Adaptive techniques for free surface flow simulations: Application to the study of the 2011 Tohoku Tsunami.

Luca Arpaia

To cite this version:

Luca Arpaia. Adaptive techniques for free surface flow simulations: Application to the study of the 2011 Tohoku Tsunami.. Numerical Analysis [math.NA]. Université de Bordeaux, 2017. English. NNT: 2017BORD0666. tel-03200417

HAL Id: tel-03200417

https://theses.hal.science/tel-03200417v1

Submitted on 16 Apr 2021

THÈSE

PRÉSENTÉE À

L'UNIVERSITÉ DE BORDEAUX

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET D'INFORMATIQUE

par **Luca Arpaia**

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : INFORMATIQUE

Techniques d'adaptation robustes pour la simulation des ondes de surface. Application à l'étude du tsunami Tohoku 2011

Date de soutenance : 18 Septembre 2017

Devant la commission d'examen composée de :

Mario Ricchiuto	Pr., Université de Bordeaux	Directeur de thése
Philippe Bonneton	Pr., Université de Bordeaux	Co-directeur de thése
Alina Chertock	Pr., North Carolina State University	Rapporteur
Carlos Pares	Pr., Universidad de Málaga	Rapporteur
Frédéric Golay	Pr., Université de Toulon	Examinateur
Rodrigo Pedreros	Dr., BRGM	Examinateur
Ata Ryadh	Dr., EDF, LHSV	Examinateur
Cecile Dobrzynski	Pr., Université de Bordeaux	Examinateur

– 2017 –

Résumé Dans cette thèse, nous implémentons les équations de Saint Venant (SV), ou de Shallow Water, sur des grilles non structurées afin de simuler des écoulements de surface libre sur des bathymétries irrégulières, incluant inondation et d'autres phénomènes complexes qui se produisent généralement dans des applications hydrodynamiques. En particulier, nous voudrions simuler avec précision les tsunamis, la propagation d'onde à grande échelle jusqu'à l'inondation très localisé. À cette fin, nous utilisons deux méthodes qui sont comparées en profondeur le long du manuscrit: la méthode des volumes finis, très populaire dans la communauté hydrodynamique et hydraulique et une technique plus récente appelée Distribution du Résidu appartenant à la classe des schémas upwind multidimensionnels. Pour améliorer la résolution de certaines caractéristiques de l'écoulement telles que le développement du déferlement et les inondations à petite échelle, nous utilisons une adaptation de maillage dynamique basée sur une redistribution des nœuds de maillage, aussi appelé adaptation de type r (r signifiant "relocalisation"). La combinaison appropriée de cette méthode avec le solveur SV est généralement appelée Méthode de Maillage Mobile. Parmi les nombreux algorithmes de maillage mobile disponibles, nous proposons une forme Arbitrary Lagrangian Eulerian (ALE) des équation SV qui permettent de faire évoluer les variables de flux d'une maille à l'autre de manière élégante. Dans ce contexte, nous soulignons les principales contributions de la thèse:

1. Nous montrons l'importance de conserver toutes les propriétés standards d'un solveur Eulérien SWE tel que la préservation du lac au repos et la conservation de la masse également sur des maillages en mouvement.
2. Notre couplage ALE est comparé à l'approche de rezoning, avec une légère augmentation de la performance globale de l'algorithme en termes de précision et de temps CPU.
3. Nous étendons l'approche ALE sur la sphère afin d'inclure l'effet de la courbure terrestre dans la dynamique de propagation des ondes à grande échelle du tsunami.
4. la simulation du tsunami 2011 de Tohoku-Honsu devrait prouver que la méthode de maillage mobile étudiée dans la thèse, bien que simple, pourrait être un bon candidat pour réduire le coût de calcul des simulations de tsunami.

Title Adaptive techniques for free surface flow simulations. Application to the study of the 2011 Tohoku Tsunami.

Abstract In this thesis we implement the Shallow Water equations (SWEs) on unstructured grids in order to simulate free surface flow over irregular bathymetries, wetting/drying and other complex phenomena that typically occurs in hydrodynamic applications. In particular we would to accurately simulate tsunami events, from large scale wave propagation up to localized runup. To this aim we use two methods that are extensively compared along the manuscript: the Finite Volume method, which is very popular in the hydrodynamics and hydraulic community and a more recent technique called Residual Distribution which belongs to the class of multidimensional upwind schemes. To enhance the resolution of important flow feature such as bore development or small scale flooding, we use a dynamic mesh adaptation based on a redistribution of mesh nodes or r-adaptation (r stands for "relocation"). The proper combination of this method with the flow solver is usually referred to as Moving Mesh Method. Among the many different moving mesh algorithms available we propose an Arbitrary Lagrangian Eulerian (ALE) form of the SWEs which elegantly permit to evolve the flow variables from one mesh to the updated one.

Keywords Shallow Water Equations, Arbitrary Lagrangian Eulerian form, Finite Volume, Multidimensional Upwind Schemes, tsunami simulations, moving mesh method, *r*-adaptation.

Mots-clés Équation de Saint-Venant, Description Arbitraire Lagrangienne Eulérienne, Volume Finis, Méthode d'upwind multidimensionel, adaptation du maillage de type-*r*.

Laboratoire d'accueil INRIA Bordeaux Sud-Ouest, 200 Avenue de la Vieille Tour, 33405 Talence

Acknowledgments

Contents

Introduction

Context of the thesis: the TANDEM project

The 11 March 2011 an earthquake of M_w 9.1 with epicenter off the coast of Tohoku triggered a tsunami on the eastern coast of the Honsu island. The devastating wave has hit the coastal prefectures of Iwate, Miyagi and Fukushima. It was the tsunami attack to cause the major part of damages and fatalities (the Fire and Disaster Management Agency counted for 16,278 fatalities and 2,994 missing in Tohoku and Kanto regions). Among the damages, the accident of the Fukushima Daichi Nuclear Power Plant (NPP) was the one that mostly impressed the public opinion worldwide. In the Fukushima site, the *designed tsunami height*, whose computation was based on a standard procedure Yanagisawa *et al.* [2007], was 6.1 $[m]$. The 11 March 2011 a tsunami wave 16 $[m]$ high hit the Fukushima Daichi NPP. The water flooded in the site, causing the loss of the cooling system which was followed by the hydrogen explosion of the reactors and by the subsequent propagation of radioactive material. In the days that followed Japan government declared nuclear emergency and thousands of people have been evacuated due to elevated radiation risk, evoking a scenario similar to the Chernobyl accident in 1986.

This event mobilized the International Atomic Energy Agency (IAEA) which asked for new security standards concerning the NPP located along coastal regions. In France the design of new guidelines was particularly urgent: 19 NPP are active and 5 of them (Blaye, Flamanville, Paluel, Penly and Gravelines) are right on the Atlantic coast. In this context, the French National Research Agency (ANR) launched the *TANDEM project* (Tsunamis in the Atlantic and the English ChaNnel Definition of the Effects through numerical Modeling) which has funded the present thesis. The objective was to study the 2011 tsunami by numerical modeling and to define the tsunami hazard for the French coastal regions. 10 French partners were involved: among them there were industrial companies (Electricté de France, Principia) and French research institutions (University of Pau, BRGM, SHOM, Ifremer, IRSN and INRIA). The project was structured in four parts or work-packages (WP):

- WP1 consists in a set of benchmarks to validate, compare and improve the different hydrodynamic softwares available among the partners. The idea was similar to the one of the Workshop on Long Wave Runup Model IIW-LWRM [1995]: collect a series of existing laboratory tests for long wave propagation and runup and check the ability of the hydrodynamic solver (both the physical model and numerical method) to well reproduce the experiments.

- WP2 studied the influence of uncertain parameters in the computation of tsunami heights and flooded areas. Sensitivity analysis on the computational and physical parameters was conducted in order to provide a quantifications of them. These includes mesh definitions as well as the seismic source.
- WP3 was dedicated to the simulation of the 2011 Tohoku tsunami which is widely recognized to be the largest instrumentally recorded earthquake in Japan history. This fact will permit us: 1) to recover realistic initial waveforms from seismic/tsunami data inversion 2) to compare the numerical simulation against high reliable data.
- WP4 provided an inventory of all the possible tsunamigenic sources for the French coast, the definition of possible tsunami scenario and their numerical simulation.

This manuscript contributes to WP1, WP2 and WP3. The main objective of the thesis consists in increasing the accuracy of hydrodynamic (and in particular tsunami) simulations through a mesh adaptation strategy, rather then employ very high order schemes. This choice is motivated by the fact that, for large scale tsunami simulations, static grids are inherently not efficient. The tsunami wave is localized in a certain area of the ocean, then crosses all the ocean basin, shoals, steepens near the coastlines and inundates the land. To enhance important features in tsunami simulations such as the resolution of wave patterns during propagation and shoaling or the runup/rundown stages in which fine scale flooding occurs, mesh adaptation could be the ideal candidate. In this thesis mesh adaptation will be achieved by redistributing (or relocating) the same number of mesh nodes within the domain.

State of the art

Tsunami science and tsunami modeling

Tsunami science is a young science. A systematic study of the subject was addressed only in the last fifty years. In the first studies (sixties) tsunamis were approximated as a one dimensional long wave that arises after an impulsive seafloor displacement. To develop a better understanding of the tsunami amplification, shoaling and runup in coastal areas, much attention was put in the derivation of analytical solutions. The physical model was quite simple: a long wave (a periodic wave, solitary waves or N-shaped waves) travels on a constant depth before reaching a sloping beach on which runup and reflection occur. Even if simple, there were important unresolved questions such as the role of non-linearity and dispersion in the wave transformation. The simplest depth averaged model describing long wave propagation is the non linear Shallow Water equations (in this thesis abbreviated as SWEs). SWEs are derived from the conservation of mass and momentum under the *hydrostatic assumption*. It is assumed that the horizontal length scale of the flow is much larger then vertical one (the scale of the water depth); the consequence is that the vertical dynamics can be neglected. Carrier and Greenspan [1958] resolved the SWEs for various waveforms climbing a sloping beach. Later Synolakis [1987] compared the analytical solution for a solitary wave with the experimental water level profile created in a wave tank: he found that the SWEs approximated very nicely the long (and non breaking) wave profiles all along the runup process. It was also found that this set of equations correctly describes non linear wave transformation and wave breaking. The success of the SWEs made them the most commonly long wave approximation for runup calculation. It was also recognized that the lack of wave dispersion could give wrong wave propagation results in deep water and shoaling. To include these effects, the Boussinesq-type equations received considerable attention, see Peregrine [1967]; Sorensen and Madsen [1992]; Nwogu [1994]. These effects are only relevant locally and for landslide tsunamis. Here we will consider the SWEs as our set of governing equations.

In the seventies we assisted to the advent of numerical methods to approximate the long waves equations. Finite Difference and Finite Volume, originally developed for gasdynamics started to be applied to the simulation of two dimensional wave propagation and runup on irregular bathymetries.

After the 1992 Flores tsunami and the 1993 Okushiri tsunami, a large tsunami experiment was conducted by Briggs *et al.* [1995]. The physical model consisted of a solitary wave hitting a a small scale reproduction of a conical island that approximated the shape of the Babi island, strongly impacted by the Flores tsunami. For the Okushiri event, an extreme tsunami runup ($30\,[m]$) was discovered in the Monai valley. This motivated a second tsunami experiment that was performed at the CRIEPI laboratory (Japan) where the complex three dimensional beach facing the Monai valley was reproduced with high accuracy. We mention this two experimental set-up because nowdays they become standard test-case for every numerical method approximating long waves, included the SWEs solver that we propose in this thesis.

From the nineties the community of tsunami modelers start to meet in Unites States at the Workshop on Long Wave Runup Models (IIW-LWRM [1995] and IIIW-LWRM [2004]). In these occasions, improved physical models and more recent numerical methods that were developed meanwhile, were tested and analyzed against experimental data in order to obtain reliable numerical tools with precise range of applicability. It was also realized that tidal gauges where unfit to record a tsunami wave. A new system of tsunamometers buoys to record tsunami height in deep ocean was implemented with an appropriate frequency response. In two thousands a tsunami forecasting system was operational in the Pacific Ocean. In 2003 there was the first warning cancellation in Hawaii. After the megathrust earthquake of 2004 in Sumatra, numerical modeling was mature to robustly simulate the complete tsunami event, from generation to propagation and runup. Numerical models were able to explain the large runups and destruction observed in coastal Thailand Ioualalen *et al.* [2007]. When this thesis started in 2014, there were already many published works on the numerical modeling of the Tohoku-Honsu tsunami, see Chen *et al.* [2014]; Løvholt *et al.* [2012]; Shimozono *et al.* [2012].

Numerical methods in hydrodynamics

SWEs are of hyperbolic type, eventually with source terms to take into account different effects such as bathymetry and friction. The hyperbolic nature of the underlying equations have permitted to transfer earlier upwind Finite Difference and Finite Volume (FV) methods, originally developed for the compressible Euler equations of gasdynamics in the aeronautical community, to hydrodynamics.

It was soon realized that there were specific issues. First and differently from gasdynamics, there was the problem of embedding at a discrete level exact solutions more complex then the trivial uniform flow with constant depth. The lake at rest, which results from the physical balance between hydrostatic pressure and the pressure force played by the bottom on the fluid, emerged as one of the most important state to be preserved numerically. This was a way to prevent the unexpected rise of numerical oscillation in regions where the flow is at rest. Such schemes are called *Well Balanced* (WB). The work of Bermudez and Vazquez-Cendon [1994] was pioneering in this area since it established the link between upwinding of the source term and Well Balanced property. This initial work in one dimension, has been led throughout the years to many different results allowing the construction of unstructured mesh discretizations verifying the WB via an appropriate coupling of the numerical flux and numerical source terms Hubbard and Garcia-Navarro [2000], or based on different forms of the equations, as the well-balanced form of Russo [2001]; Kurganov and Levy [2002]; Chertock *et al.* [2017], or the so-called *pre-balanced* form of Rogers *et al.* [2003]; Liang and Borthwick [2009]; Liang and Marche [2009]. A review on Well Balanced can be found in Castro *et al.* [2017].

In coastal engineering, modeling the moving shoreline is a second important issue. At the beginning, the shoreline was treated as a boundary condition which is changing with time and it should be determined by the computations itself. An alternative way, that proved to be simpler, is to include the wet/dry interface in the computation, introducing the presence of cells of mixed type where both dry and wet nodes appears. This method was first employed for Finite Difference by Sielecki and Wurtele [1970]. Many years after, a robust and effective treatment of wetting/drying was set in Brufau *et al.* [2002]. In a Finite Volume context the situation with cells of mixed type can be treated as a Riemann problem for which one state is wet and one state is dry, the solution consisting of a rarefaction wave advancing in the dry region and causing flooding. However unphysical oscillation may appear at the front especially for the flow speed: the flow velocity in SWEs code is typically computed as the ratio between discharge and depth, both small quantities at the shore, and this lead to not accurate values. A depth threshold value is typically fixed avoiding the computation of large velocities. Additional problems arise when bed slope is present. It was recognized the difficulty of recovering the lake at rest for WB schemes on a general irregular topography, since oscillation appeared at the shore. Modifying the difference in the bathymetry level at the interface to recover the equilibrium condition is a possible solution.

Another progress was the use of unstructured grids, which appears useful also in hydrodynamics. Their inherent advantages respect to Cartesian grids consist in the possibility to construct body fitted mesh and the flexibility when exploiting local mesh refinement. Unstructured grids are characterized by the absence of preferential direction. We recall that the Godunov scheme, in two dimensions, is based on a direct extension of the one dimensional scheme. However, while in one dimension the cell normals are always aligned to the wind direction, in two dimensions and unstructured grids this could be no longer true and upwinding is created also in the crosswind direction to the flow. For this reason, there was much interest to construct multidimensional upwind scheme with optimal diffusion property, see Roe and Sidilkover [1992]. The idea was to construct schemes directly in two dimensions, embedding the multidimensional character of the physics into the numerical methods. This was possible after that Roe [1982] had presented an alternative interpretation of the one dimensional FV for linear advection. This work shed a new light in the comprehension of upwinding and it was exploited to construct a class of *genuinly multidimensional* scheme that are nowdays referred to as *Residual Distribution* (RD). After gasdynamics Deconinck *et al.* [2000], RD were successfully extended to the SWEs by Paillere *et al.* [1998] and tested against standard runup benchmarks in Ricchiuto and Bollerman [2009].

Very high order of accuracy is the next challenge for the hydrodynamic codes. Although in this thesis we will stick to second order of accuracy, which is standard nowdays for commercial codes, we mention the increasing attention that the Discontinuous Galerkin and WENO methods have received in the last years. A review of very high order schemes obtained with these two methods can be found in Xing and Shu [2014].

Moving Mesh Methods

h-refinement is the most widely used technique to refine locally the mesh size. The idea is very simple: based on some *a posteriori* error estimates, the mesh is locally refined or coarsened adding or deleting points from the mesh. Originally this technique was developed for structured Cartesian grids by Berger and Colella [1989] under the name of Adaptive Mesh Refinement (AMR). Nowdays AMR tools are implemented in many geophysical codes such as `GeoClaw` Berger *et al.* [2011] and `Gerris Flow` Popinet [2011]. They have been already tested to improve 2011 Tohoku tsunami simulation accuracy in MacInnes *et al.* [2013] and Popinet [2012].

For unstructured grids, h-refinement (node insertion/deletion) can be performed by appropriate local remeshing strategies, with impressive results shown in Alauzet *et al.* [2007]; Isola *et al.* [2011]. In this thesis we have focused on a more recent strategy of mesh adaptation for unstructured grids based on nodes redistribution (or relocation). These are known as r-adaptation techniques. We invite the interested reader to see the recent review of Budd *et al.* [2009a]. Roughly speaking, these methods, always based on a posteriori error estimates, cluster the points of a given reference mesh, keeping the mesh topology and number of mesh points unchanged. Now days they are widespread in the CFD community including aerodynamics Li and Petzold [1997], magnetohydrodynamics Tan [2007], multiphase flow Chen *et al.* [2008], meteorology Budd *et al.* [2009b] and recently they have been applied also to hydrodynamics Zhou *et al.* [2013b]. The central idea is the one dimensional equidistribution principle of de Boor [1973]. We search for a time dependent transformation $x = M(X, t)$ from a computational domain described by coordinate X to a physical domain described by coordinate x. The map M should be computed such that it equidistributes a *monitor function* (error estimates) m on the reference domain, that is $m\Delta x = \Delta X$. Given a uniform computational grid $\Delta X = const$, if m is large (large error) the grid space Δx is automatically refined.

The main reasons behind the choice of r-adaptation can be summarized:

- Although very powerful, remeshing techniques requires much higher overheads and a very complex data structures. This results in a higher computational cost, especially compared to a single step of an explicit discretization of the SWEs (as the one we will use in this thesis). On the contrary, in r-refinement, nodal movement is obtained by solving an appropriate Moving Mesh Partial Differential Equation (MMPDE) whose computational cost is of the same order of the CFD solver.

- the potential shown in the past for these techniques to capture shocks, boundary layers and singularities. The MMPDE of Ceniceros and Hou [2001], which is the one implemented in this thesis, merged this advantages with a very efficient implementation consisting in the resolution of a sparse non-linear system.

- efficient conservative/accurate remaps of flow variables can be carried out from one grid to the updated one Tang and Tang [2003]

We have to mention that r-adaptation is less developed and less understood with respect to h-adaptation; The fact that mesh topology is fixed poses several limitations: as stated in Budd *et al.* [2009a] it may never be possible to resolve all of the fine structures of a PDE as it evolves. Moreover great care has to be put in the definition of the monitor function and in the choice of the number of mesh points, since there is the serious risk that the nodes migrate to regions with high gradient, leaving an extreme low number of mesh points to approximate smooth structure of the flow. Mesh tangling is another issue that we will discuss and that deserves further studies.

In this thesis we will take an existing MMPDE (we said, the one of Ceniceros and Hou [2001]) and we discuss extensively the coupling of the SWEs (or a general flow solver) with this MMPDE. In particular the SWEs and the MMPDE can be either solved *simultaneously* or *alternately*. The latter has been successfully implemented by Huang and Russell [1999]. Historically, depending on the framework in which we evolve the SWEs, two different *alternate* algorithms were developed and they will be tested in this thesis.

The first is the *rezoning* method introduced in Tang and Tang [2003]. The SWEs are written in Eulerian framework and they are are solved on the last updated fixed mesh at each time iteration. The advantage is that the CFD solver can be treated as a black box; the drawback is that it requires a remap/interpolation of the flow variables on the updated mesh from the previous mesh. This operation 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, Well Balancedness, mass conservation).

At the opposite, one can evolve the flow with an Arbitrary-Lagrangian-Eulerian (ALE) formulation of the governing equations, as suggested e.g. in Huang and Russell [1999]; Ni *et al.* [2009]. SWEs are rewritten in an arbitrary moving reference framework which follows mesh movement, see Donea [1983]. The resulting scheme is interpolation free and 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 is produced in the computational domain due to mesh motion. The discrete counterpart of this property is known as the DGCL (cf. Thomas and Lombard [1979], Etienne *et al.* [2009] for an overview). Ideally, in Shallow Water flows, we have to ensure the satisfaction of both the DGCL, and of the WB, while still being able to conserve mass and momentum. Up to the our knowledge, this issue has not been addressed yet and it widely discussed in this manuscript.

Tsunami simulation and earth curvature effects

There is a last key ingredient for performing large scale tsunami simulations: earth curvature effects. In this thesis we have chosen to set properly the SWEs in standard latitude-longitude curvilinear coordinates. This is the standard model for numerical weather prediction and ocean modeling. Within this community, Finite Volume on the sphere has already received considerable attention, in the context of both *Semi-Lagrangian* methods and *Eulerian* methods, see the review of Machenhauer *et al.* [2009]. In this thesis we will refer to the latter class of schemes: the Eulerian FV based on the flux form of the SWEs implemented in Ullrich *et al.* [2010]; Rossmanith *et al.* [2004]. Recently Residual Distribution have been extended to curvilinear coordinates only for steady problems Rossmanith [2013]. They appear to us very interesting due to their simplicity to include the spherical geometry starting from a code in Cartesian coordinate. We stress that our objective is the inclusion of earth curvature in the tsunami dynamics. For this reason we don't need a global scale modeling which means that the complication of the poles, where the coordinate transformation become singulars, will be neglected.

ALE-SWEs are not much discussed in the literature. Although ALE balance law on manifold were already presented in Savidis *et al.* [2008], to the best of the our knowledge this thesis represent the first attempt to set the SWEs in ALE form on the sphere.

In geophysics mesh adaptation is gaining increasing popularity. Atmospheric motion is a multiscale phenomenon and it is very hard, even with modern supercomputers to capture all the scale of interest in a single computation. Since the works of Skamarock and Klemp [1992]; Behrens [1996], h-adaptation on the sphere has strongly developed through AMR techniques, see for example the recent McCorquodale *et al.* [2015]. Recently also r-adaptation was reconsidered in order to increase the resolution of PDEs on the sphere. It was recognized that the problem is much harder with respect to r-adaptation in Cartesian coordinates. The main difficulty is related to find a unique map from the computational sphere to the physical sphere Di *et al.* [2006]; Weller *et al.* [2016]. We will bypass the problem taking advantage of the fact that our domain is only a part of the whole sphere. We will discuss a naif approach consisting in mapping a portion of the sphere into a plane.

Main results of the thesis

We highlight directly the main contribution of this thesis. In the next section we will detail each point with respect the structure of the manuscript.

1. We have developed an ALE Residual Distribution and Finite Volume scheme for the SWEs on moving mesh. These schemes are Well-Balanced and mass conserving.

2. The ALE approach has been tested (on standard long wave benchmarks) against the popular rezoning method, showing a slight increase of the overall performances in term of accuracy and CPU time. We have proposed an improved version of the rezoning algorithm which uses a simple linear advection scheme to interpolate the error estimate.

3. We have extended the ALE Residual Distribution and ALE Finite Volume for the SWEs to curvilinear coordinates.

4. We have run a moving mesh simulation of the 2011 Tohoku-Honsu tsunami.

In the following we provide also a list of the publications related to this manuscript:

1. L. A., M. Ricchiuto. ALE r-adaptive methods for the Shallow Water equations in curvilinear coordinates (in preparation).

2. L. A. and M. Ricchiuto. r-adaptation for Shallow Water flows: conservation, well balancedness, efficiency. (accepted at *Computers and Fluids*).

3. L. A., M. Ricchiuto and R. Abgrall. An ALE formulation for explicit Runge-Kutta Residual Distribution. *Journal of Scientific Computing*, 190(34):1467–1482, 2014.

For completeness we mention another project in which the author was involved during the three years of PhD. Always in the context of the numerical simulation of long waves, we have investigated the large scale mechanism of tidal bore formation in convergent alluvial estuaries. The work was inspired by the experimental campaign in the Garonne river of Bonneton *et al.* [2015] in which it was shown that the dominant dimensionless parameter for bore formation appeared to be the dissipation one, which multiplies the frictional term of the SWEs. Although the basic conditions are well known (large tidal range, shallow and convergent channel, low freshwater discharge), a parametric estuarine classification in terms of bore occurrence did not exist in the literature. It was also generally accepted that tidal bores form in estuaries which amplify the incoming tidal wave. We have numerically investigated the estuarine parameter space and proposed a new scaling for the SWEs equations that ensures a clear separation of the effect of non-linearity and friction dissipation. As a consequence, we ended up with a set of dimensionless parameters defining a space in which real alluvial estuaries developing bores are clearly divided from those where bores are not observed. The formation process of an undular tidal bore in such idealized convergent estuaries was reproduced by the use of Boussinesq-type equations and dispersive effects on the local scale of the bore were quantified. For the interested reader, the published papers related to this works are:

1. A.G. Filippini, L. A., P. Bonneton and M. Ricchiuto. Modeling analysis of tidal bore formation in convergent estuaries. (submitted to *European Journal of Mechanics / B Fluids*).

2. P. Bonneton, A.G. Filippini, L. A., N. Bonneton and M. Ricchiuto. Conditions for tidal bore formation in convergent alluvial estuaries. *Estuarine, Coastal and Shelf Science.* 172, 121-127, 2016.

Structure of the manuscript

Chapter 1 is devoted to present two existing explicit second order schemes for the *Eulerian Shallow Water equations* (SWEs) on fixed unstructured grids. They are a Residual Distribution scheme and a more classical Finite Volume scheme which is used for comparison.

In chapter 2 we propose an *Arbitrary Lagrangian Eulerian (ALE) form of the SWEs* allowing moving mesh simulations with wetting/drying fronts. In order to do this, we systematically review the forms of the SWEs which are best suited for preserving fundamental physical solutions on moving meshes such as the lake at rest or a uniform flow at constant depth. We use the resulting model equations to provide second order ALE Finite Volume and Residual Distribution discretizations. In particular we provide a simple recipe to marry the preservation of the lake at rest and mass conservation on moving meshes using a re-interpolation of the nodal bathymetry based on accurate quadrature of the given bathymetric data.

In chapter 3 the ALE moving mesh method is tested against the popular *rezoning* approach in which the Eulerian SWEs are solved on the last adapted mesh and an appropriate interpolation permits to transfer the solution from the previous mesh. We study the impact of cheaper and simpler interpolation algorithms which retains all the desired discrete properties. A novel simple ad-hoc error estimators allows to track the shorelines. Finally the ALE and the rezoning moving mesh algorithms are evaluated in terms of CPU time for a given resolution, using the standard benchmarks for near shore hydrodynamics of the WP1.

In chapter 4 we will address the numerical approximation of the ALE-SWEs in curvilinear coordinates. The validation of the partners' codes on the sphere was not addressed in the TANDEM project. In particular the benchmarking work-package WP1 did not foresee any test-case to validate the accuracy and the Well-Balanced properties on the sphere. Firstly, we have chosen to set properly the Eulerian *SWEs in standard latitude-longitude curvilinear coordinates*. We will take cases from the standard test-suite of Williamson *et al.* [1992] to validate our Finite Volume and Residual Distribution scheme for the SWEs in curvilinear coordinates on unstructured grids. Secondly we have set the ALE-SWEs on the sphere. A novel Finite Volume and Residual Distribution approximation of the resulting equations is proposed. Third, a simple moving mesh method will be established which allows point movements on the sphere. Numerical tests show that our resulting moving mesh algorithm, can improve the resolution of linear and nonlinear waves on the sphere, using a limited number of mesh points.

Finally in chapter 5 we address the issue of WP3. We present fixed and moving mesh SWEs simulations of the 2011 Tohoku-Honsu event. The numerical tools developed along this manuscript will be compared with other partners involved in the WP3. This should provide evidence that the moving mesh method studied in the thesis, although simple, could be a good candidate to reduce the large computational cost of tsunami simulations.

Chapter 1

Shallow Water equations: continuous properties and Well-Balanced approximations on unstructured grids

1.1 Shallow Water equations

The use of Shallow Water equations (SWEs) is ubiquitous in hydrodynamics and geophysics. SWEs is a nonlinear hyperbolic system of partial differential equations that describes wave propagation in a fluid that is shallow relative to the wavelength. SWEs can be derived from the conservation of mass and momentum of a vertical fluid column under a crucial assumption, the so called hydrostatic approximation. We consider the vertical fluid velocity negligible and the horizontal velocity about constant along the the vertical coordinate from the bottom to the free surface, which is true if the fluid depth is shallow compared to the horizontal length scale. Moreover since the fluid is at rest along the vertical coordinate the pressure is determined just by the weight of fluid above, that is $p = p(h)$, h here is the water depth. There are many circumstances under which these hypothesis lead to a good approximation of natural phenomena such as waves in the atmosphere, tides in ocean but also breaking waves on a beach and, more important for us, tsunamis. If we think of tsunamis generated by an offshore quake, they are very long waves with wavelength L in the order of $\sim 100\,km$, much larger then the water depth $D \sim 1\,km$. See e.g. Benoit [2016]; Lannes [2016] for the scaling analysis.

A set of Cartesian coordinates $\boldsymbol{x} = \{x^1, x^2\} \in \Omega$ defined on the orthonormal basis $\{\boldsymbol{e}_1, \boldsymbol{e}_2\}$ is given at this point. The vector basis define a plane which we assume that the water free surface in undisturbed condition lies on. Then, the

SWEs in engineering notation writes:

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial h u^j}{\partial x^j} &= 0 \\ \frac{\partial h u^i}{\partial t} + \frac{\partial T^{ij}}{\partial x^j} + S^i &= 0 \end{aligned}$$

the repeated index $j = 1, 2$ is implied in summations and the index $i = 1, 2$ is the direction of the momentum equation. The momentum discharge vector is defined as $h\boldsymbol{u} = hu^j \boldsymbol{e}_j$ and the momentum flux tensor is $\boldsymbol{T} = T^{ij}\boldsymbol{e}_i\boldsymbol{e}_j$ with components $T^{ij} = huu^{ij} + \frac{1}{2}gh^2\delta_{ij}$. The source term includes the bottom topography force and the frictional one

$$S^i = h\left(g\frac{\partial b}{\partial x^i} + c_F u^i\right)$$

$b(\boldsymbol{x})$ is the bottom topography, measured form the origin which is the undisturbed free surface level. c_F is the friction coefficient defined through Manning's empirical formula $c_F = \frac{gn^2\|\boldsymbol{u}\|}{h^{4/3}}$ and n is the Manning number. We also provide the vector form of the SWEs, which has the advantage of being more compact

$$\frac{\partial \mathsf{u}}{\partial t} + \frac{\partial \mathsf{F}^j}{\partial x^j} + \mathcal{S}(\boldsymbol{x}, \mathsf{u}) = 0, \tag{1.1}$$

$$\mathsf{u} = \begin{bmatrix} h \\ hu^i \end{bmatrix}, \; \mathsf{F}^j = \begin{bmatrix} hu^j \\ T^{ij} \end{bmatrix}, \; \mathcal{S} = \underbrace{\begin{bmatrix} 0 \\ gh\frac{\partial b}{\partial x^i} \end{bmatrix}}_{\mathcal{S}^b} + \underbrace{\begin{bmatrix} 0 \\ c_F hu^i \end{bmatrix}}_{\mathcal{S}^f} \tag{1.2}$$

We collect the flux components in the flux matrix $\mathbf{F} = [\,\mathsf{F}^1\ \mathsf{F}^2\,] \in \mathbb{R}^3 \times \mathbb{R}^2$. Equations (1.1)(1.2) represents a non-homogeneous non-linear hyperbolic system of partial differential equations. Given any vector $\boldsymbol{\xi} = \xi^i\boldsymbol{e}_i$, the flux Jacobian $\mathsf{K}(\mathsf{u}, \boldsymbol{\xi}) = \frac{\partial \mathsf{F}^j}{\partial \mathsf{u}}\xi^j$ admits a full set of real eigenvalues and linearly independent eigenvectors, namely

$$\mathsf{K} = \begin{pmatrix} 0 & \xi^1 & \xi^2 \\ c^2\xi^1 - u\boldsymbol{u}\cdot\boldsymbol{\xi} & \boldsymbol{u}\cdot\boldsymbol{\xi} + u\xi^1 & u\xi^2 \\ c^2\xi^2 - v\boldsymbol{u}\cdot\boldsymbol{\xi} & v\xi^1 & \boldsymbol{u}\cdot\boldsymbol{\xi} + v\xi^2 \end{pmatrix} \tag{1.3}$$

with eigenvalues $\lambda(\mathsf{u}, \boldsymbol{\xi})$

$$\lambda_{1,3} = u^j\xi^j \pm c\|\boldsymbol{\xi}\|, \quad \lambda_2 = u^j\xi^j \tag{1.4}$$

and $c = \sqrt{gh}$ being the local wave celerity. The nonlinear eigenstructure gives more insight about the physics of waves in the shallow water regime. Shallow water waves are propagating at speed $\pm\sqrt{gh}$ relative to the fluid, thus wave in deeper water moves faster. If the amplitude of the wave is not small compared to the water depth, the speed of every point may differ due to the different height, leading to a distortion where the crest is moving faster respect to the trough. Two typical hyperbolic phenomena occurs: the wave front steepens (hydraulic jumps) and the tail flattens (depression). This can be clearly seen for waves approaching the beach, at least before that the familiar breaking of the wave occurs.

Later in the text we will also make use of the Jacobian at rest $\mathsf{A} = \mathsf{K}(\boldsymbol{u} = 0)$

$$\mathsf{A}(h, \xi) = \begin{pmatrix} 0 & \xi^1 & \xi^2 \\ c^2\,\xi^1 & 0 & 0 \\ c^2\,\xi^2 & 0 & 0 \end{pmatrix} \tag{1.5}$$

that admits the non-linear eigenvalues $\lambda = \pm c\|\boldsymbol{\xi}\|$ that characterize the acoustics of the system.

1.2 Conservation form

The SWEs (1.1)(1.2) constitute a balance law and allow to recover the integral balance of of fundamental physical quantities such as mass and momentum. For a fixed volume C with boundary ∂C and boundary normal $\boldsymbol{n} = n^j \boldsymbol{e}_j$ we can state integral conservation for the vector of conservative variables u:

$$\begin{aligned} \int_C \mathsf{u}(\boldsymbol{x}, t)\, d\boldsymbol{x} &= \int_C \mathsf{u}(\boldsymbol{x}, 0)\, d\boldsymbol{x} \\ &- \int_0^t \left(\int_{\partial C} \mathsf{F}^j n^j ds + \int_C \mathcal{S}\, d\boldsymbol{x} \right) dt = 0 \end{aligned} \tag{1.6}$$

These are the most fundamental equations when the solution contains discontinuities. In this case, the differential form does not hold anymore but (1.6) still admits weak solutions corresponding to the Rankine-Hugoniot jump conditions, see Bernetti *et al.* [2008]:

$$\begin{aligned} s[h] &= [hu] \\ s[hu] &= [hu] + \left[\frac{1}{2} g h^2\right] - \frac{1}{2}\left(h_L^2 - (h_L - (b_L - b_R)^2)\right) \end{aligned}$$

where we defined the jump of any scalar ϕ across the discontinuity as $[\phi] = \phi_R - \phi_L$ and the shock velocity as s. As we can see, the bathymetric term disappears when the bottom is at least piecewise continuous $b_R = b_L$. On the contrary, in presence of a bottom jump $b_R \neq b_L$, this term play an important role in Rankine-Hugoniot relationship and in the resolution of the Riemann problem. It has been shown that a standing wave in correspondence of the bottom jump appears, see Bernetti *et al.* [2008], Alcrudo and Benkhaldoun [2001].

Being conservative is a delicate issue for every numerical method approximating conservation laws. It means that the numerical solution satisfies the integral form of conservation laws, mimicking what the exact solution does. For conservative methods, one can hope, as the mesh is refined, that the numerical solution converges to the weak solution of (1.6) and, even in presence of a discontinuity, we compute the correct wave speed. This was proved by Lax and Wendroff [1960].

1.3 Shallow Water entropy

An hyperbolic equation should be supplied by some conditions, which ensure that the weak solution of the problem is physically correct. These conditions are called entropy conditions because in gasdynamics the correct weak solution is selected by the second principle of thermodynamics which states that the entropy of a system must be non increasing with time. For SWEs, the role of the entropy is played by the depth-integrated total energy per unit mass, see Tadmor and Zhong [2006]

$$E(\mathsf{u}) = h\left(\frac{gh}{2} + gb + \frac{\|\boldsymbol{u}\|^2}{2}\right)$$

A conservation law for the SW energy can be obtained projecting mass and momentum equations (1.1), on the so called entropy vector

$$\mathsf{\Psi} = \frac{\partial E}{\partial \mathsf{u}} = [\,(h+b)g - \frac{\|\boldsymbol{u}\|^2}{2} \;\; u \;\; v\,]^T$$

We obtain the conservation of energy:

$$\frac{\partial E}{\partial t} + \frac{\partial \psi^j}{\partial x^j} = 0 \tag{1.7}$$

where the entropy flux is $\boldsymbol{\psi} = \psi^j \boldsymbol{e}_j$ with components $\psi^j = u^j E + \frac{1}{2} u^j g h^2$. However if we go back to the complete set of physical equation with viscosity, then we can argue that the admissible solution should be the one in the vanishing viscosity limit. Lax [1972] demonstrated that the vanishing viscosity limit for a system of conservation law with positive convex entropy $\partial^2_{\mathsf{uu}} E > 0$, is equivalent to an entropy inequality. For the SWEs this is

$$\frac{\partial E}{\partial t} + \frac{\partial \psi^j}{\partial x^j} \leq 0 \tag{1.8}$$

(1.8) states that the energy, in absence of boundary conditions, cannot grow. And this is a physical criteria that prevent from picking up unphysical weak solution, such as expansion shocks. Moreover (1.8) states that the energy is conserved whenever the solution is smooth, but it is dissipated across discontinuities. We will see that this fact is used in numerical methods to detect shock waves in the domain. Finally (1.8) can be seen as stability condition for the SWEs. Given a certain norm of the energy $\|E\|_{V(\Omega)}$, entropy inequality corresponds to:

$$\|E(t)\|_V \leq \|E_0\|_V$$

where $\|E_0\|_V$ is the norm of the energy at the initial time.

1.4 Maximum principle, entropy stability and water depth positivity

Consider for a while, the following scalar hyperbolic equation that conserves the variable $u(\boldsymbol{x}, t)$, for a general non-linear flux function $F(u)$:

$$\frac{\partial u}{\partial t} + \frac{\partial F^j}{\partial x^j} = 0, \quad \boldsymbol{x} \in \Omega, t \in [0, T] \tag{1.9}$$

L_∞ stability or maximum principle holds for the exact solution:

$$\min_{\boldsymbol{x}\in\Omega} u(\boldsymbol{x}, 0) < u(\boldsymbol{x}, t) < \max_{\boldsymbol{x}\in\Omega} u(\boldsymbol{x}, 0), \quad \forall t \in [0, T] \tag{1.10}$$

The above condition is very important when one searches for approximate solution $u_h \approx u$. In particular, maximum principle is used to have precise bounds on the numerical solution which in turn avoids the computations of oscillatory discontinuous solution, see the work of Zhang and Shu [2010] on maximum-principle-satisfying high order schemes. This stability property cannot be automatically extended to the non-linear system case since the same maximum principle is not trivial for the general problem described by (1.1). However (1.10) remains important. Historically, numerical methods have been studied and tested in the scalar case, where a non-linear stability theory is available. Only later, scalar methods were extended to the system case, more or less straightforwardly. Even if we do not have any guarantees that the stability property will work well as in the scalar case, this approach was successful. To cope with the obstacle of L_∞ stability many researchers investigated other types of stability. An important class of methods are, for example, *entropy stable* schemes (Osher [1984], Tadmor [1987]), which aim to mimic the physical constraint that the energy cannot increase. By adding a certain amount of numerical viscosity, entropy stable schemes can verify a discrete inequality analogue to the entropy one (1.8).

For the SWEs, another stability condition is a physical constraint: the positivity of water depth $h \geq 0$. Preserving the positivity of water depth is an important stability condition for numerical methods since the uncontrolled occurrence of negative depth leads soon to the code breakdown. We will discuss *positivity preserving schemes* (Bouchut [2004]) later in this chapter.

1.5 Lake at rest

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 $\eta = h + b$, is the particular steady solution characterized by the two invariants :

$$h\boldsymbol{u} = 0, \qquad h + b = \eta_0 = const \tag{1.11}$$

A numerical method approximating (1.1)(1.2) is said to enjoy the *C-property* or also to be *Well-Balanced* if (1.11) is also an exact steady solution of the discrete equations. In other words, Well-balanced schemes provide a discrete analog of the balance between hydrostatic pressure and the pressure forces exerted by the bottom

$$\frac{\partial \mathsf{F}^j}{\partial x^j} + \mathcal{S} = 0$$

allowing to preserve (1.11) exactly at the discrete level, Bermudez and Vazquez-Cendon [1994].

1.6 Discrete approximation

1.6.1 Notation for mesh, geometry and unknowns

Consider an unstructured discretization of the spatial domain Ω composed by non overlapping triangular elements. We will denote the grid (or mesh) by $\mathcal{T}_h$, h_K being the local reference element length (mesh size, hereinafter). K is the generic triangle, $|K|$ its area and $\boldsymbol{n}_j = n_j^m \boldsymbol{e}_m$ is the normal to the j-th face of the triangle scaled by the corresponding edge length (see fig.1.1). 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, its area is

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

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 neighboring triangles $K \ni i, j$ with the midpoint of the edge ij (cf. right picture in fig.1.1). 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 split over the neighboring triangles. We define the normal and the area associated to the interface ij

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

with $|C_{ij}^K| = \frac{|K|}{6}$. We highlight the following geometrical relationship between nodal normals and interface normals:

$$\frac{1}{2} \sum_{j \in K, j \neq i} \boldsymbol{n}_j^K = -\frac{\boldsymbol{n}_i^K}{2} = \sum_{j \in K, j \neq i} \boldsymbol{n}_{ij}^K \tag{1.12}$$

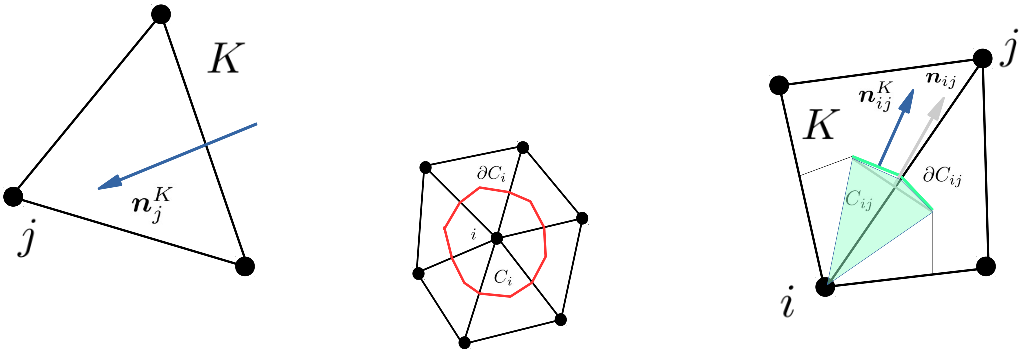

Figure 1.1: Left: Nodal normal definition. Center: Finite Volume dual cell. Right: dual cell interface with normals.

With the subscript $_i$ we distinguish discrete variables from continuous ones. It is a widespread convention in literature, however we note that engineering notation with sub/superscripts to denote vector and tensor components, could confuse the reader. In the following, we will specify whenever standard summation does not apply. For the Finite Volume method we will evolve in time approximations of cell averages of conservative variables over the standard median dual cells and we denote the average as u_i. On the contrary, the Residual Distribution method evolves values of the unknowns at mesh nodes. For simplicity, we shall still denote these values as u_i.

1.6.2 Finite Volume for SWEs

Finite Volume (FV) is the most popular class of schemes for the discretization of hyperbolic PDEs in computational hydrodynamics. An exhaustive monograph on FV for hyperbolic problem is LeVeque [2004] and a review of

FV methods for the SWEs can be found in Toro and Garcia-Navarro [2007]. Here we present the standard well balanced node-centered unstructured scheme based on Roe's linearized Riemann solver as it has been developed by Nikolos and Delis [2009]. This scheme proved to successfully simulate complex problems with irregular topography and inundations. The FV idea is to break the domain into many volumes C_i. The solution is then averaged within every cell with

$$\mathsf{u}_i(t) = \frac{1}{|C_i|}\int_{C_i} \mathsf{u}(\boldsymbol{x},t)\, d\boldsymbol{x} \tag{1.13}$$

and it is updated by a discrete integral balance, equation (1.6)

$$\mathsf{u}_i^{n+1} = \mathsf{u}_i^n - \frac{\Delta t}{|C_i|}\sum_{j\in\mathcal{D}_i} R_{ij}(\mathsf{u}_i^n, \mathsf{u}_j^n)$$

where we have

$$R_{ij} = \mathcal{F}_{ij} + \mathcal{S}_{ij} \tag{1.14}$$

with $\mathcal{F}_{ij}$ a numerical approximation of the flux along ∂C_{ij} and $\mathcal{S}_{ij}$ an approximation of the integral of the source term on C_{ij}.

$$\mathcal{F}_{ij} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}}\int_{\partial C_{ij}} \mathsf{F}^j n^j ds\, dt, \quad \mathcal{S}_{ij} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}}\int_{C_{ij}} \mathcal{S} d\boldsymbol{x}\, dt$$

After the averaging step, a Riemann problem is defined at each interface. Numerical fluxes can be computed if the complete solution of the Riemann problem is achieved, typically through the resolution of a non-linear system. Note, however, that the exact solution of the Riemann solution seems to be redundant since the same solution is roughly averaged just at the end of the time step. For this reason one can think to replace the exact Riemann Problem with an approximate one. The seminal idea consists in linearizing the flux $\frac{\partial \mathsf{F}^j}{\partial x^k} = \frac{\partial \mathsf{F}^j}{\partial \mathsf{u}}\frac{\partial \mathsf{u}}{\partial x^k}$. In this work we use the original Roe linearization Roe [1982]. Recalling expression (1.3), $\mathsf{K}_{ij} = \mathsf{K}(\mathsf{u}_{ij}^\star, \boldsymbol{n}_{ij})$ will be the flux Jacobian evaluated at the interface Roe state $\mathsf{u}_{ij}^\star$. Within this hypothesis, the Riemann solution consists in the superposition of simple or p-waves. Numerical flux evolves according to the sum of p-waves entering the median dual cell through the interface, each one carrying a jump in the solution equal to $\mathcal{W}_p$:

$$\mathcal{F}_{ij}(\mathsf{u}_i, \mathsf{u}_j) = \mathbf{F}_i \cdot \boldsymbol{n}_{ij} + \sum_{p=1,\alpha_p<0}^{m} \mathcal{W}_p = \mathbf{F}_i \cdot \boldsymbol{n}_{ij} + \mathsf{K}_{ij}^-(\mathsf{u}_j - \mathsf{u}_i) \tag{1.15}$$

$$= \mathbf{F}_j \cdot \boldsymbol{n}_{ij} - \sum_{p=1,\alpha_p>0}^{m} \mathcal{W}_p = \mathbf{F}_j \cdot \boldsymbol{n}_{ij} - \mathsf{K}_{ij}^+(\mathsf{u}_j - \mathsf{u}_i) \tag{1.16}$$

Please note that the above scalar products are standard ones and the summation convention does not apply in the above formula and in the ones which follow. Matrices are computed through eigenvalues decomposition, e.g. $\mathsf{K}_{ij}^{+} = \mathsf{R}_{ij}\Lambda_{ij}^{+}\mathsf{L}_{ij}$. To obtain expressions (1.15) and (1.16) we have decomposed the total jump in the solution as follows:

$$\mathcal{W}_p = \alpha_p \lambda_p \mathsf{r}_p, \quad \alpha_p = (\mathsf{u}_j - \mathsf{u}_i)\,\mathsf{l}_p$$

r_p, l_p are respectively the p-th right and left eingenvectors and are referred, as the eigenvalues λ_p, to the Roe state at the interface, e.g. $\lambda_p = \lambda_p(\mathsf{u}_{ij}^{\star}, \boldsymbol{n}_{ij})$. Averaging expression (1.15) and (1.16) we have the numerical flux:

$$\mathcal{F}_{ij} = \mathcal{F}_{ij}(\mathsf{u}_i,\, \mathsf{u}_j) = \frac{\mathbf{F}(\mathsf{u}_j) + \mathbf{F}(\mathsf{u}_i)}{2} \cdot \boldsymbol{n}_{ij} - \frac{|\mathsf{K}_{ij}|}{2}(\mathsf{u}_j - \mathsf{u}_i) \tag{1.17}$$

this scheme is known as *Roe scheme*. Second order of accuracy is achieved with a piecewise linear reconstruction of the vector of conservative variable over the dual cell. In practice we have replaced, in expression (1.17), nodal values u_i and u_j with linearly reconstructed ones at the cell interface, respectively $\breve{\mathsf{u}}_i$ and $\breve{\mathsf{u}}_j$. We have combined this with a Green-Gauss reconstruction Nikolos and Delis [2009]; Delis *et al.* [2011]. We will refer to this linear second order scheme as to the FROMM scheme, LeVeque [2004]. If necessary in the reconstruction step, a slope limiter is used to enforce monotonicity

$$\breve{\mathsf{u}}_i = \mathsf{u}_i + \frac{1}{2} l(\mathsf{u}_i, \mathsf{u}_j, \nabla \mathsf{u}_i)$$

with l, the Van Albada limiter Van-Albada *et al.* [1982]. The resulting scheme is a two dimensional implementation of the MUSCL scheme of Van-Leer [1979].

To march in time with second order of accuracy we use an explicit Runge-Kutta two (eRK2); the FV discrete evolution equations reads:

$$\mathsf{u}_i^{*} = \mathsf{u}_i^{n} - \frac{\Delta t}{|C_i|} \sum_{j \in \mathcal{D}_i} R_{ij}(\mathsf{u}^n) \tag{1.18}$$

$$\mathsf{u}_i^{n+1} = \mathsf{u}_i^{n} - \frac{\Delta t}{|C_i|} \sum_{j \in \mathcal{D}_i} \left(\frac{R_{ij}(\mathsf{u}^n)}{2} + \frac{R_{ij}(\mathsf{u}^*)}{2} \right) \tag{1.19}$$

Conservation

From the solution decomposition in simple waves (1.15) and (1.16), it is clear that Roe linearization ensures conservativeness. By definition the jump in the flux is:

$$(\mathbf{F}_i - \mathbf{F}_j) \cdot \boldsymbol{n}_{ij} = \mathsf{K}_{ij}(\mathsf{u}_{ij}^{\star}, \boldsymbol{n}_{ij})\,(\mathsf{u}_j - \mathsf{u}_i)$$

and thus (1.15) and (1.16) coincide with the consequent exact conservation at the interface. An advantage of the numerical flux (1.17) is that conservation is true for any linearization, for example the simple arithmetic average $\mathsf{u}_{ij}^{\star} = \frac{1}{2}(\breve{\mathsf{u}}_i + \breve{\mathsf{u}}_j)$. This fact will guarantees the method is conservative even in more complex cases for which the Roe average is not directly available, as we will in chapter 4.

Positivity of water depth

Perthame and Shu [1996] have shown that Godunov method applied to the Euler equations of gasdynamics, (with exact or approximate Riemann solver) preserves density and pressure positivity, with a rigorous CFL condition. They also presented second order pressure/density positivity preserving schemes but some other requirement on the reconstruction should be assured. In this work to avoid the appearance of negative depths, following Nikolos and Delis [2009] we propose to switch to a first order scheme when approaching regions dry regions with an appropriate CFL condition, see (1.38). The limiter is modified according to the definition given in Ricchiuto and Bollerman [2009]:

$$l^* = l(\mathsf{u}_i, \mathsf{u}_j) e^{-\frac{h_K}{L_{ref}}\frac{h_{ref}}{h_{min}}} \tag{1.20}$$

the exponential factor takes into account the occurrence of dry areas. L_{ref} is the domain reference length; h_{ref} is the maximum depth at $t = 0$; h_{min} is the minimum depth among h_i and h_j. It is always of order one but it quickly tends toward zero where h is small.

Source terms

Non differential source terms in (1.1) includes only friction. For second order of accuracy, we can use pointwise values $\mathcal{S}_i = \mathcal{S}(\boldsymbol{x}_i, \mathsf{u}_i)$ and approximate the numerical source as

$$\mathcal{S}_{ij} = \int_{C_{ij}} \mathcal{S}(\boldsymbol{x}, \mathsf{u})\, d\boldsymbol{x} = \mathcal{S}_i |C_{ij}|, \qquad \sum_{j \in \mathcal{D}_i} \mathcal{S}_{ij} = \mathcal{S}_i |C_i|$$

which corresponds to linearly reconstruct also the source term. More complex treatments can be constructed, especially when increasing the accuracy to more then second order.

Well-Balancedness

It is widely recognized that the numerical treatment of the bathymetric source term is as important as that of advective terms. In fact, this particular term contained in the numerical source $\mathcal{S}_{ij}$, in addition to not spoil the accuracy of the advective part, it should preserves of the Well-Balanced property discussed in section 1.5. The first attempt to a well-balanced source term was due to Bermúdez and Vázquez Bermudez and Vazquez-Cendon [1994] who tried to incorporate the bathymetry source term into Roe method. A second important cornerstone was the extension from first to second order accuracy proposed by Hubbard and Garcia- Navarro Hubbard and Garcia-Navarro [2000]. We present here the main conclusions of this last reference. They distinguished two contributions, the first balancing the central part of the fluxes, and the second the upwind dissipation term:

$$\mathcal{S}^b_{ij} = \mathcal{S}^c_{ij} + \mathcal{S}^*_{ij} \tag{1.21}$$

We define the following average values

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

and the bathymetry variation vectors

$$\Delta \mathsf{b}^-_{ij} = \begin{pmatrix} \breve{b}_i - b_i \\ 0 \\ 0 \end{pmatrix}, \quad \Delta \mathsf{b}_{ij} = \begin{pmatrix} \breve{b}_j - \breve{b}_i \\ 0 \\ 0 \end{pmatrix} \tag{1.22}$$

moreover we recover the definition (1.5) (Jacobian at rest) to introduce $\mathsf{A}^-_{ij} = \mathsf{A}(h^-_{ij}, \boldsymbol{n}_{ij})$, $\mathsf{A}_{ij} = \mathsf{A}(h_{ij}, \boldsymbol{n}_{ij})$. The centered component of the source can be now written as

$$\mathcal{S}^c_{ij} = \mathsf{A}^-_{ij} \Delta \mathsf{b}^-_{ij} + \frac{1}{2} \mathsf{A}_{ij} \Delta \mathsf{b}_{ij} \tag{1.23}$$

Concerning the upwind balancing term, the original definition given in Bermudez and Vazquez-Cendon [1994]; Hubbard and Garcia-Navarro [2000] leads to the following expression

$$\mathcal{S}^*_{ij} = -\frac{\operatorname{sign}(\mathsf{K}_{ij})}{2} \mathsf{A}_{ij} \Delta \mathsf{b}_{ij} \tag{1.24}$$

The proof of Well-Balancedness rests on the property of the Roe average and the fact that, on the lake at rest state, we have $\mathsf{K}_{ij} = \mathsf{A}_{ij}$. On each edge

$$\begin{aligned} R_{ij} &= \frac{\mathbf{F}(\breve{\mathsf{u}}_j) - \mathbf{F}(\breve{\mathsf{u}}_i)}{2} \cdot \boldsymbol{n}_{ij} + (\mathbf{F}(\breve{\mathsf{u}}_i) - \mathbf{F}(\mathsf{u}_i)) \cdot \boldsymbol{n}_{ij} - \frac{|\mathsf{A}_{ij}|}{2} (\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i) \\ &+ \frac{1}{2} \mathsf{A}_{ij} \, \Delta \mathsf{b}_{ij} + \mathsf{A}^-_{ij} \, \Delta \mathsf{b}^-_{ij} - \frac{\operatorname{sign}(\mathsf{A}_{ij})}{2} \mathsf{A}_{ij} \Delta \mathsf{b}_{ij} = 0 \end{aligned}$$

Using the fact that, on the selected equilibrium, $(\mathbf{F}(\breve{\mathsf{u}}_j) - \mathbf{F}(\breve{\mathsf{u}}_i)) \cdot \boldsymbol{n}_{ij} = \mathsf{A}_{ij}(\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i)$ we can sum each term of the numerical flux with the corresponding bathymetric term. Note now that $\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i + \Delta \mathsf{b}_{ij} = 0$ by hypothesis.

□

1.6.3 An introduction to Residual Distribution

At the beginning of the eighties, Roe's research on the multidimensional extension of Godunov method brought him to formalize FV into a form called Fluctuation Splitting Roe [1982, 1987]. This has paved the way to the development of multidimensional upwind schemes and to what today is referred as Residual Distribution (RD). The main ideas behind RD are recalled here, with the main purpose of introducing definitions that will be used extensively in the manuscript. The interested reader can find the rigorous foundations of the method in Deconinck and Ricchiuto [2007] and a recent review in Abgrall and Ricchiuto [2017]. We start from the FV framework, which the reader is more confident with, and, as Roe actually did, we develop the concept of fluctuation. We write a general flux (e.g. Roe fluxes with MUSCL reconstruction (1.17)) compactly as

$$\mathcal{F}_{ij} = \mathcal{H}(\breve{\mathsf{u}}_i, \breve{\mathsf{u}}_j)$$

and we decompose them in the contribution of each element, the FV update writes:

$$\begin{aligned}
\mathsf{u}_i^{n+1} &= \mathsf{u}_i^n - \frac{\Delta t}{|C_i|} \sum_{j \in \mathcal{D}_i} \mathcal{H}(\breve{\mathsf{u}}_i, \breve{\mathsf{u}}_j) \\
&= \mathsf{u}_i^n - \frac{\Delta t}{|C_i|} \sum_{K \in \mathcal{D}_i} \sum_{j \in K, j \neq i} \mathcal{H}(\breve{\mathsf{u}}_i, \breve{\mathsf{u}}_j)
\end{aligned}$$

since $\sum_{j \in \mathcal{D}_i} \mathcal{H}(\mathsf{u}_i, \mathsf{u}_i) = \mathbf{F}_i \cdot \sum_{j \in \mathcal{D}_i} \boldsymbol{n}_{ij} = 0$ we can add the following term:

$$\mathsf{u}_i^{n+1} = \mathsf{u}_i^n - \frac{\Delta t}{|C_i|} \sum_{K \in \mathcal{D}_i} \sum_{j \in K, j \neq i} \left(\mathcal{H}(\breve{\mathsf{u}}_i, \breve{\mathsf{u}}_j) - \mathcal{H}(\mathsf{u}_i, \mathsf{u}_i) \right) \tag{1.25}$$

This step seems arbitrary but it is not. In fact, in each element, we are implicitly making use of a very special flux. Using the property (1.12) allow to show that:

$$\begin{aligned}
- \sum_{j \in K, j \neq i} \mathcal{H}(\mathsf{u}_i, \mathsf{u}_i) &= - \sum_{j \in K, j \neq i} \mathbf{F}_i \cdot \boldsymbol{n}_{ij} = -\mathbf{F}_i \cdot \sum_{j \in K, j \neq i} \boldsymbol{n}_{ij} = \mathbf{F}_i \cdot \frac{\boldsymbol{n}_i}{2} \\
&= - \sum_{j \in K, j \neq i} \mathbf{F}_i \cdot \frac{\boldsymbol{n}_j}{2}
\end{aligned}$$

that is, we are taking into account also the numerical fluxes along half the edges with vertex i in common. This contributions cancels out between the elements sharing the same edge and do not modify the original scheme. However, summing over the vertexes, the internal numerical flux contribution disappears for the conservation of an element and we are left with

$$\begin{aligned}
\Phi^K &= \sum_{i \in K} \sum_{j \in K, j \neq i} \left(\mathcal{H}(\breve{\mathsf{u}}_i, \breve{\mathsf{u}}_j) - \mathcal{H}(\mathsf{u}_i, \mathsf{u}_i) \right) = - \sum_{i \in K} \sum_{j \in K, j \neq i} \mathcal{H}(\mathsf{u}_i, \mathsf{u}_i) \\
&= - \sum_{i \in K} \sum_{j \in K, j \neq i} \mathbf{F}_i \cdot \frac{\boldsymbol{n}_j}{2} = \frac{1}{2} \sum_{i \in K} \mathbf{F}_i \cdot \boldsymbol{n}_i \approx \int_{\partial K} \mathbf{F} \cdot \boldsymbol{n} \, ds
\end{aligned}$$

which is a second order quadrature formula approximating the flux balance over the element K. We call this quantity *fluctuation* of the element K. The FV scheme could be seen as defining a proper *splitting* or *distribution* Φ_i^K of the fluctuation, such that:

$$\sum_{i \in K} \Phi_i^K = \Phi^K$$

so that we don't loose consistency property $\mathcal{F}_{ij}(\mathsf{u}, \mathsf{u}) = \mathbf{F}(\mathsf{u}) \cdot \boldsymbol{n}$. The split fluctuation for FV writes (dropping superscript K in the residual notation, we take it for granted)

$$\Phi_i^{FV} = \sum_{j \in K, j \neq i} \left(\mathcal{H}(\breve{\mathsf{u}}_i, \breve{\mathsf{u}}_j) - \mathcal{H}(\mathsf{u}_i, \mathsf{u}_i) \right) \tag{1.26}$$

and verifies the consistency condition. Substituting the definition (1.26) in (1.25) we get a reformulation of the standard FV method in Godunov form

$$\mathsf{u}_i^{n+1} = \mathsf{u}_i^n - \frac{\Delta t}{|C_i|} \sum_{K \in \mathcal{D}_i} \Phi_i^{FV} \tag{1.27}$$

We call this compact update as the fluctuation form of FV. The solution is evolved at each time step according to a balance of element's fluctuations rather then interface's fluxes. All this may seem just a trivial exercise but the promising aspect is that an abstract geometrical interpretation of upwinding arises, that is, the directions in which we distribute the fluctuation with respect to the direction of simple waves celerities. We can in fact generalize the splitting step and create new schemes with improved properties. For example, this fact is used to introduce upwinding in a more clever way. Indeed early experiments demonstrated that, for multidimensional problems, first order RD were less diffusive then first order FV Roe and Sidilkover [1992]; Paillere *et al.* [1998]. We also remark that FV, as they have presented in section 1.6.2, have been implemented in the RD form (1.27) and not in the classical flux form.

RD prototype scheme

Once we have presented FV in fluctuation from (1.27), the last effort consists in generalize the splitting procedure and to consider a splitting operator different from (1.26). These lead us directly to the RD approximation of hyperbolic systems such as the SWEs (1.1) and which is constructed with the following three steps: given an approximation of conservative variables at the nodes $\mathsf{u}_i(t) = \mathsf{u}(\boldsymbol{x}_i, t)$, we introduce the following continuous numerical approximation

$$\mathsf{u}_h(\boldsymbol{x}, t) = \sum_{i \in \mathcal{T}_h} \varphi_i(\boldsymbol{x}) \mathsf{u}_i(t) \tag{1.28}$$

$\{\varphi_i\}_{i \in \mathcal{T}_h}$ will be the standard P^1 continuous piecewise linear Lagrange kernel.

- For each element K, compute the flux balance, called fluctuation:

$$\Phi^K(\mathsf{u}_h) = \int_{\partial K} \mathsf{F}^j(\mathsf{u}_h) n^j ds + \int_K \mathcal{S}(\mathsf{u}_h, \boldsymbol{x})\, d\boldsymbol{x} \tag{1.29}$$

- Split the fluctuation in contributions, one for each node of the element, through an appropriate distribution matrix $\boldsymbol{\beta}_j^K$

$$\Phi_j^K : \sum_{j \in K} \Phi_j^K = \Phi^K, \quad \text{or} \;\; \Phi_j^K = \boldsymbol{\beta}_j^K \Phi^K \tag{1.30}$$

  for consistency reason it is clear that $\Rightarrow \sum_{j \in K} \boldsymbol{\beta}_j^K = \mathsf{I}^3$.

- envoy at each node of the element the corresponding split fluctuation, updating progressively the solution at vertexes. At the end, the RD method reads

$$|C_i| \frac{d\mathsf{u}_i}{dt} + \sum_{K \in \mathcal{D}_i} \Phi_i^K = 0, \quad \forall i \in \mathcal{T}_h \tag{1.31}$$

In figure (2.6) there is a sketch which summarize the three successive steps. The key properties of the method are determined by the definition of the split fluctuation, or, if you prefer, by the distribution matrix. In the following paragraph we relate the properties of the distribution matrix to stability and accuracy results. Leaving apart a rigorous treatment of the stability and accuracy analysis, we will try instead to highlight basic concepts. We hope that this could help the reader in a better comprehension of the RD results.

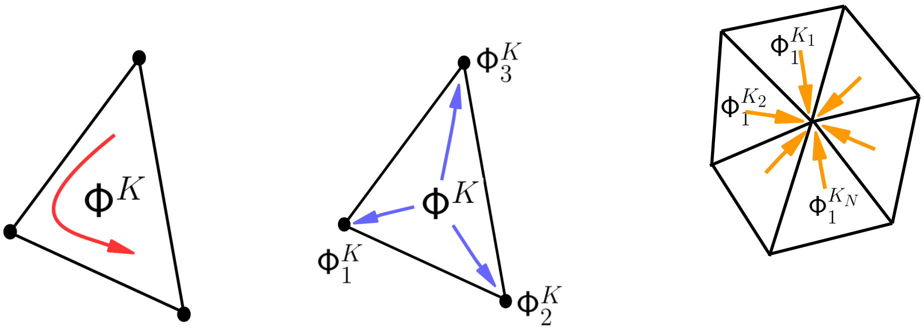

Figure 1.2: The abstract steps to construct a Residual Distribution approximation.

Conservation

RD approximate integral balance law (for an equivalence with the weak form and a bridge with the continuous Finite Element method, see Ricchiuto and Abgrall [2010]). If the divergence form of the equation is used, conservation can be incorporated quite naturally. To check this statement, we sum (1.31) over all the nodes $i \in \mathcal{T}_h$

$$\sum_{i \in \mathcal{T}_h} \left(|C_i| \frac{d\mathsf{u}_i}{dt} + \sum_{K \in \mathcal{D}_i} \Phi_i^K \right) = 0$$

We can develop the advective part with the consistency condition contained in (1.30)

$$\sum_{i \in \mathcal{T}_h} \sum_{K \in \mathcal{D}_i} \Phi_i^K = \sum_{K \in \mathcal{T}_h} \sum_{j \in K} \Phi_j^K = \sum_{K \in \mathcal{T}_h} \int_{\partial K} \mathsf{F}^j n^j ds = \int_{\partial \Omega} \mathsf{F}^j n^j ds$$

we have assumed that numerical fluxes are continuous along the triangle edges ∂K such as it happens in the continuous Finite Element method. Integrating in time with $U(t) = \sum_{i \in \mathcal{T}_h} |C_i| \mathsf{u}_i(t)$ we get

$$U(t) = U(0) - \int_0^t \int_{\partial \Omega} \mathsf{F}^j n^j \, ds dt$$

which states that we have exact conservation over the full domain if the consistency condition is verified and if numerical fluxes cancel out at the element's edges. Note that this is a continuity condition for F^j that is satisfied by several definitions of the discrete flux. We will use reinterpolation of flux's nodal values.

Positivity

Consider again the scalar conservation law (2.65). For linear schemes the scalar fluctuation ϕ_i^K can be put in a linear compact form

$$\phi_i^K = \sum_{j \in K, \neq i} c_{ij}^K (u_i - u_j) \tag{1.32}$$

one can easily check that, if the coefficients are positive $c_{ij}^K > 0$, then any update in the form (1.31) assures that local maxima are not increasing $du/dt < 0$ and local minima are not decreasing in time $du/dt > 0$. This is known as Local Extrema Diminishing (LED) property. Going further, many time schemes under CFL condition, allow the update (1.31), to verify the discrete maximum principle (1.10) with precise bounds for all the time t^n, $n \in [0, M]$. Using the fluctuation (1.32), we construct, for each element, a positive contribution to the solution:

$$u_i^{n+1} = \left(1 - \frac{\Delta t}{|C_i|} \sum_{j \in K, j \neq i} c_{ij}^K \right) u_i^n + \sum_{j \in K, j \neq i} \frac{\Delta t}{|C_i|} c_{ij}^K u_j = \sum_{j \in K} a_{ij}^K u_j$$

u_i^{n+1} can be written as a convex combination of the solution at the previous time step, enforcing the positivity of the coefficients a_{ij}^K. These schemes are said to be Positive (P). For scalar LED-RD under a proper CFL condition, L_∞ stability follows

$$\min_{j \in \mathcal{T}_h} u_j^0 < u_i^n < \max_{j \in \mathcal{T}_h} u_j^0, \qquad \forall i \in \mathcal{T}_h, \forall n \in [0, M]$$

The direct extension of Positive RD to the system case was successful in practice. However we mention that the extension of the stability result could be misleading. First of all, the existence of a maximum principle for a system of non-linear conservation laws is not a trivial task. This does not mean that a non oscillatory behavior near discontinuities is desired/expected. If we extend the positive analysis to systems, it is not easy to handle with the LED condition, this time applied to a matrix $C_{ij} < 0$. To avoid this difficulties, it is necessary to searches for some conditions relying on entropy consideration (we have seen that should always decrease) but this is beyond the scope of the brief paragraph, see Barth [1996]; Ricchiuto [2005]. Instead, for the SWEs much attention should be put to preserve the physical constraint of water depth positivity. In section 1.6.5 we will proof that a positive RD scheme applied to (1.1) preserves this property.

Linearity Preserving

We introduce now Linearity Preserving (LP) schemes which were developed for the accuracy analysis of RD. However the reader more familiar with FV will find analogies with the k-th exactness schemes of Barth. We consider a steady scalar problem with a smooth exact solution v such that

$$\frac{\partial F^j(v)}{\partial x^j} + S(\boldsymbol{x}, v) = 0$$

Keeping the notation of (1.28), v_h is the continuous piecewise polynomial approximation of v obtained with a general Lagrangian basis $\{\varphi_i\}_{i \in \mathcal{T}_h}$ of order q. First we give a trivial definition. For steady scalar problem a numerical scheme is q-th order accurate if the truncation error is of order $TE(v_h) = O(h_K^q)$. The local truncation error in RD case writes

$$TE = \sum_{i \in \mathcal{T}_h} \varphi_i \sum_{K \in \mathcal{D}_i} \phi_i^K(v_h)$$

with ϕ_i^K being the split residual in scalar notation. Less trivial is the fact that the truncation error estimate $TE(v_h) = O(h_K^q)$ is verified if the split fluctuation is

$$\phi_i^K = O(h_K^{q+1}) \tag{1.33}$$

the proof can be find in Deconinck and Ricchiuto [2007]. Given continuous and q-th order accurate flux and source terms approximations, $[F_h^1, F_h^2]$ and S_h, for a smooth exact solution one has also that

$$\phi^K = \int_{\partial K} F_h^j n^j \, ds + \int_K S_h \, d\boldsymbol{x} = O(h^{q+1})$$

Now we can take advantage of the RD formalism. The fundamental relation $\phi_i^K = \beta_i^K \phi^K$ (here in the scalar notation) leads to the concept of Linearity Preserving schemes. RD schemes are LP if the distribution coefficients are uniformly bounded with respect to the solution and data of the problem, hence exists a constant C such that

$$\max_{K \in \mathcal{T}_h} \max_{j \in K} \beta_j^K < C \quad \forall \phi^K, v_h, v_h^0$$

Using (1.33) we deduce that, for $q = 2$ (piecewise linear Lagrange basis), a scheme which is linearity preserving is second order accurate at steady state. All these results extend directly to the system case, in particular to the SWEs.

Distributions

Through a proper choice of the distribution matrix/distributed fluctuation, many classical schemes can be recovered. The FV scheme with Lax Friedrich (LxF) fluxes is important for its ability to compute non oscillatory discontinuous solutions and to have, at least on some variables Perthame and Shu [1996], precise bounds. The LxF fluctuation writes

$$\Phi_i^{LxF} = \frac{1}{3}\Phi^K + \sum_{j \in K, \neq i} \frac{\alpha^K}{3} \left(\mathsf{u}_i - \mathsf{u}_j\right), \quad \alpha^K = \max_{j \in K} \mathsf{K}_j$$

in the scalar case, the diffusion parameter α^K is chosen to verify the positivity (P) requirement of the coefficients. For systems one can take some upper bound within the time step to the the largest absolute value of the flux Jacobians evaluated in the nodes of an element. Unfortunately LxF is only first order accurate. Using the necessary condition of LP schemes, a root to the construction of non linear schemes which are LP and P consists in limiting the unbounded coefficients of a P scheme. We have used the PSI limiter of Deconink *et al.* [1993]

$$\Phi_i^{LLxF} = \sum_{m=1,3} \beta_i^m \left(\mathsf{l}_m^T \Phi_i^{LxF}\right) \mathsf{r}_m, \quad \beta_i^m = \frac{\left(\mathsf{l}_m^T \Phi^{LxF}\right)^+}{\sum \left(\mathsf{l}_m^T \Phi_j^{LxF}\right)^+} \tag{1.34}$$

where the apex LLxF stands for Limited LxF scheme. Formally this scheme is linearity preserving and positive. However, things turned out not so simple. Discontinuity are well handled while in smooth regions we encounter poor accuracy with wiggles. These oscillations can be seen as the rising of some unexpected destabilizing phenomena. Source of energy instability might be introduced by the PSI limiter. A possible solution to cure the problem is suggested by Abgrall [2006] is to add a SUPG term (which is LP but not P). The SUPG scheme of Hughes and co-workers (see Hughes and Brook [1982]; Hughes *et al.* [2010] and references therein for details) can be written in terms of fluctuations:

$$\Phi_i^{SUPG} = \boldsymbol{\beta}_i^{SUPG}\Phi^K, \quad \boldsymbol{\beta}_i^{SUPG} = \frac{1}{3} + \mathsf{K}_j\mathsf{T} \tag{1.35}$$

with $\mathsf{T} = \left(\sum_{j\in K}|\mathsf{K}_j|\right)^{-1}$. A limiter tune the streamline upwind diffusion introduced. The final distribution for such a scheme called LLxF stabilized or briefly LLxF-SUPG reads

$$\Phi_i^{LLxF-SUPG} = (1-\delta(\mathsf{u}_h))\,\Phi_i^{LLxF} + \delta(\mathsf{u}_h)\Phi_i^{SUPG}$$

The limiter is based on the energy considerations of section 1.3. Ricchiuto and Bollerman [2009] propose the following heuristic definition

$$\delta = \min\left(1, \frac{E_{ref}\|u_{ref}\||K|}{|\phi_E^K|L_{ref}}\right)$$

with ϕ_E^K an approximation of the entropy/energy fluctuation and is obtained projecting the fluctuation on the entropy vector $\phi_E^K = \Psi^T\cdot\Phi^K$. When the solution is smooth enough, we can scale the energy fluctuation as $\phi_E^K \sim E_{ref}\|u_{ref}\||K|L_{ref}^{-1}$. On cells where the solution is discontinuous the energy fluctuation becomes singular, $\phi_E^K \to \infty$ and $\delta \to 0$.

Source terms

The treatment of source term is very simple and effective in the RD framework. In fact, one could think to include every source term in its wave propagation algorithm, see LeVeque [1998]. While, for FV, this could be laborious, for RD it results automatic. According to the first step of RD algorithm (1.29), general source terms can be integrated with Gaussian quadrature formula with the prescribed order of accuracy (second for us):

$$\Phi_{\mathcal{S}}^K = \int_K \mathcal{S}(\boldsymbol{x},\mathsf{u})\,d\boldsymbol{x} = |K|\sum_{q=1}^{N_q}\omega_q\mathcal{S}_q$$

Then it is added to the fluctuation and split automatically with the distribution matrix of the advective scheme. For explicit schemes however, once should take into account these terms in the stability analysis and extends the CFL condition in presence of the source.

1.6.4 Explicit RD Runge-Kutta two scheme

The accurate simulation of unsteady problems, is crucial in many coastal and geophysical applications. We search for a time discretization that gives second order of accuracy and, in particular that leads to an explicit scheme in order to compare the results with the FV-eRK2 presented in section (1.6.2). For unsteady RD, the time derivative, once added to the fluctuation, define the element's *residual*

$$\Phi^{K(t)} = \int_K \frac{d\mathsf{u}_h}{dt}\, d\boldsymbol{x} + \int_{\partial K} \mathsf{F}^j(\mathsf{u}_h) n^j ds + \int_K \mathcal{S}(\mathsf{u}_h, \boldsymbol{x})\, d\boldsymbol{x}$$

and we denote it with the same symbol of the fluctuation, adding the superscript (t) to underline the presence of the temporal derivative. We hope that this does not generate confusion. The time part leads to the appearance of a mass matrix, (see Caraeni and Fuchs [2002] for a discussion on mass matrices) that must be inverted locally:

$$\beta_i^K \int_K \frac{d\mathsf{u}_h}{dt}\, d\boldsymbol{x} = \sum_{j\in K} m_{ij}^K \frac{d\mathsf{u}_j}{dt}, \quad m_{ij}^K = \int_K w_i \varphi_j\, d\boldsymbol{x}$$

with w_i a Petrov-Galerkin test function chosen such that a coherent RD splitting for the time part is recovered $\beta_i^K = \int_K w_i\, d\boldsymbol{x}$.

In this thesis we have implemented the explicit Runge Kutta 2 or Predictor-Corrector (PC) scheme, described in Ricchiuto and Abgrall [2010]. Through an efficient mass-lumping strategy, the RD-eRK2 scheme allows to march in time explicitly and without the unnecessary additional cost of inverting the mass matrix. The algorithm follows:

1] Predictor step: for each element $K \in \mathcal{T}_h$

- Compute the residual $\Phi^{K(1)} = \Phi^K(\mathsf{u}_h^n)$.
- Distribute the residual to the nodes of K such that $\sum_{j\in K} \Phi_j^{K(1)} = \Phi^{K(1)}$
- Compute the first order prediction of the solution, denoted as u^*

$$\mathsf{u}_i^* = \mathsf{u}_i^n - \frac{\Delta t}{|C_i|} \sum_{K\in\mathcal{D}_i} \Phi_i^{K(1)}(\mathsf{u}_h^n) \tag{1.36}$$

2] Corrector step: for each element $K \in \mathcal{T}_h$

- Compute the residual

$$\Phi^{K(2)} = \frac{1}{2}\left(\Phi^K(\mathsf{u}_h^n) + \Phi^K(\mathsf{u}_h^*)\right) + \int_K \frac{\mathsf{u}_h^* - \mathsf{u}_h^n}{\Delta t} d\boldsymbol{x}$$

- Distribute the residual to the nodes of K such that $\sum_{j\in K}\Phi_j^{K(2)}=\Phi^{K(2)}$
- Compute the second order correction from

$$\mathsf{u}_i^{n+1}=\mathsf{u}_i^{*}-\frac{\Delta t}{|C_i|}\sum_{K\in\mathcal{D}_i}\Phi_i^{K(2)}(\mathsf{u}_h^{*},\mathsf{u}_h^{n}) \tag{1.37}$$

1.6.5 Residual Distribution for SWEs

Trough this manuscript, we will compare the eRK2-FV method for the SWEs proposed by Nikolos and Delis [2009] to an RD discretization of the same system recently suggested by Ricchiuto [2015]. The numerical resolution of the SWEs with RD schemes poses two major issues. The first is numerical preservation of positive depths. The second is is the accurate discretization of source terms, bathymetry and friction up to now, recalling that we would like to embed at a discrete level the exact balance in the case of lake at rest.

Positivity of water depth

For SWEs simulations, the positivity of water depth is a physical constraint. We study the LLxF fluctuation, with a limitation performed as if we were in the scalar case, which means equation by equation, or $\mathsf{R}=\mathsf{L}=\mathsf{I}^3$ in (1.34). This variant is interesting because Ricchiuto and Bollerman [2009] showed that it preserves the positivity of water depth. We repeat the proof for sake of clarity *only* for the predictor step. The reader can find the complete proof in the reference. The distributed fluctuation for the mass equation writes

$$\begin{aligned}\phi_i^{LLxF} &= \frac{\phi_i^{LLxF}}{\phi^K}\frac{\phi^K}{\phi_i^{LxF}}\phi_i^{LxF}=\underbrace{\frac{\beta_i^{LLxF}}{\beta_i^{LxF}}}_{\gamma_i}\left(\frac{1}{3}\phi^K+\frac{\alpha^K}{3}\sum_{j\in K,\neq i}(h_i-h_j)\right)\\ &= \gamma_i\left(\frac{1}{6}\sum_{j\in K}h_j\boldsymbol{u}_j\cdot\boldsymbol{n}_j+\frac{\alpha^K}{3}\sum_{j\in K,\neq i}(h_i-h_j)\right)\\ &= \gamma_i\left(\frac{1}{6}\boldsymbol{u}_i\cdot\boldsymbol{n}_i+\frac{2\alpha^K}{3}\right)h_i+\sum_{j\in K,j\neq i}\gamma_i\left(\frac{1}{6}\boldsymbol{u}_j\cdot\boldsymbol{n}_j-\frac{\alpha^K}{3}\right)h_j\end{aligned}$$

we remark that we have put the LLxF fluctuation into the form (1.32) which is particularly suited to find local bounds of the numerical solution. Each element contributes with its residual to the update

$$\begin{aligned} h_i^* &= h_i^n - \frac{\Delta t}{|C_i|}\phi_i^{LLxF} \\ &= \left(1 - \frac{\Delta t \gamma_i}{|C_i|}\left(\frac{\boldsymbol{u}_i \cdot \boldsymbol{n}_i}{6} + \frac{2\alpha^K}{3}\right)\right) h_i + \sum_{j\in K, j\neq i} \frac{\Delta t \gamma_i}{|C_i|}\left(\frac{\alpha^K}{3} - \frac{\boldsymbol{u}_j \cdot \boldsymbol{n}_j}{6}\right) h_j^n \\ &= a_{ii}^K h_i^n + \sum_{j\in K, j\neq i} a_{ij}^K h_j^n \end{aligned}$$

$h_i^* > 0$ is positive if the extra-diagonal coefficients $a_{ij}^K > 0$, which fix a lower bound to the LxF dissipation parameter

$$\alpha^K > \frac{1}{2}\boldsymbol{u}_j \cdot \boldsymbol{n}_j, \quad \forall j \in K, \neq i$$

We have neglected $\frac{\Delta t \gamma_i}{|C_i|}$ because it is always positive. Moreover it is necessary to enforce a CFL condition to have the positivity of the diagonal coefficient $a_{ii} > 0$

$$\Delta t < \frac{3|C_i|}{\sum_{K\in\mathcal{D}_i}\left(\frac{1}{2}\boldsymbol{u}_i \cdot \boldsymbol{n}_i + 2\alpha^K\right)}$$

Ricchiuto [2015] shows that the PC scheme given by (1.36) and (1.37) with LLxF fluctuation Φ_i^{LLxF}, verifies the positivity of water depth $h_i^{n+1}\, \forall i \in \mathcal{T}_h$, for the following choice of the LxF parameter and time step

$$\alpha^K = \frac{1}{2}\max_{j\in K}\|\boldsymbol{u}_j\| \max_{j\in K}\|\boldsymbol{n}_j\|, \quad \Delta t < \min_{i\in\mathcal{T}_h}\frac{|C_i|}{\sum_{K\in\mathcal{D}_i}\alpha^K} \tag{1.38}$$

if the limitation procedure (1.34) is carried out equation by equation, that is $\mathsf{R} = \mathsf{L} = \mathsf{I}^3$ in (1.34). Ricchiuto [2015] uses this result to avoid negative depth and selects how to perform the limiting according to the water depth. In particular when $h \ll 1$ he switchs to a scalar version of LLxF scheme for each equation

$$\begin{cases} \mathsf{R} = \mathsf{L} = \mathsf{I}^3 & \text{if } \min_{j\in K} h_j < C_H \\ \mathsf{R}, \mathsf{L} & \text{else} \end{cases}$$

Well-Balancedness

For FV we have seen that bathymetric source terms are approximated in the same fashion of the fluxes, in order to recover exactly at a discrete level the balance between topographic and hydrostatic terms. This idea have been incorporated into the residual approach almost naturally, as done for any other source terms, see section (1.6.3) and Ricchiuto *et al.* [2007]; Ricchiuto and Bollerman [2009] and references therein). In the lake at rest case $\mathsf{u} = [\eta_0 -$

$b\ 0\ 0]^T$, one should check that trivially $\Phi^K = 0$. That is, that the boundary integral of the hydrostatic term in the flux balances exactly the integral of the topographic source:

$$\begin{aligned} \Phi^{\mathsf{K}} &= \int_{\partial K} \mathsf{F}^j(\mathsf{u}_h) n^j \, ds + \int_K \mathcal{S}^b(\mathsf{u}_h,\, b_h) \\ &= \begin{bmatrix} 0 \\ \int_{\partial K} \frac{1}{2} g h_h^2 n^j ds \end{bmatrix} + \begin{bmatrix} 0 \\ \int_K g h_h \frac{\partial b_h}{\partial x^i} d\boldsymbol{x} \end{bmatrix} = 0 \end{aligned} \tag{1.39}$$

This is true under the hypotheses of exact integration w.r.t. the linear variation of depth and bathymetry. In practice we have transformed

$$\int_{\partial K} \frac{1}{2} g h_h^2 n^j ds = \int_K g h_h \frac{\partial h_h}{\partial x^i} d\boldsymbol{x}$$

and then we used a second order quadrature formula to approximate both the hydrostatic and the bathymetric term.

1.6.6 Wet/Dry cell treatment

The treatment of the wetting/drying phenomenon is crucial in many coastal applications. For example, in tsunami simulation, rundown and flooding stages are crucial and must be accurately reproduced to predict runup heights and inundated areas after the tsunami attack. Wetting and drying in Shallow Water simulations is an active subject of research which is discussed thoroughly in Brufau *et al.* [2002, 2004]; Delis *et al.* [2008] and Ricchiuto and Bollerman [2009] for the FV and RD methods respectively. The interested reader is referred to these references for all details. We limit ourselves to highlight the main difficulties and to introduce the implementation details along with the reference. First, we define a *partially dry cell* or *wet/dry cell* as an element of triangulation on which the water depth passes from a positive value to zero. We 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 $h_i \leq C_H$. This artifice avoids the computations of unphysical wet cells with $0 < h \ll 1$. 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 $C_H \ll C_U$. C_H is a small quantity compared to the scale of the phenomenon, for real scale simulation is in the order of $C_H \leq 10^{-4}\, m$. The choice of C_U is less trivial as it can depends on the particular problem, Ricchiuto and Bollerman [2009] relates this threshold coefficient to the mesh size h_K:

$$C_U = \frac{h_K^2}{L_{ref}}$$

with the domain reference length L_{ref} already defined above. A second aspect concerns the preservation of Well-Balanced property in presence of partially dry cells. On these particular cells, the linear representation of the water depth generates a spurious momentum flux that spoils WB. As in Brufau *et al.* [2002], the problem is cured by using, only for the dry nodes, a modified value of the bathymetry. In particular, in the computation of the element's residual (RD) or interface's flux (FV), we define for wet-dry cells, the maximum water height at the wet nodes

$$\eta_{max}^K = \max_{j \in K, h_j > 0} (h_j + b_j)$$

and based on this values we artificially correct of the bathymetry value at dry nodes according to the following modification

$$b_i = \eta_{max}^K, \quad \text{if } b_i^K > \eta_{max}$$

such that well-balanced is recovered.

Chapter 2

Well-Balanced conservative methods for the Shallow Water equations in ALE form

In the first part of the chapter we introduce basic kinematics relationships that are involved in the formulation of the SWEs in Arbitrary Lagrangian Eulerian (ALE) framework. The reader can find these formulas in almost every book on continuum mechanics, we refer for example to Chadwick [1976]. In the second part we use the kinematics relationships to write the SWEs in a framework which is not Eulerian nor Lagrangian but it is arbitrary. We discuss different forms of the ALE-SWEs with respect to the accomplishment of Well-Balancedness and mass conservation at a discrete level. Finally it is presented a possible implementation of ALE Finite Volume and ALE Residual Distribution. A few test cases will confirm theoretical expectations.

2.1 Basic kinematics

Since the fluid motion is columnar, we will use the word *fluid column* or *fluid particle* without distinction. A point Q occupies a certain position identified by the vector $\boldsymbol{x}$ which can be expressed in Cartesian coordinates with standard summation

$$\boldsymbol{x}(Q) = x^1\boldsymbol{e}_1 + x^2\boldsymbol{e}_2 \equiv x^i\boldsymbol{e}_i$$

$\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ define the horizontal plane where the undisturbed fluid lies. We define also $\boldsymbol{e}_3$ as the the vertical axis in the upward direction. Along $\boldsymbol{e}_3$, we measure the fluid depth $h(\boldsymbol{x})$ and the bathymetry $b(\boldsymbol{x})$. The fluid free surface follows as $\eta = h + b$, see figure 2.1. Within the Shallow Water context, we call material fluid $\mathfrak{B}$, a set of fluid columns that can be put in bijective correspondence with the points of a certain configuration $\mathcal{B}$ lying on a plane. We distinguish the particles $P \in \mathfrak{B}$ from the points $Q \in \mathcal{B}$. Please note

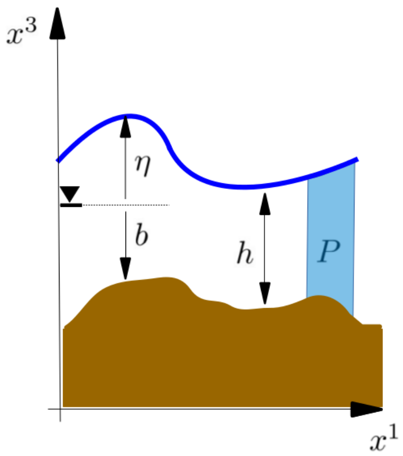

Figure 2.1: Definition of axis x^1/x^3 , water depth h, bathymetry b, free surface η with respect to an arbitrary origin. On the right, a water column/particle P.

that the bold font will be used in the following for vector and tensors. In the initial or *reference* configuration we use index $(\cdot)_0$ to highlight that time is fixed, $t = 0$. Although the notation changes from text to text, all along this manuscript, particles in the reference configuration are labeled by coordinates in capital letter $\boldsymbol{X}(Q_0)$, differently from particles in a specific configuration that are labeled by low case letters $\boldsymbol{x}(Q)$. We assume the existence of a function such that

$$F_0 : \mathfrak{B} \to \mathcal{B}_0, \quad \boldsymbol{X} = F_0(P) \tag{2.1}$$

with $\exists\, F_0^{-1}$. As sketched in figure (2.2), the body configuration changes with time. If, at each value of time is associated a unique configuration $\mathcal{B}(t)$, then the family of configurations is called motion of the fluid $\mathfrak{B}$:

$$F : \mathfrak{B} \to \mathcal{B}, \quad \boldsymbol{x} = F(P, t) \tag{2.2}$$

Combining relations (2.1) and (2.2) we obtain $\boldsymbol{x} = F(F_0^{-1}(\boldsymbol{X}), t)$ or

$$B : \mathcal{B}_0 \to \mathcal{B} \quad \boldsymbol{x} = B(\boldsymbol{X}, t)$$

We assume the function B to be continuously differentiable, thus a smooth Jacobian of transformation can be defined

$$\boldsymbol{J}_B = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}, \quad J_B = \det \boldsymbol{J}_B$$

Particle collisions is avoided through the requirement $\exists \boldsymbol{J}_B^{-1}$, $\forall P$, or equivalently $\det \boldsymbol{J}_B \neq 0$. Furthermore we add the physical condition that material volumes are positives, thus the determinant of the Jacobian will be strictly positive $J_B > 0$. A sketch of these relations can be found in figure (2.2).

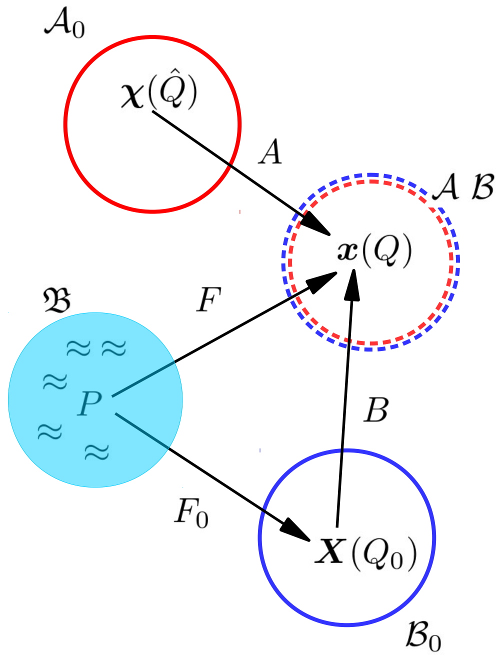

Figure 2.2: A sketch of Eulerian, Lagrangian and ALE configurations.

2.2 Lagrangian and Eulerian descriptions

The fluid $\mathfrak{B}$ is endowed with various physical properties associated to the particles P. These properties, represented by scalar ϕ, vector $\boldsymbol{v}$ and tensor fields $\boldsymbol{T}$, can be either defined with respect to the reference configuration or to the actual one. In the former case, called *Lagrangian* framework, the independent variable is $\boldsymbol{X}$. For a scalar $\phi : \mathcal{B} \to \mathbb{R}$ or $\phi_0 : \mathcal{B}_0 \to \mathbb{R}$ this corresponds to

$$\phi = \phi \circ B = \phi_0 \left(\boldsymbol{X}, t\right) \quad \text{Lagrange} \tag{2.3}$$

The pullback operator is useful because it transfer the field set from the actual description to the reference one, highlighting the transformations $\phi = \phi \circ B : \mathcal{B}_0 \to \mathcal{B} \to \mathbb{R}$. The second part of the statement (2.3) is just a shorter notation for the same concept $\phi = \phi_0 \left(\boldsymbol{X}, t\right) : \mathcal{B}_0 \to \mathbb{R}$ and will be used in the following. In presence of fluid motion, the material derivative of ϕ is the time rate of change of ϕ measured by an observer which moves with the particle labeled with X, where $\boldsymbol{X}(Q_0)$ is the particle's position in the reference configuration. We will shorten it

$$\frac{d\phi}{dt} = \left.\frac{\partial \phi}{\partial t}\right|_X \quad \text{Lagrange} \tag{2.4}$$

Analogously, if we take $\boldsymbol{x}$ as independent variable

$$\phi = \phi_0 \circ B^{-1} = \phi \left(\boldsymbol{x}, t\right) \quad \text{Euler}$$

and the time derivative with respect to a fixed point labeled with x, is

$$\frac{\partial \phi}{\partial t} = \left.\frac{\partial \phi}{\partial t}\right|_x \quad \text{Euler}$$

The velocity of the particle at point Q, $\boldsymbol{u} = u^j \boldsymbol{e}_j$ is calculated applying the aforementioned definition of Lagrangian derivative (2.4), to the position $\boldsymbol{x}(Q)$. Its components write

$$u^j = \left.\frac{\partial x^j}{\partial t}\right|_X$$

Using the pullback one can write the scalar ϕ with respect to the actual coordinates and then derive:

$$\frac{d\phi}{dt} = \left.\frac{\partial \phi_0}{\partial t}\right|_X \circ B^{-1} = \left.\frac{\partial \phi}{\partial t}\right|_x + \frac{\partial \phi}{\partial x^j} \left.\frac{\partial x^j}{\partial t}\right|_X = \left.\frac{\partial \phi}{\partial t}\right|_x + \frac{\partial \phi}{\partial x^j} u^j$$

we get an important relationship between the Lagrangian time derivative and the Eulerian one. Similar arguments can be used to transport time derivatives of a vector $\boldsymbol{v} = v^i \boldsymbol{e}_i$

$$\frac{dv^i}{dt} = \left.\frac{\partial v^i}{\partial t}\right|_X \circ B^{-1} = \left.\frac{\partial v^i}{\partial t}\right|_x + \frac{\partial v^i}{\partial x^j} u^j$$

During fluid motion the material volumes transforms according to $dx^m = J_B dX^m$, with $dx^m = dx^1 dx^2$. From this fact stems the following relationship

$$\left.\frac{\partial J_B}{\partial t}\right|_X = J_B \frac{\partial u^j}{\partial x^j} \tag{2.5}$$

or, more compactly $\partial_t|_X J_B - J_B \nabla \cdot \boldsymbol{u} = 0$. The proof can be found in every textbook on continuum mechanics.

2.3 Arbitrary Lagrangian Eulerian kinematics

To embed adaptive mesh deformation in the numerical solution of (1.1)(1.2) an appropriate Arbitrary Lagrangian Eulerian (ALE) formulation will be used. The objective of the next section is to recall some basic aspects related to ALE. For more details concerning the ALE formalism, the interested reader can refer to the original paper of Donea [1983] or to a recent review Donea *et al.* [2004].

We introduce an arbitrary configuration $\mathcal{A}$ and we will refer to it as Arbitrary Lagrangian Eulerian (ALE). For us, $\mathcal{A}$ will be the configuration which the mesh is attached to. As for the body configuration $\mathcal{B}$, it belongs to a Cartesian plane composed of points Q. The arbitrary configuration in the reference or initial state is denoted by $\mathcal{A}_0$ and the points $\hat{Q} \in \mathcal{A}_0$. If capital letters are used for the reference configuration, and lower case letters label the actual configuration, Greek letters are used to describe points in the ALE configuration, $\boldsymbol{\chi}(\hat{Q})$. Moreover we are interested in arbitrary meshes attached to a configuration that changes with time $\mathcal{A}(t)$. If, to each value of time we associate a unique arbitrary configuration, we could assume the existence of a function

$$A : \mathcal{A}_0 \to \mathcal{A} \quad \boldsymbol{x} = A(\boldsymbol{\chi}, t) \tag{2.6}$$

As done for B, we require that A is continuously differentiable, and consider the Jacobian matrix:

$$\boldsymbol{J}_A = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\chi}}, \quad J_A = \det \boldsymbol{J}_A > 0 \ \forall \hat{Q} \tag{2.7}$$

which means that also $\mathcal{A}$ cannot admit fractures, collisions and negative volumes. From the point of view of a mesh cell, the occurrence of negative volumes is also known as *tangling*, an urgent problem in ALE simulations. For example, very often, when solving the motion of $\mathcal{A}(t)$, we cannot impose explicitly the positiveness of the J_A, thus we cannot avoid mesh tangling a priori. This will be detailed in Chapter 3. For now, analogously to the particle/Lagrangian velocity, we introduce the mesh/ALE velocity, which is the velocity of the point Q which was lying, in the arbitrary reference domain, in $\boldsymbol{\chi}(\hat{Q})$

$$\sigma^j = \left.\frac{\partial x^j}{\partial t}\right|_\chi$$

$\boldsymbol{\sigma} = \sigma^j \boldsymbol{e}_j$ is arbitrary and independent of the flow (in general). In the previous paragraph we have transformed time derivative from the Lagrangian to the Eulerian description $\mathcal{B}_0 \rightarrow \mathcal{B}$. Recalling briefly that, for the scalar ϕ, the following relations holds

$$\begin{aligned} \phi &= \phi \circ A = \phi_0 \left(\boldsymbol{\chi}, t \right) \quad \text{ALE} & (2.8) \\ \phi &= \phi_0 \circ A^{-1} = \phi \left(\boldsymbol{x}, t \right) \quad \text{Euler} & (2.9) \end{aligned}$$

We can do the transformation for the arbitrary configuration $\mathcal{A}_0 \rightarrow \mathcal{A}$

$$\left.\frac{\partial \phi}{\partial t}\right|_{\chi} = \left.\frac{\partial \phi_0}{\partial t}\right|_{\chi} \circ A^{-1} = \left.\frac{\partial \phi}{\partial t}\right|_{x} + \left.\frac{\partial \phi}{\partial x^j} \frac{\partial x^j}{\partial t}\right|_{\chi} = \left.\frac{\partial \phi}{\partial t}\right|_{x} + \frac{\partial \phi}{\partial x^j} \sigma^j \tag{2.10}$$

and an interesting relationship emerge between Lagrangian and ALE time derivative:

$$\frac{d\phi}{dt} = \left.\frac{\partial \phi}{\partial t}\right|_{\chi} + \frac{\partial \phi}{\partial x^j} \left(u^j - \sigma^j \right) \tag{2.11}$$

The above relationship holds also for vector

$$\frac{dv^i}{dt} = \left.\frac{\partial v^i}{\partial t}\right|_{\chi} + \frac{\partial v^i}{\partial x^j} \left(u^j - \sigma^j \right) \tag{2.12}$$

The correct computation of ALE domain volumes in the transformation is assured by the relation $dx^m = J_A d\chi^m$ or by the so-called Geometric Conservation Law (GCL), which generalize (2.5):

$$\left.\frac{\partial J_A}{\partial t}\right|_{\chi} = J_A \frac{\partial \sigma^j}{\partial x^j} \tag{2.13}$$

More compactly $\partial_t|_X \, J_A - J_A \nabla \cdot \boldsymbol{\sigma} = 0$. This can be interpreted as a constraint that ALE Jacobian and the domain velocity both must comply during the motion. As we will see in while, people involved in ALE simulations, are particularly interested to respect explicitly the GCL. If integrated over a close domain (2.13) represents a sort of volume/area conservation, which one must also preserve at the discrete level.

2.3.1 ALE remap

Lastly, we note that for a steady function in Eulerian framework $\phi = \phi(\boldsymbol{x})$, we can write (2.10) imposing $\partial_t \phi = 0$

$$\left.\frac{\partial \phi}{\partial t}\right|_{\chi} = \frac{\partial \phi}{\partial x^j} \sigma^j \tag{2.14}$$

This last equation represents the time variation of the function $\phi(\boldsymbol{x}(t))$ measured from an observer which is following the ALE domain motion $\boldsymbol{x} = A(\boldsymbol{\chi}, t)$.

Summing eq. (2.14) (pre-multiplied by J_A) to eq. (2.13) (pre-multiplied by ϕ) leads to the conservative ALE remap equation for the function ϕ

$$\left.\frac{\partial J_A \phi}{\partial t}\right|_{\chi} - J_A \frac{\partial \phi \sigma^j}{\partial x^j} = 0 \tag{2.15}$$

2.4 Shallow Water equations in ALE form

We move away from a pure kinematic and inertial description of the flow, and we go back to hydrodynamics. Let us consider a shallow fluid column P in $\mathfrak{B}$, with depth h and flow velocity $\boldsymbol{u}$. Forces of different nature act on P. In particular pressure term participates microscopically to momentum balance and they can be added to advective flux. The bottom topography exerts also contact forces on the fluid, in particular pressure and frictional forces.

The transport formulas (2.11)(2.12) and the two volume conservation constraints, namely the fluid volume (2.5) and the geometric conservation law (2.13), are used here to transform the field equations

$$\left.\frac{\partial J_B h}{\partial t}\right|_{X} = 0 \quad \text{and} \quad \left.\frac{\partial J_B h u^i}{\partial t}\right|_{X} = J_B F^i$$

in a general arbitrary framework coincident with the domain motion. The SWEs write in this case (see also the Eulerian equations in (1.1)):

$$\left.\frac{\partial J_A h}{\partial t}\right|_{\chi} + J_A \frac{\partial}{\partial x^j}\left(h u^j - h\sigma^j\right) = 0 \tag{2.16}$$

$$\left.\frac{\partial J_A h u^i}{\partial t}\right|_{\chi} + J_A \frac{\partial}{\partial x^j}\left(T^{ij} - h u^i \sigma^j\right) + J_A S^i = 0 \tag{2.17}$$

all vectors and tensors definitions hu^i, T^{ij}, S^i were already given in section1.1 of chapter 1. Always as in chapter 1, we provide the vector form of the SWEs

$$\left.\frac{\partial J_A \mathsf{u}}{\partial t}\right|_{\chi} + J_A \frac{\partial}{\partial x^j}\left(\mathsf{F}^j - \sigma^j \mathsf{u}\right) + J_A \mathcal{S}(\boldsymbol{x}, \mathsf{u}) = 0, \tag{2.18}$$

$$\mathsf{u} = \begin{bmatrix} h \\ hu^i \end{bmatrix}, \; \mathsf{F}^j = \begin{bmatrix} hu^j \\ T^{ij} \end{bmatrix}, \; \mathcal{S} = \underbrace{\begin{bmatrix} 0 \\ gh\frac{\partial b}{\partial x^i} \end{bmatrix}}_{\mathcal{S}^b} + \underbrace{\begin{bmatrix} 0 \\ c_F u^i \end{bmatrix}}_{\mathcal{S}^f} \tag{2.19}$$

Equations (2.18)(2.19) is a non-homogeneous hyperbolic system of partial differential equations. In particular, given any vector $\boldsymbol{\xi} = \xi^i \boldsymbol{e}_i$ the flux Jacobian $\mathsf{K}^{ALE} = \frac{\partial}{\partial \mathsf{u}}(\mathsf{F}^j - \sigma^j \mathsf{u})\xi^j$ is

$$\mathsf{K}^{ALE}(\mathsf{u}, \boldsymbol{\sigma}, \boldsymbol{\xi}) = \mathsf{K} - \sigma^j \xi^j \mathsf{I}^3 \tag{2.20}$$

It admits a full set of real eigenvalues in the form $\lambda(\mathsf{u}, \boldsymbol{\sigma}, \boldsymbol{\xi})$, namely

$$\lambda_{1,3} = (u^j - \sigma^j)\xi^j \pm c\|\boldsymbol{\xi}\| \quad \lambda_2 = (u^j - \sigma^j)\xi^j$$

and linearly independent eigenvectors that does not change with respect to the standard Eulerian SWEs ones.

2.5 Discrete ALE equations

The implementation of ALE Finite Element or Finite Volume poses a new issue: the preservation of time accuracy and non-linear stability properties on moving meshes. A naif extension of the fixed grid algorithm (without worrying how to choose the mesh velocity, for example) does not assure that the GCL requirement is verified exactly and this may lead to spoil time accuracy and to the rise of numerical instabilities. Guillard and Farhat [2000] proved that satisfying the GCL is a sufficient condition for first order time accurate simulation on moving meshes. We remark that, imposing a uniform flow in (2.18), gives back the GCL. Thomas and Lombard [1979] proposed in fact to replace the GCL with the constraint of reproducing exactly uniform flows. The compliance of the GCL discretely is referred to as Discrete GCL (DGCL). In our case, a numerical method approximating solutions of (2.18)(2.19) on a moving mesh, is said to verify a DGCL if for $\mathcal{S} = 0$, the state $\mathsf{u} = \mathsf{u}_0 = const$ is an exact solution of the discrete equations. As we said, a numerical method verifies the DGCL if it also embeds an exact discretization of the GCL (2.13). Even if one could think that it is trivially important to conserve the total area of the computational domain, still the importance/benefit of the GCL remains quite controversial, see the summery of Etienne *et al.* [2009]. This debate is important but, for the SWEs, the relevance of the DGCL lies in the fact that, while avoiding the appearance of spurious oscillations, it becomes a necessary condition to preserve some exact steady state on moving meshes, which we are interested to. DGCL schemes will be of key importance to retain Well-Balancedness on moving meshes.

At a discrete level we work on integral equations, thus it is important to consider the integral form of the GCL. For example for a general volume V:

$$\left.\frac{\partial}{\partial t}\right|_{\chi} \int_V d\boldsymbol{x} = \int_V \sigma^m n^m ds \tag{2.21}$$

where the left hand side is discretized with the time scheme. It is thus crucial to choose edge fluxes (right hand side) such that the above expression is an identity. Of course at a discrete level, the closure of the problem will depend on the specific time scheme used. In general we can say that the unknown are the grid velocities at the edges and the configuration on which we integrate. Regarding these quantities in particular, Mavriplis and Yang [2006] show that the satisfaction of the DGCL is not a sufficient condition to have the prescribed high order time accuracy. In particular edge velocities and configuration should be chosen coherently with the time scheme. That is, using the quadrature points of the time integrator. Before going on we note that the grid velocity is not completely free. In particular besides the constraints on the positivity of the volume, we limit ourselves to a linear approximation of the mesh faces. This means that only straight edges are allowed (no curved mesh). As a consequence we take the velocity space to be contained in the P^1 linear finite element space

$$\boldsymbol{\sigma} = \sum_{j\in\mathcal{T}_h} \varphi_i \boldsymbol{\sigma}_j \tag{2.22}$$

with, as usual, $\{\varphi_i\}_{i\in\mathcal{T}_h}$ being the standard piecewise linear P^1 Finite Element kernel. $\boldsymbol{\sigma}_j$ is the mesh velocity vector of the node j of the triangulation. Still the grid velocity time history $\boldsymbol{\sigma}(t)$ is unknown and no hypothesis are made.

2.5.1 DGCL closure for eRK2

We consider explicit Runge Kutta type schemes, as the one presented in Chapter 1. For every RK-stage the time lapse is just one, we go from n to $n+1$ and the time discrete approximation of (2.21) for a polygon V, namely the DGCL, writes

$$\left[\int_V d\boldsymbol{x}\right]^{n+1} - \left[\int_V d\boldsymbol{x}\right]^{n} = \Delta t \int_{\partial V} \sigma^m n^m ds$$

We can further manipulate, decomposing the left-hand side in the contributions of each volume's edge

$$|V^{n+1}| - |V^n| = \Delta t \sum_{j\in V} \upsilon_j \tag{2.23}$$

where $|V|$ is the area of the polygon, $j \in V$ is an index over the set of polygon's edges. The edges' velocities result:

$$\upsilon_j = \int_{\partial V_j} \sigma^m n_j^m ds \tag{2.24}$$

and they are denoted, in a Finite Volume context, as *interface velocities*. The question is how to verify (2.23) exactly, which means with an error driven by machine roundoff?

Mavriplis and Yang [2006] and Isola *et al.* [2011], propose to evaluate geometrically the edge velocities from the volume swept by the corresponding interface. We will use extensively their closure. We state the main conclusion in the form of the following preposition.

Proposition (DGCL). *A numerical method approximating eq. (2.18) verifies the DGCL constraint eq. (2.23), if each edge velocity is computed through the (signed) area swept by the j-th edge of the polygon in one timestep Δt*

$$\upsilon_j = \frac{1}{\Delta t}\int_{Q_j} d\boldsymbol{x} \tag{2.25}$$

where the quadrangle Q_j is defined by vertexes $\boldsymbol{x}_i^n$,$\boldsymbol{x}_k^n$,$\boldsymbol{x}_k^{n+1}$,$\boldsymbol{x}_i^{n+1}$, i and k denotes the two vertexes of the j-th edge. We use the convention that, if the polygon is expanding, then the area swept is positive.

Isola [2012] explicitly developed the formula (2.25), and found the popular *midpoint* closure proposed by Lesoinne and Farhat [1996]. We summarize the main result contained in the last reference: a numerical method approximating eq. (2.18) verifies the DGCL constraint or eq. (2.23) if the midpoint configuration $V^{n+1/2}$ and constant nodal grid velocities are used to computed the ALE flux, that is replacing in (2.24) the following expressions

$$n_j^m = \frac{n_j^{m,n} + n_j^{m,n+1}}{2}, \qquad \sigma_j^m = \frac{x_j^{m,n+1} - x_j^{m,n}}{\Delta t} \tag{2.26}$$

For eRK2, the closure of Mavripils and Fahart collapses. However formally, we prefer to maintains the notation of Mavriplis and Yang [2006] and (2.25) because, in chapter 4, it will prove to be more flexible.

In the Finite Element case, the DGCL and the hypothesis (2.22) ($\boldsymbol{\sigma} \in P^1$), define a constant mesh velocity divergence on the element. We take our control polygon as the element itself $V = K$. We fix a configuration $K^t = K(t)$; the DGCL writes

$$|K^{n+1}| - |K^n| \;=\; \Delta t \int_{K^t} \frac{\partial \sigma^j}{\partial x^j} d\boldsymbol{x} = \Delta t \int_K d\boldsymbol{x} \left(\frac{\partial \sigma^j}{\partial x^j}\right)_K$$

and we obtain a rule to compute the elemental divergence of the mesh velocity vector:

$$\left(\frac{\partial \sigma^j}{\partial x^j}\right)_K = \frac{\Delta |K|}{\Delta t |K^t|} \tag{2.27}$$

2.6 Combining DGCL and Well-Balancedness on moving meshes

For a balance law, the source term $\mathcal{S}$ is different from zero, and the relevant state to be preserved may not be $\boldsymbol{u} = const$ but $\eta = const$. Note in particular, that we may write (2.18) as

$$\underbrace{J_A\left(\left.\frac{\partial \mathsf{u}}{\partial t}\right|_{\chi}-\sigma^j\frac{\partial \mathsf{u}}{\partial x^j}\right)}_{H_1}+\underbrace{\mathsf{u}\left(\left.\frac{\partial J_A}{\partial t}\right|_{\chi}-\frac{\partial \sigma^j}{\partial x^j}\right)}_{H_2}+\underbrace{J_A\left(\frac{\partial \mathsf{F}^j}{\partial x^j}+\mathcal{S}\right)}_{H_3}=0$$

Existing Eulerian discretization methods do embed integral (or even local) variants of $H_3 = 0$ (C-property) and ALE discretizations 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 (2.14) to deduce that

$$H_1+\underbrace{\left.\frac{\partial b}{\partial t}\right|_{\chi}-\sigma^j\frac{\partial b}{\partial x^j}}_{H_4}=\left.\frac{\partial \eta}{\partial t}\right|_{\chi}-\sigma^j\frac{\partial \eta}{\partial x^j}$$

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

This suggests that a better form of (2.17) for computations on moving meshes, is that obtained by summing Eq. (2.15) to Eq. (2.17). This leads to a Well-Balanced ALE form of the problem reading

$$\left.\frac{\partial J_A \mathsf{w}}{\partial t}\right|_{\chi}+J_A\frac{\partial}{\partial x^j}\left(\mathsf{F}^j-\sigma^j\mathsf{w}\right)+J_A\mathcal{S}(\boldsymbol{x},\mathsf{u})=0 \tag{2.28}$$

$$\mathsf{w}=\begin{bmatrix}\eta\\ hu^i\end{bmatrix} \tag{2.29}$$

In this case one can do much better job in the approximation of the lake at rest solution. In particular, we can write (2.28) as

$$\underbrace{J_A\left(\left.\frac{\partial \mathsf{w}}{\partial t}\right|_{\chi}-\sigma^j\frac{\partial \mathsf{w}}{\partial x^j}\right)}_{H_1+H_4}+\underbrace{\mathsf{w}\left(\left.\frac{\partial J_A}{\partial t}\right|_{\chi}-\frac{\partial \sigma^j}{\partial x^j}\right)}_{H_2}+\underbrace{J_A\left(\frac{\partial \mathsf{F}^j}{\partial x^j}+\mathcal{S}\right)}_{H_3}=0$$

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 WB, we will be able to satisfy all the compatibility requirements, and preserve steady equilibria independently on the mesh movement strategy.

As a particular case and for completeness, we recall the pre-balanced form of the Shallow Water equations of Rogers *et al.* [2003] which is obtained by introducing in (2.28) the modified flux and source functions

$$\tilde{\mathsf{F}}^j = \begin{bmatrix} hu^j \\ \tilde{T}^{ij} \end{bmatrix}, \; \tilde{\mathcal{S}}^b = \begin{bmatrix} 0 \\ g\eta \frac{\partial b}{\partial x^i} \end{bmatrix} \tag{2.30}$$

with $\tilde{T}^{ij} = huu^{ij} + \frac{1}{2} g\left(\eta^2 - 2\eta b\right)$. These definitions lead to the pre-balanced (PB) ALE form of the shallow-water system reading

$$\left.\frac{\partial J_A \mathsf{w}}{\partial t}\right|_{\chi} + J_A \frac{\partial}{\partial x^j}\left(\tilde{\mathsf{F}}^j - \sigma^j \mathsf{w}\right) + J_A \tilde{\mathcal{S}}(\boldsymbol{x}, \mathsf{u}) = 0 \tag{2.31}$$

We recall that the this form satisfies the relation, see Rogers *et al.* [2003]

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

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

2.7 Mass conservation on moving meshes

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

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

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

$$H(t) - H(0) + \int \int_{\partial\Omega(t)} (hu^j - h\sigma^j) n^j \, ds dt + B(t) - B(0) - \int \int_{\partial\Omega(t)} b\sigma^j n^j \, ds\, dt = 0 \tag{2.32}$$

which states that, modulo the boundary conditions, we have conservation over the full domain if the ALE remap equation (2.15) is satisfied, namely if

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

So a scheme approximating (2.28) will be exactly mass conservative only if the bathymetry is evolved according to an integral form of an ALE remap (2.15). This is the strategy proposed in Zhou *et al.* [2013a]. However, as pointed out in the same paper, this approach leads to changes in the bathymetric altitudes which will depend on the scheme. For example, substantial smoothing of the bed slopes is observed. To deal with this issue, in Zhou *et al.* [2013a] 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 $\boldsymbol{\sigma} \cdot \boldsymbol{n} = 0$ is verified. We can write the mass error at time t as

$$E_{mass} = H(t) - H(0) + \int \int_{\partial\Omega} hu^j n^j \, ds \, dt = B(0) - B(t)$$

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 split in integrals over the set of median dual cell areas. In case of standard FV or P^1 RD discretizations

$$B(t) = \sum_{i \in \mathcal{T}_h} \int_{C_i(t)} b \, d\boldsymbol{x} = \sum_{i \in \mathcal{T}_h} b_i(t) |C_i(t)| \tag{2.33}$$

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

$$b_i = \sum_{f=1}^{N_q} \omega_q b(\boldsymbol{x}_q(t)) \tag{2.34}$$

in figure 2.3 we provide a sketch of the treatment that we propose for the bathymetry. 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. In case we employ only one quadrature point that coincide with the node $\boldsymbol{x}_q = \boldsymbol{x}_i$ we get back to the standard choice (2.33).

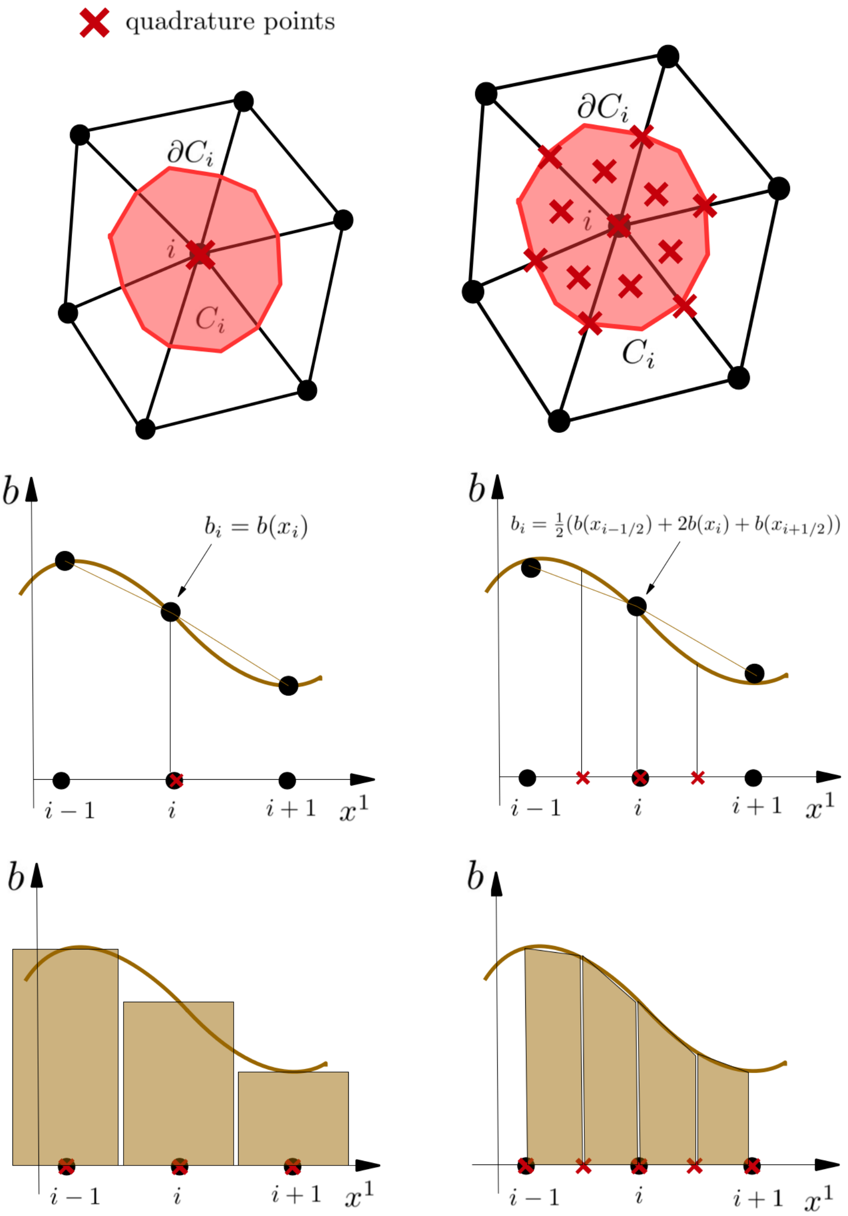

Figure 2.3: Computing the bathymetry nodal values with the standard FV method. pointwise value(left) and the method proposed here (right). Top) quadrature points. Middle) 1D sketch of bathymetry values that are computed from quadrature rule (2.34). Bottom) 1D sketch of analytical and computed bathymetry integral.

2.8 Mass conservation and Well-Balancedness in presence of wet/dry cells

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

$$\int_{\Omega_W(t)} \eta(\boldsymbol{x}(t), t)\, d\boldsymbol{x} - \int_{\Omega_{W,X}} \eta(\boldsymbol{X}, 0)\, d\boldsymbol{x} + \int \int_{\partial\Omega_W(t)} \left(hu^j - \eta\sigma^j\right) n^j \, ds\, dt = 0$$

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

$$H(t) - H(0) + \int \int_{\partial\Omega \cap \partial\Omega_W} hu^j n^j \, ds\, dt + B_W(t) - B_W(0) - \int \int_{\partial I_W(t)} b\sigma^j n^j \, ds\, dt = 0$$

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

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

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:

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

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.

A second non trivial issue related to wetting/drying and moving meshes is the Well-Balancedness. 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

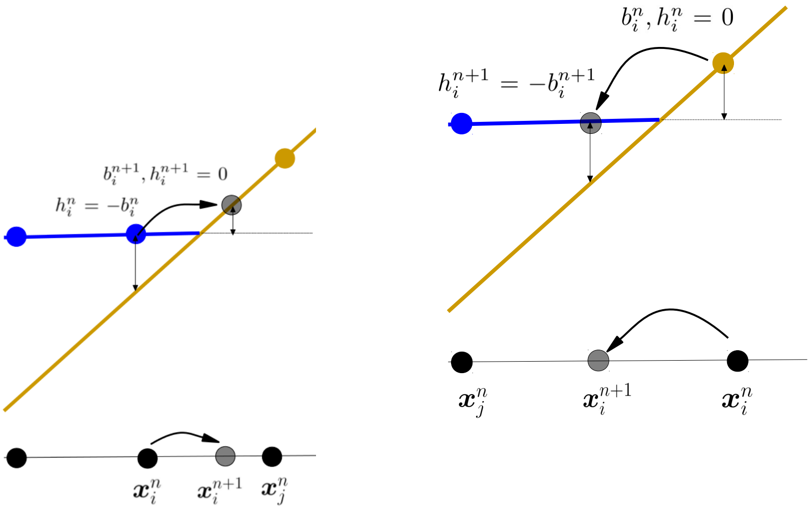

Figure 2.4: Preservation of the lake at rest on wet/dry cells. Left: a node is passing from wet to dry. Right. a node is passing from dry to wet.

water depth is computed from the free surface level obtained from the explicit updates of the well balanced ALE schemes. Given values of η_i^n, and η_i^{n+1}, obtained from the discretization of the system 2.28 or equivalently 2.31, and of b_i^n, and b_i^{n+1}, obtained using the quadrature approach, we proceed as follows

1. $\forall i \in \mathcal{T}_h$ compute the maximum water level of wet nodes belonging to the neighbor cells:

   $$\eta_{max,i} = \max_{K \in \mathcal{D}_i} \max_{\substack{j \in K \\ h_j > C_H}} \eta_j$$

   C_H was introduced in section 1.6.6 as a small threshold ($C_H \ll 1$) to define dry nodes; if $h_i \leq C_H$, then $h_i = 0$.

2. $\forall i$ set

   $$\Delta_{b_i} = b_i^* - b_i^n \tag{2.35}$$

   $$\text{with } b_i^* = \begin{cases} \eta_{max,i} & \text{if } i \text{ is dry and } b_i^n + C_H > \eta_{max,i} \\ b_i^n & \text{otherwise} \end{cases}$$

3. Compute the new water depth as

   $$h_i^{n+1} = \max(0, \eta_i^{n+1} - b_i^{n+1} + \Delta_{b_i}) \tag{2.36}$$

In the lake at rest, the value of η^{n+1} obtained from the discretization of the system 2.28 or equivalently 2.31 and to be substituted in (2.36) is $\eta_i^{n+1} = \eta_i^n$ $\forall i \in \mathcal{T}_h$ (the flux/residuals are zero everywhere). As represented in figure 2.4

there are basically two situations for the lake at rest case in presence of a mesh which is arbitrary moving. First a node can move from the wet side the dry side of the interface. From step 2, a dry node has $\Delta_{b_i} = 0$ (step 2 is not active). So we have only step 3 that writes

$$\begin{aligned} h_i^{n+1} &= \max(0, \eta_i^{n+1} - b_i^{n+1} + \Delta_{b_i}) = \max(0, \underbrace{\eta_i^n - b_i^{n+1}}_{<0}) \\ &= 0 \end{aligned}$$

which is the correct value in the dry areas. Secondly there is the vice-versa: a node moves from the dry to the wet side of the interface. This time, step 2 enforces the lake at rest:

$$\begin{aligned} h_i^{n+1} &= \eta_i^{n+1} - b_i^{n+1} + \Delta_{b_i} = b_i^n - b_i^{n+1} + \eta_{max,i} - b_i^n \\ &= \eta_{max,i} - b_i^{n+1} \end{aligned}$$

which is the exact water depth. This guarantees that a constant flat free surface level is exactly preserved also near shorelines. In figure 2.4 there is a graphical representation of the lake at rest preservation.

2.9 Notation for time dependent geometry

In the ALE case the flow 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 movement of the mesh. Keeping the same notation we have $C_i^n = C_i(t^n)$, $\partial C_i^n = \partial C_i(t^n)$ and $b^n = b(\boldsymbol{x}(t^n))$.

2.10 ALE Finite Volume for SWEs

ALE-FV are of great interest for aerodynamics problems involving moving boundaries such as aeroelastic simulations, see for example Lesoinne and Farhat [1996]. ALE methods become succeful also in the context of mesh adaptation since they can embed conservation very efficiently for both r/h-adaptation, see Isola [2012]. Instead, for the SWEs, ALE schemes are not very popular. We believe that the main reason is that many scalar quantities involved in the SWEs source term are inherently Eulerian, that is "fixed to the ground". For example, the bathymetry $b(\boldsymbol{x})$ or the friction coefficient $c_F(\boldsymbol{x})$, that also could vary in space. In the moving mesh case one should transport all this information on the new meshes. This means a time consuming computation of b^{n+1}/c_F^{n+1} at each timestep, trough an accurate interpolation or through the remap equation (2.15). For an implementation of the ALE-FV for in hydrodynamics, the reader can refer to the cell centered FV on moving mesh developed in Zhou *et al.* [2013a], which is based on a fixed grid flow evolution but with an ALE remap for the interpolation step. Here we consider the standard well balanced node-centered FV-MUSCL scheme introduced in chapter 1 and we propose an ALE extension to approximate equations in the form (2.28). We will show the equivalence between the scheme obtained using the well-balanced form (2.28), and the pre-balanced formulation (2.31) with the definition (2.30). The FV discrete evolution equations reads

$$\begin{aligned} |C_i^{n+1}|\mathsf{w}_i^* &= |C_i^n|\mathsf{w}_i^n - \Delta t \sum_{j\in\mathcal{D}_i} R_{ij}(\mathsf{w}^n, b^n) && (2.37)\\ |C_i^{n+1}|\mathsf{w}_i^{n+1} &= |C_i^n|\mathsf{w}_i^n - \frac{\Delta t}{2}\sum_{j\in\mathcal{D}_i}\left(\frac{R_{ij}(\mathsf{u}^n, b^n)}{2} + \frac{R_{ij}(\mathsf{u}^*, b^{n+1})}{2}\right) && (2.38)\end{aligned}$$

where we have again

$$R_i = \mathcal{F}_{ij} + \mathcal{S}_{ij} \tag{2.39}$$

As in chapter 1, we use the Roe-type numerical flux, this time for the well balanced formulations (2.28). ALE fluxes are evaluated at the interface:

$$\mathcal{F}_{ij}^{ALE} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}}\int_{\partial C_{ij}} \mathsf{w}\sigma^m n^m ds\, dt = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} \mathsf{w}\upsilon_{ij}\, dt$$

and they can be incorporated automatically in the resolution of the Riemann problem:

$$\begin{aligned} \mathcal{F}_{ij}(\mathsf{u}_i, \mathsf{u}_j) &= \mathbf{F}_i \cdot \boldsymbol{n}_{ij} - \mathsf{w}_i\upsilon_{ij} + \sum_{p=1,\alpha_p<0}^{m} \mathcal{W}_p \\ &= \mathbf{F}_j \cdot \boldsymbol{n}_{ij} - \mathsf{w}_j\upsilon_{ij} - \sum_{p=1,\alpha_p>0}^{m} \mathcal{W}_p \end{aligned}$$

keeping in mind that, when decomposing the solution jump (1.6.2), one should consider the velocity of simple waves is modified according to the ALE eigenvalues, see for example Isola [2012]:

$$\begin{aligned} \sum_{p=1,\alpha_p<0}^{m} \mathcal{W}_p &= (\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3)^-(\mathsf{u}_j - \mathsf{u}_i) \\ \sum_{p=1,\alpha_p>0}^{m} \mathcal{W}_p &= (\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3)^+(\mathsf{u}_j - \mathsf{u}_i) \end{aligned}$$

with I_3 the 3×3 identity matrix. At the end, the numerical flux (1.17) will result enhanced by:

$$\mathcal{F}_{ij} = \mathcal{F}_{ij}(\breve{\mathsf{u}}_i, \breve{\mathsf{u}}_j; \breve{b}_i, \breve{b}_j) = \frac{\mathbf{F}(\breve{\mathsf{u}}_j) + \mathbf{F}(\breve{\mathsf{u}}_i)}{2}\cdot\boldsymbol{n}_{ij} - \upsilon_{ij}\frac{\breve{\mathsf{w}}_j + \breve{\mathsf{w}}_i}{2} - \frac{\left|\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3\right|}{2}(\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i) \tag{2.40}$$

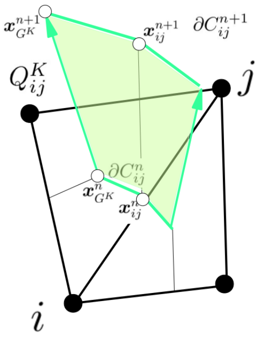

Figure 2.5: DGCL for ALE-FV. Area swept by the interface C_{ij}.

where, the $\breve{\cdot}$ values denote linearly reconstructed values of a quantity. Following the closure proposed in the previous paragraph, all the geometrical quantities needed to evaluate R_i are obtained on the mid-point averaged mesh, while the DGCL is defined by taking, in the expressions (2.23) and (2.24), $V = C_i$ and $\partial V_j = \partial C_{ij}$:

$$|C_i^{n+1}| - |C_i^n| = \Delta t \sum_{j \in \mathcal{D}_i} \upsilon_{ij} \tag{2.41}$$

The following interface interface velocity stems from the application of (2.25)

$$\begin{aligned} \upsilon_{ij} &= \int_{\partial C_{ij}} \sigma^m n^m \, ds = \sum_{K \ni i,j} \int_{\partial C^K_{ij}} \sigma^m n^m \, ds \\ &= \frac{1}{\Delta t} \sum_{K \ni i,j} \int_{Q^K_{ij}} d\boldsymbol{x} \end{aligned} \tag{2.42}$$

with the quadrilateral Q^K_{ij} defined by vertexes $\boldsymbol{x}^n_{G^K}$,$\boldsymbol{x}^n_{ij}$,$\boldsymbol{x}^{n+1}_{ij}$,$\boldsymbol{x}^{n+1}_{G^K}$, see figure 2.5.

2.10.1 Well Balancedness

The treatment of the bathymetric term does not change substantially with respect to the fixed mesh case. The definitions (1.21),(1.22) and (1.23) are valid. We only have to modify the upwind part (1.24) as follows:

$$\mathcal{S}^*_{ij} = -\frac{\mathrm{sign}(\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3)}{2}(\mathsf{A}_{ij} - \upsilon_{ij}\mathsf{I}_3)\Delta\mathsf{b}_{ij} \tag{2.43}$$

since the ALE Jacobian has an extra term due to mesh motion. With these definitions we have now the following characterization.

Proposition 1. *The finite volume discrete equations* (2.37)-(2.38) *with numerical flux and source respectively given by* (2.40)(1.21) *and with* (4.32)*,* *(1.22), (1.23), and (2.43) verify the DGCL for constant b, and the Well-Balanced 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 (2.41), 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, the proof rests on the property of the Roe average and on the fact that, on the lake at rest state, we have $\mathsf{K}_{ij} = \mathsf{A}_{ij}$. In particular, proceeding as in chapter 1

$$\begin{aligned} \mathcal{F}_{ij} &= \frac{\mathbf{F}(\breve{\mathsf{u}}_j) - \mathbf{F}(\breve{\mathsf{u}}_i)}{2} \cdot \boldsymbol{n}_{ij} + (\mathbf{F}(\breve{\mathsf{u}}_i) - \mathbf{F}(\mathsf{u}_i)) \cdot \boldsymbol{n}_{ij} - \upsilon_{ij}\mathsf{w}_0 - \frac{|\mathsf{A}_{ij} - \upsilon_{ij}\mathsf{I}_3|}{2}(\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i) \\ \mathcal{S}_{ij}^b &= \mathsf{A}_{ij}^- \Delta\mathsf{b}_{ij}^- + \frac{1}{2}\mathsf{A}_{ij}\,\Delta\mathsf{b}_{ij} - \frac{\operatorname{sign}(\mathsf{A}_{ij} - \upsilon_{ij}\mathsf{I}_3)}{2}(\mathsf{A}_{ij} - \upsilon_{ij}\mathsf{I}_3)\Delta\mathsf{b}_{ij} \end{aligned}$$

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 $\mathsf{w} = \mathsf{w}_0$.

□

2.10.2 Source term upwinding: a bridge between WB and PB form

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 (2.31) are almost identical to those presented above which are instead derived from (2.28). We neglect friction that does not play any role in deriving the results that follows. First we define

$$R_{ij} = \tilde{\mathcal{F}}_{ij} + \tilde{\mathcal{S}}_{ij}^b \tag{2.44}$$

Roe numerical flux for the PB formulation:

$$\tilde{\mathcal{F}}_{ij} = \frac{\tilde{\mathbf{F}}(\breve{\mathsf{w}}_j;\, \breve{b}_j) + \tilde{\mathbf{F}}(\breve{\mathsf{w}}_i;\, \breve{b}_i)}{2} \cdot \boldsymbol{n}_{ij} - \upsilon_{ij}\frac{\breve{\mathsf{w}}_j + \breve{\mathsf{w}}_i}{2} - \frac{\left|\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3\right|}{2}(\breve{\mathsf{w}}_j - \breve{\mathsf{w}}_i) \tag{2.45}$$

and for later purposes, we define a *non upwind numerical source* term for the PB equations:

$$\tilde{\mathcal{S}}_{ij}^b = \tilde{\mathsf{A}}_{ij}^-\Delta\mathsf{b}_{ij}^- + \frac{1}{2}\tilde{\mathsf{A}}_{ij}\Delta\mathsf{b}_{ij} \tag{2.46}$$

with (cf. equation (1.5)) $\tilde{\mathsf{A}}_{ij} = \mathsf{A}(\boldsymbol{n}_{ij}, \eta_{ij})$ 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 $\eta^-_{ij} = (\breve{\eta}_i + \eta_i)/2$.

We observe the following equivalence:

Proposition 2. *The Pre-Balanced upwind FV discretization obtained from the pre-balanced form of the SWEs* (2.31) *with numerical fluxes* (2.45) *and a non-upwind source term approximation* (2.46) *is equivalent to a minor modification of the Well-Balanced scheme which is always given by* (2.37)-(2.38) *(with all the definitions contained) but setting in* (1.21) *the upwind source term as:*

$$\mathcal{S}^*_{ij} = -\frac{\mathrm{sign}(\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3)}{2}(\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3)\Delta b_{ij} \tag{2.47}$$

instead of (2.43).

Proof. We rewrite the well balanced FV spatial discretization incorporating the new definition of source upwinding: (2.47) reads:

$$\begin{aligned} R_{ij} =& \mathcal{F}_{ij} + \mathcal{S}^b_{ij} \\ =& \frac{\mathbf{F}(\breve{\mathsf{u}}_j) - \mathbf{F}(\breve{\mathsf{u}}_i)}{2}\cdot\boldsymbol{n}_{ij} + (\mathbf{F}(\breve{\mathsf{u}}_i) - \mathbf{F}(\mathsf{u}_i))\cdot\boldsymbol{n}_{ij} - \upsilon_{ij}\frac{\breve{\mathsf{w}}_j + \breve{\mathsf{w}}_i}{2} - \frac{|\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3|}{2}(\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i) \\ &+ \mathsf{A}^-_{ij}\,\Delta b^-_{ij} + \frac{1}{2}\mathsf{A}_{ij}\,\Delta b_{ij} + \mathcal{S}^*_{ij} \end{aligned} \tag{2.48}$$

We now use the fact that definition (2.47) of $\mathcal{S}^*_{ij}$ is such that when added to dissipation of the numerical flux one gets

$$-\frac{|\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3|}{2}(\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i) + \mathcal{S}^*_{ij} = -\frac{|\mathsf{K}_{ij} - \upsilon_{ij}\mathsf{I}_3|}{2}(\breve{\mathsf{w}}_j - \breve{\mathsf{w}}_i) \tag{2.49}$$

We exploit the continuous equivalence

$$\frac{\partial \mathsf{F}^j(\mathsf{u})}{\partial x^j} + \mathcal{S}^b = \frac{\partial \tilde{\mathsf{F}}^j(\mathsf{w}, b)}{\partial x^j} + \tilde{\mathcal{S}}^b$$

which is written here at discrete level:

$$(\mathbf{F}(\breve{\mathsf{u}}_j) - \mathbf{F}(\breve{\mathsf{u}}_i))\cdot\boldsymbol{n}_{ij} + \mathsf{A}_{ij}\Delta\mathsf{b}_{ij} = \tilde{\mathbf{F}}(\breve{\mathsf{w}}_j;\,\breve{b}_j) - \tilde{\mathbf{F}}(\breve{\mathsf{w}}_i;\,\breve{b}_i) + \tilde{\mathsf{A}}_{ij}\Delta\mathsf{b}_{ij} \tag{2.50}$$

$$(\mathbf{F}(\breve{\mathsf{u}}_i) - \mathbf{F}(\mathsf{u}_i))\cdot\boldsymbol{n}_{ij} + \mathsf{A}^-_{ij}\Delta\mathsf{b}^-_{ij} = \tilde{\mathbf{F}}(\breve{\mathsf{w}}_i;\,\breve{b}_i) - \tilde{\mathbf{F}}(\mathsf{w}_i;\,b_i) + \tilde{\mathsf{A}}^-_{ij}\Delta\mathsf{b}^-_{ij} \tag{2.51}$$

Finally, substituting (2.50),(2.51) and (2.49) in the well balanced FV spatial discretization (2.48) we have

$$R_{ij} = \tilde{\mathcal{F}}_{ij} + \tilde{\mathcal{S}}^b_{ij} \tag{2.52}$$

with $\tilde{\mathcal{F}}_{ij}$ defined as in (2.45) and $\tilde{\mathcal{S}}_{ij}$ as in (2.46) which is exactly the pre-balanced FV discretization otained from (2.31) (cf. Rogers *et al.* [2003]; Liang and Borthwick [2009]; Liang and Marche [2009]).

□

The last proposition shows that the well balanced discretization of Bermudez and Vazquez-Cendon [1994]; Hubbard and Garcia-Navarro [2000] 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

$$\mathcal{S}^*_{ij} = -\frac{|\mathsf{A}_{ij} - \upsilon_{ij}\mathsf{I}_3|}{2}\Delta b_{ij} \tag{2.53}$$

which also leads to a well-balanced discretization (cf. proof of proposition 1).

2.10.3 Mass conservation on moving mesh

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. (2.34). For the FV method, the nodal values are obtained via 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 arbitrary decrease the mass error.

2.11 ALE Residual Distribution for SWEs

While first ALE FV and FE methods dates back to many decades, RD scheme in ALE framework are quite recent. Michler *et al.* [2002] achieved first order of accuracy with an Explicit Euler time integrator and later Dobes and Deconinck [2008] moved to high order time approximation such as BDF3 and Crank Nicholson. In these references the common startegy to approximate the ALE term is to split it into two contributions

$$\frac{\partial(\sigma^j\mathsf{w})}{\partial x^j} = \sigma^j\frac{\partial\mathsf{w}}{\partial x^j} + \mathsf{w}\frac{\partial\sigma^j}{\partial x^j}$$

Michler *et al.* [2002] referred to this term as Geometric Source Term and discretize it in such a way to verify the DGCL. We will take advantage of the above form and write the fluctuation as:

$$\Phi^K(\mathsf{w}_h, b_h) = \int_{\partial K} \mathsf{F}^j(\mathsf{u}_h) n^j ds - \int_K \sigma^j \frac{\partial \mathsf{w}}{\partial x^j} d\boldsymbol{x} + \int_K \mathcal{S}(\mathsf{u}_h, \boldsymbol{x})\, d\boldsymbol{x} \tag{2.54}$$

As for FV, the fluctuation is evaluated at the midpoint configuration. Thus, if not specified, for us $K = \frac{1}{2}\left(K^n + K^{n+1}\right)$ in the following. Keeping in mind this, we provide directly the ALE extension of the eRK2-RD algorithm presented in section 1.6.4. We remark that it is in a form very close to the update (1.36)- (1.37). The only remarkable difference is that the variable w appears instead of u; this is a consequence of the results of section 2.12.1 which are applicable to any scheme and thus holds also for RD. In order to verify WB a numerical method should approximate the Well-Balanced form of the SWEs (2.28):

1] Predictor step: for each element $K \in \mathcal{T}_h$

- Compute the residual $\Phi^{K(1)} = \Phi^K(\mathsf{w}_h^n, b_h^n)$.
- Distribute the fluctuation to the nodes of K such that $\sum_{j\in K} \Phi_j^{K(1)} = \Phi^{K(1)}$
- Compute the first order prediction of the solution, denoted as w^*

$$\mathsf{w}_i^* = \mathsf{w}_i^n - \frac{\Delta t}{|C_i^{n+1}|} \sum_{K\in\mathcal{D}_i} \Phi_i^{K(1)}(\mathsf{w}_h^n, b_h^n) \tag{2.55}$$

2] Corrector step: for each element $K \in \mathcal{T}_h$

- Compute the residual

$$\Phi^{K(2)} = \frac{1}{2}\left(\Phi^K(\mathsf{w}_h^n, b_h^n) + \Phi^K(\mathsf{w}_h^*, b_h^{n+1})\right) + \int_K \frac{\mathsf{w}_h^* - \mathsf{w}_h^n}{\Delta t} d\boldsymbol{x}$$

- Distribute the fluctuation to the nodes of K such that $\sum_{j\in K} \Phi_j^{K(2)} = \Phi^{K(2)}$
- Compute the second order correction from

$$\mathsf{w}_i^{n+1} = \mathsf{w}_i^* - \frac{\Delta t}{|C_i^{n+1}|} \sum_{K\in\mathcal{D}_i} \Phi_i^{K(2)}(\mathsf{w}_h^*, \mathsf{w}_h^n, b_h^n, b_h^{n+1}) \tag{2.56}$$

In Appendix D we provide the details that brought us to the above form which essentially comes from an analogy with stabilized Finite Element method. With respect to the fixed mesh algorithm, we are left with a new term into the fluctuation (4.7) related to the ALE

$$\Phi^{K,ALE} = \int_K \sigma^j \frac{\partial \mathsf{w}}{\partial x^j} d\boldsymbol{x}$$

The problem is that we have the elements' edge velocities rather then nodal values, see section 2.5.1. As remarked by Etienne *et al.* [2009] this is a common problem for Finite Element methods since the DGCL provides a rule to compute the divergence of the mesh velocity but not the velocity itself. Still we can compute the above term as:

$$\begin{aligned}\Phi^{K,ALE} &= \int_K \left(\frac{\partial \sigma^j \mathsf{w}}{\partial x^j} - \mathsf{w} \frac{\partial \sigma^j}{\partial x^j} \right) d\boldsymbol{x} \\ &= \int_{\partial K} \mathsf{w} \sigma^j n^j ds - \int_K \mathsf{w} \frac{\partial \sigma^j}{\partial x^j} d\boldsymbol{x}\end{aligned}$$

We examine the first part. Edge fluxes are computed by quadrature formula. At the same time, these fluxes should verify the DGCL. Following the closure proposed in 2.5, we replace in (2.23) and (2.24), $V = K$ and $\partial V_j = \partial K_j$ and the DGCL becomes:

$$|K^{n+1}| - |K^n| = \Delta t \sum_{j \in K} \upsilon_j \tag{2.57}$$

The velocity of the j-th edge (opposed to node j) comes from (2.25)

$$\upsilon_j = \int_{\partial K_j} \sigma^m n^m \, ds = \frac{1}{\Delta t} \int_{Q_j} d\boldsymbol{x} \tag{2.58}$$

with the quadrilateral Q_j defined by vertexes $\boldsymbol{x}_i^n$,$\boldsymbol{x}_k^n$,$\boldsymbol{x}_k^{n+1}$,$\boldsymbol{x}_i^{n+1}$, see figure 2.6. Once υ_j are recovered, we share these velocities on the edge's quadrature points through Gaussian weights $q = 1, N_Q$

$$\int_{\partial K} \mathsf{w} \sigma^j n^j ds = \sum_{j \in K} \sum_{q}^{N_Q} \omega_q \mathsf{w}_q \upsilon_j \tag{2.59}$$

The second term is computed using the constant divergence statement (4.43)

$$\begin{aligned}\int_K \mathsf{w} \frac{\partial \sigma^j}{\partial x^j} d\boldsymbol{x} &= \int_K \mathsf{w} \, d\boldsymbol{x} \left(\frac{\partial \sigma^j}{\partial x^j} \right)_K \\ &= \sum_{j \in K} \int \varphi_j \, d\boldsymbol{x} \, \mathsf{w}_j \left(\frac{\partial \sigma^j}{\partial x^j} \right)_K \\ &= \frac{\sum_{j \in K} \mathsf{w}_j}{3} \frac{\Delta |K|}{\Delta t}\end{aligned} \tag{2.60}$$

where a weighted average solution on the cell appears.

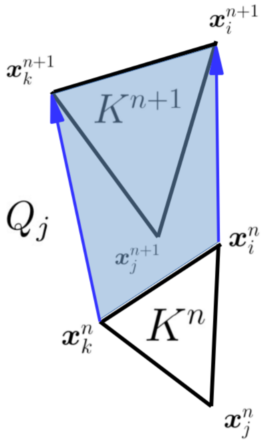

Figure 2.6: DGCL for ALE-RD. Area swept by the edge ∂K_j.

2.11.1 Well-Balanced on moving mesh

We give directly the following statement, which can be regarded as the extension of the Well-Balanced property for RD to moving meshes:

Proposition 3. *The explicit predictor corrector residual distribution prototype* (2.55), (2.56) *with edge velocities* (2.58) *verifies the DGCL for constant* b*, and the Well-Balanced property both on moving and fixed meshes, provided that the same linear piecewise continuous approximation is used for* w*,* b *and consequently for* u*, and that all integrals involving these quantities are evaluated exactly w.r.t. this variation.*

Proof. To prove the first part (DGCL), we check that for constant bathymetry and no friction, the splitting terms on the right-hand sides in (2.55) and (2.56) are identically zero for a given constant state u_0. We do the proof only for corrector, the predictor is a particular case. We use the hypothesis that quadrature formula for line integration are exact for P1 function, then the obvious $\int_{\partial K} n^j ds = 0$ and the property of Gaussian weights $\sum_q^{NQ} \omega_q = 1$:

$$\begin{aligned}
\Phi^{K(2)} &= \int_K \frac{\mathsf{u}_0 - \mathsf{u}_0}{\Delta t} d\boldsymbol{x} + \mathsf{F}_0^j \int_{\partial K} n^j \, ds - \mathsf{u}_0 \sum_{j \in K} \sum_q^{NQ} \omega_q \upsilon_j + \mathsf{u}_0 \frac{\Delta |K|}{\Delta t} \\
&= \mathsf{u}_0 \left(\frac{\Delta |K|}{\Delta t} - \sum_{j \in K} \upsilon_j \right) = 0
\end{aligned}$$

This is immediately shown if we compute the edge velocities with (2.58). We have $\mathsf{u}_i^{n+1} = \mathsf{u}_0$.

To prove the second part of the proposition (Well-Balancedness), we proceed in an identical manner, except that now we assume that we have a constant state $\mathsf{w} = \mathsf{w}_0$. As before we consider only the corrector

$$\begin{aligned}
\Phi^{K(2)} &= \int_K \frac{\mathsf{w}_0 - \mathsf{w}_0}{\Delta t} d\boldsymbol{x} + \int_{\partial K} \mathsf{F}^j n^j \, ds + \int_{\partial K} \mathcal{S}^b \, d\boldsymbol{x} - \mathsf{w}_0 \sum_{j \in K} \sum_{q}^{NQ} \omega_q \upsilon_j + \mathsf{w}_0 \frac{\Delta |K|}{\Delta t} \\
&= \mathsf{w}_0 \left(\frac{\Delta |K|}{\Delta t} - \sum_{j \in K} \upsilon_j \right) = 0
\end{aligned}$$

□

2.11.2 Mass conservation on moving mesh

In section 2.7 we have discussed the fact that mass conservation on moving meshes is related on the accuracy of the bathymetric values (2.34). For RD 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 piecewise 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 $r = 2, 3$).

2.11.3 Distributions

The distributions matrix can be extended directly from chapter 1. The SWEs Jacobian changed, see (4.8), and this must be taken into the distribution operator, especially for the SUPG matrix contained in (1.35), and the Lax Friederich parameter in (1.34).

2.11.4 Water depth positivity

The ALE-RD algorithm (2.55)(2.56) is in a form that closely resemble the Eulerian-fixed mesh (1.36)(1.37). The result of water depth positivity preserving can be extended using the same arguments of the proof contained in Ricchiuto [2015]. There are however two issues that must be taken into account. First, the fluctuation is enhanced by the ALE term. Second, the mass equation is written in the variable η and not h. The predictor update for the mass equation from (2.55), followed by the WB correction (2.34) rewrites:

$$\eta_i^* = \eta_i^n - \frac{\Delta t}{|C_i^{n+1}|} \sum_{K \in \mathcal{D}_i} \phi_i^K \tag{2.61}$$

$$h_i^* = \max(0, \eta_i^* - b_i^{n+1} + \Delta_{b_i}) \tag{2.62}$$

where Δ_{b_i} is defined in (2.35) and it is different from zero only if i is dry. As a consequence of the max operator, introduced to avoid negative depths in the lake at rest, such an update is trivially positive. However let's try to understand which is exactly the role of the max operator for a general situation where the flow is not at rest. We rewrite the update (2.61) in h and we separate the ALE fluctuation in h and b. Always for the predictor step we have:

$$\hat{h}_i^* = \underbrace{h_i^n - \frac{\Delta t}{|C_i^{n+1}|} \sum_{K \in \mathcal{D}_i} \phi_i^h}_{I} \underbrace{-b_i^{n+1} + b_i^n - \frac{\Delta t}{|C_i^{n+1}|} \sum_{K \in \mathcal{D}_i} \phi_i^b}_{II \neq 0} \tag{2.63}$$

$$h_i^* = \max(0, \hat{h}_i^* + \Delta_{b_i}) \tag{2.64}$$

where in every cell K, ϕ_i^h is a splitting of the residual in h

$$\phi^h = \int_{\partial K} h u^j n^j ds - \int_K \sigma^j \frac{\partial h}{\partial x^j} d\boldsymbol{x}$$

as we see it is enhanced by the ALE part. Similarly, in every cell K, ϕ_i^b defines a splitting of the ALE residual in b:

$$\phi^b = -\int_K \sigma^j \frac{\partial b}{\partial x^j} d\boldsymbol{x}$$

In 2.63, two parts can be put in evidence (within brackets). The last three terms underlined, named with II, should correspond to the approximation of the ALE remap equation for the bathymetry. They do not sum up to zero because we have deliberately choose to not compute b_i^{n+1} with the ALE remap equation which means by:

$$b_i^{ALE} = b_i^n - \frac{\Delta t}{|C_i^{n+1}|} \sum_{K \in \mathcal{D}_i} \phi_i^b$$

We can define II as:

$$II = b_i^{ALE} - b_i^{n+1} \neq 0$$

The first two terms on the right hand side of (2.63), named as I, have not been addressed yet. Basically they represent the update as if we were evolving the ALE mass equation in h, that is (2.16). With respect to solving the SWEs in the water depth, the term II can be seen as a perturbation related to the fact that we are not solving an equation for the bathymetry remap. Unfortunately, we do not control the sign of this perturbation. Very roughly, in (2.62)

$$h_i^* = \max(0,\ I + II + \Delta_{b_i})$$

if $II + \Delta_{b_i}$ generates a negative water depth, the max operator nullifies this contribution automatically.

However, it is important to ensure that I, the part of the mass equation update in h, is still positivity preserving. This provides our ALE scheme of a CFL condition that avoids negative depths in case of flat bathymetry and ensures that negative contributions, which are roughly nullified in (2.62), are only due to a non-exact ALE remap of the bathymetry. The fluctuation ϕ^h writes:

$$\begin{aligned}\phi^h &= \frac{1}{2}\sum_{j\in K} h_j \boldsymbol{u}_j\cdot\boldsymbol{n}_j - \sum_{j\in K} h_j\left(\frac{\upsilon_i+\upsilon_k}{2}-\frac{1}{3}\frac{\Delta|K|}{\Delta t}\right)\\ &= \frac{1}{2}\sum_{j\in K} h_j \boldsymbol{u}_j\cdot\boldsymbol{n}_j - \frac{1}{3}\sum_{j\in K} h_j\left(\frac{\upsilon_i+\upsilon_k}{2}-\upsilon_j\right)\end{aligned}$$

in last development we have used the DGCL (2.57). With respect to the fixed mesh case of section 1.6.5 the ALE fluctuation is enhanced by the ALE term which have been computed according to (2.59) and (2.60). We set for simplicity:

$$\frac{\upsilon_i+\upsilon_k}{2}-\upsilon_j=\upsilon_{ik}-\upsilon_j$$

As done in section 1.6.5 we write the LLxF fluctuation as:

$$\begin{aligned}\phi_i^{LLxF} &= \gamma_i\left(\frac{1}{6}\sum_{j\in K} h_j\boldsymbol{u}_j\cdot\boldsymbol{n}_j-\frac{1}{9}\sum_{j\in K} h_j\left(\upsilon_{ik}-\upsilon_j\right)+\frac{\alpha^K}{3}\sum_{j\in K,\neq i}(h_i-h_j)\right)\\ &= \gamma_i\left(\frac{1}{6}\boldsymbol{u}_i\cdot\boldsymbol{n}_i-\frac{1}{9}\left(\upsilon_{jk}-\upsilon_i\right)+\frac{2\alpha^K}{3}\right)h_i\\ &+ \sum_{j\in K,\neq i}\gamma_i\left(\frac{1}{6}\boldsymbol{u}_j\cdot\boldsymbol{n}_j-\frac{1}{9}\left(\upsilon_{ik}-\upsilon_j\right)-\frac{\alpha^K}{3}\right)h_j\end{aligned}$$

for the definition of $\gamma_i > 0$ see always 1.6.5. We remark that we have put the LLxF fluctuation into the form (1.32) which is particularly suited to find local bounds of the numerical solution. Considering the contribution of each element to the update (2.63) separately:

$$\begin{aligned}\hat{h}_i^* &= h_i^n-\frac{\Delta t}{|C_i^{n+1}|}\phi_i^{LLxF}\\ &= \left(1-\frac{\Delta t\gamma_i}{|C_i^{n+1}|}\left(\frac{\boldsymbol{u}_i\cdot\boldsymbol{n}_i}{6}-\frac{\upsilon_{jk}-\upsilon_i}{9}+\frac{2\alpha^K}{3}\right)\right)h_i^n\\ &+ \sum_{j\in K,j\neq i}\frac{\Delta t\gamma_i}{|C_i^{n+1}|}\left(\frac{\alpha^K}{3}-\frac{\boldsymbol{u}_j\cdot\boldsymbol{n}_j}{6}+\frac{\upsilon_{ik}-\upsilon_j}{9}\right)h_j^n\\ &= a_{ii}^K h_i^n+\sum_{j\in K,j\neq i} a_{ij}^K h_j^n\end{aligned}$$

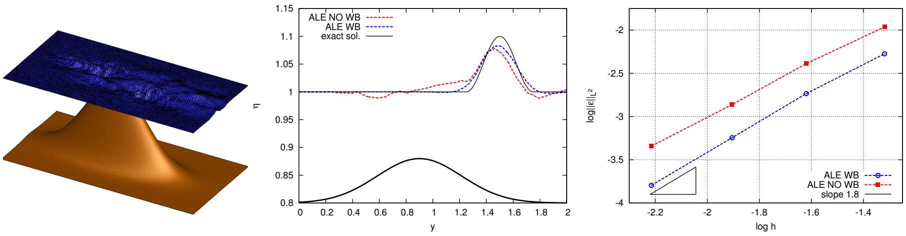

Figure 2.7: Linear Advection. Left: Lake at rest for the NO WB ALE formulation and failing in verifying Well Balanced. Middle: comparison between the numerical solution and exact one on the symmetry line $x = 0.5$. Right: convergence order for the L^2-norm of the error.

$h_i^* > 0$ is positive if the extra-diagonal coefficients $a_{ij}^K > 0$, which fix a lower bound to the LxF dissipation parameter:

$$\alpha^K > \frac{1}{2}\boldsymbol{u}_j \cdot \boldsymbol{n}_j - \frac{\upsilon_{ik} - \upsilon_j}{3}, \quad \forall j \in K, \neq i$$

We have neglected $\frac{\Delta t \gamma_i}{|C_i|}$ because it is always positive. Moreover it is necessary to enforce a CFL condition to have the positivity of the diagonal coefficient $a_{ii} > 0$

$$\Delta t < \frac{3|C_i^{n+1}|}{\sum_{K \in \mathcal{D}_i} \left(\frac{1}{2}\boldsymbol{u}_i \cdot \boldsymbol{n}_i - \frac{\upsilon_{ik} - \upsilon_j}{3} + 2\alpha^K \right)}$$

For the corrector step, the positivity analysis can be carried out exactly in the same fashion of Ricchiuto [2015] to which we refer for details (adding of course the ALE fluctuations for the predictor and corrector step). In practical calculation we have imposed:

$$\alpha^K = \frac{1}{2}\left(\max_{j \in K} \|\boldsymbol{u}_j\| \max_{j \in K} \|\boldsymbol{n}_j\| + 2 \max_{j \in K} \upsilon_j \right), \qquad \Delta t < \min_{i \in \mathcal{T}_h} \frac{|C_i^{n+1}|}{\sum_{K \in \mathcal{D}_i} \alpha^K}$$

2.12 Numerical experiments

2.12.1 Well Balanced

To illustrate some concepts and to better highlight certain numerical effects, we reconsider the simplified model (2.65). We add a general source to mimick the SWEs. This model reads

$$\frac{\partial u}{\partial t} + \frac{\partial F^j}{\partial x^j} + S(u, \boldsymbol{x}) = 0\,, \qquad \boldsymbol{x} \in [0,2] \times [0,1],\ t \in [0,1] \tag{2.65}$$

where, for a given flux vector $\boldsymbol{F}(u) = [\, F^1 \; F^2 \,]$, the source term is defined as

$$S = a^j \frac{\partial b}{\partial x^j}$$

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, $\boldsymbol{F}(u) = \boldsymbol{a}u$, with $\boldsymbol{a} = [\,1\,0\,]$. Introducing the variable $\eta = u + b$, equation (2.65) admits a non-trivial steady state given by

$$u(x, y) + b(x, y) = \eta_0 = const \tag{2.66}$$

To test the WB property on moving mesh for the ALE formulation, we use the simple case of linear advection of a smooth sinusoidal hill

$$\begin{cases} \eta_0(\boldsymbol{x}) = 1 + \cos^2(2\pi r) & \text{if } r \le 0.25,\; r = \sqrt{(x-0.5)^2 + (y-0.5)^2} \\ \eta_0(\boldsymbol{x}) = 1 & \text{otherwise} \end{cases}$$

A 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

$$x^i(t) \quad = X^i + 0.1 \sin(2\pi X^1) \sin(\pi X^2) \sin(2\pi t) \qquad i = 1, 2$$

We check the validity of the analysis of section on this scalar 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 scalar scheme, but the FV results are almost identical. The results are summarized in figure 2.7. We can confirm that: when no perturbation is added, the well balanced ALE formulation (2.28) (ALE WB in the figures) preserves the constant state to machine accuracy (not shown in the figures), while the classical ALE form (2.19) (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.

2.12.2 Accuracy

We consider the advection of a vortex problem proposed in Ricchiuto and Bollerman [2009] to test the accuracy of the SWEs-ALE schemes. The spatial domain is the square $[0, 1]$. The vortex is initially centered in $(0.5, 0.5)$ and is transported by a constant field $\boldsymbol{u} = [\,6\;\,0]$ until it has crossed the whole domain and get back to its initial position at $T = 1/6$. Periodic boundary conditions are imposed. The computations have been performed on 5 unstructured grids with the topology shown on figure 3. The coarsest has mesh size $h = 1/56$. The other 4 meshes have been generated independently, halving the mesh size at each step. The following arbitrary mapping is used to move the mesh

$$\begin{aligned} x^1(t) &\quad = \chi^1 + 0.1 \sin(2\pi\chi^1) \cos(2\pi\chi^2) \sin\left(\tfrac{2\pi t}{T}\right) \\ x^2(t) &\quad = \chi^2 + 0.1 \cos(2\pi\chi^1) \sin(2\pi\chi^2) \sin\left(\tfrac{2\pi t}{T}\right) \end{aligned}$$

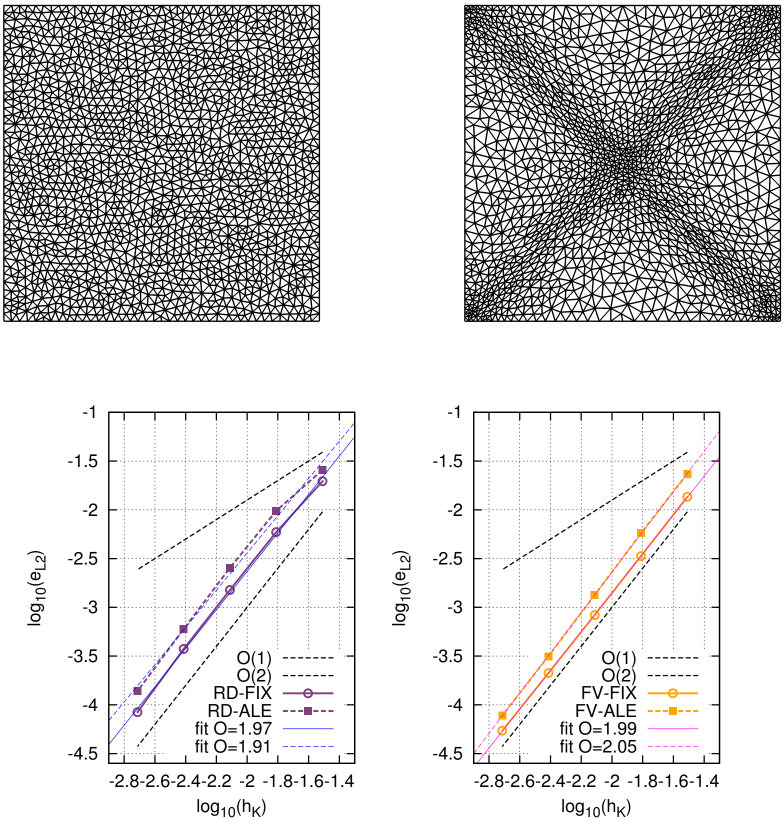

Figure 2.8: Vortex advection. 2nd level of mesh refinement: original mesh and transformed mesh. Mesh convergence for: left) RD. right) FV.

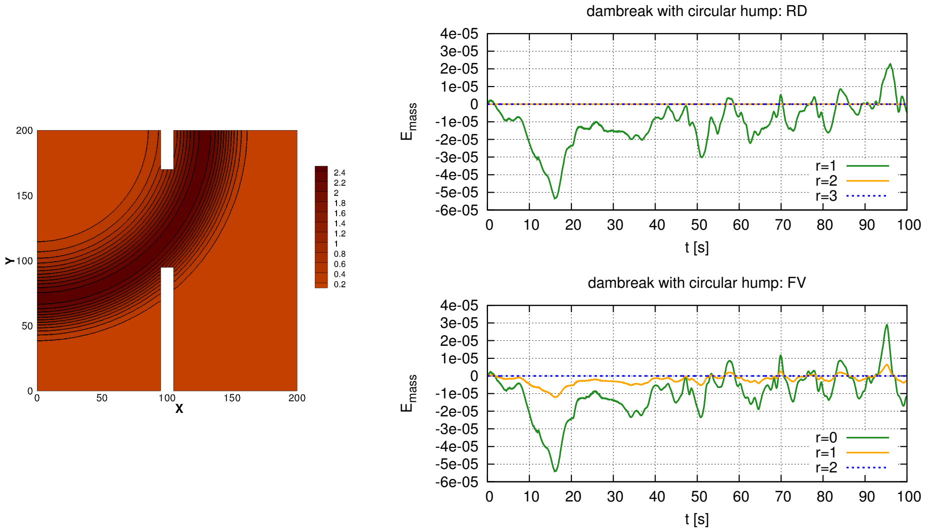

Figure 2.9: Dambreak with circular hump. Left: bathymetry. Right: dimensionless mass error for different quadrature formula of the bathymetry integral.

2.12.3 Mass conservation

We perform the classical test benchmark, taken from Seaid [2004]. 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 $5\,m$. To check our mass conservation correction (cf. section 2.7) we have added a bathymetry shaped as a circular hump centered in $(x, y) = (0, 200)\,[m]$, and defined by an exponential law in the radial direction (cf. left picture on figure 2.9). We report on the right pictures on figure 2.9 the mass error measured without any correction, and with corrections based on different quadrature formulas (for the definition of E_{mass}, see always section 2.7. We can clearly see that we are able to preserve the total mass in the domain practically up to machine accuracy.

Chapter 3

r-adaptation for hydrodynamics

In this chapter, to enhance important features in hydrodynamics simulations such as the resolution of wave patterns or the wetting/drying dynamics, we employ mesh adaptation. We point out the obvious fact that mesh adaptation introduces an additional level of complexity and must be used with great care to be effective. To improve the accuracy of the second order RD/FV-eRK2 Shallow Water solver presented in chapter 1 and 2, we employ mesh adaptation techniques based on nodes redistribution (or relocation) that are also known as *r-adaptation techniques*. Roughly speaking, these methods move the points of a given reference mesh, keeping the mesh topology and number of mesh points unchanged. In fact in this simple description, it is hidden the main advantage of r-refinement with respect to h-refinement: the data structure is unchanged and efficient conservative/accurate remaps of flow variables can be carried out from one grid to the updated one.

In the first part of the chapter we introduce the reader to r-adaptation (a vast review can be found in Budd *et al.* [2009a]). In this framework we detail the r-adaptive technique of Ceniceros and Hou [2001] which is implemented in this thesis. A small variant in the definition of the error estimate is considered to refine the mesh at the wet-dry interface. Once we have understood the tool to move the mesh, in the second part of the chapter we deal with the problem of resolving the SWEs (or general balance laws) on these moving grids. A popular method is the rezoning algorithm of Tang and Tang [2003] which is based on conservative ALE remaps. ALE remap is presented in 3.5 and permits to interpolate the numerical solution among the different grids. We also recall that an elegant way to resolve the SWEs in an arbitrary reference framework $\boldsymbol{x} = A(\boldsymbol{\chi}, t)$ has been presented in chapter 2. If we choose this arbitrary reference to coincide with the grid transformation, then the solution on the grid $\boldsymbol{x}^n$, $\mathsf{u}(\boldsymbol{x}^n, t^n)$, is evolved directly on the new grid $\boldsymbol{x}^{n+1}$, $\mathsf{u}(\boldsymbol{x}^{n+1}, t^{n+1})$. Rezoning and ALE algorithms (section 3.6) are tested on scalar problems in section 3.6.4. From this first investigation we propose a third coupling algorithm which is a cheaper variant of the rezoning approach. Finally, in section 3.7 we present the coupled algorithms in terms of accuracy, and CPU time for some classical SWEs benchmarks.

3.1 Uniform mesh in metric space

The general idea in mesh generation is to exploit the fact that we can easily generate a uniform mesh in a Riemannian space, which will correspond to a non uniform *physical* mesh in the Euclidean space, which is used for the computations. With respect to chapter 2, we switch from the term "actual" to the term "physical" in order to highlight that this is the configuration in which our physical conservation law is settle. The physical space is opposed to the computational space, which exists only to compute the mesh. At the end, points in the physical mesh must be clustered where large gradients of the solution appears. With a brief example taken from Alauzet [2010], we would like to explain why a Riemannian metric space makes possible to obtain non uniform mesh in the physical space. A manifold $\mathcal{M}$ can be seen as a surface embedded in the Euclidean space $\mathbf{E}^3$ and described with Cartesian coordinates:

$$\mathcal{M} = \left(x^1, x^2, x^3(x^1, x^2)\right) \tag{3.1}$$

with x^1, x^1 belonging to $\Omega \subseteq \mathbb{R}^2$ an open region in the Euclidean plane . First we consider a pair of auxiliary variable to parametrize the surface $\chi^1, \chi^2 \in \Omega_\chi \subseteq \mathbb{R}^2$ such that

$$\mathcal{M} = \left(x^1(\chi^1, \chi^2), x^2(\chi^1, \chi^2), x^3(\chi^1, \chi^2)\right)$$

We consider a straight curve $\boldsymbol{\gamma} \in \Omega$ parameterized by the parameter $t \in [0, 1]$ which connects two points P, Q with $P(x^1, x^2)$ and $Q(x^1, x^2)$:

$$\boldsymbol{\gamma}(t) = (x^1(t), x^2(t))$$

Its image $\boldsymbol{c}(t)$ on the surface $\mathcal{M}$ is a curve of $\mathbb{R}^3$ connecting two points lying on $\mathcal{M}$, R and S. Also the parametric coordinates can be assumed to be function of the parameter t. The curve $\boldsymbol{c}(t)$ simply is:

$$\boldsymbol{c}(t) = (x^1(\chi^1(t), \chi^2(t)), x^2(\chi^1(t), \chi^2(t)), x^3(\chi^1(t), \chi^2(t))).$$

The length of the curve $\boldsymbol{c}(t)$ on $\mathcal{M}$, is expressed by (A.14):

$$l_{\mathcal{M}}(P, Q) = \int_0^1 \sqrt{\frac{d\chi^i}{dt} M_{ij}(t) \frac{d\chi^j}{dt}}\, dt$$

We assume now that we can construct a mesh such that for every edge PQ, its arc length is constant, let's say unitary $l_{\mathcal{M}}(P, Q) = 1$. This mesh is said to be uniform in the metric space. But its image on Ω corresponds to straight

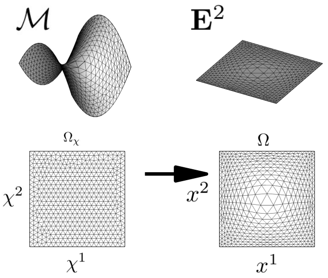

Figure 3.1: An example taken from Alauzet [2010]: a uniform mesh in a metric space and a non uniform mesh in Euclidean space.

edges with non unitary distance. If now we think to the plane Ω as a plane with Euclidean metric, we have constructed a non uniform two dimensional mesh. This illustrates why a Riemannian metric curves the space: a uniform computational mesh corresponds to a physical mesh where points cluster if curvature appears.

3.2 Elliptic Moving Mesh PDE

The theory of elliptic grid generation provides a suitable theoretical framework for developing the set of PDEs that controls mesh movement, or moving mesh PDEs (MMPDEs). For clarity we repeat here the main results contained in Thompson *et al.* [1999], and we provide the expression of the MMPDE that we chose among many, as a particular case of this general theory. Later we will see that, at least for the general case, there is an analogy with the theory of harmonic map, Ivanenko [1999].

We introduce two (unstructured), simply connected grids: the computational and the physical mesh. We recall from chapter 2 the ALE transformation which transforms the reference mesh into the actual one, see (2.6) and (2.7):

$$A : \mathcal{A}_0 \to \mathcal{A} \quad \boldsymbol{x} = A(\boldsymbol{\chi}, t), \quad \boldsymbol{J}_A = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\chi}}$$

the map should be invertible and we assume $J_A = \det \boldsymbol{J}_A > 0$ to avoid negative areas. Differently from what stated in chapter 2, we will assume here that points $\hat{Q} \in \mathcal{A}_0$ and $Q \in \mathcal{A}$ do not lie in the Euclidean plane but that they both $\mathcal{A}$ and $\mathcal{A}_0$ could belongs to a manifold in $\mathbf{E}^3$. As stated in Warsi [1999] and Spekreijse [1999], in the context of mesh adaptation/generation we would like to map a region of manifold called $\mathcal{M}$ with local curvilinear coordinates $\boldsymbol{\chi}$ into a region of a two dimensional Euclidean plane $\mathbf{E}^2$ described by Cartesian coordinates $\boldsymbol{x}$. With respect to (2.6) we consider $\mathcal{A}_0 \subset \mathcal{M}$ and $\mathcal{A} \subset \mathbf{E}^2$. We put this in evidence in the definition of the ALE map:

$$A : \mathcal{M} \to \mathbf{E}^2 \quad \boldsymbol{x} = A(\boldsymbol{\chi}, t) \tag{3.2}$$

and we specify the parametric domain to which curvilinear coordinates belongs:

$$\boldsymbol{\chi} \in \Omega_\chi \subset \mathbb{R}^2 \quad \text{and} \quad \boldsymbol{x} \in \Omega \subset \mathbb{R}^2$$

see figure 3.1. As for the rest, we refer to them respectively as *computational and physical domain.*

In the following, we will extensively make use of the concept of curvilinear coordinates. We give in appendix A a brief review of tensor analysis and a list of the formulas for divergence, gradient and Laplacian operator in generalized coordinates. Concerning the notation we use the one of appendix A but with different letters. We assume that $\mathcal{M}$ is equipped with metric tensor $\boldsymbol{M}$ of components M_{ij} ($\sqrt{M} = \sqrt{\det \boldsymbol{M}}$), inverse metric tensor $\boldsymbol{M}^{-1}$ of components M^{ij} and Christoffel symbols Γ^k_{ij}.

We start by considering the Laplacian of a scalar ϕ in the curvilinear coordinates $\boldsymbol{\chi}$, see (A.13)

$$\nabla^2 \phi = M^{ij} \left(\frac{\partial^2 \phi}{\partial \chi^i \partial \chi^j} - \Gamma^k_{ij} \frac{\partial \phi}{\partial \chi^k} \right) \tag{3.3}$$

We write the Laplacian of each physical component, separately. Setting in the above equation $\phi = x^\alpha$, we realize that

$$\nabla^2 x^\alpha = \boldsymbol{e}^i \frac{\partial x^\alpha}{\partial x^i} \cdot \boldsymbol{e}^j \frac{\partial x^\alpha}{\partial x^j} = \frac{\partial^2 x^\alpha}{\partial x^j \partial x^j} = 0$$

and we have an equation for the physical coordinates

$$M^{ij} \left(\frac{\partial^2 x^\alpha}{\partial \chi^i \partial \chi^j} - \Gamma^k_{ij} \frac{\partial x^\alpha}{\partial \chi^k} \right) = 0 \tag{3.4}$$

This equation can be further manipulated setting always in (3.3), $\phi = \chi^k$:

$$\begin{aligned} \nabla^2 \chi^k &= -M^{ij} \Gamma^k_{ij} \frac{\partial \chi^k}{\partial \chi^k} \\ &= -M^{ij} \Gamma^k_{ij} \end{aligned}$$

And we can write compactly a MMPDE for each physical coordinate

$$M^{ij}\frac{\partial^2 x^\alpha}{\partial\chi^i\partial\chi^j}+\nabla^2\chi^k\frac{\partial x^\alpha}{\partial\chi^k}=0$$

This is a set of uncoupled elliptic PDEs which gives back directly the grid in the physical space. This system can be solved numerically.

What is done typically at this point is to generate grids in the computational domain imposing Poisson equations $\nabla^2\chi^k = P^k$ with *control function* P^k that can be assigned arbitrary. For general $P^k \neq -M^{ij}\Gamma^k_{ij}$, this is a generalization of (3.4) and it could still represent a moving mesh generator, although the transformation from a metric space has no more sense and the map $\boldsymbol{x} = A(\boldsymbol{\chi}, t)$ is meant $A : \mathbf{E}^2 \to \mathbf{E}^2$. We rewrite the MMPDE with control function:

$$M^{ij}\frac{\partial^2 x^\alpha}{\partial\chi^i\partial\chi^j}+P^k\frac{\partial x^\alpha}{\partial\chi^k}=0 \tag{3.5}$$

3.3 Moving Mesh PDE from Harmonic Maps

Dvinsky [1991] noted that elliptic meshes discussed in the previous paragraph could be generated through the theory of harmonic maps, as formulated by Eell and Sampson [1964]. We refer to appendix B for a brief digression on the definition of harmonic map and the main statements. Keeping in mind that the map is $A : \mathcal{M} \to \mathbf{E}^2$, we gain that one metric is Euclidean $H_{\alpha\beta} = \delta_{\alpha\beta}$ (with $\delta_{\alpha\beta}$ the Kronecker's delta) and, as a consequence, the energy density for the map (3.2) writes, see B.1:

$$e = M^{ij}\delta_{\alpha\beta}\frac{\partial x^\alpha}{\partial\chi^i}\frac{\partial x^\beta}{\partial\chi^j} = M^{ij}\frac{\partial x^\alpha}{\partial\chi^i}\frac{\partial x^\alpha}{\partial\chi^j}$$

thus the energy functional writes

$$E(\boldsymbol{x}) = \frac{1}{2}\int M^{ij}\frac{\partial x^\alpha}{\partial\chi^i}\frac{\partial x^\alpha}{\partial\chi^j}\sqrt{M}d\boldsymbol{\chi} \tag{3.6}$$

which indeed admits the Euler-Lagrange equations, see (B.2)

$$M^{ij}\left(\frac{\partial^2 x^\alpha}{\partial\chi^i\partial\chi^j}-\Gamma^k_{ij}\frac{\partial x^\alpha}{\partial\chi^k}\right)=0$$

so we have found again (3.4). The advantage of the approach of Dvinsky consists in the fact that it comes together with a prove of existence and uniqueness of the map. The Hamilton-Yau-Shoen theorem (theorem HYS, Hamilton [1975], Schoen and Yau [1978]) states that the map exists when the following two conditions are verified:

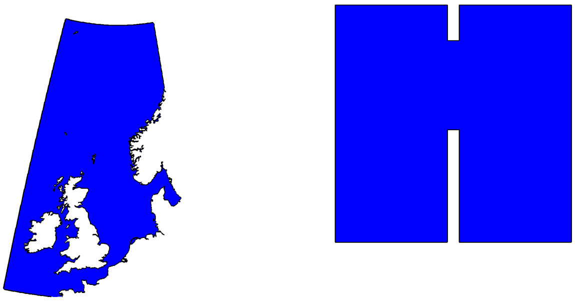

Figure 3.2: Examples of non convex physical domain in hydrodynamic simulation.

1. The curvature of the physical configuration $\mathcal{A}$ is non-positive, and
2. The boundary of the physical domain $\partial\Omega$ is convex.

The first condition is identically satisfied by the physical configuration because $\mathcal{A} \subset \mathbf{E}^2$. In addition, if the boundary of the physical domain is convex, then the maps exists. We point out that, in hydrodynamics simulations this last hypothesis seems to be a little too restrictive. Let's think for example to the physical domains in figure 3.2. For such cases we have no guarantees that a map into the logical domain exists. This was a reason why, at the beginning, much effort was spent in resolving the inverse map, that is

$$A^{-1} : \mathcal{M} \to \mathbf{E}^2 \qquad \boldsymbol{\chi} = A^{-1}(\boldsymbol{x}, t)$$

Following Dvinsky [1991] and later Huang and Russell [1999], one can choose $\mathcal{A}_0$ to have an Euclidean metric and $\mathcal{A}$ to be a region of a Riemannian manifold $\mathcal{M}$. $\mathcal{A}_0$ is flat with zero curvature and the convexity hypothesis is no more urgent, since for the computational domain we can always take a convex domain. Moreover, the HYS theorem guarantees the existence and the invertibility of the map for a general metric. In light of this and for completeness with respect to this small review on moving mesh methods, we report the (inverse) MMPDE of Huang and Russell [1999] (given the complete Euler-Lagrange equations (B.3) you may set $\Gamma^{\alpha}_{\gamma\delta} = 0$ and invert x^α with χ^α)

$$\frac{\partial}{\partial x^i}\left(M^{ij}\sqrt{M}\frac{\partial \chi^\alpha}{\partial x^j}\right) = 0$$

which works also for non convex physical domain. Of course the drawback of taking the mapping in the opposite direction is that we need to transform it, somehow, back into the curvilinear coordinates $\boldsymbol{x}$ and this makes the resulting equation much more complex.

3.4 MMPDE of Ceniceros and Hou

In the following we will stick to the direct map $\boldsymbol{x} = A(\boldsymbol{\chi}, t)$ and to the general MMPDE (3.5) that the interested reader will find in Warsi [1999]. Consider the following diagonal metric

$$M_{ij} = \frac{1}{\omega}\delta_{ij}$$

with $\sqrt{M} = \omega^{-1}$. The function ω is called *monitor function*. The inverse metric is $M^{ij} = \omega\delta^{ij}$. We choose now the control function

$$P^k = \frac{\partial \omega}{\partial \chi^k}$$

and substitute in the MMPDE (3.5)

$$\begin{aligned} \delta^{ij}\omega\frac{\partial^2 x^\alpha}{\partial \chi^i \partial \chi^j} + \frac{\partial \omega}{\partial \chi^k}\frac{\partial x^\alpha}{\partial \chi^k} &= 0 \\ \omega\frac{\partial^2 x^\alpha}{\partial \chi^i \partial \chi^i} + \frac{\partial \omega}{\partial \chi^k}\frac{\partial x^\alpha}{\partial \chi^k} &= 0 \end{aligned}$$

We end up with the simple MMPDE first obtained by Ceniceros and Hou [2001]:

$$\frac{\partial}{\partial \chi^i}\left(\omega\frac{\partial x^\alpha}{\partial \chi^i}\right) = 0 \tag{3.7}$$

This mesh generator has many advantage, among them

1. it is a set of decoupled quasi-linear elliptic equations for which many efficient numerical methods are available
2. we obtain directly the physical coordinates of the mesh points.

Among the drawbacks there is fact that folded grids can occur. Another issue is the control of the mesh quality, for example the element's skewness and the mesh alignment with respect to the flow direction. With the MMPDE (3.7) based on diagonal metric $\omega^{-1}\delta_{ij}$, we have little control on these quantities. I will briefly detail this issue later. Originally, Ceniceros and Hou [2001] have tested an implementation of (3.7) to improve the resolution of blow-up problems and heat convection. The results were promising since the method prove to be computationally efficient and capable to follow complex flow evolutions up to small scale phenomena. This approach has been quite successful, and it has been used among the others, in Tang and Tang [2003], Chen *et al.* [2008] and Zhou *et al.* [2013a].

Ceniceros and Hou [2001] derived and presented equation (3.7) as the Euler-Lagrange equation related to the following functional:

$$E(\boldsymbol{x}) = \frac{1}{2}\int_{\Omega_\chi} \omega \frac{\partial x^\alpha}{\partial \chi^i}\frac{\partial x^\alpha}{\partial \chi^i}\, d\boldsymbol{\chi} \tag{3.8}$$

It is worth to note that for this particular mesh generator with $P^k \neq -M^{ij}\Gamma^k_{ij}$, we loose the analogy with harmonics maps, as we can see by checking the difference with respect to the functional (3.6) $\rightarrow M^{ij}\sqrt{M} \neq \omega\delta^{ij}$.

Finally we think it is interesting to write (3.7) with a sort of *elastic analogy*, replacing the position with the displacement respect to the computational mesh $x^\alpha = X^\alpha + \delta^\alpha$:

$$\frac{\partial}{\partial \chi^i}\left(\omega \frac{\partial \delta^\alpha}{\partial \chi^i}\right) = \frac{\partial \omega}{\partial \chi^\alpha} \tag{3.9}$$

where $\Sigma_{\alpha i} = \omega\delta_{ij}\frac{\partial \delta^\alpha}{\partial \chi^j}$ is a *pseudo* elastic stress tensor, and the right hand side act as force on the medium $F^\alpha = \frac{\partial \omega}{\partial \chi^\alpha}$. We remark that the monitor function play a role in controlling both the stiffness of the system and the force.

3.4.1 Boundary conditions

When solving problem (3.7) we will assume that the computational domain is a closed polygon whose boundary $\partial\Omega_\chi$ is composed by the union of m segments. $\partial\Omega_\chi$ is mapped into the boundary $\partial\Omega$ and we further assume that it is invariant to the transformation. For instance we will have no moving boundaries. In particular we consider free-slip boundary conditions

$$\boldsymbol{\delta}\cdot\boldsymbol{n} = 0, \quad \boldsymbol{\chi} \in \partial\Omega_\chi \tag{3.10}$$

with $\boldsymbol{\delta} = 0$ at the polygon's vertexes. A standard method to impose boundary condition is contained in Tang and Tang [2003] where it is introduced a second map $A^\partial : \partial\mathcal{A}_0 \rightarrow \partial\mathcal{A}$ which correspond to the trace of (3.2) on the boundary. This mapping is then used as Dirichlet conditions to solve the transformation for inner points. Alternatively as shown in Li *et al.* [2002] the variational formulation could be complemented by a constraint equation to take into account (3.10). We will however stick to form (3.9), written in terms of displacement, which is suited to express directly the boundary conditions.

3.4.2 Monitor function

In general the metric tensor is related to the Jacobian of the transformation which we called $\boldsymbol{J}_A$, $\boldsymbol{M} = \boldsymbol{J}_A^T\boldsymbol{J}_A$ (see appendix A). The Jacobian controls size and orientation of mesh elements. Following Huang [2006] we write $\boldsymbol{J}_A$ using the singular value decomposition (SVD):

$$\boldsymbol{J}_A = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^T \tag{3.11}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are the orthogonal matrices associated with left and right singular vectors and $\boldsymbol{\Lambda}$ is the diagonal matrix consisting of the singular values. This decomposition has a geometrical meaning: $\boldsymbol{U}$ is a rotation matrix that specifies the orientation of physical mesh elements and $\boldsymbol{\Lambda}$ specifies the size and shape of these elements. Setting properly the SVD one can construct monitor matrix to achieve certain properties of the elements, such as anisotropy and skewness, see Alauzet [2010]. What we want to point out is the difference with the diagonal metric $\omega^{-1}\delta_{ij}$ that decouples completely the two directions. That is, x^1 and x^2 are resolved independently with the scalar monitor function ω. While this allows to achieve a quite good anisotropy, it does not guarantee the refined control of the decomposition (3.11). In practice this means that we will have limited control on mesh quality.

Now we go back to the definition of ω. A classical definition for scalar problems, given by Winslow [1967], couples the mesh motion with the gradient of the solution of the underlying PDE on the physical mesh: $\omega = \omega(\nabla u)$. As in Zhou *et al.* [2013b], we have selected the free surface η, in order to detect free surface wave patterns and bore development. We propose the following definition of the monitor function

$$\omega = \sqrt{1 + \alpha \left(\max\left(||\nabla\eta||^*, ||\nabla^2\eta||^*\right)\right)^2 + \delta\phi^2} \tag{3.12}$$

We see that when $||\nabla\eta||^*, ||\nabla^2\eta||^*, \phi \to 0$, the metric becomes the Euclidean one $M_{ij} = \delta_{ij}$ and the mesh tends to the unperturbed state. $\|\cdot\|$ represent normalized L^2-norms computed as

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

The coefficients α, β and γ are free parameters, allowing to optimize the mesh movement. Note that in all of the above formulas, the derivatives of η are computed on the physical (moving) mesh, making problem (3.7) nonlinear. In (3.12) we have also tested the influence of the Hessian of the free surface and a tracking of the wet/dry interface. There are in literature some examples of such front-tracking error functions. For example, in the context of phase change problems, J.A.Mackenzie and W.R.Mekwi [2007] defined $\omega = \alpha/\sqrt{\beta|\boldsymbol{x} - \boldsymbol{x}_{interf}| + \gamma}$. This expression, like others, 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

$$\phi = \max(\nabla f(\boldsymbol{x}), \nabla^2 f(\boldsymbol{x})) \tag{3.13}$$

and f is a function which is constant everywhere except in the narrow region

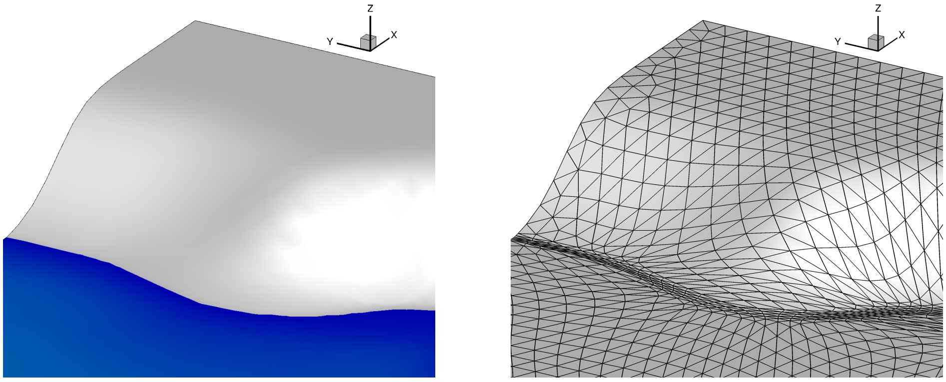

Figure 3.3: Example of mesh adaptation to the wet/dry interface.

where $C_H < h < C_U$:

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

In figure 3.3 we report an example to show how this technique may work. The mesh is refined over the wet/dry cells and, due to the presence of the Hessian norm in (3.13), it is also refined one cell after the interface in dry region. We have added the Hessian contribution after having observed that a more spread mesh refinement at the interface, ensures robust flooding simulation.

3.4.3 Mesh Smoothing

The smoothness of the mesh is measured in terms of the variation of the local element's size h_K over the domain Ω. More rigorously, given the map (2.6), a mesh has degree of regularity r if the Jacobian $\boldsymbol{J} \in C^r(\Omega)$, see Budd *et al.* [2009a]. Mesh regularity is important because it is widely recognized that abrupt variation of the mesh size leads to a deterioration of the numerical solution. Typically, meshes obtained by (3.7) are smooth, but, in presence of non convex boundaries $\partial\Omega$ loss of regularity can occur, ultimately leading to mesh tangling in extreme cases. Huang and Russell [1997] studied the property of the following one dimensional equation:

$$\left(1 - \nu \frac{\partial^2}{\partial X^2}\right) x = \hat{x}$$

and demonstrated that x satisfies the following smoothness condition:

$$|J| = \left| \frac{\partial x}{\partial X} x^{-1} \right| < \nu^{-2}$$

The two dimensional extension can be found in Budd *et al.* [2009a] and reads:

$$\left(1-\nu\frac{\partial^2}{\partial X^i\partial X^i}\right)x^\alpha_{sm}=x^\alpha \tag{3.14}$$

This MMPDE is particularly efficient to resolve because the operator

$$L=1-\nu\frac{\partial^2}{\partial X^i\partial X^i}$$

can be inverted once on the computational mesh, and then used to compute a smoother variation of a given mesh (typically the solution of (3.7)) as:

$$x^\alpha_{sm}=L^{-1}x^\alpha$$

To avoid mesh tangling at the corner of non convex domain, we have defined a variable diffusion coefficient based on the distance form the boundaries, $d_{P,\partial}=d(\boldsymbol{x}_P,\partial)$. Since the boundaries do not move, the distance is computed once and then successively interpolated.

$$\nu_i=\max\left(0,100-\left(\frac{d_{i,\partial\Omega}}{h_k}\right)^2\right)$$

In figure 3.4 we can see a case where smoothing is not applied and the mesh tangles at the corner, while, applying a sequence of the two MMPDE (3.7) and (3.14) at each time step, we observe a smoother mesh without tangling. We remark that the above definition of the diffusion parameter is heuristic and there is still no guarantee that tangling never occur.

3.4.4 Numerical resolution of MMPDE

In practice, given an initial mesh in the computational domain, the weak form of (3.7) with boundary conditions (3.10) is discretized with a standard P^1 Galerkin finite element method. We search for an approximate solution $x_h\in V_h$ such that

$$\int_{\Omega_\chi}\frac{\partial x^\alpha_h}{\partial\chi^m}\omega\frac{\partial v_h}{\partial\chi^m}\,d\boldsymbol{\chi}=0\qquad\forall v_h\in V_h,\ \alpha=1,2$$

If we use continuous piecewise polynomials, the FE solution

$$x^\alpha_h=\sum_{j\in\mathcal{T}_h}\varphi_j x^\alpha_j$$

$\{\varphi_i(x^1,x^2)\}_1^N$ is the Lagrangian basis and x^α_j coincides with the position of the node labeled by j. The standard development is:

$$\sum_{j\in\mathcal{T}_h}\int_{K_\chi}\frac{\partial\varphi_j}{\partial\chi^m}\omega\frac{\partial\varphi_i}{\partial\chi^m}\,d\boldsymbol{\chi}\,x^\alpha_j=0\qquad\forall i\in\mathcal{T}_h,\ \alpha=1,2$$

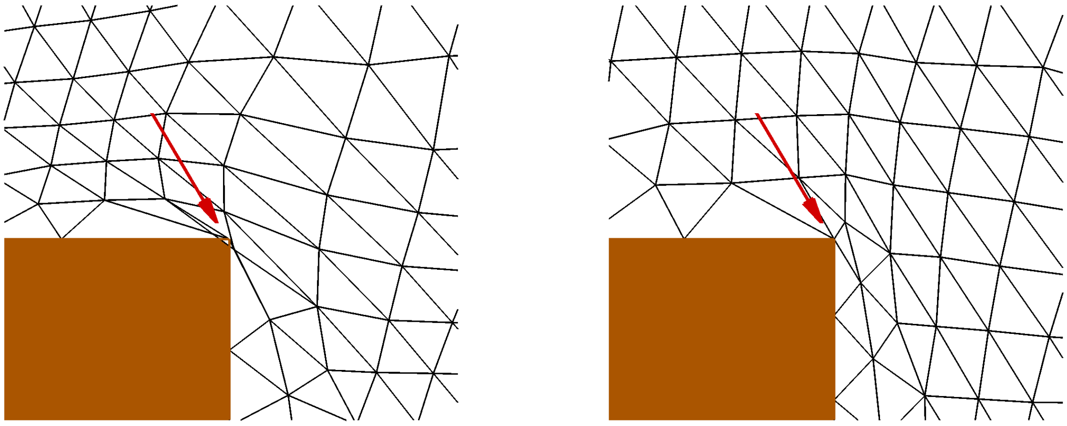

Figure 3.4: Tangling for non convex domain. right) non smoothed mesh left) smoothed mesh

Using the fact that $\frac{\partial \varphi_i}{\partial \chi^m}$ is constant on an element

$$\frac{\partial \varphi_j}{\partial \chi^m} = \frac{1}{2|K_\chi|} n_j^m$$

We can write the algebraic system in terms of displacement $\delta^\alpha = x^\alpha - \chi^\alpha$.

$$\sum_{j \in \mathcal{T}_h} a_{ij} \, \delta_j^\alpha = b_j^\alpha \qquad \forall i \in \mathcal{T}_h, \; \alpha = 1,2 \tag{3.15}$$

with:

$$\begin{aligned} b_j^\alpha &= \sum_{j \in \mathcal{T}_h} a_{ij} \chi_j^\alpha \\ a_{ij} &= \sum_{K \in \mathcal{D}_i \cup \mathcal{D}_j} \int_{K_\chi} \omega \, d\boldsymbol{\chi} \, \frac{1}{4|K|^2} n_i^m n_j^m \end{aligned}$$

Due to the dependence of ω on the derivatives of η on the new mesh, the weak form (3.15) 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 with respect to the evolution of the flow quantities with the explicit schemes discussed in chapter 2. In this regard it is worth noting that the eRK2-RD method is ten times faster compared to the implicit version, Ricchiuto [2015]. For all these reasons, we have chosen a simple explicit Newton-Jacobi iteration method, as in Chen *et al.* [2008]. In particular, at each time step, the displacement $\boldsymbol{\delta}^k = \boldsymbol{x}^k - \boldsymbol{x}^n$ is computed from the following relaxed iteration

$$\hat{\boldsymbol{\delta}}_i^{k+1} = \boldsymbol{\delta}_i^k - \frac{1}{a_{ii}} \sum_{j \in \mathcal{D}_i} a_{ij} \boldsymbol{x}_j^k \tag{3.16}$$

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^n + \mu_i \hat{\boldsymbol{\delta}}_i^k \tag{3.17}$$

Note that the update (3.16) is equal to the one of Chen *et al.* [2008], 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 Tang and Tang [2003], Chen *et al.* [2008])

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

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 a_{ij} 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 3.5.

3.4.5 Mesh Generation

We test the moving mesh algorithm defined by (3.16) on the benchmarks proposed in Tang and Tang [2003]. The monitor function is computed according to $\omega = \sqrt{1 + \alpha u^2}$, with u assigned:

$$u(x, y) = \exp(-8(x^2 + 9y^2 - 1)^2) \tag{3.18}$$

$$u(x, y) = \exp(-100(y - x^2 + 0.5)^2) \tag{3.19}$$

$$u(x, y) = 50\exp(-2500(x^2 + y^2) \tag{3.20}$$

$$u(x, y) = \begin{cases} 1 & \text{if } y = x \\ 0 & \text{if } y \neq x \end{cases} \tag{3.21}$$

with $x = x^1$ and $y = x^2$. The reference domain is a square $[-1, 1] \times [-1, 1]$. The iteration is repeated in the pseudo-time loop until convergence is reached. For the first two smooth examples reported in figure 3.5 we have set $\alpha = 100$. In the left picture of figure 3.6 we show that also singularities are well handled

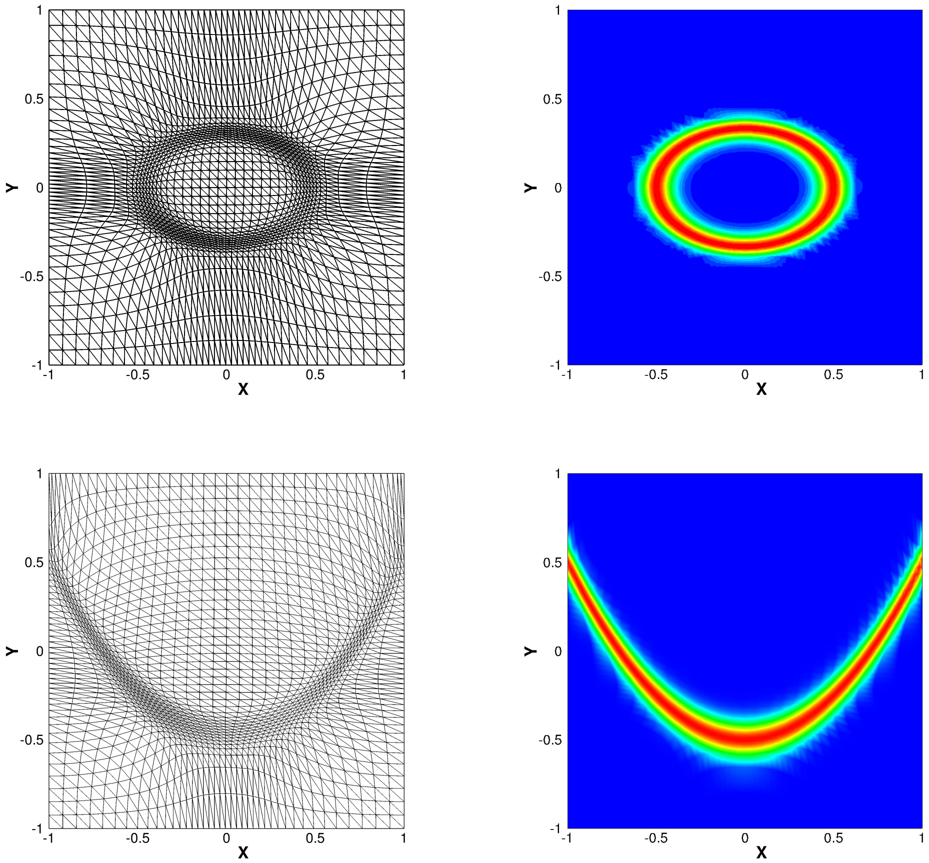

Figure 3.5: Top) test (3.18). Bottom) test (3.19)

with a proper choice of the parameter. Finally always in figure 3.6 on the right, there is a test for an oblique shock: the zoom shows that anisotropy in the direction of the shock is achieved without tangling occurrence.

3.5 High order projections from ALE remaps

As already said, the Newton-Jacobi iterations (3.16) and (3.17) require the projection of the solution values on the last updated mesh. The problem have already been formalized in section 2.3.1. We want to evaluate scalar and vector properties of the fluid through the ALE transformation. For the vector of flow variables:

$$\mathsf{w} = \mathsf{w} \circ A = \mathsf{w}_0\left(\chi\right)$$

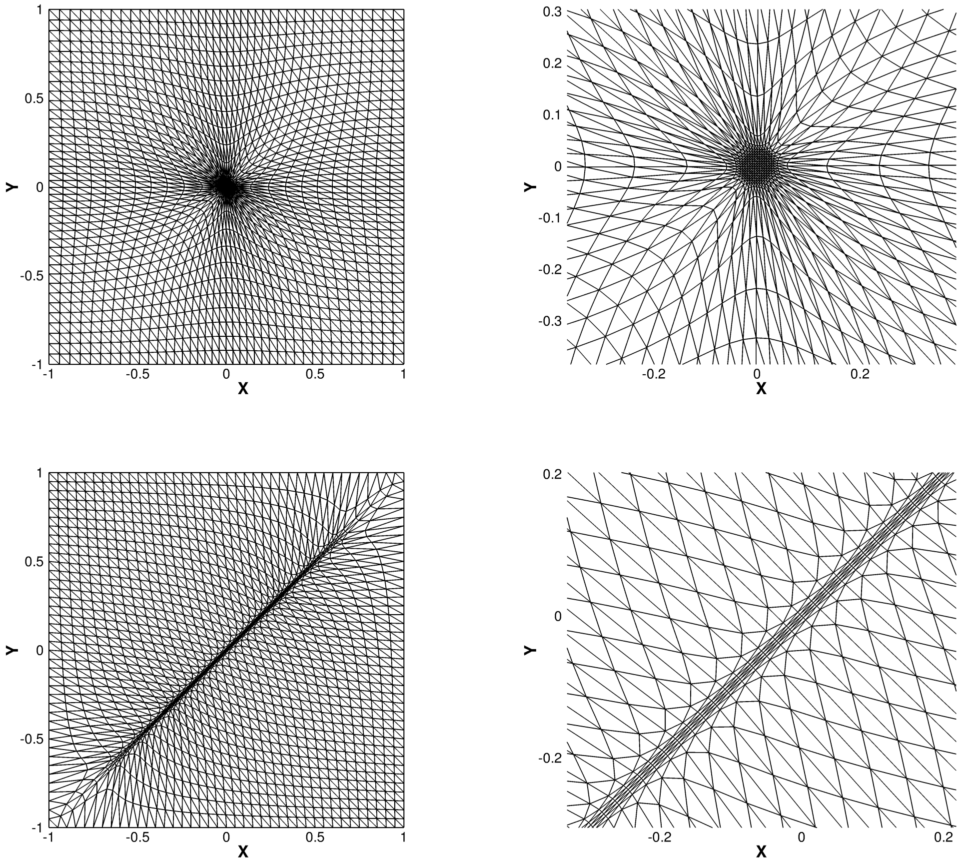

Figure 3.6: Top) test (3.20). Bottom) test (3.21).

We recall here the remap equation (2.15):

$$\left.\frac{\partial J_A \mathsf{w}}{\partial t}\right|_{\chi} - J_A \frac{\partial \mathsf{w}\sigma^j}{\partial x^j} = 0 \tag{3.22}$$

and we remark that the interpolation has been formalized into a typical hyperbolic problem in which the solution is advected by the domain movement. It is not a coincidence that it is usually referred to as *advection remap*.

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 discretize the PDEs. Such a remap corresponds to the limit for $\Delta t \to 0$ of the schemes presented in sections 2.10 and 2.11 with a "frozen" the fluid configuration $\mathcal{B}(t^*)$, but letting the mesh configuration $\mathcal{A}(t)$ continue to move. This provides an instantaneous approximation to the conservative ALE remap equation.

Practically, during the iterations (3.16) we generate a sequence of sub-grids $\mathcal{T}_h^{k+1}$ on which we want to interpolate the last available numerical solution. If we are at the k-th iteration, the last available solution is $\mathsf{w}_i^k = \mathsf{w}(\boldsymbol{x}_i^k)$.

FV ALE remap

For the FV scheme, taking the limit for $\Delta t \to 0$ of (2.37) we obtain the one step projection over the sub-grid $\mathcal{T}_h^{k+1}$

$$|C_i^{k+1}|\mathsf{w}_i^{k+1} = |C_i^k|\mathsf{w}_i^k - \sum_{j\in\mathcal{D}_i} R_{ij}(\mathsf{w}^k) \tag{3.23}$$

and

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

the interface velocity is replaced by an interface displacement (which coincides with the area swept by the same interface):

$$\Delta x_{ij} = \int_{\partial C_{ij}} \Delta x^m n^m \, ds$$

$\Delta x^m = x^{m,n+1} - x^{m,n}$ is the displacement of the interface during a single time step. 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, referred to as MUSCL in section 1.6.2. 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.

RD ALE remap

In the ALE remap the RD fluctuation defines

$$\Phi^K(\mathsf{w}_h) = -\int_K \Delta x^j \frac{\partial \mathsf{w}}{\partial x^j} d\boldsymbol{x}$$

We then take the limit $\Delta t \to 0$ in the RD predictor step, see (2.55):

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

with $\Phi_i^{K(1)} = \Phi^K(\mathsf{w}_h^k)$. For the corrector, see (2.56):

$$\mathsf{w}_i^{k+1} = \mathsf{w}_i^* - \frac{1}{|C_i^{n+1}|} \sum_{K \in \mathcal{D}_i} \Phi_i^{K(2)}(\mathsf{w}_h^*, \mathsf{w}_h^k) \tag{3.24}$$

with

$$\Phi^{K(2)} = \frac{1}{2}\left(\Phi^K(\mathsf{w}_h^k) + \Phi^K(\mathsf{w}_h^*)\right) + \int_K \frac{\mathsf{w}_h^* - \mathsf{w}_h^k}{\Delta t} d\boldsymbol{x}$$

We remark a difference with respect to FV. To obtain a second order remap the two steps projection must be carried, making the RD remap less efficient then the FV one. We believe that this is related to the presence of the mass matrix. With such a two steps projection, the non linear splitting $\Phi_i^K = \Phi_i^{LLxF-SUPG}$ of section 1.6.3 allows to project the flow variables retaining all the properties of the RD LLxF-SUPG scheme (second order, non oscillatory solutions, well-balanced and mass conservation).

3.6 Adaptive algorithms

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 algorithm, generally speaking, alternate at every time step the solution of the MMPDE and of the underlying PDEs as follows:

- compute the monitor function based on the current solution
- evolve the MMPDE and compute the new mesh
- evolve the PDEs and compute the new solution

This has been successfully implemented by Huang and Russell [1999] showing a significant reduction of stiffness problems with respect to the full-coupled approach. At the same time is it has been recognized that it can lead to a lag in the mesh movement with respect to the physical features.

In this thesis, depending on the framework in which we evolve the PDEs, two different *alternate* algorithms are tested. If the PDEs are written in Eulerian framework one gets the *rezoning* method suggested in Tang and Tang [2003]. This approach is based on a sequence of mesh and flow iterations. It 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 ALE formulation of the PDEs. In this case, the properties of the flow solutions are only determined by the scheme. In the moving mesh community this approach is called *quasi Lagrangian*, see Huang and Russell [1998] and Cao *et al.* [1999]: time derivatives are transformed along mesh trajectories and the resulting PDEs are a non conservative form of our balance laws (2.18). Given a PDE in Eulerian form

$$\frac{\partial \mathsf{u}}{\partial t} + \mathcal{L}(\mathsf{u}, \boldsymbol{x}, t) = 0$$

it can be transformed using the relationship between Eulerian and ALE time derivative (2.10)

$$\left.\frac{\partial \mathsf{u}}{\partial x}\right|_{\chi} - \left.\frac{\partial x^j}{\partial t}\right|_{\chi} \frac{\partial \mathsf{u}}{\partial x^j} + \mathcal{L}(\mathsf{u}, \boldsymbol{x}, t) \quad = \quad 0$$

We note that this approach hides the GCL. The consequences is that, approximation of the above ALE equations can represent exactly a constant uniform flow on moving meshes without the need to conserve the total mesh volume along the simulation; this latter statement in fact is not an issue in Huang and Russell [1998]. A combination of the alternate algorithm and the ALE approach in conservation form can be found in Ni *et al.* [2009].

We have now all the basic blocks to perform adaptive mesh simulations. These boil down to the flow evolution equations (section 2.10 and 2.11) and to the MMPDE, discussed in section 3.2. 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. Tang and Tang [2003] and Ni *et al.* [2009] for the ALE). Here *we will compare their impact on the overall cost of the simulation, and on the quality of the results.* The set of nodes' coordinates is called $\mathbf{x} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_i, ..., \boldsymbol{x}_N\}$ with N the number of mesh nodes. 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 a good node refinement.

3.6.1 Moving Mesh ALE algorithm (ALE)

In this approach the balance law is written using 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 $\mathsf{w}_h^n$. Set the initial conditions for the MMPDE, $\eta_h^1 = \eta_h^n$ and $\mathbf{x}^1 = \mathbf{x}^n$.

DO k=1,kmax=5

Step 2. Compute the monitor function $\omega^k = \omega(\eta_h^k)$ and, in turn, the moving mesh matrix $a_{ij} = a_{ij}\left(\omega^k\right)$. Move the mesh according to the Newton-Jacobi iteration (Eq. (3.16) and (3.17)). At each iteration we get $\mathbf{x}^{k+1}$.

Step 3. Compute the interpolated free surface $\eta_h^{k+1}$ according to the scalar version of FV/RD projections, (3.23) or (3.24).

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-eRK2 scheme, see Eq. (2.38) or Eq. (2.56) on the grid $\mathcal{T}_h^{n+\frac{1}{2}}$.

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. As a consequence, we thought that the interpolation step can be simplified a great deal without affecting the quality of the solution, as the numerical tests will confirm.

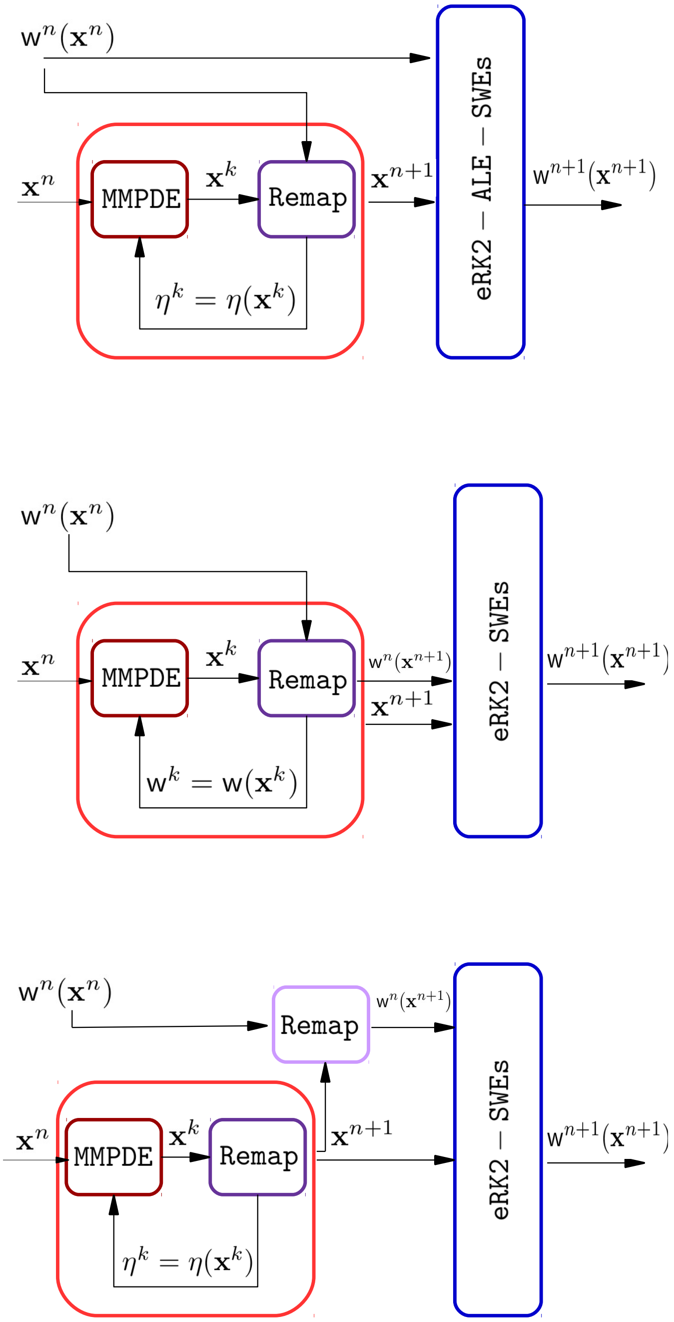

Figure 3.7: Moving mesh algorithms. Top) ALE. Middle) Rezoning (EUL1). Bottom) Rezoning (EUL2).

3.6.2 Moving Mesh Eulerian algorithm/rezoning (EUL1)

In this case, 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 $\mathsf{w}_h^n$. 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=5

    Step 2. Compute the monitor function $\omega^k = \omega(\eta_h^k)$ and matrix $a_{ij} = a_{ij}\left(\omega^k\right)$. Move the mesh according to the Newton-Jacobi iteration (Eq. (3.16) and (3.17)). At each iteration we get $\mathbf{x}^{k+1}$.

    Step 3. Compute the full interpolated solution $\mathsf{w}_h^{k+1}$ according to FV/RD projections, see (3.23) or (3.24).

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-eRK2 scheme, see Eq. (2.38) and Eq. (2.56) 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. Here ALE remaps based on high resolution schemes are used to this purpose.

3.6.3 Modified 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. If what we thought about the simplification of the ALE remap would be confirmed by numerical experiments, then we could use a simplified version of the remap algorithm to project only the free surface variable for then estimating the error. A full high resolution remap is used only after reaching $k = kmax$ in the adaptation loop, to perform the interpolation. For example for FV:

$$|C_i^{n+1}|\mathsf{w}_i^{kmax+1} = |C_i^n|\mathsf{w}_i^n - \sum_{j\in\mathcal{D}_i} R_{ij}(\mathsf{w}_i^n)$$

3.6.4 Adaptive algorithms: efficiency

To test the efficiency of the different coupling algorithms proposed in section 3.6 (ALE, EUL1, EUL2), we propose two scalar tests: linear rotation and Burgers'equation. We would like to confirm the feeling that, for ALE and EUL2, the numerical resolution of the remap equation into the MMPDE could be quite inaccurate (e.g. first order) without destroying mesh quality. Can this result leads to more efficient algorithms with respect to the classical rezoning/EUL1?

The following MMPDE parameters are used in all the tests $\alpha = 10$, $\beta = \gamma = 0.15$. We did not perform a systematic optimization relative to these parameters. The value used correspond to those that visually provided the best mesh.

Rotation

This is the classical rotation of a smooth sinusoidal hill, but with a source term. We recall our general scalar balance law (2.65):

$$\frac{\partial u}{\partial t} + \frac{\partial F^j}{\partial x^j} + S(u, \boldsymbol{x}) = 0\,, \qquad \boldsymbol{x} \in [-1,1]\times[-1,1],\ t \in [0,\pi] \tag{3.25}$$

where, for a given flux vector $\boldsymbol{F}(u) = [\,F^1\ F^2\,]$, the source term is defined as in section (2.12.1)

$$S = a^j \frac{\partial b}{\partial x^j}$$

this time with $b(x,y) = 0.8e^{\psi(x,y)}$ and $\psi = -5y^2 - 5x^2$. The following definition of the fluxes will be used, $\boldsymbol{F}(u) = \boldsymbol{a}u$, with $\boldsymbol{a} = [\,-2y\ 2x\,]$. The initial condition for the "free surface" $\eta = u + b$ is

$$\begin{cases} \eta_0(\boldsymbol{x}) = 1 + \cos^2(2\pi r) & \text{if } r \le 0.25,\ r = \sqrt{x^2 + (y-0.5)^2} \\ \eta_0(\boldsymbol{x}) = 1 & \text{otherwise} \end{cases}$$

We perform a grid convergence study, and investigate the dependence of the

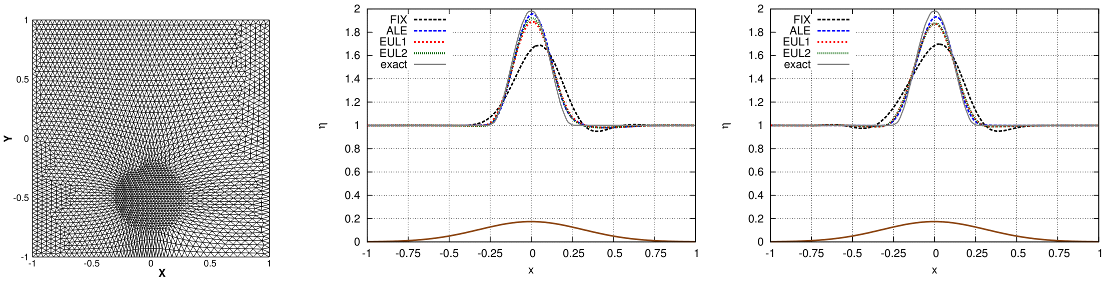

Figure 3.8: Rotation test. left) adapted mesh for the second level mesh. center) adaptive FV. right) adaptive RD.

error on the CPU time. We perform the same test for both the RD and FV scheme. In the left picture of figure 3.10 and 3.9 the convergence curves for the different combinations of moving mesh algorithms and interpolations schemes are reported. For the ALE algorithm and EUL2 (not shown in figure), we see that all the curves in blue color, corresponding to different remap schemes, are almost overlapped (dark blue is first order LxF/upwind FV, electric blue is a linear centered approximation, light blue is a second order scheme SUPG/FROMM). On the contrary for the EUL1 algorithm there is only one interpolation scheme which guarantees stable and second order accurate results, actually the one which we evolve the PDE with, namely the SUPG/FROMM scheme. We can summarize saying that, even if the interpolation of the monitor function has a positive impact on the resolution of the MMPDE and reduces mesh delay, *the specific scheme used, weakly influences mesh configuration.*

In the right picture of figure 3.10 and 3.9 the performances of the different algorithms are compared in terms of error/time. For the RD method, the ALE algorithm shows the lowest CPU time for a fixed error level (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 of the ALE and Eulerian algorithms is more similar (ALE and EUL2 80%, EUL1 70%). The reason is that, in this case, the second stage of the interpolation is not necessary, Tang and Tang [2003]. Finally, for both RD and FV, the EUL2 represents a slight improvement respect to the EUL1 algorithm.

Burgers'equation

In this section we test test if the above conclusions are true when discontinuities develop. Solutions with discontinuities are obtained with a Burgers equation and discontinuous initial conditions. We set in (2.65) $S = 0$, $\boldsymbol{F}(u) = [\,\frac{u^2}{2}\;\frac{u^2}{2}\,]$.

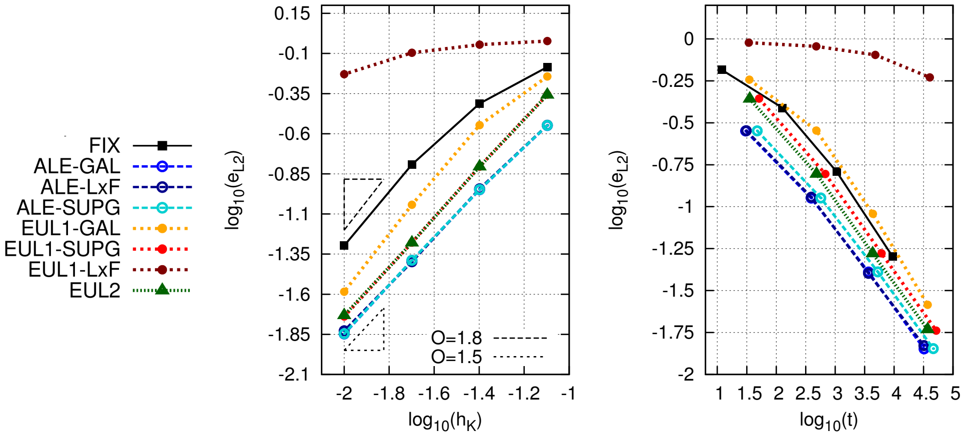

Figure 3.9: Rotation test with RD. left) order of convergence. right) Error vs CPU time.

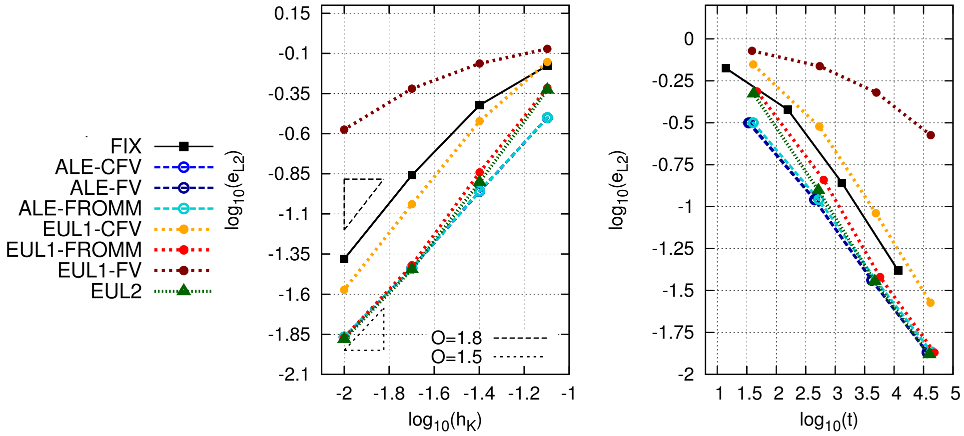

Figure 3.10: Rotation test with FV. left) order of convergence. right) Error vs CPU time.

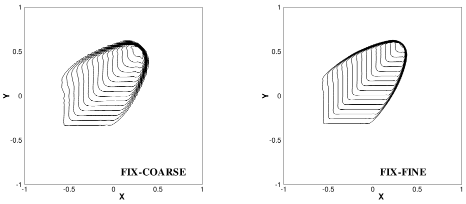

Figure 3.11: Burgers Equation computed with LLxF-SUPG scheme on fixed grid. Left: coarse mesh $h_K = 1/40$. Right: fine mesh $h_K = 1/100$.

The initial condition is

$$\begin{cases} u_0 = 1 & \text{if } \boldsymbol{x} \in [-0.6, -0.1] \times [-0.35, 0.15] \\ u_0 = 0 & \text{otherwise} \end{cases}$$

A reference solution is computed on a fixed unstructured mesh with $h_K = 1/100$, which is referred to as the fine mesh. To test the effectiveness of the adaptation algorithm, computations are performed on a fixed structured mesh with element reference size $h_K = 1/40$, which is referred to as coarse. A good mesh refinement is obtained in correspondence of the discontinuity. The comparison between EUL1, EUL2 and ALE strategies is made for both the scalar RD and FV.

From figure 3.13 to figure 3.20, the results for RD and FV are shown. This time we have found that the interpolation step slightly affect the mesh adaptation. We have reported the extreme cases where centered approximation and first order remap scheme are used. We can observe that the *first order remaps produces slightly smoother mesh at discontinuities*; of course this is as a consequence of the fact that we compute a smoother monitor function. The reader may refer to Huang and Russell [1997] for the consequence of a proper smoothing of the monitor function. For the ALE, the CPU time remains more then two times smaller respect the one obtained with the fine grid, while the two solutions are downright comparable. The EUL2 algorithm is obtained using a Galerkin remap into the MMPDE. The advantage respect to EUL1, in term of CPU time, is clear from table (3.1). In particular for RD, the full two stages eRK2 remap, makes for this test, the EUL1 not efficient.

Using these results we highlight that:

- in the following numerical test cases, *the EUL2 and ALE algorithms will be used in their faster versions with inaccurate centered approximation remap into the MMPDE.*

3.7 *r*-adaptation for Shallow Water flows

In the final section of the chapter we present a thorough study of the coupling algorithms in terms of accuracy, and CPU time for both simple Shallow Water academic problems and some classical benchmarks involving the long wave

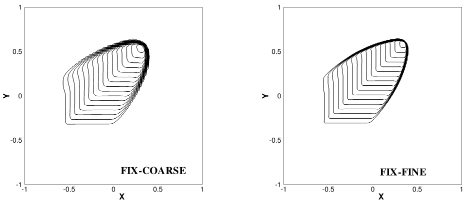

Figure 3.12: Burgers Equation computed with MUSCL scheme on fixed grid. Left: coarse mesh $h_K = 1/40$. Right: fine mesh $h_K = 1/100$.

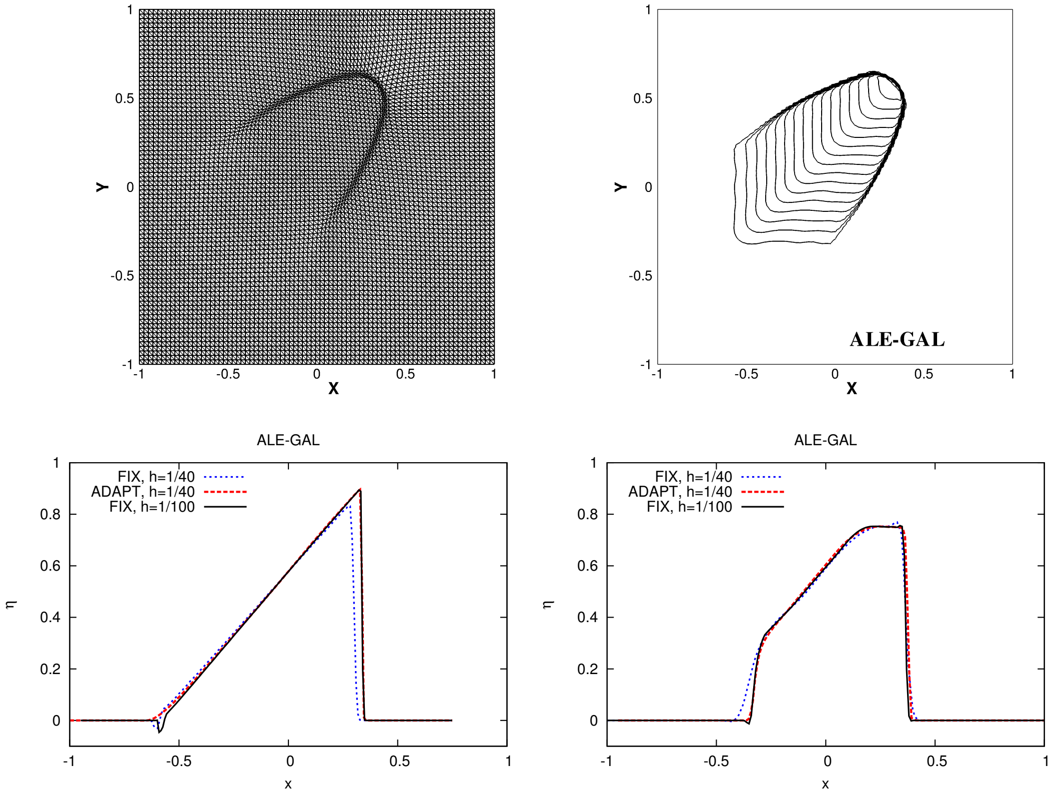

Figure 3.13: Burgers equation computed with ALE-RD with centered linear projection scheme (GAL) to compute the error estimate in the MMPDE. Top) adapted grid at final time, 20 equispaced solution isolines between 0 and 1. Bottom) comparison of the solution along the symmetry lines and the lines at $y = 0.4$

ALG.	MESH (Nodes)	RD [s]	FV [s]
FIX-COARSE	6561	58.52	61.89
FIX-FINE	40401	619.75	647.51
ADAPT-ALE	6561	271.34	282.31
ADAPT-EUL1	6561	629.77	499.30
ADAPT-EUL2	6561	406.06	406.05

Table 3.1: Burgers'equation. CPU times.

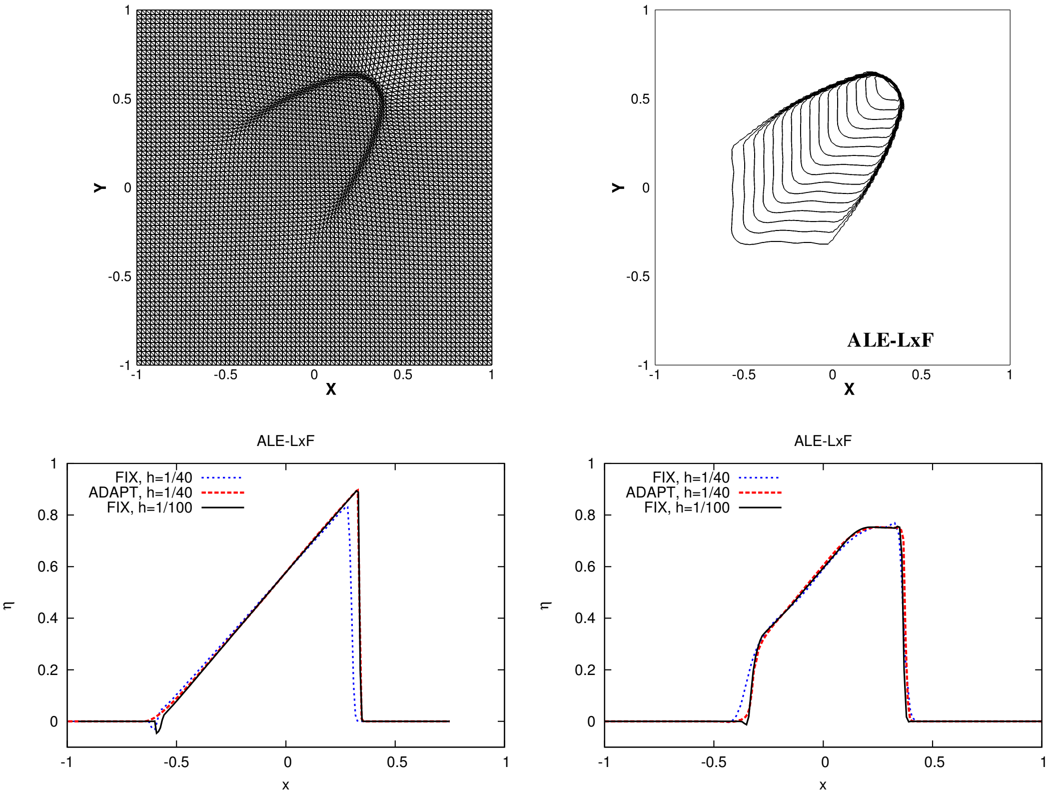

Figure 3.14: Burgers equation computed with ALE-RD with first order projection scheme (LxF) to compute the error estimate in the MMPDE. Top) adapted grid at final time, 20 equispaced solution isolines between 0 and 1. Bottom) comparison of the solution along the symmetry lines and the lines at $y = 0.4$

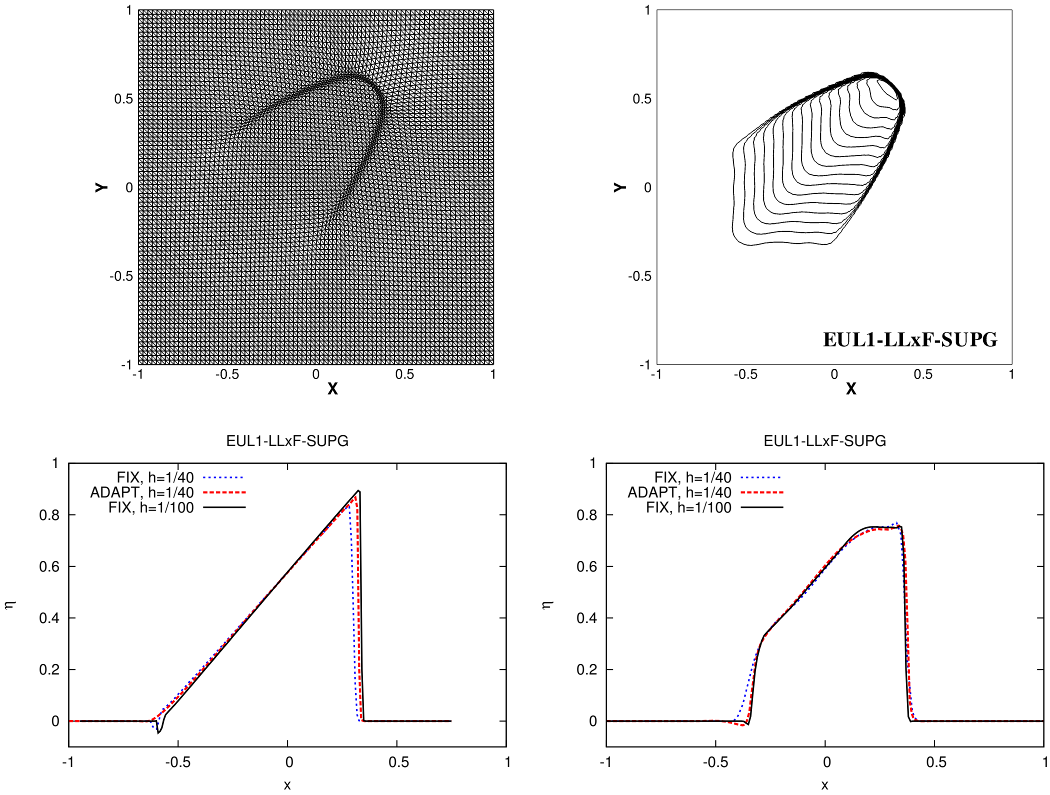

Figure 3.15: Burgers equation computed with EUL1-RD with non linear projection scheme (LLxF-SUPG). Top) adapted grid at final time, 20 equispaced solution isolines between 0 and 1. Bottom) comparison of the solution along the symmetry lines and the lines at $y = 0.4$

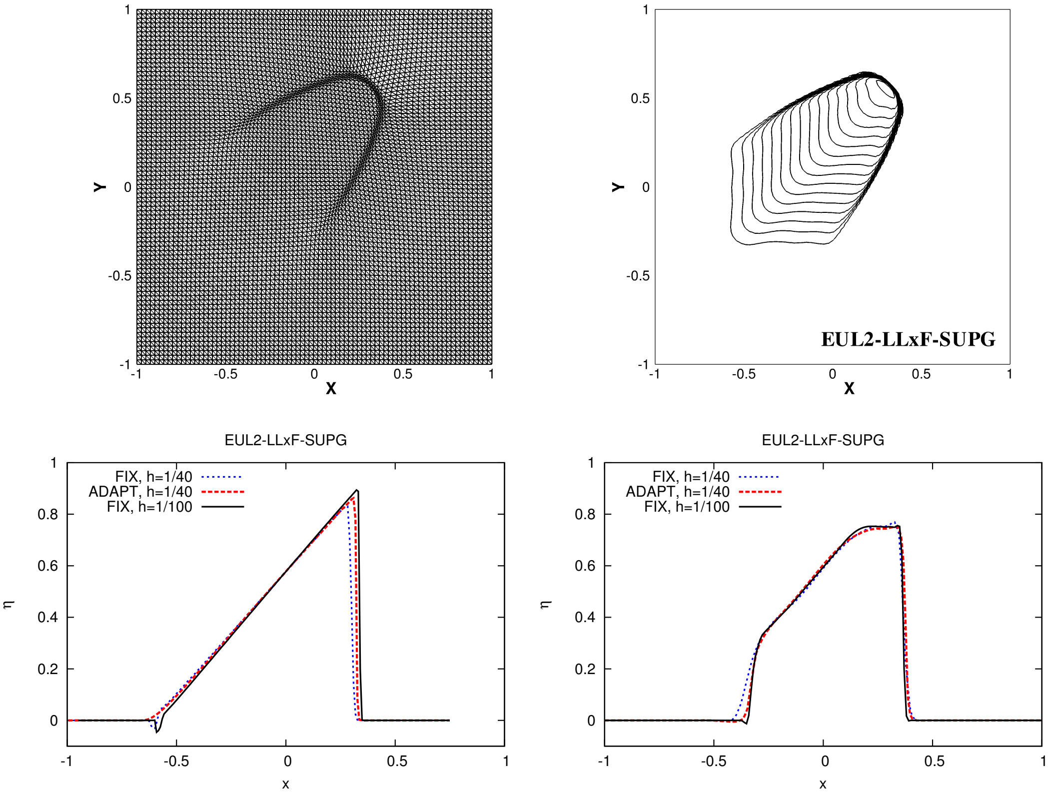

Figure 3.16: Burgers equation computed with EUL2-RD with linear centered projection scheme in the MMPDE and non linear scheme (LLxF-SUPG) in the remap of flow variables. Top) adapted grid at final time, 20 equispaced solution isolines between 0 and 1. Bottom) comparison of the solution along the symmetry lines and the lines at $y = 0.4$

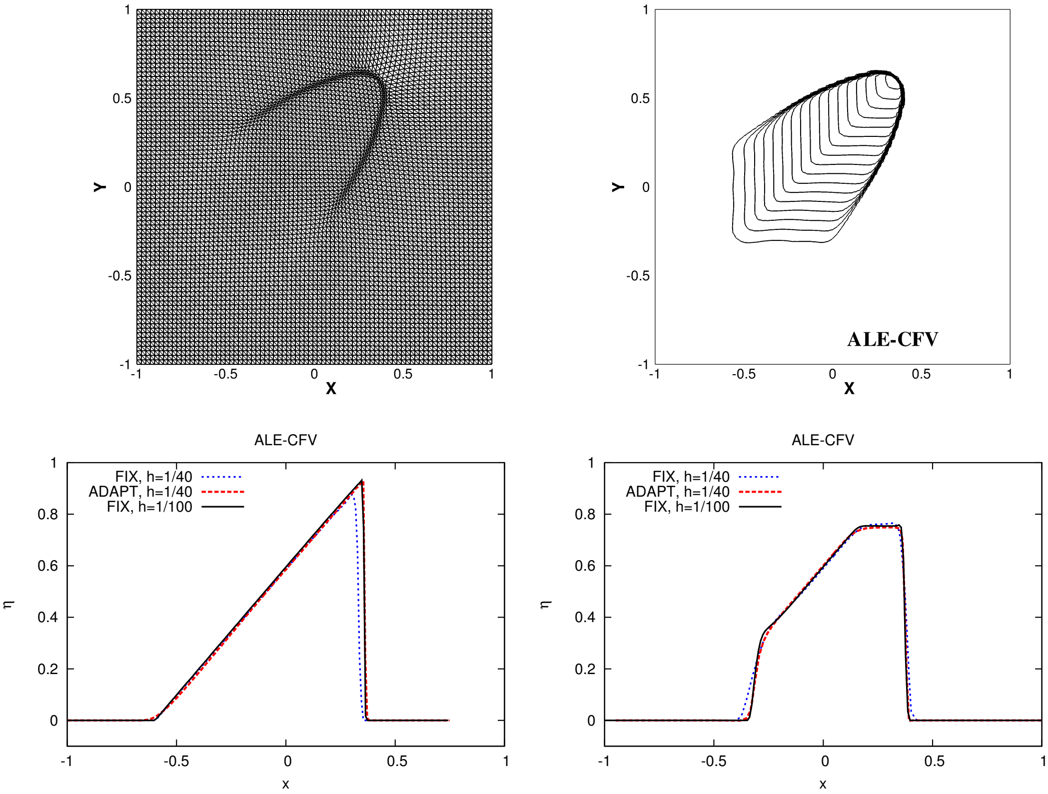

Figure 3.17: Burgers equation computed with ALE-FV with centered linear projection scheme (CFV) to compute the error estimate in the MMPDE. Top) adapted grid at final time, 20 equispaced solution isolines between 0 and 1. Bottom) comparison of the solution along the symmetry lines and the lines at $y = 0.4$

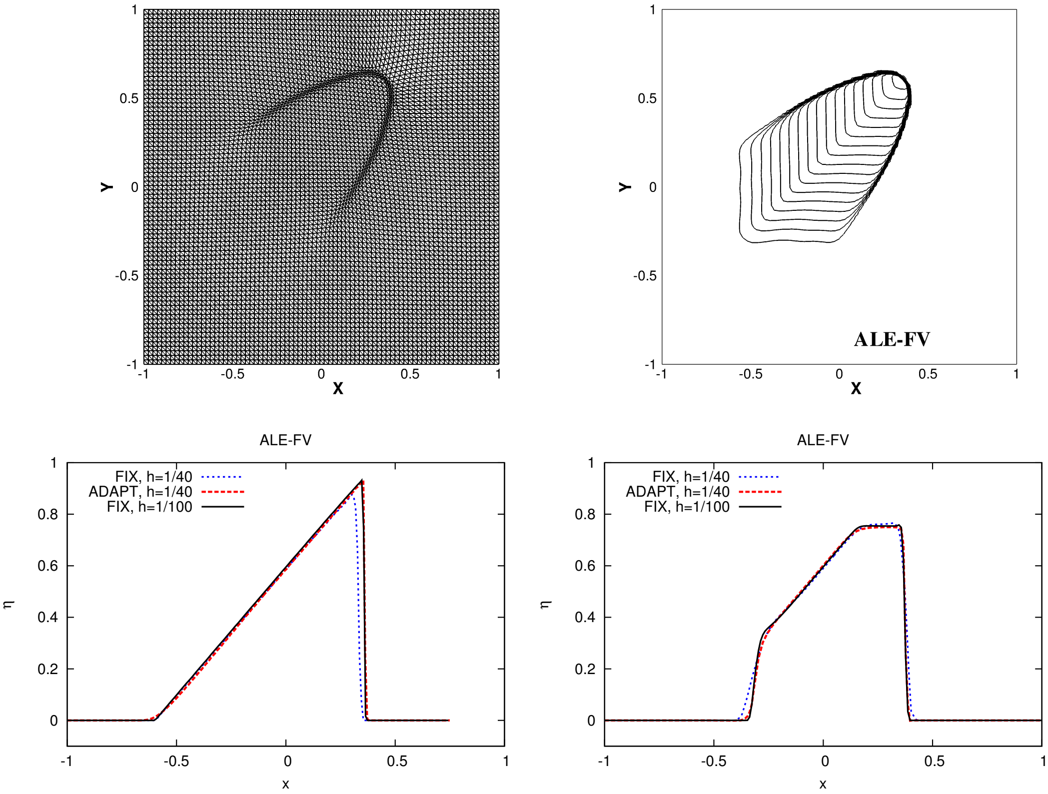

Figure 3.18: Burgers equation computed with ALE-FV with first order projection scheme (upwind FV) to compute the error estimate in the MMPDE. Top) adapted grid at final time, 20 equispaced solution isolines between 0 and 1. Bottom) comparison of the solution along the symmetry lines and the lines at $y = 0.4$

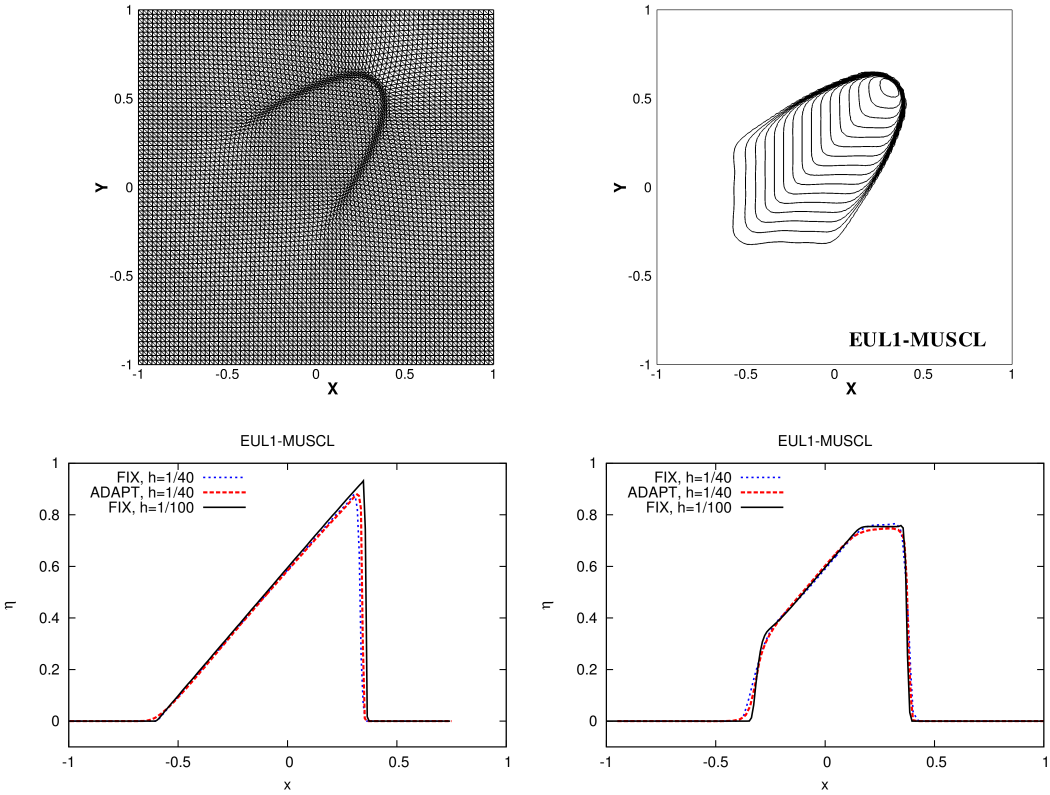

Figure 3.19: Burgers equation computed with EUL1-FV with non linear projection scheme (MUSCL). Top) adapted grid at final time, 20 equispaced solution isolines between 0 and 1. Bottom) comparison of the solution along the symmetry lines and the lines at $y = 0.4$

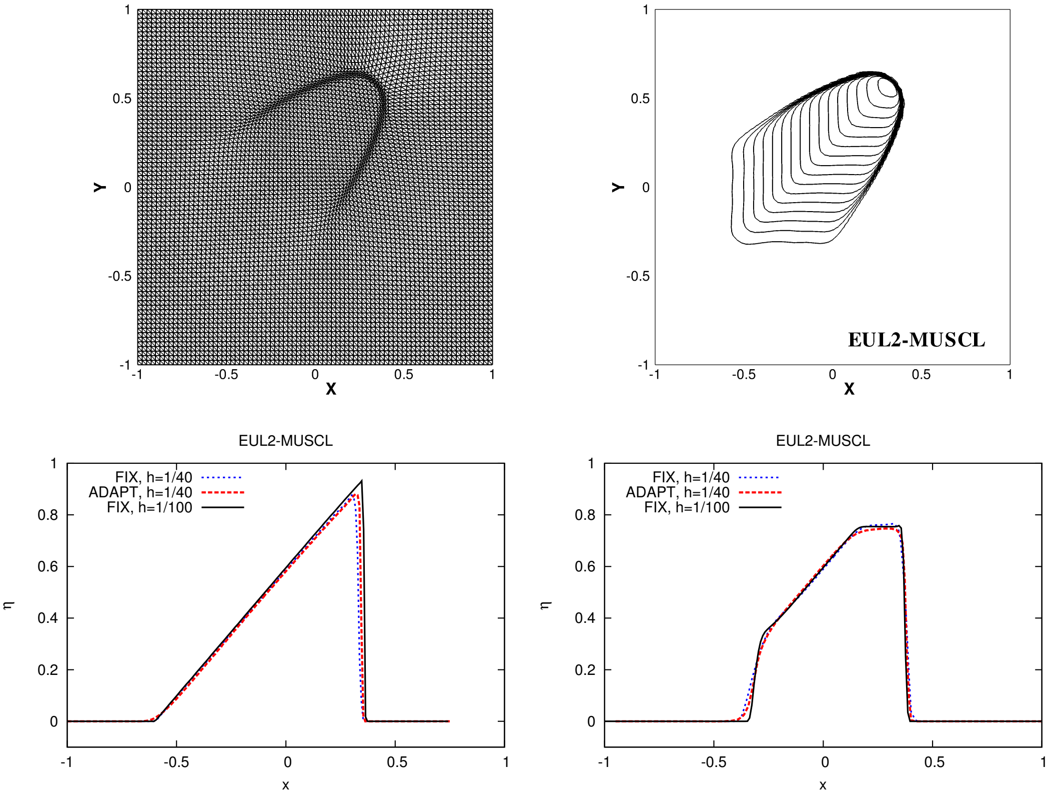

Figure 3.20: Burgers equation computed with EUL2-FV with linear centered projection scheme in the MMPDE and non linear scheme (MUSCL) in the remap of flow variables. Top) adapted grid at final time, 20 equispaced solution isolines between 0 and 1. Bottom) comparison of the solution along the symmetry lines and the lines at $y = 0.4$

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 3.2: Asymmetric dam-break. CPU times.

runup on complex bathymetries. In SWEs simulations, the MMPDE parameters used in the monitor function (see section §3.2) are $\alpha = 20$, $\beta = \gamma = 0.10$ and $\delta = 3\alpha$, unless otherwise specified. For the relaxation parameters we used $\tau = 3$ and $\vartheta = 0.7$.

3.7.1 Asymmetric dam Break

This classical test benchmark, taken from Seaid [2004], 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 basin 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 basin and a bore advancing in the lower basin. 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 3.21 and 3.22. The pictures show the potential of this adaptation procedure to *provide with considerably fewer unknowns a better resolution of the breaking bore.*

In figures 3.23,3.24 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 3.2, the savings obtained with the ALE algorithm go up to 60% for RD, and 50% for FV. For the RD scheme, the cost of a two-step interpolation, makes the EUL1 algorithm inefficient, thus the EUL2 is a clear improvement. For FV both the interpolation based algorithms (EUL1 and EUL2) are not able of providing a considerable improvement in the resolution of the peaks and the trough upstream the dam ($x \backsimeq 60\, [m]$), probably due to excessive numerical diffusion in the interpolation. Some improvement is instead observed with the ALE algorithm, which also gives a much sharper capturing of the bore.

A second test is performed to provide qualitatively evidence that r-adaptation could handle complex flows. It consist in the simultaneous break of two sym-

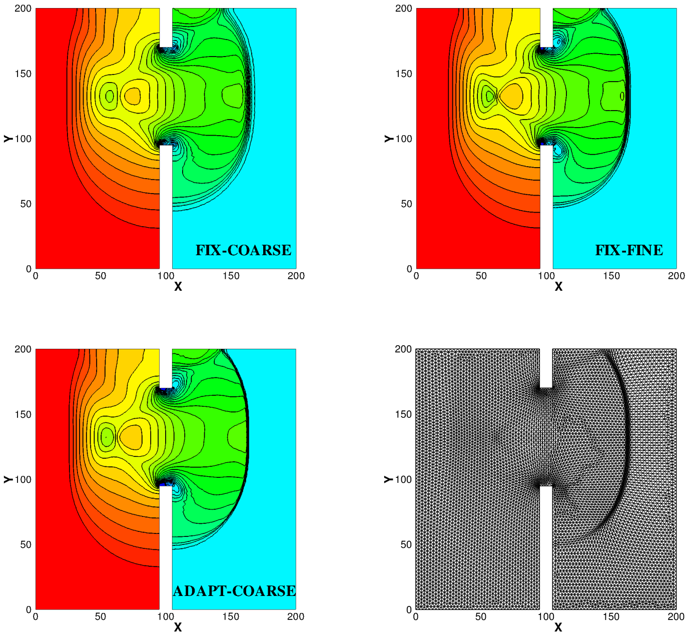

Figure 3.21: Asymmetric dam-break computed with RD scheme. 30 equispaced contour lines for h and adapted mesh.

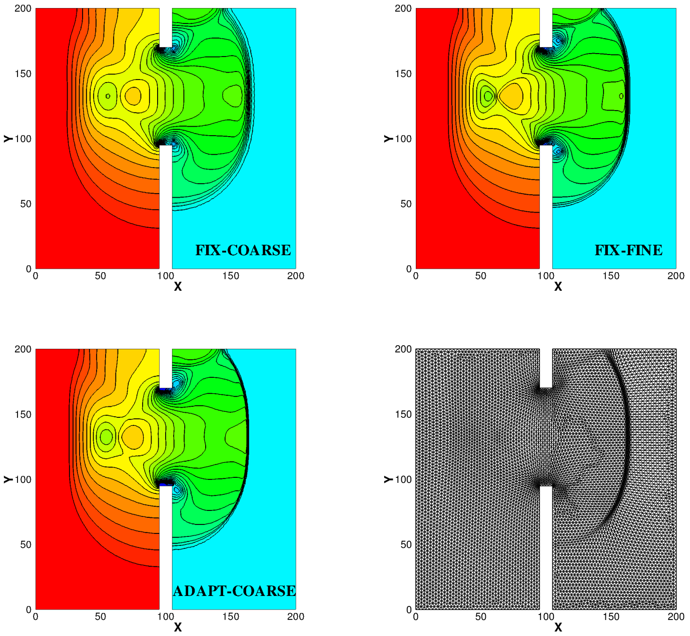

Figure 3.22: Asymmetric dam-break computed with FV scheme. 30 equispaced contour lines for h and adapted mesh.

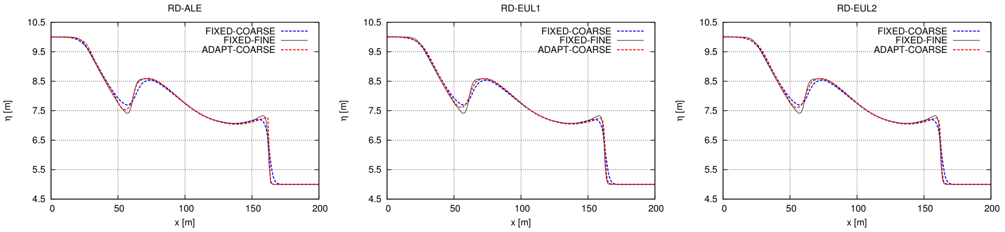

Figure 3.23: Asymmetric dam-break computed with RD scheme. Solution along the straight line at $y = 132.5$ for the different coupling. Left: ALE. Middle: EUL1. Right EUL2.

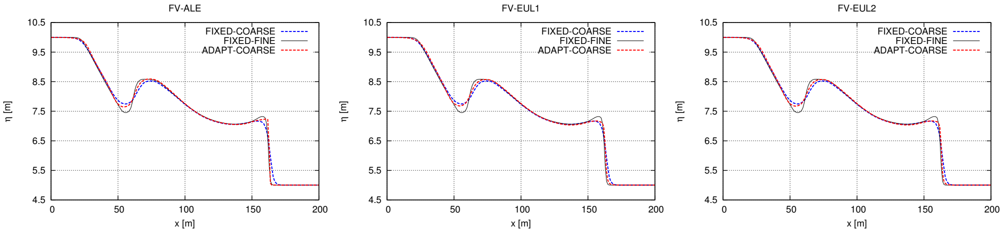

Figure 3.24: Asymmetric dam-break computed with FV scheme. Solution along the straight line at $y = 132.5$ for the different coupling. Left: ALE. Middle: EUL1. Right EUL2.

metric dams. The solution and the resulting grids are shown in Fig. (3.25). We observe *the mesh refining to capture sharp bore interaction and vortex formation.*

3.7.2 Small perturbation of a lake at rest

We consider the classical test of a small perturbation over an elliptic exponential hump (se e.g. Seaid [2004]; Ricchiuto [2015] 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 3.26 and 3.27 (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 3.3 the CPU times of all the schemes, and the water height along the line at y=0.5 on figures 3.28 and 3.29. For clarity, only the

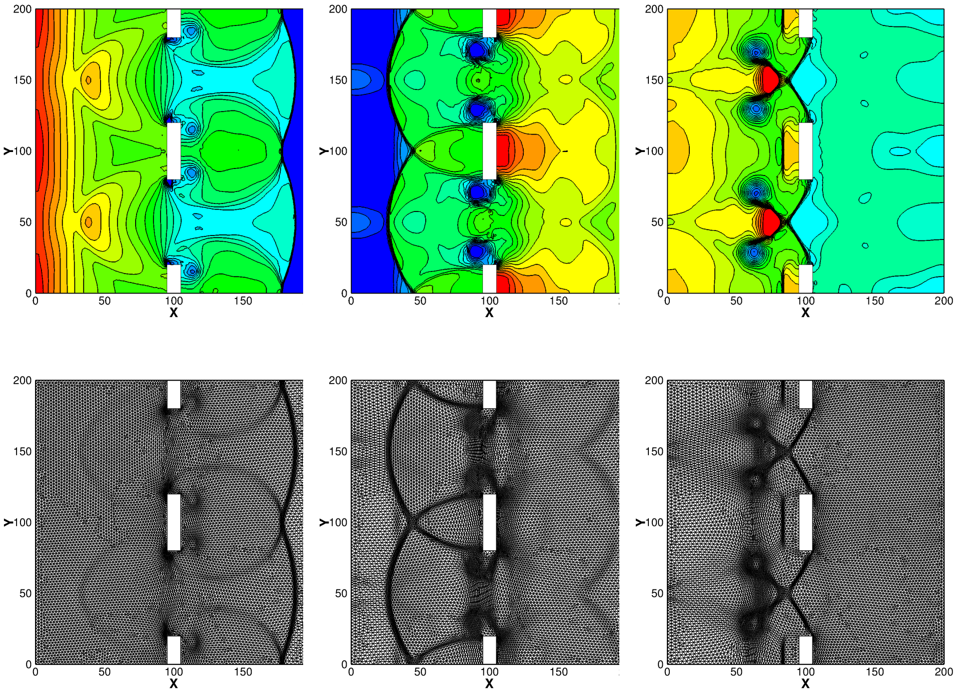

Figure 3.25: Double symmetric dam break, $t = 10$, $t = 35$, $t = 50$. Solution iso-lines and mesh computed with adaptive RD scheme.

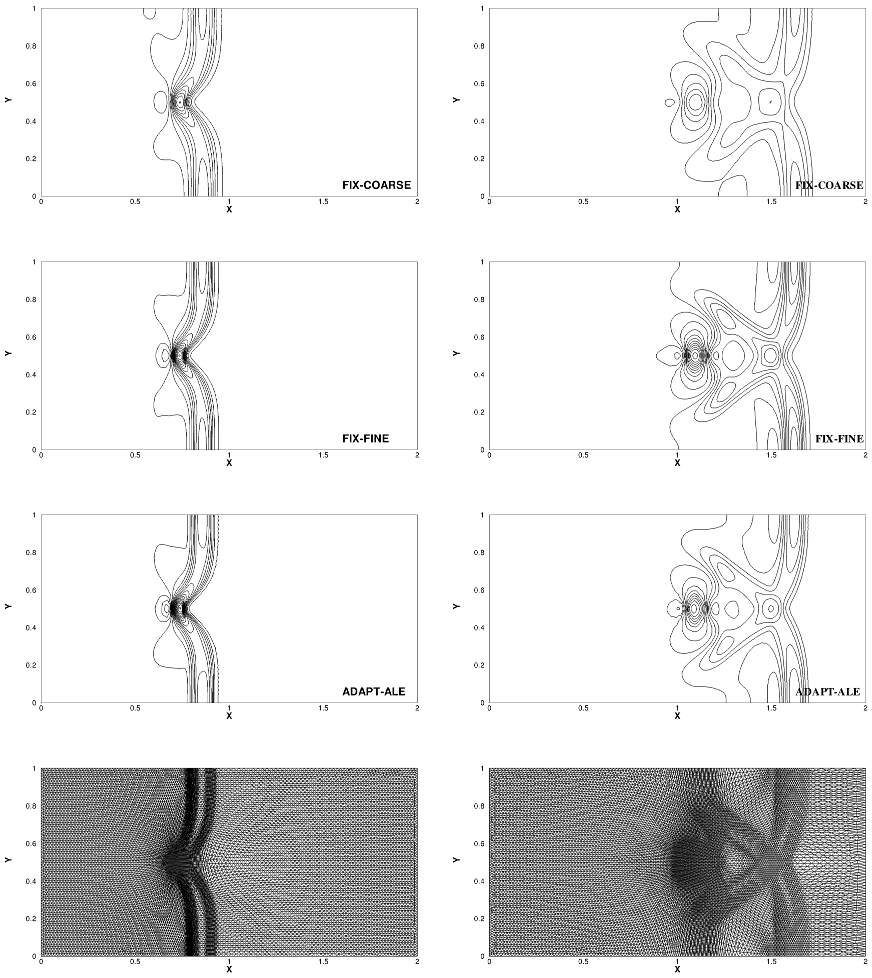

Figure 3.26: Small perturbation of a lake at rest (RD scheme). Solution isolines at $t = 0.24$, $t = 0.48$ are shown for fixed grid and adaptive computations. Top: fixed coarse grid. Middle: fixed fine grid. Bottom: adaptive ALE scheme.

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 3.30 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.

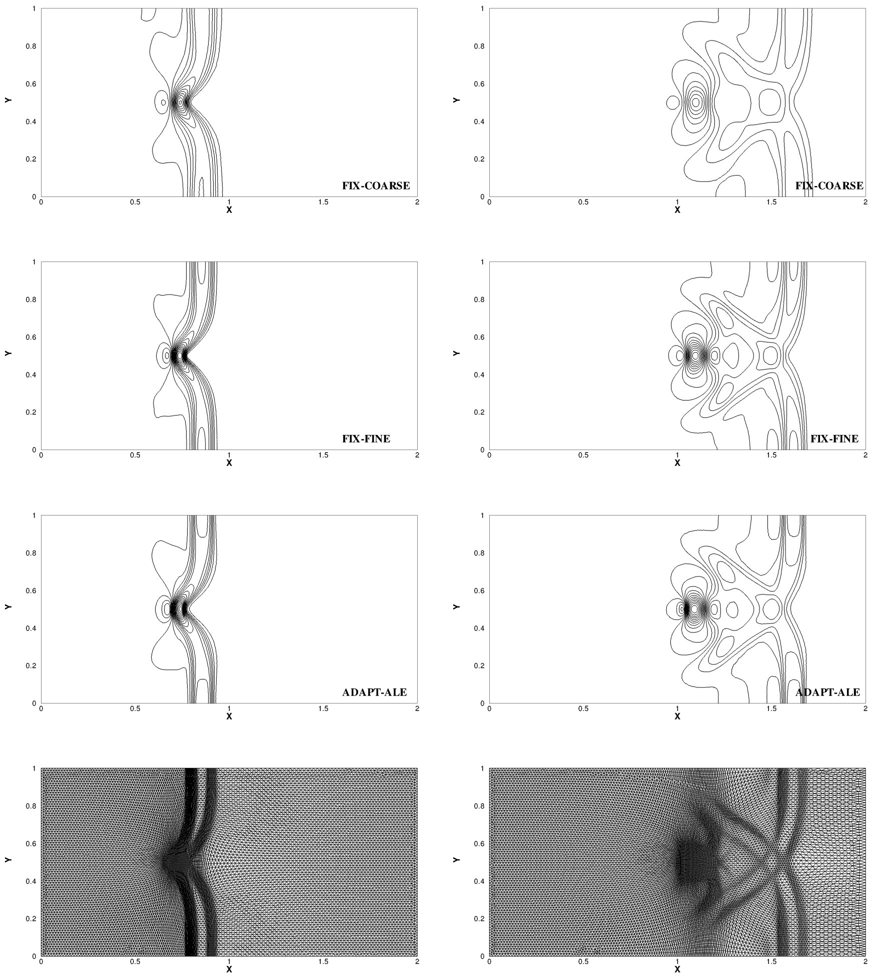

Figure 3.27: Small perturbation of a lake at rest (FV scheme). Solution isolines for $t = 0.24$, $t = 0.48$ are shown for fixed grid and adaptive computations. Top: fixed coarse grid. Middle: fixed fine grid. Bottom: adaptive ALE scheme.

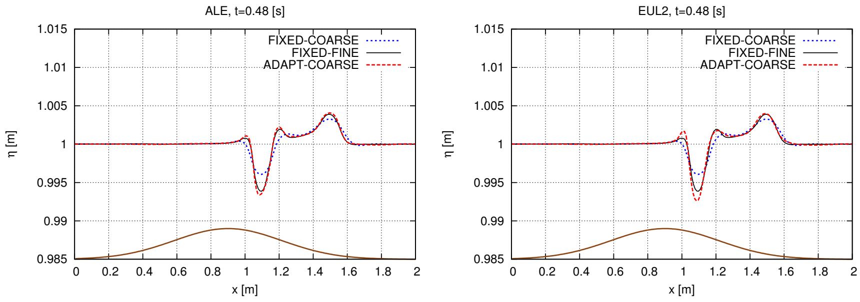

Figure 3.28: Small perturbation of a lake at rest (RD scheme). Solution at $t = 0.48$ along line $y = 0.5$. Left: ALE. Right: EUL2.

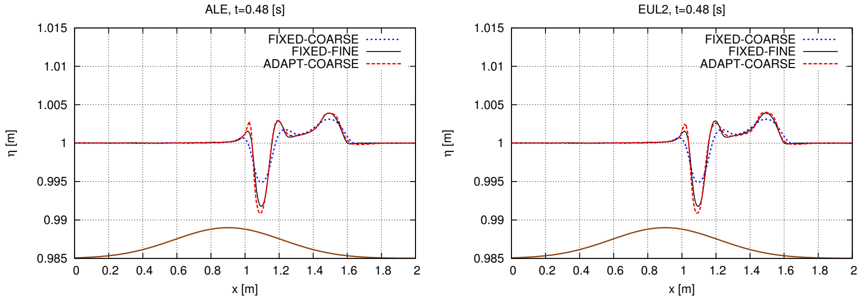

Figure 3.29: Small perturbation of a lake at rest (FV scheme). Solution at $t = 0.48$ along line $y = 0.5$. Left: ALE. Right: EUL2.

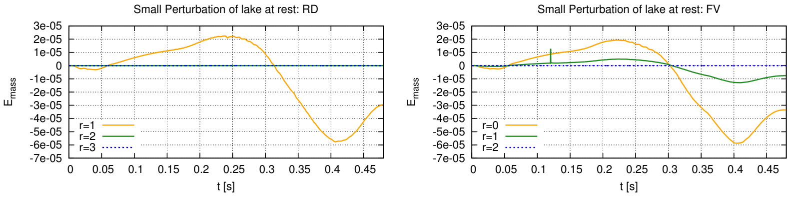

Figure 3.30: Small perturbation of a lake at rest. Dimensionless mass error for different quadrature formula of the bathymetry integral.

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 3.3: Small perturbation of a lake at rest. CPU times.

3.7.3 Runup on a conical island

This is a very classical benchmark for tsunami simulation models. It aims at reproducing some of the experiments of Briggs *et al.* [1995]. We refer to the above reference, and to Ricchiuto [2015]; Delis *et al.* [2008] 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 3.32), 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 3.33. 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 3.34 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 3.4. 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 re-computation 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 3.35 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 gen-

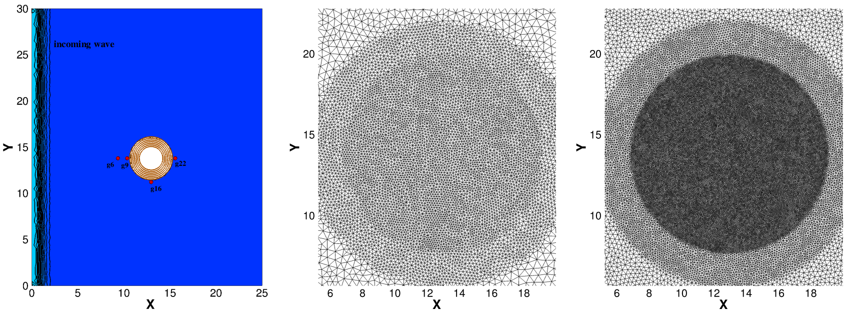

Figure 3.31: Conical island. Left: sketch of the computational domain with gauges. Middle: static coarse mesh topology. Right: static fine mesh topology.

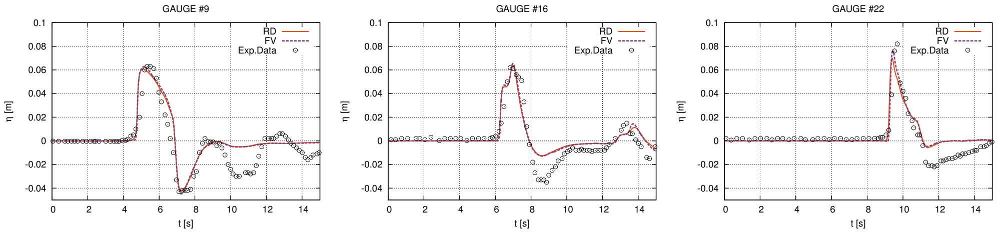

Figure 3.32: Conical Island computed with fixed fine mesh: total water height η signal registered at the gauges g9, g16, g22 and comparison with experimental data.

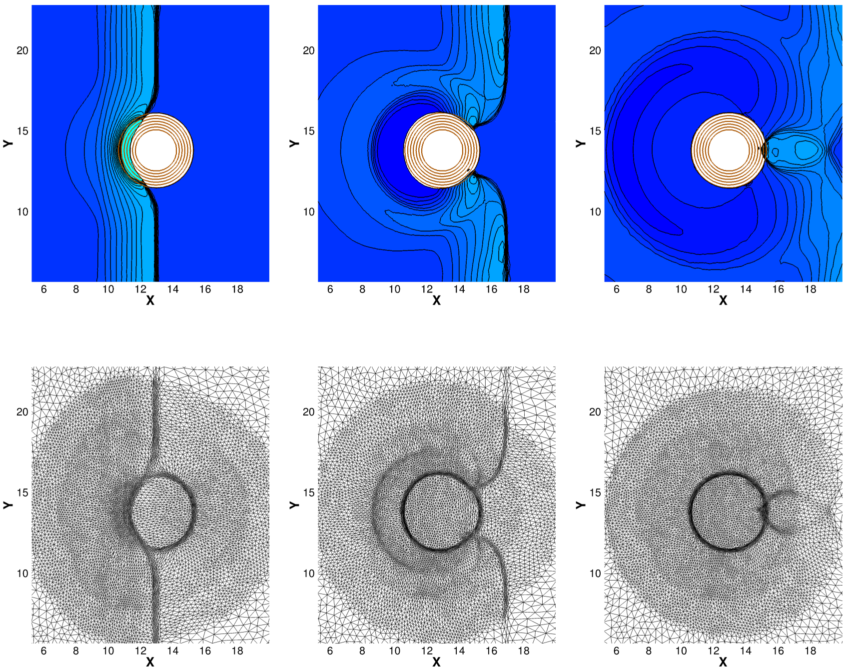

Figure 3.33: Conical Island: contour lines for total water height η and adapted mesh at different time instants, $t = 6.0, 8.0, 10.0\,[s]$

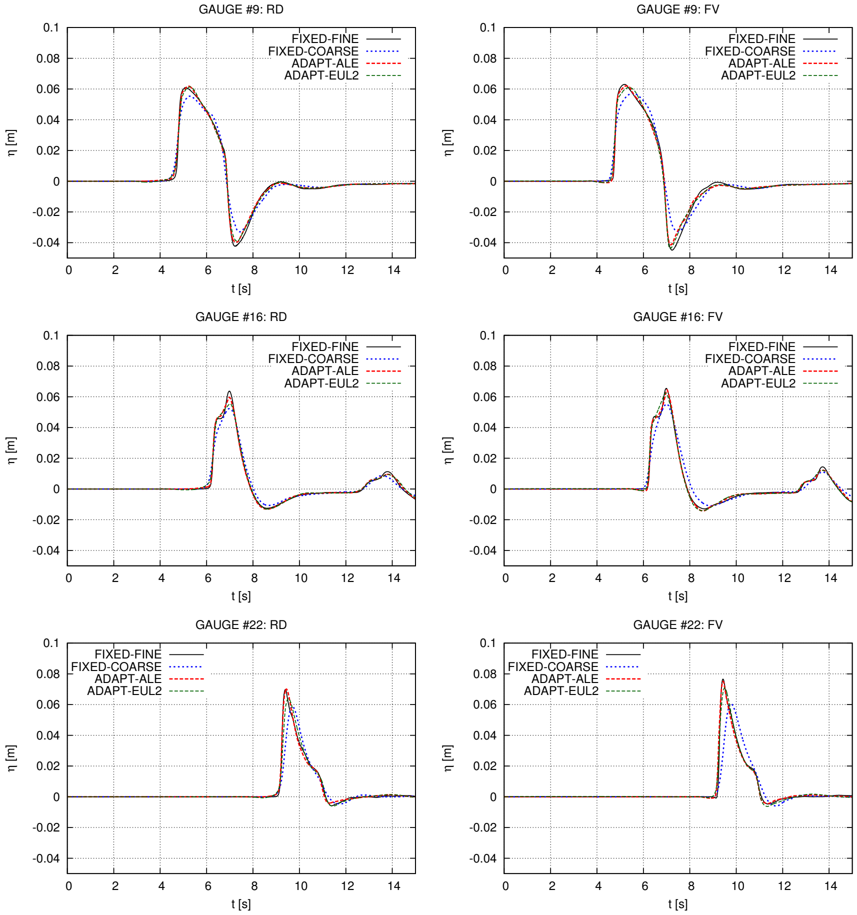

Figure 3.34: Conical Island: comparison between adaptive algorithms and fixed grid computations. Total water height η signal registered at the gauges g9, g16, g22. Left: RD scheme. Right: FV scheme.

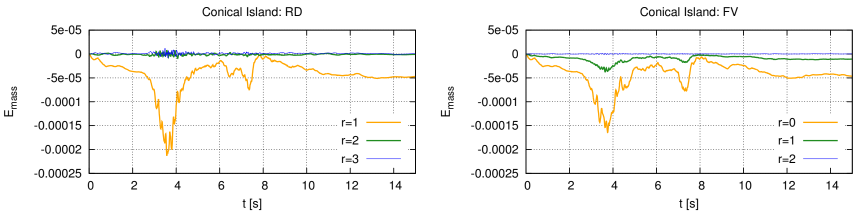

Figure 3.35: Conical island. Dimensionless mass error for different quadrature formula of the bathymetry integral.

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 3.4: Conical island. CPU times.

erated by the motion of dry nodes, allows to reduce the mass error practically to zero.

3.7.4 Monai valley benchmark

This test involves the tsunami runup over a complex 3D bathymetry, and is a standard test for tsunami simulation models Liu *et al.* [2008]. 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 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 [1993]. We have run this test on the grids reported on the right pictures of figure 3.36, statically adapted to the bathymetric variations Ricchiuto [2015]. 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 NOAA [1993]. 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 3.36, which reports plots relative to the instant of maximum runup. The

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 3.5: Okushiri experiment. CPU times.

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 r-adaptation, 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 Ricchiuto [2015], 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 3.37. A much more interesting quantity to look at is the runup plot, which is provided in the top row of figure 3.38. 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 3.38. 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 3.5.

Lastly, the evolution of the mass conservation error is reported on figure 3.39. Again we can see the improvement brought by the corrections proposed here.

3.7.5 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, involv-

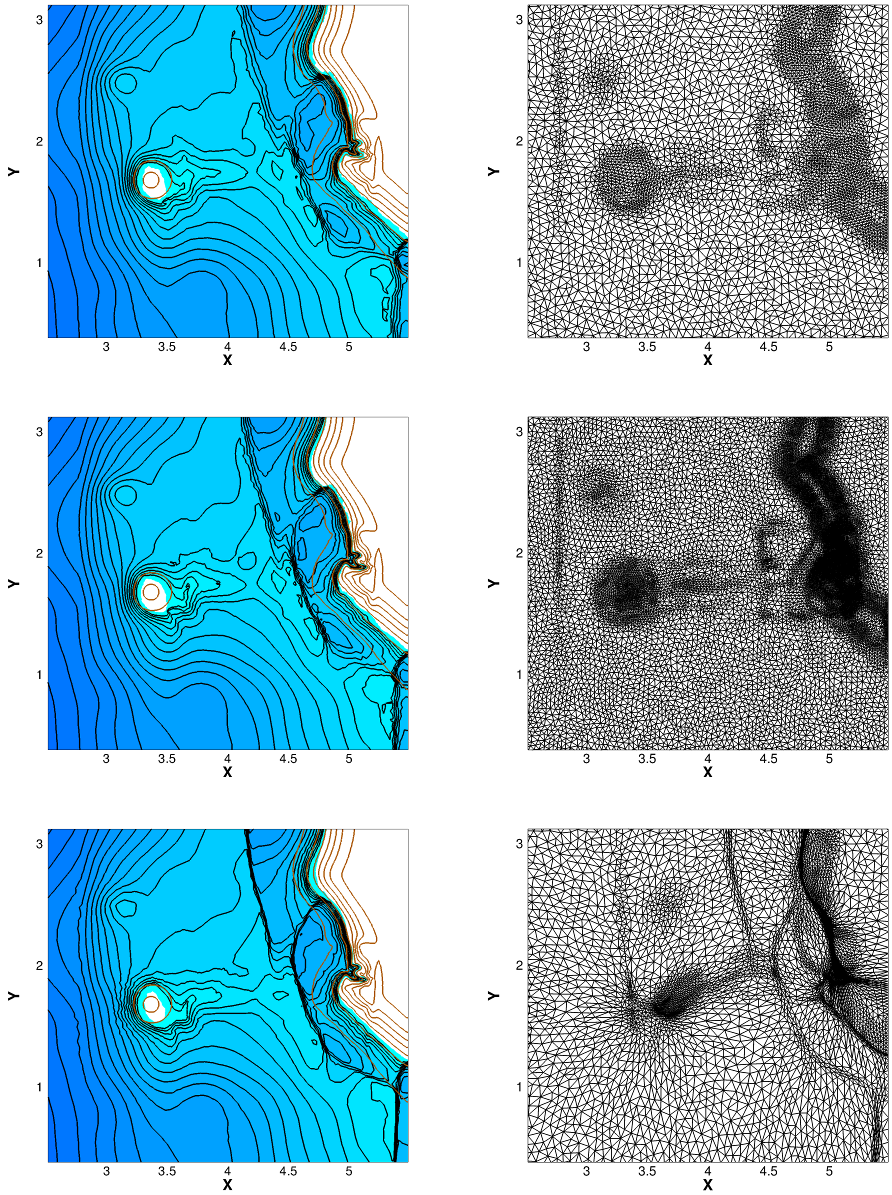

Figure 3.36: Okushiri experiment. Contour lines for h and mesh at $t = 16.5\,[s]$. Top: fixed coarse grid. Middle: fixed fine grid. Bottom: adaptive ALE scheme.

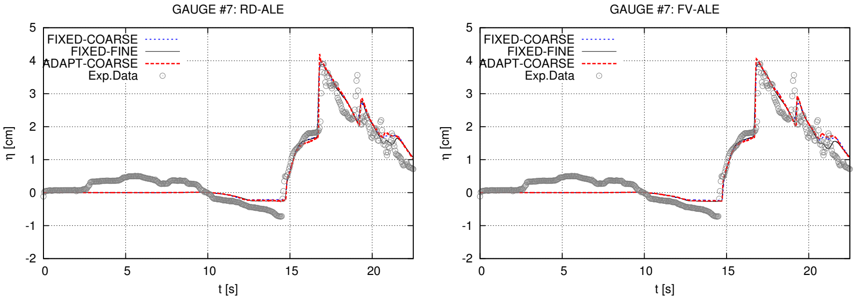

Figure 3.37: Okushiri experiment: total water height η signal registered at the gauges g7. Left: RD-ALE scheme. Right: FV-ALE scheme.

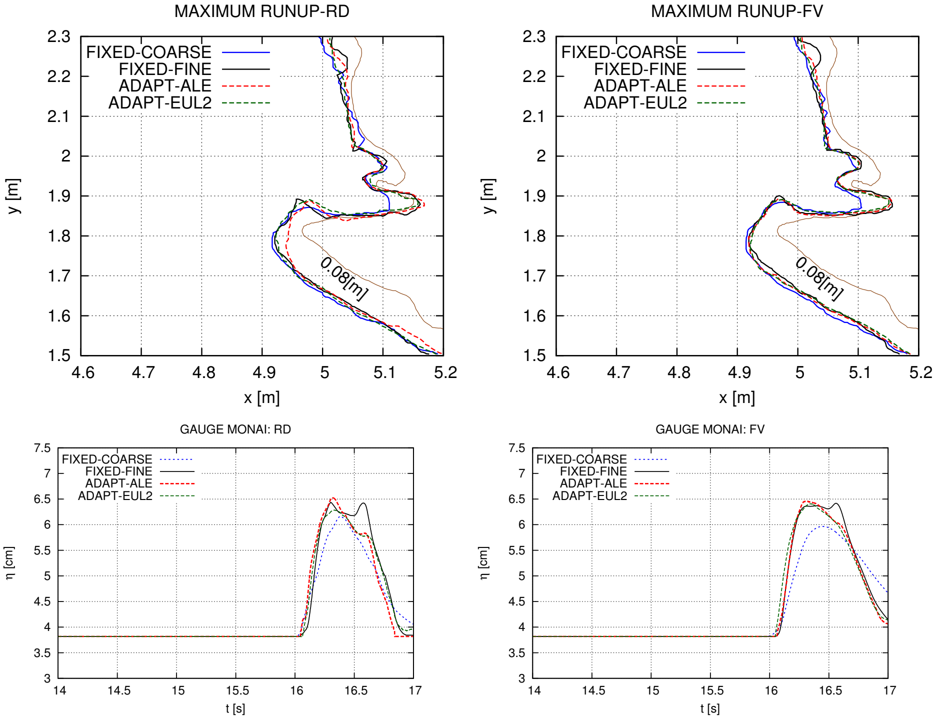

Figure 3.38: Okushiri experiment computed with adaptive ALE schemes. Top row: maximum runup for RD (left) and FV (right). Bottom row: total water height η signal registered at the gauge placed in the Monai valley. RD (left) and FV (right).

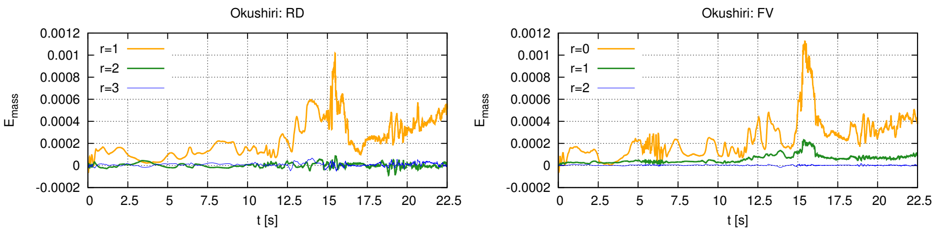

Figure 3.39: Okushiri experiment. Dimensionless mass error for different quadrature formula of the bathymetry integral.

ing 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 3.40. For this benchmark experimental time series of the water height are available 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 Lynett *et al.* [2010] (cf. also Roeber and Cheung [2012]; Kazolea *et al.* [2014]). We will compare flow velocity components in the exact location where an acoustic doppler velocimetry (ADV3) was installed. Two uniform meshes are used. The mesh 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 chosen $h_K = 0.2\,[m]$. For this test case we used the following MMPDE parameters: $\alpha = \delta = 40$ and $\beta = \gamma = 0.075$.

Figure 3.41 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 3.42 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 3.43 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.

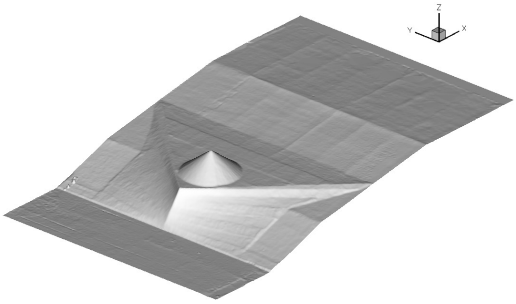

Figure 3.40: Solitary wave on a shelf: 3D visulization of the bathymetry, the scale of the $z-$axis is 5:1 with respect to x and y-axis

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 3.6: Solitary wave on a shelf. CPU times.

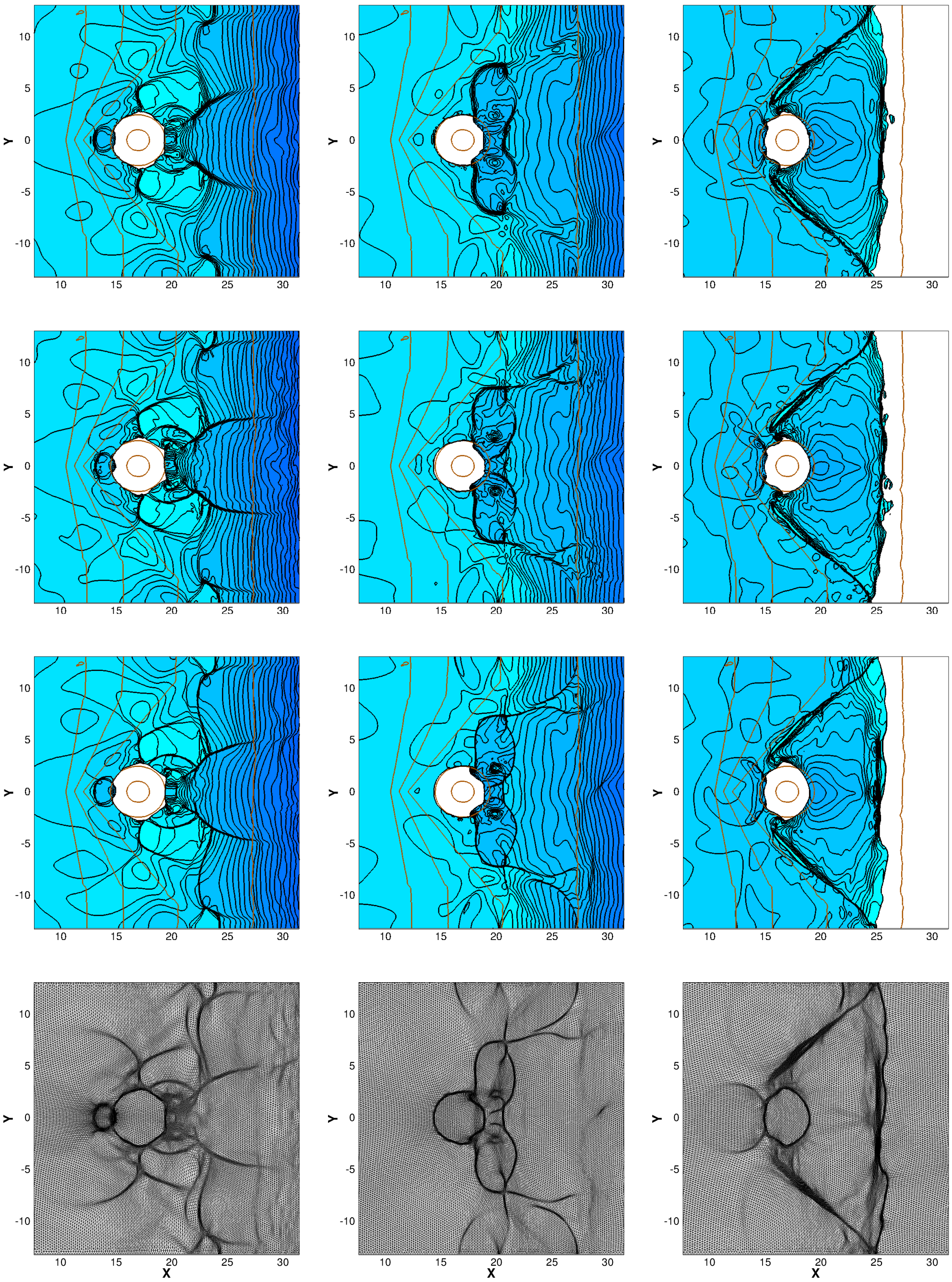

Figure 3.41: Solitary wave on a shelf (RD scheme). Solution isolines at $t = 12$, 17 and 25 $[s]$ are shown for fixed grid and adaptive computations. First row: fixed coarse grid. Second row: fixed fine grid. Third and fourth: adaptive ALE scheme.

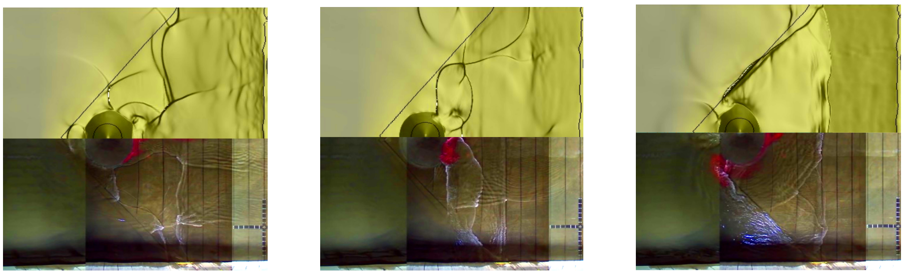

Figure 3.42: Solitary wave on a shelf (RD scheme). Snapshot at $t = 12$, 17 and 25 $[s]$ of the numerical solution and of the video from the wave tank experiment [$https://www.youtube.com/watch?v = I4uTHWBpaZg$].

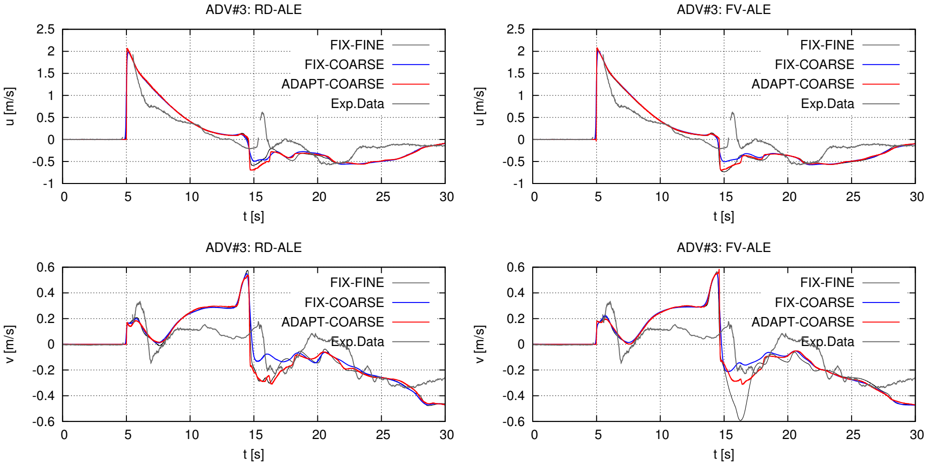

Figure 3.43: Solitary wave on a shelf: velocity components u, v registered at the gauge ADV3.

Chapter 4

ALE r-adaptive methods for the Shallaw Water equations on the sphere

The importance of setting the SWEs on the sphere stems from the fact that Earth curvature impacts wave propagation on large scale, which is particularly true for tsunami wave traveling for thousands of kilometers and crossing entire oceanic basins. In figure 4.1 the reader can find an illustrative example which explains the importance of Earth curvature in the tsunami dynamics. In this example we evaluate the computed free surface level for the 2011 Tohoku-Honsu tsunami as it is recorded by a real buoy placed in northern Japan (left picture). Two implementations of the SWEs are compared. In the right picture of figure 4.1 the red line refers to the Cartesian SWEs with Mercator coordinates and the blue line refers to the spherical SWEs in lat-lon coordinates. Only in the latter case we compute the correct arrival time of the leading tsunami wave. On the contrary the same numerical discretization applied to the Cartesian SWEs gives completely a erratic result.

We begin this chapter by presenting the SWEs in curvilinear coordinates. Then we extend the ALE transport formulas and the volume transformation statement (the GCL), developed in chapter 2 for Cartesian coordinates, to general curvilinear coordinates. Using these results, the SWEs can be written in a framework in which points moves arbitrary on a sphere. We discuss two novel implementations (FV and RD) of the resulting ALE-SWEs which are basically a development of the algorithms seen in chapter 2, see sections 2.10 and 2.11. A simplified moving mesh method will allow mesh adaptation on a portion of the sphere. Finally all the new code features (Eulerian RD/FV on the sphere, ALE-RD/FV on the sphere, moving mesh, ALE coupling with the moving mesh) are tested against standard benchmarks.

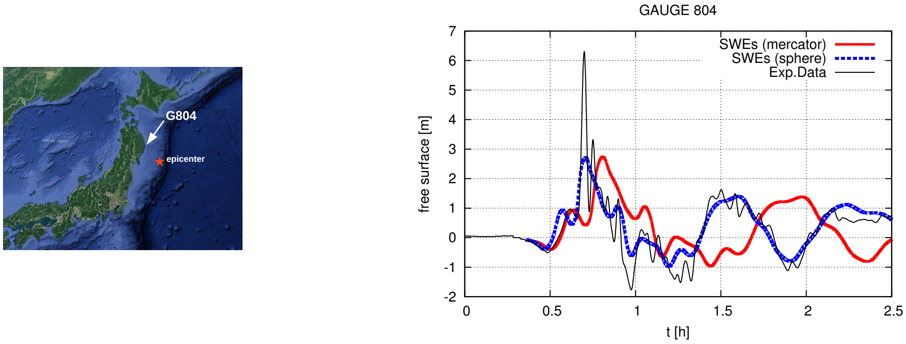

Figure 4.1: Example of a SWEs tsunami simulation of the 2011 Tohoku-Honsu event. In red) Cartesian SWEs with Mercator coordinates. In blue) spherical SWEs with lat-lon coordinates.

4.1 SWEs in curvilinear coordinates

In the community of ocean modeling and numerical weather prediction the first step is to set accurately the SWEs on a rotating sphere $\mathcal{S}^2$. There are many ways to do that, each one with its own pros and cons. One of the most popular approach Ritchie [1988]; Rossmanith *et al.* [2004]; Lauter *et al.* [2008] is to take advantage of a parametric representation of the sphere in a covariant vector basis. To allow a more complete understanding of this approach we have included, in appendix A, basic concepts from tensor analysis and differential geometry that are used here and in the following. At every point $P \in \mathcal{S}^2$ we fix an orthogonal but not orthonormal reference system with basis vectors $\{\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3\}$ together with local curvilinear coordinates $\{z^1, z^2, z^3\}$. We use this notation to avoid confusion with standard Cartesian coordinates $\{x^1, x^2, x^3\}$ adopted in all the previous chapters. A point $P \in \mathbb{R}^3$ which belongs to the sphere and is specified in Euclidean space by the vector of coordinates $\boldsymbol{x}$, can be expressed in curvilinear coordinates:

$$\boldsymbol{x} = \boldsymbol{g}_1 z^1 + \boldsymbol{g}_2 z^2 + \boldsymbol{g}_3 R$$

$\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ define the spherical surface where the undisturbed fluid lies. R is sphere radius and $\boldsymbol{g}_3$ is the axis "going out from the sphere". Along it, we measure the fluid depth $h(\boldsymbol{z})$ and the bathymetry/topography $b(\boldsymbol{z})$. The fluid free surface follows as $\eta = h + b$. We straightforwardly define the metric tensor $G_{ij} = \boldsymbol{g}_i \cdot \boldsymbol{g}_j$ and its determinant $G = \det \boldsymbol{G}$. Transforming the divergence operator of the SWEs (1.1), in curvilinear coordinates (formulas (A.9) and

(A.10)) we get:

$$\frac{\partial h}{\partial t} + \frac{1}{\sqrt{G}}\frac{\partial}{\partial z^j}\left(\sqrt{G}\, hu^j\right) = 0 \tag{4.1}$$

$$\frac{\partial hu^i}{\partial t} + \frac{1}{\sqrt{G}}\frac{\partial}{\partial z^j}\left(\sqrt{G}\, T^{ij}\right) + S^i = 0 \tag{4.2}$$

hu^i are the components of the discharge vector $h\boldsymbol{u} = hu^i \boldsymbol{g}_i$ and $T = huu^{ij} + \frac{1}{2}G^{ij}gh^2$ are the components of the flux tensor $h\boldsymbol{uu} = huu^{ij}\boldsymbol{g}_i\boldsymbol{g}_j$. We remark that the velocity components are not defined in a unitary basis. This, however, this allows to write the conservation laws in the compact form above. The source term reads

$$S^i = G^{ij} gh \frac{\partial b}{\partial z^j} + c_F hu^i + S^i_\gamma - S^i_c$$

Besides friction and bathymetry (first two addends), the source term includes also a geometrical force due to Earth curvature and the fictitious force associated to Earth rotation. The aforementioned geometrical force

$$S^i_\gamma = \Gamma^i_{jk} T^{jk}$$

comes from the transformation of the differential operator, see the appendix A for the definition of the Christoffel symbol Γ^i_{jk}. The rotation of the Earth is an important element in the dynamics of the atmosphere and ocean. Without loss of generality we let the axis of rotation be the Cartesian z-axis $\boldsymbol{e}_3$. If we solve the SWEs in the reference frame of the rotating Earth, we introduce in the momentum equations a pseudo-force of the form

$$S^i_c = f\varepsilon_{ij} hu^j$$

with $f = 2\Omega \sin\lambda$ being the Coriolis parameter, λ the latitude and ε the 2D Levi-Civita symbol. In all the experiments the Earth rotation rate is taken as $\Omega = 7.292 \times 10^{-5}\,[s^{-1}]$ unless it is differently specified.

At this point the SWEs have the following *non autonomous* vector form

$$\frac{\partial \mathsf{u}}{\partial t} + \frac{1}{\sqrt{G}}\frac{\partial \mathsf{F}^j}{\partial z^j} + \mathcal{S}(\boldsymbol{z}, \mathsf{u}) = 0, \tag{4.3}$$

$$\mathsf{u} = \begin{bmatrix} h \\ hu^i \end{bmatrix}, \; \mathsf{F}^j = \sqrt{G}\begin{bmatrix} hu^j \\ T^{ij} \end{bmatrix}, \tag{4.4}$$

$$\mathcal{S} = \underbrace{\begin{bmatrix} 0 \\ G^{ij} gh \frac{\partial b}{\partial x^j} \end{bmatrix}}_{\mathcal{S}^b} + \underbrace{\begin{bmatrix} 0 \\ c_F hu^i \end{bmatrix}}_{\mathcal{S}^f} + \underbrace{\begin{bmatrix} 0 \\ \Gamma^i_{jk} T^{jk} \end{bmatrix}}_{\mathcal{S}^\gamma} - \underbrace{\begin{bmatrix} 0 \\ f\varepsilon_{ij} hu^j \end{bmatrix}}_{\mathcal{S}^c} \tag{4.5}$$

Equations (4.3)(4.4)(4.5) is a non-homogeneous hyperbolic system of PDEs. Following LeVeque [1997], we remark that the flux vector depends explicitly on the position making the system non-autonomous. If one multiplies (4.3) by $\sqrt{G}$ a capacity coefficient appears in the time derivative. For these reason, some authors speak about a *capacity non autonomous form* of the SWEs on the sphere. The flux Jacobian $\mathsf{K}(\boldsymbol{\xi},\mathsf{u}) = \frac{\partial \mathsf{F}^j}{\partial \mathsf{u}}\xi^j$ is

$$\mathsf{K}(\boldsymbol{\xi},\mathsf{u}) = \sqrt{G}\begin{pmatrix} 0 & \xi^1 & \xi^2 \\ G^{11}gh\xi^1 - u\boldsymbol{u}\cdot\boldsymbol{\xi} & \boldsymbol{u}\cdot\boldsymbol{\xi} + u\xi^1 & u\xi^2 \\ G^{22}gh\xi^2 - v\boldsymbol{u}\cdot\boldsymbol{\xi} & v\xi^1 & \boldsymbol{u}\cdot\boldsymbol{\xi} + v\xi^2 \end{pmatrix} \tag{4.6}$$

for any vector $\boldsymbol{\xi} = \xi^j\boldsymbol{g}_j$. Note that the scalar product that appears in the expression above is not the actual inner product on the curved manifold but simply $\boldsymbol{u}\cdot\boldsymbol{\xi} = u^j\xi^j$. It become the same if the basis is orthonormal. The Jacobian (4.6) admits a full set of real eigenvalues and linearly independent eigenvectors. The eigenvalues are

$$\lambda_{1,3}(\mathsf{u},\boldsymbol{\xi}) = \sqrt{G}\left(\boldsymbol{u}\cdot\boldsymbol{\xi} \pm \sqrt{gh}\|\boldsymbol{\xi}\|_c\right), \quad \lambda_2(\mathsf{u},\xi) = \sqrt{G}\boldsymbol{u}\cdot\boldsymbol{\xi}$$

where the norm is computed with respect to the inverse metric $\|\boldsymbol{\xi}\|_c = \sqrt{\xi^i G^{ij}\xi^j}$.

At this point we remark that the eigenstructure changes with respect to the Cartesian case. Particularly important is the scaling factor G^{ij} that multiplies the acoustic part of the Jacobian, giving it the correct dimension. The Jacobian at rest on the sphere reads (cf. expression **??**):

$$\mathsf{A}(\boldsymbol{\xi},h) = \begin{pmatrix} 1 & \sqrt{G}\xi^1 & \sqrt{G}\xi^2 \\ \sqrt{G}G^{11}gh\,\xi^1 & 0 & 0 \\ \sqrt{G}G^{22}gh\,\xi^2 & 0 & 0 \end{pmatrix} \tag{4.7}$$

We mention that, starting from the work of Pons *et al.* [1998], many authors that implement FV in covariant basis, prefer to go back to an orthonormal framework (e.g. when solving the Riemann problem), namely $G^{ij} = \delta^{ij}$ and $\sqrt{G} = 1$, we refer to Rossmanith *et al.* [2004] for the details of this approach which requires a double projection: first to obtain the orthonormal Riemann problem, then to go back to covariant fluxes. Here we propose to use directly (4.6) which does not require these projections. After the resolution of the standard orthonormal Riemann problem, the fluxes are projected back in the covariant basis to update the solution of the SWEs in covariant formulation.

4.2 ALE in curvilinear coordinates

To extend the SWEs in curvilinear coordinates and in ALE framework, we upgrade the kinematics relationships seen in chapter 2. The main results, namely transport formulas between Eulerian, Lagrangian and ALE and the volume/area conservation statements are recovered here for general curvilinear

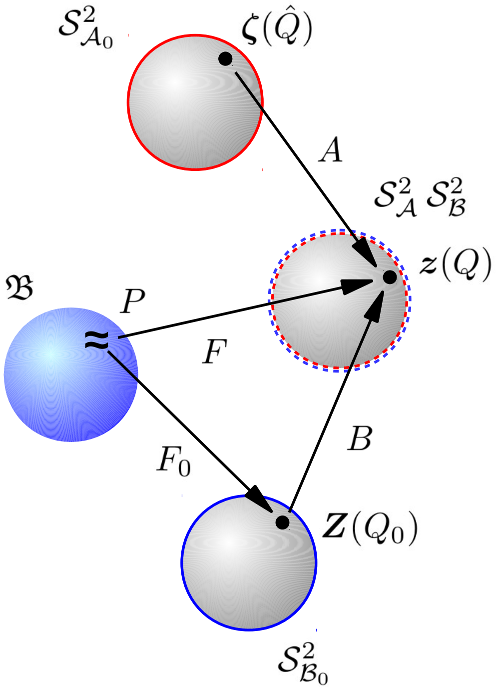

Figure 4.2: A sketch of Eulerian, Lagrangian and ALE configurations on a sphere.

coordinates. Up to the author's knowledge, the literature on the ALE balance law on manifold is not very vast. The interested reader can find a detailed treatment in Savidis *et al.* [2008].

4.2.1 Basic kinematics

We start this section setting the problem in the general case: a material fluid $\mathfrak{B}$ is a set of fluid columns ("particles") that can be put in bijective correspondence with the points of a region $\mathcal{B}$ belonging to a differentiable Riemannian manifold. We keep the same notation of chapter 2, the particles are $P \in \mathfrak{B}$ and the points $Q \in \mathcal{B}$. We assign a local covariant vector basis $\{\boldsymbol{g}_1, \boldsymbol{g}_2\}$ and local curvilinear coordinates $\{z^1, z^2\}$ to the tangent plane $T_Q\mathcal{B}$ defined at each point Q. A point position is identified by the vector $\boldsymbol{z}$ which can be expressed in the local curvilinear coordinate system

$$\boldsymbol{z}(Q) = z^1\boldsymbol{g}_1 + z^2\boldsymbol{g}_2 \equiv z^i\boldsymbol{g}_i$$

The covariant vector basis is sometimes expressed as $\{\boldsymbol{g}_i = \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} z^i}\} \in T_Q\mathcal{B}$. The initial or *reference* configuration is denoted by $\mathcal{B}_0$ and the position of the point

Q_0 in the reference configuration is denoted by capital letters $\boldsymbol{Z}(Q_0)$. For this configuration we assign a local covariant vector basis $\{\boldsymbol{G}_1, \boldsymbol{G}_2\}$ and local chart $\{Z^1, Z^2\}$ to the tangent plane $T_{Q_0}\mathcal{B}_0$

$$\boldsymbol{Z}(Q_0) = Z^1\boldsymbol{G}_1 + Z^2\boldsymbol{G}_2$$

As $\mathcal{B}$ moves, we assume the existence of a function that transform the point positions from the reference configuration to the actual one. This function is always called B, as in chapter 2:

$$B : \mathcal{B}_0 \to \mathcal{B} \quad \boldsymbol{z} = B(\boldsymbol{Z}, t)$$

Even if in principle the reference and the actual configurations could belong to different manifolds, points for us will be always attached to the sphere, that is $\mathcal{B} = \mathcal{S}^2_{\mathcal{B}}$ and $\mathcal{B}_0 = \mathcal{S}^2_{\mathcal{B}_0}$ as sketched in figure 4.2 and

$$B : \mathcal{S}^2_{\mathcal{B}_0} \to \mathcal{S}^2_{\mathcal{B}}$$

The function B is assumed to be continuously differentiable, thus a smooth Jacobian of transformation can be defined

$$\boldsymbol{J}_B = \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{Z}} \tag{4.8}$$

In order to avoid particle collision we add the smoothness requirement $\exists \boldsymbol{J}_B^{-1}$, $J_B = \det \boldsymbol{J}_B > 0$. The relationship between Lagrangian and Eulerian time derivatives for a scalar ϕ and a vector $\boldsymbol{v}$ modifies as:

$$\begin{aligned}
\frac{d\phi}{dt} &= \left.\frac{\partial \phi}{\partial t}\right|_Z \circ B^{-1} = \left.\frac{\partial \phi}{\partial t}\right|_z + \frac{\partial \phi^i}{\partial z^j} u^j \\
\frac{dv^i\boldsymbol{g}_i}{dt} &= \left.\frac{\partial v^i\boldsymbol{g}_i}{\partial t}\right|_Z \circ B^{-1} = \left(\left.\frac{\partial v^i}{\partial t}\right|_z + \frac{\partial v^i}{\partial z^j} u^j + \Gamma^i_{jk} u^j v^k \right) \boldsymbol{g}_i
\end{aligned}$$

due to the presence of Christoffel symbols in the gradient definition, (see (A.11)). The kinematic relation (2.5) expressing the derivative of a material volume modifies as follows in curvilinear coordinates

$$\left.\frac{\partial}{\partial t}\right|_Z \left(\sqrt{G} J_B\right) = J_B \frac{\partial}{\partial z^j}\left(\sqrt{G} u^j\right) = 0 \tag{4.9}$$

or, more compactly $\partial_t|_Z (\sqrt{G} J_B) - J_B \sqrt{G} \nabla \cdot \boldsymbol{u} = 0$. For us (4.9) will be the time derivative of a material volume on the sphere. This result appears here in a slightly different form with respect to Savidis *et al.* [2008]. Since we prefer the above form for future developments we prefer to give its proof.

Proof of (4.9). We extend a classical proof, see for example the lecture notes of Mantegazza [2012], to curvilinear coordinates. From the Jacobian definition (4.8) we inherit $dz^m = J_B dZ^m$ where we recall $dz^m = dz^1 dz^2$. Passing to the transformation of area differentials on manifolds $\mathcal{B}_0 \to \mathcal{B}$, the statement must be rewritten. We consider the definition of area differentials given in appendix A. In the actual configuration the infinitesimal area on the tangent plane writes, see eq. (A.15)(A.16)

$$da = |d\boldsymbol{z}_1 \times d\boldsymbol{z}_2| = \sqrt{G} dz^1 dz^2 \tag{4.10}$$

Taking the material derivative of (4.10):

$$\begin{aligned} \left.\frac{\partial}{\partial t}\right|_Z \left(\sqrt{G} dz^m\right) &= \left.\frac{\partial \sqrt{G}}{\partial t}\right|_Z dz^m + d\left.\frac{\partial z^1}{\partial t}\right|_Z \sqrt{G} dz^2 + d\left.\frac{\partial z^2}{\partial t}\right|_Z \sqrt{G} dz^1 \\ &= \left.\frac{\partial \sqrt{G}}{\partial t}\right|_X dz^m + \frac{\partial u^j}{\partial z^j} \sqrt{G} dz^m \end{aligned}$$

$\sqrt{G} = \sqrt{G(\boldsymbol{z}(t))}$ and we can use chain rule

$$\begin{aligned} \left.\frac{\partial}{\partial t}\right|_Z \left(\sqrt{G} dz^m\right) &= \left.\frac{\partial z^j}{\partial t}\right|_Z \frac{\partial \sqrt{G}}{\partial z^j} dz^m + \frac{\partial u^j}{\partial z^j} \sqrt{G} dz^m \\ &= u^j \frac{\partial \sqrt{G}}{\partial z^j} dz^m + \frac{\partial u^j}{\partial z^j} \sqrt{G} dz^m = \frac{\partial}{\partial z^j} \left(\sqrt{G} u^j\right) dz^m \\ &= \frac{\partial}{\partial z^j} \left(\sqrt{G} u^j\right) J_B dZ^m \end{aligned}$$

Developing the left-hand side with $\sqrt{G} dz^m = \sqrt{G} J_B dZ^m$ we get:

$$\left.\frac{\partial}{\partial t}\right|_Z \left(\sqrt{G} J_B\right) = J_B \frac{\partial}{\partial z^j} \left(\sqrt{G} u^j\right)$$

which is exactly (4.9). ☐

4.2.2 ALE kinematics in curvilinear coordinates

Mesh movement on the sphere will be accomplished through an arbitrary configuration $\mathcal{A}$ called ALE which we assume it belongs to a differentiable manifold. As for the material configuration, $\mathcal{A}$ is composed of points Q. Its initial or reference configuration is $\mathcal{A}_0$, composed instead of points $\hat{Q}$. A local coordinate system $\{\zeta^1, \zeta^2\}$ can be assigned in the neighborhood of every $\hat{Q}$ such that the point's position writes locally

$$\boldsymbol{\zeta}(\hat{Q}) = \zeta^i \boldsymbol{\gamma}_i$$

where $\{\boldsymbol{\gamma}_i = \frac{\boldsymbol{\partial}}{\boldsymbol{\partial} \zeta^i}\} \in T_{\hat{Q}} \mathcal{A}_0$ is the covariant vector basis. As in chapter 2 lower case letters label the actual configuration and Greek letters are used for the reference ALE configuration. If, at each time value is associated a unique configuration, then the arbitrary configuration will be in motion:

$$A : \mathcal{A}_0 \to \mathcal{A} \quad \boldsymbol{z} = A(\boldsymbol{\zeta}, t)$$

Although the ALE configuration changes with time dragging unsteady mesh movement, it will always belong to the sphere, that is $\mathcal{A} = \mathcal{S}^2_{\mathcal{A}}$ and $\mathcal{A}_0 = \mathcal{S}^2_{\mathcal{A}_0}$. As sketched in figure 4.2 we have:

$$A : \mathcal{S}^2_{\mathcal{A}_0} \to \mathcal{S}^2_{\mathcal{A}}$$

Using the definition of ALE time derivative $\frac{\partial}{\partial t}\big|_{\zeta}$, the ALE/mesh velocity of the point $\hat{Q}$ can be introduced in the current vector basis $\boldsymbol{\sigma}(\hat{Q}) = \sigma^j \boldsymbol{g}_j$

$$\sigma^j = \left.\frac{\partial z^j}{\partial t}\right|_{\zeta}$$

One can then use the chain rule to recover the relationship between Eulerian and ALE time derivative for scalars ϕ and vectors $\boldsymbol{v}$, see (2.10) in Chapter 2:

$$\left.\frac{\partial \phi}{\partial t}\right|_{\zeta} = \left.\frac{\partial \phi_0}{\partial t}\right|_{\zeta} \circ A^{-1} = \left.\frac{\partial \phi}{\partial t}\right|_{z} + \frac{\partial \phi}{\partial z^j}\left.\frac{\partial z^j}{\partial t}\right|_{\zeta} = \left.\frac{\partial \phi}{\partial t}\right|_{z} + \frac{\partial \phi}{\partial z^j}\sigma^j \qquad (4.11)$$

$$\left.\frac{\partial v^i}{\partial t}\right|_{\zeta} = \left.\frac{\partial v^i_0}{\partial t}\right|_{\zeta} \circ A^{-1} = \left.\frac{\partial v^i}{\partial t}\right|_{z} + \frac{\partial v^i}{\partial z^j}\sigma^j \qquad (4.12)$$

As in Chapter 2, a useful relationship emerges between the Lagrangian time derivative and the ALE one (4.11) and (4.12):

$$\frac{d\phi}{dt} = \left.\frac{\partial \phi}{\partial t}\right|_{\zeta} + \frac{\partial \phi}{\partial z^j}\left(u^j - \sigma^j\right) \qquad (4.13)$$

$$\frac{dv^i \boldsymbol{g}_i}{dt} = \left(\left.\frac{\partial v^i}{\partial t}\right|_{\zeta} + \frac{\partial v^i}{\partial z^j}\left(u^j - \sigma^j\right) + \Gamma^i_{jk} u^j v^k\right)\boldsymbol{g}_i \qquad (4.14)$$

We take the ALE time derivative of an infinitesimal area on the sphere. This is called geometric conservation law (GCL):

$$\left.\frac{\partial}{\partial t}\right|_{\zeta}\left(\sqrt{G} J_A\right) = J_A \frac{\partial}{\partial z^j}\left(\sqrt{G}\sigma^j\right) \qquad (4.15)$$

or, more compactly $\partial_t|_{\zeta}\,(\sqrt{G}J_A) - J_A\sqrt{G}\nabla\cdot\boldsymbol{\sigma} = 0$.

Proof. We don't repeat the proof that follows the one done in the previous paragraph. In fact, this is achieved replacing body reference quantities

with arbitrary reference ones, $\sqrt{G_0}$ with $\sqrt{\hat{G}}$, J_B with J_A and dZ with $d\zeta$.
□

The GCL (4.15) is a geometrical relation between the ALE Jacobian and the ALE velocity. One can check that the integral of (4.15) over the whole sphere states the conservation of the area of the ALE domain. In the context of the numerical solution of ALE-PDEs on the sphere, this corresponds to preserve the total area of the mesh during the simulation. For this reason in section 4.5 we will discuss how to preserve also its discrete counterpart.

4.3 ALE-SWEs in curvilinear coordinates

We consider now the field equations or balance laws for the water column in curvilinear coordinates:

$$\left.\frac{\partial \sqrt{G} J_B h}{\partial t}\right|_Z = 0 \quad \text{and} \quad \left.\frac{\partial \sqrt{G} J_B h u^i \boldsymbol{g}_i}{\partial t}\right|_Z = \sqrt{G} J_B \boldsymbol{F}^i \tag{4.16}$$

where $\boldsymbol{F}^i$ represents the forces acting on the water column. We transform them from a Lagrangian description to the ALE framework. To do this, we can employ the transport formulas (4.13),(4.14), the area transformation relationships, (4.9),(4.15), all discussed in the previous sections. In order:

1. we transform the time derivative in (4.16) using (4.13) and (4.14)
2. we substitute (4.9)
3. we multiply the resulting expression by J_A
4. finally we add (4.15) multiplied in turn by h or hu^i

As a result, the SWEs in curvilinear coordinates in the ALE framework write:

$$\begin{aligned}
\left.\frac{\partial \sqrt{G} J_A h}{\partial t}\right|_\zeta + J_A \frac{\partial}{\partial z^j}\left(h u^j - \sqrt{G} h \sigma^j\right) &= 0 &\quad (4.17)\\
\left.\frac{\partial \sqrt{G} J_A h u^i}{\partial t}\right|_\zeta + J_A \frac{\partial}{\partial z^j}\left(T^{ij} - \sqrt{G} h u^i \sigma^j\right) + \sqrt{G} J_A S^i &= 0 &\quad (4.18)
\end{aligned}$$

As done in previous chapter we benefit of the vector form to write them compactly:

$$\left.\frac{\partial}{\partial t}\right|_\zeta \left(\sqrt{G} J_A \mathsf{u}\right) + J_A \frac{\partial}{\partial z^j}\left(\mathsf{F}^j - \sqrt{G}\sigma^j \mathsf{u}\right) + \sqrt{G} J_A \mathcal{S}(\boldsymbol{z}, \mathsf{u}) = 0, \tag{4.19}$$

$$\mathsf{u} = \begin{bmatrix} h \\ h u^i \end{bmatrix}, \; \mathsf{F}^j = \sqrt{G} \begin{bmatrix} h u^j \\ T^{ij} \end{bmatrix}$$

$$\mathcal{S} = \underbrace{\begin{bmatrix} 0 \\ G^{ij} g h \frac{\partial b}{\partial x^j} \end{bmatrix}}_{\mathcal{S}^b} + \underbrace{\begin{bmatrix} 0 \\ c_F h u^i \end{bmatrix}}_{\mathcal{S}^f} + \underbrace{\begin{bmatrix} 0 \\ \Gamma^i_{jk} T^{jk} \end{bmatrix}}_{\mathcal{S}^\gamma} + \underbrace{\begin{bmatrix} 0 \\ f \varepsilon_{ij} h u^j \end{bmatrix}}_{\mathcal{S}^c} \tag{4.20}$$

SWEs are written for the water depth h and depth-integrated momentum $h\boldsymbol{u}$. So far we have seen that this form is the appropriate one to conserve, at a discrete level, total mass and momentum which remain for us the most important quantities to be conserved. We will show that our numerical approximations of (4.17) and (4.18) allow mass and momentum conservation on the sphere. We also mention that in geophysical applications it could convenient to manipulate the SWEs in order to have discrete conservation of more interesting quantities such as total energy or potential vorticity, see for example Ringler and Randall [2002]. This is left for future work.

In chapter 2 we have seen that this form is not particularly suitable to preserves the lake at rest on moving mesh. This is related to the fact that solving the lake at rest in a moving reference framework corresponds to evolve the ALE remap equation for the water depth h, to which it is associated a discretization error. To cure this problem we introduced the ALE-WB and ALE-PB form of the SWEs. The same can be done here. We write equation (4.11) for the bathymetric function (does not depends explicitly on time $b = b(\boldsymbol{z})$)

$$\left.\frac{\partial b}{\partial t}\right|_{\varsigma} = \frac{\partial b}{\partial z^j}\sigma^j$$

multiplying by $J_A\sqrt{G}$ and summing it to the (4.15) (multiplyed by b) we have the ALE remap in curvilinear coordinates

$$\left.\frac{\partial \sqrt{G}J_A b}{\partial t}\right|_{\varsigma} - J_A\frac{\partial}{\partial z^j}\left(\sqrt{G}b\sigma^j\right) = 0$$

If we add the ALE remap for b to the mass equation (4.17) we finally obtain the WB-ALE form

$$\left.\frac{\partial}{\partial t}\right|_{\varsigma}\left(\sqrt{G}J_A\mathsf{w}\right) + J_A\frac{\partial}{\partial z^j}\left(\mathsf{F}^j - \sqrt{G}\sigma^j\mathsf{w}\right) + \sqrt{G}J_A\mathcal{S}(\boldsymbol{z},\mathsf{u}) = 0, \qquad (4.21)$$

$$\mathsf{u} = \begin{bmatrix} \eta \\ hu^i \end{bmatrix}$$

4.4 Mesh and geometry in curvilinear coordinates

Points on the sphere are represented by a proper parametrization $\{z^1, z^2\}$ through the use of geometrical mappings. From now on, we will use a standard latitude longitude (lat-lon) parametrization for which the definitions of geometrical quantities that describes the sphere, including Jacobian, metric tensor and Christoffel symbols are given in appendix C. We recall that for the sphere the aforementioned quantities are available in simple analytical expression. The main drawback of this approach is that it introduces a singularity at the poles. Here we are not interested in the simulation of tsunamis that propagate all around the globe and we will simply circumvent this issue solving the SWEs far from the polar region: $(z^1, z^2) \in [-180°, 180°] \times [-75°, 75°]$ where the metric coefficients are well defined with lat-lon coordinates. We mention that this is a big limit if one is interested in global circulation model where it is urgent to accurately compute the solution at the poles. In this case the use of different grids such as the *cubed sphere* grid Ronchi *et al.* [1996], the *Yin-Yang* grid Kageyama and Sato [2004] or rotated lat-lon grids, allows to resolve the pole problem. The implementation of rotated lat-lon grids, see e.g. Rossmanith [2013], is left for future work.

We consider a discretization of the spatial domain $\mathcal{S}^2$ composed by non overlapping triangular elements which cover the sphere. For the mesh quantities, the same notation of chapter 1 is employed here, we refer to section 1.6.1 for the definitions. We just introduce a new notation for areas computed on the sphere, for which we use capital italic font (e.g. $\mathcal{A}$), in order to differentiate these from areas defined in the parametric space, for which standard italic is used as in 1.6.1. For instance the area of a spherical triangle and the median dual cell on the sphere are denoted as

$$|\mathcal{K}| = \int_K \sqrt{G} d\boldsymbol{z}, \quad |\mathcal{C}_i| = \sum_{K \in \mathcal{D}_i} \frac{|\mathcal{K}|}{3} \tag{4.22}$$

Geometrical quantities computed in the parametric space are defined as in Cartesian case. For clarity we repeat

$$|K| = \int_K d\boldsymbol{z}, \quad |C_i| = \sum_{K \in \mathcal{D}_i} \frac{|K|}{3}$$

Finally we specify $\boldsymbol{n}_j = n_j^m \boldsymbol{g}_m$ as the normal to the j-th face of the triangle scaled by the corresponding edge length.

4.5 DGCL in curvilinear coordinates

A first look to the DGCL and its relevance for SWEs-ALE simulation has been given in chapter 2. Here we come through the main definitions given in section 2.5 and extends them to general smooth manifolds (as the sphere). When approximating the SWEs the most fundamental equations are the integral ones. It is thus more useful to consider the integral GCL over a general control volume V. The time discrete approximation of (4.15), namely the Discrete Geometric Conservation Law (DGCL) writes:

$$\left[\int_V \sqrt{G} d\boldsymbol{z}\right]^{n+1} - \left[\int_V \sqrt{G} d\boldsymbol{z}\right]^{n} = \Delta t \int_{\partial V} \sqrt{G} \sigma^m n^m ds \tag{4.23}$$

or more compactly, using spherical areas

$$|\mathcal{V}^{n+1}| - |\mathcal{V}^{n}| \;=\; \Delta t \sum_{j\in K} \upsilon_j \tag{4.24}$$

we have decomposed the left-hand side in the contributions of the three edges. The edge velocity results:

$$\upsilon_j = \int_{\partial V_j} \sqrt{G}\sigma^m n_j^m ds \tag{4.25}$$

In chapter 2 we have presented a class of DGCL numerical schemes for the approximation of plane SWEs based on previous work, see Mavriplis and Yang [2006]. We have also seen that the presented closure collapses to the one of Lesoinne and Farhat [1996]. A first difference when setting the problem in curvilinear coordinates is that, for general metric tensor $\sqrt{G} \neq 1$, we cannot define the DGCL through the original characterization of Thomas and Lombard [1979], i.e. the preservation of a uniform flow. On the sphere one could think to replace uniform flows with the preservation of zonal/meridional flows. But, even general numerical schemes on fixed grid do not have this property, so one could question the importance to have it on moving meshes. Let's say however that we would like to conserve the total area of the mesh. In this sense the proof of Farhat comes to be less general: elements' areas are computed through quadrature formulas and it is hard to find proper quadrature points in time and in space to evaluate the term $\int_{t_n}^{t_{n+1}} \int \sigma^j n^j ds dt$ such that eq.(2.23) results an identity (e.g. in this case midpoint does not assure exactness). Instead the closure of Mavripils is the one that we prefer and, with a minor modification, allows to respect the GCL.

Proposition (DGCL in curvilinear coordinates) *A numerical method approximating eq. (4.3) on a smooth manifold $\mathcal{M}$, verifies the DGCL constraint or eq. (2.23) if each edge velocity is computed through the (signed) area swept on $\mathcal{M}$ by the j-th edge of the polygon in one time step*

$$\upsilon_j = \frac{1}{\Delta t} \int_{Q_j} \sqrt{G} d\boldsymbol{z} \tag{4.26}$$

where the quadrangle Q_j is defined by vertexes $\boldsymbol{z}_i^n$,$\boldsymbol{z}_k^n$,$\boldsymbol{z}_k^{n+1}$,$\boldsymbol{z}_i^{n+1}$, i and k denotes the two vertexes of the j-th edge. We use the convention that, if the polygon is expanding, then the area swept is positive.

A brief comment is necessary. Even if complicated analytical formulas exist to compute the area of a spherical quadrilateral, υ_j will be computed with a quadrature formula to approximate the integral in (2.25). This means that, at a later stage, we have to enforce explicitly the DGCL (2.23) to compute the spherical area in the new configuration

$$|\mathcal{V}^{n+1}| = |\mathcal{V}^n| + \Delta t \sum_{j \in K} \upsilon_j$$

Up to the our knowledge this is the first attempt to find DGCL fluxes in curvilinear coordinates for general metric.

4.6 Finite Volume for SWEs on the sphere

There are two main complications that one has to face when attempting to solve the SWEs on the sphere: the first concerns the non-autonomous nature of the SWEs on a manifold, and the second the resolution of the Riemann problem in curvilinear coordinates. Although the literature on Finite Volume schemes on the sphere is large Machenhauer *et al.* [2009], we refer, with respect to the two aforementioned issues, to the Lax-Wendroff scheme of Rossmanith *et al.* [2004], and more closely to the MUSCL-type scheme implemented in Ullrich *et al.* [2010].

We will compute approximations of solution averages over the standard median dual cells that we denote as u_i

$$\mathsf{u}_i(t) = \frac{1}{\mathcal{C}_i} \int_{C_i} \mathsf{u}(\boldsymbol{z}, t)\, \sqrt{G} dz^m \tag{4.27}$$

A common choice is to set $\mathcal{C}_i = |C_i|\sqrt{G}_i$. Then we define the numerical flux along the boundary of the dual cell ∂C_{ij} and the numerical source on dual cell C_{ij}, such as in chapter 1 and 2:

$$\begin{aligned}
\mathcal{F}_{ij} &= \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \int_{\partial C_{ij}} \mathsf{F}^j n^j ds\, dt \\
\mathcal{F}_{ij}^{ALE} &= \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \int_{\partial C_{ij}} \mathsf{w} \sigma^m n^m \sqrt{G} ds\, dt = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathsf{w} \upsilon_{ij}\, dt \\
\mathcal{S}_{ij} &= \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \int_{C_{ij}} \mathcal{S} \sqrt{G} d\boldsymbol{x}\, dt
\end{aligned}$$

The ALE-FV discrete evolution equations then reads:

$$|\mathcal{C}_i^{n+1}| \mathsf{w}_i^* = |\mathcal{C}_i^n| \mathsf{w}_i^n - \Delta t \sum_{j \in \mathcal{D}_i} R_{ij}(\mathsf{w}^n, b^n) \tag{4.28}$$

$$|\mathcal{C}_i^{n+1}| \mathsf{w}_i^{n+1} = |\mathcal{C}_i^n| \mathsf{w}_i^n - \frac{\Delta t}{2} \sum_{j \in \mathcal{D}_i} \left(\frac{R_{ij}(\mathsf{u}^n, b^n)}{2} + \frac{R_{ij}(\mathsf{u}^*, b^{n+1})}{2} \right) \tag{4.29}$$

with $R_{ij} = \mathcal{F}_{ij} + \mathcal{S}_{ij}$. Please note how the capacity coefficient have been naturally considered by applying the average preserving property (4.27). We make use to the Roe-type numerical fluxes, as in the planar case (see (2.40)) (to which we refer for the notation):

$$\mathcal{F}_{ij} = \frac{1}{2}(\breve{\mathbf{F}}_i + \breve{\mathbf{F}}_j) \cdot \boldsymbol{n}_{ij} - \upsilon_{ij} \frac{\breve{\mathsf{w}}_j + \breve{\mathsf{w}}_i}{2} - \frac{\left|\mathsf{K}_{ij} - \upsilon_{ij} \mathsf{I}_3\right|}{2} (\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i) \tag{4.30}$$

Note that Roe fluxes have been tested in Ullrich *et al.* [2010] for the low Mach number tests of Williamson *et al.* [1992] showing adequate accuracy. We refresh the notation of chapter 2: $\mathsf{K}_{ij} = \mathsf{K}(\boldsymbol{n}_{ij}, \mathsf{z}^*_{ij})$ and $\mathsf{R}_{ij} = \mathsf{R}(\boldsymbol{n}_{ij}, \mathsf{z}^*_{ij})$ are respectively the flux Jacobian and the matrix of right eigenvectors, evaluated with a Roe linearization z^*_{ij}. Due to the spatially varying flux function, the property of the Roe average in so no more available, however, through the numerical flux (4.30), we still compute a local conservative solution that assures $\mathcal{F}_{ij} = -\mathcal{F}_{ji}$. In our implementation we have used the following average

$$\mathsf{z}^*_{ij} = \begin{bmatrix} \frac{h_i G_i + h_j G_j}{2} \\ \frac{\sqrt{h_i} u_i \sqrt{G}_i + \sqrt{h_i} u_j \sqrt{G}_j}{\sqrt{h_j} + \sqrt{h_j}} \\ \frac{\sqrt{h_i} v_i \sqrt{G}_i + \sqrt{h_i} v_j \sqrt{G}_j}{\sqrt{h_j} + \sqrt{h_j}} \end{bmatrix} \tag{4.31}$$

For completeness concerning FV for spatially varying flux function we mention the different perspective of Bale *et al.* [2002]. Their *flux wave decomposition* allows to truly takes into account the non autonomous nature of the flux, directly into the resolution of the Riemann problem.

To accomplish the DGCL, we follow the strategy described in section 4.5 which consist of computing spherical area swept by each interface of the dual cell. The resulting interface velocities υ_{ij}, defined in (4.25), are prescribed by (4.26)

$$\begin{aligned} \upsilon_{ij} &= \int_{\partial C_{ij}} \sigma^m n^m \sqrt{G}\, ds = \sum_{K \ni i,j} \int_{\partial C^K_{ij}} \sigma^m n^m \sqrt{G}\, ds \\ &= \sum_{K \ni i,j} \int_{Q^K_{ij}} \sqrt{G}\, d\boldsymbol{z} \end{aligned} \tag{4.32}$$

using the quadrilateral Q^K_{ij} defined by vertexes $\boldsymbol{x}^n_{GK}$,$\boldsymbol{x}^n_{ij}$,$\boldsymbol{x}^{n+1}_{ij}$,$\boldsymbol{x}^{n+1}_{GK}$. Once this value is computed in the code, the dual cell area can be updated in the new configuration:

$$\mathcal{C}^{n+1}_i = \mathcal{C}^n_i + \Delta t \sum_{j \in \mathcal{D}_i} \upsilon_{ij} \tag{4.33}$$

FV algorithm in curvilinear coordinates

Although one can use the algorithm described in the previous section, additional difficulties arise in the resolution of the Riemann problem in generalized coordinates. In particular the question of the basis in which we solve the Riemann problem becomes urgent since the basis' vector is changing from point to point. Using straightforwardly the Cartesian solver makes no sense. Indeed the flux jump at the interface between two dual cells has no meaning, being the flux $\mathbf{F}_i$ and $\mathbf{F}_j$, composed by vectors and tensors defined on different basis. There is also second issue: the Finite Volume discretization of the geometric source term related to Earth curvature.

Concerning the resolution of the Riemann Problem, Rossmanith *et al.* [2004] propose to transform vectors $\mathbf{F}_i$ and $\mathbf{F}_j$ to a common coordinate system, for example the coordinate system at the interface position $\boldsymbol{z}_{ij}$, through the action of *parallel transport*. We propose an extension of their algorithm and, if possible, their conclusions to the case of unstructured grids. We define the parallel transport operator P on the vector $\boldsymbol{u}$ as the vector field tangent to the curve parametrized by the geodesic parameter λ:

$$\begin{aligned} P(\boldsymbol{u}) &= \frac{d\boldsymbol{u}}{d\lambda} \\ &= \frac{d}{d\lambda}(u^m \boldsymbol{g}_m) = \frac{du^m}{d\lambda}\boldsymbol{g}_m + \frac{dz^n}{d\lambda}\frac{d\boldsymbol{g}_m}{dz^n}u^m = \left(\frac{du^k}{d\lambda} + \frac{dz^n}{d\lambda}\Gamma^k_{nm}u^m\right)\boldsymbol{g}_k \end{aligned}$$

Parallel transport is an operation that takes a vector $\boldsymbol{u}$ at a point A and transports it to a point B along the curve $z^i(\lambda)$. Parallel transport accomplishes this in such a way that $\boldsymbol{u}$ remains parallel to $z^i(\lambda)$. This constraint is the parallel transport equation:

$$P(\boldsymbol{u}) = 0 \qquad \Rightarrow \frac{du^k}{d\lambda} + \Gamma^k_{nm}\frac{dz^n}{d\lambda}u^m = 0$$

If one evaluate the linearization of the above expression at the interface gets:

$$\begin{aligned} u^k_{ij} &= u^k_i + \frac{\Delta\lambda}{2}\left.\frac{du^k}{d\lambda}\right|_i + \mathcal{O}(\Delta\lambda^2) \\ &= u^k_i - \frac{\Delta\lambda}{2}\frac{dz^n}{d\lambda}\left(\Gamma^k_{nm}u^m\right)_i + \mathcal{O}(\Delta\lambda^2) \end{aligned}$$

where the indexes m, n and k loop with standard summation convention while indexes i and j represent the nodes; node i in particular is the node around which we have performed linearization. Index ij denotes values at the interface between node i and node j. We can transport the full vector of conservative variables:

$$\begin{bmatrix} h \\ hu^k \end{bmatrix}_{ij} = \left(\begin{bmatrix} 1 & 0 \\ 0 & \mathsf{I}^2 \end{bmatrix} - \frac{\Delta z^n_{ij}}{2}\begin{bmatrix} 0 & 0 \\ 0 & \Gamma^k_{nm} \end{bmatrix}_i\right)\begin{bmatrix} h \\ hu^m \end{bmatrix}_i$$

in our implementation we have simplified $\Delta\lambda\frac{dz^n}{d\lambda} = \Delta z^n_{ij} = z^n_j - z^n_i$. We can compactly write the *discrete parallel transport operator* which transports the vector of conservative variable from node i to the interface ij:

$$\mathsf{u}_{ij} = \mathsf{P}_{ij}\mathsf{u}_i \quad \mathsf{P}_{ij} = \begin{bmatrix} 1 & 0 \\ 0 & \mathsf{I}^2 \end{bmatrix} - \frac{\Delta z^n_{ij}}{2}\begin{bmatrix} 0 & 0 \\ 0 & \Gamma^k_{nm} \end{bmatrix}_i = \mathsf{I}^3 + \mathsf{B}_{ij}$$

defining a *parallel transport matrix* P_{ij}. Once conservative variables are transported at the common interface, then the resolution of the Riemann problem can be carried, through (1.15)(1.16) or (1.17) (chapter 1). Let us reconsider the Riemann problem (on a fixed grid for simplicity) defined at the interface between nodes i and j, both lying on a certain geodesic parametrized with λ. The fluxes in these cells are $\mathbf{F}_i$ and $\mathbf{F}_j$ respectively. Let us apply the appropriate parallel transport matrices so that $\mathbf{F}_i$ and $\mathbf{F}_j$ both get (approximately) represented in the coordinate basis at $\boldsymbol{z}_{ij}$. Relations (1.15)(1.16) become:

$$\begin{aligned} \mathcal{F}_{ij} &= \mathsf{P}_{ij}\mathbf{F}_i \cdot \boldsymbol{n}_{ij} - \sum_{p=1,\beta^p<0}^{m} \mathcal{W}_p \\ &= \mathsf{P}_{ji}\mathbf{F}_j \cdot \boldsymbol{n}_{ij} + \sum_{p=1,\beta^p>0}^{m} \mathcal{W}_p \end{aligned}$$

Once the solution jump has been transported to the interface, it can be decomposed in p-waves and resolved for the coefficient α_p

$$\mathcal{W}_p = \alpha_p \lambda_p \mathsf{r}_p, \quad \alpha_p = \mathsf{l}_p \left(\mathsf{P}_{ji}\mathsf{u}_j - \mathsf{P}_{ij}\mathsf{u}_i\right)$$

First order numerical fluxes are written in a compact form

$$\mathcal{F}_{ij} = \frac{1}{2}(\mathbf{F}_j + \mathbf{F}_i) \cdot \boldsymbol{n}_{ij} - \frac{1}{2}|\mathsf{K}_{ij}|(\mathsf{u}_j - \mathsf{u}_i) + \mathcal{S}^\gamma_{ij} \tag{4.34}$$

where the following terms can be isolated in a source

$$\mathcal{S}^\gamma_{ij} = \frac{1}{2}\left(\mathsf{B}_{ji}\mathbf{F}_j \cdot \boldsymbol{n}_{ij} + \mathsf{B}_{ij}\mathbf{F}_i \cdot \boldsymbol{n}_{ij}\right) - \frac{1}{2}|\mathsf{K}_{ij}|(\mathsf{B}_{ji}\mathsf{u}_j - \mathsf{B}_{ij}\mathsf{u}_i) \tag{4.35}$$

To obtain a formally second order update we can write numerical fluxes in the following form

$$\mathcal{F}_{ij} = \frac{1}{2}(\breve{\mathbf{F}}_j - \breve{\mathbf{F}}_i) \cdot \boldsymbol{n}_{ij} + (\breve{\mathbf{F}}_i - \mathbf{F}_i) \cdot \boldsymbol{n}_{ij} - \frac{1}{2}|\mathsf{K}_{ij}|(\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i) + \breve{\mathcal{S}}^\gamma_{ij}$$

This approximate source term result enhanced by a second order correction:

$$\breve{\mathcal{S}}^\gamma_{ij} = \mathcal{S}^\gamma_{ij} + \mathcal{S}^{\gamma-}_{ij}$$

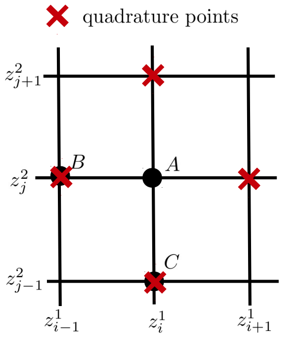

Figure 4.3: Cartesian grid of Rossmanith *et al.* [2004] and quadrature points for numerical approximation of the geometric source term

with the correction that writes:

$$\mathcal{S}^{\gamma-}_{ij} = \mathsf{B}_{ij}\left(\breve{\mathbf{F}}_i - \mathbf{F}_i\right)\cdot \boldsymbol{n}_{ij}$$

What does this source term $\breve{\mathcal{S}}^{\gamma}_{ij}$ represent? On structured quadrilateral (*Cartesian*) grids this corresponds to have directly an approximation for the geometric source term $\mathcal{S}^{\gamma}$ in (4.3). In this case, the application of parallel transport within the wave propagation algorithm produces a natural numerical approximation of the geometric source term, which was the second issue reported.

Proof. Consider a Cartesian grid on the sphere with the following nodes:

$$z^1_i = z^1_l + \left(i - \frac{1}{2}\right)\Delta z^1$$
$$z^2_j = z^2_l + \left(j - \frac{1}{2}\right)\Delta z^2$$

with z^1_l, z^2_l is the lower left corner of the computational rectangular domain. Let's consider a median dual cell area in figure 4.3 centered in $A = (z^1_i, z^2_j)$. Let's consider the vertical interface between node A and $B = (z^1_{i-1}, z^2_j)$ where $n^1 = -\Delta z^2$ and $n^2 = 0$. Moreover

$$\Delta z^1_{i-1,i} = z^1_i - z^1_{i-1} = \Delta z^1 \quad \Delta z_{i,i-1} = z^1_{i-1} - z^1_i = -\Delta z^1$$

We consider the central part of the source term (4.35) that, on a Cartesian grid reads

$$\mathcal{S}^{\gamma}_{i-1/2,j} = \frac{1}{2}\Delta z^1(-\Delta z^2)\begin{bmatrix} 0 \\ -\frac{a}{2}\Gamma^k_{m1} huu^{m1}\end{bmatrix}_{i-1,j} - \frac{1}{2}(-\Delta z^1)(-\Delta z^2)\begin{bmatrix} 0 \\ -\frac{a}{2}\Gamma^k_{m1} huu^{m1}\end{bmatrix}_{i,j}$$

Let's consider the horizontal interface between node A and $C = (z_i^1, z_{j-1}^2)$ where $n^1 = 0$, $n^2 = -\Delta z^1$ and

$$\Delta z_{j-1,j}^2 = z_j^2 - z_{j-1}^2 = \Delta z^2 \quad \Delta z_{j,j-1}^2 = z_{j-1}^2 - z_i^2 = -\Delta z^2$$

and

$$\mathcal{S}_{i,j-1/2}^{\gamma} = \frac{1}{2}(-\Delta z^1)\Delta z^2 \begin{bmatrix} 0 \\ -\frac{a}{2}\Gamma_{m2}^k huu^{m2} \end{bmatrix}_{i,j-1} - \frac{1}{2}(-\Delta z^1)(-\Delta z^2) \begin{bmatrix} 0 \\ -\frac{a}{2}\Gamma_{m2}^k huu^{m2} \end{bmatrix}_{i,j}$$

Summing all the contributions from the four interfaces, taking into account the source term appears with a minus sign in the global update (4.28):

$$\begin{aligned} \mathcal{S}_{ij}^{\gamma} &= -\Delta z^1 \Delta z^2 \begin{bmatrix} 0 \\ \sum_{q1} \omega_{q1} a_{q1} \Gamma_{1n}^k huu_{q1}^{1n} + \sum_{q2} \omega_{q2} a_{q2} \Gamma_{2n}^k huu_{q2}^{2n} \end{bmatrix} \\ &\approx \int_{\Delta z^1} \int_{\Delta z^2} \mathcal{S}^{\gamma}(\boldsymbol{z})\, d\boldsymbol{z} \end{aligned}$$

we get a quadrature formula for the integral of the geometric source term. Different quadrature points are used for the the two terms containing Christoffel symbols Γ_{1m}^k and Γ_{1m}^k. For the first, quadrature points are placed on the midpoint of the upper and lower edge of the rectangular dual cell, for the latter the two quadrature points coincide with the midpoints of left and right edges, see again figure 4.3. Similar arguments hold for the upwind part. □

For unstructured grids such an approximation of the geometric source is not obvious since the inclusion of parallel transport into the resolution of the Riemann Problem can become complicated. In fact, taking the central part of the source term (4.35) we have

$$\mathsf{B}_{ij} \mathbf{F}_i \cdot \boldsymbol{n}_{ij} = \begin{bmatrix} 0 \\ -\frac{a \Delta z_{ij}^n}{2} \Gamma_{mn}^k huu^{mq} n_{ij}^q \end{bmatrix}_i$$

which *includes* but not coincides with a discretization of the geometric source term. We will refer to this formulation as the one of Rossmanith *et al.* [2004] since it is an extension of their work to unstructured grids.

An alternative yet simpler approximation of the geometric source consists in approximating $\mathcal{S}^{\gamma}$ directly on the underlying grid as done by Ullrich *et al.* [2010] on a structured grid. An extension of this last work to unstructured grids consists in using the pointwise value:

$$\int_{C_i} \mathcal{S}^{\gamma} \sqrt{G} d\boldsymbol{z} \approx |C_i| \sqrt{G_i} \mathcal{S}_i^{\gamma} = |C_i| \sqrt{G_i} \begin{bmatrix} 0 \\ \Gamma_{mn}^k huu^{mn} \end{bmatrix}_i \qquad (4.36)$$

such a discretization is second order accurate. We have tested both the formulations in figure 4.4 for a circular hump propagating on a sphere. Even if they provide similar accuracy, the complexity of (4.35), together with the additional cost to include high order reconstruction, drove us to the implement directly (4.36). This will represent the final implementation of the FV.

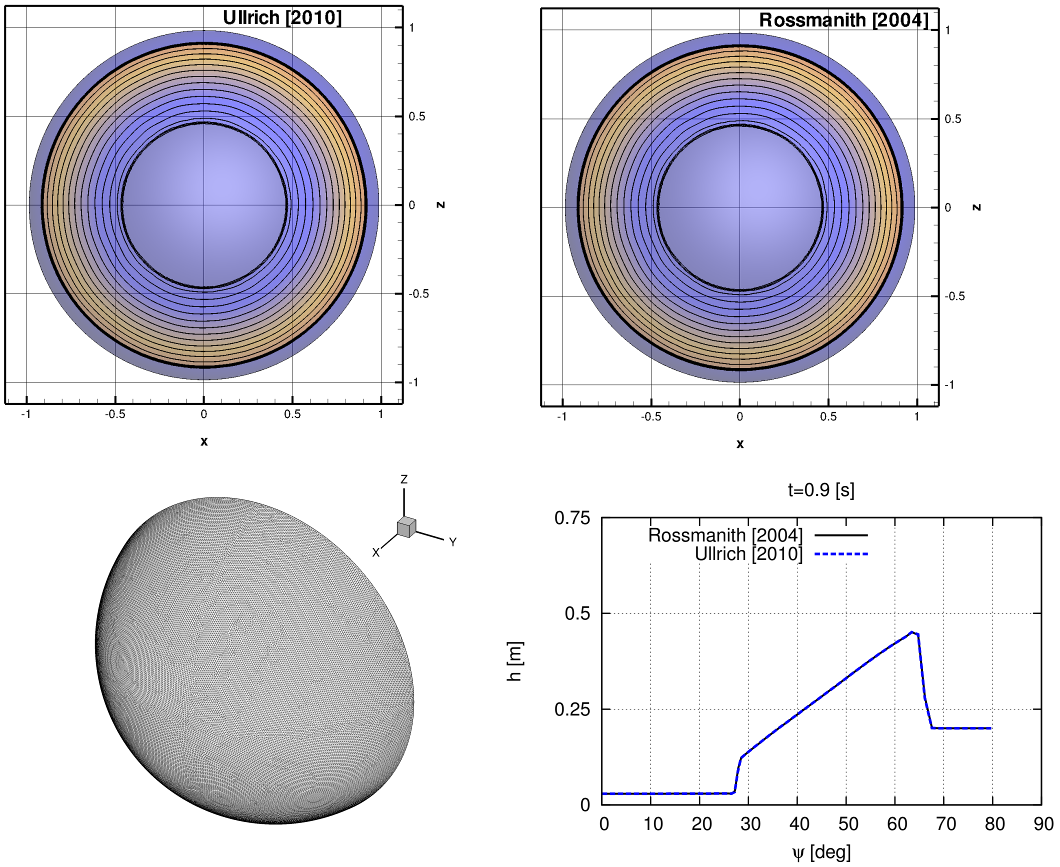

Figure 4.4: Circular hump propagating on a sphere. Comparison between two different treatments of the geometric source term on the fixed unstructured grid in the bottom-right picture. Dashed-blue) extension to unstructured grids of the centered approximation of Ullrich *et al.* [2010]. Continuous-black) extension to unstructured grids of the wave propagation algorithm Rossmanith *et al.* [2004]

Well Balanced for moving mesh on the sphere

The discretization of the topography/bathymetry term follows what done for plane Cartesian coordinates (see sections 1.6.2 and 2.10.1) which in turns consisted in the upwind discretization of Hubbard and Garcia-Navarro [2000]; Nikolos and Delis [2009]. We distinguish two contributions $\mathcal{S}^b_{ij} = \mathcal{S}^c_{ij} + \mathcal{S}^*_{ij}$. We introduce the nodal differences

$$\Delta h = \begin{bmatrix} h_j - h_i \\ 0 \\ 0 \end{bmatrix}, \quad \Delta b = \begin{bmatrix} b_j - b_i \\ 0 \\ 0 \end{bmatrix}$$

and the following Jacobian at rest $\mathsf{A}_{ij} = \mathsf{A}(\boldsymbol{n}_{ij}, h_{ij})$ and $\mathsf{A}^-_{ij} = \mathsf{A}(\boldsymbol{n}_{ij}, h^-_{ij})$ with average values (see (4.7)):

$$h_{ij} = \frac{\breve{h}_i \sqrt{G}_i G_i^{km} + \breve{h}_j \sqrt{G}_j G_j^{km}}{2}, \quad h^-_{ij} = \frac{\sqrt{G}_i G_i^{km} \breve{h}_i + \sqrt{G}_j G_j^{km} h_i}{2}$$

The central part

$$\mathcal{S}^c_{ij} = \frac{1}{2} \mathsf{A}_{ij} \Delta b + \mathsf{A}^-_{ij} \Delta b^-_{ij} \tag{4.37}$$

and the upwind dissipation term :

$$\mathcal{S}^*_{ij} = -\frac{\left|\mathsf{K}_{ij} - \upsilon_{ij} \mathsf{I}_3\right|}{2} \Delta b_{ij} \tag{4.38}$$

However for the FV scheme presented so far we cannot prove Well Balanced. The reason is that we cannot use the Roe average property in the demonstration of (1.6.2), spoiling Well Balancedness. To cure this main problem we have isolated the hydrostatic part of the flux and the geometric source term. Setting $\Pi = \frac{1}{2} g h^2$ we have to approximate

$$\int_{C_{ij}} P G^{ij} n^j \sqrt{G} d\boldsymbol{z} + \int_{C_{ij}} \Gamma^i_{jk} \, P G^{jk} \sqrt{G} d\boldsymbol{z} \tag{4.39}$$

We now write the above quantity into a *non conservative form.* Putting the flux (first term) in divergence form and deriving term by term we get

$$\frac{1}{\sqrt{G}} \frac{\partial}{\partial z^j} \left(\sqrt{G} \Pi G^{ij} \right) = \Gamma^m_{mj} \left(\Pi G^{ij} \right) + \frac{\partial G^{ij}}{\partial z^j} \Pi + \frac{\partial \Pi}{\partial z^j} G^{ij}$$

where we have used relation (A.6). We add the metric term and using Ricci's Lemma (A.7) the first, second and fourth term cancel out. We can simplify expression (4.39):

$$\Pi \left(\frac{\partial G^{ij}}{\partial z^j} + G^{ij} \Gamma^m_{mj} \right) + \frac{\partial \Pi}{\partial z^j} G^{ij} + \Pi \Gamma^i_{jk} G^{jk} = g h \frac{\partial h}{\partial z^j} G^{ij}$$

It will be this non conservative form of the hydrostatic term that we approximate as

$$\int_{C_{ij}} \sqrt{G} G^{ij} g h \frac{\partial}{\partial z^j} \begin{bmatrix} 0 \\ h \end{bmatrix} d\boldsymbol{z} = \frac{1}{2} \mathsf{A}_{ij} \Delta h_{ij} + \mathsf{A}^-_{ij} \Delta h^-_{ij}$$

As in the Cartesian case two contributions appear, one related to first-order piece-wise constant solution approximation, the second one comes from the integration of the solution's variations due to the gradient-reconstruction. We remark that the presence of variable-coefficients in the matrix A makes the non-conservative discretization no longer equivalent to the conservative one. Strictly speaking, we are loosing exact conservation to the extent that *Rankine-Hugoniot condition are no more satisfied exactly at a discrete level.* We remark however that the variation of the metric coefficients are very smooth. This problem of Well-Balanced scheme on the sphere is already known and we note that our WB approximation on the sphere resembles to the work of Ullrich *et al.* [2010]. For different WB scheme for the SWEs on the sphere, we refer to Rossmanith *et al.* [2004] and Yang and Cai [2011].

Now we have all the formulas to prove Well Balancedness with moving meshes.

Proposition 1. *The finite volume discrete equations* (4.28)-(4.29) *with definitions* (4.30), (4.32),(4.37) *and* (4.38) *verifies the Well-Balanced property both on moving and fixed mesh, provided that the same reconstruction procedure is used for* u *and* b*.*

Proof. For the constant lake at rest solution $\mathsf{w}_0 = [const\, 0\, 0]^T$, numerical flux and the source term reduce to

$$\begin{aligned} \mathcal{F}_{ij} &= \frac{1}{2} \mathsf{A}_{ij}\, \Delta h_{ij} + \mathsf{A}^-_{ij}\, \Delta h^-_{ij} - \upsilon_{ij} \mathsf{w}_0 - \frac{|\mathsf{A}_{ij} - \upsilon_{ij} \mathsf{I}_3|}{2} (\breve{\mathsf{u}}_j - \breve{\mathsf{u}}_i) \\ \mathcal{S}_{ij} &= \frac{1}{2} \mathsf{A}_{ij}\, \Delta b_{ij} + \mathsf{A}^-_{ij}\, \Delta b^-_{ij} - \frac{|\mathsf{A}_{ij} - \upsilon_{ij} \mathsf{I}_3|}{2} \Delta b_{ij} \end{aligned}$$

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. We are left with the DGCL

$$|\mathcal{C}^{n+1}_i| \mathsf{w}^{n+1}_i = |\mathcal{C}^n_i| \mathsf{w}^n_0 - \mathsf{w}_0 \Delta t \sum_{j \in \mathcal{D}_i} \upsilon_{ij}$$

which is verified by hypothesis (4.33). This complete the proof:

$$\mathsf{w}^{n+1}_i = \frac{\left(|\mathcal{C}^n_i| - \Delta t \sum\limits_{j \in \mathcal{D}_i} \upsilon_{ij} \right) \mathsf{w}_0}{|\mathcal{C}^{n+1}_i|} = \mathsf{w}_0$$

□

We remark that the proof does not rely, as usually done for the Cartesian coordinates, on the property of the Roe average, which again is not available in this case.

4.7 Residual Distribution for SWEs on the sphere

Steady Residual Distribution have been extended to the rotating sphere in Rossmanith [2013]. In this section we propose a high order extension for time dependent problem on moving meshes. This is the first such generalization of RD to our knowledge. To obtain second order of accuracy in time, we extend the eRK2 ALE-RD of Arpaia *et al.* [2014] detailed in chapter 2 and appendix D. Compared to FV, the extension to the sphere appears almost straightforward. In the following we provide directly the algorithm that we comment extensively later:

Given the fluctuation, see (4.7):

$$\Phi^K(\mathsf{w}_h, b_h) = \int_{\partial K} \mathsf{F}^j(\mathsf{u}_h) n^j ds + \int_K \sigma^j \frac{\partial \mathsf{w}}{\partial z^j} \sqrt{G} d\boldsymbol{z} + \int_K \mathcal{S}(\mathsf{u}_h, \boldsymbol{z}) \sqrt{G} d\boldsymbol{z}$$

If not specified, for us $K = \frac{1}{2}\left(K^n + K^{n+1}\right)$ in the following. Keeping in mind this, the two steps writes

1] Predictor step: for each element $K \in \mathcal{T}_h$

- Compute the residual $\Phi^{K(1)} = \Phi^K(\mathsf{w}_h^n, b_h^n)$.
- Distribute the fluctuation to the nodes of K such that $\sum_{j \in K} \Phi_j^{K(1)} = \Phi^{K(1)}$
- Compute the first order prediction of the solution, denoted as w^*

$$\mathsf{w}_i^* = \mathsf{w}_i^n - \frac{\Delta t}{|\mathcal{C}_i^{n+1}|} \sum_{K \in \mathcal{D}_i} \Phi_i^{K(1)}(\mathsf{w}_h^n, b_h^n) \tag{4.40}$$

2] Corrector step: for each element $K \in \mathcal{T}_h$

- Compute the residual

$$\Phi^{K(2)} = \frac{1}{2}\left(\Phi^K(\mathsf{w}_h^n, b_h^n) + \Phi^K(\mathsf{w}_h^*, b_h^{n+1})\right) + \int_K \frac{\mathsf{w}_h^* - \mathsf{w}_h^n}{\Delta t} \sqrt{G} d\boldsymbol{z} \tag{4.41}$$

- Distribute the fluctuation to the nodes of K such that $\sum_{j \in K} \Phi_j^{K(2)} = \Phi^{K(2)}$

- Compute the second order correction from

$$\mathsf{w}_i^{n+1} = \mathsf{w}_i^* - \frac{\Delta t}{|\mathcal{C}_i^{n+1}|} \sum_{K \in \mathcal{D}_i} \Phi_i^{K(2)}(\mathsf{w}_h^*, \mathsf{w}_h^n, b_h^n, b_h^{n+1}) \tag{4.42}$$

All the integral are still approximated via Gaussian quadrature. We believe it is worth noting both the simplicity and the rigor of the RD extension from Cartesian coordinates to curvilinear ones, specially if compared to FV. In particular:

1. the heterogeneity of the medium is taken into account integrating the coefficient $\sqrt{G}$ (hidden in the fluxes or in the area definition) with the precision degree of the quadrature formula.
2. the geometric source term comes from the fact that the flux is written in curvilinear coordinates. As for any other source term, it is included automatically in the wave propagation algorithm through the splitting approach, see section 1.6.3. We recall that for FV an analogous approach was much more complicated.

4.7.1 RD mass matrix computation on manifold

In this paragraph we detail the practical computation of the split residual (4.41). In particular it is worth to report how we compute the unsteady term in (4.41) since this could impact consistency and (we will see) conservation properties on the sphere. First, we compute the following important quantities as:

$$m_{ij}^{GAL} = \int_K \varphi_i \varphi_j \sqrt{G}\, d\boldsymbol{z} \approx \frac{\mathcal{K}}{12}(1+\delta_{ij}) \qquad |\mathcal{K}_j| = \int_K \varphi_j \sqrt{G}\, d\boldsymbol{z} \approx \frac{\mathcal{K}}{3} \tag{4.43}$$

where the $|\mathcal{K}|$ has been defined in (4.22). We refer to m_{ij}^{GAL} as the Galerkin mass matrix. On Cartesian plane the above definitions collapse respectively to the Galerkin mass matrix and to $|\mathcal{K}_j| = \frac{|K|}{3}$. In chapter 1 we have already mentioned that the time discretization in RD methods is obtained from an analogy with the theory of Finite Element. From the weak form it is well known that an element mass matrix of the form

$$m_{ij}^K = \int_K w_i \varphi_j \sqrt{G}\, d\boldsymbol{z}$$

emerges, where w_i is the Petrov-Galerkin test function. Within a RD context, it is important to build mass matrices which are consistent with the space approximation, thus with a given set of distribution coefficients. Assigned a certain distribution $\{\beta_i^K\}_{i\in K}$, this consistency property is translated in the following constraint on w_i Ricchiuto and Abgrall [2010]:

$$\beta_i^K = \frac{1}{\int_K \sqrt{G}\, d\boldsymbol{z}} \int_K w_i \sqrt{G}\, d\boldsymbol{z} \tag{4.44}$$

which is verified, for example, by:

$$w_i = \varphi_i + \beta_i^K - \aleph_i^K, \quad m_{ij}^K = m_{ij}^{GAL} + |\mathcal{K}_j|(\beta_i^K - \aleph_i^K) \tag{4.45}$$

where $\aleph_i^K$ is:

$$\aleph_i^K = |\mathcal{K}_i||\mathcal{K}|^{-1} \tag{4.46}$$

With this in mind, we can treat separately the splitting of time residual (4.41). Invoking the analogy RD/Finite Elements (4.44) we let a mass matrix appears, mass matrix that can be computed with the consistent w_i of (4.45). At the end, the corrector residual (4.41) will be distributed according to:

$$\Phi_i^{K(2)} = \boldsymbol{\beta}_i^K \left(\frac{1}{2}\left(\Phi^K(\mathsf{w}^*) + \Phi^K(\mathsf{w}^n)\right) + \sum_{j\in K} \mathcal{K}_j \frac{\mathsf{w}_j^* - \mathsf{w}_j^n}{\Delta t} \right) + \sum_{j\in K} \delta m_{ij}^K \frac{\mathsf{w}_j^* - \mathsf{w}_j^n}{\Delta t} \tag{4.47}$$

We see that the last two terms in the above splitting (4.47) are related to the the three contributions of the mass matrix (4.45). In fact, two of them have been grouped in the so-called dissipation matrix, see Ricchiuto and Abgrall [2010]:

$$\delta m_{ij}^K = m_{ij}^{GAL} - |\mathcal{K}_j|\aleph_i^K \tag{4.48}$$

4.7.2 ALE part

We treat the ALE residual as done in chapter 2:

$$\begin{aligned}
\Phi^{K,ALE} &= \int_K \sqrt{G}\sigma^j \frac{\partial \mathsf{w}}{\partial z^j} d\boldsymbol{z} = \int_K \left(\frac{\partial \sqrt{G}\sigma^j \mathsf{w}}{\partial z^j} - \mathsf{w}\frac{\partial \sqrt{G}\sigma^j}{\partial z^j} \right) d\boldsymbol{z} \\
&= \int_{\partial K} \sqrt{G}\mathsf{w}\sigma^j n^j ds - \int_K \mathsf{w}\frac{\partial \sqrt{G}\sigma^j}{\partial z^j} d\boldsymbol{z}
\end{aligned}$$

We examine the first part. Edge fluxes are computed by quadrature formula. Moreover these fluxes should verify the DGCL which means that every edge should sweep the area $\Delta t \upsilon_j$. We can split these areas on the edge quadrature points through the weights

$$\int_{\partial K} \sqrt{G}\mathsf{w}\sigma^j n^j ds = \sum_{j\in K} \sum_{q}^{NQ} \omega_q \mathsf{w}_q \upsilon_j$$

while the second term using the usual passages

$$
\begin{aligned}
\int_K \mathsf{w}\frac{\partial\sqrt{G}\sigma^j}{\partial x^j} d\boldsymbol{z} &= \int_K \mathsf{w}\sqrt{G} d\boldsymbol{z}\, (\nabla\cdot\boldsymbol{\sigma})_K \\
&= \sum_{j\in K}\int \varphi_j \sqrt{G} d\boldsymbol{z}\, \mathsf{w}_j\, (\nabla\cdot\boldsymbol{\sigma})_K \\
&= \sum_{j\in K} |\mathcal{K}_j| \mathsf{w}_j \frac{\Delta|\mathcal{K}|}{\Delta t|\mathcal{K}|} \\
&= \frac{\sum_{j\in K}|\mathcal{K}_j|\mathsf{w}_j}{|\mathcal{K}|}\frac{\Delta|\mathcal{K}|}{\Delta t}
\end{aligned}
$$

where a weighted average solution on the cell appears. Finally using definition (4.43) for this term we get form

$$
\int_K \mathsf{w}\frac{\partial\sqrt{G}\sigma^j}{\partial x^j} d\boldsymbol{z} = \frac{\sum_{j\in K}\mathsf{w}_j}{3}\frac{\Delta|\mathcal{K}|}{\Delta t} \tag{4.49}
$$

4.7.3 Properties of the RD scheme

In the following we analyze some of the properties of the scheme proposed.

Conservation

Flows at geophysical scale are typically very smooth and for numerical methods approximating very smooth phenomena, the verification of Rankine-Hugoniot conditions could be not as urgent as it was for coastal and hydraulic applications. However global conservation of the flow variables on the sphere remains an important issue. It is interesting to check the global conservation statement for RD, in light of all the definitions given. For sake of clarity we take the source term equal to zero, since it will not affect the analysis. We sum the corrector (4.42) over all the nodes of the triangulation $i\in\mathcal{T}_h$:

$$
\sum_{i\in\mathcal{T}_h}\mathcal{C}_i^{n+1}\frac{\mathsf{w}^{n+1}-\mathsf{w}^*}{\Delta t} = -\sum_{K\in\mathcal{T}_h}\sum_{j\in K}\Phi_j^{K(2)}
$$

As seen in previous chapters, fluxes in divergence form cancel in the sums. On the right hand side, (4.47) reduce to the sum of the unsteady terms and the ALE term (4.49)

$$
\begin{aligned}
\sum_{i\in\mathcal{T}_h}\mathcal{C}_i^{n+1}\frac{\mathsf{w}_i^{n+1}-\mathsf{w}_i^*}{\Delta t} = &- \sum_{K\in\mathcal{T}_h}\sum_{j\in K}\boldsymbol{\beta}_j^K\sum_{m\in K}\left(\underbrace{\frac{\mathsf{w}_m^*+\mathsf{w}_m^n}{2}\frac{\Delta|\mathcal{K}|}{3\Delta t}}_{(4.49)}+\frac{|\mathcal{K}|}{3}\frac{\mathsf{w}_m^*-\mathsf{w}_m^n}{\Delta t}\right) \\
&- \sum_{K\in\mathcal{T}_h}\sum_{j\in K}\sum_{m\in K}\delta m_{jm}^K\frac{\mathsf{w}_m^*-\mathsf{w}_m^n}{\Delta t}
\end{aligned}
$$

Recalling the definition of the dissipation mass matrix (4.48), we observe that:

$$\sum_{i\in K}\sum_{j\in K}\delta m_{ij}^{K}(\mathsf{w}_j^{*}-\mathsf{w}_j^{n})=0$$

if the following condition is verified:

$$\sum_{j\in K} m_{ij}^{GAL}=|\mathcal{K}_i| \tag{4.50}$$

which is true for the definitions given in (4.43). Now we use consistency $\sum_{j\in K}\boldsymbol{\beta}_j^K=\mathsf{I}_3$:

$$\sum_{i\in\mathcal{T}_h}\left(\mathcal{C}_i^{n+1}\mathsf{w}_i^{n+1}-\mathcal{C}_i^{n+1}\mathsf{w}_i^{*}\right)=-\sum_{i\in\mathcal{T}_h}\sum_{K\in\mathcal{D}_i}\left(\frac{\mathsf{w}_i^{*}+\mathsf{w}_i^{n}}{2}\frac{\Delta|\mathcal{K}|}{3}+\frac{|\mathcal{K}|}{3}\left(\mathsf{w}_i^{*}-\mathsf{w}_i^{n}\right)\right)$$

where we have also used again $\sum_{K\in\mathcal{T}_h}\sum_{j\in K}=\sum_{i\in\mathcal{T}_h}\sum_{K\in\mathcal{D}_i}$. We sum and subtract $\sum_{i\in\mathcal{T}_h}\mathcal{K}_i^n\mathsf{w}_i^n$ to the left hand side and, using definition (4.22) we get:

$$\sum_{i\in\mathcal{T}_h}\left(\mathcal{C}_i^{n+1}\mathsf{w}_i^{n+1}-\mathcal{C}_i^{n}\mathsf{w}_i^{n}\right)=-\sum_{i\in\mathcal{T}_h}\sum_{K\in\mathcal{D}_i}\left(\frac{\mathsf{w}_j^{*}+\mathsf{w}_j^{n}}{2}\frac{\Delta|\mathcal{K}|}{3}-\frac{|\Delta\mathcal{K}|}{3}\frac{\mathsf{w}_j^{*}-\mathsf{w}_j^{n}}{2}\right)$$

which proves that we conserves the flow variables over the sphere:

$$\sum_{i\in\mathcal{T}_h}\mathcal{C}_i^{n+1}\mathsf{w}_i^{n+1}=\sum_{i\in\mathcal{T}_h}\mathcal{C}_i^{n}\mathsf{w}_i^{n}$$

Well Balancedness for moving mesh on the sphere

The discretization of the hydrostatic term have to ensure the discrete balance between hydrostatic and bottom forces when approximating a lake at rest flow. We can check that ($\frac{1}{2}gh_h^2=\Pi$ for shortness):

$$\begin{aligned}\Phi^{\mathsf{K}} &= \int_{\partial K}\mathsf{F}^j(\mathsf{u}_h)n^j\,ds+\int_K\mathcal{S}(\mathsf{u}_h,\,b_h)\,\sqrt{G}d\boldsymbol{z} \\ &= \begin{bmatrix}0\\ \int_{\partial K}\Pi n^j\,G^{ij}\sqrt{G}ds\end{bmatrix}+\begin{bmatrix}0\\ \int_K gh_h\frac{\partial b_h}{\partial z^j}\,G^{ij}\sqrt{G}\end{bmatrix}+\begin{bmatrix}0\\ \int_K\Pi\,\Gamma^i_{jk}G^{jk}\sqrt{G}\end{bmatrix} \\ &\neq 0\end{aligned} \tag{4.51}$$

the residual is not zero. Instead, for Cartesian coordinates ($\sqrt{G}=1$) under the assumption of P^1 and exact integration, the above statement is an equality. Unfortunately, in generalized curvilinear coordinates we loose these discrete equivalences because of the non linear metric coefficients which spoils the exactness of the integrals. As done for FV, it is convenient to rewrite the hydrostatic pressure term in a *non-conservative form*

$$\frac{1}{\sqrt{G}}\frac{\partial}{\partial z^j}\left(\sqrt{G}\,\Pi G^{ij}\right)+\Gamma^i_{jk}\,\Pi G^{jk}=gh\frac{\partial h}{\partial z^j}G^{ij}$$

and then integrate exactly with respect to linear variation of h and b. One can show the following result:

Proposition 3. *The explicit predictor corrector residual distribution prototype (4.40), (4.42) with edge velocities (2.25) verifies the Well-Balanced property both on moving and fixed meshes, provided that the same linear piecewise continuous approximation is used for* w*,* b *and consequently for* u*.*

Proof. We do the proof only for corrector, the predictor is a particular case. We impose $\mathsf{w}_0 = [const\, 0\, 0]^T$ into the residual (4.41) ($\Delta \mathsf{w}_0 = 0$):

$$\begin{aligned}\Phi^{K(2)} &= \int_{\partial K} \mathsf{F}_0^j n^j\, ds - \mathsf{w}_0 \sum_{j\in K} \sum_{q}^{NQ} \omega_q \upsilon_j + \mathsf{w}_0 \frac{\Delta|\mathcal{K}|}{\Delta t} + \int_K \mathcal{S}_0 \sqrt{G} d\boldsymbol{z} \\ &= \mathsf{w}_0 \left(\frac{\Delta|\mathcal{K}|}{\Delta t} - \sum_{j\in K} \upsilon_j \right) + \int_{\partial K} \mathsf{F}_0^j n^j\, ds + \int_K \mathcal{S}_0 \sqrt{G} d\boldsymbol{z} = 0\end{aligned}$$

where we have used the property of Gaussian weights $\sum_q^{NQ} \omega_q = 1$ to develop the second term. The term in brackets is zero if we compute the edge velocities with (2.25). We are left with the balance between hydrostatic flux and bathymetry terms. The flux term is now transformed to a non conservative form (4.52) which is more suited to accomplish Well Balancedness. We integrate the flux term in the divergence form and we can write the balance (4.51) as

$$\begin{bmatrix} 0 \\ \int_K g h_h \frac{\partial h_h}{\partial x^j} G^{ij} \sqrt{G} \end{bmatrix} + \begin{bmatrix} 0 \\ \int_K g h_h \frac{\partial b_h}{\partial x^j} G^{ij} \sqrt{G} \end{bmatrix} = 0$$

which is zero if the same linear piecewise continuous approximation is used for h, b. To have Well Balancedness it is sufficient to employ the same quadrature formula for the two integrals. Under these hypothesis $\Phi^{K(2)} = 0$ and we have $\mathsf{w}_i^{n+1} = \mathsf{w}_0$.

□

We remark that the integrals, even under the piecewise linear approximation, cannot be computed exactly, spoiling exact conservation. This loss of conservation is the sense that *Rankine-Hugoniot condition are no more satisfied exactly at a discrete level.* However the integral that has to be computed is in the following form:

$$\int_K \varphi_i\, G^{ij} \sqrt{G} d\boldsymbol{z}$$

The metric coefficients, which are infinitely smooth on the sphere, could be expanded in Taylor series and the whole term could be integrated with an arbitrary high degree of precision making the error much smaller then the discretization error.

4.8 Moving mesh on the sphere

Although h-refinement on the sphere has already received considerable attention (see the Adaptive Mesh Refinement methods of Behrens [1996, 1998]; Hubbard and Nikiforakis [2003]), the moving mesh method on the sphere is a quite recent subject of research. Up to the author's knowledge there are only two published works: the harmonic map of Di *et al.* [2006] and the Monge-Ampere equation of Weller *et al.* [2016]. In chapter 3 we have seen that a powerful and general tool for mesh adaptation is the theory of harmonic map (see also appendix B). Immediately one can think to use this general framework to map from the computational manifold with its own metric to the physical mesh which lie on a sphere (in the same fashion as we where considering, in two dimensional case, the Euclidean plane):

$$A : \mathcal{M} \rightarrow \mathcal{S}^2_{\mathcal{A}} \quad \boldsymbol{z} = A(\boldsymbol{\zeta}, t), \quad \boldsymbol{J}_A = \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\zeta}}$$

Unfortunately such a mapping violates the hypothesis of the HYS theorem: the positive curvature of a sphere makes harmonic map not unique. Keeping in mind that we will deal with problems at large scale *but not at global scale* we can hope to simplify the problem. Weller *et al.* [2016] for example suggest that a naif idea could be to project the mesh on a plane, adapt on the plane with a certain MMPDE from the ones seen in chapter 3 and finally project back to the sphere. The same authors, which were always interested in global circulation models, were then forced to move away from this because the problem was moved on how to impose proper boundary conditions to the mesh node position, for examples at the poles. However, *if one is not interested in moving the mesh around the poles*, this idea remains valid.

We assume that the computational domain is described by a certain parametrization of the sphere $\boldsymbol{\zeta}$ with metric components γ_{ij} and square root of the metric tensor $\sqrt{\gamma}$. We consider then following direct transformation from the sphere to a two dimensional Euclidean plane:

$$A : \mathcal{S}^2_{\mathcal{A}_0} \rightarrow \mathbf{E}^2 \quad \boldsymbol{z} = A(\boldsymbol{\zeta}, t), \quad \boldsymbol{J}_A = \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{\zeta}}$$

We can extend the functional of Ceniceros and Hou [2001] with minor modifications (recall the ALE notation $\boldsymbol{\zeta} = \zeta^i \boldsymbol{\gamma}_i$)

$$\begin{aligned} E(\boldsymbol{z}) &= \frac{1}{2}\int_{\Omega_\zeta} \omega \boldsymbol{\gamma}^i \frac{\partial z^\alpha}{\partial \zeta^i} \cdot \boldsymbol{\gamma}^j \frac{\partial z^\alpha}{\partial \zeta^j} \sqrt{\gamma} d\boldsymbol{\zeta} \\ &= \frac{1}{2}\int_{\Omega_\zeta} \omega \gamma^{ij} \frac{\partial z^\alpha}{\partial \zeta^i} \frac{\partial z^\alpha}{\partial \zeta^j} \sqrt{\gamma} d\boldsymbol{\zeta} \end{aligned} \tag{4.52}$$

to which the following Euler-Lagrange equations or MMPDE is related

$$\frac{1}{\sqrt{\gamma}} \frac{\partial}{\partial \zeta^i} \left(\omega \sqrt{\gamma} \gamma^{ij} \frac{\partial z^\alpha}{\partial \zeta^i} \right) = 0 \tag{4.53}$$

This will be our MMPDE on the sphere. It consist, as in the two dimensional case, of a decoupled system of PDEs. Minor modification in the algorithm must be done to implement the Laplace-Beltrami operator, see (A.13), instead of the Laplacian. We recall also that $\sqrt{\gamma}$ and γ_{ij} are defined in the computational domain (greek letters): they don't depend from that physical configuration. The only non-linearity is associated to the monitor function $\omega = \omega(\boldsymbol{z})$ which depends on the evolving underlying solution.

4.8.1 Mesh generation

The moving mesh algorithm defined by the sequence of iterations (3.16) can be used to approximate our new MMPDE (4.53). We remark that, as done for the CFD, we do not solve the MMPDE in the polar regions. Roughly speaking, points will not move at the poles.

We extend the benchmarks proposed in Tang and Tang [2003] and tested in section 3.4.5 to the sphere. The monitor function is computed according to $\omega = \sqrt{1 + \alpha u^2}$, with u assigned:

$$u(\psi, \lambda) = \exp(-8(4\psi^2 + 9\lambda^2 - 1)^2) \tag{4.54}$$

$$u(\psi, \lambda) = 50 \exp(-250(\psi^2 + \lambda^2)) \tag{4.55}$$

with $\psi = z^1$ and $\lambda = z^2$. The iteration is repeated in the pseudo-time loop until convergence is reached. For the first smooth example reported on the left of figure 3.5 we have set $\alpha = 100$. In the left picture of figure 4.6 we show that also singularities are well handled with a proper choice of the parameter.

4.9 Numerical experiments

4.9.1 Global steady state zonal geostrophic flow

Case #2 from Williamson test suite Williamson *et al.* [1992] is a steady state solution of (4.2) which allow to measure the order of accuracy of numerical

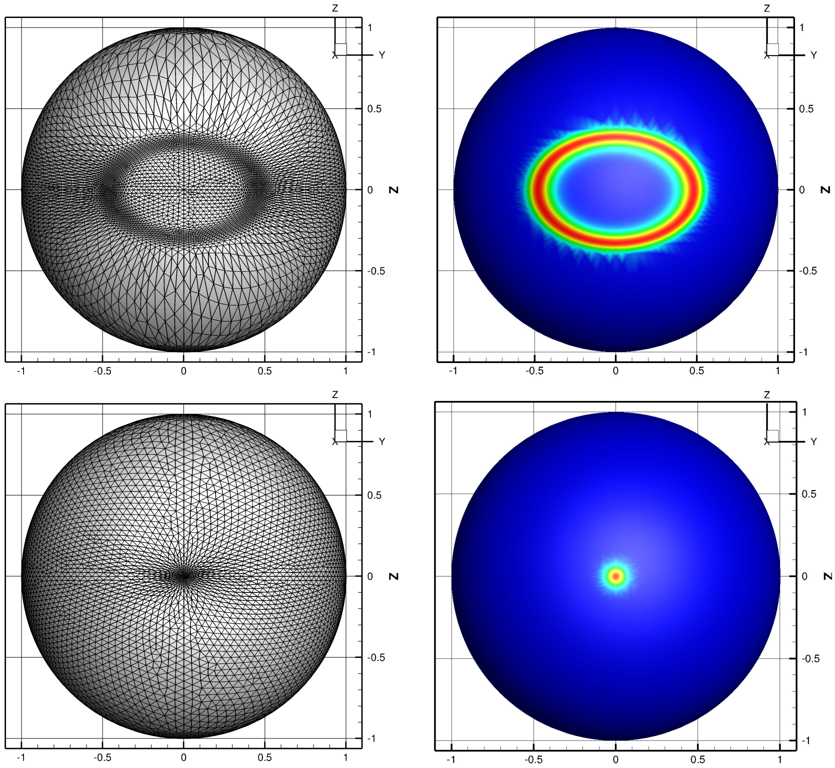

Figure 4.5: Top) test (4.54). Bottom) test (4.55)

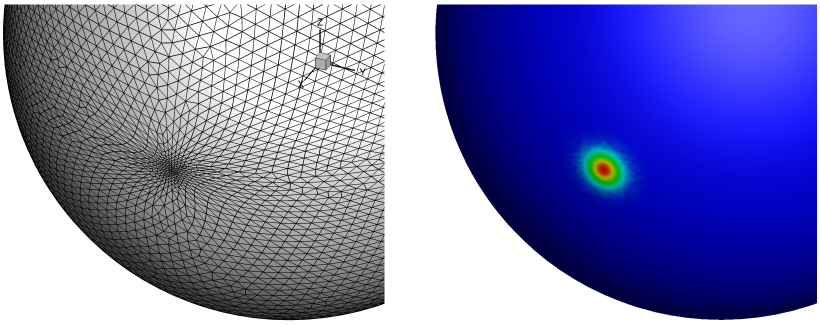

Figure 4.6: Zoom of test (4.55)

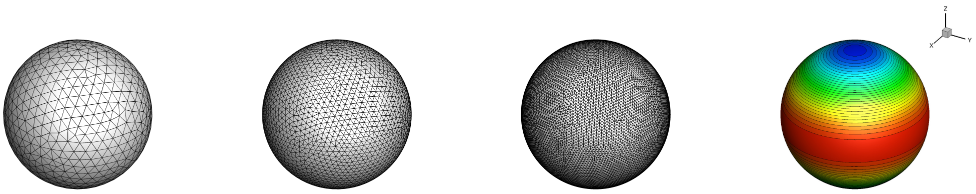

Figure 4.7: Steady zonal flow. Left: mesh 1-2 and 3. Right: fluid depth isolines.

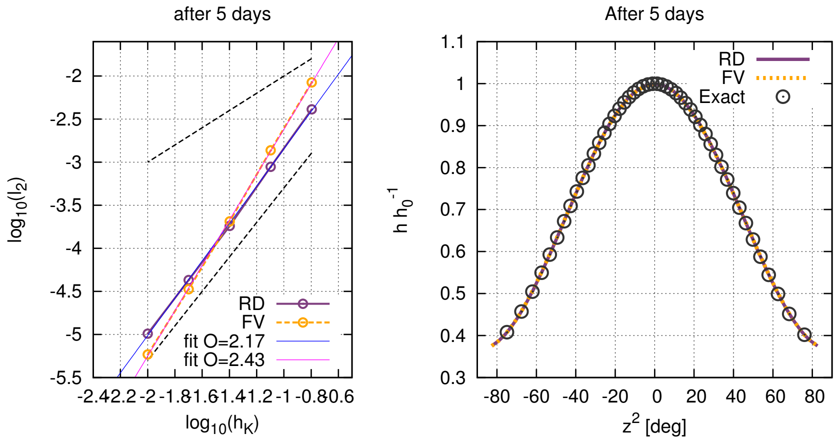

Figure 4.8: Steady zonal flow. Left: L2 error against mesh size. Right: fluid depth profile along a meridional cut and the corresponding exact solution.

schemes in spherical coordinates and in presence of Earth rotation. It consists of a zonal flow with the corresponding geostrophic height field. Typically several rotation orientations are specified. In our case, in order to avoid flow entering/leaving the pole regions, only the case where the axis of rotation coincides with the z-axis will be simulated. The velocity and height fields are initially (and for all time) given by:

$$\begin{aligned} h(\boldsymbol{z},0) &= h_0 - \frac{1}{g}\left(\Omega R\, u_0 + \frac{u_0^2}{2}\right)\sin z^2 \sin z^2 \\ u^1(\boldsymbol{z},0) &= u_0 \cos z^2 \\ u^2(\boldsymbol{z},0) &= 0 \end{aligned}$$

with $gh_0 = 2.94 \times 10^4\,[m]^2[s]^{-2}$ and $u_0 = \frac{2\pi R}{12 days}$. Simulation are performed on 5 unstructured grids, halving the mesh size. The meshes are generated always with the meshing software `Gmsh` Geuzaine and Remacle [1997-2017], with the frontal algorithm (`Mesh.Algorithm=6`). In figure 4.7 we can see the mesh topology for the first three levels of mesh refinement. The error norm is computed following Williamson *et al.* [1992]:

$$||e||_{L2} = \sqrt{\int_0^{2\pi}\int_{-\pi/2}^{\pi/2} \left(h(\boldsymbol{z}) - h_{ex}\right)^2 \sqrt{G}\, d\boldsymbol{z}}$$

In figure 4.11 we report the dimensionless error against the mesh size, after 5 days of computations, as specified by always in Williamson *et al.* [1992]

$$l_2 = \frac{||e||_{L2}}{||h_{ex}||_{L2}}$$

For this steady case we have run only fixed mesh computations. We observe that the error is of the same order of magnitude for FV and RD. For both schemes it decreases with more then second order of accuracy, this being particularly evident for FV.

4.9.2 Advection of cosine bell

We revisit test case #1 from the Williamson test suite Williamson *et al.* [1992] in order to assess numerical accuracy for unsteady problems and in presence of mesh movement. A cosine bell is transported once along the equator ($R = 6371220$, $g = 9.80616$ and $\Omega = 0$):

$$\begin{aligned} h(\boldsymbol{z},0) &= \left[\begin{array}{ll} \frac{h_0}{2}\left(1+\cos\frac{\pi r}{\pi/3}\right)+h_0 & \text{if } r < \frac{\pi}{3} \\ h_0 \quad \text{otherwise} & \end{array} \right. \\ u^1(\boldsymbol{z},0) &= u_0 \cos z^2 \\ u^2(\boldsymbol{z},0) &= 0 \end{aligned} \tag{4.56}$$

where $h_0 = 1000$ and $u_0 = \frac{2\pi R}{12 days}$. r is the great circle distance between $\boldsymbol{z}$ and the center of the cosine bell $\boldsymbol{z}_C$:

$$r = R \arccos\left(\sin z_C^2 \sin z^2 + \cos z_C^2 \cos z^2 \cos(z^1 - z_C^1)\right)$$

In general several orientations for the wind are specified but so far we have tried only the zonal configuration to avoid interaction with polar regions. For this test, Williamson propose to test the advective components in isolation. We prefer to test all the SWEs using the technique of manifactured solution. See appendix E for the derivation of the source term to be added to the SWEs in order that (E.1) is an exact solution. Moreover, with respect to Williamson original solution we have translated the initial height of h_0 to avoid the appearance of dry regions. Simulation are performed on 5 unstructured grids generated with the meshing software `Gmsh`, halving the mesh size. In figure 4.9 there is an example of the grid topology for the first three levels of mesh refinement. With respect to the steady case, the order of accuracy fall below

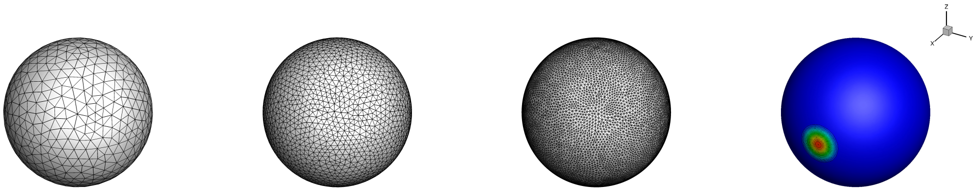

Figure 4.9: Advection of cosine bell. Left: mesh topology. Right: fluid depth isolines.

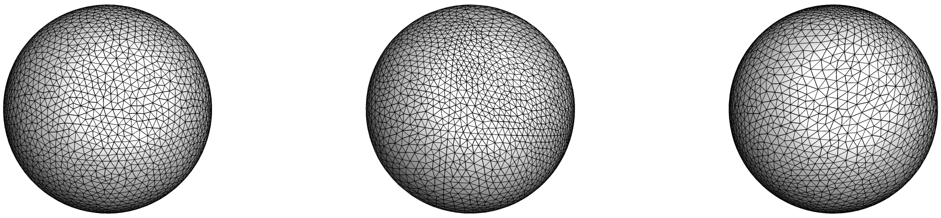

Figure 4.10: Advection of cosine bell. 2nd level of mesh refinement: original mesh and transformed meshes at two different instants.

the second order. RD seems to be slightly more accurate in terms of global error and order of accuracy (1.88 against 1.78), see figure 4.11.

To test the accuracy of the ALE scheme we have added an unsteady perturbation to the position of mesh nodes, see figure 4.10. They will be mapped according to:

$$\begin{aligned} z^1(t) &= \zeta^1 + 0.3\cos(\zeta^1)\cos(\zeta^2)\sin\left(\tfrac{4\pi t}{T}\right) \\ z^2(t) &= \zeta^2 + 0.3\cos(\zeta^1)\cos(\zeta^2)\sin\left(\tfrac{4\pi t}{T}\right) \end{aligned}$$

Always in figure 4.11, with dashed lines are reported the convergence lines for the ALE scheme. We observe that, for both RD and FV, the unsteady and arbitrary mesh perturbation does not spoil the order of accuracy.

4.9.3 Circular hump on a non rotating sphere

We consider the test contained in Rossmanith *et al.* [2004]. For the initial conditions we place a “circular” depth disturbance at the equator:

$$\begin{aligned} h(\boldsymbol{z},0) &= \begin{cases} 2 & \text{if } \arccos(\cos(z^1)\cos(z^2)) \leq 0.2 \\ 0.2 & \text{otherwise} \end{cases} \\ \boldsymbol{u}(\boldsymbol{z},0) &= 0 \end{aligned}$$

This initial condition is symmetric about the point $(z^1, z^2) = (0°, 0°)$. Therefore, the solution should remain symmetric for all time. We will use two

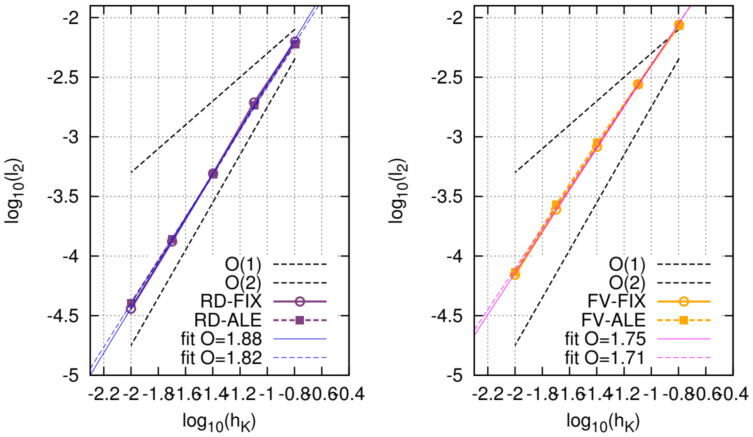

Figure 4.11: Advection of cosine bell. Left: convergence for RD. Right: convergence for FV.

semi-spherical grids: a coarse one with 7122 grid points and 14034 elements; a fine one with 39699 and 78900 elements. Isolines (only for FV) are plotted in figure 4.12 for the coarse and the fine simulation (first two rows), showing a good shock capturing and *good symmetry in all radial directions.*

Then the coarse mesh is used as the computational domain for a moving mesh ALE simulation. The moving mesh parameter are $\alpha = 50$ and $\beta = \gamma = 0.15$ for RD and $\alpha = 50$ and $\beta = \gamma = 0.3$ for FV. In figure 4.12 (last row) we can observe that the meshes exhibit a strong adaptation on the shocks and also a good symmetry in the radial direction. As a consequence the ALE simulation provides a good resolution of both the inner and the outer shock, of the order of simulation run on the fixed fine mesh.

In figure 4.13 we show some cuts of the fluid depth along the equator. In the same figure our FV and RD implementations are also compared against the numerical solution of Rossmanith *et al.* [2004] which consists in a high-resolution FV (Lax-Wendroff flux with MC limiter) on a Cartesian grid composed of 34680 points. From the cuts and from the table 4.1 reporting the CPU times, we can conclude that the moving mesh method is effective. We can observe that the shock resolution improves with respect to the fixed coarse mesh and we also see a consistent CPU time reduction with respect to the fixed fine mesh (58% for RD and 65% for FV).

4.9.4 Circular hump on a rotating sphere

This test case is also taken form Rossmanith *et al.* [2004]. It is an extension of the previous one in presence of the sphere rotation. The dimensionless rotation rate is $\Omega = 5$. We have run this test with two semi-spherical grids: a coarse one with 8361 nodes and 16488 elements and a fine one with 46521 nodes and 92488

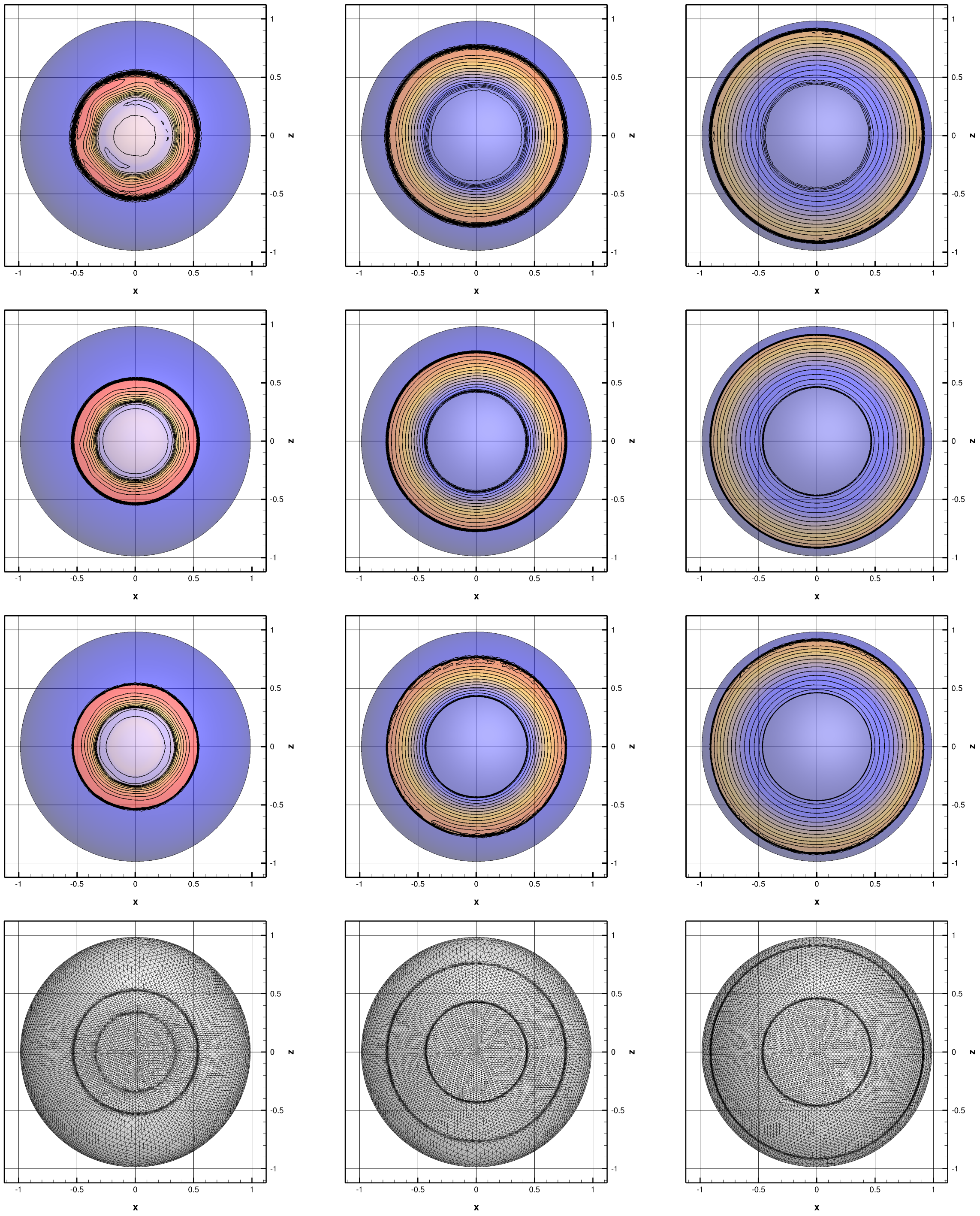

Figure 4.12: Circular hump on a sphere. Snapshots of FV depth isolines (20 levels between 0 and 0.55) at different times, $t = 0.3, 0.6, 0.9$. Top) coarse mesh. Middle: fine mesh. Bottom: adaptive ALE with adaptive mesh.

ALG	Mesh (Nodes)	RD[s] (%MMPDE)	FV[s] (%MMPDE)
FIX-COARSE	7122	34.58	34.70
FIX-FINE	39699	485.37	593.52
ADAPT-ALE	7122	201.05 (37.5%)	203.10 (38.0%)
Rossmanith	34680	-	

Table 4.1: Circular hump on a non rotating sphere. CPU times.

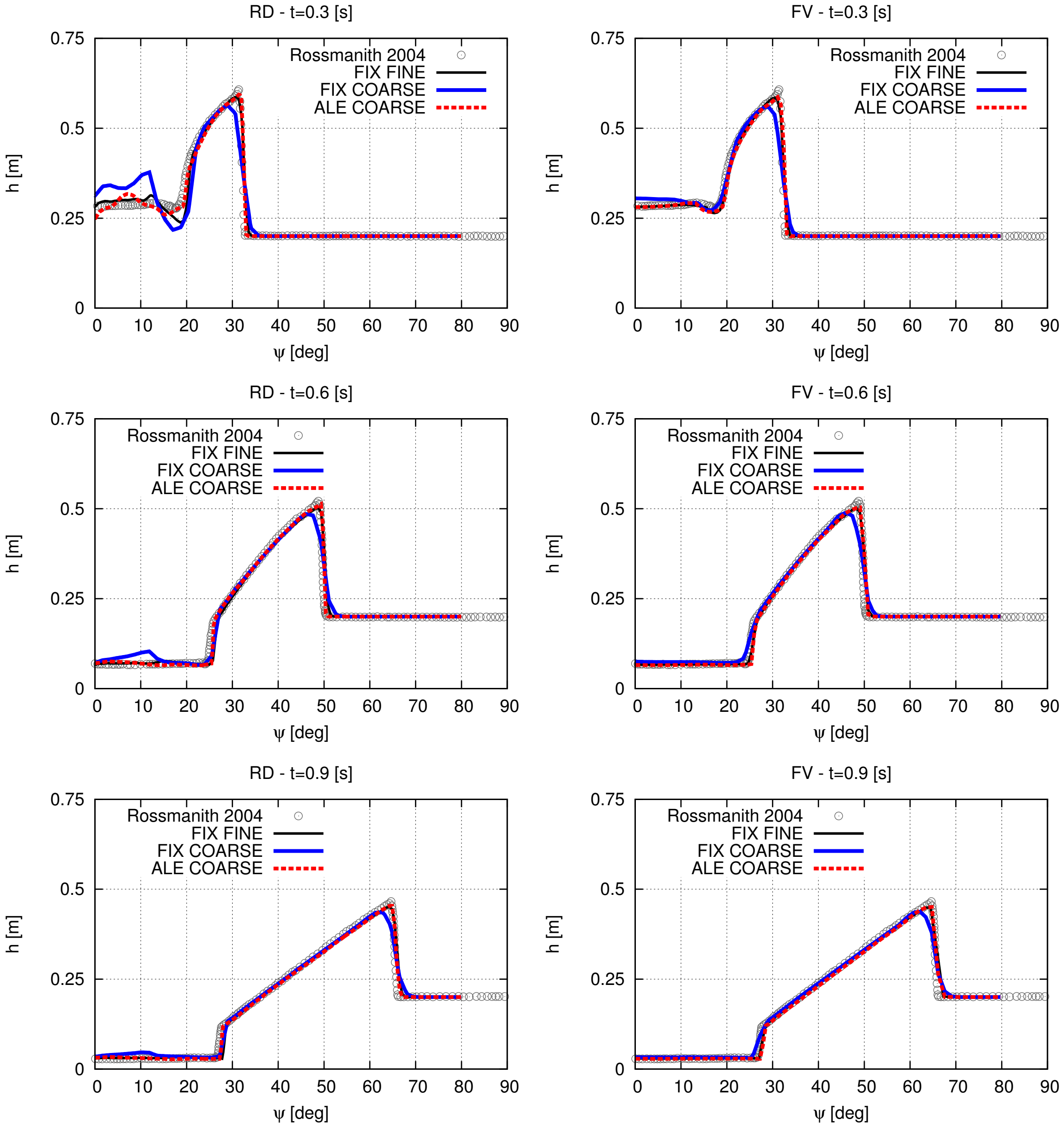

Figure 4.13: Circular hump on a sphere. Solution along a symmetry line. Comparison between fix mesh simulations and adaptive ALE.

elements. As before the coarse mesh represents the computational domain for an moving mesh ALE simulation. The moving mesh parameter are $\alpha = 50$ and $\beta = \gamma = 0.1$. In figure 4.14 we report snapshots of the depth isolines and of the adaptive meshes. This times mesh nodes are clustered both around the shock waves and also near some smoother feature where high gradients appear. We remark that, for this case, FV gives better and smoother result with respect to RD where some oscillations appear (visible in figure 4.14). This is probably related to a non proper activation of the limiter and it subject of current investigation.

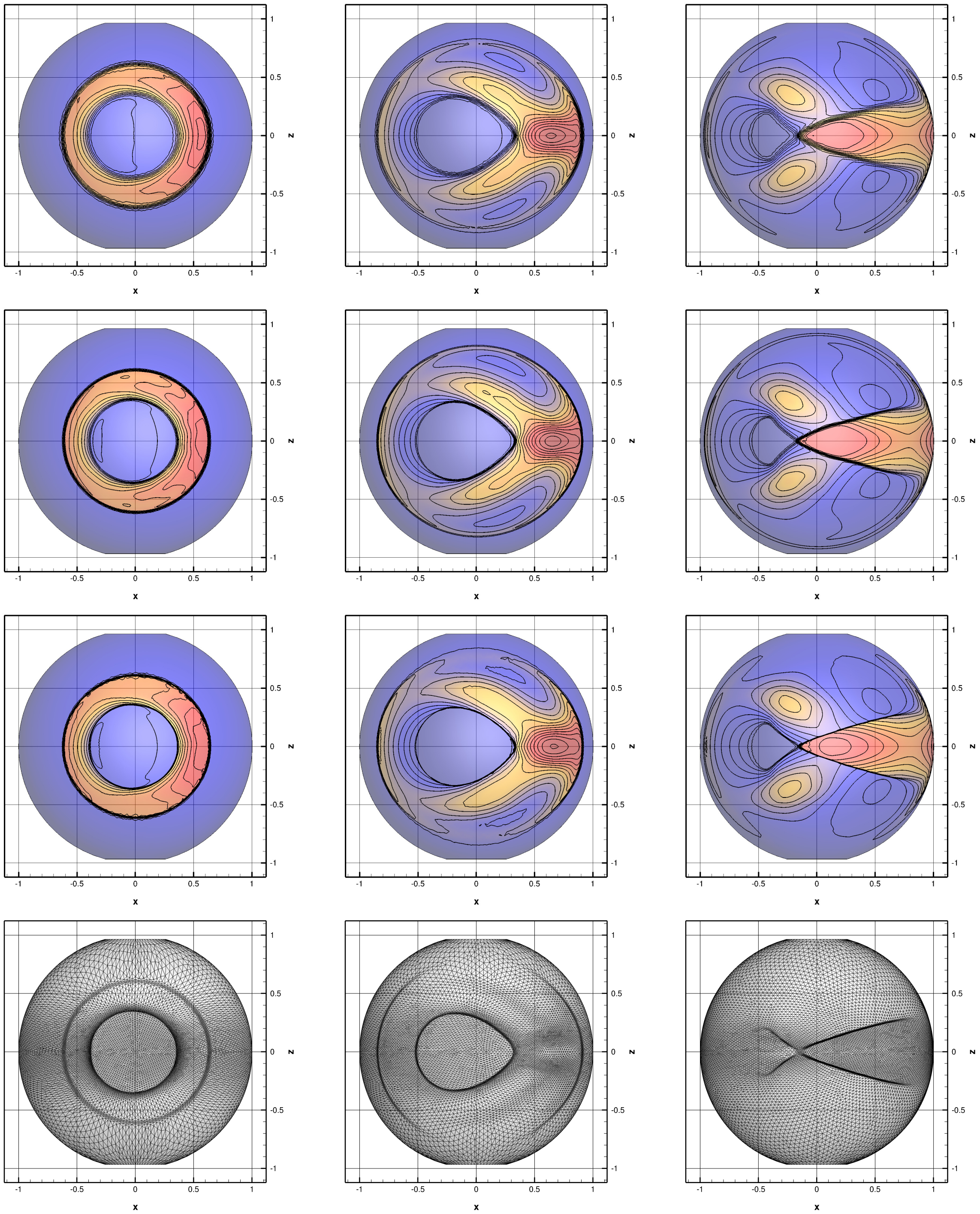

Figure 4.14: Circular hump on a sphere. Snapshots of FV depth isolines (20 levels between 0 and 0.55) at different times, t = 0.4, 0.8, 1.2. top) coarse mesh. Middle: fine mesh. Bottom: adaptive ALE.

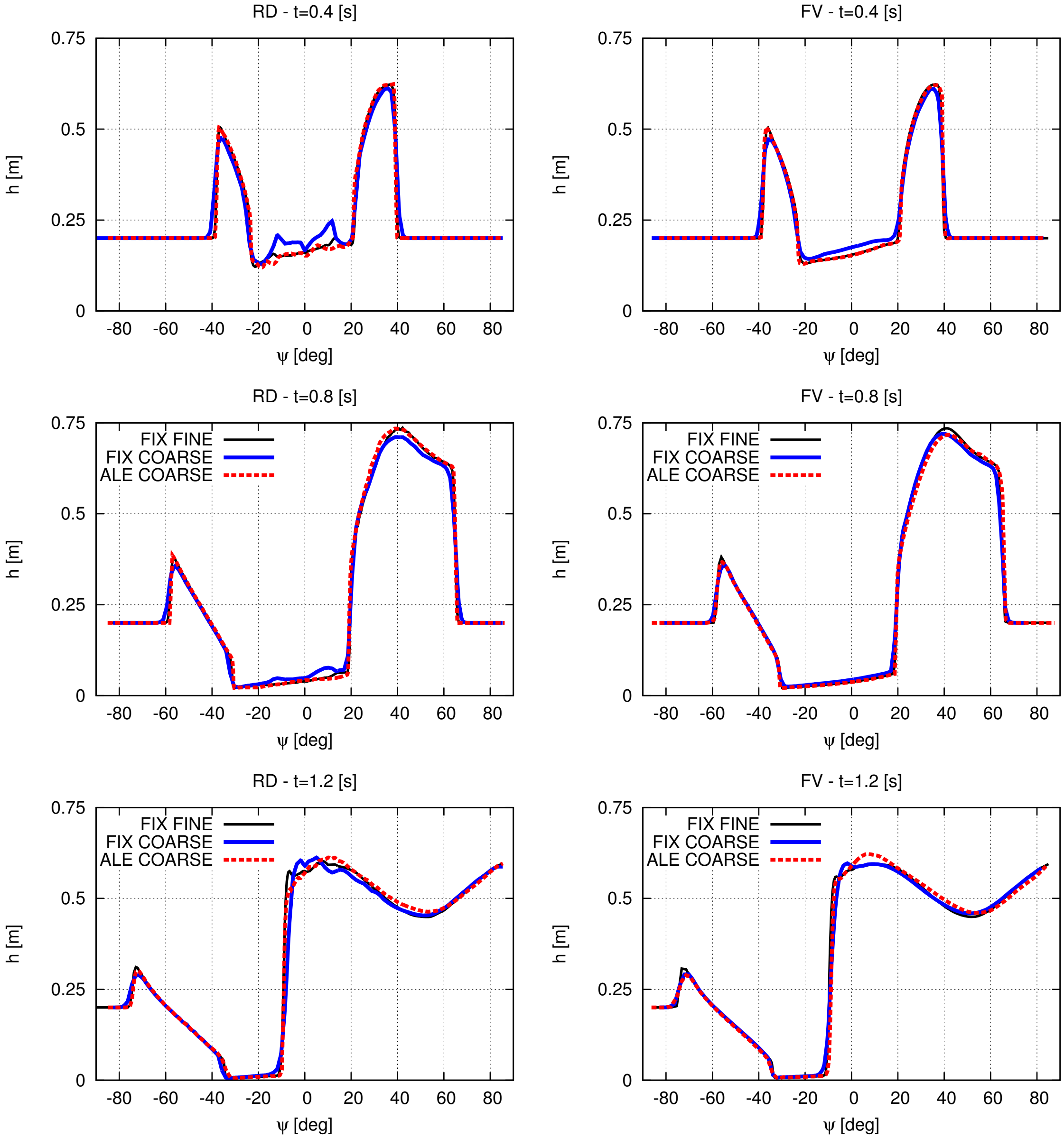

Figure 4.15: Circular hump on a sphere. Solution along a symmetry line. Comparison between fix mesh simulations and adaptive ALE.

Chapter 5

Tohoku-Honsu tsunami simulation

The 11 March 2011 a catastrophic earthquake of magnitude $M_w 9.0$ ruptured the plate off the coast of the Tohoku region. This is usually referred to as the 2011 Tohoku-Honsu earthquake. Many authors that have studied the fault rupture mechanism agree that the earthquake was compact with huge plate slips concentrated near the trench axis, see Fujii *et al.* [2011]; Shao *et al.* [2011]; Ammon *et al.* [2011]; Iinuma *et al.* [2011]; Satake *et al.* [2013]. We will give a definition of the slip vector in 5.1.1, for now see figure 5.1 for an illustration of its physical meaning. Maximum slips may vary from one author to the other but they stay in the range $30 - 70\,[m]$. The fact that large slips occurred near the trench, at rather low depths, produced a large seafloor uplift that in turn triggered a devastating tsunami wave. At the end, significant damages occurred in the coastal prefectures of Fukushima, Myiagy and Iwate and they were mostly caused not by the earthquake itself, but by abnormal wave heights (up to $40\,[m]$ in northern Japan) that overwhelmed protection barriers, breakwaters and other sea defenses.

Very interestingly for us, the Tohoku-Honsu earthquake was the largest instrumentally recorded earthquake in Japan history. A part from extensive ground motion dataset, the tsunami was recorder by 10 nearshore GPS gauges, 32 tidal gauges and by the offshore DART buoy system. All these data are available online: for the GPS wave gauges data one can access the website of the Nationwide Ocean Wave information network for Ports and Harbours (NOWPHAS) NOWPHAS [2017]; the DART buoy data are available from the website of the National Oceanic and Atmospheric Administration (NOAA) NOAA [2017a]. Equally important was the post-tsunami field survey which have been conducted meticulously along the coast of Japan. Inundation areas as long as runup and tsunami height[1] from *in situ* measurement, are reported

---

[1] Hereafter for the tsunami terminology we refer to the NOAA [2017b]:
Inundation area: an area that is flooded with water.
Runup: Maximum height of the water onshore observed above a reference sea level. Usually measured at the horizontal inundation limit.

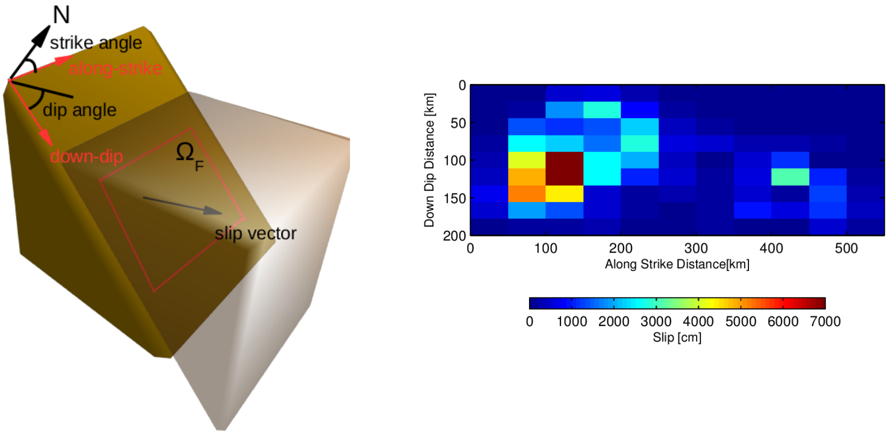

Figure 5.1: Left) Fault geometry and nomenclature. Right) Approximation of the fault plane in subfaults and approximated slip field, (to generate this image we have used the software `SlipReal` Mai and Beroza [2002])

for example in Mori and Takahashi [2012]. This big amount of data helped researchers in better understanding the phenomenon. For instance, data were used to recover initial waveform; this inverse problem is explained in section 5.1 with all the physical hypotheses that are introduced. Both buoy data and and field survey are also very useful to test the capability of hydrodynamics codes in modeling the main physical features of a tsunami event. In section 5.2 and 5.3 we compare the accuracy of the numerical simulation against observations.

5.1 Initial waveform

The selection of the initial waveform is crucial to obtain realistic wave heights and runups. First, the waveform depends on the fault's rupture dynamics. Then the rupture pulse produces seismic waves which propagate to the seafloor. Finally the seafloor displacement generates wave motion in the ocean up to the free surface. The resulting waveform represents the initial condition for the SWEs code. As we see the phenomenon is complex, involving non linear wave propagation in different media. Typically, in the context of tsunami modeling, it can be simplified using many physical assumptions.

5.1.1 Fault rupture model

First of all we need to understand how to properly account for the rupture mechanism. The rupture geometry is described in the sketch in figure 5.1. We

---

Tsunami height: the height, relative to a stated reference level, to which a particular location at certain time instant is covered by water.

can see two plates moving w.r.t one another having the fault surface Ω_F as interface; the movement is characterized by

- two angles: the strike angle formed between the nord and the alongstrike direction and the dip angle formed by the horizontal and the downdip direction
- an heterogeneous field of displacement vectors contained in the fault plane. The relevant quantity is the module of the slip vector which is simply called the slip and for us is $d(\boldsymbol{x}, t) \in \mathbb{R}$.

Fault's rupture is a complex phenomenon which depends from pre-rupture stress condition, geometrical setting and frictional property of the fault which are largely unknowns. To simplify the problem we approximate the scalar slip field: the fault plane is subdivided in N_F rectangular subfaults, each one with its slip defined at the center $d_i, \quad i = 1, N_F$, see the right picture of figure 5.1. If the dynamics of the rupture is taken into account, the time domain is also approximated and subdivided in multiple windows allowing a time dependent activation of the subfaults. The approximated slip field d_i is also called *source model*. The computation of the slip field d_i from available time dependent data (both seismometeres and tsunamometers) is referred to as the inversion problem. This inversion problem could be very hard to solve and strongly depends on the algorithm used and on the data, leading to quite different slip fields for the same earthquake/tsunami event. For the 2011 Tohoku tsunami we have assisted to a proliferation of source models among which is quite difficult to select a better one. Moreover the choice of the specific inversion is determinant to reproduce accurately wave heights and runups. This issue was addressed recently in MacInnes *et al.* [2013] and in the TANDEM project. The source model of Satake *et al.* [2013] emerged as one of the best for matching both nearshore and offshore height records as well as inundations.

Satake used 55 subfaults, each 50 $[km]$ long and 50 $[km]$ wide, only the shallower column of subfaults having an halved width of 25 $[km]$. The strike and slip angles are adopted from USGS W-phase inversion result (strike 193°, slip angle 81°) while the dip angle varies along the depth from 8° for the shallowest subfaults to 16° for the deeper ones. It has been realized that, for great earthquakes, the rupture propagates over large distances in a finite time, in the order of a few minutes. To consider these time dependent effects in the inversion, Satake *et al.* [2013] used a delayed rupture mechanism in such a way that, at the end, a temporal variation of the slip field was included. 5 time-windows were considered: every subfault activates at time $m\Delta t + t_d$. The time interval was set to $\Delta t = 30\,[s]$ based on physical observations; $m = 0, 1, 2, 3, 4$ is an integer that represents each time-window and $t_d = |x_i|/V_r$ is a delay which depends on the rupture velocity V_r and on the distance $|x_i|$ of the subafault's center from the hypocenter. Satake computed 275 unknowns values of slip (55 subfaults to be computed at each of the five time windows) based on tsunami waveform data (GPS buoys, tidal gauges and DART buoys). Practically, the slip distribution obtained with this inversion algorithm is sampled every 30 $[s]$, from 0 $[s]$ (the time origin is 14:46:18 Japan Standard Time) until 300 $[s]$ when the rupture mechanism stops. The result of the inversion can be found in the original paper Satake *et al.* [2013] and we don't report them here. For clarity, at each sampling time, the slip looks like the right picture of figure 5.1.

5.1.2 Okada dislocation theory

Seismic wave propagation in principle can be performed with modern Finite Elements techniques. In the tsunami community what is commonly done is to work within the following simplifying hypotheses:

1. the earth is an elastic material
2. wave propagation is neglected and one computes an instantaneous equilibrium configuration.

The problem reduces to a classical *dislocation problem* van Zwieten *et al.* [2013]: an elastic body contains a dislocation plane (fault plane) on which a discontinuous displacement instantaneously occurs. The question is which is the equilibrium configuration of the body after this discontinuous initial condition has been imposed. An analytical solution exists for a finite rectangular dislocation plane immersed in a homogeneous half space (the domain boundary consists of a flat surface and a far field boundary which is at rest; isotropy and homogeneity of the material are considered). This is the celebrated Okada solution Okada [1985]. For earthquakes, the interesting quantity is the displacement at the flat boundary which represents the seafloor. The great advantage of this approach is that we can easily implement the subfaults configuration of Satake. Due to linearity, we superpose at each point of the seafloor the solution related to each subfault displacement, as if it was acting independently from the others. In this way, the slip distribution interpolated every 30 $[s]$ is used as initial condition for the Okada analytical expression to compute the seafloor displacement.

5.1.3 Free surface motion

The free surface displacement is assumed to be identical to the static *vertical displacement of the seafloor*, basically as if the water would translate instantaneously. This rough assumption is classically justified by several observations (see the review in Dutykh *et al.* [2011]): since tsunamis are long waves, the vertical dynamics can be neglected 2) elastic waves are much faster then gravity waves (in the order of ten times faster), as a consequence the seafloor

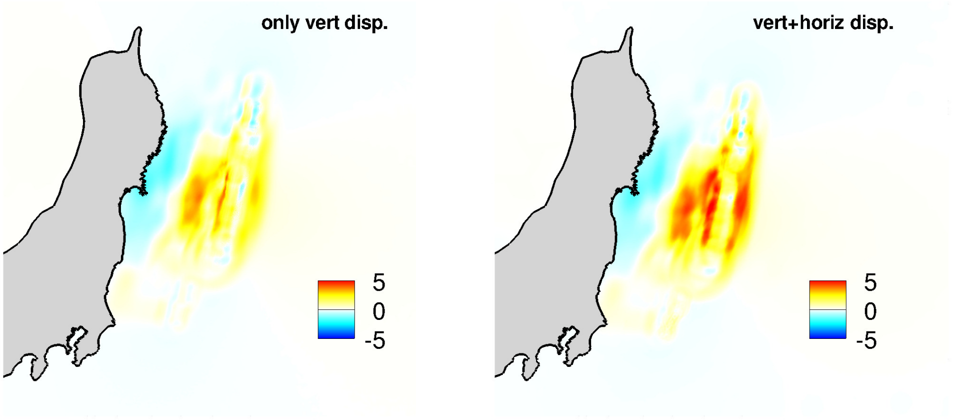

Figure 5.2: Waveforms in $[m]$ at $t = 300\,[s]$ used as initial condition for the SWEs code and computed accounting for: left) only vertical seafloor displacement Right) both horizontal and vertical seafloor displacement.

deformation is assumed to be instantaneous 3) the bathymetry has in general mild slope, which means that the effect of horizontal seafloor deformation is negligible for tsunami generation.

For the 2011 Tohoku-Honsu earthquake the horizontal seafloor displacement was large near the trench axis. If large horizontal seafloor displacements occur where the bottom present a strong slope, such as in correspondence of the trench, the tsunami wave can be amplified by 30% Tanioka and Satake [1996]. For this reason we have accounted also for the horizontal seafloor displacement into the computation of the waveform. A partner of the TANDEM project, the Bureau de Recherche en Géologiques et Miniérs (BRGM) provided us the free surface adding the horizontal displacement to the waveform.

Then the initial condition for the SWEs code is generated as follows. Starting from the time origin, we impose the computed seafloor displacement at $t = 0\,[s]$ as initial condition for the SWEs code. This initial condition is propagated and at every sampling time $(30\,[s])$ the waveform related to the interpolated slip is superposed to the propagating wave. Progressing this sequence of wave propagation and superposition of free surface displacement, we obtain the waveform at $t = 300\,[s]$ which is then propagated up to the coast. The initial waveforms at $t = 300\,[s]$ are reported for completeness in figure 5.2 (with and without horizontal displacement).

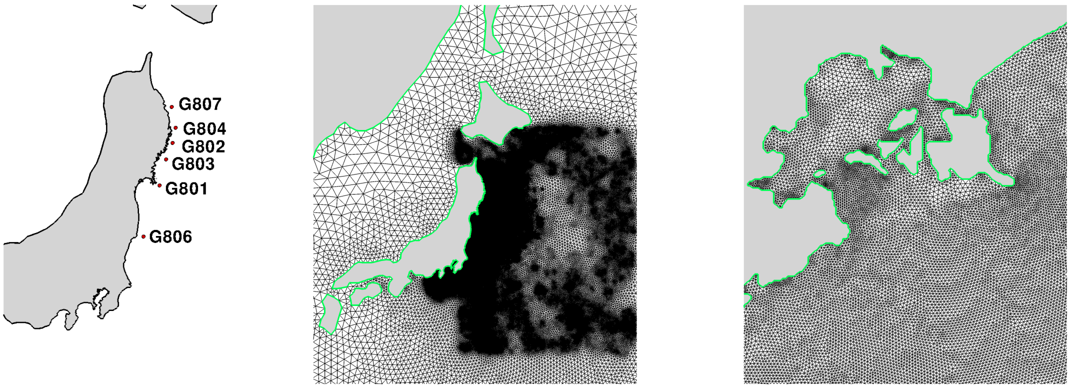

Figure 5.3: Left) GPS buoy position. Right) Mesh topology. zoom of the Japan coast and zoom of the Sendai bay.

5.2 Results with fitted mesh

We present the numerical simulation of the Tohoku-Honsu tsunami with the Residual Distribution and the Finite Volume schemes developed in previous chapters. The initial waveform does not include horizontal displacement, thus it is the one of the left picture in figure 5.2. A body fitted mesh was created by Electricité de France (EDF) and provided to us within the partnership of the TANDEM project. EDF made available also the results of the numerical SWEs simulation that they run with the software `Telemac2D`. This is a Finite Element hydrodynamics software developed at EDF and freely available online, the reader may refer to Hervouet [2007]. The details concerning this `Telemac2D` tsunami simulation can be found in Legal [2017]. This will allow a comparison, for the same set-up of the Tohoku-Honsu event, of different SWEs codes which was one of the main objective of the TANDEM project.

The mesh has 1080181 elements and 547469 nodes with a strong variation of the local mesh size that ranges from $100000\,[m]$ to $120\,[m]$ in the coastal region, where the maximum resolution is necessary to correctly reproduce the complex Japan's boundary. In figure 5.3 the reader can appreciate the mesh topology. The mesh is adapted to the bathymetry variation and to the distance to boundaries. Reflective boundary conditions are imposed everywhere along the coastline.

In figure 5.4 the simulated sea level displacement is compared against experimental data recorded by the GPS buoys, see figure 5.3 to check the position of the GPS buoys. *The three numerical schemes provide similar free surface levels.* We can see in particular that both our RD and FV implementations match well with the results produced by `Telemac2D`. We can underline that

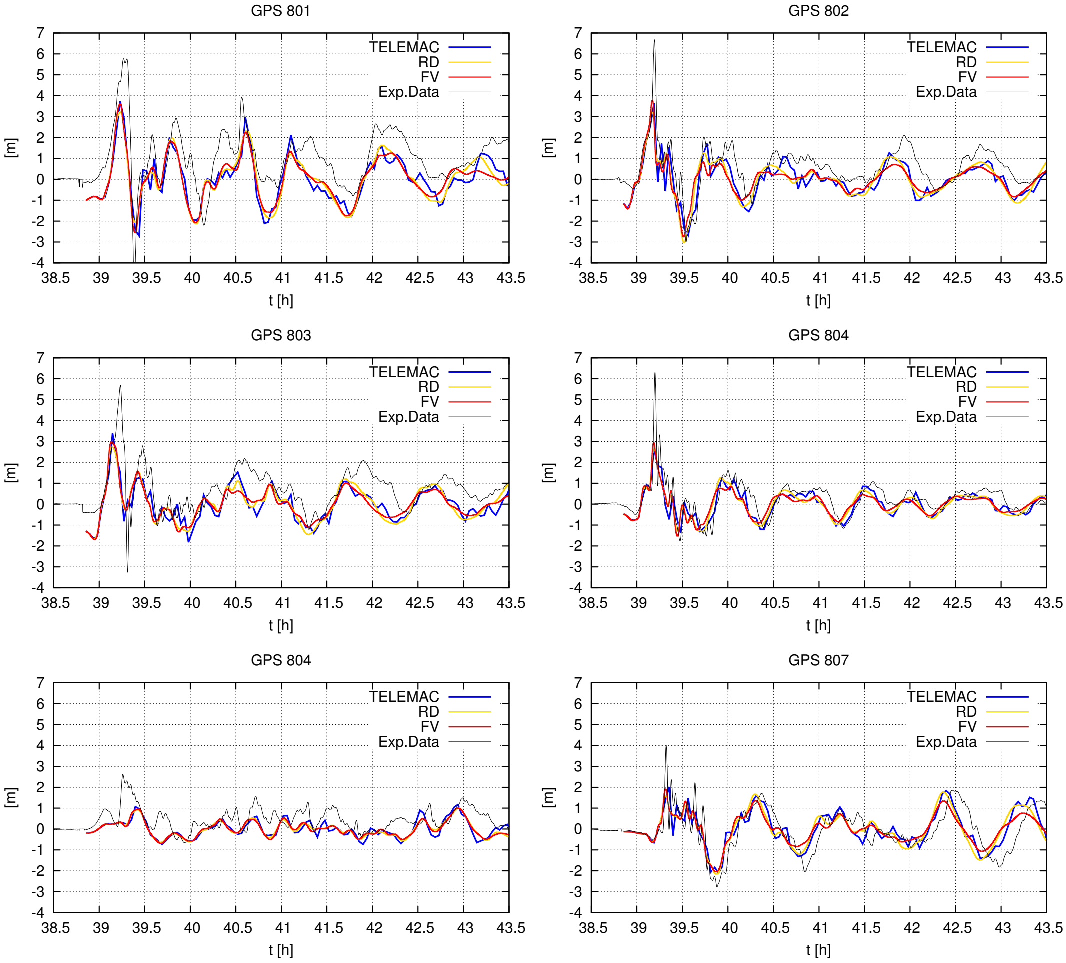

Figure 5.4: Comparisons of the simulated and observed sea level displacement recorded at GPS buoys. The simulated curves refer to FV, RD and `Telemac2D`.

RD scheme is slightly less dissipative then FV, providing better resolved secondary peaks. The tsunami simulation agrees quite well with experimental data, with an accurate computation of the arrival time of the leading wave and well reproducing the complex interaction between incoming and trapped waves. In figure 5.5 and as observed by Chen *et al.* [2014], we see that the elevation wave, once reflected, undergoes a complex transformation: reflected waves travel parallel to the coast from north to south and from south to north or focus to form higher waves such as it happens in the Sendai Bay (see the red area in the last picture of 5.4). Always from figure 5.4 we clearly see that all *the numerical simulations largely underestimate the first elevation wave peak*, up to 3.5 $[m]$ error with respect to the observations at the GPS802. Aside form this drawback we can conclude that the Shallow Water model reproduce a realistic scenario for the Tohoku event. The tsunami wave was a weakly non-

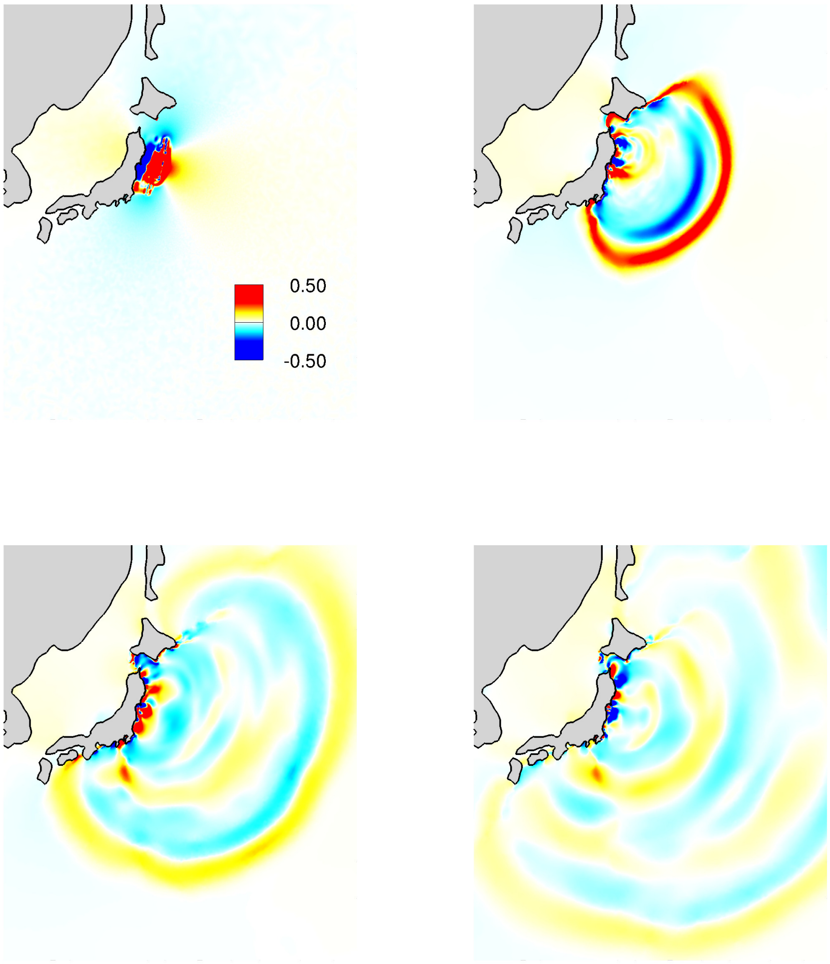

Figure 5.5: Snapshot of the simulated offshore (RD) free surface level in meters every one hour starting from time 14:51:18: $t = $ 0:00, 1:00, 2:00 and 3:00.

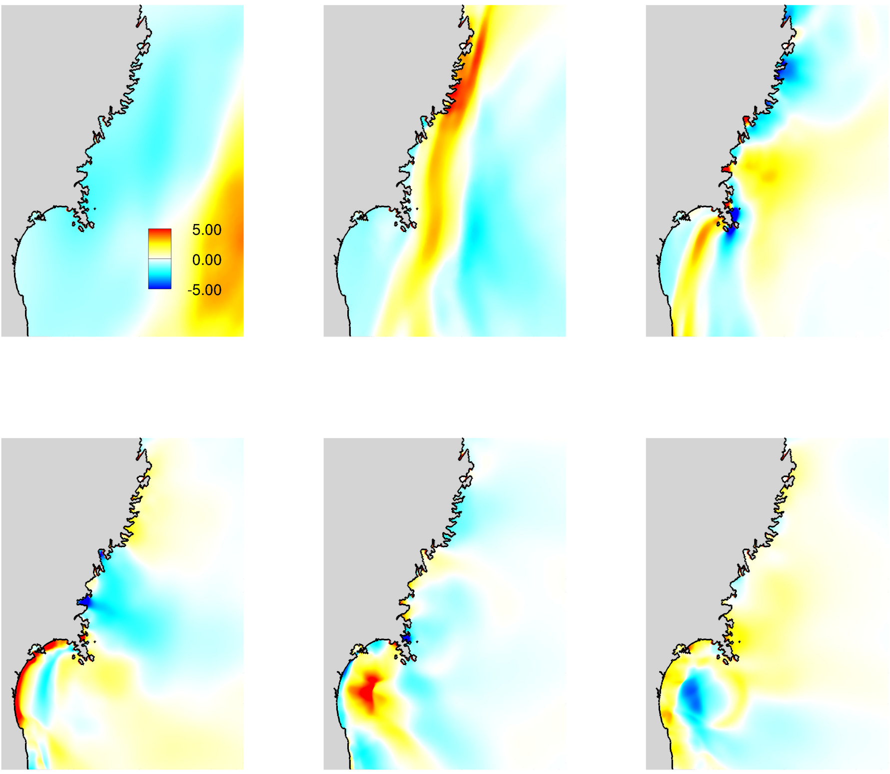

Figure 5.6: Snapshot of the simulated nearshore (RD) free surface level in meters every twenty minutes starting from time 14:51:18: t = 0:00, 0:20, 0:40, 1:00 and 1:20.

linear and weakly dispersive very long wave. It passes from the linear regime in deep water (figure 5.5) suddenly to the shallow water regime (figure 5.6), across the very steep shelf which characterize the bottom. We point out that in none of the simulations we have observed a bore. Nonlinearity and non-hydrostatic dynamics became instead important in localized shallow regions where frequency dispersion and wave breaking have been observed in the real event, see again the study of Chen *et al.* [2014]. Modeling these local effects, for which Boussinesq-type equations together and a much finer grid are needed, is beyond the scope of the present study.

One possible explanation for the leading wave underestimation could be the absence of the contribution of horizontal seafloor displacement in the waveform computation considered so far. Horizontal displacement has been included and its effect can be analyzed by inspecting the GPS buoys in figure 5.7. Actually the computation of the *first wave peak improves for all the buoys* in the case in which horizontal displacement is retained. In particular the wave height recorded at the buoy G802 (where the highest tsunami height can be observed) is one meter higher. Despite this improvement, maximum wave heights at the buoys are still underestimated with respect to observation (up to 2.5 $[m]$ at the GPS802).

We have tried to improve the results with the moving mesh algorithm discussed in chapter 3 but, unfortunately, this was not immediate. First of all, the topology of the mesh in figure 5.3 is not suited for moving mesh computations due to the strong variation of the mesh size. Here, to perform a moving mesh tsunami simulation, we have used a computational grid which is refined smoothly as the boundary is approached. The minimum mesh size at the coast is $h_k = 1000\,[m]$ compared to a minimal size of 120 $[m]$ of the mesh provided by EDF, while the maximum is $h_k = 50000\,[m]$. The grid has 151650 nodes and 299207 elements. In the initial waveform *it is not* considered the horizontal displacement. The moving mesh parameters, see section 3.4.2, are $\alpha = 50$, $\beta = 0.1$ and $\delta = 200$.

The initial propagation of the tsunami wave and the corresponding mesh movement are visualized in figure 5.8. The images show snapshots of the initial state, after $20'$ and $40'$. We can observe that mesh adaptation increases the resolution of wave shoaling and allows to well capture the smooth wave pattern. However as the wave interact with the boundary mesh tangling has revealed to be more important then we expected. In figure 5.10 we show typical tangling situations at the boundaries that we have faced during the simulation. The smoothing technique discussed in section 3.4.3 become fundamental to solve this issue and avoid tangling in most situations. However to be able to perform a full simulation without mesh tangling, an a posteriori automatic untangling procedure has been set up. This procedure consist in an *a-posteriori* reduction of the nodal displacement by locally increasing the value of the relaxation parameter μ_i in equation (3.17). It is based on the following loop which ends

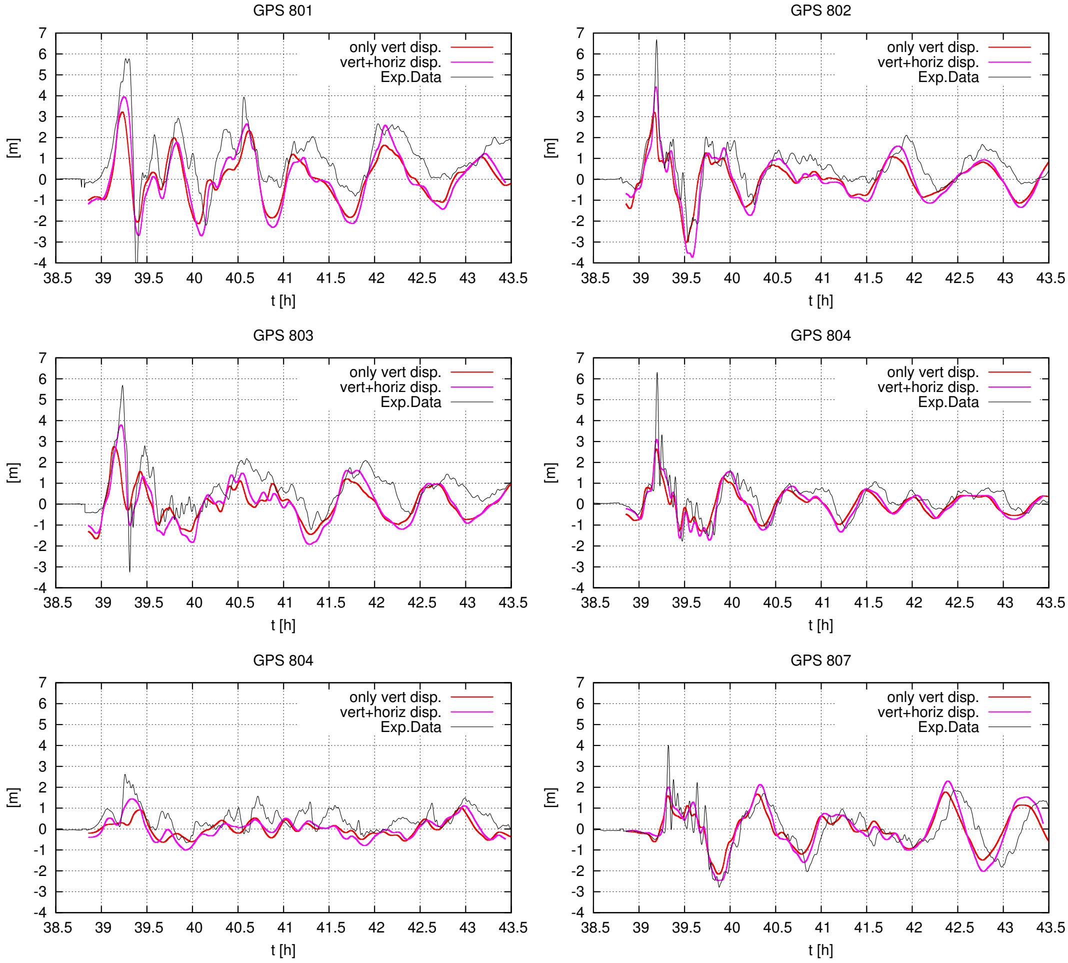

Figure 5.7: Simulated (RD) free surface level recorded at GPS buoys with and without seafloor horizontal displacement in the waveform computation.

when all the cells have been untangled:

1. we set a counter $n = 0$ and a constant relaxation factor $\mu_i = 1 \quad \forall i \in \mathcal{T}_h$. Then we compute the relaxation step with (3.17)
$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^n + \hat{\boldsymbol{\delta}}_i^k$$
2. $\forall K \in \mathcal{T}_h$ we check tangling. If there is no tangling the loops ends. Otherwise, if one cell tangles $|K| < 0$ we flag it and we compute its center position $\boldsymbol{x}_G^K$, then we set
$$\begin{aligned} n &= n+1 \\ \tau &= 2n^2 \end{aligned}$$
and we compute for every node $\forall i \in \mathcal{T}_h$
$$\mu_i = \min\left(\mu_i, 1 - e^{-\frac{d_{G,i}}{h_k \tau}}\right)$$
with $d_{G,i}$ the distance between the center of the tangled element and node i.
3. we perform a new relaxation step (3.17):
$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^n + \mu_i \hat{\boldsymbol{\delta}}_i^k$$
and we go back to step 2.

The simulated free surface level for the adaptive ALE simulation is reported in figure 5.11 where it is compared with the fixed grid simulation. We observe that the moving mesh is slightly curing the excessive diffusion of the leading tsunami wave encountered for the fixed mesh simulation. This should suggest that the fixed computation obtained with the mesh of EDF is locally not converged yet. As a final remark we note that the overall cost of the moving mesh algorithm with the untangling procedure grows making it ineffective. For this reason we have moved to an *embedded mesh* that, in the benchmarks of sections 3.7.3, 3.7.4 and 3.7.5, *provided an accurate wave reflection when coupled with a strong mesh adaptation at the shoreline.* In that cases tangling was not an issue.

5.3 Results with embedded mesh

We have tested an embedded approach in which the whole coast has simply been treated as a wet-dry boundary. We will in particular present simulations preformed with our RD approach on both a fixed and a moving grid.

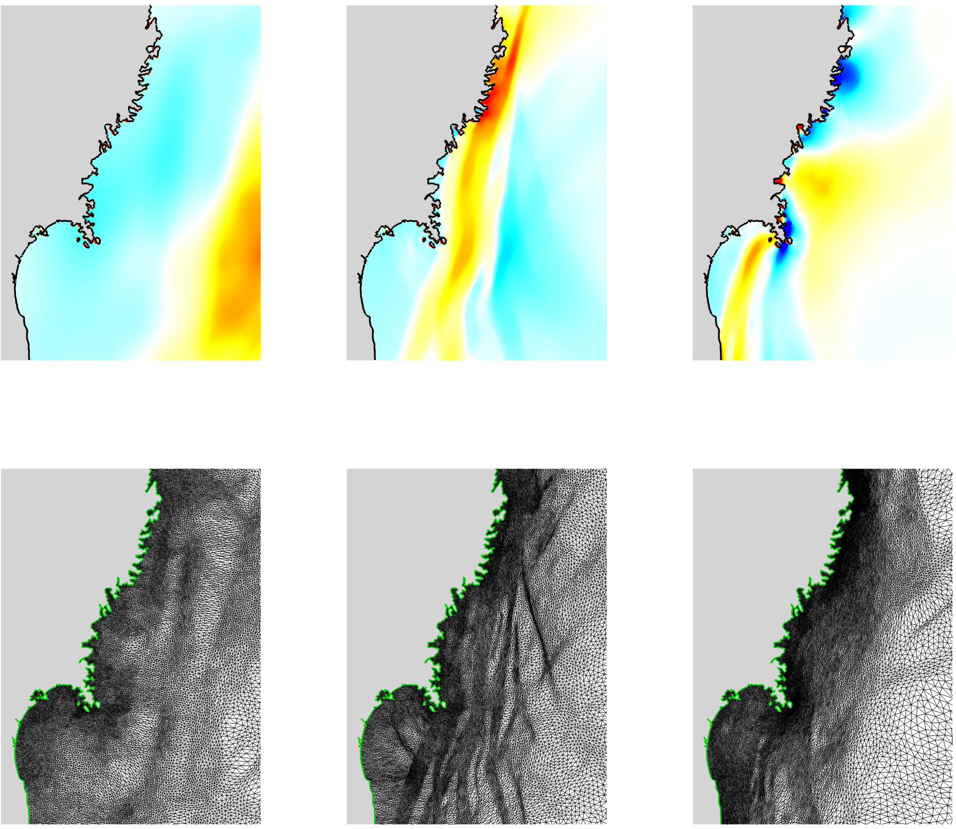

Figure 5.8: Snapshot of free surface level (in meters) and corresponding moving mesh refinement every twenty minutes starting from time 14:51:18: t = 0:00, 0:20. and 0:40.

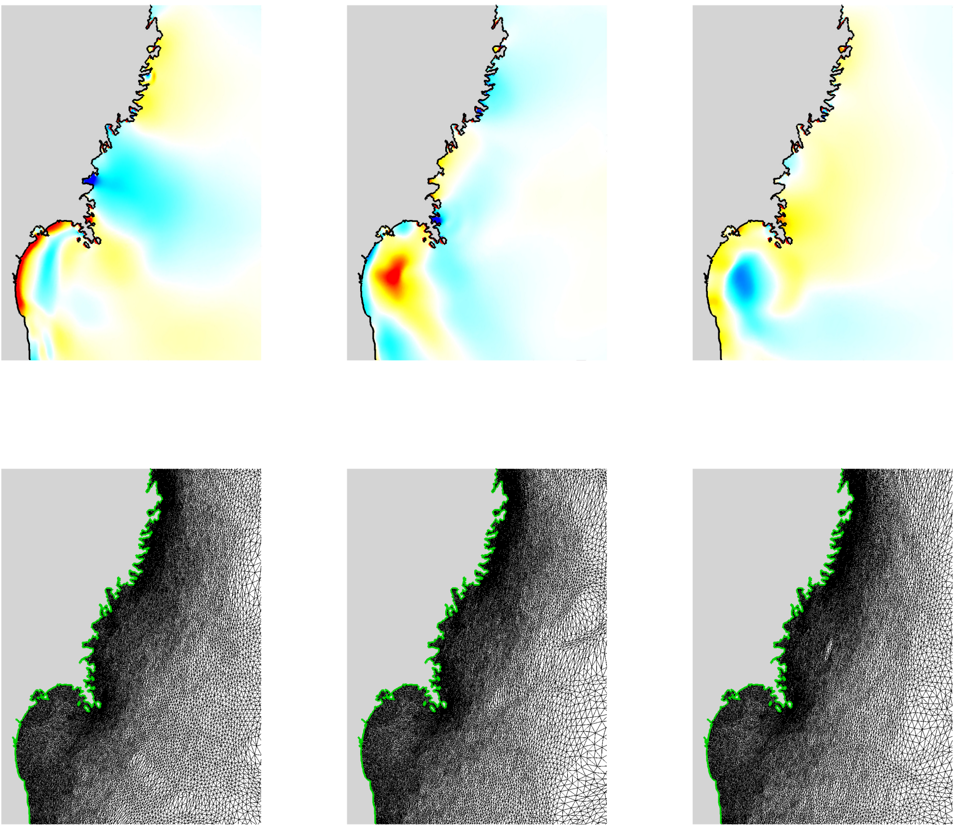

Figure 5.9: Snapshot of the simulated (ALE-RD) free surface level (in meters) and corresponding moving mesh refinement every twenty minutes starting from time 14:51:18: t = 0:40, 1:00 and 1:20.

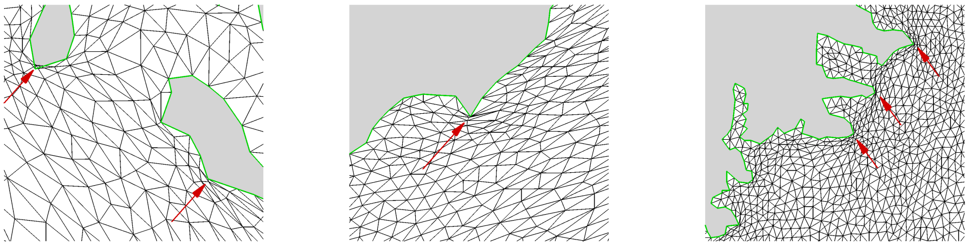

Figure 5.10: Examples of mesh tangling at the boundaries.

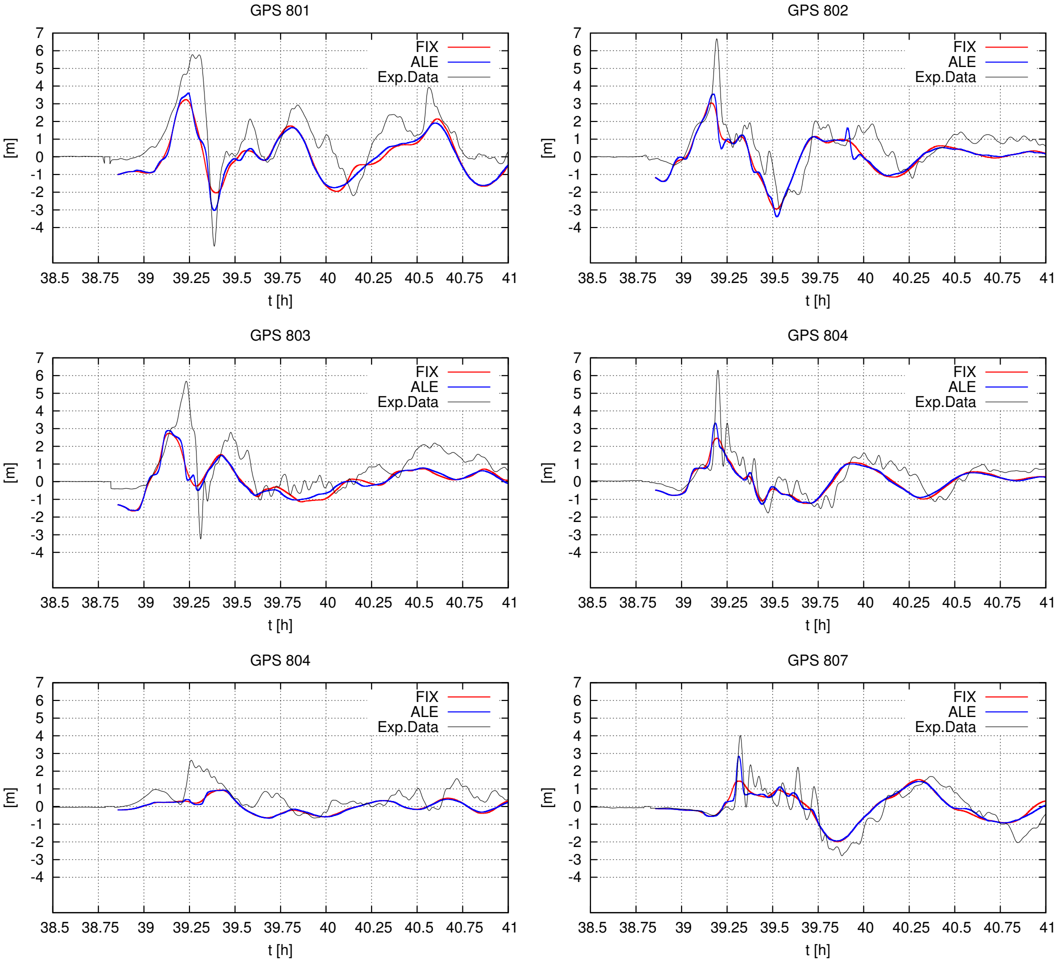

Figure 5.11: Simulated (RD) free surface level recorded at GPS buoys with fixed fitted mesh and ALE moving fitted mesh.

In the above section we got the evidence that, for the Tohoku-Honsu event, the horizontal displacements must be taken into account in the computation of the initial waveform. For this reason, the initial condition considered in all the tests of this section will include this effect and will coincide with the one in the right picture of figure 5.2.

With the embedded approach, friction becomes important during the inundation stage. It is well known that having accurate maps of the Manning coefficient (*roughness maps*) has a considerable influence on inundations, specially in densely populated areas. Here we will employ a simpler, yet widely used approach of considering a constant Manning of $n = 0.03125\,[sm^{-1/3}]$. This value represents land covered with coarse sand and is typically used when no detailed roughness map is available Gayer *et al.* [2010].

First we have run a fixed grid computation with an unstructured embedded mesh composed of 689861 nodes and 1379054 triangles. The mesh is adapted to the bathymetry: the minimum local mesh size is at the coast $(120\,[m])$ and the maximum mesh size is $5\,[km]$. We note that this grid is finer with respect to the fitted one detailed in the previous section and it is used here to run a reference simulation. In figure 5.12 the reader can find the simulated free surface level at the buoys. As suspected from the previous paragraph, we realize that the more refined mesh reduce numerical diffusion and the wave profile of the leading wave appear much sharper. More importantly, *the height of the first wave is considerably closer to the experimental value* with respect to the result obtained with the fitted mesh of the previous section. A side effect is that, due to the friction in the wetting/drying phase, the reflected wave are slightly damped.

With the embedded reference mesh we model also the flooding of the Japan coastal prefectures. We focus the attention on three bays in the south of the Iwate prefecture: in figure 5.13, as they appear from south to north, they are the Kesennuma, Hirota and Ofunato bay. In figure 5.14 we provide a comparison between the inundated areas predicted by the RD scheme and the surveyed areas for these three densely populated bays. With a mesh size of 120 meter we can say to have a good prediction of the runup process, except very fine scale flooding such as the inundation of two narrow channels at the end of Hirota bay. In the simulation they remain dry even if water was observed.

We test now if the moving mesh methods could provide the same resolution of the reference simulation, using a coarser computational mesh. The computational mesh is adapted to the bathymetry but the minimum local mesh size (in the proximity o the coast) is $360\,[m]$, three times larger with respect to the reference mesh. It results a computational mesh composed by 364864 nodes and 728874 elements. The moving mesh parameters are $\alpha = 50$, $\beta = 0.1$ and $\delta = 200$.

From figure 5.15 to 5.20 we show the solution and the mesh, in an alternate fashion, for a zoom of the physical domain close to Japan coast. The snapshots

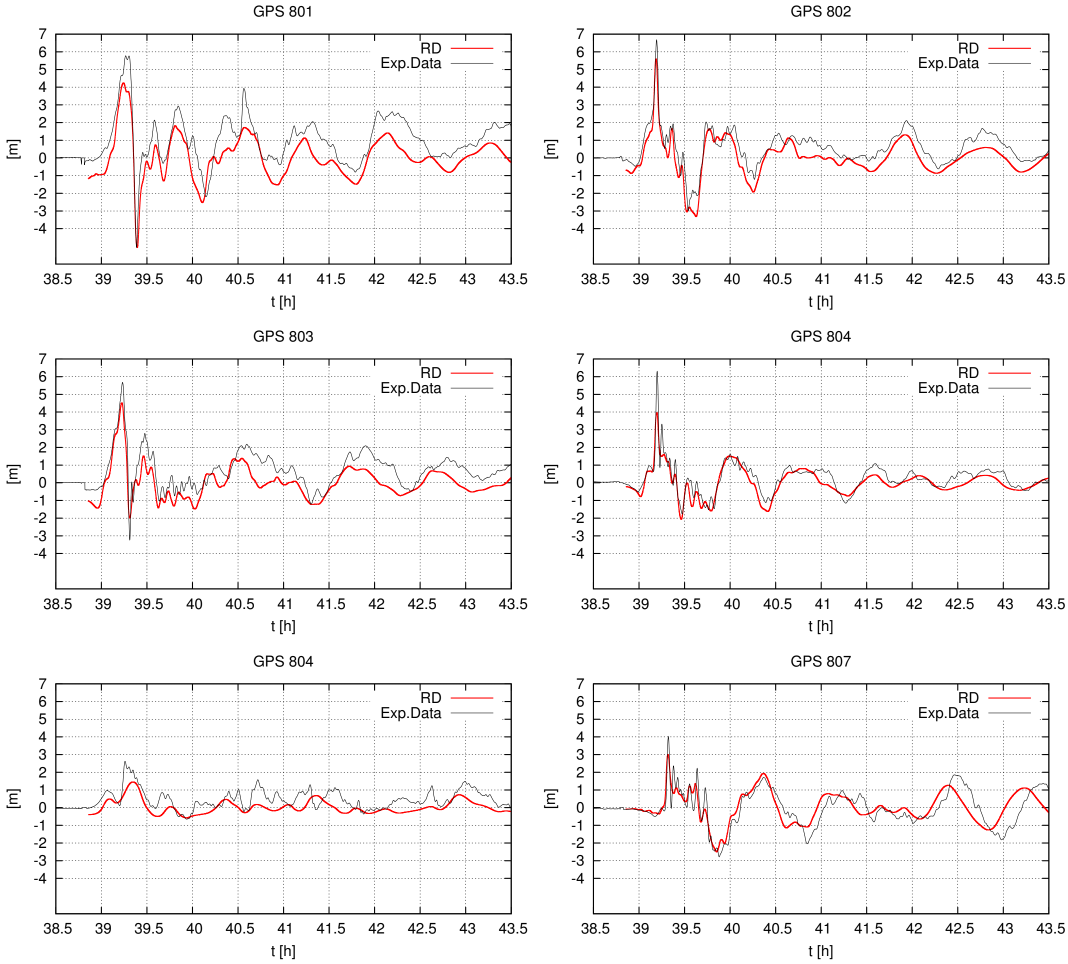

Figure 5.12: Simulated (RD) sea level displacement recorded at GPS buoys with an embedded reference mesh.

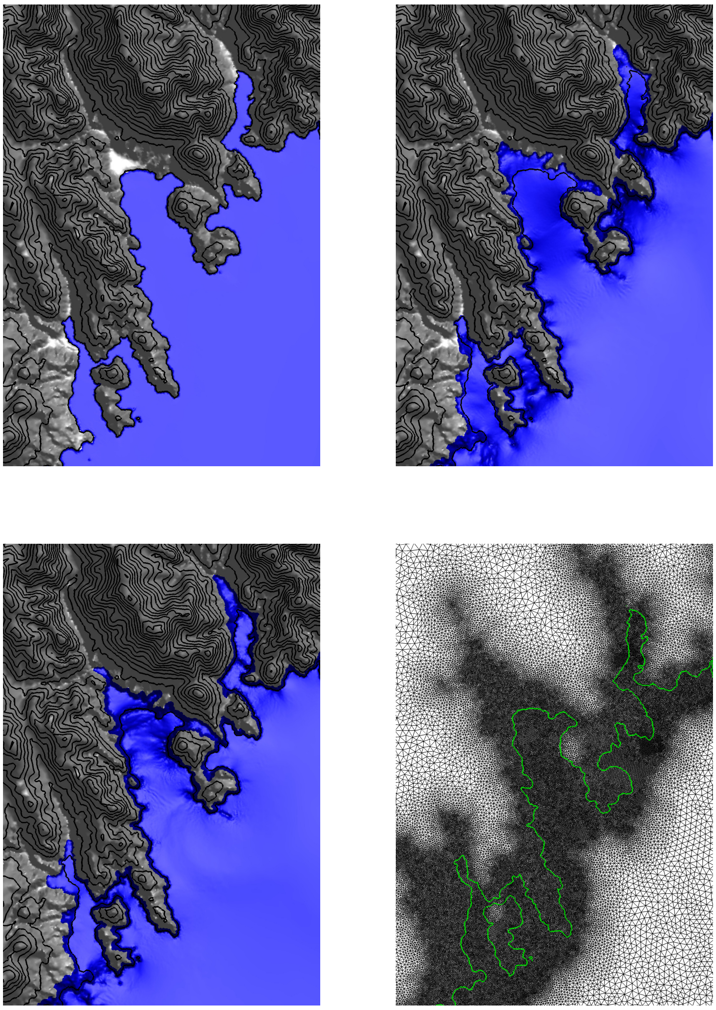

Figure 5.13: Simulated flooding in the southern Iwate prefecture. Snapshot of the numerical simulation (RD) at t = 0:00, 0:40 and 0:50. The picture in bottom-right position is the embedded mesh with the coastline drawn in green.

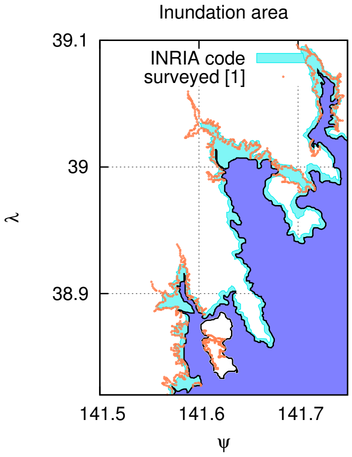

Figure 5.14: Left. Comparisons between simulated (RD) and surveyed flooded areas in the south Iwate prefecture.

of the free surface level and the corresponding adapted meshes are plotted every twenty minutes at $t = 00'$ (fig. 5.15 and 5.16), $t = 20'$ ((fig. 5.17 and 5.18) and $t = 40'$ (fig. 5.19 and 5.20). We observe that the mesh points move and cluster around the complex tsunami wave pattern. This becomes visible especially when the wave shoals and the mesh points gather in correspondence of the largest gradients of the incoming waves train. The shoreline is also detected by the mesh. We recall that tangling at the coast was the main limitation of the fitted method. In this embedded moving mesh computation tangling has not occurred even without employing the smoothing step and the a-posteriori relaxation implemented in the previous section. In figure 5.21 we can observe a closer zoom of a *very complex coastline detected by the moving mesh method without tangling*. Finally from figure 5.22 we can see that the simulated sea level computed with the moving mesh at the wave gauges is similar to the one computed with the reference. Mesh adaptation at the shoreline allows to reproduce sharply the complex reflection of the tsunami wave.

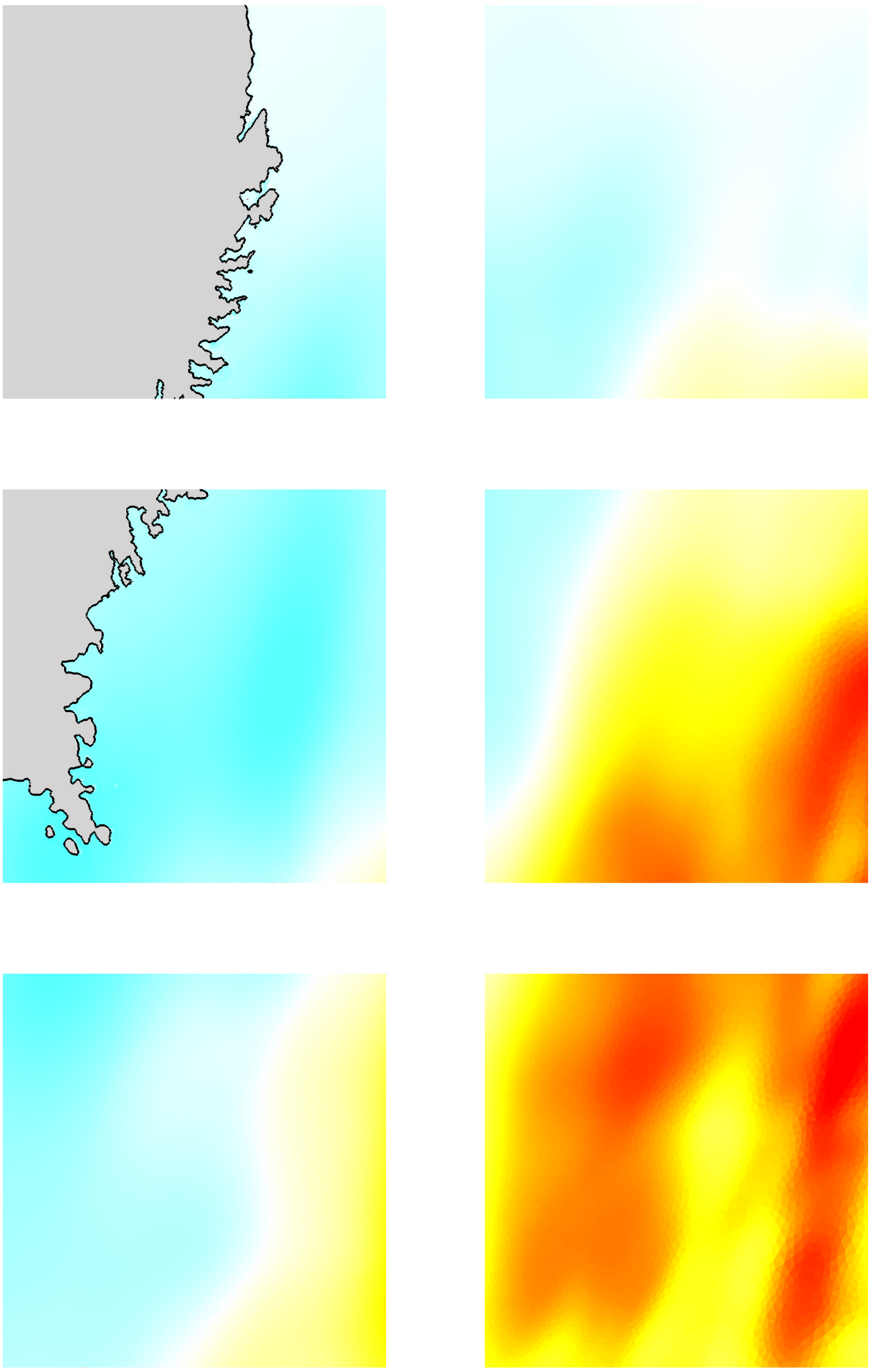

Figure 5.15: Snapshot of the free surface level (same color legend of previous figures) starting from time 14:51:18 at $t = 0{:}00$.

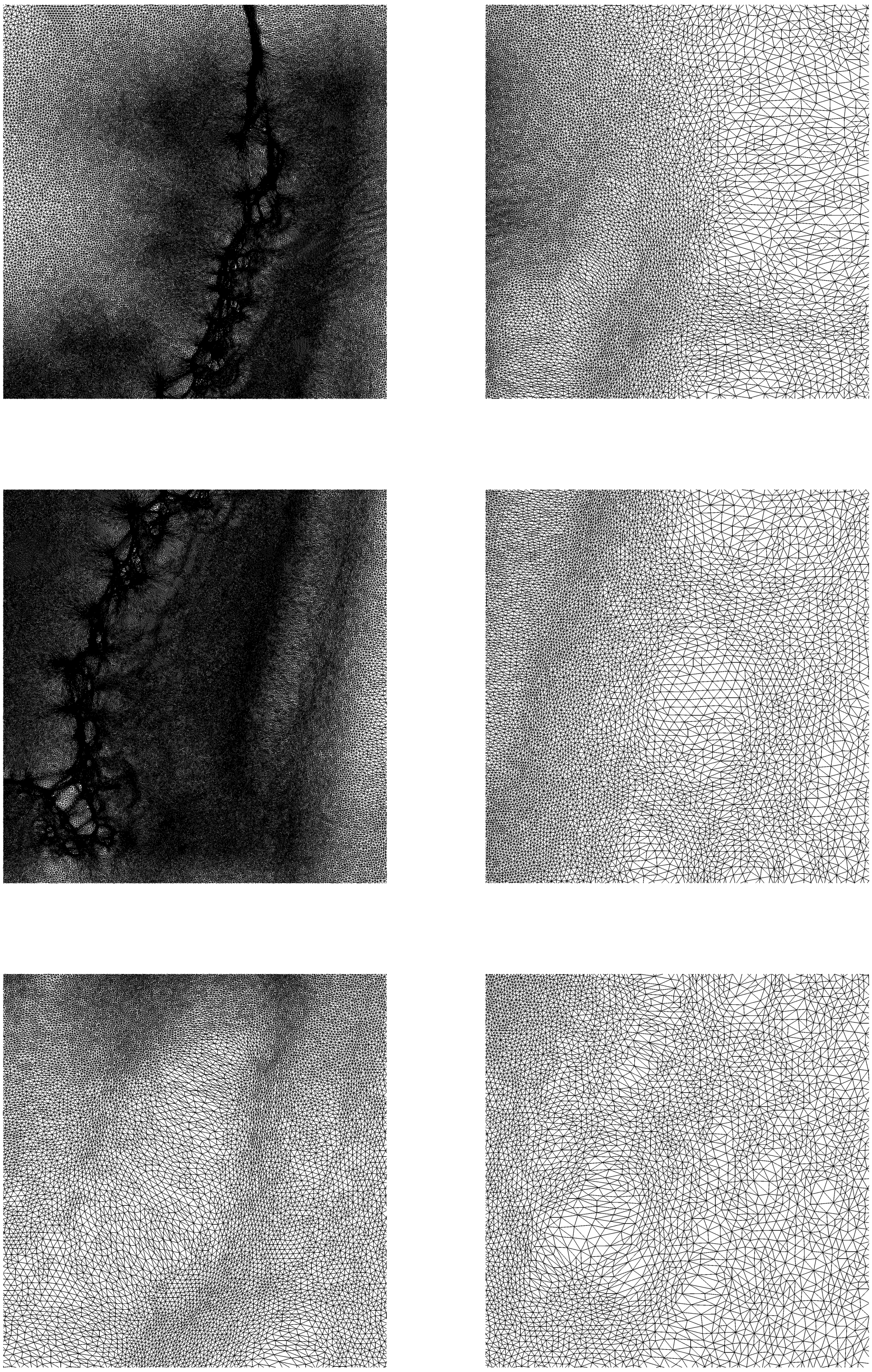

Figure 5.16: Snapshot of moving mesh (same color legend of previous figures) starting from time 14:51:18 at t = 0:00.

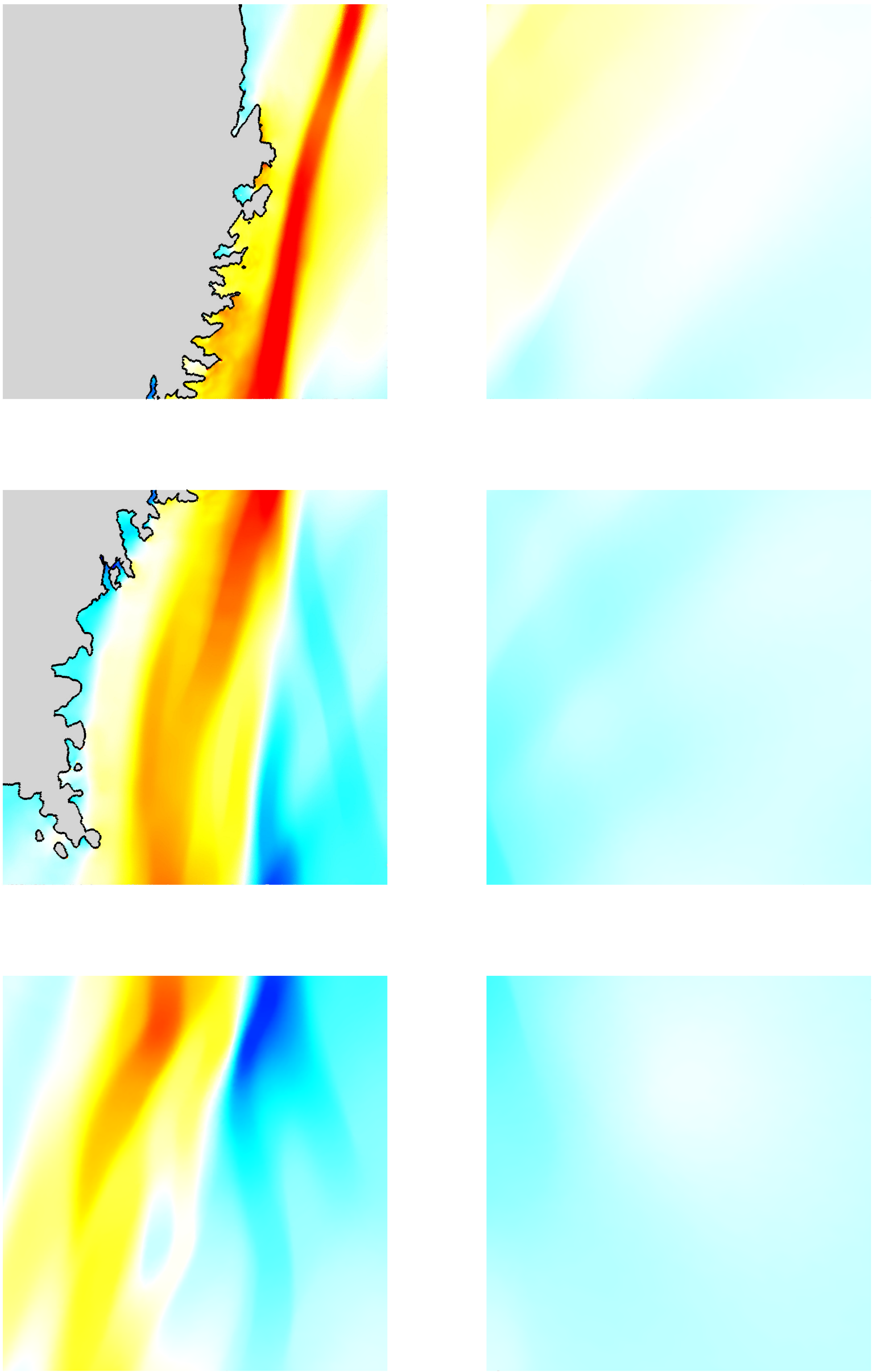

Figure 5.17: Snapshot of the free surface level (same color legend of previous figures) starting from time 14:51:18 at $t = 0{:}20$.

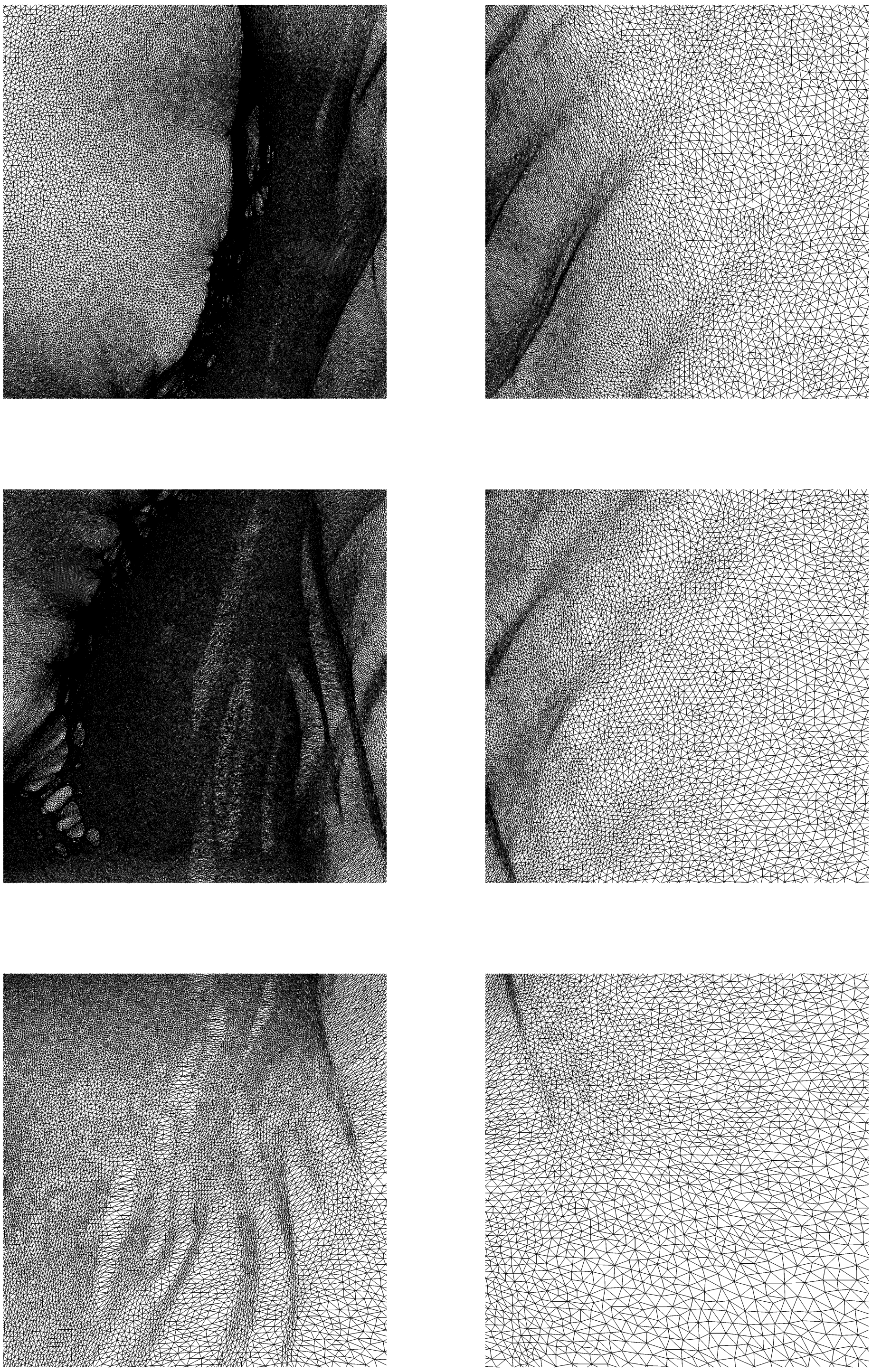

Figure 5.18: Snapshot of moving mesh (same color legend of previous figures) starting from time 14:51:18 at $t = 0{:}20$.

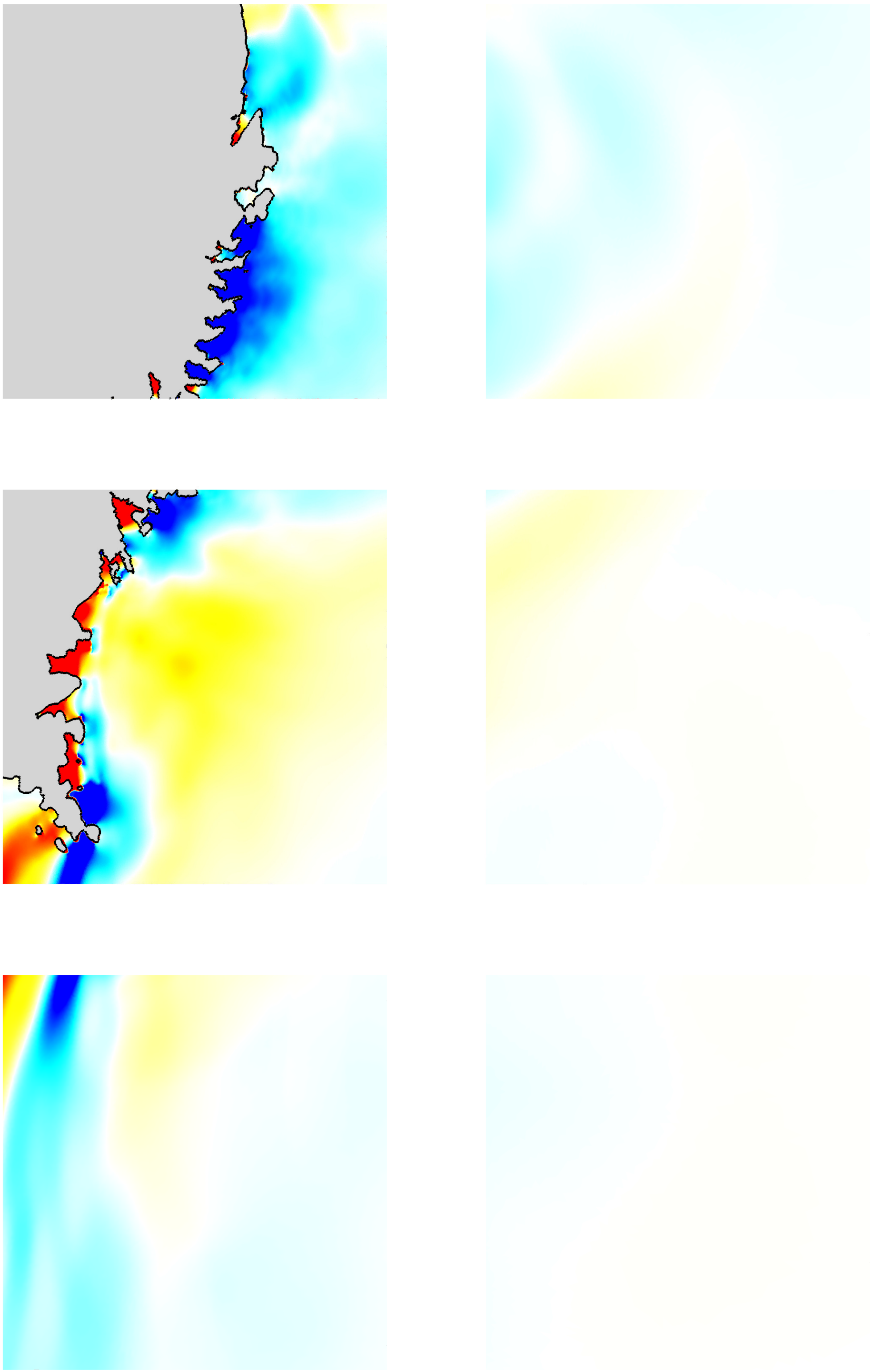

Figure 5.19: Snapshot of the free surface level (same color legend of previous figures) starting from time 14:51:18 at $t = $ 0:40.

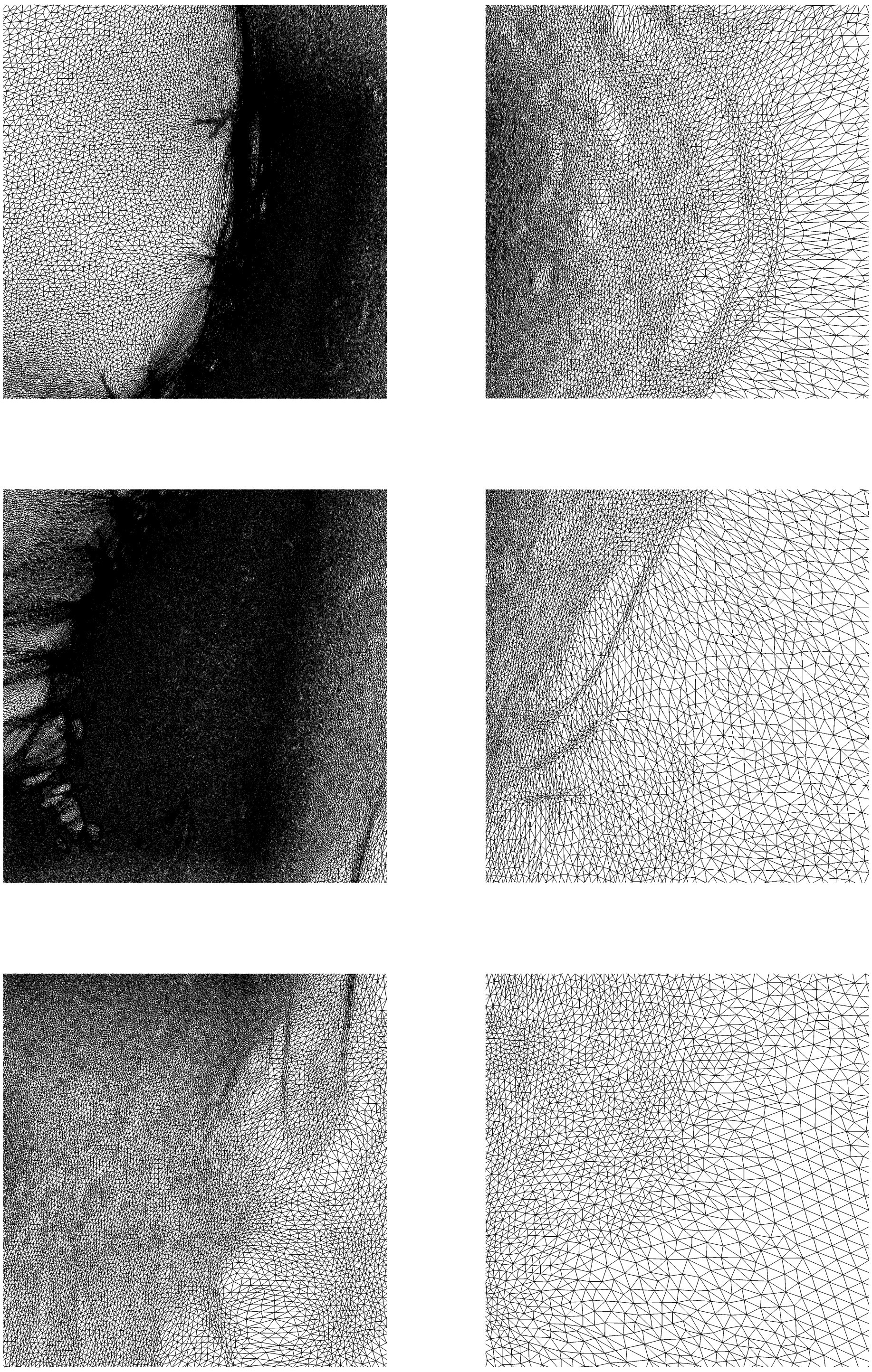

Figure 5.20: Snapshot of moving mesh (same color legend of previous figures) starting from time 14:51:18 at $t = $ 0:40.

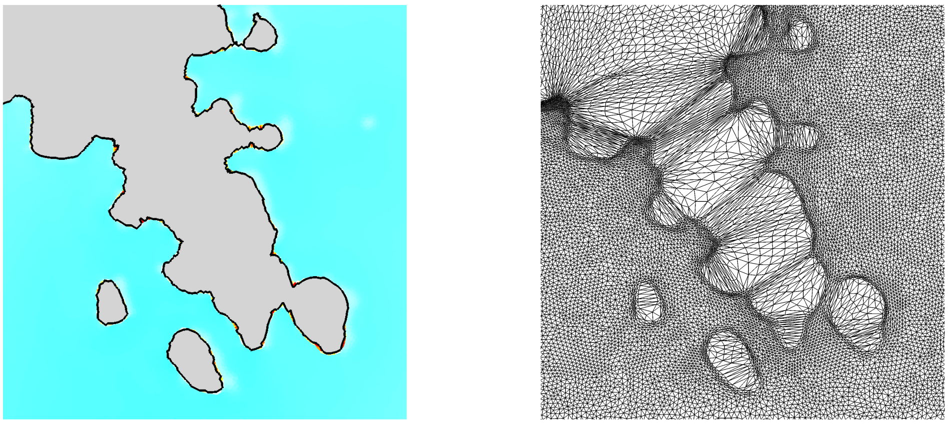

Figure 5.21: Mesh adapted to a complex shoreline (Miyagi prefecture).

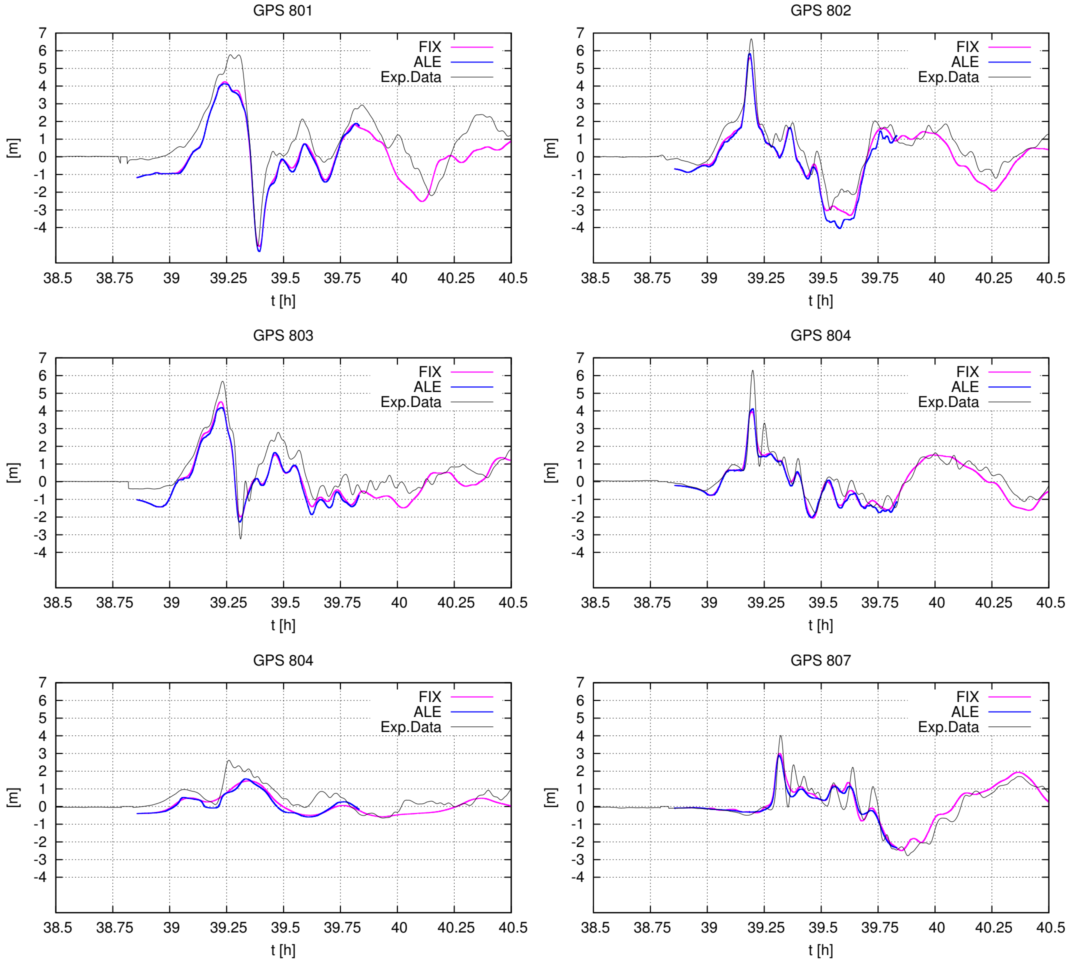

Figure 5.22: Simulated sea level displacement recorded at GPS buoys with fixed embedded mesh and moving embedded mesh.

Conclusions

Summery of the thesis results

In this thesis we have analyzed the use of r-adaptation for the shallow water simulation of complex wave interactions and wave runup on irregular bathymetries.

In chapter 2 we have provided a thorough theoretical setting to construct Well Balanced Finite Volume and Residual Distribution schemes on moving grids with an ALE and a rezoning approach. This schemes includes a mass conserving correction of the nodal bathymetric heights, based on a quadrature of the given bathymetric data. Being based on the actual data, this correction requires no re-initialization, as e.g. the ALE remap used in Zhou *et al.* [2013a] which leads to a numerical deviation from the real bathymetric data. Both simple academic test cases and long wave benchmarks have confirmed our theoretical expectations in terms of the conservation of steady equilibria and mass conservation.

In chapter 3 we have coupled these schemes with the Laplacian-type r-adaptation method of Ceniceros and Hou [2001] and we have investigated different coupling strategies between the flow solver and the adaptation 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. 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 Tang and Tang [2003]; Ceniceros and Hou [2001]; Ni *et al.* [2009]; Zhou *et al.* [2013a,b]; Chen *et al.* [2008] providing quantitative as well as qualitative elements.

In chapter 4 we have extended the Shallow Water solver (both Residual Distribution and Finite Volume) and the moving mesh method to curvilinear coordinates. The Residual Distribution approximation enhances to unsteady problems the scheme of Rossmanith [2013] and it is appealing due to its simplicity and rigor in treating all the issues that the spherical geometry poses such as the medium heterogeneity and the geometric source term. Numerical results show that the resulting ALE r-adaptive method can improve the resolution of steep solutions of the SWEs on the sphere.

In chapter 5 we have employed r-adaptation to the simulation of the Tohoku-Honsu tsunami. This includes the large scale wave propagation and coastal flooding. We show that with a combined use of r-adaptation an embedded meshes it is possible to perform accurate tsunami simulations with a limited number of mesh points.

Limitations and future perspective

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 Filippini *et al.* [2016]. 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 mesh adaptation method, as well as its discretization, and iterative solution will be object of future work. Although successful we have observed that the implementation of the simple Laplacian poses some strong limitation on the control of mesh quality. A possible avenue is the combination of the simple Laplacian approach used here, providing a very sharp approximation of discontinuous features, with elastic deformation Stein *et al.* [2004] 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 Tang and Tang [2003] 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.

Future work will be also dedicated to overcome the pole singularity and to propose a Residual Distribution method for global circulation model. The pole problem for the moving mesh method is more complicated and necessitates further study. However the work of Di *et al.* [2006] could be implemented to then test the ALE-SWEs moving mesh method against benchmarks on the whole sphere.

Appendix A

Tensors, vectors and nabla operators in curvilinear coordinates

Notation for curvilinear coordinates

Consider a transformation from a Cartesian coordinate system $\boldsymbol{x} = \{x^1, x^2, x^3\}$ to a general curvilinear coordinates system $\boldsymbol{z} = \{z^1, z^2, z^3\}$

$$\boldsymbol{x} = G(\boldsymbol{z})$$

The orthonormal basis in Cartesian coordinates is $\{\boldsymbol{e}_i\} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$. The non-orthonormal covariant basis vectors is defined by $\{\boldsymbol{g}_j\} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3\}$. A vector $\boldsymbol{x}$ in Euclidean space can be expressed in curvilinear coordinates as

$$\boldsymbol{x} = \boldsymbol{g}_1 z^1 + \boldsymbol{g}_2 z^2 + \boldsymbol{g}_3 z^3 = \sum_{i=1}^{3} \boldsymbol{g}_i z^i \equiv \boldsymbol{g}_i z^i$$

where the symbol $\equiv$ means that we have used the standard summation convention. The basis vectors of the curvilinear framework can be calculated from the orthonormal basis associated to the Cartesian coordinate system

$$\boldsymbol{g}_i = \frac{\partial \boldsymbol{x}}{\partial z^i} = \sum_{j=1}^{3} \frac{\partial \boldsymbol{x}}{\partial x^j} \frac{\partial x^j}{\partial z^i} \equiv \frac{\partial \boldsymbol{x}}{\partial x^j} \frac{\partial x^j}{\partial z^i} = \boldsymbol{e}_j \frac{\partial x^j}{\partial z^i} \tag{A.1}$$

The components $\frac{\partial x^j}{\partial z^i}$ fully describe the transformation from cartesian to curvilinear coordinate system and they can be collected in a transformation tensor, or jacobian

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial x^1}{\partial z^1} & \frac{\partial x^1}{\partial z^2} & \frac{\partial x^1}{\partial z^3} \\ \frac{\partial x^2}{\partial z^1} & \frac{\partial x^2}{\partial z^2} & \frac{\partial x^2}{\partial z^3} \\ \frac{\partial x^3}{\partial z^1} & \frac{\partial x^3}{\partial z^2} & \frac{\partial x^3}{\partial z^3} \end{bmatrix} \tag{A.2}$$

which admits a determinant $\det(\boldsymbol{J}) \neq 0$. The contravariant basis is useful to describe tangent plane to a given surface. Given a point P embedded in cartesian space, the contravariant basis $\boldsymbol{g}^k$ is perpendicular to the plane defined by the covariant basis $\boldsymbol{g}_i$ and $\boldsymbol{g}_j$ in P. It is defined as

$$\boldsymbol{g}^k = \frac{\boldsymbol{g}_i \times \boldsymbol{g}_j}{(\boldsymbol{g}_i \times \boldsymbol{g}_j) \cdot \boldsymbol{g}_k}$$

The metric tensor $\boldsymbol{G}$ is composed of the coefficients given by the scalar product between the covariant basis vector

$$G_{ij} = \boldsymbol{g}_i \cdot \boldsymbol{g}_j$$

with determinant $\det(\boldsymbol{G}) \neq 0$. The inverse metric tensor $\boldsymbol{G}^{-1}$ is:

$$G^{ij} = \boldsymbol{g}^i \cdot \boldsymbol{g}^j$$

An important relation between the metric tensor and the jacobian $\boldsymbol{G} = \boldsymbol{J}^T \boldsymbol{J}$. In the following we will define the square root of the determinant of the metric tensor with $\sqrt{G}$ which will mean:

$$\sqrt{G} = \sqrt{\det \boldsymbol{G}} = \det \boldsymbol{J} \tag{A.3}$$

Physical components of tensors

Now, let $\boldsymbol{v}$ be a vector in the curvilinear framework, it can be written in the covariant basis as $\boldsymbol{v} = v^j \boldsymbol{g}_j$, always using standard summation convention. For our purposes it is useful also to define physical components of a vector in a unitary covariant basis. This means that the curvilinear basis vectors have to be normalized

$$\boldsymbol{g}_i^* = \frac{\boldsymbol{g}_i}{|\boldsymbol{g}_i|}$$

The vector reads

$$\boldsymbol{v} = (v^j |\boldsymbol{g}_j|) \boldsymbol{g}_j^* = v^{*j} \boldsymbol{g}_j^*$$

using the following components

$$v^{*j} = |\boldsymbol{g}_j| v^j \tag{A.4}$$

the star denotes physical contravariant vector components defined on the unitary basis. For tensors $\boldsymbol{T} = T^{ij} \boldsymbol{g}_i \boldsymbol{g}_j$ the physical contravariant tensor components in the unitary contravariant basis write

$$T^{*ij} = |\boldsymbol{g}_i||\boldsymbol{g}_j| T^{ij} \tag{A.5}$$

Christoffel symbol

The 3×3 second-kind Christoffel symbol reads

$$\Gamma^k_{ij} = \boldsymbol{g}^k \cdot \frac{\partial \boldsymbol{g}_i}{\partial z^j}$$

We recall two important property of the Christoffel symbol which we will use later in the text. The first one reads as follows

$$\Gamma^i_{ij} = \frac{1}{\sqrt{G}} \frac{\partial \sqrt{G}}{\partial z^j} \tag{A.6}$$

the second important relationship is known as Ricci's Lemma and concerns the derivative of the coefficients of the metric covariant coefficient G^{ij} with respect to z^k. For us it will be useful the derivation with respect to z^j. In this particular case:

$$\frac{\partial G^{ij}}{\partial z^j} = -G^{mj}\Gamma^i_{jm} - G^{im}\Gamma^j_{jm} \tag{A.7}$$

for the demonstration, see Nguyen-Shafer and Schmidth [2014].

Nabla operator in curvilinear coordinates

Nabla operator in a general curvilinear coordinates reads

$$\nabla = \boldsymbol{g}^i \frac{\partial}{\partial z^i} \tag{A.8}$$

Given a scalar ϕ, a vector $\boldsymbol{v}$ and a second order tensor $\boldsymbol{T}$, we list the nabla operator in curvilinear coordinates that we use in this thesis. The divergence of a vector $\boldsymbol{v}$ in curvilinear coordinates writes

$$\nabla \cdot \boldsymbol{v} = \left(\boldsymbol{g}^i \frac{\partial}{\partial z^i} \right) \cdot \boldsymbol{v} = \frac{1}{\sqrt{G}} \frac{\partial}{\partial z^i} \left(\sqrt{G}\, v^i \right), \quad i = 1, 2, 3 \tag{A.9}$$

The divergence of a second order tensor is

$$\nabla \cdot \boldsymbol{T} = \left(\boldsymbol{g}^k \frac{\partial}{\partial z^k} \right) \cdot \boldsymbol{T} = \left(\frac{1}{\sqrt{G}} \frac{\partial}{\partial z^i} \left(\sqrt{G}\, T^{ij} \right) + T^{ik}\Gamma^j_{ik} \right) \boldsymbol{g}_j, \quad i, j = 1, 1, 2 \tag{A.10}$$

The gradient of a vector is

$$\nabla \boldsymbol{v} = \left(\boldsymbol{g}^i \frac{\partial}{\partial z^i} \right) \boldsymbol{v} = \boldsymbol{g}_j \boldsymbol{g}^i \frac{\partial v^j}{\partial z^i}, \quad i = 1, 2, 3 \tag{A.11}$$

Finally the Laplacian of a scalar (also called Laplace-Beltrami operator) is reported here with two different formulas, both useful:

$$\nabla^2 \phi = \boldsymbol{g}^i \frac{\partial}{\partial z^i} \cdot \left(\boldsymbol{g}^j \frac{\partial \phi}{\partial z^j} \right) = G^{ij} \left(\frac{\partial^2 \phi}{\partial z^i \partial z^j} - \Gamma^k_{ij} \frac{\partial \phi}{\partial z^k} \right) \tag{A.12}$$

$$= \frac{1}{\sqrt{G}} \frac{\partial}{\partial z^i} \left(\sqrt{G} G^{ij} \frac{\partial \phi}{\partial z^j} \right) \tag{A.13}$$

Elementay differential geometry

We consider a 2-dimensional Riemannian manifold $\mathcal{M}$. At a point R defined by curvilinear coordinates $\{z^1, z^2\}$, $R(z^1, z^2)$ let $\{\boldsymbol{g}_1, \boldsymbol{g}_2\}$ be the vector basis. In the nighbourhood of this point we construct a tangent plane.

Arc lenght between points in curvilinear coordinates

Consider now two points $P(z^1, z^2)$ and $Q(z^1, z^2)$ connected by a curve $\boldsymbol{c}$ parametrized with $t \in [0, 1]$. The arc length of ds between the points Q and Q is given by

$$\left(\frac{ds}{dt}\right)^2 = \frac{d\boldsymbol{x}}{dt} \cdot \frac{d\boldsymbol{x}}{dt} = \boldsymbol{g}_i \frac{dz^i}{dt} \cdot \boldsymbol{g}_j \frac{dz^j}{dt}$$

$$ds = \sqrt{\frac{dz^i}{dt} G_{ij} \frac{dz^j}{dt}}\, dt$$

And the arc lenght of PQ is computed with a line integral from $t = 0$ to $t = 1$:

$$l_{\mathcal{M}}(P, Q) = \int_0^1 \sqrt{\frac{dz^i}{dt} G_{ij} \frac{dz^j}{dt}}\, dt \tag{A.14}$$

Surface area in curvilinear coordinates

The area differential of the tangent plane can be calculated using contravariant components of an infinitesimal displacement in the tangent plane

$$d\boldsymbol{x} = dz^1 \boldsymbol{g}_1 + \underbrace{dz^2 \boldsymbol{g}_2}_{d\boldsymbol{z}_2} = dz^i \boldsymbol{g}_i \tag{A.15}$$

We get

$$dA = |\boldsymbol{g}_1 \times \boldsymbol{g}_2| dz^1 dz^2 \tag{A.16}$$

with $|\boldsymbol{g}_1 \times \boldsymbol{g}_2| = \sqrt{G}$.

Appendix B

Harmonic maps

We provide the definition of harmonic maps between Riemannian manifold, as emerged from the work of Eell and Sampson [1964]. Given two Riemannian manifold $\mathcal{M}$ and $\mathcal{N}$, the former is described by curvilinear coordinates $\boldsymbol{\chi}$ with metric tensor $\boldsymbol{G}$ while the latter is described by curvilinear coordinates $\boldsymbol{x}$, being $\boldsymbol{H}$ its metric. Let $A : \mathcal{M} \to \mathcal{N}$ be a smooth map. Let Ω be a domain of $\mathcal{M}$. We define an energy functional in local coordinates:

$$E(\boldsymbol{x}) = \frac{1}{2} \int_\Omega e \sqrt{G} d\boldsymbol{\chi}$$

where the energy density is $e = ||\nabla \boldsymbol{x}||^2$. From (A.11), the gradient of a vector is the following tensor:

$$\nabla \boldsymbol{x} = \boldsymbol{g}^i \boldsymbol{h}_\alpha \frac{\partial x^\alpha}{\partial \chi^i}$$

computing its norm, the energy functional reads:

$$||\nabla \boldsymbol{x}||^2 = G^{ij} H_{\alpha\beta} \frac{\partial x^\alpha}{\partial \chi^i} \frac{\partial x^\beta}{\partial \chi^j}$$

where standard summation convention is assumed. The energy functional reads:

$$E(\boldsymbol{x}) = \frac{1}{2} \int_\Omega G^{ij} H_{\alpha\beta} \frac{\partial x^\alpha}{\partial \chi^i} \frac{\partial x^\beta}{\partial \chi^j} \sqrt{G} d\boldsymbol{\chi} \tag{B.1}$$

The map $\boldsymbol{x} = A(\boldsymbol{\chi})$ is called harmonic if it is a critical point of the energy functional. Moreover, the harmonic map is the solution of the following Euler-Lagrange equation:

$$G^{ij} \left(\frac{\partial^2 x^\alpha}{\partial \chi^i \partial \chi^j} - \Gamma^k_{ij} \frac{\partial x^\alpha}{\partial \chi^k} + \Gamma^\alpha_{\gamma\delta} \frac{\partial x^\gamma}{\partial \chi^i} \frac{\partial x^\delta}{\partial \chi^j} \right) = 0 \tag{B.2}$$

Here Γ^k_{ij} and $\Gamma^\alpha_{\gamma\delta}$ denote the Christoffel symbols on $\mathcal{M}$ and $\mathcal{N}$ respectively. Using (A.13) we get:

$$\frac{1}{\sqrt{G}} \frac{\partial}{\partial \chi^i} \left(\sqrt{G} G^{ij} \frac{\partial x^\alpha}{\partial \chi^j} \right) + G^{ij} \Gamma^\alpha_{\gamma\delta} \frac{\partial x^\gamma}{\partial \chi^i} \frac{\partial x^\delta}{\partial \chi^j} = 0 \tag{B.3}$$

Appendix C

Lat-Lon coordinates

We consider a sphere $\mathcal{S}^2$ with radius R described by orthogonal curvilinear coordinates $\{z^1, z^2\} = \{\psi, \lambda\}$ whose basis are mutually perpendicular but not unitary. $z^2 = \lambda$ is the latitude and $z^1 = \psi$ the longitude. A vector $\boldsymbol{x}$

$$\begin{aligned}\boldsymbol{x} &= (R\cos\lambda\cos\psi)\boldsymbol{e}_1 + (R\cos\lambda\sin\psi)\boldsymbol{e}_2 + (R\sin\lambda)\boldsymbol{e}_3 \\ &= x^1\boldsymbol{e}_1 + x^2\boldsymbol{e}_2 + x^3\boldsymbol{e}_3\end{aligned}$$

We got the transformation $\{x^i\} = G(\{z^j\})$

$$\begin{aligned}x^1 &= R\cos(z^2)\cos(z^1) \\ x^2 &= R\cos(z^2)\sin(z^1) \\ x^3 &= R\sin(z^2)\end{aligned} \tag{C.1}$$

For lat-lon coordinates, the transformation tensor, according to equation (A.2), is

$$\boldsymbol{J} = \begin{bmatrix} -R\cos\lambda\sin\psi & -R\sin\lambda\cos\psi \\ R\cos\lambda\cos\psi & -R\sin\lambda\sin\psi \\ 0 & R\cos\lambda \end{bmatrix}$$

The covariant basis results

$$\begin{aligned}\boldsymbol{g}_1 &= (-R\cos\lambda\sin\psi)\boldsymbol{e}_1 + (R\cos\lambda\cos\psi)\boldsymbol{e}_2 + (0)\boldsymbol{e}_3 \Rightarrow |\boldsymbol{g}_1| = R\cos\lambda \\ \boldsymbol{g}_2 &= (-R\sin\lambda\cos\psi)\boldsymbol{e}_1 + (-R\sin\lambda\sin\psi)\boldsymbol{e}_2 + (\rho\cos\lambda)\boldsymbol{e}_3 \Rightarrow |\boldsymbol{g}_2| = R\end{aligned}$$

The metric tensor reads

$$\boldsymbol{G} = \begin{bmatrix} R^2\cos^2\lambda & 0 \\ 0 & R^2 \end{bmatrix}$$

with $\sqrt{G} = R^2\cos\lambda$. The infinitesimal area of a sphere is computed as

$$dA = \sqrt{G}\,dz^1dz^2 = R^2\cos\lambda\,d\psi d\lambda$$

and Christofell symbols are

$$\mathbf{\Gamma}^1 = \boldsymbol{g}^1 \cdot \frac{\partial \boldsymbol{g}_i}{\partial z^j} = \begin{bmatrix} 0 & -\tan\lambda \\ -\tan\lambda & 0 \end{bmatrix}, \quad \mathbf{\Gamma}^2 = \boldsymbol{g}^2 \cdot \frac{\partial \boldsymbol{g}_i}{\partial z^j} = \begin{bmatrix} \sin\lambda\cos\lambda & 0 \\ 0 & 0 \end{bmatrix}$$

Appendix D

Proof of the eRK2-RD-ALE update

We extend the same steps contained in Ricchiuto and Abgrall [2010] to the ALE framework. As the authors of the same paper underline, the procedure to construct a second order explicit RD scheme is based on the following three steps:

- analogy between RD and Stabilized Finite Element, valid for the P1 case
- step shifting operator for the time part in the stabilization bubble
- high order mass-lumping

The reader will find these steps in the following. The test function is $w_i = \varphi_i + \gamma_i$. Well-Balanced SWEs in ALE framework (2.28) are multiplied by the the test function and integrated over the domain Ω. Using $dx^m = J_A d\chi^m$ and the hypothesis that $\partial_t|_\chi \varphi_i = 0$, one obtain the weak form (Galerkin)

$$\underbrace{\Delta^k \int_{\Omega_i} \varphi_i \mathsf{w}\, d\boldsymbol{x}}_{I} \underbrace{+\Delta t \int_{\Omega_i} \varphi_i \left(\frac{\partial \mathsf{F}^j}{\partial x^j} - \sigma^j \frac{\partial \mathsf{w}}{\partial x^j} \right)^k d\boldsymbol{x}}_{II} \underbrace{-\Delta t \int_{\Omega_i} \varphi_i \mathsf{w}^k \frac{\partial \sigma^j}{\partial x^j} d\boldsymbol{x}}_{III}$$

$$= \underbrace{\Delta t \int_{\Omega_i} \varphi_i \mathcal{S}^k \, d\boldsymbol{x}}_{IV}$$

the apex k is the RK stage,

$$k = 1 \;\Rightarrow\; \Delta^1 = [\cdot]^* - [\cdot]^n, \quad [\cdot]^1 = [\cdot]^n$$

$$k = 2 \;\Rightarrow\; \Delta^2 = [\cdot]^{n+1} - [\cdot]^n, \quad [\cdot]^2 = \frac{1}{2}([\cdot]^n + [\cdot]^*)$$

We have split the ALE term in a pure advective term and a Geometric Source Term, as is commonly done in other RD-ALE discretizations Dobes and Deconinck [2008]; Michler *et al.* [2002]. The next step is to add the contribution of the stabilization bubble: for this part we use a non conservative form, see Arpaia *et al.* [2014], and secondly we use the step shifting operator $\overline{\Delta}^k$ instead of the original Δ^k. We mention that this approximation is necessary to end up with a fully explicit scheme and does not spoil the formal order of accuracy. The details can be found in Ricchiuto and Abgrall [2010]

$$\begin{aligned} k = 1 \quad &\Rightarrow \quad \overline{\Delta}^1 = [\cdot]^n - [\cdot]^n \\ k = 2 \quad &\Rightarrow \quad \overline{\Delta}^2 = [\cdot]^* - [\cdot]^n \end{aligned}$$

The stabilization part is

$$\Delta t \sum_{K\in\mathcal{D}_i} \left(\underbrace{\int_K \gamma_i \frac{\overline{\Delta}^k \mathsf{w}}{\Delta t}\, d\boldsymbol{x}}_{V_K} + \underbrace{\int_K \gamma_i \left(\frac{\partial \mathsf{F}^j}{\partial x^j} - \sigma^j \frac{\partial \mathsf{w}}{\partial x^j} \right)^k d\boldsymbol{x}}_{VI_K} = \underbrace{\int_K \gamma_i \mathcal{S}^k \, d\boldsymbol{x}}_{VII_K} \right)$$

Now we can consider the sum of I and III:

$$\begin{aligned} I + III \quad &= \quad \Delta^k \int_{\Omega_i} \varphi_i \mathsf{w}\, d\boldsymbol{x} - \Delta t \int_{\Omega_i} \varphi_i \mathsf{w}^k \frac{\partial \sigma^j}{\partial x^j} d\boldsymbol{x} \\ &= \quad \sum_{K\in\mathcal{D}_i} (I_K + III_K) \end{aligned}$$

where we have decomposed the integral over neighbor elements. Using the fact that $\mathsf{w}^k = \bar{\Delta}^k \mathsf{w}/2 + \mathsf{w}^n$, we can write on each element

$$I_K + III_K = \Delta^k \left(\sum_{j\in K} \int_K \varphi_i \varphi_j \, d\boldsymbol{x} \mathsf{w}_j \right) - \Delta t \sum_{j\in K} \int_K \varphi_i \varphi_j \frac{\partial \sigma^j}{\partial x^j} d\boldsymbol{x} \left(\frac{\overline{\Delta}^k \mathsf{w}_j}{2} + \mathsf{w}_j^n \right)$$

Moreover we can use the definition of constant divergence over the element, eq.(4.43)

$$\begin{aligned} \int_K \varphi_i \varphi_j \frac{\partial \sigma^j}{\partial x^j} d\boldsymbol{x} \quad &= \quad \int_K \varphi_i \varphi_j \nabla \cdot \boldsymbol{\sigma} \, d\boldsymbol{x} \\ &= \quad \int_K \varphi_i \varphi_j \, d\boldsymbol{x} \frac{\Delta |K|}{\Delta t |K|} \end{aligned}$$

Wee see the Galerkin mass matrix $m_{ij}^G = \int_K \varphi_i \varphi_j d\boldsymbol{x}$.

$$I_K + III_K \quad = \quad \Delta^k \left(\sum_{j\in K} m_{ij}^G \mathsf{w}_j \right) - \sum_{j\in K} m_{ij}^G \left(\frac{\overline{\Delta}^k \mathsf{w}_j}{2} + \mathsf{w}_j^n \right) \frac{\Delta |K|}{|K|}$$

We follow exactly Ricchiuto and Abgrall [2010] and we perform mass lumping that is $\sum_{j\in K} m_{ij}^G = \int_K \varphi_i d\boldsymbol{x} = |K|/3$ so that

$$I_K + III_K \;=\; \frac{1}{3}\Delta^k\left(|K|\mathsf{w}_i\right) - \frac{1}{3}\Delta|K|\mathsf{w}_i^n - \underbrace{\sum_{j\in K}\frac{1}{2}m_{ij}^G\frac{\Delta|K|}{|K|}\overline{\Delta}^k\mathsf{w}_j}_{VIII_K}$$

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

$$\begin{aligned} I + III &= \Delta^k\left(|C_i|\mathsf{w}_i\right) - \Delta^k|C_i|\mathsf{w}_j^n - \sum_{K\in\mathcal{D}_i}\sum_{j\in K}\frac{1}{2}m_{ij}^G\frac{\Delta|K|}{|K|}\overline{\Delta}^k\mathsf{w}_j \\ &= |C_i^{n+1}|\Delta^k\mathsf{w}_i - \sum_{K\in\mathcal{D}_i}\sum_{j\in K}\frac{1}{2}m_{ij}^G\frac{\Delta|K|}{|K|}\overline{\Delta}^k\mathsf{w}_j \end{aligned}$$

for each element we rewrite $V_K = IX_K + X_K$

$$\int_K \gamma_i\overline{\Delta}^k\mathsf{w}\,d\boldsymbol{x} = \underbrace{\int_K w_i\overline{\Delta}^k\mathsf{w}\,d\boldsymbol{x}}_{IX_K} - \underbrace{\int_K \varphi_i\overline{\Delta}^k\mathsf{w}\,d\boldsymbol{x}}_{X_K}$$

and then we sum $II_K + IV_K + V_K + VI_K + VII_K$ into the residual of element K

$$\mathsf{\Phi}_i^{K(k)} = \int w_i\frac{\overline{\Delta}^k\mathsf{w}}{\Delta t}\,d\boldsymbol{x} + \int w_i\left(\frac{\partial \mathsf{F}^j}{\partial x^j} - \sigma^j\frac{\partial \mathsf{w}}{\partial x^j}\right)^k d\boldsymbol{x} + \int_K \mathcal{S}^k d\boldsymbol{x}$$

we remark that we have computed the fluctuation at the k as it has been defined in (4.7). We miss X_K and $VIII_K$ but they can be summed up together:

$$-\sum_{j\in K} m_{ij}^G\overline{\Delta}^k\mathsf{w}_j - \sum_{K\in\mathcal{D}_i}\sum_{j\in K}\frac{1}{2}m_{ij}^G\frac{\Delta|K|}{|K|}\overline{\Delta}^k\mathsf{w}_j \;=\; -\sum_{j\in K}\left(1 + \frac{1}{2}\frac{\Delta|K|}{|K|}\right)m_{ij}^G\overline{\Delta}^k\mathsf{w}_j$$

Lumping again this matrix:

$$-\sum_{j\in K}\left(1 + \frac{1}{2}\frac{\Delta|K|}{|K|}\right)\frac{|K|}{3}\overline{\Delta}^k\mathsf{w}_j \;=\; -|C_i^{n+1}|\overline{\Delta}^k\mathsf{w}_i$$

We define $\Delta^k - \overline{\Delta}^k = \widetilde{\Delta}^k$, the whole algorithm can be recast in compact form as

$$\widetilde{\Delta}^k\mathsf{w}_i = -\frac{\Delta t}{|C_i^{n+1}|}\sum_{K\in\mathcal{D}_i}\mathsf{\Phi}_i^{K(k)} \tag{D.1}$$

We obtain the two stage, for $k = 1$ the predictor (2.55) and for $k = 2$ the corrector (2.56). ☐

Appendix E

Manifactured source term

We construct a source term to run test case #1 from Williamson *et al.* [1992]. We impose in equation (4.2) the following advection bell travelling over a background irrotatational velocity field $\boldsymbol{a} = a^j \boldsymbol{g}_j$

$$\begin{aligned} h(\boldsymbol{z}, t) &= f(\boldsymbol{z} - \boldsymbol{a}t) \\ a^1 &= 1 \\ a^2 &= 0 \end{aligned} \tag{E.1}$$

called "manifactured" solution. We get a residual:

$$\begin{aligned} \frac{\partial f}{\partial t} + \frac{1}{\sqrt{G}} \frac{\partial}{\partial z^j}\left(\sqrt{G}\, f a^j\right) &= S^h \\ \frac{\partial f a^i}{\partial t} + \frac{1}{\sqrt{G}} \frac{\partial}{\partial z^j}\left(\sqrt{G}\, f a a^{ij}\right) + G^{ij} g f \frac{\partial f}{\partial z^j} + \Gamma^i_{jk} f a a^{jk} &= S^{hu} \end{aligned}$$

The continuity equation is satified exactly:

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial z^j}\left(f a^j\right) + \frac{f a^j}{\sqrt{G}} \frac{\partial}{\partial z^j}\left(\sqrt{G}\right) = 0$$

Since $\frac{\partial G}{\partial z^1} = 0$ we have

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial z^j}\left(f a^j\right) = S^h = 0$$

which is an advection equation exactly satified by the linear wave solution (E.1). For momentum equation

$$\begin{aligned} a^i \frac{\partial f}{\partial t} + \frac{\partial}{\partial z^j}\left(f a a^{ij}\right) + \frac{f a a^{ij}}{\sqrt{G}} \frac{\partial}{\partial z^j}\left(\sqrt{G}\right) + G^{ii} g f \frac{\partial f}{\partial z^i} + \Gamma^i_{km} f a a^{km} &= S^i \\ a^i \left(\frac{\partial f}{\partial t} + \frac{\partial}{\partial z^j}\left(f a^j\right)\right) + G^{ii} g f \frac{\partial f}{\partial z^i} + \Gamma^i_{km} f a a^{km} &= S^i \end{aligned}$$

The terms which are left represents a source term for the Shallow Water system

$$S^i = G^{ii} g f \frac{\partial f}{\partial z^i} + \Gamma^i_{km} f a a^{km}$$

These extra terms provide that (E.1) is an exact solution of the system. The term $g f \nabla f$ balances the nonlinearity nullifying the hydrostatic part and the term $\Gamma_{km} f a a^{km}$ balances the geometrical term. The modified shallow water system writes

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{1}{\sqrt{G}} \frac{\partial}{\partial z^j} \left(\sqrt{G}\, h u^j \right) &= 0 \\ \frac{\partial h u^i}{\partial t} + \frac{1}{\sqrt{G}} \frac{\partial}{\partial z^j} \left(\sqrt{G}\, h u u^{ij} \right) + G^{ij} g h \frac{\partial \eta}{\partial z^j} + \Gamma^i_{jk}\, h u u^{jk} &= S^i \end{aligned}$$

Bibliography

ABGRALL, R., 2006. Essentially non-oscillatory residual distribution schemes for hyperbolic problems. *J. Comput. Phys.*, 214:773–808.

ABGRALL, R., 2011. A review of residual distribution schemes for hyperbolic and parabolic problems: The july 2010 state of the art. *Commun. Comput. Phys.*, 11(4):1043–1080.

ABGRALL, R. and RICCHIUTO, M., 2017. High order methods for cfd. In R. de Borst E. Stein and T. Hughes, editors, *Encyclopedia of Computational Mechanics*. John Wiley and Sons.

ALAUZET, F., 2010. Metric-based anisotropic mesh adaptation. In *CEA-EDF-INRIA Numerical Analysis Summer School*. 14-25 June.

ALAUZET, F., FREY, P., GEORGE, P.L. and MOHAMMADI, B., 2007. 3d transient fixed point mesh adaptation for time dependent problems. application to cfd simulations. *J. Comp. Phys.*, 222:592–623.

ALCRUDO, F. and BENKHALDOUN, F., 2001. Exact solutions to the riemann problem of the shallow water equations with a bottom step. *Computers Fluids*, 30:643–671.

AMMON, C.J., LAY, T., KANAMORI, H. and CLEVELAND, M., 2011. A rupture model of the 2011 off the pacific coast of tohoku earthquake. *Earth Planets Space*, 63:693–696.

ARPAIA, L., RICCHIUTO, M. and ABGRALL, R., 2014. An ale formulation for explicit runge-kutta residual distribution. *J. Sci. Comput.*, 190(34):1467–1482.

BALE, D.S., LEVEQUE, R.J., MITRAN, S. and ROSSMANITH, J.A., 2002. A wave propagation method for conservation laws and balance laws with spatially varying flux function. *J. Sci. Comput.*, 24(3):955–978.

BARTH, T.J., 1996. An energy look at the n scheme. Technical report Working notes, NASA Ames research center, CA, USA,.

Behrens, J., 1996. An adaptive semi-lagrangian advection scheme and its parallelization. *Mon. Weather Rev.*, 124:2386–2395.

Behrens, J., 1998. Atmospheric and ocean modeling with an adaptive finite element solver for the shallow-water equations. *Appl. Numer. Math.*, 26:217–226.

Benoit, M., 2016. Long-distance propagation of tsunamis and surface waves: on the relative importance of dispersion and nonlinearity. In *Tandem and Defi Littoral: tsunami school.* 25-29 May 2016. Bordeaux, France.

Berger, M.J. and Colella, P., 1989. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84.

Berger, M.J., George, D.L., LeVeque, R.J. and Mandli, K.T., 2011. The GeoClaw software for depth-averaged flows with adaptive refinement. *Adv. Water Res.*, 34:1195–1206.

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

Bernetti, R., Titarev, V.A. and Toro, E.F., 2008. Exact solution of the riemann problem for the shallow water equations with discontinuous bottom geometry. *J. Comput. Phys.*, 227:3212–3243.

Bonneton, P., Bonneton, N., Parisot, J-P. and Castelle, B., 2015. Tidal bore dynamics in funnel-shaped estuaries. *J. Geophys. Res.*, 120:DOI: 10.1002/2014JC010267.

Bouchut, F., 2004. *Nonlinear stability of finite volume methods for hyperbolic conservation laws, and well-balanced schemes for sources.* Frontiers in Mathematics. Birkhauser.

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

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

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

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

Budd, C.J., Piggott, M.D. and Williams, J.F., 2009b. Adaptive numerical methods and the geostrophic coordinate transformation. *Mon. Weather Rev.*, 73:266–283.

Cao, W., Huang, W. and Russell, R.D., 1999. An r-adaptive finite element method based upon moving mesh pdes. *J. Comput. Phys.*, 149:221–244.

Caraeni, D. and Fuchs, L., 2002. Compact third-order multidimensional upwind scheme for navier stokes simulations. *Theor. Comput. Fluid Dynam.*, 15:373–401.

Carrier, G.F. and Greenspan, H.P., 1958. Water waves of finite amplitude on a sloping beach. *J. Fluid. Mech.*, 17:97–110.

Castro, M.J., de Luna, T. Morales and Parés, C., 2017. Well-balanced schemes and path-conservative numerical methods. In R. Abgrall and C-W. Shu, editors, *Handook of Numerical Analysis: Handbook of Numerical Methods for Hyperbolic Problems Applied and Modern Issues*, chapter 6. Elsevier B.V.

Ceniceros, H.D. and Hou, T.Y., 2001. An efficient dynamically adaptive mesh for potentially singular solutions. *J. Comput. Phys.*, 172:609–639.

Chadwick, P., 1976. *Continuum Mechanics: Concise Theory and Problems.* Dover Books on Physics, 2nd ed. Edition.

Chen, C., Lai, Z., Beardsley, R.C., Sasaki, J., Lin, J., Lin, H., Ji, R. and Sun, Y., 2014. The march 11, 2011 tohoku m9.0 earthquake-induced tsunami and coastal inundation along the japanese coast: A model assessment. *Prog. Oceanogr.*, 123:84–104.

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

Chertock, A., Dudzinsky, M., Kurganov, A. and Lukacova-Medvidova, M., 2017. Well-balanced schemes for the shallow water equations with coriolis forces. Technical report, North Carolina State University.

de Boor, C., 1973. *Good Approximations by Splines with Variable Knots II*, volume 363. Springer, Berlin.

Deconinck, H. and Ricchiuto, M., 2007. Residual distribution schemes: Foundations and analysis. In John Wiley and Sons, editors, *Encyclopedia of Computational Mechanics.*

Deconinck, H., Sermeus, K. and Abgrall, R., 2000. Status of multidimensional upwind residual distribution schemes and applications in aeronautics. *AIAA Journal*, pages 2000–2328.

Deconink, H., Roe, P.L. and Struijs, R., 1993. A multidimensional of roe flux difference splitter for the euler equations. *Computers Fluids*, 22:215–222.

Delis, A.I., M.Kazolea and Kampanis, N.A., 2008. A robust high-resolution finite volume scheme for the simulation of long waves over complex domains. *Int. J. Numer. Meth. Fluids*, 56:419–452.

Delis, A.I., Nikolos, I.K. and Kazolea, M., 2011. 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.

Di, Y., Li, R., Tang, T. and Zhang, P., 2006. Moving mesh methods for singular problems on a sphere using perturbed harmonic mappings. *J. Sci. Comput.*, 28(4):1490–1508.

Dobes, J. and Deconinck, H., 2008. Second order blended multidimensional upwind residual distribution scheme for steady and unsteady computations. *J. Comput. Appl. Math.*, 215:378–389.

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

Donea, J., Huerta, A., Ponthot, J.P. and Rodriguez-Ferran, A., 2004. Arbitrary lagrangian eulerian methods. In Rene de Borst Erwin Stein and Thomas J.R. Hughes, editors, *Encyclopedia of Computational Mechanics, Ltd*, chapter 14. John Wiley and Sons.

Dutykh, D., Mitsotakis, D., Chubarov, L. and Shokin, Y., 2011. The contribution of horizontal sea-bed displacements into tsunami generation processes. Technical report hal-00530999v2, www.lama.univ-savoie.fr.

Dvinsky, A.S., 1991. Adaptive grid generation from harmonic maps on riemannian manifolds. *J. Comput Phys.*, 95:450–475.

Eell, J. and Sampson, J.H., 1964. Harmonic mappings of riemannian manifolds. *Amer. J. Math.*, 86:109–160.

Etienne, S., Garon, A. and Pelletier, D., 2009. Geometric conservation law and finite element methods for 3d unsteady simulations of incompressible flow. In *39th AIAA Fluid Dynamics Conference*. 22-25 June 2009. San Antonio, Texas.

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

Fujii, Y., Satake, K., Sakai, S., Shinohara, S. and Kanazawa, T., 2011. Tsunami source of the 2011 off the pacific coast of tohoku earthquake. *Earth Planets Space*, 63:815–820.

Gayer, G., Leschka, S., Nohren, I., Larsen, O. and Gunther, H., 2010. Unconditionally stable space–time discontinuous residual distribution for shallow-water flows. *Nat. Hazards Earth Syst. Sci.*, 10:1679–1687.

Geuzaine, C. and Remacle, J-F., 1997-2017. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. `http://www.geuz.org/gmsh`.

Guillard, H. and Farhat, C., 2000. On the significance of the geometric conservation law for flow computations on moving meshes. *Comput. Meth. Appl. Mech. Engrg*, 190:1467–1482.

Hamilton, R., 1975. *Harmonic Maps of Manifolds with Boundary*. Springer-Verlag, New York.

Hervouet, J.M., 2007. *Hydrodynamics of Free Surface Flows: Modelling with the finite element method*. Mechanical Engineering. John Wiley and Sons, Ltd.

Huang, W., 2006. Mathematical principles of anisotropic mesh adaptation. *Comm. Comput. Phys.*, 1:276–310.

Huang, W. and Russell, R.D., 1997. Analysis of moving mesh partial differential equations with spatial smoothing. *J. Numer. Anal.*, 34(3):1106–1126.

Huang, W. and Russell, R.D., 1998. A high dimensional moving mesh strategy. *Appl. Numer. Math.*, 26:63–76.

Huang, W. and Russell, R.D., 1999. Moving mesh strategy based on a gradient flow equation for two-dimensional problems. *J. Sci. Comput.*, 20(3):998.

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

Hubbard, M.E. and Nikiforakis, N., 2003. A three-dimensional, adaptive, godunov-type model for global atmospheric flows. *Mon. Weather Rev.*, 131:1848–1864.

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

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

IIIW-LWRM, 2004. In P.L-F. Liu H. Yeh and C.E. Synolakis, editors, *The Third International Workshop on Long-Wave Runup Models*. June 17-18 2004. Catalina Island, California.

Iinuma, T., Ohzono, M., Ohta, Y. and Miura, S., 2011. Coseismic slip distribution of the 2011 off the pacific coast of tohoku earthquake (m9.0) estimated based on gps data - was the asperity in miyagi-oki ruptured? *Earth Planets Space*, 63:643–648.

IIW-LWRM, 1995. In P.L-F. Liu H. Yeh and C.E. Synolakis, editors, *International Workshop on Long-Wave Runup Models*. 12-17 September 1995. Friday Harbor, USA.

Ioualalen, M., Asavanant, J., Kaewbanjak, N., Grilli, S.T., Kirby, J.T. and Watts, P., 2007. Modeling the 26 december 2004 indian ocean tsunami: Case study of impact in thailand. *J. Geophys. Res.*, 112:1–21.

Isola, D., 2012. *An Interpolation Free Two-Dimensional Conservative ALE scheme over Adaptive Unstructured Grids for Rotorcraft Aerodynamics*. PhD Thesis, Politecnico di Milano. Department of Aerospace Engeneering. Sec. 2.4.1.

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

Ivanenko, S.A, 1999. Harmonic mappings. In J.F. Thompson, B.K. Soni and N.P. Weatherill, editors, *Handbook of Grid Generation*, chapter 8. CRC Press.

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

Kageyama, A. and Sato, T., 2004. “yin-yang grid”: An overset grid in spherical geometry. *Geochem. Geophys. Geosyst.*, 5:Q09005.

KAZOLEA, M., DELIS, A.I. and SYNOLAKIS, C.E., 2014. Numerical treatment of wave-breaking on unstructured finite volume approximations for extended boussinesq-type equations. *J.Comput.Phys*, 271:281–305.

KIM, S.K., LIU, P.L-F. and LIGGET, J.A., 1983. Boundary integral equation solutions for solitary wave generation propagation and run-up. *Cwstal Engng*, 7:299–317.

KURGANOV, A. and LEVY, D., 2002. Central-upwind schemes for the Saint-Venant system. *ESAIM: M2AN*, 36(3):397–425.

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

LANNES, D., 2016. Pde modelling (parameters, notion of asymptotics). In *Tandem and Defi Littoral: tsunami school.* 25-29 May 2016. Bordeaux, France.

LAUTER, M., GIRALDO, F.X., HANDORF, D. and DETHLOFF, K., 2008. A discontinuous galerkin method for the shallow water equations in spherical triangular coordinates. *J. Comput. Phys.*, 227:10226–10242.

LAX, P. D., 1972. Hyperbolic system of conservation laws and the mathematical theory of shock waves. *SIAM Conference Lecture in Applied Mathematics*, 11.

LAX, P. D. and WENDROFF, B., 1960. Systems of conservation laws. *Comm. Pure Appl. Math.*, 13:217–237.

LEGAL, M., 2017. *Blabla.* PhD Thesis, Institute of Physics.

LESOINNE, M. and FARHAT, C., 1996. Geometric conservation laws for flow problems with moving boundaries and deformable meshs, and their impact on aeroelastic computations. *Comput. Meth. Appl. Mech. Engrg*, 134:71–90.

LEVEQUE, R.J., 1997. Wave propagation algorithm for multidimensional hyperbolic systems. *J. Comput. Phys.*, 131:3279–353.

LEVEQUE, R.J., 1998. Balancing source terms and flux gradients in high-resolution godunov methods. *J. Comput. Phys.*, 146:346–365.

LEVEQUE, R.J., 2004. *Finite Volume Methods for Hyperbolic problems.* Cambridge University Press.

LI, R., TANG, T. and ZHANG, P., 2002. A moving mesh finite element algorithm for singular problems in two and three space dimensions. *J. Comput. Phys.*, 177(2):365–393.

Li, S.T. and Petzold, L.R., 1997. Moving mesh methods with upwinding schemes for time dependent pdes. *J. Comput Phys.*, 131:368–377.

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

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

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

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

Løvholt, F., Kaiser, G., Glimsdal, S., Scheele, L., Harbitz, C.B., and Pedersen, G., 2012. Modeling propagation and inundation of the 11 march 2011 tohoku tsunami. *Nat. Hazards Earth Syst. Sci.*, 12:1017–1028.

Machenhauer, B., Kaas, E. and Lauritzen, P.H., 2009. Finite-volume methods in meteorology. In R.M. Temam and J.J. Tribbia, editors, *Handook of Numerical Analysis: Computational Methods for the Atmosphere and the Oceans*, chapter 14. Elsevier B.V.

MacInnes, B.T., Gusman, A.R., LeVeque, R.J. and Tanioka, Y., 2013. Comparison of earthquake source models for the 2011 tohoku event using tsunami simulations and near-field observations. *Bull. Seismol. Soc. Am.*, 103:1256–1274.

Mai, P.M. and Beroza, G.C, 2002. A spatial random field model to characterize complexity in earthquake slip. *J. Geophys. Res-Sol. Ea.*, 107(10):10–21.

Mantegazza, P., 2012. Balance laws in eulerian and lagrangian form. In *Dynamics and control of aerospace strucures, Lecture Notes*, chapter 1. Politecnico di Milano. Department of Aerospace Engeneering.

Mavriplis, D.J. and Yang, Z., 2006. Construction of the discrete geometric conservation law for high-order time-accurate simulations on dynamic meshes. *J. Comput. Phys.*, 213:557–573.

McCorquodale, P., Ullrich, P.A., Johansen, H. and Colella, P., 2015. Adaptive grid refinement for two-dimensional and three-dimensional nonhydrostatic atmoshperic flows. *Adv. Water Resour.*, 10(2):121–162.

Michler, C., Sterck, H. De and Deconinck, H., 2002. An arbitrary lagrangian eulerian formulation for residual distribution schemes on moving grids. *Computers Fluids*, 32:59–71.

Mori, N. and Takahashi, T., 2012. The 2011 tohoku earthquake tsunami joint survey group (2012), nationalwide post event survey and analysis of the 2011 tohoku earthquake tsunami. *Coast. Eng. J.*, 54.

Nguyen-Shafer, H. and Schmidth, J-P., 2014. *Tensor Analysis and Elementary Differential Geometry for Physisict and Engineers*, volume 21 of *Mathematical Engineering*. Springer.

Ni, G., Jiang, S. and Xu, K., 2009. Remapping-free ale-type kinetic method for flow computations. *J. Comp. Phys.*, 228:3154–3171.

Nikolos, I.K. and Delis, A.I., 2009. 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.

NOAA, 1993. Tsunami runup onto a complex three-dimensional beach; monai valley. `http://nctr.pmel.noaa.gov/benchmark/Laboratory/Laboratory_MonaiValley/index.html`.

NOAA, 2017a. National data buoy center. http://www.ndbc.noaa.gov/dart.shtml.

NOAA, 2017b. Tsunami vocabulary and terminology. http://www.tsunami.noaa.gov/terminology.html.

Nouveau, L. and Ricchiuto, M., 2016. Ale methods for predictor corrector time integrator. Technical report, INRIA Bordeaux-Sud Ouest.

NOWPHAS, 2017. Nationwide ocean wave information network for ports and harbours. http://nowphas.mlit.go.jp/nowphasdata/sub301.htm.

Nwogu, O., 1994. An alternative form of the boussinesq equations for nearshore wave propagation. *J. Waterw. Port C. Div.*, pages 618–638.

Okada, Y., 1985. Surface deformation due to shear and tensile faults in a half-space. *Bull. Seismol. Soc. Am.*, 75(4):1135–1154.

Osher, S., 1984. Riemann solvers, the entropy condition and difference approximations. *SIAM J. Numer. Anal.*, 21:217–235.

Paillere, H., Degrez, G. and Deconinck, H., 1998. Multidimensional upwind schemes for the shallow water equations. *Int. J. Numer. Meth. Fl.*, 26(8):987–1000.

PEREGRINE, D.H., 1967. Long waves on a beach. *J. Fluid. Mech.*, 27.

PERTHAME, B. and SHU, C.W., 1996. On positivity preserving finite volume schemes for euler equations. *Numer. Math.*, (73):119–130.

PONS, J.A., FONT, J.A., IBANEZ, J.M. and J.M. MARTI, J.A. Miralles, 1998. General relativistic hydrodynamics with special relativistic riemann. *Astron. Astrophys.*, 339:638–642.

POPINET, S., 2011. Quadtree-adaptive tsunami modelling. *Ocean Dynamics*, 61(9):1261–1285.

POPINET, S., 2012. Adaptive modelling of long-distance wave propagation and fine-scale flooding during the tohoku tsunami. *Nat. Hazards Earth Syst. Sci.*, 12(4):1213–1227.

RICCHIUTO, M., 2005. *Contruction and analysis of compact residual discretizations for conservation laws on unstructured meshes.* PhD Thesis, von Karman Institute for Fluid Dynamics and Universite Libre de Bruxelles.

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

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

RICCHIUTO, M., ABGRALL, R. and DECONINCK, H., 2007. Application of conservative residual distribution schemes to the solution of the shallow water equations on unstructured meshes,. *J. Comput. Phys.*, 222:287–331.

RICCHIUTO, M. and BOLLERMAN, A., 2009. Stabilized residual distribution for shallow water simulations. *J.Comput.Phys*, 228:1071–111.

RINGLER, T.D. and RANDALL, D.A., 2002. A potential enstrophy and energy conserving numerical scheme for solution of the shallow-water equations on a geodesic grid. *Mon. Wea. Rev.*, 130:1397–1410.

RITCHIE, H., 1988. Application of the semi-lagrangian method to a spectral model of the shallow water equations. *Mon. Weather Rev.*, 116:1587–1598.

ROE, P.L., 1982. Fluctuations and signals - a framework for numerical evolution problems. In K.W. Morton and M.J. Baines, editors, *Numerical Methods for Fluids Dynamics*, pages 219–257. Academic Press.

ROE, P.L., 1987. Linear advection schemes on triangular meshes. Technical report CoA 8720, Cranfield Institute of Technology.

Roe, P.L. and Sidilkover, D., 1992. Optimum positive linear schemes for advection in 2 and 3 dimensions. *SIAM J. Numer. Anal.*, 29(6):1542–1568.

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

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

Ronchi, C., Iacono, R. and Paolucci, P.S., 1996. The "cubed sphere": A new method for the solution of partial differential equations in spherical geometry. *J. Comput. Phys.*, 124:93–114.

Rossmanith, J.A., 2013. Residual distribution schemes for hyperbolic balance laws in generalized coodinates. In *Numerical Modeling of Space Plasma Flows*, volume 359 of *ASP Conference Series*. Nikolai V. Pogorelov and Gary P. Zank.

Rossmanith, J.A., Bale, D.S. and LeVeque, R.J., 2004. A wave propagation algorithm for hyperbolic systems on curved manifold. *J. Comput. Phys.*, 199:631–662.

Russo, G., 2001. 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. ISBN 978-3-0348-9538-5.

Satake, K., Fujii, Y., Harada, T. and Namegaya, Y., 2013. Time and space distribution of coseismic slip of the 2011 tohoku earthquake as inferred from tsunami waveform data. *Bull. Seismol. Soc. Am.*, 103:1473–1492.

Savidis, S.A., Aubram, D. and Rackwit, F., 2008. Arbitrary lagrangian-eulerian finite element formulation for geotechnical construction processes. *Journal of Theoretical and Applied Mechanics*, 38:165–174.

Schoen, R. and Yau, S.T., 1978. On univalent harmonic maps between surfaces. *Invent. Math.*, 44:265–278.

Seaid, M., 2004. Non-oscillatory relaxation methods for the shallow water equations in one and two space dimensions. *Int. J. Numer. Meth. Fl.*, 46:457–484.

Shao, G., Li, X., Ji, C. and Maeda, T., 2011. Focal mechanism and slip history of the 2011 m9.1 off the pacific coast of tohoku earthquake, constrained with teleseismic body and surface waves. *Earth Planets Space*, 63:559–564.

Shimozono, T., Sato, S., Okayasu, A., Tajima, Y., Fritz, H.M, , Liu, H. and Takagawa, T., 2012. Propagation and inundation characterstics of the 2011 tohoku tsunami on the central sanriku coast. *Coast. Eng. J.*, 54:1017–1028.

Sielecki, A. and Wurtele, M.G., 1970. The numerical integration of the nonlinear shallow-water equations with sloping boundaries. *J. Comp. Phys.*, 6:219–236.

Skamarock, W.C. and Klemp, J.B., 1992. An adaptive multiblock high-order finite-volume method for solving the shallow-water equations on the sphere. *Mon. Weather Rev.*, 121:788–804.

Sorensen, O.R. and Madsen, P.A., 1992. A new form of the boussinesq equations with improved linear dispersion characteristics. *Coast. Eng. J.*, 18:183–204.

Spekreijse, S.P., 1999. Elliptic generation systems. In J.F. Thompson, B.K. Soni and N.P. Weatherill, editors, *Handbook of Grid Generation*, chapter 4. CRC Press.

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

Synolakis, C.E., 1987. The runup of solitary waves. *J. Fluid. Mech.*, 185:523–545.

Sármány, D., Hubbard, M.E. and Ricchiuto, M., 2013. Unconditionally stable space–time discontinuous residual distribution for shallow-water flows. *J. Comput. Phys.*, 253:86–113.

Tadmor, E., 1987. The numerical viscosity of entropy stable schemes for systems of conservation laws. *Math. Comput.*, 49(179):91–103.

Tadmor, E. and Zhong, W., 2006. Energy-preserving and stable approximations for the two-dimensional shallow water equations. In Hans Munthe-Kaas and Brynjulf Owren, editors, *Abel Symposium*, volume 3, pages 67–94. 25-27 May 2006. Alesund, Denmark.

Tan, Z., 2007. Adaptive moving mesh methods for two-dimensional resistive magneto-hydrodynamic pde models. *Computers Fluids*, 36:758–771.

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

TANIOKA, Y. and SATAKE, K., 1996. Tsunami generation by horizontal displacement of ocean bottom. *Geophysical Research Letters*, 23:861–864.

THOMAS, P.D. and LOMBARD, C.K, 1979. Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal*, pages 1030–1037.

THOMPSON, J.F., SONI, B.K. and WEATHERILL, N.P., 1999. *Handbook of Grid Generation*. CRC Press.

TORO, E.F. and GARCIA-NAVARRO, P., 2007. Godunov-type methods for free surface shallow water flows: a review. *J.Hydraul.Res*, 45:746.

ULLRICH, P.A., JABLONOWKSI, C. and VAN-LEER, B., 2010. High-order finite-volume methods for the shallow water equations on the sphere. *J. Comput. Phys.*, 229:6104–6134.

VAN-ALBADA, G.D., VAN-LEER, B. and ROBERTS, W.W., 1982. A compartive study of computational methods for cosmeic gas dynamics. *Astron. Astrophys.*, (108):76–84.

VAN-LEER, B., 1979. Towards the ultimate conservative difference scheme v. a second-order sequel to godunov's method. *J. Comput. Phys.*, (32).

VAN ZWIETEN, G.J., HANSSEN, R.F. and GUTIÉRREZ, M.A., 2013. Overview of a range of solution methods for elastic dislocation problems in geophysics. *J. Geophys. Res-Sol. Ea.*, 118:1–12.

WARSI, Z.U.A., 1999. Mathematics of space and surface grid generation. In J.F. Thompson, B.K. Soni and N.P. Weatherill, editors, *Handbook of Grid Generation*, chapter 2. CRC Press.

WELLER, H., BROWNE, P., BUDD, C. and CULLEN, M., 2016. Mesh adaptation on the sphere using optimal transport and the numerical solution of a monge–ampère type equation. *J. Comput. Phys.*, 308:102–123.

WILLIAMSON, D.L., DRAKE, J.B., HACK, J.J., R.JAKOB and SWARZTRAUBER, P.N., 1992. A standard test set for numerical approximations to the shallow water equations in spherical geometry. *J. Comput. Phys.*, 102:211–224.

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

XING, Y. and SHU, C-W., 2014. A survey of high order schemes for the shallow water equations. *J. Math. Study*, (47):1–28.

Yanagisawa, K., Imamura, F., Sakakiyama, T., Annaka, T., Takeda, T. and Shuto, N., 2007. Tsunami assessment for risk management at nuclear power facilities in japan. *Pure Appl. Geophys.*, 164:565–576.

Yang, C. and Cai, X.C., 2011. A parallel well-balanced finite volume method for shallow water equations with topography on the cubed-sphere. *J. Comput. Appl. Math.*, 235:5357–5366.

Zhang, X. and Shu, C-W., 2010. On maximum-principle-satisfying high order schemes for scalar conservation laws. *J. Comput. Phys.*, (229):3091–3120.

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

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