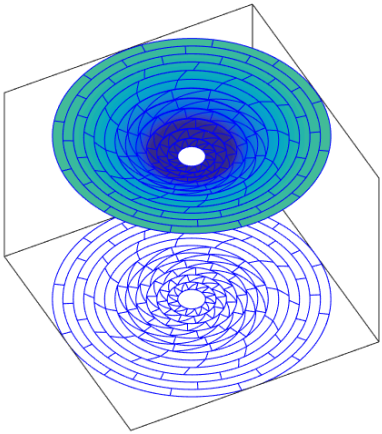
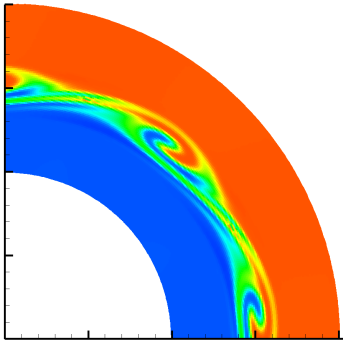
Doctoral School in Mathematics

Well balanced Arbitrary-Lagrangian-Eulerian Finite Volume schemes on moving nonconforming meshes for non-conservative Hyperbolic systems

Elena Gaburro





UNIVERSITÀ DEGLI STUDI DI TRENTO
Dipartimento di Matematica

2018

Doctoral thesis in **Applied Mathematics**, **XXX cycle**.

Department of Mathematics,
University of Trento.

Academic year **2017/2018**.

Supervisor : **Prof. Dr.-Ing. Michael Dumbser**

University of Trento
Trento, Italy
2018

"*Behind every result is a new challenge*"

Preface

This thesis has been developed during the three-year Doctoral Program of the Doctoral School in Mathematics at the University of Trento in agreement with the University of Verona (Italy).

The presented research has been partially funded by the European Research Council (ERC) under the European Union's Seventh Framework Programme (FP7/2007-2013) with the research project *STiMulUs*, ERC Grant agreement no. 278267. Most of the work has been carried out at the Department of Mathematics of the University of Trento under the supervision of Prof. Dr.-Ing. Michael Dumbser.

Moreover, I spent four months (January - April 2017) at the Laboratoire Jacques-Louis Lions of the University Pierre et Marie Curie, Paris (France), collaborating with Prof. Bruno Després and Dr. Stéphane Del Pino, funded by the Erasmus+ Programme of the European Union.

Finally, I worked for seven months (May 2016 and June-November 2017) with Prof. Manuel J. Castro and Prof. Carlos Parés at the University of Málaga (Spain). In particular, I obtained a six months S-ESR Marie Sklodowska-Curie ITN fellowship with the research project ModCompShock under the European Union's Horizon 2020 Research and Innovation Programme, Grant agreement no. 642768.

Trento, February 2018

Elena Gaburro

Acknowledgments

First of all, I would like to thank my supervisor Michael Dumbser for introducing me in a so wide scientific community letting me know lots of interesting people, for his smart inputs and expert knowledge, and in particular for his enthusiasm which I have extremely appreciated and that motivated me throughout my thesis.
Many thanks also to Manuel J. Castro and Christian Klingenberg for having accepted to carefully review this manuscript and together with Bruno Després and Vincenzo Casulli for taking part to my evaluation committee.

I am very grateful to Manuel J. Castro and Carlos Parés for welcoming me in Málaga in a familiar atmosphere, and for the patience and expertise through which they answered my questions and solved my doubts. I must express my gratitude to Bruno Després and Sthépane Del Pino for the enriching collaboration and all the time spent working together. My gratitude goes also to Francesca Rapetti and Victorita Dolean who encouraged and supported me at the very beginning of my scientific researches in Nice.
I extend my thanks to my past professors Zeffiro Mari, Ada Gallina, Sisto Baldo, Leonard Peter Bos and Marco Caliari for making me realize the power of Mathematics, and to Giandomenico Orlandi for his subtle hints and the willingness shown with my troubled academic choices.

Of course, I cannot forget the holidays and the nights spent with my closest friends Debora and Giulia, and Carlotta, Elena, Filippo, Guglielmo, Luca & Luca, Matteo & Matteo, Pierluigi, Riccardo and Simone. Very special thanks go also to Ernesto, Marco and Mauro for the pleasant office days, to Tommaso for the unexpected friendship without which my stay in Paris would not have been the same, and to Marcella for sharing with me free and study time, and to be always a reliable source of scientific, bureaucratic and linguistic advices.

Finally, I am grateful to my family, both to those that do not know nothing about me but they are proud of me and those that have always been supportive and helpful, in particular my grandmother Rosy and my mother Emanuela who experienced all of the ups and downs of my research and help me keep things in perspective.

Grazie a tutti, Merci à tous, Gracias a todos, Danke euch allen!

Contents

Symbols

c	Speed of sound
d	Number of space dimensions
i	Element index
n	Current time level
$n+1$	Future time level
t	Time
T	Control volume
$\|T\|$	Length, area or volume (respectively $d=1,2,3$) of T
x	Horizontal direction in the physical system
y	Lateral direction in the physical system
z	Vertical direction in the physical system
β	Linear basis function
γ	Ratio of specific heats
Δt	Time step
ρ	Density
ρE	Total energy per mass unit
p	Pressure
$\mathbf{u}$	Vector of velocity
ξ	Horizontal direction in the reference system
η	Lateral direction in the reference system
ζ	Vertical direction in the reference system
τ	Relative time with respect to time level t^n
χ_1, χ_2	Face parameters on the element boundary
Ω	Computational domain
$\mathbf{B}$	Non-conservative nonlinear flux tensor
$\mathbf{F}$	Conservative nonlinear flux tensor
$\mathbf{I}$	Identity matrix
$\mathbf{n}$	Element boundary normal vector in space
$\tilde{\mathbf{n}}$	Element boundary normal vector in space and time
$\mathcal{O}$	Order of the scheme
$\mathcal{V}_i^n$	Voronoi neighbors of element i at time t^n

$\mathcal{S}_i^n$	Edge neighbors of element i at time t^n
$\mathbf{Q}$	Vector of conserved variables
$\mathbf{S}$	Algebraic source term
$\mathbf{V}$	Vector of the local mesh velocity
$\mathbf{x}$	Physical spatial coordinate vector
$\boldsymbol{\xi}$	Reference spatial coordinate vector

Abbreviations

ADER	Arbitrary high order scheme using derivatives
ALE	Arbitrary-Lagrangian-Eulerian scheme
AMC	Angular Momentum Preserving
CFL	Courant-Friedrichs-Levy number
CPU	Central processing unit
CUDA	Compute Unified Device Architecture
ENO	Essentially non-oscillatory
FV	Finite Volume
GCL	Geometric conservation law
GPU	Graphics Processing Unit
LTS	Local time stepping
ODE	Ordinary differential equation
PDE	Partial differential equation
TVD	Total variation diminishing
WB	Well balanced
WENO	Weighted essentially non-oscillatory

Abstract

This PhD thesis presents a novel second order accurate *direct* Arbitrary-Lagrangian-Eulerian (ALE) Finite Volume scheme for nonlinear hyperbolic systems, written both in *conservative* and *non-conservative* form, whose peculiarity is the *nonconforming* motion of interfaces. Moreover it has been coupled together with specifically designed path-conservative well balanced (WB) techniques and angular momentum preserving (AMC) strategies. The obtained result is a method able to preserve many of the physical properties of the system: besides being conservative for mass, momentum and total energy, also any known steady equilibrium of the studied system can be exactly maintained up to machine precision. Perturbations around such equilibrium solutions are resolved with high accuracy and minimal dissipation on moving contact discontinuities even for very long computational times.

The core of our ALE scheme is the use of a space-time conservation formulation in the construction of the final Finite Volume scheme: the governing PDE system is rewritten at the aid of the space-time divergence operator and then a fully discrete one-step discretization is obtained by integrating over a set of closed space-time control volumes. In order to avoid the typical mesh distortion caused by *shear flows* in Lagrangian-type methods, we adopt a nonconforming treatment of sliding interfaces, which requires the dynamical insertion or deletion of nodes and edges, and produces hanging nodes and space-time faces shared between more than two cells. In this way, the elements on both sides of the shear wave can move with a different velocity, without producing highly distorted elements, the mesh quality remains high and, as a direct consequence, also the time step remains almost constant in time, even for highly sheared vortex flows. Moreover, due to the space-time conservation formulation, the geometric conservation law (GCL) is *automatically* satisfied by *construction*, even on moving nonconforming meshes.

Our nonconforming ALE scheme is especially well suited for modeling in polar coordinates *vortical flows* affected by strong differential rotation: in particular, the novel *combination* with the well balancing make it possible to obtain

great results for challenging astronomical phenomena as the rotating Keplerian disk. Indeed, we have formulated a new HLL-type and a novel Osher-type flux that are both able to guarantee the well balancing in a gas cloud rotating around a central object, maintaining up to machine precision the equilibrium between pressure gradient, centrifugal force and gravity force that characterizes the stationary solutions of the Euler equations with gravity. To the best knowledge of the author this work is original for various reasons: it is the first time that the little dissipative Osher scheme is modified in order to be well balanced for non trivial equilibria, and it is the first time that WB is coupled with ALE for the Euler equations with gravity; moreover the use of a well balanced Osher scheme joint with the Lagrangian framework allows, for the first time within a Finite Volume method, to maintain exactly even *moving* equilibria.

In addition, the introduced techniques demonstrate a wide range of applicability from steady vortex flows in shallow water equations to complex free surface flows in two-phase models. In the last case, studied on fixed Cartesian grids, the new well balanced methods have been implemented in *parallel* exploiting a *GPU*-based platform and reaching the very high *efficiency* of ten million of volumes processed per seconds.

Finally, in the case of vortical flows we propose a preliminary analysis on how to increase the accuracy of the method by exploiting the *redundant conservation law* that can be written for the angular momentum, as proposed in [56]. Indeed, an easy manipulation of the Euler equations allows to write its additional conservation law: clearly it does not add any supplementary information from the analytical point of view, but from a numerical point of view it provides extra information in particular in the case of rotating systems. We present both a master-slave approach, to deduce *a posteriori* a more precise approximation of the velocity, and some coupled approaches to investigate how the entire process can take advantage from considering *directly* the angular momentum during the computation within a strong coupling with other variables.

A large set of different numerical tests has been carried out in order to check the accuracy and the robustness of the new methods for both smooth and discontinuous problems, close and far away from the equilibrium, in one and two space dimensions. Many of the presented results show a great *enhancement* with respect to the state of the art.

1 Introduction

The development of new reliable and effective numerical methods for solving hyperbolic partial differential equations (PDEs) has become a central discipline in present-day computational science. Indeed, the variety and broad applicability of hyperbolic equations cover a wide range of interesting phenomena, from environmental, geological and electromagnetic problems to blood flow in the human circulation system, as well as computational fluid dynamics, seismic wave propagation and astrophysical applications, including magnetized plasma flows and even general relativity. And, since exact analytical solutions are rarely available, a continuously increasing interest is focused on numerical strategies able to model such complex situations.

In particular, the Finite Volume (FV) method has been the most used approach to discretize hyperbolic equations in the last decades. The main idea consists in replacing the continuous problem given by the PDEs on a domain Ω by a finite set of discrete values, which represent the average of the considered physical quantities on each portion, called indeed *control volume*, of a partition of Ω. These averages are then evolved in time through the integral form of the governing equations and a specific *numerical flux* function that resolves the discontinuities at the interfaces between two control volumes. From its introduction, that dates back before 1960 with the seminal works of Godunov, Lax, Von Neumann, Richtmyer and Wendroff [86, 104, 105, 160], until today many improvements have been proposed in order to adapt the method to any specific situation and increase its overall accuracy and efficiency. Between the mayor enhancements, which would be impossible to list all here, I would like to recall those that have been the starting point for the research presented in this PhD thesis: first of all the overcoming of the Godunov theorem (which states that linear monotone schemes for hyperbolic PDE are at most of order one) that has led to significant progress on *high order schemes* both on structured and unstructured grids in any dimension of space; secondly, the introduction and developments of the complex *Lagrangian framework* that allows to reduce the errors due to convective terms paying the price of a more complex moving mesh, whose quality is often very difficult to control; and finally, the growing success of the *well balanced* techniques which increment the power of resolution

of the method close to steady state solutions.

This work focuses on the development of numerical methods able to maintain *exactly* even at discrete level many of the physical properties of the studied hyperbolic systems, with a particular interest for vortical flows. With this aim we have developed a completely new class of well balanced Arbitrary-Lagrangian-Eulerian Finite Volume schemes on moving nonconforming meshes. In this chapter, after having introduced the basic notation in Section 1.1, we recall the state of the art on Lagrangian-type methods (Section 1.2), on path-conservative well balance techniques (Section 1.3), and on vortical flows (Section 1.4). Finally, Section 1.5 provides a summary of the PhD thesis.

1.1 Finite volume schemes for nonlinear hyperbolic systems

We consider here a very general formulation of the governing equations in order to model a wide class of physical phenomena, namely all the ones which are governed by equations that can be cast into the following form,

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{Q}) + \mathbf{B}(\mathbf{Q}) \cdot \nabla \mathbf{Q} = \mathbf{S}(\mathbf{Q}), \quad \mathbf{x} \in \Omega(t) \subset \mathbb{R}^d, \ t \in \mathbb{R}_0^+. \tag{1.1}$$

In this system, $\mathbf{x}$ is the spatial position vector, $d = [1, 2, 3]$ denotes the number of space dimensions, t represents the time, and $\Omega(t)$ is the computational domain at time t. $\mathbf{Q}(\mathbf{x}, t) = (q_1(\mathbf{x}, t), q_2(\mathbf{x}, t), \ldots, q_\nu(\mathbf{x}, t))$ is the vector of the conserved variables defined in the space of the admissible states $\Omega_{\mathbf{Q}} \subset \mathbb{R}^\nu$, $\mathbf{F}(\mathbf{Q}) = (\mathbf{f}(\mathbf{Q}), \mathbf{g}(\mathbf{Q}), \mathbf{h}(\mathbf{Q}))$ is the nonlinear flux tensor, $\mathbf{B}(\mathbf{Q}) = (\mathbf{B}_1(\mathbf{Q}), \mathbf{B}_2(\mathbf{Q}), \mathbf{B}_3(\mathbf{Q}))$ is a matrix collecting the non-conservative terms, and $\mathbf{S}(\mathbf{Q})$ represents a nonlinear algebraic source term. The system (1.1) can also be written in the following quasi-linear form

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{A}(\mathbf{Q}) \cdot \nabla \mathbf{Q} = \mathbf{S}(\mathbf{Q}), \quad \mathbf{x} \in \Omega(t) \subset \mathbb{R}^d, \ t \in \mathbb{R}_0^+, \tag{1.2}$$

with the system matrix $\mathbf{A}(\mathbf{Q}) = \partial \mathbf{F} / \partial \mathbf{Q} + \mathbf{B}(\mathbf{Q})$. The system is *hyperbolic* if for any normal direction $\mathbf{n} \neq \mathbf{0}$ the matrix $\mathbf{A}(\mathbf{Q}) \cdot \mathbf{n}$ has ν real eigenvalues and a full set of ν linearly independent eigenvectors for all $\mathbf{Q} \in \Omega_{\mathbf{Q}}$. PDE systems like (1.1) include as particular cases systems of *conservation laws* ($\mathbf{B} = \mathbf{0}$, $\mathbf{S} = \mathbf{0}$), systems of conservation laws with source terms or *balance laws* ($\mathbf{B} = \mathbf{0}$), and even *non-conservative* hyperbolic systems ($\mathbf{B} \neq \mathbf{0}$). They appear in many different physical models: in particular, in this thesis we will take into account the shallow water equations, the Euler equations of gas dynamics

with and without gravity and a simplification of the Baer Nunziato models for compressible multi-phase flows.

In order to compute a numerical solution of (1.1) one can use a Finite Volume scheme. To do so, one has to introduced a discretization of the domain Ω at time t^n done with a total number N_E of elements T_i^n $i = 1, \dots, N_E$, called *spatial control volumes*, with volume $|T_i^n|$. Then (1.1) is integrated in space over each T_i^n and in time from t^n to the next time step $t^{n+1} = t^n + \Delta_t$

$$\int_{t^n}^{t^{n+1}} \int_{T_i^n} \frac{\partial \mathbf{Q}}{\partial t} \, d\mathbf{x}dt + \int_{t^n}^{t^{n+1}} \int_{T_i^n} \nabla \cdot \mathbf{F}(\mathbf{Q}) + \mathbf{B}(\mathbf{Q}) \cdot \nabla \mathbf{Q} \, d\mathbf{x}dt = \int_{t^n}^{t^{n+1}} \int_{T_i^n} \mathbf{S}(\mathbf{Q}) \, d\mathbf{x}dt. \tag{1.3}$$

For this introductory part of the thesis we restrict ourselves to the case $\mathbf{B} = \mathbf{0}$. The case $\mathbf{B} \neq \mathbf{0}$ will be introduced in Section 1.3 and developed in Chapters 3-5. Now, by applying the Gauss theorem, the volume integral of the flux can be rewritten as a surface integral

$$\int_{t^n}^{t^{n+1}} \int_{T_i^n} \frac{\partial \mathbf{Q}}{\partial t} \, d\mathbf{x}dt + \int_{t^n}^{t^{n+1}} \int_{\partial T_{ij}^n} \mathbf{F}(\mathbf{Q}) \cdot \mathbf{n}_{ij} \, d\mathbf{x}dt = \int_{t^n}^{t^{n+1}} \int_{T_i^n} \mathbf{S}(\mathbf{Q}) \, d\mathbf{x}dt, \tag{1.4}$$

where ∂T_{ij}^n represents the boundary between T_i^n and its Voronoi neighbors $T_j^n \in \mathcal{V}_i^n$, $\mathbf{n}_{ij}$ the outward-pointing unit normal vector, and $|\partial T_{ij}^n|$ the area of the boundary. Fundamental definitions in this context are those of the *cell-average* at time t^n

$$\mathbf{Q}_i^n = \frac{1}{|T_i^n|} \int_{T_i^n} \mathbf{Q}(\mathbf{x}, t^n) d\mathbf{x}, \tag{1.5}$$

and of the *time-averaged flux* at the boundary ∂T_{ij}^n

$$\mathbf{F}_{ij}^n \cdot \mathbf{n}_{ij} = \frac{1}{\Delta_t} \int_{t^n}^{t^{n+1}} |\partial T_{ij}^n| \mathbf{F}(\mathbf{Q}(\mathbf{x}|_{\partial T_{ij}^n}, t^n)) \cdot \mathbf{n}_{ij} \, dt. \tag{1.6}$$

By substituting (1.5) and (1.6) into (1.4) the following *exact* relation can be derived

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \frac{\Delta t}{|T_i^n|} \sum_{T_j^n \in \mathcal{V}_i^n} |\partial T_{ij}^n| \, \mathbf{F}_{ij}^n \cdot \mathbf{n}_{ij}. \tag{1.7}$$

So far, no numerical scheme has been introduced yet and the exact solution of (1.1) satisfies also (1.7) exactly, but in this context it is in general impossible to compute (1.6) exactly because $\mathbf{Q}(\mathbf{x}|_{\partial T_{ij}^n}, t^n)$ could be not available. Hence, the main ingredient in order to construct a Finite Volume scheme is to define

the *numerical* flux function at the interface $\mathbf{F}_{ij}^n \cdot \mathbf{n}_{ij}$ as function of the cell averages $\mathbf{Q}_i^n$ and $\mathbf{Q}_j^n$, since in the Finite Volume framework only the cell-averaged solution is known. And for this last reason also the approximation of the boundary values $\mathbf{Q}(\mathbf{x}|_{\partial T_{ij}^n}, t^n)$ plays an important role in the structure of the method: the procedure through which $\mathbf{Q}(\mathbf{x}|_{\partial T_{ij}^n}, t^n)$ is obtained is called *reconstruction step*.

Defining and improving the numerical fluxes and the reconstruction step, together with many other enhancements, make the *history* of the Finite Volume methods: from the first works of Godunov in $d = 1$ and $\mathbf{B} = \mathbf{0}$ up to very recent researches. For a general and complete introduction to the topic, in particular for the theoretical framework, we refer to the classical books of Toro [150], LeVeque [107], Hirsch [92, 93], Kroner [100] and Godlewski Raviart [85]. Here we recall only some fundamental stages, giving the theoretical framework, in particular basic concepts on existence and uniqueness of weak solutions in the case of Riemann problems, as known.

Godunov was the first that suggested to obtain the numerical fluxes by solving Riemann problems at each interface, in order to resolve the discontinuities. Early work was about the *exact* solution of the Riemann problem [54, 86]; then a large number of *approximate* Riemann solvers were developed, such as the linearized Riemann solver of Roe [138], the HLL and HLLE Riemann solvers [78, 91] and the local Lax-Friedrichs (LLF) solver proposed by Rusanov [140]. While the above-mentioned HLL schemes are very robust, they smear out contact discontinuities. An improvement was made by Einfeldt and Munz in [79] with the introduction of the HLLEM Riemann solver, where the intermediate state was assumed piecewise linear instead of piecewise constant. Another well-known improvement of the original HLL scheme is due to Toro et al. in [152] with the design of the HLLC Riemann solvers that use an enhanced wave model that is able to capture also the intermediate contact wave. In [127] Osher et al. introduced a class of approximate Riemann solvers based on path integrals, where the paths were obtained by an approximation of the solution of the Riemann problem by rarefaction fans. A simpler and more general version of the Osher flux has recently been forwarded by Dumbser and Toro in [74, 75]. All those one-dimensional Riemann solvers can be used even in two and three-dimensional problems, where the discontinuities are resolved at each boundary of the control volume along the normal direction.

For what concerns the reconstruction step, the simplest solution consists in supposing $\mathbf{Q}(\mathbf{x}, t)$ piecewise constant with value $\mathbf{Q}_i^n$ within each cell, so that the boundary values $\mathbf{Q}(\mathbf{x}|_{\partial T_{ij}^n}^-, t^n)$ and $\mathbf{Q}(\mathbf{x}|_{\partial T_{ij}^n}^+, t^n)$ coincide respectively with

$\mathbf{Q}_i^n$ and $\mathbf{Q}_j^n$. Note that $\mathbf{Q}(\mathbf{x}|^-_{\partial T_{ij}^n}, t^n)$ represents the values at the left of the interface, i.e. seen from $\mathbf{Q}_i^n$, and $\mathbf{Q}(\mathbf{x}|^+_{\partial T_{ij}^n}, t^n)$ represents the values at the right of the interface, i.e. seen from $\mathbf{Q}_i^n$. This choice leads to first order methods. In order to design *higher order* schemes, a reconstruction operator in space and time is needed in such a way to recover from the cell average of T_i^n and of its neighbors $\mathcal{V}_i^n$ more accurate values at the interface. Since linear monotone schemes are at most of order one, as stated by the Godunov theorem [86], a first contribution for the improvement of the order of accuracy has been provided by the class of second order accurate TVD schemes, which adopts a linear reconstruction in space and time, like the MUSCL scheme of van Leer [156] and has to be used together with a *limiter* such as the minmod function in one dimension, or the Barth and Jespersen limiter [11] in two dimensions. If more than second order of accuracy is required one can adopts the nonlinear ENO reconstructions on unstructured grids introduced in [1, 145] as well as WENO reconstructions [82, 94, 143], which provide a nonlinear high order accurate and non-oscillatory spatial reconstruction. Then a suitable time stepping technique has to be used to guarantee the final order of accuracy. The Runge-Kutta methods, that perform a multi-stage time-integration, represent the most popular method to evolve the numerical solution from the current time level t^n to the next time level t^{n+1}. But, in recent years a valid alternative was proposed by Toro et al., who developed the ADER approach [8, 35, 67, 72, 123, 148, 149], which is one-step fully discrete and of arbitrary order of accuracy in space and time.

All the literature cited above provides a useful introduction to classical Finite Volume schemes in particular in the case of balance laws, i.e. $\mathbf{B} = \mathbf{0}$, and within an Eulerian framework, where the flow is observed in a fixed reference system which does neither change nor move in time. Although this provides effective tools it is not always enough: for example non-conservative terms need a specific treatment just to guarantee that a weak discontinuous solution is well-defined, where instead the errors due to nonlinear convective terms need a different approach to be considerably reduced. For the former case one can adopt a path-conservative approach (see Section 1.3) and for the latter it is more convenient to study the problem from a Lagrangian point of view (see Section 1.2).

1.2 Lagrangian-type methods on moving meshes

Lagrangian-type Finite Volume schemes are characterized by a moving computational mesh: at each time step the new position of all the nodes is recomputed according to a prescribed mesh velocity, which generally is chosen as close as possible to the local fluid velocity. In the Lagrangian description of the fluid the nonlinear convective terms disappear and Lagrangian schemes exhibit virtually no numerical dissipation at contact waves and material interfaces. So the aim of these methods is to reduce the numerical dissipation errors due to the convective terms, hence to capture contact discontinuities sharply and to precisely identify and track material interfaces. Among the different variants, that will be briefly recalled here, I would like to underline the role of the *direct* Arbitrary-Lagrangian-Eulerian (ALE) schemes. In this case the mesh velocity can be chosen in an arbitrary way, so usually it is chosen close to the fluid velocity, but the fact that it can be slightly modified allows first, some flexibility in the mesh motion reducing the mesh distortion, second let us perform the rezoning before the computation of the numerical flux, so that the remapping stage is no more needed, and finally it naturally extends to complex unstructured meshes and to slide lines treatment.

Lagrangian schemes were already of interest in the 1950 when John von Neumann and Richtmyer were working on the one-dimensional case [160], then Wilkins proposed a two-dimensional extension in 1964, see [161]. For a general review one can refer to [13]. A first natural approach, since the fluid velocity is required at each node and at each time step, is a *staggered* discretization, where the momentum is defined at the grid vertexes and all the other flow variables are defined at the cell center. Despite some drawbacks of the initial version of staggered Lagrangian schemes, which was not conservative and which produced some spurious modes in the numerical solution, it was widely used in the last forty years and many improvements have been made in the meantime; for further details one can refer to the papers of Caramana and Shashkov [32, 33] and of Loubère at al. [111,112]. Moreover, examples on general polygonal grids have been presented in [113]. An alternative consists in a *conservative cell-centered* discretization, which was first introduced by Godunov in [86]. An early application of conservative cell-centered Godunov-type schemes to the compressible Euler equations of gas dynamics in a Lagrangian framework on moving grids was provided by Munz in [124], using Roe-type and HLL-type approximate Riemann solvers. In many recent papers, see for example Després et al. [34,57,58] and Maire et al. [114–117], the conservative cell-centered Godunov-type approach is used both with structured and unstructured mov-

ing grids and in two and three space dimensions, respectively. Successively also better than second-order accurate schemes were introduced: *high order* of accuracy in *space* was first achieved by Cheng and Shu [49, 109] by means of a nonlinear ENO reconstruction, and high order in *time* was obtained either by the use of Runge-Kutta type methods or by adapting the ADER-WENO schemes to the Lagrangian framework, see for example Dumbser et al. [77] and Cheng and Toro [50]. Recent work on high order Lagrangian discontinuous Galerkin finite element methods can be found in the papers of Vilar et al. [157–159], Yu et al. [108] and Boscheri and Dumbser [24], while high order Lagrangian continuous finite elements have been studied by Scovazzi et al. [126, 142] and Dobrev and Rieben et al. [61–63].

For all the cell-centered methods an important step is the computation of the fluid velocity at the nodes, since this information is not directly available in the scheme, but it has to be extrapolated from the adjacent cells. To obtain these values three different types of *node solvers* can be employed. The simplest one is that proposed in the above mentioned papers by Cheng and Shu [49, 109], somehow employed also in this work, where the node velocity is obtained as arithmetic average among the near states; another possibility, suggested by Després et al. (GLACE scheme) [34] and Maire (EUCCLHYD scheme) [114], is to solve multiple one-dimensional half-Riemann problems around a vertex, in order to get an approximate solution of the multi-dimensional (generalized) Riemann problem at the node; the most recent method introduced by Balsara et al. [5–7, 9] consists in solving approximately a multidimensional Riemann problem at the nodes, using a new family of genuinely multidimensional HLL-type Riemann solvers. They are all compared with each other within a high-order ADER-WENO ALE scheme in the recent paper of Boscheri et al. [26].

Although all these different schemes are widely used, especially to describe compressible multi-material flows, a common problem that affects all Lagrangian methods is the severe mesh *distortion* or the mesh tangling that happens in the presence of shear flows and that may even destroy the computation. Hence, all Lagrangian methods must be in general combined with an algorithm to (locally) *rezone* the mesh at least from time to time and to remap the solution from the old mesh to the new mesh in a conservative manner. Lagrangian remesh and remap ALE schemes are very popular and some recent work on that topic can be found in [16, 19, 30, 103, 162]. Extensions of the remesh-remap approach to better than second-order of accuracy can be found in [18, 110]. In contrast to indirect ALE schemes (purely Lagrangian phase, remesh and subsequent remap phase) there are the so-called direct ALE schemes, where the local rezoning is performed before the computation of the numerical fluxes, hence changing di-

rectly the chosen mesh velocity of the ALE approach, see for example [21–23] for recent work in that direction based on high order ADER-WENO schemes. Moreover, when dealing with shear flows at material interfaces in realistic applications, see e.g. [101], a special treatment of *slide lines* may be required. The introduction of slide lines goes back to an idea of Wilkins [161], successively studied and refined by Caramana [31], Barlow et al. [10] and Loubère at al. [102]; the main ideas adopted in their papers regard the subdivision of the nodes at the interface in master and slave nodes and the study of the forces between the two sides of the slide lines. Another very interesting approach to slide lines was presented by Clair et al. in [51, 52] and by Del Pino et al. in [17, 134]. In [32] a staggered Lagrangian code has been presented, where the internal interfaces are handled with a special type of boundary condition. A very original solution to the problem of shear flows in Lagrangian simulations has been recently proposed by Springel in [146], where the connectivity of the moving mesh is dynamically regenerated via a moving unstructured but conforming Voronoi tessellation of the domain.

At this point we also would like to refer to some recent works on high order Eulerian and ALE schemes on moving meshes with time-accurate *local time stepping* (LTS) presented in [27,47,66], where each element is allowed to run at its own optimal local time step according to a *local* CFL stability condition. The resulting algorithms use a *conforming* grid in space, but naturally produce a *nonconforming* mesh in time. Therefore, the new nonconforming ALE method presented in this thesis, which produces a nonconforming mesh in both space and time, is naturally related to some of the ideas forwarded in [27, 66] in the context of local time stepping.

1.3 Well balanced path-conservative methods

Coming back to the complete system (1.1) with $\mathbf{B} \neq \mathbf{0}$ the main difficulty, both from the theoretical and the numerical point of view, comes from the presence of non-conservative products that do not make sense in the standard framework of distributions when the solution $\mathbf{Q}$ develops discontinuities. From the theoretical point of view, we assume the definition of non-conservative products as Borel measures given by Dal Maso, Le Floch and Murat in [55]. This definition allows to generalize the classical Rankine-Hugoniot conditions, which relate the jump of the conserved quantities from a left state $\mathbf{Q}_L$ to a right state $\mathbf{Q}_R$ across a discontinuity with propagation speed σ, and are fundamental to give a rigorous definition of the weak solution. The generalized Rankine-

Hugoniot conditions read

$$\sigma\left(\mathbf{Q}_R-\mathbf{Q}_L\right)-\int_0^1 \mathbf{A}(\Phi(s,\mathbf{Q}_L,\mathbf{Q}_R))\frac{\partial\Phi}{\partial s}\,ds=\mathbf{0} \tag{1.8}$$

so it is clear that they depend on the choice of a family of paths in the phases-space. Different integration paths result in different jump relations. Note that in the case when $\mathbf{A}(\mathbf{Q})$ is the Jacobian of a flux $\mathbf{F}$, i.e. $\mathbf{B}=\mathbf{0}$, the generalized Rankine-Hugoniot conditions reduce to the classical ones, independent of the particular choice of the path.

Starting from this theory, the family of so-called *path-conservative* methods has been developed: it is based on a prescribed family of paths and provides a generalization of standard conservative schemes. The first special case of a path-conservative scheme has been proposed by Toumi in [153], where he used a weak formulation of the method of Roe based on the integration along a particular path. The framework of path-conservative schemes has then been successively extended and analyzed by Parés [128] and Castro et al. [38, 43] in the Finite Volume context and by Rhebergen et al. [137] in the discontinuous Galerkin Finite Element context. The first better than second order accurate path-conservative schemes on general unstructured meshes in two and three space dimensions have been published in [71, 73, 76].

Moreover, by choosing a particular path the method can be made *well balanced* that means able to maintain up to machine precision the stationary solutions of the considered system. The design of numerical methods with such a good property is a very active front of research: see, for instance [3, 15, 29, 39, 45, 87–90, 106, 130–132, 135, 136, 147, 151], and the recent overview on path-conservative well-balanced schemes [36]. It has to be emphasized that conventional numerical schemes are in general *not* able to preserve such stationary solutions, especially on coarse meshes, although the source term is discretized in a consistent manner, but consistency alone is not enough to achieve good results on coarse grids. This leads to erroneous numerical solutions especially when trying to compute small perturbations around the steady states necessitating the need for very fine meshes. Many recent papers have been devoted to this topic, in particular we refer to [14, 28, 48, 59, 60, 98, 99, 141] and the references therein, for a complete state of the art.

1.4 Fluid solvers for vortical flows

One of the aims of this thesis is to contribute to a long lasting debate in computational fluid dynamics which is the enhancement of the accuracy of

compressible fluid solvers for vortical flows. ALE and well balancing represent one of the techniques that we have successfully adopted, based on ameliorating the employed numerical method. However, another strategy consists in playing directly with the equations formulation. In particular for what concerns the Euler equations, in the literature one can distinguish two main approaches: the first one based on *vorticity* and the second one based on *angular momentum*. For a general review on vorticity in standard Finite Volume schemes we refer to Roe [139] and for the case of Lagrangian methods we mention the seminal work of Dukowicz and Meltz in [64]. Other references are in the recent contribution [133]. Concerning angular momentum, we cite the recent results of Després and Labourasse [56] where the angular momentum is added to the system (with an initial approach similar to the one we propose in Chapter 6) and treated with a partial Discontinous Galerkin discretization. In particular, they show that considering the angular momentum enhances the accuracy of implosion calculations and minimizes the mesh imprint.

There exists a wide range of applications for which the *conservation of angular momentum* is an issue. A first example is fluid simulations of the atmosphere around the earth for which the angular momentum of the atmosphere may interact with the angular momentum of the planet itself, we refer to [125] for an early work on the theme. A completely different physical problem is rotation of MHD flows in Tokamaks for which angular preservation is clearly a fundamental issue. It is addressed in the context of MHD solvers, either full MHD or reduced MHD, and a general review can be found in [96]. We notice that Finite Volume techniques are rarely used in the Tokamaks community. On the other hand Godunov solvers are widely used for astrophysical flows, and angular momentum is a key feature for an accurate numerical treatment of rotation of stars and planets: many works are devoted to this issue on Cartesian fixed grids and we quote only on few of them such as [121, 144]. In this context, Käppeli and Mishra have recently proposed a Godunov scheme in the Eulerian frame to address the issue of angular momentum conservation [97]. A last case regards the chemical reactions into the combustion chamber of engines [2]: in this situation the initial stage of turbulent flows is dominated by strong vortexes inside the flow and so its study could be improved taking into account the angular momentum.

1.5 PhD thesis summary

The main contributions of this PhD thesis are briefly summarized in what follows.

Direct ALE schemes on moving nonconforming unstructured meshes. In Chapter 2, we present a novel second-order accurate Arbitrary-Lagrangian-Eulerian Finite Volume scheme which proposes a new and effective technique to move the nodes at the interfaces between regions with different characteristics, in order to avoid the typical mesh distortion of Lagrangian-type methods.

In particular, in our new approach the interfaces could be both prescribed a priori by the user or automatically detected by the algorithm, if the tangential velocity difference across an element interface is sufficiently large. The new scheme then subdivides the neighbors of an interface edge into two groups, each of which has similar properties, namely the *left* and the *right* neighbors. The nodes far away from the shear discontinuity are moved with a standard node solver, while for those at the interface a *new node* is inserted in such a way that the old one moves according to an averaged velocity over the left neighbors, and the new one slides along the edge according to the average velocity prescribed by the right group.

We refer to the resulting mesh as *nonconforming* because each edge can be shared between *more* than two elements and a node can lie on an edge not only at its extremities, i.e. we explicitly allow so-called *hanging nodes*. This gives us more flexibility in the grid motion and helps to maintain a high quality mesh: indeed, the elements on both sides of the shear wave can move with a different velocity, without producing highly distorted elements. The core of the proposed method is the *local* update of all the necessary connectivity tables, as a consequence of the insertion or the deletion of nodes and edges, and the computation of the numerical fluxes between the space-time control volumes, taking in particular into account that the lateral faces can be shared between more than two elements.

The final ALE finite volume scheme is based directly on a space-*time* conservation formulation of the governing PDE system integrated over a set of closed space-*time* control volume C_i^n

$$|T_i^{n+1}|\,\mathbf{Q}_i^{n+1} = |T_i^n|\,\mathbf{Q}_i^n - \int_{\partial C_i^n} \tilde{\mathbf{F}}_{ij}\cdot\tilde{\mathbf{n}}_{ij} + \int_{C_i^n} \mathbf{S}\,,$$

where the discontinuities at the interfaces are resolved through an ALE numerical flux function $\tilde{\mathbf{F}}_{ij}\cdot\tilde{\mathbf{n}}_{ij}$ of Rusanov, Osher or HLL -type. This strategy

completely avoids the need of an additional remapping stage, hence the new method is a so-called *direct* ALE scheme, and automatically satisfies the GCL condition. Second order of accuracy in space and time is obtained by using a MUSCL-Hancock strategy, together with a Barth and Jespersen slope limiter and a time-evolution of the reconstruction polynomials.

A large set of different numerical tests has been carried out in order to check the accuracy and the robustness of the new method for both smooth and discontinuous problems. In particular we have compared the results for a steady vortex in equilibrium solved with a standard *conforming* ALE method (without any rezoning technique) and with our new *nonconforming* ALE scheme, to show that the new nonconforming scheme is able to avoid mesh distortion even after very long simulation times, refer to Figure 2.14.

This work is the object of the publication [84], written in collaboration with Michael Dumbser and Manuel J. Castro, published on Computer & Fluids.

Well balanced path-conservative methods. Chapters 3, 4 and 5 focus on well balanced path-conservative schemes, i.e. methods appropriate to treat non-conservative products and able to maintain up to machine precision families of equilibria of the studied system. These methods are based on the choice of a *path* when integrating from a left state to a right one across a discontinuity. In particular we propose a path Φ which directly exploits the known stationary solution (and so it is exact on it), and treats in a approximate way only the fluctuations around the equilibrium

$$\Phi = \Phi^E + \Phi^f.$$

The same idea of using an approximate technique only on the fluctuations appears also in the reconstruction process

$$\mathbf{q}_i^n = \mathbf{Q}_i^E + \mathcal{P}_i^f,$$

where $\mathcal{P}_i^f$ is a standard reconstruction operator applied only to fluctuations. This concept was already introduced in [128], [129], [38] and [40]. Based on it, we have added the following contributions.

We have formulated a new HLL-type and a novel Osher-type numerical flux. For what concerns the HLL-type well balanced flux we have followed the strategy called *identity modification* introduced in [40] and [46], modifying the standard HLL viscosity by substituting the appearing identity matrix with a diagonal matrix that vanishes at the equilibrium. It is the first time that this

strategy is applied to the Euler equations with gravity in polar coordinates and is inserted in an ALE moving mesh framework.

For what concerns the Osher-type well balanced flux instead, it is already an innovation from the theoretical point of view in one dimension. To the knowledge of the authors, it is the first time that the little dissipative path-conservative Osher scheme proposed by [75] is modified in order to be well balanced for non trivial equilibria. In particular, starting from its standard viscosity, which depends on the *absolute value* of the Jacobian matrix multiplied by the jump of the conserved variables at the interface, we have reformulated it in order to exploit directly the well balanced technique used in the non-dissipative part. The final formulation of our Osher-Romberg viscosity depends on the *sign* of the Jacobian matrix and it is directly multiplied by the discretization of flux and non-conservative terms, computed in strategic points given by the Romberg quadrature rule (instead of the Gauss rule usually employed). This strategy is in our opinion of particular interest because it makes quite automatical the formulation of a well balanced viscosity.

Moreover, the well balanced reconstruction is performed in the ADER context i.e. by evolving the conserved variables directly in space and *time* (as already done once in [70]), not using as usual in the well balanced community, the classical features of Runge-Kutta steps.

The ideas presented in Chapter 3 represent the theoretical foundation of the paper [83], written in collaboration with Manuel J. Castro and Michael Dumbser, published on Monthly Notices of the Royal Astronomical Society.

WB ALE for the Euler equations with gravity. Chapter 4 is especially dedicated to a very challenging physical problem in computational astrophysics that is the simulation of rotating gas clouds for very long computational times and with high accuracy. The physical situation we want to study is described by the Euler equations of compressible gas dynamics with an externally given gravitational field generated by a central object. A very important family of stationary solutions of the governing equations is characterized by the *equilibrium* between pressure gradient, centrifugal force and gravity force. In order to model with high accuracy and minimal dissipation small perturbations around those equilibria for a very long period of time we propose to use the well balanced technique introduced above coupled with our nonconforming ALE framework. In particular, we work on a non-conservative version of the Euler equations with gravity in polar coordinates where the source terms are rewritten via non-conservative products.

The use of the WB ALE combination solves the main problems connected with this type of simulation. First, small physical perturbations are not hidden by spurious numerical oscillations because the employed method is exact up to machine precision on the equilibrium part even at discrete level. Second, the Lagrangian context reduces the errors due to convected terms, and finally, our nonconforming technique allows to preserve a high mesh quality despite the shear flow due to the differential rotation. From the numerical results presented in Sections 4.2.6 - 4.2.9 one can appreciate that is really the coupling between these two techniques that allows to obtain so good results.

We furthermore have provided a thorough comparison of our new numerical method with the results that can be obtained with the PLUTO code [120] [122], a well established freely-distributed software for the numerical solution of mixed hyperbolic/parabolic systems of partial differential equations targeting high Mach number flows in astrophysical fluid dynamics. This code is based on finite volumes and therefore is rather close to the scheme proposed in this thesis: the provided comparison shows the great enhancement introduced by our novel techniques.

The results presented in Chapters 4 are also part of the publication [83].

WB ALE for the shallow water equations. The same coupled approach is also applied to shallow water equations in polar coordinates whose source terms are rewritten via non-conservative products. The equilibria between centrifugal and gravitational forces are maintained up to machine precision. The immediacy with which the WB ALE combination can be applied to different contexts proves the general applicability of the introduced method, which is not limited to one set of equations or one family of equilibria.

The obtained results are part of the publication [84].

Well balancing for a simple two-phase model with GPU parallel implementation. In Chapter 5 we propose a simple and efficient two-phase interface-capturing algorithm for the simulation of complex non-hydrostatic free surface flows. The physical model is given by a special case of the more general Baer-Nunziato system for compressible multi-phase flows: in particular it does not neglect accelerations in gravity direction, as it is in the classical shallow water system; moreover, the free surface is not necessary given by a single value function so even complex shapes as those of breaking waves can be captured. The model was first introduced by Dumbser in [65] and here we solve it in a well balanced manner and through a GPU parallel implementation.

The novel well balanced Osher-Romberg scheme assures lower numerical dissipation at the free surface and the preservation of equilibria between air and water even in the presence of obstacles. Furthermore, the parallel implementation allows the use of very fine meshes. In particular our parallel algorithm is composed by three main kernels: the first one for the reconstruction procedure, the second one to compute the numerical fluxes at the interfaces, and the third one to evolve the cell averages from the current time step to the next one. Each kernel automatically subdivides the work between the cores of the GPU multiprocessors, thanks to the NVIDIA CUDA framework. As in any parallel implementation, we pay particular attention to not modify the same data at the same moment. For this reason, the second kernel is additionally separated into four non intersecting kernels each one working on a subset of the edges with non intersecting neighbors. The code is further accelerated by subdividing the data in $2D$ grids, so that neighbor cells or edges are treated by the same multiprocessors reducing slow memory accesses.

The results presented in Chapter 5 have been obtained collaborating with Manuel J. Castro at the University of Malaga (Spain). For this research period abroad, I was funded through a S-ESR Marie Sklodowska-Curie ITN fellowship within the project ModCompShock (European Union's Horizon 2020, grant agreement no. 642768).

Angular momentum preserving schemes for compressible Euler equations. In Chapter 6 we investigate the conservation of the *angular momentum* for the Euler equations of compressible gas dynamics. We propose a method able to discretize, besides the standard quantities, *i.e.* mass, inertial momentum and energy, the angular momentum w, and we study the positive effects of considering this adjoint physical variable on the entire system. With respect to the previous chapters, here we do not work on ameliorating the numerical scheme (we use basically the one of Chapter 2) but on introducing a new formulation of the governing equations.

So, first we deduce from the linear momentum equation the conservation law regarding the angular momentum

$$\partial_t (\rho w) + \nabla \cdot \left(\rho \mathbf{u} w + p(\mathbf{x} - \mathbf{x_c})^{\perp} \right) = 0,$$

which is redundant from the analytical point of view but extremely interesting for numerical purposes, especially for *vortical flows*. Indeed, using a standard finite volume scheme to solve steady rotational problems the velocity rapidly

dissipates. So, for long times, even using an high order scheme and a very fine mesh $||\mathbf{u}||_{L_2} \to 0$ and consequently $w = \mathbf{u} \wedge (\mathbf{x} - \mathbf{x_c}) \to 0$ instead of maintaining its constant value. While, we have seen that simply adding the angular momentum equation as slave equation in the system conspicuously reduces the dissipations, if computing *a posteriori* $||\mathbf{u}||_{L_2}$.

Thus, we propose also some coupled approaches, namely a global coupling and a local coupling, in order to exploit the benefit provided by considering the conservation law regarding the angular momentum directly *during* the computation coupling it with the velocity. The coupling is obtained by noticing that, chosen a center of rotation $\mathbf{x}_c$ and an orthonormal basis $(\mathbf{e}_r, \mathbf{e}_\theta)$, the velocity can be rewritten as the sum of the two components along this basis: we call $\mathbf{v}$ the velocity when written in this way

$$\mathbf{v} = \mathbf{v}_r + \mathbf{v}_\theta, \quad \mathbf{v}_r = \frac{1}{r^2} \langle \mathbf{u}, (\mathbf{x} - \mathbf{x}_c) \rangle (\mathbf{x} - \mathbf{x}_c), \quad \mathbf{v}_\theta = -\frac{1}{r^2} w (\mathbf{x} - \mathbf{x}_c)^\perp,$$

In particular if $\mathbf{u}$ is a radial field and $\mathbf{e}_r$ lies along the radial direction then $\mathbf{v}_r$ is null, which is easy to maintain even at numerical level, since classically $||\mathbf{u}||_{L_2}$ rapidly dissipates. Hence, the correctness of $\mathbf{v}$ is strictly connected with w, and one can suppose that if w is computed in a good way this can produce even positive effects on $||\mathbf{u}||_{L_2}$. For this reason we suggest the following global coupled formulation

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0, \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{v}) + \nabla p = \mathbf{0}, \\ \partial_t (\rho e) + \nabla \cdot (\rho \mathbf{v} e + p \mathbf{v}) = 0, \\ \partial_t (\rho w) + \nabla \cdot (\rho \mathbf{v} w) + \nabla \wedge (p(\mathbf{x} - \mathbf{x}_c)) = 0, \end{cases}$$

obtaining some encouraging results. However, simple test cases open important issues in terms of imposition of appropriate boundary conditions and proper definition of the angular momentum, in particular when the center of rotation is not known *a priori*. For this last reason, we introduce a *detector* able to reconstruct locally the centers of rotation or of explosion of a generic problem given its velocity field and its pressure field. This detector provides supplementary information over the studied system and can also be applied for a convenient definition of the angular momentum. In particular it is used within the local coupling approach.

Finally, we consider the Kidder test case, emblematic of strong implosion in stars or for inertial confinement devices, and we show how to add a rotation to the initial condition.

The results presented in Chapter 6 have been obtained collaborating with Bruno Després and Stéphane Del Pino at the Laboratoire Jacques-Louis Lions - UPMC, Paris (France) and are the object of a submitted publication. For this research period abroad, I was funded through the Erasmus+ Programme of the European Union.

2 Direct ALE Finite Volume schemes on moving nonconforming meshes

Direct Arbitrary-Lagrangian-Eulerian (ALE) Finite Volume schemes are characterized by a moving computational mesh: at each time step the new position of all the nodes is recomputed according to a prescribed mesh velocity, and the solution is evolved in time directly through a space-time conservation formulation of the governing PDE system, without needing of further remapping stages.

In this chapter we provide a detailed description of our nonconforming ALE scheme for conservative systems of hyperbolic equations; the non-conservative case will be treated in the central part of the thesis, see Chapter 4. After having introduced in Section 2.1 the equations and the complete notation for a very flexible and general discretization of the two dimensional domain, we prepare all the necessary geometric information about the so called *control volumes* in Section 2.1.1. Then, in Section 2.1.2 we present the space–time reconstruction operator that provides second order of accuracy to our direct ALE scheme, which is described in Section 2.1.3. In Section 2.2 we explain how to deal with the moving nonconforming (hanging) nodes and the corresponding *local* update of the connectivity tables.

In Section 2.4 some numerical test problems are presented in order to check the efficiency and the robustness of the proposed approach in maintaining a high quality mesh, local and global volume conservation, and in satisfying the GCL condition. The numerical results presented in this chapter concern the shallow water equations both in Cartesian coordinates (see Section 2.3.1), and in polar coordinates (see Section 2.3.2). Other interesting applications will be shown in Chapter 4 where the nonconforming ALE framework will be coupled together with a well balancing technique both for studying the shallow water equations and the Euler equations with gravity.

The algorithm presented here is not necessarily limited to logically straight slipe lines. In Section 2.5 we therefore show first preliminary results for general, logically non-straight slide lines.

2.1 ALE framework on moving nonconforming meshes

We consider here two-dimensional nonlinear hyperbolic systems of conservation laws, i.e. a subset of (1.1) that can be cast in the following general form

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{Q}) = \mathbf{S}(\mathbf{Q}), \quad \mathbf{x} \in \Omega(t) \subset \mathbb{R}^2, \mathbf{Q} \in \Omega_{\mathbf{Q}} \subset \mathbb{R}^{\nu}, \tag{2.1}$$

where $\mathbf{x} = (x, y)$ is the spatial position vector, t represents the time, $\mathbf{Q} = (q_1, q_2, \ldots, q_\nu)$ is the vector of conserved variables defined in the space of the admissible states $\Omega_{\mathbf{Q}} \subset \mathbb{R}^{\nu}$, $\mathbf{F}(\mathbf{Q}) = (\,\mathbf{f}(\mathbf{Q}), \mathbf{g}(\mathbf{Q})\,)$ is the nonlinear flux tensor, and $\mathbf{S}(\mathbf{Q})$ represents a nonlinear algebraic source term.

To discretize the moving domain, we consider a total number N_E of polygonal elements T_i^n of area $|T_i^n|$, each one with an arbitrary number of vertexes $N\mathsf{v}(i)$, $i = 1, \ldots, N_E$: the union of all these elements results in an unstructured mesh $\mathcal{T}_\Omega^n$ which covers the computational domain $\Omega(\mathbf{x}, t^n) = \Omega^n$ at the current time t^n and which contains a total number N_{node}^n of nodes and a total number N_{edge}^n of edges. Note that the number of elements, nodes and edges, as well as the shape of each element are allowed to change at each time step. Moreover, we refer to our mesh as *nonconforming* because each edge can be shared between more than two elements, and therefore a node can lie *on* an edge not only at its extremities, i.e. we explicitly allow so-called *hanging nodes*. Refer to Figure 2.1 for an example.

In general, with index i we refer to an element T_i^n, with index j we refer to one of its neighbors T_j^n (which shares with T_i^n an edge or a vertex depending on the situation), and with index k we refer to a node. The coordinate of each node at time t^n are denoted by $\mathbf{x}_k^n$, and $\overline{\mathbf{V}}_k^n$ represents the velocity at which it is supposed to move, so that its new coordinates at time t^{n+1} are given from the following relation

$$\mathbf{x}_k^{n+1} = \mathbf{x}_k^n + \Delta t \overline{\mathbf{V}}_k^n, \tag{2.2}$$

More details on how to obtain $\overline{\mathbf{V}}_k^n$ and $\mathbf{x}_k^{n+1}$ will be given in Section 2.2.3.

2.1.1 Nonconforming space–time control volumes

As introduced in Section 1.1, Finite Volume schemes are based on integrating the governing equation in space over a spatial control volume and then in time, see (1.3). In the case of our *direct* ALE scheme the integration is performed directly over the *space–time control volumes*, which are constructed as follow.

For each element T_i^n the new vertex coordinates x_k^{n+1}, $k = 1, \ldots, N\mathsf{v}(i)$, are connected to the old coordinates $\mathbf{x}_k^{n+1}$ via straight line segments, yielding a

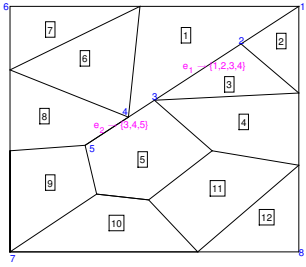


Figure 2.1: Example of a nonconforming mesh that can be treated by our algorithm. The mesh contains $N_E = 12$ elements: triangles, quadrilaterals and five-sided polygons. The mesh is nonconforming: note for example edge e_1. It is shared between the elements T_1, T_2, T_3, T_5 and on it we can find two intermediate nodes $2, 3$ called hanging nodes. A similar situation can be noted for edge e_2.

multidimensional space-time control volume $C_i^n = T_i^n(t) \times [t^n, t^{n+1}]$, that involves overall $N\mathsf{v}(i) + 2$ space-time sub-surfaces. Specifically, the space-time volume C_i^n is bounded on the bottom and on the top by the element configuration at the current time level T_i^n and at the new time level T_i^{n+1}, respectively, while it is closed with a total number of $N\mathsf{v}(i)$ lateral space–time surfaces $\partial C_{ij}^n = \partial T_{ij}^n(t) \times [t^n; t^{n+1}]$ that are given by the evolution of each edge ∂T_{ij}^n of element T_i^n within the time step $\Delta_t = t^{n+1} - t^n$. Therefore the space-time volume C_i^n is bounded by its surface ∂C_i^n which is given by

$$\partial C_i^n = \left(\bigcup_j \partial C_{ij}^n \right) \cup T_i^n \cup T_i^{n+1}. \tag{2.3}$$

For a graphical interpretation one can refer to Figure 2.2, where we have reported an example of a control volume and of the parametrization of the lateral space–time surfaces.

Now, for each control volume we can compute its barycenter and for each sub–surfaces their areas, normal vectors, and space–time midpoints. The upper space–time sub–surface T_i^{n+1} and the lower space–time sub–surface T_i^n are the simplest, since they are orthogonal to the time coordinate. The space–time unit normal vectors are respectively

$$\tilde{\mathbf{n}} = (0, 0, 1) \quad \text{and} \quad \tilde{\mathbf{n}} = (0, 0, -1). \tag{2.4}$$

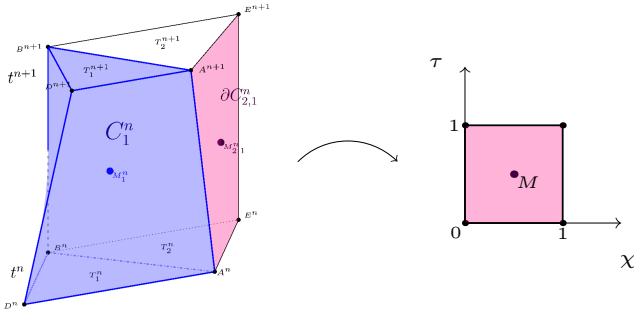


Figure 2.2: Left. In blue we show the physical space–time control volume C_1^n obtained by connecting via straight line segments each vertex of T_1^n with the corresponding vertex of T_1^{n+1}, and its space-time midpoint M_1^n. In pink we show one of the lateral surfaces of C_2^n, $\partial C_{2,1}^n$, together with its space–time midpoint $M_{2,1}^n$. Right. The reference system (χ, τ) adopted for the bilinear parametrization of the lateral surfaces ∂C_{ij}^n.

To compute their areas we can use the *shoelace formula* or *Gauss's area formula* which is valid for any type of polygonal element

$$|\, T_i^n \,| = \frac{1}{2} \left| x_{N\mathsf{v}(i)}^n y_1^n - x_1^n y_{N\mathsf{v}(i)}^n + \sum_{j=1}^{N\mathsf{v}(i)-1} \left(x_j^n y_{j+1}^n - x_{j+1}^n y_j^n \right) \right|, \tag{2.5}$$

where $\mathbf{x}_j^n = (x_j^n, y_j^n)$, $j = 1, \ldots, N\mathsf{v}(i)$, are the coordinates of the vertexes of element T_i^n numbered in a counterclockwise order. Next, the lateral space–time surfaces of C_i^n are parametrized using a set of bilinear basis functions

$$\partial C_{ij}^n = \tilde{\mathbf{x}}\,(\chi, \tau) = \sum_{k=1}^{4} \beta_k(\chi, \tau)\, \tilde{\mathbf{x}}_{ij,k}^n, \qquad 0 \le \chi \le 1, \quad 0 \le \tau \le 1, \tag{2.6}$$

where the $\tilde{\mathbf{x}}_{ij,k}^n$ represent the physical space–time coordinates of the four vertexes of ∂C_{ij}^n (two of them belonging to T_i^n and the other two being their evolution in time and belonging to T_i^{n+1}), and the $\beta_k(\chi, \tau)$ functions are defined as follows

$$\begin{aligned} \beta_1(\chi, \tau) &= (1-\chi)(1-\tau), & \beta_2(\chi, \tau) &= \chi(1-\tau), \\ \beta_3(\chi, \tau) &= \chi\tau, & \beta_4(\chi, \tau) &= (1-\chi)\tau. \end{aligned} \tag{2.7}$$

The mapping in time is given by the transformation

$$t = t^n + \tau\,\Delta t, \qquad \tau = \frac{t - t^n}{\Delta t}, \tag{2.8}$$

hence the Jacobian matrix $J_{\partial C_{ij}^n}$ of the parametrization is

$$J_{\partial C_{ij}^n} = \begin{pmatrix} \vec{e}_x & \vec{e}_y & \vec{e}_t \\ \frac{\partial x}{\partial \chi} & \frac{\partial y}{\partial \chi} & \frac{\partial t}{\partial \chi} \\ \frac{\partial x}{\partial \tau} & \frac{\partial y}{\partial \tau} & \frac{\partial t}{\partial \tau} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{e}} \\ \frac{\partial \tilde{\mathbf{x}}}{\partial \chi} \\ \frac{\partial \tilde{\mathbf{x}}}{\partial \tau} \end{pmatrix}. \tag{2.9}$$

The space–time unit normal vector $\tilde{\mathbf{n}}_{ij}$ can be evaluated computing the normalized cross product between the transformation vectors of the mapping (2.6), i.e.

$$|\partial C_{ij}^n| = \left| \frac{\partial \tilde{\mathbf{x}}}{\partial \chi} \times \frac{\partial \tilde{\mathbf{x}}}{\partial \tau} \right|, \quad \tilde{\mathbf{n}}_{ij} = \left(\frac{\partial \tilde{\mathbf{x}}}{\partial \chi} \times \frac{\partial \tilde{\mathbf{x}}}{\partial \tau} \right) / |\partial C_{ij}^n|, \tag{2.10}$$

where $|\partial C_{ij}^n|$ is the determinant of the Jacobian matrix $J_{\partial C_{ij}^n}$ and represents also the area of the lateral surfaces. Moreover, exploiting the parametrization in (2.6)-(2.8) and choosing $\chi = 0.5$ and $\tau = 0.5$ we recover the coordinates $M_{i,j}^n$ of the space–time midpoint of the lateral surfaces. Moreover, the space–time barycenter M_i^n of each control volume C_i^n reads

$$M_i^n = \left(\frac{\mathbf{x}_i^n + \mathbf{x}_i^{n+1}}{2}, \frac{t^n + t^{n+1}}{2} \right),$$

where the spatial barycenter $\mathbf{x}_i^n = (x_i^n, y_i^n)$ of T_i^n is given by the explicit formula

$$\mathbf{x}_i^n = \frac{1}{6\,|T_i^n|} \sum_{j=1}^{N\mathsf{v}(i)} \left(\mathbf{x}_j^n + \mathbf{x}_{j+1}^n \right) \left(x_j^n y_{j+1}^n - x_{j+1}^n y_j^n \right), \tag{2.11}$$

with the convention that $j = N\mathsf{v}(i) + 1$ coincides with $j = 1$.

Finally, let us consider the nonconforming case, i.e. the case when on the same edge we can find more than the two extreme nodes that means that more then two control volumes share the same edge. In this case the surface can be subdivided in sub–surfaces. The treatment of the nonconforming lateral space–time surfaces basically requires only to repeat the computation of the necessary geometric information over each sub–surface (the same will hold for the flux computation, which will be simply split in several parts).

For example, consider the case of $\partial C_{i,j}^n$ with the four standard vertexes and two more hanging nodes on the edges orthogonal to the time coordinate (as

in the middle of Figure 2.3). Then the lateral surface is shared between three (and not two, as usual) control volumes. However it can be subdivided into two pieces, each one shared between only two control volumes, which are still trapezoidal, so each of them can be mapped into the reference element using the standard map in (2.6), just taking care to select in a correct way the vertexes of each piece. Hence areas, normal vectors, and space–time midpoints can be computed exactly as in the conforming case but on each part. Next, we have to analyze the two extreme situations that happens due to our dynamical insertion and deletion of nodes. On the left and on the right of Figure 2.3 we have reported these two limiting cases: first, at time t^{n+1} a new node has been inserted, which at the previous time t^n did not yet exist; or vice-versa, at time t^{n+1} a hanging node is merged together with one of the other vertexes and hence it disappears. In these cases the lateral surfaces can still be divided into two parts, and even if one of them is triangular it can still be treated as a degenerate quadrilateral face, so all the computations can be performed, once again, as above. The coordinates of a hanging node at the moment of its creation or destruction will be set equal to those of the vertex from which the hanging node was born, or those of the vertex to whom it was merged, respectively.

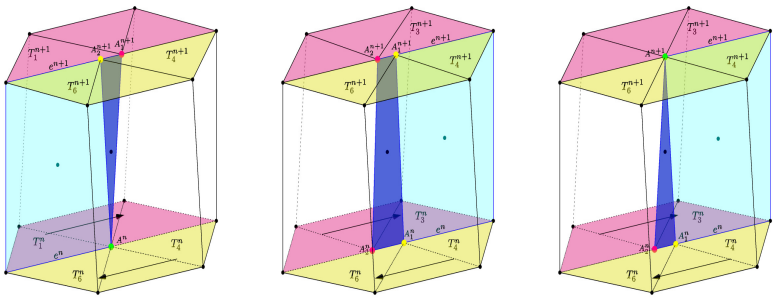
2.1.2 MUSCL type space–time reconstruction

As usual in a classical cell-centered Finite Volume scheme, at the beginning of each time step t^n we dispose of the cell averages $\mathbf{Q}_i^n$ of the conserved variables for each spatial control volume T_i^n, defined as

$$\mathbf{Q}_i^n = \frac{1}{|T_i^n|} \int_{T_i^n} \mathbf{Q}(\mathbf{x}, t^n)\, d\mathbf{x},$$

as introduced in (1.5). These are the data computed and stored at the previous time, and which will be used to evolve the solution during the current time step. To construct a method which is better than first order accurate we cannot compute the numerical fluxes directly with these piecewise constant data, but we have to reconstruct for each T_i^n a piecewise space-time polynomial $\mathbf{q}_h(\mathbf{x}, t^n)$, exploiting the cell averages of the cell and its neighbors, combined with a time-evolution procedure.

Here, second order of accuracy in space and time is achieved by using the MUSCL-Hancock strategy that was for the first time proposed by van Leer in [155] and which is very well explained in [150], slightly adapted to our context



(a) Insertion of a new node **(b)** Motion of hanging nodes **(c)** Fusion of two nodes

Figure 2.3: Suppose that at time t^n across the pink and the yellow elements the tangential fluid velocity changes sharply, as suggested by the arrows; at t^{n+1} the pink elements will move in one direction and the yellow ones will move in the opposite direction. In (a) at time t^n we have a conforming mesh, but in order to avoid a severe mesh distortion in the following time steps we decide to double the green node A^n. So at time t^{n+1} there are both A_1^{n+1} and A_2^{n+1}: A_1^{n+1} is a vertex for the pink elements and A_2^{n+1} is a vertex for the yellow elements. Moreover A_2^{n+1} is hung to edge e^{n+1}. So the blue lateral face of T_i^n, which has e^n and e^{n+1} as bases, is composed by two pieces: the one in light blue which is trapezoidal and touches elements T_1 and T_6, and the one in dark blue which is triangular and touches elements T_1 and T_4. Note in particular that we need to compute the flux between T_1 and T_4 during the interval $[t^n, t^{n+1}]$ even if at time t^n they were not in contact. In (b) we show the intermediate situation where a hanging node slides along an edge. In this case the blue surface is still divided into two parts and it is shared between three elements T_3, T_4 and T_6, so the computation of two fluxes will be required. In order to compute the fluxes and to maintain the second order of accuracy of the entire method the reconstruction polynomial $\mathbf{q}_h(\mathbf{x}, t)$ will be evaluated at the midpoints of each of the part of the lateral surface. Finally, in (c) we report the last limiting case: A_1^n and A_2^n are close and at t^{n+1} will be even closer since they are moving one towards the other, so we decide to merge them and to restore the conforming and simpler situation, in particular to avoid that A_1^n will leave edge e^{n+1} at time t^{n+1}. Eventually A^{n+1} could be doubled again at t^{n+2} if the tangential velocity difference across the interface is sufficiently large.

of a nonconforming moving mesh. For the *spatial* reconstruction, let us define a polynomial $\mathbf{w}_h(\mathbf{x}, t^n)$ of the form

$$\mathbf{w}_h(\mathbf{x}, t^n)\,|_{T_i^n} = \mathbf{w}_i(\mathbf{x}, t^n) = \mathbf{Q}_i^n + \nabla \mathbf{Q}_i(\mathbf{x} - \mathbf{x}_i), \qquad \mathbf{x} \in T_i^n,$$

where $\mathbf{x}_i$ is the barycenter of cell T_i^n. We denote by $\mathcal{S}_i^n$ the set of neighbors of T_i^n that share a common edge with T_i^n (the set $\mathcal{S}_i^n$ may change at each time step). To compute $\nabla \mathbf{Q}_i$, integral conservation is imposed on each element of $\mathcal{S}_i^n$

$$\frac{1}{|T_j^n|} \int_{T_j^n} \mathbf{w}_h(\mathbf{x}, t^n)\, d\mathbf{x} = \mathbf{Q}_j^n \qquad \forall T_j^n \in \mathcal{S}_i^n. \tag{2.12}$$

The above system is in general over-determined, so we add the constraint that equation (2.12) holds exactly at least for T_i^n. This is easily satisfied by rewriting the equations as

$$\frac{1}{|T_j^n|} \int_{T_j^n} \nabla \mathbf{Q}_i(\mathbf{x} - \mathbf{x}_i)\, d\mathbf{x} = \mathbf{Q}_j^n - \mathbf{Q}_i^n \qquad \forall T_j^n \in \mathcal{S}_i^n, \tag{2.13}$$

then we solve (2.13) via a classical least-squares approach using the normal equation of (2.13), and we thus obtain the *non-limited* slope $\nabla \mathbf{Q}_i$.

To ensure that new extrema are not created in the reconstruction process, we employ the classical *slope limiter* function Φ_i presented by Barth and Jespersen in [11]. The idea is to find the largest admissible Φ_i in such a way that

$$\tilde{\mathbf{w}}_h(\mathbf{x}, t^n) = \mathbf{Q}_i^n + \Phi_i \nabla \mathbf{Q}_i(\mathbf{x} - \mathbf{x}_i)$$

satisfies

$$\min_{j \in \mathcal{V}_i^n} \mathbf{Q}_j^n = \mathbf{Q}_i^{min} \leq \tilde{\mathbf{w}}_h(\mathbf{x}, t^n) \leq \mathbf{Q}_i^{max} = \max_{j \in \mathcal{V}_i^n} \mathbf{Q}_j^n,$$

where $\mathbf{Q}_i^{max}$ and $\mathbf{Q}_i^{min}$ are the componentwise maximum and minimum among the cell-averages of the set $\mathcal{V}_i^n$, respectively. The set $\mathcal{V}_i^n$ contains *all* the vertex neighbors of T_i^n and the element T_i^n itself. Since $\mathbf{w}_h$ is obtained as a piecewise linear reconstruction of the data, its extrema occur at the vertices of T_i^n. Hence, to compute the limiter for each conserved variable, it suffices to find for all vertices $\mathbf{x}_j$ of T_i^n

$$\Phi_{i,j} = \begin{cases} \min\left(1, \frac{\mathbf{Q}_i^{max} - \mathbf{Q}_i^n}{\mathbf{w}_{h,j} - \mathbf{Q}_i^n}\right), & \text{if } \mathbf{w}_{h,j} - \mathbf{Q}_i^n > 0 \\ \min\left(1, \frac{\mathbf{Q}_i^{min} - \mathbf{Q}_i^n}{\mathbf{w}_{h,j} - \mathbf{Q}_i^n}\right), & \text{if } \mathbf{w}_{h,j} - \mathbf{Q}_i^n < 0 \\ 1 & \text{if } \mathbf{w}_{h,j} - \mathbf{Q}_i^n = 0. \end{cases}$$

with $\mathbf{w}_{h,j} = \mathbf{w}_h(\mathbf{x}_j, t^n)$ (ratios and inequalities are to be understood componentwise). Then, the slope limiter is defined as

$$\Phi_i = \min_j (\Phi_{i,j}).$$

Finally, second order of accuracy in *time* is achieved by an element-local predictor stage that evolves the reconstructed polynomials $\mathbf{w}_i(\mathbf{x}, t^n)$ within each element $T_i^n(t)$ during the time interval $[t^n, t^{n+1}]$. The piecewise space-time polynomials are denoted by $\mathbf{q}_h(\mathbf{x}, t)$, and are of the form

$$\begin{aligned}\mathbf{q}_h(\mathbf{x},t)|_{T_i^n} = \mathbf{q}_i(\mathbf{x},t^n) = \mathbf{Q}_i^n + \Phi_i \nabla \mathbf{Q}_i(\mathbf{x}-\mathbf{x}_i) + \partial_t \mathbf{Q}_i (t-t^n),\\ \mathbf{x} \in T_i(t),\ t \in [t^n, t^{n+1}].\end{aligned} \tag{2.14}$$

The value of $\partial_t \mathbf{Q}_i$ can be easily recovered through the strong form of the PDE

$$\partial_t \mathbf{Q} = -\mathbf{f}_x(\mathbf{Q}) - \mathbf{g}_y(\mathbf{Q}) + \mathbf{S}(\mathbf{Q}), \tag{2.15}$$

where the r.h.s of (2.15) can be easily computed. Indeed the fields $\mathbf{f}$ and $\mathbf{g}$ over T_i^n are approximated as linear fields

$$\begin{aligned}\mathbf{f}(x,y) &= \mathbf{f}_0 + \mathbf{f}_1(x - x_i) + \mathbf{f}_2(y - y_i),\\ \mathbf{g}(x,y) &= \mathbf{g}_0 + \mathbf{g}_1(x - x_i) + \mathbf{g}_2(y - y_i),\end{aligned}$$

whose coefficients $\mathbf{f}_i$ and $\mathbf{g}_i$ are determined interpolating the values of the fields computed at the vertices j of T_i^n, i.e.

$$\mathbf{f}(\tilde{\mathbf{w}}_h(\mathbf{x}_j, t^n)) \quad \text{and} \quad \mathbf{g}(\tilde{\mathbf{w}}_h(\mathbf{x}_j, t^n)), \quad \forall j = 1, \ldots, N\mathbf{v}(i).$$

Then $\mathbf{f}_1 = \mathbf{f}_x(\mathbf{Q})$ and $\mathbf{g}_2 = \mathbf{g}_y(\mathbf{Q})$. Besides, the source $\mathbf{S}(\mathbf{Q})$ is computed at the barycenter $\mathbf{x}_i$ of T_i^n, i.e. $S(\tilde{\mathbf{w}}_h(\mathbf{x}_i, t^n))$.

2.1.3 Direct ALE FV scheme

Once $\mathbf{q}_h(\mathbf{x}, t)$ has been computed for each T_i^n, we are in the position to introduce the one-step space-time Finite Volume scheme. As proposed in 2 [22], the governing PDE (1.1) is first reformulated in a space–time divergence form as

$$\tilde{\nabla} \cdot \tilde{\mathbf{F}} = \mathbf{S}(\mathbf{Q}), \tag{2.16}$$

with

$$\tilde{\nabla} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial t} \right)^T, \qquad \tilde{\mathbf{F}} = (\mathbf{F}, \mathbf{Q}) = (\mathbf{f}, \mathbf{g}, \mathbf{Q}), \tag{2.17}$$

and it is integrated over each space–time control volume C_i^n

$$\int_{C_i^n} \tilde{\nabla} \cdot \tilde{\mathbf{F}} \, d\mathbf{x}dt = \int_{C_i^n} \mathbf{S} \, d\mathbf{x}dt. \tag{2.18}$$

Then, the volume integral of the space-time divergence flux on the l.h.s. can be rewritten as a surface integral by applying the Gauss theorem

$$\int_{\partial C_i^n} \tilde{\mathbf{F}} \cdot \ \tilde{\mathbf{n}} \ dS = \int_{C_i^n} \mathbf{S} \, d\mathbf{x}dt, \tag{2.19}$$

where $\tilde{\mathbf{n}} = (\tilde{n}_x, \tilde{n}_y, \tilde{n}_t)$ is the outward pointing space–time unit normal vector on the space–time surface ∂C_i^n, derived in (2.4) and (2.10). Substituting the physical boundary fluxes $\tilde{\mathbf{F}} \cdot \ \tilde{\mathbf{n}}$ with appropriate numerical fluxes leads to a consistent and conservative Finite Volume discretization. In principle, the entire structure of the numerical scheme is already given by (2.19).

The final one–step direct ALE Finite Volume scheme is then obtained from equation (4.4) as

$$|T_i^{n+1}| \, \mathbf{Q}_i^{n+1} = |T_i^n| \, \mathbf{Q}_i^n - \sum_j \int_0^1 \int_0^1 |\partial C_{ij}^n| \tilde{\mathbf{F}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} \, d\chi d\tau + \int_{C_i^n} \mathbf{S}(\mathbf{q}_h) \, d\mathbf{x}dt, \tag{2.20}$$

where the discontinuity of the solution at the space–time surface ∂C_{ij}^n is resolved by an *ALE numerical flux function* $\tilde{\mathbf{F}}_{ij} \cdot \tilde{\mathbf{n}}_{ij}$, which computes the flux between two neighbors across the intermediate space–time lateral surface. In particular when the lateral surface is shared between more than two control volumes (as shown in Figure 2.3) we have to compute the fluxes across each sub-surface and sum each contribution. The results presented in this chapter are obtained using a Rusanov–type or an Osher–type ALE flux; let us introduce some notation before presenting the fluxes. Let $\mathbf{q}_h^-(\mathbf{x}, t)$ be the reconstructed numerical solution inside the element C_i^n and $\mathbf{q}_h^+(\mathbf{x}, t)$ be the reconstructed numerical solution inside one of the neighbors of T_i^n through $\partial C_{i,j}^n$; let $\mathbf{q}_h^-$ and $\mathbf{q}_h^+$ the values of these polynomials evaluated at the space-time midpoint $M_{i,j}^n$ of the considered piece of the lateral surface. Define the ALE Jacobian matrix w.r.t. the normal direction in space

$$\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\mathbf{Q}) = \left(\sqrt{\tilde{n}_x^2 + \tilde{n}_y^2} \right) \left[\frac{\partial \mathbf{F}}{\partial \mathbf{Q}} \cdot \mathbf{n} - (\mathbf{V} \cdot \mathbf{n}) \, \mathbf{I} \right], \qquad \mathbf{n} = \frac{(\tilde{n}_x, \tilde{n}_y)^T}{\sqrt{\tilde{n}_x^2 + \tilde{n}_y^2}}, \tag{2.21}$$

with $\mathbf{I}$ representing the identity matrix and $\mathbf{V} \cdot \mathbf{n}$ denoting the local normal mesh velocity.

Then, the expression for the *Rusanov flux* is given by

$$\tilde{\mathbf{F}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} = \frac{1}{2}\left(\tilde{\mathbf{F}}(\mathbf{q}_h^+) + \tilde{\mathbf{F}}(\mathbf{q}_h^-)\right) \cdot \tilde{\mathbf{n}}_{ij} - \frac{1}{2} s_{\max}\left(\mathbf{q}_h^+ - \mathbf{q}_h^-\right), \tag{2.22}$$

where $s_{\max}$ is the maximum eigenvalue of $\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\mathbf{q}_h^+)$ and $\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\mathbf{q}_h^-)$.

The *Osher–type flux* formulation has been proposed in the Eulerian framework in [74] and has been subsequently extended to moving meshes in one and two space dimensions in [77] and [22], respectively. It is defined as

$$\tilde{\mathbf{F}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} = \frac{1}{2}\left(\tilde{\mathbf{F}}(\mathbf{q}_h^+) + \tilde{\mathbf{F}}(\mathbf{q}_h^-)\right) \cdot \tilde{\mathbf{n}}_{ij} - \frac{1}{2}\left(\int_0^1 \left|\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\boldsymbol{\Psi}(s))\right| ds\right)\left(\mathbf{q}_h^+ - \mathbf{q}_h^-\right), \tag{2.23}$$

where we choose to connect the left and the right state across the discontinuity using a simple straight–line segment path

$$\boldsymbol{\Psi}(s) = \mathbf{q}_h^- + s\left(\mathbf{q}_h^+ - \mathbf{q}_h^-\right), \qquad 0 \le s \le 1. \tag{2.24}$$

The absolute value of the dissipation matrix in (2.23) is evaluated as usual as

$$|\mathbf{A}| = \mathbf{R}|\boldsymbol{\Lambda}|\mathbf{R}^{-1}, \qquad |\boldsymbol{\Lambda}| = \mathrm{diag}\left(|\lambda_1|, |\lambda_2|, ..., |\lambda_\nu|\right), \tag{2.25}$$

where $\mathbf{R}$ and $\mathbf{R}^{-1}$ denote, respectively, the right eigenvector matrix and its inverse of the ALE Jacobian $\mathbf{A}_{\mathbf{n}}^{\mathbf{V}} = \frac{\partial \mathbf{F}}{\partial \mathbf{Q}} \cdot \mathbf{n} - (\mathbf{V} \cdot \mathbf{n})\mathbf{I}$.

Note that the numerical flux is always composed by two parts: a standard one, given by the fluxes computed at the left and the right of the interface, and a second one, called *numerical viscosity*, that stabilizes the method and depends on the eigenstructure of the Jacobian matrix and on the jump of the conserved variables at the interface.

In (2.20) the time step Δt is given by

$$\Delta t = \mathrm{CFL} \min_{T_i^n} \frac{d_i}{|\lambda_{\max,i}|}, \qquad \forall T_i^n \in \Omega^n, \tag{2.26}$$

where CFL is the Courant-Friedrichs-Levy number, d_i represents the encircle diameter of element T_i^n and $|\lambda_{\max,i}|$ is the maximum absolute value of the eigenvalues computed from the solution $\mathbf{Q}_i^n$ in T_i^n. As stated in [150] in Chapter 16, for linear stability in two space dimensions the Courant number must satisfy CFL ≤ 0.5. Finally, note that in time we have used the upwind flux due to the causality principle, which naturally leads to the terms $|T_i^n|\,\mathbf{Q}_i^n$ and $|T_i^{n+1}|\,\mathbf{Q}_i^{n+1}$ in (2.20).

Properties

We underline that the integration over a closed space–time control volume, as done above, automatically satisfies the so-called geometric conservation law (GCL), since from the Gauss theorem follows

$$\int_{\partial \mathcal{C}_i^n} \tilde{\mathbf{n}} \, dS = 0. \tag{2.27}$$

The relation between (2.27) and the usual form of the *GCL* that is typically employed in the community working on Lagrangian schemes has been established in the appendix of [23]. For all the numerical test problems shown later it has been explicitly verified that property (2.27) holds for all elements and for all time steps up to machine precision, even on moving nonconforming meshes. Moreover the scheme is locally and globally *conservative* for mass and inertial momentum.

We would like to emphasize that the direct ALE scheme presented here does in general *not* lead to a vanishing mass flux across element boundaries, similar to previous work on direct ALE schemes presented in [22, 23]. The mass flux is exactly zero only for isolated contact discontinuities moving in uniform flow when using appropriate Riemann solvers that resolve contact waves, like the Godunov method, or the Roe, HLLC, HLLEM and Osher flux.

2.2 Nonconforming mesh motion

In this section we focus on the detailed description of the procedure needed to determine how the computational mesh moves. Indeed, this is a crucial point in any moving mesh algorithm, because following the fluid exactly with its own velocity may become sometimes very complex, leading to highly deformed or degenerate control volumes. Moreover these damaged elements may drastically reduce the admissible time step, which is computed under a classical CFL stability condition, see (2.26), causing a slowdown or even an interruption of the algorithm. Many techniques have been developed in order to overcome this problem, and what we propose here is a novel and effective nonconforming treatment of the sliding lines, separating two regions with different characteristics.

In particular, at each time step the algorithm computes the new node positions through the following intermediate stages. First, the edges along relevant shear flows are detected (see Section 2.2.2) and the nodes on these edges are

marked as problematic. Then the new node positions are computed according to the type of the considered node, in particular

a) Regular non-hanging nodes that are not in regions of relevant shear flow (i.e. they have not been marked as problematic) are evolved using a mass-weighted Cheng and Shu node solver, see (2.31).

b) Regular non-hanging nodes that are in regions of relevant shear flow (i.e. they have been marked as problematic) are *doubled*; their new position is projected along the nearest interface edge, and they subsequently change their type from regular non-hanging nodes to *hanging* nodes (refer to Section 2.2.4).

c) Hanging nodes on an edge are allowed to slide only along that edge (see Section 2.2.5), and if they get too close to other nodes, they are *merged* together (deleted), eventually changing back their type from hanging nodes to regular non-hanging nodes (refer to Section 2.2.6).

Associated to *b*) and *c*) there is a procedure for updating all the connectivities of the unstructured mesh: we would like to underline that all these procedures are done at a *local* level, so affecting only the neighbors of the considered node and nothing else.

2.2.1 Connectivity matrices

Since the core of the proposed method is the motion and the changing of the nonconforming mesh topology in time, we have to know all the connectivities of the mesh and to maintain them updated. In this way we will have enough information both to rearrange the mesh after the insertion of a new node, or the fusion of two existing nodes, and to know all the neighbors of each space–time lateral surface during the numerical flux computation.

As in the standard conforming case for each element T_i^n we save the global numbering of its vertexes $V_1, \ldots, V_{\mathrm{Nv}(i)}$ in row i of a matrix called `tri` in counterclockwise order, and in matrix `Elem2Edge` we store the global numbering of its edges $E_1, \ldots, E_{\mathrm{Nv}(i)}$. However, in the nonconforming case, additional connectivity tables are needed, since more than two elements can share the same edge and more than two points can belong to it. For each edge E_j^n, we store the elements that share it in row j of matrix `Edge2Elem`, and all the nodes that belong to E_j^n in row j of matrix `Edge2Vertex` in such a way that the first two entries of each row contain the endpoints of the corresponding edge. Then, for each node we memorize the edge to which it belongs in `Vertex2Edge`

(both if this node is an endpoint of the edge or an intermediate point) and the elements for which it is a vertex in `Vertex2Elem` (note that if a node N_i belongs to an edge of an element but it is not one of its vertices, that element will not appear in the row N_i of this last matrix). Moreover, each node has a label that indicates whether the node is free to move everywhere, if it has been doubled, or if it is constrained to slide along a particular edge, i.e. if it is a hanging node.

Besides, we allow our data structures to be completely dynamic in such a way that nodes and edges can appear and disappear in time: so rows can be added to our matrices or be nullified, and the information regarding which global numbering of nodes and edges is currently used is always available.

2.2.2 Shear interface detector

Since the sliding interfaces are not defined *a priori* by the user, at the beginning of each time step the algorithm has first to identify along which *edges* the shear interfaces lie, and mark the corresponding edges and nodes. Basically an edge e will be considered at the interface if the tangential velocity difference ΔV_e across e exceeds a certain threshold value κ_e. So for each edge we need to compute ΔV_e and κ_e.

Given the set of nodes S_e^n over the edge e, and the set of neighbors S_j^n of each node j, the threshold value κ_e is computed as

$$\kappa_e = \min_{j \in S_e^n} \kappa_j, \qquad \text{with} \quad \kappa_j = \max_{i \in S_j^n} \left(\frac{\alpha \, d_i}{||J_i||} \right), \tag{2.28}$$

where d_i is the encircle diameter of element T_i^n, J_i is the Jacobian of the transformation that maps element T_i^n in the corresponding reference element, the norm is the two-norm of Frobenius divided by $\sqrt{2}$ (other matrix norms could also be used), and α is chosen in $[0, 1]$ according to the desired sensitivity of the detector. If the velocity jump at the interface is very large, the value of α does not matter. Instead, where the velocity field changes smoothly, the number of interfaces, and as a consequence the number of new nodes, will be dependent on α. Moreover, in the limit $\alpha \to +\infty$ we recover the standard conforming algorithm.

Once the threshold value has been fixed we loop over all the edges of the mesh: for each edge e we consider all its neighbors and we compute their tangential velocity with respect to e. Say, for example, that two elements $A = T_a^n$ and $B = T_b^n$ with area $|T_a^n|$ and $|T_b^n|$ share the same edge e and their tangential

velocities are $v_{t,a}^n$ and $v_{t,b}^n$. If the quantity ΔV_e exceeds κ_e

$$\Delta V_e = \frac{\left| v_{t,a}^n \, |T_a^n| - v_{t,b}^n \, |T_b^n| \right|}{\left(|v_{t,a}^n||T_a^n| + |v_{t,b}^n||T_b^n| + \varepsilon \right)} \geq \kappa_e, \tag{2.29}$$

with $\varepsilon = 10^{-14}$ to avoid division by zero, then edge e is marked as an edge at a shear interface, and the elements A and B are divided into two different groups: the elements on the left and the ones on the right with respect to this particular edge e. Afterwards, we also need to find the *nodes* that have *to be doubled* and to separate their Voronoi neighbors (the elements stored in `Vertex2Elem`) into two groups. So we loop over the nodes considering the ones which belong to an interface edge. If in their list of Voronoi neighbors there are elements from both the sides of the interface, according to the previous subdivision, we mark them and we separate their Voronoi neighbors into two groups which are stored in two matrices. Note that the two cycles, the one over the edges and the other over the nodes, are not nested one into the other, but are run one after the other.

2.2.3 Node motion

At this point we are able to distinguish between nodes far away from the interfaces, hanging nodes and nodes which lie at the interface. So we loop over the nodes and according to their labels we choose what to do. First, consider a *regular* non-hanging node k located in a smooth region. We compute its coordinates at the new time level t^{n+1} simply by

$$\mathbf{x}_k^{n+1} = \mathbf{x}_k^n + \Delta t \overline{\mathbf{V}}_k^n, \tag{2.30}$$

where $\overline{\mathbf{V}}_k^n$ is obtained using the node solver of Cheng and Shu. Cheng and Shu introduced a very simple and general formulation for obtaining the final node velocity, which is chosen to be the arithmetic average velocity among all the contributions coming from the Voronoi neighbor elements $\mathcal{V}_k^n$. Moreover, following the ideas presented in [26] we take a mass weighted average velocity among the neighborhood $\mathcal{V}_k^n$, that is,

$$\overline{\mathbf{V}}_k^n = \frac{1}{\mu_k} \sum_{T_j^n \in \mathcal{V}_k} \mu_{k,j} \overline{\mathbf{V}}_{k,j} \tag{2.31}$$

with

$$\mu_k = \sum_{T_j^n \in \mathcal{V}_k} \mu_{k,j}, \quad \mu_{k,j} = \rho_j^n |T_j^n|. \tag{2.32}$$

The local weights $\mu_{k,j}$, which are the masses of the elements T_j^n, are defined by multiplying the cell averaged value of density ρ_j^n (or water depth h_j^n for shallow water flows) with the cell area $|T_j^n|$. The local contributions $\overline{\mathbf{V}}_{k,j}$ in a pure Lagrangian context represent the fluid velocity in the j^{th} neighbor of vertex k, while in the ALE framework they can be obtained either according to an arbitrary, prescribed mesh velocity function or by the local fluid velocity.

Now let us consider the *nodes at the interfaces*. The following considerations are carried out by supposing for the moment that each interface is separated from the others and lies on a straight line. Even if this is a rigid constraint, already with this configuration interesting test cases can be studied; a proof of concept that the extension to the general case of piece-wise linear interfaces is feasible will be presented in Section 2.5.

2.2.4 Insertion of a new node

The first situation we encounter is a node k that has some of its Voronoi neighbors on the left of the interface, call them *left neighbors*, $\mathcal{V}_{k,\text{left}}$, and the others on the right of the same interface, call them *right neighbors*, $\mathcal{V}_{k,\text{right}}$; these two sets of neighbors have been provided by the *interface detector* described above. We apply the node solver of Cheng and Shu at the two sets of neighbors obtaining two different new coordinates

$$\tilde{\mathbf{x}}_{k,\,\text{left}}^{n+1} = \mathbf{x}_k^n + \Delta t \sum_{T_j^n \in \mathcal{V}_{k,\text{left}}} \frac{\mu_{k,j}}{\mu_k} \overline{\mathbf{V}}_{k,j}, \quad \text{and} \quad \tilde{\mathbf{x}}_{k,\,\text{right}}^{n+1} = \mathbf{x}_k^n + \Delta t \sum_{T_j^n \in \mathcal{V}_{k,\text{right}}} \frac{\mu_{k,j}}{\mu_k} \overline{\mathbf{V}}_{k,j}. \tag{2.33}$$

We allow this kind of nodes to move only along the interface, so basically according to their averaged tangential velocity with respect to the interface: for this reason we need to find the nearest interface edges and to project onto them the coordinates in (2.33) obtaining thus $\mathbf{x}_{k,\,\text{left}}^{n+1}$ and $\mathbf{x}_{k,\,\text{right}}^{n+1}$. Call the nearest interface edges belonging to the left elements e_1^ℓ and e_2^ℓ, and the nearest interface edges belonging to the right elements e_1^r and e_2^r (suppose also that $e_1^{\ell,r}$ are closer to $\tilde{\mathbf{x}}_{k,\,\text{left}}^{n+1}$ than to $\tilde{\mathbf{x}}_{k,\,\text{right}}^{n+1}$, so that $\mathbf{x}_{k,\,\text{left}}^{n+1}$ is obtained by projecting $\tilde{\mathbf{x}}_{k,\,\text{left}}^{n+1}$ onto e_1^ℓ, and $\mathbf{x}_{k,\,\text{right}}^{n+1}$ by projecting $\tilde{\mathbf{x}}_{k,\,\text{right}}^{n+1}$ onto e_1^r). We decide to assign $\mathbf{x}_{k,\text{left}}^{n+1}$ as new coordinate to the old node k

$$\mathbf{x}_k^{n+1} = \mathbf{x}_{k,\text{left}}^{n+1} \tag{2.34}$$

and to create a new node with global number k_{new} and coordinates (at time n and $n+1$)

$$\mathbf{x}^n_{k_{\text{new}}} = \mathbf{x}^n_k \quad \text{and} \quad \mathbf{x}^{n+1}_{k_{\text{new}}} = \mathbf{x}^{n+1}_{k,\,\text{right}}. \tag{2.35}$$

The global number k_{new} can be larger than N^n_{node} if all the numbers between 1 and N^n_{node} are currently used, otherwise we choose the first of the unused numbers (indeed if two nodes have been merged together one of their global numbers is no more used, see Section 2.2.6).

Now we have to update the connectivity tables taking into account the insertion of this new node. See also Figure 2.4 to follow our construction. First, in matrix `tri` we substitute k with k_{new} in all the right elements; moreover, we consider matrix `Vertex2Elem` and in row k we leave only the left elements and we put the others in row k_{new} (because now k is no more a vertex for the right neighbors). Then we have to deal with the edges: if $e^\ell_1 = e^r_1$ we need to substitute e^r_1 with a new edge $e^r_{1_{\text{new}}}$. In matrix `Elem2Edge` all the right neighbors change e^r_1 with $e^r_{1_{\text{new}}}$, and in matrix `Edge2Elem` we insert a new row $e^r_{1_{\text{new}}}$ equal to row e^r_1 (the new edge inherits all the characteristics from the old one). The same has to be done if $e^\ell_2 = e^r_2$. The endpoints of these new edges are the endpoints of the substituted edges seen from the right (so basically there is k_{new} instead of k). The endpoints of the left edges do not change. Besides we add k as intermediate point in e^r_1 and k_{new} as intermediate point of e^ℓ_2, (note that an edge is allowed to have more than one intermediate point). In this way also matrix `Edge2Vertex` has been updated. Matrix `Vertex2Edge` is easily modified at the same time. Finally, we have to revise the list of neighbors: in particular the edges that gained an intermediate point (e^r_1 and e^ℓ_2) gain also neighbors. In particular the new neighbors of e^r_1 are the left neighbors of e^ℓ_2 and the new neighbors of e^ℓ_2 are the right neighbors of e^r_1. This allows us to update `Edge2Elem` and `Elem2Edge`.

At the end we mark with a label the nodes which are intermediate for an edge: we call them *hanging nodes* and they are constrained to move along that edge. Note that in the case of straight slip-lines no distinction between master and slave nodes is required, since both will move along the same straight interface. To extend the algorithm to the case of piece-wise linear interfaces, this distinction is introduced in such a way that only slave nodes will be constrained to slide along edges, while the master nodes can move freely. For some first preliminary results concerning the extension to completely general slide lines, see Section 2.5.

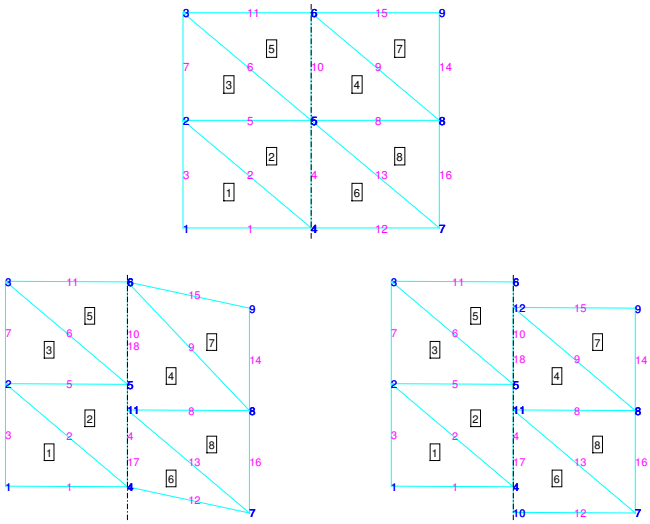


Figure 2.4: Example of how to double a node. At the top we show the situation before a nonconforming motion, and at the bottom after the motion and the corresponding update of the connectivity matrices. Precisely at the bottom on the left we have supposed to move in a nonconforming way only one of the nodes at the interface, while the realistic motion of all the nodes at the interface is shown on the right. The black vertical dotted line represents the interface: suppose that the elements on the left $\{1, 2, 3, 5\}$ move with velocity $\mathbf{v} = (0, 2)$ and the elements on the right $\{4, 6, 7, 8\}$ move with velocity $\mathbf{v} = (0, -2)$. We want to double vertex number $k = 5$, so we insert a new node $k_{\text{new}} = 11$. The nearest interface edges on which we project the new coordinates of node 5 are $e_1^\ell = e_1^r = 10$ and $e_2^\ell = e_2^r = 4$. Note that edges $e_1^{\ell,r}$ are closer to k than to k_{new}. Since the edges from the left and from the right are equal we create two new edges $e^r_{1_{\text{new}}} = 18$ and $e^r_{2_{\text{new}}} = 17$. The endpoints of edges 10 and 4 remain untouched. Edge 4 gains an intermediate point, the node 11, and edge 18 gains the node 5. To better understand we list now the vertexes of each edge at the end of the updating process (first we write the endpoints and then, if existing, the intermediate points): $e_1^\ell = 10 \to \{5, 8\}$, $e_2^\ell = 4 \to \{4, 5, 11\}$, $e_1^r = 18 \to \{11, 6, 5\}$ and $e_2^r = 17 \to \{4, 11\}$. Finally, elements $\{1, 3, 5, 6, 7, 8\}$ maintain the same edge neighbors, while the neighbors of elements 2 and 4 are augmented: indeed edge 4 has neighbors $\{2, 6, 4\}$ and edge 18 has neighbors $\{4, 5, 2\}$. Note that the situation on the right appears to be more complicated only because also nodes 4 and 6 have been doubled and so the corresponding update of the connectivity matrices has been done.

2.2.5 Hanging nodes

Consider a hanging node k which lies on the edge e: it is at the interface and it is a vertex only of elements lying on one side of the interface, so its Voronoi neighbors are in the same smooth region. However it is not free to move everywhere but it must slide along that edge, to avoid creation of holes or superposition of elements in the mesh.

We apply the averaged node solver of Cheng and Shu among its Voronoi neighbors, we find its new coordinates $\tilde{\mathbf{x}}_k^{n+1}$ and we project them over edge e, obtaining $\mathbf{x}_k^{n+1}$. Now, we compute also the new coordinates of the other points over edge e. If two of them are sufficiently close, we decide to merge them (see Section 2.2.6), otherwise the computed coordinates $\mathbf{x}_k^{n+1}$ are the new coordinates of such a node and no update of the connectivity matrices is required.

2.2.6 Fusion of two existing nodes

Suppose we computed the new coordinates at time t^{n+1} of all the nodes k_i over the same edge e denoted by $\mathbf{x}_{k_i}^{n+1}$, which are assumed to be already projected onto the straight line spanned by edge e. If the new coordinates of two of them, say k_1 and k_2, are too close, we decide to merge them. Moreover, if one intermediate node of edge e falls outside the edge, we decide to merge it with the closest endpoint of the edge. Since the loop over the nodes is carried out according to the increasing global numbering of the nodes, we decide to remove the node with the largest global number (we call it *dead node*, k_{dn}) because we have not worked with it yet, and to maintain the one with the smallest global numbering (call it *fusion node*, k_{fn}) assigning to it as new coordinates the average between $\mathbf{x}_{k_1}^{n+1}$ and $\mathbf{x}_{k_2}^{n+1}$

$$\mathbf{x}_{k_{\mathrm{fn}}}^{n+1} = \frac{\mathbf{x}_{k_1}^{n+1} + \mathbf{x}_{k_2}^{n+1}}{2}. \tag{2.36}$$

We assign the same coordinates also to the dead node

$$\mathbf{x}_{k_{\mathrm{dn}}}^{n+1} = \frac{\mathbf{x}_{k_1}^{n+1} + \mathbf{x}_{k_2}^{n+1}}{2}. \tag{2.37}$$

Now, we need to update the connectivity tables. See also Figure 2.5 to follow our construction. This process is somehow more complicated than the nodes splitting. Indeed when we insert a new node at time t^{n+1} we only add information without losing anything about the previous time step, and even if

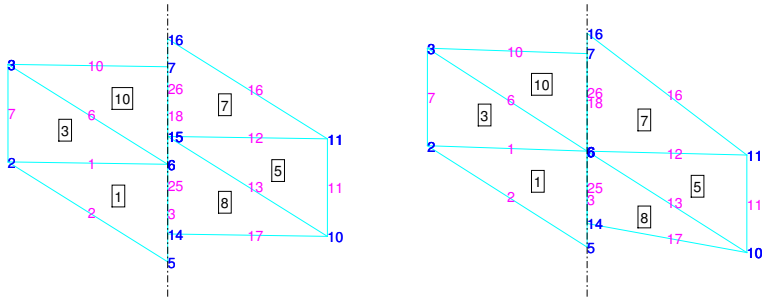


Figure 2.5: Example of how to merge two existing nodes. The black dotted line represents the interface: suppose that the elements on the left $\{1, 3, 10\}$ move with positive velocity and the elements on the right $\{5, 7, 8\}$ move with negative velocity. On the left we show the situation at time t^n and on the right at time t^{n+1}. Nodes 6 and 15 at t^{n+1} will be so close that we decide to merge them (as in the previous example, for the sake of clarity, we present on the right the situation after the fusion of only two nodes). We maintain the smallest global number so $k_{\text{fn}} = 6$ and we remove $k_{\text{dn}} = 15$. In `triNew` elements $\{5, 7, 8\}$ substitute $k_{\text{dn}} = 15$ with $k_{\text{fn}} = 6$. Note that in `tri` nothing changes, so some elements refer to node 6 and some other to node 15, but everything works because at time t^{n+1} they have the same new coordinates $\mathbf{x}_{k_{\text{dn}}}^{n+1} = \mathbf{x}_{k_{\text{fn}}}^{n+1}$ and at the successive time step t^{n+2} `tri` will no longer exist because it will be overwritten by `triNew`. In row k_{fn} of matrix `Vertex2ElemNew` there are listed elements $\{1, 3, 5, 7, 8, 10\}$, while row k_{dn} is empty. In row k_{fn} of matrix `Vertex2EdgeNew` there are edges $\{1, 3, 6, 12, 13, 18, 25, 26\}$, while row k_{dn} is empty. List `Edge[dn-fn]` contains edges $\{18, 25\}$ and list `Elem[dn-fn]` contains elements $\{8, 10\}$. Knowing these lists we can update matrices `Edge2ElemNew` because we remove element 8 from the neighbor of edge 18 and element 10 from the neighbors of edge 25. In this case even if we removed the segment $\overline{6, 15}$ no edge becomes equal so we do not need to merge edges neither to update `Elem2EdgeNew`.

it is true that the right neighbors of a doubled node k change their node k with a new one k_{new}, we can dispose of all its reference simply by giving to k_{new} at time t^n the same coordinates of k, see also (2.33). On the contrary, when we remove a node we lose all the reference to it, reference that, only for time t^{n+1}, we still need during the computation of the interface fluxes in the Finite Volume scheme (it is for this reason that in (2.37) we have assigned the coordinates $\mathbf{x}_{k_{\text{dn}}}^{n+1}$ even to the dead node). So we decide to duplicate some of the connectivity tables, creating `triNew`, `Elem2EdgeNew`, `Edge2ElemNew`, `Edge2VertexNew`, and

`Vertex2ElemNew`. During the insertion procedure we modify in the same way both the old and the new matrices, while during the fusion we modify only the `new` matrices. Hence we can use the old ones in the Finite Volume scheme, because they store all the needed information (for example they refer both to the dead and the fusion node which have the same coordinates at the new time t^{n+1}), while when we advance in time, to t^{n+2}, we maintain updated only the new ones because the information about two previous time steps are no longer necessary and so we can overwrite the old connectivity matrices with the new ones.

First, in matrix `triNew` we substitute k_{dn} with k_{fn} in all the neighbors of the dead node; moreover, we consider matrix `Vertex2ElemNew`, in row k_{fn} we put both the neighbors of the dead and the fusion node and we nullify row k_{dn}. We do the same with matrix `Vertex2EdgeNew`: we nullify row k_{dn} and we put in row k_{fn} all the edges that contain k_{fn} or k_{dn}. Then all the edges that contain k_{dn} substitute it with k_{fn} (in matrix `Edge2VertexNew`), whereas the edges with both k_{dn} and k_{fn} (that we memorize in a list `Edge[dn-fn]`) remove k_{dn}. We note that merging k_{dn} and k_{fn} we are removing the segment in between, so we look for the edges that contain it (listed in `Edge[dn-fn]`) and its neighbor elements that we list in `Elem[dn-fn]`. We update now matrix `Edge2ElemNew` because the edges in `Edge[dn-fn]` have no more one of the neighbors in `Elem[dn-fn]`. Afterward we check if the absence of this segment makes some edges in `Edge[dn-fn]` equal: in this case we remove one of them (the one with the largest global number) and we update correspondingly the new connectivity matrices. Besides we modify the labels telling us if a node is hung to some edges and which nodes and edges are currently existing. This last step prevents us to work again with disappeared nodes and allows us to reuse their global numbering when we want to insert a new node or a new edge.

2.3 Shallow water equations

In the next Section, we will solve a large set of numerical tests in order to validate the presented nonconforming direct ALE scheme. The robustness of the method is checked both on smooth and discontinuous problems related to the shallow water equations written both in Cartesian and in polar coordinates. These equations are briefly recalled here.

2.3.1 Cartesian coordinates

We consider the shallow water equations with flat bottom topography, which can be cast into form (2.1) with

$$\mathbf{Q} = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}, \tag{2.38}$$

where the variables h, u and v are respectively the water depth, the velocity in $x-$direction, and the velocity in $y-$direction. The term $\overline{p} = \frac{1}{2}gh^2$ denotes the averaged pressure force along the water depth (normalized with the fluid density ρ), where g is the gravity acceleration along the vertical direction. Since the shallow water equations are derived from depth-integrating the incompressible Navier-Stokes equations assuming that the horizontal length scale is much larger than the vertical one, the vertical pressure is assumed to be hydrostatic.

The Jacobian matrices, necessary for the computation of the ALE Jacobian matrix in (2.21), are

$$\frac{\partial \mathbf{f}}{\partial \mathbf{Q}} = \begin{pmatrix} 0 & 1 & 0 \\ a^2 - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix}, \quad \frac{\partial \mathbf{g}}{\partial \mathbf{Q}} = \begin{pmatrix} 0 & 0 & 1 \\ -uv & v & u \\ a^2 - v^2 & 0 & 2v \end{pmatrix}. \tag{2.39}$$

where $a = \sqrt{gh}$.

2.3.2 Polar coordinates

We recover here the formulation of the shallow water equations in polar coordinates. Consider the usual relation between polar (r, φ) and Cartesian (x, y) coordinates

$$x = r\cos\varphi\,, \qquad \text{and} \qquad y = r\sin\varphi\,, \tag{2.40}$$

and the corresponding relations for the derivatives

$$\frac{\partial}{\partial x} = \cos\varphi\,\frac{\partial}{\partial r} - \frac{\sin\varphi}{r}\,\frac{\partial}{\partial \varphi}, \quad \text{and} \quad \frac{\partial}{\partial y} = \sin\varphi\,\frac{\partial}{\partial r} + \frac{\cos\varphi}{r}\,\frac{\partial}{\partial \varphi} \tag{2.41}$$

and let u_r and u_φ be respectively the radial and the tangential component of the velocity, linked to u and v by

$$u = \cos\varphi\, u_r - \sin\varphi\, u_\varphi, \qquad v = \sin\varphi\, u_r + \cos\varphi\, u_\varphi\ . \tag{2.42}$$

Now by substituting into (2.38) the expressions given in (2.42) and (2.41), after some calculations, we derive a new set of hyperbolic equations

$$\begin{cases} \frac{\partial rh}{\partial t} + \frac{\partial rhu_r}{\partial r} + \frac{\partial hu_\varphi}{\partial \varphi} = 0, \\ \frac{\partial rhu_r}{\partial t} + \frac{\partial}{\partial r}\left(rhu_r^2 + \frac{1}{2}grh^2\right) + \frac{\partial hu_r u_\varphi}{\partial \varphi} = hu_\varphi^2 + \frac{1}{2}gh^2, \\ \frac{\partial rhu_\varphi}{\partial t} + \frac{\partial rhu_r u_\varphi}{\partial r} + \frac{\partial}{\partial \varphi}\left(hu_\varphi^2 + \frac{1}{2}gh^2\right) = -hu_r u_\varphi, \end{cases} \tag{2.43}$$

which, however, does not yet fit into the form (2.1), since the fluxes in the above system depend explicitly on the spatial coordinate r (i.e. the system is not autonomous). Thus, we add to the system the trivial equation,

$$\frac{\partial r}{\partial t} = 0, \tag{2.44}$$

obtaining finally

$$\begin{cases} \frac{\partial rh}{\partial t} + \frac{\partial rhu_r}{\partial r} + \frac{\partial hu_\varphi}{\partial \varphi} = 0, \\ \frac{\partial rhu_r}{\partial t} + \frac{\partial}{\partial r}\left(rhu_r^2 + \frac{1}{2}grh^2\right) + \frac{\partial hu_r u_\varphi}{\partial \varphi} = hu_\varphi^2 + \frac{1}{2}gh^2, \\ \frac{\partial rhu_\varphi}{\partial t} + \frac{\partial rhu_r u_\varphi}{\partial r} + \frac{\partial}{\partial \varphi}\left(hu_\varphi^2 + \frac{1}{2}gh^2\right) = -hu_r u_\varphi, \\ \frac{\partial r}{\partial t} = 0. \end{cases} \tag{2.45}$$

The vector of the conserved variables, the nonlinear flux, and the source can now be written as

$$\mathbf{Q} = \begin{pmatrix} rh \\ rhu_r \\ rhu_\varphi \\ r \end{pmatrix}, \ \mathbf{f} = \begin{pmatrix} rhu_r \\ rhu_r^2 + \frac{1}{2}grh^2 \\ rhu_r u_\varphi \\ 0 \end{pmatrix}, \ \mathbf{g} = \begin{pmatrix} hu_\varphi \\ hu_r u_\varphi \\ hu_\varphi^2 + \frac{1}{2}gh^2 \\ 0 \end{pmatrix}, \ \mathbf{S} = \begin{pmatrix} 0 \\ vhu_\varphi^2 + \frac{1}{2}gh^2 \\ -hu_r u_\varphi \\ 0 \end{pmatrix}. \tag{2.46}$$

and the Jacobian matrices, necessary for the computation of the ALE Jacobian matrix in (2.21), are

$$\frac{\partial \mathbf{f}}{\partial \mathbf{Q}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -u_r^2 + gh & 2u_r & 0 & -\frac{1}{2}gh^2 \\ -u_r u_\varphi & u_\varphi & u_r & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \frac{\partial \mathbf{g}}{\partial \mathbf{Q}} = \begin{pmatrix} 0 & 0 & \frac{1}{r} & -\frac{hu_\varphi}{r} \\ -\frac{u_r u_\varphi}{r} & \frac{u_\varphi}{r} & \frac{u_r}{r} & -\frac{hu_r u_\varphi}{r} \\ -\frac{u_\varphi^2}{r} + g\frac{h}{r} & 0 & \frac{2u_\varphi}{r} & -\frac{hu_\varphi^2}{r} - g\frac{h^2}{r} \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.47}$$

2.4 Numerical results

The test cases presented in this chapter are carried out by supposing straight slip lines and using either the Rusanov or the Osher type flux; the value of α in (2.28) is always taken equal to $\alpha = 1$ unless otherwise specified, and the CFL number is chosen as CFL$= 0.3$. Furthermore, the order of convergence is verified first fixing for the mesh motion an arbitrary velocity, then in the case of a steady vortex in equilibrium using the local fluid velocity.

2.4.1 Sanity checks: pure sliding

The numerical examples reported in this section are sanity checks testing the ability of the method to detect and maintain straight slip-line interfaces.

First, we consider the shallow water equations (2.38), with initial computational domain given by $\Omega(t_0) = [-2, 2] \times [0, 4]$ and the following initial condition

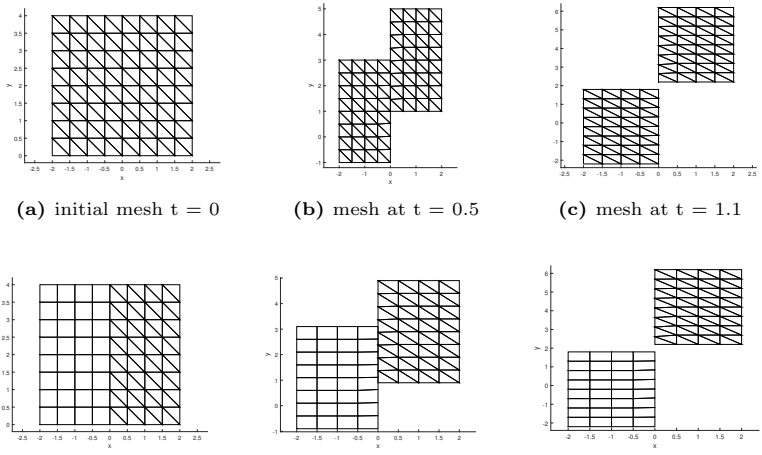
$$\mathbf{Q}(\mathbf{x}, 0) = \begin{cases} (1, 0, -2) & \text{if} \quad x \leq 0, \\ (1, 0, 2) & \text{if} \quad x > 0, \end{cases} \tag{2.48}$$

which also coincides with the exact solution at any time. We impose *wall* boundary conditions on the left and on the right side of the domain, respectively, whereas at the top and at the bottom of the domain we impose *transmissive* boundary conditions. In Figure 2.6 we show the numerical results over a triangular mesh and then over a mixed mesh composed of both, triangular and quadrilateral elements. The chosen mesh velocity coincides exactly with the fluid velocity, as in a pure Lagrangian context. At each time step we have verified that the total water volume is conserved up to machine precision both locally and globally and that relation (2.27), the GCL, is verified also up to machine precision.

Next, we consider as initial condition

$$\mathbf{Q}(\mathbf{x}, 0) = (1, 0, 0.5 \,\mathrm{floor}\,(2x)), \quad -2 \leq x \leq 2, \tag{2.49}$$

with $\mathrm{floor}(x) = \lfloor x \rfloor$ denoting the lower Gauss bracket, and we run our algorithm until a final time $t = 0.7$ with different *threshold* values, see (2.28), in such a way that there will be a different number of interfaces along which nodes have to be doubled and merged in time. The discretization of the computational domain is reported in Figure 2.7. Also in this case we reach the exact solution (that is the initial condition translated in the motion direction), the total volume



(a) initial mesh t = 0 **(b)** mesh at t = 0.5 **(c)** mesh at t = 1.1

(d) initial mesh t = 0 **(e)** mesh at t = 0.4 **(f)** mesh at t = 1.1

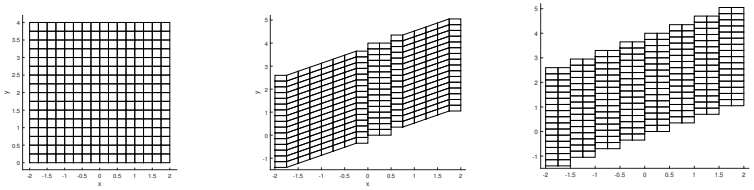
Figure 2.6: Slide lines test case with initial condition as in equation (2.48). The mesh is moved with the local fluid velocity, which at $x = 0$ is discontinuous: so nodes over there are handled in a nonconforming way. At the top we show the results obtained employing a triangular mesh and at the bottom using a mesh made of both triangular and quadrilateral elements. We report the mesh at three different computational times: note that the computational domain can also be split in two non connected parts. The level of the water, the total area and the total volume are conserved at any time step, and the solution coincides with the exact one up to machine precision.

of water is conserved and relation (2.27) is verified up to machine precision at each time step and on each element.

Finally, we want to show that the interface can be along any straight line (provided that edges lie over this line): we take as initial condition

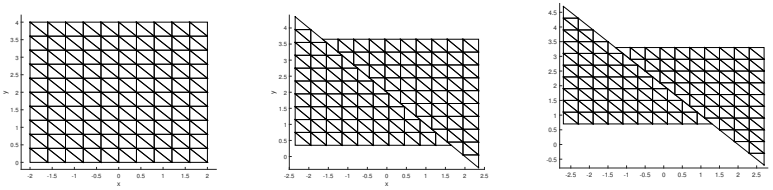
$$\mathbf{Q}(\mathbf{x}, 0) = \begin{cases} (1, -1, 1) & \text{if} \quad x + y - 2 \leq 0, \\ (1, 1, -1) & \text{if} \quad x + y - 2 > 0, \end{cases} \tag{2.50}$$

and in Figure 2.8 we report the computational domain at different times. Again, the numerical solution matches the exact one and as expected, the total volume is conserved and equation (2.27) is satisfied up to machine precision.



(a) initial mesh t = 0 **(b)** final mesh with $\alpha = 1$ **(c)** final mesh with $\alpha = 0.4$

Figure 2.7: Slide lines test case with initial condition as in equation (2.49). We start with a conforming quadrilateral mesh; using a value of $\alpha = 1$ in (2.28) we obtain only two slip-lines (at $x = 0$ and $x = 0.5$), whereas using $\alpha = 0.4$, which makes the detector more strict, the mesh slides along each straight line where the fluid velocity changes.



(a) initial mesh t = 0 **(b)** mesh at t = 0.35 **(c)** mesh at t = 0.7

Figure 2.8: Oblique slide line. We show the discretization of the computational domain at three different times. The corresponding numerical solution matches the exact one.

2.4.2 Periodic boundary conditions

The tests reported in the previous section can be run also by imposing periodic boundary conditions on the top and at the bottom of the computational domain. In Figure 2.9 we show the discretization of the computational domain at time $t = 100.2$ for the initial conditions in (2.48) and in (2.49). We would like to underline that no distortion of the mesh elements appears even after a very long computational time, and as a direct consequence the time steps remain almost constant during the computation. As always in this type of test cases the volume conservation holds and the numerical solution is equal to the exact

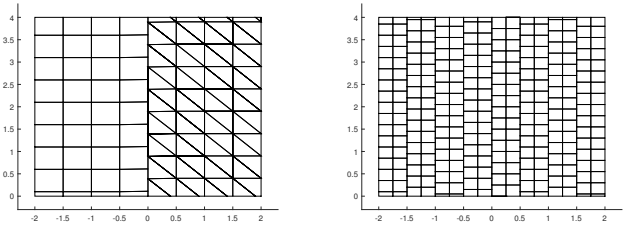


Figure 2.9: Slide lines with periodic boundary conditions. We report the final computational domain at time $t = 100.2$ corresponding to the initial condition in (2.48) on the left, and the one corresponding to the initial condition in (2.49) on the right. No distortion of the computational domain appears neither at the interfaces, and the numerical solution coincides with the exact one.

one up to machine precision.

2.4.3 Riemann problem

Let us now consider as initial condition a Riemann problem with a discontinuity in the water level

$$\mathbf{Q}(\mathbf{x}, 0) = \begin{cases} (1, 0, 0) & \text{if} \quad x \leq 0, \\ (0.5, 0, 0) & \text{if} \quad x > 0, \end{cases} \tag{2.51}$$

that originates a left-traveling rarefaction fan and a right-moving shock wave. We decided to move the mesh with an arbitrary mesh velocity function

$$\mathbf{V} = (\, 0,\ 0.5 \,\mathrm{floor}\,(2x)\,) \qquad -2 \leq x \leq 2,$$

in order to check the robustness of the algorithm also in the presence of discontinuities. We impose periodic boundary conditions on the top and on the bottom of the square, and wall boundary conditions on the left and on the right. The final discretization of the computational domain together with the comparison between the numerical and the exact solution are depicted in Figure 2.10 both for the first order accurate scheme (i.e. without the MUSCL-Hancock strategy for the reconstruction) and the second order accurate scheme.

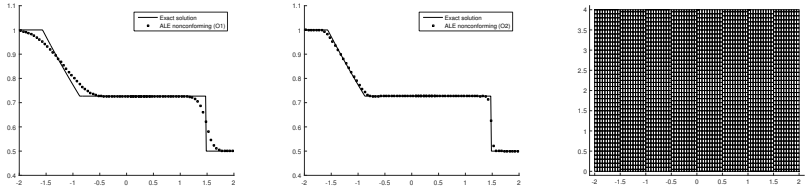


Figure 2.10: Riemann problem with an arbitrary mesh velocity. Taking $\alpha = 0.4$ in (2.28) the algorithm identifies 7 interfaces which are then handled in a nonconforming way. In the figure we report the final discretization of the computational domain, and the comparison between the exact solution and the numerical solutions obtained with our nonconforming method showing first order results (left), second order results (center) and the mesh at the final time (right).

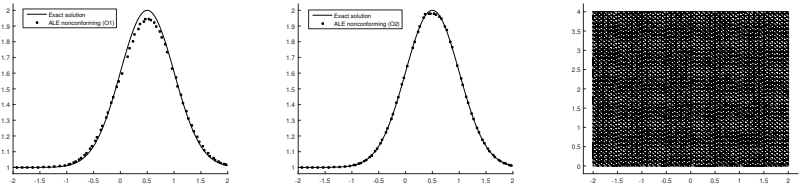


Figure 2.11: Comparison of the exact solution for the quantity c with the numerical solution obtained on moving nonconforming meshes. The results obtained with the first order algorithm are shown on the left, while those obtained with the second order MUSCL-Hancock method are presented in the center. The comparison is done at time $t = 0.5$ taking a cut of the profile of c corresponding to $y = 2$. On the right we show the discretization of the computational domain at time $t = 0.5$.

2.4.4 Convergence test

To verify the order of convergence of the proposed method we study the passive transport of a quantity c, that at time $t = 0$ is taken equal to a Gaussian profile and then will be passively transported in the direction of the fluid flow without changing its shape. The PDE system describing this situation is obtained from the standard shallow water equations (2.38) with the addition of the

Table 2.1: Numerical convergence results for the passive transport of a Gaussian profile on moving nonconforming meshes. The error norms refer to the variable c at time $t = 0.5$. On the left we report the result for the first order method (i.e. without using the MUSCL-Hancock reconstruction procedure) and on the right using the second order accurate scheme.

$\mathcal{O}1$				$\mathcal{O}2$			
mesh points	$h(\Omega(t_f))$	ε_{L_2}	$\mathcal{O}(L_2)$	mesh points	$h(\Omega(t_f))$	ε_{L_2}	$\mathcal{O}(L_2)$
12×12	1.95E-01	1.44E-01	-	12×12	1.95E-01	4.96E-02	-
24×24	9.78E-02	7.58E-02	0.93	24×24	9.78E-02	1.23E-02	2.02
40×40	5.88E-02	4.69E-02	0.94	40×40	5.88E-02	4.24E-03	2.10
80×80	2.95E-02	2.41E-02	0.97	80×80	2.95E-02	1.01E-03	2.09
120×120	1.98E-02	1.62E-02	0.99	120×120	1.98E-02	4.51E-04	2.01

concentration c of a passive tracer,

$$\mathbf{Q} = \begin{pmatrix} h \\ hu \\ hv \\ hc \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \\ huc \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \\ hvc \end{pmatrix}. \tag{2.52}$$

We fix the following initial condition

$$\mathbf{Q}(\mathbf{x}, 0) = \left(1, u, 0, 1 + \mathrm{e}^{\frac{-0.5\left(x^2 + (y - 0.5\,p)^2\right)}{0.5^2}}\right), \quad -2 \le x \le 2, \quad 0 \le y \le p, \tag{2.53}$$

where we use a fluid velocity of $u = 1$ and where we have taken the period $p = 4$. The mesh is moved with the velocity

$$\mathbf{V} = (0,\, 0.5\,\mathrm{floor}\,(x)) \quad -2 \le x \le 2, \tag{2.54}$$

according to the ALE framework, where the mesh velocity can be chosen arbitrarily. We prescribed periodic boundary conditions on the upper and lower side of the rectangular domain, and wall boundary conditions on the left and right sides. Since the exact solution is known ($\mathbf{Q}(\mathbf{x}, t) = \mathbf{Q}(\mathbf{x} - ut, 0)$) and it is smooth, we can verify the order of convergence of our method. In Table 2.1 we report the order of convergence of the basic first order Finite Volume method,

and of its second order extension that uses the MUSCL-Hancock strategy for the reconstruction procedure in space and time. Moreover, in Figure 2.11 we compare the numerical solution for the variable c with the profile of the exact solution and we show the mesh at the final time.

2.4.5 Steady vortex in equilibrium

To show that our method is also robust enough for vortex flows, we simulate the case of a steady vortex in equilibrium and we will compare the results obtained with our nonconforming method with a standard conforming algorithm (without any rezoning technique) looking at the differences after long simulation times.

Consider the shallow water in polar coordinates (2.46) and the following initial condition

$$h(r,\varphi,0)=1-\frac{1}{2g}\mathrm{e}^{-(r^2-1)},\quad u_r(r,\varphi,0)=0,\quad u_\varphi(r,\varphi,0)=r\mathrm{e}^{-\frac{1}{2}(r^2-1)},\quad (2.55)$$

which is a stationary solution, and so coincides with the exact solution at any time. We performe our test both with the Osher-type and the Rusanov-type fluxes and with a mesh made of triangles, quadrilaterals or both. The considered computational domain is $\Omega(r,\varphi) = [0.2, 2] \times [0, 2\pi]$ which is easily mapped into the annulus with radius $[0.2, 2]$. Indeed the choice of considering the shallow water equations in polar coordinates allows us to study the vortex over a rectangular domain with periodic boundary conditions (at $\varphi = 0$ and $\varphi = 2\pi$) instead of dealing with circles. At $r = 0.2$ and $r = 2$ we impose reflective boundary conditions. In particular using the polar coordinates the detected shear interfaces lie over straight lines and so they are perfectly handled by our algorithm. The images presented in this section are then obtained by mapping back our results to Cartesian coordinates, as shown in Figure 2.12.

First, Table 2.2 confirms the designed order of convergence of our algorithm in multiple situations: so primarily we can say that the mesh motion does not affect the standard order of convergence of the MUSCL-Hancock strategy and moreover this shows once again that the numerical flux computation, even at the nonconforming interfaces, is carried out correctly. The numerical solution at $t = 15$ is compared with the analytical one in Figure 2.13.

Then we compare the results with a standard conforming method. First, let us underline that when the velocity changes even within the same element the only way to overcome the mesh distortion would be to split the element itself. For this reason, where the velocity field changes smoothly and as a consequence

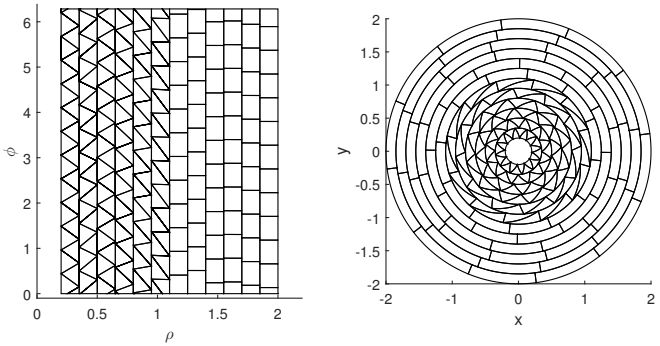


Figure 2.12: Domain discretization at time $t = 15$. On the left we report the grid in polar coordinates where the shear discontinuities lie over straight lines. On the right the corresponding grid in Cartesian coordinates.

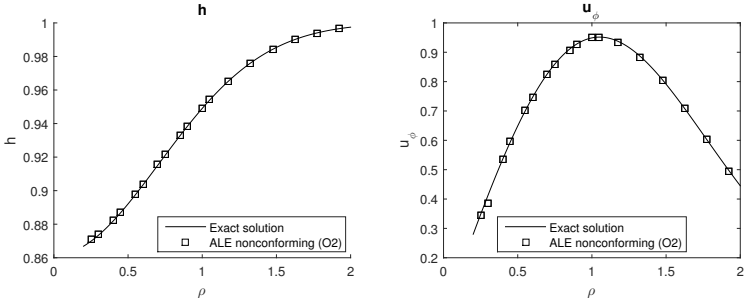


Figure 2.13: Comparison between analytical solution and second-order accurate numerical results for the water level h (left) and the tangential component of the velocity u_φ (right), with $\varphi = 2\pi$ and $r \in [0.2, 2]$.

the shear flow affects all the vertexes of the same element, at a certain time the mesh will become invalid even in the nonconforming case. This would not happen if the velocity field were uniform within each element, i.e. if each element moved all its vertexes with the same velocity, e.g. the velocity of the barycenter. The main difference between the new nonconforming algorithm and a conventional conforming method is the final time at which the computation

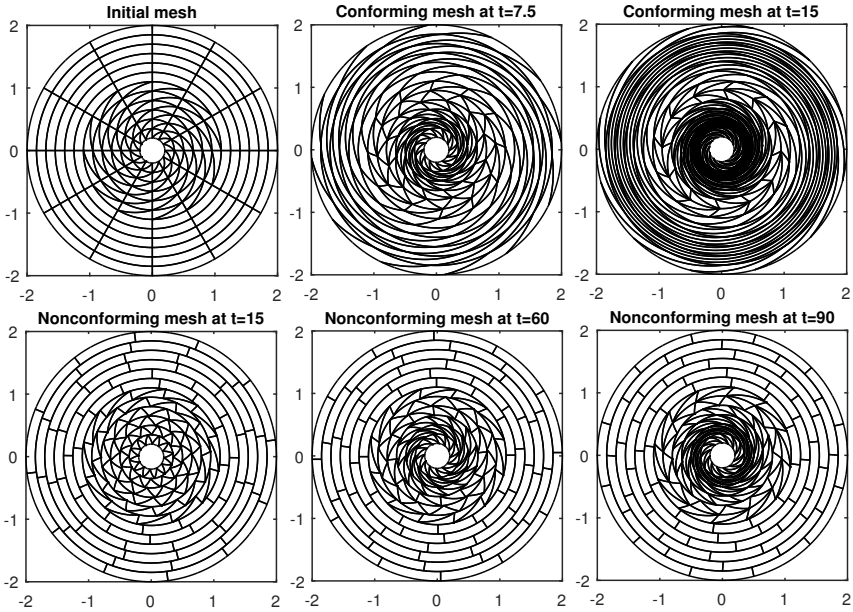


Figure 2.14: Steady vortex in equilibrium. We compared the behavior of a standard conforming algorithm (without any rezoning technique) and of our new nonconforming method. Using the conforming algorithm the elements are deformed in a very short time, the time step is heavily reduced and hence the computation is slower. On the contrary, the nonconforming slide lines introduced by our scheme are able to maintain a good shape of each element and an almost constant time step for a long computational time. Indeed only at time $t = 90$ some elements with $r \to 0$ are deformed because of the presence of shear *inside* the elements, which could be remedied only by subdividing the elements themselves or by removing them.

stops due to an invalid mesh, and the time step restriction that depends on the smallest encircle diameter of the elements.

In Table 2.3 we report the employed number of time steps and their dimension for different kinds of meshes and at different times. We remark that a larger value of Δt decreases the required number of time steps and in this way also the total amount of computational time. The last results of each group refer

Table 2.2: Numerical convergence results for the steady vortex in equilibrium using nonconforming meshes. In the left table we report the results obtained on a quadrilateral mesh using the Osher type flux. For the results on the right we have employed a triangular mesh and the Rusanov type flux. The error refers to the difference between the computed water level h and the exact one at time $t_f = 0.5$.

$\mathcal{O}2$, Osher flux, quadrilateral elements				$\mathcal{O}2$, Rusanov flux, triangular elements			
mesh points	$h(\Omega(t_f))$	ε_{L_2}	$\mathcal{O}(L_2)$	mesh points	$h(\Omega(t_f))$	ε_{L_2}	$\mathcal{O}(L_2)$
12×12	2.33E-01	1.36E-03	-	20×20	7.18E-02	5.97E-04	-
24×24	1.17E-01	3.42E-04	1.99	30×30	5.21E-02	2.54E-04	2.11
32×32	8.74E-02	1.94E-04	1.97	40×40	3.91E-02	1.43E-04	2.01
44×44	6.36E-02	1.03E-04	1.98	55×55	2.84E-02	7.76E-05	1.91
60×60	4.66E-02	5.57E-05	1.99	60×60	2.60E-02	6.58E-05	1.91

to the moment at which the algorithm breaks due to an invalid mesh: one can easily see that the nonconforming method is able to run almost eight times longer than a conventional ALE method on conforming grids. Finally, looking at Figure 2.14 one can appreciate that the conforming method destroys the mesh immediately and then breaks, whereas the new nonconforming algorithm maintains a high quality mesh for a very long time, even with a very coarse mesh.

2.5 Proof of concepts: general slide lines

All test problems shown before were limited to logically straight slide lines. However, there is no intrinsic limitation to logically straight slide lines in our algorithm, since the integral space-time conservation form (4.4) of the conservation law is valid for *arbitrary* closed space-time control volumes. This simple, elegant but at the same time very powerful formulation allows also to dynamically *add* and *remove* elements or to change their type during the simulation in a consistent manner that respects the GCL as well as local and global conservation. All these features are trivially built in *by construction*, due to the integral formulation on closed space-time control volumes. In Figure 2.15 we show examples of space-time control volumes that result when elements change type or when elements are dynamically added and removed during a simula-

Table 2.3: In this table we report the number of time steps n necessary to reach the time t and the dimension of the time step Δt at that time. We used three different meshes with N_E total number of elements (triangles or quadrilaterals). The results are obtained by applying a standard conforming method and our new nonconforming algorithm. Looking at the bold data one can see that with almost the same number of time steps one reaches a simulation time that is twice as large with the nonconforming algorithm compared to a classical conforming one. Besides the final simulation time that can be reached before obtaining an invalid mesh is almost 8 times larger.

$N_E \rightarrow$	216			264			300		
	conforming algorithm								
	t	n	Δt	t	n	Δt	t	n	Δt
	1	110	9.58E-03	1	180	5.40E-03	1	180	5.71E-03
	8	1163	4.13E-03	8	2180	2.52E-03	10	2071	3.11E-03
	12	2370	2.70E-03	12	4035	1.89E-03	15	4098	2.04E-03
stop at $\rightarrow$	15.3	**3773**	2.06E-03	15.5	**6072**	1.54E-03	17	5190	1.78E-03
	nonconforming algorithm								
	1	110	9.58E-03	1	180	5.82E-03	1	175	5.68E-03
	8	851	9.50E-03	8	1410	5.52E-03	10	1720	5.92E-03
	30	**3175**	9.30E-03	30	**6033**	4.06E-03	15	2565	5.94E-03
	60	7757	4.90E-03	60	15010	2.84E-03	80	15979	3.34E-03
stop at $\rightarrow$	119	26430	2.24E-03	129	35791	1.94E-03	132	36275	2.13E-03

tion. For logically non-straight slide lines, it is necessary to divide elements and nodes into masters and slaves, where the master elements maintain their number of nodes, while the slave elements must in general change their element type during the sliding process. Also note that master nodes are free to move anywhere, while slave nodes must slide along the master edges. Furthermore, small elements need to be removed if they lead to excessively small time steps due to the CFL condition.

We now repeat the same shallow water vortex test problem as described in the previous section, but using the PDE in Cartesian coordinates. This leads to logically non-straight slide lines. The comparison between the classical conforming moving mesh algorithm and our new nonconforming approach is

shown in Fig. 2.16 and Table 2.4. We observe the improved mesh quality and time step size compared to the classical conforming approach, in particular when the moving nonconforming mesh is combined with the removal of small elements. The obtained results look promising and justify further research in this direction in the future.

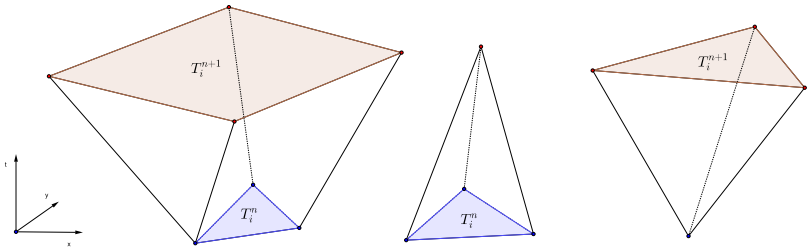


Figure 2.15: Dynamic change of element type (left), element removal (center) and element insertion (right) between time t^n and time t^{n+1}. Nodes and element T_i^n at time t^n are highlighted in blue, nodes and element T_i^{n+1} at time t^{n+1} are colored in red.

Table 2.4: Time step size for three different moving mesh algorithms. The main improvement is achieved when using a nonconforming algorithm combined with small element removal. This allows to maintain reasonable time steps also for longer simulation times.

	Time step size		
time	conforming	nonconforming	nonconforming + element removal
0.3	3.8E-3	3.2E-3	3.2E-2
0.6	3.6E-3	2.1E-3	2.1E-3
1.0	1.9E-3	9.0E-4	1.2E-3
1.3	5.8E-4	1.2E-4	1.4E-3
1.7	-	-	1.4E-3

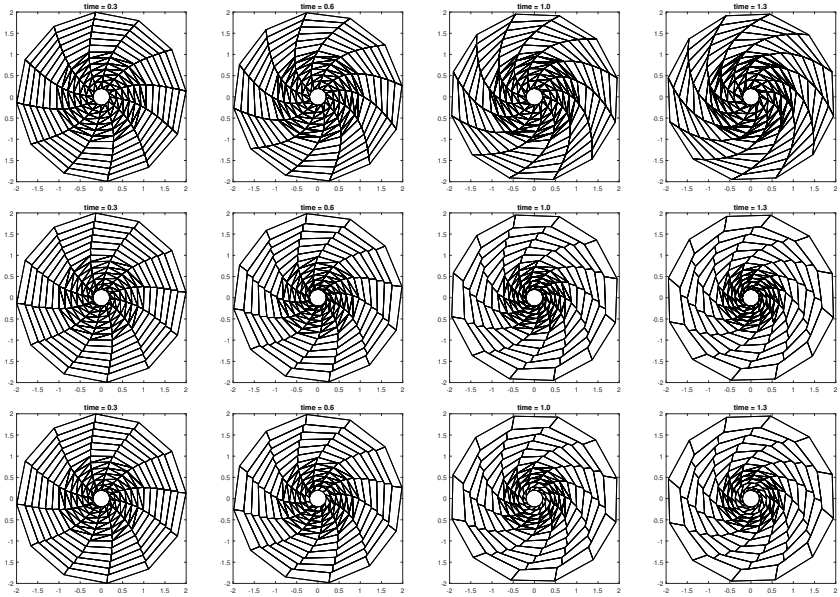


Figure 2.16: Isolated vortex in Cartesian coordinates. Classical conforming algorithm without any rezoning technique (top). Moving mesh obtained with the new *nonconforming* algorithm at different times (center) without small element removal. Moving *nonconforming* mesh with small element removal (bottom), which allows to control the time step size and to maintain a better mesh quality. The nonconforming algorithms used here use logically *non-straight* slide lines. The sliding edges are automatically detected based on the tangential velocity difference.

Well balanced path-conservative methods

This central part of the thesis is dedicated to the presentation of a completely new family of well balanced (WB) path-conservative schemes, able to maintain up to machine precision the equilibrium solutions of the studied system and to drastically increase the power of resolutions on small perturbations that may appear around those equilibria. To the best knowledge of the author this work is original for various reasons: it is the first time that the little dissipative Osher scheme is modified in order to be well balanced for nontrivial equilibria on moving meshes, and it is the first time that WB is coupled with ALE for the Euler equations with gravity; moreover the use of a well balanced Osher scheme joint with the Lagrangian framework allows, for the first time within a Finite Volume method, to maintain exactly even *moving* equilibria. This topic is covered in the next three chapters.

In Chapter 3, we concentrate on the one-dimensional case. In Section 3.1 we recall the basic notions on well balanced path-conservative schemes introduced by Dal Maso et al. [119] and Parés et al. [128] [37]. Then we have decided to detail our novel well balanced scheme in the particular case of the Euler equations of gas dynamics with gravity in polar coordinates, which are recovered in Section 3.2. Indeed, the method is based on very general and powerful funding concepts, but some computations differ from system to system and depend on the family of equilibria that has to be maintained, so in order to present all its specific features we have to introduce a concrete case. However, it is easy to reproduce the same reasoning for other systems following the presented line, as it is done for shallow water equations and a reduced compressible two-phase flow model in Chapters 4 and 5.

Moreover, the Euler equations with gravity represent a very interesting and highly challenging case of study. They allow to explore complex *astrophysical* phenomena as the rotating *Keplerian disk*: it is characterized by a family of steady equilibria between pressure gradient, centrifugal force and gravity force and by a high shear flow due to the differential rotation in the disk. In order to model with high accuracy and minimal dissipation small perturbations

around those equilibria for very long period of times we propose to use a well balanced method coupled with our nonconforming ALE framework. This coupling is presented in Section 4.1, together with a large set of numerical test cases in Section 4.2 that witnesses the power of resolution of our method both close and far away from the equilibria. Besides, a comparison with *not* well balanced schemes or *not* Lagrangian-type methods show the superiority of the proposed scheme with respect to the state of the art, demonstrated also by the comparison with the PLUTO code (a well established software targeting astrophysical simulations); we would like to emphasize the excellent quality of the results obtained in Chapter 4. Then we conclude by showing that the same coupling between ALE and WB can be applied to the shallow water equations, refer to Section 4.3.

In Chapter 5, we concentrate ourselves on a simple compressible two-phase model for complex free surface flows. The model is derived from the Baer Nunziato system and results to be more accurate than standard shallow water equations both for the initial stages of dambreaks, because it does not neglect vertical accelerations, and for breaking waves, because the free surface, given by the volume fraction, is not constraint to be necessarily a single value function. This three-equation model was first introduced by Dumbser in [65]. The novelties presented in this thesis consist first in a well balanced treatment of the source terms, in order to preserve exactly equilibria between air and water even in the presence of obstacles; and second in a *parallel* implementation exploiting a *GPU*-based platform that allows very high computational efficiency on fine meshes.

To conclude, we would like to focus the attention on the key idea for the construction of our well balanced scheme: the introduction of a path which directly exploits the known stationary solution (and so it is exact on it), and treats in a approximate way only the fluctuations around the equilibrium, see equation (4.40). The same idea of using an approximate technique only on the fluctuations appears even in the reconstruction process, see equation (3.66)-(3.68). This simple idea can guide to the understanding of the method.

3 Well balancing: one dimensional case

3.1 General framework

For the numerical approximation of the a one dimensional system, the spatial domain is discretized by N fixed intervals $T_i^n = I_i = [x_{i-1/2}, x_{i+1/2}]$ of regular size $\Delta x = x_{i+1/2} - x_{i-1/2}$, $i = 1, \dots, N$. As a standard Finite Volume scheme, a path-conservative scheme is obtained first by integrating the governing PDE (1.1) in space and time obtaining (1.3). After that, instead of introducing the time-averaged flux as in (1.6), following [128] and [41] we write the scheme as follows

$$\begin{aligned}\frac{d\mathbf{Q}_i}{dt}(t) = &- \frac{\Delta t}{\Delta x}\left(\mathbf{D}^+_{i-\frac{1}{2}}\left(\mathbf{q}^-_{i-\frac{1}{2}}(t), \mathbf{q}^+_{i-\frac{1}{2}}(t)\right) + \mathbf{D}^-_{i+\frac{1}{2}}\left(\mathbf{q}^-_{i+\frac{1}{2}}(t), \mathbf{q}^+_{i+\frac{1}{2}}(t)\right)\right) \\ &- \frac{\Delta t}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial}{\partial x}\mathbf{f}\left(\mathbf{q}_i(x,t)\right) dr \\ &- \frac{\Delta t}{\Delta x}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{B}_1\left(\mathbf{q}_i(x,t)\right)\frac{\partial}{\partial x}\left(\mathbf{q}_i(x,t)\right) dr.\end{aligned} \tag{3.1}$$

In the scheme, $\mathbf{q}_i(x,t)$ is the approximation of the conserved variables inside cell I_i at time t, computed via a reconstruction operator from the conserved variables in I_i and its neighbors, while $\mathbf{q}^+_{i-\frac{1}{2}}(t) = \mathbf{q}_i(x_{i-1/2}, t)$ and $\mathbf{q}^-_{i+\frac{1}{2}}(t) = \mathbf{q}_i(x_{i+1/2}, t)$ denote the evaluation of $\mathbf{q}_i(x,t)$ at the left and right boundaries of cell I_i. According to the above reference, $\mathbf{D}^\pm_{i+\frac{1}{2}}$ is defined as follows

$$\begin{aligned}\mathbf{D}^\pm_{i+\frac{1}{2}}\left(\mathbf{q}^-_{i+\frac{1}{2}}, \mathbf{q}^+_{i+\frac{1}{2}}\right) = \frac{1}{2}\Bigg(&\mathbf{f}(\mathbf{q}^+_{i+\frac{1}{2}}) - \mathbf{f}(\mathbf{q}^-_{i+\frac{1}{2}}) + \\ &\mathcal{B}_{i+\frac{1}{2}}\left(\mathbf{q}^+_{i+\frac{1}{2}} - \mathbf{q}^-_{i+\frac{1}{2}}\right) \pm \mathcal{V}_{i+\frac{1}{2}}\left(\mathbf{q}^+_{i+\frac{1}{2}} - \mathbf{q}^-_{i+\frac{1}{2}}\right)\Bigg),\end{aligned} \tag{3.2}$$

where $\mathbf{f}(\mathbf{q})$ is the physical flux, $\mathcal{B}_{i+\frac{1}{2}}\left(\mathbf{q}^+_{i+\frac{1}{2}} - \mathbf{q}^-_{i+\frac{1}{2}}\right)$ is the discretization of the non-conservative terms and $\mathcal{V}_{i+\frac{1}{2}}\left(\mathbf{q}^+_{i+\frac{1}{2}} - \mathbf{q}^-_{i+\frac{1}{2}}\right)$ is the viscosity term, that characterizes the method. In (3.2), the dependency on t has been dropped for simplicity. $\mathcal{B}_{i+\frac{1}{2}}\left(\mathbf{q}^+_{i+\frac{1}{2}} - \mathbf{q}^-_{i+\frac{1}{2}}\right)$ and $\mathcal{V}_{i+\frac{1}{2}}\left(\mathbf{q}^+_{i+\frac{1}{2}} - \mathbf{q}^-_{i+\frac{1}{2}}\right)$ have to be defined in terms of a family of paths $\Phi(s; \mathbf{q}^-_{i+\frac{1}{2}}, \mathbf{q}^+_{i+\frac{1}{2}})$, $s \in [0,1]$. In this work the family of paths will be chosen so that a class of stationary solutions could be preserved.

The scheme as written in (3.1) is similar to the wave-propagation formulation proposed by LeVeque in [106] for standard conservative systems. We refer the reader interested in the complete theoretical framework of path-conservative schemes to the very detailed paper of Parés [128], especially to Chapter 3 where the classical Finite Volume scheme is first rewritten in terms of distributions and then generalized to non-conservatives products. In particular it is shown that the terms $\mathbf{D}^{\pm}_{i+\frac{1}{2}}$ allow to decompose the punctual masses placed at the interface in such a way that $\mathbf{D}^-_{i+\frac{1}{2}}$ contributes to cell I_i and $\mathbf{D}^+_{i+\frac{1}{2}}$ to cell I_{i+1}, leading to the key definition (see equation (3.7)) for resolving the discontinuities at the interface. Moreover we refer to [55] for the definition of the non-conservative products as Borel measure.

For the sake of clarity let us now recall some definitions, taken from the above references.

Definition 3.1.1 (Family of paths)**.** *A family of paths in $\Omega \subset \mathbb{R}^d$ is a locally Lipschitz map*

$$\Phi : [0,1] \times \Omega \times \Omega \mapsto \Omega, \tag{3.3}$$

such that

i. $\Phi(0; \mathbf{Q}_L, \mathbf{Q}_R) = \mathbf{Q}_L$, $\Phi(1; \mathbf{Q}_L, \mathbf{Q}_R) = \mathbf{Q}_R$, $\Phi(s; \mathbf{Q}, \mathbf{Q}) = \mathbf{Q}$, *for any* $\mathbf{Q}_L$, $\mathbf{Q}_R$, $\mathbf{Q} \in \Omega$;

ii. for every arbitrary bounded set $\mathcal{O} \subset \Omega$, there exists a constant k such that

$$\left| \frac{\partial \Phi}{\partial s}(s; \mathbf{Q}_L, \mathbf{Q}_R) \right| \leq k \left| \mathbf{Q}_R - \mathbf{Q}_L \right|, \tag{3.4}$$

for any $\mathbf{Q}_L, \mathbf{Q}_R \in \mathcal{O}$ and almost every $s \in [0,1]$;

iii. for every bounded set $\mathcal{O} \subset \Omega$, there exists a constant K such that

$$\left| \frac{\partial \Phi}{\partial s}(s; \mathbf{Q}^1_L, \mathbf{Q}^1_R) - \frac{\partial \Phi}{\partial s}(s; \mathbf{Q}^2_L, \mathbf{Q}^2_R) \right| \leq K\left(\left| \mathbf{Q}^1_L - \mathbf{Q}^2_L \right| + \left| \mathbf{Q}^1_R - \mathbf{Q}^2_R \right| \right), \tag{3.5}$$

for any $\mathbf{Q}_L^1$, $\mathbf{Q}_R^1$, $\mathbf{Q}_L^2$, $\mathbf{Q}_R^2 \in \mathcal{O}$ *and almost every* $s \in [0,1]$.

Definition 3.1.2 (Path-conservative)**.** *Given a family of path* Φ*, a numerical scheme is said to be* $\Phi-$*conservative or path-conservative if it can be written under the form* (3.1) *with*

$$\mathbf{D}^{\pm}(\mathbf{Q},\mathbf{Q}) = \mathbf{0} \quad \forall \mathbf{Q} \in \Omega, \tag{3.6}$$

and

$$\mathbf{D}^{-}(\mathbf{Q}_L,\mathbf{Q}_R)+\mathbf{D}^{+}(\mathbf{Q}_L,\mathbf{Q}_R) = \int_0^1 \mathbf{A}(\Phi(s;\mathbf{Q}_L,\mathbf{Q}_R))\frac{\partial \Phi}{\partial s}(s;\mathbf{Q}_L,\mathbf{Q}_R)ds, \tag{3.7}$$

for every $\mathbf{Q}_L$, $\mathbf{Q}_R \in \Omega$.

Note that, in the case of a one dimensional system i.e. (1.1) with $\mathbf{g}, \mathbf{h} = \mathbf{0}$ and $\mathbf{B}_{2,3} = \mathbf{0}$, we can rewrite equation (3.7) as

$$\mathbf{D}^{-}(\mathbf{Q}_L,\mathbf{Q}_R)+\mathbf{D}^{+}(\mathbf{Q}_L,\mathbf{Q}_R) = \mathbf{f}(\mathbf{Q}_R) - \mathbf{f}(\mathbf{Q}_L) + \mathcal{B}_{LR}(\mathbf{Q}_R - \mathbf{Q}_L), \tag{3.8}$$

where

$$\mathcal{B}_{LR}(\mathbf{Q}_R - \mathbf{Q}_L) = \int_0^1 \mathbf{B}_1(\Phi(s;\mathbf{Q}_L,\mathbf{Q}_R))\frac{\partial \Phi}{\partial s}(s;\mathbf{Q}_L,\mathbf{Q}_R)ds. \tag{3.9}$$

Definition 3.1.3 (Well balance)**.** *Let* Γ *be the set of all the integral curves gamma of a linearly degenerate field of* $\mathbf{A}(\mathbf{Q})$ *such that the corresponding eigenvalues vanishes on* Γ*. Given a curve* $\gamma \in \Gamma$*, a numerical scheme*

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n + \frac{\Delta t}{\Delta x}\mathbf{H}(\mathbf{Q}_{i-q}^n, \dots, \mathbf{Q}_{i+p}^n) \tag{3.10}$$

is said to be exactly well balanced for γ *if, given any* $\mathcal{C}^1$ *function* $x \in (\alpha,\beta) \subset \mathbb{R} \mapsto \mathbf{Q}(x) \in \Omega$ *such that*

$$\mathbf{Q}(x) \in \gamma, \quad \forall x \in (\alpha,\beta), \tag{3.11}$$

and $p+q+1$ *points in* (α,β) *such that* $x_{-q} \leq \dots \leq x_p$*, then*

$$H(\mathbf{Q}(x_{-q}), \dots, \mathbf{Q}(x_p)) = 0. \tag{3.12}$$

Proposition 3.1.4. *A first order path-conservative scheme is exactly well balanced for a curve* $\gamma \in \Gamma$ *if and only if*

$$D^{\pm}(\mathbf{Q}_L,\mathbf{Q}_R) = 0, \quad \forall\ \mathbf{Q}_L,\mathbf{Q}_R \in \gamma. \tag{3.13}$$

Proposition 3.1.5. *A reconstruction operator $\mathcal{P}$ is said to be exactly well balanced for a curve $\gamma \in \Gamma$ if, given a sequence $\{\mathbf{Q}_i\} \in \gamma$, it satisfies*

$$\mathcal{P}(x, \mathbf{Q}_{i-l}, \dots, \mathbf{Q}_{i+r}) \in \gamma \quad \forall x \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}], \tag{3.14}$$

for every i.

Proposition 3.1.6. *The numerical scheme in* (3.1) *is exactly well balanced for a curve $\gamma \in \Gamma$ if both the underlying first order scheme and the reconstruction operator are exactly well-balanced for γ.*

This part was to introduce the basic notions on well balanced path-conservative schemes for general systems. From now on, we will focus on a specific set of equations and on a family of equilibria. However, we would like to underline once again that all the methods we will propose, for example in Section 3.3 in the particular case of the Euler equations with gravity in polar coordinates, are based on very general and powerful concepts so that they can be easily extended to other systems. This will be shown by studying both the shallow water equations in Section 4.3, and a two-phase model in Section 5.2, following always the same key ideas.

3.2 Euler equations with gravity

The Euler equations of compressible gas dynamics with an externally given gravitational field allow to study problems in computational astrophysics connected with the rotation of gas clouds around a central object. In particular, we are interested in situations close to equilibrium solutions and affected by strong shear flows. What we propose is to use the nonconforming ALE technique presented in the previous chapter in order to maintain a good quality mesh despite the differential rotation, and a well balanced method in order to preserve exactly the steady states and to reduce the numerical errors close to them. But in order to apply these two techniques *coupled together* we have first to recover a suitable formulation of the equations.

In this section we present the classical Cartesian form of the Euler equations with gravity (Section 3.2.1), than we rewrite them in polar coordinates (Section 3.2.2) so that the nonconforming ALE can be used in the more simple case of straight slip lines. And finally, in Section 3.2.3 we eliminate the algebraic source terms, which would make the equilibria preservation very difficult, by introducing some non-conservative products. This last non-conservative formulation (3.23) represents a very useful framework for applying our *new* well balanced techniques.

3.2.1 Cartesian coordinates

The Euler equations with gravity in two space dimensions represent a strongly hyperbolic system of equations that can be cast in the form of balance laws, by taking in (1.1)

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u_x \\ \rho u_y \\ \rho E \end{pmatrix}, \quad \mathbf{f}(\mathbf{Q}) = \begin{pmatrix} \rho u_x \\ \rho u_x^2 + p \\ \rho u_x u_y \\ u_x(\rho E + p) \end{pmatrix}, \quad \mathbf{g}(\mathbf{Q}) = \begin{pmatrix} \rho u_y \\ \rho u_x u_y \\ \rho u_y^2 + p \\ u_y(\rho E + p) \end{pmatrix},$$

$$\mathbf{B}(\mathbf{Q}) = 0, \qquad \mathbf{S}(\mathbf{Q}) = \begin{pmatrix} 0 \\ -\cos\varphi\, \rho\, \frac{G\, m_s}{r^2} \\ -\sin\varphi\, \rho\, \frac{G\, m_s}{r^2} \\ -(u_x \cos\varphi + u_y \sin\varphi)\rho \frac{G m_s}{r^2} \end{pmatrix}. \tag{3.15}$$

Here ρ is the density, u_x and u_y are respectively the velocities along the x and y directions, $r = \sqrt{x^2 + y^2}$, $\varphi = \arctan(y/x)$, E is the specific total energy (excluding the gravitational energy), m_s is the mass of the central object, G is the gravitational constant and the pressure p is given by

$$p = (\gamma - 1)\left(\rho E - \frac{1}{2}\rho\left(u_x^2 + u_y^2\right)\right), \quad \gamma = \frac{c_p}{c_v} > 1, \tag{3.16}$$

where γ is the ratio of the specific heats at constant pressure and at constant volume, and which is supposed to be constant.

3.2.2 Polar coordinates

Now we are interested in studying *rotational phenomena* affected by sheared vortex flows, so we decide to rewrite the Euler equations in polar coordinates (r, φ). We follow the same procedure of Section 2.3.2. To shorten the notation, from now on when referring to the Euler equations in polar coordinates, we denote the radial velocity u_r by u, and the angular velocity u_φ by v. The

resulting hyperbolic system still takes the form (1.1) with

$$\mathbf{Q} = \begin{pmatrix} r\rho \\ r\rho u \\ r\rho v \\ r\rho E \\ r \end{pmatrix}, \quad \mathbf{f}(\mathbf{Q}) = \begin{pmatrix} r\rho u \\ r\rho u^2 + rp \\ r\rho uv \\ ru(\rho E + p) \\ 0 \end{pmatrix}, \quad \mathbf{g}(\mathbf{Q}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E + p) \\ 0 \end{pmatrix},$$
$$\mathbf{B}(\mathbf{Q}) = 0, \qquad \mathbf{S}(\mathbf{Q}) = \begin{pmatrix} 0 \\ -\rho\frac{Gm_s}{r} + p + \rho v^2 \\ -\rho uv \\ -\rho u\frac{Gm_s}{r} \\ 0 \end{pmatrix}. \tag{3.17}$$

Note that the system is written in terms of conserved variables, which is made possible by the insertion of an additional trivial equation as in (2.44)

$$\frac{\partial r}{\partial t} = 0, \tag{3.18}$$

which implies that the radius r is both a coordinate and a conserved variable.

3.2.3 Non-conservative formulation and equilibrium solutions

The aim of our work is to construct a Finite Volume scheme that is second order accurate in general situations, and, at the same time, can solve exactly (i.e. up to machine precision) a class of stationary solutions given by

$$\rho = \rho(r), \qquad u = 0, \qquad \frac{\partial v}{\partial \varphi} = 0. \tag{3.19}$$

Looking at the second equation in (3.17) and at the equilibrium constraints in (3.19), we notice that equilibria should balance the pressure and gravitational forces. More precisely

$$\frac{\partial rp}{\partial r} = -\rho\left(\frac{Gm_s}{r} - v^2\right) + p. \tag{3.20}$$

This relation has to be achieved also at the discrete level in order to preserve these stationary solutions. In standard Finite Volume schemes, fluxes and

sources are typically discretized in different ways and therefore, the balancing between them is usually lost. In order to construct a numerical scheme that exactly preserves those stationary solutions, here we first rewrite the equations in the following way, where both, pressure and gravitational forces (3.20) are treated as non-conservative terms. Thus, by exploiting some trivial equalities as

$$\frac{\partial rp}{\partial r} = p + r\frac{\partial p}{\partial r} \quad \text{and} \quad \frac{\partial r}{\partial r} = 1, \tag{3.21}$$

the forces in (3.20) can be rearranged as

$$r\frac{\partial p}{\partial r} + \left(\rho\frac{Gm_s}{r} - \rho v^2\right)\frac{\partial r}{\partial r} = 0, \tag{3.22}$$

and finally the Euler equations with gravity in polar coordinates can be cast in form (1.1) with non trivial non-conservative terms and with zero algebraic source term as

$$\mathbf{Q} = \begin{pmatrix} r\rho \\ r\rho u \\ r\rho v \\ r\rho E \\ r \end{pmatrix}, \quad \mathbf{f}(\mathbf{Q}) = \begin{pmatrix} r\rho u \\ r\rho u^2 \\ r\rho uv \\ ru(\rho E + p) \\ 0 \end{pmatrix}, \quad \mathbf{g}(\mathbf{Q}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E + p) \\ 0 \end{pmatrix},$$

$$\mathbf{S}(\mathbf{Q}) = \mathbf{0}, \qquad \mathbf{B}(\mathbf{Q})\cdot\nabla\mathbf{Q} = \begin{pmatrix} 0 \\ r\frac{\partial p}{\partial r} + \left(\rho\frac{Gm_s}{r} - \rho v^2\right)\frac{\partial r}{\partial r} \\ (\rho uv)\,\frac{\partial r}{\partial r} \\ \rho u\frac{Gm_s}{r}\frac{\partial r}{\partial r} \\ 0 \end{pmatrix}, \tag{3.23}$$

i.e.

$$\mathbf{B}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ r\frac{\partial p}{\partial q_1} & r\frac{\partial p}{\partial q_2} & r\frac{\partial p}{\partial q_3} & r\frac{\partial p}{\partial q_4} & r\frac{\partial p}{\partial q_5} + \rho\frac{Gm_s}{r} - \rho v^2 \\ 0 & 0 & 0 & 0 & \rho uv \\ 0 & 0 & 0 & 0 & \rho u\frac{Gm_s}{r} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{3.24}$$

$$\mathbf{B}_2 = \mathbf{0},$$

where q_i, $i = 1, \ldots, 5$ denotes the i-th component of vector $\mathbf{Q}$. Notice that it is possible to write the source terms as non-conservative products thanks to the introduction of the coordinate r also as conserved variables (see the added equation in (3.18)), which is the typical strategy adopted in [44, 87–90].

3.3 WB for the Euler equations with gravity

In this section we focus on the one dimensional version of the previous system (3.23), i.e. $\mathbf{g}$ and $\mathbf{B}_2$ are not considered, and we achieve an exact balancing in the radial direction r (we use r instead of x to indicate the spatial domain). Then in Section 4.1, we will extend the method to two space dimensions and moving nonconforming meshes. In both cases the key point of our new numerical method is the discretization of the terms in (3.22).

The rest of this section is organized as follows: we start by proposing two different first order well balanced schemes, the first one is named as Osher-Romberg scheme, and the second one is a well balanced HLL scheme. Next we propose a second order scheme constructed using the previous first order schemes in combination with a second order well balanced reconstruction operator.

3.3.1 First order well balanced schemes

Let us remark first, that the scheme (3.1) reduces to

$$\frac{d\mathbf{Q}_i}{dt}(t) = -\frac{\Delta t}{\Delta r}\left(\mathbf{D}^+_{i-\frac{1}{2}}\left(\mathbf{q}^-_{i-\frac{1}{2}}(t), \mathbf{q}^+_{i-\frac{1}{2}}(t)\right) + \mathbf{D}^-_{i+\frac{1}{2}}\left(\mathbf{q}^-_{i+\frac{1}{2}}(t), \mathbf{q}^+_{i+\frac{1}{2}}(t)\right)\right), \tag{3.25}$$

if $\mathbf{q}_i(r,t) = \mathbf{Q}_i(t)$ is constant within each cell, for every time t and coincides with the cell average $\mathbf{Q}_i(t)$. The time derivative is discretized by the first order explicit Euler method. Thus, the resulting scheme will be first order accurate in space and time. Moreover, $\mathbf{q}^-_{i+\frac{1}{2}} = \mathbf{q}_i = \mathbf{Q}_i$ and $\mathbf{q}^+_{i+\frac{1}{2}} = \mathbf{q}_{i+1} = \mathbf{Q}_{i+1}$.

Therefore, to determine the numerical scheme we should define $\mathcal{B}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ and $\mathcal{V}_{i+\frac{1}{2}}(\mathbf{q}_{i+1} - \mathbf{q}_i)$. In order to define $\mathcal{B}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$, a family of paths should be prescribed, so that the resulting scheme is well balanced for (3.19)-(3.20). Note that if the standard segment path is prescribed, that is

$$\Phi(s; \mathbf{q}_i, \mathbf{q}_{i+1}) = \mathbf{q}_i + s(\mathbf{q}_{i+1} - \mathbf{q}_i), \tag{3.26}$$

then, the resulting scheme is not well balanced for our set of stationary solutions.

Here we propose the following *family of paths*. Let $\Phi^E(s, \mathbf{Q}_i^E, \mathbf{Q}_{i+1}^E)$ be a reparametrization of a stationary solution given by (3.19)-(3.20) that connects the state $\mathbf{Q}_i^E$ with $\mathbf{Q}_{i+1}^E$, where $\mathbf{Q}_i^E$ is the cell average of the given stationary solution in the cell I_i. Note that in the case of first and second order schemes $\mathbf{Q}_i^E$ could be approximated by the evaluation of the stationary solution at the center of the cell. Then we define $\Phi(s; \mathbf{q}_i, \mathbf{q}_{i+1})$ as follows

$$\Phi(s; \mathbf{q}_i, \mathbf{q}_{i+1}) = \Phi^E(s; \mathbf{Q}_i^E, \mathbf{Q}_{i+1}^E) + \Phi^f(s; \mathbf{q}_i^f, \mathbf{q}_{i+1}^f), \tag{3.27}$$

where $\mathbf{q}_i^f = \mathbf{q}_i - \mathbf{Q}_i^E$ and $\mathbf{q}_{i+1}^f = \mathbf{q}_{i+1} - \mathbf{Q}_{i+1}^E$ and

$$\Phi^f(s; \mathbf{q}_i^f, \mathbf{q}_{i+1}^f) = \mathbf{q}_i^f + s(\mathbf{q}_{i+1}^f - \mathbf{q}_i^f). \tag{3.28}$$

That is, Φ^f is a segment path on the *fluctuations* with respect to a given stationary solution. With this choice, it is clear that if $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$ lie on the same stationary solution satisfying (3.19)-(3.20), then $\mathbf{q}_i^f = \mathbf{q}_{i+1}^f = \mathbf{0}$ and Φ reduces to Φ^E. In such situations we have that $\mathbf{f}(\mathbf{q}_{i+1}) = \mathbf{f}(\mathbf{q}_i) = \mathbf{0}$ and

$$\mathcal{B}_{i+\frac{1}{2}}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \int_0^1 \mathbf{B}_1(\Phi^E(s, \mathbf{q}_i, \mathbf{q}_{i+1})) \frac{\partial \Phi^E}{\partial s}(s; \mathbf{q}_i, \mathbf{q}_{i+1}) ds = \mathbf{0}. \tag{3.29}$$

Therefore

$$\mathbf{f}(\mathbf{q}_{i+1}) - \mathbf{f}(\mathbf{q}_i) + \mathcal{B}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \mathbf{0}. \tag{3.30}$$

For the sake of simplicity, in the following we will use the notation $\Phi(s)$ instead of $\Phi(s; \mathbf{q}_i, \mathbf{q}_{i+1})$ when there is no confusion.

Let us now define $\mathcal{B}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ in the general case, where $\mathbf{q}_{i+1}$ and $\mathbf{q}_i$ do not lie on a stationary solution. In this case we have that

$$\mathcal{B}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \left(b_1^{i+1/2} \; b_2^{i+1/2} \; b_3^{i+1/2} \; b_4^{i+1/2} \; b_5^{i+1/2} \right)^T. \tag{3.31}$$

It is clear from the definition of $\mathbf{B}_1$ that

$$b_1^{i+1/2} = b_5^{i+1/2} = 0, \tag{3.32}$$

$$b_2^{i+1/2} = \int_0^1 \Phi_r(s) \frac{\partial \Phi_p}{\partial s}(s) + \Phi_{(r\rho)}(s) \Phi_{\zeta_r}(s) \frac{\partial \Phi_r}{\partial s}(s) ds, \tag{3.33}$$

where $\Phi_r(s) = \Phi_r(s; r_i, r_{i+1}) = r_i + s(r_{i+1} - r_i)$, $\Phi_p(s) = \Phi_p^E(s) + \Phi_p^f(s)$, $\Phi_{(r\rho)}(s)(s) = \Phi_{(r\rho)}^E(s) + \Phi_{(r\rho)}^f(s)$ and, finally, $\Phi_{\zeta_r}(s) = \Phi_{\zeta_r}^E(s) + \Phi_{\zeta_r}^f(s)$ where

$$\zeta_r(r) = \left(\frac{G m_s}{r^2} - \frac{v^2}{r} \right), \quad \text{with} \quad \zeta(r) = \int \zeta_r(r) dr. \tag{3.34}$$

Taking into account that

$$\int_0^1 \Phi_r(s)\frac{\partial \Phi_p^E}{\partial s}(s) + \Phi_{(r\rho)}^E(s)\Phi_{\zeta_r}^E(s)\frac{\partial \Phi_r}{\partial s}(s)ds = 0, \tag{3.35}$$

$b_2^{i+1/2}$ can be rewritten as follows:

$$\begin{aligned} b_2^{i+1/2} &= \int_0^1 \Phi_r(s)\frac{\partial \Phi_p^f}{\partial s}(s)ds \\ &\quad + \int_0^1 \Big(\Phi_{(r\rho)}^E(s)\Phi_{\zeta_r}^f(s) + \Phi_{(r\rho)}^f(s)\Phi_{\zeta_r}(s)\Big)\frac{\partial \Phi_r}{\partial s}(s)ds. \end{aligned} \tag{3.36}$$

Note that, $\frac{\partial \Phi_p^f}{\partial s}(s) = p_{i+1}^f - p_i^f$ and $\frac{\partial \Phi_r}{\partial s}(s) = r_{i+1} - r_i = \Delta r_{i+1/2}$. Observe that in uniform meshes $\Delta r_{i+1/2} = \Delta r$. With the previous notation $b_2^{i+1/2}$ reduces to

$$\begin{aligned} b_2^{i+1/2} &= r_{i+1/2}\Delta p_{i+1/2}^f \\ &\quad + \left(\int_0^1 \Big(\Phi_{(r\rho)}^E(s)\Phi_{\zeta_r}^f(s) + \Phi_{(r\rho)}^f(s)\Phi_{\zeta_r}(s)\Big)\, ds\right)\Delta r_{i+1/2}, \end{aligned} \tag{3.37}$$

where $r_{i+1/2} = \frac{r_i + r_{i+1}}{2}$ and $\Delta p_{i+1/2}^f = p_{i+1}^f - p_i^f$. In general, the integral term could be difficult to compute, therefore we propose to use a numerical quadrature formula. Here the mid-point rule is used. In this case, we define $b_2^{i+1/2}$ as follows:

$$\begin{aligned} b_2^{i+1/2} &= \Big((r\rho)_{i+1/2}^E(\zeta_r)_{i+1/2}^f + (r\rho)_{i+1/2}^f(\zeta_r)_{i+1/2}\Big)\Delta r_{i+1/2} \\ &\quad + r_{i+1/2}\Delta p_{i+1/2}^f, \end{aligned} \tag{3.38}$$

where

$$(r\rho)_{i+1/2}^E = \Phi_{(r\rho)}^E(1/2),\ (\zeta_r)_{i+1/2}^f = \frac{(\zeta_r^f)_i + (\zeta_r^f)_{i+1}}{2}, \tag{3.39}$$

$$(r\rho)_{i+1/2}^f = \frac{(r\rho)_i^f + (r\rho)_{i+1}^f}{2},\ \text{and}\ \ (\zeta_r)_{i+1/2} = \Phi_{\zeta_r}(1/2). \tag{3.40}$$

It is clear from the definition that $b_2^{i+1/2} = 0$ if $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$ lie on the same stationary solution as $\Delta p_{i+1/2}^f = 0$, $(r\rho)_{i+1/2}^f = 0$ and $(\zeta_r)_{i+1/2}^f = 0$. Finally,

terms $b_3^{i+1/2}$ and $b_4^{i+1/2}$ could be approximated in the same way. Nevertheless, as those terms explicitly depend on u and we are interested in preserving equilibria with $u = 0$, a simpler approach can be used. Thus, $b_3^{i+1/2}$ is defined as

$$b_3^{i+1/2} = \frac{(r\rho u)_{i+1/2}}{r_{i+1/2}} v_{i+1/2} \Delta r_{i+1/2}, \tag{3.41}$$

where

$$(r\rho u)_{i+1/2} = \frac{(r\rho u)_i + (r\rho u)_{i+1}}{2}, \; v_{i+1/2} = \frac{v_i + v_{i+1}}{2}, \tag{3.42}$$

and $b_4^{i+1/2}$ as

$$b_4^{i+1/2} = (r\rho u)_{i+1/2} \frac{G m_s}{r_{i+1/2}^2} \Delta r_{i+1/2}. \tag{3.43}$$

Note that both terms vanish when $u = 0$.

As pointed in [128] and in Proposition 3.1.4, a sufficient condition for a first order path-conservative scheme to be well balanced is that $\mathbf{D}^{\pm}_{i+1/2}(\mathbf{q}_i, \mathbf{q}_{i+1}) = \mathbf{0}$, if $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$ lie on the same stationary solution. Therefore, with the previous choice of paths, $\mathbf{D}^{\pm}_{i+1/2} = 0$ if $\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \mathbf{0}$. In the next paragraph we are going to present two different schemes defined in terms of two different viscosity, both of them verifying that $\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = 0$ for stationary solutions (3.19)-(3.20).

3.3.1.1 Osher-Romberg scheme

A path-conservative Osher-type scheme following [42,74,75] can be cast in form (3.2) with $\mathcal{V}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ being defined as follows

$$\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \int_0^1 |\mathbf{A}(\Phi(s))| \, \partial_s \Phi(s) ds, \quad 0 \le s \le 1, \tag{3.44}$$

with $|\mathbf{A}| = \mathbf{R}|\mathbf{\Lambda}|\mathbf{R}^{-1}$ being the usual definition of the matrix absolute value operator given in terms of the right eigenvector matrix $\mathbf{R}$, its inverse $\mathbf{R}^{-1}$ and the diagonal matrix of the absolute values of the eigenvalues $|\mathbf{\Lambda}| = \text{diag}\,(|\lambda_1|, |\lambda_2|, ..., |\lambda_\nu|)$. For the numerical approximation of the viscosity matrix, first we notice that it can be written as

$$\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \int_0^1 \text{sign}\,(\mathbf{A}(\Phi(s))) \, \mathbf{A}(\Phi(s)) \, \partial_s \Phi(s) ds, \tag{3.45}$$

with $\text{sign}(\mathbf{A}) = \mathbf{R}\,\text{sign}(\mathbf{\Lambda})\mathbf{R}^{-1}$ and $\text{sign}(\mathbf{\Lambda})$ the diagonal matrix containing the signs of all eigenvalues of $\mathbf{A}$. Then, we approximate the previous expression by a quadrature formula as follows:

$$\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \sum_{j=1}^{l} \omega_j \text{sign}\left(\mathbf{A}(\Phi(s_j))\right) \mathbf{A}(\Phi(s_j))\partial_s\Phi(s_j). \tag{3.46}$$

Now, we propose to approximate $\mathbf{A}(\Phi(s_j))\partial_s\Phi(s_j)$ by the following expression:

$$\mathbf{A}(\Phi(s_j))\partial_s\Phi(s_j) \approx \frac{\mathbf{A}_{\Phi_j}}{2\varepsilon_j}\left(\Phi(s_j+\varepsilon_j) - \Phi(s_j-\varepsilon_j)\right), \tag{3.47}$$

where $\mathbf{A}_{\Phi_j} = A(\Phi(s_j-\varepsilon_j), \Phi(s_j+\varepsilon_j))$ is a Roe-matrix associated to the system (see [128] for details), that is a matrix satisfying

$$\begin{aligned}\mathbf{A}_{\Phi_j}\left(\Phi(s_j+\varepsilon_j) - \Phi(s_j-\varepsilon_j)\right) = \mathbf{f}(\Phi(s_j+\varepsilon_j)) - \mathbf{f}(\Phi(s_j-\varepsilon_j)) \\ + \mathcal{B}_{\Phi_j}\left(\Phi(s_j+\varepsilon_j) - \Phi(s_j-\varepsilon_j)\right),\end{aligned} \tag{3.48}$$

where $\mathcal{B}_{\Phi_j}\left(\Phi(s_j+\varepsilon_j) - \Phi(s_j-\varepsilon_j)\right)$ is defined as in the previous section using the states $\Phi(s_j-\varepsilon)$ and $\Phi(s_j+\varepsilon)$. Therefore, the viscosity term reads as follows

$$\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \sum_{j=1}^{l} \omega_j \text{sign}\left(\mathbf{A}(\Phi(s_j))\right) \frac{\mathcal{R}_j}{2\varepsilon_j}, \tag{3.49}$$

where

$$\begin{aligned}\mathcal{R}_j \quad = \quad & \mathbf{f}(\Phi(s_j+\varepsilon_j)) - \mathbf{f}(\Phi(s_j-\varepsilon_j)) \\ & + \mathcal{B}_{\Phi_j}\left(\Phi(s_j+\varepsilon_j) - \Phi(s_j-\varepsilon_j)\right).\end{aligned} \tag{3.50}$$

Note that if $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$ lie on the same stationary solution we have $\Phi(s) = \Phi^E(s)$ and $\mathcal{R}_j = \mathbf{0}$, $j = 1, \ldots, l$ and $\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ vanishes. Therefore, the numerical scheme (3.25) with (3.2), where $\mathcal{B}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ is defined as (3.31), (3.32), (5.19), (3.41) and (3.43) and $\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ is defined by (3.49) is exactly well balanced for stationary solutions given by (3.19)-(3.20).

Here we propose the Romberg method with $l = 3$ and

$$\begin{aligned}&s_1 = 1/4,\ s_2 = 3/4,\ s_3 = 1/2,\\ &\omega_1 = 2/3,\ \omega_2 = 2/3,\ \omega_3 = -1/3,\\ &\varepsilon_1 = 1/4,\ \varepsilon_2 = 1/4,\ \varepsilon_3 = 1/2.\end{aligned} \tag{3.51}$$

With this choice, the viscosity term $\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i)$ of the Osher-Romberg method reads as follows

$$\begin{aligned}
&\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i) = \\
&\frac{4}{3}\,\mathrm{sign}(\mathbf{A}(\Phi(1/4)))\left(\mathbf{f}(\Phi(1/2))-\mathbf{f}(\mathbf{q}_i)+\mathcal{B}_{i+1/4}\left(\Phi(1/2)-\mathbf{q}_i\right)\right) \\
&\frac{4}{3}\,\mathrm{sign}(\mathbf{A}(\Phi(3/4)))\left(\mathbf{f}(\mathbf{q}_{i+1})-\mathbf{f}(\Phi(1/2))+\mathcal{B}_{i+3/4}\left(\mathbf{q}_{i+1}-\Phi(1/2)\right)\right) \\
&-\frac{1}{3}\,\mathrm{sign}(\mathbf{A}(\Phi(1/2)))\left(\mathbf{f}(\mathbf{q}_{i+1})-\mathbf{f}(\mathbf{q}_i)+\mathcal{B}_{i+1/2}\left(\mathbf{q}_{i+1}-\mathbf{q}_i\right)\right).
\end{aligned} \tag{3.52}$$

Note that the major drawback in the previous expression is that the complete eigenstructure of the matrix $\mathbf{A}(\mathbf{Q})$ is required since $\mathrm{sign}(\mathbf{A}) = \mathbf{R}\,\mathrm{sign}(\mathbf{\Lambda})\,\mathbf{R}^{-1}$. However, on the other hand, the Osher-Romberg method is very little dissipative and is stable under the standard CFL condition.

We would like to underline that, without a well balanced way of treating the viscosity, the entire scheme will lose the property of preserving equilibria up to machine precision; instead this choice of the viscosity term guarantees the well balancing and moreover does it in a automatic way, so that it can be easily applied to other systems of equations and family of equilibria.

3.3.1.2 HLL scheme

Following [46], the standard HLL scheme can be written in the form (3.25) with (3.2), where the numerical viscosity term is given by

$$\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i) = \alpha^0_{i+1/2} I_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i) + \alpha^1_{i+1/2}\mathcal{R}_{i+1/2}, \tag{3.53}$$

where $I_{i+1/2}$ is the identity matrix,

$$\mathcal{R}_{i+1/2} = \mathbf{f}(\mathbf{q}_{i+1}) - \mathbf{f}(\mathbf{q}_i) + \mathcal{B}_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i) \tag{3.54}$$

and

$$\alpha^0_{i+1/2} = \frac{S^R_{i+1/2}|S^L_{i+1/2}| - S^L_{i+1/2}|S^R_{i+1/2}|}{S^R_{i+1/2}-S^L_{i+1/2}},\ \alpha^1_{i+1/2} = \frac{|S^R_{i+1/2}| - |S^L_{i+1/2}|}{S^R_{i+1/2}-S^L_{i+1/2}} \tag{3.55}$$

being $S^L_{i+1/2}$ and $S^R_{i+1/2}$ respectively the minimum and the maximum of the wave speeds of the Riemann problem associated to states $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$. In particular to compute $S^L_{i+1/2}$ we recover the eigenvalues associated to $\mathbf{q}_i$ and $\frac{\mathbf{q}_i+\mathbf{q}_{i+1}}{2}$ and we consider the minimum value, similarly to compute $S^R_{i+1/2}$ we

take the maximum of the eigenvalues associated to $\frac{\mathbf{q}_i+\mathbf{q}_{i+1}}{2}$ and $\mathbf{q}_{i+1}$. It is clear that $\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i)$ does not vanish if $\mathbf{q}_{i+1}$ and $\mathbf{q}_i$ lie on a stationary solution: $\mathcal{R}_{i+1/2}$ vanishes, but it is not the case for the term $\alpha^0_{i+1/2} I_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i)$.

Here, we follow the ideas described in [40] and [46] to modify the viscosity term such that the resulting scheme is exactly well balanced for the stationary solutions (3.19)-(3.20). In particular $I_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i)$, will be replaced by $\widetilde{I}_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i)$ that vanishes when a stationary solution is considered. Here we consider the following expression for $\widetilde{I}_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i)$

$$\widetilde{I}_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i)=\begin{pmatrix} b_2^{i+1/2}\left(\frac{\rho}{\gamma p}\right)_{i+1/2} \\ \Delta\left(r\rho u\right)_{i+1/2} \\ b_2^{i+1/2}\left(\frac{\rho}{\gamma p}\right)_{i+1/2}(v)_{i+1/2} \\ b_2^{i+1/2}\left(\frac{\rho}{\gamma p}\right)_{i+1/2}(z)_{i+1/2} \\ 0 \end{pmatrix} \tag{3.56}$$

where $b_2^{i+1/2}$ is given in (5.19), $\left(\frac{\rho}{\gamma p}\right)_{i+1/2}=\frac{\rho_{i+1}+\rho_i}{\gamma(p_{i+1}+p_i)}$, $\Delta\left(r\rho u\right)_{i+1/2}=(r\rho u)_{i+1}$ $-(r\rho u)_i$, $(v)_{i+1/2}=\frac{v_{i+1}+v_i}{2}$, $(z)_{i+1/2}=\frac{z_{i+1}+z_i}{2}$, being $z=\frac{\partial(ru(\rho E+p))}{\partial q_2}$. Following [40] and [46] $\widetilde{I}_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i)$ is obtained as follows: we start by computing the eigenstructure of the extended Jacobian matrix $\mathbf{A}$ at the equilibrium

$$\mathbf{A}(\mathbf{Q})=\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ r\frac{\partial p}{\partial q_1} & 0 & r\frac{\partial p}{\partial q_3} & r\frac{\partial p}{\partial q_4} & \rho\frac{Gm_s}{r}+\rho v^2 \\ 0 & v & 0 & 0 & 0 \\ 0 & \frac{\partial(ru(\rho E+p))}{\partial q_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.57}$$

In this situation the eigenstructure is easy to be computed: let $\mathbf{R}$ the matrix of the right-eigenvectors and $\mathbf{\Lambda}=\mathrm{diag}(\lambda_1,\lambda_2,\ldots,\lambda_5)$ the diagonal matrix of the eigenvalues of (3.57). In particular we have

$$\mathbf{\Lambda}=\mathrm{diag}\left(\frac{\rho u+\sqrt{\gamma\rho p}}{\rho},\frac{\rho u-\sqrt{\gamma\rho p}}{\rho},u,u,0\right),\quad \text{with } u=0. \tag{3.58}$$

Then $\widetilde{I}_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i)$ is given by

$$\widetilde{I}_{i+1/2}(\mathbf{q}_{i+1}-\mathbf{q}_i)=\mathbf{R_{i+1/2}}\widetilde{\Lambda}(\mathbf{R_{i+1/2}})^{-1}(\mathbf{q}_{i+1}-\mathbf{q}_i), \tag{3.59}$$

where

$$\widetilde{\Lambda} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.60}$$

Note that $\widetilde{\Lambda}$ is a diagonal matrix composed with 0 and 1, where the 0 elements correspond to the zero eigenvalues at the stationary solution. The final expression (3.56) is obtained considering the following relation that it is derived from (3.9)

$$\begin{aligned} r_{i+1/2}\Bigg(&\left(\frac{\partial p}{\partial q_1}\right)_{i+1/2}\Delta q_{1,i+1/2} + \left(\frac{\partial p}{\partial q_3}\right)_{i+1/2}\Delta q_{3,i+1/2} \\ &+ \left(\frac{\partial p}{\partial q_4}\right)_{i+1/2}\Delta q_{4,i+1/2} + \left(\rho\frac{Gm_s}{r^2} - \frac{\rho v^2}{r}\right)_{i+1/2}\Delta r_{i+1/2}\Bigg) = b_2^{i+1/2}. \end{aligned} \tag{3.61}$$

We underline once again that this *identity modification* allows the viscosity to vanish at the equilibria, so that the scheme is exactly well balanced. Finally, we would like to note that a similar HLL scheme could also be obtained within the framework of path-conservative HLLEM methods recently proposed by [68], in which according to [79] the intermediate HLL state is assumed to be linear rather than constant.

3.3.2 2nd order well balanced reconstruction

Let us recall the numerical scheme presented in (3.1) considering the space-time conservation form of the PDE

$$\begin{aligned} \mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n &- \frac{\Delta t}{\Delta r}\left(\mathbf{D}^+_{i-\frac{1}{2}}\left(\mathbf{q}_{i-\frac{1}{2}}^{n^+,-}, \mathbf{q}_{i-\frac{1}{2}}^{n^+,+}\right) + \mathbf{D}^-_{i+\frac{1}{2}}\left(\mathbf{q}_{i+\frac{1}{2}}^{n^+,-}, \mathbf{q}_{i+\frac{1}{2}}^{n^+,+}\right)\right) \\ &- \frac{\Delta t}{\Delta r}\int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \frac{\partial}{\partial r}\mathbf{f}\left(\mathbf{q}_i^{n^+}(r)\right) dr \\ &- \frac{\Delta t}{\Delta r}\int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \mathbf{B}_1\left(\mathbf{q}_i^{n^+}(r)\right)\frac{\partial}{\partial r}\left(\mathbf{q}_i^{n^+}(r)\right) dr, \end{aligned} \tag{3.62}$$

where $\mathbf{q}_i^n(r,t)$ is the approximation of the conserved variables inside cell I_i at time t^n, $\mathbf{q}_{i-\frac{1}{2}}^{n^+,+}(t) = \mathbf{q}_i^n(r_{i-1/2}, t^{n+1/2})$ and $\mathbf{q}_{i+\frac{1}{2}}^{n^+-}(t) = \mathbf{q}_i^n(r_{i+1/2}, t^{n+1/2})$,

that is the evaluation of $\mathbf{q}_i^n(r,t)$ at the two boundaries of cell I_i at the time-midpoint of $[t^n, t^{n+1}]$. We would like to underline that in order to obtain a second order scheme $\mathbf{q}_i^n$ should be a second order reconstruction of the cell averages $\mathbf{Q}_{i-1}^n, \mathbf{Q}_i^n, \mathbf{Q}_{i+1}^n$.

According to [128], [38] and Proposition 3.1.6, scheme (3.62) is well balanced if both, the underlying first order scheme and the reconstruction operator are well balanced, and all the integrals that appear in (3.62) are computed exactly. Therefore, in order to define a second order scheme, a second order well balanced reconstruction operator should be defined.

The most popular way to define a second order reconstruction operator is based on the MUSCL method introduced by van Leer in [156] joint with the *minmod* limiter. He proposed to reconstruct $\mathbf{q}_i^n$ using a linear polynomial in space and time as follows

$$\mathcal{P}_i^n(r,t) = \mathbf{Q}_i^n + \frac{\Delta \mathbf{Q}_i^n}{\Delta r}(r - r_i) + \partial_t \mathbf{Q}_i^n (t - t^n), \tag{3.63}$$

where

$$\Delta \mathbf{Q}_i^n = \text{minmod}\left(\Delta \mathbf{Q}_{i-1/2}^n, \Delta \mathbf{Q}_{i+1/2}^n\right) \tag{3.64}$$

with $\Delta \mathbf{Q}_{i-1/2}^n = \mathbf{Q}_i^n - \mathbf{Q}_{i-1}^n$, $\Delta \mathbf{Q}_{i+1/2}^n = \mathbf{Q}_{i+1}^n - \mathbf{Q}_i^n$ and

$$\text{minmod}(a,b) = \begin{cases} 0, & \text{if } ab \leq 0 \\ a, & \text{if } |a| < |b| \\ b, & \text{if } |a| \geq |b|. \end{cases} \tag{3.65}$$

It is clear that the standard MUSCL method is only well balanced for linear stationary solutions, which is not the case here. In this work we therefore follow the strategy proposed in [37], where the reconstruction operator is defined as a combination of a smooth stationary solution together with a standard reconstruction operator to reconstruct the *fluctuations* with respect to the given stationary solution, that is

$$\mathbf{q}_i^n(r,t) = \mathbf{Q}_i^E(r) + \mathcal{P}_i^f(r,t), \quad r \in I_i,\ t \in [t^n, t^{n+1}], \tag{3.66}$$

where $\mathcal{P}_i^f(r,t)$ is the standard MUSCL reconstruction operator applied to the fluctuations around the stationary solution at every cell of the stencil. Thus, if we define

$$\mathbf{Q}_i^{f,n} = \mathbf{Q}_i^n - \mathbf{Q}_i^E, \quad \mathbf{Q}_{i-1}^{f,n} = \mathbf{Q}_{i-1}^n - \mathbf{Q}_{i-1}^E, \quad \mathbf{Q}_{i+1}^{f,n} = \mathbf{Q}_{i+1}^n - \mathbf{Q}_{i+1}^E, \tag{3.67}$$

then, $\mathcal{P}_i^f(r,t)$ is defined as follows:

$$\mathcal{P}_i^{f,n}(r,t) = \mathbf{Q}_i^{f,n} + \frac{\Delta \mathbf{Q}_i^{f,n}}{\Delta r}(r - r_i) + \partial_t \mathbf{Q}_i^n (t - t^n), \tag{3.68}$$

where

$$\Delta \mathbf{Q}_i^{f,n} = \text{minmod}\left(\Delta \mathbf{Q}_{i-1/2}^{f,n}, \Delta \mathbf{Q}_{i+1/2}^{f,n}\right) \tag{3.69}$$

with

$$\Delta \mathbf{Q}_{i-1/2}^{f,n} = \mathbf{Q}_i^{f,n} - \mathbf{Q}_{i-1}^{f,n}, \quad \Delta \mathbf{Q}_{i+1/2}^{f,n} = \mathbf{Q}_{i+1}^{f,n} - \mathbf{Q}_i^{f,n}. \tag{3.70}$$

Note that we have replaced $\partial_t \mathbf{Q}_i^{f,n}(t-t^n)$ by $\partial_t \mathbf{Q}_i^n(t-t^n)$ in (3.68) as $\partial_t \mathbf{Q}_i^E = 0$. It is clear from its construction that the reconstruction operator is exactly well balanced, and it is second order accurate for non-stationary solutions as $\mathbf{Q}^E(r)$ is a smooth stationary solution. The term $\partial_t \mathbf{Q}_i^n$ indicates the time derivative of $\mathbf{Q}$ and it can be computed using a discrete version of the governing equation

$$\begin{aligned} \partial_t \mathbf{Q}_i^n &= -\frac{\mathbf{f}(\mathbf{q}_{i+1/2}^{n,-}) - \mathbf{f}(\mathbf{q}_{i-1/2}^{n,+})}{\Delta r} - \frac{\mathcal{B}_i(\mathbf{q}_{i+1/2}^{n,-} - \mathbf{q}_{i-1/2}^{n,+})}{\Delta r}, \\ \mathbf{q}_{i\pm 1/2}^{n,\mp} &= \mathbf{q}_i(x_{i\pm 1/2}^{\mp}, t^n), \end{aligned} \tag{3.71}$$

where the fluxes have been approximated by a central finite difference with respect to the cell center r_i, and $\mathcal{B}_i(\mathbf{q}_{i+1/2}^{n,-} - \mathbf{q}_{i-1/2}^{n,+})$ is obtained in the same way of (3.31),(5.19),(3.41),(3.43), where by replacing for example $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$ by $\mathbf{q}_{i+1/2}^{n,-}$ and $\mathbf{q}_{i-1/2}^{n,+}$ respectively, and using as central value the cell average one obtains

$$\begin{aligned} &\frac{\mathcal{B}_i(\mathbf{q}_{i+1/2}^{n,-} - \mathbf{q}_{i-1/2}^{n,+})}{\Delta r} = \left(b_1^i \; b_2^i \; b_3^i \; b_4^i \; b_5^i\right)^T \text{ with} \\ &b_1^i = b_5^i = 0, \quad b_3^i = \rho_i u_i v_i, \quad b_4^i = \rho_i u_i \frac{G m_s}{r_i}, \\ &b_2^i = r_i \left(p_{i+1/2}^{f,n,-} - p_{i-1/2}^{f,n,+}\right) + \left(r_i \rho_i^E (\zeta_r)_i^f + r_i \rho_i^f (\zeta_r)_i\right). \end{aligned} \tag{3.72}$$

The last ingredient for a second order scheme is the computation of the integrals in (3.62): the first one can be computed exactly

$$\int_{r_{i-1/2}}^{r_{i+1/2}} \frac{\partial}{\partial r} \mathbf{f}\left(\mathbf{q}_i(r,t)\right) dr = \mathbf{f}(\mathbf{q}_{i+1/2}^{n^+,-}) - \mathbf{f}(\mathbf{q}_{i-1/2}^{n^+,+}). \tag{3.73}$$

Note that this first integral vanishes for stationary solutions with $u = 0$. The second integral is more sophisticated, and it is not easy to compute it exactly, except in some particular situations. Therefore we will use a quadrature formula to approximate this integral, but this must be done carefully to maintain the well balanced property of the scheme: effectively, a wrong choice in the quadrature formula will destroy all the work we have done up to now in order to define a well balanced scheme. Here we proceed as follows: first we express the particular form of the reconstruction operator: $\mathbf{q}_i^n(x,t) = \mathbf{Q}_i^E(x) + \mathcal{P}_i^f(x,t)$ and we use the fact that

$$\int_{r_{i-1/2}}^{r_{i+1/2}} \mathbf{B}_1(\mathbf{Q}_i^E(r)) \frac{\partial \mathbf{Q}_i^E(r)}{\partial r} dr = \mathbf{0}. \tag{3.74}$$

Here, we only show the details for the second component of

$$\int_{r_{i-1/2}}^{r_{i+1/2}} \mathbf{B}_1(\mathbf{q}_i^n(r)) \frac{\partial \mathbf{q}_i^n(r)}{\partial r} dr, \tag{3.75}$$

i.e.

$$\begin{aligned} &\int_{r_{i-1/2}}^{r_{i+1/2}} r \left[\frac{\partial p}{\partial r} + \rho \left(\frac{G m_s}{r^2} - \frac{v^2}{r} \right) \right] dr \\ &\quad = \int_{r_{i-1/2}}^{r_{i+1/2}} r \left[\frac{\partial (p^E + p^f)}{\partial r} + \left(\rho^E + \rho^f \right) \left(\zeta^E + \zeta^f \right)_r \right] dr \\ &\quad = \int_{r_{i-1/2}}^{r_{i+1/2}} r \frac{\partial p^f}{\partial r} + r \rho^E \zeta_r^f + r \rho^f \zeta_r dr. \end{aligned} \tag{3.76}$$

Now, the mid-point quadrature formula is used to ensure second order accuracy obtaining that

$$\begin{aligned} &\int_{r_{i-1/2}}^{r_{i+1/2}} r \left[\frac{\partial p}{\partial r} + \rho \left(\frac{G m_s}{r^2} - \frac{v^2}{r} \right) \right] dr \\ &\quad \approx \Delta r \left[r_i \left(\Delta p^f \right)_i + \left(r \rho^E \right)_i \left(\zeta_r^f \right)_i + \left(r \rho^f \right)_i (\zeta_r)_i \right], \end{aligned} \tag{3.77}$$

where $\left(\Delta p^f \right)_i = \dfrac{p_{i+1/2}^{f,-} - p_{i-1/2}^{f,+}}{\Delta r}$,

$$(\zeta_r)_i = \frac{G m_s}{r_i^2} - \frac{v_i^2}{r_i}, \qquad (\zeta_r^f)_i = \frac{{v_i^E}^2}{r_i} - \frac{v_i^2}{r_i}.$$

It is clear that this approximation is second order accurate and, moreover, will vanish for stationary solutions (3.19)-(3.20). For the third and fourth component we could perform the same procedure, but, again, as both trivially vanish when $u = 0$, we could use directly the mid-point rule.

Finally, note that $r^{\pm}_{i+1/2} = r_{i+1/2}$ and therefore $\Delta r_{i+1/2} = 0$. Therefore $\mathcal{B}_{i+1/2}(\mathbf{q}^{n^+,+}_{i+\frac{1}{2}} - \mathbf{q}^{n^+,-}_{i+\frac{1}{2}})$ reduces to

$$\mathcal{B}_{i+1/2}(\mathbf{q}^{n^+,+}_{i+\frac{1}{2}} - \mathbf{q}^{n^+,-}_{i+\frac{1}{2}}) = (0, b_2^{i+1/2}, 0, 0, 0)^T$$

where

$$b_2^{i+1/2} = r_{i+1/2}\Delta p^f_{i+1/2} = r_{i+1/2}\left(p^{f,+}_{i+1/2} - p^{f,-}_{i+1/2}\right).$$

3.4 Numerical results

First of all, we show the ability of both schemes to preserve a wide class of stationary solutions of the Euler equations with gravity of the form (3.19)-(3.20); we also report the convergence tables for some smooth solutions. Then, we test both methods with a classical Riemann problem: this show that our methods are able to deal with situations far from the equilibrium, hence they do not fall into the case of perturbation methods. And finally, we study their behavior in capturing small perturbations around the equilibrium.

3.4.1 Stationary solutions with constant pressure

Simple, but non trivial, stationary solutions of the Euler equations can be obtained by considering velocities as in (3.19) and a constant pressure p. It is easy to verify that under these conditions for any density profile the velocity in the angular direction v must satisfy

$$v = \sqrt{\frac{Gm_S}{r}}, \tag{3.78}$$

while $u = 0$. For the numerical simulations we consider a spatial domain $r \in [1, 2]$, $G = 1$, $m_s = 1$, $\gamma = 1.4$, $p = 1$ and two density profiles:

$$\rho_1 = r, \tag{3.79}$$

$$\rho_2 = \begin{cases} 1, & \text{if } r < 1.5 \\ 0.1, & \text{if } r \geq 1.5. \end{cases} \tag{3.80}$$

Table 3.1: Constant pressure equilibrium. The following results show the capability of the schemes to preserve equilibria both for a hierarchy of meshes for a fixed time $t = 1$ (on the left) and for a fixed mesh ($N = 64$ cells) and for increasing computational times. The table on the top refers to the L_1-norm error between the continuous ρ_1 profile and the table on the bottom refers to the discontinuous ρ_2 profile. Data have been obtained using either the Osher or HLL flux (and no significant differences have been noticed).

tend = 1		$N = 64$	
N	E_ρ - Osher	time	E_ρ - Osher
64	9.54E-17	1	9.54E-17
128	9.54E-17	2	2.36E-16
256	6.49E-16	5	8.85E-16
512	6.23E-16	10	1.67E-15
1024	1.21E-15	50	6.24E-17

tend = 1		$N = 64$	
N	E_ρ - HLL	time	E_ρ - HLL
64	8.45E-18	1	8.45E-18
128	1.38E-16	2	1.19E-17
256	5.54E-16	5	6.71E-16
512	2.64E-15	10	2.42E-15
1024	5.05E-16	50	1.13E-13

In Table 3.1 we report the errors between the exact equilibrium and the numerical solution obtained with both schemes using a hierarchy of meshes and long term time integration. We can notice that all the errors are of the order of machine precision and no significant differences can be noticed between the two fluxes. Moreover the method is perfectly well balanced both with continuous and discontinuous density profiles, as expected.

3.4.2 General equilibrium

Using the equilibrium relation between the pressure and the gravitational forces in (3.20) and ζ given by (3.34), we obtain another class of stationary solutions

Table 3.2: General equilibrium. L_1-norm error for the density between the exact and the numerical solution. On the left we have the error for different meshes at $t = 1$ and on the right we show the error for a given mesh ($N = 64$ cells) for different computational times.

tend = 1		$N = 64$	
N	E_ρ - OSHER	time	E_ρ - HLL
64	6.28E-15	1	5.03E-15
128	1.17E-14	2	1.01E-14
256	1.70E-14	5	2.65E-14
512	2.15E-14	10	7.21E-14
1024	3.19E-14	50	3.07E-12

of the Euler equations

$$\rho = \rho_0 e^{-\zeta(r)}, \quad p = \rho + p_0, \quad v = \sqrt{r\left(\frac{Gm_s}{r^2} - \zeta_r\right)}. \tag{3.81}$$

We have applied both schemes to two different choices of ζ obtaining always a well balanced result. Table 3.2 shows the L_1-norm error for the density between the equilibrium and the numerical solution in the case

$$\zeta = kr,\ k = -1,\ \rho = \rho_0 e^{-kr},\ \rho_0 = 1,\ p = \rho + p_0,\ p_0 = 1. \tag{3.82}$$

Again, both methods are able to exactly preserve these non-trivial equilibria.

3.4.3 Order of convergence

To study numerically the order of convergence of both schemes we have considered the following equilibrium situation

$$\rho = 1, \quad u = 0, \quad p = 1, \quad v = \sqrt{r\left(\frac{Gm_s}{r^2} - \zeta_r\right)}, \tag{3.83}$$

and at the initial time, we have added a small perturbation (with a Gaussian profile) to the velocity field

$$\begin{aligned} \tilde{u} &= u + 10^{-5}\exp\left(\frac{-0.5\,(r-1.5)^2}{0.01}\right), \\ \tilde{v} &= v + 10^{-5}\exp\left(\frac{-0.5\,(r-1.5)^2}{0.01}\right). \end{aligned} \tag{3.84}$$

Table 3.3: Perturbation around a stationary solution. The reference solution has been obtained with the second order Osher-type scheme over 2^{13} cells. L_1-norm errors for ρ at time $t = 0.1$ are shown: on the left we report the result obtained using the Osher-type flux and on the right using the HLL-type flux.

Osher $\mathcal{O}2$			HLL $\mathcal{O}2$		
N	$\varepsilon(\rho)$	$\mathcal{O}(L_1)$	N	$\varepsilon(\rho)$	$\mathcal{O}(L_1)$
16	1.59E-07	-	16	1.16E-07	-
32	3.82E-08	2.06	32	2.90E-08	2.01
64	9.50E-09	2.00	64	7.22E-09	2.00
128	2.31E-09	2.04	128	1.77E-09	2.03
256	5.72E-10	2.01	256	4.44E-10	1.99
512	1.45E-10	1.97	512	1.14E-10	1.96

We have computed a reference solution using our method with the Osher-type flux over a fine mesh ($N = 2^{13} = 8192$). In Table 3.3 we report the L_1 norm errors for the density ρ with respect to our reference solution and both numerical schemes achieve second order of convergence.

3.4.4 Riemann Problem

To show that our method is accurate even far away from an equilibrium, we consider as initial condition a classical Riemann problem with non-vanishing angular velocity

$$\rho_L = 1.0, u_L = 0, v_L = \sqrt{\frac{Gm_s}{r}}, p_L = 1.0, r = r,\ 1 \leq r \leq 4.5,$$
$$\rho_R = 0.1, u_R = 0, v_R = \sqrt{\frac{Gm_s}{r}}, p_R = 0.1, r = r,\ 4.5 < r \leq 8,$$

and we compute the solution by employing the schemes set up to preserve the equilibrium in (3.80). We report the results obtained with the first and second order scheme and with the HLL and Osher-type flux in Figure (3.1). Note that both schemes produce quite similar results.

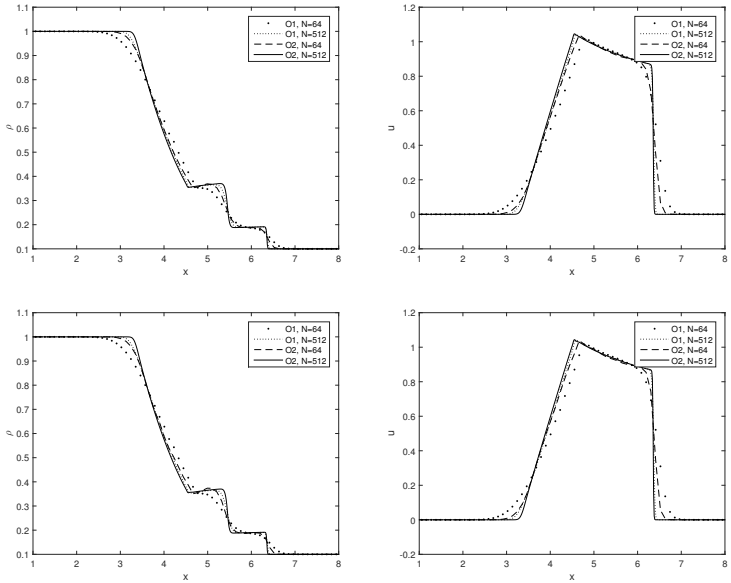


Figure 3.1: Riemann problem at final time $t_f = 1$. On the top the density and velocity profiles obtained using the HLL scheme and on the bottom the profiles obtained with the Osher-type flux. We have employed two meshes: a coarse one with $N = 64$ elements and a fine one with $N = 512$ elements. Moreover, we have compared the first and second order schemes.

3.4.5 Evolution of perturbations

Following the idea presented in [99] we have tried to study small perturbations around the equilibrium. We have considered a density profile as in (3.79) and we have imposed a periodic perturbation on the velocity u through the left boundary conditions, by imposing

$$u_0 = A \sin\left(6\frac{2\pi t^n}{t_f}\right), \quad t_f = 1. \tag{3.85}$$

Two situations are analyzed. First we consider a big perturbation, with $A = 10^{-2}$ and we simulate the evolution using the second order well balanced HLL

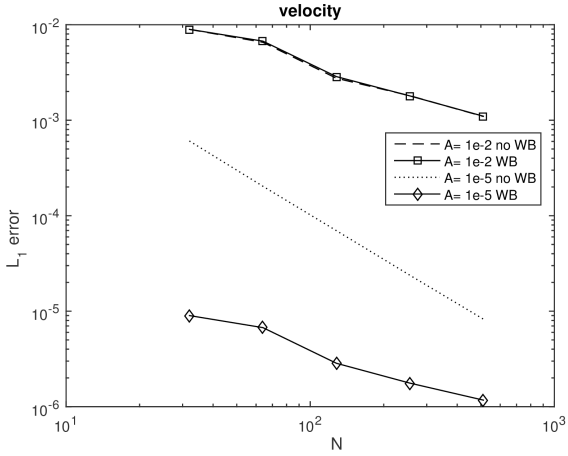


Figure 3.2: Error in the L_1 norm between a reference solution and the numerical solutions computed with the well balanced HLL method and a second order non well balanced scheme. Well balanced and non well balanced methods perform equally well for large perturbations, while well balanced schemes perform significantly better for the small perturbation problem.

scheme and a standard second order HLL scheme using a hierarchy of grids with increasing number of cells. A reference solution computed with the second order well balanced HLL method is also considered using a fine grid composed of $N = 2048$ cells. Figure (3.2) shows the errors for the different meshes. Note that in this case no big differences are visible between the well balanced and not well balanced schemes as the perturbations are so large so that shocks are quickly generated and the solution is far away from the stationary profile. The situation changed significantly when a small perturbation is considered ($A = 10^{-5}$). In that case the well balanced method performs much better than the non well balanced scheme on the finer grid, as shown in Figure (3.2).

4 Well balancing: coupling with nonconforming ALE

Now, we extend our well balanced method to the two dimensional ALE context on moving nonconforming meshes. In particular, the scheme will *inherit* the well balanced property of the previous one-dimensional scheme in the radial direction and the addition of some constraints on the mesh structure will guarantee the well balancing in the angular direction. The extension concerns first the Euler equations of gas dynamics with gravity, see Section 4.1; a large number of numerical test are presented in Section 4.2. Then we consider the shallow water equations (in Section 4.3) and in particular the steady vortex test case already introduced in Section 2.4.5. One can appreciate as the same basic ideas can be applied to a different set of equations and of equilibrium solutions.

4.1 WB ALE for the Euler equations with gravity

Consider the Euler equations of gas dynamics with gravity in the two dimensional non-conservative form stated in (3.23). This section is organized as follows: first, we revisit the moving domain discretization introduced in Section 2.1 by specifying the required shape of the elements and the mesh velocity. Next, we derive the one-step path-conservative ALE scheme that extends to the non-conservative case (2.20), and we explain where the 1D well balanced techniques are employed in order to guarantee the well balancing of the scheme even in a two dimensional moving mesh framework.

4.1.1 Domain discretization and mesh constraints

To discretize the moving domain, we consider a nonconforming mesh $\mathcal{T}_\Omega^n$ as in Section 2.1, but with a fixed total number $N_E = N \times M$ of elements that we suppose to be all quadrilaterals.

These elements should satisfy the following conditions:

i. their barycenters should be aligned along straight lines with $r = r_i = const,\ i=1,\dots N$,

ii. the two bounding edges of each element in radial direction must be aligned with $r = r_{i\pm1/2} = const,\ i=1,\dots N+1$,

iii. the other two bounding edges must be parallel between them.

For example a Cartesian grid satisfies these conditions, but we could accept even something more general (which allows us to move the computational domain). See Figure 4.1 for a general mesh that satisfies the above constraints. In Section 4.1.2.1 these choices will be justified. We emphasize that our numerical scheme works for completely general unstructured and nonconforming moving meshes, but it will be well balanced only if the mesh satisfies these special conditions. Indeed a general well balanced scheme for completely unstructured meshes is more difficult to be achieved, since the considered equilibria have $v \neq 0$, which implies that $\mathbf{g} \neq \mathbf{0}$ (recall that instead $\mathbf{f} = \mathbf{0}$). This does not mean that it is impossible, but it will be the object of another work, because the above conditions do not appear to us as restrictive for our scopes.

We have decided to couple the well balanced techniques with a Lagrangian-type method in order to reduce the errors due to the advection. And our nonconforming ALE algorithm results to be particularly *well suited* for this situation, where the gas at the equilibrium is advected with a *known* equilibrium velocity field $\mathbf{V}(\mathbf{x}) = \left(u^E(\mathbf{x}), v^E(\mathbf{x})\right)$ which reads

$$u^E(\mathbf{x}) = 0, \quad v^E(\mathbf{x}) = \sqrt{r\left(\frac{Gm_s}{r^2} - \zeta_r\right)}. \tag{4.1}$$

Note that the *a priori* knowledge of the velocity field significantly simplifies the application of an ALE scheme: indeed, we can move the nodes following directly the exact equilibrium velocity, which is not affected by any physical or numerical perturbation. The node velocity $\overline{\mathbf{V}}_k^n$ of (2.31) can be obtained by choosing

$$\overline{\mathbf{V}}_{k,j} = \mathbf{V}(\mathbf{x}_j^n),$$

being $\mathbf{x}_j$ the barycenter of element T_j^n. This allows us to control the movement of the mesh avoiding the violation of the above conditions: indeed the radial component of $\overline{\mathbf{V}}_k^n$ will be always zero, hence nodes will slide along straight lines with $r = const$ where the edges lie. Besides, since the barycenters are placed on the straight lines with $r = r_i$, all nodes lying on the same edge will move with the same velocity maintaining the parallelism constraint between the edges.

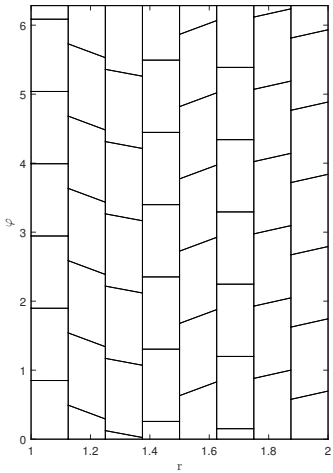


Figure 4.1: Example of a mesh that allows a well balanced treatment of the fluxes. Each element has two vertical edges and the other two are parallel between them. Besides the vertical edges lie on straight lines and the barycenters are aligned along $r = r_i$. Moreover the domain is periodic so that $\varphi = 0$ coincides with $\varphi = 2\pi$.

Furthermore, the presence of slide lines clearly requires a special approach to preserve a high level of grid quality during the mesh motion, and this is automatically provided by our nonconforming treatment of the interfaces, in addition simplified by the fact that their position is *a priori* known: $r = const.$

4.1.2 Well balanced direct ALE scheme

In order to obtain a space-time formulation of a direct path-conservative ALE scheme, as proposed in [69], the governing PDE (1.1) is first reformulated in a space-time divergence form as

$$\tilde{\nabla} \cdot \tilde{\mathbf{F}}(\mathbf{Q}) + \tilde{\mathbf{B}}(\mathbf{Q}) \cdot \tilde{\nabla}\mathbf{Q} = \mathbf{S}(\mathbf{Q}), \qquad \tilde{\nabla} = (\partial_r, \, \partial_\varphi, \, \partial_t)^T \tag{4.2}$$

with

$$\tilde{\mathbf{F}} = (\mathbf{F}, \, \mathbf{Q})^T = (\mathbf{f}, \, \mathbf{g}, \, \mathbf{Q})^T, \; \tilde{\mathbf{B}} = (\mathbf{B}, \mathbf{0})^T = (\mathbf{B}_1, \mathbf{0}, \mathbf{0})^T, \text{ and } \mathbf{S} = \mathbf{0},$$

and it is then integrated over the space-time control volume C_i^n

$$\int_{C_i^n} \left(\tilde{\nabla} \cdot \tilde{\mathbf{F}}(\mathbf{Q}) + \tilde{\mathbf{B}}(\mathbf{Q}) \cdot \tilde{\nabla} \mathbf{Q} \right) d\mathbf{x}dt = \mathbf{0} \,. \tag{4.3}$$

Now, the space-time volume integral of $\tilde{\nabla} \cdot \tilde{\mathbf{F}}(\mathbf{Q})$ can be rewritten using the Gauss theorem as

$$\int_{\partial C_i^n} \tilde{\mathbf{F}} \cdot \tilde{\mathbf{n}} + \int_{C_i^n} \tilde{\mathbf{B}} \cdot \tilde{\nabla} \mathbf{Q} = \mathbf{0}, \tag{4.4}$$

where $\tilde{\mathbf{n}} = (\tilde{n}_r, \tilde{n}_\varphi, \tilde{n}_t)$ is the outward pointing space-time unit normal vector on the space-time surface ∂C_i^n. Taking into account the jump of $\tilde{\mathbf{B}}$ at the interfaces, the final high order ALE one-step Finite Volume scheme is then obtained from equation (4.4) as

$$\begin{aligned} |T_i^{n+1}| \, \mathbf{Q}_i^{n+1} = |T_i^n| \, \mathbf{Q}_i^n - \sum_j \int_0^1 \int_0^1 |\partial C_{ij}^n| \, \tilde{\mathbf{D}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} d\chi d\tau \\ - \int_{C_i^n} \tilde{\mathbf{B}}(\mathbf{q}_i^n) \cdot \tilde{\nabla} \mathbf{q}_i^n \; d\mathbf{x}dt \end{aligned} \tag{4.5}$$

where $\mathbf{q}_i^n(\mathbf{x}, t)$ is a well balanced second order reconstruction of the conserved variables $\mathbf{Q}$ inside cell T_i^n at time t^n, and the discontinuity of the solution at the space-time sub–face ∂C_{ij}^n is resolved by a well balanced path-conservative ALE flux $\tilde{\mathbf{D}}_{ij} \cdot \tilde{\mathbf{n}}_{ij}$, which accounts for the jump in the discrete solution between two neighbors across the intermediate space-time lateral surface. This *generalizes* the ALE scheme introduced in (2.20) for conservative systems to the non-conservative case. Recall that when the lateral surface is shared between more than two control volumes we have to compute the flux across each sub-piece and sum each contribution.

4.1.2.1 Well balanced ALE numerical flux function

The core of the well balanced method in (4.5) is the design of the well balanced space-time flux function. Its final expression will be

$$\begin{aligned} \tilde{\mathbf{D}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} = \; & \frac{1}{2} \left(\tilde{\mathbf{F}}(\mathbf{q}^+) + \tilde{\mathbf{F}}(\mathbf{q}^-) + \mathcal{B}_{ij} \left(\mathbf{q}^+ - \mathbf{q}^- \right) \right) \cdot \tilde{\mathbf{n}}_{ij} \\ & - \frac{1}{2} \mathcal{V}_{ij} \left(\mathbf{q}^+ - \mathbf{q}^- \right), \end{aligned} \tag{4.6}$$

where $\mathbf{q}^-$ is the value of the reconstructed numerical solution inside the element C_i^n evaluated at the space-time midpoint $M_{i,j}^n$ of the lateral surface ∂C_{ij}^n, and

$\mathbf{q}^+$ is the evaluation at the same point of the reconstructed numerical solution inside the neighbor C_j^n at ∂C_{ij}^n. Besides, generalizing the notation introduced in Section 3.1, $\tilde{\mathbf{F}}$ is the physical flux, the term $\mathcal{B}_{ij}\left(\mathbf{q}^+ - \mathbf{q}^-\right)$ represents a well balanced way to write the non-conservative products, and $\mathcal{V}_{i+\frac{1}{2}}\left(\mathbf{q}^+ - \mathbf{q}^-\right)$ is the viscosity term.

As already pointed out, according to [128] (summarized in Definition 3.1.2 and Proposition 3.1.4), the numerical flux should satisfy the following properties

$$\tilde{\mathbf{D}}_{ij}(\mathbf{Q},\mathbf{Q})\cdot\tilde{\mathbf{n}}_{ij} = \mathbf{0} \quad \forall \mathbf{Q}\in\Omega, \text{ and} \tag{4.7}$$

$$\tilde{\mathbf{D}}_{ij}(\mathbf{q}^-,\mathbf{q}^+)\cdot\tilde{\mathbf{n}}_{ij} = \int_0^1 \mathbf{A}_{\mathbf{n}}^{\mathbf{V}}\left(\Phi(s;\mathbf{q}^-,\mathbf{q}^+)\right)\frac{\partial\Phi}{\partial s}(s;\mathbf{q}^-,\mathbf{q}^+)ds, \tag{4.8}$$

where, due to the ALE framework,

$$\begin{aligned}
&\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\mathbf{Q}) = \sqrt{\tilde{n}_r^2+\tilde{n}_\varphi^2}\left(\left(\frac{\partial\mathbf{F}}{\partial\mathbf{Q}}+\mathbf{B}\right)\cdot\mathbf{n}-(\mathbf{V}\cdot\mathbf{n})\,\mathbf{I}\right),\\
&\mathbf{n} = (n_r,n_\varphi) = \frac{(\tilde{n}_r,\tilde{n}_\varphi)^T}{\sqrt{\tilde{n}_r^2+\tilde{n}_\varphi^2}},
\end{aligned} \tag{4.9}$$

with $\mathbf{I}$ representing the identity matrix and $\mathbf{V}\cdot\mathbf{n}$ denoting the local normal mesh velocity.

We explain now how to discretize $\mathcal{B}_{ij}$ and $\mathcal{V}_{ij}$ in (4.6) in a well balanced way. Here we perform our reasoning edge–by–edge and we distinguish two situations: the first one across the vertical edges, which evolving in time originate a surface orthogonal to the radial direction, easier to be treated, and the second one across the other two parallel edges (see the constraints stated at the beginning of Section 4.1.1). For the sake of clarity, in Appendix A we present the proof that our scheme is well balanced taking into account a single element.

First of all, it is easy to see that the flux across the lateral surfaces evolved from the vertical edges coincides with the one dimensional flux. Indeed, in this case, $\mathbf{n} = (n_r, 0)$, $\mathbf{V} = (0, v)$ and so $\mathbf{V}\cdot\mathbf{n} = 0$. Hence $\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\mathbf{Q}) = \mathbf{J}_{\mathbf{f}}(\mathbf{Q}) + \mathbf{B}_1(\mathbf{Q})$ which coincides with the one dimensional Jacobian. So we can discretize $\mathcal{B}_{ij}$ as stated in (3.31)-(5.19)-(3.41)-(3.43), and $\mathcal{V}_{ij}$ by using the Osher-Romberg method (3.52) or the modified HLL scheme as described in Section 3.3.1.2. Therefore the scheme is well balanced in the radial direction and second order accurate provided that the reconstruction $\mathbf{q}_i^n$ and the integrals in (4.5) are computed in a well balanced manner and with second order of accuracy (see Section 4.1.2.2).

For what concerns the flux through the other two surfaces (see Point (*iii*) of Section 4.1.1) let us first state the following remark.

Remark 4.1.1. *Given an element T_i^n consider its two edges which are parallel between them but not vertical. Their evolution in time originates two parallel surfaces with the same areas and opposite normal vectors. Moreover call T_{j_1} and T_{j_2} the two neighbors of T_i through these edges. Since the barycenters of T_i, T_{j_1} and T_{j_2} are aligned on the same vertical line, i.e. their r-coordinate is the same, the equilibrium values $\mathbf{Q}_i^E$, $\mathbf{Q}_{j_1}^E$ and $\mathbf{Q}_{j_2}^E$ coincide.*

Now let us rewrite (4.9) as

$$\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\mathbf{Q}) = \sqrt{\tilde{n}_r^2+\tilde{n}_\varphi^2}\,\left((\mathbf{J_f}+\mathbf{B}_1)\,n_r + \mathbf{J_g} n_\varphi - (\mathbf{V}\cdot\mathbf{n})\,\mathbf{I}\right). \tag{4.10}$$

and (4.8) as

$$\tilde{\mathbf{D}}_{ij}(\mathbf{q}^-,\mathbf{q}^+)\cdot\tilde{\mathbf{n}}_{ij} = \sqrt{\tilde{n}_r^2+\tilde{n}_\varphi^2}\int_0^1 ((\mathbf{J_f}+\mathbf{B}_1)\,n_r + \mathbf{J_g} n_\varphi - (\mathbf{V}\cdot\mathbf{n})\,\mathbf{I})\Phi(s)\frac{\partial\Phi}{\partial s}(s)ds. \tag{4.11}$$

Thus, by exploiting the linearity of the integral, we can give the discretization of $\tilde{\mathbf{D}}_{ij}\cdot\tilde{\mathbf{n}}_{ij}$ in (4.6) as the sum of the following contributions

$$\begin{aligned}\tilde{\mathbf{D}}_{ij}\cdot\tilde{\mathbf{n}}_{ij} = \quad & \frac{1}{2}\left(\mathbf{f}(\mathbf{q}^+)+\mathbf{f}(\mathbf{q}^-)+\mathcal{B}_{ij}\left(\mathbf{q}^+-\mathbf{q}^-\right)\right)\tilde{n}_r \\ & +\frac{1}{2}\left(\mathbf{g}(\mathbf{q}^+)+\mathbf{g}(\mathbf{q}^-)\right)\tilde{n}_\varphi \\ & +\frac{1}{2}\left(\mathbf{q}^+ + \mathbf{q}^-\right)\,\tilde{n}_t - \frac{1}{2}\mathcal{V}_{ij}\left(\mathbf{q}^+-\mathbf{q}^-\right).\end{aligned} \tag{4.12}$$

Note that, whereas the discretization of $\tilde{\mathbf{F}}$ and of $\mathcal{B}_{ij}$ can be splitted, the same cannot be done automatically for the viscosity $\mathcal{V}_{ij}$, whose expression depends on the chosen method (Osher-Romberg, HLL or others).

The expression in (4.12) results to be well balanced, provided that a well balanced expression for $\mathcal{V}_{ij}$ is given. Indeed the first row coincides with the one dimensional flux along the radial direction for which $\mathcal{B}_{ij}$ is given by (3.31)-(5.19)-(3.41)-(3.43) that are well balanced. With regards to the second line we know that in general it is not zero evaluated at the equilibrium because, as already pointed out at the beginning of the section, $\mathbf{g}$ is not zero evaluated at the equilibrium. But, if we consider, together with the flux between T_i and T_{j_1}, also the flux between T_1 and T_{j_2} and we sum them up, we can see that all the values at the equilibrium cancel exactly, thanks to the properties stated in

Remark 4.1.1, that follows from the geometrical constraints we have imposed on our mesh. Finally, the same argument shows that also the third line goes to zero when $\mathbf{q}^- = Q_i^E$ and $\mathbf{q}^+ = Q_{j_1,j_2}^E$.

Viscosity term

To end with the formulation of the well balanced ALE flux (4.6) across this second kind of surfaces, we have to provide an expression for the viscosity $\mathcal{V}_{ij}(\mathbf{q}^+ - \mathbf{q}^-)$ which vanishes on stationary solutions (3.19)-(3.20).

First of all, it is easy to generalize the Osher-Romberg scheme introduced in Section 3.3.1.1. Indeed in the two dimensional ALE context the viscosity matrix introduced in (3.44) can be written as

$$\mathcal{V}_{ij}(\mathbf{q}^+ - \mathbf{q}^-) = \int_0^1 \left| \mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\mathbf{Q})\left(\Phi(s)\right) \right| \partial_s \Phi(s), \quad 0 \le s \le 1. \tag{4.13}$$

Following the same reasoning of Section 3.3.1.1 we get the following expression

$$\mathcal{V}_{ij}(\mathbf{q}^+ - \mathbf{q}^-) = \sum_{j=1}^{l} \omega_j \text{sign}\left(\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\Phi(s_j))\right) \frac{\mathcal{R}_j}{2\varepsilon_j}, \tag{4.14}$$

where

$$\begin{aligned} \mathcal{R}_j = \tilde{\mathbf{F}}(\Phi(s_j + \varepsilon_j)) - \tilde{\mathbf{F}}(\Phi(s_j - \varepsilon_j)) \\ + \tilde{\mathcal{B}}_{\Phi_j}\left(\Phi(s_j + \varepsilon_j) - \Phi(s_j - \varepsilon_j)\right) \end{aligned} \tag{4.15}$$

is discretized as explained in the 1D case above and the Romberg quadrature formula with $l = 3$ is still used. Hence, if $\mathbf{q}_i^n$ and $\mathbf{q}_{i+1}^n$ lie on the same stationary solution $\Phi(s) = \Phi^E(s)$ and $\mathcal{R}_j = \mathbf{0}$, $j = 1, \ldots, l$. Thus, the extension to two dimensions of the Osher-Romberg scheme results to be straightforward. The only drawback is that the complete eigenstructure of the extended Jacobian matrix $\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}$ should be computed, which could be costly in particular when edges are not parallel to the axis (we underline that $\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}$ does not enjoy the property of rotational invariance that characterizes the Euler equations in Cartesian coordinates). As counter part, the method is very little dissipative and allows us to obtain very good results in convective transport problems.

The generalization of the HLL scheme is simpler. equation (3.53) can be rewritten in two dimensions as

$$\mathcal{V}_{ij}(\mathbf{q}^+ - \mathbf{q}^-) = \alpha_{ij}^0 I_{ij}(\mathbf{q}^+ - \mathbf{q}^-) + \alpha_{ij}^1 \mathcal{R}_{ij}, \tag{4.16}$$

where I_{ij} is the identity matrix,

$$\mathcal{R}_{ij} = \mathbf{F}(\mathbf{q}_{i+1}) - \mathbf{F}(\mathbf{q}_i) + \mathcal{B}_{ij}(\mathbf{q}^+ - \mathbf{q}^-) \tag{4.17}$$

(which can be discretized as described in Section 4.1.2.1 to maintain the well balanced properties), and $\alpha_{ij}^{0,1}$ can be computed as in (3.55) being S^L and S^R the minimum and the maximum eigenvalues of $\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\mathbf{q}_{i,i+1}^n)$.

For the same reasons stated in Section 3.3.1.2, I_{ij} must be replaced by a matrix that vanishes when a stationary solution is considered. In particular we choose the following identity modification

$$\tilde{I}_{ij} = \tilde{I}_{i+1/2}\, n_r + I n_\varphi, \tag{4.18}$$

where $\tilde{I}_{i+1/2}$ is given by (3.56), which we already know to vanish for stationary solutions. Moreover it follows from Remark 4.1.1 that when $n_\varphi \neq 0$ the term $I n_\varphi$ cancels at the equilibrium (by considering the two contributions of the neighbors T_{j_1} and T_{j_2} of T_i).

4.1.2.2 2nd order well balanced reconstruction

The missing ingredient for (4.5) to be well balanced up to second order is the definition of a second order well balanced reconstruction operator. As in the one dimensional case we are going to employ a combination of a smooth stationary solution together with the standard MUSCL method, hence our reconstruction will be of the form

$$\mathbf{q}_i^n(\mathbf{x},t) = \mathbf{Q}_i^E(\mathbf{x},t) + \mathcal{P}_i^f(\mathbf{x},t), \quad \mathbf{x} \in C_i^n, \tag{4.19}$$

where, as in Section 3.3.2, $\mathcal{P}_i^f(\mathbf{x},t)$ is the standard MUSCL method applied in order to reconstruct the fluctuations with respect to the given stationary solution computed for all the neighbors T_j^n of T_i^n as

$$\mathbf{Q}_j^{f,n} = \mathbf{Q}_j^n - \mathbf{Q}_j^{E,n}. \tag{4.20}$$

The expression of the reconstruction operator is

$$\mathcal{P}_i^{f,n}(\mathbf{x},t) = \mathbf{Q}_i^{f,n} + \Phi_i \nabla \mathbf{Q}_i^{f,n}(\mathbf{x}-\mathbf{x}_i) + \partial_t \mathbf{Q}_i^n (t-t^n), \tag{4.21}$$

where $\mathbf{x}_i$ is the barycenter of cell T_i^n. To compute $\nabla Q_i^{f,n}$ we use the standard MUSCL method (see [156]) together with the Barth and Jespersen limiter (see [11]), as described in Section 2.1.2. We would like to remark that the employed methods are standard, the novelty is in the fact that both are applied only to the fluctuations.

Finally, the term $\partial_t \mathbf{Q}_i^n$ indicates the time derivative of $\mathbf{Q}$ and it can be computed using a discrete version of the governing equation

$$\partial_t \mathbf{Q}_i^n = (\mathbf{J_f} + \mathbf{B}_1)|_{\mathbf{x}_i} \partial_r \mathbf{Q}(\mathbf{x}_i) + \mathbf{J_g}|_{\mathbf{x}_i} \partial_\varphi \mathbf{Q}(\mathbf{x}_i), \tag{4.22}$$

evaluated at the barycenter $\mathbf{x}_i$ of T_i^n. In particular the gradient of the conserved variables must be expressed as the gradient of the equilibrium plus the previously computed gradient of the fluctuation, i.e.

$$\nabla Q = \nabla \mathbf{Q}_i^E + \nabla \mathbf{Q}_i^f = \nabla \mathbf{Q}_i^E + \nabla \mathbf{Q}_i^{f,n}, \tag{4.23}$$

in order to preserve the well balancing.

The same idea of (4.23) can be exploited in order to rewrite

$$\int_{C_i^n} \tilde{\mathbf{B}}(\mathbf{q}_i^n) \cdot \tilde{\nabla} \mathbf{q}_i^n \; d\mathbf{x}dt, \tag{4.24}$$

where, as in Section 3.3.2, the equilibrium terms cancel and the remaining terms all contain fluctuations. So the integral can be computed through the mid-point quadrature rule which is second order accurate on the fluctuations without affecting the equilibrium.

4.2 Numerical results

4.2.1 Equilibrium preservation

First of all we want to show the accuracy of our scheme in preserving some equilibrium of interest. We consider a discontinuous equilibrium

$$\begin{gathered} \rho = 1, \text{ if } r < r_m, \quad \rho = 0.1, \text{ if } r \geq r_m, \\ u = 0, \quad v = \sqrt{\frac{Gm_s}{r}}, \quad p = 1, \end{gathered} \tag{4.25}$$

with $r_m = 1.5$, $G = 1$, $m_s = 1$, over the computational domain $[r, \varphi] \in [1, 2] \times [0, 2\pi]$. In Figure 4.2 we depict the density profile at the equilibrium and in Table 4.1 we report the maximum error, committed using the HLL flux, with respect to the exact solution after long computational times over a coarse mesh, both for order 1 and 2. The equilibrium results to be perfectly preserved.

Then we consider a hydrostatic equilibrium without tangential velocity, so that the gravity force is perfectly balanced with the pressure gradient. The initial data is given by

$$\rho = 1, \quad \mathbf{u} = (u, v) = \mathbf{0}, \quad P = 1/r, \quad G = m_s = 1. \tag{4.26}$$

Table 4.1: Maximum error between the exact and the numerical density obtained with the first and the second order well balanced methods (using the HLL flux). We underline that similar results have been obtained using the Osher-Romberg flux and that the same precision is achieved for the velocities.

points 20×40		
time	$\mathcal{O}1$	$\mathcal{O}2$
10	7.32E-13	4.20E-13
40	2.83E-12	8.18E-12
80	3.92E-12	1.72E-11
100	2.25E-12	1.99E-11

We consider a computational domain $[r, \varphi] \in [1, 2] \times [0, 2\pi]$ covered by a coarse mesh of 20×40 elements. In Table 4.2 we show the error between the analytical solution and our numerical solution obtained with the second order Osher-Romberg scheme. Since the scheme is exactly well balanced the errors are maintained at the order of machine precision for very long computational times. Similar results are also achieved with our well balanced HLL-type flux.

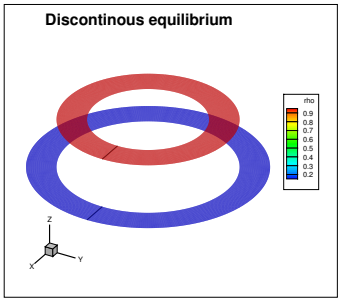


Figure 4.2: Discontinous density profile for the equilibrium solution considered in the test case of Section 4.2.1.

Table 4.2: Hydrostatic equilibrium. Maximum error in L_∞ norm between the exact solution and the numerical results for density, velocity and pressure at different times. The values refer to the second order Osher-Romberg ALE scheme, but similar results have been obtained at first order and with the HLL-type flux.

time	E_ρ	E_u	E_v	E_P
1	7.77E-15	3.29E-16	3.95E-16	3.33E-16
10	1.60E-14	3.16E-16	1.05E-15	3.33E-16
40	2.66E-14	3.58E-16	1.37E-15	3.33E-16
80	3.02E-13	1.30E-13	4.98E-14	3.87E-14

4.2.2 Order of convergence

To study numerically the order of convergence of our method we consider a smooth isentropic vortex, similar to the one proposed in [95]. The initial condition in polar coordinates is given by

$$\begin{aligned} &\rho = 1 + \delta\rho, \quad u = 0, \quad v = \delta v, \quad P = 1 + \delta P, \\ &\delta v = r\frac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}}, \quad \delta T = -\frac{(\gamma-1)^{\varepsilon^2}}{8\gamma\pi} e^{1-r^2}, \\ &\delta P = (1+\delta T)^{\frac{1}{\gamma-1}} - 1, \quad \delta\rho = (1+\delta T)^{\frac{\gamma}{\gamma-1}} - 1, \end{aligned} \tag{4.27}$$

with $\varepsilon = 5$, $G = 0$, $m_s = 0$ and $\gamma = 1.4$ and the computational domain defined as $[r, \varphi] = [1, 2] \times [0, 2\pi]$. The final time is $t_f = 1$. Our new scheme is able to preserve this equilibrium up to machine precision if we impose the above initial data (4.27) also as the equilibrium profile to be preserved. However, it is also possible to impose a different equilibrium profile to be maintained, e.g. the one given by (4.25). In this way, equilibrium and initial condition are not close one to the other so the method comes back to its standard order of convergence, i.e. second order. Refer to Table 4.3 and Figure 4.3 for the numerical results, which confirm that our scheme is indeed second order accurate away from the prescribed equilibrium profile. Finally, we would like to remark that we are working with a moving nonconforming grid. In Figure 4.4 we report an example of the final mesh configuration obtained with our Osher-Romberg scheme.

Table 4.3: Order of convergence, isentropic vortex. We report the results obtained with our second order accurate well-balanced Osher-Romberg ALE scheme. The mesh size h is computed as the maximum incircle diameter of the elements of the final mesh. The errors refer to the L_1 norm of the difference between our numerical solution and the exact one. The last column refers to the setting where the initial data (4.27) are also imposed as the smooth known equilibrium to be maintained, hence in this case the scheme is accurate up to machine precision. The other results are for the setting where the code is used to evolve a different equilibrium profile (4.25) that does not coincide with the initial data (4.27), so that we can show its formal order of accuracy.

mesh size h	E_ρ, eq. (4.25)	$\mathcal{O}(L_1)$	E_ρ, eq. (4.27)
5.59E-2	1.48E-4	-	1.86E-14
2.80E-2	3.60E-5	2.04	1.45E-13
1.86E-2	1.58E-5	2.03	4.78E-13
1.40E-2	8.85E-6	2.02	5.36E-13

4.2.3 Riemann problem

To show the correctness of our method also in the presence of shock waves we solve a classical Riemann problem with non-vanishing angular velocity using both the well balanced HLL and Osher-Romberg ALE schemes. We consider the computational domain $[r, \varphi] = [1, 4] \times [0, 2\pi]$ and we impose the following initial conditions

$$\begin{aligned} &\rho = 1, \text{ if } r < r_m, \quad \rho = 0.1, \text{ if } r \geq r_m, \\ &u = 0, \qquad v = \sqrt{G\, m_s / r}, \\ &p = 1, \text{ if } r < r_m, \quad p = 0.1, \text{ if } r \geq r_m, \end{aligned} \tag{4.28}$$

with $r_m = 2.5$. The results at the final computational time $t_f = 0.5$ are shown in Figure 4.5 where we report a cut along $\varphi = \pi/2$ and a comparison with a one-dimensional reference solution computed on a fine grid using 1024 elements. We note a good agreement between the numerical solution obtained with the well balanced ALE scheme on moving non-conforming meshes and the reference solution also in this case where the solution is far from any equilibrium. Moreover we show the order of convergence of our method with respect to the reference solution in Figure 4.6: obviously it cannot reach order two because of the presence of shocks. However, the observed convergence order is higher than one.

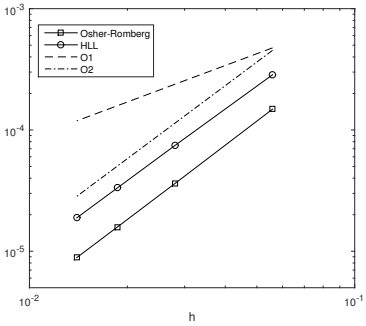


Figure 4.3: Order of convergence, isentropic vortex for imposed equilibrium (eq.) given by (4.25), i.e. different from the initial data of the isentropic vortex (4.27). We report the L_1 error norm of the density obtained with our well-balanced Osher-Romberg and HLL ALE schemes. The dashed lines represent the theoretical slopes of order one and two, respectively.

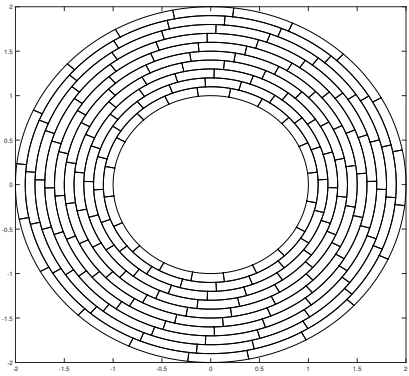


Figure 4.4: Isentropic vortex, final mesh. We report the final mesh configuration at time $t_f = 1$ obtained with our Osher-Romberg scheme in the case of a very coarse mesh of 10×20 elements so that the nonconforming motion is clearly visible.

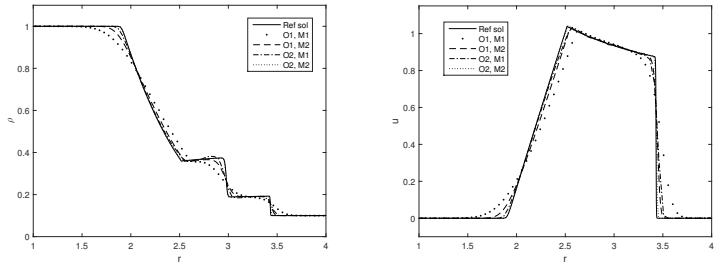


Figure 4.5: Riemann problem in a 2D domain. The test heave been carried out over two meshes: the first one, M_1, with 64×20 control volumes and the second one, M_2, with 256×40 control volumes. The reported results have been obtained using the well balanced HLL scheme with first and second order of accuracy. On the left we report the results for the density and on the right for the velocity at the final time $t_f = 0.5$. The graphs have been obtained as a 1D cut along $\varphi = \pi/2$. One can observe that the second order scheme captures the discontinuities sharply. The results are compared against a reference solution obtained with our second order well balanced HLL scheme in one space dimension with $N = 1024$.

4.2.4 Noh shock test

The Noh shock test consists of a circular infinite strength shock propagating out from the origin. We have chosen this test case to prove that our method can deal also with highly supersonic flows, low pressure atmospheres and shocks of infinite strength. Consider a gas with $\gamma = 5/3$ initialized with density $\rho = 1$, radial velocity $u = -1$, tangential velocity $v = 0$, and pressure $P = 10^{-6}$ as an approximation to zero pressure. The shock wave propagates with speed $1/3$. The exact solution inside the shock region, i.e. $r \leq \frac{t}{3}$, is given by the following relations

$$\rho = 16, \quad P = 16/3, \quad u = 0, \quad v = 0, \tag{4.29}$$

and outside the shock region, i.e. $r > \frac{t}{3}$, by

$$\rho = 1 + \frac{t}{r}, \quad P = 0, \quad u = -1, \quad v = 0. \tag{4.30}$$

We consider an initial domain $[r, \varphi] \in [0, 1] \times [0, \pi/2]$. We impose periodic boundary conditions on $\varphi = 0 = \pi/2$, and we exploit the exact solution to impose the boundary conditions at $r = 0$ and on the moving outer boundary.

The presented results have been obtained with the HLL-type scheme. First we have considered the Eulerian case, hence we have imposed a zero mesh

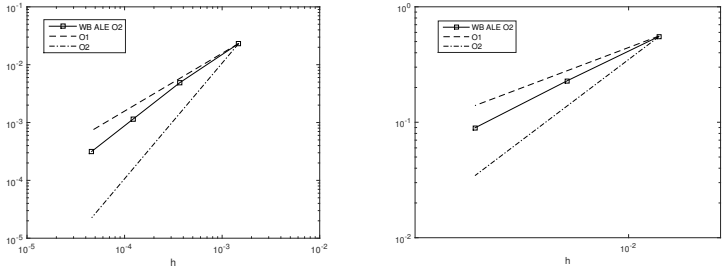


Figure 4.6: Convergence results. Left: we refer to the Riemann problem (4.28) and compare the results obtained with our WB ALE HLL code with a fine grid reference solution. Right: we refer to the Noh shock test of Section 4.2.4 and we compare our results with the exact solution. Note that the L_1 norm of our numerical errors are depicted with squares and is compared with the theoretical slopes of order one and two (dotted lines), respectively. It is evident that the method is better than first order accurate even in presence of shocks.

velocity. The results at time $t_f = 1.2$ obtained with the second order scheme are shown in Figure 4.7. Then we have employed the ALE framework moving the mesh with the local fluid velocity. Due to the absence of shear flow, the mesh remains conforming. The results obtained with the moving mesh are shown in Figure 4.8, where the well-known wall heating problem is visible. Apart from the wall heating, in both the cases the method shows a good agreement with the exact solution. For what concerns the convergence performances of our code refer to Figure 4.6.

4.2.5 Comparison with the PLUTO code

For the following test cases that concern Keplerian discs, we compare the results obtained with our new second order well balanced Osher-Romberg scheme with the results one can obtain with the PLUTO code. PLUTO is a freely-distributed software for the numerical solution of mixed hyperbolic/parabolic systems of partial differential equations (conservation laws) targeting high Mach number flows in astrophysical fluid dynamics. The code has been systematically checked against several benchmarks available in the literature in the papers [120] and [122], using fixed uniform and AMR grids. It provides a multi-physics and multi-algorithm modular environment, where one can choose the Newtonian description for the fluid motion (`HD` option) and add a potential

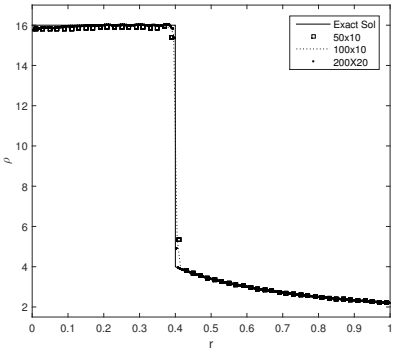


Figure 4.7: Noh shock test. We show the numerical results obtained with our second order HLL-type flux at time $t_f = 1.2$ on three fixed grids with respectively 50×10, 100×10 and 200×20 elements. In the figure the density profile ρ has been depicted along the radial direction r, compared with the exact solution.

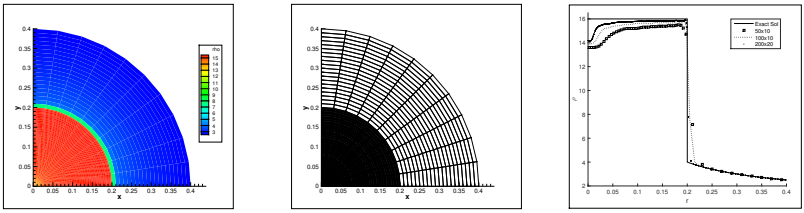


Figure 4.8: Noh shock test. We show the density profile (left) and the final mesh (center) obtained with the second order ALE HLL-type scheme at time $t_f = 0.6$, using a moving grid of 100×10 elements. On the right we compare the density profile along the radial direction r with the exact solution for three different meshes with respectively 50×10, 100×10 and 200×20 elements.

$\Phi = -\frac{Gm}{r}$ to the right hand side by setting the option `BODY_FORCE` equal to `POTENTIAL`. In this way one can study (3.15) within this code. Then we select `POLAR GEOMETRY` and we do not activate any other options.

The modular structure allows us to choose between different numerical fluxes, limiters, spatial reconstructions and time integrators. In particular, we have selected a little dissipative setting by imposing `LIMITER` equal to `MC_LIM` (the

monotonized central difference limiter), or sometimes equal to `MINMOD_LIM` (the classical minmod limiter), and using the `Roe solver` as numerical flux. Then we have compared our second order scheme with both a *second order* configuration of PLUTO (with `LINEAR` reconstruction in space and `RK2` in time) and a *third order* configuration (with `WENO3` reconstruction in space and `RK3` in time). Finally, for the comparison we set the number of elements in PLUTO either equal to the number of elements used for our scheme, or we double it in each dimension.

We remark that within PLUTO special care is taken for the treatment of source and pressure terms when a polar (cylindrical or spherical) geometry is chosen, because in those cases the equations are discretized in angular momentum conserving form and pressure terms are treated separately. For this reason the results are more accurate than those obtained with standard finite volume techniques.

4.2.6 Mass transport in a Keplerian disc

Let us consider a steady state solution of the Euler equations with gravity which satisfies the constraints in (3.19)-(3.20) and with a constant density profile,

$$\rho_E = 1, \quad u_E = 0, \quad v_E = \sqrt{\frac{Gm_s}{r}}, \quad p = 1, \tag{4.31}$$

over the computational domain $[r, \varphi] \in [1, 2] \times [0, 2\pi]$. At the initial time, we perturb this equilibrium solution by imposing a higher density $\rho = 2$ within the disc defined in Cartesian coordinates as $(x - 1.5)^2 + y^2 \leq (0.15)^2$.

The expected result is the transport of this density fluctuation (contact discontinuity) at different velocities which are bigger at the interior and smaller at the exterior, without any dissipation. The velocity and the pressure field should remain constant in time, according to the equilibrium solution. In Figure 4.9 we compare the results obtained with different numerical methods with the exact solution: Eulerian and ALE schemes coupled or not with the well balanced Osher Romberg scheme. As expected, the Eulerian scheme is very dissipative, even when coupled with our new well balanced technique. The dissipation is evident in the angular direction, since the radial velocity in this problem is zero and the Osher scheme is a complete Riemann solver that is able to resolve steady contact waves exactly. The ALE scheme, without well balancing does not dissipate too much in the angular direction, but if it is not coupled with a proper well balanced technique, some spurious velocity oscillations appear which lead to unphysical dissipation in the radial direction and

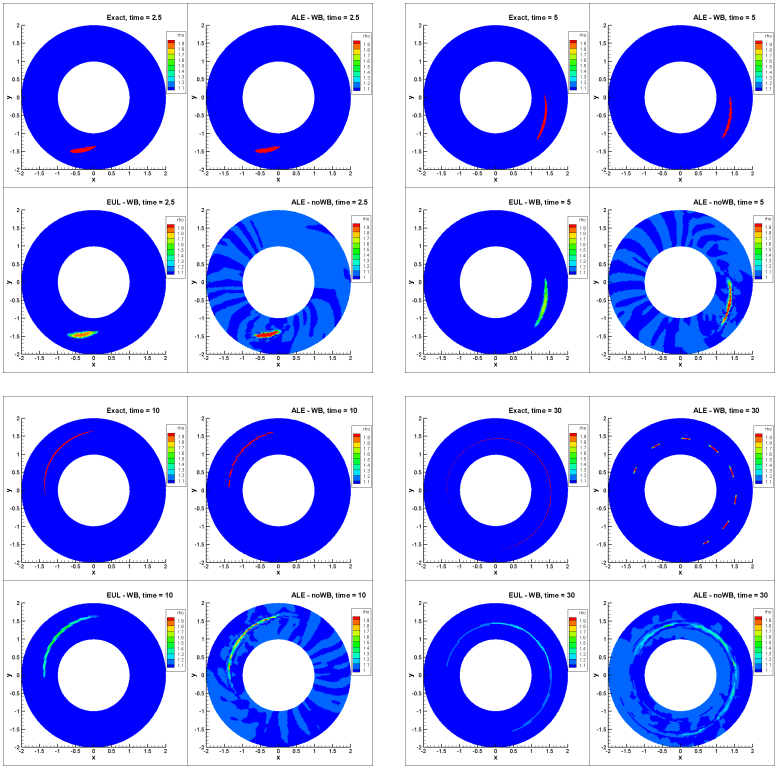


Figure 4.9: We compare the exact solution with the numerical solutions obtained with different methods at times $t = 2.5$ (top-left), $t = 5$ (top-right), $t = 10$ (bottom-left), $t = 30$ (bottom-right). For all the cases the employed numerical flux is an Osher-type flux. The Lagrangian algorithms show their ability in reducing the viscosity along the angular direction. The well balanced methods do not diffuse the quantities in the radial direction. When coupled together (top-right of each square) we obtain a result very close to the exact solution (top-left of each square). We want to remark that in the well balanced ALE case (top-right of each square), the quantity with higher density remains in the same cells in which it is confined at the initial time since the method is very little diffusive in any direction and the differential rotation is treated in a nonconforming way. Moreover, only the well balanced ALE scheme is able to maintain the concentration of the higher density gas.

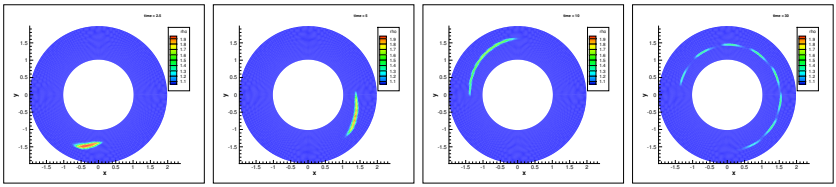


Figure 4.10: Results obtained with PLUTO, using the Roe solver combined with the mc_lim limiter, linear reconstruction in space and RK2 in time on a grid of 30×350 elements. One can observe that the results are more dissipative compared to those shown in Fig. 4.9.

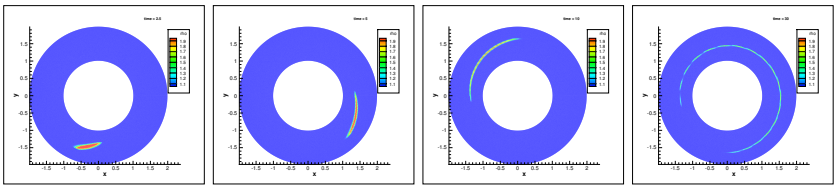


Figure 4.11: Results obtained with PLUTO, using the Roe solver, a third order WENO reconstruction in space combined with the mc_lim limiter and a third order RK3 time integrator on a grid of 60×700 elements.

which also produce some oscillations on the density profiles, which are evident even for short computational times.

The coupling between the two techniques reduces the dissipation both in the radial and in the angular directions. In our computations we have observed that for this test problem the error in the pressure and in the velocity field was always of the order of *machine precision*, since the advection of a contact discontinuity does *not* affect the equilibrium of pressure and velocity. We emphasize that this property of conserving even moving equilibria (density is not constant in time here) is anything else than trivial to achieve and to the best knowledge of the authors, the scheme presented here is the first Finite Volume method to achieve it. Referring to Table 4.4, one can notice that indeed the precision achieved by our code on angular velocity and pressure is of the order of machine precision (even at time $t = 30$), where instead this is not the case for various PLUTO configurations.

Finally, we report the results obtained with PLUTO by selecting the configuration setting described in Section 4.2.5 with the `MC_LIM`. First, in Figure

Table 4.4: The results shown in this table testify that our code is able to maintain up to machine precision even moving equilibria. Indeed for the test cases presented both in Section 4.2.6 and Section 4.2.7 the L_1 norm of the difference between the numerical solution computed with our WB ALE Osher Romberg scheme and the exact stationary profiles of angular velocity v and pressure P, at the respective final times ($t = 30$ and $t = 15$), is of the order of machine precision. The other two lines refer to the results obtained with PLUTO both with second and third order of accuracy.

		Test Section 4.2.6		Test Section 4.2.7	
Method	Elem	$\|\|v-v_E\|\|_{L_1}$	$\|\|P-P_E\|\|_{L_1}$	$\|\|v-v_E\|\|_{L_1}$	$\|\|P-P_E\|\|_{L_1}$
WB ALE Osher-R.	100×200	2.17E-12	7.19E-14	2.13E-12	6.36E-14
PLUTO O2 minmod	100×200	5.56E-7	2.36E-6	5.44E-7	9.89E-6
PLUTO O3 mc_lim	200×400	1.30E-7	5.28E-7	1.49E-7	2.44E-6

4.10 we use the described second order method and 30×350 elements. Then, in Figure 4.11 we use the third order method and 60×700 elements. In both the cases the density is dissipated faster than with our method: this shows that it is not a finer grid or a higher order of accuracy that can solve this type of problem, but a very specific treatment of the equilibrium together with the Lagrangian framework proposed in this paper.

4.2.7 Keplerian disc with density perturbations

For this test we have considered the equilibrium profile in (4.31) and we have added a periodic perturbation over the density profile as follows

$$\rho = \rho_E + A\sin(k_1\varphi)(0.25 - |r_m - r|),\ r \in [r_1, r_2] \tag{4.32}$$

with $A = 0.5$, $k_1 = 12$, $r_1 = 1.25$, $r_2 = 1.75$, $r_m = 1.5$. The goal of the this test is to show that our well balanced ALE scheme is able to maintain the equilibrium pressure and velocity exactly and that the numerical method does *not* generate any spurious numerical perturbations of pressure and velocity that would usually lead to Kelvin-Helmholtz type flow instabilities for density fluctuations combined with shear flow as in the above setup. In Figure 4.12 we show the evolution of the perturbations at different times. They are properly transported with different velocities with only very little numerical dissipation. As in the previous case we stress that the velocity and pressure

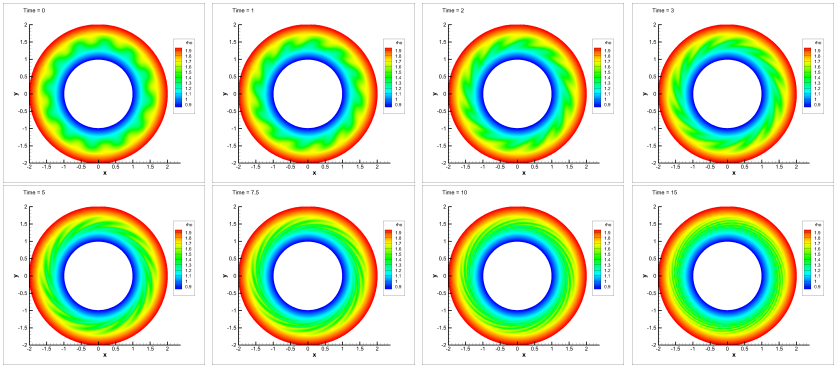


Figure 4.12: Evolution of periodic density perturbations in an equilibrium disc obtained with the well balanced ALE scheme with Osher-Romberg flux. The perturbations are perfectly convected (with an inner velocity bigger than the outer one), and no spurious Kelvin-Helmholtz instabilities are generated, even after long computational times.

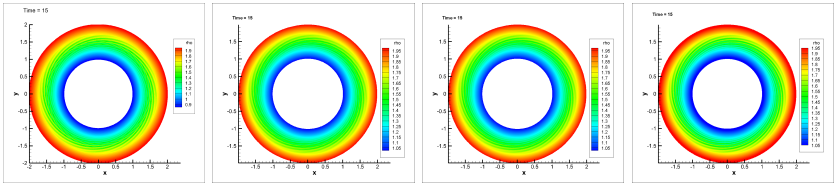


Figure 4.13: Method comparison at time $t = 15$. The first image is obtained with our code and 50×500 elements. The second and the third one with PLUTO using 50×500 elements and respectively a second order scheme with mc_lim limiter and a third order scheme with minmod_lim limiter. The last image is obtained with the third order version of PLUTO using mc_lim and 100×1000 elements. All images are drawn with the same color map. Even if the results are similar, one can notice that to obtain the same resolution of our code we need the third order version of PLUTO and a finer mesh.

remain at the *equilibrium solution* up to *machine accuracy* throughout the entire simulation. No spurious Kelvin-Helmhotz instabilities are generated, since the equilibrium pressure and velocity are exactly maintained for arbitrary long simulation times.

Finally, we compare our result at time $t = 15$ with the results obtained

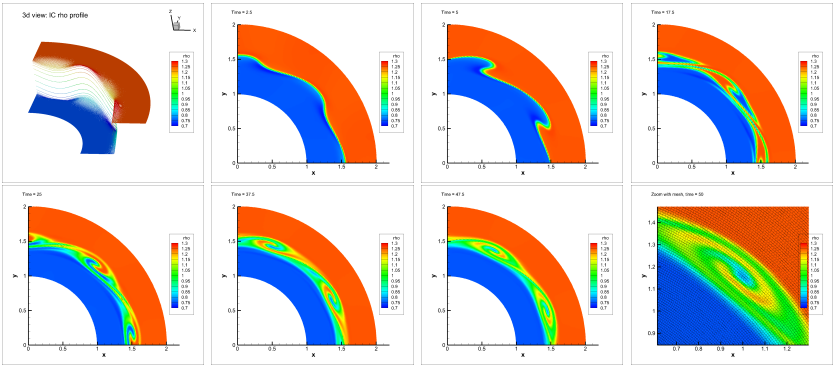


Figure 4.14: Kelvin-Helmholtz instabilities I. In the panel we show the evolution of the imposed periodic perturbations at different times. The results have been obtained with our second order well balanced ALE Osher-Romberg scheme over a grid with 100×200 control volumes.

with PLUTO, refer to Figure 4.13. For the visualization we have always used the software Tecplot and the same colormap; even if the results look similar, one can notice that to obtain the same resolution of our code (left image of the panel) we need the third order version of PLUTO and a finer mesh (last image of the panel). We stress that our code maintains u and P up to machine precision, whereas PLUTO produces standard numerical errors, see Table 4.4.

4.2.8 Keplerian disc with Kelvin-Helmholtz instabilities I

Let us consider an equilibrium solution which satisfies the equilibrium constraints in (3.19)-(3.20) so that

$$\rho_E = \rho_0 + \rho_1 \tanh\left(\frac{r - r_m}{\sigma}\right), \; u_E = 0, \; v_E = \sqrt{\frac{G m_s}{r}}, \; p_E = 1, \tag{4.33}$$

with $G = 1$, $m_s = 1$, $\rho_0 = 1$, $\rho_1 = 0.25$, $r_m = 1.5$ and $\sigma = 0.01$. It shows a steep gradient in the density for $r \to 1.5$. We consider as computational domain a ring sector with radius $r \in [1, 2]$ and $\varphi \in [0, \pi/2]$. For the boundary conditions we exploit the exact solution when $r = 1, 2$, and we impose periodic boundary conditions for $\varphi = 0, \pi/2$.

As confirmed by the previous tests, our well balanced ALE scheme is able to maintain the equilibrium up to machine precision for very long computational

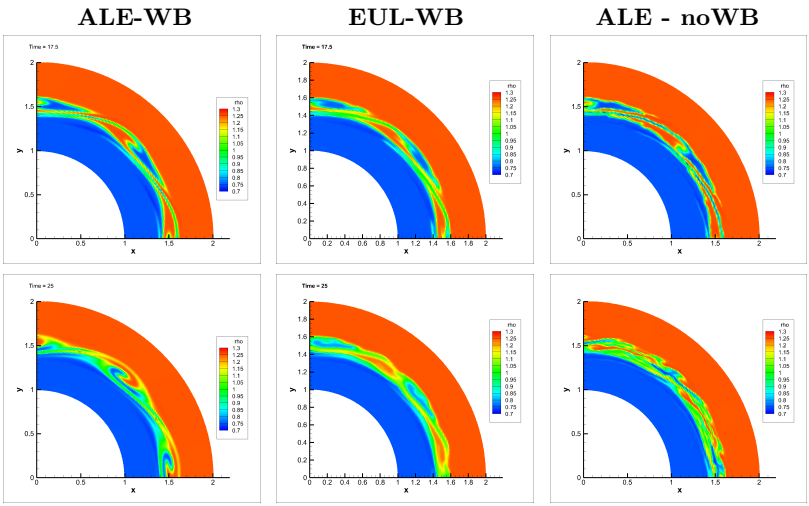


Figure 4.15: Kelvin-Helmholtz instabilities I. In the panel we show the obtained solution for the density profile at time $t = 17.5$ (first row) and time $t = 25$ (second row). The results presented in the first column have been obtained using the Osher-Romberg well balanced ALE scheme. The ones in the second column have been obtained using a zero velocity mesh (Eulerian case) and the well balanced Osher-Romberg scheme. The third column is obtained with a standard nonconforming ALE scheme (i.e. using the ALE Osher type flux without well balancing). One can apreciate that it is really the coupling between the ALE and the well balancing that allows to achieve this high resolution.

times. So we can study with high accuracy the evolution of the perturbations over the density, the radial velocities and the pressure prescribed by the following initial condition

$$\begin{cases} \rho = \rho_E + A\rho_0 \sin(k\varphi)\exp\left(-\frac{(r-r_m)^2}{s}\right), \\ u = u_E + A\sin(k\varphi)\exp\left(-\frac{(r-r_m)^2}{s}\right), \quad v = v_E, \\ p = p_E + A\sin(k\varphi)\exp\left(-\frac{(r-r_m)^2}{s}\right), \end{cases} \tag{4.34}$$

with $A = 0.1$, $k = 8$, $s = 0.005$. The computational results are depicted in Figure 4.14. In particular, for this flow configuration with physical perturbations in all flow quantities we observe the appearance of Kelvin-Helmholtz instabil-

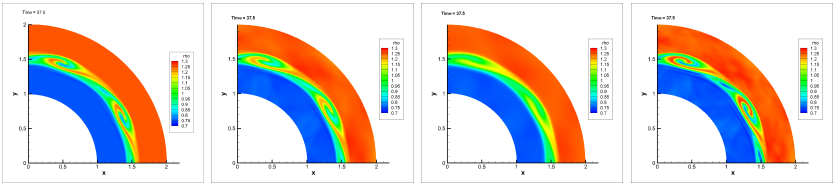


Figure 4.16: Kelvin-Helmholtz instabilities I. Method comparison at time $t = 37.5$. The first image is obtained with our code and 100×200 elements. The second and the third one with PLUTO, 100×200 elements and respectively a second order scheme with mc_lim limiter and a third order scheme and minmod_lim limiter. The last image is obtained with the third order version of PLUTO using mc_lim and 200×400 elements. All images are drawn with the same color map.

ities and a very good resolution of the developing vortices, which is achieved thanks to the ALE technique and despite the rather coarse mesh of 100×200 elements used here.

Moreover we have compared our well balanced ALE scheme with a well balanced *Eulerian* method on a fixed grid, which appears to be quite diffusive, and a *not* well balanced ALE scheme, which produces visible spurious oscillations in the density profile. The results are presented in Figure 4.15 and, once again, they show that it is indeed the coupling between the well balanced techniques and the moving mesh framework that allows to achieve a high resolution on small perturbations around an equilibrium solution for very long computational times.

We also compare our numerical results at time $t = 37.5$ with those obtained by PLUTO, see Figure 4.16. In order to obtain the same accuracy of our new second order well balanced Osher Romberg ALE scheme (left image of the panel) one needs the third order version of PLUTO on a finer mesh (last image of the panel).

4.2.9 Keplerian disc with Kelvin-Helmholtz instabilities II

We finally consider another equilibrium solution which satisfies the equilibrium constraints in (3.19)-(3.20) and which reads

$$\rho_E = r, \quad u_E = 0, \quad v_E = \sqrt{\frac{Gm_s}{r}}, \quad p_E = 1, \tag{4.35}$$

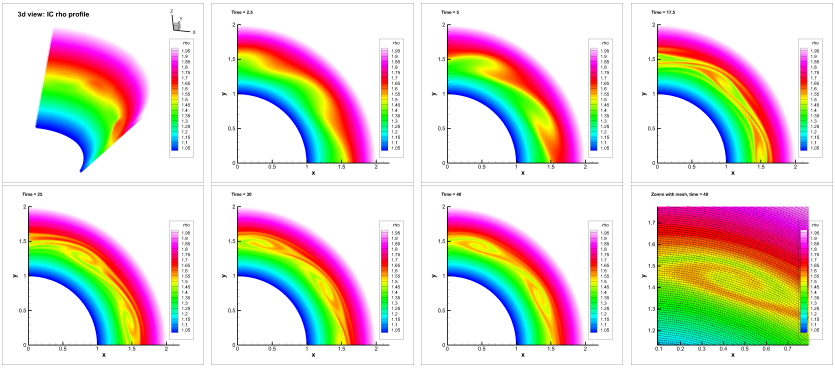


Figure 4.17: Kelvin-Helmholtz instabilities II. In the panel we show the evolution of the imposed periodic perturbations at different times. The results have been obtained with our second order Osher-Romberg scheme over a grid with 100×200 control volumes.

with $G = 1$, $m_s = 1$ and $r_m = 1.5$. With respect to the previous example, here the density profile is linear. However, also in this example we expect the Kelvin-Helmholtz instabilities to arise if some perturbations are added to the stationary profile. The computational domain and the boundary conditions are chosen as before. The initial condition used in this test problem reads

$$\begin{cases} \rho = \rho_E + A\sin(k\varphi)\exp\left(-\frac{(r-r_m)^2}{s}\right), \\ u = u_E + A\sin(k\varphi)\exp\left(-\frac{(r-r_m)^2}{s}\right), \quad v = v_E, \\ p = p_E + A\sin(k\varphi)\exp\left(-\frac{(r-r_m)^2}{s}\right), \end{cases} \tag{4.36}$$

with $A = 0.1$, $k = 8$, $s = 0.005$, i.e. we are again solving a problem that is close to an equilibrium and therefore difficult to solve with standard numerical techniques that are not well balanced. The computational results are depicted in Figure 4.17. Again we observe the appearance of Kelvin-Helmholtz instabilities that are well resolved also on a rather coarse mesh, without any visible spurious numerical oscillations. Finally, we compare once again our code with results obtained with PLUTO, refer to Figure 4.18. A similar resolution of the vortices is obtained with our second order code and the third order version of PLUTO with a finer mesh (refer to Section 4.2.5 for the details on the PLUTO configuration we have chosen). In this case we want to underline also that

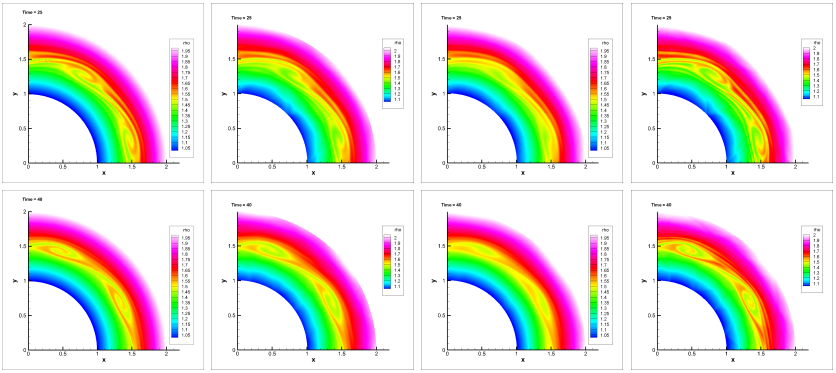


Figure 4.18: Kelvin-Helmholtz instabilities II. Method comparison at time $t = 25$ (first row) and at time $t = 40$ (second row). The first images are obtained with our code and 100×200 elements. The second and the third ones with PLUTO, 100×200 elements and using respectively a second order (mc_lim) and a third order (minmod_lim) scheme. The last images are obtained with PLUTO using a third order scheme (mc_lim) and 200×400 elements. All images are drawn with the same color map. The vortices have a similar resolution in the leftmost and rightmost images.

our code avoids other oscillations that instead can be noticed in the images obtained with PLUTO.

4.3 WB ALE for the shallow water equations

Coming back to the steady vortex test case presented in Section 4.3 we know that our method is able to maintain a high quality mesh even in the case of strong shear flows, to preserve the physical properties of the system (mass, momentum, energy) for very long computational times, and that it is very little dissipative for contact discontinuities. In this section we want to extend the algorithm in such a way that in addition it can preserve also exactly certain relevant and non-trivial equilibrium solutions. In particular, our interest is focused on the shallow water equations in polar coordinates given by (2.46), and to stationary solutions characterized by the equilibrium between the centrifugal and the gravitational forces. Since the two forces appear one in the flux and the other one in the source term, it is difficult to maintain the exact balancing until they are discretized with different techniques. So once again (as in Sec-

tion 3.2.3), we rewrite the source terms via non-conservative products so that all the terms connected with the equilibrium could be treated together. By adding as auxiliary variables the radius r and the bottom topography b such that the free surface is $\eta(r,\varphi) = b + h(r,\varphi)$, our equations can be cast in form (1.1) with

$$\mathbf{Q}=\begin{pmatrix} rh \\ rhu_r \\ rhu_\varphi \\ rb \\ r \end{pmatrix}, \quad \mathbf{f}=\begin{pmatrix} rhu_r \\ rhu_r^2 \\ rhu_r u_\varphi \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{g}=\begin{pmatrix} hu_\varphi \\ hu_r u_\varphi \\ hu_\varphi^2 + \frac{1}{2}gh^2 \\ 0 \\ 0 \end{pmatrix},$$
$$\mathbf{S}=\mathbf{0}, \qquad \mathbf{B}_1 \cdot \nabla \mathbf{Q}=\begin{pmatrix} 0 \\ grh\frac{\partial \eta}{\partial r} - hu_\varphi^2 \frac{\partial r}{\partial r} \\ hu_r u_\varphi \frac{\partial r}{\partial r} \\ 0 \\ 0 \end{pmatrix}, \qquad \mathbf{B}_2 \cdot \nabla \mathbf{Q} = \mathbf{0} \tag{4.37}$$

4.3.1 First order WB ALE scheme

In what follows we explain how to adapt the first order Osher-Romberg scheme proposed for the Euler equations to (4.37), in particular with the aim of preserving equilibrium solutions such that

$$u_r = 0, \quad \frac{\partial u_r}{\partial \varphi} = \frac{\partial u_\varphi}{\partial \varphi} = \frac{\partial \eta}{\partial \varphi} = 0, \quad \text{and} \quad \frac{\partial \eta}{\partial r} = \frac{u_\varphi^2}{gr}. \tag{4.38}$$

For the domain discretization let us consider the same setting introduced in Section 4.1.1, and for the basic form of a path-conservative ALE scheme refer to (4.5). The core of the scheme is the design of the well balanced space-time flux function, whose final expression is

$$\begin{aligned} \tilde{\mathbf{D}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} = \; & \frac{1}{2}\left(\tilde{\mathbf{F}}(\mathbf{q}^+) + \tilde{\mathbf{F}}(\mathbf{q}^-) + \mathcal{B}_{ij}\left(\mathbf{q}^+ - \mathbf{q}^-\right)\right) \cdot \tilde{\mathbf{n}}_{ij} \\ & - \frac{1}{2}\mathcal{V}_{ij}\left(\mathbf{q}^+ - \mathbf{q}^-\right), \end{aligned} \tag{4.39}$$

with the same notation of Section 4.1.2.1. We can also follow the same reasoning of Section 4.1.2.1 regarding the well balancing in the angular direction and Sections 3.3.1.1 - 4.1.2.1 for the Osher-Romberg viscosity term. So the

missing ingredients for the first order scheme are the well balanced path and the definition of $\mathcal{B}_{ij}(q^+ - q^-)$.

Here we propose a family of paths that is connected to the known equilibrium profiles for the free surface η and the angular velocity u_φ, whereas for the other variables the segment path is sufficient (because they assume simple profiles at the equilibrium). Let $\Phi^E(s; \mathbf{q}_E^-, \mathbf{q}_E^+)$ be a reparametrization of a stationary solution given by (4.38) that connects the two equilibrium states $\mathbf{q}_E^-$ with $\mathbf{q}_E^+$, then we define $\Phi(s; \mathbf{q}^-, \mathbf{q}^+)$ as follows

$$\Phi(s; \mathbf{q}^-, \mathbf{q}^+) = \Phi^E(s; \mathbf{q}_E^-, \mathbf{q}_E^+) + \Phi^f(s; \mathbf{q}_f^-, \mathbf{q}_f^+), \tag{4.40}$$

where $\mathbf{q}_f^- = \mathbf{q}^- - \mathbf{q}_E^-$ and $\mathbf{q}_f^+ = \mathbf{q}^+ - \mathbf{q}_E^+$ and

$$\Phi^f(s; \mathbf{q}_f^-, \mathbf{q}_f^+) = \mathbf{q}_f^- + s(\mathbf{q}_f^+ - \mathbf{q}_f^-).$$

That is Φ^f is a segment path on the *fluctuations* with respect to a stationary solution. With this choice, it is clear that if $\mathbf{q}^-$ and $\mathbf{q}^+$ lie on the same stationary solution satisfying (4.38), then $\mathbf{q}_f^- = \mathbf{q}_f^+ = \mathbf{0}$ and Φ, reduces to Φ^E. In such situation we have that $\mathbf{f}(\mathbf{q}^+) = \mathbf{f}(\mathbf{q}^-) = \mathbf{0}$ and

$$\mathcal{B}_{ij}(\mathbf{q}^+ - \mathbf{q}^-) = \int_0^1 \mathbf{B}_1(\Phi(s; \mathbf{q}^-, \mathbf{q}^+))\, \tilde{n}_r\, \frac{\partial \Phi^E}{\partial s}(s; \mathbf{q}^-, \mathbf{q}^+) ds = \mathbf{0}.$$

Therefore

$$\mathbf{f}(\mathbf{q}^+) - \mathbf{f}(\mathbf{q}^-) + \mathcal{B}_{ij}\left(\mathbf{q}^+ - \mathbf{q}^-\right) = \mathbf{0}.$$

Let us now define $\mathcal{B}_{ij}\left(\mathbf{q}^+ - \mathbf{q}^-\right)$ in the general case, where $\mathbf{q}^+$ and $\mathbf{q}^-$ do not lie on a stationary solution. In such case we have that

$$\mathcal{B}_{ij}\left(\mathbf{q}^+ - \mathbf{q}^-\right) = \left(b_1^{ij},\ b_2^{ij},\ b_3^{ij},\ b_4^{ij},\ b_5^{ij}\right)^T. \tag{4.41}$$

It is clear from the definition of $\mathbf{B}$ that $b_1^{ij} = b_4^{ij} = b_5^{ij} = 0$. What is interesting is the discretization of the second term that can be rewritten as

$$\left(grh\frac{\partial \eta}{\partial r} - hu_\varphi^2 \frac{\partial r}{\partial r}\right)\tilde{n}_r = \left(grh\frac{\partial \eta}{\partial r} - grh\left[\int \frac{u_\varphi^2}{rg}\, dr \pm \int \frac{u_{\varphi,E}^2}{rg}\, dr\right]_r\right)\tilde{n}_r, \tag{4.42}$$

where $u_{\varphi,E}$ is any known profile for the angular velocity at the equilibrium; moreover call $\zeta(r)$ a primitive of $\frac{u_{\varphi,E}^2}{rg}$, i.e. $\zeta(r) = \int \frac{u_{\varphi,E}^2}{rg}\, dr$. In this way we obtain that

$$b_2^{ij} = \int_0^1 \left(g\Phi_{rh}(s)\frac{\partial \Phi_\eta(s)}{\partial s} - g\Phi_{rh}(s)\frac{\Phi_A(s)}{rg}\frac{\partial \Phi_r(s)}{\partial s} \right)\tilde{n}_r\, ds,$$

where for variables r and rh we can employ a standard segment path to connect the left and the right states

$$\Phi_r(s) = \Phi_r(s; r^-, r^+) = r^- + s\,(r^+ - r^-),$$
$$\Phi_{rh}(s) = \Phi_{rh}(s; (rh)^-, (rh)^+) = (rh)^- + s\,((rh)^+ - (rh)^-).$$

Instead, following the idea in (4.40) and considering the terms in (4.42) we define

$$\Phi_A(s) = \Phi_A(s; u_\varphi^-, u_\varphi^+) = \Phi^E_{\zeta_r}(s) + \frac{\Phi^f_{u_\varphi}(s)}{rg} \tag{4.43}$$

which exploits the reparametrization of $\zeta(r)$ at the equilibrium and approximates with a segment path the fluctuations of the angular velocity

$$\Phi^f_{u_\varphi}(s) = \Phi^f_{u_\varphi}(s; u^-_{\varphi,f}, u^+_{\varphi,f}) = \frac{1}{rg}\left(u^-_{\varphi,f} + s\,(u^+_{\varphi,f} - u^-_{\varphi,f})\right).$$

A similar approach is used for $\Phi_\eta(s)$ defined as

$$\Phi_\eta(s) = \Phi_\eta(s; \eta^-, \eta^+) = \Phi^E_\eta(s) + \Phi^f_\eta(s).$$

Taking into account that

$$\int_0^1 \left(g\Phi_{rh}(s)\frac{\partial \Phi^E_\eta(s)}{\partial s} - g\Phi_{rh}(s)\frac{\Phi^E_{\zeta_r}(s)}{rg}\frac{\partial \Phi_r(s)}{\partial s} \right) \tilde{n}_r\, ds = 0,$$

b_2^{ij} could be rewritten as follows

$$b_2^{ij} = \int_0^1 \left(g\Phi_{rh}(s)\frac{\partial \Phi^f_\eta(s)}{\partial s} - g\Phi_{rh}(s)\frac{\Phi^f_{u_\varphi}(s)}{rg}\frac{\partial \Phi_r(s)}{\partial s} \right) \tilde{n}_r\, ds. \tag{4.44}$$

Note that

$$\frac{\partial \Phi^f_\eta(s)}{\partial s} = \eta_f^+ - \eta_f^- = \Delta\eta_f \quad \text{and} \quad \frac{\partial \Phi_r(s)}{\partial s} = r^+ - r^- = \Delta r,$$

therefore b_2^{ij} reduces to

$$b_2^{ij} = \left(g\,(rh)_{ij}\,\Delta\eta_f - g\,(rh)_{ij}\left(\frac{u_\varphi^2 - (u_\varphi^E)^2}{rg}\right)_{ij}\Delta r \right)\tilde{n}_r$$

where we have employed the mid point rule to approximate the integrals and the following notation holds $(\cdot)_{ij} = (\cdot_i + \cdot_j)/2$. Finally, term b_3^{ij} could be

approximated in the same way. Nevertheless, as this terms explicitly depends on u_r and we are interested to preserve equilibria with $u_r = 0$, a more simple approach could be used. Thus, b_3^{ij} could be defined as

$$b_3^{ij} = \Big((rhu_r)_{ij} (u_\varphi)_{ij} \Delta r \Big) \tilde{n}_r \,, \tag{4.45}$$

which vanishes when $u_r = 0$. For any other detail of the scheme one can refer to the previous sections and adapt the method introduced for the Euler equations to this new context. Indeed the method is very general and easy to be generalized to different sets of equations and equilibria.

4.4 Numerical results

In this section, first we want to show that the well balanced method works in general situations and not only close to the equilibria of the system. In this way, it will be clear that it can be applied in any context without corrupting the standard characteristics of the scheme, and it will perform better than classical schemes when near to a prescribed equilibrium. Then, we will see that the coupling between our nonconforming techniques with the well balanced strategy allow us to study the vortex flow of Section 2.4.5 even for longer periods of time.

4.4.1 Riemann problem

To show the correctness of our method in the presence of shocks we solve a classical Riemann problem with our well balanced Osher-Romberg ALE scheme. We consider the system of equation in (2.46), and as computational domain $[r, \varphi] = [1, 5] \times [0, 2\pi]$. We impose the following initial conditions

$$\begin{cases} h = 1, \text{ if } r < r_m, \quad h = 0.125, \text{ if } r \geq r_m, \\ u_r = u_\varphi = 0 \end{cases} \tag{4.46}$$

with $r_m = 3$. The results at the final computational time $t_f = 0.4$ are shown in Figure 4.19, where we report a cut along $\varphi = \pi/4$. The method, even if it is set up to preserve the smooth stationary profile described in Section 2.4.5, converges properly to the reference solution of this problem, despite the presence of discontinuities.

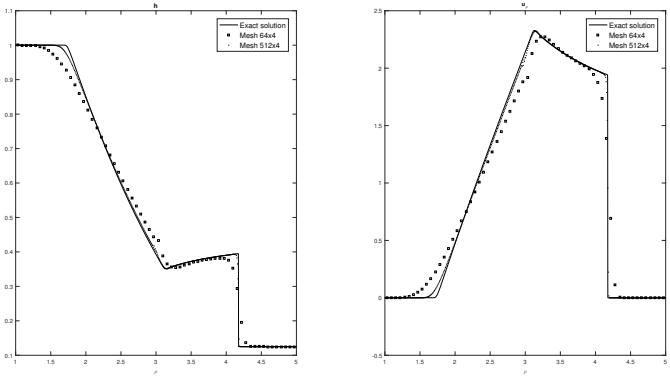


Figure 4.19: Comparison between the exact and the numerical solution for the Riemann problem. The numerical solution is obtained with the well balanced scheme of order one with two different meshes (a coarser and a finer one). On the left we show the water level h and on the right the radial velocity u_r for $r \in [1, 5]$ at a fixed angle $\varphi = \frac{\pi}{4}$.

4.4.2 Steady vortex in equilibrium

Test A. Let us consider again the test case of Section 2.4.5, with the initial condition of (2.55). The coupling between our novel nonconforming ALE scheme together with the well balanced techniques gives us, even after a very long computational time, a good mesh quality (see Figure 4.20) and a numerical solution equal to the exact one up to machine precision (refer to Table 4.5).

Note that we have employed a mesh of squares with the constraints that interfaces lie over straight lines with constant radius. This automatically implies that each square of the mesh has two edges parallel to the $\varphi-$axis: over this kind of edges the $\mathbf{g}$ component of the flux does not play any role, and so the method is well balanced simply because the $\mathbf{f}$ component of the flux is zero for stationary vortex-type solutions and (4.41) has been proved to be discretized in the correct way. The other two edges are parallel between them, so at the equilibrium, fluxes through them cancel.

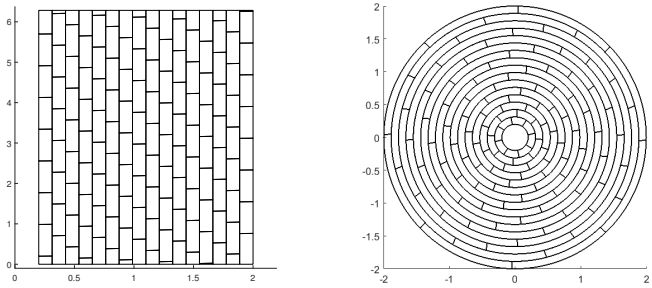


Figure 4.20: Stationary vortex in equilibrium obtained with well-balanced ALE schemes on moving nonconforming meshes. The mesh is shown at time $t = 200$. On the left we report the grid in polar coordinates where the shear discontinuities lie over straight lines. On the right the corresponding grid is shown in Cartesian coordinates.

Test B. Moreover, to show that the method is able to preserve *any* known stationary solution that satisfies the constraint in (4.38), we have performed a similar test but starting from a different stationary condition

$$h(r, \varphi, 0) = \frac{r^2}{2g}, \qquad u_r(r, \varphi, 0) = 0, \qquad u_\varphi(r, \varphi, 0) = r, \tag{4.47}$$

over the same computational domain $\Omega(r, \varphi) = [0.2, 2] \times [0, 2\pi]$. Even in this case the numerical solution remains close to the exact one up to machine precision for very long times, as also shown in Table 4.5.

Table 4.5: Stationary vortex in equilibrium. Maximum error on the water level h between the exact and the numerical solution obtained with the first order well balanced nonconforming ALE method. In the left column we show the error for Test A with finer and finer meshes with a fixed final time, in the central column we choose a coarse mesh and show the error for longer and longer times. In the right column, the results for Test B are shown.

Test A				Test B	
$tend = 10$		points 16×8		points 16×8	
points	error	time	error	time	error
12×6	1.42E-14	10	1.28E-14	10	2.11E-13
16×8	1.28E-14	50	3.74E-14	100	4.84E-13
24×12	3.04E-14	150	4.02E-14	150	3.25E-13
36×18	6.68E-14	200	4.88E-14	200	2.62E-13

5 Well balancing: GPU Parallel implementation

In this chapter we apply our well balanced techniques to a simplified compressible two-phase model for complex non-hydrostatic free surface flows. The well balancing guarantees the preservation of equilibrium solutions between air and water even on coarse meshes and in the presence of obstacles. Moreover a very efficient parallel implementation on a GPU-based platform ensures high performances on very fine grids, necessary to capture all the details of complex breaking waves.

Here, we first introduce in Section 5.1 our physical model given by a special case of the more general Baer-Nunziato system for compressible multi-phase flows. Then in Section 5.2-5.3 we present our well balanced scheme in the particular case of a Cartesian mesh and of a parallel implementation in CUDA. Finally, in Section 5.4 we show our numerical results and we exhibit the high efficiency of our code.

5.1 Physical model for complex free surface flow

We propose a simple and efficient two-phase interface-capturing algorithm for the simulation of complex non-hydrostatic free surface flows. The physical model is given by a special case of the more general Baer-Nunziato model for compressible multi-phase flows, which was introduced for the first time by Baer and Nunziato in [4]. This system does not adopt the classical assumptions of the shallow water equations, so it does not neglect the vertical accelerations and the resulting pressure distribution. In this way it results to be remarkably more accurate to model phenomena as breaking waves, drops, or the initial stages of dambreak flows, where vertical acceleration plays an important role.

In this section we present the physical model starting from the original Baer-Nunziato system and introducing some appropriate simplifications that allow us to reduce the number of equations from seven to three, as was done for the first time by Dumbser in [65]. Then we rewrite the source terms via non-conservative products.

The original Baer-Nunziato system with gravity effects reads

$$\begin{cases} \partial_t(\alpha_1\rho_1) + \nabla\cdot(\alpha_1\rho_1\mathbf{u}_1) = 0 \\ \partial_t(\alpha_1\rho_1\mathbf{u}_1) + \nabla\cdot(\alpha_1(\rho_1\mathbf{u}_1\otimes\mathbf{u}_1 + p_1\mathbf{I})) = p_I\nabla\alpha_1 + \alpha_1\rho_1\mathbf{g} \\ \partial_t(\alpha_1\rho_1E_1) + \nabla\cdot(\alpha_1\mathbf{u}_1(\rho_1E_1 + p_1)) = -p_I\partial_t\alpha_1 \\ \partial_t(\alpha_2\rho_2) + \nabla\cdot(\alpha_2\rho_2\mathbf{u}_2) = 0 \\ \partial_t(\alpha_2\rho_2\mathbf{u}_2) + \nabla\cdot(\alpha_2(\rho_2\mathbf{u}_2\otimes\mathbf{u}_2 + p_2\mathbf{I})) = p_I\nabla\alpha_2 + \alpha_2\rho_2\mathbf{g} \\ \partial_t(\alpha_2\rho_2E_2) + \nabla\cdot(\alpha_2\mathbf{u}_2(\rho_2E_2 + p_2)) = -p_I\partial_t\alpha_2 \\ \partial_t\alpha_1 + \mathbf{u}_I\cdot\nabla\alpha_1 = 0, \end{cases} \tag{5.1}$$

where α_j is the volume fraction of phase number j with $\alpha_1 + \alpha_2 = 1$, ρ_j is the fluid mass density, $\mathbf{u}_j = (u_j, v_j)$ the velocity vector, p_j the pressure and $\rho_j E_j$ the total energy per mass unit of phase number j, respectively. Moreover, $\mathbf{g} = (0, -g)$ is the vector of gravity acceleration, $g = 9.81$. The model must be closed by the equations of state (EOS) for each phase that link the pressure p_j to the density and the internal energy, and furthermore the model requires a proper choice of the interface velocity $\mathbf{u}_I$ and the interface pressure p_I. Baer and Nunziato proposed the following choice

$$p_I = p_2, \quad \text{and} \quad \mathbf{u}_I = \mathbf{u}_1, \tag{5.2}$$

which we also use here.

As explained in details in [65], the three-equation model we will study is based on the following *simplifications*. The first assumption is that all pressures are relative pressures with respect to the atmospheric reference pressure p_0, hence we can define $p_0 = 0$. Second, the gas surrounding the liquid is supposed to remain always at atmospheric reference conditions, i.e. the gas pressure is

$$p_2 = p_0 = 0 = \text{const}, \tag{5.3}$$

which is a standard assumption for free surface flows in fluid mechanics, see for example [154]. It is based on the fact that for low Mach number flows the pressure fluctuations p_0 are approximately proportional to $\rho\mathbf{u}^2$ according to Bernoulli's law and since the liquid density is several orders of magnitude larger than the density of the gas ($\rho_1 \gg \rho_2$) the pressure fluctuations of the gas phase are much smaller than the pressure fluctuations in the liquid phase ($p_2' \ll p_1'$). We therefore can neglect all evolution equations related to the gas phase $j = 2$. Furthermore, according to (5.2)-(5.3) the interface pressure automatically results $p_I = p_2 = p_0 = 0$. This is consistent with the usual

standard assumption for free surface flows, where, at the free surface of the liquid, atmospheric reference pressure boundary conditions are imposed. Also the choice of the interface velocity $\mathbf{u}_I = \mathbf{u}_1$ according to (5.2) is consistent, since the interface will obviously propagate with the speed of the liquid phase. Third, the pressure of the liquid is computed by the Tait equation of state [12]: the key idea is that according to the first assumption it yields a relative pressure with respect to the atmospheric reference pressure ($p_0 = 0$). We therefore have

$$p_1 = k_0 \left(\left(\frac{\rho_1}{\rho_0} \right)^{\gamma} - 1 \right), \tag{5.4}$$

where k_0 is a constant that governs the compressibility of the fluid and hence the speed of sound, ρ_1 is the liquid density, ρ_0 is the liquid reference density at atmospheric standard conditions and γ is a parameter that is used to fit the EOS with experimental data. Since the EOS (5.4) does not depend explicitly on the internal energy, also the liquid energy equation can be dropped.

So, we obtain the following reduced *three-equation model*

$$\begin{cases} \partial_t (\alpha\rho) + \nabla \cdot (\alpha\rho\mathbf{u}) = 0 \\ \partial_t (\alpha\rho\mathbf{u}) + \nabla \cdot (\alpha (\rho\mathbf{u} \otimes \mathbf{u} + p\mathbf{I})) = \alpha\rho\mathbf{g} \\ \partial_t \alpha + \mathbf{u} \cdot \nabla\alpha = 0, \end{cases} \tag{5.5}$$

where the index has been dropped to simplify the notation. We work in a two-dimensional framework $\mathbf{x} = (x, z)$ where z indicates the gravity direction. We underline that the free surface here is captured automatically by the volume fraction function α which is evolved in time by the last equation (advection equation) and allows us to distinguish between the portion of the domain occupied by the liquid Ω_ℓ, with $\alpha \sim 1$, and the one occupied by the gas Ω_a, with $\alpha \sim 0$.

An important *family of equilibria* of (5.11) consists in the water at rest solution given by

$$\mathbf{u} = \mathbf{0}, \quad \alpha = \text{ const}, \tag{5.6}$$

and density and pressure obtained through the momentum equation in $z-$direction as follows

$$\frac{\partial p}{\partial z} = \frac{\partial}{\partial z} k_0 \left(\left(\frac{\rho(z)}{\rho_0} \right) - 1 \right) = -\rho(z) g, \tag{5.7}$$

which for the particular case $\gamma = 1$ and $\rho(0) = \rho_0$ gives the simple solution

$$\rho(z) = \rho_0 \exp\left(-\frac{g\rho_0}{k_0} z \right), \tag{5.8}$$

and hence the pressure distribution

$$p(z) = k_0 \left(\exp\left(-\frac{g\rho_0}{k_0} z \right) - 1 \right). \tag{5.9}$$

To be able to preserve the equilibrium (5.7), that involves both terms in the flux and in the source, we need to rewrite all the involved terms in such a way that they could be treated together, so we add z as conserved variables through the following trivial equation

$$\frac{\partial z}{\partial t} = 0, \tag{5.10}$$

and we obtain our final non-conservative system

$$\begin{cases} \partial_t (\alpha\rho) + \nabla \cdot (\alpha\rho \mathbf{u}) = 0 \\ \partial_t (\alpha\rho u) + \frac{\partial}{\partial x} \left(\alpha\rho u^2 + \alpha p \right) + \frac{\partial}{\partial z} (\alpha\rho uv) = 0 \\ \partial_t (\alpha\rho v) + \frac{\partial}{\partial x} (\alpha\rho uv) + \frac{\partial}{\partial z} \left(\alpha\rho v^2 \right) + \frac{\partial}{\partial z} (\alpha p) + \alpha\rho g \frac{\partial z}{\partial z} = 0 \\ \partial_t \alpha + \mathbf{u} \cdot \nabla \alpha = 0, \\ \partial_t z = 0, \end{cases} \tag{5.11}$$

which can be cast in the general form (1.1) with

$$\mathbf{Q} = \begin{pmatrix} \alpha\rho \\ \alpha\rho u \\ \alpha\rho v \\ \alpha \\ z \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \alpha\rho u \\ \alpha\rho u^2 + \alpha p \\ \alpha\rho uv \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \alpha\rho v \\ \alpha\rho uv \\ \alpha\rho v^2 \\ 0 \\ 0 \end{pmatrix},$$

$$\mathbf{B}_1 \cdot \nabla \mathbf{Q} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ u \frac{\partial \alpha}{\partial x} \\ 0 \end{pmatrix}, \quad \mathbf{B}_2 \cdot \nabla \mathbf{Q} = \begin{pmatrix} 0 \\ 0 \\ \frac{\partial \alpha p}{\partial z} + \alpha\rho g \frac{\partial z}{\partial z} \\ v \frac{\partial \alpha}{\partial z} \\ 0 \end{pmatrix}, \quad \mathbf{S} = \mathbf{0}. \tag{5.12}$$

5.2 Well balancing

We solve the three-equation two-phase model (5.11) with our second order well balanced path-conservative Osher-Romberg method on a structured grid. We

remark that in this way we can preserve the equilibria for long times, which is, up to our knowledge, a novelty with respect to the literature.

In order to clarify how the Osher-Romberg scheme, already presented in details in Section 3.3.1.1, can be applied even in this context we have to specify the chosen path and the computation of the non trivial elements of $\mathcal{B}_{ij}$ in (4.6). For everything else (viscosity and second order reconstruction) one can follow the line given in Chapter 3-4.

The choice of the path is done as in (4.40) by summing up two parts: a reparametrization of the equilibrium profile (5.6)-(5.8)-(5.9) plus a segment path over the fluctuations.

For what concern the discretization of the non-conservative terms $\mathcal{B}_{ij}(\mathbf{q}_j - \mathbf{q}_i)$ we have only three non null components

$$\mathcal{B}_{ij}(\mathbf{q}^+ - \mathbf{q}-) = (0,0,0,b_4^x,0)\,n_x + \left(0,0,b_3^z,b_z^4,0\right)n_z. \tag{5.13}$$

The terms b_4^x and b_4^z can be written in a very easy way as

$$b_4^x = u_{ij}\Delta\alpha, \quad b_4^z = v_{ij}\Delta\alpha, \tag{5.14}$$

where $\Delta\alpha = \alpha_j - \alpha_i$ and we propose to use the Roe average velocity to compute u_{ij} and v_{ij}

$$u_{ij} = \frac{u_i\sqrt{\rho_i} + u_j\sqrt{\rho_j}}{\sqrt{\rho_i}+\sqrt{\rho_j}}, \quad v_{ij} = \frac{v_i\sqrt{\rho_i} + v_j\sqrt{\rho_j}}{\sqrt{\rho_i}+\sqrt{\rho_j}}. \tag{5.15}$$

The above expressions are so easy thank to the equilibrium constraint (5.6) that ensure $b_4^x = b_4^z = 0$ when $\mathbf{q}_i$ and $\mathbf{q}_j$ lie on the same stationary solution. Finally let us recover b_3^z which verifies

$$b_3^z = \int_0^1 \frac{\partial\Phi_\alpha(s)}{\partial s}\frac{\partial\Phi_p(s)}{\partial s} + \Phi_\alpha(s)\Phi_\rho(s)g\frac{\partial\Phi_z(s)}{\partial s}ds, \tag{5.16}$$

where $\Phi_z(s) = \Phi_z(s;z_i,z_j) = z_i + s(z_j - z_i)$, $\Phi_{\alpha,\rho,p}(s) = \Phi^E_{\alpha,\rho,p}(s) + \Phi^f_{\alpha,\rho,p}(s)$ respectively. Taking into account that

$$b_3^z = \int_0^1 \frac{\partial\Phi^E_\alpha(s)}{\partial s}\frac{\partial\Phi^E_p(s)}{\partial s} + \Phi^E_\alpha(s)\Phi^E_\rho(s)g\frac{\partial\Phi_z(s)}{\partial s}ds = 0 \tag{5.17}$$

b_3^z can be rewritten as follows

$$\begin{aligned} b_3^z \quad = \quad & \int_0^1\left(\frac{\partial\Phi^E_\alpha(s)}{\partial s}\frac{\partial\Phi^f_p(s)}{\partial s} + \frac{\partial\Phi^f_\alpha(s)}{\partial s}\frac{\partial\Phi^E_p(s)}{\partial s} + \frac{\partial\Phi^f_\alpha(s)}{\partial s}\frac{\partial\Phi^f_p(s)}{\partial s}\right)ds \\ & + \int_0^1\left(\Phi^E_\alpha(s)\Phi^f_\rho(s) + \Phi^f_\alpha(s)\Phi^E_\rho(s) + \Phi^f_\alpha(s)\Phi^f_\rho(s)\right)g\frac{\partial\Phi_z(s)}{\partial s}ds. \end{aligned} \tag{5.18}$$

In general, the integral term could be difficult to compute, therefore we propose to use a numerical quadrature formula. Here the mid-point rule is used. In this case, we define b_3^z as follows

$$\begin{aligned} b_3^z &= \Delta\alpha_{ij}^E \Delta p_{ij}^f + \Delta\alpha_{ij}^f \Delta p_{ij}^E + \Delta\alpha_{ij}^f \Delta p_{ij}^f \\ &\quad + \left(\alpha_{ij}^E \rho_{ij}^f + \alpha_{ij}^f \rho_{ij}^E + \alpha_{ij}^f \rho_{ij}^f\right) g\, \Delta z_{ij}, \end{aligned} \tag{5.19}$$

where

$$\begin{aligned} \Delta p_{ij}^{E,f} = p_j^{E,f} - p_i^{E,f}, \Delta\alpha_{ij}^{E,f} = \alpha_j^{E,f} - \alpha_i^{E,f}, \Delta z_{ij} = z_j - z_i \\ \alpha_{ij}^{E,f} = \frac{\alpha_i^{E,f} + \alpha_{ij}^{E,f}}{2}, \quad \text{and} \quad \rho_{ij}^{E,f} = \frac{\rho_i^{E,f} + \rho_{ij}^{E,f}}{2}. \end{aligned} \tag{5.20}$$

It is clear from the definition that $b_3^z = 0$ if $\mathbf{q}_i$ and $\mathbf{q}_j$ lie on the same stationary solution because all the fluctuation terms are zero.

For completeness we recover here the expression of the extended Jacobians whose eigenstructure is needed in the Osher-Romberg viscosity (3.49)

$$\begin{aligned} \mathbf{A}_1 = \frac{\partial \mathbf{f}}{\partial \mathbf{Q}} + \mathbf{B}_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ c^2 - u^2 & 2u & 0 & p - \rho c^2 & 0 \\ -uv & v & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \\ \mathbf{A}_2 = \frac{\partial \mathbf{g}}{\partial \mathbf{Q}} + \mathbf{B}_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ -uv & v & u & 0 & 0 \\ c^2 - v^2 & 0 & 2v & p - \rho c^2 & \alpha\rho g \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \end{aligned} \tag{5.21}$$

where the speed of sound c is defined as

$$c^2 = \frac{\partial p}{\partial \rho} = \frac{\gamma k_0}{\rho_0} \left(\frac{\rho}{\rho_0}\right)^{\gamma - 1} \tag{5.22}$$

In particular for $\alpha = 0$ we lose the hyperbolicity (it can be shown by computing the eigenvectors, see [65]), so we always choose the initial conditions for the volume fraction α as

$$\alpha(\mathbf{x}, 0) = \begin{cases} 1 - \varepsilon, & \text{if } \mathbf{x} \in \Omega_l \\ \varepsilon, & \text{if } \mathbf{x} \in \Omega_a \end{cases} \tag{5.23}$$

where $\varepsilon > 0$ is a small parameter.

To conclude, we point out that in order to avoid wasting computational time we use the second order version of the code only wherever $\alpha > 10\varepsilon$ otherwise, i.e. in pure gas cells, we switch to the first order method.

5.3 CUDA

The introduced system is particularly well suited to simulate breaking waves and dambreak problems, since the physical model is complete and the employed numerical method is of high order and exactly well-balanced for relevant equilibrium solutions. However to capture all the details of a complex free surface flow we need also a very fine mesh. Thus, in order to increase the effectiveness of our method, we have decided for a parallel implementation, choosing in particular a GPU-based platform that is even faster than parallel codes run on CPU.

The NVIDIA's CUDA (Compute Unified Device Architecture) framework is a hardware and software platform that allows to easily exploit the NVIDIA GPUs and parallelize many costly algorithms. We refer to the recent book [118] for a complete introduction to CUDA for Scientific Computing. The images of this section are indeed courtesy of its authors. Here we limit ourselves to a small introduction and to some notes on our specific implementation.

According to the CUDA framework, a GPU is viewed as part of a computing device, with their own Dynamic Random Access (DRAM) memory, which works as a coprocessor for a host which includes the CPU and its (DRAM) memory, see Figure 5.1. It is possible to copy data from the host DRAM memory (linked to CPU) to the DRAM memory of the device (linked to GPU) and viceversa.

The CUDA framework assumes a unified architectural view of the GPU where a GPU is formed by many multiprocessors, each one having multiple cores (see Figure 5.2). At any clock cycle, each core of the multiprocessor executes the same instruction, but operates on different data. The data which are processed in each multiprocessor can be stored in the *global memory*, that can be shared by all execution units but is slower than the other type of memory i.e. *registers* and *shared memory* which belong respectively to each core and each multiprocessor.

The multiprocessors of a GPU are specialized in the parallel execution of a huge number of CUDA threads. A CUDA thread represents a sequential computational task which executes an instance of a function and is executed

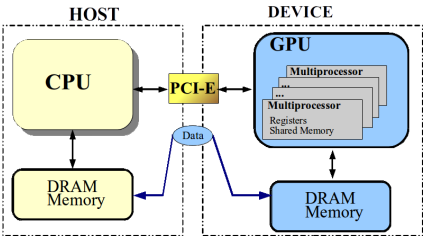


Figure 5.1: GPU device: coprocessor for the host system.

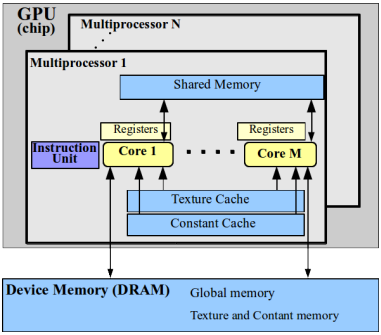


Figure 5.2: CUDA hardware model.

logically in parallel with respect to other CUDA threads (associated to the same function but operating on different data) on the cores of a GPU multiprocessor (see Figure 5.3). To specify the function to be executed by each thread on the GPU, the programmer must define a special C function, called CUDA *kernel*. A CUDA kernel is called from the CPU and is executed on the GPU by each CUDA thread. When a kernel is launched, every thread executes the same function but the processed data depend on the value of several built-in variables which identify the position of each particular thread, from which one can deduce for example the element or the edge on which the thread is operating. According to the data structure the threads are organized in a $1D$, $2D$ or $3D$ grids, see Figure 5.4. In this way data referring to neighbors cells in the physical problem will be grouped together even during the computation

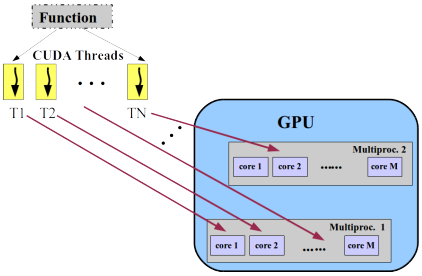


Figure 5.3: Execution of multiple CUDA threads associated with the same kernel.

contributing in accelerating the algorithm, since the interactions between neighbors are frequent and the proximity of the data reduces the memory accesses.

5.3.1 Notes on our implementation

Our algorithm is composed by three main kernel functions. The first one is used to reconstruct the polynomial representing the conserved variables inside each cell. Data are organized in a $2D$ grid and each thread works on one element of the mesh.

The second one is used to solve the Riemann problem at each interface, i.e. each edge of the mesh. We compute the numerical flux between two elements across an edge: the obtained quantity should be added (with different signs) to each of the two elements. As in any parallel implementation, it is important to guarantee that there are no superpositions, i.e. that we do not modify at the same time the same element. So we cannot simply launch the kernel function indiscriminately on all the edges at the same time: because each element has 4 edges and so many possibilities of superpositions. To overcome this problem the idea consists simply in subdividing the edges in four groups. Since the mesh is Cartesian we can distinguish the edges between vertical and horizontal ones and we can number each group incrementally form left to right and from the bottom up. The four groups are then obtained by considering: the vertical edges with odd numbering, the vertical edges with even numbering, the horizontal edges with odd numbering, the horizontal edges with even numbering. So the computation of the flux at the interfaces is subdivided into four kernel functions

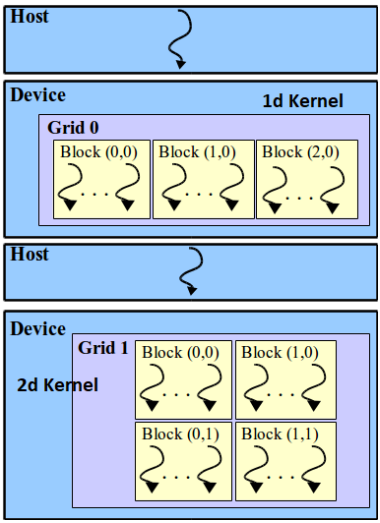


Figure 5.4: Structure of a CUDA Program. A kernel is a function to be executed by each thread (represented by the curved arrow). According to the data structure the threads are organized in a $1D$, $2D$ or $3D$ grid.

each one operating on a non intersecting set of data.

The third kernel function is organized as the first one and is used to update the solution in each cell from time t^n to time t^{n+1} summing up all the flux contributions.

5.4 Numerical Results

The numerical results presented here demonstrate that our method preserves the equilibria up to machine precision, that solves dambreak problems more accurately than shallow water in the first instances after the break, and that it is very efficient on fine meshes.

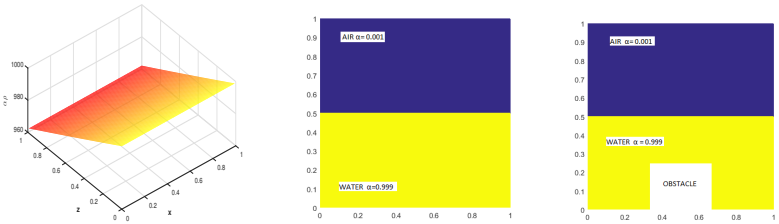


Figure 5.5: Density distribution at the equilibrium. Left: only water ($\alpha = 1$) with hydrostatic pressure distribution. Middle and right discontinuous α, interface water-air without (middle) and with (right) an obstacle. Velocity $\mathbf{v} = \mathbf{0}$.

5.4.1 Equilibria

First of all, we have considered an equilibrium profile characterized by (5.6)-(5.8)-(5.9), with $k_0 = 10^7$, $\rho_0 = 1$ and $\gamma = 1$ on a domain $\Omega = [-0.5, 0.5] \times [0, 1]$ in three different contexts.

a) In the first case the domain is totally covered by the liquid $\Omega_\ell = \Omega$ and $\alpha = 1$.

b) In the second case we consider both air and water, more precisely $\Omega_\ell = [-0.5, 0.5] \times [0, 0.5]$ with $\alpha = 0.999$ and $\Omega_a = [-0.5, 0.5] \times [0.5, 1]$ with $\alpha = 0.001$.

c) The third case coincides with the second one with the insertion of an obstacle in $[-0.125, 0.125] \times [0, 0.25]$.

The equilibrium profiles are shown in Figure 5.5. Our scheme preserves the equilibria up to machine precision: in Table 5.1 we report the maximum error between our numerical results, in the three cases, and the exact solution for what concerns the absolute value of the velocity (in order to compare the results with [65]); Machine precision is reached also on all the other quantities.

5.4.2 Dambreak

A very typical application for shallow water-type models is the so-called dambreak. It consists of the sudden collapse/removal of a vertical wall that separates two different piecewise constant states of water from each other. Since

Table 5.1: Maximum error between the numerical and the exact solution for the modulus of the velocity in the three considered cases. Equilibria are maintained up to machine precision for very long times.

time	$\|\|\mathbf{u}\|\|_\infty$		
1	3.2E-12	7.4E-14	5.1E-13
10	3.2E-12	8.0E-14	3.0E-12
100	3.3E-12	8.0E-14	3.3E-12
1000	3.4E-12	8.1E-14	3.4E-12

in the initial stages of dambreak flows, the classical shallow water assumption of small vertical velocities and accelerations does not hold, it is of interest to apply our more complete model to this well-studied phenomenon.

We consider the classical dambreak over a dry bed. We take a computational domain $\Omega = [-50, 50] \times [0, 4]$ covered with a very fine Cartesian mesh of $3 \cdot 10^6$ elements. At the initial time $t = 0$ the liquid is contained in $\Omega_\ell = [-50, 0] \times [0, 1.4618]$. The constants that characterize the problem are chosen to be $k_0 = 6.37 \cdot 10^5$, $\rho_0 = 1000$ and $\gamma = 1$ (in this way the Mach number is $M = 0.3$ with a maximum expected velocity $|\mathbf{u}| = 7.57$). The initial velocity is $\mathbf{u} = \mathbf{0}$ everywhere, and the initial hydrostatic pressure and density distributions in Ω_ℓ are given by

$$\begin{aligned} p(\mathbf{x}, 0) &= \rho_0 g(z_0(x) - z), \\ \rho(\mathbf{x}, 0) &= \rho_0 \left(\frac{\rho_0 g(z_0(x) - z) + k_0}{k_0} \right)^{(1/\gamma)}, \end{aligned} \tag{5.24}$$

where $z_0(x)$ is the initial vertical position of the free surface as function of x. In the air we impose $p(\mathbf{x}, 0) = 0$, and $\rho(\mathbf{x}, 0) = \rho_0 = 1000$. The boundary conditions are reflective wall on the left, bottom and right border of the computational domain and transmissive boundaries on the top.

The results presented in Figure 5.6 clearly show the superiority of our more complete model with respect to the classical shallow water assumptions. And after longer times the two methods perform equally well, which guarantees the reliability of our scheme both for short and long times.

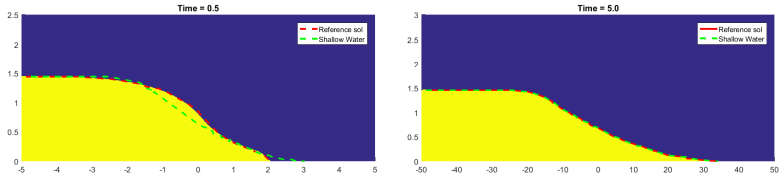


Figure 5.6: Dambreak problem. We compare our numerical results (water in yellow and air in blu) with a reference solution taken from Dumbser [65] in red, and the solution obtained using the shallow water model in green. On the left we show the results at time $t = 0.5$ and on the right the results at time $t = 5$. It is clear that at short time our method fits better the results with respect to shallow water and at longer times they perfectly agree. We have imposed that water corresponds to $\alpha \geq 0.5$ and air to $\alpha < 0.5$; with different threshold values one obtain very similar results.

5.4.3 Dambreak and impact against a vertical wall

The last test problem we want to show consists of a dambreak flow with successive impact against a vertical wall, which leads to the reflection of the incident wave and successive wave breaking. The setup of this test case is taken from [81] and [53]; the results could be compared also with [65]. The computational domain is $\Omega = [0, 3.2] \times [0, 1.8]$ with reflective boundaries on all the sided a part from the top where we impose transmissive boundary conditions. At $t = 0$ the liquid occupies $\Omega_\ell = [0, 1.2] \times [0.6]$, and density and pressure distributions are given once again by (5.24). We set $k_0 = 2.62 \cdot 10^5$ that gives a Mach number $M = 0.3$, and we cover the domain with a mesh of $8 \cdot 10^5$ elements. The numerical results are depicted in Figure 5.8. This test shows the capability of our scheme to capture complex free surface flows as those produced by breaking waves. In particular, since the free surface interface is given by the volume fraction α it is not constraint to be necessarily a a single-valued function as it is in standard shallow water context.

Efficiency

In order to prove the efficiency of our implementation we have run this test up to the final time $t_f = 1$ on finer and finer meshes using as graphics card a GeForce Titan Black, with the double precision performance option enabled. The number of volumes processed per seconds with respect to the total number of processed volumes is reported in Figure 5.7: we have reached the order of

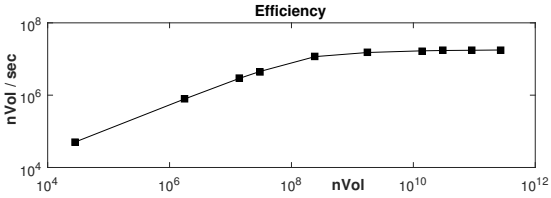


Figure 5.7: Impact against a vertical wall: efficiency. In the figure we show the number of volumes processed per seconds when employing finer and finer meshes to solve the dambreak problem of Section 5.4.3 up to a final time $t_f = 1$. (Note that with `nVol` on the x-axis we refer to the number of elements of the mesh multiplied by the number of timesteps.)

ten millions of volumes processed per second.

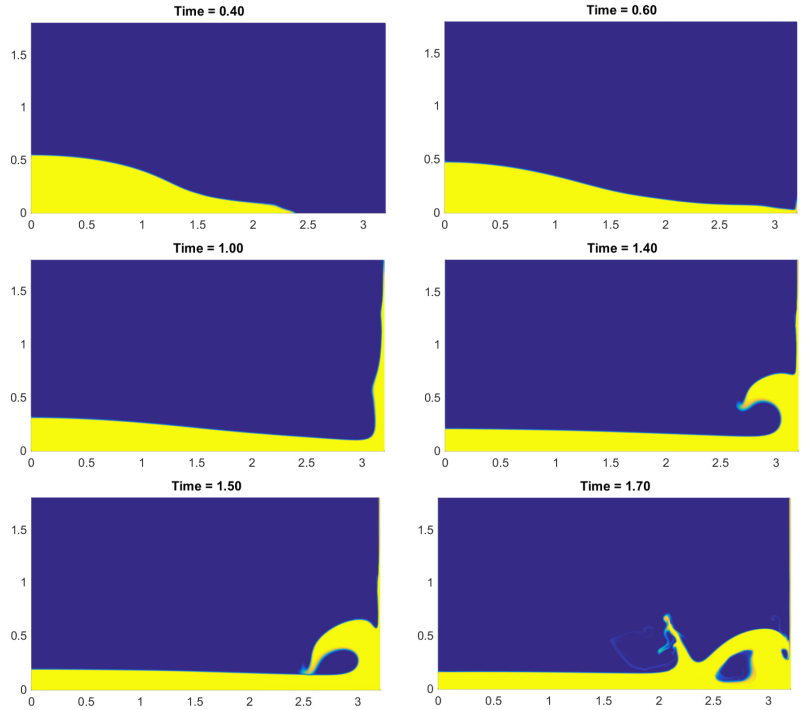


Figure 5.8: Impact against a vertical wall at different times. In the figure we depict the value of α which goes from $\alpha \sim 0$ (blue) i.e. air, to $\alpha \sim 1$ (yellow) i.e. liquid.

6 Angular Momentum preserving schemes for the compressible Euler equations

What we propose here is to *exploit* in various ways the *redundant conservation law* that can be written for the *angular momentum*, in order to increase the accuracy of the final results. Our study is based on considering an augmented Euler system of equations where we take into account both the inertial momentum conservation law as well as the angular momentum conservation law.

We present here three different *formulations*: the master-slave approach, the global-coupling and the local-coupling. In the master-slave approach we propose a straightforward discretization of the augmented system, where the angular momentum will have no influence on the other conserved variables. In the global-coupling the angular momentum is strongly coupled with the entire system: in this case we will assume the *a priori* knowledge of a fixed center of rotation. Finally, for the local-coupling we will explore a solution in which the rotation center varies locally. In these three approaches the number of conserved variables is $\nu = 5$ whereas for the initial Euler system $\nu = 4$.

The *validation* of the algorithms will be performed with three different types of test problems. The first one is standard and is used to compare the results for the three approaches (master-slave approach, global-coupling and local-coupling). It is a solid body rotation of a gas with constant density. Density and velocity have been chosen in such a way that also the angular momentum is constant. As computational domain we consider a ring centered in the origin with radius $r \in [1, 2]$ and the initial condition, that corresponds also to the stationary analytical solution, is the following

$$\rho = 1, \quad \mathbf{u} = \left(\frac{-y}{x^2 + y^2}, \frac{x}{x^2 + y^2} \right), \quad p = -\frac{1}{2(x^2 + y^2)} + 1, \quad w = 1. \tag{6.1}$$

In Table 6.1 with this test, we illustrate the interest of solving the angular momentum within an augmented system. The second type of test is made of variants of the first one, to illustrate specific features of the different methods. The third test problem is a generalization of the Kidder test problem [20] which

Table 6.1: In the central column (*without Angular Momentum Conservation* – without AMC) we report the error in L_2 norm between the analytical value of the angular momentum and the one obtained at later times by computing *a posteriori* $w = \mathbf{u} \wedge (\mathbf{x} - \mathbf{x_c})$, by solving a system of size $\nu = 4$. The errors reported in the right column (*with AMC*) are obtained with the master-slave approach (see Section 6.2), by solving an augmented system of size $\nu = 5$. We have employed a mesh of 1600 elements and both a method of order 1 and 2. This simple test proves that post-processing may generate important errors with respect to a direct computation of an augmented system.

	without AMC ($\nu = 4$)		with AMC ($\nu = 5$, master-slave)	
time	order 1	order 2	order 1	order 2
1	3.2	5.1E-3	2.6E-16	1.6E-16
5	7.2	3.0E-2	1.0E-15	8.2E-16
10	9.4	5.3E-2	2.1E-15	1.7E-15
20	-	1.0E-1	3.7e-15	3.2E-15

is emblematic of strong implosion in stars or inertial confinement fusion devices. In our study we extend the physics considered in the standard Kidder problem by adding non-zero angular velocity. To our knowledge, this new analytical solution in cylindrical coordinates which combines compression and rotation is original with respect to the literature.

The rest of the chapter is organized as follows. First in Section 6.1, we introduce the augmented system of equations. In Section 6.2 we introduce the master-slave formulation. Section 6.3 is devoted to the global-coupling algorithm. The principles of a center-detector are introduced in Section 6.4. Next, Section 6.5 is devoted to the local-coupling method. The coupling with the ALE techniques is investigated in Section 6.6. In Section 6.7, we study a modified Kidder problem with rotation, whose derivation details are given in the Appendix B.

6.1 Augmented Euler equations

Let us recall the standard Euler equations of compressible gas dynamics in two dimensions of space which represent a hyperbolic system

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0, \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = \mathbf{0}, \\ \partial_t (\rho e) + \nabla \cdot (\rho \mathbf{u} e + p \mathbf{u}) = 0. \end{cases} \tag{6.2}$$

Here ρ is the density, $\mathbf{u} = (u_1, u_2)$ is the velocity, e is the total energy density and p the pressure. For a perfect gas one has

$$p = (\gamma - 1) \left(\rho e - \frac{1}{2} \rho \, ||\mathbf{u}||^2 \right), \quad \gamma = \frac{c_p}{c_v} > 1, \tag{6.3}$$

where γ is the ratio between heats at constant pressure and volume, which is taken to be constant. In particular, we underline that the system states $\nu = 4$ conservation laws: one for the mass, two for the inertial momentum and the last for the energy. In any direction defined by the unit vector $\mathbf{n}$, (6.2) admits four eigenvalues: $u_n - c, u_n, u_n + c, u_n$, where c denotes the sound speed $c = \sqrt{\gamma p / \rho}$ and $u_n = \mathbf{u} \cdot \mathbf{n}$. From now on, for the sake of clarity, we detail everything in $2D$, but most of the ideas are general and can be easily extended to $3D$.

By choosing arbitrarily a special center point $\mathbf{x_c} = (x_c, y_c)$ the *angular momentum* can be defined as

$$w = \mathbf{u} \wedge (\mathbf{x} - \mathbf{x_c}), \qquad \text{where } \wedge \text{ denotes the vector product.} \tag{6.4}$$

Note that $\mathbf{x_c} = \mathbf{0}$ is a possible choice and it is adopted when the center of rotation of the studied system is unique and known for any time. For the other cases we will describe a technique to determine it. Straightaway, a *redundant conservation law* for the angular momentum can be easily derived from the equations of the inertial momentum, by computing

$$\begin{aligned} (y - y_c) \left(\partial_t (\rho u_1) + \partial_x (\rho u_1^2 + p) + \partial_y (\rho u_1 u_2) \right) \; - \\ (x - x_c) \left(\partial_t (\rho u_2) + \partial_x (\rho u_1 u_2) + \partial_y (\rho u_2^2 + p) \right), \end{aligned} \tag{6.5}$$

from which we obtain

$$\partial_t (\rho w) + \nabla \cdot \left(\rho \mathbf{u} w + p (\mathbf{x} - \mathbf{x_c})^\perp \right) = 0. \tag{6.6}$$

It is clear that, from an analytical point of view, this equation does not add any supplementary information to the system, since it is directly derived from the

other ones. But from a numerical point of view it provides extra information in particular in the case of rotating systems.

Thus one can define the augmented 2D Euler system as

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0 \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = 0 \\ \partial_t (\rho e) + \nabla \cdot (\rho \mathbf{u} e + p \mathbf{u}) = 0 \\ \partial_t (\rho w) + \nabla \cdot \left(\rho w \mathbf{u} + p (\mathbf{x} - \mathbf{x_c})^{\perp} \right) = 0. \end{cases} \tag{6.7}$$

The system is still hyperbolic with an extra eigenvalue equal to u_n.

In Sections 6.2, 6.3 and 6.5 we will reformulate this system in order to obtain numerical methods with different characteristics. All these augmented formulations can be cast in the form

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{Q}, \mathbf{x}) = \mathbf{S}(\mathbf{Q}), \quad \mathbf{x} \in \Omega(t) \subset \mathbb{R}^2, \quad \mathbf{Q}(t, \mathbf{x}) \in \mathbb{R}^{\nu}, \tag{6.8}$$

a special case of (1.1) where in particular the flux $\mathbf{F}(\mathbf{Q}, \mathbf{x})$ depends both on $\mathbf{Q}$ and on the position $\mathbf{x}$. They will be automatically discretized thanks to our up to second order accurate direct ALE PDE solver presented in Chapter 2. The ALE context allows us to first employ a zero velocity mesh in order to investigate what happens in the Eulerian framework; we will switch to a moving domain in Section 6.6.

6.2 Master-slave approach

The first method we propose consists simply in discretizing (6.7) by setting

$$\mathbf{Q} := \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ \rho e \\ \rho w \end{pmatrix} \quad \text{and} \quad \mathbf{F}(\mathbf{Q}, \mathbf{x}) := \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + \mathbf{I} p \\ \rho \mathbf{u} e + p \mathbf{u} \\ \rho w \mathbf{u} + p (\mathbf{x} - \mathbf{x_c})^{\perp} \end{pmatrix}$$

in (6.8) and use our general code. We refer to the fifth equation as slave because the Euler equations are solved independently, then angular momentum equation is computed *a posteriori*. This is regrettable because the benefits of angular momentum preservation cannot be seen by the other physical variables. However, by construction we automatically obtain the conservation in L_1 norm of w as adjoint variable. See Table 6.2 for some numerical results.

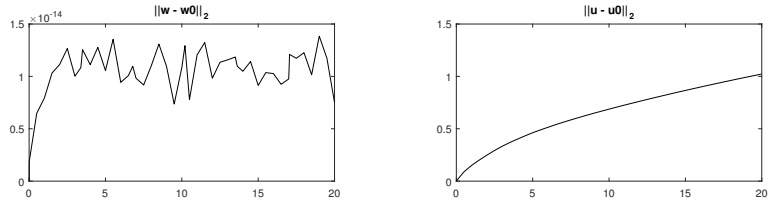


Figure 6.1: We consider the solid body rotation described in (6.1) and the master-slave formulation. We have used order one and a mesh of 1600 elements. On the left we show that the L_2 norm of the angular momentum remains stable for long times during the computation. However the L_2 norm of the velocity is rapidly dissipated and so the error grows. This means that with this formulation the velocity cannot enjoy any positive effects of including w in the system.

We would like to underline that the value assumed by w evolved through its conservation law may be different from the exact one and from the one computed *a posteriori* as $\mathbf{u} \wedge (\mathbf{x} - \mathbf{x_c})$: indeed the evolution of $\mathbf{u}$ and w are not linked in the master-slave formulation. For this reason the exact conservation of the total angular momentum does not imply directly an improvement on the final results. So, to measure the impact of our modified methods on the simulations we consider the following quantities

$$||w - w_0||_{L_2} = \sqrt{\sum_{i=1}^{N_E} |T_i| \, (w_i - w_{0,i})^2}, \quad ||\mathbf{u} - \mathbf{u}_0||_{L_2} = \sqrt{\sum_{i=1}^{N_E} |T_i| \, ||\mathbf{u}_i - \mathbf{u}_{0,i}||^2}. \tag{6.9}$$

The first one tells us if the value assumed by the variable w adjoint to the system coincides with the initial value, and the second quantity is an indicator of the effects of w on the entire system, in particular on the velocity. This choice is moreover justified by the fact that our tests are based on stationary solutions so the initial values coincide also with the exact ones.

In Figure 6.1 we show the values of the quantities in (6.9) for the test case (6.1) at different times. Even if the angular momentum is perfectly maintained, the velocity cannot enjoy any positive effects of including w in the system. Therefore, in Section 6.3 we propose a *coupled* formulation in such a way that the behavior of w and $\mathbf{u}$ are linked together. Before presenting the second approach which couples strongly the angular momentum and the rest of the variables, we describe the test cases that will be used to analyze our methods.

Table 6.2: In this table we report the error in L_1 norm between the total angular momentum at the beginning of the simulation and after different times. The errors refer to three different test cases: *test* 1 refers to the solid body rotation described in (6.1), in *test* 2 the isentropic vortex of (6.10) is taken into account, and finally, in the third one we refer to the four vortexes test case described in (6.11). The results show clearly the exact conservation of the angular momentum obtained with the master-slave approach both with order 1 and 2.

	test 1		test 2		test 3	
time	order 1	order 2	order 1	order 2	order 1	order 2
1	2.6E-16	1.6E-16	9.5E-14	7.3E-14	5.5E-16	7.3E-16
5	1.0E-15	8.2E-16	1.5E-14	6.7E-14	1.6E-15	1.1E-15
10	2.1E-15	1.7E-15	3.6E-15	7.9E-14	1.1E-14	7.2E-16
20	3.7E-15	3.2E-15	1.0E-13	8.9E-14	2.1E-14	1.2E-16

Isentropic vortex tests for Master-Slave

We consider a single isentropic vortex over a ring centered in the origin with radius $r \in [1, 2]$. The initial stationary condition is given by

$$\begin{aligned} \rho(r) &= \left[1 - \frac{(\gamma-1)\beta^2}{8\gamma\pi^2} e^{(1-r^2)}\right]^{\frac{1}{\gamma-1}}, \\ \mathbf{u}(r) &= \frac{\beta}{2\pi} e^{\left(\frac{1-r^2}{2}\right)} \Big(-y, x\Big), \quad p(r) = \rho(r)^\gamma, \quad w = \mathbf{u} \wedge \mathbf{x}, \end{aligned} \tag{6.10}$$

with $r = ||\mathbf{x}||$, $\beta = 5$, $\gamma = 7/5$. See Figure 6.2 for the density and w profiles. In this case the center of rotation is obviously defined and coincides with the origin. A more complex example consists in considering four isentropic vortexes centered respectively in $C_1 = (2.5, 2.5), C_2 = (-2.5, 2.5), C_3 = (-2.5, -2.5), C_4 = (2.5, -2.5)$. The computational domain is a square $[-5, 5] \times [5, 5]$. The initial

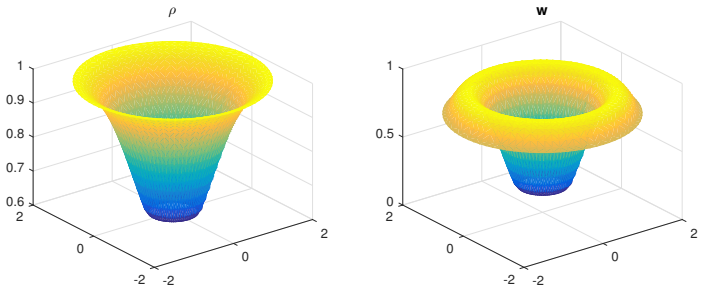


Figure 6.2: Density (left) and angular momentum (right) profiles for the single isentropic vortex stationary solution (6.10).

stationary condition is given by (6.10) with in particular

$$r = \begin{cases} ||\mathbf{x} - C_1|| & \text{if} \quad x \geq 0, \text{ and } y \geq 0 \\ ||\mathbf{x} - C_2|| & \text{if} \quad x < 0, \text{ and } y \geq 0 \\ ||\mathbf{x} - C_3|| & \text{if} \quad x < 0, \text{ and } y < 0 \\ ||\mathbf{x} - C_4|| & \text{if} \quad x \leq 0, \text{ and } y < 0 \end{cases},$$
$$\mathbf{u} = \begin{cases} \frac{\beta}{2\pi} e^{\left(\frac{1-r^2}{2}\right)} \Big(-y, x\Big), & \text{if} \quad xy \geq 0 \\ \frac{\beta}{2\pi} e^{\left(\frac{1-r^2}{2}\right)} \Big(y, -x\Big), & \text{if} \quad xy < 0 \end{cases}. \tag{6.11}$$

See Figure 6.3 for the density and w profiles.

For all test problems presented in this chapter, unless otherwise specified, the reflective wall boundary conditions are implemented by assigning a state at the wall boundary which solves the inverse Riemann problem at the element interface such that the normal velocity $\mathbf{u} \cdot \mathbf{n}$ vanishes at the interface. A first order version of the code is employed in order to compare the three different approaches, in such a way to avoid the effects of reconstruction procedures and boundary conditions in the analysis. Finally, test (6.1) and (6.10) are run with a mesh of 1600 elements, whereas for (6.11) we use a mesh of 6400 elements.

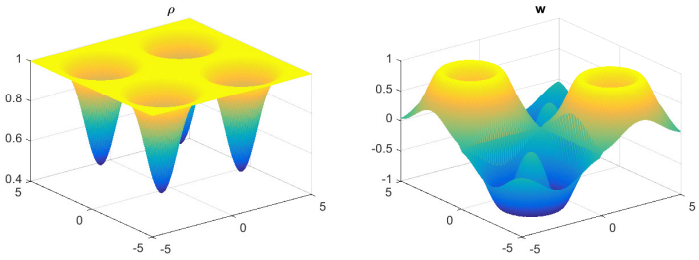


Figure 6.3: Density (left) and angular momentum (right) profiles for the four isentropic vortexes stationary solution (6.11).

6.3 Global-coupling

In order to exploit the angular momentum preservation, we propose a coupled approach which we call *global-coupling*, referring with the term *global* to a fixed global center of rotation and in opposition to the *local-coupling* approach which will be proposed in Section 6.5. The global-coupling is obtained by rewriting the Euler system as follows

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0, \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{v}) + \nabla p = \mathbf{0}, \\ \partial_t (\rho e) + \nabla \cdot (\rho \mathbf{v} e + p \mathbf{v}) = 0, \\ \partial_t (\rho w) + \nabla \cdot (\rho \mathbf{v} w) + \nabla \wedge (p(\mathbf{x} - \mathbf{x}_c)) = 0, \end{cases} \tag{6.12}$$

where

$$\begin{aligned} &\mathbf{v} = \mathbf{v}_r + \mathbf{v}_\theta, \quad \mathbf{v}_r = \frac{1}{r^2} \langle \mathbf{u}, (\mathbf{x} - \mathbf{x}_c) \rangle (\mathbf{x} - \mathbf{x}_c), \quad \mathbf{v}_\theta = -\frac{1}{r^2} w (\mathbf{x} - \mathbf{x}_c)^\perp, \\ &r \mathbf{e}_r = \mathbf{x} - \mathbf{x}_c, \quad r = ||\mathbf{x} - \mathbf{x}_c||, \quad \mathbf{e}_\theta = \mathbf{e}_r^\perp, \\ &e = \varepsilon + \frac{1}{2} \|\mathbf{v}_r\|^2 + \frac{1}{2r^2} w^2, \\ &p = (\gamma - 1) \left(\rho e - \frac{1}{2} \rho \left(||\mathbf{v}_r||^2 + \frac{1}{r^2} w^2 \right) \right). \end{aligned} \tag{6.13}$$

The system remains hyperbolic with an adjoint eigenvalue equal to u_n. Moreover we propose to compute the eigenvalues using $\mathbf{v}$ instead of $\mathbf{u}$. Obviously, on the continuous level, one has $\mathbf{v} = \mathbf{u}$.

This formulation is obtained by noticing that, chosen a center of rotation and an orthonormal basis $(\mathbf{e}_r, \mathbf{e}_\theta)$, the velocity can be rewritten as the sum of the two components along this basis: we call $\mathbf{v}$ the velocity when written in this way. In particular if $\mathbf{u}$ is a radial field and $\mathbf{e}_r$ lies along the radial direction then $\mathbf{v}_r$ is null, which is easy to maintain even at numerical level, since classically $||\mathbf{u}||_{L_2}$ rapidly dissipates. Hence, the correctness of $\mathbf{v}$ strictly depends on the preservation of w. Therefore, being able to conserve w, the expected results of employing this formulation are the following:

a) The error on $\mathbf{v}$, computed *a posteriori* using (6.13), should be less than the error on $\mathbf{u}$, for this reason we introduce another indicator to measure the precision of the results

$$||\mathbf{v} - \mathbf{v}_0||_{L_2} = \sqrt{\sum_{i=1}^{N_E} |T_i| \, ||\mathbf{v}_i - \mathbf{v}_{0,i}||^2}, \quad \text{with} \quad \mathbf{v}_0 = \mathbf{u}_0.$$

b) The use of $\mathbf{v}$, instead of $\mathbf{u}$, in (6.12) should reduce the error even on $||\mathbf{u} - \mathbf{u}_0||_{L_2}$ with respect to the master-slave approach.

On the other side the application of (6.12) is not trivial. First, appropriate boundary conditions should be defined for $\mathbf{v}$, w and $\mathbf{x}$ and moreover the center of rotation should be known. As previously, the numerical method is built using our general solver by defining

$$\mathbf{Q} := \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \rho e \\ \rho w \end{pmatrix} \quad \text{and} \quad \mathbf{F}(\mathbf{Q}, \mathbf{x}) := \begin{pmatrix} \rho\mathbf{v} \\ \rho\mathbf{u} \otimes \mathbf{v} + \mathbf{I}p \\ \rho\mathbf{v}e + p\mathbf{v} \\ \rho\mathbf{v}w + p(\mathbf{x} - \mathbf{x_c})^\perp \end{pmatrix}$$

in (6.8).

Numerical results for Global-Coupling

We have applied the global-coupling to the test case (6.1) and to (6.10). In these two cases the center of rotation is known and coincides with the origin. However the treatment of boundary condition is complex. Consider a boundary element i and call j its *phantom* neighbor. An easy definition consists in considering again the reflective boundary condition and moreover to set $w_j = w_i$, $\mathbf{x}_j = \mathbf{x}_i$, and finally recovering $\mathbf{v}$ through (6.13). This setting guarantees the conservation of $||w||_{L_1}$. Also $||\mathbf{v} - \mathbf{v}_0||_{L_2}$ is significantly reduced with respect to $||\mathbf{u} - \mathbf{u}_0||_{L_2}$.

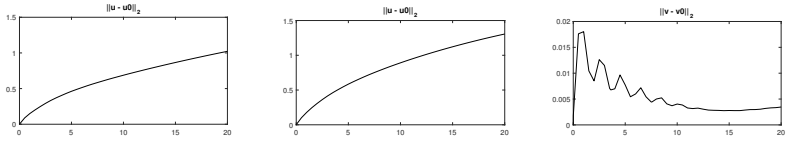


Figure 6.4: Consider the solid body rotation (6.1) and the global-coupling with standard reflective boundary conditions. In the middle we report the $||\mathbf{u} - \mathbf{u}_0||_{L_2}$ obtained with the global-coupling where we cannot appreciate an improving with respect to the master-slave approach (left). But instead the $||\mathbf{v} - \mathbf{v}_0||_{L_2}$ (right) is greatly reduced with respect to $||\mathbf{u} - \mathbf{u}_0||_{L_2}$.

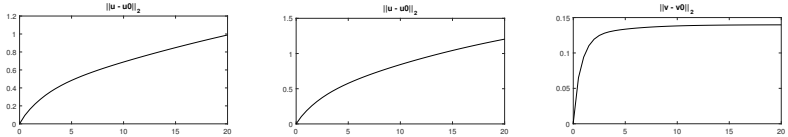


Figure 6.5: Consider the isentropic vortex in (6.10) and the global-coupling with standard reflective boundary conditions. In the middle we report the $||\mathbf{u} - \mathbf{u}_0||_{L_2}$ obtained with the global-coupling where we cannot appreciate an improving with respect to the master-slave approach (left). But instead the $||\mathbf{v} - \mathbf{v}_0||_{L_2}$ (right) is greatly reduced with respect to $||\mathbf{u} - \mathbf{u}_0||_{L_2}$.

Unfortunately, no positive effects can be registered on $||\mathbf{u} - \mathbf{u}_0||_{L_2}$ with respect to the master-slave approach whose error is actually slightly increased. Refer to Figure 6.4 and 6.5 for the numerical results.

Another possible choice for the boundary condition consists in imposing the exact solution in the phantom element j, setting in particular $\mathbf{x}_j$ equal to the barycenter of j. Strictly speaking we lose the conservation, but only because there is an exchange with the exterior. With this choice the performance on $||\mathbf{u} - \mathbf{u}_0||_{L_2}$ is highly increased and at the same time $||\mathbf{v} - \mathbf{v}_0||_{L_2}$ grows slowly. Refer to Figure 6.6 and 6.7 for the numerical results.

These two test cases witness the potential of this formulation despite some intrinsic defects. For example, trying to apply the same scheme to the four vortexes test case (6.11) would not improve the results, since the center of rotation, even if it is known, is not unique. With this motivation we propose the local-coupling approach where multiple centers of rotation can be considered at the same time. At first, we propose a way to detect locally the center of rotation in the following paragraph.

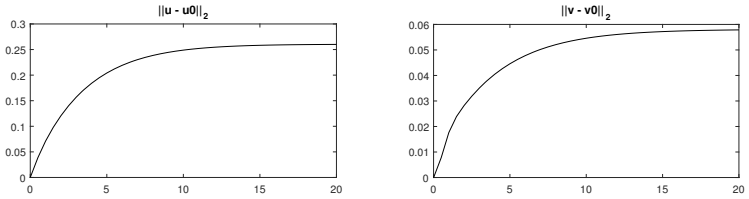


Figure 6.6: Consider the solid body rotation (6.1) and the global-coupling with the boundary conditions that exploit the exact solution. We first notice the great improving on $||\mathbf{u} - \mathbf{u}_0||_{L_2}$ and as counterpart only a small worsening on $||\mathbf{v} - \mathbf{v}_0||_{L_2}$.

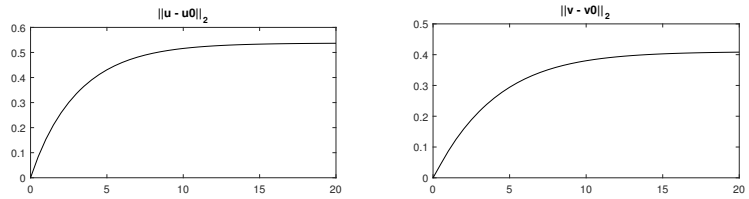


Figure 6.7: Consider the isentropic vortex in (6.10) and the global-coupling with the boundary conditions that exploit the exact solution. We first notice the great improving on $||\mathbf{u} - \mathbf{u}_0||_{L_2}$ and as counterpart only a small worsening on $||\mathbf{v} - \mathbf{v}_0||_{L_2}$.

6.4 Center-Detector

Let us consider a velocity field $\mathbf{u}(\mathbf{x})$ whose value is locally given at the barycenter of each control volume $T_i^n \in \mathcal{T}_\Omega$ at any time step n. It can be described by the following relation

$$\mathbf{u}(\mathbf{x}) = \omega(r)(\mathbf{x} - \mathbf{x_c})^\perp + \varphi(r)(\mathbf{x} - \mathbf{x_c}), \tag{6.14}$$

where $\mathbf{x_c}$ is the center of rotation of the field, r is the distance from the center, $\omega(r)$ represents the angular velocity and $\varphi(r)$ the expansion coefficient.

The aim of this section is to propose a method able to reconstruct the center $\mathbf{x_c}$ and the values of $\omega(r)$ and $\varphi(r)$, given local information about the velocity field $\mathbf{u}(\mathbf{x})$ and a radial pressure field $p = p(r)$. To fix the notation, let us consider an element T_i and its neighbors: let $\mathcal{L}(T_i)$ the set of neighbors of T_i

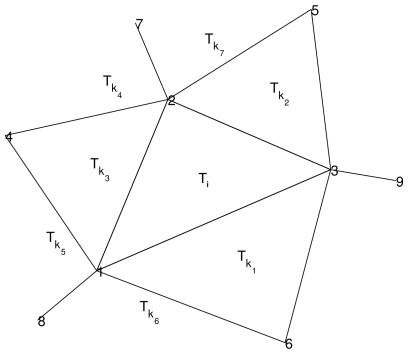


Figure 6.8: The element T_i and its neighbors T_{k_i} $i = 1, 2, \ldots, 7$, *i.e.* T_{k_1}, T_{k_2}, $\ldots$, $T_{k_7} \in \mathcal{L}(T_j)$.

that shares with T_i an edge, and $\mathcal{V}(T_i)$ the set of neighbors of T_i that shares with T_i a vertex. The barycenter of an element T_i is denoted by $\mathbf{x}_i$. All the quantities evaluated at the midpoint between two elements T_i and T_k will be denoted by a star, namely $\mathbf{x}^*_{i,k}, \mathbf{u}^*_{i,k}, r^*_{i,k}$, and for the sake of simplicity by $\mathbf{x}^*, \mathbf{u}^*, r^*$ when there is no confusion. Refer to Figure 6.8 for the notation. The core of the procedure is given by the following proposition.

Proposition 6.4.1. *Let*

$$\varphi^*_{i,k} = \frac{\left\langle \mathbf{u}_i - \mathbf{u}^*_{i,k}, \mathbf{x}_i - \mathbf{x}^*_{i,k} \right\rangle}{\|\mathbf{x}_i - \mathbf{x}^*_{i,k}\|^2}, \quad \omega^*_{i,k} = \frac{\left\langle \mathbf{u}_i - \mathbf{u}^*_{i,k}, \left(\mathbf{x}_i - \mathbf{x}^*_{i,k}\right)^\perp \right\rangle}{\|\mathbf{x}_i - \mathbf{x}^*_{i,k}\|^2}.$$

be a local approximation of the angular velocity and of the expansion factor valid in a neighborhood of element T_i and its neighbor $T_k \in \mathcal{V}(T_i)$. Then the following first order approximations hold

$$\omega^*_{i,k} = \omega(r_i) + \omega'(r_i)\, r^* \left\langle \frac{\mathbf{x}^* - \mathbf{x_c}}{r^*}, \frac{\mathbf{x}_i - \mathbf{x}_k}{\|\mathbf{x}_i - \mathbf{x}_k\|} \right\rangle^2 + \mathcal{O}(h), \tag{6.15}$$

$$\varphi^*_{i,k} = \varphi(r_i) + \varphi'(r_i)\, r^* \left\langle \frac{\mathbf{x}^* - \mathbf{x_c}}{r^*}, \frac{\mathbf{x}_i - \mathbf{x}_k}{\|\mathbf{x}_i - \mathbf{x}_k\|} \right\rangle^2 + \mathcal{O}(h), \tag{6.16}$$

where h is the characteristic dimension of the elements.

Proof. We present here the proof of (6.15), the proof of (6.16) can be obtained following the same procedure. From (6.14) the following equalities can be deduced

$$\begin{aligned}\left\langle \mathbf{u}_i - \mathbf{u}^*, (\mathbf{x}_i - \mathbf{x}^*)^\perp \right\rangle &= \left\langle \omega(r_i)\,(\mathbf{x}_i - \mathbf{x_c})^\perp - \omega(r^*)\,(\mathbf{x}^* - \mathbf{x_c})^\perp, (\mathbf{x}_i - \mathbf{x}^*)^\perp \right\rangle \\ &= \left\langle \omega(r_i)(\mathbf{x}_i - \mathbf{x}^*)^\perp + (\omega(r_i) - \omega(r^*))\,(\mathbf{x}^* - \mathbf{x_c})^\perp, (\mathbf{x}_i - \mathbf{x}^*)^\perp \right\rangle \\ &= \omega(r_i)\ \|\mathbf{x}_i - \mathbf{x}^*\|_2 + (\omega(r_i) - \omega(r^*))\,\langle \mathbf{x}^* - \mathbf{x_c}, \mathbf{x}_i - \mathbf{x}^* \rangle \\ &= \left(\omega(r_i) + \frac{\omega(r_i) - \omega(r^*)}{\|\mathbf{x}_i - \mathbf{x}^*\|_2^2}\, r^* \left\langle \frac{\mathbf{x}^* - \mathbf{x_c}}{r^*}, \mathbf{x}_i - \mathbf{x}^* \right\rangle \right) \|\mathbf{x}_i - \mathbf{x}^*\|_2^2 .\end{aligned} \tag{6.17}$$

By introducing a Taylor approximation we have that

$$\omega(r_i) - \omega(r^*) = \omega'(r_i) \left\langle \frac{\mathbf{x}^* - \mathbf{x_c}}{r^*}, \mathbf{x}_i - \mathbf{x}^* \right\rangle + \mathcal{O}(h^2),$$

and so by substituting this last expression in (6.17) we finally obtain

$$\begin{aligned}&\left\langle \mathbf{u}_i - \mathbf{u}^*, (\mathbf{x}_i - \mathbf{x}^*)^\perp \right\rangle \\ &\quad = \left(\omega(r_i) + \omega'(r_i) r^* \left\langle \frac{\mathbf{x}^* - \mathbf{x_c}}{r^*}, \frac{\mathbf{x}_i - \mathbf{x}^*}{\|\mathbf{x}_i - \mathbf{x}^*\|} \right\rangle^2 \right) \|\mathbf{x}_i - \mathbf{x}^*\|_2^2 + \mathcal{O}(h^3),\end{aligned} \tag{6.18}$$

and dividing the two members by $\|\mathbf{x}_i - \mathbf{x}^*\|_2^2$ we easily recover (6.15). □

The quantity

$$\left\langle \frac{\mathbf{x}^* - \mathbf{x_c}}{r^*}, \frac{\mathbf{x}_i - \mathbf{x}_k}{\|\mathbf{x}_i - \mathbf{x}_k\|} \right\rangle \tag{6.19}$$

can be obtained without explicitly knowing the center $\mathbf{x_c}$ but exploiting the radial pressure field, characterizing any flux subject to a rotation or an expansion.

Proposition 6.4.2. *Let $p = p(r)$ be a radial pressure field then*

$$p_i - p^* = \frac{p'(r^*)}{r^*} \left\langle \mathbf{x}^* - \mathbf{x_c}, \frac{\mathbf{x}_i - \mathbf{x}_k}{\|\mathbf{x}_i - \mathbf{x}_k\|} \right\rangle + \mathcal{O}\left(h^2\right), \tag{6.20}$$

where $p_i = p(\mathbf{x}_i)$ and $p^* = p(\mathbf{x}^*_{ik})$, with T_k a neighbor of T_i .

Proof. The relation can be easily obtained as a Taylor expansion of p which has been supposed to depend only on r. □

Equation (6.20) can be rewritten as

$$p_i - p^* = \lambda \left\langle \frac{\mathbf{x}^* - \mathbf{x_c}}{r^*}, \frac{\mathbf{x}_i - \mathbf{x}_k}{\|\mathbf{x}_i - \mathbf{x}_k\|} \right\rangle + \mathcal{O}\left(h^2\right), \tag{6.21}$$

with

$$\lambda = \|\mathbf{x}^* - \mathbf{x_c}\| \frac{p'(r^*)}{r^*}.$$

To simplify the notation, we remark that the two vectors in (6.19) and (6.20) are unit vectors, hence we call μ the angle between the center of rotation $\mathbf{x_c}$ and x^*, $\alpha = \cos\mu$, $\beta = \sin\mu$, and θ_k the angle between $\mathbf{x}_i$ and $\mathbf{x}_k$. Now in order to obtain (6.19) for the element T_i by exploiting (6.21) we propose to find α and β such that minimize

$$\sum_{k \in \mathcal{L}(T_i)} \left| \lambda \left(\alpha \cos\theta_{k_j} + \beta \sin\theta_k \right) - \xi_k \right|, \quad \text{with} \quad \xi_k = \frac{p_i - p_k}{x_i - x_k}. \tag{6.22}$$

Then, the two unknowns $\omega(r_i)$ and $\omega'(r_i)$ in (6.15) can be recovered by exploiting the values of $\omega^*_{i,k}$ obtained from all the neighbors of T_i and by another minimization procedure. So, finally we have to solve

$$\begin{aligned}
\omega(r_i), \omega'(r_i) &= \underset{z_1, z_2}{\arg\min} \sum_{k \in \mathcal{V}(\mathcal{T}_i)} \left| z_1 + z_2 \left\langle \frac{\mathbf{x}^*_{i,k} - \mathbf{x_c}}{r^*_{i,k}}, \frac{\mathbf{x}_i - \mathbf{x}_k}{\|\mathbf{x}_i - \mathbf{x}_k\|} \right\rangle^2 - \omega^*_{i,k} \right|^2 \\
&= \underset{z_1, z_2}{\arg\min} \sum_{k \in \mathcal{V}(\mathcal{T}_i)} \left| z_1 + z_2 \left\langle \begin{bmatrix} \cos\mu \\ \sin\mu \end{bmatrix}, \begin{bmatrix} \cos\theta_k \\ \sin\theta_k \end{bmatrix} \right\rangle^2 - \omega^*_{i,k} \right|^2 \\
&= \underset{z_1, z_2}{\arg\min} \sum_{k \in \mathcal{V}(\mathcal{T}_i)} \left| z_1 + z_2 \left(\cos\mu\cos\theta_k + \sin\mu\sin\theta_k \right)^2 - \omega^*_{i,k} \right|^2 \\
&= \underset{z_1, z_2}{\arg\min} \sum_{k \in \mathcal{V}(\mathcal{T}_i)} \left| z_1 + z_2 \cos^2(\mu - \theta_k) - \omega^*_{i,k} \right|^2.
\end{aligned} \tag{6.23}$$

Proposition 6.4.3. *The uniqueness of the least square solution of* (6.23) *is guaranteed if and only if the number of neighbors $T_k \in \mathcal{V}(T_i)$ with different θ_k is greater or equal than* 5.

Proof. Relation (6.23) can be rewritten as

$$\omega(r_i), \omega'(r_i) = \underset{z \in \mathbb{R}^2}{\arg\min} \|Az - b\|_2$$

with

$$A = \begin{bmatrix} 1 & \cos^2(\mu - \theta_{k_1}) \\ 1 & \cos^2(\mu - \theta_{k_2}) \\ \vdots & \vdots \\ 1 & \cos^2(\mu - \theta_{k_m}) \end{bmatrix}, \quad z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \quad b = \begin{bmatrix} \omega^*_{k_1} \\ \omega^*_{k_2} \\ \vdots \\ \omega^*_{k_m} \end{bmatrix}, \quad m = \#\mathcal{V}(T_i), \quad (6.24)$$

which has a unique solution if and only if $rank(A)$ is maximal, *i.e.* it is equal to two. Thus, to ensure the uniqueness, we need that at least two elements T_{k_j} and $T_{k_\ell} \in \mathcal{V}(T_i)$ are such that

$$\cos^2(\mu - \theta_{k_j}) \neq \cos^2(\mu - \theta_{k_\ell}),$$

which implies

$$\mu - \theta_{k_i} \neq \begin{cases} \pm(\mu - \theta_{k_\ell}) \\ \pm(\mu - \theta_{k_\ell}) + \pi \end{cases} \quad \rightarrow \quad \theta_{k_i} \neq \begin{cases} \theta_{k_\ell} \\ \theta_{k_\ell} + \pi \\ 2\mu - \theta_{k_\ell} \\ 2\mu - \theta_{k_\ell} + \pi \end{cases} . \quad (6.25)$$

It is clear that if $\#\mathcal{V}(T_i) \leq 4$ the angles θ_k could be exactly linked by the relations in (6.25). Then we need at least 5 neighbors along 5 different directions to ensure that $rank(A) = 2$. □

Remark 6.4.4. *The condition given by Proposition 6.4.3 is in general not restrictive and always verified by any Delaunay triangulation and any structured grid. However, this tells us that the elements in $\mathcal{L}(T_i)$ are not enough to ensure the uniqueness of the solution, and so we really need to consider the set of all the neighbors $\mathcal{V}(T_i)$.*

Let us resume the fundamental steps of the algorithm: first one compute α and β through (6.22), then it is possible to recover $\omega(r_i), \omega'(r_i)$ with (6.23) and finally the center $\mathbf{x_c}$ can be obtained through (6.14), (6.15) and (6.16).

6.5 Local-coupling

As already noticed, we are interested in studying problems where the center of rotation could be unknown or there could be more than a center, so we propose a second coupled approach, that we will call *local-coupling*. The advantage of such a formulation is the possibility of defining a different center of rotation for each element of the mesh. In this way we can treat both problems with multiple *known* rotation centers and even situations for which the center is *a priori* unknown but only approximated from each element, and hence affected by numerical errors. In the latter case, we shall use the local center-detector proposed in the previous section.

The difference with respect to the previous case lies in the flux computation which is done at a local level. We propose to rewrite Euler system as follows

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho \mathbf{v}^*) = 0, \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{v}^*) + \nabla p = \mathbf{0}, \\ \partial_t (\rho e) + \nabla \cdot (\rho \mathbf{v}^* e + p \mathbf{v}^*) = 0, \\ \partial_t (\rho w) + \nabla \cdot (\rho \mathbf{v}^* w) + \nabla \wedge (p(\mathbf{x} - \mathbf{x}^*)) = 0, \end{cases} \tag{6.26}$$

where

$$\begin{aligned} & w = \mathbf{u} \wedge \mathbf{x}, \quad w^* = w - \mathbf{u} \wedge \mathbf{x}^* \\ & \mathbf{e}^* = \frac{\mathbf{x} - \mathbf{x}^*}{r^*}, \quad r^* = \|\mathbf{x} - \mathbf{x}^*\|, \\ & \mathbf{v}^* = (\mathbf{u}, \mathbf{e}^*)\mathbf{e}^* - \frac{\mathbf{w}^*}{r^*} \mathbf{e}^{*\perp}. \end{aligned} \tag{6.27}$$

$\mathbf{x}^*$ is the local center of rotation. In this later case, the numerical method is defined by writing

$$\mathbf{Q} := \begin{pmatrix} \rho \\ \rho \mathbf{u} \\ \rho e \\ \rho w \end{pmatrix} \quad \text{and} \quad \mathbf{F}(\mathbf{Q}, \mathbf{x}) := \begin{pmatrix} \rho \mathbf{v}^* \\ \rho \mathbf{u} \otimes \mathbf{v}^* + Ip \\ \rho \mathbf{v}^* e + p \mathbf{v}^* \\ \rho \mathbf{v}^* w + p(\mathbf{x} - \mathbf{x}^*)^\perp \end{pmatrix}$$

in (6.8). Notice that if $\mathbf{x}^*$ coincides with a unique rotation center the local-coupling method coincides with the global one.

In practical applications, $\mathbf{x}^*$ is defined on the edges of the mesh (to compute the numerical fluxes). Logically $\mathbf{x}^*$ is different at any edge of the mesh. Another possible choice, when computing the flux between the elements i and k through the edge ℓ_{ik}, consists in taking $\mathbf{x}^*$ equal to the midpoint of ℓ_{ik}. With this

choice the method is stable, but not significant improvements can be achieved on $||\mathbf{u} - \mathbf{u}_0||_{L_2}$.

Numerical results for Local-Coupling

We can test the algorithm on our test (6.1). Our procedure results to be quite accurate in computing the center especially when the data (*i.e.* pressure and velocity fields) are well discretized. Moreover we can couple our detector with the local-coupling method. Let us call $\overline{\mathbf{x}_{\mathbf{c},i}}$ the approximation of the center obtained through our procedure considering an element T_i and when computing the flux between T_i and T_j choose $\mathbf{x}^* = \mathbf{x}^*_{ij}$ as

$$\mathbf{x}^*_{ij} = \frac{\overline{\mathbf{x}_{\mathbf{c},i}} + \overline{\mathbf{x}_{\mathbf{c},j}}}{2}.$$

In Figure 6.9 we compare the quantities $||\mathbf{u} - \mathbf{u}_0||_{L_2}$ obtained with the three methods we have proposed on the solid body rotation test case (6.1). This example shows that the local-coupling method, that employs only an approximation of the center, gives results similar to the global-coupling procedure that instead makes use of the exact center. Similar numerical results can be obtained by considering the isentropic vortex test case, see Figure 6.10.

6.5.1 Numerical criteria for general cases

In practical implementation, it does not make sense to consider the center detected by all the elements: because some of them could be affected by a shock in the velocity field, by an almost constant pressure, by a field which is not perfectly radial or by a mesh configuration particularly ill conditioned with respect to the angle of rotation μ.

For all these reasons we propose here some numerical criteria to understand when the center detected by our algorithm could be considered as a good approximation of the exact one.

First of all, we can accept a result only when the pressure field in the neighborhood of the element has really a radial shape. So we will first exclude the elements across which pressure differences are too small (*i.e.* less that 10^{-8} for example). Moreover, since the detection method is first order accurate, and from (6.20) we know that α and β should be approximate with an error of the order of $\mathcal{O}(h^2)$, if the minimization procedure is affected by an error greater than

$$c_1 h$$

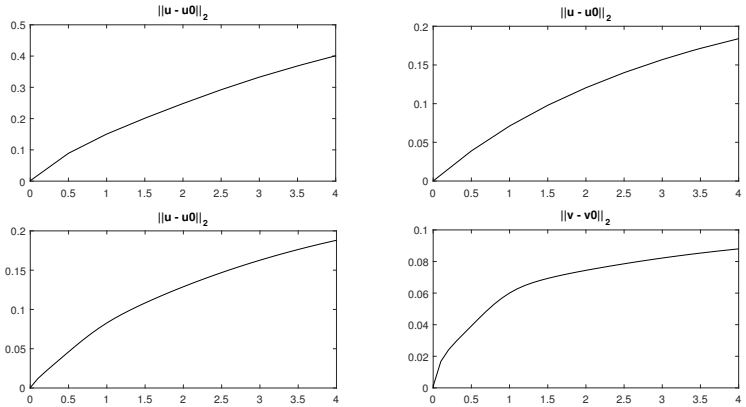


Figure 6.9: Consider the test case (6.1). Top left master-slave approach. Top right global-coupling. Bottom local-coupling with center-detector. This example shows that the local-coupling, which employs only an approximation of the center, gives results similar to the global-coupling procedure that makes use of the exact center.

we will exclude these elements. Indeed, we deduce from an high variation on the estimate of α and β a non radial pressure field in the neighborhood of the considered element.

The second limitation is connected with the least squares procedure described in (6.23). Even in this case we know the expected order of accuracy in the computations of ω and ω', which is of order of $\mathcal{O}(h)$. We decide to exclude

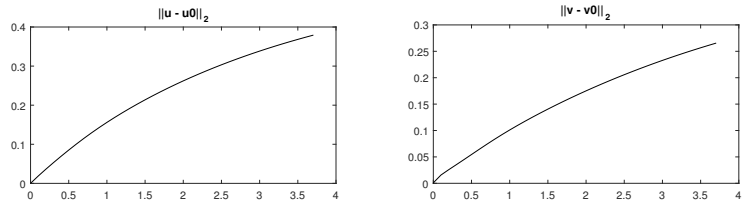


Figure 6.10: Consider the isentropic vortex in (6.10) and the local-coupling with center-detector. Even on this example this approximated procedure gives us a reduction in the velocity dissipation.

all the elements for which the residuum of the least squares exceeds a certain value of the form of

$$c_2 h.$$

In this way we exclude the element for which matrix A in (6.24) is ill conditioned.

Last, recover the center by exploiting (6.14) could be a very difficult task because it requires to invert a matrix whose determinant is proportional to $\Lambda = \omega^2 + \varphi^2$. If Λ is to small, its inverse will be too big and the final computation is ill conditioned. For this reason we will exclude from the computation all elements for which

$$\Lambda < c_3 h.$$

While no limitations are required if the pressure and the velocity fields are radial and well approximated, in more complex test cases the choice of c_1, c_2, c_3 can be relevant and not trivial.

By considering the four vortexes test case (6.11) we have verified that our detector (without the application of the previous criteria) fails in all the elements where the velocity field is too small to detect a radial field, in the elements too close to the rotation center or in the diagonal direction where the mesh configuration is particularly ill conditioned. So in Table 6.3 we report the results obtained by applying the numerical criteria described above: by eliminating the elements which are not suited for our detector the algorithm maintains a good precision in the determination of the center.

However its coupling with the local method does not give an improving on the computation of the velocities: indeed for all the elements from which the center cannot be computed the choice of $\mathbf{x}^*$ is not clear and even the boundary conditions for $\mathbf{x}^*$ and w^* need a more sophisticated investigation. In particular choosing $\mathbf{x}^*$ equal to the midpoint when the center is not known does not increase the quality of the results with respect to not consider w.

6.6 Coupling with ALE techniques

We underline that all the test cases presented up to now are run with a zero mesh velocity. The coupling with the ALE techniques is not trivial: again because of the boundary conditions of the adjoint quantities $\mathbf{x}, w, \mathbf{v}$ but even because of the standard distortion problems of considering a moving domain. However, taking into account the test case (6.1) and imposing the boundary conditions through the exact solution we can see that the global formulation

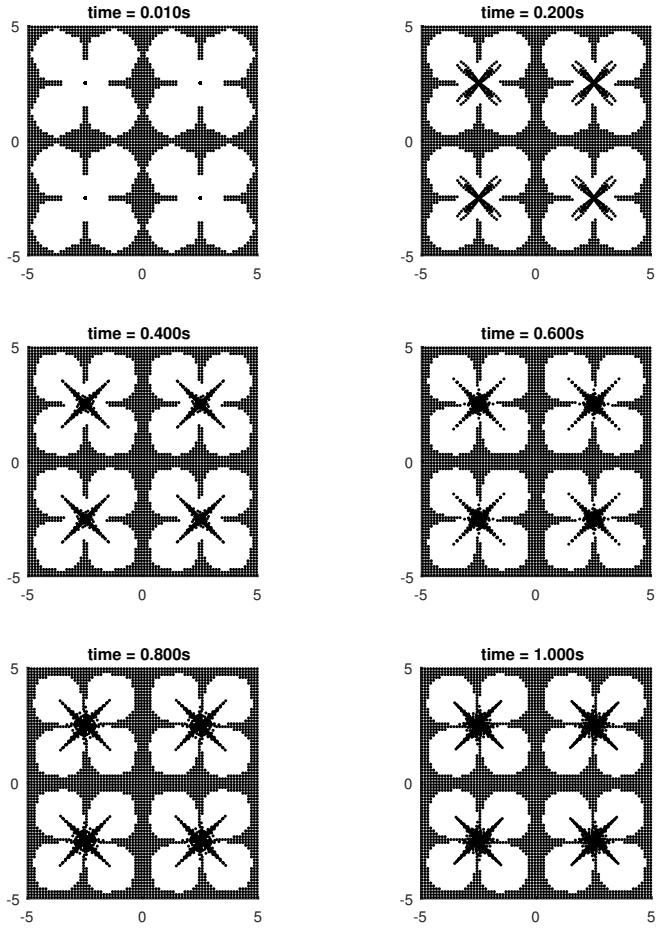


Figure 6.11: Four vortexes test case (6.11). The black points represent the barycenter of the elements for which our center-detector fails. The white areas show the location of the elements for which our detector computes the center with a good precision at different times.

Table 6.3: Four vortexes test case (6.11). We have applied our detector with the numerical criteria described in 6.5.1 choosing in particular $c_1 = 6$, $c_2 = 10$, $c_3 = 0.1$. In the table we report the percentage of active elements and the mean error in the computation of the center obtained with a Cartesian mesh with characteristic mesh size equal to 0.125 at two different times. By mean error we mean $\frac{\sum_{i \in \text{active elements}} ||\mathbf{x_c}, i - \mathbf{x_c}||}{\#\text{ active elements}}$. The error are of the order of the mesh size, as expected.

time	active elements	error
0	37%	0.15
1.2	33%	0.18

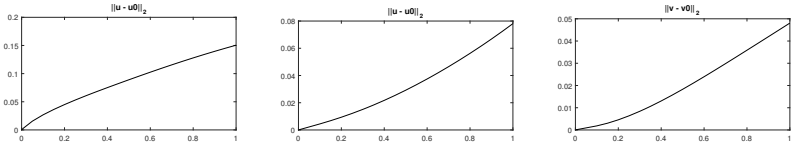


Figure 6.12: Consider the solid body rotation (6.1) with boundary condition given by the exact solution. We compare the results obtained with the master-slave approach with a zero velocity mesh (left) and the global-coupling formulation in the ALE framework (middle and right). We can notice that even in the complex situation of a moving mesh the global-coupling allows a better preservation of the velocity norm.

allows a better preservation of the velocity norm even in this context at least for small times, refer to Figure 6.12.

6.7 A Kidder problem with rotation

A solution depending on three coefficients $\overline{\rho} > 0$, $\alpha \geq 0$ and $R_0 > 0$ reads

$$\begin{cases} \rho_0(R) = \overline{\rho}\left(\frac{R^2}{R_0^2} + \alpha\right), \\ p_0(R) = \widehat{p}\left(\frac{R^2}{R_0^2} + \alpha\right)^2, \quad \widehat{p} = \frac{1}{4}\overline{\rho}\frac{R_0^2}{\overline{\tau}^2} = \left(1 + \omega^2\tau^2\right)\frac{1}{4}\overline{\rho}\frac{R_0^2}{\tau^2}. \end{cases} \tag{6.28}$$

The full derivation is in the Appendix B. The other initial data are $u_\theta(R) = \omega R$ and $u_r(R) = 0$. This solution is very similar to the Kidder solution for $\omega = 0$, but the pressure is premultiplied by the constant factor $1 + \omega^2\tau^2$ to counterbalance the centrifugal force created by the angular solid body rotation.

Table 6.4: Kidder with rotation test case at the final time $t_f = 0.6\tau$ with zero mesh velocity. We report the L_2 norm of the error over ρ, $\mathbf{u}$ and w with respect to the exact solution $\overline{\rho}$, $\overline{\mathbf{u}}$, $\overline{w}$.

mesh	master-slave approach			global-coupling approach		
h	$\|\|\rho - \overline{\rho}\|\|_{L_2}$	$\|\|\mathbf{u} - \overline{\mathbf{u}}\|\|_{L_2}$	$\|\|w - \overline{w}\|\|_{L_2}$	$\|\|\rho - \overline{\rho}\|\|_{L_2}$	$\|\|\mathbf{u} - \overline{\mathbf{u}}\|\|_{L_2}$	$\|\|w - \overline{w}\|\|_{L_2}$
2.03E-04	4.15E-2	4.82E-1	7.79E-1	6.26E-2	6.96E-1	9.48E-1
9.50E-05	2.77E-2	2.99E-1	4.97E-1	3.81E-2	4.30E-1	6.09E-1

For our test case we have chosen $R_2 = 1$, $R_1 = 0.9$, $\rho_2 = 2$ and $\rho_1 = 1$. This corresponds to a focusing time of $\tau \simeq 0.21794$ and we run the simulation until $t_f = 0.6\tau$. The initial angular velocity is $\omega = \tau^{-1}$ so that the pressure is, with respect to the classical Kidder solution, multiplied by a constant factor 2. The boundary conditions are imposed by prescribing the outer exact value of density, velocity and pressure.

In Figure 6.14 we report the density and the angular momentum contours of our numerical solution (obtained with the master-slave approach) compared with the analytical one.

Moreover, in Table 6.4 we report the errors over the density, the velocity and the angular momentum obtained both with the master-slave and the global-coupling approaches. We underline that the results obtained with the global-coupling do not improve the overall quality because in this test case the velocity field has a non null radial component: so even if the angular component is approximated through the angular momentum, no positive effects can be seen on the radial component. However this does not prevent the convergence of the method.

Finally, we have performed the same test moving the mesh with the fluid velocity. In particular the Cheng and Shu node solver [49,109] has been employed to compute the velocity of each node of the mesh. We report our numerical results, obtained both with the master-slave and the global-coupling approaches, in Figure 6.14 and in Table 6.5.

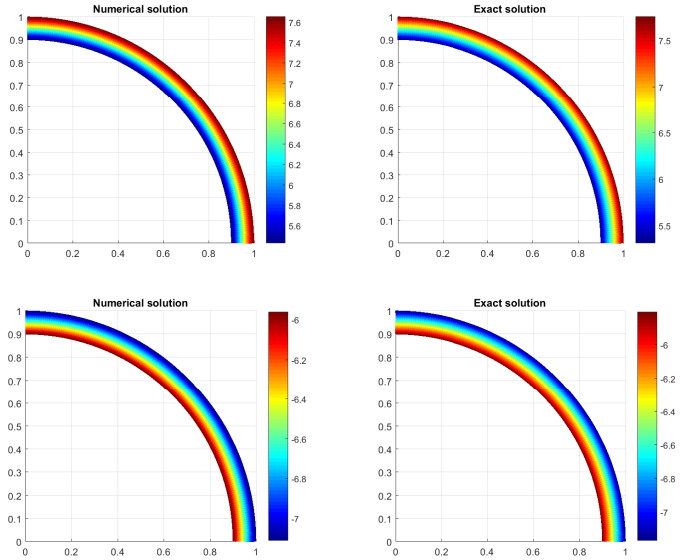


Figure 6.13: Kidder with rotation test case at the final time $t_f = 0.6\tau$ with an Eulerian scheme. We compare the numerical solution (left) with the analytical one (right), considering the density profile (top) and the angular momentum (bottom). The numerical results have been obtained with the first order master-slave approach and a mesh with 6280 quadrilateral elements.

Table 6.5: Kidder with rotation test case at the final time $t_f = 0.6\tau$ with the ALE code. We report the L_2 norm of the error over ρ, $\mathbf{u}$ and w with respect to the exact solution $\overline{\rho}$, $\overline{\mathbf{u}}$, $\overline{w}$.

mesh	master-slave approach			global-coupling approach		
h $(t=0)$	$\|\|\rho-\overline{\rho}\|\|_{L_2}$	$\|\|\mathbf{u}-\overline{\mathbf{u}}\|\|_{L_2}$	$\|\|w-\overline{w}\|\|_{L_2}$	$\|\|\rho-\overline{\rho}\|\|_{L_2}$	$\|\|\mathbf{u}-\overline{\mathbf{u}}\|\|_{L_2}$	$\|\|w-\overline{w}\|\|_{L_2}$
2.03E-04	4.11E-2	2.18E-1	1.90E-1	3.82E-2	5.09E-1	4.27E-1
9.50E-05	2.12E-2	1.78E-1	9.63E-2	2.56E-2	3.75E-1	3.09E-1

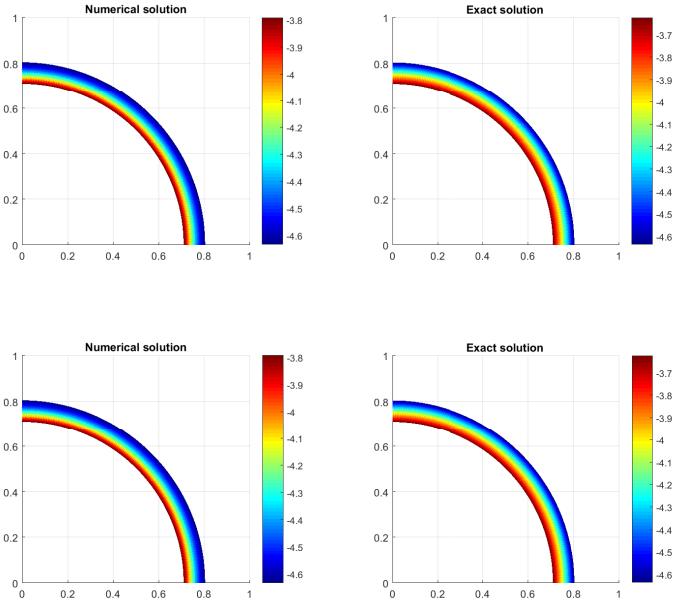


Figure 6.14: Kidder with rotation test case at the final time $t_f = 0.6\tau$ with the ALE scheme. We compare the numerical solution (left) with the analytical one (right), considering the density profile (top) and the angular momentum (bottom). The numerical results have been obtained with the first order master-slave approach and a mesh of 6280 quadrilateral elements.

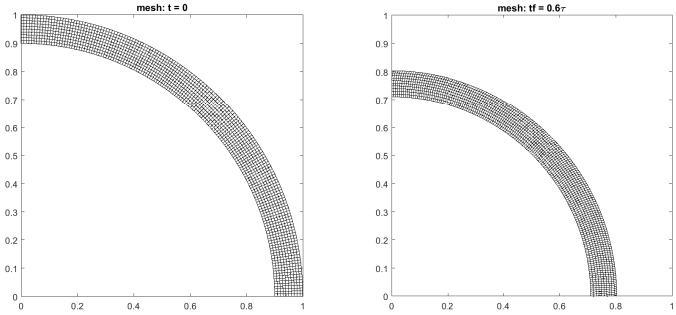


Figure 6.15: Initial and final mesh relative to the Kidder with rotation test case performed with the ALE code presented in Chapter 2 on a coarse mesh, with Rusanov-type numerical flux and without nonconforming sliding lines.

7 Conclusions and outlooks

To conclude, I would like to underline that the research presented in this thesis is based on three powerful founding concepts. First the space time conservation formulation of the governing PDEs which naturally applies to the novel moving nonconforming ALE framework; second the idea to treat the nonconservative products in a well balanced way by directly employing the steady equilibrium in the path construction; finally the exploitation of an adjoint conservation law, namely the angular momentum conservation law, to increase the accuracy of the entire system. These original ideas have been applied in different contexts and coupled together obtaining promising results that justify further research in many directions.

In particular the introduced nonconforming ALE techniques enables the resolution even of complex shear flows with differential rotations in an effective and accurate way. It is noteworthy to stress again that standard *conforming* Lagrangian schemes will *crash* after *finite times* for any vortex flow with differential rotation due to *mesh tangling*. Moreover the reduced dissipation characterizing the Lagrangian methods, together with the high mesh quality provided by the nonconforming treatment of sliding lines, and the increased accuracy near the equilibria given by the well balanced techniques, allow us to obtain significant improvements compared to the existing state of the art. The major benefits are achieved with our new class of schemes when studying physical phenomena that arise close to stationary solutions, where standard discretizations would hide the flow physics with spurious oscillations and excessive numerical dissipation.

Future research will consider an extension to three space dimensions as well as to more general classes of stationary solutions and an automatic detector of the equilibrium profiles in order to extend our method to situations in which the equilibrium is not known exactly *a priori*. With respect to the two-phase model for example, an automatic detection of the water level of equilibrium could be used to restore the water at rest solution after a dambreak.

Furthermore, based on the high order path-conservative methods introduced in [80] we intend to use the algorithms developed in this thesis in order to design exactly well balanced schemes for gravity driven equilibrium flows in *general*

relativity, where the use of well balanced methods appears to be still rather unknown. We also plan to increase the *order* of accuracy of the presented method by extending the Lagrangian ADER-WENO and ADER-DG schemes proposed in [23, 25, 27] to moving nonconforming unstructured meshes in a well balanced manner. Finally, we envisage to remove the mesh constraints and design a well balanced scheme for completely general moving nonconforming unstructured meshes.

For what concerns the angular momentum preserving schemes, the conservation of the angular momentum is guaranteed by solving the augmented system for all the studied test cases. Moreover, we have presented some test cases where the preservation of the angular momentum allows also to maintain good velocity profiles for long times, better than the one obtained with standard methods. Despite the good results achieved with simple test cases on symmetric domains we consider this work only as starting point. Indeed the extension to more complex situations and further exploitation of our center detector would need supplementary investigations.

A Proof of well balancing for a general element in 2D

In this section we recall the first order ALE one-step finite volume scheme in two space dimensions for the Euler Equations with gravity, and we show that our formulation is well balanced for each element of a mesh that satisfies the constraints stated at the beginning of Section 4.1.1.

Consider a generic element I and its neighbors $J_i, i = 1, \dots 6$, respectively through the edges $\Gamma_j, i = 1, \dots, 6$, as depicted in Figure A.1. As derived in Section 4.1.2 our first order ALE scheme can be written as

$$|T_I^{n+1}|\, \mathbf{Q}_I^{n+1} = |T_I^n|\, \mathbf{Q}_I^n - \sum_{J_i} \int_0^1 \int_0^1 |\partial C_{I,J_i}^n|\, \tilde{\mathbf{D}}_{I,J_i} \cdot \tilde{\mathbf{n}}_{I,J_i}\, d\chi d\tau \quad \text{(A.1)}$$

and a sufficient condition to be well balanced is that

$$\sum_{J_i} \int_0^1 \int_0^1 |\partial C_{I,J_i}^n|\, \tilde{\mathbf{D}}_{I,J_i} \cdot \tilde{\mathbf{n}}_{I,J_i}\, d\chi d\tau = \mathbf{0} \quad \text{(A.2)}$$

when evaluated on equilibrium states.

Note that $\Gamma_{3,4,5,6}$ are parallel to the radial direction so the normal vectors are $\tilde{\mathbf{n}} = (n_r, 0, 0)$, hence the flux across these edges is exactly the 1D flux, which has already been proven to be zero when evaluated on stationary solutions. Therefore (A.2) reduces to

$$\int_0^1 \int_0^1 \left(|\partial C_{I,J_1}^n|\, \tilde{\mathbf{D}}_{I,J_1} \cdot \tilde{\mathbf{n}}_{I,J_1} + |\partial C_{I,J_2}^n|\, \tilde{\mathbf{D}}_{I,J_2} \cdot \tilde{\mathbf{n}}_{I,J_2} d\chi d\tau \right) \quad \text{(A.3)}$$

where, since $\Gamma_{1,2}$ are parallel and have the same length,

$$\tilde{\mathbf{n}}_{I,J_2} = -\tilde{\mathbf{n}}_{I,J_1} = (\tilde{n}_r, \tilde{n}_\varphi, \tilde{n}_t) \quad \text{and} \quad |\partial C_{I,J_2}^n| = |\partial C_{I,J_1}^n|, \quad \text{(A.4)}$$

so we can rewrite

$$\int_0^1 \int_0^1 |\partial C_{I,J_1}^n| \left(\tilde{\mathbf{D}}_{I,J_1} \cdot \tilde{\mathbf{n}}_{I,J_1} - \tilde{\mathbf{D}}_{I,J_2} \cdot \tilde{\mathbf{n}}_{I,J_1} d\chi d\tau \right). \quad \text{(A.5)}$$

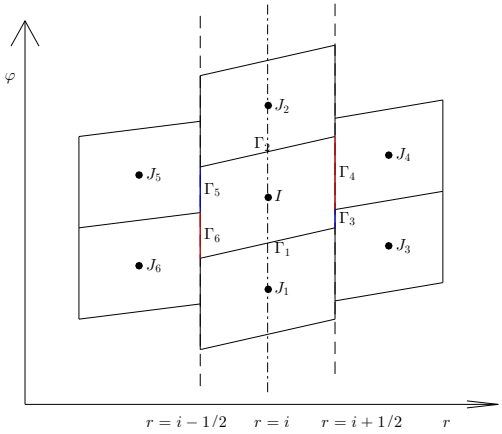


Figure A.1: Portion of a general nonconforming mesh that satisfies the constraints in Section 4.1.1. We consider an element I, and its neighbors $J_i, i = 1, \dots 6$, respectively through the edges $\Gamma_j, i = 1, \dots, 6$. In particular $\Gamma_{1,2}$ are parallel, $\Gamma_{3,4,5,6}$ lie on vertical straight lines and the barycenter of I, J_1 and J_2 have the same r coordinate.

Now, by exploiting (4.12) the integrand can be rewritten as

$$\begin{aligned}
&\left|\partial C^n_{I,J_1}\right| \left(\frac{1}{2}\left(\mathbf{f}(\mathbf{q}^E_{J_1}) + \mathbf{f}(\mathbf{q}^E_I) + \mathcal{B}_{I,J_1}\left(\mathbf{q}^E_{J_1} - \mathbf{q}^E_I\right)\right)\tilde{n}_r \right. \\
&+\frac{1}{2}\left(\mathbf{g}(\mathbf{q}^E_{J_1})+\mathbf{g}(\mathbf{q}^E_I)\right)\tilde{n}_\varphi + \frac{1}{2}\left(\mathbf{q}^E_{J_1}+\mathbf{q}^E_I\right)\tilde{n}_t - \frac{1}{2}\mathcal{V}_{I,J_1}\left(\mathbf{q}^E_{J_1}-\mathbf{q}^E_I\right) \\
&-\frac{1}{2}\left(\mathbf{f}(\mathbf{q}^E_{J_2}) + \mathbf{f}(\mathbf{q}^E_I) + \mathcal{B}_{I,J_2}\left(\mathbf{q}^E_{J_1} - \mathbf{q}^E_I\right)\right)\tilde{n}_r \\
&\left.-\frac{1}{2}\left(\mathbf{g}(\mathbf{q}^E_{J_2})+\mathbf{g}(\mathbf{q}^E_I)\right)\tilde{n}_\varphi - \frac{1}{2}\left(\mathbf{q}^E_{J_2}+\mathbf{q}^E_I\right)\tilde{n}_t + \frac{1}{2}\mathcal{V}_{I,J_2}\left(\mathbf{q}^E_{J_2}-\mathbf{q}^E_I\right)\right).
\end{aligned} \tag{A.6}$$

We already know that the component multiplied by $\tilde{n}_r$ vanishes at the equilibrium. Moreover, since the barycenter of I, J_1, J_2 are aligned along the same vertical straight line $r = i$,

$$\mathbf{q}^E_{J_2} = \mathbf{q}^E_{J_1}, \tag{A.7}$$

and so the terms multiplied by $\tilde{n}_\varphi$ and $\tilde{n}_t$ cancel between them.

For what concerns the viscosity, in the case of the Osher-Romberg scheme we refer to (4.14)-(4.15) that proves

$$\mathcal{V}_{I,J_i}\Big(\mathbf{q}^E_{J_i} - \mathbf{q}^E_I\Big) = 0 \tag{A.8}$$

provided that the rest of the scheme is well balanced (as we have just proven). For the HLL-type flux we have

$$\begin{aligned} &-\alpha^0_{I,J_1}\left(\tilde{I}_{\frac{I+J_1}{2}}\, n_r + I n_\varphi\right)\left(\mathbf{q}^E_{J_1} - \mathbf{q}^E_I\right) - \alpha^1_{I,J_1}\mathcal{R}_{I,J_1} \\ &+\alpha^0_{I,J_2}\left(\tilde{I}_{\frac{I+J_2}{2}}\, n_r + I n_\varphi\right)\left(\mathbf{q}^E_{J_2} - \mathbf{q}^E_I\right) + \alpha^1_{I,J_2}\mathcal{R}_{I,J_2} \end{aligned} \tag{A.9}$$

where $\tilde{I}_{\frac{I+J_i}{2}}$ vanishes as in the one dimensional case, $\mathcal{R}_{I,J_i}$ vanishes because we have already proven that the rest of the scheme vanishes, and the term multiplied by n_φ cancels because of (A.7).

B Implosion of Kidder type with rotation

The Kidder test problem is emblematic of strong implosion in stars or for inertial confinement devices. Starting from the standard Kidder problem, we show how to add a rotation to the initial condition. The derivation of the analytical solution is described using the seminal method of [20]. One starts with the Euler equations in general dimension $d \geq 1$

$$\left\{ \begin{array}{l} D_t\rho + \rho\nabla\cdot\mathbf{v} = 0, \quad D_t = \partial_t + \mathbf{v}\cdot\nabla, \\ \rho D_t\mathbf{v} + \nabla p = \mathbf{0}, \\ D_t(p/\rho^\gamma) = 0. \end{array} \right.$$

For a flow with rotation invariance it can be recast as

$$\left\{ \begin{array}{ll} D_t\rho + \rho r^{-(d-1)}\partial_r(r^{d-1}u_r) = 0, & \mathbf{v} = u_r\mathbf{e}_r + u_\theta\mathbf{e}_\theta, \\ \rho D_t u_r + \partial_r p = F_{\text{centrifugal}}, & F_{\text{centrifugal}} = \rho u_\theta^2/r, \\ D_t(u_\theta r) = 0, & \text{(conservation of angular momentum)} \\ D_t(p/\rho^\gamma) = 0. & \end{array} \right.$$

The new feature with respect to the standard Kidder solution is the non zero angular velocity u_θ. One looks for a self-similar solution $r = Rf(t)$ with $u_r = D_t r = Rf'(t) = rf'(t)f(t)^{-1}$.

One gets $\rho = \rho_0(R)f(t)^{-d}$ where $\rho_0(R)$ is the density at time $t = 0$. This can be checked as follows: one has

$$D_t\rho = \partial_{t|R}\left(\rho_0(R)f(t)^{-d}\right) = -\rho_0(R)df(t)^{-(d+1)}$$

and

$$\rho r^{-(d-1)}\partial_r(r^{d-1}u_r) = \rho\partial_r u_r + \rho(d-1)r^{-1}u_r = \rho f'(t)f(t)^{-1} + \rho(d-1)f'(t)f(t)^{-1}$$

$$= \rho df'(t)f(t)^{-1} = \rho_0(R)df'(t)f(t)^{-(d+1)}.$$

By summation, one gets the continuity equation $D_t\rho + \rho r^{-(d-1)}\partial_r(r^{d-1}u_r) = 0$. The adiabaticity of the flow yields $p = p_0(r)(\rho/\rho_0(R))^\gamma$ that is $p = p_0(R)f(t)^{-\gamma d}$.

Moreover one has the identities

$$\begin{cases} \rho D_t u_r = f(t)^{-d}\rho_0(R)Rf''(t), \\ \partial_r p = f(t)^{-1}\partial_R p = f(t)^{-(\gamma d+1)}p_0'(R), \\ \rho u_\theta^2/r = f(t)^{-(d+3)}\rho_0(R)(u_\theta)_0(R)^2/R. \end{cases}$$

The Newton equation $\rho D_t u_r + \partial_r p = \rho u_\theta^2/r$ is an identity between the above 3 terms. It yields

$$f(t)^{-d}\rho_0(R)Rf''(t) + f(t)^{-(\gamma d+1)}p_0'(R) = f(t)^{-(d+3)}\rho_0(R)(u_\theta)_0(R)^2/R. \quad \text{(B.1)}$$

As in [20], the solution is seek by separation of variables.

However a preliminary manipulation is necessary because the identity (B.1) is made with 3 contributions. We equate the power of the terms which are non differential with respect to $f(t)$ using $\gamma d + 1 = d + 3 \iff \gamma = \frac{d+2}{d} = 2$. One gets

$$\rho_0(R)Rf(t)^3 f''(t) = \rho_0(R)w_0(R)^2/R - p_0'(R).$$

A classical solution [20] by separation of variables of such an equation is $f(t) = \sqrt{1 - t^2/\tau^2}$ where $\tau > 0$ is a focalization time, so that $f(t)^3 f''(t) = -\tau^{-2}$. It remains to discuss the reduced equation

$$p_0'(R) = \rho_0(R)R\tau^{-2} + \rho_0(R)w_0(R)^2/R. \quad \text{(B.2)}$$

We decide for convenience of a rigid body rotation which corresponds to $w_0(R) = \omega R$ where ω is a given angular velocity. Let us define $\hat{\tau}^{-2} = \tau^{-2} + \omega^2$ so that (B.2) rewrites

$$p_0'(R) = \rho_0(R)R\hat{\tau}^{-2}. \quad \text{(B.3)}$$

Bibliography

[1] R. Abgrall. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. *Journal of Computational Physics*, 144:45–58, 1994.

[2] A. A. Amsden, P. O'rourke, and T. Butler. Kiva-ii: A computer program for chemically reactive flows with sprays. Technical report, Los Alamos National Lab., NM (USA), 1989.

[3] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, and B. Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing*, 25(6):2050–2065, 2004.

[4] M. Baer and J. Nunziato. A two-phase mixture theory for the deflagration-to-detonation transition (ddt) in reactive granular materials. *International journal of multiphase flow*, 12(6):861–889, 1986.

[5] D. Balsara. A two-dimensional HLLC riemann solver for conservation laws: Application to euler and magnetohydrodynamic ows. *Journal of Computational Physics*, 231:7476–7503, 2012.

[6] D. Balsara. Multidimensional Riemann problem with self-similar internal structure - Part I - Application to hyperbolic conservation laws on structured meshes. *Journal of Computational Physics*, 277:163–200, 2014.

[7] D. Balsara and M. Dumbser. Multidimensional Riemann problem with self-similar internal structure - Part II - Application to hyperbolic conservation laws on unstructured meshes. *Journal of Computational Physics*, 287:269–292, 2015.

[8] D. Balsara, T. Rumpf, M. Dumbser, and C. Munz. Efficient, high accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics. *Journal of Computational Physics*, 228:2480–2516, 2009.

[9] D. S. Balsara, M. Dumbser, and R. Abgrall. Multidimensional hllc riemann solver for unstructured meshes-with application to euler and mhd flows. *Journal of Computational Physics*, 261:172–208, 2014.

[10] A. J. Barlow and J. Whittle. Mesh adaptivity and material interface algorithms in a two dimensional Lagrangian hydrocode. *Chemical Physics*, 19:15–26, 2000.

[11] T. Barth and D. Jespersen. The design and application of upwind schemes on unstructured meshes. *AIAA Paper 89-0366*, pages 1–12, 1989.

[12] G. Batchelor. An introduction to fluid mechanics, 615 pp. *Cambridge University Press, New York*, 1967.

[13] D. J. Benson. Computational methods in lagrangian and eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99(2):235 – 394, 1992.

[14] A. Bermúdez, X. López, and M. E. Vázquez-Cendón. Numerical solution of non-isothermal non-adiabatic flow of real gases in pipelines. *Journal of Computational Physics*, 323:126–148, 2016.

[15] A. Bermudez and M. E. Vazquez. Upwind methods for hyperbolic conservation laws with source terms. *Computers & Fluids*, 23(8):1049–1071, 1994.

[16] M. Berndt, J. Breil, S. Galera, M. Kucharik, P. H. Maire, and M. Shashkov. Two–step hybrid conservative remapping for multimaterial arbitrary Lagrangianï¿½-Eulerian methods. *Journal of Computational Physics*, 230:6664–6687, 2011.

[17] S. Bertoluzza, S. D. Pino, and E. Labourasse. A conservative slide line method for cell-centered semi-Lagrangian and ALE schemes in 2D. *ESAIM: Mathematical Modelling and Numerical Analysis (M2AN)*, 50:187–214, 2016.

[18] G. Blanchard and R. Loubère. High order accurate conservative remapping scheme on polygonal meshes using a posteriori MOOD limiting. *Computers and Fluids*, 136:83–103, 2016.

[19] P. Bochev, D. Ridzal, and M. J. Shashkov. Fast optimization-based conservative remap of scalar fields through aggregate mass transfer. *Journal of Computational Physics*, 246:37–57, 2013.

[20] D. L. Book and I. B. Bernstein. Fluid instabifities of a uniformly imploding ablatively driven shell. *Journal of Plasma Physics*, 23(3):521–533, 1980.

[21] W. Boscheri, D. Balsara, and M. Dumbser. Lagrangian ADER-WENO Finite Volume Schemes on Unstructured Triangular Meshes Based On Genuinely Multidimensional HLL Riemann Solvers. *Journal of Computational Physics*, 267:112–138, 2014.

[22] W. Boscheri and M. Dumbser. Arbitrary–Lagrangian–Eulerian One–Step WENO Finite Volume Schemes on Unstructured Triangular Meshes. *Communications in Computational Physics*, 14:1174–1206, 2013.

[23] W. Boscheri and M. Dumbser. A direct Arbitrary-Lagrangian-Eulerian ADER-WENO finite volume scheme on unstructured tetrahedral meshes for conservative and non-conservative hyperbolic systems in 3d. *Journal of Computational Physics*, 275:484 – 523, 2014.

[24] W. Boscheri and M. Dumbser. Arbitrary–Lagrangian–Eulerian discontinuous Galerkin schemes with a posteriori subcell finite volume limiting on moving unstructured meshes. *Journal of Computational Physics*, 346:449–479, 2017.

[25] W. Boscheri and M. Dumbser. Arbitrary-Lagrangian-Eulerian discontinuous Galerkin schemes with a posteriori subcell finite volume limiting on moving unstructured meshes. *Journal of Computational Physics*, 346:449 – 479, 2017.

[26] W. Boscheri, M. Dumbser, and D. Balsara. High Order Lagrangian ADER-WENO Schemes on Unstructured Meshes – Application of Several Node Solvers to Hydrodynamics and Magnetohydrodynamics. *International Journal for Numerical Methods in Fluids*, 76:737–778, 2014.

[27] W. Boscheri, M. Dumbser, and O. Zanotti. High Order Cell-Centered Lagrangian-Type Finite Volume Schemes with Time-Accurate Local Time Stepping on Unstructured Triangular Meshes. *Journal of Computational Physics*, 291:120–150, 2015.

[28] N. Botta, R. Klein, S. Langenberg, and S. Lützenkirchen. Well balanced finite volume methods for nearly hydrostatic flows. *Journal of Computational Physics*, 196:539–565, 2004.

[29] F. Bouchut. *Nonlinear stability of finite Volume Methods for hyperbolic conservation laws: And Well-Balanced schemes for sources.* Springer Science & Business Media, 2004.

[30] J. Breil, T. Harribey, P. Maire, and M. Shashkov. A multi-material ReALE method with MOF interface reconstruction. *Computers and Fluids*, 83:115–125, 2013.

[31] E. J. Caramana. The implementation of slide lines as a combined force and velocity boundary condition. *Journal of Computational Physics*, 228:3911–3916, 2009.

[32] E. J. Caramana, D. E. Burton, M. J. Shashkov, and P. P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *Journal of Computational Physics*, 146:227–262, 1998.

[33] E. J. Caramana and M. J. Shashkov. Elimination of artificial grid distorsion and hourglass–type motions by means of lagrangian subzonal masses and pressures. *Journal of Computational Physics*, 142:521–561, 1998.

[34] G. Carré, S. D. Pino, B. Després, and E. Labourasse. A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. *Journal of Computational Physics*, 228:5160–5183, 2009.

[35] C. C. Castro and E. F. Toro. Solvers for the high-order riemann problem for hyperbolic balance laws. *Journal of Computational Physics*.

[36] M. Castro, T. M. de Luna, and C. Parés. Well-balanced schemes and path-conservative numerical methods. In *Handbook of Numerical Analysis*, volume 18, pages 131–175. Elsevier, 2017.

[37] M. Castro, J. Gallardo, J. López, and C. Parés. Well-balanced high order extensions of godunov's method for semilinear balance laws. *SIAM Journal of Numerical Analysis*, 46:1012–1039, 2008.

[38] M. Castro, J. Gallardo, and C. Parés. High-order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products. Applications to shallow-water systems. *Mathematics of Computation*, 75:1103–1134, 2006.

[39] M. Castro, J. Macías, and C. Parés. A q-scheme for a class of systems of coupled conservation laws with source term. application to a two-layer 1-d shallow water system. *ESAIM: Mathematical Modelling and Numerical Analysis*, 35(1):107–127, 2001.

[40] M. Castro, A. Pardo, C. Parés, and E. Toro. On some fast well-balanced first order solvers for nonconservative systems. *Mathematics of computation*, 79(271):1427–1472, 2010.

[41] M. J. Castro, E. Fernández, A. Ferriero, J. A. García, and C. Parés. High order extensions of Roe schemes for two dimensional nonconservative hyperbolic systems. *Journal of Scientific Computing*, 39:67–114, 2009.

[42] M. J. Castro, J. M. Gallardo, and A. Marquina. Approximate Osher–Solomon schemes for hyperbolic systems. *Applied Mathematics and Computation*, 272:347–368, 2016.

[43] M. J. Castro, P. G. LeFloch, M. L. Muñoz-Ruiz, and C. Parés. Why many theories of shock waves are necessary: Convergence error in formally path-consistent schemes. *Journal of Computational Physics*, 227(17):8107–8129, 2008.

[44] M. J. Castro, A. Pardo Milanés, and C. Parés. Well-balanced numerical schemes based on a generalized hydrostatic reconstruction technique. *Mathematical Models and Methods in Applied Sciences*, 17(12):2055–2113, 2007.

[45] M. Castro Díaz, T. Chacón Rebollo, E. D. Fernández-Nieto, and C. Pares. On well-balanced finite volume methods for nonconservative nonhomogeneous hyperbolic systems. *SIAM Journal on Scientific Computing*, 29(3):1093–1126, 2007.

[46] M. J. Castro Díaz and E. D. Fernández-Nieto. A class of computationally fast first order finite volume solvers: Pvm methods. *SIAM J. Scientific Computing*, 34(4), 2012.

[47] J. R. Cavalcanti, M. Dumbser, D. da Motta-Marques, and C. R. F. Junior. A conservative finite volume scheme with time-accurate local time stepping for scalar transport on unstructured grids. *Advances in Water Resources*, 86:217–230, 2015.

[48] P. Chandrashekar and C. Klingenberg. A second order well-balanced finite volume scheme for Euler Equations with gravity. *Journal on Scientific Computing* , 37:B382–B402, 2015.

[49] J. Cheng and C. Shu. A high order ENO conservative Lagrangian type scheme for the compressible Euler equations. *Journal of Computational Physics*, 227:1567–1596, 2007.

[50] J. Cheng and E. Toro. A 1D conservative Lagrangian ADER scheme. *Chinese Journal of Computational Physics*, 30:501–508, 2013.

[51] G. Clair, B. Despres, and E. Labourasse. A new method to introduce constraints in cell-centered Lagrangian schemes. *Computer Methods in Applied Mechanics and Engineering*, 261-262:56–65, 2013.

[52] G. Clair, B. Despres, and E. Labourasse. A one-mesh method for the cell-centered discretization of sliding. *Computer Methods in Applied Mechanics and Engineering*, 269:315–333, 2014.

[53] A. Colagrossi and M. Landrini. Numerical simulation of interfacial flows by smoothed particle hydrodynamics. *Journal of computational physics*, 191(2):448–475, 2003.

[54] P. Colella. A direct eulerian muscl scheme for gas dynamics. *SIAM J. Sci. Statist. Comput.*, 6:104 – 117, 1985.

[55] G. Dal Maso, P. G. LeFloch, and F. Murat. Definition and weak stability of nonconservative products. *J. Math. Pures Appl.* , 74:483–548, 1995.

[56] B. Després and E. Labourasse. Angular momentum preserving cell-centered lagrangian and eulerian schemes on arbitrary grids. *Journal of Computational Physics*, 290:28–54, 2015.

[57] B. Després and C. Mazeran. Symmetrization of Lagrangian gas dynamic in dimension two and multimdimensional solvers. *C.R. Mecanique*, 331:475–480, 2003.

[58] B. Després and C. Mazeran. Lagrangian gas dynamics in two-dimensions and Lagrangian systems. *Archive for Rational Mechanics and Analysis*, 178:327–372, 2005.

[59] V. Desveaux, M. Zenk, C. Berthon, and C. Klingenberg. A well-balanced scheme for the euler equation with a gravitational potential. In *Finite Volumes for Complex Applications VII-Methods and Theoretical Aspects*, pages 217–226. Springer, 2014.

[60] V. Desveaux, M. Zenk, C. Berthon, and C. Klingenberg. A well-balanced scheme to capture non-explicit steady states in the euler equations with gravity. *International Journal for Numerical Methods in Fluids*, 81(2):104–127, 2016.

[61] V. A. Dobrev, T. E. Ellis, T. Kolev, and R. N. Rieben. Curvilinear Finite elements for Lagrangian hydrodynamics. *International Journal for Numerical Methods in Fluids*, 65:1295–1310, 2011.

[62] V. A. Dobrev, T. E. Ellis, T. Kolev, and R. N. Rieben. High Order Curvilinear Finite Elements for Lagrangian Hydrodynamics. *SIAM Journal on Scientific Computing*, 34:606–641, 2012.

[63] V. A. Dobrev, T. E. Ellis, T. Kolev, and R. N. Rieben. High Order Curvilinear Finite Elements for axisymmetric Lagrangian Hydrodynamics. *Computers and Fluids*, 83:58–69, 2013.

[64] J. K. Dukowicz and B. J. Meltz. Vorticity errors in multidimensional lagrangian codes. *Journal of Computational Physics*, 99(1):115–134, 1992.

[65] M. Dumbser. A simple two-phase method for the simulation of complex free surface flows. *Computer Methods in Applied Mechanics and Engineering*, 200(9):1204–1219, 2011.

[66] M. Dumbser. Arbitrary–Lagrangian–Eulerian ADER–WENO finite volume schemes with time–accurate local time stepping for hyperbolic conservation laws. *Computer Methods in Applied Mechanics and Engineering*, 280:57–83, 2014.

[67] M. Dumbser, D. Balsara, E. F. Toro, and C. D. Munz. A unified framework for the construction of one–step finite–volume and discontinuous Galerkin schemes. *Journal of Computational Physics*, 227:8209–8253, 2008.

[68] M. Dumbser and D. S. Balsara. A new, efficient formulation of the HLLEM Riemann solver for general conservative and non-conservative

hyperbolic systems. *Journal of Computational Physics*, 304:275–319, 2016.

[69] M. Dumbser and W. Boscheri. High-order unstructured lagrangian one-step weno finite volume schemes for non-conservative hyperbolic systems: applications to compressible multi-phase flows. *Computers & Fluids*, 86:405–432, 2013.

[70] M. Dumbser, M. Castro, C. Parés, and E. Toro. ADER schemes on unstructured meshes for non-conservative hyperbolic systems: Applications to geophysical flows. *Computers and Fluids*, 38:1731–ï¿½1748, 2009.

[71] M. Dumbser, M. Castro, C. Parés, and E. F. Toro. Ader schemes on unstructured meshes for nonconservative hyperbolic systems: Applications to geophysical flows. *Computers & Fluids*, 38(9):1731–1748, 2009.

[72] M. Dumbser, C. Enaux, and E. F. Toro. Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. *Journal of Computational Physics*, 227:3971–4001, 2008.

[73] M. Dumbser, A. Hidalgo, M. Castro, C. Parés, and E. F. Toro. Force schemes on unstructured meshes ii: Non-conservative hyperbolic systems. *Computer Methods in Applied Mechanics and Engineering*, 199(9):625–647, 2010.

[74] M. Dumbser and E. F. Toro. On universal Osher–type schemes for general nonlinear hyperbolic conservation laws. *Communications in Computational Physics*, 10:635–671, 2011.

[75] M. Dumbser and E. F. Toro. A simple extension of the Osher Riemann solver to non-conservative hyperbolic systems. *Journal of Scientific Computing*, 48:70–88, 2011.

[76] M. Dumbser and E. F. Toro. A simple extension of the osher riemann solver to non-conservative hyperbolic systems. *Journal of Scientific Computing*, 48(1):70–88, 2011.

[77] M. Dumbser, A. Uuriintsetseg, and O. Zanotti. On Arbitrary–Lagrangian–Eulerian One–Step WENO Schemes for Stiff Hyperbolic Balance Laws. *Communications in Computational Physics*, 14:301–327, 2013.

[78] B. Einfeldt. On godunov-type methods for gas dynamics. *SIAM J. Numer. Anal.*, 25:294 – 318, 1988.

[79] B. Einfeldt, C. D. Munz, P. L. Roe, and B. Sjögreen. On godunov-type methods near low densities. *Journal of Computational Physics*, 92:273–295, 1991.

[80] F. Fambri, M. Dumbser, S. Köppel, L. Rezzolla, and O. Zanotti. ADER discontinuous Galerkin schemes for general-relativistic ideal magnetohydrodynamics. *Monthly Notices of the Royal Astronomical Society (MNRAS)*, 2018. submitted. https://arxiv.org/abs/1801.02839.

[81] A. Ferrari, M. Dumbser, E. F. Toro, and A. Armanini. A new 3d parallel sph scheme for free surface flows. *Computers & Fluids*, 38(6):1203–1217, 2009.

[82] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *Journal of Computational Physics*, 144:194–212, 1998.

[83] E. Gaburro, M. J. Castro, and M. Dumbser. Well-balanced arbitrary-lagrangian-eulerian finite volume schemes on moving nonconforming meshes for the euler equations of gas dynamics with gravity. *Monthly Notices of the Royal Astronomical Society*, 477(2):2251–2275, 2018.

[84] E. Gaburro, M. Dumbser, and M. J. Castro. Direct arbitrary-lagrangian-eulerian finite volume schemes on moving nonconforming unstructured meshes. *Computers and Fluids*, 159:254–275, 2017.

[85] E. Godlewski and P. A. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer, 1996.

[86] S. Godunov. Finite difference methods for the computation of discontinuous solutions of the equations of fluid dynamics. *Mathematics of the USSR: Sbornik*, 47:271–306, 1959.

[87] L. Gosse. A well-balanced flux-vector splitting scheme designed for hyperbolic systems of conservation laws with source terms. *Computers & Mathematics with Applications*, 39(9):135–159, 2000.

[88] L. Gosse. A well-balanced scheme using non-conservative products designed for hyperbolic systems of conservation laws with source terms. *Mathematical Models and Methods in Applied Sciences*, 11(02):339–365, 2001.

[89] J. Greenberg, A. Leroux, R. Baraille, and A. Noussair. Analysis and approximation of conservation laws with source terms. *SIAM Journal on Numerical Analysis*, 34(5):1980–2007, 1997.

[90] J. M. Greenberg and A.-Y. Leroux. A well-balanced scheme for the numerical processing of source terms in hyperbolic equations. *SIAM Journal on Numerical Analysis*, 33(1):1–16, 1996.

[91] A. Harten, P. Lax, and B. van Leer. On upstream differencing and godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25(1):35–61, 1983.

[92] C. Hirsch. *Numerical Computation of Internal and External Flows Vol I: Fundamentals of Numerical Discretisation.* Wiley, 1988.

[93] C. Hirsch. *Numerical Computation of Internal and External Flows Vol II: Computational Methods for Inviscid and Viscous Flow.* Wiley, 1988.

[94] C. Hu and C. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150:97–127, 1999.

[95] C. Hu and C.-W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150(1):97–127, 1999.

[96] S. C. Jardin. Review of implicit methods for the magnetohydrodynamic description of magnetically confined plasmas. *Journal of Computational Physics*, 231(3):822–838, 2012.

[97] R. Kaeppeli and S. Mishra. Structure preserving schemes. *ETH Zürich*, 2014.

[98] R. Käppeli and S. Mishra. Well-balanced schemes for the euler equations with gravitation. *Journal of Computational Physics*, 259:199–219, 2014.

[99] R. Käppeli and S. Mishra. A well-balanced finite volume scheme for the euler equations with gravitation. *Astronomy and Astrophysics*, 587:A94, 2016.

[100] D. Kröner. *Numerical Schemes for Conservation Laws*. Wiley - Teubner, 1997.

[101] M. Kucharik, R. Liska, J. Limpouch, and P. Váchal. ALE simulations of high-velocity impact problem. *Journal of Computational Physics*, 76:737–778, 2014.

[102] M. Kucharik, R. Loubére, L. Bednàrik, and R. Liska. Enhancement of Lagrangian slide lines as a combined force and velocity boundary condition. *Computers & Fluids*, 83:3–14, 2013.

[103] M. Kucharik and M. J. Shashkov. One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian-Eulerian methods. *Journal of Computational Physics*, 231:2851–2864, 2012.

[104] P. Lax and B. Wendroff. Systems of conservation laws. *Communications in Pure and Applied Mathematics*, 13:217–237, 1960.

[105] P. D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical approximation. *Comm. Pure Appl. Math.*, 7:159–193, 1954.

[106] R. J. LeVeque. Balancing source terms and flux gradients in high-resolution godunov methods: the quasi-steady wave-propagation algorithm. *Journal of computational physics*, 146(1):346–365, 1998.

[107] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[108] Z. Li, X. Yu, and Z. Jia. The cell–centered discontinuous Galerkin method for Lagrangian compressible Euler equations in two dimensions. *Computers and Fluids*, 96:152–164, 2014.

[109] W. Liu, J. Cheng, and C. Shu. High order conservative Lagrangian schemes with Lax–Wendroff type time discretization for the compressible Euler equations. *Journal of Computational Physics*, 228:8872–8891, 2009.

[110] R. Loubère, M. Dumbser, and S. Diot. A new family of high order unstructured MOOD and ADER finite volume schemes for multidimensional systems of hyperbolic conservation law. *Communications in Computational Physics*, 16:718–763, 2014.

[111] R. Loubère, P. Maire, and P. Váchal. 3D staggered Lagrangian hydrodynamics scheme with cell-centered Riemann solver-based artificial viscosity. *International Journal for Numerical Methods in Fluids*, 72:22–42, 2013.

[112] R. Loubère, P. H. Maire, and P. Váchal. A second-order compatible staggered Lagrangian hydrodynamics scheme using a cell-centered multidimensional approximate Riemann solver. *Procedia Computer Science*, 1:1931–1939, 2010.

[113] R. Loubère and M. Shashkov. A subcell remapping method on staggered polygonal grids for arbitrary–lagrangian–eulerian methods. *Journal of Computational Physics*, 23:155–160, 2004.

[114] P. Maire. A high-order cell-centered lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *Journal of Computational Physics*, 228:2391–2425, 2009.

[115] P. Maire. A unified sub-cell force-based discretization for cell-centered lagrangian hydrodynamics on polygonal grids. *International Journal for Numerical Methods in Fluids*, 65:1281–1294, 2011.

[116] P. H. Maire. A high-order one-step sub-cell force-based discretization for cell-centered lagrangian hydrodynamics on polygonal grids. *Computers and Fluids*, 46(1):341–347, 2011.

[117] P. H. Maire, R. Abgrall, J. Breil, and J. Ovadia. A cell-centered lagrangian scheme for two-dimensional compressible flow problems. *SIAM Journal on Scientific Computing*, 29:1781–1824, 2007.

[118] J. M. Mantas, M. De la Asunción, and M. J. Castro. An introduction to gpu computing for numerical simulation. In *Numerical Simulation in Physics and Engineering*, pages 219–251. Springer, 2016.

[119] G. D. Maso, P. LeFloch, and F. Murat. Definition and weak stability of nonconservative products. *J. Math. Pures Appl.*, 74:483–548, 1995.

[120] A. Mignone, G. Bodo, S. Massaglia, T. Matsakos, O. Tesileanu, C. Zanni, and A. Ferrari. Pluto: A numerical code for computational astrophysics. *The Astrophysical Journal Supplement Series*, 170(1):228, 2007.

[121] A. Mignone, M. Flock, M. Stute, S. Kolb, and G. Muscianisi. A conservative orbital advection scheme for simulations of magnetized shear flows with the pluto code. *Astronomy & Astrophysics*, 545:A152, 2012.

[122] A. Mignone, C. Zanni, P. Tzeferacos, B. Van Straalen, P. Colella, and G. Bodo. The pluto code for adaptive mesh computations in astrophysical fluid dynamics. *The Astrophysical Journal Supplement Series*, 198(1):7, 2011.

[123] R. Millington, E. Toro, and L. Nejad. *Arbitrary High Order Methods for Conservation Laws I: The One Dimensional Scalar Case*. PhD thesis, Manchester Metropolitan University, Department of Computing and Mathematics, June 1999.

[124] C. Munz. On Godunov–type schemes for Lagrangian gas dynamics. *SIAM Journal on Numerical Analysis*, 31:17–42, 1994.

[125] A. H. Oort. Angular momentum cycle in the atmosphere-ocean-solid earth system. *Bulletin of the American Meteorological Society*, 70(10):1231–1242, 1989.

[126] A. L. Ortega and G. Scovazzi. A geometrically–conservative, synchronized, flux–corrected remap for arbitrary Lagrangian–Eulerian computations with nodal finite elements. *Journal of Computational Physics*, 230:6709–6741, 2011.

[127] S. Osher and F. Solomon. Upwind difference schemes for hyperbolic conservation laws. *Math. Comput.*, 38:339–374, 1982.

[128] C. Parés. Numerical methods for nonconservative hyperbolic systems: a theoretical framework. *SIAM Journal on Numerical Analysis*, 44:300–321, 2006.

[129] C. Parés. Numerical methods for nonconservative hyperbolic systems: a theoretical framework. *SIAM Journal on Numerical Analysis*, 44(1):300–321, 2006.

[130] C. Parés and M. Castro. On the well-balance property of roe's method for nonconservative hyperbolic systems. applications to shallow-water systems. *ESAIM: mathematical modelling and numerical analysis*, 38(5):821–852, 2004.

[131] B. Perthame and C. Simeoni. A kinetic scheme for the saint-venant system with a source term. *Calcolo*, 38(4):201–231, 2001.

[132] B. Perthame and C. Simeoni. Convergence of the upwind interface source method for hyperbolic conservation laws. In *Hyperbolic problems: theory, numerics, applications*, pages 61–78. Springer, 2003.

[133] I. Petropoulos, M. Costes, and P. Cinnella. Development and analysis of high-order vorticity confinement schemes. *Computers & Fluids*, 2017.

[134] S. D. Pino. A curvilinear finite-volume method to solve compressible gas dynamics in semi-Lagrangian coordinates. *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, 348:1027–1032, 2010.

[135] T. C. Rebollo, A. D. Delgado, and E. D. F. Nieto. A family of stable numerical solvers for the shallow water equations with source terms. *Computer methods in applied mechanics and engineering*, 192(1):203–225, 2003.

[136] T. C. Rebollo, A. D. Delgado, and E. D. F. Nieto. Asymptotically balanced schemes for non-homogeneous hyperbolic systems–application to the shallow water equations. *Comptes Rendus Mathematique*, 338(1):85–90, 2004.

[137] S. Rhebergen, O. Bokhove, and J. van der Vegt. Discontinuous Galerkin finite element methods for hyperbolic nonconservative partial differential equations. *Journal of Computational Physics*, 227:1887–1922, 2008.

[138] P. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.

[139] P. Roe and B. Morton. Preserving vorticity in finite-volume schemes. *Finite Volume For Complex Applications. Volum 2, Problems and Perspectives*, 2:347, 1999.

[140] V. V. Rusanov. Calculation of Interaction of Non–Steady Shock Waves with Obstacles. *J. Comput. Math. Phys. USSR*, 1:267–279, 1961.

[141] K. Schaal, A. Bauer, P. Chandrashekar, R. Pakmor, C. Klingenberg, and V. Springel. Astrophysical hydrodynamics with a high-order discontinuous galerkin scheme and adaptive mesh refinement. *Monthly Notices of the Royal Astronomical Society*, 453(4):4278–4300, 2015.

[142] G. Scovazzi. Lagrangian shock hydrodynamics on tetrahedral meshes: A stable and accurate variational multiscale approach. *Journal of Computational Physics*, 231:8029–8069, 2012.

[143] J. Shi, C. Hu, and C. Shu. A technique of treating negative weights in WENO schemes. *Journal of Computational Physics*, 175:108–127, 2002.

[144] M. A. Skinner and E. C. Ostriker. The athena astrophysical magnetohydrodynamics code in cylindrical geometry. *The Astrophysical Journal Supplement Series*, 188(1):290, 2010.

[145] T. Sonar. On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations: polynomial recovery, accuracy and stencil selection. *Computer Methods in Applied Mechanics and Engineering*, 140:157–181, 1997.

[146] V. Springel. E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Monthly Notices of the Royal Astronomical Society (MNRAS)*, 401:791–851, 2010.

[147] H. Tang, T. Tang, and K. Xu. A gas-kinetic scheme for shallow-water equations with source terms. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 55(3):365–382, 2004.

[148] V. Titarev and E. Toro. ADER: Arbitrary high order Godunov approach. *Journal of Scientific Computing*, 17(1-4):609–618, December 2002.

[149] V. Titarev and E. Toro. ADER schemes for three-dimensional nonlinear hyperbolic systems. *Isaac Newton Institute for Mathematical Sciences Preprint Series*, 2003.

[150] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, second edition, 1999.

[151] E. F. Toro. *Shock-capturing methods for free-surface shallow flows*. John Wiley, 2001.

[152] E. F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the Harten-Lax-van Leer Riemann solver. *Journal of Shock Waves*, 4:25–34, 1994.

[153] I. Toumi. A weak formulation of Roe's approximate Riemann solver. *Journal of Computational Physics*, 102:360–373, 1992.

[154] E. F. Truckenbrodt. *Bd. 1: Grundlagen und elementare Strömungsvorgänge dichtebeständiger Fluide*. Springer, 1980.

[155] B. van Leer. Towards the ultimate conservative difference scheme II: Monotonicity and conservation combined in a second order scheme. *Journal of Computational Physics*, 14:361–370, 1974.

[156] B. van Leer. Towards the ultimate conservative difference scheme V: A second order sequel to Godunov's method. *Journal of Computational Physics*, 32:101–136, 1979.

[157] F. Vilar. Cell-centered discontinuous Galerkin discretization for two-dimensional Lagrangian hydrodynamics. *Computers and Fluids*, 64:64–73, 2012.

[158] F. Vilar, P. H. Maire, and R. Abgrall. Cell-centered discontinuous Galerkin discretizations for two-dimensional scalar conservation laws on unstructured grids and for one-dimensional Lagrangian hydrodynamics. *Computers and Fluids*, 46(1):498–604, 2010.

[159] F. Vilar, P. H. Maire, and R. Abgrall. A discontinuous Galerkin discretization for solving the two-dimensional gas dynamics equations written under total Lagrangian formulation on general unstructured grids. *Journal of Computational Physics*, 276:188–234, 2014.

[160] J. von Neumann and R. Richtmyer. A method for the calculation of hydrodynamics shocks. *Journal of Applied Physics*, 21:232–237, 1950.

[161] M. L. Wilkins. Calculation of elastic-plastic flow. *Methods in Computational Physics*, 3, 1964.

[162] Y. V. Yanilkin, E. A. Goncharov, V. Kolobyanin, V. Sadchikov, J. R. Kamm, M. J. Shashkov, and W. J. Rider. Multi-material pressure relaxation methods for lagrangian hydrodynamics. *Computers and Fluids*, 83:137–143, 2013.

List of Tables

List of Figures