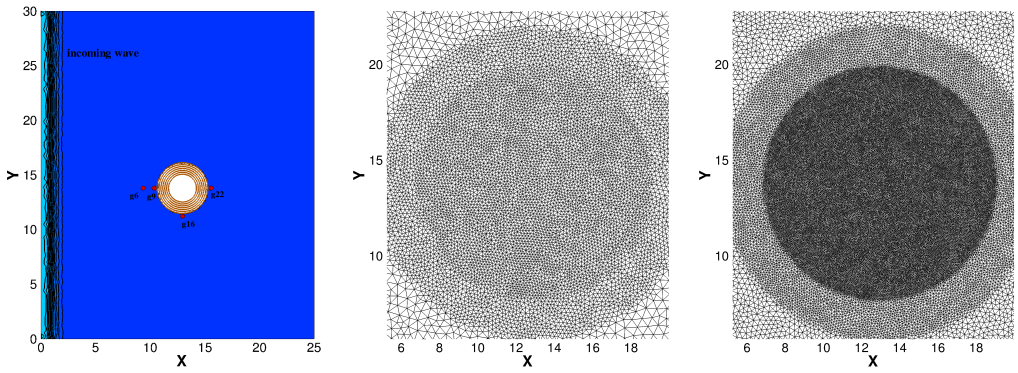

Figure 15: Conical island. Left: sketch of the computational domain with gauges. Middle: static coarse mesh topology. Right: static fine mesh topology.

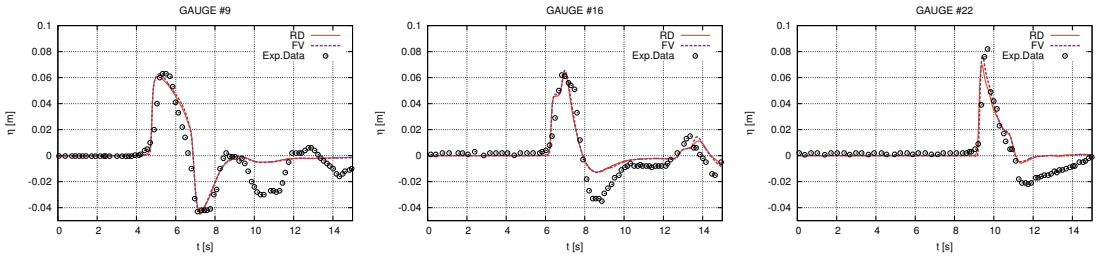

Figure 16: Conical Island computed with fixed fine mesh: total water height η signal registered at the gauges g9, g16, g22 and comparison with experimental data.

ALG.	Mesh (Nodes)	RD[s] (%MMPDE)	FV[s] (%MMPDE)
FIX-COARSE	10401	171.30	210.37
FIX-FINE	37982	1785.96	1959.02
ADAPT-ALE	10401	510.65 (38.8%)	574.52 (37.4%)
ADAPT-EUL1	10401	1115.98 (73.2%)	1086.66 (68.1%)
ADAPT-EUL2	10401	608.41 (51.3%)	653.14 (46.8%)

Table 3: Conical island. CPU times.

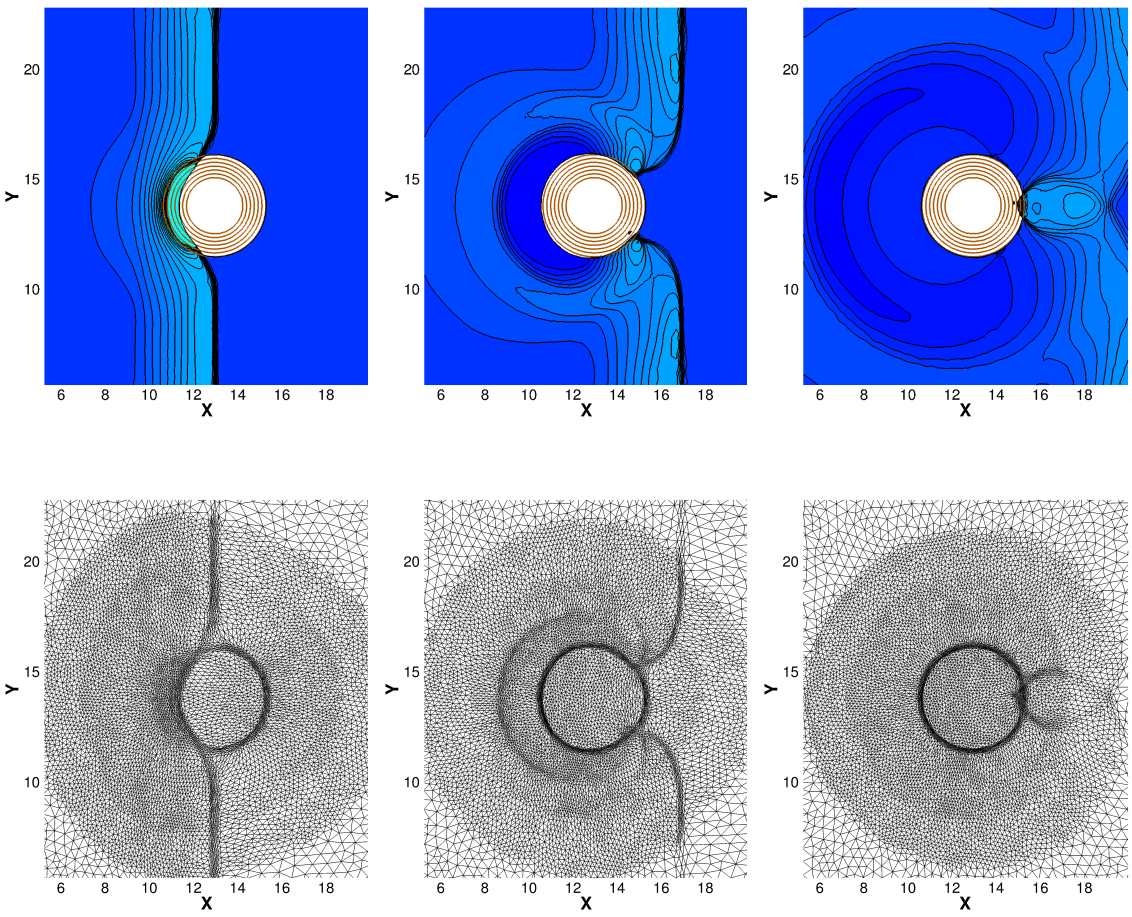

Figure 17: Conical Island: contour lines for total water height η and adapted mesh at different time instants, $t = 6.0, 8.0, 10.0\,[s]$

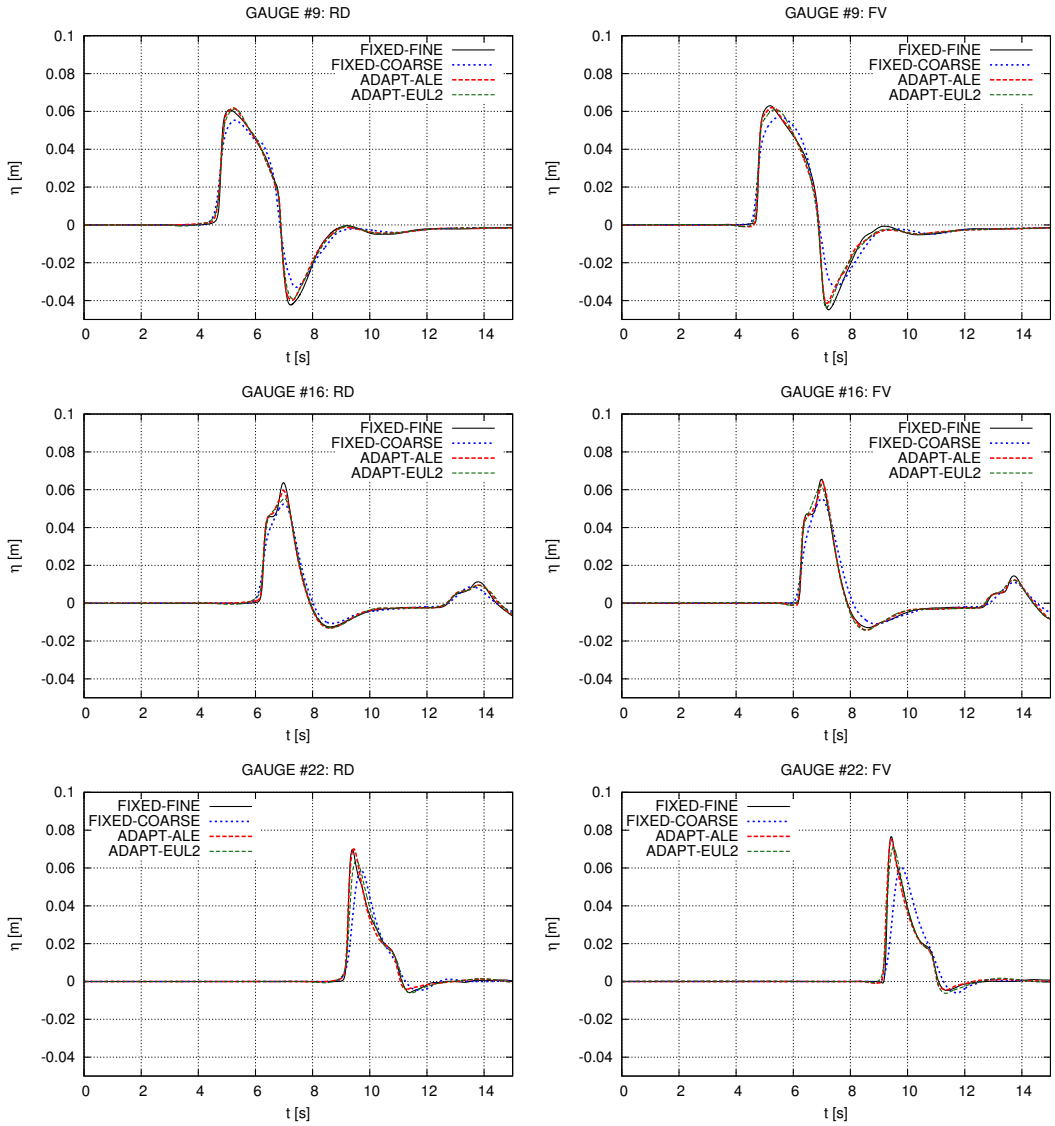

Figure 18: Conical Island: comparison between adaptive algorithms and fixed grid computations. Total water height η signal registered at the gauges g9, g16, g22. Left: RD scheme. Right: FV scheme.

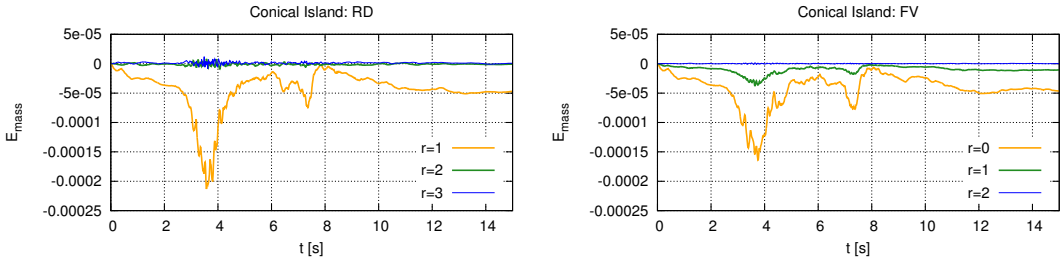

Figure 19: Conical island. Dimensionless mass error for different quadrature formula of the bathymetry integral.

setup, including all the necessary data to run the test, and with the results from the experiments, we refer to page of the center for Tsunami research at NOAA [51]. We have run this test on the grids reported on the right pictures of figure 20, statically adapted to the bathymetric variations [14]. The coarse one contains 7000 nodes and 13720 triangles, with mesh sizes h_K ranging from 0.1 to 0.025 meters; the fine mesh contains 36911 nodes and 18711 triangles, with sizes ranging from 0.05 to 0.01 meters. Note that the prescribed uniform mesh size for this test is usually of 1.4 cm [51]. We have used the fine mesh results as a reference, to compare against the solutions obtained with adaptive mesh deformation on the coarse grid.

The qualitative impact of the adaptation algorithms has been visualized on figure 20, which reports plots relative to the instant of maximum runup. The top rows report the fix grid results, while the bottom one shows the solution on the adaptive grid, and the mesh itself. The moving adaptive result shows a clear improvement in the reflected bores, and, as we will see in more detail shortly, runup heights very close to those obtained on the fine mesh. Note that this is a difficult test for the overall method, as the initial non-uniform mesh size distribution leads to strongly anisotropic triangles in the adaptive case, as clearly visible in the figure.

As already remarked in [14], there is little influence of the mesh size on the gauge signals. This is shown clearly by the water height signal in gauge 7, reported for completeness in figure 21. A much more interesting quantity to look at is the runup plot, which is provided in the top row of figure 22. In the pictures, the brown line represents the height of the maximum runup observed in the experimental setup in the narrow gulley with a cove at $(x, y) \approx (5.15, 1.875)[m]$ in the scaled down model. The figure shows that only with finer grids the correct runup height can be reached, and that both the ALE and rezoning methods allow to obtain the correct prediction on the coarser grid. To corroborate this result, we have placed an additional gauge (not present in the experiment). Its position is at $(x_g, y_g) = (5.05, 1.9)[m]$, very close to the maximum runup point. The water height time series in this gauge are reported in the bottom row of figure 22. These pictures confirm that the ALE algorithm is superior in allowing to retain the correct values of the maximum water heights, even though failing in reproducing the exact shape of the signal. The rezoning methods also provide a considerable improvement over the coarse mesh result, with water heights very close to the reference. CPU times are given in table 4.

Lastly, the evolution of the mass conservation error is reported on figure 23. Again we can see the improvement brought by the corrections proposed here.

8.5 Solitary wave on a shelf with an island

Finally, as an application to a more complex flow, we consider a laboratory experiment, conducted in the wave tank of the Oregon State University, involving the solitary wave runup over a shelf with a conical island. The bathymetry used here is a perturbed variant of the piecewise analytical one, provided within the French TANDEM research program $[http://www-tandem.cea.fr]$. A 3D view of the bathymetry is reported on figure 24. For this benchmark experimental time series of the water height are available

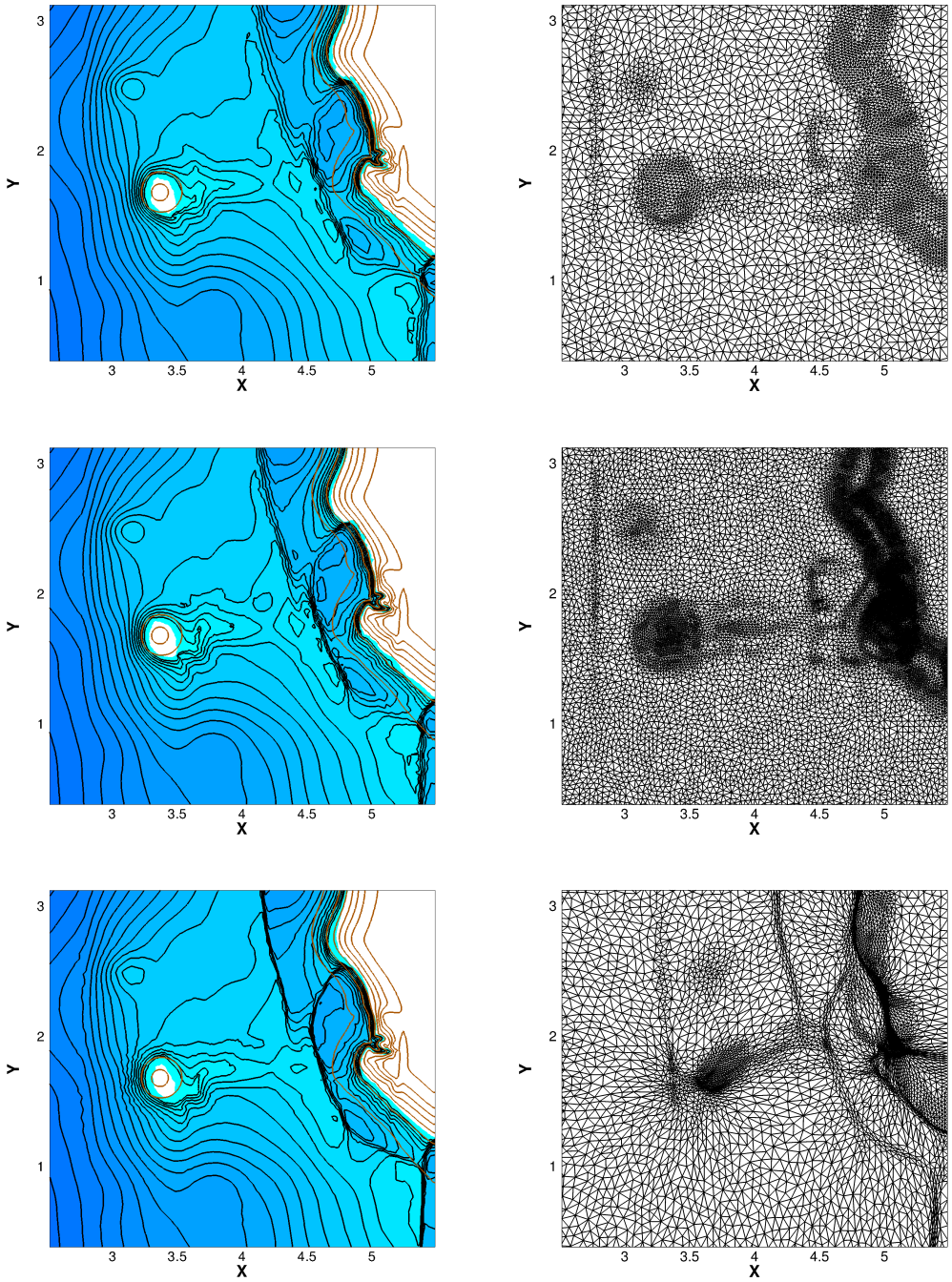

Figure 20: Okushiri experiment. Contour lines for h and mesh at $t = 16.5\,[s]$. Top: fixed coarse grid. Middle: fixed fine grid. Bottom: adaptive ALE scheme.

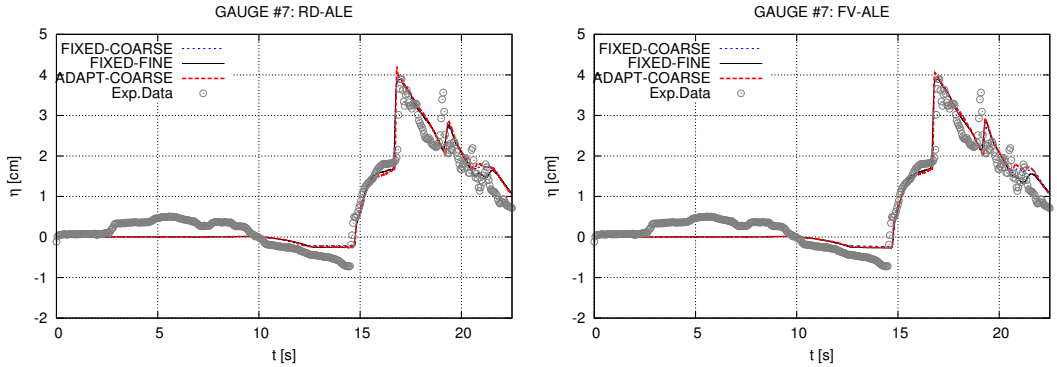

Figure 21: Okushiri experiment: total water height η signal registered at the gauges g7. Left: RD-ALE scheme. Right: FV-ALE scheme.

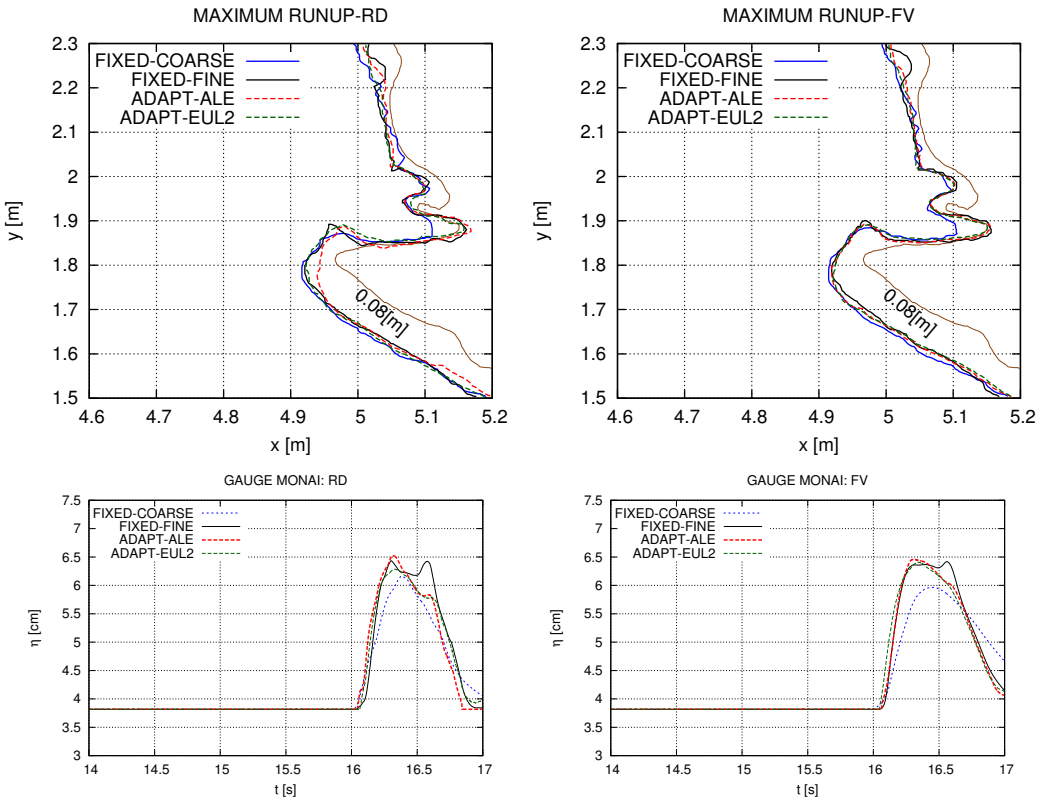

Figure 22: Okushiri experiment computed with adaptive ALE schemes. Top row: maximum runup for RD (left) and FV (right). Bottom row: total water height η signal registered at the gauge placed in the Monai valley. RD (left) and FV (right).

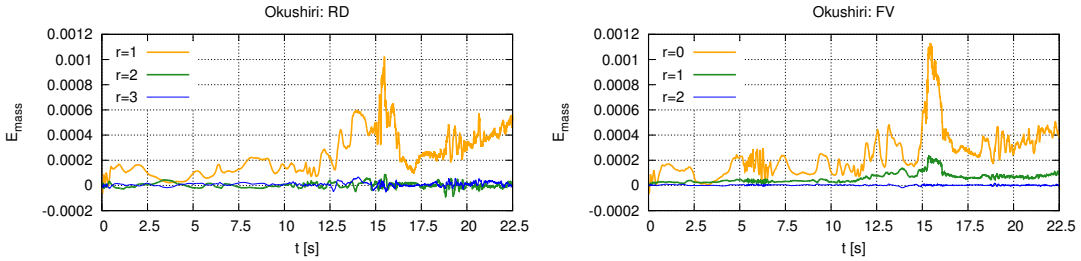

Figure 23: Okushiri experiment. Dimensionless mass error for different quadrature formula of the bathymetry integral.

ALG.	Mesh (Nodes)	RD[s] (%MMPDE)	FV[s](%MMPDE)
FIX-COARSE	7000	391.33	453.89
FIX-FINE	18711	2876.06	3301.62
ADAPT-ALE	7000	1179.23 (37.6%)	1466.02 (37.8%)
ADAPT-EUL1	7000	2930.10 (73.2%)	2454.45 (67.8%)
ADAPT-EUL2	7000	1408.25 (51.0%)	1565.96 (43.5%)

Table 4: Okushiri experiment. CPU times.

in 9 gauges placed upstream and downstream of the island, while velocities time series are provides in three gauges. For the set up of the test we refer to [52] (cf. also [53, 54]). We will compare flow velocity components in the exact location where an acoustic doppler velocimetry (ADV3) was installed. Two uniform meshes are used. The elements'size of the finer mesh is $h_K = 0.1\,[m]$ and has been prescribed in the TANDEM test case RS03 in order to compare different codes. For the coarse mesh we have choosen $h_K = 0.2\,[m]$. For this test case we used the following MMPDE parmeters: $\alpha = \delta = 40$ and $\beta = \gamma = 0.075$.

Figure 25 shows visualizations of the wave patterns arising from this complex interaction. In the figure, the top row shows the results obtained on the coarse grid. The second row reports the results on the fine grid. The ALE results, and the corresponding grids, are reported in the third and fourth row. Figure 26 shows visualizations comparing the ALE results (top half of the pictures, with snapshots of the video of the experiment, available online $[https://www.youtube.com/watch?v=I4uTHWBpaZg]$. The results are those obtained with the RD scheme, but very close ones are obtained with the FV method, not reported here due to shorten the presentation. The ALE results on the coarse mesh provide a flow description which is even clearer of the one obtained on the fine mesh, and clearly allows to resolve wave and vortical structures otherwise absent on the fixed coarse grid simulations. The comparison with the experimental snapshots shows a very satisfactory qualitative agreement with the patterns observed in the wave tank. Finally, figure 27 provides the time series in gauge ADV3. We can see that the adaptive simulation computes better resolved profiles of the waves reflected from the bar. The gain in time is between 40-50% with respect to using a reference mesh.

9 Conclusion

In this paper we have studied the use of r-adaptation for the shallow water simulation of complex wave interactions and wave runup on irregular bathymetries. We have provided a thorough theoretical setting to construct well balanced Finite Volume and Residual Distribution schemes on moving grids, including a mass conserving correction of the nodal bathymetric heights, based on a quadrature of the given bathymetric

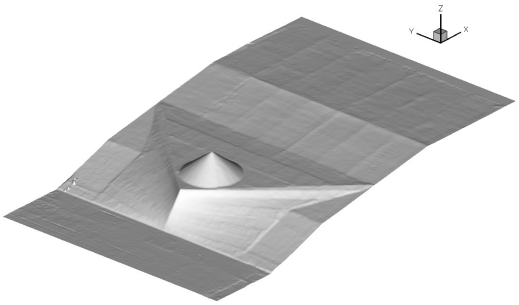

Figure 24: Solitary wave on a shelf: 3D visulization of the bathymetry, the scale of the $z-$axis is 5:1 with respect to x and y-axis

ALG.	Mesh (Nodes)	RD[s] (%MMPDE)	FV[s] (%MMPDE)
FIX-COARSE	32954	1772.22	1285.57
FIX-FINE	130439	15204.03	13707.38
ADAPT-ALE	32954	8735.69 (47.2%)	6358.76 (48.3%)

Table 5: Solitary wave on a shelf. CPU times.

data. Being based on the actual data, this correction requires no re-initialization, as e.g. the ALE remap used in [28] which leads to a numerical deviation from the real bathymetric data. We have coupled these schemes with a Laplacian-type r-adaptation method and investigated different coupling strategies in terms of accuracy and cost (CPU time). The delicate point is here the overhead of the mesh adaptation method when the flow solver is based on fully explicit multi-stage methods.

Besides confirming our theoretical expectations in terms of conservation of steady equilibria and mass conservation, our results show that, as long as possible, one should stick to the use of a fully ALE method coupled with the mesh deformation solver, used with a simplified solution remapping for the error sensor. This turns out to be the most efficient in terms of accuracy for a given CPU time, as well as the most robust in providing substantial improvements both for smooth and non-smooth features, including an improved prediction of runup. We have also proposed a simplified rezoning method which allows to run the flow solver on a fixed mesh. The method proposed allows to save significant CPU time and can be used in situations where local remeshing is necessary, and a full ALE method with finite time step values cannot be used. These results improve on, and complete the studies done in the past in e.g. [17, 21, 25, 28, 29, 45] providing quantitative as well as qualitative elements.

Future developments will involve the extension of our analysis to both multistep, and higher order methods, as well as the addition of dispersive effects, based on the approach of [55]. The reduction of the adaptation overhead obtained with the full ALE approach (and with the simplified rezoning) also opens the door to new developments. In particular, both the underlying PDE adaptation method, as well as its discretization, and iterative solution will be object of future work. A possible avenue is the combination of the simple Laplacian approach used here, providing a very sharp approximation of discontinuous features, with elastic deformation [56] allowing greater control on mesh quality. Concerning the discretization and resolution of the MMPDE, improved iterative methods can certainly be beneficial to produce grids with improved quality. In [17] it is mentioned that the algebraic system is solved with Gauss-Seidel iterations, but no quantitative informations whatsoever are given w.r.t. the cost overhead, or of the total cost of the adaptive simulations compared to a fixed fine mesh one. Other developments will include the extension to curved elements, in the context of high order methods, based on the use of local Bezier polynomials, following e.g. the initial work of [57].

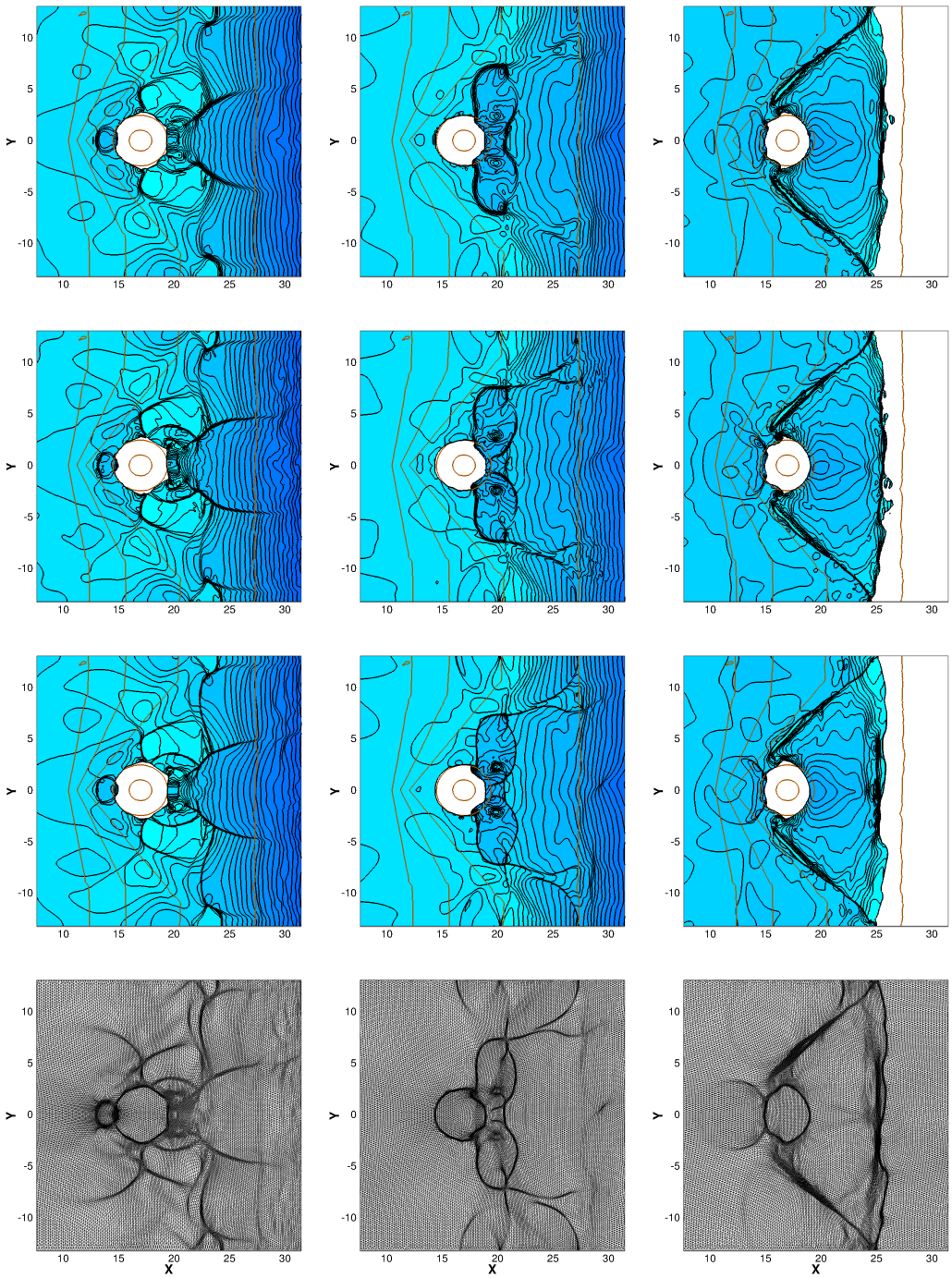

Figure 25: Solitary wave on a shelf (RD scheme). Solution isolines at $t = 0.24$, $t = 0.48$ are shown for fixed grid and adaptive computations. First row: fixed coarse grid. Second row: fixed fine grid. Third and fourth: adaptive ALE scheme.

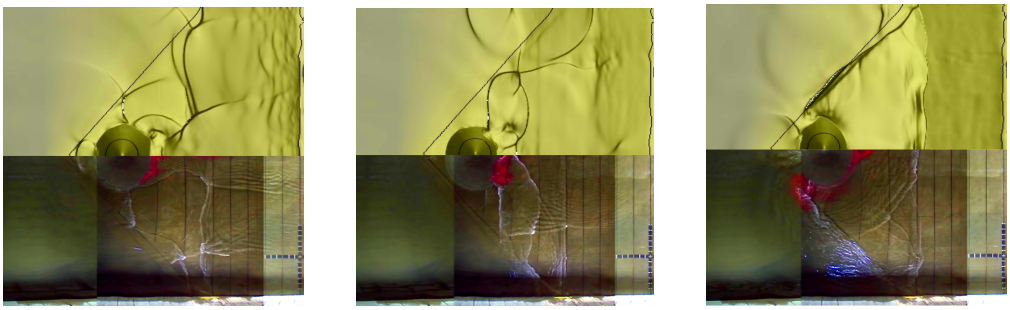

Figure 26: Solitary wave on a shelf (RD scheme). Comparison between numerical solution and experiment.

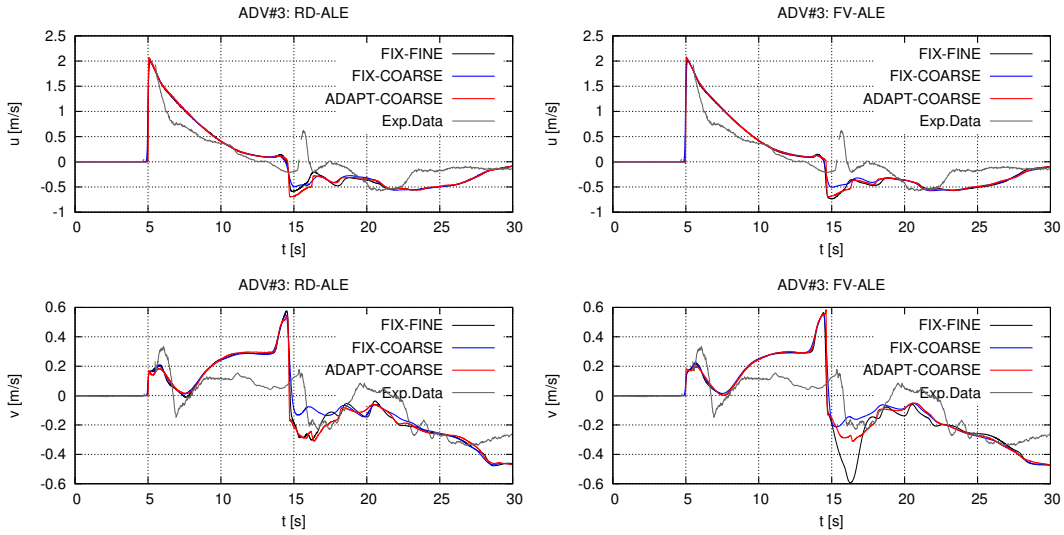

Figure 27: Solitary wave on a shelf: velocity components u, v registered at the gauge ADV3.

10 Acknowledgments

Work partially funded by the TANDEM contract, reference ANR-11-RSNR-0023-01 of the French Programme Investissements d'Avenir.

References

[1] A. Bermudez and M.E. Vazquez Cendon. Upwind methods for hyperbolic conservation laws with source terms. *Comput. Fluids*, 235(8):1049–1071, 1994.

[2] M.E. Hubbard and P. Garcia-Navarro. Flux difference splitting and the balancing of source terms and flux gradients. *J. Comput. Phys*, 165:89–125, 2000.

[3] E.F. Toro and P. Garcia Navarro. Godunov-type methods for free surface shallow water flows: a review. *J.Hydraul.Res*, 45:746, 2007.

[4] G. Russo. Central schemes for balance laws. In Heinrich Freistauer and Gerald Warnecke, editors, *Hyperbolic Problems: Theory, Numerics, Applications*, volume 141 of *ISNM International Series of Numerical Mathematics*, pages 821–829. Birkhauser Basel, 2001.

[5] A. Kurganov and D. Levy. Central-upwind schemes for the Saint-Venant system. *ESAIM: M2AN*, 36(3):397–425, 2002.

[6] A. Kurganov and G. Petrova. A second-order well-balanced positivity preserving central-upwind scheme for the Saint-Venant system. *Commun. Math. Sci.*, 5:133–160, 2007.

[7] B.D. Rogers, A. Borthwick, and P. Taylor. Mathemathical balancing of flux gradient and source terms prior to using roe's approximate riemann solver. *J.Comput.Phys.*, 192:422–451, 2003.

[8] Q. Liang and A.G.L. Borthwick. Adaptive quadtree simulation of shallow flows with wet-dry fronts over complex topography. *Comput. Fluids*, 38:221–234, 2009.

[9] Q. Liang and F. Marche. Numerical resolution of well-balanced shallow water equations with complex source terms. *Adv. Water Resour.*, 32:873–884, 2009.

[10] P. Brufau and P. Garcia-Navarro. Unsteady free surface flow simulation over complex topography with a multidimensional upwind technique. *J. Comput. Phys*, 186(2):503 – 526, 2003.

[11] M. Ricchiuto, R. Abgrall, and H. Deconinck. Application of conservative residual distribution schemes to the solution of the shallow water equations on unstructured meshes,. *J. Comput. Phys.*, 222:287–331, 2007.

[12] M. Ricchiuto and A. Bollerman. Stabilized residual distribution for shallow water simulations. *J.Comput.Phys*, 228:1071–111, 2009.

[13] M. Ricchiuto. On the c-property and generalized c-property of residual distribution for the shallow water equations. *Journal of Scientific Computing*, 48:304–318, 2011.

[14] M. Ricchiuto. An explicit residual based approach for shallow water flows. *J.Comput.Phys.*, 80:306–344, 2015.

[15] F. Alauzet, P. Frey, P.L. George, and B. Mohammadi. 3d transient fixed point mesh adaptation for time dependent problems. application to cfd simulations. *J. Comp. Phys.*, 222:592–623, 2007.

[16] D. Isola, A. Guardone, and G. Quaranta. Arbitrary lagrangian eulerian formulation for two-dimensional flows using dynamic meshes with edge swapping. *J. Comp. Phys.*, 230:7706–7722, 2011.

[17] H. Tang and T. Tang. Adaptive mesh methods for one and two-dimensional hyperbolic conservation laws. *SIAM J. Numer. Anal.*, 41(2):487–515, 2003.

[18] C.J. Budd, W. Huang, and R.D. Russell. Adaptivity with moving grids. chapter 1, pages 1–131. Cambridge University Press, 2009.

[19] J.F. Thompson and N.P. Weatherill. Fundamental concepts and approaches. In J.F. Thompson, B.K. Soni, and N.P. Weatherill, editors, *Hanbook of Grid Generation*, chapter 1, page 10. CRC Press, 1999.

[20] A.M. Winslow. Numerical solution of the quasi-linear poisson equation. *J. Comput. Phys*, 1:149–172, 1967.

[21] H.D. Ceniceros and T.Y. Hou. An efficient dynamically adaptive mesh for potentially singular solutions. *J. Comput. Phys*, 172:609–639, 2008.

[22] C.J. Budd, M.J.P. Cullen, and E.J. Walsh. Monge-ampere based moving mesh methods for numerical weather prediction, with applications to the eady problem. Technical report, 2012.

[23] W.Huang. Anisotropic mesh adaptation and movement. Workshop on Adaptive Method, Theory and Application, June 20 – August 20, 2005.

[24] J.U. Brackbill and J.S. Saltzman. Adaptive zoning for singular problems in two dimensions. *J. Comput. Phys*, 46:342, 1982.

[25] X.Xu, G.Ni, and S.Jiang. A high-order moving mesh kinetic scheme based on weno reconstruction for compressible flows on unstructured meshes. *J. Comp. Phys.*, 57:278–299, 2013.

[26] Z.J. Wang and H.Q Yang. Unsteady flow simulation using a zonal multigrid approach with moving boundaries. *AIAA Paper*, 94-0057, 1994.

[27] M. Lesoinne and C. Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Comput. Method Appl. M.*, 134(1-2):71–90, 1996.

[28] F. Zhou, G. Chen, Y. Huang, J.Z. Yang, and H. Feng. An adaptive moving finite volume scheme for modelling flood inundation over dry and complex topography. *Water Resour. Res.*, 49:1914–1928, 2013.

[29] F.Zhou, G.Chen, S.Noelle, and H.C.Guo. A well-balanced stable grp scheme for shallow water equations for adaptive unstructured triangular meshes. *Int J Numer Meth Fl*, 73:266–283, 2013.

[30] J.Donea. Arbitrary lagrangian eulerian finite element methods. In *Computational Methods for Transient Analysis*, chapter 10. Elsevier Science Publisher, Amsterdam, 1983.

[31] I.K.Nikolos and A.I.Delis. An unstructured node-centered finite volume scheme for shallow water flows with wet/dry fronts over complex topography. *Comput.Methods Appl.Mech.Engrg*, 198:3723–3750, 2009.

[32] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction or shallow water flows. *SIAM J. Sci. Comput.*, 25(6):2050– 2065, 2004.

[33] L. Arpaia, M. Ricchiuto, and R. Abgrall. An ale formulation for explicit runge-kutta residual distribution. *J. Sci. Comput.*, 190(34):1467–1482, 2014.

[34] D.Isola. *An Interpolation Free Two-Dimensional Conservative ALE scheme over Adaptive Unstructured Grids for Rotorcraf Aerodynamics*. PhD thesis, Politecnico di Milano. Dipartimento di Ingegneria Aerospaziale, 2012. Sec. 2.4.1.

[35] A.I. Delis, I.K. Nikolos, and M. Kazolea. Performance and comparison of cell-centered and node-centered unstructured finite volume discretizations for shallow water free surface flows. *Archives of Computational Methods in Engineering*, 18(1):57–118, 2011.

[36] R.J. LeVeque. High-resolution methods. In *Finite Volume Methods for Hyperbolic problems*. Cambridge University Press, 2004.

[37] P. Brufau, M.E. Vazquez-Cendon, and P. Garcia-Navarro. A numerical model for the flooding and drying of irregular domains. *Int. J. Numer. Meth. Fluids*, 39:247–275, 2002.

[38] P. Brufau, P. Garcia-Navarro, and M.E. Vazquez-Cendon. Zero mass error using unsteady wetting-drying conditions in shallow flows over dry irregular topography. *Int. J. Numer. Meth. Fluids*, 45:1047–1082, 2004.

[39] 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, 2008.

[40] M. Ricchiuto and R. Abgrall. Explicit Runge-Kutta residual distribution schemes for time dependent problems: Second order case. *J. Comput. Phys*, 229(16):5653 – 5691, 2010.

[41] T.J.R. Hughes and A. Brook. Streamline upwind Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Engrg.*, 32:199–259, 1982.

[42] T.J.R. Hughes, G. Scovazzi, and T. Tezduyar. Stabilized methods for compressible flows. *J. Sci. Comp.*, 43:343–368, 2010.

[43] R. Abgrall. Essentially non oscillatory residual distribution schemes for hyperbolic problems. *J. Comput. Phys*, 214(2):773–808, 2006.

[44] M.J. Baines and M.E. Hubbard. Multidimensional upwinding for grid adaptation. *Numerical Methods for Wave Propagation*, page 33–54, 1998.

[45] G. Chen, H. Tang, and P. Zhang. Second-order accurate godunov scheme for multicomponent flows on moving triangular meshes. *J. Sci. Comput*, 34:64–86, 2008.

[46] R. Li, T. Tang, and P.Zhang. A moving mesh finite element algorithm for signular problems in two and three space dimensions. Technical report.

[47] J.A.Mackenzie and W.R.Mekwi. On the use of moving mesh methods to solve pdes. In T. Tang and J. Xu, editors, *Adaptive Computations: Theory and Algorithms*, page 242–278. Science Press, Beijing, 2007.

[48] M. Seaid. Non-oscillatory relaxation methods for the shallow water equations in one and two space dimensions. *Int. J. for Numerical Methods in Fluids*, 46:457–484, 2004.

[49] M.J. Briggs, C.E. Synolakis, G.S.Harkins, and D.R. Green. Laboratory experiments of tsunami runup on a circular island. *Pure Appl. Geophys.*, 144:569–593, 1995.

[50] P. L.-F. Liu, H. Yeh, and C. Synolakis, editors. *Advanced Numerical Models for Simulating Tsunami Waves and Runup*, volume 10 of *Advances in Coastal and Ocean Engineering*. World Scientific, 2008.

[51] NOAA Center for tsunami research. Tsunami runup onto a complex three-dimensional beach; monai valley. `http://nctr.pmel.noaa.gov/benchmark/Laboratory/Laboratory_MonaiValley/index.html`.

[52] P. Lynett, D. Swigler, S. Sangyoung, D.Bryant, and S. Socolofsky. Experimental study of a solitary wave evolution over a 3d shallow shelf. *Proc. 32th Conf. Coast. Engng.*, page 813, 2010.

[53] V. Roeber and K.F. Cheung. Boussinesq-type model for energetic breaking waves in fringing reef environments. *Coast. Eng.*, 70:1–20, 2012.

[54] M. Kazolea, A.I. Delis, and C.E. Synolakis. Numerical treatment of wave-breaking on unstructured finite volume approximations for extended boussinesq-type equations. *J.Comput.Phys*, 271:281–305, 2014.

[55] A.G. Filippini, M. Kazolea, and M. Ricchiuto. A flexible genuinely nonlinear approach for nonlinear wave propagation, breaking and run-up. *J.Comput.Phys*, 310:381–417, 2016.

[56] K. Stein, T.E. Tezduyar, and R. Benney. Automatic mesh update with the solid-extension mesh moving technique. *Comput. Method Appl. M.*, 193:2019–2032, 2004.

[57] R. Abgrall, C. Dobrzynski, and A. Froehly. A method for computing curved meshes via the linear elasticity analogy, application to fluid dynamics problems. *Int. J. Numeric. Meth. Fl.*, 76(4):246–266, 2014.