



UNIVERSITÀ DEGLI STUDI  
DI TRENTO

---

DEPARTMENT OF CIVIL, ENVIRONMENTAL AND MECHANICAL  
ENGINEERING

**Doctoral School in Department of Civil, Environmental and Mechanical  
Engineering**

XXIX CYCLE

NUMERICAL METHODS FOR COMPRESSIBLE MULTI-PHASE  
FLOWS WITH SURFACE TENSION

**Nguyen Tri Nguyen**

Advisor

Prof. Michael Dumbser

Università degli Studi di Trento

---

April 2017



# Abstract

In this thesis we present a new and accurate series of computation methods for compressible multi-phase flows with capillary effects based upon the full seven-equation Baer-Nunziato model. For that reason, there are some numerical methods to obtain high accuracy solutions, which will be shown here. First, a high resolution shock capturing Total Variation Diminishing (TVD) finite volume scheme is used on both Cartesian and unstructured triangular grids. Regarding the TVD finite volume scheme on the unstructured grid, time-accurate local time stepping (LTS) is applied to compute the solutions of the governing PDE system, in which the results are also compared with time-accurate global time stepping. Second, we propose a novel high order accurate numerical method for the solution of the seven equation Baer-Nunziato model based on ADER discontinuous Galerkin (DG) finite element schemes combined with *a posteriori* subcell finite volume limiting and adaptive mesh refinement (AMR).

In multi-phase flows, the difficulty is to design accurate numerical methods for resolving the phase interface, as well as the simulation of the phenomena occurring at the interface, such as surface tension effects, heat transfer and friction. This is because of the interactions of the fluids at the phase interface and its complex geometry. So the accurate simulation of compressible multi-phase flows with surface tension effects is currently still one of the most challenging problems in computational fluid dynamics (CFD). In this work, we present a novel path-conservative finite volume discretization of the continuum surface force method (CSF) of Brackbill et al. to account for the surface tension effect due to curvature of the phase interface. This is achieved in the context of a diffuse interface approach, based on the seven equation BaerNunziato model of compressible multi-phase flows. Such diffuse interface methods for compressible multi-phase flows including capillary effects have first been proposed by Perigaud and Saurel. Regarding the high order accuracy of a diffuse interface approach, the interface is captured and allowed to travel across one single possibly refined cell, and is computed in the context of multi-dimensional

high accurate space/time DG schemes with AMR and a posteriori sub-cell stabilization. The surface tension terms of the CSF approach are considered as a part of the non-conservative hyperbolic system. We propose to integrate the CSF source term as a non-conservative product and *not* simply as a source term, following the ideas on path-conservative finite volume schemes put forward by Castro and Parés. For the validation of the current numerical methods, we compare our numerical results with those published previously in the literature.

THE MEMORY OF MY GRANDMOTHER PHAM THI CAT (1936 - 2014).



# Acknowledgements

My sincere appreciation and deep respect to my supervisor, Professor Michael Dumbser, for his unlimited support with continuous and patient advice during my studies. Much of this work could not have been achieved without his ideas, encouragement, thorough review, and dedication.

I would like to thank all the members of the numerical analysis group for helping me one way or another during my studies at DICAM, Trento University, Italy. I also would like to thank Trento University for providing the funding for supporting my research.

Last but not least, I would like to express my loving thanks to my wife Vu Thi Kim Yen, my daughter Nguyen Vu Thien Minh, and my parents. They always helped me in sharing my difficulties throughout the years of my studies.





# Contents

<b>List of Tables</b>	<b>ix</b>
<b>List of Figures</b>	<b>xii</b>
<b>Symbols</b>	<b>xvii</b>
<b>Abbreviations</b>	<b>xix</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Introduction . . . . .	1
1.2. The compressible Euler and Navier-Stokes equations for single phase flows	6
1.2.1. 1D Euler equations in conservation form . . . . .	8
1.2.2. 1D Euler equations in physical form . . . . .	9
1.3. Baer Nunziato equations . . . . .	10
1.3.1. The seven-equation Baer-Nunziato model with relaxation, surface tension, viscosity and gravity effects . . . . .	11
1.3.2. Eigenstructure of the Baer-Nunziato model in 1D . . . . .	15
1.4. Young-Laplace Equation . . . . .	18
1.5. Overview of the thesis . . . . .	19
<b>2. Path-conservative Finite Volume Schemes</b>	<b>21</b>
2.1. Finite volume methods . . . . .	21
2.2. Conservation laws in generalized coordinates . . . . .	22
2.3. Total-Variation-Diminishing Schemes . . . . .	24
2.4. Path-conservation Finite volume schemes . . . . .	25
2.4.1. Path-conservation Finite volume schemes for the 2D Baer-Nunziato equation on the Cartesian meshes . . . . .	25
2.4.2. Curvature computation . . . . .	28
2.4.3. Second order MUSCL-type method on unstructured meshes with global time stepping . . . . .	28
2.4.4. Local time stepping (LTS) . . . . .	31

<b>3. Riemann Solvers</b>	<b>35</b>
3.1. Introduction . . . . .	35
3.2. Riemann problem . . . . .	36
3.3. Approximate Riemann solvers . . . . .	38
3.3.1. Rusanov scheme (LLF) . . . . .	39
3.3.2. Osher-type scheme (DOT) . . . . .	39
3.3.3. Roe-type scheme . . . . .	39
3.4. Jump terms in the non-conservative product . . . . .	40
3.5. A path-conservative HLLEM scheme for non-conservative systems . . . . .	41
<b>4. High Order Extension</b>	<b>45</b>
4.1. Introduction . . . . .	45
4.2. ADER-DG AMR scheme with <i>a posteriori</i> sub-cell finite volume limiter . . . . .	47
4.3. ADER-DG scheme . . . . .	49
4.3.1. The local space-time predictor . . . . .	49
4.3.2. Fully discrete one-step ADER-DG scheme . . . . .	51
4.3.3. Timestep restriction, high order of accuracy, sub-cell resolution and <i>a posteriori</i> stabilization . . . . .	53
4.4. <i>A posteriori</i> sub-cell stabilization: detection and ADER-WENO recomputation . . . . .	53
4.4.1. Detection criteria . . . . .	54
4.4.2. Sub-cell ADER-WENO recomputation . . . . .	55
4.5. AMR framework . . . . .	56
4.6. Curvature computation . . . . .	58
4.6.1. Test #1: static curvature computation . . . . .	59
4.6.2. Test #2: simple dynamic curvature computation . . . . .	60
<b>5. Numerical Results</b>	<b>65</b>
5.1. One-dimensional tests . . . . .	65
5.1.1. TVD finite volume schemes. . . . .	65
5.1.2. Path-conservative HLLEM-type scheme. . . . .	66
5.2. Two dimensional tests . . . . .	72
5.2.1. TVD finite volume schemes on a two dimensional Cartesian grid . . . . .	72
5.2.1.1. Steady bubble in equilibrium and numerical mesh convergence study . . . . .	74
5.2.1.2. Droplet oscillation . . . . .	75

5.2.1.3.	Dynamics of a droplet under gravity and surface tension forces . . . . .	77
5.2.1.4.	A rising gas bubble under buoyancy forces . . . . .	79
5.2.1.5.	Head-on Collision of Binary Drops . . . . .	81
5.2.2.	Path-conservative HLLEM-type scheme. . . . .	81
5.2.2.1.	Numerical Convergence Results . . . . .	81
5.2.2.2.	Dynamics of a droplet under gravity and surface tension forces . . . . .	83
5.2.3.	TVD finite volume schemes on a two dimensional unstructured grid . . . . .	83
5.2.3.1.	Numerical convergence studies . . . . .	83
5.2.3.2.	2D Riemann problems . . . . .	86
5.3.	High order extension . . . . .	87
5.3.1.	Numerical convergence studies . . . . .	92
5.3.2.	Riemann problems . . . . .	93
5.3.3.	Young-Laplace law . . . . .	96
<b>6.</b>	<b>Conclusion</b>	<b>105</b>
<b>A.</b>	<b>Appendix</b>	<b>107</b>
A.1.	Local time stepping . . . . .	107
A.2.	Boundary conditions . . . . .	108



# List of Tables

5.1. Initial state left(L) and right(R) for the Baer-Nunziato problem solved in 1D with the surface tension effect. . . . .	66
5.2. Numerical verification of the exact well-balanced property of the DOT Riemann solver for the steady bubble in equilibrium in 1D for different machine precisions. The $L_\infty$ errors refer to the velocities of the two phases $u_1$ and $u_2$ , respectively. In both tests the exact curvature $\kappa = -1$ and the exact pressure jump $\Delta p = \sigma\kappa$ have been prescribed. . . . .	72
5.3. Numerical convergence results for the compressible Baer-Nunziato equations with surface tension using the DOT Riemann solver. The error norms refer to the pressure jump at time $t = 0.5$ s. . . . .	75
5.4. Numerical convergence results for the compressible Baer-Nunziato equations with surface tension using three different numerical schemes. The error norms refer to the pressure jump at time $t = 0.5$ s. . . . .	83
5.5. Numerical convergence results for the compressible Baer-Nunziato equations with comparing global time stepping (GTS) with local time stepping (LTS) in terms of errors. The error norms refer to the variable $\rho_1 u_1 \alpha_1$ at time $t = 10.0$ . . . . .	86
5.6. Initial states left (L) and right (R) for the Riemann problems solved in 2D with the Baer-Nunziato model. Values for $\gamma_i$ , $\pi_i$ and the final time $t_e$ are also given. . . . .	87
5.7. Summary of the main components of our code along with associated features and key words. . . . .	92
5.8. $L^1$ , $L^2$ and $L^\infty$ errors and convergence rates for the 2D isentropic vortex problem for the ADER-DG- $\mathbb{P}_N$ scheme with sub-cell limiter and adaptive mesh refinement for variable $\rho_s$ at final time 10 using $CFL = 1/2$ . . . . .	94
5.9. Initial states left (L) and right (R) for the Riemann problems solved in 2D. Values for $\gamma_i$ , $\pi_i$ and the final time $t_e$ are also given. . . . .	97



# List of Figures

1.1.	The strategy of computer simulations . . . . .	2
1.2.	The wave structure of an idealized Riemann problem for the seven-equation multi-phase flow model. . . . .	18
2.1.	Piecewise constant data representation in a first order finite volume scheme (left) and piecewise linear data (right) in the second order finite volume MUSCL scheme. . . . .	24
3.1.	The linear Riemann problem with initial conditions (left) and Riemann fan for the one-dimensional Euler equations(right). . . . .	37
3.2.	Piecewise constant data representation in the finite volume Godunov's scheme. . . . .	37
4.1.	Curvature computation in the case of a static initial 2D ellipsoidal bubble. Top-left: solid volume fraction. Top-right: curvature. Bottom-left: limiter. Bottom-right: zoom on the mesh (thick line), number of degree of freedom in DG cell (submesh in blue cells) and true sub-cell resolution used in limited red cells. . . . .	61
4.2.	Curvature computation in the case of an evolving 2D initial ellipsoidal bubble. Solid volume fraction. From top-left to bottom-right intermediate times. Last panel is at later time after several oscillations. . . . .	62
4.3.	Curvature computation in the case of an evolving 2D initial ellipsoidal bubble. Limiter value (color) and solid volume fraction (elevation). Right panels are at later time after several oscillations whereas the first two are after one oscillation. Bottom panels present a zoom on the submesh and of the solid volume fraction to show the underlying resolution. . . . .	63
5.1.	Numerical results for the two phase RP1 at time $t = 0.15$ s. . . . .	67
5.2.	Numerical results for the two phase RP2 at time $t = 0.15$ s. . . . .	68
5.3.	Numerical results for the two phase RP3 at time $t = 0.15$ s. . . . .	69
5.4.	Numerical results for the two phase RP4 at time $t = 0.15$ s. . . . .	70

5.5. Numerical results for the two phase RP4 at time $t = 0.15$ s. . . . .	71
5.6. Numerical results for Riemann problems RP1-RP5 (from top to bottom). Left: Solid density $\rho_1$ . Right: Gas density $\rho_2$ . . . . .	74
5.7. 3D surface plot for the mixture pressure obtained with a path-conservative Osher-type scheme corresponding to four different mesh sizes listed in Table 5.3. . . . .	76
5.8. The oscillations of the droplet in time . . . . .	78
5.9. The oscillations of the droplet in time . . . . .	78
5.10. Drop falling under gravity effect and break-up. . . . .	79
5.11. Interface positions in form of volume fraction contour ( $\alpha_2$ ) for a rising 2D bubble. Mesh size is $100 \times 200$ . . . . .	80
5.12. Transient flow visualization of a 2D binary drop collision. Mesh size is $200 \times 200$ . . . . .	82
5.13. 3D surface plot for the pressure obtained with the HLLEM-type Riemann solver corresponding to four different mesh sizes listed in Table 5.4. . . . .	84
5.14. Drop falling under gravity effect and break-up. . . . .	85
5.15. Results for Riemann problem RP1 on the unstructured grid. . . . .	88
5.16. Results for Riemann problem RP2 on the unstructured grid. . . . .	89
5.17. Results for Riemann problem RP3 on the unstructured grid. . . . .	90
5.18. Results for Riemann problem RP4 on the unstructured grid. . . . .	91
5.19. 2D isentropic vortex problem — Error in logscale as a function of the number of degrees of freedom for the DG schemes tested in table 5.8, from 3rd order up to 10th order of accuracy. . . . .	95
5.20. Results for Riemann problem RP1. . . . .	98
5.21. Results for Riemann problem RP2. . . . .	99
5.22. Results for Riemann problem RP3. . . . .	100
5.23. Results for Riemann problem RP4. . . . .	101
5.24. Sketch of the expected behavior for the bubble deformation test starting from an ellipse bubble. A loss of kinetic energy is expected in the dissipative case (green) whereas in an (ideal) non dissipative case the energy remains the same (blue). . . . .	103
5.25. Laplace law test case — 2D initial ellipse bubble — Kinetic energy as a function of time. . . . .	104
A.1. Illustration of the local time stepping algorithm with 3 different elements. .	108



A.2. Boundary conditions. Ghost cells outside the computational domain are created. . . . .	109
A.3. Sketch of the boundary conditions. Ghost cell outside the computational domain is taken, and it is based on a logically connected mesh. . . . .	110



## Symbols

$a$	Speed of sound
$c$	Color function
$d$	Number of space dimensions
$i$	Element index
$j$	Neighbor element index
$h$	Characteristic mesh size
$H$	The enthalpy
$M$	Maximum degree of the reconstruction
$n$	Current time level
$n + 1$	Future time level
$p$	Pressure
$p_I$	Interface pressure
$u_I$	Interface velocity
$t$	Time
$T$	Control volume
$u$	Velocity component in $x$ -direction
$v$	Velocity component in $y$ -direction
$w$	Velocity component in $z$ -direction
$x$	Horizontal direction in the physical system
$y$	Lateral direction in the physical system
$z$	Vertical direction in the physical system
$\gamma$	Ratio of specific heats
$\mu$	Dynamic viscosity
$\pi$	Material constant
$\delta_{ij}$	Kronecker symbol
$\Delta t$	Timestep
$\rho$	Density

$\alpha$	Volume fraction
$\kappa$	The curvature
$\sigma$	Surface tension coefficient
$\lambda$	Velocity relaxation
$\beta$	Pressure relaxation
$\mathbf{g}$	Gravity acceleration
$Z$	Acoustic impedance
$\Theta$	Dissipation matrix
$\rho E$	Total energy per mass unit
$\theta$	Space-time basis function and test function
$\psi$	Space basis function
$\xi$	Horizontal direction in the reference system
$\eta$	Lateral direction in the reference system
$\zeta$	Vertical direction in the reference system
$\tau$	Relative time with respect to timelevel $t^n$
$\tau$	Viscous shear stress tensor
$\Omega$	Computational domain
$\mathbf{B}$	Non-conservative nonlinear flux tensor
$\mathbf{F}$	Conservative nonlinear flux tensor
$\mathbf{I}$	Identity matrix
$l$	Level of refinement
$\mathbf{n}$	Element boundary normal vector in space
$\mathcal{N}_i$	Neumann neighborhood of element $i$
$\tau$	Refinement factor
$\mathbf{Q}$	Vector of conserved variables
$\mathbf{W}$	Vector of primitive variables
$\mathbf{S}$	Algebraic source term
$\mathbf{x}$	Physical spatial coordinate vector
$\tilde{\mathbf{x}}$	Physical space-time coordinate vector
$\xi$	Reference spatial coordinate vector
$\tilde{\xi}$	Reference space-time coordinate vector
$\mathbf{q}_h$	High order space-time predictor solution
$\mathbf{w}_h$	High order reconstruction polynomial

## Abbreviations

ADER	Arbitrary high order scheme using derivatives
ALE	Arbitrary-Lagrangian-Eulerian scheme
AMR	Adaptive Mesh Refinement
BN	Baer-Nunziato
CFD	Computational Fluid Dynamics
CG	Continuous Galerkin method
CFL	Courant-Friedrichs-Levy number
CPU	Central processing unit
CSF	Continuum surface force
DG	Discontinuous Galerkin method
ENO	Essentially non-oscillatory
FV	Finite Volume
GRP	Generalized Riemann problem
GTS	Glocal time stepping
LS	Level set
LTS	Local time stepping
MOOD	Multi-dimensional optimal order detection
MUSCL	Monotonic Upstream-Centered Scheme for Conservation Laws
NAD	Numerical Admissible Criteria
ODE	Ordinary differential equation
PAD	Physical Admissible Criteria
PLIC	Piecewise linear interface reconstruction
PPM	Piecewise parabolic method
PDE	Partial differential equation
RS	Riemann Solver
SPH	Smooth Particle Hydrodynamics
TEU	Total element updates
TVD	Total variation diminishing
VOF	Volume of fluid
WENO	Weighted essentially non-oscillatory

# 1. Introduction

## 1.1. Introduction

Multi-phase flow problems including surface tension and capillary effects are of great interest in mechanical, chemical and aerospace engineering. They appear in many industrial processes, such as liquid fuel sprays injected into car, air- and spacecraft engines; mixing processes in chemical engineering; breakup of liquid jets; condensation in nuclear reactors; off-shore engineering and bio-medical applications.

Laboratory experiments and computer simulations are the two main approaches for the study of complex flow problems. So far, most of the knowledge obtained is experimental, but this approach tends to be very complex and is subject to technical limitations when obtaining some measurements. Experimental data tends to include uncertainties that do not allow correct validation of the theory. As a consequence, numerical simulations with modern state of the art methods can help one to understand the complexity involved in these fluid flows more clearly without having to perform time consuming, expensive and complicated experiments. Advances in modern computers combined with advances in numerical schemes for the solution of the governing partial differential equations can provide important information, especially at complicated conditions where it can be difficult to extract measurements experimentally. This simulation approach is known as Computational Fluid Dynamics (CFD). In CFD, one has been trying to improve numerical schemes in order to simulate increasingly complex flows with greater accuracy and efficiency. The strategy of computer simulations is presented in Figure 1.1.

The simulation of the physics and the mechanics of the interaction of different materials is of vital importance for the application of multi-phase flows, such as in high-speed flows with droplets, bubbles and particles, and high pressure problems with strong shock waves. The interface between the two different fluids is the location where complex phenomena, such as heat and mass transfer and surface tension can occur. Capillary effects are also of increasing interest in nano-mechanics, for example for the design and analysis of the performance of new hydrophobic, non-wetting or self-cleaning surfaces. Therefore, in this thesis the effect of surface tension is carefully studied in the context of compressible

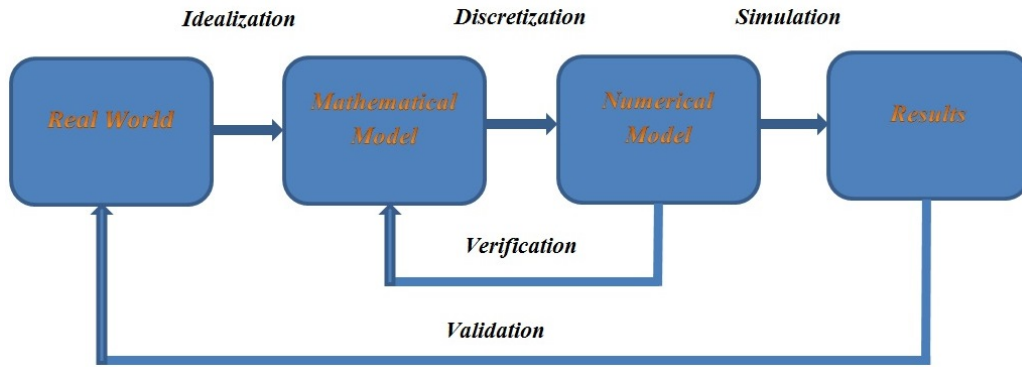


Figure 1.1.: The strategy of computer simulations

multi-phase flows.

In the past, many mathematical models and numerical methods have been developed for compressible and incompressible multi-phase flows. A first problem is to determine the location of the interface where the interactions between different fluids take place. Further difficulties arise when shock waves travel across the interface between two fluids. This is due to the fact that the numerical simulation of compressible multi-phase flows is far more complex than the simulation of single phase flows.

The numerical schemes used to solve the interface tracking problem can be classified into three basic categories: tracking methods (moving grid or Lagrangian approach), capturing methods (fixed grid or Eulerian approach) and combined methods.

In general, Lagrangian methods [14] explicitly track the interface of the two fluids via a moving mesh and thus always provide the exact location of the interface. However, mesh-based Lagrangian schemes are typically not suitable for multi-phase flows with complex vortex structures or where a complex merging and separation of the phase interface occurs. Thus, a completely different approach for the modeling of multi-phase flows with surface tension is based upon meshless Lagrangian particle schemes, such as the well-known Smooth Particle Hydrodynamics (SPH) method [110]. SPH is well-suited for the simulation of complex interface flows including surface tension, complex vortex flow and separation and merging of the phase interface. The SPH method provides excellent interface tracking capabilities, but it is also well known to exhibit several numerical instabilities, such as the tensile instability, which require artificial viscosity and other stabilization techniques. It is also important to note that SPH is computationally more expensive than most of the other methods. Furthermore, the method in general lacks even zeroth order consistency with the governing PDE. In recent years, novel SPH techniques have been developed to provide accurate and stable solutions for weakly compressible free surface

flows, see e.g [73, 158].

In Eulerian schemes based on a fixed mesh there are two very popular methods for the capturing of the phase interface, namely the volume of fluid (VOF) method [85] and the level set (LS) method [142, 115, 112]. A combination of VOF and LS proposed in [141, 154] has shown to produce a robust method for flows with complex geometries and interface deformations in the setting of incompressible fluids. The disadvantage of Eulerian methods applied to multiphase flow problems is a significant amount of numerical dissipation, which requires proper interface reconstruction techniques to avoid excessive smearing of the phase boundary and to restore a sharp interface. However, this may become rather cumbersome in complex configurations. The level set method also needs a periodic re-initialization to restore the signed distance function property of the level set function, which requires the additional solution of a Hamilton-Jacobi equation. Furthermore, VOF has difficulties in simulating highly compressible multi-phase flows, hence most of the applications of the VOF method are restricted to the simulation of incompressible fluids. While the VOF method is perfectly conservative, the level-set approach is not. Due to the piecewise linear interface reconstruction (PLIC) used in the VOF context and due to the signed distance function property of the level-set method, both approaches are called *sharp interface* methods.

A very recent and completely different method for simulating compressible multi-phase flows is a novel type of model that uses a *diffuse interface* approach based upon extended hyperbolic systems with stiff relaxation. The basic philosophy of this new type of models is similar to the capturing of discontinuities (shockwaves) in gas dynamics. These methods were presented for the first time by Saurel et al. in [131, 135]. The diffuse interface approach is stabilized by the numerical diffusion provided by the Riemann solver at the interface, and when the mesh size tends to zero, also the interface thickness is approaching zero. Hence, the diffuse interface method is a very interesting alternative approach, and in the limit, also the diffuse interface model tends to a sharp interface representation, but based on a totally different mathematical formulation. The first applications of the diffuse interface method to compressible multi-phase flows with surface tension in two space dimensions have been presented in [121, 21], with a subsequent extension to three space dimensions carried out in [120]. Further recent research on multi-phase flows with surface tension has been presented in [104, 80]. A common problem of all Eulerian methods, i.e. for both sharp and diffuse interface approaches, is the correct calculation of the interface curvature.

Most models contain non-conservative products, so the two-phase flow of the models can be cast into the following general form of a nonlinear system of PDE in multiple space



dimensions

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}) + \mathbf{B}(\mathbf{u}) \cdot \nabla \mathbf{u} = \mathbf{S}(\mathbf{u}), \quad (1.1)$$

where  $\mathbf{u}$  is the state vector;  $\mathbf{F}(\mathbf{u}) = [\mathbf{f}(\mathbf{u}), \mathbf{g}(\mathbf{u}), \mathbf{h}(\mathbf{u})]$  is the flux tensor for the conservative part of the PDE system, with  $\mathbf{f}(\mathbf{u})$ ,  $\mathbf{g}(\mathbf{u})$  and  $\mathbf{h}(\mathbf{u})$  expressing the fluxes along the  $x$ ,  $y$  and  $z$  directions, respectively;  $\mathbf{B}(\mathbf{u}) = [\mathbf{B}_1(\mathbf{u}), \mathbf{B}_2(\mathbf{u}), \mathbf{B}_3(\mathbf{u})]$  represents the non-conservative part of the system, written in block-matrix notation. Finally,  $\mathbf{S}(\mathbf{u})$  is the source term, which may in principle be stiff. When written in quasilinear form, the system (1.1) becomes

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{A}(\mathbf{u}) \cdot \nabla \mathbf{u} = \mathbf{S}(\mathbf{u}), \quad (1.2)$$

where the matrix  $\mathbf{A}(\mathbf{u}) = [\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3] = \partial \mathbf{F}(\mathbf{u}) / \partial \mathbf{u} + \mathbf{B}(\mathbf{u})$  contains both the conservative and the non-conservative contributions.

The diffuse interface method of Saurel et al. is used for the capturing of the interface. The interface is identified by a function that represents the volume fraction of one of the two phases. The surface tension effects are included by the continuum surface force (CSF) method [20]. Since the CSF approach produces a source term that contains the gradient of the volume fraction function, we propose to treat this term as a non-conservative product rather than a classical volume source term. Recent work on numerical schemes for systems of equations involving non-conservative terms, like Eq. (1.1), includes the family of so-called path-conservative schemes of Castro and Parés et al. [119, 25, 75, 24] which are based on the theory proposed by Dal Maso, Le Floch and Murat [108] and are a generalization of the usual concept of conservative methods for systems of conservation laws. Note that the weak formulation of the Roe method by Toumi [151] can also be considered as a path-conservative scheme. It has to be clearly stressed that path-conservative schemes have known deficiencies, which have been studied in detail in [2, 26].

In this thesis, various high order path-conservative schemes are used to solve hyperbolic systems of PDEs with non-conservative products and stiff source terms (1.1) in multiple space dimensions. The first one is the second order TVD version of the path-conservative finite volume method. Within this scheme, the smooth part of the CSF source term is integrated like a classical volume source term, while the contributions due to the jumps in the volume fraction function at the element interfaces are naturally taken into account by the path-conservative finite volume scheme. The second one is the extension of high order path-conservative schemes which are used within the ADER approach together with space-time *Adaptive Mesh Refinement* (AMR) and Discontinuous Galerkin schemes. Due to the path-conservative framework, the surface tension force is naturally included

into the Riemann solver. The physical properties of the interface with curvature  $\kappa$  are solved by the approximate Riemann solver without spurious oscillations near the material interface. Furthermore, the scheme satisfies the Abgrall condition [1] and is *well-balanced* for a circular bubble at rest that obeys the Young-Laplace relation, i.e. where the surface tension force is exactly balanced by the pressure jump across the interface. This well-balancing has been shown via numerical evidence and under the assumption of an exact knowledge of the interface curvature.

Among the numerical methods specifically developed to solve hyperbolic problems, two well-known families are finite volume (FV) methods and Discontinuous Galerkin (DG) schemes. While FV methods are still extremely popular, nowadays, DG methods are becoming increasingly popular. First introduced by Reed and Hill [125] to solve a first order neutron transport equation, DG schemes are applied in different fields, in particular those related to fluid dynamics. In a series of well-known papers, Cockburn and Shu [35, 34, 33, 32, 36] provided a rigorous formal framework of these methods, contributing significantly to their popularization. DG methods are very robust and, among high order numerical methods, they show high flexibility and adaptivity [124]. Moreover, Jiang and Shu [86] proved that DG methods verify an entropy condition leading to nonlinear  $L_2$  stability. Unfortunately explicit DG methods have a strong stability limitation, since usually the CFL restriction for these schemes is very severe and the time step in  $d$  space dimensions is constrained as  $\Delta t \leq \frac{h}{d|\lambda_{max}|} \frac{1}{2N+1}$  [66], where  $h$  is a characteristic mesh size,  $\lambda_{max}$  is the maximum signal velocity and  $N$  is the degree of the basis polynomial.

The goal of this thesis is to provide a new and accurate computational method for compressible multi-phase flows with capillary effects based upon the full seven-equation Baer-Nunziato model [6]. As already mentioned before, the surface tension terms of the CSF approach are considered as a part of the non-conservative hyperbolic system and not as pure source terms. For that purpose, there are some techniques to obtain high accuracy schemes. Firstly, we use a high resolution shock capturing TVD finite volume scheme based on the path-conservative version of the new generalized Osher-type Riemann solver (DOT), the path-conservative extension of the Roe and Rusanov scheme, as well as a novel path-conservative HLLEM Riemann solver. The key idea of our approach is to obtain an approximate Riemann solver at the interface such that the surface tension effect depends on the jump of the volume fraction function across the interface. Second, we extend to a very high-order almost sharp diffuse interface method, which can be appropriately and accurately treated with high accurate numerical method such as our ADER DG with AMR, and combined with the *a posteriori* sub-cell finite volume limiter. Last but not least, a strategy of local time stepping in TVD finite volume schemes on

unstructured meshes is applied. For various applications of the Baer-Nunziato model the reader is referred to the literature, see e.g. [130, 5, 137, 39, 147, 63, 59, 60]. To our knowledge, this is the first time that a path-conservative scheme has been applied to the problem of surface tension effects in compressible multi-phase flows in the context of the full seven-equation Baer-Nunziato model.

## 1.2. The compressible Euler and Navier-Stokes equations for single phase flows

Before going to compressible multi-phase flows, here we briefly recall the compressible single phase flow equations. These equations are the fundamental conservation laws of mass, momentum and energy, which are given below.

Conservation of mass

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \quad (1.3)$$

conservation of momentum

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \nabla \cdot \mathbb{T} = \rho \mathbf{g}, \quad (1.4)$$

conservation of energy

$$\frac{\partial (\rho e + \frac{1}{2} \rho \mathbf{u}^2)}{\partial t} + \nabla \cdot \left[ \left( \rho e + \frac{1}{2} \rho \mathbf{u}^2 \right) \mathbf{u} \right] + \nabla \cdot \mathbf{q} - \nabla \cdot (\mathbb{T} \cdot \mathbf{u}) = \rho \mathbf{g} \cdot \mathbf{u} + q_s. \quad (1.5)$$

Here,  $\rho$  is the density,  $\mathbf{u}$  denotes the velocity vector,  $\mathbf{g}$  is the gravity acceleration,  $e$  represents the internal energy,  $\mathbf{q}$  is the heat flux and  $q_s$  is the body heating source. The stress tensor is given by  $\mathbb{T} = -p \mathbb{I} + \boldsymbol{\tau}$ , where  $p$  represents the fluid pressure,  $\mathbb{I}$  denotes the unit tensor and  $\boldsymbol{\tau}$  is the viscous shear stress tensor, respectively. The equations (1.3)-(1.5) are well established flow equations in the literature. They can be either derived from the basic principles of continuum mechanics, or from kinetic gas theory as the asymptotic limit of the Boltzmann equation in the small Knudsen number regime. It is important to mention that if heat flux, viscosity and gravity effects are neglected then these equations reduce to the Euler equations of compressible gas dynamics. The Euler system is a system of non-linear hyperbolic conservation laws that governs the dynamics of compressible materials, such as gases but also liquids or granular materials at high pressures under the assumption that viscous effects can be neglected. We consider the homogeneous three

dimensional Euler system as the following.

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0, \\ \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p &= 0, \\ \frac{\partial E}{\partial t} + \nabla \cdot [\mathbf{u}(E + p)] &= 0.\end{aligned}\tag{1.6}$$

Here, the density  $\rho$ , the pressure  $p$  and the velocity vector are mentioned in (1.3)-(1.5). The total energy  $E$  consists of the internal energy and kinetic energy

$$E = \rho e + \frac{1}{2} \rho \mathbf{u}^2\tag{1.7}$$

and the enthalpy  $H$  denotes the specific total enthalpy in the form,

$$H = \frac{E + P}{\rho}\tag{1.8}$$

The system (1.6) has to be closed with an equation of state that relates pressure to density and internal energy  $e = e(\rho, p)$ . For gases, the ideal gas equation of state is given by

$$p = (\gamma - 1) \left( E - \frac{1}{2} \rho \mathbf{u}^2 \right),\tag{1.9}$$

with  $\gamma$  the adiabatic exponent. For liquid fluids like water, the so-called stiffened gas equation of state can be written by

$$p = (\gamma - 1) \left( E - \frac{1}{2} \rho \mathbf{u}^2 \right) - \gamma \pi,\tag{1.10}$$

The constant  $\pi$  characterises the compressibility of the fluid.

The speed of sound  $a$  is defined by

$$a^2 = \left( \frac{\partial p}{\partial t} \right)_s\tag{1.11}$$

The speed of sound is a speed at which a disturbance produced by a sound wave travels in a medium. The disturbance is generally very small (a sound wave is not at all as strong as a shock wave), so that the process is considered as isentropic. For a perfect gas, we have the isentropic relation  $p/\gamma = \text{constant}$ , and thus,

$$a^2 = \gamma \frac{p}{\rho}\tag{1.12}$$

### 1.2.1. 1D Euler equations in conservation form

The one-dimensional Euler equations can be cast in conservative form as follows,

$$\mathbf{U}_t + \mathbf{F}_x = 0, \quad (1.13)$$

where

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u H \end{pmatrix}. \quad (1.14)$$

The flux Jacobian reads

$$\mathbb{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 & 0 \\ (\gamma - 3) \frac{u^2}{2} & (\gamma - 3) u & \gamma - 1 \\ \left(\frac{\gamma-1}{2}u^2 - H\right) u & H + (1 - \gamma) u^2 & \gamma u \end{bmatrix}. \quad (1.15)$$

The matrix  $\mathbb{A}$  is diagonalizable

$$\mathbb{A} = \mathbb{R} \Lambda \mathbb{L} \quad (1.16)$$

where  $\Lambda$ ,  $\mathbb{R}$ ,  $\mathbb{L}$  are the eigenvalues, right and left eigenvectors of the systems (1.13), respectively. They are given by

$$\Lambda = \begin{bmatrix} u - a & 0 & 0 \\ 0 & u & 0 \\ 0 & 0 & u \end{bmatrix}, \quad (1.17)$$

$$\mathbb{R} = \begin{bmatrix} 1 & 1 & 1 \\ u - a & u + a & u \\ H - ua & u^2/2 & H + ua \end{bmatrix}, \quad (1.18)$$

$$\mathbb{L} = \begin{bmatrix} \frac{1}{2} \left( \frac{\gamma-1}{2a^2} u^2 + \frac{u}{a} \right) & -\frac{1}{2} \left( \frac{\gamma-1}{a^2} u^2 + \frac{1}{a} \right) & \frac{\gamma-1}{2a^2} \\ 1 - \frac{\gamma-1}{2a^2} u^2 & \frac{\gamma-1}{a^2} u & -\frac{\gamma-1}{a^2} \\ \frac{1}{2} \left( \frac{\gamma-1}{2a^2} u^2 - \frac{u}{a} \right) & -\frac{1}{2} \left( \frac{\gamma-1}{a^2} u^2 - \frac{1}{a} \right) & \frac{\gamma-1}{2a^2} \end{bmatrix}, \quad (1.19)$$

$$\mathbb{L} d\mathbf{U} = \begin{bmatrix} \frac{dp - \rho a d u}{2a^2} \\ -\frac{dp - a^2 d \rho}{a^2} \\ \frac{dp + \rho a d u}{2a^2} \end{bmatrix}. \quad (1.20)$$

where  $a$  is the sound speed, considering the ideal equation of state  $e = p/\rho/(\gamma - 1)$ , the sound speed is given by  $a = \sqrt{\gamma p/\rho}$ . In (1.20), the quantity  $\frac{dp - a^2 d \rho}{a^2}$  corresponds to the

entropy change  $ds$  and the corresponding right eigenvector indicates how the conservative variables change due the entropy change,

$$\begin{bmatrix} \partial\rho \\ \partial(\rho u) \\ \partial(\rho E) \end{bmatrix} \propto \begin{bmatrix} 1 \\ u \\ u^2/2 \end{bmatrix}. \quad (1.21)$$

This is often called an entropy wave. The associated eigenvalue  $\mathbf{u}$  gives the speed at which the entropy wave travels: it moves with the local fluid velocity.

### 1.2.2. 1D Euler equations in physical form

The Euler equations can be written in terms of primitive or physical variables as follows,

$$\mathbf{W}_t + \mathbb{A}^w \mathbf{W}_x = 0, \quad (1.22)$$

where

$$\mathbf{W} = \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}, \quad \mathbb{A}^w = \begin{pmatrix} u & \rho & 0 \\ 0 & u & \frac{1}{\rho} \\ 0 & \rho a^2 & u \end{pmatrix}. \quad (1.23)$$

$\mathbb{A}^w$  is the coefficient matrix of the primitive form of the Euler equations.

Eigenstructure

$$\mathbb{A}^w = \mathbb{R}^w \Lambda \mathbb{L}^w \quad (1.24)$$

with

$$\Lambda = \begin{bmatrix} u - a & 0 & 0 \\ 0 & u & 0 \\ 0 & 0 & u \end{bmatrix}, \quad \mathbb{R}^w = \begin{bmatrix} -\frac{\rho}{2a} & 1 & \frac{\rho}{2a} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ -\frac{\rho a}{2} & 0 & \frac{\rho a}{2} \end{bmatrix}, \quad (1.25)$$

$$\mathbb{L}^w = \begin{bmatrix} 0 & 1 & -\frac{1}{\rho a} \\ 1 & 0 & -\frac{1}{a^2} \\ 0 & 1 & \frac{1}{\rho a} \end{bmatrix}, \quad \mathbb{L}^w d\mathbf{U} = \begin{bmatrix} d u - \frac{dp}{\rho a} \\ d\rho - \frac{dp}{a^2} \\ d u + \frac{dp}{\rho a} \end{bmatrix}. \quad (1.26)$$

The second component of (1.25, 1.26) represents the entropy wave, and we thus find

$$\begin{bmatrix} \partial\rho \\ \partial u \\ \partial p \end{bmatrix} \propto \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \quad (1.27)$$

From (1.27), the density changes due to the entropy wave; the velocity and the pressure are not effected.

Change of variables

$$d\mathbf{U} = \frac{\partial\mathbf{U}}{\partial\mathbf{W}} d\mathbf{W}, \quad d\mathbf{W} = \frac{\partial\mathbf{W}}{\partial\mathbf{U}} d\mathbf{U}, \quad (1.28)$$

where

$$\frac{\partial \mathbf{U}}{\partial \mathbf{W}} = \begin{bmatrix} 1 & 0 & 0 \\ u & \rho & 0 \\ u^2/2 & \rho u & 1/(\gamma - 1) \end{bmatrix}, \quad (1.29)$$

$$\frac{\partial \mathbf{W}}{\partial \mathbf{U}} = \left( \frac{\partial \mathbf{U}}{\partial \mathbf{W}} \right)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{u}{\rho} & \frac{1}{\rho} & 0 \\ (\gamma - 1)u^2/2 & -(\gamma - 1)u & \gamma - 1 \end{bmatrix}. \quad (1.30)$$

The coefficient matrix  $\mathbb{A}^w$  is related to the Jacobian matrix  $\mathbb{A}$  of the conservative form by

$$\mathbb{A}^w = \frac{\partial \mathbf{W}}{\partial \mathbf{U}} \mathbb{A} \frac{\partial \mathbf{U}}{\partial \mathbf{W}}. \quad (1.31)$$

Hence, the eigenvectors are given by

$$\mathbb{L} = \mathbb{L}^w \frac{\partial \mathbf{W}}{\partial \mathbf{U}}, \quad \mathbb{R} = \frac{\partial \mathbf{U}}{\partial \mathbf{W}} \mathbb{R}^w. \quad (1.32)$$

### 1.3. Baer Nunziato equations

In general, averaging is a well established mathematical tool to derive multi-phase flow models from the single-phase equations. There are several types of averaging procedures which can be classified into four categories, such as time averaging [106], volume averaging [157], time and volume averaging [43], and ensemble averaging [42]. These procedures have been used by Saurel and Abgrall in [130] to obtain a seven equation model consisting of phase-specific mass, momentum, and energy conservation equations; one phase is considered to be consisting of an ensemble of particles embedded in a carrier fluid. A transport equation for the volume fraction or color function (of one of the phases) completes the system of equations. The main difference between Baer and Nunziato's work and that of Saurel and Abgrall is related to the use of relaxation terms and the definition of the interface velocity and pressure. Saurel and Abgrall introduced the notion of relaxation parameters enabling a wider application of the seven equation model. The approach by Saurel and Abgrall [130] has been shown to apply to both interface as well as homogenous two-phase flows.

In this section, multi-phase flows will be presented in the form of the original Baer-Nunziato equations [6]. Two phase flows can be obtained by applying the single-phase flow equations on both sides of the interfaces. The balance laws for each phase are similar to those for an isolated gas, i.e., each phase can be defined by a system of compressible Euler equations plus non-conservative terms (called nozzling terms) which take into account

the interphase exchange of pressure and velocity. As mentioned in the literature, there are several difficulties related to the non-conservation property of the balance equations. Hence, the design of efficient algorithms or novel methods for solving the equations has been a very challenging problem. Since the non-conservative terms do not represent fluxes, they have to be integrated as source terms. In some studies, for simple schemes, the nozzling terms are ignored to get numerical stability and non-oscillatory behavior of numerical methods [79, 117]. However, to guarantee that the entropy inequality is satisfied [137] these terms are required. Similar to the shallow-water equations, where a numerical scheme must preserve the lake-at-rest solution, a two-phase flow with initially uniform velocity and pressure should remain as such for all times, as derived by Abgrall in [1]. In recent years some other seven-equation models have been suggested by [131] and Romenski et al [128]. However, it has also been reduced to a five equation model in [89, 113, 134, 93].

### 1.3.1. The seven-equation Baer-Nunziato model with relaxation, surface tension, viscosity and gravity effects

In this thesis we consider the full seven-equation Baer-Nunziato model for compressible two-phase flows including surface tension and gravity effects. The original model has been described in [6] and has been successively modified in [131]. The original Baer-Nunziato model has been proposed for the description of the deflagration-detonation transition in high-energy reactive materials. The full model with relaxation terms, gravity, viscosity and surface tension effects is given by the following non-conservative system:

$$\begin{aligned}
\frac{\partial}{\partial t} (\alpha_1 \rho_1) + \nabla \cdot (\alpha_1 \rho_1 \mathbf{u}_1) &= 0, \\
\frac{\partial}{\partial t} (\alpha_1 \rho_1 \mathbf{u}_1) + \nabla \cdot (\alpha_1 (\rho_1 \mathbf{u}_1 \otimes \mathbf{u}_1 + p_1 \mathbf{I} - \boldsymbol{\tau}_1)) - p_\sigma \nabla \alpha_1 &= \alpha_1 \rho_1 \mathbf{g} - \lambda (\mathbf{u}_1 - \mathbf{u}_2), \\
\frac{\partial}{\partial t} (\alpha_1 \rho_1 E_1) + \nabla \cdot [\alpha_1 ((\rho_1 E_1 + p_1) \mathbf{I} - \boldsymbol{\tau}_1) \cdot \mathbf{u}_1] - p_\sigma \mathbf{u}_I \cdot \nabla \alpha_1 &= \\
\alpha_1 \rho_1 \mathbf{g} \cdot \mathbf{u}_1 - \lambda (\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{u}_I - \beta p_I (p_1 - p_2), & \\
\frac{\partial}{\partial t} (\alpha_2 \rho_2) + \nabla \cdot (\alpha_2 \rho_2 \mathbf{u}_2) &= 0, \\
\frac{\partial}{\partial t} (\alpha_2 \rho_2 \mathbf{u}_2) + \nabla \cdot (\alpha_2 (\rho_2 \mathbf{u}_2 \otimes \mathbf{u}_2 + p_2 \mathbf{I} - \boldsymbol{\tau}_2)) - p_I \nabla \alpha_2 &= \alpha_2 \rho_2 \mathbf{g} + \lambda (\mathbf{u}_1 - \mathbf{u}_2), \\
\frac{\partial}{\partial t} (\alpha_2 \rho_2 E_2) + \nabla \cdot [\alpha_2 ((\rho_2 E_2 + p_2) \mathbf{I} - \boldsymbol{\tau}_2) \cdot \mathbf{u}_2] - p_I \mathbf{u}_I \cdot \nabla \alpha_2 &= \\
\rho_2 \mathbf{g} \cdot \mathbf{u}_2 + \lambda (\mathbf{u}_1 - \mathbf{u}_2) \cdot \mathbf{u}_I + \beta p_I (p_1 - p_2), & \\
\frac{\partial}{\partial t} \alpha_1 + \mathbf{u}_I \cdot \nabla \alpha_1 &= \beta (p_1 - p_2). \tag{1.33}
\end{aligned}$$



In the above system  $\alpha_j$  is the volume fraction of phase number  $j$ , with  $j \in \{1, 2\}$ , and the constraint  $\alpha_1 + \alpha_2 = 1$ . The constraint immediately yields the relation  $\nabla\alpha_2 = -\nabla\alpha_1$ . Furthermore,  $\rho_j$ ,  $\mathbf{u}_j$ ,  $p_j$  and  $\rho_j E_j$  represent the density, the velocity vector, the pressure and the total energy per unit mass for phase number  $j$ , respectively. The viscous stress tensor  $\boldsymbol{\tau}_j$  of phase number  $j$  that is supposed to be a Newtonian fluid under Stokes' hypothesis is given by

$$\boldsymbol{\tau}_j = \mu_j (\nabla\mathbf{u}_j + \nabla\mathbf{u}_j^T) - \frac{2}{3}\mu_j (\nabla \cdot \mathbf{u}_j) \mathbf{I}, \quad (1.34)$$

where  $T$  represents the transpose operator,  $\mu_j$  is the dynamic viscosity of fluid  $j$  and  $\mathbf{I}$  is the identity matrix. The relaxation parameter for the interphase drag is given by  $\lambda$ , the parameter governing the pressure relaxation is denoted by  $\beta$  and  $\mathbf{g}$  is the vector of gravity acceleration. Furthermore, the interface pressure of phase 1 that includes the surface tension effect is abbreviated with  $p_\sigma = p_I - \sigma\kappa$ , where  $\sigma$  is the surface tension coefficient and  $\kappa$  is the interface curvature.

The model (1.33) is closed by the stiffened gas equation of state (EOS) for each phase

$$e_j = \frac{p_j + \gamma_j \pi_j}{\rho_j (\gamma_j - 1)}, \quad (1.35)$$

and the definition of the total energy density

$$\rho_j E_j = \rho_j e_j + \frac{1}{2} \rho_j \mathbf{u}_j^2, \quad (1.36)$$

where  $\gamma_j$  is the ratio of specific heats and  $\pi_j$  is a material constant. In this work, we choose  $\mathbf{u}_\mathbf{I} = \mathbf{u}_1$  for the interface velocity and the interface pressure is assumed to be  $p_\mathbf{I} = p_2$ . This corresponds to the original choice proposed in [6]. Other options are possible, see for example in [131],  $\mathbf{u}_\mathbf{I}$  is defined as velocity of the center of mass and  $p_\mathbf{I}$  is defined by mixture pressure and they are given as follows

$$\mathbf{u}_\mathbf{I} = \frac{\alpha_1 \rho_1 \mathbf{u}_1 + \alpha_2 \rho_2 \mathbf{u}_2}{\alpha_1 \rho_1 + \alpha_2 \rho_2}, \quad p_\mathbf{I} = \alpha_1 p_1 + \alpha_2 p_2. \quad (1.37)$$

Moreover, other closure relations were derived in [132] which are given by

$$p_\mathbf{I} = \frac{Z_1 p_2 + Z_2 p_1}{Z_1 + Z_2} + \frac{Z_1 Z_2}{Z_1 + Z_2} \frac{\nabla\alpha_1}{|\nabla\alpha_1|} \cdot (\mathbf{u}_2 - \mathbf{u}_1) \quad (1.38)$$

$$\mathbf{u}_\mathbf{I} = \frac{Z_1 \mathbf{u}_2 + Z_2 \mathbf{u}_1}{Z_1 + Z_2} + \frac{Z_1 Z_2}{Z_1 + Z_2} \frac{\nabla\alpha_1}{|\nabla\alpha_1|} \cdot (p_2 - p_1) \quad (1.39)$$

Here,  $Z_k$  represents the acoustic impedance with

$$Z_k = \rho_k c_k, \quad k = 1, 2, \quad (1.40)$$

where  $c_k$  is the speed of sound defined by

$$c_k^2 = \frac{\frac{p_k}{\rho_k} - \left(\frac{\partial e_k}{\partial \rho_k}\right)}{\left(\frac{\partial e_k}{\partial \rho_k}\right)_{p_k}} \quad k = 1, 2. \quad (1.41)$$

As mention above (1.33), the interaction between the two fluids is given by the non-conservative terms as well as by the pressure and velocity relaxation terms. The pressure relaxation terms  $\beta(p_1 - p_2)$  and  $\pm\beta p_{\mathbf{I}}(p_1 - p_2)$  appear in the volume fraction equations and energy equations, respectively. Pressure relaxation and interface pressure condition were mentioned by Saurel and Abgrall in [131]. These terms are important in many aspects, for example, they control the relaxation phenomena behind the shock and pressure waves. Also, these terms are responsible for the interfacial pressure condition between the fluids and this condition expresses the equality of pressure.

The velocity relaxation terms  $\pm\lambda(\mathbf{u}_1 - \mathbf{u}_2)$  appear in the momentum equations and the terms  $\pm\lambda\mathbf{u}_{\mathbf{I}}(\mathbf{u}_1 - \mathbf{u}_2)$  in the energy equations, where  $\lambda > 0$  is the velocity relaxation parameter. This parameter gives the rate at which the velocities relax to a common value. Such terms are used to model the drag force. The drag force is because of the imbalance of the pressure and due to the viscous stresses of an elementary particle, i.e. solid particle, bubble or droplet. From experiments the relation for the drag force is defined as

$$\mathbf{F}_{drag} = \lambda(\mathbf{u}_1 - \mathbf{u}_2). \quad (1.42)$$

The parameter  $\lambda$  depends upon the geometric parameters of the particle and the Reynolds number which describe the nature of the flow. Note that in the case  $\lambda \rightarrow +\infty$  and  $\beta \rightarrow +\infty$  the system instantaneously relaxes to the mechanical equilibrium equations if the surface tension and viscosity effects are ignored. As mentioned in [133], pressure and velocity relaxation procedures are used in the seven-equation model to make it valid for several different applications and a large variety of problems like interface flows, detonation, and cavitation in multi-phase flows. According to [20], the surface force per unit volume reads

$$F_S = \sigma\kappa \frac{\nabla\Phi}{[\Phi]}, \quad (1.43)$$

where  $\sigma$  is the surface tension coefficient, which depends on the kind of fluids under consideration.  $\Phi$  is a color function, e.g. a volume fraction, and according to the discussion in [121]  $\Phi$  is taken as the liquid volume fraction in the following, i.e.  $\Phi = \alpha_1$ . The symbol  $[\Phi]$  denotes the jump of the color function across the interface and  $\kappa$  is the interface curvature. While surface tension is in reality a surface force, it can alternatively be also

approximated as an equivalent, distributed volume force. This is exactly the approach of the continuum surface force (CSF) method introduced by Brackbill et al. [20]. Some numerical results obtained with the CSF approach can be seen in [107, 114, 142, 152]. For a steady circular bubble in equilibrium, the pressure jump across the interface of two different fluids is given by the Young-Laplace equation which will be presented in section 1.4. The curvature  $\kappa$  is defined in terms of the normal vector to the phase boundary  $\mathbf{m}$  as follows:

$$\kappa = \nabla \cdot \mathbf{m} = \nabla \cdot \left( \frac{\nabla \alpha_1}{\|\nabla \alpha_1\|} \right). \quad (1.44)$$

In (1.33), the surface tension terms appear only in the momentum and the energy equation of the liquid phase (phase number 1). The interactions of the two phases are given by the relaxation source terms, which are purely algebraic, and by the non-conservative terms involving the gradient of the volume fractions (nozzling terms). These non-conservative terms directly affect the wave structure of the Baer-Nunziato system. The system (1.33) can be written in the following compact matrix-vector notation:

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{Q}, \nabla \mathbf{Q}) + \mathbf{B}(\mathbf{Q}, \kappa) \cdot \nabla \mathbf{Q} = \mathbf{S}(\mathbf{Q}), \quad (1.45)$$

with the state vector  $\mathbf{Q} = \mathbf{Q}(\mathbf{x}, t)$ , the vector of spatial coordinates  $\mathbf{x} = (x, y)$  and the nonlinear flux tensor  $\mathbf{F}(\mathbf{Q}, \nabla \mathbf{Q}) = (\mathbf{f}, \mathbf{g})$  that depends on the state vector and its gradient, to take into account also viscous effects. The purely non-conservative terms are represented by  $\mathbf{B}(\mathbf{Q}, \kappa) \cdot \nabla \mathbf{Q}$ , while the algebraic source terms like relaxation and gravity effects are contained in  $\mathbf{S}(\mathbf{Q})$ . We furthermore introduce the symbol  $\mathbf{A}(\mathbf{Q}, \kappa) = \partial \mathbf{F} / \partial \mathbf{Q} + \mathbf{B}(\mathbf{Q}, \kappa)$ , which groups the Jacobian of the flux tensor with the non-conservative terms. In the description of system (1.45), the local curvature  $\kappa$  is considered as a parameter. In the absence of viscous terms, the system (1.45) is called hyperbolic if for all unit normal vectors  $\mathbf{n}$  with  $\|\mathbf{n}\| = 1$  the matrix

$$\mathbf{A}(\mathbf{Q}, \kappa, \mathbf{n}) = \mathbf{A}(\mathbf{Q}, \kappa) \cdot \mathbf{n} = \mathbf{R}(\mathbf{Q}, \kappa, \mathbf{n}) \mathbf{\Lambda}(\mathbf{Q}, \kappa, \mathbf{n}) \mathbf{R}^{-1}(\mathbf{Q}, \kappa, \mathbf{n}) \quad (1.46)$$

is diagonalizable with  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_9)$  real and a full set of linearly independent right eigenvectors  $\mathbf{R}$ .

### 1.3.2. Eigenstructure of the Baer-Nunziato model in 1D

If we ignore the viscous effects, the gravity and relaxation terms in (1.33), the Baer-Nunziato system with surface tension reads in one space dimension as follows:

$$\begin{aligned}
\frac{\partial(\alpha_1\rho_1)}{\partial t} + \frac{\partial(\alpha_1\rho_1u_1)}{\partial x} &= 0, \\
\frac{\partial(\alpha_1\rho_1u_1)}{\partial t} + \frac{\partial(\alpha_1(\rho_1u_1u_1 + p_1))}{\partial x} - p_\sigma \frac{\partial\alpha_1}{\partial x} &= 0, \\
\frac{\partial(\alpha_1E_1)}{\partial t} + \frac{\partial(\alpha_1u_1(E_1 + p_1))}{\partial x} - p_\sigma u_I \frac{\partial\alpha_1}{\partial x} &= 0, \\
\frac{\partial(\alpha_2\rho_2)}{\partial t} + \frac{\partial(\alpha_2\rho_2u_2)}{\partial x} &= 0, \\
\frac{\partial(\alpha_2\rho_2u_2)}{\partial t} + \frac{\partial(\alpha_2(\rho_2u_2u_2 + p_2))}{\partial x} + p_I \frac{\partial\alpha_1}{\partial x} &= 0, \\
\frac{\partial(\alpha_2E_2)}{\partial t} + \frac{\partial(\alpha_2u_2(E_2 + p_2))}{\partial x} + p_I u_I \frac{\partial\alpha_1}{\partial x} &= 0, \\
\frac{\partial\alpha_1}{\partial t} + u_I \frac{\partial\alpha_1}{\partial x} &= 0.
\end{aligned} \tag{1.47}$$

We can rewrite system (1.47) in a more compact non-conservative form (1.48), given below,

$$\frac{\partial\mathbf{Q}}{\partial t} + \frac{\partial\mathbf{f}(\mathbf{Q})}{\partial x} + \mathbf{B}(\mathbf{Q}, \kappa) \frac{\partial\mathbf{Q}}{\partial x} = 0, \quad x \in \Omega \subset \mathbf{R}, \quad t \in \mathbf{R}_0^+, \tag{1.48}$$

where  $\mathbf{Q}$  is the vector of state variables,  $\mathbf{f}(\mathbf{Q})$  is the flux vector for the purely conservative part of the system and  $\mathbf{B}(\mathbf{Q})$  contains the purely non-conservative part of the system. The quantities  $\mathbf{Q}$ ,  $\mathbf{f}(\mathbf{Q})$  and  $\mathbf{B}(\mathbf{Q}, \kappa)$  are defined as,

$$\mathbf{Q} = \begin{pmatrix} \alpha_1\rho_1 \\ \alpha_1\rho_1u_1 \\ \alpha_1E_1 \\ \alpha_2\rho_2 \\ \alpha_2\rho_2u_2 \\ \alpha_2E_2 \\ \alpha_1 \end{pmatrix}, \quad \mathbf{f}(\mathbf{Q}) = \begin{pmatrix} \alpha_1\rho_1u_1 \\ \alpha_1(\rho_1u_1u_1 + p_1) \\ \alpha_1u_1(E_1 + p_1) \\ \alpha_2\rho_2u_2 \\ \alpha_2(\rho_2u_2u_2 + p_2) \\ \alpha_2u_2(E_2 + p_2) \\ 0 \end{pmatrix}, \tag{1.49}$$

$$\mathbf{B}(\mathbf{Q}, \kappa) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -p_\sigma \\ 0 & 0 & 0 & 0 & 0 & 0 & -p_\sigma u_I \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & p_I \\ 0 & 0 & 0 & 0 & 0 & 0 & u_I p_I \\ 0 & 0 & 0 & 0 & 0 & 0 & u_I \end{pmatrix}. \tag{1.50}$$

System (1.47) can also be written in the following quasilinear form

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{A}(\mathbf{Q}, \kappa) \frac{\partial \mathbf{Q}}{\partial x} = 0, \quad (1.51)$$

with the system matrix given by  $\mathbf{A}(\mathbf{Q}, \kappa) = \partial \mathbf{f}(\mathbf{Q}) / \partial \mathbf{Q} + \mathbf{B}(\mathbf{Q}, \kappa)$ , where we have dropped the dependence on the normal vector  $\mathbf{n}$  for the 1D case. After having introduced the vector of primitive variables  $\mathbf{W} = (\rho_1, u_1, p_1, \rho_2, u_2, p_2, \alpha_1)^T$ , we can rewrite the quasilinear system (1.52) in terms of the primitive variables as

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{C}(\mathbf{W}, \kappa) \frac{\partial \mathbf{W}}{\partial x} = 0, \quad (1.52)$$

with the system matrix

$$\mathbf{C}(\mathbf{W}, \kappa) = \frac{\partial \mathbf{W}}{\partial \mathbf{Q}} \mathbf{A}(\mathbf{Q}(\mathbf{W}), \kappa) \frac{\partial \mathbf{Q}}{\partial \mathbf{W}}, \quad \text{and} \quad \frac{\partial \mathbf{W}}{\partial \mathbf{Q}} = \left( \frac{\partial \mathbf{Q}}{\partial \mathbf{W}} \right)^{-1}. \quad (1.53)$$

In one space dimension the matrix  $\mathbf{C}(\mathbf{W}, \kappa)$  and the Jacobian  $\frac{\partial \mathbf{Q}}{\partial \mathbf{W}}$  read as follows:

$$\mathbf{C}(\mathbf{W}, \kappa) = \begin{pmatrix} u_1 & \rho_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & u_1 & \frac{1}{\rho_1} & 0 & 0 & 0 & \frac{p_1 - p_2 + \sigma \kappa}{\alpha_1 \rho_1} \\ 0 & \alpha_1^2 \rho_1 & u_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & u_2 & \rho_2 & 0 & \frac{\rho_2 (u_1 - u_2)}{\alpha_2} \\ 0 & 0 & 0 & 0 & u_2 & \frac{1}{\rho_2} & 0 \\ 0 & 0 & 0 & 0 & \alpha_2^2 \rho_2 & u_2 & \frac{(u_1 - u_2) \alpha_2^2 \rho_2}{\alpha_2} \\ 0 & 0 & 0 & 0 & 0 & 0 & u_1 \end{pmatrix}, \quad (1.54)$$

$$\frac{\partial \mathbf{Q}}{\partial \mathbf{W}} = \begin{pmatrix} \alpha_1 & 0 & 0 & 0 & 0 & 0 & \rho_1 \\ \alpha_1 u_1 & \alpha_1 \rho_1 & 0 & 0 & 0 & 0 & \rho_1 u_1 \\ \frac{1}{2} \alpha_1 u_1^2 & \alpha_1 \rho_1 u_1 & \frac{\alpha_1}{\gamma_1 - 1} & 0 & 0 & 0 & \rho_1 E_1 \\ 0 & 0 & 0 & \alpha_2 & 0 & 0 & -\rho_2 \\ 0 & 0 & 0 & \alpha_2 u_2 & \alpha_2 \rho_2 & 0 & -\rho_2 u_2 \\ 0 & 0 & 0 & \frac{1}{2} \alpha_2 u_2^2 & \alpha_2 \rho_2 u_2 & \frac{\alpha_2}{\gamma_2 - 1} & -\rho_2 E_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (1.55)$$

The sound speeds of the liquid and the gas phases are given by

$$a_j = \sqrt{\frac{\gamma_j (p_j + \pi_j)}{\rho_j}}. \quad (1.56)$$

The eigenvalues of the matrix  $\mathbf{C}(\mathbf{W}, \kappa)$  are then given by  $\mathbf{\Lambda} = \text{diag}(u_1 - a_1, u_1, u_1 + a_1, u_2 - a_2, u_2, u_2 + a_2, u_1)$ . It is important to notice that the eigenvalues of the system (1.47) with

and without the surface tension effect are the same. The right and left eigenvector matrix corresponding to these eigenvalues are,

$$\mathbf{R} = \begin{pmatrix} -\frac{\rho_1}{a_1} & 1 & \frac{\rho_1}{a_1} & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\rho_1 a_1 & 0 & \rho_1 a_1 & 0 & 0 & 0 & -(a_2^2 - u_1^2 + 2u_1 u_2 - u_2^2)\alpha_2 b \\ 0 & 0 & 0 & -\frac{\rho_2}{a_2} & 1 & \frac{\rho_2}{a_2} & -\rho_2(u_1^2 - 2u_1 u_2 + u_2^2)\alpha_1 \\ 0 & 0 & 0 & 1 & 0 & 1 & -(u_1 - u_2)\alpha_1 \alpha_2^2 \\ 0 & 0 & 0 & -a_2 \rho_2 & 0 & a_2 \rho_2 & -a_2^2 \alpha_1 p_2 (u_1^2 - 2u_1 u_2 + u_2^2) \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha_2 (a_2^2 - u_1^2 - 2u_1 u_2 - u_2^2)\alpha_1 \end{pmatrix},$$

and

$$\mathbf{R}^{-1} = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2\rho_1 a_1} & 0 & 0 & 0 & -\frac{p_1 - p_2 + \sigma \kappa}{2\rho_1 a_1 \alpha_1} \\ 1 & 0 & -\frac{1}{a_1^2} & 0 & 0 & 0 & -\frac{p_1 - p_2 + \sigma \kappa}{a_1^2 \alpha_1} \\ 0 & \frac{1}{2} & \frac{1}{2\rho_1 a_1} & 0 & 0 & 0 & \frac{p_1 - p_2 + \sigma \kappa}{2\rho_1 a_1 \alpha_1} \\ 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2a_2 \rho_2} & \frac{(u_1 - u_2)a_2}{2(a_2 - u_2 + u_1)\alpha_2} \\ 0 & 0 & 0 & 1 & 0 & -\frac{1}{a_2^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2a_2 \rho_2} & \frac{(u_1 - u_2)a_2}{2(a_2 + u_2 - u_1)\alpha_2} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\alpha_2 (a_2^2 - u_1^2 + 2u_1 u_2 - u_2^2)\alpha_1} \end{pmatrix}, \quad (1.57)$$

with  $b = (p_1 - p_2 + \sigma \kappa)$ . Matrix  $\mathbf{C}(\mathbf{W}, \kappa)$  can thus be written as  $\mathbf{C}(\mathbf{W}, \kappa) = \mathbf{R} \Lambda \mathbf{R}^{-1}$  and therefore one immediately also gets the eigenstructure of the matrix  $\mathbf{A}(\mathbf{Q}, \kappa)$  as

$$\mathbf{A}(\mathbf{Q}, \kappa) = \left( \frac{\partial \mathbf{Q}}{\partial \mathbf{W}} \mathbf{R} \right) \Lambda \left( \mathbf{R}^{-1} \frac{\partial \mathbf{W}}{\partial \mathbf{Q}} \right).$$

From the above eigensystem we can conclude that the model (1.47) is hyperbolic with only real eigenvalues and a full set of eigenvectors. Under surface tension effect at the interface, the wave structure of the Riemann problem [148] for the seven-equation model is illustrated in Figure 1.2. The pressure jump is discontinuous at the interface region, and obeys the Young-Laplace law [74, 13]. We denote the two physical states on the left and the right of the Riemann problem with  $\mathbf{W}_L$  and  $\mathbf{W}_R$ , respectively. For exact Riemann solvers of the Baer-Nunziato model without surface tension effects, see [5, 39, 137]. The major difficulty for numerical methods applied to system (1.47) consists in the non-conservative product, i.e. the non-conservative terms containing the gradient of the volume fraction. For that purpose, we rely in the following on the family of path-conservative schemes forwarded by Parés and Castro in [119, 25].

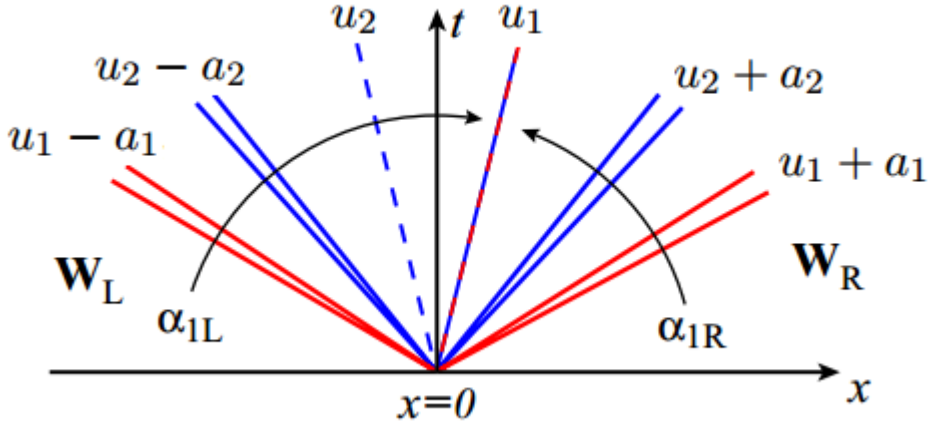


Figure 1.2.: The wave structure of an idealized Riemann problem for the seven-equation multi-phase flow model.

## 1.4. Young-Laplace Equation

In the model (1.33), the surface tension is constructed as boundary values present only at the interface, so that discontinuities or abrupt changes occur in the various variables. The phase interface can be resolved using different methods by deriving the types of the jump conditions. As discussed in [138, 106], these jump conditions actually contain the exchange of mass, momentum and energy across the interface; the properties of the phase interface are very similar the other phases in multi-phase flow systems. The jump conditions are given by mass jump

$$[\rho(\mathbf{u} - \mathbf{u}_i)] \cdot \mathbf{n}, \quad (1.58)$$

momentum jump

$$[\rho\mathbf{u}(\mathbf{u} - \mathbf{u}_i) + \mathbf{T}] \cdot \mathbf{n} = m_i^\sigma, \quad (1.59)$$

energy equation

$$\left[ \rho \left( e + \frac{1}{2} \mathbf{u}^2 \right) \mathbf{u}(\mathbf{u} - \mathbf{u}_i) + \mathbf{T} \cdot \mathbf{u} - \mathbf{q} \right] \cdot \mathbf{n} = e_i^\sigma, \quad (1.60)$$

where for any variable  $k$  at both sides of the interface, the jump is defined as  $[k] = k^+ - k^-$ . In (1.58) - (1.60),  $\mathbf{u}_i$  represents the velocity of the interface,  $\mathbf{n}$  is the outward pointing unit normal vector. The interactions at the phase interface are affected by the surface tension  $\sigma$  and energy related with the phase interface  $e_i^\sigma$ . Here we only consider the condition (1.59), in the case of non-equilibrium of a circular bubble, the pressure gradient and the capillary forces at the interface are unbalanced, giving rise to a nonzero velocity field

in the vicinity of the interface that tends to reshape an initially elliptical bubble into a circular one. Otherwise, for a steady circular bubble in equilibrium, the pressure jump across the interface of two fluids is given by Young-Laplace Equation as

$$p_1 - p_2 = \Delta p = \sigma \kappa, \quad (1.61)$$

see [74, 13]. The curvature  $\kappa$  is defined in (1.44), and  $\sigma$  is the surface tension coefficient of the fluid between the gaseous and the liquid phase. The surface tension coefficient  $\sigma$  is assumed to be a positive constant and it also is dependent on the properties of the fluid.

## 1.5. Overview of the thesis

The focus of this thesis is on the design of numerical methods for compressible multi-phase flows by using the Baer-Nunziato system with surface tension effects. The outline of this thesis is as follows. In the next chapter, we describe the second order TVD finite volume schemes on Cartesian and unstructured triangular meshes. In chapter 3 we discuss different Riemann solvers, in particular the approximate Riemann solvers of Rusanov, Roe and Osher. Also a novel path-conservative HLLEM-type Riemann solver will be presented there. A high order extension of numerical methods with ADER-DG schemes is outlined in chapter 4, while numerical results in one and two space dimensions are presented in chapter 5. Some concluding remarks and an outlook to future research are given in chapter 6.





## 2. Path-conservative Finite Volume Schemes

In this chapter we are going to discuss the numerical methods, which are applied for solving the equations of compressible multi-phase flows with surface tension. First, we describe the general framework of finite volume schemes for hyperbolic conservation laws, second we extend to path-conservative finite volume schemes on Cartesian and unstructured meshes, as well as a finite volume scheme with time-accurate local time stepping (LTS) on unstructured grids. Finally, the curvature computation is discussed.

### 2.1. Finite volume methods

Assuming the family of hyperbolic PDEs is written in the differential form of conservation laws. In Cartesian coordinates and in three space dimensions, it is written as

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} + \frac{\partial \mathbf{h}}{\partial z} = 0, \quad (2.1)$$

where  $\mathbf{u}$  is the vector of conserved variables,  $\mathbf{f}, \mathbf{g}, \mathbf{h}$  are the flux vectors in the direction of  $x, y, z$ , respectively. In order to obtain the finite volume formulation of (2.1), we discretize the computational domain  $\Omega$  in space-time control volumes defined as  $\mathbf{I}_{ijk} = \mathbb{I}_{ijk} \times [t^n, t^n + \Delta t] = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}] \times [z_{k-\frac{1}{2}}, z_{k+\frac{1}{2}}] \times [t^n, t^n + \Delta t]$ , with  $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \Delta y_j = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}, \Delta z_k = z_{k+\frac{1}{2}} - z_{k-\frac{1}{2}}$  and  $\Delta t = t^{n+1} - t^n$ . After integration of (2.1) over a space-time control volume  $\mathbf{I}_{ijk}$  one obtains the following finite volume formulation:

$$\bar{\mathbf{u}}_{ijk}^{n+1} = \bar{\mathbf{u}}_{ijk}^n - \frac{\Delta t}{\Delta x_i} [\mathbf{f}_{i+\frac{1}{2},j,k} - \mathbf{f}_{i-\frac{1}{2},j,k}] - \frac{\Delta t}{\Delta y_j} [\mathbf{g}_{i,j+\frac{1}{2},k} - \mathbf{g}_{i,j-\frac{1}{2},k}] - \frac{\Delta t}{\Delta z_k} [\mathbf{h}_{i,j,k+\frac{1}{2}} - \mathbf{h}_{i,j,k-\frac{1}{2}}], \quad (2.2)$$

where

$$\bar{\mathbf{u}}_{ijk}^n = \frac{1}{\Delta x_i} \frac{1}{\Delta y_j} \frac{1}{\Delta z_k} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}} \mathbf{u}(x, y, z, t^n) dz dy dx \quad (2.3)$$

is the spatial average of the solution in the element  $\mathbf{I}_{ijk}$  at the time  $t^n$ , while  $\mathbf{f}_{i\pm\frac{1}{2},j,k}$ ,  $\mathbf{g}_{i,j,\pm\frac{1}{2},k}$ , and  $\mathbf{h}_{i,j,k\pm\frac{1}{2}}$  are the average fluxes along each Cartesian direction and in time as follows:

$$\mathbf{f}_{i\pm\frac{1}{2},j,k} = \frac{1}{\Delta t} \frac{1}{\Delta y_j} \frac{1}{\Delta z_k} \int_{t^n}^{t^{n+1}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}} \tilde{\mathbf{f}} \left( \mathbf{u}(x_{i\pm\frac{1}{2},j,k}, y, z, t) \right) dz dy dt, \quad (2.4)$$

$$\mathbf{g}_{i,j,\pm\frac{1}{2},k} = \frac{1}{\Delta t} \frac{1}{\Delta x_i} \frac{1}{\Delta z_k} \int_{t^n}^{t^{n+1}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \int_{z_{k-\frac{1}{2}}}^{z_{k+\frac{1}{2}}} \tilde{\mathbf{g}} \left( \mathbf{u}(x, y_{i,j,\pm\frac{1}{2},k}, z, t) \right) dz dx dt, \quad (2.5)$$

$$\mathbf{h}_{i,j,k\pm\frac{1}{2}} = \frac{1}{\Delta t} \frac{1}{\Delta x_i} \frac{1}{\Delta y_j} \int_{t^n}^{t^{n+1}} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \int_{y_{k-\frac{1}{2}}}^{y_{k+\frac{1}{2}}} \tilde{\mathbf{h}} \left( \mathbf{u}(x, y, z_{i,j,k\pm\frac{1}{2}}, t) \right) dy dx dt. \quad (2.6)$$

## 2.2. Conservation laws in generalized coordinates

In general computations, geometries are not simple and, therefore, the grids associated with these geometries are also not trivial. So, to account for these non-uniformities, modifications to the above described algorithm become necessary, one considers the coordinate transformation:

$$\xi = \xi(x, y, z), \quad \eta = \eta(x, y, z), \quad \zeta = \zeta(x, y, z), \quad (2.7)$$

which defines a map between the physical coordinates  $(x, y, z)$  and the computational (or generalized) coordinates  $(\xi, \eta, \zeta)$ . It maps a curvilinear grid in the physical space to a Cartesian grid in the computational space. Computing the first order derivative of any variable  $\phi$  with regard to  $x, y$  and  $z$ , respectively is done using the chain rule according to the following formulae:

$$\frac{\partial \phi}{\partial x} = \frac{\partial \xi}{\partial x} \frac{\partial \phi}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial \phi}{\partial \eta} + \frac{\partial \zeta}{\partial x} \frac{\partial \phi}{\partial \zeta}, \quad (2.8)$$

$$\frac{\partial \phi}{\partial y} = \frac{\partial \xi}{\partial y} \frac{\partial \phi}{\partial \xi} + \frac{\partial \eta}{\partial y} \frac{\partial \phi}{\partial \eta} + \frac{\partial \zeta}{\partial y} \frac{\partial \phi}{\partial \zeta}, \quad (2.9)$$

$$\frac{\partial \phi}{\partial z} = \frac{\partial \xi}{\partial z} \frac{\partial \phi}{\partial \xi} + \frac{\partial \eta}{\partial z} \frac{\partial \phi}{\partial \eta} + \frac{\partial \zeta}{\partial z} \frac{\partial \phi}{\partial \zeta}. \quad (2.10)$$

The metric coefficients such as  $\frac{\partial \xi}{\partial x}, \frac{\partial \xi}{\partial y}, \frac{\partial \xi}{\partial z}, \frac{\partial \eta}{\partial x}, \frac{\partial \eta}{\partial y}, \frac{\partial \eta}{\partial z}$ , and  $\frac{\partial \zeta}{\partial x}, \frac{\partial \zeta}{\partial y}, \frac{\partial \zeta}{\partial z}$  are given by,

$$\frac{\partial \xi}{\partial x} = \mathbf{J}(y_\eta z_\zeta - y_\zeta z_\eta), \quad \frac{\partial \xi}{\partial y} = \mathbf{J}(x_\zeta z_\eta - x_\eta z_\zeta) \quad \frac{\partial \xi}{\partial z} = \mathbf{J}(x_\eta y_\zeta - x_\zeta y_\eta) \quad (2.11)$$

$$\frac{\partial \eta}{\partial x} = \mathbf{J} (y_\zeta z_\xi - y_\xi z_\zeta), \quad \frac{\partial \eta}{\partial y} = \mathbf{J} (x_\xi z_\zeta - x_\zeta z_\xi), \quad \frac{\partial \eta}{\partial z} = \mathbf{J} (x_\zeta y_\xi - x_\xi y_\zeta) \quad (2.12)$$

$$\frac{\partial \zeta}{\partial x} = \mathbf{J} (y_\xi z_\eta - y_\eta z_\xi), \quad \frac{\partial \zeta}{\partial y} = \mathbf{J} (x_\eta z_\xi - x_\xi z_\eta), \quad \frac{\partial \zeta}{\partial z} = \mathbf{J} (x_\xi y_\eta - x_\eta y_\xi) \quad (2.13)$$

where

$$\mathbf{J} = \frac{1}{x_\xi (y_\eta z_\zeta - y_\zeta z_\eta) - x_\eta (y_\eta z_\zeta - y_\zeta z_\eta) - x_\zeta (y_\xi z_\eta - y_\eta z_\xi)}, \quad (2.14)$$

which can be used to compute spatial derivatives in  $(x, y, z)$  from variables mapped on to  $(\xi, \eta, \zeta)$ . The set of these formulae of the derivatives  $x_\xi, x_\eta, x_\zeta, y_\xi, y_\eta, y_\zeta$  and  $z_\xi, z_\eta, z_\zeta$  are given by

$$x_\xi = \frac{x_{i+1,j,k} - x_{i-1,j,k}}{2}, \quad x_\eta = \frac{x_{i,j+1,k} - x_{i,j-1,k}}{2}, \quad x_\zeta = \frac{x_{i,j,k+1} - x_{i,j,k-1}}{2}, \quad (2.15)$$

$$y_\xi = \frac{y_{i+1,j,k} - y_{i-1,j,k}}{2}, \quad y_\eta = \frac{y_{i,j+1,k} - y_{i,j-1,k}}{2}, \quad y_\zeta = \frac{y_{i,j,k+1} - y_{i,j,k-1}}{2}, \quad (2.16)$$

$$z_\xi = \frac{z_{i+1,j,k} - z_{i-1,j,k}}{2}, \quad z_\eta = \frac{z_{i,j+1,k} - z_{i,j-1,k}}{2}, \quad z_\zeta = \frac{z_{i,j,k+1} - z_{i,j,k-1}}{2}. \quad (2.17)$$

For more details, see [3, 4]. From the above transformations, the differential form of the conservation law (2.1) will be given as

$$\frac{\partial \mathbf{u}^*}{\partial t} + \frac{\partial \mathbf{f}^*}{\partial \xi} + \frac{\partial \mathbf{g}^*}{\partial \eta} + \frac{\partial \mathbf{h}^*}{\partial \zeta} = 0, \quad (2.18)$$

where

$$\mathbf{u}^* = \frac{\mathbf{u}}{\mathbf{J}} \quad (2.19)$$

$$\mathbf{f}^* = \frac{1}{\mathbf{J}} \left( \mathbf{f} \frac{\partial \xi}{\partial x} + \mathbf{g} \frac{\partial \xi}{\partial y} + \mathbf{h} \frac{\partial \xi}{\partial z} \right) \quad (2.20)$$

$$\mathbf{g}^* = \frac{1}{\mathbf{J}} \left( \mathbf{f} \frac{\partial \eta}{\partial x} + \mathbf{g} \frac{\partial \eta}{\partial y} + \mathbf{h} \frac{\partial \eta}{\partial z} \right) \quad (2.21)$$

$$\mathbf{h}^* = \frac{1}{\mathbf{J}} \left( \mathbf{f} \frac{\partial \zeta}{\partial x} + \mathbf{g} \frac{\partial \zeta}{\partial y} + \mathbf{h} \frac{\partial \zeta}{\partial z} \right). \quad (2.22)$$

If  $\mathbf{f}, \mathbf{g}, \mathbf{h}$  depend on the derivatives of  $\mathbf{u}$ , these derivatives are transformed by Equations of (2.8) - (2.10). In cases when the geometrical grid is uniform and aligned with the coordinate axis, the metrics degenerate to an identity matrix.

## 2.3. Total-Variation-Diminishing Schemes

The most well known and one of the most successful schemes for the numerical solution of nonlinear systems of conservation laws is the method of Godunov. Here, data are represented by piecewise constant cell averages, and these piecewise constant data are also used to compute the numerical fluxes. Since the original Godunov method is only first-order accurate, it is too diffusive. Therefore, we need to develop high order schemes for hyperbolic problems. One of the main drawbacks of high order schemes are spurious oscillations at shock waves and discontinuities, the so-called Gibbs' phenomenon. According to the Godunov theorem, there are no better than first order accurate linear monotone schemes. This means, that in order to circumvent the theorem, high order schemes for conservation laws must be necessarily *nonlinear*, to overcome the drawbacks of high order schemes and achieve high order of accuracy, one of the most popular schemes is the MUSCL scheme, which is both monotonicity preserving and higher than first order accurate. This scheme was first proposed by Kolgan [90] and later independently rediscovered by van Leer [153]. This scheme makes use of piecewise linear data inside every computational cell, while first order finite volume schemes rely only on a piecewise constant data representation see Figure 2.1. In this section we will briefly introduce the scheme, which relate to a

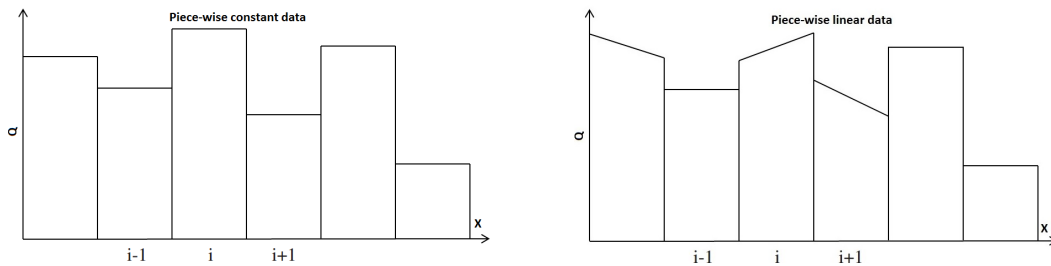


Figure 2.1.: Piecewise constant data representation in a first order finite volume scheme (left) and piecewise linear data (right) in the second order finite volume MUSCL scheme.

linear polynomial, variations and oscillations in nonlinear systems, that is the so-called Total Variation Diminishing (TVD) of property Harten et al [83], for a general overview, the reader is referred to [148, 96]. The total variation for a scalar conservation law of a discrete function  $q$ ,

$$TV(q) = \int \left| \frac{\partial q}{\partial x} \right| dx, \quad (2.23)$$

does not increase. With the definition of Harten, the total variation (TV) of a discrete function  $q_h^n$  is given as,

$$TV(q_h^n) = \sum_{i=-\infty}^{\infty} |q_{i+1}^n - q_i^n|, \quad (2.24)$$

with  $q_i^n \rightarrow \text{const.}$  for  $i \rightarrow \pm\infty$ . A numerical method

$$q_i^{n+1} = H(q_{i-l}^n, \dots, q_{i+r}^n) \quad (2.25)$$

is called TVD if

$$TV(q_h^{n+1}) \leq TV(q_h^n). \quad (2.26)$$

If a conservative scheme is TVD, then it converges to the weak solution. A piecewise linear polynomial reconstruction for cell  $i$  reads

$$P_i^n(x) = Q_i^n + \frac{\Delta Q_i^n}{\Delta x}(x - x_i), \quad (2.27)$$

where  $Q_i^n$  is the Godunov cell average and the slope  $\frac{\Delta Q_i^n}{\Delta x}$  must be properly defined in order to ensure the TVD property. This is usually achieved at the aid of so-called slope limiters.

In (2.24) we can see that oscillations in the computational results will increase the total variation. So numerical schemes for which the total variation of the solution satisfies the condition in (2.26) are called TVD schemes.

## 2.4. Path-conservation Finite volume schemes

Here, we recall the path-conservative finite volume schemes of Parés and Castro [119, 25] that we are going to use in order to discretize the non-conservative governing PDE system (1.45). We also briefly summarize the new generalized Osher-type Riemann solver (DOT) presented in [64, 65], as well as the more popular path-conservative Rusanov and Roe schemes. For all three methods, we will, however, rely on the *numerical* evaluation of the path integral along the straight line segment path by using high Gauss-Legendre quadrature rules according to [64, 65, 63]. The method is implemented inside a classical second order accurate high resolution TVD framework.

### 2.4.1. Path-conservation Finite volume schemes for the 2D Baer-Nunziato equation on the Cartesian meshes

The computational domain  $\Omega$  is discretized by a set of non-overlapping control volumes  $T_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ . In the following we assume an equidistant mesh spacing

of constant width  $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$  and  $\Delta y = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}$ , respectively. We denote the spatial barycenter of  $T_{i,j}$  with  $\mathbf{x}_{i,j} = (x_i, y_j) = \left(\frac{1}{2}(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}), \frac{1}{2}(y_{j-\frac{1}{2}} + y_{j+\frac{1}{2}})\right)$ . The time step is denoted by  $\Delta t = t^{n+1} - t^n$ . The cell averages at time  $t^n$  on  $T_{i,j}$  are defined as usual by

$$\mathbf{Q}_{i,j}^n = \frac{1}{\Delta x \Delta y} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \mathbf{Q}(x, y, t^n) dy dx, \quad (2.28)$$

from which a second order TVD scheme in space can be constructed by using the following nonlinear slope reconstruction:

$$\nabla \mathbf{Q}_{i,j}^n = \begin{pmatrix} \partial_x \mathbf{Q}_{i,j}^n \\ \partial_y \mathbf{Q}_{i,j}^n \end{pmatrix} = \begin{pmatrix} \min\text{mod} \left( \frac{\mathbf{Q}_{i+1,j}^n - \mathbf{Q}_{i,j}^n}{\Delta x}, \frac{\mathbf{Q}_{i,j}^n - \mathbf{Q}_{i-1,j}^n}{\Delta x} \right) \\ \min\text{mod} \left( \frac{\mathbf{Q}_{i,j+1}^n - \mathbf{Q}_{i,j}^n}{\Delta y}, \frac{\mathbf{Q}_{i,j}^n - \mathbf{Q}_{i,j-1}^n}{\Delta y} \right) \end{pmatrix}, \quad (2.29)$$

with the classical minmod slope limiter detailed, for example, in [148], where

$$\min\text{mod}(a, b) = \begin{cases} a & \text{if } |a| \leq |b| \text{ and } ab > 0, \\ b & \text{if } |a| > |b| \text{ and } ab > 0, \\ 0 & \text{if } ab \leq 0. \end{cases} \quad (2.30)$$

A second order MUSCL-type scheme in time can then be obtained by the following approximation of the first derivative of  $\mathbf{Q}$  in time, which is based on a discrete form of the governing PDE as follows:

$$\begin{aligned} \partial_t \mathbf{Q}_{i,j}^n &= - \frac{\mathbf{f}(\mathbf{Q}_{i,j}^n + \frac{\Delta x}{2} \partial_x \mathbf{Q}_{i,j}^n, \nabla \mathbf{Q}_{i,j}^n) - \mathbf{f}(\mathbf{Q}_{i,j}^n - \frac{\Delta x}{2} \partial_x \mathbf{Q}_{i,j}^n, \nabla \mathbf{Q}_{i,j}^n)}{\Delta x} \\ &\quad - \frac{\mathbf{g}(\mathbf{Q}_{i,j}^n + \frac{\Delta y}{2} \partial_y \mathbf{Q}_{i,j}^n, \nabla \mathbf{Q}_{i,j}^n) - \mathbf{g}(\mathbf{Q}_{i,j}^n - \frac{\Delta y}{2} \partial_y \mathbf{Q}_{i,j}^n, \nabla \mathbf{Q}_{i,j}^n)}{\Delta y} \\ &\quad - \mathbf{B}(\mathbf{Q}_{i,j}^n, \kappa_{i,j}^n) \cdot \nabla \mathbf{Q}_{i,j}^n + \mathbf{S}(\mathbf{Q}_{i,j}^n). \end{aligned} \quad (2.31)$$

The resulting piecewise linear space-time reconstruction polynomial inside cell  $T_{i,j}$  then reads

$$\mathbf{Q}(\mathbf{x}, t)|_{T_{i,j}} = \mathbf{Q}_{i,j}^n + \nabla \mathbf{Q}_{i,j}^n \cdot (\mathbf{x} - \mathbf{x}_{i,j}) + \partial_t \mathbf{Q}_{i,j}^n (t - t^n). \quad (2.32)$$

The second order TVD version of the path-conservative finite volume method is now obtained by using the piecewise linear data representation (2.32) and by integration of the governing PDE system over the space-time control volume  $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}] \times$

$[t^n, t^{n+1}]$ :

$$\begin{aligned}
\mathbf{Q}_{i,j}^{n+1} &= \mathbf{Q}_{i,j}^n - \frac{\Delta t}{\Delta x} \left( \mathbf{f}_{i+\frac{1}{2},j} - \mathbf{f}_{i-\frac{1}{2},j} \right) - \frac{\Delta t}{\Delta y} \left( \mathbf{g}_{i,j+\frac{1}{2}} - \mathbf{g}_{i,j-\frac{1}{2}} \right) \\
&\quad - \frac{\Delta t}{\Delta x} \left( \mathbf{D}_{i+\frac{1}{2},j} + \mathbf{D}_{i-\frac{1}{2},j} \right) - \frac{\Delta t}{\Delta y} \left( \mathbf{D}_{i,j+\frac{1}{2}} + \mathbf{D}_{i,j-\frac{1}{2}} \right) \\
&\quad - \Delta t \mathbf{B} \left( \mathbf{Q}_{i,j}^{n+\frac{1}{2}}, \kappa_{i,j}^n \right) \cdot \nabla \mathbf{Q}_{i,j}^n + \Delta t \mathbf{S} \left( \mathbf{Q}_{i,j}^{n+\frac{1}{2}} \right), \tag{2.33}
\end{aligned}$$

with  $\mathbf{Q}_{i,j}^{n+\frac{1}{2}} = \mathbf{Q}(\mathbf{x}_{i,j}, t^{n+\frac{1}{2}})$ . In (2.33), the  $\mathbf{f}_{i\pm\frac{1}{2},j}$  and  $\mathbf{g}_{i,j\pm\frac{1}{2}}$  denote the numerical fluxes for the purely conservative and the viscous part of the PDE, while the terms  $\mathbf{D}_{i\pm\frac{1}{2},j}$  and  $\mathbf{D}_{i,j\pm\frac{1}{2}}$  take into account the jump of the piecewise linear data reconstruction (2.32) at the element interfaces. The term  $\mathbf{B} \left( \mathbf{Q}_{i,j}^{n+\frac{1}{2}}, \kappa_{i,j}^n \right) \cdot \nabla \mathbf{Q}_{i,j}^n$  accounts for the smooth part of the non-conservative product and  $\mathbf{S} \left( \mathbf{Q}_{i,j}^{n+\frac{1}{2}} \right)$  is an explicit discretization of the algebraic source term.

The numerical flux  $\mathbf{f}_{i+\frac{1}{2},j}$  in  $x$  direction is given by

$$\mathbf{f}_{i+\frac{1}{2},j} = \frac{1}{2} \left( \mathbf{f} \left( \mathbf{Q}_{i+\frac{1}{2},j}^+, \nabla \mathbf{Q}_{i+\frac{1}{2},j}^n \right) + \mathbf{f} \left( \mathbf{Q}_{i+\frac{1}{2},j}^-, \nabla \mathbf{Q}_{i+\frac{1}{2},j}^n \right) \right) - \frac{1}{2} \Theta_{i+\frac{1}{2},j} \left( \mathbf{Q}_{i+\frac{1}{2},j}^+ - \mathbf{Q}_{i+\frac{1}{2},j}^- \right), \tag{2.34}$$

where  $\Theta_{i+\frac{1}{2},j} > 0$  is a positive definite dissipation matrix that depends on the approximate Riemann solver to be used and that will be defined later. Furthermore,  $\mathbf{Q}_{i+\frac{1}{2},j}^- = \mathbf{Q}(x_{i+\frac{1}{2}}^-, y_j, t^{n+\frac{1}{2}})$  and  $\mathbf{Q}_{i+\frac{1}{2},j}^+ = \mathbf{Q}(x_{i+\frac{1}{2}}^+, y_j, t^{n+\frac{1}{2}})$  denote the boundary-extrapolated values taken at the half time level at the element interface from the left and the right, respectively, while  $\nabla \mathbf{Q}_{i+\frac{1}{2},j}^n$  denotes the *unlimited gradient* at the edge that is used for the computation of the viscous fluxes and that is defined as the average of the *corner gradients*  $\nabla \mathbf{Q}_{i+\frac{1}{2},j\pm\frac{1}{2}}^n$  as follows:

$$\nabla \mathbf{Q}_{i+\frac{1}{2},j}^n = \frac{1}{2} \left( \nabla \mathbf{Q}_{i+\frac{1}{2},j+\frac{1}{2}}^n + \nabla \mathbf{Q}_{i+\frac{1}{2},j-\frac{1}{2}}^n \right)$$

with

$$\nabla \mathbf{Q}_{i+\frac{1}{2},j+\frac{1}{2}}^n = \frac{1}{2} \left( \frac{\mathbf{Q}_{i+1,j+1}^n - \mathbf{Q}_{i,j+1}^n}{\Delta x} + \frac{\mathbf{Q}_{i+1,j}^n - \mathbf{Q}_{i,j}^n}{\Delta y} \right). \tag{2.35}$$

The jump terms due to the non-conservative product are simply defined as

$$\begin{aligned}
\mathbf{D}_{i\pm\frac{1}{2},j} &= \frac{1}{2} \int_0^1 \mathbf{B} \left( \psi(\mathbf{Q}_{i\pm\frac{1}{2},j}^-, \mathbf{Q}_{i\pm\frac{1}{2},j}^+, s), \kappa_{i\pm\frac{1}{2},j}^n \right) \cdot \mathbf{n}_{i\pm\frac{1}{2},j} ds \\
&\approx \frac{1}{2} \sum_l \omega_l \mathbf{B} \left( \psi(\mathbf{Q}_{i\pm\frac{1}{2},j}^-, \mathbf{Q}_{i\pm\frac{1}{2},j}^+, s_l), \kappa_{i\pm\frac{1}{2},j}^n \right) \cdot \mathbf{n}_{i\pm\frac{1}{2},j}. \tag{2.36}
\end{aligned}$$



## 2.4.2. Curvature computation

The surface tension effect is naturally included inside the non-conservative terms of system (1.33). Since the term  $p_\sigma$  in system (1.33) contains the curvature  $\kappa$ , defined according to (1.44), we first need to compute the gradient of the volume fraction function at the beginning of each time step at the vertices of the two dimensional Cartesian grid, i.e. at the location  $\mathbf{x}_{i+\frac{1}{2},j+\frac{1}{2}}$ , while the cell centers are located at  $\mathbf{x}_{i,j}$ . By using this vertex-based staggering, according to Perigaud et al [121], the unlimited corner gradient of the volume fraction function is computed as follows:

$$\nabla \alpha_{1,i+\frac{1}{2},j+\frac{1}{2}}^n = \frac{1}{2} \left( \frac{\alpha_{1,i+1,j+1}^n - \alpha_{1,i,j+1}^n}{\Delta x} + \frac{\alpha_{1,i+1,j}^n - \alpha_{1,i,j}^n}{\Delta x} \right. \\ \left. \frac{\alpha_{1,i+1,j+1}^n - \alpha_{1,i+1,j}^n}{\Delta y} + \frac{\alpha_{1,i,j+1}^n - \alpha_{1,i,j}^n}{\Delta y} \right). \quad (2.37)$$

From  $\nabla \alpha_{1,i+\frac{1}{2},j+\frac{1}{2}}^n$  the normal vector to the interface is given in the vertices by the relation

$$\mathbf{m}_{i+\frac{1}{2},j+\frac{1}{2}}^n = \begin{cases} \frac{\nabla \alpha_{1,i+\frac{1}{2},j+\frac{1}{2}}^n}{\|\nabla \alpha_{1,i+\frac{1}{2},j+\frac{1}{2}}^n\|}, & \text{if } \|\nabla \alpha_{1,i+\frac{1}{2},j+\frac{1}{2}}^n\| \neq 0, \\ 0, & \text{if } \|\nabla \alpha_{1,i+\frac{1}{2},j+\frac{1}{2}}^n\| = 0. \end{cases} \quad (2.38)$$

The normal vector on the cell edges is then obtained by simple averaging

$$\mathbf{m}_{i+\frac{1}{2},j}^n = \frac{1}{2} \left( \mathbf{m}_{i+\frac{1}{2},j+\frac{1}{2}}^n + \mathbf{m}_{i+\frac{1}{2},j-\frac{1}{2}}^n \right), \quad (2.39)$$

from which finally the discrete curvature in the cell centers  $\kappa_{i,j}^n$  can be computed via

$$\kappa_{i,j}^n = \frac{\mathbf{m}_{i+\frac{1}{2},j}^n - \mathbf{m}_{i-\frac{1}{2},j}^n}{\Delta x} + \frac{\mathbf{m}_{i,j+\frac{1}{2}}^n - \mathbf{m}_{i,j-\frac{1}{2}}^n}{\Delta y}. \quad (2.40)$$

## 2.4.3. Second order MUSCL-type method on unstructured meshes with global time stepping

In this section we will extend to second-order accurate schemes to two space dimensions on unstructured meshes, which build the construction of an appropriate linear representation of the solution within a computational cell. As mentioned above, we use the technique for limiting the local solution gradients in order to avoid spurious oscillations in the solution due to the presence of local extrema resulting from the high order schemes. As usual, the global time step size is defined by  $\Delta t = t^{n+1} - t^n$ , and we use a MUSCL-Hancock strategy of van Leer [153] to produce a space-time reconstruction of the data from the known cell averages. In here the reconstructed profile is considered as a linear profile, i.e. a plane, leading to second-order spatial accuracy of the results. The purpose of the use of the

piecewise linear data reconstruction is to achieve higher order of accuracy than the first order Godunov scheme. In a finite volume framework, the data are stored and evolved at the time level  $t^n$  by the cell averaged value in the control volume  $T_i^n$ , defined as

$$\mathbf{Q}_i^n = \frac{1}{|T_i^n|} \int_{T_i^n} \mathbf{Q}(\mathbf{x}, t^n) d\mathbf{x}, \quad (2.41)$$

where  $|T_i^n|$  denoted the area of  $T_i^n$ . According to the Godunov's theory [78], we can only get a first order accurate scheme by using the piecewise constant data in (2.41) for calculating the numerical fluxes. So in order to obtain high order accurate schemes, which are better than first order ones, we need to construct piecewise space-time polynomials for each element  $T_i^n$  from the known cell averages  $\mathbf{Q}_i^n$ . In this section, second order of accuracy in space and time is given by using the well-known MUSCL-Hancock scheme [153, 148]. We first consider the reconstruction in space and the linear polynomial  $\mathbf{w}_h(\mathbf{x}, t)$  is defined as

$$\mathbf{w}_h(\mathbf{x}, t)|_{T_i^n} = \mathbf{Q}_i^n + \nabla \mathbf{Q}_i(\mathbf{x} - \mathbf{x}_i), \quad \mathbf{x} \in T_i^n, \quad (2.42)$$

where  $\mathbf{x}_i$  is the barycenter of cell  $T_i^n$ . We denote  $S_i$  a reconstruction stencil that consists of element  $T_i$  and its direct side neighbors. To compute the slope  $\nabla \mathbf{Q}_i$ , the corresponding integral conservation of the reconstruction equations read

$$\frac{1}{|T_j^n|} \int_{T_j^n} \mathbf{w}_h(\mathbf{x}, t) d\mathbf{x} = \mathbf{Q}_j^n \quad \forall T_j \in S_i. \quad (2.43)$$

The system (2.43) is in general over-determined, which can be solved with a classical constrained least-squares algorithm. The constraint is given by the requirement that (2.43) holds exactly at least for  $T_i^n$ . Finally we obtain the non-limited slope  $\nabla \mathbf{Q}_i$  in (2.42).

To obtain a stable solution with a new reconstruction process, the idea of Barth and Jespersen [12] for using the slope limiter is applied. The main idea is to find the largest admissible  $\Phi_i$  for which the values of the reconstructed variables do not exceed the maximum and the minimum values of the cell averages. The equation reads as

$$\tilde{\mathbf{w}}_h(\mathbf{x}, t^n) = \mathbf{Q}_i^n + \Phi_i \nabla \mathbf{Q}_i(\mathbf{x} - \mathbf{x}_i) \quad (2.44)$$

with the condition given as follows:

$$\mathbf{Q}_i^{min} \leq \tilde{\mathbf{w}}_h(\mathbf{x}, t^n) \leq \mathbf{Q}_i^{max} \quad (2.45)$$

Here  $\mathbf{Q}_i^{min}$  and  $\mathbf{Q}_i^{max}$  are the componentwise minimum and maximum among the cell-averages of the set  $S_i = T_i \cup N_i$ , and they are calculated as following equations.

$$\mathbf{Q}_i^{min} = \min_{j \in S_i} (\mathbf{Q}_j^n) \quad \text{and} \quad \mathbf{Q}_i^{max} = \max_{j \in S_i} (\mathbf{Q}_j^n) \quad (2.46)$$

In this case, the extreme values occur at the vertices of each element  $T_i^n$ . From the unlimited slope  $\nabla \mathbf{Q}_i$  in (2.42), one can determine a value  $\Phi_{i,j}$  for all vertices  $\mathbf{x}_j$  as follows:

$$\Phi_{i,j} = \begin{cases} \min \left( 1, \frac{\mathbf{Q}_i^{max} - \mathbf{Q}_i^n}{\mathbf{w}_{h,j} - \mathbf{Q}_i^n} \right), & \text{if } \mathbf{w}_{h,j} - \mathbf{Q}_i^n > 0 \\ \min \left( 1, \frac{\mathbf{Q}_i^{min} - \mathbf{Q}_i^n}{\mathbf{w}_{h,j} - \mathbf{Q}_i^n} \right), & \text{if } \mathbf{w}_{h,j} - \mathbf{Q}_i^n < 0 \\ 1 & \text{if } \mathbf{w}_{h,j} - \mathbf{Q}_i^n = 0. \end{cases} \quad (2.47)$$

with  $\mathbf{w}_{h,j} = \mathbf{w}_h(\mathbf{x}_j, t^n)$ . Then, from these values  $\Phi_{i,j}$ , the slope limiter is calculated as

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

From (2.44), one has only obtained a high order scheme in space, hence to achieve a high order scheme in time, it is necessary to build piecewise linear space-time polynomials  $\tilde{\mathbf{w}}_i(\mathbf{x}, t)$  inside each element  $T_i^n$ . Finally, one achieves a piecewise linear space-time reconstruction polynomial  $\mathbf{q}_h(\mathbf{x}, t)$  inside element like (2.32) with the limited slope that reads as follows:

$$\mathbf{q}_h(\mathbf{x}, t)|_{T_i^n} = \mathbf{Q}_i^n + \Phi_i \nabla \mathbf{Q}_i(\mathbf{x} - \mathbf{x}_i) + \partial_t \mathbf{Q}_i(t - t^n), \quad \mathbf{x} \in T_i(t), \quad t \in [t^n, t^{n+1}]. \quad (2.48)$$

Here  $\partial_t \mathbf{Q}_i$  is its first time derivative, which is based on a discrete form of the governing PDE as follows:

$$\partial_t \mathbf{Q}_i = -\frac{1}{|T_i|} \sum_j |\partial T_{ij}| \mathbf{F}(\mathbf{Q}_i^n + \nabla \mathbf{Q}_i^n(\mathbf{x}_{ij} - \mathbf{x}_i)) \cdot \mathbf{n}_{ij} - \frac{1}{|T_i|} \mathbf{B}(\mathbf{Q}_i^n) \cdot \Phi_i \nabla \mathbf{Q}_i^n. \quad (2.49)$$

Here,  $j$  denotes the index of edges of  $T_i^n$ ,  $\partial T_{ij}$  is the edge length,  $\mathbf{x}_{ij}$  denotes the midpoint of the common edge  $\partial T_{ij}$  shared by elements  $T_i$  and  $T_j$ ,  $\mathbf{n}_{ij}$  is the outward-pointing unit normal vector;  $\mathbf{x}_i$  is the barycenter of cell  $T_i^n$ . Then the numerical flux is given by

$$\mathbf{F}_{ij} \cdot \mathbf{n}_{ij} = \frac{1}{2} (\mathbf{F}(\mathbf{q}_h^+) + \mathbf{F}(\mathbf{q}_h^-)) \cdot \mathbf{n}_{ij} - \frac{1}{2} \Theta(\mathbf{q}_h^+ - \mathbf{q}_h^-), \quad (2.50)$$

where  $\Theta$  is again a positive definite dissipation matrix.

Finally the TVD finite volume scheme for nonconservative systems reads

$$|T_i^{n+1}| \mathbf{Q}_i^{n+1} = |T_i^n| \mathbf{Q}_i^n - |\partial T_{ij}| \sum_{j \in N_i} \left( \mathbf{F}_{ij}(\mathbf{q}_h^-, \mathbf{q}_h^+) + \mathbf{D}_{ij}(\mathbf{q}_h^-, \mathbf{q}_h^+) \right) - \mathbf{B}(\mathbf{Q}_i^{n+\frac{1}{2}}) \cdot \nabla \mathbf{Q}_i^n. \quad (2.51)$$

Here  $N_i$  is the set of neighbors of element  $T_i$ .

#### 2.4.4. Local time stepping (LTS)

As mentioned in the literature almost all the algorithms, which the time step is computed under a classical global CFL stability condition that is so-called an explicit *global* time step (GTS). The global time step  $\Delta t$  is computed as in (2.52).

$$\Delta t^n = \text{CFL} \min_{T_i^n} \frac{h_i}{\max |\Lambda(\mathbf{Q}_i)|}, \quad \forall T_i^n \in \Omega^n \quad (2.52)$$

where  $h_i$  denotes the insphere or incircle diameter of element  $T_i^n$  and  $\max |\Lambda(\mathbf{Q}_i)|$  is the maximum eigenvalue in cell  $T_i$ . CFL  $< 1/d$  is the Courant-Friedrichs-Lewy number,  $d$  represents the number of space dimensions, for the CFL condition, the reader is referred to [148].

One of the main disadvantages of an explicit time discretization for computing time step is that the global minimum is taken over all elements in the computational domain, so the GTS restriction may become very severe and small distorted elements may cause a very small time step for all elements of the domain. For these reasons the computational efficiency of the algorithm drastically decreases. So another algorithm is used to save computational time. In this section we will now describe a different method that will replace a global time-stepping strategy by using a *local* time-stepping (LTS) one. In LTS we will give up the assumption that all grid cells with the same time step. A major advantage of fully discrete TVD finite volume schemes based on piecewise polynomial reconstruction in space and time is the possibility of making use of a local time stepping scheme as has been shown in [29]. For previous works including LTS see in [44, 19, 52, 51, 54]. The LTS scheme is used to reduce the constraints of the CFL stability criterion. Therefore, within each of the cells in the computational domain is defined a local time step value, which is the maximum allowed by the CFL stability condition. The time step with an LTS strategy is given in (2.53)

$$\Delta t_i^n = \text{CFL} \min_{T_i^n} \frac{h_i}{\max |\Lambda(\mathbf{Q}_i^n)|}, \quad \forall T_i^n \in \mathbf{S}_i, \quad (2.53)$$

and the next local time level in  $T_i$  is given as

$$t_i^{n+1} = t_i^n + \Delta t_i^n, \quad (2.54)$$

where  $\mathbf{S}_i$  is the element and its direct neighbors. The  $\Delta t^n$  in (2.52) has been replaced by the minimum of the local  $\Delta t_i^n$  (2.53). In a local time stepping algorithm, we denote a new definition to recognize the updated time step that is the so-called cycle. For more details about the cycles, see [101, 54]. Following [29], the finite volume scheme for nonconservative

systems with an LTS in space and time for those elements  $T_i$  is given by

$$\begin{aligned} |T_i^{n+1}| \mathbf{Q}_i^{n+1} = & |T_i^n| \mathbf{Q}_i^n - |\partial T_{ij}| \sum_{j \in N_i} \Delta t_{ij} \left( \mathbf{F}_{ij}(\mathbf{Q}_{ij}^-, \mathbf{Q}_{ij}^+) + \mathbf{D}_{ij}(\mathbf{Q}_{ij}^-, \mathbf{Q}_{ij}^+) \right) \\ & - \Delta t_{ij} \mathbf{B} \left( \mathbf{Q}_{ij}^{n+\frac{1}{2}} \right) \cdot \nabla \mathbf{Q}_{ij}^n + \mathbf{Q}_i^M, \end{aligned} \quad (2.55)$$

where  $|T_i^n|$  and  $|T_i^{n+1}|$  are the surfaces of triangle  $T_i$  at the current time level  $t_i^n$  and the future time level  $t_i^{n+1}$ , respectively.

If we want to achieve time-accurate LTS, the following *evolve* condition or *update* criterion has to be satisfied for a element  $T_i$  at a time level  $t_i^n$

$$t_i^n + \Delta t_i^n \leq t_j^n + \Delta t_j^n \quad \text{or} \quad t_i^{n+1} \leq t_j^{n+1}, \quad \forall j \in N_i. \quad (2.56)$$

This condition constrains that a cell can be only updated if its future time is less or equal than all the future times of the Neumann neighbors. The numerical fluxes between two elements  $T_i$  and  $T_j$  have to be computed in the time interval

$$[t_{ij}^n, t_{ij}^{n+1}] = [\max(t_i^n, t_j^n), \min(t_i^{n+1}, t_j^{n+1})], \quad \Delta t_{ij} = t_{ij}^{n+1} - t_{ij}^n, \quad \Delta t_{ij}^{n+\frac{1}{2}} = \frac{1}{2} (t_{ij}^n + t_{ij}^{n+1}) \quad (2.57)$$

In order to a new high resolution TVD finite volume scheme with time-accurate LTS on unstructured grids, some parts of the flux integral will be computed using a *memory variable*  $\mathbf{Q}_i^M$ , according to [44, 19, 29]. The *memory variable* contains all fluxes through the element interfaces in the past between  $t_i^n$  to  $t_{ij}^n$ . We stress that the *memory variables* are used only for computing the flux contributions of the neighbour elements. After a local element  $T_i$  is updated according to (2.55), the memory variable of the element itself is reset to zero, and the fluxes across the element boundary are added the memory variables of the neighbor element  $\mathbf{T}_j$ . As mentioned in [29] the memory variables have to update as follows:

$$\mathbf{Q}_i^M := 0, \quad \mathbf{Q}_j^M := \mathbf{Q}_j^M + \Delta t_{ij} |\partial T_{ij}| \left( \mathbf{F}_{ij}(\mathbf{Q}_{ij}^-, \mathbf{Q}_{ij}^+) + \mathbf{D}_{ij}(\mathbf{Q}_{ij}^-, \mathbf{Q}_{ij}^+) \right), \quad \forall T_j \in N_i. \quad (2.58)$$

The boundary extrapolated values  $\mathbf{Q}_{ij}^-$  and  $\mathbf{Q}_{ij}^+$  used in (2.55) are computed from the local space-time reconstruction like a piecewise linear space-time reconstruction polynomial  $\mathbf{q}_h(\mathbf{x}, t)$  inside element (2.48) simply as

$$\mathbf{Q}_{ij}^-(\mathbf{x}, t) = \mathbf{Q}_i^n + \Phi_i \nabla \mathbf{Q}_i^n (\mathbf{x}_{ij} - \mathbf{x}_i) + \partial_t \mathbf{Q}_i^n (t_{ij}^{n+\frac{1}{2}} - t_i^n), \quad (2.59)$$

$$\mathbf{Q}_{ij}^+(\mathbf{x}, t) = \mathbf{Q}_j^n + \Phi_j \nabla \mathbf{Q}_j^n (\mathbf{x}_{ij} - \mathbf{x}_j) + \partial_t \mathbf{Q}_j^n (t_{ij}^{n+\frac{1}{2}} - t_j^n). \quad (2.60)$$

The first time derivative  $\partial_t \mathbf{Q}_i^n$  and  $\partial_t \mathbf{Q}_j^n$  in (2.59) and (2.60), respectively are computed like (2.49) as follows:

$$\partial_t \mathbf{Q}_i^n = -\frac{1}{|T_i|} \sum_j |\partial T_{ij}| \mathbf{F}(\mathbf{Q}_i^n + \nabla \mathbf{Q}_i^n (\mathbf{x}_{ij} - \mathbf{x}_i)) \cdot \mathbf{n}_{ij} - \frac{1}{|T_i|} \mathbf{B}(\mathbf{Q}_i^n) \cdot \nabla \mathbf{Q}_i^n. \quad (2.61)$$

$$\partial_t \mathbf{Q}_j^n = -\frac{1}{|T_i|} \sum_j |\partial T_{ij}| \mathbf{F}(\mathbf{Q}_j^n + \nabla \mathbf{Q}_j^n (\mathbf{x}_{ij} - \mathbf{x}_j)) \cdot \mathbf{n}_{ij} - \frac{1}{|T_i|} \mathbf{B}(\mathbf{Q}_j^n) \cdot \nabla \mathbf{Q}_j^n. \quad (2.62)$$

As well as in (2.50) then the numerical flux is given by

$$\mathbf{F}_{ij} \cdot n_{ij} = \frac{1}{2} (\mathbf{F}(\mathbf{Q}_{ij}^+) + \mathbf{F}(\mathbf{Q}_{ij}^-)) \cdot n_{ij} - \frac{1}{2} \Theta (\mathbf{Q}_{ij}^+ - \mathbf{Q}_{ij}^-), \quad (2.63)$$

where  $\Theta$  a positive definite dissipation matrix.

We can stress that the use of local time stepping instead of a global time stepping scheme does not change the fundamental properties of the TVD finite method on unstructured grids nor its accuracy as is described in more detail in [29]. LTS methods have been implemented to save the computational time for the complexity of geometry and minimize the total number of time steps for a computation with fixed end time.



## 3. Riemann Solvers

In this chapter, we briefly review the Riemann problem and approximate Riemann solvers, such as the Rusanov Riemann solver [129], the new generalized Osher-type Riemann solver (DOT) [65, 148], the Roe Riemann solver [151] and a new HLLEM Riemann solver that has been recently proposed by Dumbser and Balsara [61], as well as their applications to the multi-phase flows and non-conservative hyperbolic systems.

### 3.1. Introduction

A fundamental ingredient for designing finite volume methods and discontinuous Galerkin methods for hyperbolic conservation laws is the numerical flux, which is computed by the approximate solution of a Riemann problem. Essentially two approaches to obtain the numerical flux are the centered approach and the Godunov approach. We shall see that Riemann solvers are based on the concept of the Riemann problem; an exhaustive overview of existing Riemann solvers can be found in [148]. For computing exact Riemann solvers, the reader is referred to [78, 153]. The theory of Riemann solvers is available for both linear and nonlinear hyperbolic systems. The exact Riemann solution is often too expensive to compute for nonlinear problems. To obtain high order of accuracy and computational efficiency, approximate Riemann solvers are used as alternative to exact ones when implemented within a numerical method. These techniques of approximate Riemann solvers are available. The most powerful linear Riemann solver is the Roe solver which has the particular advantage that it recognizes shock waves and transports all characteristics nicely, it was first developed by Roe in [127] and then reformulated by Tóuimi in [151] which makes use of a weak integral formulation. In recent years, Parés and Castro have been used the theory of Dal Maso, Le Floch and Murat in [108] to extend the Roe solver to non-conservative hyperbolic systems, see [25, 119]. One of these disadvantages of the Roe solver is that it is not able to recognize sonic points, so it needs an entropy fix procedure. Some techniques for the entropy fix are presented in [148]. In the context of the family of path-conservative schemes, Dumbser and Toro have proposed the generalised Osher-type Riemann solver (DOT) [65, 148], which was first introduced by Osher and Solomon in [116]



and which has been successively applied to general nonlinear hyperbolic conservation laws and to non-conservative hyperbolic systems. The classical Osher-Solomon scheme is very complex and computationally expensive. Hence a new Osher solver is applied by using an appropriate Gaussian quadrature rule for computing path integrals that arise in the definition of the dissipative part of the Osher-Solomon flux. The DOT scheme has some general attractive advantages, such as computational robustness, entropy satisfaction, good behaviour for slowly-moving shocks and smoothness. A much simpler approach is that the local Lax-Friedrichs or the Rusanov method [129], which uses a one-wave model. Since the intermediate waves are not considered, the Rusanov Riemann solver contains the penalty of a high level of numerical dissipation. The Rusanov method is a so-called incomplete Riemann solver. In a very recent paper of Dumbser et al [61], a novel HLLEM Riemann solver has been extended to general conservative and non-conservative hyperbolic systems. This method was first proposed by Einfeldt [67] and Einfeldt et al [69] with its applications to the compressible Euler equations. It assumes no longer a constant intermediate state, but a piecewise linear distribution. For other approximate Riemann solvers, the reader is referred to [148, 96].

## 3.2. Riemann problem

A Riemann problem in the theory of hyperbolic equations is a Cauchy problem in which the piecewise constant initial state of the system is defined as follows:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial}{\partial x} f(\mathbf{q}) = 0, \quad x \in \mathcal{R}, t > 0, \quad (3.1)$$

$$\mathbf{q}(x, t = 0) = \begin{cases} \mathbf{q}_L & \text{for } x \leq 0, \\ \mathbf{q}_R & \text{for } x > 0, \end{cases} \quad (3.2)$$

where  $\mathbf{q}_L, \mathbf{q}_R \in \mathcal{Q}$ ,  $\mathcal{Q}$  is a distributional solution to the Cauchy problem. For hydrodynamic problems one can consider this to be a 1-D hydrodynamics problem or shock tube problem. Following Godunov's theory [78], the use of the Riemann problem solution as a building block for a finite volume scheme was proposed. We assume  $\mathbf{Q}_{i-1}$  and  $\mathbf{Q}_i$  are the cell averages in two neighboring grid cells on a finite volume grid, for solving the Riemann problem of equation (3.1) with the initial values as:  $\mathbf{q}_L = \mathbf{Q}_{i-1}$  and  $\mathbf{q}_R = \mathbf{Q}_i$ , see Figure 3.1. From information of the initial problem, we can compute the numerical fluxes at cell interfaces over a time step. With a large number of hydrodynamic problems, shock tube tests are used to test the performance of numerical hydrodynamics algorithms. This problem was first proposed by Sod, see in [139]. In the classical paper of Godunov, the

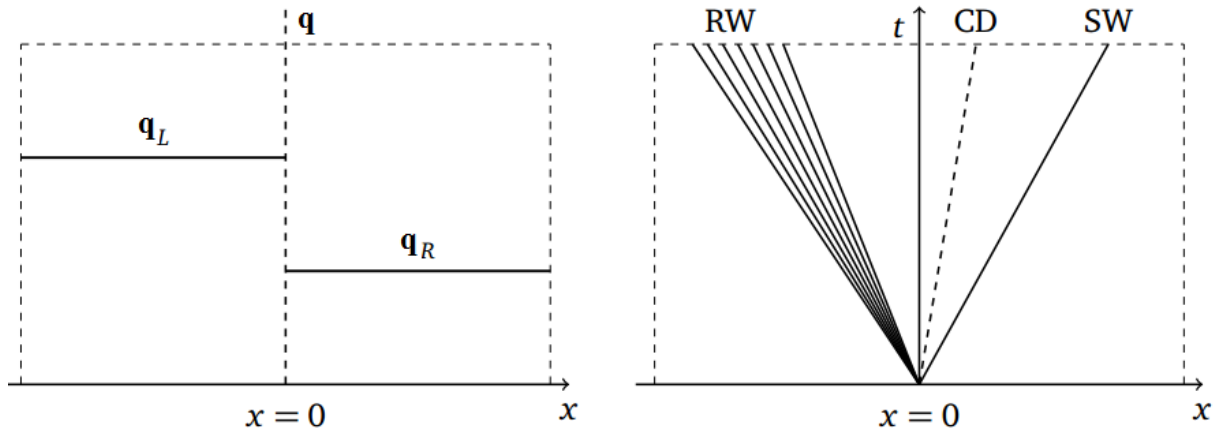


Figure 3.1.: The linear Riemann problem with initial conditions (left) and Riemann fan for the one-dimensional Euler equations(right).

use of a piecewise constant data representation was used to construct numerical fluxes, see Figure 3.2, so Godunov's scheme achieves first order of accuracy and is monotone. So to obtain high-order extensions of the Godunov method, alternative methods have been proposed, like the MUSCL scheme of van Leer [153], the piecewise parabolic method (PPM) of Woodward and Colella [37], the essentially non-oscillatory (ENO) schemes of Harten et al [82], the weighted essentially non-oscillatory schemes (WENO) of Jiang and Shu [87]. See in [148, 96], an overview of the Riemann problem in linear and nonlinear

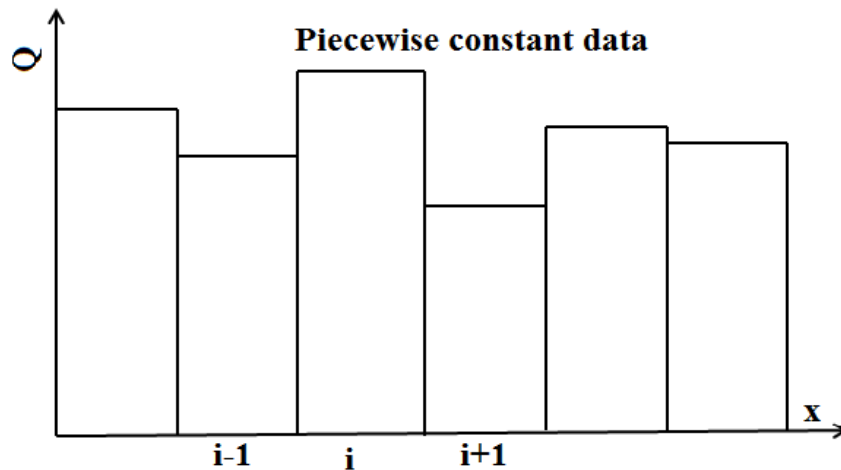


Figure 3.2.: Piecewise constant data representation in the finite volume Godunov's scheme.

hyperbolic systems. The conservation law (3.1) is called strictly hyperbolic if the Jacobian  $A(\mathbf{q}) = \frac{\partial f}{\partial \mathbf{q}}$  has  $m$  distinct real eigenvalues

$$\lambda_1(\mathbf{q}) < \lambda_2(\mathbf{q}) < \cdots < \lambda_m(\mathbf{q}) \quad \forall \mathbf{q} \in \mathcal{Q}. \quad (3.3)$$

The right eigenvectors and left eigenvectors of equation (3.1) are  $\{r_1(\mathbf{q}), \dots, r_m(\mathbf{q})\}$  and  $\{l_1(\mathbf{q}), \dots, l_m(\mathbf{q})\}$ , respectively. For the solution  $\mathbf{q} \in \mathcal{Q}$  we have the relationship of the Jacobian matrix  $A$  with eigenvalues and eigenvectors as follows:

$$A(\mathbf{q}) r_i(\mathbf{q}) = \lambda_i(\mathbf{q}) r_i(\mathbf{q}), \quad l_i(\mathbf{q})^T A(\mathbf{q}) = \lambda_i(\mathbf{q}) l_i(\mathbf{q})^T, \quad i = 1, \dots, m. \quad (3.4)$$

The right and left eigenvectors are orthogonal,

$$|r_i(\mathbf{q})| = 1, \quad l_j(\mathbf{q}) \cdot r_i(\mathbf{q}) = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \quad (3.5)$$

A  $\lambda_i$  characteristic field is said to be linearly degenerate if

$$\nabla \lambda_i(\mathbf{q}) \cdot r_i(\mathbf{q}) = 0, \quad \forall \mathbf{q} \in \mathcal{Q}, \quad (3.6)$$

or genuinely nonlinear if

$$\nabla \lambda_i(\mathbf{q}) \cdot r_i(\mathbf{q}) \neq 0, \quad \forall \mathbf{q} \in \mathcal{Q}. \quad (3.7)$$

Here  $\nabla \lambda_i(\mathbf{q})$  is the gradient of the eigenvalue  $\lambda_i(\mathbf{q})$ , say

$$\nabla \lambda_i(\mathbf{q}) = \left( \frac{\partial}{\partial \mathbf{q}_1} \lambda_i, \frac{\partial}{\partial \mathbf{q}_2} \lambda_i, \dots, \frac{\partial}{\partial \mathbf{q}_m} \lambda_i \right)^T. \quad (3.8)$$

### 3.3. Approximate Riemann solvers

In this section we will introduce the three Riemann solvers, which we use to compute the numerical flux on the Cartesian grid in (2.34) and we also extend it in a similar way to the unstructured grid in (2.50) and (2.63). A new HLLEM Riemann solver will be defined later. The dissipation matrix  $\Theta_{i+\frac{1}{2},j}$  is a function of the matrix  $\mathbf{A}(\mathbf{Q}, \kappa, \mathbf{n})$  and hence of the normal vectors w.r.t. the element interface,  $\mathbf{n}_{i\pm\frac{1}{2},j} = (1, 0)$  and  $\mathbf{n}_{i,j\pm\frac{1}{2}} = (0, 1)$ , which are pointing in the corresponding coordinate direction. In the following, we will also need the so-called straight-line *segment path* given by

$$\psi(\mathbf{Q}_L, \mathbf{Q}_R, s) = \mathbf{Q}_L + s(\mathbf{Q}_R - \mathbf{Q}_L), \quad s \in [0, 1]. \quad (3.9)$$

### 3.3.1. Rusanov scheme (LLF)

The simplest approximate Riemann solver is the one of Rusanov [129], where the entire wave structure of the Riemann problem is approximated by two symmetric waves moving to the left and to the right of the interface with the maximum wave speeds  $\pm s_{\max}$ . The resulting viscosity matrix in the numerical flux is a simple diagonal matrix of the form

$$\Theta_{i+\frac{1}{2},j}^{\text{LLF}} = s_{\max} \mathbf{I}, \quad (3.10)$$

with  $s_{\max} = \max \left( \left| \Lambda \left( \mathbf{Q}_{i+\frac{1}{2},j}^+, \kappa_{i+\frac{1}{2},j}^n, \mathbf{n}_{i+\frac{1}{2},j} \right) \right|, \left| \Lambda \left( \mathbf{Q}_{i+\frac{1}{2},j}^-, \kappa_{i+\frac{1}{2},j}^n, \mathbf{n}_{i+\frac{1}{2},j} \right) \right| \right)$ , and  $\kappa_{i+\frac{1}{2},j}^n = \frac{1}{2}(\kappa_{i,j}^n + \kappa_{i+1,j}^n)$ . Note that  $\Lambda(\mathbf{Q}, \kappa, \mathbf{n})$  is the diagonal matrix of eigenvalues of  $\mathbf{A}(\mathbf{Q}, \kappa, \mathbf{n})$  defined in (1.46). This scheme is often also called the local Lax-Friedrichs flux (LLF).

### 3.3.2. Osher-type scheme (DOT)

In the DOT Riemann solver recently proposed in [65, 64], the viscosity matrix is defined as the numerical approximation of a path integral along the straight-line segment path (3.9). It reads

$$\begin{aligned} \Theta_{i+\frac{1}{2},j}^{\text{DOT}} &= \int_0^1 \left| \mathbf{A} \left( \psi(\mathbf{Q}_{i+\frac{1}{2},j}^-, \mathbf{Q}_{i+\frac{1}{2},j}^+, s), \kappa_{i+\frac{1}{2},j}^n, \mathbf{n}_{i+\frac{1}{2},j} \right) \right| ds \\ &\approx \sum_l \omega_l \left| \mathbf{A} \left( \psi(\mathbf{Q}_{i+\frac{1}{2},j}^-, \mathbf{Q}_{i+\frac{1}{2},j}^+, s_l), \kappa_{i+\frac{1}{2},j}^n, \mathbf{n}_{i+\frac{1}{2},j} \right) \right|, \end{aligned} \quad (3.11)$$

with the usual definition of the matrix absolute value operator  $|\mathbf{A}| = \mathbf{R}|\Lambda|\mathbf{R}^{-1}$  and the Gauss-Legendre quadrature points  $s_l$  with associated weights  $\omega_l$  on the unit interval  $[0, 1]$ . In all numerical test problems shown later, we use 3 quadrature points, as suggested in [65, 64]. The DOT Riemann solver is a *complete Riemann solver*, since it makes use of the entire eigenstructure of the hyperbolic part of the governing PDE system. Since the system (1.45) is hyperbolic for physically admissible values of  $\mathbf{Q}$ , the matrix  $\mathbf{A}$  appearing in the summation is always diagonalizable and thus the operator  $|\mathbf{A}|$  is always well defined.

### 3.3.3. Roe-type scheme

A simple weak formulation of a Roe-type Riemann solver can be obtained according to [65, 64] by defining the Roe matrix again at the aid of a path integral, following the

original ideas of Toumi [151]. For the simple straight line segment path (3.9) the viscosity matrix of the resulting Roe-type Riemann solver reads:

$$\begin{aligned} \Theta_{i+\frac{1}{2},j}^{\text{Roe}} &= \left| \int_0^1 \mathbf{A} \left( \psi(\mathbf{Q}_{i+\frac{1}{2},j}^-, \mathbf{Q}_{i+\frac{1}{2},j}^+, s), \kappa_{i+\frac{1}{2},j}^n, \mathbf{n}_{i+\frac{1}{2},j} \right) ds \right| \\ &\approx \left| \sum_l \omega_l \mathbf{A} \left( \psi(\mathbf{Q}_{i+\frac{1}{2},j}^-, \mathbf{Q}_{i+\frac{1}{2},j}^+, s_l), \kappa_{i+\frac{1}{2},j}^n, \mathbf{n}_{i+\frac{1}{2},j} \right) \right|. \end{aligned} \quad (3.12)$$

In comparison with the DOT Riemann solver (3.11) we find that for the Roe-type method defined in (3.12) the matrix absolute value operator has been exchanged with the integral / summation. In practice, the eigenstructure of the matrix resulting from the summation can be easily computed numerically. Again, the path integral is approximated by Gauss-Legendre quadrature points  $s_l$  with associated weights  $\omega_l$  on the unit interval  $[0, 1]$ . Again, we use 3 quadrature points to evaluate the path integral in (3.12) numerically. Also the Roe-type Riemann solver (3.12) is a *complete Riemann solver*, since it makes use of the entire eigenstructure of the hyperbolic part of the governing PDE system. However, it can not be guaranteed in general that the matrix resulting from the summation is always diagonalizable, while the matrices appearing in the DOT solver (3.11) are always diagonalizable.

### 3.4. Jump terms in the non-conservative product

Since the entire numerical dissipation based on the matrix  $\mathbf{A} \cdot \mathbf{n}$  is already contained in the numerical fluxes (2.34), the jump terms due to the non-conservative product are simply defined as

$$\begin{aligned} \mathbf{D}_{i\pm\frac{1}{2},j} &= \frac{1}{2} \int_0^1 \mathbf{B} \left( \psi(\mathbf{Q}_{i\pm\frac{1}{2},j}^-, \mathbf{Q}_{i\pm\frac{1}{2},j}^+, s), \kappa_{i\pm\frac{1}{2},j}^n \right) \cdot \mathbf{n}_{i\pm\frac{1}{2},j} ds \\ &\approx \frac{1}{2} \sum_l \omega_l \mathbf{B} \left( \psi(\mathbf{Q}_{i\pm\frac{1}{2},j}^-, \mathbf{Q}_{i\pm\frac{1}{2},j}^+, s_l), \kappa_{i\pm\frac{1}{2},j}^n \right) \cdot \mathbf{n}_{i\pm\frac{1}{2},j}. \end{aligned} \quad (3.13)$$

These terms (3.13) give a meaning to the derivative of the piecewise continuous function  $\mathbf{Q}(\mathbf{x}, t)$  at the location of discontinuities. This requires an interpretation of  $\mathbf{B}(\mathbf{Q}, \kappa) \cdot \nabla \mathbf{Q}$  in the sense of distributions, according to the definition of weak solutions of non-conservative hyperbolic PDE proposed by Dal Maso, Le Floch and Murat in [108]. For a more detailed description of path-conservative schemes, the reader is referred to [25, 119, 64] and for open problems concerning the discretization of non-conservative hyperbolic PDE, see [26].

The numerical fluxes and jump terms in the  $y$  direction can be derived in a completely analogous way.

### 3.5. A path-conservative HLLEM scheme for non-conservative systems

In this section we extend the new HLLEM-type Riemann solver of Dumbser and Balsara [69, 68, 61], combined together with a path-conservative finite volume scheme [119, 25] to track the material interface very accurately and robustly. The HLLEM-type Riemann solvers have been adapted to hyperbolic systems with non-conservative products. We also show that the new HLLEM scheme is computationally cheaper than other well established Riemann solvers, like the Osher-type scheme proposed in [65, 64].

Our aim is to provide a new computational model for compressible multi-phase flows based upon the full Baer-Nunziato equations [6] with a highly accurate shock capturing TVD finite volume scheme that is based on the new HLLEM-type Riemann solver, and the surface tension terms must be considered as a part of the non-conservative part of the hyperbolic system. The new idea of our approach is to obtain an approximate Riemann solver at the interface such that the surface tension effect depends on the jump of the volume fraction function across the interface. For simple implementation, the interactions over the material interface are computed by using the Eulerian approach on fixed grids. We recall non-linear hyperbolic systems in (1.48) in one space dimensions without source terms as follows:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{Q})}{\partial x} + \mathbf{B}(\mathbf{Q}) \frac{\partial \mathbf{Q}}{\partial x} = 0, \quad \mathbf{x} \in \Omega \subset \mathbf{R}, \quad \mathbf{t} \in \mathbf{R}_0^+. \quad (3.14)$$

A second order path-conservative finite volume discretisation of system (3.14) based upon the path-conservative HLLEM scheme of Dumbser and Balsara [61] reads as follows:

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \frac{\Delta t}{\Delta x} \left( \mathbf{f}_{i+1/2}^- - \mathbf{f}_{i-1/2}^+ \right) - \frac{\Delta t}{\Delta x} \left( \mathbf{D}_{i+1/2}^- + \mathbf{D}_{i-1/2}^+ \right) - \frac{\Delta t}{\Delta x} \mathbf{B} \left( \mathbf{Q}_i^{n+1/2} \right) \Delta \mathbf{Q}_i^n, \quad (3.15)$$

where  $\Delta x = x_{i+1/2} - x_{i-1/2}$  and  $\Delta t = t^{n+1} - t^n$  represent the mesh spacing and the time step, respectively. The superscripts  $n$  and  $n + 1$  denote two successive time steps  $t^n$  and  $t^{n+1}$ . This method is stable under conventional CFL condition:

$$\Delta t = \text{CFL} \frac{\Delta x}{\max_{i=1} |\Lambda(\mathbf{Q}_i^n)|}, \quad \text{with} \quad \text{CFL} \leq 1. \quad (3.16)$$

In the new HLLEM Riemann solver, the jump terms and the boundary-extrapolated fluxes at the half time level  $t^{n+\frac{1}{2}}$  are written as below:

$$\mathbf{D}_{i+\frac{1}{2}}^- = \mathbf{D}_{\text{HLLEM}}^- \left( \mathbf{Q}_{i+\frac{1}{2}}^{n+\frac{1}{2},-}, \mathbf{Q}_{i+\frac{1}{2}}^{n+\frac{1}{2},+} \right), \quad \mathbf{D}_{i-\frac{1}{2}}^+ = \mathbf{D}_{\text{HLLEM}}^+ \left( \mathbf{Q}_{i-\frac{1}{2}}^{n+\frac{1}{2},-}, \mathbf{Q}_{i-\frac{1}{2}}^{n+\frac{1}{2},+} \right), \quad (3.17)$$

and

$$\mathbf{f}_{i+\frac{1}{2}}^\pm = \mathbf{f}(\mathbf{Q}_{i+\frac{1}{2}}^{n+\frac{1}{2},\pm}). \quad (3.18)$$

According to the derivation in [61], we find the jump terms  $\mathbf{D}_{\text{HLLEM}}^+$  and  $\mathbf{D}_{\text{HLLEM}}^-$  as follows:

$$\mathbf{D}_{\text{HLLEM}}^-(\mathbf{Q}_L, \mathbf{Q}_R) = \mathbf{D}_{\text{HLL}}^-(\mathbf{Q}_L, \mathbf{Q}_R) - \mathbf{T}, \quad (3.19)$$

$$\mathbf{D}_{\text{HLLEM}}^+(\mathbf{Q}_L, \mathbf{Q}_R) = \mathbf{D}_{\text{HLL}}^+(\mathbf{Q}_L, \mathbf{Q}_R) + \mathbf{T}, \quad (3.20)$$

with the HLL jump terms

$$\begin{aligned} \mathbf{D}_{\text{HLL}}^-(\mathbf{Q}_L, \mathbf{Q}_R) &= -\frac{S_L}{S_R - S_L} \left[ \mathbf{f}_R - \mathbf{f}_L + \tilde{\mathbf{B}}(\mathbf{Q}_L, \mathbf{Q}_*) (\mathbf{Q}_* - \mathbf{Q}_L) \right. \\ &\quad \left. + \tilde{\mathbf{B}}(\mathbf{Q}_*, \mathbf{Q}_R) (\mathbf{Q}_R - \mathbf{Q}_*) \right] + \frac{S_L S_R}{S_R - S_L} (\mathbf{Q}_R - \mathbf{Q}_L), \end{aligned} \quad (3.21)$$

$$\begin{aligned} \mathbf{D}_{\text{HLL}}^+(\mathbf{Q}_L, \mathbf{Q}_R) &= +\frac{S_R}{S_R - S_L} \left[ \mathbf{f}_R - \mathbf{f}_L + \tilde{\mathbf{B}}(\mathbf{Q}_L, \mathbf{Q}_*) (\mathbf{Q}_* - \mathbf{Q}_L) \right. \\ &\quad \left. + \tilde{\mathbf{B}}(\mathbf{Q}_*, \mathbf{Q}_R) (\mathbf{Q}_R - \mathbf{Q}_*) \right] - \frac{S_L S_R}{S_R - S_L} (\mathbf{Q}_R - \mathbf{Q}_L), \end{aligned} \quad (3.22)$$

and the HLL state

$$\begin{aligned} \mathbf{Q}_* &= \frac{1}{(S_R - S_L)} \left[ (\mathbf{Q}_R S_R - \mathbf{Q}_L S_L) - (\mathbf{f}_R - \mathbf{f}_L) \right. \\ &\quad \left. - \left( \tilde{\mathbf{B}}(\mathbf{Q}_L, \mathbf{Q}_*) (\mathbf{Q}_* - \mathbf{Q}_L) + \tilde{\mathbf{B}}(\mathbf{Q}_*, \mathbf{Q}_R) (\mathbf{Q}_R - \mathbf{Q}_*) \right) \right]. \end{aligned} \quad (3.23)$$

The equation (3.23) is in general non-linear in terms of  $\mathbf{Q}_*$  and needs to be iterated to convergence. For the sake of clarity, and to facilitate the reader in the practical computation of the HLL state, see [61]. Here we summarize as follows:  $\mathbf{Q}_*^0$  is an initial guess of  $\mathbf{Q}_*$ , i.e.

$$\mathbf{Q}_*^0 = \frac{1}{(S_R - S_L)} \left[ (\mathbf{Q}_R S_R - \mathbf{Q}_L S_L) - (\mathbf{f}_R - \mathbf{f}_L) - \tilde{\mathbf{B}}(\mathbf{Q}_L, \mathbf{Q}_R) (\mathbf{Q}_R - \mathbf{Q}_L) \right]. \quad (3.24)$$

The HLL state  $\mathbf{Q}_*$  can be obtained by a very simple iterative scheme by using the initial guess (3.24) as

$$\begin{aligned} \mathbf{Q}_*^{r+1} &= \frac{1}{(S_R - S_L)} \left[ (\mathbf{Q}_R S_R - \mathbf{Q}_L S_L) - (\mathbf{f}_R - \mathbf{f}_L) \right. \\ &\quad \left. - \left( \tilde{\mathbf{B}}(\mathbf{Q}_L, \mathbf{Q}_*) (\mathbf{Q}_*^r - \mathbf{Q}_L) + \tilde{\mathbf{B}}(\mathbf{Q}_*, \mathbf{Q}_R) (\mathbf{Q}_R - \mathbf{Q}_*^r) \right) \right]. \end{aligned} \quad (3.25)$$

In alternative, a quasi-Newton type iterative scheme is used to find the roots as

$$\begin{aligned} \mathbf{g}(\mathbf{Q}_*) &= \mathbf{Q}_* - \frac{1}{(S_R - S_L)} \left[ (\mathbf{Q}_R S_R - \mathbf{Q}_L S_L) - (\mathbf{f}_R - \mathbf{f}_L) \right. \\ &\quad \left. - \left( \tilde{\mathbf{B}}(\mathbf{Q}_L, \mathbf{Q}_*) (\mathbf{Q}_* - \mathbf{Q}_L) + \tilde{\mathbf{B}}(\mathbf{Q}_*, \mathbf{Q}_R) (\mathbf{Q}_R - \mathbf{Q}_*) \right) \right] = 0. \end{aligned} \quad (3.26)$$

To simplify the algorithm and keep the simple Riemann solver, we can ignore the derivatives of the Roe matrices  $\tilde{\mathbf{B}}$  with respect to  $\mathbf{Q}_*$ , so the iteration can read as follows:

$$\left( \mathbf{I} + \frac{\tilde{\mathbf{B}}(\mathbf{Q}_L, \mathbf{Q}_*) - \tilde{\mathbf{B}}(\mathbf{Q}_*, \mathbf{Q}_R)}{S_R - S_L} \right) \Delta \mathbf{Q}_*^r = \mathbf{g}(\mathbf{Q}_*^r), \quad (3.27)$$

$$\mathbf{Q}_*^{r+1} = \mathbf{Q}_*^r - \Delta \mathbf{Q}_*^r, \quad (3.28)$$

with  $\mathbf{I}$  is the identity matrix.

The expression  $\tilde{\mathbf{B}}(\mathbf{Q}_a, \mathbf{Q}_b)$  is defined by the path integral [25, 119]

$$\tilde{\mathbf{B}}(\mathbf{Q}_a, \mathbf{Q}_b) = \int_0^1 \mathbf{B}(\psi(\mathbf{Q}_a, \mathbf{Q}_b, s)) ds, \quad (3.29)$$

with the linear segment path

$$\psi(\mathbf{Q}_a, \mathbf{Q}_b, s) = \mathbf{Q}_a + s(\mathbf{Q}_b - \mathbf{Q}_a), \quad \text{with} \quad 0 \leq s \leq 1. \quad (3.30)$$

The slopes and the boundary-extrapolated data of the state vector  $\mathbf{Q}$  at time  $t^n$  in (3.15) are given as

$$\Delta \mathbf{Q}_i^n = \text{minmod}(\mathbf{Q}_{i+1}^n - \mathbf{Q}_i^n, \mathbf{Q}_i^n - \mathbf{Q}_{i-1}^n), \quad \mathbf{Q}_{i\pm\frac{1}{2}}^{n,\mp} = \mathbf{Q}_i^n \pm \frac{1}{2} \Delta \mathbf{Q}_i^n. \quad (3.31)$$

The first derivative of the state vector  $\mathbf{Q}$  in time, and the evolution to the half time level in (3.15), (3.17) and (3.18) are based on a discrete form of the governing PDE (3.14) as follows:

$$\partial_t \mathbf{Q}_i^n = - \frac{\left( \mathbf{f}(\mathbf{Q}_{i+\frac{1}{2}}^{n,-}) - \mathbf{f}(\mathbf{Q}_{i-\frac{1}{2}}^{n,+}) \right)}{\Delta x} - \mathbf{B}(\mathbf{Q}_i^n) \frac{\Delta \mathbf{Q}_i^n}{\Delta x} \quad (3.32)$$

$$\mathbf{Q}_{i\pm\frac{1}{2}}^{n+\frac{1}{2},\mp} = \mathbf{Q}_i^n \pm \frac{1}{2} \Delta \mathbf{Q}_i^n + \frac{1}{2} \Delta t \partial_t \mathbf{Q}_i^n, \quad \mathbf{Q}_i^{n+\frac{1}{2}} = \mathbf{Q}_i^n + \frac{1}{2} \Delta t \partial_t \mathbf{Q}_i^n. \quad (3.33)$$



The anti-diffusive contribution of the HLLEM method  $\mathbf{T}$  in (3.19) and (3.20), which has been used in [67, 68, 69, 61], using the mean state  $\bar{\mathbf{Q}} = \frac{1}{2}(\mathbf{Q}_R + \mathbf{Q}_L)$ , is given by

$$\mathbf{T} = \varphi \frac{S_R S_L}{S_R - S_L} \mathbf{R}_*(\bar{\mathbf{Q}}) \boldsymbol{\delta}_*(\bar{\mathbf{Q}}) \mathbf{L}_*(\bar{\mathbf{Q}}) (\mathbf{Q}_R - \mathbf{Q}_L), \quad (3.34)$$

where  $0 \leq \varphi \leq 1$  is a flattener, see [7] and  $\mathbf{R}_*$  is the matrix of right eigenvectors, associated with the linearly degenerate intermediate waves. Likewise,  $\mathbf{L}_*$  contains the left eigenvectors of the intermediate characteristic fields of the system (1.48). To control the amount of anti-diffusion, the diagonal matrix  $\boldsymbol{\delta}_*(\bar{\mathbf{Q}})$  is given by

$$\boldsymbol{\delta}_*(\bar{\mathbf{Q}}) = \mathbf{I} - \frac{\Lambda_*^-}{S_L} - \frac{\Lambda_*^+}{S_R}, \quad 0 < \boldsymbol{\delta}_*(\bar{\mathbf{Q}}) \leq 1, \quad (3.35)$$

with  $\mathbf{I}$  is the identity matrix. The computation in (3.35) is based on the diagonal elements  $\Lambda_* = \Lambda_*(\bar{\mathbf{Q}})$ . We use the following wave speed estimates for  $S_L$  and  $S_R$

$$S_L = \min(0, \Lambda(\mathbf{Q}_L), \Lambda(\bar{\mathbf{Q}})), \quad S_R = \max(0, \Lambda(\bar{\mathbf{Q}}), \Lambda(\mathbf{Q}_R)). \quad (3.36)$$

Last but not least, if we ignore viscous and non-conservative terms in the PDE in (1.48), the numerical HLLEM flux is computed as

$$\begin{aligned} \mathbf{f}_{\text{HLLEM}} &= \frac{S_R \mathbf{f}_L - S_L \mathbf{f}_R}{S_R - S_L} + \frac{S_L S_R}{S_R - S_L} (\mathbf{Q}_R - \mathbf{Q}_L) \\ &\quad - \varphi \frac{S_R S_L}{S_R - S_L} \mathbf{R}_*(\bar{\mathbf{Q}}) \boldsymbol{\delta}_*(\bar{\mathbf{Q}}) \mathbf{L}_*(\bar{\mathbf{Q}}) (\mathbf{Q}_R - \mathbf{Q}_L). \end{aligned} \quad (3.37)$$

## 4. High Order Extension

In this chapter we extend our implementation of high order path-conservative schemes for non-conservative systems of the type (1.45) using the ADER approach together with space-time *Adaptive Mesh Refinement* (AMR) and the Discontinuous Galerkin discretization framework. The seven equation Baer-Nunziato model (1.33) is written under a non-conservative form with *stiff* source terms and *surface tension effects* are modeled through the *Continuum Surface Force* (CSF) model [20] involving local curvature of the interface separating the two fluids. In our approach several ingredients are gathered to form an efficient numerical simulation tool. First an effective high order accurate in space and time ADER Discontinuous Galerkin (DG) finite element method is considered [45]. Second a high order accurate *a posteriori* sub-cell ADER-WENO finite volume limiter is employed to stabilize the former scheme in presence of steep gradients and shock waves [66]. Third space-time adaptive Cartesian meshes are considered leading to an efficient Adaptive Mesh Refinement AMR simulation code [159]. Fourth a specific high order accurate reconstruction of the curvature, which is a central part of the multiphase model, along with the DG and AMR treatments allows for an almost sharp capturing of the interface between phases even in the diffuse interface methodology employed in this work. In fact the interface is captured and allowed to travel across one single possibly refined cell. Such high accuracy allows the computation of very accurate curvature as needed in the CSF model.

### 4.1. Introduction

As mentioned in the literature, ADER schemes, originally developed by E.F. Toro and collaborators in [149] and extensively used in the context of hyperbolic systems [145, 144, 52, 8, 9], are a class of numerical methods that obtain high order of accuracy in one-step in time without the use of backward time levels, like in Adams-Bashforth type time integrators, and also without the use of substages, as used inside Runge-Kutta time integrators. The original ADER approach [149, 145, 144] suffers from the drawback that it uses Taylor expansions in time where time derivatives are substituted in place of spatial

derivatives through a repeated use of the governing system of equations. This so-called Cauchy-Kowalewski procedure is rather cumbersome when dealing with complex systems of equations, and fails in the presence of stiff source terms. Contrarily a successful alternative was proposed in [48] where the Taylor series expansions and the Cauchy-Kowalewski procedure are replaced by a local space-time Galerkin method, that is to say by a weak formulation of the PDE in space-time [45, 57, 8, 9]. In this work we employ this time integrator.

ADER-DG schemes are notoriously very accurate in smooth regions, but in the presence of sharp gradients and/or shock waves, they develop Gibbs phenomenon and give rise to spurious numerical oscillations, since they are linear in the sense of Godunov. Indeed according to Godunov's theorem [78] there are no linear and monotone schemes of order higher than the first. In the finite volume framework Godunov's theorem is circumvented by carrying out non linear reconstructions (piecewise linear with slope limiter, ENO or WENO, etc.). In the Discontinuous Galerkin approach, no spatial reconstruction is needed, but in practice it is necessary to introduce some sort of limiting or stabilization to avoid spurious oscillations. Among the most relevant limiters for DG proposed so far we can mention the use of artificial viscosity, (H)WENO limiting procedure or slope limiting as that performed in TVB methods [34]. In [66], a different and more efficient formulation has been introduced. This formulation is an *a posteriori* subcell limiting approach for DG schemes, which is based on first computing a solution by means of an *unlimited* numerical scheme and detect *a posteriori* the cells *with troubles*, using certain criteria. Once the *problematic* cells have been found, a subgrid is created within these cells and a finite volume ADER-WENO approach is used on the sub-cells. The idea of introducing an *a posteriori* approach to the problem of limiting was established in the finite volume context in [31, 40, 41, 103] by means of the paradigm denominated Multi-dimensional Optimal Order Detection (MOOD). This paradigm may be considered as the origin of the *a posteriori* limiting procedure for DG schemes.

In the present work an ADER-DG scheme with *a posteriori* sub-cell limiter has been adopted but the resolution is even more improved by introducing an Adaptive Mesh Refinement technique contrarily to the fixed grid approach developed in [66]. Adaptive Mesh Refinement (AMR) was first proposed by Marsha J. Berger and collaborators in the seminal works [16, 15, 17]. They introduced a patch-boxed block-structured AMR approach developed in finite difference. These techniques have extensively been used in different fields of research. High order accurate ADER finite volume schemes on space-time adaptive AMR grids have been proposed in [58] for conservation laws, and extended to nonconservative systems in [50].

When dealing with multi dimensional computations carried out with high order accurate DG scheme under AMR, the use of parallel computing by means of the Message Passing Interface (MPI) system is needed. The whole numerical method has been therefore parallelized to adapt to massively parallel machines.

Concerning the capillary effect in the BN model (1.33), a good resolution of material interfaces is mandatory. To achieve this goal the use of Lagrangian methods is one possibility [105, 22, 18, 23, 30, 99, 62], or ghost-fluid and level-set methods [70, 71, 115, 112, 72], little dissipative Riemann solvers combined with high order schemes [146, 150, 64] and, of course, the use of adaptive mesh refinement (AMR). In this chapter we combine high order accurate Discontinuous Galerkin schemes with AMR for the solution of compressible multi-phase flow problems to assure an accurate resolution of the material interfaces. Nevertheless an accurate description of the numerical interface does not imply an accurate, non-oscillatory computation of its associated curvature which often appears as a coefficient in front of surface tension forces like in the Continuum Surface Force model [20, 121]. Curvature may be computed by the introduction of a smooth level set or color function defined around the interface and constructed from the local quantities such as the normal to the interface [20, 121], supplemented with different strategies employing complex operators. In this thesis, we propose a novel method for the computation of curvature within the context of multi-dimensional high accurate space/time DG schemes with AMR and *a posteriori* sub-cell stabilization.

## 4.2. ADER-DG AMR scheme with a posteriori sub-cell finite volume limiter

The purpose of this section is to provide a consistent, stable and accurate numerical method devoted to solve system (1.1). Recently the authors have developed a space-time DG scheme using one-step ADER scheme in time stabilized with an *a posteriori* sub-cell finite volume limiter from [66] for homogeneous hyperbolic system of conservation laws. Alternatively, in [159], a cell-by-cell mesh adaptation technique has been supplemented and validated in the same DG framework. Let us describe in the following how these schemes and techniques are combined together to form our basic scheme, a highly accurate one step ADER DG scheme with *a posteriori* sub-cell based stabilization under the Adaptive Mesh Refinement framework. The computational domain  $\Omega$  is discretized by a

Cartesian grid composed by conforming elements  $T_i$ , namely

$$\Omega = \bigcup_{i=1}^{N_E} T_i, \quad (4.1)$$

where the index  $i$  ranges from 1 to the total number of elements  $N_E$ , which, in the adaptive mesh refinement framework, is a time-dependent quantity. In the following, we denote the cell volume by  $|T_i| = \int_{T_i} d\mathbf{x}$ . At the beginning of each time-step, the numerical solution of equation (1.1) is represented within each cell  $T_i$  by piecewise polynomials of maximum degree  $N \geq 0$  as

$$\mathbf{u}_h(\mathbf{x}, t^n) = \sum_{l=0}^M \Phi_l(\mathbf{x}) \hat{\mathbf{u}}_l^n = \Phi_l(\mathbf{x}) \hat{\mathbf{u}}_l^n \quad \mathbf{x} \in T_i, \quad (4.2)$$

where  $\mathbf{u}_h$  is referred to as the *discrete representation* of the solution, while the coefficients  $\hat{\mathbf{u}}_l^n$  are usually called the *degrees of freedom*. Note that we here use the Einstein summation convention implying summation over indices appearing twice. Basis functions  $\Phi_l(\mathbf{x})$  in (4.2) are chosen as the Lagrange interpolation polynomials of maximum degree  $N$  which pass through the  $N+1$  tensor-product Gauss-Legendre quadrature points [140, 91, 77, 92]. Following [159] we claim that the ADER-DG scheme with sub-cell limiter under AMR used in this thesis is the composition of four essential bricks, which can be schematically described as

*A predictor step* [81] in which (1.1) is solved within each cell *in the small* by means of a locally implicit space-time Discontinuous Galerkin scheme. In other words this solve does not need any interaction with any cell neighbor. The predictor is therefore a space-time DG polynomial solution of the PDE (1.1) *but neglecting the influence of the neighborhood of cell  $T_i$* . As such this step is entirely parallel, see section 4.3.1;

*A pure Discontinuous Galerkin (DG) scheme* which, by exploiting the information obtained from the predictor step, allows to compute the solution at the next time level through a single one-step corrector. This corresponds to the unlimited  $\mathbb{P}_N\mathbb{P}_N$  scheme according to [45] see section 4.3.2 for a detailed description. By *unlimited* we emphasize here the fact that no artificial mechanism is employed to stabilize the numerical scheme up to this step. The so-called candidate solution is therefore the solution obtained from a highly accurate DG scheme, which is, obviously, prone to spurious numerical oscillations due to Gibbs phenomenon.

*An a posteriori sub-cell limiter* [66] which recomputes the solution for troubled cells needing some dissipation (limiting), by an ADER-WENO finite volume scheme acting at the sub-cell level. Note that this step implies that troubled cells must be

detected amongst the cells of the candidate solution obtained at the end of the time step after the DG scheme has operated. Back to  $t^n$ : DG polynomials in the troubled cells are scattered onto sub-cells and evolved with a finite volume ADER-WENO scheme, see section 4.4; in other words, this step can also be called an element-local checkpointing and restarting of the code, but using for the restart a more robust scheme on a fine subgrid.

An *Adaptive Mesh Refinement* (AMR) approach [159], which employs a classical cell-by-cell strategy with the combination of specific treatments for sub-cell data involved when the ADER-WENO scheme is triggered on troubled cells, see section 4.5.

In the following we provide a minimal description of the entire scheme but we urge the reader to consult exhaustive descriptions and validations of each step in the following references [45, 57, 84, 76, 10, 66, 159].

## 4.3. ADER-DG scheme

### 4.3.1. The local space-time predictor

The local space-time predictor step in our approach is inherited from the essence and origin of the ADER approach [136, 144] or its more recent version adopted in this thesis [48, 45]. In a DG framework it consists in a time evolution of the representation polynomials at time  $t^n$   $\mathbf{u}_h(\mathbf{x}, t^n)$  of Eq. (4.2) independently of any neighbor effect. Each cell is considered isolated from the rest of the mesh, and, as such, during this local evolution, it does not receive any flux-like contribution from its neighbors. The so-called predictor,  $\mathbf{q}_h(\mathbf{x}, t)$ , is then be defined for each point  $\mathbf{x}$  in the current space-time cell  $T_i \times [t^n, t^{n+1}]$  and is a local solution of the system of PDEs considered. System of Eq. (1.1) is transformed into a space-time reference coordinate system  $(\boldsymbol{\xi}, \tau)$  of the space-time reference element  $[0; 1]^d \times [0; 1]$ . Spatial reference element is denoted  $T_E = [0; 1]^d$ , whereas the time is transformed according to  $t = t^n + \Delta t \tau$ . As a result, we obtain

$$\frac{\partial \mathbf{u}}{\partial \tau} + \nabla_{\boldsymbol{\xi}} \cdot \mathbf{F}^*(\mathbf{u}) + \mathbf{B}^*(\mathbf{u}) \cdot \nabla_{\boldsymbol{\xi}} \mathbf{u} = \mathbf{S}^*(\mathbf{u}), \quad (4.3)$$

where

$$\begin{aligned} \mathbf{F}^*(\mathbf{u}) &:= \Delta t (\partial \boldsymbol{\xi} / \partial \mathbf{x})^T \cdot \mathbf{F}(\mathbf{u}), \\ \mathbf{B}^*(\mathbf{u}) \cdot \nabla_{\boldsymbol{\xi}} \mathbf{u} &:= \Delta t \mathbf{B}(\mathbf{u}) \cdot (\partial \boldsymbol{\xi} / \partial \mathbf{x})^T \nabla \mathbf{u}, \quad \mathbf{S}^*(\mathbf{u}) := \Delta t \mathbf{S}(\mathbf{u}). \end{aligned} \quad (4.4)$$

and where  $\nabla_\xi = \partial \boldsymbol{\xi} / \partial \mathbf{x} \cdot \nabla$ . Next the multiplication of (4.3) with a space-time test function  $\theta_k = \theta_k(\boldsymbol{\xi}, \tau)$  and subsequent integration over the space-time reference control volume  $T_E \times [0; 1]$  yields

$$\begin{aligned} & \int_0^1 \int_{T_E} \theta_k \frac{\partial \mathbf{u}}{\partial \tau} d\boldsymbol{\xi} d\tau + \int_0^1 \int_{T_E} \theta_k \nabla_\xi \cdot \mathbf{F}_h^*(\mathbf{u}) d\boldsymbol{\xi} d\tau \\ & + \int_0^1 \int_{T_E} \theta_k \mathbf{B}_h^*(\mathbf{u}) \cdot \nabla_\xi \mathbf{u} d\boldsymbol{\xi} d\tau = \int_0^1 \int_{T_E} \theta_k \mathbf{S}_h^*(\mathbf{u}) d\boldsymbol{\xi} d\tau . \end{aligned} \quad (4.5)$$

Mimicking (4.2), we introduce the discrete space-time solution of equation (4.5), denoted by  $\mathbf{q}_h$ , as well as the corresponding ones for the flux, non-conservative part and the source term i.e.

$$\begin{aligned} \mathbf{q}_h &= \mathbf{q}_h(\boldsymbol{\xi}, \tau) = \theta_l \hat{\mathbf{q}}_l, & \mathbf{F}_h^* &= \mathbf{F}_h^*(\boldsymbol{\xi}, \tau) = \theta_l \hat{\mathbf{F}}_l^*, \\ \mathbf{B}_h^* &= \mathbf{B}_h^*(\boldsymbol{\xi}, \tau) = \theta_l \hat{\mathbf{B}}_l^*, & \mathbf{S}_h^* &= \mathbf{S}_h^*(\boldsymbol{\xi}, \tau) = \theta_l \hat{\mathbf{S}}_l^*, \end{aligned} \quad (4.6)$$

moreover

$$\nabla \mathbf{q}_h = \nabla \mathbf{q}_h(\boldsymbol{\xi}, \tau) = \nabla \theta_l \hat{\mathbf{q}}_l. \quad (4.7)$$

A very convenient choice for our nodal basis associates the degrees of freedom for the fluxes and the point-wise evaluation of the physical fluxes, that is

$$\hat{\mathbf{F}}_l^* = \mathbf{F}^*(\hat{\mathbf{q}}_l), \quad \hat{\mathbf{B}}_l^* = \mathbf{B}^*(\hat{\mathbf{q}}_l), \quad \hat{\mathbf{S}}_l^* = \mathbf{S}^*(\hat{\mathbf{q}}_l). \quad (4.8)$$

As already said this predictor step remarkably neglects the effect of neighbor cells, and, as such an integration by part of (4.5) is relatively simple (notice the integration by part of the first term taking into account  $\mathbf{q}_h(\boldsymbol{\xi}, \tau = 0) = \mathbf{u}_h(\boldsymbol{\xi})$ ):

$$\begin{aligned} & \left( \int_{T_E} \theta_k(\boldsymbol{\xi}, 1) \mathbf{q}_h d\boldsymbol{\xi} - \int_{T_E} \theta_k(\boldsymbol{\xi}, 0) \mathbf{u}_h d\boldsymbol{\xi} - \int_0^1 \int_{T_E} \frac{\partial \theta_k}{\partial \tau} \mathbf{q}_h d\boldsymbol{\xi} d\tau \right) + \int_0^1 \int_{T_E} \theta_k \nabla_\xi \cdot \mathbf{F}_h^* d\boldsymbol{\xi} d\tau \\ & + \int_0^1 \int_{T_E} \theta_k \mathbf{B}_h^* \cdot \nabla_\xi \mathbf{q}_h d\boldsymbol{\xi} d\tau = \int_0^1 \int_{T_E} \theta_k \mathbf{S}_h^* d\boldsymbol{\xi} d\tau. \end{aligned} \quad (4.9)$$

Substituting (4.6) and (4.7) into (4.9) we obtain the equation to be solved to get access to a valid predictor [45, 84, 57]

$$\begin{aligned}
& \left( \int_{T_E} \theta_k(\boldsymbol{\xi}, 1) \theta_l(\boldsymbol{\xi}, 1) d\boldsymbol{\xi} - \int_0^1 \int_{T_E} \frac{\partial \theta_k}{\partial \tau} \theta_l d\boldsymbol{\xi} d\tau \right) \hat{\mathbf{q}}_l = \\
& \left( \int_{T_E} \theta_k(\boldsymbol{\xi}, 0) \Phi_l d\boldsymbol{\xi} \right) \hat{\mathbf{u}}_l^n - \left( \int_0^1 \int_{T_E} \theta_k \nabla_\xi \theta_l d\boldsymbol{\xi} d\tau \right) \mathbf{F}^*(\hat{\mathbf{q}}_l) \\
& + \left( \int_0^1 \int_{T_E} \theta_k \theta_l \nabla \theta_j d\boldsymbol{\xi} d\tau \right) \hat{\mathbf{B}}^*(\hat{\mathbf{q}}_l) \hat{\mathbf{q}}_j + \left( \int_0^1 \int_{T_E} \theta_k \theta_l d\boldsymbol{\xi} d\tau \right) \hat{\mathbf{S}}^*(\hat{\mathbf{q}}_l) \quad (4.10)
\end{aligned}$$

Equations (4.10) corresponds to a nonlinear system to be solved in the unknown expansion coefficients  $\hat{\mathbf{q}}_l$  of the local space-time predictor solution. The terms  $\hat{\mathbf{u}}_l^n$  are known degrees of freedom from the DG polynomial at  $t^n$  [45, 84].

### 4.3.2. Fully discrete one-step ADER-DG scheme

The spacetime predictor  $\mathbf{q}_h$  obtained by means of (4.10) is not a solution of the PDEs at time level  $t^{n+1}$ . Therefore a fully discrete one-step ADER DG step [55, 143, 122], will now couple the current cell and its neighbors. Following the classical DG framework we multiply the governing PDE (1.1) by a test function  $\Phi_k \in \mathcal{U}_h$ , identical to the spatial basis functions for convenience. Then we integrate over the space-time control volume  $T_i \times [t^n; t^{n+1}]$ , noticing that an integration by part in space is applied to the flux divergence term in order to separate flux terms involving neighbor interaction from terms local to the current cell. This yields

$$\begin{aligned}
& \int_{t^n}^{t^{n+1}} \int_{T_i} \Phi_k \frac{\partial \mathbf{u}_h}{\partial t} d\mathbf{x} dt + \int_{t^n}^{t^{n+1}} \int_{\partial T_i} \Phi_k \mathbf{F}(\mathbf{u}_h) \cdot \mathbf{n} dS dt - \int_{t^n}^{t^{n+1}} \int_{T_i} \nabla \Phi_k \cdot \mathbf{F}(\mathbf{u}_h) d\mathbf{x} dt \\
& + \int_{t^n}^{t^{n+1}} \int_{T_i} \Phi_k \mathbf{B}(\mathbf{u}_h) \cdot \nabla \mathbf{u}_h d\mathbf{x} dt = \int_{t^n}^{t^{n+1}} \int_{T_i} \Phi_k \mathbf{S}(\mathbf{u}_h) d\mathbf{x} dt, \quad (4.11)
\end{aligned}$$

where  $\mathbf{n}$  is the outward pointing unit normal vector on the boundary  $\partial T_i$  of element  $T_i$ . The second term of (4.11) is computed through the solution of a Riemann problem which guarantees the upwind character of the method [35, 34, 33, 32, 36]. In this work we have only considered the simple and robust Rusanov (local Lax Friedrichs) flux [129], denoted as  $\mathcal{G}$ . Note that it is not a limitation of the method and other more advanced numerical



fluxes could be considered.

It is utterly important that the time integration of the terms of (4.11) are performed to the desired order of accuracy. This is the reason for the existence of the local space-time predictor  $\mathbf{q}_h$  allowing to compute the numerical flux function as  $\mathcal{G}(\mathbf{q}_h^-, \mathbf{q}_h^+)$  and the physical flux of the third term as  $\mathbf{F}(\mathbf{q}_h)$ . We emphasize that  $\mathbf{q}_h^-$  and  $\mathbf{q}_h^+$  are the left and right states of the Riemann problem.

Moreover after substitution of  $\mathbf{u}_h$  (4.2) in the first term of (4.11) we obtain the following arbitrary high order accurate one-step Discontinuous Galerkin (ADER-DG) scheme as

$$\begin{aligned}
& \left( \int_{T_i} \Phi_k \Phi_l d\mathbf{x} \right) (\hat{\mathbf{u}}_l^{n+1} - \hat{\mathbf{u}}_l^n) + \int_{t^n}^{t^{n+1}} \int_{\partial T_i} \Phi_k \mathcal{G}(\mathbf{q}_h^-, \mathbf{q}_h^+) \cdot \mathbf{n} dS dt \\
& - \int_{t^n}^{t^{n+1}} \int_{T_i} \nabla \Phi_k \cdot \mathbf{F}(\mathbf{q}_h) d\mathbf{x} dt + \int_{t^n}^{t^{n+1}} \int_{\partial T_i} \Phi_k \mathcal{D}^-(\mathbf{q}_h^-, \mathbf{q}_h^+, \mathbf{n}) ds dt \\
& = \int_{t^n}^{t^{n+1}} \int_{T_i} \Phi_k \mathbf{S}(\mathbf{q}_h) d\mathbf{x} dt, \tag{4.12}
\end{aligned}$$

where  $\mathbf{q}_h^-$  denotes the boundary extropolated value from within  $\partial T_i$  whereas  $\mathbf{q}_h^+$  is the one from the neighbor cell. The jumps in  $\mathbf{q}_h$  at the element boundaries are resolved by a path-conservative method that defines a weak derivative in the sense of a Borel measure by means of jump terms  $\mathcal{D}^-(\mathbf{q}_h^-, \mathbf{q}_h^+, \mathbf{n})$  which still remain to be defined [95, 151, 118, 25, 111, 24, 126, 63, 64, 50]. More precisely we use the Dal Maso–Le Floch–Murat theory [108] where the non-smooth part of the non-conservative term is defined as a Borel measure and the smooth part is integrated classically. For the known limitations and deficiencies of path-conservative schemes see [2, 27]. Consequently  $\mathcal{D}^-$  is defined as a function so that  $\mathcal{D}^-(\mathbf{q}, \mathbf{q}, \mathbf{n}) = 0$  and

$$\mathcal{D}^-(\mathbf{q}_h^-, \mathbf{q}_h^+, \mathbf{n}) + \mathcal{D}^-(\mathbf{q}_h^+, \mathbf{q}_h^-, -\mathbf{n}) = \int_0^1 \mathbf{B}(\Psi(\mathbf{q}_h^-, \mathbf{q}_h^+, s) \cdot \mathbf{n}) \frac{\partial \Psi}{\partial s} ds, \tag{4.13}$$

where in this work a simple linear path is considered:  $\Psi(\mathbf{q}_h^-, \mathbf{q}_h^+, s) = \mathbf{q}_h^- + s(\mathbf{q}_h^+ - \mathbf{q}_h^-)$ . A purely numerical integration is used to define the integral in the previous equation as described in [47, 63].

Finally at the end of the timestep we have computed the new solution  $\mathbf{u}_h^*(\mathbf{x}, t^{n+1})$  starting from data at  $t^n$   $\mathbf{u}_h(\mathbf{x}, t^n)$ .

### 4.3.3. Timestep restriction, high order of accuracy, sub-cell resolution and a posteriori stabilization

The Courant-Friedrichs-Lewy (CFL) restriction imposed by the ADER-DG scheme in multiple space dimensions is restricted as [94]

$$\Delta t \leq \frac{1}{d} \frac{1}{(2N + 1)} \frac{h}{|\lambda_{\max}|}, \quad (4.14)$$

where  $h$  and  $|\lambda_{\max}|$  are a characteristic mesh size and the maximum signal velocity, respectively. The factor  $2N + 1$  in the time step is tremendously restrictive particularly in multi-dimensions ( $d = 2$  or  $3$ ). This has to be balanced with the fact that a DG scheme can achieve sub-cell resolution on coarse grid. Obviously, this statement is only true if the DG scheme is able to maintain an overall stability and, at the same time, maintaining the sub-cell resolution. Indeed the nominal space and time order of accuracy of the ADER-DG scheme depicted in these sections is  $N + 1$  if the underlying solution remains smooth enough. The DG scheme as it is described is a so-called *unlimited* scheme which, obviously can not handle steep gradients or a discontinuity (shock, contact, etc.) due to the Gibbs phenomenon leading to spurious numerical oscillations, and, ultimately to code crash. Consequently it is mandatory to design some sort of numerical dissipation mechanism (sometimes called limiter, artificial viscosity, etc.) to *spread* those oscillations before they grow unexpectedly. Because the main advantage of a DG scheme compared to a high order finite volume one is its ability to achieve sub-cell resolution even on coarse grids, the design of the dissipative mechanism is of great importance to maintain this property. Here we employ the new *a posteriori* sub-cell limiter designed, developed and validated in [66] which is briefly recalled in the following section.

## 4.4. A posteriori sub-cell stabilization: detection and ADER-WENO recomputation

The discrete representation of the solution within a general cell  $T_i$  at the beginning of the time step is denoted  $\mathbf{u}_h(\mathbf{x}, t^n)$ . At the next time level  $t^{n+1}$  we first calculate a so-called *candidate* solution, denoted as  $\mathbf{u}_h^*(\mathbf{x}, t^{n+1})$ , which results from the *unlimited* ADER-DG scheme previously described. From these data the *a posteriori* sub-cell limiter will act in two stages [66]:

1. Detect troubled cells in the candidate solution, that is at  $t^{n+1}$ , which are not acceptable given physical and numerical criteria,

2. Re-compute these troubled cells with a robust Finite Volume (FV) type of scheme acting on a large number of sub-cells as to conserve the subcell resolution properties of the DG scheme.

These stages are described in details in the following section 4.4.1 and 4.4.2, respectively.

#### 4.4.1. Detection criteria

The appearance of possible spurious oscillations due to the effect of Gibbs phenomenon renders the candidate solution not acceptable everywhere in the numerical domain. Our sub-cell limiting strategy, like any limiting strategy, must design a mechanism to detect troubled/problematic/bad cells. Therefore a number of detection criteria are devised in order to promote the candidate solution in good cells to *acceptable*. As already mentioned in the Multi-dimensional Optimal Order Detection (MOOD) approach [31, 40, 41, 103] from which our limiter is inspired, the detection of troubled cells is based on physical considerations and it mostly consists of checking if  $\mathbf{u}_h^*(\mathbf{x}, t^{n+1})$  verifies some physical admissibility constraints for the cell  $T_i$  which are dictated by the system of PDEs. In the Baer-Nunziato model Eq. (1.33) physical admissibility is more demanding as it requires the positivity of  $\rho_1, \rho_2, e_1, e_2, p_1, p_2$  and the boundedness of  $\alpha_1$ :  $0 \leq \alpha_1 \leq 1$ . Note that the bound preservation of  $\alpha_1$  is assured thanks to  $\alpha_1 + \alpha_2 = 1$ . Finally, if the curvature  $\kappa$  is large enough, that is the difference between the maximal and minimal local curvature is larger than the characteristics length of the cell, then the cell is considered as problematic because the surface tension force will need extra care. These criteria are referred to as Physical Admissibility Detection criteria (PAD).

The second set of detection criteria deals with numerical issues and is referred to as Numerical Admissibility Detection criteria (NAD). NAD rely on relaxed *discrete maximum principle* expressed for polynomials  $\mathbf{u}_h$  as

$$\min_{\mathbf{y} \in \mathcal{V}_i} \left( \mathbf{u}_h(\mathbf{y}, t^n) \right) - \delta \leq \mathbf{u}_h^*(\mathbf{x}, t^{n+1}) \leq \max_{\mathbf{y} \in \mathcal{V}_i} \left( \mathbf{u}_h(\mathbf{y}, t^n) \right) + \delta \quad \forall \mathbf{x} \in T_i, \quad (4.15)$$

where the set  $\mathcal{V}_i$  contains the current cell  $T_i$  and the cells sharing at least a common node with  $T_i$ . (4.15) expresses the fact that the polynomial representing the candidate solution  $\mathbf{u}_h^*(\cdot, t^{n+1})$  in cell  $T_i$  must remain between the minimum and the maximum values of the polynomials representing the solution at the previous time step  $\mathbf{u}_h(\cdot, t^n)$  in the set  $\mathcal{V}_i$ . The small number  $\delta$  in (4.15) is a parameter used to relax the maximum principle thus allowing for small undershoots and overshoots which permits to maintain a good accuracy

when dealing with smooth extrema. The value used in [66] and adopted in this work is

$$\delta = \epsilon \left( \max_{\mathbf{y} \in \mathcal{V}_i} \left( \mathbf{u}_h(\mathbf{y}, t^n) \right) - \min_{\mathbf{y} \in \mathcal{V}_i} \left( \mathbf{u}_h(\mathbf{y}, t^n) \right) \right), \quad (4.16)$$

with  $\epsilon = 0.001$ . In other words, parameter  $\delta$  defined by (4.16) allows the occurrence of new extrema the values of which do not exceed one thousandth of the local jump present at  $t^n$  in the neighborhood of the current cell.

In practice, if a cell does not fulfill the PAD criteria, then it is flagged as problematic and must be recomputed. Next the NAD criteria are tested for the remaining cells which may or may not be flagged as problematic. The result of this step is a list of problematic cells along with a patch of surrounding neighbor cells; these form the so-called troubled cells. In this approach, physical and the numerical criteria are totally independent. Consequently the relaxation of the maximum principle never affects the positivity of the solution. More important, the detection is performed at time level  $t^{n+1}$  whereas classical indicators typically use information from one time level, generally  $t^n$ . This subtle but crucial difference allows for a *simple observation* of problems which have occurred during the timestep, contrarily to classical limiting strategy using  $t^n$  data which must solve the more *difficult* problem of *predicting* their occurrence.

#### 4.4.2. Sub-cell ADER-WENO recomputation

Considering one cell  $T_i$  of the list of troubled cells obtained from the detection step, we pave  $T_i$  with a sub-grid made of  $(N_s)^d$  sub-cells  $S_{i,j}$ , with  $j = 1, \dots, (N_s)^d$ , where  $N_s = 2N + 1$ . An *alternative data representation*  $\mathbf{v}_h(\mathbf{x}, t^n)$  onto the sub-cells expressed by a set of piecewise constant sub-cell averages  $\mathbf{v}_{i,j}^n$ , computed as the  $L_2$  projection of  $\mathbf{u}_h$  onto  $S_{i,j}$  as:

$$v_{i,j}^n = \frac{1}{|S_{i,j}|} \int_{S_{i,j}} \mathbf{u}_h(\mathbf{x}, t^n) d\mathbf{x} = \frac{1}{|S_{i,j}|} \int_{S_{i,j}} \phi_l(\mathbf{x}) d\mathbf{x} \hat{\mathbf{u}}_l^n, \quad \forall S_{i,j} \in \mathcal{S}_i, \quad (4.17)$$

where we denote by  $\mathcal{S}_i = \bigcup_j S_{i,j}$  the set of the sub-grid cells, i.e. the sub-cells. Following [66] the choice  $N_s = 2N + 1$  is doubly motivated. On the one hand, this choice, by construction, guarantees that the timestep of the ADER-DG scheme on the main grid computed by (4.14) matches the timestep of the finite volume scheme on the sub-grid. As a consequence there is no need for extra manipulation such as local time-stepping technique. On the other hand, the choice  $N_s = 2N + 1$  leads to the maximum admissible CFL number, thus minimizing the error, which scales as  $(1 - \text{CFL})$  for the linear advection equation [56]. Using the new data representation  $\mathbf{v}_h(\mathbf{x}, t^n)$  (i.e. piecewise constant data

$v_{i,j}^n$  on sub-cells) as initial conditions, the discrete solution at time  $t^{n+1}$  is re-computed by means of a robust finite volume type of scheme on the sub-grid. Any scheme from the finite volume family can be considered as long as it is robust for the system of PDEs considered; 1st order FV, 2nd order TVD, 3rd/5th order WENO, etc. In this work we adopt a third order ADER-WENO scheme [51, 52, 45, 11, 58, 50]. Because the ADER-DG (main grid) and ADER-WENO (sub-grid) are one-step temporal schemes, the total amount of MPI communications is greatly reduced compared to traditional schemes requiring substages, such as Runge-Kutta schemes. Once a robust sub-cell solution  $\mathbf{v}_h(\mathbf{x}, t^{n+1})$  for all sub-cells of the troubled cells has been recomputed, the DG polynomial on the main grid must be recovered. This is achieved by requiring that the sub-cell solution  $v_{i,j}^{n+1}$  in  $S_{i,j}$  be equal to the  $L_2$  projection of the unknown DG polynomial  $\mathbf{u}_h(\cdot, t^{n+1})$  onto  $S_{i,j}$

$$\int_{S_{i,j}} \mathbf{u}_h(\mathbf{x}, t^{n+1}) d\mathbf{x} = \int_{S_{i,j}} \mathbf{v}_h(\mathbf{x}, t^{n+1}) d\mathbf{x}, \quad \forall S_{i,j} \in \mathcal{S}_i. \quad (4.18)$$

The previous equation is equivalent to solving

$$\int_{S_{i,j}} \phi_l(\mathbf{x}) d\mathbf{x} \hat{\mathbf{u}}_l^{n+1} = v_{i,j}^{n+1}, \quad \forall S_{i,j} \in \mathcal{S}_i, \quad (4.19)$$

which is a standard reconstruction problem arising both within the finite volume context as well as for spectral finite volume methods [156, 100, 155]<sup>1</sup>.

## 4.5. AMR framework

In this section we briefly describe the AMR framework which is exhaustively described in [159]. Due to its simple tree-type structure and for its more general formulation, we have adopted the cell-by-cell strategy for which each cell is individually refined. This framework is particularly well suited for the local space-time DG predictor and the fully discrete one-step ADER-DG scheme. First of all the DG predictor acts on every active cell without exchanging any information with neighbors, second, the ADER-DG scheme has a direct unstructured non-conforming grid interpretation in the presence of different levels of grid refinement between two adjacent cells. By defining the maximum level of refinement  $\ell_{\max}$ , each level of refinement is indicated by  $\ell$  such that  $0 \leq \ell \leq \ell_{\max}$ . We are therefore considering up to  $\ell_{\max}$  overlaying uniform lattices, whose cells are activated only

---

<sup>1</sup>In the case a cell is marked as troubled for the subsequent timestep we keep the sub-cell data  $\mathbf{v}_h(\mathbf{x}, t^{n+1})$  obtained from the ADER-WENO finite volume scheme of the previous time step to avoid extra  $L_2$  projection. If the cell is no more problematic then the sub-cell data are canceled.

if necessary. Every cell is labeled by a positive integer number  $m$  and can be denoted as  $\mathcal{C}_m$ , with  $m \leq N_e$ , where  $N_e$  is the (time-dependent) total number of the cells. Moreover, each cell of a given  $\ell$ -th level has one status  $\Sigma$  among three possible ones:

- $\Sigma = 0$  for *active* cells updated with the finite element ADER-DG scheme described in Section 4.3.2;
- $\Sigma = 1$  for *virtual children* cells, or virtual children, updated according to standard  $L_2$  projection of the high order polynomial of the so-called *mother* cell at level  $(\ell - 1)$ -th;
- $\Sigma = -1$  for *virtual mother* cells, or virtual mothers, updated by averaging recursively the *children* of the upper refinement levels, from the  $(\ell + 1)$ -th to the level of the corresponding *active* children.

Whenever  $\mathcal{C}_m$  is refined, it generates  $\mathfrak{r}^d$  children, such that

$$\Delta x_\ell = \mathfrak{r} \Delta x_{\ell+1}, \quad \Delta y_\ell = \mathfrak{r} \Delta y_{\ell+1}, \quad \Delta z_\ell = \mathfrak{r} \Delta z_{\ell+1}, \quad (4.20)$$

where  $\mathfrak{r}$  is the so-called *refinement factor*. The time steps can be chosen locally (see [53, 144, 102]) depending on the refinement level, such that  $\Delta t_\ell = \mathfrak{r} \Delta t_{\ell+1}$ , to increase the performance.

Like for any AMR technique we design a criterion to flag any given active cell  $\mathcal{C}_m$  as a cell requiring refinement or recoarsening. We therefore introduce a refinement diagnostics function  $\chi_m$ , built according to [98, 109, 159], which involves up to the second order derivative of an *indicator function*  $\Phi$ , i.e.

$$\chi_m(\Phi) = \sqrt{\frac{\sum_{k,l} \left( \frac{\partial^2}{\partial x_k \partial x_l} \Phi \right)^2}{\sum_{k,l} \left[ \left( \left| \frac{\partial \Phi}{\partial x_k} \right|_{i+1} + \left| \frac{\partial \Phi}{\partial x_k} \right|_i \right) / \Delta x_l + \varepsilon \left| \frac{\partial^2}{\partial x_k \partial x_l} \Phi \right| |\Phi| \right]^2}}. \quad (4.21)$$

The sum  $\sum_{k,l}$  is intended to be the double summation over all the spatial directions, so that cross derivatives contributions are properly taken into account. The refinement diagnostics ( $\mathcal{R}$ ) is built as

$$(\mathcal{R}) \begin{cases} \chi_m > \chi_{\text{ref}} & \text{then } \mathcal{C}_m \text{ needs refinement,} \\ \chi_m < \chi_{\text{rec}} & \text{then } \mathcal{C}_m \text{ needs recoarsening.} \end{cases} \quad (4.22)$$

The *free* parameters are  $\Phi = \Phi(\mathbf{u})$  a generic function of the conservative variables,  $\chi_{\text{ref}}$ ,  $\chi_{\text{rec}}$  upper and lower bounds and  $\varepsilon$  a filter-parameter that avoids unnecessary mesh-refinement in regions affected by wiggles. In this work we have used  $\Phi(\mathbf{u}) = \rho_1\alpha_1 + \rho_2\alpha_2$ ,  $\chi_{\text{ref}}$  and  $\chi_{\text{rec}}$  are typically chosen in the range  $\sim [0.2, 0.25]$  and  $\sim [0.05, 0.15]$ , respectively, and  $\varepsilon = 0.01$ . Full details about the implementation, the parallelization and the incorporation of the sub-cell limiter are available in [58, 50, 159].

## 4.6. Curvature computation

At first glance the computation of a curvature given the volume fraction function  $\phi(\mathbf{x})$  may seem relatively simple by following Perrigaud *et al.* [121] and this formula has been applied in section 2.4.2 as follows:

$$\kappa = \nabla \cdot \mathbf{m} = \nabla \cdot \left( \frac{\nabla \phi}{\|\nabla \phi\|} \right). \quad (4.23)$$

In order to formally *derive* two times  $\phi$ , enough numerical smoothness is needed to avoid spurious oscillations. This is the main reason why several authors, instead of using function  $\phi$  in (4.23) rather use a so-called color function  $c$  which is a smoothed enough version of  $\phi$ . Obviously the devil lies in the details on how  $c$  is computed and how spurious oscillation on  $\mathbf{m}$  and later  $\kappa$  could be avoided. Facing the same problem we have chosen to rely on a color function initially defined by

$$c(\mathbf{x}, t = 0) = f(\phi(\mathbf{x}, t = 0)), \quad (4.24)$$

where  $f = \text{erf}$  function. This color function evolves as  $\phi$  with the interface velocity and is considered as an extra variable to our system of PDEs

$$\frac{\partial}{\partial t} c + \mathbf{v}_I \nabla c = \nu(p_1 - p_2). \quad (4.25)$$

As already described in the *a posteriori* sub-cell stabilization section 4.4, any cell capturing the interface between the two phases is detected as a troubled cell. As a consequence the starting data are not exactly the DG polynomial in cell  $T_i$  but the finite volume data on sub-cells  $S_{i,j}$  located in the vicinity of  $T_i$ ,  $c_{i,j} \in \mathbb{P}_0(S_{i,j})$ . Because a 3rd order WENO scheme is used in the sub-cell, we have access to the reconstruction  $c \in \mathbb{P}_2(S_{i,j})$  in each sub-cell. From these sub-cell polynomials we increase the regularity by reconstructing  $\tilde{c}_{i,j} \in \mathbb{P}_4(S_{i,j})$ . Next we formally derive  $\tilde{c}_{i,j}$  as  $\mathbf{m} = (m_1, m_2, m_3)^t = \nabla \tilde{c}_{i,j} / \|\nabla \tilde{c}_{i,j}\|$  and evaluate  $\mathbf{m}$  for each  $\mathbf{x}_k$  which is a degree of freedom of the  $\mathbb{P}_2$  basis. Therefore we have access to  $\mathbf{m} = (m_1, m_2, m_3)^t$ ,  $m_k \in \mathbb{P}_2(S_{i,j})$  in each sub-cell. Mimicking the treatment of

$c$ , we start again from  $\mathbf{m} \in \mathbb{P}_2(S_{i,j})$  and reconstruct  $\widetilde{\mathbf{m}} = (\widetilde{m}_1, \widetilde{m}_2, \widetilde{m}_3)^t$ ,  $\widetilde{m}_k \in \mathbb{P}_4(S_{i,j})$  and, formally, apply the divergence operator to evaluate the curvature  $\kappa(\mathbf{x}) = \nabla \cdot \widetilde{\mathbf{m}}(\mathbf{x})$  for any point  $\mathbf{x} \in S_{i,j}$ . A threshold value of the curvature is used to cut too small/large values, which in any case would have no effect on the timestep results but may lead in unexploitable curvature evaluations.

$$\kappa = \begin{cases} \kappa & \text{if } |\kappa| > \kappa_0 \\ 0 & \text{else} \end{cases} \quad (4.26)$$

where  $\kappa_0 = (2N + 1)/L$  and  $L$  is the characteristic length of the sub-cell. In our approach the access to a genuinely high order accurate polynomial description of  $\phi$  and  $c$  in the main cell of the DG scheme, or, in the sub-cell of the WENO scheme, is somewhat equivalent to having an almost interface capturing method on a sub-cell based resolution. In other words the interface between phases is never spread over more than one DG cell length, rather it lives as a *sub-cell* entity (i.e. spread over a maximum of three to four sub-cells). Consequently in our approach, the diffused interface can travel across one cell.

In order to assess this statement we perform several static and dynamic computations of curvature starting from relatively simple but sharp initial interfaces. For these computations we plot the true WENO polynomials on the sub-cells for the interface function  $\phi$ , the color function  $c$  and the curvature  $\kappa$ . (In other word the true values which are actually used in the numerical scheme are displayed.) The 7th order accurate ADER DG scheme with 3rd order WENO *a posteriori* sub-cell limiting is used. This means that the number of sub-cells per direction is 13 (i.e  $2N + 1$  with  $N = 6$ ). The number of initial cells in the domain is restricted on purpose to  $15^d$  with  $d$  the dimension of space <sup>2</sup>. The following two tests are only designed to illustrate the behavior of this curvature reconstruction algorithm. Numerical tests involving the full machinery to solve physically relevant problems will be proposed in the next section.

#### 4.6.1. Test #1: static curvature computation

Let us initialize a simulation with an ellipsoidal interface centered at the origin and elongated in the  $x$  direction

$$\phi(\mathbf{x}) = ax^2 + by^2 + cz^2. \quad (4.27)$$

In this section, we first verify the computation, which corresponds to an ellipse in 2D ( $a = 4, b = 16, c = 0$ ). In Figure 4.1 are presented the initial data  $\phi$  and the associated

---

<sup>2</sup>We have observed better results with higher resolution, however the main reason why using a DG scheme is to explicitly avoid employing ultra fine meshes.



computed curvature  $\kappa$  and limiter in this computation (recall that the red DG cells are limited, hence treated with a sub-cell finite volume scheme). The plotted data are the true DG  $\mathbb{P}_6$  polynomials and not a mean value of those polynomials in the cells. The last panel presents a zoom on the mesh and number of degree of freedom in DG cell (sub-mesh in blue cells). On the same figure we plot for limited cells (red) the actual sub-cell resolution used for the FV scheme. As expected the representation of the polynomials in the sub-cells is almost continuous and symmetric with correct values for the interface as well as for the ellipse. Note that the color function is a diffused version of the interface data.

#### 4.6.2. Test #2: simple dynamic curvature computation

In this test one starts from the 2D ellipse initialized in the previous test and let it evolve in time to retrieve a cylindrical shape as the surface tension force acts. We do not measure the physical validity of the results (this will be studied in the numerical section later), only do we observe the evolution of the discrete curvature to assess that our sub-cell computation does not destroy its shape, and, more important, is able to advect the interface across one coarse DG cell. In Figure 4.2 we present several snapshots in time of the volume fraction of the solid phase. The snapshots are at early time apart from the last one which show the volume fraction at late time after several oscillations. As expected the bubble adopts an oscillatory motion, deforming from an  $x$  aligned ellipse into a  $y$  aligned one and vice-versa. We observe that there is little diffusion of the interface, i.e. it is captured at sub-cell level. Moreover an almost symmetry preservation is observed as the simulation marched in time even if the interface endures significant distortion which are non-aligned with the main directions of the mesh. In Figure 4.3 are shown the limiter values: red cells are treated with the subcell limiter and we observe that the curvature computation takes place within these cells. Note that other cells away from the interface may also be detected as problematic but this does not affect the overall scheme. At later time (right panels) the interface is still well captured and not extremely diffused. The top panels present the DG cells which are not limited in blue and, in red, the FV sub-cells. The middle panels show a 3D view where the elevation corresponds to the solid volume fraction and the color to the limiter at the same intermediate times than for the top panels. We clearly observe that the interface is properly captured within one DG cell. Last, the bottom panels show the solid volume fraction in color and elevation along with the underlying resolution.

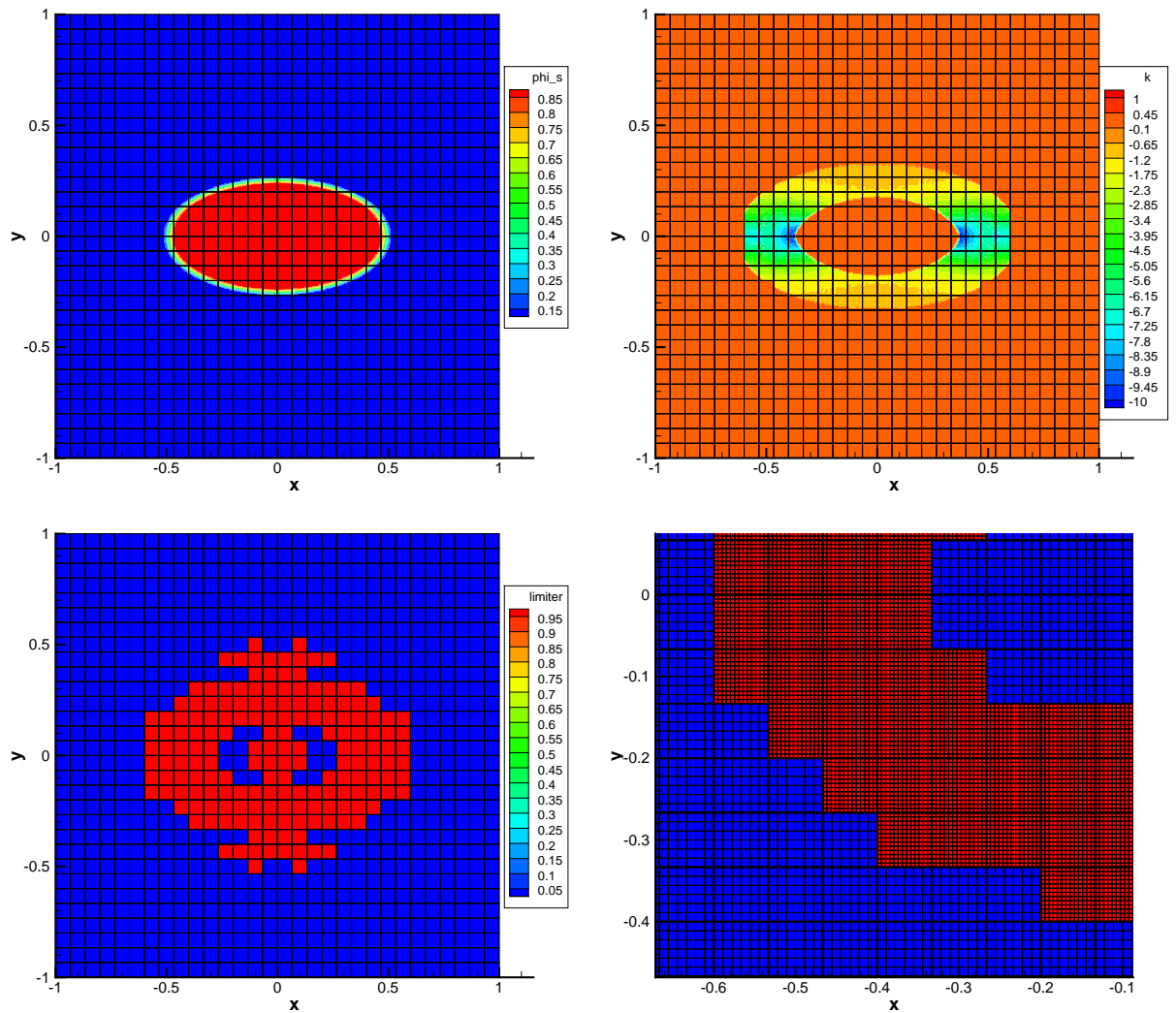


Figure 4.1.: Curvature computation in the case of a static initial 2D ellipsoidal bubble. Top-left: solid volume fraction. Top-right: curvature. Bottom-left: limiter. Bottom-right: zoom on the mesh (thick line), number of degree of freedom in DG cell (submesh in blue cells) and true sub-cell resolution used in limited red cells.

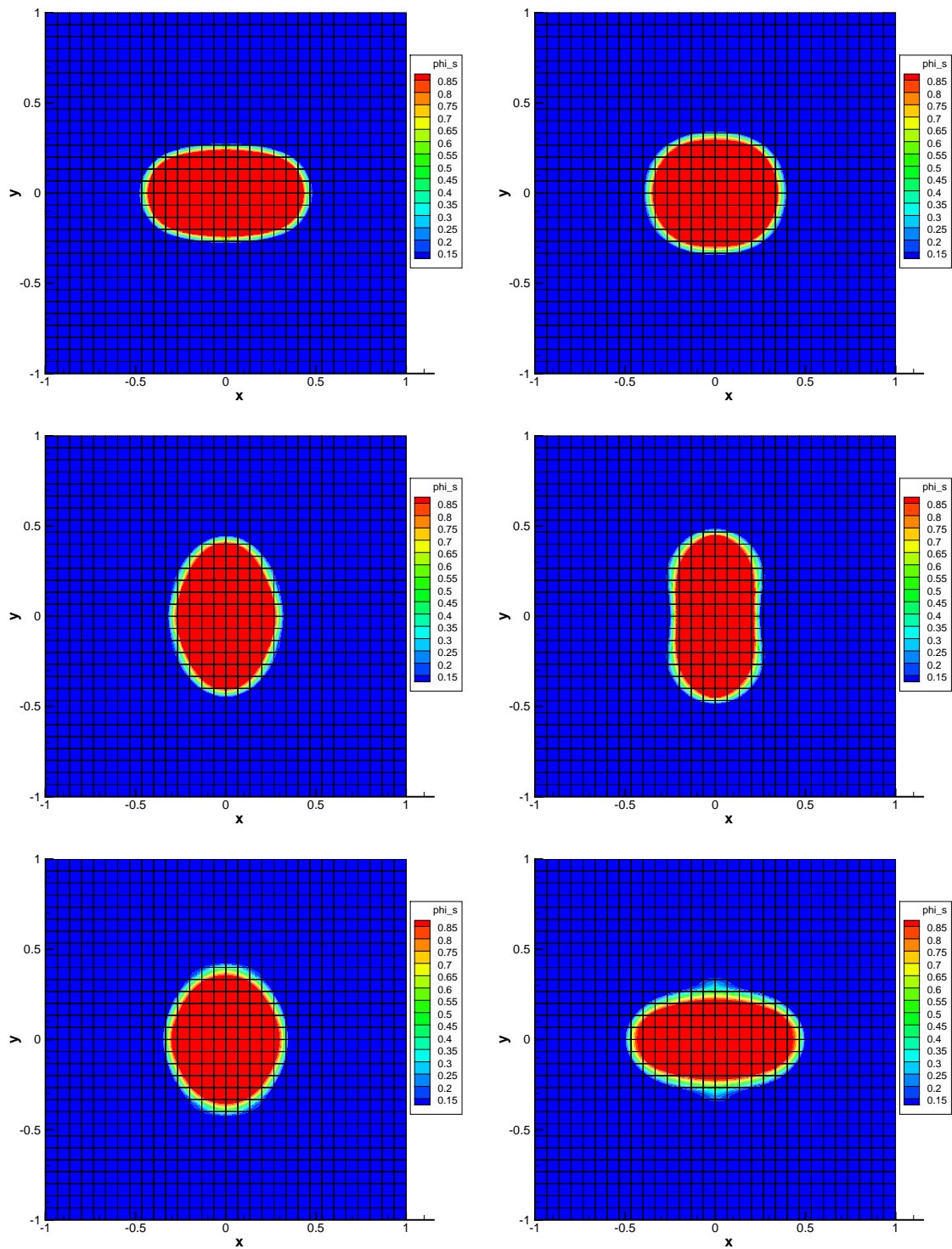


Figure 4.2.: Curvature computation in the case of an evolving 2D initial ellipsoidal bubble. Solid volume fraction. From top-left to bottom-right intermediate times. Last panel is at later time after several oscillations.

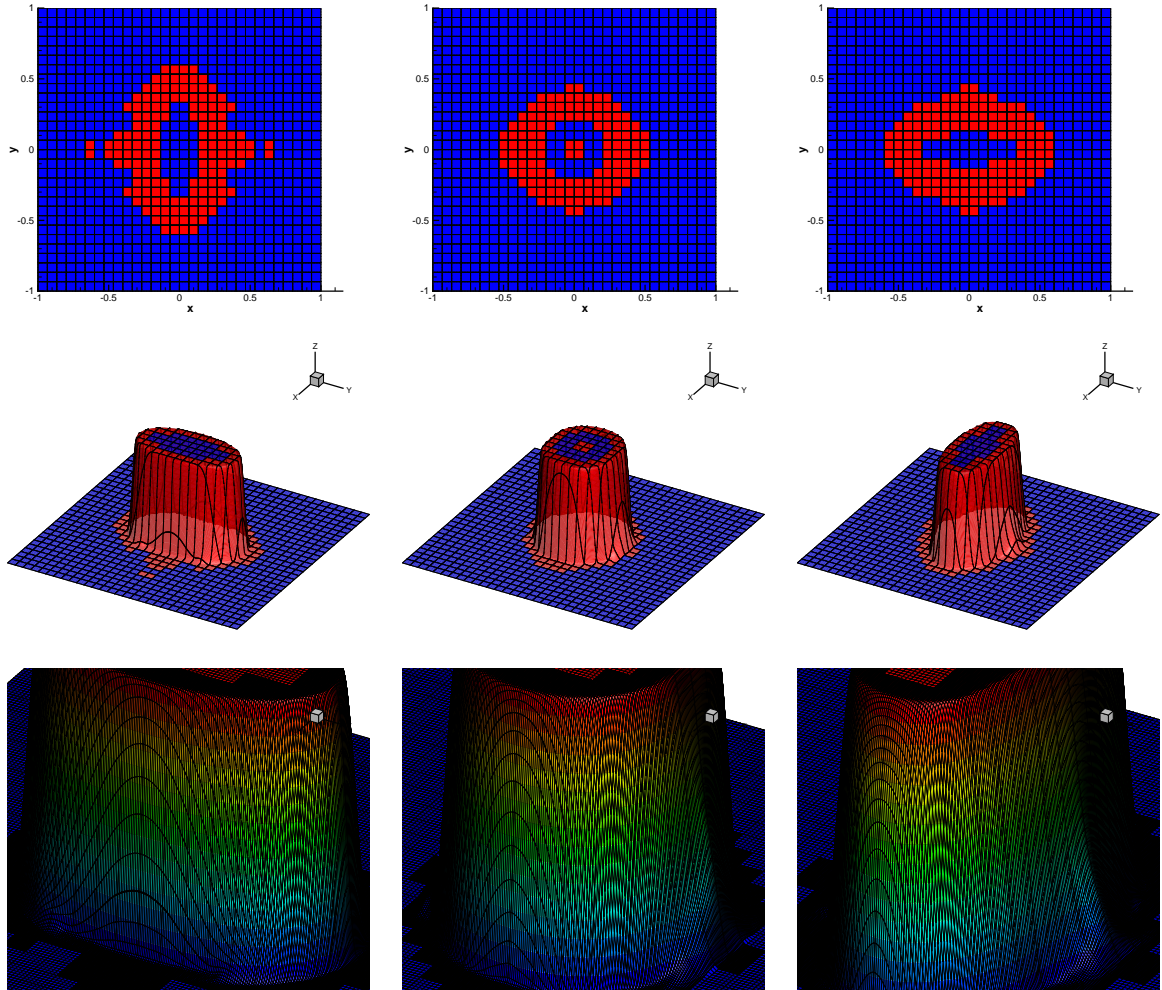


Figure 4.3.: Curvature computation in the case of an evolving 2D initial ellipsoidal bubble. Limiter value (color) and solid volume fraction (elevation). Right panels are at later time after several oscillations whereas the first two are after one oscillation. Bottom panels present a zoom on the submesh and of the solid volume fraction to show the underlying resolution.



# 5. Numerical Results

In this chapter we show numerical results for the numerical methods presented in this thesis.

## 5.1. One-dimensional tests

### 5.1.1. TVD finite volume schemes.

In this section, we first verify our scheme on simple one-dimensional shock tube problems. In particular, we verify if our scheme is able to maintain a bubble in equilibrium according to the Young-Laplace law. For simplicity, we used the initial conditions for the Riemann problems provided in [62], but including surface tension and a constant (prescribed) value of the curvature. The data used for the 1D Riemann problems (RP) are given in Table 5.1. The five test problems are solved in the spatial domain  $[0, 1]$  using 400 cells and the Courant-Friedrichs-Levy (CFL) number is taken as  $CFL= 0.9$ . For these 1D test problems, viscosity, gravity and relaxation terms have been neglected.

The numerical results with the new generalized Osher-type scheme (3.11) are compared with the path-conservative Rusanov-type scheme (3.10) and with the path-conservative Roe-type scheme (3.12), see [129, 127, 25, 119, 28, 47, 64, 65]. For the test RP1, the value of the surface tension  $\sigma$  and the curvature  $\kappa$  are constant, but the initial pressures of the two fluids are not consistent with the Young-Laplace condition, hence the complex wave structure of the Riemann problem for the BN system develops, consisting of shocks and rarefaction waves in both phases, together with the material contact wave. The numerical results are shown in Figure 5.1. For test RP2 and RP4, without the surface tension effect, the system becomes the original Baer-Nunziato model, and the numerical results are seen in Figure 5.2 and Figure 5.4, respectively. Its behavior is in agreement with the solutions observed in [62], where also a comparison with the exact Riemann solvers [5, 137, 39] has been provided. In the test RP3 and test RP5, the value of the surface tension  $\sigma$  and the curvature  $\kappa$  are constant, like in test 1, but the pressure jump at the interface this time obeys the Young-Laplace equation. The numerical solutions are shown in Figure 5.3 and

RP	$\rho_1$	$u_1$	$p_1$	$\rho_2$	$u_2$	$p_2$	$\alpha_1$	$t_e$
<b>RP1</b>	$\gamma_1 = 1.4$	$\pi_1 = 0$	$\gamma_2 = 1.4$	$\pi_2 = 0$	$\sigma = 1$	$\kappa = -1$		
<b>L</b>	1.0	0.0	1.0	0.5	0.0	1.0	0.4	0.15
<b>R</b>	2.0	0.0	2.0	1.5	0.0	2.0	0.8	
<b>RP2</b>	$\gamma_1 = 3.0$	$\pi_1 = 100$	$\gamma_2 = 1.4$	$\pi_2 = 0$	$\sigma = 0$	$\kappa = -1$		
<b>L</b>	800.0	0.0	500.0	1.5	0.0	2.0	0.4	0.15
<b>R</b>	1000.0	0.0	600.0	1.0	0.0	1.0	3.0	
<b>RP3</b>	$\gamma_1 = 1.4$	$\pi_1 = 0$	$\gamma_2 = 1.4$	$\pi_2 = 0$	$\sigma = 1$	$\kappa = -1$		
<b>L</b>	2.0	0.0	2.0	1.0	0.0	1.0	0.9	0.15
<b>R</b>	2.0	0.0	2.0	1.0	0.0	1.0	0.1	
<b>RP4</b>	$\gamma_1 = 3.0$	$\pi_1 = 3400$	$\gamma_2 = 1.35$	$\pi_2 = 0$	$\sigma = 0$	$\kappa = -1$		
<b>L</b>	1900.0	0.0	10.0	2.0	0.0	3.0	0.2	0.15
<b>R</b>	1950.0	0.0	1000.0	1.0	0.0	1.0	0.9	
<b>RP5</b>	$\gamma_1 = 3.0$	$\pi_1 = 3400$	$\gamma_2 = 1.35$	$\pi_2 = 0$	$\sigma = 1$	$\kappa = -1$		
<b>L</b>	1900.0	0.0	2.0	1.0	0.0	1.0	0.99	0.15
<b>R</b>	1900.0	0.0	2.0	1.0	0.0	1.0	0.01	

Table 5.1.: Initial state left(L) and right(R) for the Baer-Nunziato problem solved in 1D with the surface tension effect.

Figure 5.5. We find that the initial constant density and pressure profiles are perfectly maintained. The detailed  $L_\infty$  errors for the velocities of both phases are reported in Table 5.2 for both cases and for various machine precisions. We find that the error is always close to the chosen machine accuracy. From these results we observe numerically, that our path-conservative scheme is well-balanced up to machine precision for the Baer-Nunziato system including the surface tension effect. In other words, our scheme is *well-balanced* for a steady circular bubble in equilibrium, if the exact curvature  $\kappa$  and the pressure jump according to the Young-Laplace law (1.61) are prescribed, and if the interface is exactly located on an edge of the computational grid.

### 5.1.2. Path-conservative HLLEM-type scheme.

In this section, like in section 5.1.1 we first verify our scheme on simple one-dimensional shock tube problems. The data used for the 1D Riemann problems (RP) are given in Table 5.1. The five test problems are solved in the spatial domain  $[0,1]$  using 400 cells and the Courant-Friedrichs-Levy (CFL) number is taken as  $CFL = 0.9$ . A constant factor  $\varphi = 1$  is taken for the HLLEM Riemann solver, hence the flattener is not used here. The numerical results obtained with the new HLLEM-type scheme are compared with the

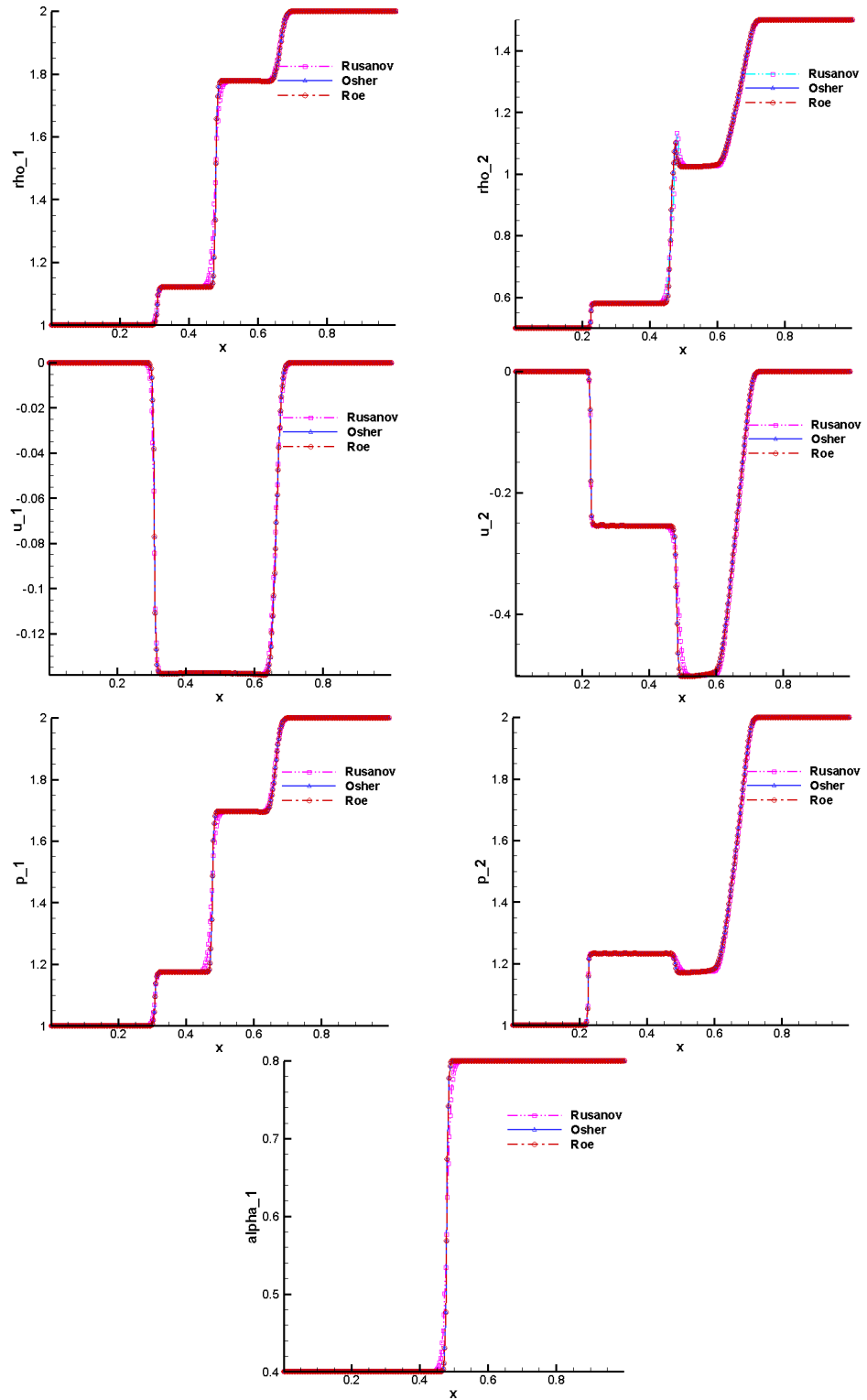


Figure 5.1.: Numerical results for the two phase RP1 at time  $t = 0.15$  s.



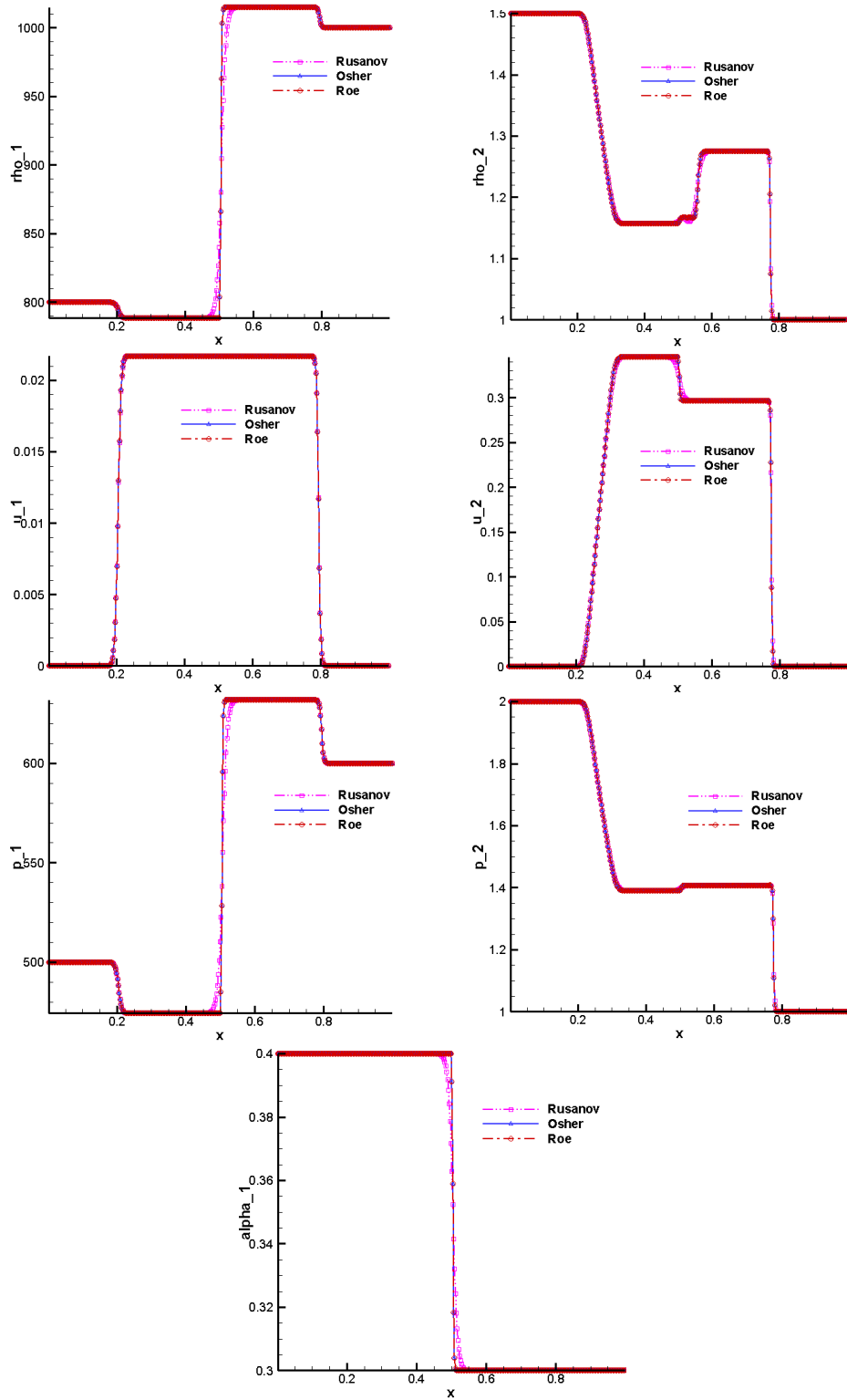


Figure 5.2.: Numerical results for the two phase RP2 at time  $t = 0.15$  s.

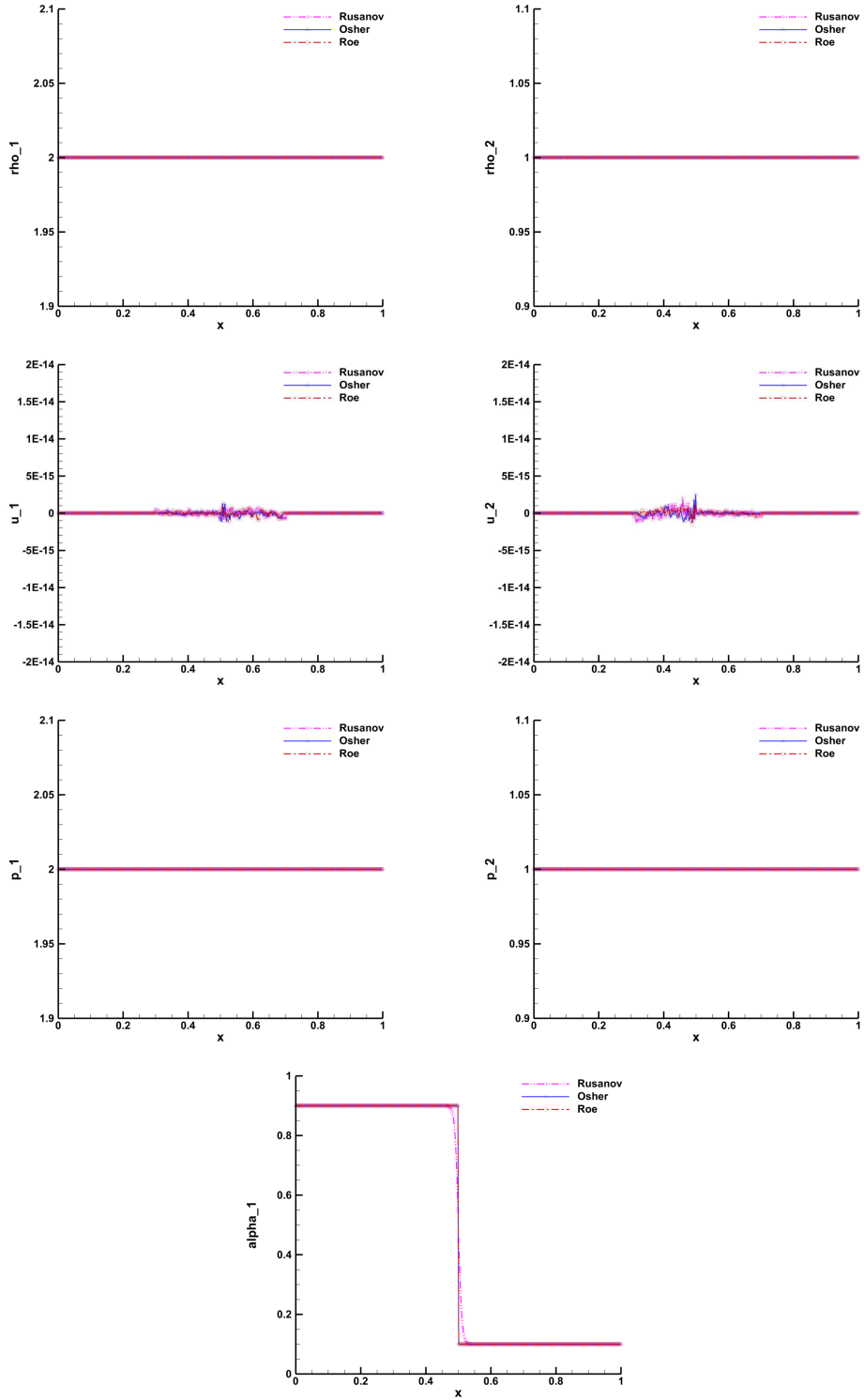


Figure 5.3.: Numerical results for the two phase RP3 at time  $t = 0.15$  s.

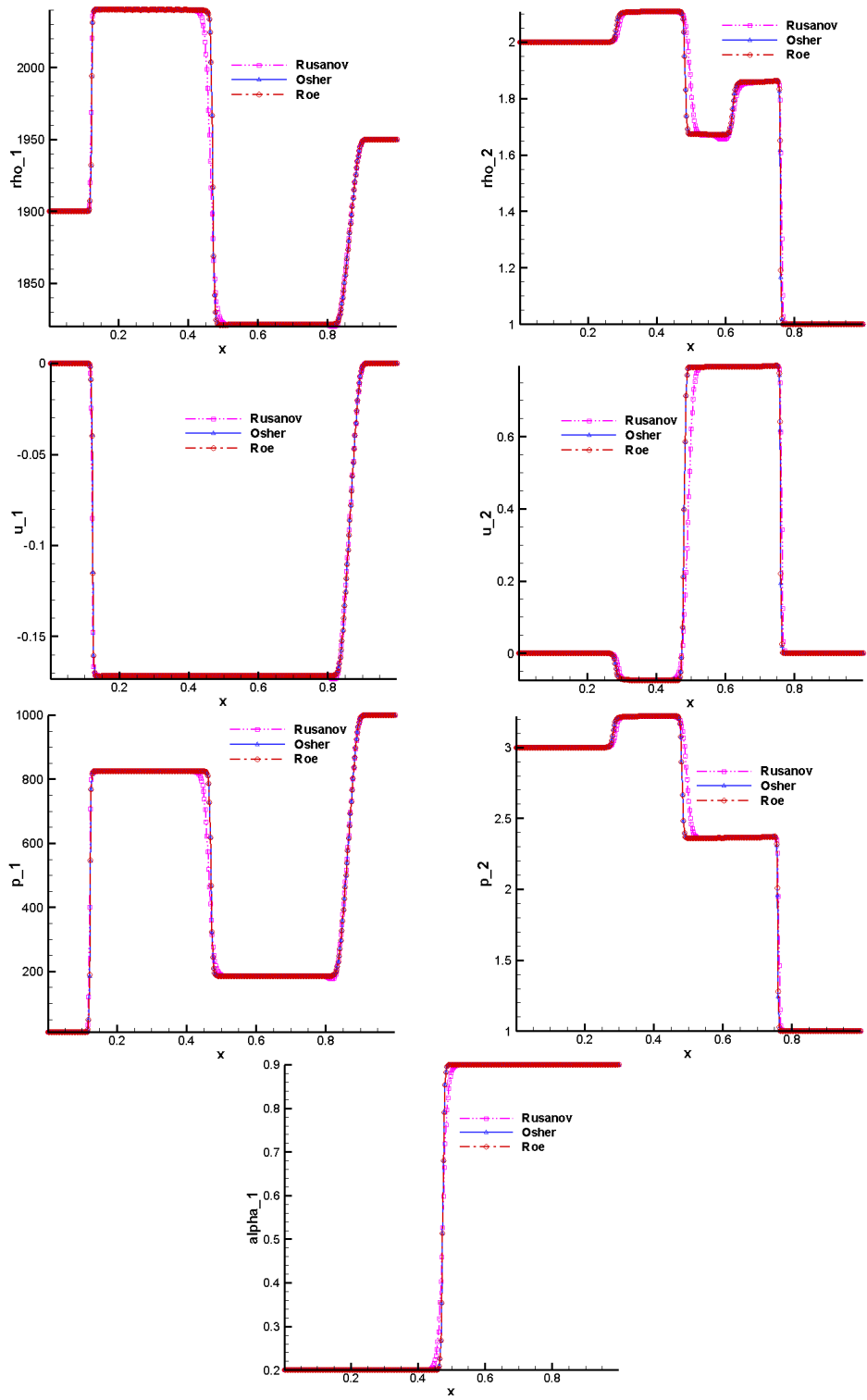


Figure 5.4.: Numerical results for the two phase RP4 at time  $t = 0.15$  s.

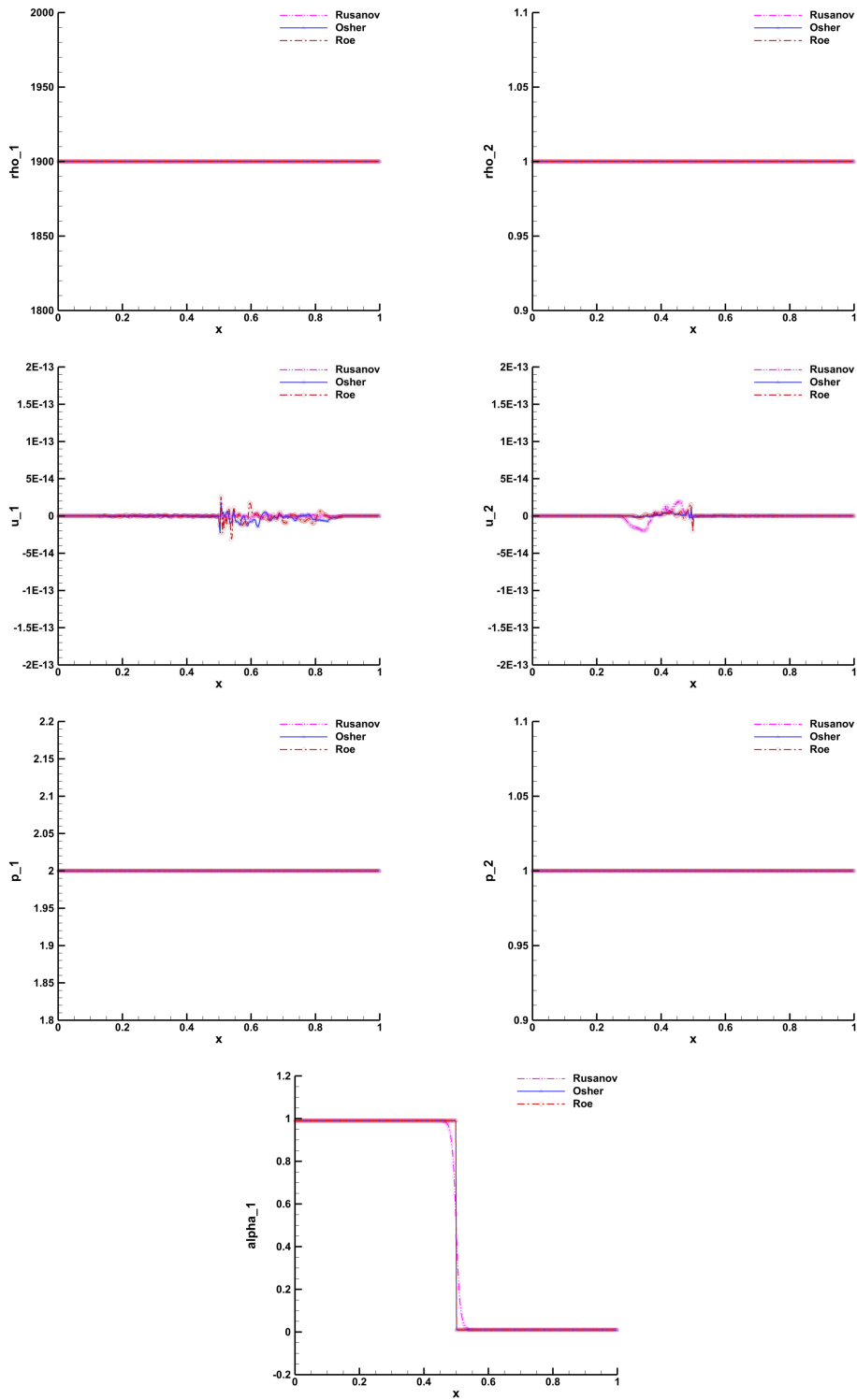


Figure 5.5.: Numerical results for the two phase RP4 at time  $t = 0.15$  s.

<b>Case RP3</b>		$L_\infty$ error $u_1$	$L_\infty$ error $u_2$
Single precision	REAL(4)	$1.0714515 \cdot 10^{-6}$	$1.7865468 \cdot 10^{-6}$
Double precision	REAL(8)	$1.3435343 \cdot 10^{-15}$	$2.5479857 \cdot 10^{-15}$
Quadruple precision	REAL(16)	$1.4892956 \cdot 10^{-33}$	$2.1358886 \cdot 10^{-33}$
<b>Case RP5</b>		$L_\infty$ error $u_1$	$L_\infty$ error $u_2$
Single precision	REAL(4)	$1.2037556 \cdot 10^{-5}$	$5.4810471 \cdot 10^{-6}$
Double precision	REAL(8)	$2.3044841 \cdot 10^{-14}$	$1.1247634 \cdot 10^{-14}$
Quadruple precision	REAL(16)	$1.3201379 \cdot 10^{-32}$	$7.8649468 \cdot 10^{-33}$

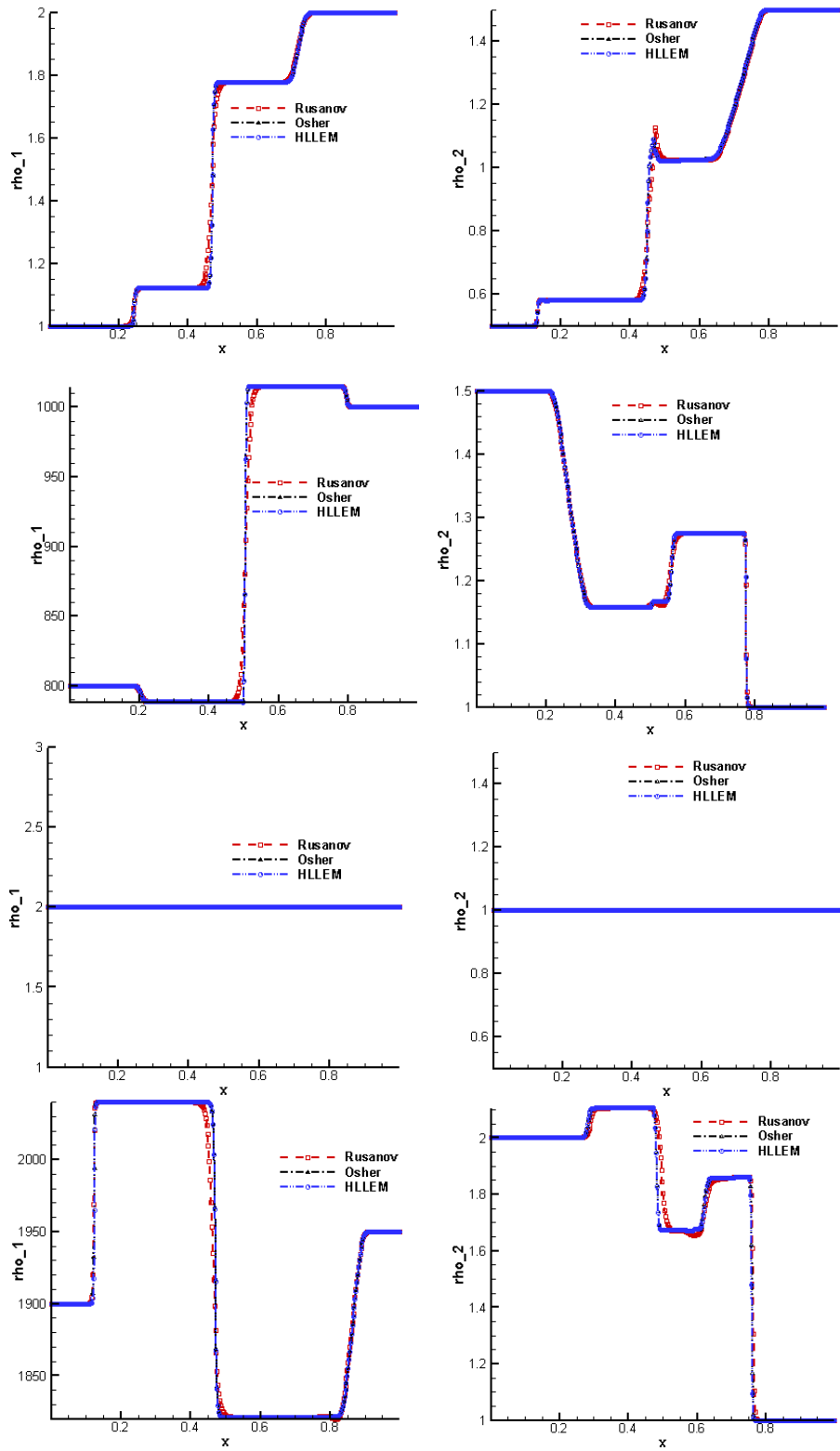
Table 5.2.: Numerical verification of the exact well-balanced property of the DOT Riemann solver for the steady bubble in equilibrium in 1D for different machine precisions. The  $L_\infty$  errors refer to the velocities of the two phases  $u_1$  and  $u_2$ , respectively. In both tests the exact curvature  $\kappa = -1$  and the exact pressure jump  $\Delta p = \sigma \kappa$  have been prescribed.

Rusanov scheme [129] and the new Osher-type scheme of Dumbser and Toro [65, 64]. The numerical results like in section 5.1.1, for RP1, the value of the surface tension  $\sigma$  and the curvature  $\kappa$  are constant but the pressure jump is not consistent with the Young-Laplace principle. In RP2 and RP4 without the surface tension effect, the system becomes the original Baer-Nunziato system. For Riemann problems RP3-RP5, we confirm that the new HLLEM-type scheme satisfies the pressure equilibrium condition under the surface tension effect exactly at the discrete level. In other words, our scheme is *well-balanced* for the bubble in equilibrium, if the exact curvature and pressure jumps are prescribed, and if the interface is exactly located on an edge of the computational grid. The numerical results of RP1-RP5 are shown in Figure 5.6.

## 5.2. Two dimensional tests

### 5.2.1. TVD finite volume schemes on a two dimensional Cartesian grid

In this section, various two-dimensional test problems are solved according to some test cases proposed in the literature, see [121]. In the first test, we run a numerical mesh



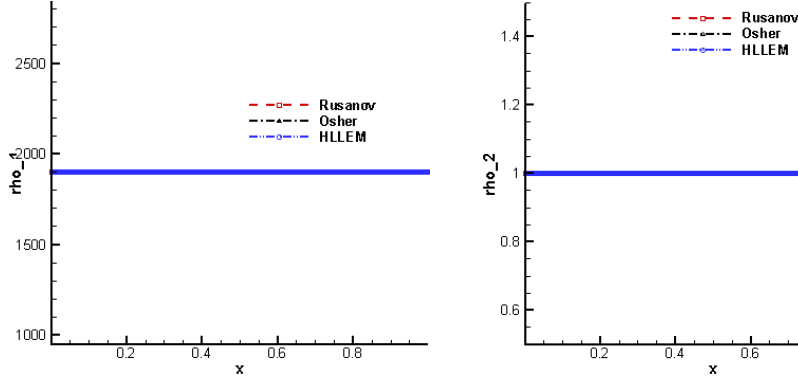


Figure 5.6.: Numerical results for Riemann problems RP1-RP5 (from top to bottom).  
Left: Solid density  $\rho_1$ . Right: Gas density  $\rho_2$ .

convergence study for a static droplet in equilibrium. The second test considers the oscillation of a deforming droplet under surface tension effect, but without gravity. In the third test, the dynamics of an initially wall-bound droplet with gravity and surface tension effects are considered. Finally, we consider the influence of viscosity, gravity and surface tension on the rising of a gas bubble. The CFL number is taken as  $CFL = 0.45$  for all 2D test problems.

### 5.2.1.1. Steady bubble in equilibrium and numerical mesh convergence study

In this section, the numerical results will be confirmed as follows:

We first carry out a mesh convergence study, in order to validate our path-conservative Osher-type scheme for the Baer-Nunziato system with surface tension effects. The initial condition is given as follows: a circular bubble with an initial radius of  $R = 0.15$  is centered at  $(0.5, 0.5)$ . The computational domain is defined as  $\Omega = [0, 1] \times [0, 1]$ , which is discretized with a two-dimensional Cartesian grid. The parameters of the liquid phase 1 are  $\gamma_1 = 2.4$  and  $\pi_1 = 10^7$ , and the parameters of the gas phase 2 are  $\gamma_2 = 1.4$  and  $\pi_1 = 0$ . Both phases are initially at rest, i.e.  $\mathbf{u}_1 = \mathbf{u}_2 = 0$ . We furthermore set  $\rho_1 = 1000$ ,  $\rho_2 = 1$  and  $\sigma = 200$ . The exact pressure jump based on the Young-Laplace principle is  $\Delta p = \sigma\kappa = 1333.33$ . The pressures are initialized with  $p_2 = 1$  and  $p_1 = p_2 + \Delta p$ . The error norms refer to the pressure jump ( $\Delta p = \sigma\kappa$ ), and the error norm in  $L_2$  norm is computed from the numerical pressure jump  $\Delta p_h$  as

$$\epsilon = \sqrt{\frac{1}{\Delta P} \int_{\Omega} (\Delta p - \Delta p_h)^2 dx dy}, \quad (5.1)$$

Grid	$\Delta p_h$ [Pa]	$\epsilon$ [Pa]	$t_{\text{CPU}}$ [s]
70×70	1329.5	0.0306	12760.52
100×100	1307.7	0.0299	37153.31
120×120	1312.0	0.0254	64244.35
150×150	1329.7	0.0219	124831.52

Table 5.3.: Numerical convergence results for the compressible Baer-Nunziato equations with surface tension using the DOT Riemann solver. The error norms refer to the pressure jump at time  $t = 0.5$  s.

For this test problem, the curvature is computed *numerically* according to (2.40). The resulting numerical convergence rates are given in Table 5.3 at a final time of  $t_f = 0.5$ .

From the results reported in Table 5.3 we find that the error norms decrease with an increasing number of grid cells, while the CPU time increases for finer meshes. In Figure 5.7, we also show the final pressure field of the mixture pressure  $p = \alpha_1 p_1 + \alpha_2 p_2$ , and its dependence on the mesh size in 2D using the path-conservative finite volume scheme based on the DOT solver (3.11).

If we repeat this test problem on a  $100 \times 100$  grid and now prescribe the *exact* value of the curvature  $\kappa = 1/R$  together with the exact pressure jump  $\Delta p = \sigma \kappa$ , the steady bubble in equilibrium is preserved up to *machine precision*. In double precision arithmetics, the error norms obtained for the absolute values of the velocities in  $L_\infty$  norm after 17216 time steps at the final time  $t = 0.5$  are  $\max(|\mathbf{u}_1|) = 2.0120619 \cdot 10^{-12}$  and  $\max(|\mathbf{u}_2|) = 6.7706877 \cdot 10^{-12}$ , respectively. This confirms that our path-conservative finite volume method is also *well-balanced* for the steady bubble in equilibrium in the two-dimensional case, if the *exact values* of the curvature and the pressure jump are prescribed. We thus can conclude, that the errors reported in Table 5.3 are due to the numerical errors produced in the discrete curvature computation (2.40) and not due to the path-conservative scheme (2.33) used to discretize the governing PDE system.

### 5.2.1.2. Droplet oscillation

Here, we consider the oscillation of an initially elliptic droplet due to the effect of surface tension at the interface of two immiscible fluids. We are particularly interested in the temporal evolution of the total kinetic energy and the oscillation period. For the initially ellipsoidal shape of the droplet that separates phase 1 from phase 2, we use the same



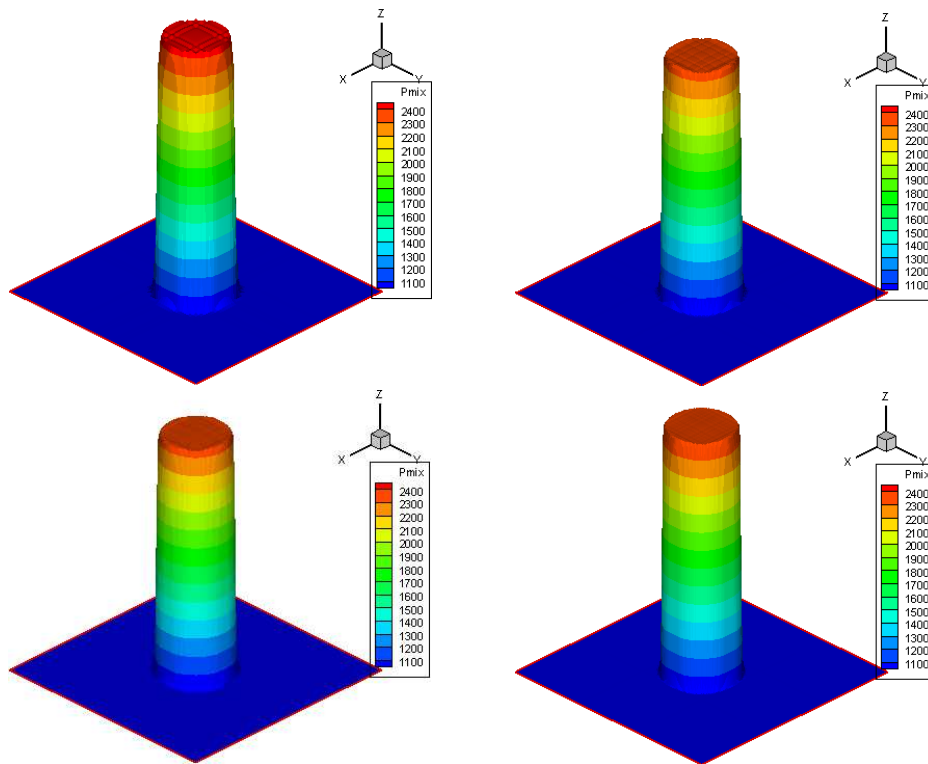


Figure 5.7.: 3D surface plot for the mixture pressure obtained with a path-conservative Osher-type scheme corresponding to four different mesh sizes listed in Table 5.3.

equation given in [121]:

$$\frac{(x - 0.5)^2}{0.2^2} + \frac{(y - 0.5)^2}{0.12^2} = 1.$$

We use the square domain  $\Omega = [0, 1] \times [0, 1]$ . The liquid phase inside the droplet has the thermodynamic parameters  $\gamma_1 = 2.4$ ,  $\pi_1 = 10^7$  and density  $\rho_1 = 100$ . The gas phase is the surrounding air, which is an ideal gas. The properties of the ideal gas are  $\gamma_2 = 1.4$ ,  $\pi_2 = 0$  and  $\rho_2 = 1$ . The surface tension coefficient is  $\sigma = 342$ . The number of cells for our simulation is  $111 \times 111$ , and we apply no slip wall boundary conditions on all domain boundaries. The volume fraction is  $\alpha_1 = 0.99$  inside the droplet and  $\alpha_1 = 0.01$  outside. The initial velocities are  $\mathbf{u}_1 = \mathbf{u}_2 = 0$  and the initial pressure is set to  $p_1 = p_2 = 1000$  everywhere in the computational domain. Due to the surface tension effect, the droplet starts to oscillate from the initially ellipsoidal shape to a circular intermediate shape and back to an ellipsoidal one, see the sketch in Figure 5.8. The oscillation is due to the transfer between the potential energy of the interface and the kinetic energy of the fluid. As suggested in [121], the surface tension coefficient is set to zero for the first three time steps, and a diffusive Riemann solver (Rusanov scheme) is used. This is in order to spread the interface over several cells in order to allow an accurate computation of the curvature  $\kappa$  according to (2.40). We call this preprocessing technique initial artificial interface diffusion method (ADNM). The difference in the numerical results with and without the use of this initial ADNM treatment are reported in Figure 5.9.

According to the Rayleigh formula [123] and [74], in the two dimensional case the oscillation frequency of the droplet is given as:

$$\omega^2 = (o^3 - o) \frac{\sigma}{(\rho_1 + \rho_2) R^3}, \quad (5.2)$$

where  $\rho_1$ , and  $\rho_2$  are the liquid and the gas density, respectively,  $o$  is the oscillation mode,  $R$  is the equilibrium drop radius and  $\sigma$  is the surface tension coefficient. The oscillation period can then be computed as

$$T = \frac{2\pi}{\omega}. \quad (5.3)$$

Here, we have  $R = 0.15825$  and  $o = 2$ , according to [121], hence the exact period is  $T = 0.0878$ . From Figure 5.8 and Figure 5.9, we can estimate the numerical period as  $T = 0.09031$ . The error is about 3% in our case, while the error was about 2% in [121] and 4% in [104] on a  $128 \times 128$  grid.

### 5.2.1.3. Dynamics of a droplet under gravity and surface tension forces

In this test we want to simulate the elongation and breakup of the falling of a water droplet under surface tension and gravity effects. The physical parameters are as follows:

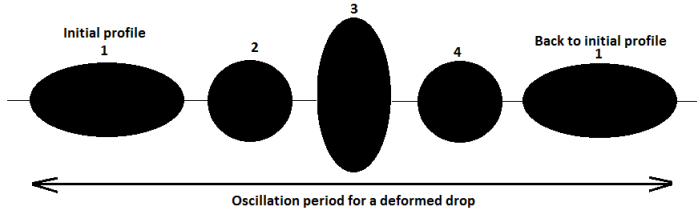


Figure 5.8.: The oscillations of the droplet in time

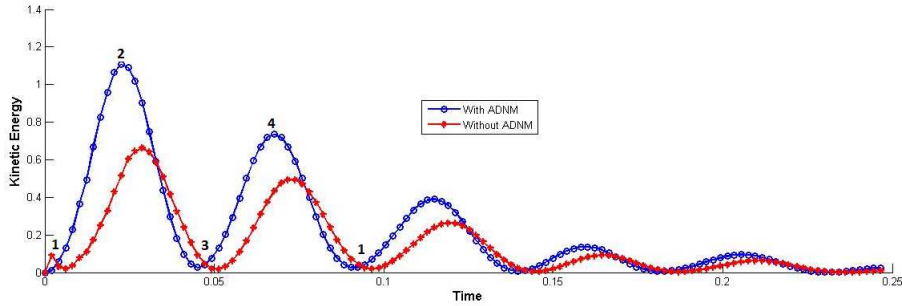


Figure 5.9.: The oscillations of the droplet in time

a drop of initial radius  $R = 0.25$  is located on the upper wall centered at  $[0.5, 1.6]$ . The computational domain is defined by  $\Omega = [0, 1] \times [0, 1.5]$ . The fluid properties are:  $\rho_1 = 10^3$ ,  $\sigma = 200$ ,  $\gamma_1 = 4.1$ ,  $\pi_1 = 5 \times 10^7$  for the phase 1 (liquid phase), and  $\rho_2 = 1.17$ ,  $\gamma_2 = 1.4$ ,  $\pi_2 = 0$  for the phase 2 (the surrounding ideal gas). The gravity acting downwards is taken to be  $\mathbf{g} = (0, -25)$ . Viscosity and relaxation effects are neglected. The initial velocities are zero and the pressure obeys the Young-Laplace equation. The contact angle at the wall boundary is taken equal to  $25^\circ$ . The number of cells for our simulation is  $205 \times 305$ . The numerical results are computed in 125000 time steps up to  $t = 0.6$  in physical time. At the initial time, the water droplet is hung up on the upper wall as a spherical cap, and then due to the gravity, the water droplet moves downward and quickly distorts the spherical cap to an elongated sack-like structure. But a part of the cap still adheres to the upper wall due to the surface tension effect. A filament is produced that connects the cap on the upper wall with the downward moving droplet. Our numerical results for the process of such a falling droplet under gravity and surface tension effects are shown in Figure 5.10. They are in good qualitative agreement with numerical and experimental results provided in [121].

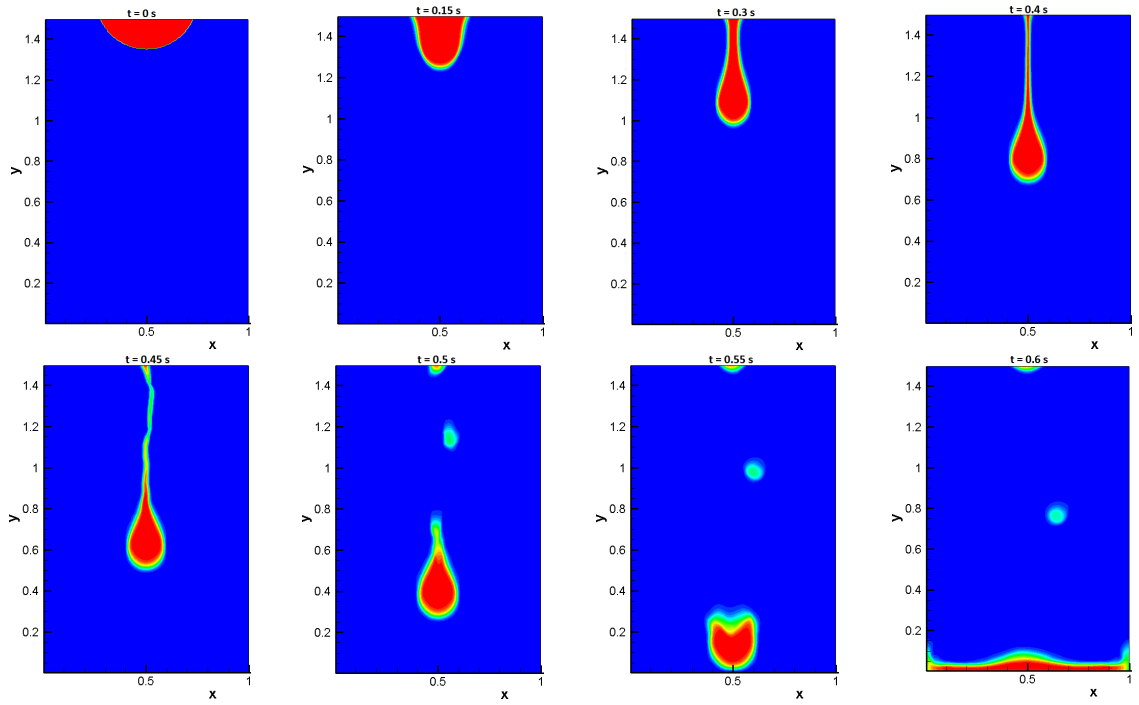


Figure 5.10.: Drop falling under gravity effect and break-up.

#### 5.2.1.4. A rising gas bubble under buoyancy forces

In this section, we simulate a rising gas bubble due to the buoyancy force effect. The shape of the bubble is dependent on viscosity, surface tension and gravity effects. The computational domain is the rectangle  $\Omega = [0, 1] \times [0, 2]$ , the initial position of the bubble is  $[0.5, 0.5]$ , the initial diameter is 0.5. The properties of the phase 1 are:  $\rho_1 = 1000$ ,  $\mu_1 = 1.14 \times 10^{-3}$ ,  $\gamma_1 = 2.1$  and  $\pi_1 = 2 \times 10^7$ . The properties of the phase 2 are:  $\rho_2 = 1$ ,  $\mu_2 = 1.78 \times 10^{-5}$ ,  $\gamma_2 = 1.4$  and  $\pi_2 = 0$ . The surface tension coefficient is  $\sigma = 0.0728$ . The gravity acceleration is taken to be  $\mathbf{g} = (0, -9.8)$ . Our computations were carried out on a Cartesian mesh of size  $100 \times 200$ . In this test, the surface tension is very small, so the rising bubble is unstable due to the density and velocity differences, respectively. In Figure 5.11, the numerical results are shown at six different times of  $t = 0$ ,  $t = 0.2$ ,  $t = 0.3$ ,  $t = 0.4$ ,  $t = 0.5$  and  $t = 0.65$ . The solutions are similar compared with the results presented in [88, 97].

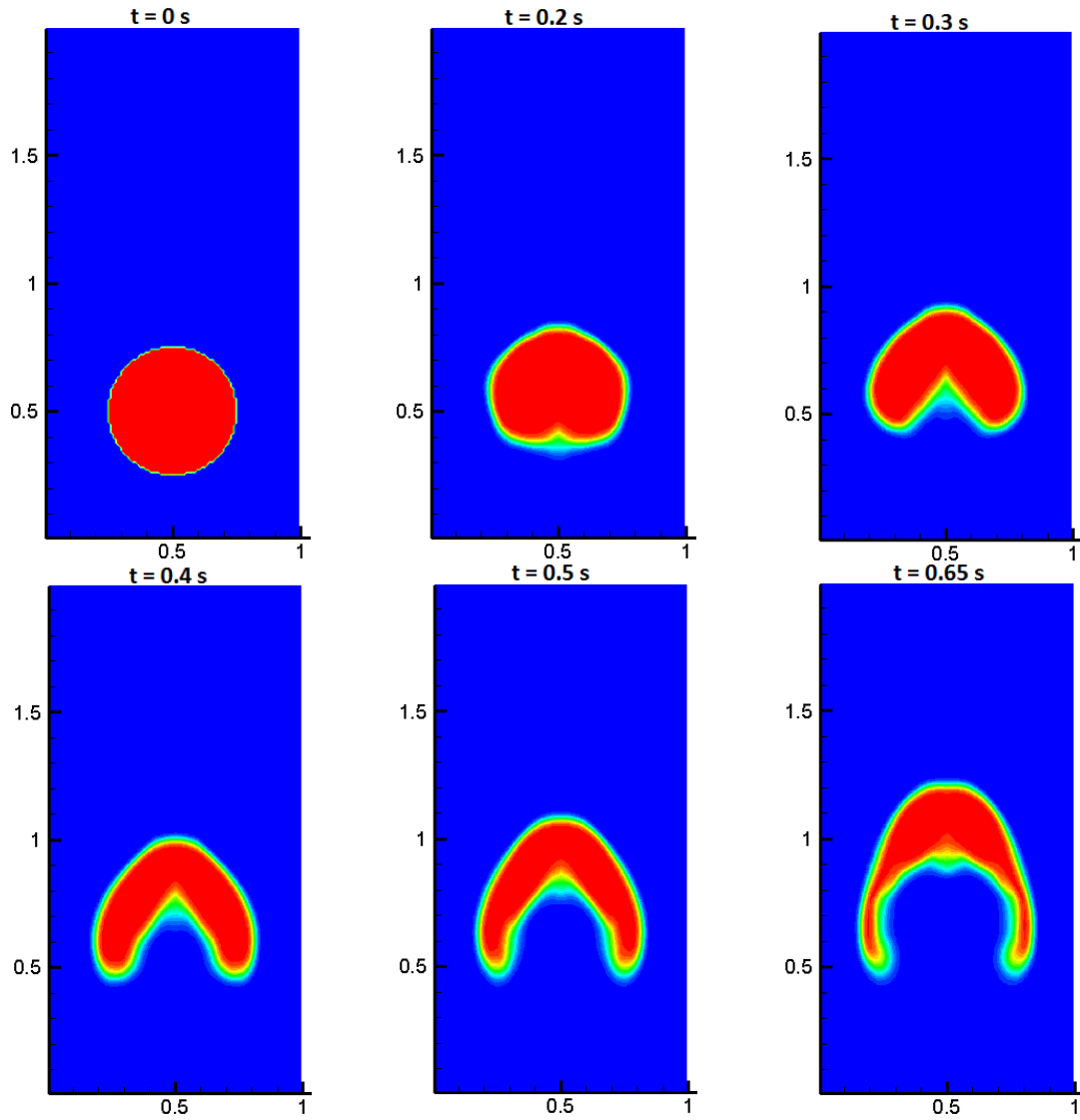


Figure 5.11.: Interface positions in form of volume fraction contour ( $\alpha_2$ ) for a rising 2D bubble. Mesh size is  $100 \times 200$ .

### 5.2.1.5. Head-on Collision of Binary Drops

In this test case, we give an example of the simulation of the head-on collision of two equally sized liquid drops moving at a relative velocity of 10. The computational domain is the square  $\Omega = [0, 1] \times [0, 1]$  with four periodic boundary conditions, the initial diameter of two equally sized droplets is 0.2, and they are initially located at points (0.3, 0.5) and (0.7, 0.5), respectively. The properties of the drops are:  $\rho_1 = 1000$ ,  $\mu_1 = 1.14 \times 10^{-3}$ ,  $\gamma_1 = 4.1$  and  $\pi_1 = 5 \times 10^7$ . The properties of the gas are:  $\rho_2 = 1.17$ ,  $\mu_2 = 1.8 \times 10^{-5}$ ,  $\gamma_2 = 1.4$  and  $\pi_2 = 0$ . The surface tension coefficient is  $\sigma = 831$ . The computational domain is discretized on a Cartesian mesh of size  $200 \times 200$ . The initial pressure is  $P = 10^5$  for both drops and gas, and gravity acceleration is neglected. The numerical results were obtained by using a Osher-type scheme (DOT) at eight different times in Figure 5.12. At the initial time  $t = 0$ , the drops are located in two-diameter distance apart from their centers, in the future time two approaching drops will be merged together and obtained a head-on collision. The liquid keeps moving forward in lateral direction away to get further expansion. Depending on the initial velocity and the surface tension coefficient, the expansion may lead to a liquid filament formation and eventually breaking up into discrete drops. Finally, after few cycles of oscillations between the elliptical and the circular shape, a circular shape of the liquid will be established.

### 5.2.2. Path-conservative HLLEM-type scheme.

As shown above 5.2.1, the path-conservative finite volume method is a well-balanced scheme for the steady bubble in equilibrium in the two-dimensional case. In this section, we just show two numerical tests of the path-conservative HLLEM-type scheme as follows: In the first test, we validate the numerical convergence with surface tension when a static droplet is in equilibrium. In the second test, the dynamics of a droplet with gravity and surface tension effects are considered. The CFL number is taken as 0.45 for all tests, and a constant factor  $\varphi = 1$  is taken in the HLLEM scheme, hence the flattener is not used in the following simulations.

#### 5.2.2.1. Numerical Convergence Results

To validate the new HLLEM-type scheme, a stationary droplet without gravity is simulated, in order to test the static performance and evaluate the pressure jump across the interface of two immiscible fluids. The initial conditions are chosen like in section 5.2.1.1. The resulting numerical convergence rates are given in Table 5.4 at a final time of  $t_f =$

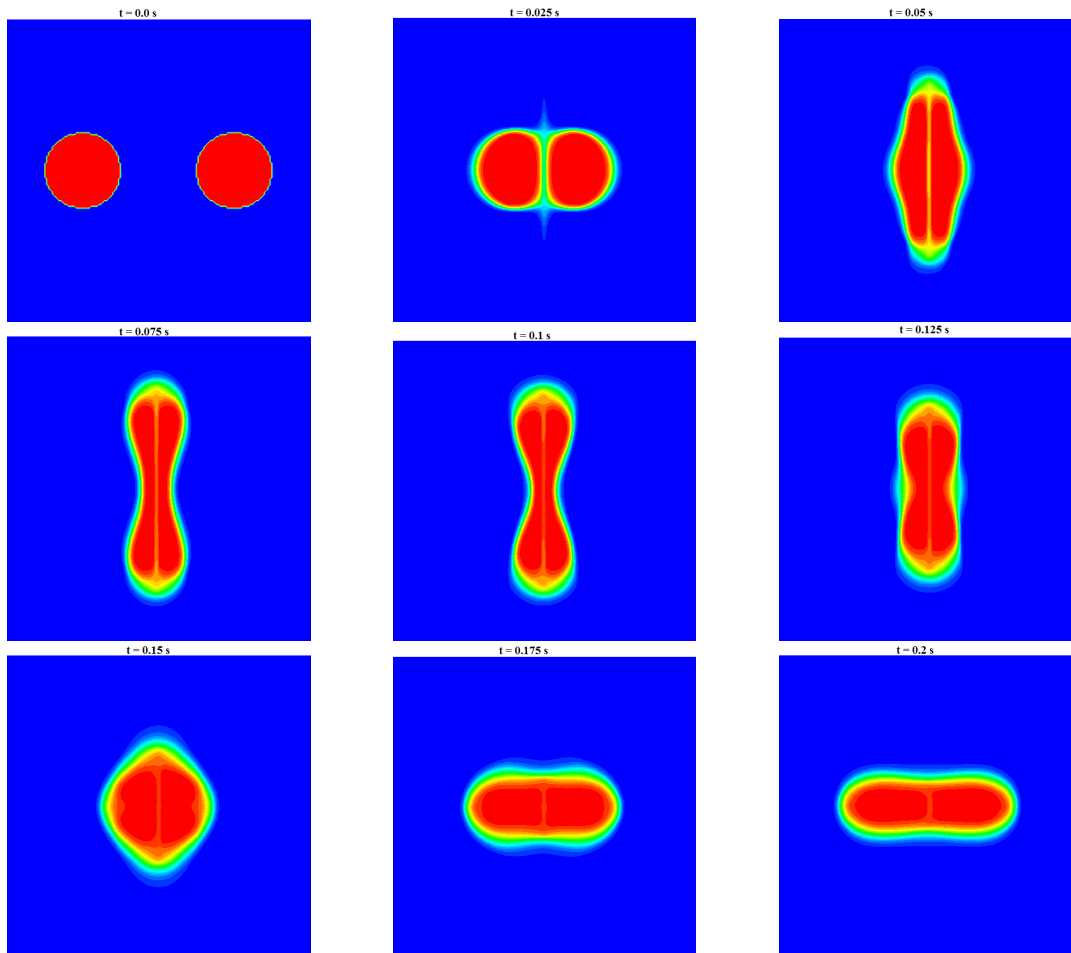


Figure 5.12.: Transient flow visualization of a 2D binary drop collision. Mesh size is  $200 \times 200$ .

Grid	Rusanov		HLLEM		Osher	
	$\epsilon$	$t_{cpu}$	$\epsilon$	$t_{cpu}$	$\epsilon$	$t_{cpu}$
70×70	0.3056	1874.62	0.0335	7473.49	0.0306	12760.52
100×100	0.4479	5389.30	0.0195	21696.72	0.0299	37153.31
120×120	0.4456	9580.51	0.0157	37048.27	0.0254	64244.35
150×150	0.4218	18363.42	0.0125	72945.30	0.0219	124831.52

Table 5.4.: Numerical convergence results for the compressible Baer-Nunziato equations with surface tension using three different numerical schemes. The error norms refer to the pressure jump at time  $t = 0.5$  s.

0.5. The results obtained with the new HLLEM-type Riemann solver are compared with the Rusanov- and the Osher-type scheme. In Figure 5.13 we show the pressure field and its dependence on the mesh size in 2D using the HLLEM-type scheme. From the CPU times listed in Table 5.4, the Rusanov-type scheme is the most efficient one, but it is also the least accurate scheme. Otherwise, the HLLEM-type scheme produces the lowest errors and is computationally still very efficient. The last one, the Osher-type method presented in [65, 64], is the most expensive scheme.

### 5.2.2.2. Dynamics of a droplet under gravity and surface tension forces

In this test we want to simulate the elongation and breakup of the falling of a water droplet under surface tension and gravity effects. The initial conditions are chosen like in section 5.2.1.3. The process of a falling droplet under gravity and the surface tension effects are shown in Figure 5.14 and is in good agreement with numerical and experimental results provided in section 5.2.1.3 and [121, 21].

## 5.2.3. TVD finite volume schemes on a two dimensional unstructured grid

### 5.2.3.1. Numerical convergence studies

In this section a numerical convergence study for the compressible BN model is performed in two space dimensions. The test problem is similar to those presented in [49, 50, 62, 64]. In [49] the authors have derived for the BN model the equivalent of the smooth vortex originally designed for Euler equations. This vortex is stationary and axisymmetric with



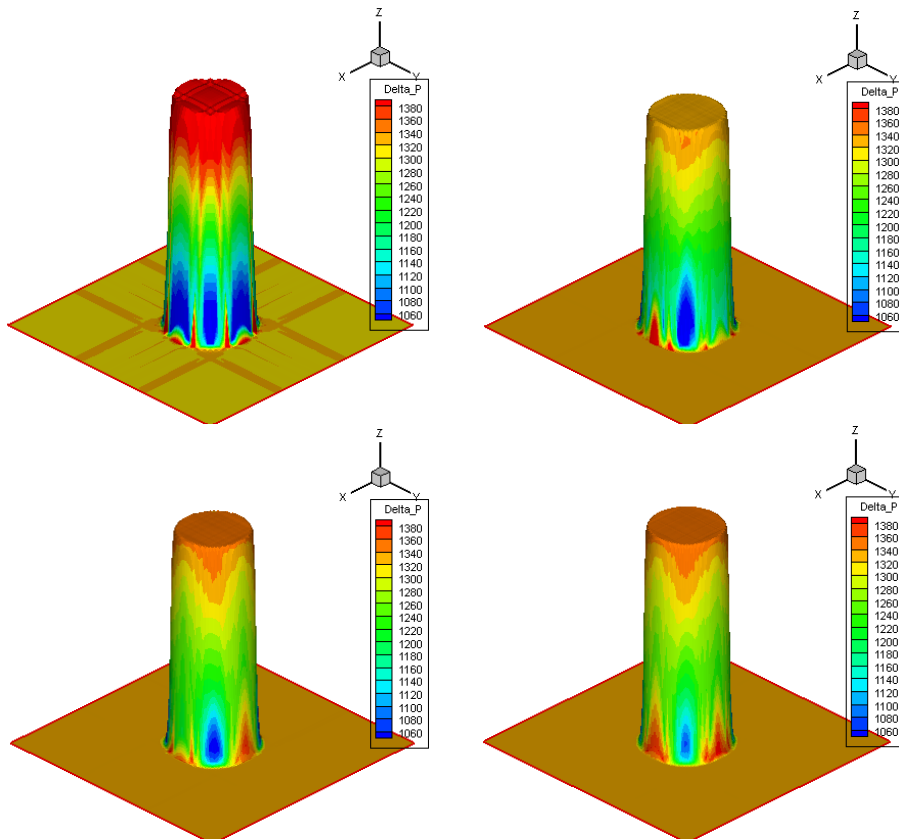


Figure 5.13.: 3D surface plot for the pressure obtained with the HLLEM-type Riemann solver corresponding to four different mesh sizes listed in Table 5.4.

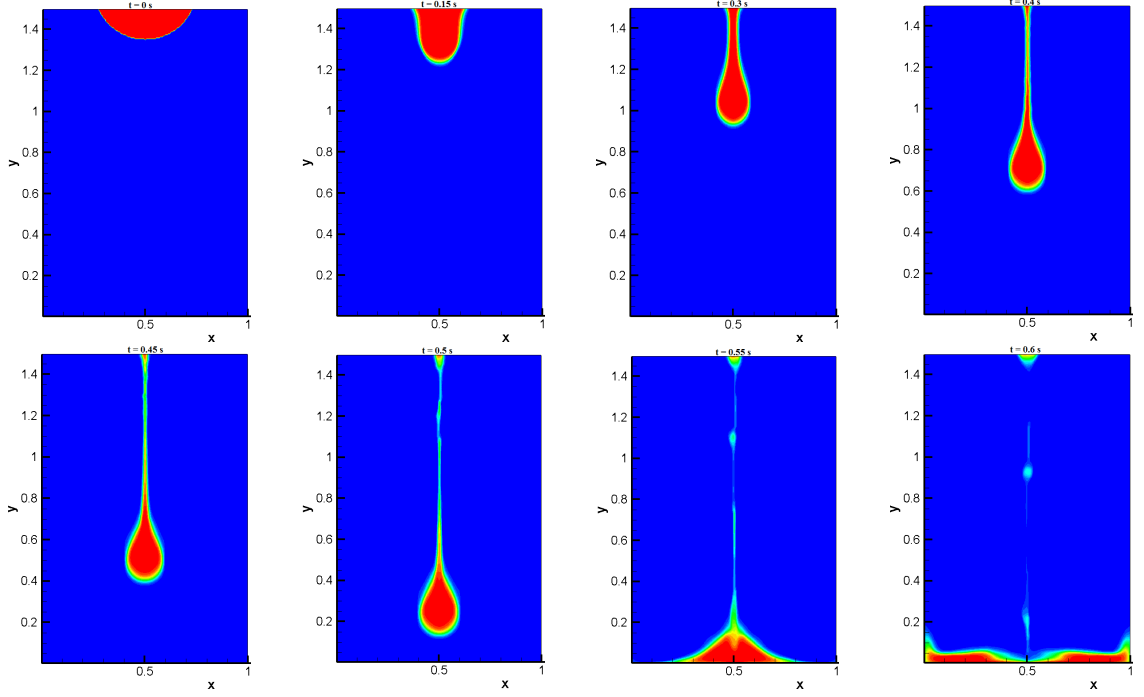


Figure 5.14.: Drop falling under gravity effect and break-up.

no motion in the radial direction. Consequently continuity and energy equations are fulfilled by construction. Following [49, 50, 62, 64] we choose

$$p_k = p_{k0} \left( 1 - \frac{1}{4} e^{(1 - r^2/s_k^2)} \right), \quad (k = 1, 2), \quad (5.4)$$

$$\phi_1 = \frac{1}{3} + \frac{1}{2\sqrt{2\pi}} e^{-r^2/2}, \quad (5.5)$$

the momentum equations can be solved to provide the velocity field as

$$u_1^\theta = \frac{1}{2s_1 D} \sqrt{rD \left[ p_{10} \left( 4\sqrt{2\pi} F_1 + 6H_1 - 12G s_1^2 + 3H_1 s_1^2 \right) + 3p_{20} s_1^2 (4G - H_2) \right]}, \quad (5.6)$$

$$u_2^\theta = \frac{r\sqrt{2}}{2\rho_2 s_2} \sqrt{\rho_2 p_{20} F_2}, \quad (5.7)$$

where

$$H_k = e^{-\frac{2r^2 + r^2 s_k^2 - 2s_k^2}{2s_k^2}}, \quad F_k = e^{-\frac{(r - s_k)(r + s_k)}{s_k^2}}, \quad (k = 1, 2), \quad (5.8)$$

and

$$G = e^{-r^2/2}, \quad D = \rho_1 \left( 2\sqrt{2\pi} + 3G \right). \quad (5.9)$$

The vortex is further boosted along the diagonal of the computational domain through a Galilean transformation of the velocity, with components  $\bar{u} = \bar{v} = 2$ . In this work we

Glocal time stepping (GTS)					
Grid	$L^1$ error	$L^2$ error	$L^1$ order	$L^2$ order	CPU time
24×24	3.4113E-02	7.557E-03	—	—	4.20E+01
32×32	1.5411E-02	4.250E-03	2.0	2.0	7.28E+01
64×64	2.6149E-03	1.020E-03	2.6	2.1	4.12E+02
128×128	9.4785E-04	3.530E-04	1.5	1.5	2.17E+03
Local time stepping (LTS)					
Grid	$L^1$ error	$L^2$ error	$L^1$ order	$L^2$ order	CPU time
24×24	3.4113E-02	7.557E-03	—	—	4.84E+01
32×32	1.5411E-02	4.250E-03	2.8	2.0	8.39E+01
64×64	2.6810E-03	1.050E-03	2.5	2.0	3.28E+02
128×128	9.2190E-04	3.330E-04	1.5	1.7	1.97E+03

Table 5.5.: Numerical convergence results for the compressible Baer-Nunziato equations with comparing global time stepping (GTS) with local time stepping (LTS) in terms of errors. The error norms refer to the variable  $\rho_1 u_1 \alpha_1$  at time  $t = 10.0$ .

have chosen the remaining free parameters as

$$\gamma_1 = 1.4, \quad \gamma_2 = 1.35, \quad \pi_1 = \pi_2 = 0, \quad \mu = \lambda = 0. \quad (5.10)$$

$$\rho_1 = 1, \quad \rho_2 = 2, \quad p_{10} = 1, \quad p_{20} = \frac{3}{2}, \quad s_1 = \frac{3}{2}, \quad s_2 = \frac{7}{5}. \quad (5.11)$$

In this configuration the computational domain  $\Omega = [-10; 10] \times [-10; 10]$  enjoys four periodic boundary conditions, in such a way that, the exact solution at time  $t=10$  is given again by the initial condition. The numerical convergence rates are computed for the conservative variable  $\mathbf{Q}_1 = \rho_1 u_1 \alpha_1$  at the final time  $t = 10.0$  on a sequence of successively refined meshes obtained by fixing a number of elements  $N_G$  along each direction. The numerical convergence results are showed in Table 5.5 with using both the global and local time stepping algorithm, seeing that they also obtain the second order of accuracy. In Table 5.5, we also make a comparison about CPU time for both GTS and LTS method.

### 5.2.3.2. 2D Riemann problems

In this section, we validate and compare the solutions of the the TVD finite volume schemes on a two dimensional unstructured grid with time-accurate global time stepping and local time stepping. For simple computations we use the compressible Baer-Nunziato

<b>RP</b>	$\rho_1$	$u_1$	$p_1$	$\rho_2$	$u_2$	$p_2$	$\alpha_1$	$t_e$
<b>RP1</b> [38]	$\gamma_1 = 1.4$	$\pi_1 = 0$	$\gamma_2 = 1.4$	$\pi_2 = 0$				
<b>L</b>	1.0	0.0	1.0	0.5	0.0	1.0	0.4	0.10
<b>R</b>	2.0	0.0	2.0	1.5	0.0	2.0	0.8	
<b>RP2</b> [38]	$\gamma_1 = 3.0$	$\pi_1 = 100$	$\gamma_2 = 1.4$	$\pi_2 = 0$				
<b>L</b>	800.0	0.0	500.0	1.5	0.0	2.0	0.4	0.10
<b>R</b>	1000.0	0.0	600.0	1.0	0.0	1.0	3.0	
<b>RP3</b> [38]	$\gamma_1 = 1.4$	$\pi_1 = 0$	$\gamma_2 = 1.4$	$\pi_2 = 0$				
<b>L</b>	1.0	0.9	2.5	1.0	0.0	1.0	0.9	0.10
<b>R</b>	1.0	0.0	1.0	1.2	1.0	2.0	0.2	
<b>RP4</b> [137]	$\gamma_1 = 3.0$	$\pi_1 = 3400.0$	$\gamma_2 = 1.35$	$\pi_2 = 0$				
<b>L</b>	1900.0	0.0	10.0	2.0	0.0	3.0	0.2	0.15
<b>R</b>	1950.0	0.0	1000.0	1.0	0.0	1.0	0.9	

Table 5.6.: Initial states left (L) and right (R) for the Riemann problems solved in 2D with the Baer-Nunziato model. Values for  $\gamma_i$ ,  $\pi_i$  and the final time  $t_e$  are also given.

equation without the surface tension and source term effects. The initial two-dimensional computational domain is given by  $\Omega(0) = [-0.5; 0.5] \times [-0.5; 0.5]$ , and the initial discontinuity between the left state  $\mathbf{Q}_L$  and  $\mathbf{Q}_R$  at the interface is located at  $x_0 = 0$ . The domain is paved with triangular meshes using a characteristic mesh spacing of  $h = 1/200$ , corresponding to a total number of  $N_E = 89992$  elements. The boundary conditions are similar [49, 62], these are periodic boundary conditions for the y direction and transmissive boundary conditions for all other directions. The initial conditions are given in Table 5.6. The numerical results are shown in Figure 5.15 - Figure 5.18 with using  $CFL = 0.5$ .

### 5.3. High order extension

In this section we present the results produced by our scheme on classical and also more demanding test cases. The methodology of testing rests upon numerically showing that

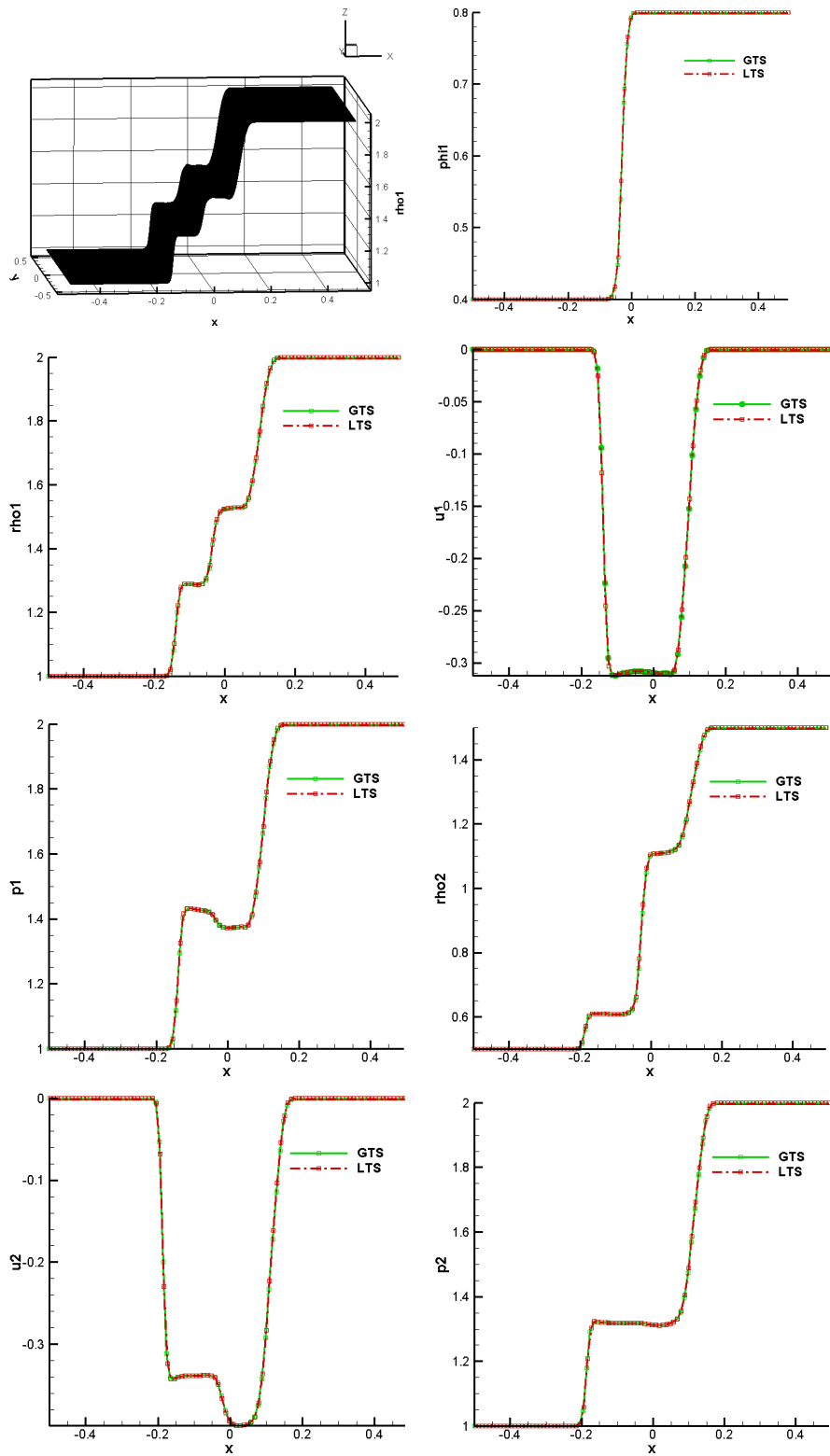


Figure 5.15.: Results for Riemann problem RP1 on the unstructured grid.

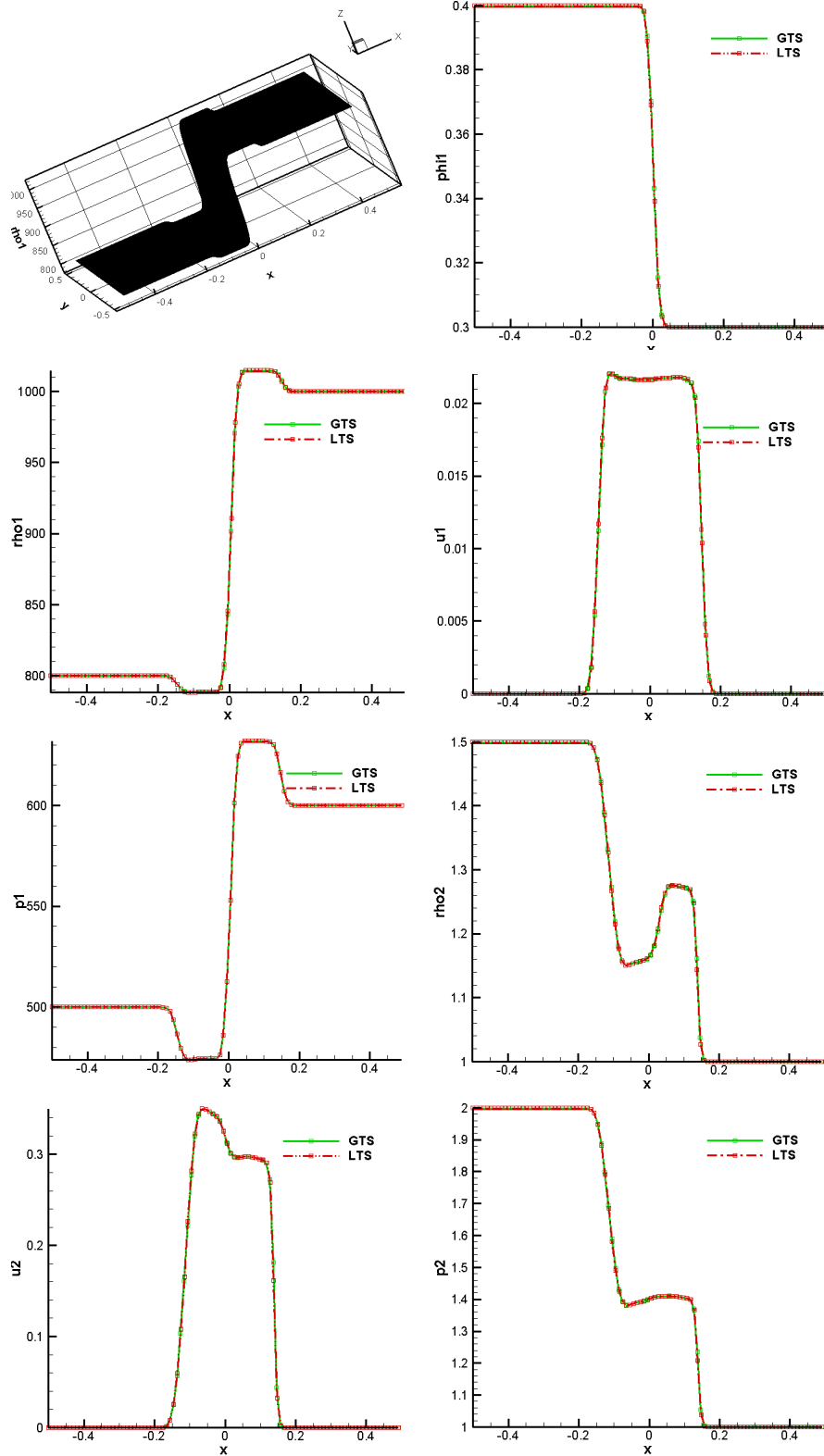


Figure 5.16.: Results for Riemann problem RP2 on the unstructured grid.

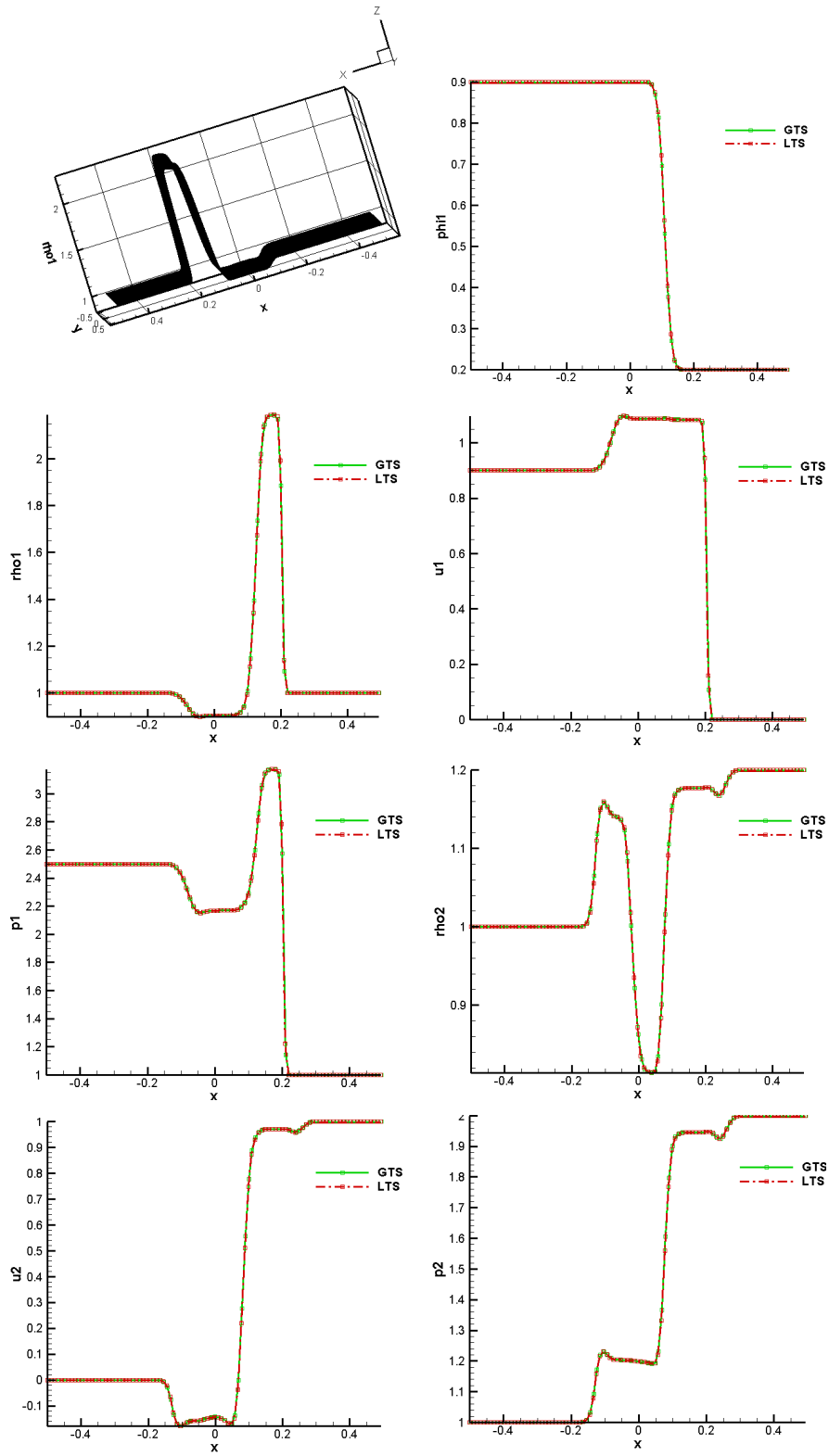


Figure 5.17.: Results for Riemann problem RP3 on the unstructured grid.

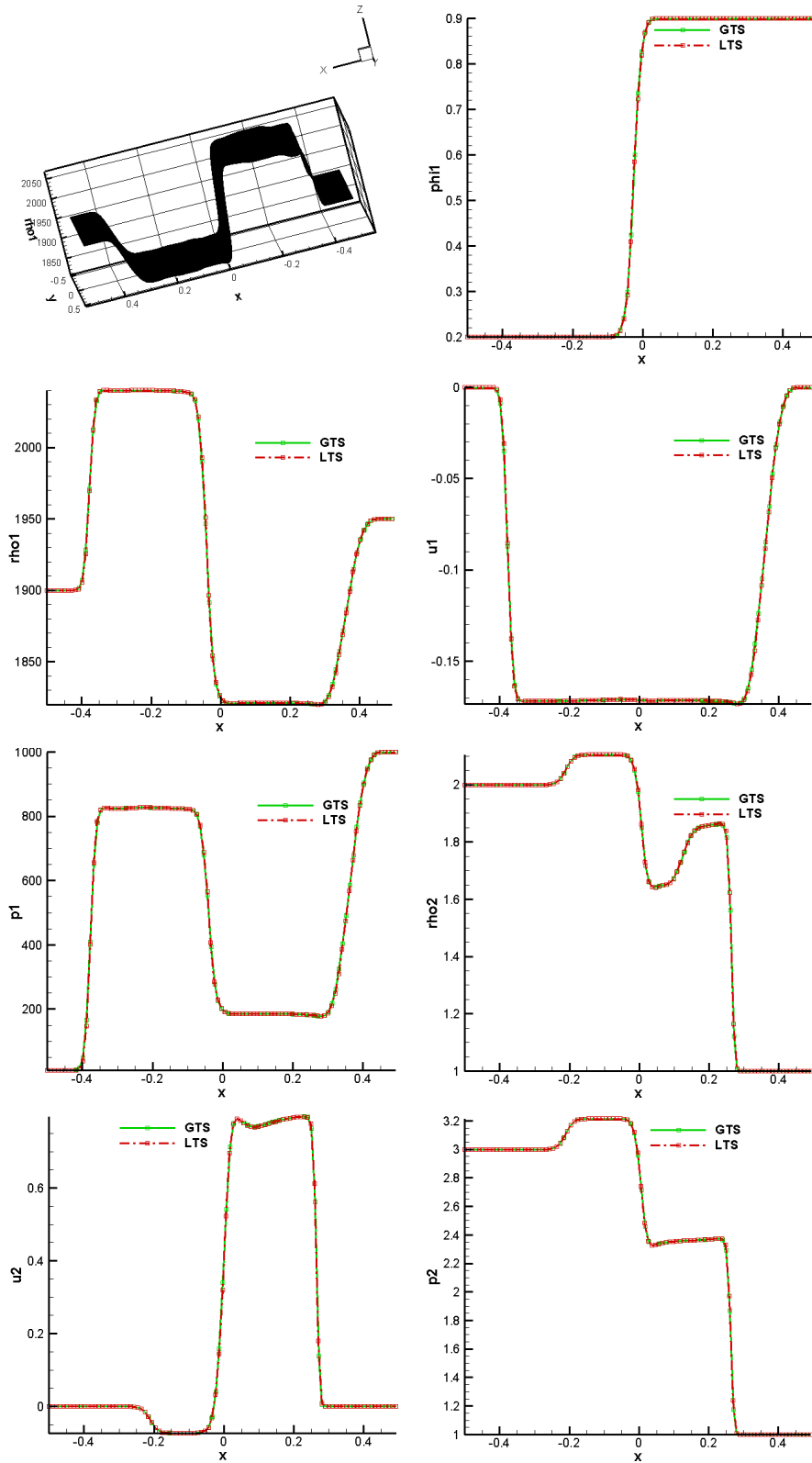


Figure 5.18.: Results for Riemann problem RP4 on the unstructured grid.



Components	Purpose	Feature/Key words
DG	Spatial accuracy	Subscale resolution, polynomial basis
ADER	Time accuracy	Local predictor, fully-discrete one-step schemes
Sub-cell limiter	Nonlinear stabilization	<i>a posteriori</i> detection criteria
Sub-cell ADER-WENO	Space/time accuracy	High order accurate subscale resolution
AMR	Efficiency, accuracy	Cell-by-cell AMR
MPI	Efficiency by parallelization	Massively parallel machines
Curvature computation	Spatial accuracy	Multiple polynomial reconstructions

Table 5.7.: Summary of the main components of our code along with associated features and key words.

- an effective optimal high order of accuracy is maintained on smooth flows with the results of a convergence study on the smooth vortex problem;
- an accurate resolution of material interfaces in multi-phase flows is obtained with the numerical evidences gathered for 1D Riemann problems for the Baer Nunziato model with exact solutions solved with our 2D method;
- Physical phenomena involving surface tension effect can be reproduced like in the Young-Laplace test case in which an oscillating ellipsoidal bubble is monitored.

For the Young-Laplace test case, like in section 5.2.1.2 we validate the capability of the method to produce physically relevant solutions to the BN model when surface tension forces are present. For these last test cases we remove the AMR capability in order to present the reaction of the scheme to the surface tension forces without polluting the solution or hiding small spurious phenomena under the AMR technology. Nonetheless the AMR methodology works well even with surface tension forces and this is illustrated with the last test case of the thesis where the whole ADER-DG AMR with sub-cell limiter machinery is employed. For all these verification test cases the full numerical machinery is employed, that is the main scheme is ADER-DG- $\mathbb{P}_M$  ( $M = 5, 6$ ) with *a posteriori* sub-cell 3rd order ADER-WENO limiter under AMR framework, see Table 5.7 where we summarize the main components of the simulation code.

### 5.3.1. Numerical convergence studies

In this section, we validate the numerical convergence studies of the ADER-DG with sub-cell limiter by using the initial conditions, mentioned in section 5.2.3.1. In this con-

figuration the computational domain  $\Omega = [-10; 10] \times [-10; 10]$  uses four periodic boundary conditions, in such a way that, given the velocity field, the exact solution of the problem coincides with the initial one after  $T = 10$ . We have tested the DG scheme with *a posteriori* sub-cell WENO limiter with optimal accuracy ranging from 3rd order up to 10th order. In Table 5.8 we have reported the results of the convergence tests. Here  $\rho_{1,0} = \rho_{2,0} = 1$  have been chosen. The convergence rates are computed with respect to an initially uniform mesh, see [16]. The value of  $N_x$  in the table indicates the resolution of the  $\ell = 0$  grid in each direction. The number of degrees of freedom is computed as  $N_x^2 \times (N+1)^2$  where  $N$  is the degree of polynomial reconstructions. As expected, even if the *a posteriori* sub-cell limiter is active, the formal order of convergence of the DG scheme is attained. Note that the number of DG cells is reduced when the order of accuracy increases, because the number of degrees of freedom for a single 2D DG- $\mathbb{P}_9$  cells already attains 100. In figure 5.19 we present the error in logscale as a function of the number of degrees of freedom for the DG schemes tested in table 5.8, from 3rd order up to 10th order of accuracy. As expected the higher accurate DG schemes are more efficient than lower ones at fixed number of degrees of freedom.

### 5.3.2. Riemann problems

We first propose to solve a set of 1D shock tube problems on space-time adaptive Cartesian meshes in 2D with our  $\mathbb{P}_5$  ADER-DG scheme with *a posteriori* sub-cell WENO3 limiter and AMR. The purpose is to test the behavior of the numerical machinery, more precisely the ability of the limiter to maintain the solution essentially non-oscillating, the capability of the AMR framework to perform with such a limiter. Exact solutions for the Riemann problems can be located in [5, 137, 38]. Unlike [50] only a set of four Riemann problems is considered, see Table 5.9. Note that numerical results obtained with high order unstructured one-step WENO finite volume schemes using centered path-conservative methods can also be found in [49, 64, 50]. Let us consider a two-dimensional computational domain  $\Omega = [-0.5; 0.5] \times [0; 1]$  discretized at the level 0 grid with a coarse grid made of  $50 \times 5$  cells. A maximum number of two refinement levels ( $\ell_{\max} = 2$ ) is chosen, together with a refinement factor of  $\tau = 2$ . The discontinuity is located initially at  $x = 0$ . Initial configurations and final simulation times are provided in Table 5.9. Transmissive (periodic) boundary conditions are imposed in  $x$  (resp.  $y$ )-direction. For all test cases a 6th order ADER-DG scheme with 3rd order WENO sub-cell limiter is used. The parameters for the refinement criterion are  $\rho_{1,0} = \rho_{2,0} = 1$  apart for RP2 where  $\rho_{1,0} = 1000$  and  $\rho_{2,0} = 1$ . Figures from 5.20 to 5.23 are organized likewise: The top left panel presents the

2D isentropic vortex problem — ADER-DG- $\mathbb{P}_N$ + WENO3 SCL									
	$N_x$	$L^1$ error	$L^2$ error	$L^\infty$ error	$L^1$ order	$L^2$ order	$L^\infty$ order	Theor.	d.o.f
DG- $\mathbb{P}_2$	25	3.1373E-01	4.6640E-02	3.0606E-02	—	—	—	3	5625
	50	2.7100E-02	7.4266E-03	7.2766E-03	3.5	2.7	2.1		22500
	75	2.7108E-03	5.1779E-04	4.2721E-04	5.7	6.6	7.0		50625
	100	1.0479E-03	2.0447E-04	2.8775E-04	3.3	3.2	1.4		90000
DG- $\mathbb{P}_3$	25	1.4063E-02	3.2029E-03	3.5704E-03	—	—	—	4	1000
	50	2.5865E-04	5.9017E-05	6.8442E-05	5.8	5.8	5.7		40000
	75	3.4212E-05	6.6685E-06	7.7136E-06	5.0	5.4	5.4		90000
	100	9.0307E-06	1.5123E-06	1.9290E-06	4.6	5.2	4.8		160000
DG- $\mathbb{P}_4$	25	5.0585E-04	9.5326E-05	1.1607E-04	—	—	—	5	15625
	50	2.0510E-05	5.4802E-06	6.1274E-06	4.6	4.1	4.2		62500
	75	2.8471E-06	8.3349E-07	1.1184E-06	4.9	4.6	4.2		140625
	100	7.6122E-07	1.9214E-07	2.7314E-07	4.6	5.1	4.9		250000
DG- $\mathbb{P}_5$	10	3.5927E-02	5.5746E-03	8.1916E-03	—	—	—	6	3600
	20	7.2292E-04	8.2757E-05	5.1444E-05	5.6	6.1	7.3		14400
	40	5.1988E-06	1.0141E-06	1.1510E-06	7.1	6.4	5.5		57600
	50	9.1567E-07	2.1590E-07	2.7839E-07	7.8	6.9	6.4		90000
DG- $\mathbb{P}_6$	10	1.9108E-02	1.4978E-03	1.5898E-03	—	—	—	7	4900
	20	5.2733E-05	7.1536E-06	7.8438E-06	8.5	7.7	7.7		19600
	25	4.8075E-06	1.0215E-06	1.3230E-06	10.7	8.7	8.0		30625
	30	1.5320E-06	3.1872E-07	4.1243E-07	6.3	6.4	6.4		44100
DG- $\mathbb{P}_7$	5	4.7139E-01	6.5843E-02	3.0455E-02	—	—	—	8	1600
	10	3.2259E-03	2.6498E-04	2.7646E-04	7.2	8.0	6.8		6400
	15	9.8869E-05	1.1159E-05	1.2436E-05	8.6	7.8	7.6		14400
	20	5.9890E-06	6.3143E-07	4.6989E-07	9.7	10.0	11.4		25600
DG- $\mathbb{P}_8$	4	1.7938E+00	2.7979E-01	1.2589E-01	—	—	—	9	1296
	8	4.5217E-03	4.1161E-04	4.2700E-04	8.6	9.4	8.2		5184
	12	1.0525E-04	9.8229E-06	7.9624E-06	9.3	9.2	9.8		11664
	16	6.6067E-06	7.5932E-07	7.8281E-07	9.6	8.9	8.1		20736
DG- $\mathbb{P}_9$	4	5.2641E-01	7.5568E-02	4.1828E-02	—	—	—	10	1600
	8	1.1603E-03	1.0307E-04	7.0770E-05	8.8	9.5	9.2		6400
	12	1.1211E-05	1.2532E-06	1.0546E-06	11.4	10.9	10.4		14400
	16	5.4982E-07	8.1196E-08	1.0714E-07	10.5	9.5	7.9		25600

Table 5.8.:  $L^1, L^2$  and  $L^\infty$  errors and convergence rates for the 2D isentropic vortex problem for the ADER-DG- $\mathbb{P}_N$  scheme with sub-cell limiter and adaptive mesh refinement for variable  $\rho_s$  at final time 10 using  $CFL = 1/2$ .

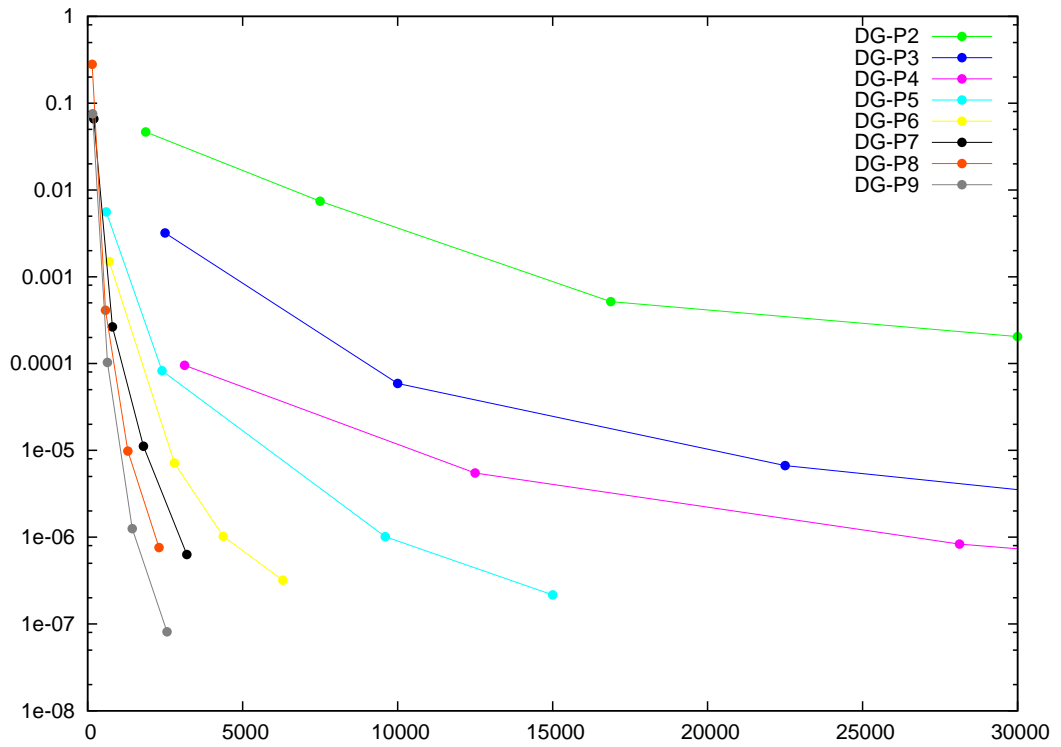


Figure 5.19.: 2D isentropic vortex problem — Error in logscale as a function of the number of degrees of freedom for the DG schemes tested in table 5.8, from 3rd order up to 10th order of accuracy.

final AMR mesh, the cell color corresponds to the activation of the limiter: blue cells are computed with 6th order pure DG on coarse cell level (*a posteriori* limiter off) whereas red cells are updated with the 3rd order WENO on sub-cell level (*a posteriori* limiter on). The other panels displays the a cut through the reconstructed numerical solution  $\mathbf{u}_h$  on 200 *equidistant* points along the  $x$ -axis for the following variables:  $\phi_s$ ,  $\rho_s$ ,  $\rho_g$ ,  $u_s$ ,  $u_g$ ,  $p_s$  and  $p_g$ . For all problems we obtain an excellent agreement between the high order AMR ADER-DG computations and the exact reference solutions from [5, 137, 38]. The discontinuities are captured on a sub-cell level. This corresponds to our initial claim that the *a posteriori* limiter is able to maintain the sub-cell resolution of the DG scheme even when it is switched on and the sub-cell WENO3 scheme is triggered. Even when the limiter is activated in inappropriate regions (RP2 for instance), we do not see any associated spurious effect on the numerical solution. As such, a sharp discontinuity can truly travel across a DG cell without being diffused on more than one or two sub-cells. In addition the AMR capability increases even more the local resolution for steep gradients leading to almost perfectly rendered discontinuities. The comparison with the results from [159] using a 3rd order ADER-WENO and AMR using 100 cells in  $x$  direction, which is more or less the same nominal accuracy than our ADER-DG- $\mathbb{P}_6$ , seems in favor of our new approach. Specifically for the resolution of the constact discontinuities for all problems, see for instance figure 5.22 the panel for  $u_s$  or  $\rho_g$  where the contact is almost exact (that is captured within one sub-cell!) compared to [159] 4th or 5th panel where the contact is spread over 5-6 visualization points, that is over one cell length. This illustrates the ability of the whole scheme to maintain the DG subscale resolution even when limiting occurs thanks, according to us, to the *a posteriori* subcell finite volume limiting paradigm. Obvisouly some spurious numerical effects are still produced (see RP4 for  $\rho_s$  variable at  $x \simeq -0.05$ ) however these mostly correspond to initial entropy deposition errors. As such they can not be avoided by our scheme, and most of the other robust schemes present in the literature. In conclusion this ADER-DG method with its add-ons produces highly accurate results and extremely sharp discontinuities.

### 5.3.3. Young-Laplace law

In this section we use the code in its operational configuration to solve problems involving active surface tension forces. The Laplace law expresses a relation at the material interface between the pressure jump and the surface tension force when equilibrium has been reached. This law states that  $\Delta p = \sigma \kappa$ , where  $\sigma$  is the surface tension coefficient and  $\kappa$  the interface curvature and  $\Delta p$  the pressure jump. In order to numerically show

<b>RP</b>	$\rho_s$	$u_s$	$p_s$	$\rho_g$	$u_g$	$p_g$	$\phi_s$	$t_e$
<b>RP1</b> [38]	$\gamma_1 = 1.4$	$\pi_1 = 0$	$\gamma_2 = 1.4$	$\pi_2 = 0$				
<b>L</b>	1.0	0.0	1.0	0.5	0.0	1.0	0.4	0.1
<b>R</b>	2.0	0.0	2.0	1.5	0.0	2.0	0.8	
<b>RP2</b> [38]	$\gamma_1 = 3.0$	$\pi_1 = 100$	$\gamma_2 = 1.4$	$\pi_2 = 0$				
<b>L</b>	800.0	0.0	500.0	1.5	0.0	2.0	0.4	0.1
<b>R</b>	1000.0	0.0	600.0	1.0	0.0	1.0	3.0	
<b>RP3</b> [137]	$\gamma_1 = 1.4$	$\pi_1 = 0$	$\gamma_2 = 1.4$	$\pi_2 = 0$				
<b>L</b>	1.0	0.0	1.0	0.2	0.0	0.3	0.8	0.2
<b>R</b>	1.0	0.0	1.0	1.0	0.0	1.0	0.3	
<b>RP4</b> [5]	$\gamma_1 = 1.4$	$\pi_1 = 0$	$\gamma_2 = 1.4$	$\pi_2 = 0$				
<b>L</b>	0.2068	1.4166	0.0416	0.5806	1.5833	1.375	0.1	0.1
<b>R</b>	2.2263	0.9366	6.0	0.4890	-0.70138	0.986	0.2	

Table 5.9.: Initial states left (L) and right (R) for the Riemann problems solved in 2D. Values for  $\gamma_i$ ,  $\pi_i$  and the final time  $t_e$  are also given.

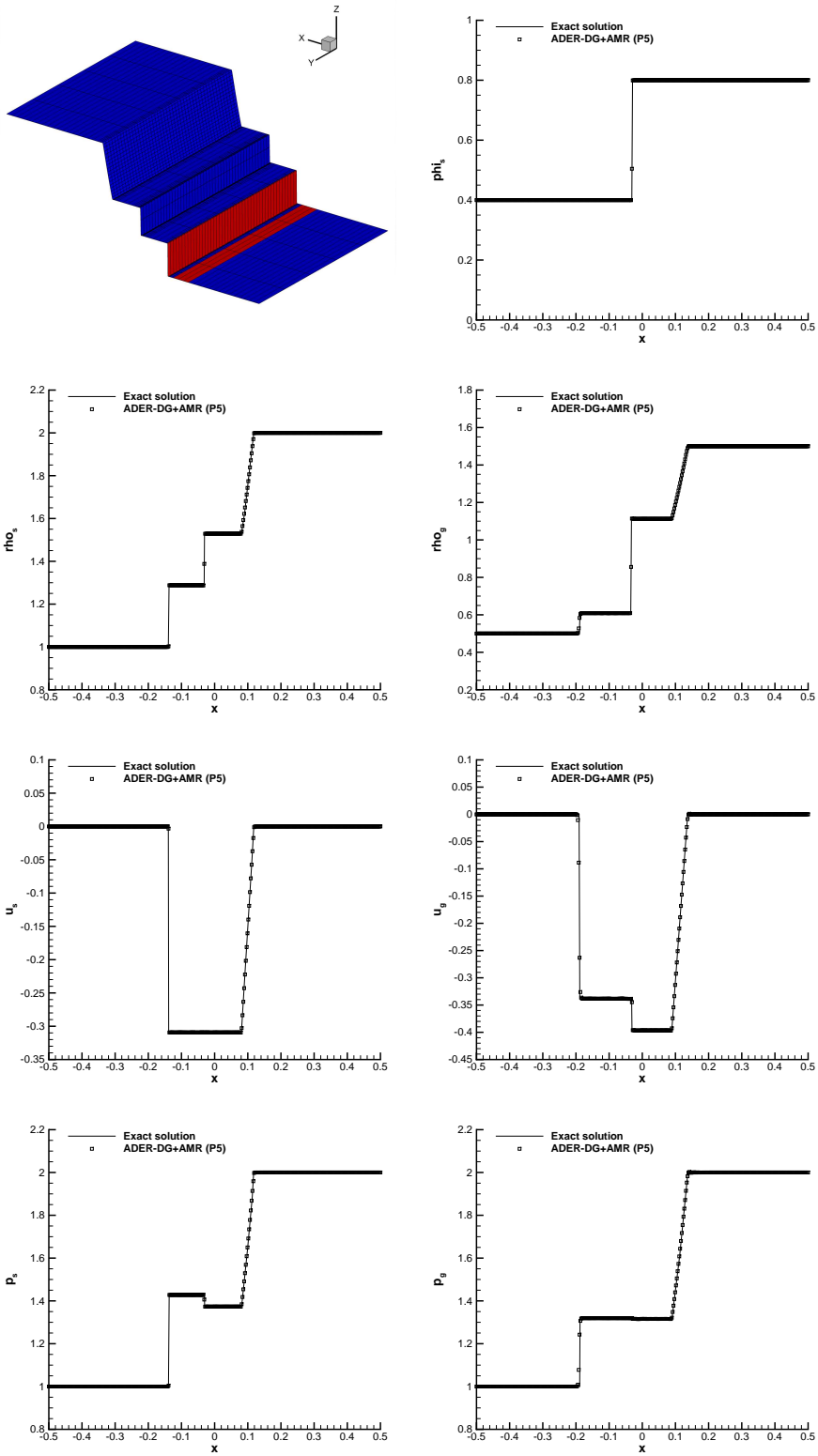


Figure 5.20.: Results for Riemann problem RP1.

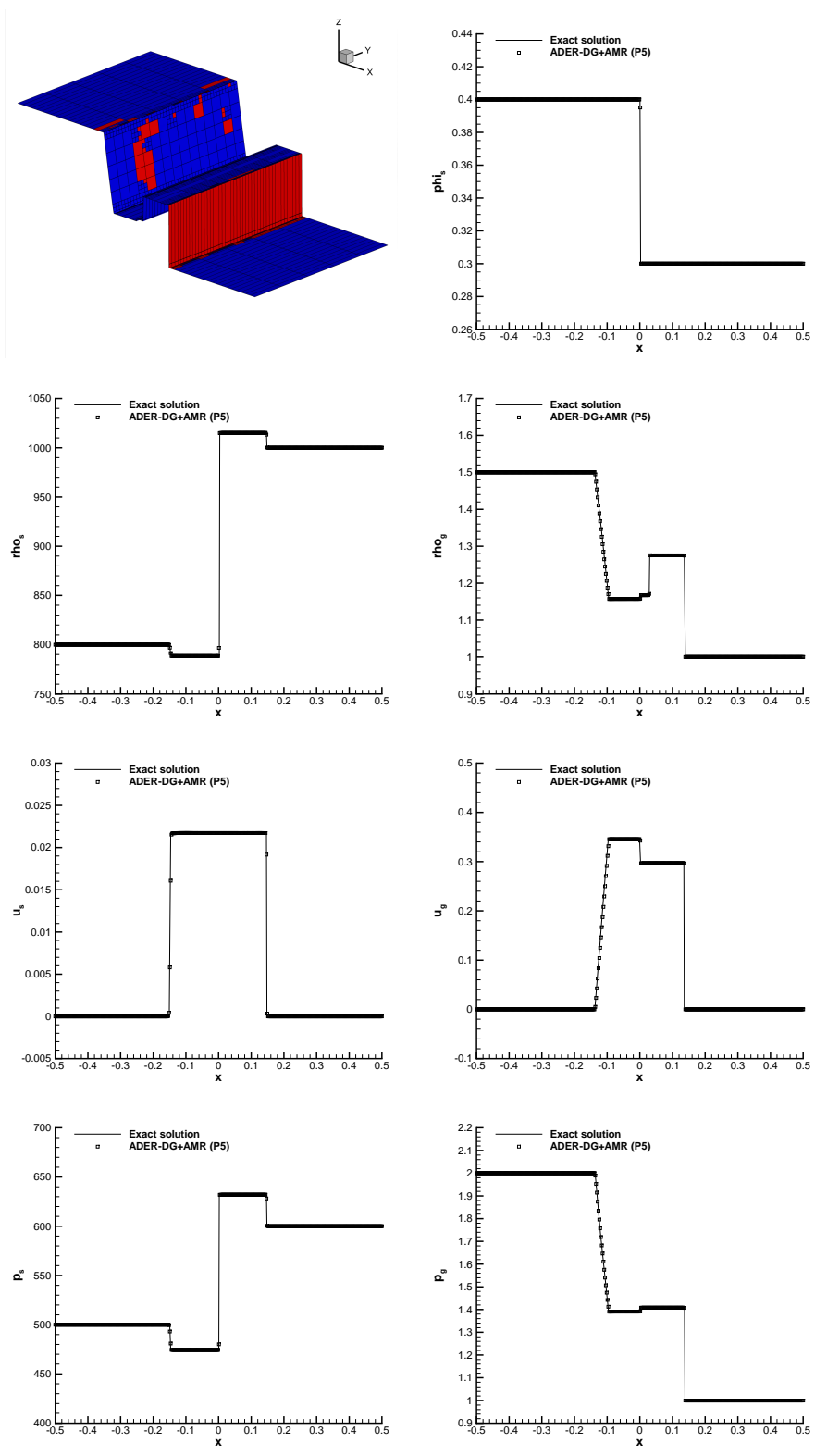


Figure 5.21.: Results for Riemann problem RP2.



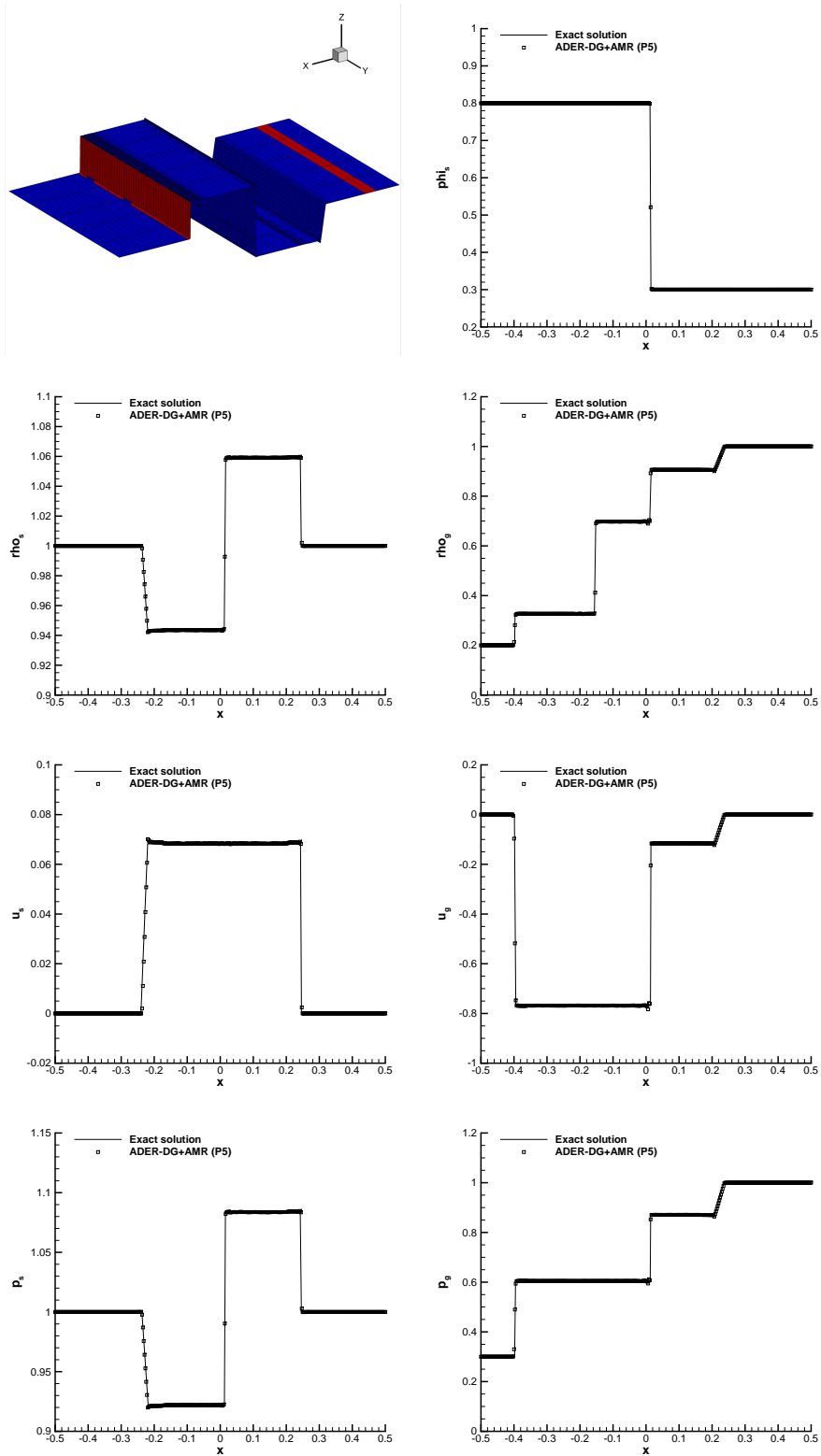


Figure 5.22.: Results for Riemann problem RP3.

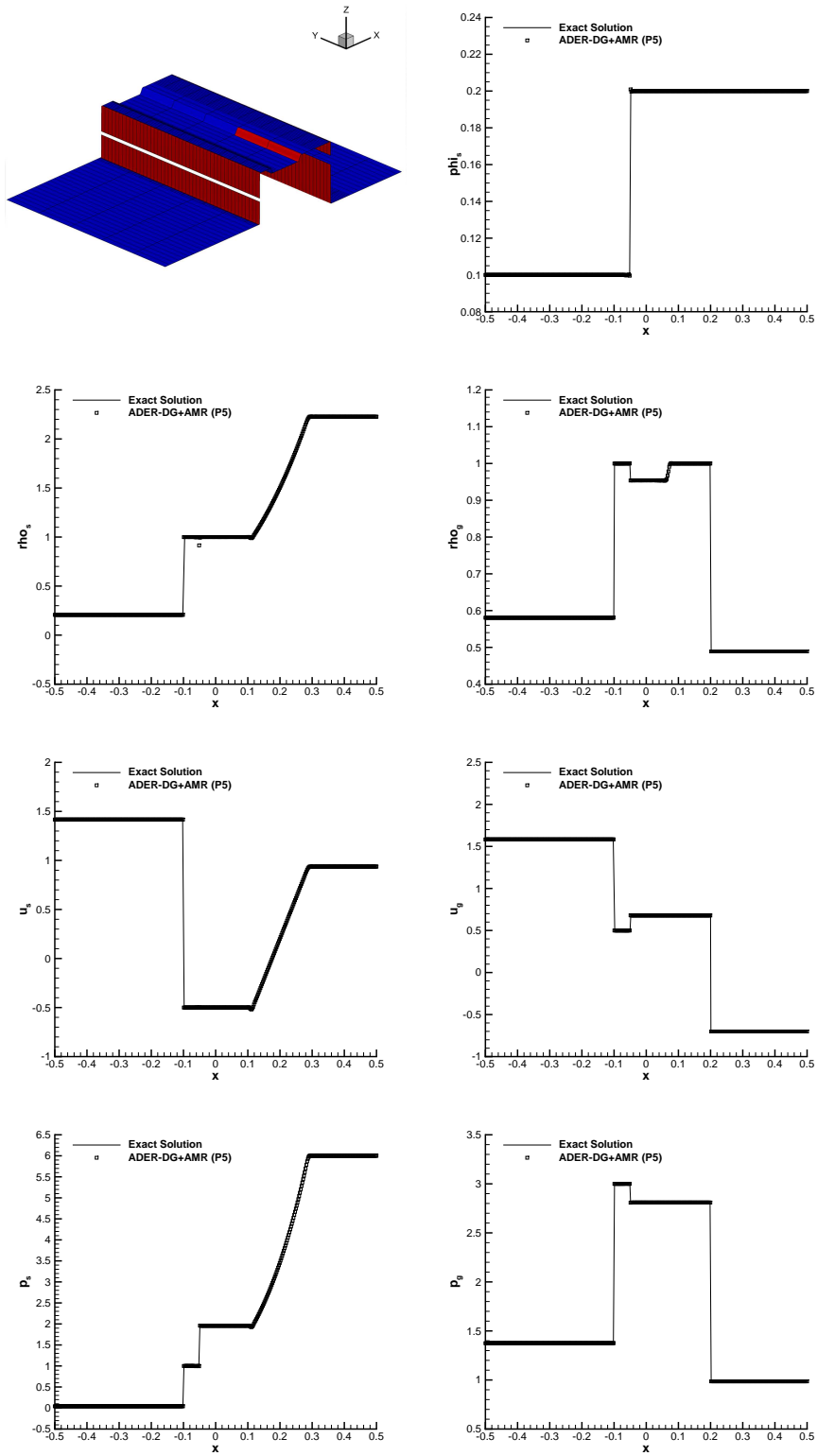


Figure 5.23.: Results for Riemann problem RP4.

that our method can recover this law, one observes the behavior of our scheme when a liquid ellipse in 2D is embedded into a gas at rest [121, 21]. The ellipse is initially at rest and kinetic energy is zero. Because of surface tension forces the ellipse starts to deform to get a circular shape. However most of the potential energy due to surface tension has been converted into kinetic one so that the circular drop cannot remain in this equilibrium state, and, enters an other cycle of deformation into an ellipse. Contrarily this ellipse is a shape for which the kinetic energy has been converted into potential energy. As a consequence, although the velocity is close to zero almost everywhere, the interface deformation is large and the bubble tends to return to a circular shape again. The drop keeps on oscillating between ellipsoidal and circular shapes due to the transfer between potential capillary energy and kinetic one, see Figure 5.24. In the ideal case of zero dissipation this oscillating motion may continue indefinitely. Due to the presence of numerical dissipation, ultimately, the ellipse ends up into a static disk configuration at convergence when  $t \rightarrow \infty$ . According to the generalized Young-Laplace law one must retrieve:  $\Delta P = \sigma/R$  where  $R$  is the radius of the disk of same surface than the initial ellipse.

The code is setup with  $M = 6$  (7th order nominally accurate DG scheme) with *a posteriori* sub-cell ADER-WENO limiter without AMR capability. The goal being here to validate the treatment of the surface tension force, on the full machinery without AMR. The computational domain is set to  $\Omega = [-1 : 1]^d$  and paved with  $N = 30^d$  cells for  $d = 2$  in 2D. The ellipse is filled with liquid surrounded by air. Both phases are modeled with the stiffened gas equation of state with  $\gamma_1 = 1.4$ ,  $\pi_1 = 20$  for liquid and  $\gamma_2 = 1.4$ ,  $\pi_2 = 0$  for air. The initial states are homogeneous and at rest initially with  $\rho_s = 10$ ,  $\rho_g = 1$ , and a jump in pressure  $p_s = 2$ ,  $p_g = 1$ . We set the relaxation coefficients to  $\lambda = 0$  and  $\beta = 50$  and  $\sigma = \frac{1}{2}$ . The bubble is an initial ellipse of the form (4.27) with  $a = 4$ ,  $b = 16$ ,  $c = 0$ . The volume fraction function is initialized as

$$\phi(r) = \frac{1}{2} \left( 1 + \frac{4}{5} \operatorname{erf}(-25(r-1)) \right), \quad \text{with} \quad r^2 = 4x^2 + 16y^2, \quad (5.12)$$

accordingly the color function  $c$  is initially given by

$$c(r) = \frac{1}{2} (1 + \operatorname{erf}(-5(r-1))). \quad (5.13)$$

The final time is fixed at  $t_{\text{final}} = 15$  s and we monitor the kinetic energy loss during this period and refer to the figures in section 4.6 for the interface and curvature pictures.

For the **2D ellipse**, as already seen in Figure 4.2 the interface and the curvature are captured at a sub-cell level thanks to our approach. Moreover the DG method on cells

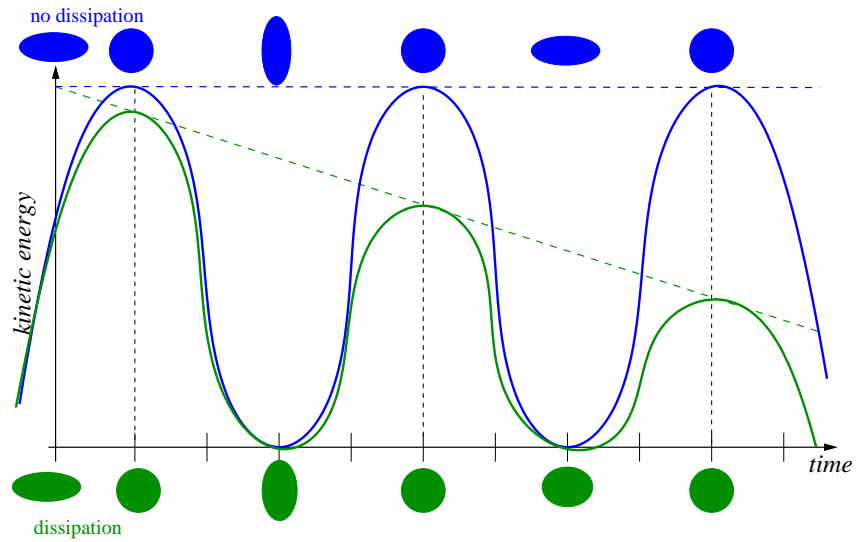


Figure 5.24.: Sketch of the expected behavior for the bubble deformation test starting from an ellipse bubble. A loss of kinetic energy is expected in the dissipative case (green) whereas in an (ideal) non dissipative case the energy remains the same (blue).

and WENO on sub-cells both being high accurate, we expect that the loss of kinetic energy depicted during the energy transfer  $potential \leftrightarrow kinetic$  remains small. We plot in Figure 5.25 the total kinetic energy in the system as a function of time for several oscillations of the bubble. We observe that the maximal kinetic energy is of the order 0.1325 and, on average, during the simulation, the kinetic energy for the bubble remains beyond  $\simeq 0.11$  which is more than 83% of the initial energy.

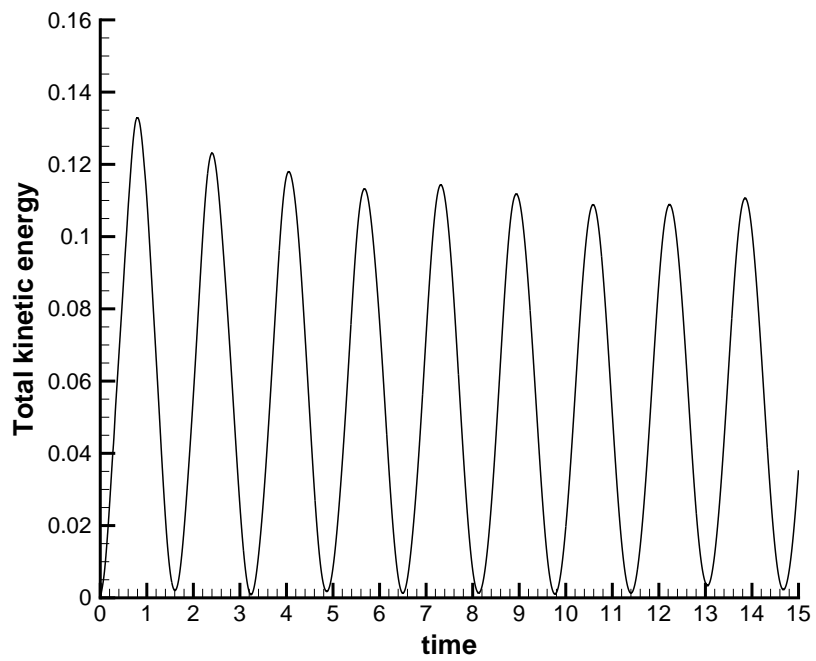


Figure 5.25.: Laplace law test case — 2D initial ellipse bubble — Kinetic energy as a function of time.

## 6. Conclusion

This thesis has proposed a wide range of numerical methods for non-conservative hyperbolic systems. From a physical, mathematical view point as well as real-life applications of compressible multi-phase flows, the Baer-Nunziato model with surface tension effects has been considered. First, we have designed a second order TVD finite volume scheme on both Cartesian and unstructured meshes. In the Cartesian grid, several numerical tests in one and two space dimensions are presented. The mathematical model under consideration includes gravity, viscosity and capillary effects and is based on the full seven-equation BaerNunziato model. The continuum surface force method (CSF) of Brackbill et al. [20] is used, which replaces the surface tension force by a volume force. The family of path-conservative finite volume schemes by using Roe-, Rusanov-, Osher- and HLLEM-type Riemann solvers has been proposed. For a steady bubble in equilibrium for which the pressure jump at the interface is given by the Young-Laplace condition, the path-conservative finite volume scheme is exactly well-balanced at the discrete level if the exact interface curvature is known. While in the case of unstructured grids, we developed our scheme with time-accurate local time stepping, which was implemented for non-conservative hyperbolic systems by using a classical TVD finite volume scheme. We have discussed and compared the differences between the GTS and LTS method, concerning accuracy and CPU time. The LTS algorithm shown lowers CPU times for the test with refined meshes. We verified the numerical convergence, both LTS and GTS algorithm have nicely obtained the second order accuracy. A time-accurate local time stepping method, the fluxes across the element interfaces are computed by adding the *memory variables* of the neighbor elements.

Second, we also have successfully developed an effective ADER-DG numerical method to solve the seven equation Baer-Nunziato model with CSF surface tension model and diffuse interface model as an illustration of the ability of the method to handle hyperbolic systems of equations, non-conservative products, stiff source terms and surface tension effects. More precisely the ADER-DG scheme is supplemented with *a posteriori* sub-cell limiter employing a 3rd order ADER-WENO scheme on sub-cell level along with AMR capability, the numerical accuracy got better than classical second order high resolutions

schemes can obtain with an AMR technique. Curvature computation is done by polynomial reconstructions of the color function (a diffused version of the volume fraction function) and of the gradient of the color function. Regarding curvature computations, the *a posteriori* sub-cell limiter was applied for computing the color function by considering the interface phase as troubled cells, and the curvature computation is handled by this strategy. The presented results show that the surface tension forces involving the curvature computation can be simulated with 6th or 7th order accurate in space and time DG scheme without large spurious oscillations. We have constructed the numerical scheme (ADER-DG), which reaches the effective optimal orders of accuracy for smooth solutions (up to 7th order of accuracy) even when the sub-cell limiter is on.

We have described the framework of path-conservative schemes of Parés and Castro [119, 25], which can naturally discretize the non-conservative products  $\mathbf{B}(\mathbf{Q}) \cdot \nabla \mathbf{Q}$  of the seven-equation system. For the Young-Laplace condition, the loss of kinetic energy in the ADER-DG scheme is less than the loss of kinetic energy in the TVD finite volume scheme.

Future research on the numerical methods presented in this thesis, i.e. both the second order TVD finite volume schemes and ADER-DG with *a posteriori* sub-cell limiting will be extended to three space dimensions. We also plan to investigate an *a posteriori* sub-cell finite volume limiter for the discontinuous Galerkin method on unstructured meshes, as well as to the more general framework of the new  $P_N P_M$  method proposed in [46].

For the curvature computation will be concerned to a cut-cell interface tracking, and the phase interface will be added by the physical phenomena, i.e. temperature and friction will be consider at the phase interface to get closer to real-world problems.

# A. Appendix

## A.1. Local time stepping

In this section we briefly describe the algorithm of the local time stepping (LTS), which is given in 2.4.4. We recall the criterion of updating of a element  $T_i$  at a time level  $t_i^n$  and the formula to compute the numerical fluxes between two elements  $T_i$  and  $T_j$  as follows:

$$t_i^n + \Delta t_i^n \leq t_j^n + \Delta t_j^n \quad \text{or} \quad t_i^{n+1} \leq t_j^{n+1}, \quad \forall j \in N_i. \quad (\text{A.1})$$

ans

$$[t_{ij}^n, t_{ij}^{n+1}] = [\max(t_i^n, t_j^n), \min(t_i^{n+1}, t_j^{n+1})], \quad \Delta t_{ij} = t_{ij}^{n+1} - t_{ij}^n, \quad \Delta t_{ij}^{n+\frac{1}{2}} = \frac{1}{2}(t_{ij}^n + t_{ij}^{n+1}) \quad (\text{A.2})$$

We consider a sequence of four time steps with three non-equidistantly spaced elements  $T^{(1)}$  to  $T^{(3)}$  as see in Figure A.1. Assume that at the beginning of the time marching, all elements start from a common time level  $t_0 = 0$  but their local time steps are different, from  $\Delta t_1$  to  $\Delta t_3$ , and  $t = t_1 > 0$  is the final time for all elements. The initial condition is used by the cell averages and a second Total Variation Diminishing (TVD) reconstruction can be done for all elements based on the cell averages  $\mathbf{Q}_j$ . After the reconstruction, all contributions are gathered, and the piecewise linear space-time reconstruction polynomial like the element-local predictor solution  $\mathbf{q}_h$  is used in each element. We stress that the stability criterion for each element an element-local time step is given by to the local CFL condition (2.53). The LTS method is a fully asynchronous algorithm, which means that it is not organized in time steps, so we denote a new definition to recognize the updated time step that is the so-called cycle. In our example, in the first cycle, the only element  $T^2$  that satisfies (A.1) and can be evolved to  $t_2^1$ ,  $t_2^1 = \Delta t^2 + t_0$ . We will discuss on the element  $T^2$ , the flux integrals are calculated by  $\Delta t^2$  at the left interface with neighbour  $T^1$ ,  $\partial T_{2,-\frac{1}{2}}$  and at the right interface with neighbour  $T^3$ ,  $\partial T_{2,+\frac{1}{2}}$ . The flux memory variables  $Q_{(1)}^M$  and  $Q_{(3)}^M$  of both neighbours are updated according to (2.58). In cycle 2, only element  $T_1$  allows the evolve condition (A.1) and can perform an update to  $t_1^1$ , see in the lower left corner of Figure A.1, like the element  $T_2$ ,  $t_1^1 = t_0 + \Delta t^1$ . From (A.2), the flux contribution



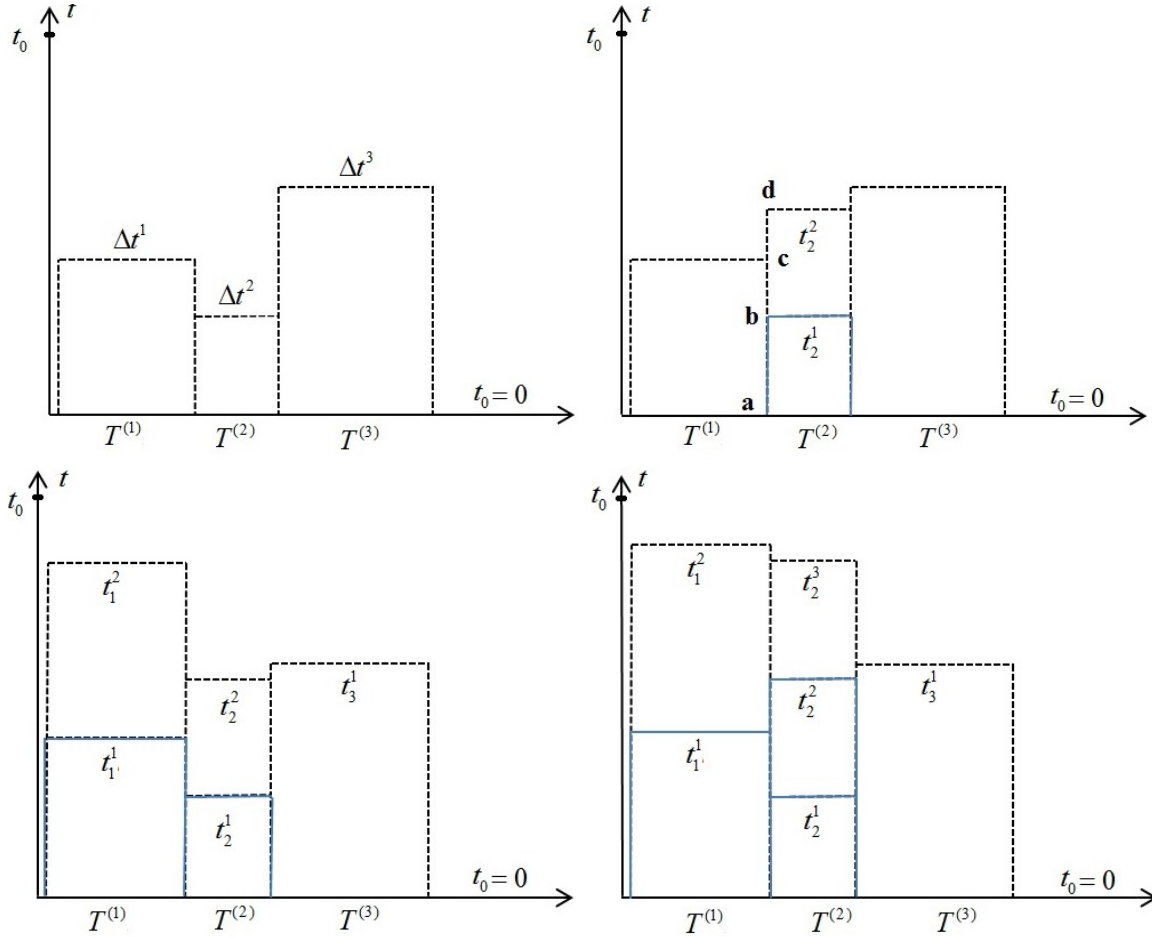


Figure A.1.: Illustration of the local time stepping algorithm with 3 different elements.

of the neighbour element  $T_2$  has already been added to the flux memory variables  $Q_{(1)}^M$  in the interval time  $[a, b]$ , so only the missing flux contribution in the interval time  $[b, c]$  will be computed to add the flux memory variables  $Q_{(1)}^M$ . In practice at the end of cycle 2  $Q_{(1)}^M$  is reset to zero. In the manner, the LTS algorithm continues doing similar ways to search the elements, which are satisfied by the criterion (A.1), and this procedure will be stopped, when all elements have reached the final  $t = t_0$ .

## A.2. Boundary conditions

In this section we will briefly review the boundary conditions, which used in the frame work of the family of the Path-conservative finite volume methods on both Cartesian and Unstructured meshes. For the numerical tests in this thesis, we consider only two types

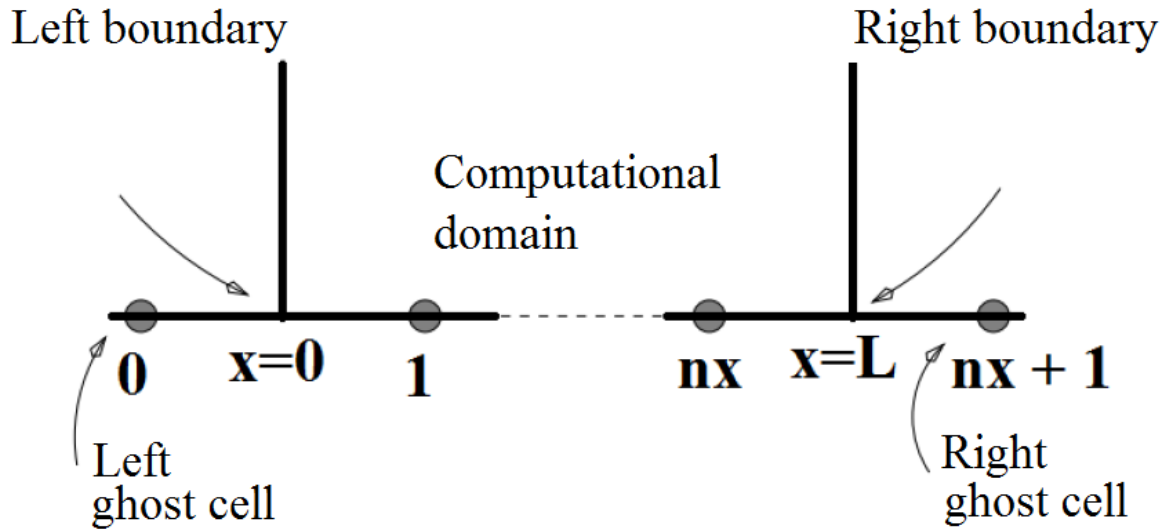


Figure A.2.: Boundary conditions. Ghost cells outside the computational domain are created.

of boundaries: reflective and transparent or transmissive.

We first describe for the Cartesian mesh as follows: for a computational domain  $[0, L]$  is discretised into  $nx$  computing cells of length  $\Delta$ , in Figure A.2. We need to set up the boundary conditions at the boundaries  $x = 0$  and  $x = L$ , we will introduce ghost cells outside the computational domain see in Figure A.2.

- Reflective Boundaries

We consider the boundary at  $x = L$ , these boundary conditions are defined as follows:  $\mathbf{Q}_{nx+1} = \mathbf{Q}_{nx}$ ,  $\mathbf{u}_{nx+1} = -\mathbf{u}_{nx}$ , here  $\mathbf{Q}$ ,  $\mathbf{u}$  are the state vector, and the velocity state, respectively. These states is described for the boundary state.

- Transmissive Boundaries

We also consider the boundary at  $x = L$ , these boundary conditions are defined as follows:  $\mathbf{Q}_{nx+1} = \mathbf{Q}_{nx}$ .

For a general coordinate, the ghost cell  $T_g$  of the element  $T_i$ , see in Figure A.3. The boundary condition will be given as follows:

- Transmissive Boundaries

In this condition the fluid across the domain boundary, the condition is computed as:  $\mathbf{Q}_g = \mathbf{Q}_i$ . Here  $\mathbf{Q}_g$  and  $\mathbf{Q}_i$  are the boundary state, and the state of element  $T_i$ , respectively.

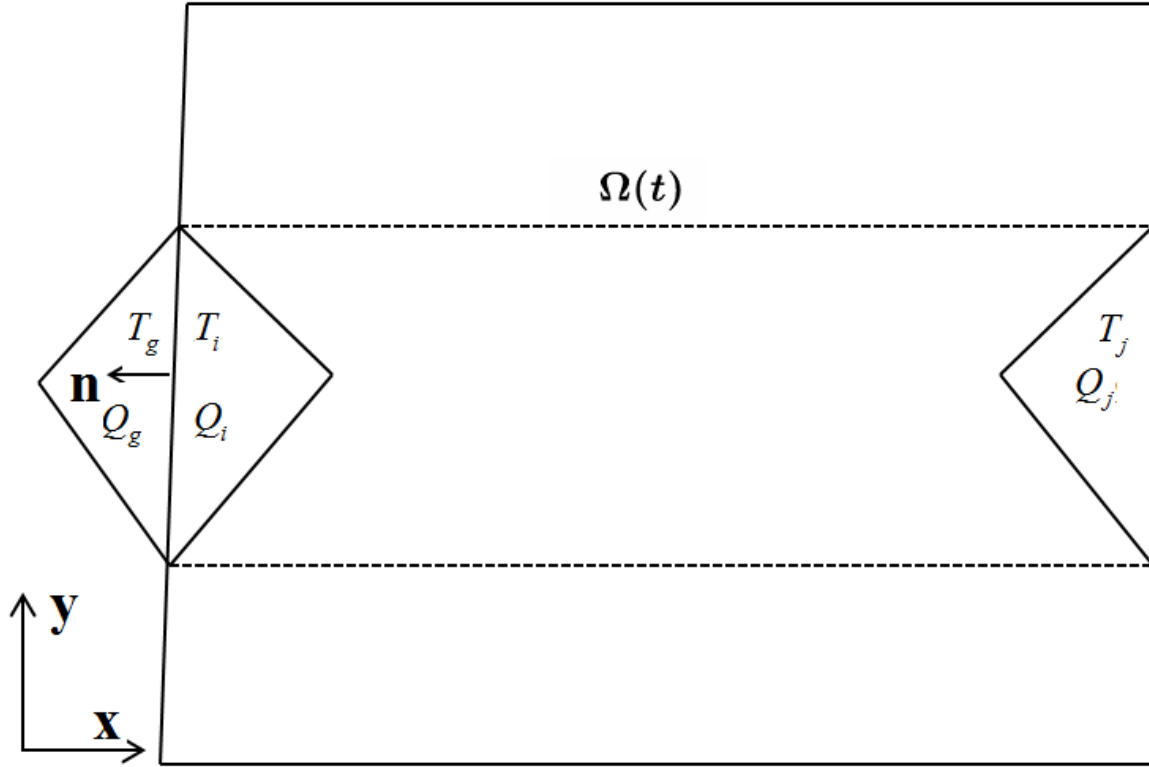


Figure A.3.: Sketch of the boundary conditions. Ghost cell outside the computational domain is taken, and it is based on a logically connected mesh.

- Reflective Boundaries

These conditions are given to treat the wall boundaries, and they are defined as follows:  $\mathbf{Q}_g = \mathbf{Q}_i$ , and the velocity vector at the boundary state is as  $\mathbf{u}_g = \mathbf{u}_i - 2(\mathbf{u}_i \cdot \mathbf{n})\mathbf{n}$ , where  $\mathbf{n}$  is outward pointing unit normal vector on the boundary edge of element  $T_i$ .

- Periodic Boundaries

We apply periodic boundary conditions to simulate an infinite computational domain, and can be given as follows:  $\mathbf{Q}_g = \mathbf{Q}_j$ , with  $T_j$  representing the logical Neumann neighbor of element  $T_i$ , see in A.3.

# Bibliography

- [1] R. Abgrall. On Essentially Non-oscillatory Schemes on Unstructured Meshes: Analysis and Implementation. *Journal of Computational Physics*, 114(1):45–58, 1994.
- [2] R. Abgrall and S. Karni. A comment on the computation of non-conservative products. *Journal of Computational Physics*, 229:2759–2763, 2010.
- [3] Dale Anderson. *Computational Fluid Dynamics: The Basic with Applications*. McGraw-Hill, 1995.
- [4] Dale Anderson, John C. Tannehill, and Richard H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. Taylor & Francis, third edition 2012.
- [5] N. Andrianov and G. Warnecke. The Riemann problem for the Baer-Nunziato two-phase flow model. *Journal of Computational Physics*, 212:434–464, 2004.
- [6] M.R. Baer and J.W. 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.
- [7] D. S. Balsara. Self-adjusting, positivity preserving high order schemes for hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231:7504–7517, September 2012.
- [8] D. S. Balsara, T. Rumpf, M. Dumbser, and C.-D. Munz. Efficient, high accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics. *Journal of Computational Physics*, 228:2480–2516, April 2009.
- [9] D.S. Balsara, C. Meyer, M. Dumbser, H. Du, and Z. Xu. Efficient implementation of ADER schemes for Euler and magnetohydrodynamical flows on structured meshes – Speed comparisons with Runge–Kutta methods. *Journal of Computational Physics*, 235:934–969, 2013.

- [10] D.S. Balsara, C. Meyer, M. Dumbser, H. Du, and Z. Xu. Efficient implementation of ADER schemes for euler and magnetohydrodynamical flows on structured meshes speed comparisons with rungekutta methods. *Journal of Computational Physics*, 235:934 – 969, 2013.
- [11] D.S. Balsara, T. Rumpf, M. Dumbser, and C.D. Munz. Efficient, high accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics. *Journal of Computational Physics*, 228:2480–2516, 2009.
- [12] T.J. Barth and D.C. Jespersen. The design and application of upwind schemes on unstructured meshes. *AIAA Paper 89-0366*, pages 1–12, 1989.
- [13] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 2000.
- [14] David J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99(2-3):235–394, 1992.
- [15] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82:64–84, May 1989.
- [16] M. J. Berger and J. Olinger. Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations. *Journal of Computational Physics*, 53:484, March 1984.
- [17] Marsha J. Berger, David L. George, Randall J. LeVeque, and Kyle T. Mandli. The geoclaw software for depth-averaged flows with adaptive refinement. *Advances in Water Resources*, 34(9):1195 – 1206, 2011.
- [18] M. Berndt, J. Breil, S. Galera, M. Kucharik, P.H. Maire, and M. Shashkov. Two-step hybrid conservative remapping for multimaterial arbitrary Lagrangian–Eulerian methods. *Journal of Computational Physics*, 230:6664–6687, 2011.
- [19] Walter Boscheri, Michael Dumbser, and Olindo 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.
- [20] J.U Brackbill, D.B Kothe, and C Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335–354, 1992.

- [21] Benjamin Braconnier and Boniface Nkonga. An all-speed relaxation scheme for interface flows with surface tension. *Journal of Computational Physics*, 228(16):5722–5739, 2009.
- [22] J. Breil, S. Galera, and P.H. Maire. Multi-material ALE computation in inertial confinement fusion code CHIC. *Computers and Fluids*, 46:161–167, 2011.
- [23] G. Carré, S. Del 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.
- [24] M.J. Castro, J.M. Gallardo, J.A. 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.
- [25] M.J. Castro, J.M. 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.
- [26] 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*, 227:8107–8129, 2008.
- [27] 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*, 227:8107–8129, 2008.
- [28] M.J. Castro, A. Pardo, C. Parés, and E.F. Toro. On some fast well-balanced first order solvers for nonconservative systems. *Mathematics of Computation*, 79:1427–1472, 2010.
- [29] Jos Rafael Cavalcanti, Michael Dumbser, David da Motta-Marques, and Carlos Ruberto Fragoso Junior. A conservative finite volume scheme with time-accurate local time stepping for scalar transport on unstructured grids. *Advances in Water Resources*, 86, Part A:217 – 230, 2015.
- [30] J. Cheng and C.W. Shu. A high order ENO conservative Lagrangian type scheme for the compressible Euler equations. *Journal of Computational Physics*, 227:1567–1596, 2007.

- [31] S. Clain, S. Diot, and R. Loubère. A high-order finite volume method for systems of conservation laws multi-dimensional optimal order detection (MOOD). *Journal of Computational Physics*, 230(10):4028 – 4050, 2011.
- [32] B. Cockburn, S. Hou, and C. W. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Mathematics of Computation*, 54:545–581, 1990.
- [33] B. Cockburn, S. Y. Lin, and C.W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems. *Journal of Computational Physics*, 84:90–113, 1989.
- [34] B. Cockburn and C. W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Mathematics of Computation*, 52:411–435, 1989.
- [35] B. Cockburn and C. W. Shu. The Runge-Kutta local projection P1-Discontinuous Galerkin finite element method for scalar conservation laws. *Mathematical Modelling and Numerical Analysis*, 25:337–361, 1991.
- [36] B. Cockburn and C. W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. *Journal of Computational Physics*, 141:199–224, 1998.
- [37] Phillip Colella and Paul R. Woodward. The piecewise parabolic method (ppm) for gas-dynamical simulations. *Journal of Computational Physics*, 54(1):174–201, 1984.
- [38] V. Deledicque and M.V. Papalexandris. An exact riemann solver for compressible two-phase flow models containing non-conservative products. *Journal of Computational Physics*, 222:217–245, 2007.
- [39] Vincent Deledicque and Miltiadis V. Papalexandris. An exact riemann solver for compressible two-phase flow models containing non-conservative products. *Journal of Computational Physics*, 222(1):217–245, 2007.
- [40] S. Diot, S. Clain, and R. Loubère. Improved detection criteria for the multi-dimensional optimal order detection (MOOD) on unstructured meshes with very high-order polynomials. *Computers and Fluids*, 64:43 – 63, 2012.

- [41] S. Diot, R. Loubère, and S. Clain. The MOOD method in the three-dimensional case: Very-high-order finite volume method for hyperbolic systems. *International Journal of Numerical Methods in Fluids*, 73:362–392, 2013.
- [42] D A Drew. Mathematical modeling of two-phase flow. *Annual Review of Fluid Mechanics*, 15(1):261–291, 1983.
- [43] Donald A. Drew. Averaged field equations for two-phase media. *Studies in Applied Mathematics*, 50(2):133–166, 1971.
- [44] M. Dumbser. Arbitrary-Lagrangian-Eulerian ADER-WENO Finite Volume Schemes with Time-Accurate Local Time Stepping for Hyperbolic Conservation Laws. *Computational Methods in Applied Mechanics and Engineering*, 280:57–83, 2014.
- [45] 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.
- [46] M. Dumbser, D.S. Balsara, E.F. Toro, and C.-D. Munz. A unified framework for the construction of one-step finite volume and discontinuous galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227:8209 – 8253, 2008.
- [47] M. Dumbser, M. Castro, C. Parés, and E.F. Toro. ADER schemes on unstructured meshes for non-conservative hyperbolic systems: Applications to geophysical flows. *Computers and Fluids*, 38:1731–1748, 2009.
- [48] 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.
- [49] 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*, 199:625–647, 2010.
- [50] M. Dumbser, A. Hidalgo, and O. Zanotti. High Order Space-Time Adaptive ADER-WENO Finite Volume Schemes for Non-Conservative Hyperbolic Systems. *Computer Methods in Applied Mechanics and Engineering*, 268:359–387, 2014.
- [51] M. Dumbser and M. Kaeser. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics*, 221:693–723, February 2007.



- [52] M. Dumbser, M. Kaeser, V. A. Titarev, and E. F. Toro. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *Journal of Computational Physics*, 226:204–243, September 2007.
- [53] M. Dumbser, M. Käser, and E. F. Toro. An arbitrary high order discontinuous Galerkin method for elastic waves on unstructured meshes V: Local time stepping and  $p$ -adaptivity. *Geophysical Journal International*, 171:695–717, 2007.
- [54] M. Dumbser, M. Kser, and E. F. Toro. An arbitrary high-order discontinuous galerkin method for elastic waves on unstructured meshes - v. local time stepping and  $p$ -adaptivity. *Geophysical Journal International*, 171(2):695–717, 2007.
- [55] M. Dumbser and C.D. Munz. Building blocks for arbitrary high order discontinuous Galerkin schemes. *Journal of Scientific Computing*, 27:215–230, 2006.
- [56] M. Dumbser, T. Schwartzkopff, and C.D. Munz. Arbitrary high order finite volume schemes for linear wave propagation. In *Computational Science and High Performance Computing II*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design (NNFM), pages 129–144. Springer, 2006.
- [57] M. Dumbser and O. Zanotti. Very high order PNPM schemes on unstructured meshes for the resistive relativistic MHD equations. *Journal of Computational Physics*, 228:6991–7006, 2009.
- [58] M. Dumbser, O. Zanotti, A. Hidalgo, and D.S. Balsara. ADER-WENO Finite Volume Schemes with Space-Time Adaptive Mesh Refinement. *Journal of Computational Physics*, 248:257–286, 2013.
- [59] Michael Dumbser. A simple two-phase method for the simulation of complex free surface flows. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12):1204–1219, 2011.
- [60] Michael Dumbser. A diffuse interface method for complex three-dimensional free surface flows. *Computer Methods in Applied Mechanics and Engineering*, 257(0):47–64, 2013.
- [61] Michael Dumbser and Dinshaw 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.

- [62] Michael Dumbser and Walter 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(0):405–432, 2013.
- [63] Michael Dumbser, Arturo Hidalgo, Manuel Castro, Carlos Pars, and Eleuterio F. Toro. FORCE schemes on unstructured meshes II: Non-conservative hyperbolic systems. *Computer Methods in Applied Mechanics and Engineering*, 199(9-12):625–647, 2010.
- [64] Michael Dumbser and Eleuterio F. Toro. A Simple Extension of the Osher Riemann Solver to Non-conservative Hyperbolic Systems. *Journal of Scientific Computing*, 48(1–3):70–88, 2011.
- [65] Michael Dumbser and Eleuterio F. Toro. On universal Osher-type schemes for general nonlinear hyperbolic conservation laws. *Communications in Computational Physics*, 10(3):635–671, 2011.
- [66] Michael Dumbser, Olindo Zanotti, Raphal Loubre, and Steven Diot. A posteriori subcell limiting of the discontinuous galerkin finite element method for hyperbolic conservation laws. *Journal of Computational Physics*, 278:47 – 75, 2014.
- [67] B. Einfeldt. On Godunov-type methods for gas dynamics. *SIAM J. Numer. Anal.*, 25:294–318, 1988.
- [68] B. Einfeldt. On godunov-type methods for gas dynamics. *J. Plasma Phys.*, 65:29–58, 2001.
- [69] 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.
- [70] R. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152:457–492, 1999.
- [71] R.P. Fedkiw, T. Aslam, and S. Xu. The Ghost Fluid method for deflagration and detonation discontinuities. *Journal of Computational Physics*, 154:393–427, 1999.
- [72] A. Ferrari, C.D. Munz, and B. Weigand. A High Order Sharp Interface Method with Local Timestepping for Compressible Multiphase Flows. *Communications in Computational Physics*, 9:205–230, 2011.

- [73] Angela Ferrari, Michael Dumbser, Eleuterio F. Toro, and Aronne Armanini. A new 3D parallel SPH scheme for free surface flows. *Computers & Fluids*, 38(6):1203–1217, 2009.
- [74] D.E Fyfe, E.S Oran, and M.J Fritts. Surface tension and viscosity with lagrangian hydrodynamics on a triangular mesh. *Journal of Computational Physics*, 76(2):349–384, 1988.
- [75] J.M. Gallardo, C. Parés, and M.J. Castro. On a well-balanced high-order finite volume scheme for shallow water equations with topography and dry areas. *Journal of Computational Physics*, 227:574–601, 2007.
- [76] G. Gassner, M. Dumbser, F. Hindenlang, and C.D. Munz. Explicit one-step time discretizations for discontinuous Galerkin and finite volume schemes based on local predictors. *Journal of Computational Physics*, 230:4232–4247, 2011.
- [77] G. Gassner and D.A. Kopriva. A comparison of the dispersion and dissipation errors of Gauss and Gauss-Lobatto discontinuous Galerkin spectral element methods. *SIAM Journal on Scientific Computing*, 33(5):2560–2579, 2011.
- [78] S. K. Godunov. A difference method for numerical calculations of discontinuous solutions of the equations of hydrodynamics. *Mat. Sb.*, 47:271, 1959. in Russian.
- [79] Keith A. Gonthier and Joseph M. Powers. A high-resolution numerical method for a two-phase model of detonation transition. *J. Comput. Phys*, pages 376–433, 2000.
- [80] L.H. Han, X.Y. Hu, and N.A. Adams. Scale separation for multi-scale modeling of free-surface and two-phase flows with the conservative sharp interface method. *Journal of Computational Physics*, 280(0):387–403, 2015.
- [81] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71:231–303, 1987.
- [82] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly High Order Accurate Essentially Non-oscillatory Schemes III. *Journal of Computational Physics*, 71:231–303, 1987.
- [83] A. Harten, P. D. Lax, and B. van Leer. On upstream differencing and godunov-type schemes for hyperbolic conservation laws. *SIAM Rev.*, 25:35, 1983.

- [84] A. Hidalgo and M. Dumbser. ADER schemes for nonlinear systems of stiff advection-diffusion-reaction equations. *Journal of Scientific Computing*, 48:173–189, 2011.
- [85] C.W Hirt and B.D Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries . *Journal of Computational Physics*, 39(1):201–225, 1981.
- [86] G. Jiang and C.W. Shu. On a cell entropy inequality for discontinuous Galerkin methods. *Mathematics of Computation*, 62:531–538, 1994.
- [87] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted eno schemes. *J. Comput. Phys*, 126:202–228, 1996.
- [88] Myungjoo Kang, RonaldP. Fedkiw, and Xu-Dong Liu. A Boundary Condition Capturing Method for Multiphase Incompressible Flow. *Journal of Scientific Computing*, 15(3):323–360, 2000.
- [89] A. K. Kapila, R. Menikoff, J. B. Bdzil, S. F. Son, and D. S. Stewart. Two-phase modeling of deflagration-to-detonation transition in granular materials: Reduced equations. *Physics of Fluids*, 13(10):3002–3024, 2001.
- [90] V. P. Kolgan. Application of the minimum-derivative principle in the construction of finite-difference schemes for numerical analysis of discontinuous solutions in gas dynamics. *Transactions of the Central Aerohydrodynamics Institute*, 3(6):68–77, 1972. in Russian.
- [91] D.A. Kopriva. Metric identities and the discontinuous spectral element method on curvilinear meshes. *Journal of Scientific Computing*, 26(3):301–327, 2006.
- [92] D.A. Kopriva and G. Gassner. On the quadrature and weak form choices in collocation type discontinuous galerkin spectral element methods. *Journal of Scientific Computing*, 44(2):136–155, 2010.
- [93] Jasper J. Kreeft and Barry Koren. A new formulation of kapilas five-equation model for compressible two-fluid flow, and its numerical treatment. *Journal of Computational Physics*, 229(18):6220 – 6242, 2010.
- [94] L. Krivodonova and R.Qin. An analysis of the spectrum of the discontinuous galerkin method. *Applied Numerical Mathematics*, 64:1–18, 2013.
- [95] P. G. LeFloch. Shock waves for nonlinear hyperbolic systems in nonconservative form. Institute for Math. and its Appl., Minneapolis, Preprint 593.

- [96] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, New York, 2002.
- [97] Randall J. LeVeque and Zhilin Li. Immersed Interface Methods for Stokes Flow with Elastic Boundaries or Surface Tension. *SIAM J. Sci. Comput.*, 18(3):709–735, 1997.
- [98] Rainald Lhner. An adaptive finite element scheme for transient problems in {CFD}. *Computer Methods in Applied Mechanics and Engineering*, 61(3):323 – 338, 1987.
- [99] W. Liu, J. Cheng, and C.W. 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.
- [100] Y. Liu, M. Vinokur, and Z.J. Wang. Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids V: Extension to Three-Dimensional Systems. *Journal of Computational Physics*, 212:454–472, 2006.
- [101] F. Lörcher, G. Gassner, and C.-D. Munz. A discontinuous galerkin scheme based on a space–time expansion. i. inviscid compressible flow in one space dimension. *Journal of Scientific Computing*, 32(2):175–199, 2007.
- [102] F. Lörcher, G. Gassner, and C. D. Munz. A discontinuous Galerkin scheme based on a space-time expansion. I. inviscid compressible flow in one space dimension. *Journal of Scientific Computing*, 32:175–199, 2007.
- [103] 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 laws. *Communication in Computational Physics*, 16:718–763, 2014.
- [104] J. Luo, X.Y. Hu, and N.A. Adams. A conservative sharp interface method for incompressible multiphase flows. *Journal of Computational Physics*, 284(0):547–565, 2015.
- [105] 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.
- [106] Takashi Hibiki. Mamoru Ishii. *Thermo-fluid dynamic theory of two-phase flow*. Springer, second edition 2011.

- [107] S. Le Martelot, R. Saurel, and B. Nkonga. Towards the direct numerical simulation of nucleate boiling flows. *International Journal of Multiphase Flow*, 66:62 – 78, 2014.
- [108] G. Dal Maso, P.G. LeFloch, and F. Murat. Definition and weak stability of non-conservative products. *J. Math. Pures Appl.*, 74:483–548, 1995.
- [109] 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. *Astrophysical Journal Suppl.*, 198:7, January 2012.
- [110] Joseph P. Morris. Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, 33(3):333–353, 2000.
- [111] M.L. Muñoz and C. Parés. Godunov method for nonconservative hyperbolic systems. *Mathematical Modelling and Numerical Analysis*, 41:169–185, 2007.
- [112] W. Mulder, S. Osher, and J.A. Sethian. Computing interface motion in compressible gas dynamics. *Journal of Computational Physics*, 100:209–228, 1992.
- [113] A. Murrone and H. Guillard. A five equation reduced model for compressible two phase flow problems. *Journal of Computational Physics*, 202:664–698, 2005.
- [114] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2003.
- [115] S. Osher and J.A. Sethian. Fronts propagating with curvature–dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
- [116] S. Osher and F. Solomon. Upwind difference schemes for hyperbolic conservation laws. *Math. Comput.*, 38:339–374, 1982.
- [117] Miltiadis V. Papalexandris. A two-phase model for compressible granular flows based on the theory of irreversible processes. *Journal of Fluid Mechanics*, 517:103–112, 2004.
- [118] C. Parés and M.J. Castro. On the well-balance property of roe’s method for nonconservative hyperbolic systems. applications to shallow-water systems. *Mathematical Modelling and Numerical Analysis*, 38:821–852, 2004.

- [119] Carlos Parés. Numerical methods for nonconservative hyperbolic systems: a theoretical framework. *SIAM Journal on Numerical Analysis*, 44(1):300–321, 2006.
- [120] Nayan Patel. *Simulation of hydrodynamic fragmentation from a fundamental and an engineering perspective*. PhD Thesis in the School of Aerospace Engineering, Georgia Institute of Technology, 2007.
- [121] Guillaume Perigaud and Richard Saurel. A compressible flow model with capillary effects. *Journal of Computational Physics*, 209(1):139–178, 2005.
- [122] J. Qiu, M. Dumbser, and C.W. Shu. The discontinuous Galerkin method with Lax-Wendroff type time discretizations. *Computer Methods in Applied Mechanics and Engineering*, 194:4528–4543, 2005.
- [123] Lord Rayleigh. On the Capillary Phenomena of Jets. *Proceedings of the Royal Society of London*, 29(196–199):71–97, 1879.
- [124] R. Biswas, K. D. Devine, and J. E. Flaherty. Parallel, adaptive finite element methods for conservation laws. *APPL. NUMER. MATH*, 14:255–283, 1994.
- [125] W.H. Reed and T.R. Hill. Triangular mesh methods for neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
- [126] S. Rhebergen, O. Bokhove, and J.J.W. van der Vegt. Discontinuous Galerkin finite element methods for hyperbolic nonconservative partial differential equations. *Journal of Computational Physics*, 227:1887–1922, 2008.
- [127] P.L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 1981.
- [128] E. Romenski, A. D. Resnyansky, and E. F. Toro. Conservative hyperbolic formulation for compressible two-phase flow with different phase pressures and temperatures. *Quarterly of Applied Mathematics*, 65:259–279, 2007.
- [129] V. V. Rusanov. Calculation of Interaction of NonSteady Shock Waves with Obstacles. *J. Comput. Math. Phys. USSR*, 1:267–279, 1961.
- [130] R. Saurel and R. Abgrall. A Simple Method for Compressible Multifluid Flows. *SIAM Journal on Scientific Computing*, 21:1115–1145, 1999.

- [131] Richard Saurel and R. Abgrall. A Multiphase Godunov Method for Compressible Multifluid and Multiphase Flows. *Journal of Computational Physics*, 150(2):425–467, 1999.
- [132] Richard Saurel, Sergey Gavriluk, and Francois Renaud. A multiphase model with internal degrees of freedom: application to shockbubble interaction. *Journal of Fluid Mechanics*, 495:283–321, 2003.
- [133] Richard Saurel and Oliver Lemetayer. A multiphase model for compressible flows with interfaces, shocks, detonation waves and cavitation. *Journal of Fluid Mechanics*, 431:239–271, 2001.
- [134] Richard Saurel, Fabien Petitpas, and Ray A. Berry. Simple and efficient relaxation methods for interfaces separating compressible fluids, cavitating flows and shocks in multiphase mixtures. *Journal of Computational Physics*, 228(5):1678 – 1712, 2009.
- [135] Richard Saurel, Fabien Petitpas, and Ray A. Berry. Simple and efficient relaxation methods for interfaces separating compressible fluids, cavitating flows and shocks in multiphase mixtures . *Journal of Computational Physics*, 228(5):1678–1712, 2009.
- [136] T. Schwartzkopff, C.D. Munz, and E.F. Toro. ADER: A high order approach for linear hyperbolic systems in 2d. *Journal of Scientific Computing*, 17(1-4):231–240, 2002.
- [137] D.W. Schwendeman, C.W. Wahle, and A.K. Kapila. The Riemann problem and a high-resolution Godunov method for a model of compressible two-phase flow. *Journal of Computational Physics*, 212(2):490–526, 2006.
- [138] John C Slattery, Leonard Sagis, and Eun-Suok Oh. *Interfacial Transport Phenomena*. Springer, second edition 2007.
- [139] Gary A Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1):1 – 31, 1978.
- [140] A.H. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1971.
- [141] Mark Sussman and Elbridge Gerry Puckett. A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows. *Journal of Computational Physics*, 162(2):301–337, 2000.



- [142] Mark Sussman, Peter Smereka, and Stanley Osher. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics*, 114(1):146–159, 1994.
- [143] A. Taube, M. Dumbser, D. Balsara, and C.D. Munz. Arbitrary high order discontinuous Galerkin schemes for the magnetohydrodynamic equations. *Journal of Scientific Computing*, 30:441–464, 2007.
- [144] V.A. Titarev and E.F. Toro. ADER: Arbitrary high order Godunov approach. *Journal of Scientific Computing*, 17(1-4):609–618, December 2002.
- [145] V.A. Titarev and E.F. Toro. ADER schemes for three-dimensional nonlinear hyperbolic systems. *Journal of Computational Physics*, 204:715–736, 2005.
- [146] V.A. Titarev and E.F. Toro. WENO schemes based on upwind and centred TVD fluxes. *Computers and Fluids*, 34:705–720, 2005.
- [147] S.A. Tokareva and E.F. Toro. HLLC-type Riemann solver for the BaerNunziato equations of compressible two-phase flow. *Journal of Computational Physics*, 229(10):3573–3604, 2010.
- [148] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, third edition 2009.
- [149] E.F. Toro, R.C. Millington, and L.A.M Nejad. Towards very high order Godunov schemes. In E.F. Toro, editor, *Godunov Methods. Theory and Applications*, pages 905–938. Kluwer/Plenum Academic Publishers, 2001.
- [150] E.F. Toro and V.A. Titarev. TVD fluxes for the high-order ADER schemes. *Journal of Scientific Computing*, 24:285–309, 2005.
- [151] I. Toumi. A weak formulation of Roe’s approximate Riemann solver. *Journal of Computational Physics*, 102:360–373, 1992.
- [152] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169(2):708 – 759, 2001.
- [153] Bram van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method . *Journal of Computational Physics*, 32(1):101–136, 1979.

- [154] Zhaoyuan Wang, Jianming Yang, Bonguk Koo, and Frederick Stern. A coupled level set and volume-of-fluid method for sharp interface simulation of plunging breaking waves. *International Journal of Multiphase Flow*, 35(3):227–246, 2009.
- [155] Z.J. Wang and Y. Liu. Extension of the spectral volume method to high-order boundary representation. *Journal of Computational Physics*, 211:154–178, 2006.
- [156] Z.J. Wang, L. Zhang, and Y. Liu. Spectral (finite) volume method for conservation laws on unstructured grids iv: Extension to two-dimensional euler equations. *Journal of Computational Physics*, 194:716–741, 2004.
- [157] S Whitaker. *The method of volume averaging*. Kluwar Academic Publishers, Netherlands, 1999.
- [158] Rui Xu, Peter Stansby, and Dominique Laurence. Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach. *Journal of Computational Physics*, 228(18):6703–6725, 2009.
- [159] Olindo Zanotti, Francesco Fambri, Michael Dumbser, and Arturo Hidalgo. Space-time adaptive ADER discontinuous galerkin finite element schemes with a posteriori sub-cell finite volume limiting. *Computers and Fluids*, 118:204 – 224, 2015.