

A MULTIPHASE FLUID-STRUCTURE COMPUTATIONAL
FRAMEWORK
FOR UNDERWATER IMPLOSION PROBLEMS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF MECHANICAL
ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Arthur Seiji Daniel Rallu

May 2009

© Copyright by Arthur Seiji Daniel Rallu 2009

All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Charbel Farhat) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Rémi Abgrall)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Ron Fedkiw)

Approved for the University Committee on Graduate Studies.

Abstract

The intense implosive collapse of an air-filled underwater structure can lead to ultra-high compressions. Shocks emanating from this process are a potential threat to a nearby submarine hull, particularly in the presence of UNDEX loading. Therefore, with the projected increase of the number of air-backed volumes external to a submarine hull, implosion has become a source of concern to the Navy who now requires an improved capability to design and qualify submarine external payloads for implosion avoidance and platform survivability. However, the development of such a computational model for simulating payload implosion and predicting platform damage is a formidable challenge. It requires accounting for all of the fluid, gas, and structural aspects of this complex problem as well as their interactions. It also requires incorporating in the computations material and failure models, UNDEX loading, and capturing the precise effects on the pressure peaks of many factors such as the rate of structural collapse, hydrodynamic instability at the fluid/bubble interface, and cavitation when it occurs in the fluid. To this effect, this thesis makes essential contributions to computational mechanics that pave the way for the simulation of implosion and the assessment of platform survivability. These include a novel finite volume scheme for the solution of two-phase gas/water problems characterized by strong contact discontinuities. This discretization scheme is equipped with an exact,

local, two-phase Riemann solver for computing the fluxes at the material interface without crossing it, which makes it robust with respect to a large discontinuity of the density and a strong pressure jump at the material interface. This two-phase flow scheme is also proven to be contact preserving and nearly conservative. To achieve computational efficiency for complex equations of state such as that of Jones, Wilkins and Lee, this scheme is also equipped with a sparse grid method for tabulating and interpolating data relevant to the exact solution of a Riemann problem. The contributions of this thesis also include an exact local fluid-structure Riemann solver for embedded fluid methods, the implementation of all aforementioned CFD methods in a state-of-the-art fluid-structure analysis environment, and the application of the developed computational technology to the simulation of the implosion of aluminum cylinders with diameter-thickness ratio of approximately 54. The obtained numerical predictions are found to correlate well with the results of the corresponding experiments performed simultaneously at the University of Texas at Austin.

Acknowledgements

I am very grateful to my advisor Prof. Charbel Farhat for convincing me of further pursuing my studies through this thesis, but also for providing me with this interesting research topic. I also thank him for his constant support and guidance in my research.

I would like to thank Professors Rémi Abgrall and Ron Fedkiw who, in addition to forming my thesis committee, have contributed invaluable insights to this work. I am also most grateful to professors Eli Turkel and Jean-Frédéric Gerbeau for their most precious help and for their constant encouragement.

Of course, I am grateful to my present and former colleagues at the FRG. My special thanks go to Dr. Charbel Boumosle, David Amsallem and Axel Strang.

I am also thankful to all my friends, here and there, as they either cheered me up and helped me take a step back from my research when necessary.

Finally, I wish to thank my family and my fiancée.

Contents

Abstract	v
Acknowledgements	vii
1 Problem Definition	1
1.1 Motivations	1
1.2 Objectives	5
1.3 Thesis Accomplishments and Outline	8
2 Physical Model and Mathematical Properties	11
2.1 Assumptions	12
2.2 Governing Equations	15
2.2.1 Fluid Governing Equations	15
2.2.2 Structure Governing Equations	16
2.2.3 Transmission Conditions	17
2.3 Thermodynamics, Equations of State and Constitutive Models	18
2.3.1 Equations of State	18
2.3.2 Fluid Thermodynamics	23
2.3.3 Constitutive Laws for a Structure	26

2.4	Mathematical Properties	28
2.4.1	Hyperbolicity of the Euler Equations	29
2.4.2	Genuinely Non-Linear and Linearly Degenerate Fields	32
2.4.3	Existence of Discontinuous Solutions	35
2.4.4	Rarefaction Waves	37
2.4.5	Review of the One-Dimensional Riemann Problem	39
3	Computational Framework	43
3.1	Introduction	43
3.2	Arbitrary-Lagrangian-Eulerian Framework	47
3.3	Finite Volume Semi-Discretization of the Fluid Governing Equations for a Single-Phase Flow	49
3.4	Time Integration of Semi-Discrete Equations on Moving Grids	56
3.4.1	Second-Order Runge-Kutta	58
3.4.2	Three-Point Backward Difference	59
3.5	Level Set Equation	62
3.6	Finite Element Semi-Discretization of the Structure Governing Equations	66
3.7	Time Discretization of the Structure Governing Equations	68
3.7.1	Explicit Central Difference Scheme	68
3.7.2	Midpoint Rule	71
4	A Two-Phase Riemann Solver Based Approach	73
4.1	Introduction	74
4.2	Basic Ideas	78
4.2.1	Exact Two-Phase Riemann Solvers	79
4.2.2	Numerical Interfacial Flux	86

4.2.3	Rationale	90
4.2.4	Update of Phase-Changing Nodes	93
4.3	Performance Considerations for the Exact Two-Phase Riemann Solver	96
4.3.1	Riemann Invariants	99
4.3.2	Tabulation Requirements	100
4.3.3	Sparse Grids	102
4.3.4	Application to an Analytical Function	111
4.3.5	Application to the JWL EOS	112
4.4	Second-Order Reconstruction	115
4.5	Time Discretization and Complete Algorithm	118
4.5.1	Interfacial Treatment	120
4.5.2	Second-Order Runge-Kutta (I)	122
4.5.3	Second-Order Runge-Kutta (II)	124
4.6	Contact-Preserving Property	125
4.7	Mass Conservation	127
4.8	One-Dimensional Two-Phase Flow Benchmark Problems	128
5	Fluid-Structure Coupling	143
5.1	Introduction	143
5.2	Mesh Motion	146
5.3	Three-Field Problem	149
5.4	Semi-discretization of the Three-field Equations	150
5.4.1	Semi-Discretization of the Mesh Equation	151
5.4.2	Compatibility Equations	152
5.4.3	Transmission Conditions and Load Transfer	154
5.5	Time Discretization	155

5.5.1	The GSS Procedure	155
5.5.2	Fluid Time Discretization	156
5.5.3	Structure Time Discretization	160
5.5.4	Second-Order Time-Accurate Loosely-Coupled Solution Algorithms	163
5.6	Numerical Accuracy Study	172
6	Applications to Underwater Implosions	179
6.1	Collapsing Bubble	179
6.2	Collapsing Cylinders	185
6.2.1	Influence of the Initialization	189
6.2.2	Collapse of Three-Dimensional Cylinders	193
6.3	Summary	205
7	Conclusions and Perspectives for Future Work	207
7.1	Summary and Conclusions	207
7.2	Perspectives for Future Work	209
A	Second-Order Accuracy of the GSS Procedure	211
B	One-Dimensional Two-Phase Riemann Problem	215
B.1	Non-Linear Equation Form	216
B.1.1	Shock Wave Relations for a Stiffened Gas	217
B.1.2	Expansion Wave Relations for a Stiffened Gas	218
B.1.3	Shock Wave Relations for Tait's EOS	218
B.1.4	Expansion Wave Relations for Tait's EOS	219
B.1.5	Local Solution by Newton's Method	219

B.2	System Form	220
B.2.1	Shockwave Relation	220
B.2.2	Expansion Wave Relation	221
B.2.3	Local Solution by Newton's Method	221
	Bibliography	225

List of Tables

2.1	Conventional parameter values for the JWL EOS when TNT explosive is considered.	21
4.1	Average timings (in seconds) of flux computation with and without an exact Riemann solver between two stiffened gas	96
4.2	Average timings (in seconds) of flux computation with and without an exact Riemann solver when a fluid with a JWL EOS undergoes a rarefaction. When an exact Riemann solver is used, the number of integration steps to compute the Riemann invariant is specified in parentheses.	97
4.3	Accuracy of the sparse grid tabulation of the Riemann invariant for the JWL EOS with respect to the number of grid points and the level of refinement in each direction. The range of the tabulation is $(\rho, s) \in [3.0, 1630.0] \times [-10^6, 10^{10}]$	115
4.4	Perfect gas - stiffened gas: limits of the GFM and GFMP and advantage of the FVM-ERS for problems with a strong interfacial contact discontinuity.	138
5.1	Time steps for the different simulations	173

List of Figures

2.1	Different bounds on the pressure with respect to the density for the JWL EOS to ensure hyperbolicity, genuine non-linearity and unicity of the solution to the Riemann problem.	35
2.2	Initial Riemann problem (bottom), structure of the solution (middle) with a rarefaction on the left, a shock on the right and a contact discontinuity in the middle separating the two interfacial states, and solution at a specific time (top)	40
3.1	Control volume (lighter lines) in an unstructured tetrahedral (heavier lines) mesh (only one of the tetrahedra needed to construct the graphically depicted control volume is shown).	51
4.1	The material interface located between two nodes is assumed to go through the intersection $\partial\Omega_{ij} \cap [ij]$ of the two corresponding control volume boundary and of the corresponding edge. The normals of the material interface and of the control surface are however different. . .	87

4.2	Decomposition of the two-phase Riemann problem (top) into two single-phase Riemann problems (bottom): the initial conditions are plotted in blue dotted lines, the solutions at a given time $t > 0$ are in black full lines.	92
4.3	Phase change update: node i is about to change phases and extrapolation will proceed only along edges with red nodes	95
4.4	Construction of the one-dimensional hierarchical basis (on the right) in comparison to the conventional nodal basis (on the left).	104
4.5	Construction of the two-dimensional hierarchical basis based on the second-order tensor product of one-dimensional hierarchical basis. . .	105
4.6	Example of a one-dimensional function and its approximations on different hierarchical basis.	107
4.7	Analytical example: solution obtained from the analytical formula . .	113
4.8	Analytical example: solution obtained from the sparse grid by interpolation	113
4.9	Analytical example: absolute errors	114
4.10	Analytical example: relative errors	114
4.11	Example of a two-dimensional sparse grid for the tabulation of the JWL Riemann invariant in the (ρ, s) -range $[100, 1700] \times [10^3, 10^7]$. .	116
4.12	Solutions of the Sod shock-tube problem given by the FVM-ERS in comparison to the exact solution at $t = 0.2$ as well as mass conservation of the problem.	131
4.13	Solutions of the shock-tube problem given by the FVM-ERS in comparison to the exact solution at $t = 0.015$ as well as mass conservation of the problem.	132

4.14	Reflection-less gaseous shock problem: variations of the density, pressure, and velocity at $t = 0.06$ along the shock tube (FVM-ERS, $\Delta x = 1/201$) — Zoom on the main oscillation is shown for the pressure field at the bottom right part of the figure.	134
4.15	Perfect gas - stiffened gas: variations of the density, pressure, and velocity at $t = 2.4 \times 10^{-4}$ along the length of the shock-tube ($\Delta x = 1/201$) — Zoom on the “plateau” region is shown for the density field at the top right part of the figure.	136
4.16	Perfect gas - stiffened gas: variations of the density, pressure, and velocity at $t = 2.4 \times 10^{-4}$ along the length of the shock-tube ($\Delta x = 1/801$) — Zoom on the “plateau” region is shown for the density field at the top right part of the figure.	137
4.17	Air (perfect gas) - water (barotropic fluid/stiffened gas): variations of the density, pressure, and velocity at $t = 4.0 \times 10^{-4}$ along the length of the shock-tube (FVM-ERS, $\Delta x = 1/201$).	141
4.18	Underwater explosion: variations of the density, pressure, and velocity at $t = 3.0 \times 10^{-5}$ along the length of the shock-tube.	142
5.1	The GSS procedure	157
5.2	Mesh	175
5.3	Displacement errors for the structure subsystem using schemes A through E (linear structural model).	176
5.4	Total state errors for the fluid subsystem using schemes A through E (linear structural model).	176
5.5	Displacement errors for the structure subsystem with scheme A (non-linear structural model)	177

5.6	Total state errors for the fluid subsystem with scheme A (nonlinear structural model)	177
5.7	Displacement errors for the structure subsystem using scheme F (non-linear structural model)	178
5.8	Total state errors for the fluid subsystem using scheme F (nonlinear structural model)	178
6.1	Pressure time-histories recorded at one of the sensors during the four experiments described in [82].	181
6.2	Schematic of the implosion experiment reported in [82] and corresponding computational domain (dashed lines represent the non-reflecting boundaries of the two-dimensional computational domain adopted in [82]).	183
6.3	Comparison of the pressure time-histories predicted by AERO-F and DYSMAS as well as their corresponding test data.	184
6.4	Density contour plots (in kg/m^3) predicted by AERO-F equipped with the FVM-ERS in the vicinity of the air bubble during its collapse. . .	186
6.5	Experimental setup for the observation of the mode-2 collapse of an aluminum cylinder	188

6.6	Pressure signals in the vicinity of a cylinder collapsing in mode 2 at the critical pressure of 184.5 psi. The curves correspond to different initializations of the simulation: (a) the critical pressure is initially applied to an imperfect cylinder with a 1%-mode 2 deformation, (b) pressure is increased until the critical pressure is reached while the cylinder has an initial 1%-mode 2 deformation, (c) the critical pressure is initially applied to an imperfect cylinder with a 1%-mode 4 deformation, (d) the critical pressure is initially applied to a perfect cylinder being deformed by an indenter.	191
6.7	Pressure signals in the vicinity of a cylinder collapsing in mode 2 at the critical pressure of 184.5 psi. The curves correspond to different initializations of the simulation. All simulations start from a steady-state of the flow and the structure. Collapse of the cylinder is triggered by: (a) a sudden 1%-mode 2 deformation, (b) a sudden 1%-mode 4 deformation, (c) a deformation due to an indenter, (d) a sudden weakness of one of the element	192
6.8	Mode 2 collapse: schematic drawing of the cylinder (striped area correspond to the end caps).	194
6.9	Photography of the cylinder after a mode-2 collapse. No crack is observed.	194
6.10	Mode 2 collapse: the coarseness of the mesh inside the cylinder is necessary to allow for mesh motion until contact	195
6.11	Mode 2 collapse: final deformation of the cylinder.	196

6.12	Mode 2 collapse: comparison of numerical and experimental pressure time-histories for two different setups of the contact algorithm in the structural solver.	198
6.13	Mode 4 collapse: schematic drawing of the cylinder (striped area correspond to the end caps).	199
6.14	Mode 4 collapse: experimental pressure time-histories.	200
6.15	Photography of the cylinder after a mode-4 collapse. Cracks of different sizes can be observed.	201
6.16	Mode 4 collapse: initial fluid mesh in the symmetry plane.	202
6.17	Mode 4 collapse: fluid mesh deformations at the contact in the symmetry plane.	203
6.18	Mode 4 collapse: numerical pressure time-histories recorded at different locations around the structure including at the locations of the experimental sensors.	204
6.19	Mode 4 collapse: comparison of numerical and experimental pressure time-histories.	205

Chapter 1

Problem Definition

The present chapter provides an introduction to implosion problems and to the interests of the NAVY in such problems. Next, the NAVY research project to tackle the study of implosion problems is briefly described to define the framework of this thesis. This chapter ends with a list of the contributions of the present work to the field of numerical methods for implosion problems and an overview of the thesis.

1.1 Motivations

Submarines have emerged as powerful instruments of war during the Battle of the Atlantic in World War II and the Cold War. Now, submarines are an integral part of the projection and dissuasion forces of several countries. Although the design of these submarines have come a long way since World War II, their failure and the loss of the human lives aboard are always a possibility because of their extreme operational conditions. For this reason, the U.S. Navy has a strong interest in submarine design and failure analysis.

Though submarine designs can vary, the pressure hull is a common component of

all submarines. This is the main part of the submarine that houses most equipment and the submarine crew. Because of its obvious importance, its design constraints have been studied extensively and are now fairly well established. Submarine hulls must be designed not only to withstand intense hydrostatic pressures, but also to maximize the ability to withstand the effects of an underwater explosion (UNDEX).

Today's submarines are required to perform more specialized missions than in the past. As submarines are assigned to more specific missions, additional equipment must be either integrated into the original submarine design or mounted to the submarine hull. However, the additional equipment has a potential to fail or implode before the submarine hull since each piece of equipment has its own design constraints which may differ from those of the submarine hull. Such a failure poses a serious threat to the rest of the submarine, as is explained below. While the U.S. Navy's interests in the general vulnerability of submarine hulls is obvious, the increased use of specialized equipment has necessitated the consideration of more specific problems, namely the failure analysis of implodable volumes near underwater structures. Implodable volumes are defined by the Navy as "any non-compensated pressure housing containing a compressible fluid at a pressure below the external ambient sea pressure (at any depth down to maximum operating depth) which has the potential to collapse" [70]. "Volumes external to the pressure hull, designed to maintain a static, dry, nominally one atmosphere environment within the volume for the purpose of protecting pressure-sensitive components are usually classified as implodables." Examples include the Universal Modular Mast, Unmanned Undersea Vehicles (UUV) more commonly called drones, and even lighting equipment. The number of these air-filled structures is expected to increase in the future while their potential

collapse damage to nearby structures is not very well understood. One recent example of the complexity and the lack of accurate understanding of underwater explosion and implosion damage is given by Wardlaw [5]. It was shown that in the case of an underwater explosion close enough to a structure, the physics of the problem are not only governed by the interaction of the shock and the structure, but also by the interaction between the shock reflected off the structure and the explosion bubble. As the resulting expansion wave further decreases the pressure in the region between the structure and the bubble, tensile rupture of the water is favored potentially leading to the formation of a new bubble. The subsequent collapse of this bubble can have more destructive effects than the shock induced by the explosion. Because the collapse of implodable volumes can have devastating consequences, it is essential to improve the understanding of the underlying physics of implosion problems in order to better define design constraints for both submarine hulls and implodable volumes.

In the case of implosion problems, the structural failure of an implodable volume leads to the compression of the air inside. The pressure can become thousands of times greater than the atmospheric pressure and much greater than the hydrostatic pressure. The differential of pressures between the air and the water forces the air bubble to expand, generating a shock wave that travels throughout the water. Since the implodable volumes are usually mounted to the submarine hull, the shock wave's emission point is very close to the submarine and its strength is only slightly decreased before it reaches the submarine hull. Thus, it may be the effects of the implosion that lead to the failure of the submarine hull rather than the initial shock wave induced by the explosion.

These implosion problems are characterized by their violent nature and by the

strong interactions between the different mechanical systems, the structures and the fluids. In addition, up to three very different fluids, the air, water and gaseous products of explosives, interact not only with each other but also with one or more structures. Strong pressure waves travel through all media, leading to large deformations of the structures and potentially to their failure in the form of cracks or collapse. These failures are accompanied by violent motions of the flow near the structures. At the same time, other regions of the flow are, momentarily at least, unperturbed. Hence, implosions form a complex system to study as they involve the strong interactions of systems with very different behaviors and in a vast range of regimes.

While all these characteristics seem very specific, they are encountered in other problems of engineering. Scientific interest in collapsing bubbles in a liquid near a structure is not limited to the study of submarine hulls. Cavitation damage is probably the most well-known example. The consequent structural damage poses serious problems in most hydraulic engineering fields, such as the gradual destruction of ship propellers and rudders. The damage mechanism is slightly different than in the case of implosion problems, but the physics remain the same. In both cases, the expansion and compression of bubbles are the underlying cause of structural damage. In the first case bubbles come from the structural failure of implodable volumes, while in the second case they result from liquid vaporization due to pressure drops. Bubble-liquid-structure interactions are also found in medical applications as reported by [24], which includes well known shock wave lithotripsy to treat kidney stones. In a water bath, shock waves are focused towards the patient's kidney stones where impurities are the center of small bubbles. The compression and expansion of these bubbles then helps break the kidney stones into pieces small enough to be eliminated naturally by

the body. All these problems are current research topics.

1.2 Objectives

In order to provide a better understanding of the physics of implosion phenomenon and to be able to quantitatively predict its effects on nearby structures, the Office of Naval Research has set up a Multi-University Research Initiative (MURI) Project with 4 U.S. universities in order to carry out both experimental and computational work. Given the obvious impossibility of conducting repeated real-scale experiments, the project is aimed at paving the way for developing a simulation software platform and validating it using relevant laboratory experimental data.

In the experimental part of the research project, implosion tests on model cylinders are used to understand the underlying physics of the problem. Experiments focus on the collapse of cylindrical structures in a constant external pressure environment. The issues of interest include different modes of dynamic collapse, the possible fracture of the structures and their interaction with the surrounding fluid and the internal air bubble. At the same time, the results provide the basis for evaluating the numerical tools developed for solving this class of problems. These tests are performed at the University of Texas at Austin by Professor Kyriakides's team.

The computational part of the project is aimed at providing numerical tools to quantitatively predict the effects of underwater implosion of air-filled structures. A brief analysis of this class of problems helps to define two different regimes with

very different characteristics, leading to different numerical approaches. Initially, a one-atmosphere dry environment is maintained by the structure of the implodable volume. As long as cracking of the structure does not occur, air and water are kept separated. This configuration with no cracking of the structure can be observed at least in the early stages of an implosion, if not in the entire process of the collapse of a structure. However, in some cases, cracking can be initiated and propagated and hence, air and water come into contact. This second configuration differs from the first one in many aspects. Both structure and fluid are characterized by the appearance of discontinuities in the form of a crack and of a contact discontinuity respectively, while larger deformations of the structure and movements of the flow can be expected. For this reason, a two-pronged strategy is being developed to simulate implosion problems corresponding to the two possible regimes of implosion problems.

In the first stages of an implosion problem when cracking is not yet initiated, the structural deformations remain limited while air and water are still separated. This configuration does not require the modeling of a two-phase flow nor the modeling of cracks. It is then possible to use state-of-the-art mature, accurate and reliable numerical methods in both the structural and flow solvers. More precisely, given that the deformations of the structure, and hence the deformations of the fluid domain, are expected to remain limited, an Arbitrary-Lagrangian-Eulerian framework is considered for fluid-structure interactions. In the subsequent stages of an implosion problem, it becomes necessary to switch to other methods in order to model the crack initiation and propagation in the structure and the multiphase nature of the flow. In this case, the mesh motion algorithms necessary in the Arbitrary-Lagrangian-Eulerian framework are not expected to be able to handle the large deformations of the fluid domain. A purely Eulerian approach should then be considered to solve the flow. In

addition to crack initiation and propagation algorithms, new numerical methods are developed to simultaneously treat the multiphase nature of the flow and the fluid-structure interaction. This defines the general setting of an implosion problem. It can be furthermore extended to include the shock wave generation with explosives. In that case, it becomes necessary to consider the use of a multiphase flow algorithm in the Arbitrary-Lagrangian-Eulerian computational framework.

This thesis takes place within the framework described above of this MURI project and is aimed at contributing to the multidisciplinary failure analysis of submerged implodable volumes. However, it does not address all the aforementioned issues to develop the corresponding numerical tools for implosion problems. The modeling of the various fluids, their interaction with each other and their interaction with a structure are the core issues of focus of this thesis. More specifically, the modeling of both single-phase and multi-phase fluids are considered in general, while the fluid-structure interaction modeling is restricted to the Arbitrary-Lagrangian-Eulerian framework for the simulation of the early stages of implosion phenomena. The study of the structure modeling is not part of this thesis as it was performed by Professor Belytschko's team at Northwestern University and Professor Wierzbicki's team at MIT. However, given the complexity and the interdependence of the fluids and the structure, the modeling of the structure cannot be omitted. For this reason, the structural equations of motion and their numerical discretization are briefly described along the numerical methods for the fluid equations of motion.

1.3 Thesis Accomplishments and Outline

The major contributions of this dissertation are as follows

- Development of a new robust, quasi-conservative, contact-preserving interface and shock-capturing scheme for the simulation of compressible multiphase flows:
 - on unstructured meshes,
 - based on a two-phase Riemann problem,
 - applicable to any equation of state for which a one-dimensional two-phase Riemann problem can be solved.
- Examination of the applicability of sparse grids for the tabulation of Riemann invariants in order to reduce the computational costs of solving certain Riemann problems.
- Simulation of the collapse of an underwater air bubble and validation of the results by comparison with experimental data.
- Development of a provably second-order time-accurate staggered loosely-coupled fluid-structure solver in an Arbitrary-Lagrangian-Eulerian framework in order to allow for the simultaneous handling of multiphase flows and crack propagation.
- Verification of the order of time-accuracy of the fluid-structure solver.
- Assessment of the feasibility of conducting numerical simulations of imploding cylinders in the current computational framework.

The thesis is organized as follows. Chapter 2 defines the physical models for both the fluids and the structure and their mathematical properties with an emphasis on

the fluid modeling. Chapter 3 details the computational framework of the present work, where numerical schemes for the resolution of single-phase compressible flow and for the resolution of the structural equations of motion. While it does not address multiphase flows, chapter 3 also overviews the different methods available for numerically advecting an interface. In chapter 4, a new explicit compressible multiphase flow scheme on unstructured meshes is presented. Chapter 5 presents an algorithm that enables the simultaneous time-integration of the compressible multiphase equations and the structure equations of motion with crack propagation in an Arbitrary-Lagrangian-Eulerian framework. In chapter 6, the use of the compressible multiphase flow scheme presented in chapter 4 and that of the fluid-structure time-integrator presented in chapter 5 are assessed with applications to implosion problems. Lastly, chapter 7 provides conclusions as well as perspectives for future work.

Chapter 2

Physical Model and Mathematical Properties

The present chapter details the mathematical model of the physical problem at hands and its underlying assumptions. In that perspective, section 2.1 analyzes the physics of the implosion problem and the possible physical approximations through a clarification of the important phenomena that needs to be modelled and through a dimensional analysis. These assumptions are included in the resulting mathematical models for both the fluid and the structure, which are presented in section 2.2 while the specification of the material properties of both fluid and structure is given in section 2.3. Finally, section 2.4 reviews some fundamental mathematical properties of the fluid Euler model in order to define the mathematical framework of the present study.

2.1 Assumptions

The physical problem consists of the structural failure of a submarine hull –submerged in water – in the presence of nearby explosions and/or implosions. As mentioned in the first chapter, the goal of the present work is the corresponding failure analysis that must lead to the understanding of the mechanisms leading to that failure and to the knowledge of the corresponding critical conditions. Thus the problem is primarily characterized by a complex spatial and dynamic interaction between two subsystems that are the fluid and the structure. In order to model this interaction with fidelity, modelings of both the structure and the fluid appear necessary.

In the fluid, the problem is characterized by the propagation of acoustic waves in the water due to implosions and/or explosions. During the implosion of a bubble of air, the gas is first highly compressed, reaching its pressure peak when the bubble reaches its minimal volume, and then expands. During the explosion of materials such as TNT, the solid explosive is quickly detonated and the resulting gaseous products follow an expansion and then possibly a compression. In both cases, each gas can undergo several of those cycles and the expansion of the gas in one of those cycles results in pressure waves in the surrounding fluid, that is water here. The accurate propagation of these resulting pressure waves is key to the structure failure analysis. For all these reasons, it appears necessary to model all fluids as compressible fluids. In general, water is most commonly modeled by an incompressible fluid, but the violent phenomena that needs to be captured actually requires a compressible modeling of water. The violence of the phenomena also leads to large motions of the flow and the structure. While the flow may be mostly unperturbed far away from implosion/explosion bubbles and from the structure, the fluid undergoes large variations leading to fluid states ranging several orders of magnitude, therefore forbidding

the use of a purely acoustic model. In light of these characteristics of the flow, a complete modeling of the fluid seems necessary, with specifications of thermodynamic quantities and velocities in both space and time dimensions.

A dimensional analysis now proves useful to determine which effects must be taken into account in the fluid. Mach numbers in all types of fluids are expected to vary from 0 (or close to 0) to values of the order of 1, since compressibility effects are non-negligible. The time scale of the problem varies from a fraction of a millisecond to 100 milliseconds, while the characteristic length scale is of the order of 1 to 10 meters given the size of implodable volumes on actual submarines. In that time frame, heat and momentum diffusions are negligible as Peclet number - which measures heat advection with respect to heat diffusion - varies around 10^7 in air and 10^9 in water, and Reynolds number - which measures momentum advection with respect to viscous diffusion - takes values around 10^7 in air and 10^8 in water. Gravity may need to be modeled in some cases as the Froude number - which measures the ratio of inertial forces to gravity - takes values around 10 and below. In this case, buoyancy effects are taken into account allowing to model the rising of gas bubbles (with lower density than that of water). Given these values, the compressible Euler equations seem to form the most appropriate model for this physical problem, with the possible need to model gravity.

At the interface between any two fluids, effects such as evaporation and surface tension are present. However, the time scale of the physical problem of interest here is much smaller than the one of mass diffusion and the Schmidt number, which compares the effects of momentum and mass diffusions, is such that mass diffusion can be neglected. Due to the violence of the phenomena and the size of the bubble, the Weber number, which compares dynamic pressure to surface tension, has values

around 10^8 , suggesting that surface tension is negligible. This leads to a modeling of the interface by a contact discontinuity, or a free surface, meaning that any two fluids are immiscible. The contact discontinuity is characterized by density around 1,000 and initial pressure ratios varying from 10 to 10,000, if not more.

Modeling in space and time also seems necessary for the structure – due to the same reasons as the ones given for the fluid subsystem – and thus the structure is modeled in the framework of continuum mechanics. Since the time scale of heat conduction in the solid is expected to be substantially smaller than the one of mechanical response, the thermal and mechanical phenomena are uncoupled. Thus, only a purely mechanical problem needs to be solved. In addition, the small aspect ratio of the thickness of the structure and the size of the structure suggests the use of shell theory. Similarly to the fluid, the structure undergoes, in general, large displacements in the considered physical problem. Two possibly successive regimes can be distinguished. First, the structure deforms without failing anywhere, but large displacements are possible. Potentially, part of the structure can then fail leading to even greater displacements and changes in topology. This last part is beyond the scope of this work, as mentioned earlier. However, the displacements of the structure are large enough to induce a highly nonlinear behavior that must be modeled appropriately. For these reasons, a model handling nonlinear deformation of an inelastic material is required. In particular, the aluminum sheets that are used to build the structures considered in this work must be modeled with a plasticity constitutive law.

The interaction between these two subsystems is assumed to be purely mechanical. Indeed, given the time scale of the physical phenomena, the walls of the structure can be considered adiabatic. This also justifies why the thermal effects inside the structure are not modeled. The fluid-structure interaction is then modeled by a

dynamic equilibrium between the two at any point of the interface and at all times. This translates into a contact discontinuity and an equality of the forces on both sides, due to the principle of action and reaction.

Next, the mathematical models specific to each subsystem - fluid and structure - are presented.

2.2 Governing Equations

The governing equations for the fluid and the structure subsystems are presented in this section, as well as their interactions.

2.2.1 Fluid Governing Equations

Let $\Omega(\mathbf{x}, t) \subseteq \mathbb{R}^3 \times [0, \infty)$ be the flow domain of interest of a compressible fluid where \mathbf{x} and t denote respectively space and time coordinates. The behavior of this fluid is governed by the compressible Euler equations, which account for mass, momentum and energy conservations. Their strong conservative form can be written down as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot [\rho \mathbf{u}] = 0 \quad (2.1)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot [\rho \mathbf{u} \otimes \mathbf{u}] = -\nabla p \quad (2.2)$$

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot [(\rho e + p) \mathbf{u}] = 0 \quad (2.3)$$

where ρ , \mathbf{u} , p and e denote respectively the density, velocity vector, pressure and total energy per unit mass. The operator \otimes is the tensor product. The total energy is the sum of the internal energy and the kinetic energy $e = \epsilon + \frac{1}{2} \mathbf{u}^2$. In the remainder of the thesis, the compressible Euler equations will be mostly written in the following

compact vectorial form

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = 0$$

where the notations of the conservative variables

$$\mathbf{U} = (\rho, \rho \mathbf{u}, \rho e)^T \quad (2.4)$$

(where the superscript T denotes the transposition operation) and of the fluxes

$$\mathbf{F}(\mathbf{U}) = (\rho \mathbf{u}, \rho \mathbf{u} \otimes \mathbf{u} + p \mathbf{I}, (\rho e + p) \mathbf{u})^T \quad (2.5)$$

have been adopted. Additionally, the primitive variables

$$\mathbf{V} = (\rho, \mathbf{u}, p)^T$$

are introduced as they are more convenient to express physical states in some situations, such as boundary and initial condition specifications or Riemann problems.

The closure of the compressible Euler equations is obtained with an equation of state (EOS) which relates the thermodynamic variables ρ , p and ϵ together. In its most general form, it will be denoted by $\text{EOS}(\rho, p, \epsilon) = 0$. However, this work only focuses on EOS that have the explicit form $p = p(\rho, \epsilon)$.

2.2.2 Structure Governing Equations

It is beyond the scope of this work to study the physical behavior, the mathematical modeling and the mathematical framework of the structure governing equations.

However, since they represent one of two subsystems involved in any fluid-structure interaction problem, they are now briefly introduced.

Considering a structural domain of interest $\Omega_s(\mathbf{x}, t)$, the general equations governing the structure can be written [41]:

$$\rho_s \frac{\partial^2 \mathbf{u}_s}{\partial t^2} - \nabla_x \cdot \sigma_s(\mathbf{u}_s, \dot{\mathbf{u}}_s) = \mathbf{f}_s^{\text{ext}} \quad \text{in } \Omega_s(t)$$

where \mathbf{u}_s denotes the displacement field of the structure, $\dot{\mathbf{u}}_s$ the velocity field of the structure, ρ_s and σ_s its density and its Cauchy stress tensor respectively, $x \in \Omega_s(0)$ the position of the structure in its reference configuration, and $\mathbf{f}_s^{\text{ext}}$ the external forces acting on the structure. In the remainder of this work, dotted quantities represents the time derivative of the quantity considered. A constitutive law allows to express the stress tensor in terms of the displacement and velocity fields of the structure. Geometric nonlinear behavior of the structure is also modeled.

In the remainder of this work, unless specified otherwise, the subscript s will always refer to a structural field or quantity, at the continuous and the discrete level.

2.2.3 Transmission Conditions

The goal of this work is to study effects of the interactions between a fluid and a structure. The interactions between the two subsystems is entirely localized at their common interface $\Gamma(t) = \partial\Omega(\mathbf{x}, t) \cap \partial\Omega_s(t)$. Besides the general assumptions that the fluid domain and the structure domain do not interpenetrate and that the forces between the fluid and the structure are at equilibrium at all time, no restriction on the fluid-structure coupling is considered. In this case and considering the above framework for the fluid and structure equations, the Dirichlet and Neumann transmission

conditions between the fluid and the structure can be expressed as

$$\mathbf{u} \cdot \mathbf{n}_\Gamma = \dot{\mathbf{u}}_s \cdot \mathbf{n}_\Gamma \quad (2.6)$$

$$-p \mathbf{n}_\Gamma = \sigma_s(\mathbf{u}_s, \dot{\mathbf{u}}_s) \cdot \mathbf{n}_\Gamma \quad (2.7)$$

where \mathbf{n}_Γ denotes the unitary normal of the fluid-structure interface $\Gamma(t)$.

2.3 Thermodynamics, Equations of State and Constitutive Models

The governing equations presented in the above section are very general and do not make assumptions on the material behavior of each fluid or each structure. In the present section, different existing equations of state for the fluid subsystem and constitutive models for the structure subsystem are described as they will be used to characterize the respective material behaviors of both subsystems. A few notions on thermodynamics in general and entropy in particular are also reviewed in the framework of the compressible Euler equations.

2.3.1 Equations of State

The material properties of a fluid and thus its thermodynamic behavior are defined by an equation of state (EOS). However, solving the Euler equations requires only the partial specification of the thermodynamics of the material through an incomplete EOS. This incomplete EOS does not allow to recover all the thermodynamic quantities of a system, contrary to the complete EOS [67]. In general, the incomplete EOS is not necessarily derived from a thermodynamically consistent one. In fact, it sometimes

tries to reproduce some experimental results by appropriately choosing values of the coefficients of a given analytical formula. In the present work, four different EOS are considered to model different fluids:

- polytropic gas
- stiffened gas
- Jones-Wilkins-Lee gaseous products of high explosives
- Tait barotropic liquid

Each of these EOS will lead to different behaviors of the fluid.

Polytropic Gas (PG) This gas model comes from kinetic theory. It assumes that molecules do not interact with each other which is, for most real gas, a valid assumption at standard pressures and temperatures. Because of its simplicity and the historical study of aerodynamics, this EOS is well understood and hence very popular for gas dynamics. However, at high pressures or low temperatures, the underlying assumption may not be valid as molecules are closer together and start interacting with each other. It may be then necessary to switch to a real-gas EOS, which is beyond the scope of this work and consequently, gas such as air will always be modeled by a polytropic gas. Its EOS is

$$p = (\gamma - 1)\rho\epsilon$$

where γ is the constant ratio of specific heats. For air, γ is usually set to 1.4.

Stiffened Gas (SG) This EOS is very versatile as it has been used to model liquids and even solid materials, and to propagate waves in such stiff media [31]. It

is a common generalization of the polytropic gas one and has proven to be as simple to use in gas dynamics, see for e.g. [69]. Its EOS is

$$p = (\gamma - 1)\rho\epsilon - \gamma p_{sg}$$

where γ and p_{sg} are two constants. These constants are usually chosen such that a given shock Mach number is reproduced correctly by the EOS [43]. This equation is thermodynamically consistent as was shown by [67]. Usual numerical values of these constants vary from 4.4 to 7.15 for γ and from 6×10^8 Pa to 2×10^9 Pa for p_{sg} .

Jones-Wilkins-Lee (JWL) The detonation of high explosives such as TNT, results in gaseous products at very high pressures, temperatures and densities. It is followed by an expansion of those gaseous products and an explosion in the surrounding area. Polytropic gas and other traditional models fail to properly reproduce the thermodynamic behavior of these detonation products. The JWL EOS is the most common one for high explosives and their gaseous products [7]. The popularity of that EOS is partially due to its ability to reproduce experimental results on the explosive cylinder test, the primary source of data for high explosives EOS [18]. In this test, a copper cylinder is filled with explosive and detonated. The expansion profile of the gaseous products is recorded and used to calibrate parameters of the considered EOS. Therefore, one cannot expect results based on this EOS to be accurate or even valid in other types of situations, e.g. when shocks are present. The JWL EOS has the form

$$p = \omega\rho\epsilon + f(\rho)$$

A_1	=	3.712×10^{11} Pa
A_2	=	3.23×10^9 Pa
R_1	=	4.15
R_2	=	0.95
ω	=	0.28
ρ_0	=	1630.0 kg.m ⁻³

Table 2.1: Conventional parameter values for the JWL EOS when TNT explosive is considered.

where

$$f(\rho) = A_1 \left(1 - \frac{\omega\rho}{R_1\rho_0}\right) e^{-\frac{R_1\rho_0}{\rho}} + A_2 \left(1 - \frac{\omega\rho}{R_2\rho_0}\right) e^{-\frac{R_2\rho_0}{\rho}} \quad (2.8)$$

and ω , A_1 , A_2 , R_1 , R_2 and ρ_0 are constants. ρ_0 is the density of the explosive before detonation, which means that the density of the gaseous products cannot be expected to be more than ρ_0 . Note that this EOS is asymptotically similar to a polytropic gas EOS as density nears zero. When the explosive is TNT, conventional values for the constants of this EOS are given by table 2.1 and the range of validity of the equation (with these numerical parameters) is $\rho \in [0, \rho_0]$. Even though those values are most often used, [35] showed that calibration of the EOS's parameters using experimental underwater detonation of TNT led to slightly different values. However, in the present work, the first numerical values will be used for ease of comparison with other simulations.

Tait law This EOS [71] models compressible fluids whose bulk modulus is an affine function of pressure $k_1 + k_2 p = \rho \frac{dp}{d\rho}$ where the constants k_1 and k_2 are measured

experimentally for the considered fluid. The EOS can be written under the form

$$p = p_0 + \alpha \left(\left(\frac{\rho}{\rho_0} \right)^\beta - 1 \right) \quad (2.9)$$

where (p_0, ρ_0) is a given reference state and $\alpha = p_0 + \frac{k_1}{k_2}$ and $\beta = k_2$ are constants.

As for any barotropic law, it can only model isentropic fluids. Thus, for isentropic flows, this EOS makes perfect sense. However, the physical meaning of this model becomes unclear when shocks develop in the mathematical solution, as the physical entropy is assumed to be constant across the shock. This model still retains some validity for truly isentropic flows and for flows with weak shocks. In the presence of shocks, the stiffened gas model is better suited since the Tait law is recovered from the stiffened gas model when the fluid is assumed to be isentropic. However, its widespread use in the community of underwater explosions imposes to consider it for the present work.

The energy equation is decoupled from the mass conservation and the momentum equations. In this case, the internal energy of the fluid is given by

$$\epsilon - \epsilon_0 = \frac{-p_0 + \alpha + \frac{\alpha}{\beta - 1} \left(\frac{\rho}{\rho_0} \right)^\beta}{\rho}.$$

The energy equation can then be disregarded.

In general, values of k_1 and k_2 are $2.07 \times 10^9 \text{kg.m}^{-3}.\text{s}^{-2}$ and 7.15 respectively.

General form The first three EOS can all be written under the form

$$p = (\gamma - 1)\rho\epsilon - f(\rho) \quad (2.10)$$

where the function f is respectively 0, $-\gamma p_{sg}$ and Eq. (2.8) for the PG EOS, the SG EOS and the JWL EOS. In addition, in the case of the JWL EOS, γ is defined by $\gamma = \omega + 1$.

2.3.2 Fluid Thermodynamics

Thermodynamics involves the study of equilibrium states of a system whose characteristic properties are independent of space and time. The system is then determined by a few independent state variables and all other state variables can be expressed in terms of the independent ones using an equation of state.

In a moving fluid, variations to the equilibrium are assumed to be sufficiently small in order to use classical thermodynamic relations and results at any point in space and time of the flow. In addition, the thermodynamic state of the fluid is determined by the same state variables as the ones used in thermodynamics, the only difference being that the thermodynamics state variables are then considered as field functions. In that situation, the equations of state relating the different fluid state variables are identical to the ones encountered in classical thermodynamic. Thus all flow state quantities such as pressure, density, energy, enthalpy and entropy can be related to one another at any location of the fluid domain through the thermodynamic equation of state (EOS).

Among all thermodynamic quantities, entropy (denoted by s) has a particular place in the resolution of the Euler equations. As mentioned earlier, it is not always possible to express entropy without a thermodynamically consistent EOS that links energy ϵ , density ρ and entropy s . Moreover, such an EOS is not necessary to solve the Euler equations, as only an incomplete EOS [67] that links pressure p to density ρ and energy ϵ is needed. However, even though a complete EOS may not be available,

the entropy is not a complete unknown. Most importantly, the differential relation

$$T ds = d\epsilon + p d\left(\frac{1}{\rho}\right) \quad (2.11)$$

always holds. The second principle of thermodynamics states that the entropy of a fluid particle remains constant in a smooth Eulerian flow. This can be written as

$$\frac{\partial s}{\partial t} + u \cdot \nabla s = 0$$

or in combination with the mass conservation equation

$$\frac{\partial \rho s}{\partial t} + \nabla \cdot (\rho u s) = 0.$$

The thermodynamics of air, water and gaseous products of high explosives are all different and thus, several equations of state are needed to model all these fluids. The equations of state considered in this work can be cast under the general EOS (2.10). There is no isentropic form for this general EOS, but all three equations for the PG, SG and JWL do have one such form. For a fluid particle that undergoes an isentropic change from the state (p_a, ρ_a) to the state (p, ρ) , it can be shown using (2.11), that the two states satisfy

$$\frac{p}{\rho^\gamma} = \frac{p_a}{\rho_a^\gamma} \quad (2.12)$$

in the case of a fluid with the behavior of a polytropic gas,

$$\frac{p + p_s}{\rho^\gamma} = \frac{p_a + p_s}{\rho_a^\gamma} \quad (2.13)$$

in the case of a fluid with the behavior of a stiffened gas, and

$$\frac{p - A_1 e^{-\frac{R_1 \rho_0}{\rho}} - A_2 e^{-\frac{R_2 \rho_0}{\rho}}}{\rho^\gamma} = \frac{p_a - A_1 e^{-\frac{R_1 \rho_0}{\rho_a}} - A_2 e^{-\frac{R_2 \rho_0}{\rho_a}}}{\rho_a^\gamma} \quad (2.14)$$

in the case of a fluid with JWL EOS.

Under certain circumstances, physical discontinuities called shocks are known to appear in a fluid. They all satisfy an entropy condition given by the second principle of thermodynamics, that is the entropy of a fluid particle increases as it crosses the shock. From a mathematical point of view, the Euler equations (2.1)-(2.3) do not express any such condition on the entropy. Thus even if the Euler equations admit discontinuous solutions, they are not necessarily physically admissible. A new condition needs to be added to the mathematical model to ensure that only physically admissible solutions are considered. While in general hyperbolic systems of several dimensions, no condition ensures the existence and uniqueness of the solution and there is only little theory on the subject, all solutions need at least to satisfy the second principle of thermodynamics. For the one-dimensional Euler equations, if the equation of state is convex, it is sufficient to rely on that principle to obtain existence and uniqueness of the solution. However, in general, more stringent criterion need to be considered, such as the one requiring that realizable shock solutions should admit viscous profiles. More will be said in section 2.4.3 on the existence of discontinuous solutions for the Euler equations.

2.3.3 Constitutive Laws for a Structure

In order to solve the structural equations (2.6), it is necessary to specify both the kinematics (relation between strain and displacement) and the behavior of the structure material under the form of a constitutive law which accounts for the properties of the material that needs to be modeled. Since geometric nonlinearities are essential in this problem, the kinematics are given by

$$\epsilon_s = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T + \nabla \mathbf{u} \otimes \nabla \mathbf{u}^T)$$

where ϵ_s denotes Green's symmetric strain tensor in the solid. The constitutive law then relates the stress tensor to the strain of the structure. The present work is aimed at the structural failure analysis of specific thin-wall metallic structures, such as submarine hulls. An elastic constitutive law is first described. However, given that the displacements of the structure to be modeled are far from being small, this model is not sufficient to describe material properties of a metallic structure, even though geometrical nonlinearities may be taken into account. For that purpose, the J_2 -flow theory plasticity model is described. In addition, it is assumed that the material has an isotropic behavior.

Elasticity The isotropic elastic constitutive law is a linear law between the stress tensor and the strain tensor. It properly describes material behavior as long as no permanent strains appear upon unloading, which means that after a cycle of loading and unloading, the structure returns to its initial configuration. Two physical parameters must be specified to characterize the elastic constitutive law. The Young's modulus measures the stiffness of the material and the Poisson ratio measures the ratio of transverse contraction strain to longitudinal extension strain in the direction

of stretching force.

Plasticity In practice, the elastic behavior of the metallic material is limited to a small state-domain. When the structure stress reaches a yield strength, the material starts to deform plastically instead of elastically. Contrary to the case where the material is elastic, once the yield strength is reached, part of the deformation is permanent and irreversible. These permanent strains that appear upon unloading are called plastic strains. The unloading process is itself elastic with the same stiffness (given by the Young's modulus) as in the original loading. The following reloading is also elastic until the yield strength is reached. Besides, the development of the plastic strains may vary the elastic domain. In order to model this plastic behavior of the material, a rate-independent response of the material is considered. In particular, the J_2 -flow theory plasticity is used as it was specifically developed for metal plasticity [41]. In that case, two additional coefficients are necessary to characterize the material: the yield stress and the hardening modulus.

Experiments and simulations shown in chapter 6 involve cylinders made of aluminum (Al-6061-T6). This material is modeled by the J_2 -flow theory plasticity. The elastic behavior of this aluminum is characterized by a Young's modulus and a Poisson ratio with values of $10.08 \cdot 10^3$ ksi and 0.3 respectively (corresponding to 69.5 GPa and 0.3 in SI units). The plastic behavior is characterized by the yield strength and the hardening modulus with values of 40.08 ksi and 92.0 ksi respectively (corresponding to 275 MPa and 635 MPa in SI units). These numerical values are obtained experimentally and are standard values used in the industry.

2.4 Mathematical Properties

Detailed reviews of the mathematical properties of the fluid equations and of the structural equations are available in the literature. In particular for the fluid equations, the reader is referred to [23] or [72], where hyperbolicity of nonlinear systems, nonlinearity, linear degenerescence, Riemann problems and such mathematical properties are discussed. For the structural equations, the reader is referred, for e.g., to [80].

In the following, some key mathematical properties of the Euler equations are briefly reviewed. The purpose is to clearly define the mathematical framework in which the Euler equations are studied, that is the one of a hyperbolic system of nonlinear equations with genuinely nonlinear and linearly degenerate fields. This is the modeling for classical gas dynamics. For each equation of state, this results in different bounds to some of the physical variables. Discontinuous solutions and the Riemann problem are also briefly introduced.

As advocated by [67], key mathematical properties can be derived with the knowledge of only a few non-dimensional parameters which characterize the thermodynamic states of the fluid. Within this work, three such quantities will be used. They are the adiabatic exponent $\tilde{\gamma}$, the Gruneisen coefficient $\tilde{\Gamma}$ and the fundamental derivative of thermodynamics $\tilde{\mathcal{G}}$. They respectively measure the slopes of isentropes, the spacing between the isentropes in the $\log p$ - $\log \frac{1}{\rho}$ plane, and the convexity of the isentropes in

the p - ρ plane. Their general expressions are given by

$$\begin{aligned}\tilde{\gamma} &= \left. \frac{\rho}{p} \frac{\partial p}{\partial \rho} \right|_{\epsilon} + \left. \frac{1}{\rho} \frac{\partial p}{\partial \epsilon} \right|_{\rho} \\ \tilde{\Gamma} &= \left. \frac{1}{\rho} \frac{\partial p}{\partial \epsilon} \right|_{\rho} \\ \tilde{\mathcal{G}} &= - \left. \frac{1}{2} \frac{1}{\tilde{\gamma} p} \frac{\partial^2 p}{\partial \rho^2} \right|_s\end{aligned}$$

which in the case of a model EOS of the form (2.10) can be rewritten

$$\begin{aligned}\tilde{\gamma} &= \gamma - \frac{f(\rho) - \rho f'(\rho)}{p} \\ \tilde{\Gamma} &= \gamma - 1 \\ \tilde{\mathcal{G}} &= \frac{\gamma + 1}{2\tilde{\gamma}p} \left(\gamma p - f(\rho) + \rho f'(\rho) + \frac{\rho^2 f''(\rho)}{\gamma + 1} \right)\end{aligned}$$

These quantities have a determining role in characterizing the mathematical properties of the Euler equations.

2.4.1 Hyperbolicity of the Euler Equations

The three-dimensional conservative Euler equations (2.1)-(2.3) can be linearized to obtain a quasi-linear system of equations. This system is said to be hyperbolic if any linear combination of the Jacobians of the fluxes with respect to the variables, that is $\frac{d\mathbf{F}}{d\mathbf{U}} \cdot \mathbf{n}$, is diagonalizable for all \mathbf{n} in \mathbb{R}^3 and if the eigenvalues at any point in space, time and state are real. In the case of the three-dimensional Euler equations (see [23] for e.g.) , the Jacobian is diagonalizable with eigenvalues $\mathbf{u} \cdot \mathbf{n}$, $\mathbf{u} \cdot \mathbf{n} + c$ and $\mathbf{u} \cdot \mathbf{n} - c$ having multiplicity 3, 1 and 1 respectively, and c denoting the speed of sound. The speed of sound depends on the material properties of the medium and the state of

the fluid through the pressure p and the density ρ and for any EOS is given by

$$c^2 = \left. \frac{\partial p}{\partial \rho} \right|_s = \frac{\tilde{\gamma} p}{\rho}$$

which in the considered case here boils down to

$$c^2 = \frac{\gamma p - f(\rho) + \rho f'(\rho)}{\rho}. \quad (2.15)$$

Assuming that the thermodynamic quantities ρ and p are positive, the parameter $\tilde{\gamma}$ must be positive to ensure thermodynamic stability, which itself ensures that c^2 is positive. In the case of the polytropic gas EOS, the system of Euler equations is hyperbolic if

$$\rho > 0, \quad p > 0.$$

In the case of the stiffened gas EOS, the system of Euler equations is hyperbolic if

$$\rho > 0, \quad p > -p_s,$$

which means that pressure can take negative values.

In the case of the JWL EOS, the system of Euler equations is hyperbolic if

$$\rho > 0, \quad p > \frac{f(\rho) - \rho f'(\rho)}{\gamma}, \quad (2.16)$$

which means that pressure can take negative values as well, but there also exists a positive lower bound depending on the value of the density (see Figure 2.1) and therefore some positive values of the pressure are not acceptable.

In the case of the Tait EOS, the system of Euler equations is hyperbolic if

$$\rho > 0,$$

and the pressure p can take any value given by formula (2.9), including negative values. Note that for the first three equations of state, the bounds on density and pressure can be summarized by (2.16), choosing the appropriate function for $f(\rho)$.

In the remainder of this work, the study of the Euler equations is restricted to the aforementioned bounds on the density and pressure, so that the Euler equations can always be considered a hyperbolic system.

Remark When the fluid is modeled by a JWL EOS or a SG EOS, the hyperbolicity of the Euler equations allows for negative values of pressure. It does not mean either that physical values of the pressure should be negative. However, it has been shown that fluids such as water can withstand tensile stress, which can only be seen as negative pressures in the frame of the Euler equations. However the function of the EOS considered in this work does not include the modeling of tensile stress. Thus, negative pressures within the aforementioned bounds were allowed in this work in order to preserve conservation properties of the Euler equations, but they should not necessarily be considered as adequate representation of tensile stress.

2.4.2 Genuinely Non-Linear and Linearly Degenerate Fields

For the sake of simplicity, only the one-dimensional Euler equations are considered now under the general form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0, \quad x \in \mathbb{R}, \quad t > 0$$

where $\mathbf{U} = (\rho, \rho u, \rho e)^T$ and $\mathbf{F} = (\rho u, \rho u^2 + p, (\rho e + p)u)^T$. Linearization of the system gives

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A} \frac{\partial \mathbf{U}}{\partial x} = 0$$

where $\mathbf{A} = \partial \mathbf{F} / \partial \mathbf{U}$ is the Jacobian of the flux function and depends on \mathbf{U} . As for any gas dynamics system - composed of the Euler equations and an equation of state - the Jacobian has increasing eigenvalues $(\lambda_1, \lambda_2, \lambda_3) = (u - c, u, u + c)$. It is diagonalizable and there exists a matrix of right eigenvectors $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ such that

$$\mathbf{R}^{-1} \mathbf{A} \mathbf{R} = \mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \lambda_3).$$

Characteristic variables \mathcal{W} can then be defined by $d\mathcal{W} = \mathbf{R}^{-1} d\mathbf{U}$ and a change of variables leads to the characteristic form of the Euler equations

$$\frac{\partial \mathcal{W}}{\partial t} + \mathbf{\Lambda} \frac{\partial \mathcal{W}}{\partial x} = 0.$$

Note that only the differentials of the characteristic variables are known. In general, the characteristic variables of a hyperbolic are known only when the coefficients of the

matrix A are constant. The decomposition of the one-dimensional Euler equations into characteristic equations leads to three fields, each being associated with a scalar equation and an eigenvalue: two nonlinear acoustic fields, labelled the 1-field and the 3-field as they are associated with the eigenvalues $\lambda_1 = u - c$ and $\lambda_3 = u + c$, and an entropy field, labelled the 2-field and associated with the eigenvalue $\lambda_2 = u$.

The characteristic fields of a system of nonlinear equations can be parted in three categories: genuinely nonlinear fields, linearly degenerate fields, and the others. Genuinely nonlinear fields satisfy the property

$$\nabla \lambda_k(\mathbf{U}) \cdot \mathbf{r}_k(\mathbf{U}) \neq 0$$

for all admissible \mathbf{U} , while linearly degenerate fields satisfy the property

$$\nabla \lambda_k(\mathbf{U}) \cdot \mathbf{r}_k(\mathbf{U}) = 0$$

for all admissible \mathbf{U} .

The genuine nonlinearity of the 1- and 3-fields is directly related to the sign ¹ of the fundamental derivative of thermodynamics

$$\nabla \lambda_k(\mathbf{U}) \cdot \mathbf{r}_k(\mathbf{U}) = \pm \frac{1}{c} \tilde{\mathcal{G}} \quad \text{for } k \in \{1, 3\}$$

where a positive sign is used for the left-facing wave ($k = 1$) and a negative sign for the right-facing wave ($k = 3$). The two acoustic fields being genuinely nonlinear ensures that the nonlinear waves issued from the Riemann problem are special waves

¹When the fundamental derivative of thermodynamics is positive, shocks are compressive and rarefactions are expansive. In the case when it is negative, shocks become expansive and rarefactions compressive.

called shocks and rarefactions, that will be defined in the next sections. Otherwise, solutions of the Riemann problem may include mixt and/or split waves. The study of these types of waves is beyond the scope of this work, and therefore only the (p, ρ) -domain such that $\tilde{\mathcal{G}} > 0$ is considered. In the case of a polytropic gas or a stiffened gas, acoustic waves are genuinely nonlinear. While acoustic waves are in general not genuinely nonlinear when the JWL EOS is considered on the (p, ρ) -domain defined by the bounds (2.16), they are genuinely nonlinear if an appropriate restriction of the (p, ρ) -domain is considered, that is

$$p > \frac{1}{\gamma} \left(f(\rho) - \rho f'(\rho) - \frac{\rho^2 f''(\rho)}{\gamma - 1} \right) \quad (2.17)$$

so that the mathematical framework remains the same as for the other EOS ².

The 2-wave is always linearly degenerate. In particular, in the solution of a Riemann problem, it is a contact discontinuity characterized by a jump in density and continuity of both velocity and pressure.

The physical model and its mathematical framework are now restricted to ranges of pressures and densities such that the system of Euler equations is hyperbolic with genuinely nonlinear waves (and linearly degenerate entropy waves as is always the case for the Euler equations). For example, in the case of the JWL EOS, the conditions (2.16) and (2.17) that must be satisfied by the pressure are plotted on Figure 2.1. The two conditions on the adiabatic exponent and the fundamental derivative of thermodynamics above have been more generally studied by [34, 36]. They define the general classical gas dynamics framework, where $\tilde{\gamma}$ and $\tilde{\mathcal{G}}$ are both positive.

²For the JWL EOS, the previous bounds on pressure and density given by the positivity of c^2 do not ensure that the new bounds are satisfied. In general, there is no relation of one bound implying the other.

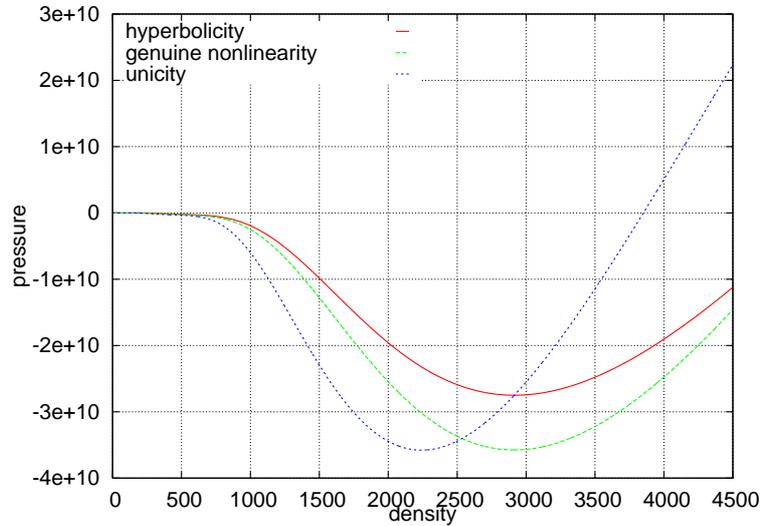


Figure 2.1: Different bounds on the pressure with respect to the density for the JWL EOS to ensure hyperbolicity, genuine non-linearity and unicity of the solution to the Riemann problem.

2.4.3 Existence of Discontinuous Solutions

The strong formulation (2.1)-(2.3) of the Euler equations assumes that the solution \mathbf{U} is continuous and differentiable. However discontinuous solutions of the flow exist and in some cases can eventually develop even when the flow solution is originally smooth. The differential equations are not valid in the classical sense for such solutions. However, the mathematical theory of partial differential equations (cf [23] for e.g.) provides a framework in which such discontinuous solutions are possible and are called weak solutions of the hyperbolic system. We now briefly explain what conditions these weak solutions must satisfy.

Jump conditions at a discontinuity in the flow must satisfy the so-called Rankine-Hugoniot conditions

$$\mathbf{F}(\mathbf{U}_2) - \mathbf{F}(\mathbf{U}_1) = \sigma (\mathbf{U}_2 - \mathbf{U}_1) \quad (2.18)$$

where \mathbf{U}_1 and \mathbf{U}_2 are the two states on each side of the discontinuity and σ is the velocity of the discontinuity.

As proven by Lax, weak solutions have the property of satisfying these jump conditions. This has led to the development of conservative schemes that mimic the properties of the integral form of the hyperbolic systems such as Godunov-type finite volume methods. It should be noted however that among the weak solutions, only some of them are physically relevant.

It is possible to distinguish between two types of discontinuities: contact discontinuities and shocks.

Contact discontinuities are characterized by continuity of the pressure and of the normal velocity across the discontinuity and by jumps in density and tangential velocities.

Across a shock, a particle suddenly sees all its physical properties change. Thus, in the case of a shock and not a contact discontinuity, the equations (2.18) lead to a scalar equation relating the two states on both sides of the shock discontinuity:

$$\epsilon_2 - \epsilon_1 = -\frac{1}{2}(p_2 + p_1) \left(\frac{1}{\rho_2} - \frac{1}{\rho_1} \right). \quad (2.19)$$

Given a physical state \mathbf{U}_1 , the jump conditions allow to define a family of acceptable states \mathbf{U}_2 described by the Hugoniot curve. However, in order to get a physical solution, there is no general criterion that must be met by shock solutions. One of the most stringent condition is that the shock solution should admit a viscous profile as a limit of the Navier-Stokes equations when viscosity vanishes, but it does not always allow to choose a single physical solution. Indeed there is only little theory on this active research topic. However, in the classical gas dynamics framework, satisfying the entropy condition inferred from the second principle of thermodynamics

is enough to select physical solutions. In addition, the Bethe-Weyl theorem shows that the entropy condition corresponds to shock waves being compressive, that is the state behind the shock wave has higher pressure. As a consequence, only a branch of the Hugoniot curve leads to physically relevant states. Thus, a fluid particle crossing a shock wave has its entropy increased, or equivalently in classical gas dynamics, it has its pressure increased.

2.4.4 Rarefaction Waves

Rarefaction waves are another type of particular waves that can be studied in isolation in the sense that they are single waves from only one of the characteristic families. Contrary to contact discontinuities and shock waves, they are characterized by the facts that they are continuous solutions connecting two constant states, that they are associated with only one characteristic, and that they are self similar. This translates into a solution of the form:

$$\mathbf{U}(x, t) = \begin{cases} \mathbf{U}_L & x \leq \xi_L t \\ \tilde{\mathbf{U}}(x/t) & \xi_L t < x < \xi_R t \\ \mathbf{U}_R & \xi_R t \leq x \end{cases}$$

where $\tilde{\mathbf{U}}$ is a continuous solution of the Euler equations and satisfies $\tilde{\mathbf{U}}(\xi_L) = \mathbf{U}_L$ and $\tilde{\mathbf{U}}(\xi_R) = \mathbf{U}_R$. Each rarefaction wave corresponds to a characteristic genuinely nonlinear field and its associated eigenvalue. In order to be a solution of the Euler equations, $\tilde{\mathbf{U}}$ must satisfy an ordinary differential equation. The integration of these ODEs lead to constant quantities of the flow in the rarefaction wave, the so-called *Riemann invariants* of that characteristic field. However, it is not always possible to integrate these ODEs.

The Riemann invariants of a genuinely nonlinear field always include the entropy, and can be written under the ODE form:

$$\frac{dp}{d\rho} = c^2(p, \rho)$$

For the one-dimensional Euler equations, the other Riemann invariant has the following ODE form:

$$\frac{du}{d\rho} = \mp \frac{c(p(\rho), \rho)}{\rho}$$

where the sound speed c depends eventually of only the density ρ since the entropy is constant and thus $p(\rho)$ is given by the first Riemann invariant. The minus sign corresponds to a 1-Riemann invariant (and eigenvalue $u - c$), while the plus sign corresponds to a 3-Riemann invariant (and eigenvalue $u + c$).

In the case of a SG EOS, the equality of the Riemann invariants between two states \mathbf{V}_2 and \mathbf{V}_1 can be written as

$$\frac{p_2 + p_s}{\rho_2^\gamma} = \frac{p_1 + p_s}{\rho_1^\gamma} \quad (2.20)$$

$$u_2 \mp \frac{2}{\gamma - 1} c_2 = u_1 \mp \frac{2}{\gamma - 1} c_1 \quad (2.21)$$

where c_k is the sound speed corresponding to the state \mathbf{V}_k , $k = 1, 2$. Note that the first of these equations is the same as equation (2.13).

In the case of the JWL EOS, only the first ODE can be integrated leading to equation (2.14)

$$\frac{p_2 - A_1 e^{-\frac{R_1 \rho_0}{\rho_2}} - A_2 e^{-\frac{R_2 \rho_0}{\rho_2}}}{\rho_2^\gamma} = \frac{p_1 - A_1 e^{-\frac{R_1 \rho_0}{\rho_1}} - A_2 e^{-\frac{R_2 \rho_0}{\rho_1}}}{\rho_1^\gamma}$$

while the second Riemann invariant for the 1- and 3-characteristics cannot be expressed analytically. It becomes necessary to compute it in an approximate fashion, for e.g. by numerically integrating the ODE or by integrating an approximation of the integrand [58], [68].

2.4.5 Review of the One-Dimensional Riemann Problem

The Riemann problem for the one-dimensional Euler equations consists in finding the solution $\mathbf{U}(x, t)$ to

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0, \quad x \in \mathbb{R}, \quad t > 0$$

$$\mathbf{U}(x, 0) = \begin{cases} \mathbf{U}_L & , \quad x < x_0 \\ \mathbf{U}_R & , \quad x > x_0 \end{cases}$$

where \mathbf{U}_L and \mathbf{U}_R correspond to the equivalent primitive initial conditions given by \mathbf{V}_L and \mathbf{V}_R respectively.

In the context of classical gas dynamics, the mathematical study of hyperbolic systems for gas dynamics shows that the solution is composed of three waves that separate constant states. The left-facing wave (or 1-wave) and the right-facing wave (or 3-wave) are either rarefactions or shocks and the middle wave (or 2-wave) is a contact discontinuity. States left of the 1-wave and right of the 2-wave are given by initial conditions. The two constant states on both sides of the contact discontinuity differ only by their densities, since velocity and pressure are constant across a contact.

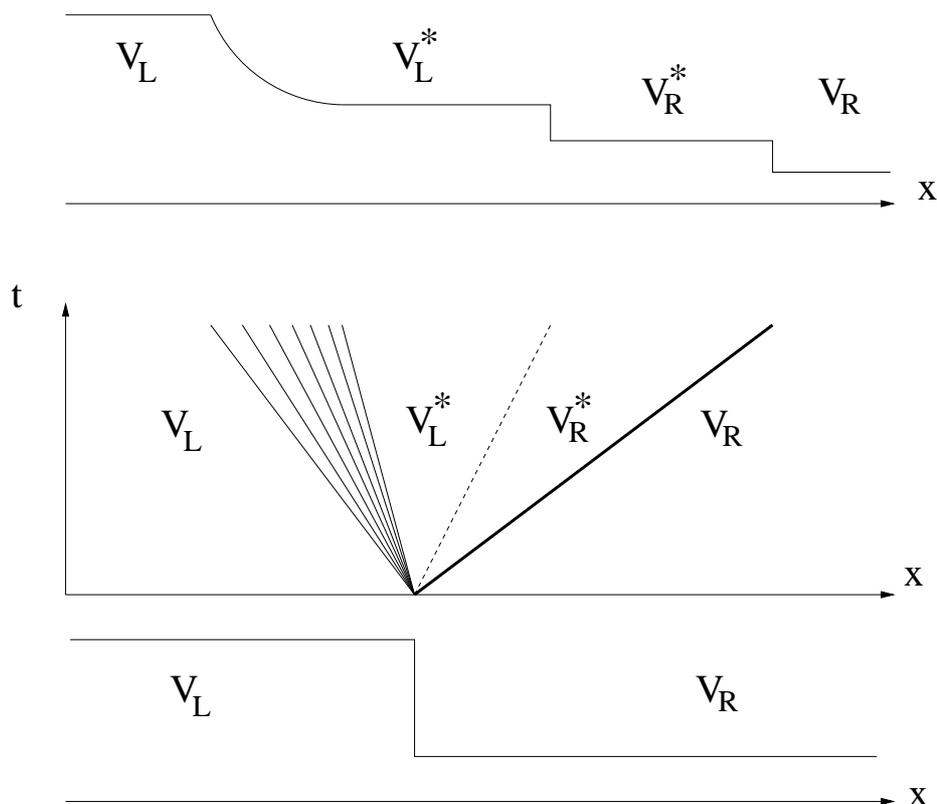


Figure 2.2: Initial Riemann problem (bottom), structure of the solution (middle) with a rarefaction on the left, a shock on the right and a contact discontinuity in the middle separating the two interfacial states, and solution at a specific time (top)

The two constant states can be denoted

$$\mathbf{V}_L^* = (\rho_R^*, u^*, p^*)$$

$$\mathbf{V}_R^* = (\rho_L^*, u^*, p^*)$$

Figure 2.2 shows the structure of one such possible solution.

In the cases of the PG EOS and of the SG EOS, the Bethe-Weyl theorem states that there always exists a unique solution to the Riemann problem, provided that the

formation of voids is allowed. The unicity of the solution of the Riemann problem (if it exists) is related to the sign of the fundamental derivative of thermodynamics. In the case of the JWL, it is not possible to prove that there always exists a unique solution since the fundamental derivative is positive only in a limited range of the state-space. However, in our range of interest $0 < \rho < \rho_0$, if there exists a solution, it is unique since $\tilde{\mathcal{G}} > 0$ in that range as shown in Fig. 2.1, where the blue curve represents the bound $\tilde{\mathcal{G}} = 0$. Solutions that lie outside that domain are not correctly described by this EOS and one cannot expect to have physically relevant results. In the modeling of expansions of gaseous products by that EOS (which was calibrated for that purpose), density and pressure will decrease, falling within the bounds of our model.

In order to solve for the constant states \mathbf{V}_L^* and \mathbf{V}_R^* (or equivalently \mathbf{U}_L^* and \mathbf{U}_R^*), several algorithms are available. To solve for the SG EOS and the PG EOS, the reader is referred to [29] and to solve for the JWL EOS, an algorithm similar to the one given by [50] was used. For the sake of consistency of this work, detailed algorithms are given in Appendix A.

Chapter 3

Computational Framework

3.1 Introduction

In the present chapter, the respective computational frameworks for a single-phased fluid and for a structure are presented separately without consideration for the interfaces between two fluids or between a fluid and a structure. The specific numerical treatment at the interfaces will be studied in chapters 4 and 5. However, some general ideas are outlined here since the choices for the interface treatment can be intrinsically connected to the choices for the numerical methods for the single-phased fluid and for the structure.

As described in the previous chapter, the physical problem involves two systems, the fluid system and the structure system. In addition to the physical phenomena specific to each one of them, the systems are strongly interacting with each other. Indeed, the instantaneous position of the structure at least partially determines the fluid domain boundaries. Flow features are then affected as well as the fluid load on the structure. Problems with such moving boundaries between fluid and structure

are of great concern in many scientific and engineering applications such as biomedical, civil, mechanical and aerospace engineering. Examples include aircraft flutter, stability of suspension bridges and blood flow in arteries. Fluid and structure systems must then be solved simultaneously. Given the complexity of the global system and the different mathematical properties of both the fluid and structure systems that have led to different mature numerical methods, the fluid and the structure systems (and the chosen corresponding numerical methods) are first studied separately in the present chapter.

The fluid system alone presents added complexity due to the presence of at least water and the gas within an implodable volume. It becomes necessary to distinguish if a point of the domain of interest is in one fluid or another, knowing that the material interface between the different fluids is completely dependent on the flow behavior from both fluids. Similar to the material interface between a fluid and a structure, the material interface between two fluids defines a certain boundary, and is not known a priori since it is part of the solution. In addition, the advection of the material interface by the flow can lead to complex changes of its topology such as mergers and break-ups. Problems with such two-phase flows are of great interest to the scientific and engineering communities. Shockwave lithotripsy is an example of such a problem in the medical field, cavitation damage is a well-known problem in the naval industry, the optimization of mixing of two different fluids is also a common problem in areas as different as the food and the auto industries. To avoid dealing with the issues pertaining to two-phase flows immediately, the fluid system considered in this chapter involves a single fluid.

The study of problems involving interfaces, either between two fluids or between a fluid and a structure, has led to numerous algorithms depending on the nature

and the characteristics of the interface. For example, fluid-structure interactions involving only infinitesimal displacements of the interface have been solved with the transpiration technique. With the development of numerical algorithms and of computing power, problems with larger displacements of material interfaces have been considered leading to several approaches, where the tracking or capturing of the interface is required. Most fall in either one of three categories: moving mesh methods, where the mesh follows the interface, level sets (including embedded methods) and front-tracking methods, which give the position of the interface on a given grid, and mixture algorithms, where the interface is not given by any direct quantity, but rather captured. On the one hand, moving mesh methods include the popular ALE method, the closely-related dynamic mesh method, the co-rotational approach, the space-time formulation. Advantages of some of these methods and of the ALE method in particular come from the high-order accuracy that can be achieved in both space and time and the treatment of the boundary conditions. This also includes the possibility of high mesh resolution for boundary layers in viscous flows. On the other hand, the other classes of methods such as level set, front-tracking and mixture methods belong to the purely Eulerian framework and have been used successfully to applications in free-surface flows, two-phase flows, combustion, solidification, as well as computer vision and image processing. The way to treat the interface can then be done in numerous ways for both fluid-fluid and fluid-structure interactions. Even though the application of boundary conditions is complex and it is then difficult to obtain higher-order accuracy, these Eulerian methods present the advantage of handling larger deformations than the ALE methods, as well as topological changes which are not easily possible with moving mesh methods.

Considering that a Lagrangian description will be used for the structure and given

that the interest in this work is to simulate the first stages of an implosion problem where cracking is yet to occur as explained in the first chapter, an ALE approach is used in the fluid to treat the fluid domain deformations due to the structure. In addition to that, tracking of the interface between two fluids on the ALE moving mesh is done by considering a level set method. An unstructured mesh is adopted for the representation of the fluid as it allows for a fairly easy meshing of the fluid domain in the presence of structures with complex geometries. The use of an unstructured mesh also allows for a better behavior of the mesh motion when the mesh deformations are not known a priori. These choices allow the use of some proven numerical methods in both the single-phase fluid and in the structure.

In this chapter, the first section is dedicated to the presentation of the Arbitrary-Lagrangian-Eulerian framework. The finite volume semi-discretization and the time discretization used for a single-phase flow are then presented in sections 3.3 and 3.4. Even though mesh motion is an important aspect of the ALE framework for practical purposes, its presentation is postponed until chapter 5. The ALE framework requires the knowledge of certain quantities given by the mesh, but not the mesh itself. Section 3.5 introduces the level set function and its discretization, while its use for the numerical treatment of the interface in two-phase flow problems will be detailed in the next chapter. Finally, the structural equations are semi-discretized using a Finite Element approach with integration in time with an explicit central difference scheme or a midpoint rule scheme.

3.2 Arbitrary-Lagrangian-Eulerian Framework

Structural deformations are crucial to the flow resolution in the applications considered in the present work. However, structure and fluid problems are most often solved using completely different approaches. The most classical approach to solve a structural problem involves a Lagrangian description while an Eulerian description of the flow is most convenient for fluid problems. Both descriptions are two classical descriptions of motions in the algorithms of continuum mechanics, but are fundamentally different in the point of view they adopt to look at the continuum. In Lagrangian methods which are most often used in structural mechanics, material particles are tracked during their motion by the algorithm and are followed by the computational mesh. In Eulerian methods, the computational mesh is fixed and captures the motion of the continuum allowing for large distortions, hence explaining its widespread use in computational fluid dynamics. Contrary to front-tracking or embedded methods which can make use of the Lagrangian description for the structure and the Eulerian description for the fluid, the ALE description [39] combines these two previous descriptions, meaning that any node of a computational mesh can be arbitrarily chosen to be fixed, as in the Eulerian description, or to be moved as in a Lagrangian description but without necessarily following any fluid particles. The ALE formulation is therefore well-suited to flow problems with moving/deforming boundaries. Indeed, when flow resolution is heavily impacted by structural deformations, the flexibility of moving the computational domain in an arbitrary way allows to combine some of the best features of both approaches. Moving the boundaries with the accuracy characteristic of Lagrangian methods and solving the flow with the same ease as Eulerian methods allow for handling deformation and resolution requirements.

However, several limitations need to be recognized. First of all, the mesh has

to be modeled and a specific description of the mesh motion is required which adds complexity and incurs some additional computational cost of the algorithm. More importantly, mesh motion robustness can become an issue when element distortions and entanglements are likely to happen, meaning that large changes in the computational domain are not always possible and small time-steps might be necessary. Break-up of the computational domain is an extreme case which cannot be handled with ALE methods, explaining why the use of an ALE description is more widespread in fluid-structure interaction than in two-phase flow computations for instance. All these problems are inherent to the method. In cases such as gas compression in a combustion chamber where the motion of the piston is often assumed to be a given parameter of the problem, it is possible to avoid some of these difficulties as the mesh motion is given a priori. In general, however, the motion of the mesh is not known a priori and an ALE approach is highly dependent on the mesh motion algorithm. Without regard to the mesh motion, other issues also arise in numerical schemes derived from ALE formulation of conservation equations. ALE-formulated Euler equations retain the hyperbolic character of the classical Euler equations, and methods used to semi-discretize Euler equations on fixed grids can be reused, with minor changes, for semi-discretization of ALE-formulated Euler equations. Even though their semi-discretization does not generate more problems than the fixed-grid counterpart, time integration on moving meshes is far from being a trivial task. Besides leading to a loss of accuracy, an arbitrary extension of a fixed-grid time integrator to solve flow problems on a moving grid can also generate instabilities in some situations. Thus some stability and accuracy requirements need to be addressed in the design of time integrators on moving meshes. While these issues must be carefully considered, ALE methods can be and have been designed to be high-order space- and time-accurate.

This is not the case for all of the aforementioned methods.

In the next two sections, finite volume semi-discretization of the Euler equations on moving grids and the time-integration of those semi-discrete equations are presented.

3.3 Finite Volume Semi-Discretization of the Fluid Governing Equations for a Single-Phase Flow

Let $\Omega_x(t)$ be the flow domain of interest and $\Gamma(t)$ its moving and possibly deforming boundary. The instantaneous configuration $\Omega_x(t)$ - where grid point space coordinates are denoted by x and time is denoted by t - and the reference configuration $\Omega_\xi(\tau)$ - where grid point space coordinates are denoted by ξ and time is denoted by τ - are mapped by the following mapping function

$$\begin{cases} x &= x(\xi, \tau) \\ t &= \tau \end{cases} .$$

Using the same notations as in chapter 2, the ALE strong form of the Euler equations is written as

$$\begin{aligned} \frac{\partial J\rho}{\partial t}\Big|_\xi + J\nabla_x \cdot [\rho(\mathbf{u} - \dot{\mathbf{x}})] &= 0 \\ \frac{\partial J\rho\mathbf{u}}{\partial t}\Big|_\xi + J\nabla_x \cdot [\rho\mathbf{u} \otimes (\mathbf{u} - \dot{\mathbf{x}})] &= -J\nabla_x p \\ \frac{\partial J\rho e}{\partial t}\Big|_\xi + J\nabla_x \cdot [\rho e(\mathbf{u} - \dot{\mathbf{x}}) + p\mathbf{u}] &= 0 \end{aligned}$$

where $J = \det\left(\frac{\partial x}{\partial \xi}\right)$ is the Jacobian of the mapping and $\dot{\mathbf{x}} = \frac{\partial x}{\partial t}\Big|_\xi$ denotes the velocity of the instantaneous configuration. In the special case of a barotropic fluid modeled

by the Tait EOS, the energy equation can be disregarded since the energy is given directly from the values of the other quantities of the flow (see section 2.3.1).

In any case, the system can be written under vector form as follows:

$$\left. \frac{\partial J\mathbf{U}}{\partial t} \right|_{\xi} + J \nabla_x \cdot \mathbf{F}(\mathbf{U}, \dot{\mathbf{x}}) = 0$$

where

$$\mathbf{F}(\mathbf{U}, \dot{\mathbf{x}}) = \mathbf{F}(\mathbf{U}) - \dot{\mathbf{x}}\mathbf{U}$$

is the convective ALE flux and $\mathbf{F}(\mathbf{U})$ and \mathbf{U} have been defined by Eqs. (2.4) and (2.5).

Spatial discretization of these equations is as follows. The computational domain of interest Ω is subdivided in a set of non-overlapping control volumes. In the present work, Ω is discretized by a tetrahedral mesh from which a median dual mesh (cf Figure 3.1) is derived to obtain the control volumes $\{\Omega_i\}$. The unknowns of each control volume are located at the vertices of the tetrahedral mesh. A finite volume formulation is used to semi-discretize these equations on the aforementioned vertex-centered mesh.

Integrating this equation over a reference control volume $\Omega_i(0)$ in the ξ -space, leads to

$$\frac{d}{dt} \int_{\Omega_i(0)} J\mathbf{U} d\Omega_{\xi} + \int_{\Omega_i(0)} J \nabla_x \cdot \mathbf{F}(\mathbf{U}, \dot{\mathbf{x}}) d\Omega_{\xi} = 0$$

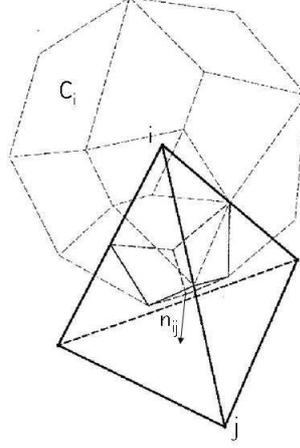


Figure 3.1: Control volume (lighter lines) in an unstructured tetrahedral (heavier lines) mesh (only one of the tetrahedra needed to construct the graphically depicted control volume is shown).

A change of variables from the reference configuration to the instantaneous configuration (x -space) transforms it to

$$\frac{d}{dt} \int_{\Omega_i(t)} \mathbf{U} d\Omega_x + \int_{\Omega_i(t)} \nabla_x \cdot \mathbf{F}(\mathbf{U}, \dot{\mathbf{x}}) d\Omega_x = 0$$

The use of Green's theorem on the second term leads to

$$\frac{d}{dt} \int_{\Omega_i(t)} \mathbf{U} d\Omega_x + \int_{\partial\Omega_i(t)} \mathbf{F}(\mathbf{U}, \dot{\mathbf{x}}) \cdot \mathbf{n}_i(t) ds = 0$$

where $\mathbf{n}_i(t)$ is the outward unitary normal to the control surface $\partial\Omega_i(t)$ of the control volume $\Omega_i(t)$ at any point on that control surface.

For any node i , $\mathcal{N}(i)$ denotes the set of nodes connected to node i and $\partial\Omega_{ij}(t) = \partial\Omega_i(t) \cap \partial\Omega_j(t)$ denotes the common boundary between the two connected nodes i

and j . For an interior node i , the second term can then be split over all the neighbors of node i

$$\frac{d}{dt} \int_{\Omega_i(t)} \mathbf{U} d\Omega_x + \sum_{j \in \mathcal{N}(i)} \mathbf{F}_{ij}(\mathbf{U}, \mathbf{x}, \dot{\mathbf{x}}) = 0$$

where

$$\mathbf{F}_{ij}(\mathbf{U}, \mathbf{x}, \dot{\mathbf{x}}) = \int_{\partial\Omega_{ij}(t)} \mathbf{F}(\mathbf{U}, \dot{\mathbf{x}}) \cdot \mathbf{n}_{ij}(t) ds$$

and $\mathbf{n}_{ij}(t)$ is the unitary normal to $\partial\Omega_{ij}(t)$, both of which depend on \mathbf{x} .

If \mathbf{U} is known everywhere, it is then possible to exactly update the space averaged value of \mathbf{U} in a given control volume, that is for the control volume Ω_i around node i

$$\mathbf{U}_i = \frac{1}{|\Omega_i(t)|} \int_{\Omega_i(t)} \mathbf{U} d\Omega_x$$

where $|\cdot|$ denotes the measure of the geometric quantity (\cdot) .

However, at the discrete level, the flux terms are then approximated by a numerical flux function Φ that is conservative and consistent with the continuous flux function

$$\mathbf{F}_{ij}(\mathbf{U}, \mathbf{x}, \dot{\mathbf{x}}) \approx |\partial\Omega_{ij}(t)| \Phi(\mathbf{U}_i, \mathbf{U}_j, \nu_{ij}(t), \kappa_{ij}(t))$$

where $\nu_{ij}(t)$, $\kappa_{ij}(t)$ are the unitary normal to the control surface $\partial\Omega_{ij}(t)$ and its normal velocity. They are defined by

$$\nu_{ij}(t) = \frac{1}{|\partial\Omega_{ij}(t)|} \int_{\partial\Omega_{ij}(t)} \mathbf{n}_{ij}(t) ds \quad (3.1)$$

$$\kappa_{ij}(t) = \frac{1}{|\partial\Omega_{ij}(t)|} \int_{\partial\Omega_{ij}(t)} \dot{\mathbf{x}} \cdot \mathbf{n}_{ij}(t) ds. \quad (3.2)$$

In general, the averaged normals ν_{ij} and the averaged normal velocities κ_{ij} are nonlinear functions of the grid point positions [1]. In particular, the vector of the normals ν can be written as $\nu(\mathbf{x})$, where $\nu(\cdot)$ is a nonlinear function.

In the present work, Roe's approximate Riemann solver [51] is used to compute the numerical flux and the numerical flux function Φ is defined by

$$\Phi(\mathbf{U}_i, \mathbf{U}_j, \nu_{ij}, \kappa_{ij}) = \frac{\mathbf{F}(\mathbf{U}_i, \nu_{ij}, \kappa_{ij}) + \mathbf{F}(\mathbf{U}_j, \nu_{ij}, \kappa_{ij})}{2} - d(\mathbf{U}_i, \mathbf{U}_j, \nu_{ij}, \kappa_{ij}) \quad (3.3)$$

where the flux function $\mathbf{F}(\mathbf{U}, \nu, \kappa)$ is

$$\mathbf{F}(\mathbf{U}, \nu, \kappa) = \mathbf{F}(\mathbf{U}) \cdot \nu - \kappa \mathbf{U},$$

$d(\mathbf{U}_i, \mathbf{U}_j, \nu, \kappa)$ is the upwinding term. The upwinding term is originally given by

$$d(\mathbf{U}_i, \mathbf{U}_j, \nu, \kappa) = |\mathcal{A}(\mathbf{U}_i, \mathbf{U}_j, \nu, \kappa)| (\mathbf{U}_j - \mathbf{U}_i)$$

where \mathcal{A} is the Jacobian $\frac{\partial \mathbf{F}}{\partial \mathbf{U}}$ of the flux function evaluated at the Roe-averaged quantities. However, in order to overcome the numerical difficulties encountered by this compressible flow solver in the low Mach regime [28, 17], the upwinding term is modified when the equations of state considered is either a stiffened gas equation of state or the Tait barotropic liquid equation of state. The original upwinding term is then replaced by

$$d(\mathbf{U}_i, \mathbf{U}_j, \nu, \kappa) = P^{-1} |P \mathcal{A}(\mathbf{U}_i, \mathbf{U}_j, \nu, \kappa)| (\mathbf{U}_j - \mathbf{U}_i)$$

where P is a preconditioning matrix [27, 16, 26]. For a three-dimensional flow where

the fluid follows the stiffened gas equation of state, the preconditioner takes the form

$$P = \frac{\partial \mathbf{U}}{\partial \mathbf{W}} \begin{pmatrix} \beta^2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \frac{\partial \mathbf{W}}{\partial \mathbf{U}}$$

where $\mathbf{W} = (p, \mathbf{u}, s)^T$. For a three-dimensional flow where the fluid follows the Tait equation of state, as the energy equation does not need to be considered the preconditioner takes the form

$$P = \frac{\partial \mathbf{U}}{\partial \mathbf{W}} \begin{pmatrix} \beta^2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \frac{\partial \mathbf{W}}{\partial \mathbf{U}}$$

where $\mathbf{W} = (p, \mathbf{u})^T$. The preconditioning matrix is controlled by the parameter β which is adequately chosen to recover the proper behavior of the pressure at low Mach number regimes. Its expression is given by

$$\beta^2 = \min \left(\max \left(M^2, \beta_{min}^2, \frac{2|p_j - p_i|}{p_i + p_j} \right), 1.0 \right)$$

similar to what was proposed in [27, 22]. In a supersonic flow, the preconditioner is removed, while a minimum value is specified in order to avoid getting a singular preconditioner. The minimum value is usually taken as the inflow Mach number for external flow. Due to the presence of shock waves and strong rarefactions even

in subsonic flows and to avoid smearing them, it becomes important to include a pressure-gradient term that turns off the preconditioning in these regions.

The semi-discretized equation for any interior node i can hence be rewritten as

$$\frac{d(\Omega_i \mathbf{U}_i)}{dt} + f_i(\mathbf{U}, \nu, \kappa) = 0$$

where

$$f_i(\mathbf{U}, \nu, \kappa) = \sum_{j \in \mathcal{N}(i)} |\partial\Omega_{ij}(t)| \Phi(\mathbf{U}_i, \mathbf{U}_j, \nu_{ij}(t), \kappa_{ij}(t))$$

Second-order accuracy is achieved through a Monotonic Upwind Scheme for Conservation Laws (MUSCL) approach [91]. The values \mathbf{U}_i and \mathbf{U}_j used in the numerical flux function (3.3) are respectively replaced by reconstructed values \mathbf{U}_{ij} and \mathbf{U}_{ji} of \mathbf{U} at the boundary of the control volumes $\partial\Omega_{ij}$,

$$\mathbf{U}_{ij} = \mathbf{U}_i + \frac{1}{2} (\nabla \mathbf{U})_{ij} \cdot \overrightarrow{x_i x_j} \quad (3.4)$$

$$\mathbf{U}_{ji} = \mathbf{U}_j + \frac{1}{2} (\nabla \mathbf{U})_{ji} \cdot \overrightarrow{x_j x_i} \quad (3.5)$$

where x_k denotes the coordinates of node k in the instantaneous configuration. On unstructured meshes, different gradient computations are possible. In the present work, a simplified least squares linear reconstruction is considered. The gradient $(\nabla \mathbf{U})_{ij}$ is based on a least-square nodal gradient around node i [2] and a central difference gradient along the edge $i - j$. The one-dimensional Van Albada limiter is used [90]. Given two gradients (or finite differences) of a smooth solution, the Van Albada limiter returns the mean of the two quantities, whereas the solution tends to the smallest quantity of the two where the solution is not smooth. Thus, it prevents

central differencing across a discontinuity in the solution or in its first derivative, which would lead to numerical oscillations. It also prevents undesirable clipping of a smooth extremum.

Open flow boundary conditions are imposed via the normal flux method [42] which is particularly well suited for approximations in the Finite Volume setting as it computes directly the normal flux at the boundary. Its generality allows to use it for all the equations of state studied in the present work. The boundary condition at a wall is given by the equality of the normal velocities of the fluid and of the structure as given by Eq. (2.6). It is numerically enforced via the flux

$$\Phi_w(\mathbf{U}, \nu_\Gamma, \kappa_\Gamma) = (0, p\nu_\Gamma, p\kappa_\Gamma)^T$$

where \mathbf{U} is the space averaged value in the considered cell and Γ is the interface between the wall (or structure) and the considered cell.

3.4 Time Integration of Semi-Discrete Equations on Moving Grids

This section focuses on the ALE extensions of time integration schemes employed for integrating the semi-discrete equations on moving grids, which in the case of a node i in the interior of the fluid domain can be written as

$$\frac{d(\Omega_i \mathbf{U}_i)}{dt} + f_i(\mathbf{U}, \nu, \kappa) = 0$$

This equation involves both the position and velocity of the underlying fluid dynamic mesh through the values of ν and κ . In the case of a moving and possibly

deforming mesh, it is not obvious which mesh configuration(s) should be used to integrate the equation between two times t^n and t^{n+1} . Indeed, the proper computation of $\int_{t^n}^{t^{n+1}} f_i(\mathbf{U}, \nu(t), \kappa(t)) dt$ raises the question of where to integrate the convective fluxes: on mesh configuration x^n (at t^n), or on x^{n+1} (at t^{n+1}), or somewhere else, or on a combination of all those configurations? For a given fixed-grid time integrator, arbitrary choices of those mesh configurations for a corresponding moving-grid time integrator may not lead to ALE extensions that retain the order of accuracy of the fixed grid one [6]. Moreover, it was shown in some cases that the computational solutions obtained with such time integrators exhibit weak instabilities [54]. Even when such issues do not arise with an ad-hoc procedure, one cannot assume that it will not happen in general. These issues were addressed in a series of papers. In particular, the nonlinear stability of these schemes was linked to the enforcement of the Discrete Geometric Conservation Law (DGCL) for Nonlinear Scalar Conservation Law (NSCL) [13]. One interpretation of a given time-integrator satisfying its DGCL is that constant states of the flow are preserved. A general methodology is now available [61, 12] for designing robust ALE extensions of a fixed-grid multi-step implicit time integrator for the solution of unsteady flow problems on dynamic grids. The newly designed time integrators retain the accuracy of the original time integrators and preserve the nonlinear stability in the sense of the discrete maximum principle of their fixed-grid counterparts.

As mentioned in chapter 1, two stages of the implosion of a structure must be considered, namely before any cracking of the structure where the air and the water are separated by a structural interface and after the cracking where the two fluids are in direct contact. More generally, two cases are distinguished depending on whether two fluids are in direct contact. As it will be seen later, the numerical time integration

of the fluid equations of motion in the presence of a fluid-fluid interface considered only an explicit scheme. However, when no such interface is present, an implicit time-integrator is preferred for stability reasons and computational costs. In what follows, extensions on moving grids of the second-order explicit SSP Runge-Kutta scheme [76, 75] and of the second-order three-point-backward-difference [61] scheme are presented. Both these schemes present the advantage of being second-order time-accurate on moving grids and of satisfying their DGCL. This last feature will be crucial to the two-phase flow integration as will be seen in the next chapter and in particular in section 4.6.

3.4.1 Second-Order Runge-Kutta

The fixed-grid 2nd-order accurate Runge-Kutta scheme was extended to a moving-grid second-order accurate Runge-Kutta one in [75]. It was shown to satisfy both nonlinear stability and second-order accuracy requirements. Whereas it is usually said that the DGCL is a sufficient condition for nonlinear stability in the sense of the discrete maximum, it is here emphasized that enforcing the DGCL guarantees that constant solutions of the flow are preserved. The scheme is as follows

$$\Omega_i^{n+1} \mathbf{U}_i^{n+1} = \Omega_i^n \mathbf{U}_i^n + \frac{\Delta t}{2} (k_1 + k_2)$$

where

$$\begin{aligned} k_1 &= -\mathbf{f}_i(\mathbf{U}^n, \bar{\nu}, \bar{\kappa}) \\ k_2 &= -\mathbf{f}_i\left(\frac{\Omega_i^n \mathbf{U}^n + \Delta t k_1}{\Omega_i^{n+1}}, \bar{\nu}, \bar{\kappa}\right) \end{aligned}$$

and

$$\begin{aligned}
 \bar{\nu} &= \sum_{k=1}^{K_f} \alpha_k \nu(x_f^{(k)}) \\
 \bar{\kappa} &= \sum_{k=1}^{K_f} \alpha_k \mathbf{w}^{(k)} \cdot \nu(x_f^{(k)}) \\
 x_f^{(k)} &= \zeta_k^{n+1} x^{n+1} + \zeta_k^n x^n + \zeta_k^{n-1} x^{n-1} \\
 \mathbf{w}^{(k)} &= \frac{\theta_k^{n+1} x^{n+1} + \theta_k^n x^n + \theta_k^{n-1} x^{n-1}}{\Delta t}.
 \end{aligned}$$

The explicit second-order Runge-Kutta scheme on moving grids used in the present work is described by $K_f = 2$ and

$$\alpha_1 = \alpha_2 = \frac{1}{2}$$

$$\begin{aligned}
 \zeta_1^{n+1} &= \frac{1}{2} \left(1 + \frac{1}{\sqrt{3}} \right), & \zeta_1^n &= \frac{1}{2} \left(1 - \frac{1}{\sqrt{3}} \right), & \zeta_1^{n-1} &= 0, \\
 \zeta_2^{n+1} &= \frac{1}{2} \left(1 - \frac{1}{\sqrt{3}} \right), & \zeta_2^n &= \frac{1}{2} \left(1 + \frac{1}{\sqrt{3}} \right), & \zeta_2^{n-1} &= 0,
 \end{aligned}$$

$$\theta_1^{n+1} = 1, \quad \theta_1^n = -1, \quad \theta_1^{n-1} = 0,$$

$$\theta_2^{n+1} = 1, \quad \theta_2^n = -1, \quad \theta_2^{n-1} = 0$$

3.4.2 Three-Point Backward Difference

The multistage time integration schemes have been studied by [61] and [12]. The second-order time-accurate three-point backward difference integration scheme on moving grids presented below has the advantage of requiring a single flux computation on a unique computational mesh configuration (as opposed to several flux computations on different computational mesh configurations) [6]. Also, in the same way that no stability criterion exists for the time integration of the Euler equations

using the explicit second-order Runge-Kutta time-integrator on moving grids, there exists no stability criterion for their time integration with the present three-point backward difference scheme on moving grids. Hence, while the present scheme is unconditionally stable in the linear case, it is not necessarily unconditionally stable for the Euler equations. However, it is observed that time-steps larger than the ones with the explicit second-order Runge-Kutta time-integrator are possible. In the present work, the implicit time integration of the Euler equations is done as follows

$$\frac{a_{n+1}\Omega_i^{n+1}\mathbf{U}_i^{n+1} + a_n\Omega_i^n\mathbf{U}_i^n + a_{n-1}\Omega_i^{n-1}\mathbf{U}_i^{n-1}}{\Delta t^n} + \mathbf{f}_i(\mathbf{U}_i^{n+1}, \bar{\nu}, \bar{\kappa}) = 0$$

where

$$a_{n+1} = \frac{1 + 2\tau}{1 + \tau}, \quad a_n = -1 - \tau, \quad a_{n-1} = \frac{\tau^2}{1 + \tau}, \quad \tau = \frac{\Delta t^n}{\Delta t^{n+1}}$$

and

$$\begin{aligned} \bar{\nu} &= \sum_{k=1}^{K_f} \alpha_k \nu(x_f^{(k)}) \\ \bar{\kappa} &= \sum_{k=1}^{K_f} \alpha_k \mathbf{w}^{(k)} \cdot \nu(x_f^{(k)}) \end{aligned}$$

The different configurations $(x_f^{(k)}, \mathbf{w}^{(k)})$ are given by

$$\begin{aligned} x_f^{(k)} &= \zeta_k^{n+1}x^{n+1} + \zeta_k^n x^n + \zeta_k^{n-1}x^{n-1} \\ \mathbf{w}^{(k)} &= \frac{\theta_k^{n+1}x^{n+1} + \theta_k^n x^n + \theta_k^{n-1}x^{n-1}}{\Delta t} \end{aligned}$$

The three-point backward difference implicit scheme on moving grids used in the present work is described by $K_f = 4$ and

$$\alpha_1 = \alpha_2 = \frac{3}{4}, \quad \alpha_3 = \alpha_4 = -\frac{1}{4}$$

$$\begin{aligned} \zeta_1^{n+1} &= \frac{1}{2} \left(1 + \frac{1}{\sqrt{3}} \right), & \zeta_1^n &= \frac{1}{2} \left(1 - \frac{1}{\sqrt{3}} \right), & \zeta_1^{n-1} &= 0, \\ \zeta_2^{n+1} &= \frac{1}{2} \left(1 - \frac{1}{\sqrt{3}} \right), & \zeta_2^n &= \frac{1}{2} \left(1 + \frac{1}{\sqrt{3}} \right), & \zeta_2^{n-1} &= 0, \\ \zeta_3^{n+1} &= 0, & \zeta_3^n &= \frac{1}{2} \left(1 + \frac{1}{\sqrt{3}} \right), & \zeta_3^{n-1} &= \frac{1}{2} \left(1 - \frac{1}{\sqrt{3}} \right), \\ \zeta_4^{n+1} &= 0, & \zeta_4^n &= \frac{1}{2} \left(1 - \frac{1}{\sqrt{3}} \right), & \zeta_4^{n-1} &= \frac{1}{2} \left(1 + \frac{1}{\sqrt{3}} \right), \\ \theta_1^{n+1} &= \theta_2^{n+1} = 1, & \theta_1^n &= \theta_2^n = -1, & \theta_1^{n-1} &= \theta_2^{n-1} = 0, \\ \theta_3^{n+1} &= \theta_4^{n+1} = 0, & \theta_3^n &= \theta_4^n = 1, & \theta_3^{n-1} &= \theta_4^{n-1} = -1 \end{aligned}$$

Finally, the flux term can be linearized, leading to

$$\begin{aligned} &\left(a_{n+1} \Omega_i^{n+1} + \Delta t^n \frac{\partial \mathbf{f}_i}{\partial \mathbf{U}}(\mathbf{U}_i^n, \bar{\nu}, \bar{\kappa}) \right) (\mathbf{U}_i^{n+1} - \mathbf{U}_i^n) = \\ &- (a_n \Omega_i^n \mathbf{U}_i^n + a_{n-1} \Omega_i^{n-1} \mathbf{U}_i^{n-1}) - \Delta t^n \mathbf{f}_i(\mathbf{U}_i^n, \bar{\nu}, \bar{\kappa}) \end{aligned}$$

This linear system is solved using a Newton-Krylov approach [93] in which the matrix-vector product of the Jacobian and of the state difference is approximated by finite differencing the second-order fluxes. The computational cost of one iteration of this implicit scheme, mostly due to the solving of a linear system, is much more than the one with the second-order Runge-Kutta scheme. However, due to its unconditional stability, the present scheme allows for much larger time-steps and can hence become overall computationally less expensive.

3.5 Level Set Equation

As mentioned earlier, methods involving moving meshes that follow an interface have been proposed and successfully used on some problems. However, these types of methods are highly dependent on the robustness of the mesh motion. Whereas this type of algorithm can be efficient for certain types of fluid-structure interactions, it does not realistically allow the simulation of flows of two immiscible fluids, as these flows can be such that large displacements and/or distortions of the interface between the two fluids appear. Assumptions on the particular behavior of the interface such as spherical symmetry have led to some successful computations [74], but the distortion of the interface is then always small. Without these assumptions, general motion of the interface must be considered with not only large distortions but also mergers and/or break-ups of the interface and a moving mesh method is not suitable.

Other methods have been developed that embrace a purely Eulerian description of the flow. Among them, the front-tracking approach is similar to what is done in fluid-structure interaction to track the structure. This method has been applied to the multiphase flow problem [43, 40] where the interface has no “material significance”. The method discretizes the interface with an unstructured moving mesh that is transported in a Lagrangian fashion. The benefit of a Lagrangian scheme is offset by the frequent remeshings that are necessary to represent the interface accurately and by the inability of the method to handle topological changes automatically.

One of the most popular methods in incompressible fluid dynamics to numerically transport an interface is the Volume-Of-Fluid method (VOF) [20]. It uses volume fractions to reconstruct the interface and hence ensures discrete mass conservation. However the need for a specific geometric advection scheme limits its accuracy and its time step size. In addition, the normals to the interface are not easily computed as the

interface is not given and needs to be reconstructed. All these reasons make the VOF a computationally expensive method. In addition to that, its extension to compressible fluid mechanics does not seem obvious, as another layer of complexity is added in order to recover the density field. Other classes of methods do not numerically transport an interface and do not reconstruct this interface. Mixture and/or averaging methods [69, 49] are such examples. Some of these methods consists in writing the fluid equations for each phase, each phase having its own density, momentum and energy, and then in averaging the equations. Hence, in one dimensional space, the system to solve for a two-phase flow is composed of 6 equations. While these methods obviate the need for a very local description of the flow and retain the possibility of solving flows with two highly different phases, some of these methods introduce new terms in the equations lead to modeling and numerical difficulties, as the equations are not hyperbolic anymore (see [4] and the references therein). This is especially true at the interface where highly nonlinear source terms appear. These issues can be alleviated by introducing another equation, which leads to the strictly hyperbolic Baer-Nunziato models. However, an additional layer of complexity is added as 7 equations need then be solved (see [4] and the references therein). In both cases with 6 or 7 equations in one dimension, the computational cost remains a major drawback. Simpler techniques introduce a volumetric fraction to average the densities of the two fluids in mixed cells, whereas velocities and pressures can be each described by a single field for both phases. However, an “averaged” equation of state needs to be considered in mixed cells. This can prove very difficult when the equations of state of each fluid are very different. Another example of methods that do not numerically transport an interface per say, and do not reconstruct the interface, was proposed in [64]. It had been noticed that conservative schemes for two-phase flows

were not able to preserve a contact discontinuity [77]. To tackle this issue when both fluids are modeled by a stiffened gas equation of state (but with different equation of state parameter values), the parameters cannot be assumed to be constant in each phase. Instead, combinations of these parameters must satisfy advection equations. However, it applies only to two-phase flows where both fluids follow a stiffened gas equation of state. Even if it is possible to apply this method to other equations of state, it is much less obvious to apply it to two-phase flows where the two fluids have different types of equation of state.

Another class of methods to numerically transport an interface are level set methods [78, 44], which represent an interface implicitly by a smooth function. Any iso-value of the level set can represent an interface. However, the interface of interest is traditionally represented by the zero iso-value. Its wide-spread use is due to some of its mathematical properties. The smooth function allows for the computation of geometric quantities related to the interface such as normal to and curvature of the interface. Depending on the application field, the level set satisfies different dynamic equations. In the present problem, the interface is advected and the zero level-set (that implicitly describes the interface) must satisfy the advection equation where the advection velocity is given by the flow velocity at the interface. It is then convenient to advance the level set field ϕ in the entire domain using the equation

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \quad (3.6)$$

where \mathbf{u} is the velocity of the flow. As a consequence, the handling of topological changes mentioned earlier is done naturally and efficiently. In addition, numerical tools already abound in the literature to solve this equation and its conservative

counterpart

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \mathbf{u} \phi) = 0 \quad (3.7)$$

(where \mathbf{u} and ρ denote respectively the velocity and the density of the fluid) and parallelization is straightforward and efficient. For all these reasons, the level set method was chosen to numerically detect the interface between two immiscible fluids in the present problem. However, it must be noticed that while topology changes are handled naturally, the dynamic creation of interfaces is not easy. This can happen for instance when cavitation occurs and a new gaseous phase is created in a liquid phase. For the modeling of this type of problems, it may be best to consider another of the aforementioned methods.

The level set function ϕ is most often chosen to initially be the signed distance function to the interface Γ ,

$$\phi(\mathbf{x}, t) = \pm |\mathbf{x} - \min_{s \in \Gamma} \mathbf{x}(s)|.$$

In all simulations of the present work, the level set was directly computed since only geometric objects such as spheres were considered. The interface is therefore described by $\phi = 0$ and one of the two fluids lies in the domain where the level set is negative while the other lies in the domain where the level set is positive. Note however that the advection of a level-set equation is the source of some problems. First, the mathematical problem of the advection of the level set is ill-posed due to the lack of a boundary condition. Second, the initial profile of the level set field is not preserved by the advection equation. Consequently, the level set can lose its smoothness in some cases and the computation of the normal to the interface can

become inaccurate. A reinitialization procedure [30] is used to recover a distance function allowing to treat both these issues. Other reinitialization procedures are available in the literature [44, 55]. Since the algorithm presented in the next chapter does not require the distance to the interface, the level set is reinitialized only when the gradient of the level set near the interface becomes either too flat or too steep.

Given that the level-set is written in conservative form and that it is solved for on the same moving grid as the Euler equations, the ALE computational framework of the Euler equations presented in the previous two sections can be applied to the numerical solution of the level-set, and hence is not presented again. The only difference is that the velocity intervening in the level-set equation is provided by the Euler equations.

3.6 Finite Element Semi-Discretization of the Structure Governing Equations

The equation of motion governing the structure is recalled here:

$$\rho_s \frac{\partial^2 \mathbf{u}_s}{\partial t^2} - \nabla_x \cdot \sigma_s(\mathbf{u}_s, \dot{\mathbf{u}}_s) = \mathbf{f}_s^{\text{ext}} \quad \text{in } \Omega_s(t) \quad (3.8)$$

where \mathbf{u}_s denotes the displacement field of the structure, $\dot{\mathbf{u}}_s$ its velocity field, ρ_s and σ_s its density and its stress tensor respectively, $x \in \Omega_s(0)$ the position of the structure in its reference configuration, and $\mathbf{f}_s^{\text{ext}}$ the body forces acting on the structure. The kinematics are given by

$$\epsilon_s = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T + \nabla \mathbf{u} \otimes \nabla \mathbf{u}^T)$$

and the material is either considered to be elastic or plastic, in which case a J_2 -flow plasticity theory is used.

Semi-discretization of this equation by finite elements (FE) [84, 83] is performed. First a weak form of Eq. (3.8) is obtained. While the solution of the problem belongs to a given space, approximate solutions of the problem are restricted on a subspace which defines the representation of the solution on a discretized domain. Hence, when the domain Ω_s is subdivided in non-overlapping elements Ω_s^e , nodal displacements are represented by functions of the form

$$\mathbf{u}(\mathbf{x}, t) = \sum_I \mathbf{u}_I(t) \mathbf{N}_I(\mathbf{x})$$

where the u_I are nodal values and $\mathbf{N}_I(\mathbf{x})$ are called the shape functions and depend on the subspace chosen to approximate the solution. A weak formulation associated with a Galerkin projection leads to a system of ordinary differential equations of second order in time with the very general form

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{f}_s^{\text{int}}(\mathbf{u}(t), \dot{\mathbf{u}}(t)) = \mathbf{f}_s^{\text{ext}}(t) \quad (3.9)$$

where \mathbf{M} denotes the FE mass matrix, \mathbf{u} the generalized displacement vector, $\mathbf{f}_s^{\text{int}}$ the vector of internal forces, $\mathbf{f}_s^{\text{ext}}$ the vector of remaining external forces. All the work presented in subsequent chapters does not assume the use of any particular type of element. For that reason, any further description of the numerical method for the semi-discretization of the structural equations is omitted.

3.7 Time Discretization of the Structure Governing Equations

A large number of time integration algorithms have been developed to solve both linear and nonlinear structural dynamics problems with different properties in terms of not only stability, accuracy, and computational cost, but also dispersion, dissipation, overshoot and starting conditions.

Two time-integration schemes for the structural equation (3.9) are used in the present work and are presented in this section, the explicit central difference scheme and the midpoint rule. Their consideration is driven by the implosion problem and the approach proposed in chapter 1 to model such problems. The modelisation of crack initiation and propagation has been developed at Northwestern University. The time integration of the structure equations of motion with cracking has been done with an explicit central difference scheme. While the explicit scheme allows for the treatment of cracks, it requires smaller time steps and is eventually computationally more expensive. For this reason, an implicit scheme is also considered as long as cracking does not occur.

3.7.1 Explicit Central Difference Scheme

The explicit central difference scheme stands among the most popular explicit and conditionally stable methods in computational mechanics and physics for its computational efficiency, its ease of implementation and its robustness. While time integration algorithms that perform well for linear structural dynamics are not necessarily well suited for nonlinear dynamic problems, the explicit central difference scheme proves to be mostly unaffected whether geometric and material nonlinearities are considered. In

terms of accuracy, it satisfies the minimum requirement of at least second-order time-accuracy for structural dynamics [38] as first-order accurate methods have typically excessive dissipation. At the same time, the considered problems are not smooth as they involve contact and fracture phenomena, and in that regard, higher-order methods are not the most suited algorithms. The nature of the problems considered in this work also dictate a small time-step in order to resolve the timescales of interest and the time variations of the solution, thus the stability constraint of the central difference scheme is not expected to be a considerable drawback. However, whereas the conditional stability of the scheme for linear systems is dictated by $\omega\Delta t < 2$ where ω can be considered as the highest frequency of the undamped mechanical system and is given by an eigenvalue analysis of individual elements in the finite element semi-discretization [88] –meaning that its value depends on the mesh size–, such stability criterion does not exist for nonlinear dynamic problems and the linear stability does not guarantee the stability of the algorithm for solving nonlinear problems. This is true for most time integrators.

In the following, the explicit central difference scheme is presented as given in [84]. Damping is modeled by a linear viscous force so that the considered semi-discrete equations are

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{f}_s^{\text{int}}(\mathbf{u}(t)) = \mathbf{f}_s^{\text{ext}}(t)$$

where \mathbf{C} is the damping matrix. Note that the method is not strictly speaking explicit unless the mass matrix \mathbf{M} is diagonal. It is then necessary to inverse a small local matrix. Lumping is adopted here to avoid this matrix inversion. As the equations are integrated from time t^n to t^{n+1} , the velocity $\dot{\mathbf{u}}(t^{n+\frac{1}{2}})$ and the acceleration $\ddot{\mathbf{u}}(t^n)$

are approximated by $\mathbf{u}^{n+\frac{1}{2}}$ and \mathbf{u}^n respectively via central difference:

$$\begin{aligned}\mathbf{u}^{n+1} &= \mathbf{u}^n + (t^{n+1} - t^n) \dot{\mathbf{u}}^{n+\frac{1}{2}} \\ \dot{\mathbf{u}}^{n+\frac{1}{2}} &= \dot{\mathbf{u}}^{n-\frac{1}{2}} + \left(t^{n+\frac{1}{2}} - t^{n-\frac{1}{2}}\right) \ddot{\mathbf{u}}^n\end{aligned}$$

where $t^{n+\frac{1}{2}} = \frac{1}{2}(t^{n+1} + t^n)$ and $\mathbf{u}(t^n)$ is approximated by \mathbf{u}^n . Thus, the discrete equations can be written as

$$\mathbf{M} \frac{\dot{\mathbf{u}}^{n+\frac{1}{2}} - \dot{\mathbf{u}}^{n-\frac{1}{2}}}{t^{n+\frac{1}{2}} - t^{n-\frac{1}{2}}} + \mathbf{C} \dot{\mathbf{u}}^{n-\frac{1}{2}} = \mathbf{f}^n$$

where $\mathbf{f}^n = \mathbf{f}_s^{\text{ext}}(t^n) - \mathbf{f}_s^{\text{int}}(\mathbf{u}(t^n))$.

The algorithm can then be written as follows:

- given \mathbf{u}^n and $\dot{\mathbf{u}}^n$ and $\dot{\mathbf{u}}^{n-\frac{1}{2}}$, compute $\ddot{\mathbf{u}}^n = \mathbf{M}^{-1} \left(\mathbf{f}^n - \mathbf{C} \dot{\mathbf{u}}^{n-\frac{1}{2}} \right)$
- update the nodal velocities $\dot{\mathbf{u}}^{n+\frac{1}{2}} = \dot{\mathbf{u}}^n + (t^{n+\frac{1}{2}} - t^n) \ddot{\mathbf{u}}^n$
- update nodal displacements $\mathbf{u}^{n+1} = \mathbf{u}^n + (t^{n+1} - t^n) \dot{\mathbf{u}}^{n+\frac{1}{2}}$
- compute $\ddot{\mathbf{u}}^{n+1}$
- update nodal velocities $\dot{\mathbf{u}}^{n+1} = \dot{\mathbf{u}}^{n+\frac{1}{2}} + (t^{n+1} - t^{n+\frac{1}{2}}) \ddot{\mathbf{u}}^{n+1}$

As mentioned in the previous chapter, the thermal and mechanical phenomena are assumed to be decoupled. The conservation of energy can then be used to monitor how well the evolution of the mechanical system is treated. This specific implementation allows to proceed to an energy balance check, which allows to detect some nonlinear instabilities that may not be obvious in view of displacement and velocity results.

Note also that the need to factorize the non-diagonal damping matrix was obviated by computing $\mathbf{C}\dot{\mathbf{u}}^{n-\frac{1}{2}}$ instead of $\mathbf{C}\dot{\mathbf{u}}^n$ and without changing the overall accuracy of the algorithm [89, 33].

3.7.2 Midpoint Rule

Similar to the explicit central difference scheme, the midpoint rule [87] is popular among structural dynamicists. It presents, for the most part, the same advantages as the explicit central difference scheme except that it is unconditionally stable for linear problems and that it is implicit. Hence, contrary to the central difference time-integration scheme, a full linear system must be solved at each iteration – possibly more than one full linear system if the structure has a nonlinear behavior. The equilibrium of the structure is enforced at the half-time step $t^{n+\frac{1}{2}}$ to advance the solution from t^n to $t^{n+1} = t^n + \Delta t$

$$\mathbf{M}\ddot{\mathbf{u}}(t^{n+\frac{1}{2}}) + \mathbf{f}_s^{\text{int}}(\mathbf{u}(t^{n+\frac{1}{2}})) = \mathbf{f}_s^{\text{ext}}(t^{n+\frac{1}{2}}) \quad (3.10)$$

while the relations between the approximations to the displacements, velocities and accelerations at the different time-steps and at the half-time-step are

$$\mathbf{u}^{n+\frac{1}{2}} = \frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2} \quad (3.11)$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \dot{\mathbf{u}}^{n+\frac{1}{2}} = \frac{\dot{\mathbf{u}}^{n+1} + \dot{\mathbf{u}}^n}{2} \quad (3.12)$$

$$\ddot{\mathbf{u}}^{n+\frac{1}{2}} = \frac{\dot{\mathbf{u}}^{n+1} - \dot{\mathbf{u}}^n}{\Delta t} \quad (3.13)$$

leading to the following nonlinear equation

$$\frac{4}{\Delta t^2} \mathbf{M} \left(\mathbf{u}^{n+\frac{1}{2}} - \mathbf{u}^n - \frac{\Delta t}{2} \dot{\mathbf{u}}^n \right) + \mathbf{f}_{\text{int}}(\mathbf{u}^{n+\frac{1}{2}}) = \mathbf{f}_{\text{ext}}^{n+\frac{1}{2}} \quad (3.14)$$

solved with a Newton iterative method. After solving for $\mathbf{u}^{n+\frac{1}{2}}$, one can deduce displacement and velocity fields at time t^{n+1} with the above relations (3.11-3.13).

Chapter 4

A Two-Phase Riemann Solver Based Approach

One of the goal of the current research project is to allow for the solution of two-phase problems. Within the setting of implosion problems as defined in the first chapter, these two-phase flow problems can occur at two different levels. Before the implosion of any structure, the generation of a pressure wave in the water may need to be modeled. Typically, this pressure wave emanates from the disequilibrium between the states of the water and of the gaseous products of high explosives, thus defining a fluid-fluid interface. Another fluid-fluid interface may need to be considered when a crack is initiated and propagated on the structure during its collapse. The current chapter presents the numerical scheme developed to solve these two-phase flow problems. In later chapters, this scheme will be referred to as the FVM-ERS method.

4.1 Introduction

Scientific interest in underwater explosions and hence in compressible multiphase flows goes back before World War II with the development of military submarines. Research on the subject has been of great importance as shown by [71]. The research tools have since then evolved to include computational tools. Actually, the use of computational tools is in itself a research topic as many investigators try to develop computationally inexpensive, accurate and robust numerical methods to allow the reproduction of experimental results and the prediction of underwater explosions/implosions effects.

As mentioned in the previous chapters, several methods ranging from the ALE methods to the front-tracking methods through level set methods and mixture methods have been developed for the solution of multiphase flow problems.

Probably the most well-known method is the Volume-Of-Fluid method [20] which uses volume fractions to reconstruct the interface and hence ensures discrete mass conservation. However, the need for a specific geometric advection scheme limits its accuracy and its time step size. In addition, the normals to the interface are not easily computed as the interface is not given and thus needs to be reconstructed. All these reasons make the VOF method a computationally expensive method. In addition to that, it was primarily designed for incompressible flow problems and its extension to compressible fluid mechanics is not straightforward, as another layer of complexity is added in order to recover the density field.

Another class of methods consists in tracking the interface between the two fluids. This approach discretizes the interface with an unstructured moving mesh that is transported in a Lagrangian fashion. The benefit of a Lagrangian scheme is offset by the frequent remeshings that are necessary to represent the interface accurately and

by the inability of the method to handle topological changes automatically. However, it was shown that this method can be successfully applied in computational fluid dynamics as well [40].

Various attempts were made to use conventional conservative schemes to solve two-phase flows. However, it was pointed out that these schemes could not preserve contact discontinuities as pressure oscillations were introduced and oscillations in other quantities hence subsequently arose [63] and a primitive formulation of the problem at the interface was first proposed instead[77]. As a consequence, quasi-conservative schemes have since been developed [64, 25]. Given the fluid equations of motion and a numerical scheme to compute the fluxes, advection equations for the parameters that define the equations of state considered can be derived in order to always satisfy the contact preserving property. Thus, no oscillation in pressure or other quantities numerically arose at a contact discontinuity. While such methods are very successful, they require that the two equations of state have the same analytical expression. In addition, to the best of the author's knowledge, it was only derived for equations of state of stiffened gas type. In the present work, it does not seem obvious how such a scheme could easily be applied to multifluid flow with one fluid satisfying the stiffened gas EOS and with another satisfying the JWL EOS (which is defined by five independent parameters).

Conservation and contact preserving properties are also offered by Arbitrary-Lagrangian Eulerian formulation. While these ALE-based methods apply to any EOS, the impracticality of some of them becomes apparent when the flow features lead to mesh entanglements. This is why moving mesh methods have been applied only in simple geometry flow problems such as spherical explosions [74]. An ALE

method with reconstructed control volumes was also proposed when a cartesian two-dimensional computational grid was considered [94]. In a three-dimensional unstructured dual grid, the computational cost of such a method is expected to significantly increase as the control volumes need to be cut and the control surfaces need to be recomputed.

Section 3.5 presented the level set as a way to locate and numerically transport the interface. The development of level set algorithms have led to many different compressible multiphase methods. Most if not all of these methods differ by the computation of the fluxes at the interface. Among them, the most popular one is probably the Ghost Fluid Method [73, 66]. It allows multifluid computations to be performed in the vicinity of the interface as if they pertained to a single medium domain, by exploiting the concept of ghost and real fluid control volumes. In a region close to the interface, the GFM sets the values of pressure and normal velocity in the ghost fluid cells to those in the real fluid cells. To eliminate an otherwise spurious “over-heating” phenomenon, it computes the density of the ghost fluid cells using an isobaric fix [66]. In its basic form, the GFM is non-conservative. However, it can be equipped with an a posteriori correction procedure that first measures the discrete conservation errors generated in the neighborhood of the interface during a given time-step, then offsets them using an error redistribution technique. This correction procedure was proposed in [21] where it was applied to stiff detonation problems. Unfortunately, the GFM fails to solve some multifluid problems as was analytically proven in [86]. An improved version of this method incorporating in a one-step time-integration scheme an approximate two-phase Riemann solver at the material interface that assumes either a two-shock or two-rarefaction wave structure was proposed in [86] for the solution of gas/water problems and illustrated with

simple 1D and 2D calculations. Like the original GFM, this enhanced version relies on the isobaric technique for eliminating the spurious over-heating phenomenon. For two- and three-dimensional applications, this isobaric fix requires the solution of yet another auxiliary partial differential equation [73] and therefore increases further the computational complexity of the method.

The ghost fluid method for the poor (GFMP) [65] is a variant method which avoids most of this overhead by computing two numerical fluxes at the material interface: one using the thermodynamic parameters of the fluid on one side of the interface, and another one using the thermodynamic parameters of the other fluid medium. It is an elegant, computationally efficient, and nearly conservative method in the sense that it conserves all conservative variables except the energy across the material interface. The GFMP was developed in [65] for one-dimensional problems using a one-step explicit time-integration algorithm and assuming that each given fluid is a stiffened gas. It involves a subtle but crucial conversion from conservative to primitive variables (and vice-versa) before and after advancing in time the solution of the level-set equation. Its extension to multiple dimensions is straightforward.

As mentioned in the previous chapter, the level set method was chosen to locate and transport the interface. Given the context above, the objective of this section is to develop an interfacial treatment that possesses the following properties

- contact preservation,
- robustness and consistency for two-phase flow problems with arbitrary large contact discontinuities and strong pressure jumps,
- applicability to a variety of different equations of state,
- computational efficiency for three-dimensional simulations.

To this effect, the remainder of this chapter is organized as follows. Section 4.2 presents the basic ideas of the new interfacial treatment. In section 4.3 the issue of computational efficiency is addressed. Reconstruction of the state variables at the interface in the framework of the MUSCL approach is presented in section 4.4. The complete algorithm with the time discretization is described in section 4.5. Sections 4.6 and 4.7 study contact-preserving and mass conservation properties. Finally, section 4.8 presents numerical results of simple academic one-dimensional shock tube problems to verify the method.

4.2 Basic Ideas

The robustness problem of multifluid methods comes from the discontinuity in both the values of the state variables and the equations of state. The same type of problem arises in single-fluid flows with a shock. In the same way that numerical algorithms tend to avoid discretizing equations across a shock discontinuity, it seems necessary to avoid discretizing equations, that is computing a numerical flux, using data from both sides of the material interface. Somehow, the stencil of the scheme near the interface needs to be modified to exclude nodes on the other side of such an interface. Thus, the new method proposed here is based, unlike most two-phase flow methods, on the computation of a numerical interfacial flux that does not directly use data from both sides of the material interface. In order to do that, an exact two-fluid Riemann problem at the interface between two different fluids is solved to obtain the two constant interfacial states which are used in the computation of the numerical interfacial flux. This method addresses the robustness issue encountered particularly for low space-order methods as the discretization scheme does not cross the material discontinuity.

It alleviates issues related to the consistency of some methods in particular test cases. It can be used with any numerical flux function that already exists for single-phase flows and requires the addition of an exact Riemann solver in the functions called in a code. More importantly, it can be applied for any EOS as long as Riemann invariants can be computed which is always the case in the framework of classical gas dynamics. The proposed numerical scheme is only subject to the existence and uniqueness of the solution of the Riemann problem and to the convergence of the exact Riemann problem. Because of the use of the exact solution of the Riemann problem, the proposed numerical scheme is referred to as the FVM-ERS (Finite Volume Method with Exact Riemann Solver) method in the remainder of this dissertation.

In this section, the two-fluid Riemann problem is presented and resolution algorithms are proposed for the EOS considered in this work, then the computation of the numerical interfacial flux is explained. A rationale is then given to explain why such a interfacial flux computation is considered and some of its shortcomings are assessed. For nodes that change fluids in the duration of the simulation, an update algorithm is then presented.

4.2.1 Exact Two-Phase Riemann Solvers

The one-dimensional Riemann problem for a single-phase flow has been briefly presented in section 2.4.5 and is reminded here:

$$\frac{\partial \mathbf{U}}{\partial \tau} + \frac{\partial \mathbf{F}}{\partial \xi} = 0, \quad \tau > 0$$

$$\mathbf{U}(\xi, 0) = \begin{cases} \mathbf{U}_L, & \xi < 0 \\ \mathbf{U}_R, & \xi > 0 \end{cases}$$

where τ and ξ respectively denote time and space coordinates. Under the assumptions of classical gas dynamics, the structure of its solution is composed of one left-facing and one right-facing genuinely non-linear waves and of a single centered contact discontinuity. Each non-linear wave can be either a rarefaction wave or a shock wave. Resolution methods for this problem are considered by [56] for the polytropic gas and stiffened gas equations of state, where a non-linear scalar equation must be solved in terms of the pressure at the interface. A similar method in the case of a Tait equation of state was also presented therein, but the problem was formulated in terms of the density. Colella and Glaz also proposed a method using a single non-linear scalar equation for more general equations of state in [60]. More recently, a 2-by-2 system of nonlinear equations where the unknowns are the densities on both sides of the interface was considered in [50]. Other methods are considered in the literature, but all of them express in different ways the continuity of pressure and velocity across the interface and use the Rankine-Hugoniot conditions for shock waves and/or the conditions given by the Riemann invariants for rarefaction waves. For each equation of state that is considered in this work, the resolution methods for the one-dimensional Riemann problem of single-phase flow are presented in Appendix B.

As already mentioned in [52, 67], a more general one-dimensional Riemann problem can be considered where the two equations of state on both sides of the initial discontinuity are different. In that case, the study of the Riemann problem done in the framework of classical gas dynamics for a single-phase flow remains valid for a two-phase flow as long as the fluids remain immiscible. The two fluids evolve on their own side of the contact discontinuity. Since a non-linear wave exists on each side of the contact discontinuity, each must satisfy conditions formulated with the appropriate equation of state. For instance, if the right-facing non-linear wave is a

shock, the flow must satisfy the Rankine-Hugoniot conditions formulated with the equation of state of the fluid at the right of the original discontinuity. In turn, this means that the resolution methods used for the one-dimensional one-phase Riemann problem could be used for the two-phase case. However, this is not exactly the case since the formulation of the one-phase Riemann problem may be different for each equation of state. It is then sometimes necessary to reformulate the two-phase Riemann problem, especially in cases where the two equations of state differ not only in their parameters values but also in their formulas. Each of the two-phase Riemann problem is reviewed below.

Stiffened Gas EOS - Stiffened Gas EOS In the following, the polytropic gas equation of state is not discussed explicitly, since it is a special case of the stiffened gas equation of state. (It can also be viewed as a special case of the JWL equation of state). When the two fluids involved have different SG EOS, the same formulation using a non-linear scalar equation in terms of the interfacial pressure can be used, similarly to the one-phase case. The non-linear equation in term of interfacial pressure p_I is

$$\mathcal{R}(p_I; \mathbf{U}_L, \mathbf{U}_R) = \mathcal{R}_L^{SG}(p_I; p_L, \rho_L, EOS_L) + \mathcal{R}_R^{SG}(p_I; p_R, \rho_R, EOS_R) + u_R - u_L = 0$$

where \mathcal{R}_L^{SG} and \mathcal{R}_R^{SG} each express Rankine-Hugoniot condition for a shock or conditions given by the Riemann invariants for a rarefaction wave and EOS_L and EOS_R represent the parameters of the two SG EOSs on the left and right side of the interface respectively.

Tait EOS - Tait EOS When the two fluids involved have different Tait EOS, the original formulation for the one-phase case needs to be modified. Indeed, in the one-phase Riemann problem, the pressure is continuous across the contact discontinuity, and so is the density given that the EOS is (2.9). In the two-phase case, continuity in pressure across the interface is not equivalent to continuity in density and it becomes necessary to reformulate the problem in terms of the pressure rather than in terms of the density.

$$\mathcal{R}(p_I; \mathbf{U}_L, \mathbf{U}_R) = \mathcal{R}_L^{Tait}(p_I; \rho_L, EOS_L) + \mathcal{R}_R^{Tait}(p_I; \rho_R, EOS_R) + u_R - u_L = 0$$

where \mathcal{R}_L^{Tait} and \mathcal{R}_R^{Tait} each express Rankine-Hugoniot condition for a shock or conditions given by the Riemann invariants for a rarefaction wave and EOS_L and EOS_R represent the parameters of the two Tait EOSs on the left and right side of the interface respectively.

JWL EOS - JWL EOS Given an internal energy and a pressure, the form of EOS such as a Tait EOS or SG EOS allows for an easy computation of the corresponding density. With more complicated EOS such as the JWL EOS, such a computation might not be straightforward and require the resolution of a non-linear equation through an iterative procedure. A non-linear equation in term of interfacial pressure, such as the one given for a SG EOS Riemann problem, will lead to two embedded iterative loops. To overcome this burden, the approach proposed by [50] is considered here. Instead of solving a scalar nonlinear equation in terms of the interfacial pressure, a system of 2 scalar nonlinear equations in terms of the two interfacial densities imposes the equality of velocity and pressure on both sides of the contact discontinuity. The state values on both sides of a rarefaction wave or of a shock wave are linked by

the Riemann invariants in the first case and by the Rankine-Hugoniot relations in the second. In particular, if one state \mathbf{U}_L is known completely on side L and the density ρ_{IL} is also known on the other side IL , then it is possible by using the Riemann invariants or the Rankine-Hugoniot relations, to write both velocity and pressure on side IL as

$$\begin{aligned} u_{IL}^{JWL}(\rho_{IL}; \mathbf{U}_L, EOS_L) \\ p_{IL}^{JWL}(\rho_{IL}; \mathbf{U}_L, EOS_L). \end{aligned}$$

Thus, applying the previous relations on the two nonlinear waves of a shock tube problem and imposing the equality of pressure and velocity at the contact discontinuity lead to

$$\begin{cases} u_{IL}^{JWL}(\rho_{IL}; \mathbf{U}_L, EOS_L) = u_{IR}^{JWL}(\rho_{IR}; \mathbf{U}_R, EOS_R) \\ p_{IL}^{JWL}(\rho_{IL}; \mathbf{U}_L, EOS_L) = p_{IR}^{JWL}(\rho_{IR}; \mathbf{U}_R, EOS_R) \end{cases}$$

where u_{Ik} and p_{Ik} are respectively the interfacial velocity and pressure on the k -side of the interface. The unknowns are then the densities ρ_{IR} and ρ_{IL} on both sides of the contact discontinuity. Similarly to previous Riemann problems, the system of nonlinear equations is solved using an iterative procedure. Once the densities are obtained, the interfacial pressure and velocity can be inferred.

SG EOS - Tait EOS The two EOS involved in this two-phase Riemann problem are fundamentally different in the sense that the analytical expressions of these EOS are different. However, in this particular case, it is still possible to use a single-equation method to solve for the interfacial states. Without loss of generality, by assuming that the Tait EOS fluid lies on the left side of the interface - and is denoted

EOS_L - and that the SG EOS lies on the right side of the interface - and is denoted EOS_R -, the nonlinear scalar equation can be written under the form:

$$\mathcal{R}(p_I; \mathbf{U}_L, \mathbf{U}_R) = \mathcal{R}_L^{Tait}(p_I; \rho_L, EOS_L) + \mathcal{R}_R^{SG}(p_I; p_R, \rho_R, EOS_R) + u_R - u_L = 0$$

where \mathcal{R}_L^{Tait} expresses Rankine-Hugoniot conditions across a shock wave or conditions given by the Riemann invariants across a rarefaction wave for a fluid whose EOS is the Tait EOS, and \mathcal{R}_R^{SG} expresses similar conditions for a fluid whose EOS is the SG EOS.

In addition to solving the two-phase Riemann problem, problems involving a stiffened gas EOS and a Tait EOS also requires a different resolution treatment as the fluid modeled by the Tait EOS has one less equation – the energy conservation equation – than the fluid modeled by the stiffened gas EOS. As mentioned in section 2.3.1, the barotropic nature of the Tait EOS fluid does not require to solve for all conservation equations as the energy of the fluid is directly given once density and velocity are known. In order to solve the two-phase problem, all conservation equations are considered for the stiffened gas EOS fluid while only mass and momentum conservation equations are for the Tait EOS fluid. At the interface between the two fluids, the two-phase Riemann problem is formulated as above using density, velocity and pressure on the stiffened gas side and using density and velocity (or equivalently pressure) on the Tait EOS side. It is therefore never required to compute the energy in the Tait EOS fluid. However, it is always possible to recover the energy as shown in section 2.3.1.

SG EOS - JWL EOS This case mixes two EOS whose Riemann problem formulations as seen above are different. The resolution method proposed by [50] is used

once more. Assuming, without loss of generality, that the JWL EOS fluid lies on the left side of the interface and that the SG EOS fluid lies on the right side of the interface,

$$\begin{cases} u_{IL}^{JWL}(\rho_{IL}; \mathbf{U}_L, EOS_L) = u_{IR}^{SG}(\rho_{IR}; \mathbf{U}_R, EOS_R) \\ p_{IL}^{JWL}(\rho_{IL}; \mathbf{U}_L, EOS_L) = p_{IR}^{SG}(\rho_{IR}; \mathbf{U}_R, EOS_R) \end{cases}$$

where u_{Ik} and p_{Ik} are respectively the interfacial velocity and pressure on the k -side of the interface linked to the state \mathbf{U}_k via either a rarefaction wave or a shock wave for the fluid on the k side.

Tait EOS - JWL EOS This case has not been studied in this work.

All these nonlinear equations, may they be scalar or vectorial, can be cast under the form $F(x) = 0$. Resolution of this equation is done via an iterative Newton method:

$$x^{n+1} = x^n - \left(\frac{\partial F}{\partial x}(x^n) \right)^{-1} F(x^n)$$

where $\left(\frac{\partial F}{\partial x}(x^n) \right)^{-1}$ denotes the inverse of $\frac{\partial F}{\partial x}(x^n)$ as long as it exists. Iterations are stopped when the following convergence criterion is met:

$$\frac{x^{n+1} - x^n}{x^{n+1} + x^n} < \epsilon$$

where ϵ is a convergence tolerance.

4.2.2 Numerical Interfacial Flux

As mentioned earlier, it is not obvious how to compute a numerical flux between two neighboring nodes across their common control surface, knowing that the material interface lies somewhere in between the two points. A method based on the two-phase Riemann problem and its exact resolution is presented hereafter in order to compute that numerical interfacial flux.

The problem is formalized as follows. Two control volumes Ω_i and Ω_j associated to nodes i and j are considered with flow variable values \mathbf{U}_i and \mathbf{U}_j respectively. The level set $\phi = 0$ lies in between the two considered points, and for example but without loss of generality, it is assumed that $\phi_i < 0$ and $\phi_j > 0$. In addition, control volume Ω_k with $k \in \{i, j\}$ contains a fluid whose equation of state is EOS_k . The primary assumption of the method is that the material interface given by the level set coincides with the intersection of the control volume boundary between Ω_i and Ω_j and the edge $i-j$ as shown in Figure 4.1. The correct use of a numerical flux function that takes the two states \mathbf{U}_i and \mathbf{U}_j as arguments is not obvious as the two fluids can be highly different not only in their state but also in their material properties. It seems preferable to avoid using data from opposite sides of a discontinuity. However, near the interface, a locally one-dimensional two-phase Riemann problem normal to the material interface can be defined:

$$\frac{\partial \mathbf{U}}{\partial \tau} + \frac{\partial \mathbf{F}}{\partial \xi} = 0, \quad \tau > 0 \quad (4.1)$$

$$\mathbf{U}(\xi, 0) = \begin{cases} \mathbf{U}_i, & \xi > 0 \\ \mathbf{U}_j, & \xi < 0 \end{cases} \quad (4.2)$$

where τ and ξ are locally defined variables in time and space coordinates. Since the

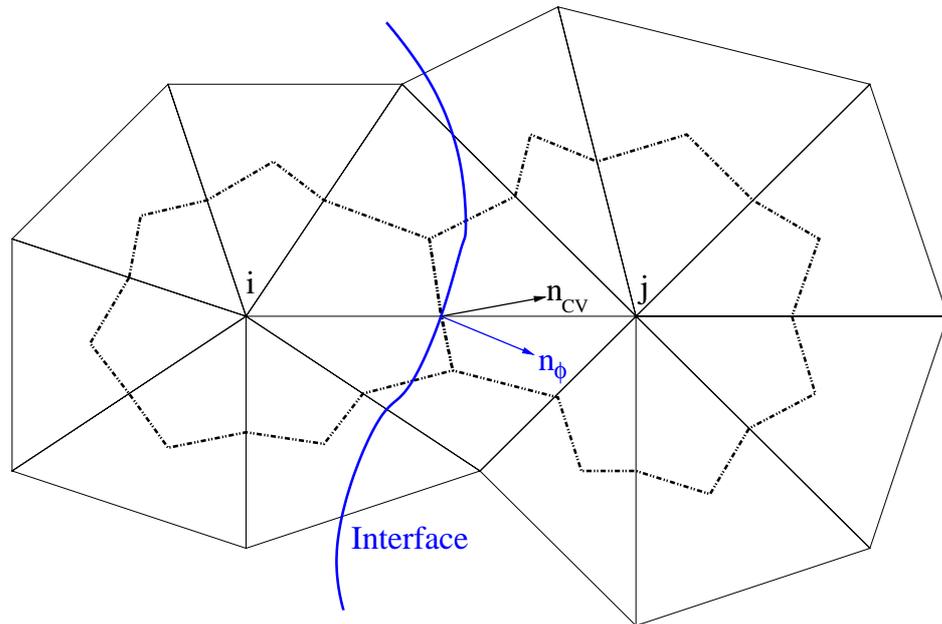


Figure 4.1: The material interface located between two nodes is assumed to go through the intersection $\partial\Omega_{ij} \cap [i,j]$ of the two corresponding control volume boundary and of the corresponding edge. The normals of the material interface and of the control surface are however different.

one-dimensional problem consider only a velocity normal to the material interface, a decomposition of the three-dimensional velocity vector into a normal component and a tangential component is done. It is the normal to the material interface, that is \mathbf{n}_ϕ , that determines the normal and tangential parts of the velocities, and not the normal to the boundaries of the control volumes (see Figure 4.1). The decomposition can be written

$$\mathbf{u} = u_{n_\phi} \mathbf{n}_\phi + \mathbf{u}_{t_\phi}$$

with

$$\begin{aligned} u_{n_\phi} &= \mathbf{u} \cdot \mathbf{n}_\phi \\ \mathbf{u}_{t_\phi} &= \mathbf{u} - (\mathbf{u} \cdot \mathbf{n}_\phi) \mathbf{n}_\phi. \end{aligned}$$

In the present work, the mathematical properties of the level set are used to obtain the normal to the material interface. In particular, for two neighboring nodes i and j , this normal is computed with

$$\mathbf{n}_\phi = \frac{(\nabla\phi)_i + (\nabla\phi)_j}{2}$$

where $(\nabla\phi)_k$ is a least-square gradient [2] - presented in chapter 3 - of ϕ at node k which takes into account all neighboring nodes. Even the nodes across the interface are taken into account in the formulation of the corresponding least-square problem since the level set is a smooth function across the interface. It is however possible to consider different numerical normals. The initial conditions to the local

one-dimensional two-phase Riemann problem are in primitive variable form

$$\begin{aligned} & (\rho_i, u_{i,n_\phi}, p_i)^T \\ & (\rho_j, u_{j,n_\phi}, p_j)^T \end{aligned}$$

As seen earlier, this two-phase Riemann problem can be solved for the equations of state considered in the present work. Most importantly, its solution includes two constant states on both sides i and j of the contact discontinuity for $\tau > 0$ and are respectively given in their primitive variable form by

$$\begin{aligned} & (\rho_i^{\mathcal{R}}, u_{n_\phi}^{\mathcal{R}}, p^{\mathcal{R}})^T \\ & (\rho_j^{\mathcal{R}}, u_{n_\phi}^{\mathcal{R}}, p^{\mathcal{R}})^T . \end{aligned}$$

Three-dimensional interfacial states are then reconstructed via the same procedure mentioned earlier

$$\begin{aligned} \mathbf{V}_i^{\mathcal{R}} &= (\rho_i^{\mathcal{R}}, u_{n_\phi}^{\mathcal{R}} \mathbf{n}_\phi + \mathbf{u}_{i,t_\phi}, p^{\mathcal{R}})^T \\ \mathbf{V}_j^{\mathcal{R}} &= (\rho_j^{\mathcal{R}}, u_{n_\phi}^{\mathcal{R}} \mathbf{n}_\phi + \mathbf{u}_{j,t_\phi}, p^{\mathcal{R}})^T . \end{aligned}$$

Their conservative form are written $\mathbf{U}_i^{\mathcal{R}}$ and $\mathbf{U}_j^{\mathcal{R}}$ in the rest of the paragraph. Next, the computation of the interfacial fluxes is done. It differs for the two control volumes from the usual numerical flux in a single phase flow problem since, for control volume Ω_i , the state vectors \mathbf{U}_i and $\mathbf{U}_i^{\mathcal{R}}$ are used to compute the numerical interfacial flux

$$\Phi_{ij} = \Phi(\mathbf{U}_i, \mathbf{U}_i^{\mathcal{R}}, EOS_i, \nu_{ij}, \kappa_{ij})$$

where the dependency of the numerical flux on the equation of state is explicitly expressed. Similarly, the numerical interfacial flux for control volume Ω_j is given by

$$\Phi_{ji} = \Phi(\mathbf{U}_j, \mathbf{U}_j^{\mathcal{R}}, EOS_j, \nu_{ji}, \kappa_{ji}).$$

Note that the normal ν_{ji} of the boundaries of the control volumes Ω_i and Ω_j is used in the flux computation, and not the normal to the material interface which is used only in the decomposition of three-dimensional velocities into normal and tangential components and in the reciprocal operation. The normal to the control volume boundaries are used in order to remain consistent with the finite volume formulation.

With the above flux computation for a given control volume belonging to a given fluid, the direct use of state variable quantities from another fluid is avoided. Thus, the spatial discretization stencil for the state variables (density, momentum and energy) does not cross the interface, in the sense that it does not include nodes that lie on the other side of the interface. Obviously the two interfacial fluxes for two neighboring nodes are not conservative. Conservation errors are incurred and will be discussed later.

4.2.3 Rationale

The reason for the proposed interfacial flux computation described above is provided by the structure of the solution of the two-phase Riemann problem. This solution is composed of four constant states \mathbf{U}_i , $\mathbf{U}_i^{\mathcal{R}}$, $\mathbf{U}_j^{\mathcal{R}}$ and \mathbf{U}_j separated by non-linear waves and a contact discontinuity.

Consider first the following one-phase Riemann problem associated with a fluid

whose equation of state is EOS_i ,

$$\frac{\partial \mathbf{U}}{\partial \tau} + \frac{\partial \mathbf{F}}{\partial \xi} = 0, \quad \tau > 0 \quad (4.3)$$

$$\mathbf{U}(\xi, 0) = \begin{cases} \mathbf{U}_i, & \xi < 0 \\ \mathbf{U}_i^{\mathcal{R}}, & \xi > 0 \end{cases} \quad (4.4)$$

and then the similar one-phase Riemann problem associated with a fluid whose EOS is EOS_j ,

$$\frac{\partial \mathbf{U}}{\partial \tau} + \frac{\partial \mathbf{F}}{\partial \xi} = 0, \quad \tau > 0 \quad (4.5)$$

$$\mathbf{U}(\xi, 0) = \begin{cases} \mathbf{U}_j^{\mathcal{R}}, & \xi < 0 \\ \mathbf{U}_j, & \xi > 0. \end{cases} \quad (4.6)$$

The solution of problem (4.3)-(4.4) is composed of two constant states, \mathbf{U}_i and $\mathbf{U}_i^{\mathcal{R}}$, and a single non-linear wave connecting them. The restriction of $\xi < \xi_{\text{contact}}$ of this solution, where ξ_{contact} denotes the coordinate of the contact discontinuity (which has zero strength in this case), is identical to the restriction to $\xi < \xi_{\text{contact}}$ of the solution of the original two-phase Riemann problem. Similarly, the restriction to $\xi > \xi_{\text{contact}}$ of the solution to the problem (4.5)-(4.6) is identical to the restriction to $\xi > \xi_{\text{contact}}$ of the solution of the original two-phase Riemann problem (4.1)-(4.2). These two results hold for any combination of equations of state ($\text{EOS}_i, \text{EOS}_j$) and for any initial states ($\mathbf{U}_i, \mathbf{U}_j$). Therefore, using $\mathbf{U}_i^{\mathcal{R}}$ ($\mathbf{U}_j^{\mathcal{R}}$) in the Roe flux function associated with node i (j), which itself is an approximate Riemann solver, can be expected to give the sought-after accuracy and robustness effects.

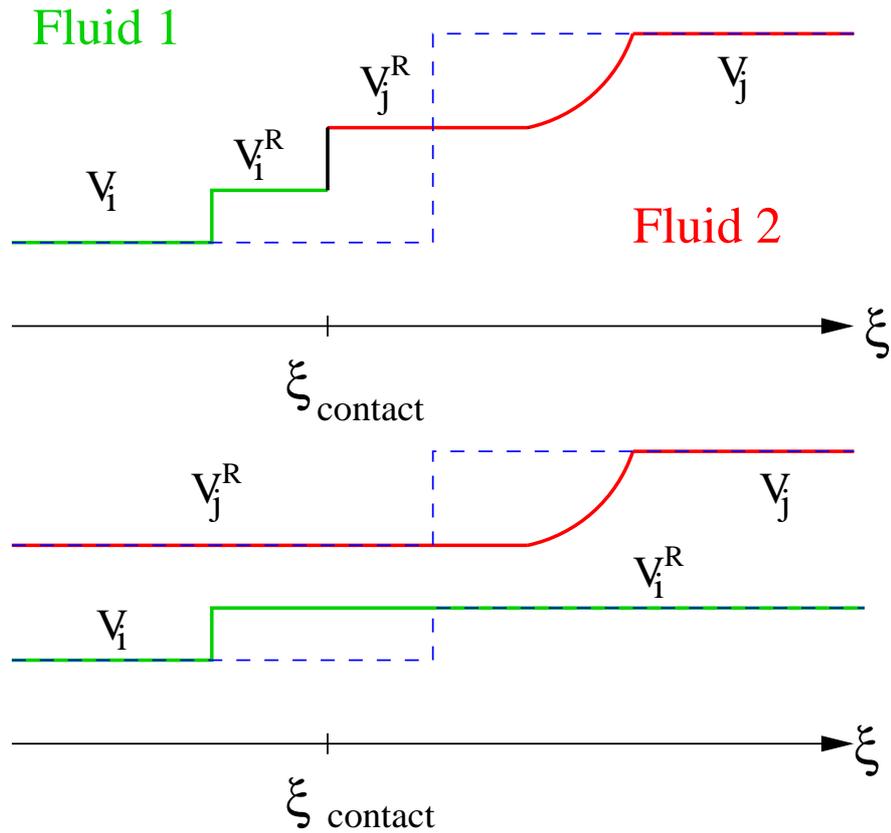


Figure 4.2: Decomposition of the two-phase Riemann problem (top) into two single-phase Riemann problems (bottom): the initial conditions are plotted in blue dotted lines, the solutions at a given time $t > 0$ are in black full lines.

4.2.4 Update of Phase-Changing Nodes

The computation of both interfacial fluxes presented in section 4.2.2 avoids using thermodynamically different states, as both arguments of the numerical flux function represent states from the same fluid. This raises the issue of updating the state variables of a node when this node changes fluid. In single-phase flow, nodes that are being passed by a contact discontinuity or a shock wave see their states updated naturally via flux computation. However, nothing in the numerical interfacial flux function is geared toward this issue. Thus the update of any phase-changing node must be done manually by checking which nodes have changed fluids and by providing appropriate state variables. This section presents how this can be done.

One possibility is given by using the solution of the Riemann problem, that was computed during the computation of the fluxes. In the one-dimensional Riemann problem, the constant interfacial states $\mathbf{U}_L^{\mathcal{R}}$ and $\mathbf{U}_R^{\mathcal{R}}$ that develop on both sides of the contact discontinuity are known and their locations are given by the locations of the contact discontinuity and of either a shock wave or the tail of a rarefaction wave. Thus, the state of a point close to the original discontinuity – assumed in this discussion, without loss of generality, to be on the left side of the discontinuity – will change from its initial value to the constant interfacial state $\mathbf{U}_L^{\mathcal{R}}$, and if, in addition, the contact discontinuity moves to the left, the considered node eventually changes fluids and its state will vary abruptly from $\mathbf{U}_L^{\mathcal{R}}$ to $\mathbf{U}_R^{\mathcal{R}}$. This remark on the solution of the Riemann problem provides an obvious algorithm to update the numerical values of the nodes that change fluids in a one-dimensional problem. However, while such an algorithm is based on the solution of the Riemann problem at the continuous level, it is applied to a discrete system where each state value corresponds to an average state in a control volume (with a positive volume). Thus, depending on the contact discontinuity speed

which is also the speed of the flow, the use of the Riemann interfacial states to update phase-changing nodes may or may not be appropriate. Consider a node that changes phases between time steps n and $n + 1$. If the contact discontinuity speed is large with respect to the grid spacing and the time step, most of the control volume of this node can change fluids and its average value is well described by the Riemann interfacial state. If the contact discontinuity speed is small, its neighboring node that was in another phase at time-step n and that is in the same phase at time-step $n + 1$ may have state values at time-step $n + 1$ that better describe the state of the fluid at the phase-changing node. This avoids having two neighboring nodes in the same fluid and close to the interface with potentially very different state values. It is therefore necessary to consider updating phase-changing node state values with either an extrapolation of neighboring nodes state values or an extrapolation of the Riemann interfacial state values, if not a combination of the two.

In a three-dimensional problem, extrapolation of several Riemann interfacial state values or from several neighboring nodes state values may need to be considered. A simple averaging procedure would consist of the arithmetic mean of all the state values considered for the extrapolation. However, the fluids are being advected and it seems reasonable that the physics of the problem should not be disregarded. In order to take into account the direction of the flow, the averaging procedure is modified as follows. A node i changing fluids between time step n and $n + 1$ is considered. The set $\mathcal{N}(i)$ denotes the set of its neighboring nodes that are on the opposite side of the interface at time step n . Among those neighboring nodes, only the nodes with positive $\overrightarrow{x_k x_i} \cdot \mathbf{u}_k$ are considered, since negative values correspond to flow velocities that carry a fluid particle at the location of node k away from node i (see Figure 4.3). The corresponding non-dimensional quantities serve as weights to the averaging

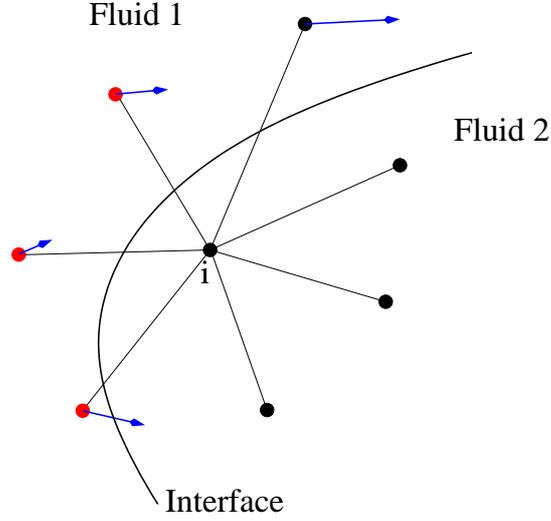


Figure 4.3: Phase change update: node i is about to change phases and extrapolation will proceed only along edges with red nodes

of the extrapolated values, may they be neighboring node state values or Riemann interfacial state values. The new value of the node i at time step $n + 1$ can be written as

$$\mathbf{U}_i^{n+1} = \frac{1}{\sum_{k \in \mathcal{N}(i)} w_k} \sum_{k \in \mathcal{N}(i)} w_k \mathbf{U}_k^* \quad (4.7)$$

where

$$w_k = \max \left(0, \frac{\overrightarrow{x_k x_i}}{|\overrightarrow{x_k x_i}|} \cdot \frac{\mathbf{u}_k}{|\mathbf{u}_k|} \right) \quad (4.8)$$

$$\mathbf{U}_k^* = \begin{cases} \mathbf{U}_k^{\mathcal{R}} & \text{if a Riemann interfacial state value update algorithm is chosen} \\ \mathbf{U}_k^n & \text{if a neighboring node state value update algorithm is chosen} \end{cases} \quad (4.9)$$

4.3 Performance Considerations for the Exact Two-Phase Riemann Solver

It is often criticized that the exact resolution of a Riemann problem is computationally too expensive to be used in computational fluid dynamics and this has led to the development of new approximate Riemann flux schemes, such as the Roe scheme, the HLLE scheme, the HLLC scheme and so forth. Even if it is truly expensive to use a Godunov flux in a space of dimension d , the use of a Godunov scheme on the fluids interface (which is a space of dimension $d-1$) and of any approximate Riemann flux scheme in the rest of the domain does not always incur a significant cost. As an example, the timings of the flux computation for a given three-dimensional simulation are reported in Table 4.1. In the first case, the two fluids on both sides of the interface have the same stiffened gas EOS and no exact Riemann solver is used at the interface. In the second case, the two fluids are different stiffened gas and an exact Riemann solver is used. The difference between the two timings is due to the use of an exact Riemann solver at the interface.

solver type	per iteration	per iteration per edge
single-phase flow	0.206	8.0×10^{-6}
two-phase flow	0.216	8.0×10^{-6}

Table 4.1: Average timings (in seconds) of flux computation with and without an exact Riemann solver between two stiffened gas

However, other EOS need also be considered. In the current research project, the JWL EOS, presented in chapter 2, is used to model the gaseous products of high explosives after detonation. This EOS belongs to the general class of EOS for which Riemann invariants cannot be expressed in close form and are only defined by

ODEs. For these EOS, the repeated computation of Riemann invariants via ODE integration can become excessively expensive as it needs to be computed at every iteration during the iterative solve of the Riemann problem and thus, such an ODE integration is inefficient for a realistic simulation. Indeed, the cost of a single Riemann problem between a stiffened gas and a JWL fluid, in which the JWL fluid undergoes a rarefaction, is of the order of 2.0×10^{-3} second when a Runge-Kutta 2 scheme is used with 50 iterations. When the number of iterations is decreased to 10, the timing is of the order of 4.0×10^{-4} second. Clearly, the solving of the Riemann problem is dominated by the computation of the Riemann invariants in the JWL fluid. The same simulation as the one mentioned above had timings given by Table 4.2. One clearly sees that the flux computation becomes dominated by the computation of the solution to the Riemann problem. Notice however, that the slow down in a simulation involving a rarefaction wave in a JWL fluid is not entirely due to the computation of the exact Riemann problem. The JWL EOS is by itself computationally expensive as it uses two exponential operations in all subroutines directly related to the JWL EOS. Hence, a single-phase JWL flow computation is more expensive than a single-phase SG flow computation, as shown by comparison of single-phase flow computation timings in Tables 4.1 and 4.2.

solver type	per iteration	per iteration per edge
single JWL flow	0.321	1.28×10^{-5}
SG-JWL(10)	1.49	5.94×10^{-5}
SG-JWL(50)	6.1	2.44×10^{-4}

Table 4.2: Average timings (in seconds) of flux computation with and without an exact Riemann solver when a fluid with a JWL EOS undergoes a rarefaction. When an exact Riemann solver is used, the number of integration steps to compute the Riemann invariant is specified in parentheses.

To alleviate the cost of integrating the Riemann invariants, the use of a tabulation associated with an interpolation procedure is proposed. The use of tabulation is already widely used either to avoid the expensive computational costs of some function or simply because no analytical function is available to compute certain quantities. One example is given in computational fluid dynamics with combustion, where the enthalpy depends on the thermodynamical state of the fluid, but is not necessarily given by any analytical equation. Since the Riemann invariants can be computed, the proposed strategy consists in computing different values of the Riemann invariants and storing them on a grid as a preprocessing step. Then, instead of integrating an expensive ODE to compute values of the Riemann invariant, linear interpolation on the considered grid is done. The rest of this section is dedicated to the assessment of the use of such a method and its potential in the current framework. To this end, the rest of the section is organized as follows. First, the Riemann invariants necessary to solve the Riemann problem are reviewed in the case of a general EOS and in the case of the JWL EOS. This brief study allows to clarify requirements on the possible types of tabulation. Sparse grid based on truncated tensor products meet these requirements and are presented. Finally, a tabulation using such a sparse grid is applied to the JWL EOS and its timings and accuracy are tested in a Matlab program.

4.3.1 Riemann Invariants

In the case of a general EOS, the Riemann invariants may be computed by integrating the following ODEs:

$$\begin{cases} \frac{dp}{d\rho} = c^2(p, \rho) \\ \frac{du}{d\rho} = \pm \frac{c(p, \rho)}{\rho} \end{cases}$$

along the isentropic curve going through the initial state $(\rho_a, p(\rho_a) = p_a, u(\rho_a) = u_a)$. Since an analytical formula of the entropy is not available, the algorithm to solve the Riemann problem within the framework given by [50] requires to integrate the equations from ρ_a to ρ_b with initial conditions (p_a, u_a) to obtain (p_b, u_b) , where ρ_b is the density given by the previous iteration of the Newton algorithm.

In the case of the JWL EOS, using notations of chapter 2, the first Riemann invariant is given by

$$\frac{p - A_1 e^{-\frac{R_1 \rho_0}{\rho}} - A_2 e^{-\frac{R_2 \rho_0}{\rho}}}{\rho^\gamma} = \frac{p_a - A_1 e^{-\frac{R_1 \rho_0}{\rho_a}} - A_2 e^{-\frac{R_2 \rho_0}{\rho_a}}}{\rho_a^\gamma} \quad (4.10)$$

$$(4.11)$$

This quantity acts as an entropy

$$s(p, \rho) = \frac{p - A_1 e^{-\frac{R_1 \rho_0}{\rho}} - A_2 e^{-\frac{R_2 \rho_0}{\rho}}}{\rho^\gamma}$$

providing an explicit relation to compute pressure in terms of entropy and density

$$p(s, \rho) = s\rho^\gamma + A_1 e^{-\frac{R_1 \rho_0}{\rho}} + A_2 e^{-\frac{R_2 \rho_0}{\rho}}$$

It is then possible to substitute this equation in the second ODE

$$\begin{aligned}\frac{du}{d\rho} &= \pm \frac{c(p(s, \rho), \rho)}{\rho} \\ &= \frac{c(s, \rho)}{\rho}\end{aligned}$$

where a slight abuse of notation was used since the same letter c designates two different functions, but the same physical quantity. Along the integration path of this ODE, the entropy s is constant

$$s = s_a = \frac{p_a - A_1 e^{-\frac{R_1 \rho_0}{\rho_a}} - A_2 e^{-\frac{R_2 \rho_0}{\rho_a}}}{\rho_a^\gamma}$$

and thus, the ODE is integrated from ρ_a to final density ρ_b with a constant entropy s_a with initial state u_a to obtain the final state u_b . The final pressure is obtained from the equality of entropy between the two initial and final states using Eq. (4.10).

4.3.2 Tabulation Requirements

The analysis of the Riemann invariants shows that a three-dimensional tabulation is a priori necessary. For a general EOS, initial density ρ_a , initial pressure p_a and final density ρ_b are required as inputs to the tabulation. The required outputs can be written under the general form

$$\int_{\mathcal{I}_s(\rho_a, \rho_b)} c^2(p, \rho) d\rho \tag{4.12}$$

$$\int_{\mathcal{I}_s(\rho_a, \rho_b)} \pm \frac{c(p, \rho)}{\rho} d\rho \tag{4.13}$$

$$\tag{4.14}$$

where $\mathcal{I}_s(\rho_a, \rho_b)$ designates the isentropic curve from (ρ_a, s) to (ρ_b, s) . The tabulation of these quantities with three inputs can become memory-wise extremely expensive. The most naive approach would consist in a 3D cartesian grid with constant spacings in each direction. Depending on the range of each input and the resolution requirement, this may not be an affordable solution. For example, the case of an underwater explosion or implosion, or of any violent phenomenon, has density and pressure ranges spanning several orders of magnitude and the number of entries can easily reach a billion. In the case of a JWL EOS fluid, even if three inputs - initial density ρ_a , final density ρ_b and entropy $s_a = s_b$ - also need to be specified, the tabulation requires only two inputs. Indeed, the difference in velocity can be computed as follows:

$$\int_{\mathcal{I}_s(\rho_a, \rho_b)} \pm \frac{c(s, \rho)}{\rho} d\rho = \int_{\mathcal{I}_s(\rho_c, \rho_b)} \pm \frac{c(s, \rho)}{\rho} d\rho - \int_{\mathcal{I}_s(\rho_c, \rho_a)} \pm \frac{c(s, \rho)}{\rho} d\rho$$

where ρ_c is an arbitrary density value and s is by definition constant along $\mathcal{I}_s(\rho_a, \rho_b)$ and $s = s_a = s_b$. Thus, a two-dimensional tabulation is possible as long as two calls to the tabulation are made. Even if the range of each input is large, the number of entries is likely to be much less and within reasonable bounds.

While the use of a non uniform cartesian grid would help alleviate the memory requirements of tabulations to a certain degree - which is not always enough to reach reasonable bounds to the number of entries of the grid - , the choice of the local spacing in each coordinate would still remain an issue, as different equation of state would lead to different spacings.

In order to address these issues of memory management and resolution within the framework of tabulation, a very general approach is suggested that allows the tabulation of Riemann invariants, independently of the equation of state considered.

4.3.3 Sparse Grids

One possible solution is offered by sparse grids [37, 3], which prove to be advantageous in many regards. Based on a hierarchical basis equivalent to the conventional nodal basis and on a sparse tensor product, a sparse grid requires far fewer nodes while its accuracy is only slightly degraded in order to represent a function defined over a d -dimensional space.

In the following paragraphs, sparse grid methods are presented for the purpose of tabulating a bivariate function. Direct application is the tabulation of Riemann invariants for an equation of state which has an algebraic isentropic form such as the JWL one. Results are still valid in higher dimensions and it is expected that the method can be applied to the three-variate tabulation of the Riemann invariants of a general EOS. The reader is invited to read [3] and the references therein for more details on the subject of sparse grids.

For the JWL EOS, let us consider the following notations. For the sake of clarity, the presentation considers the domain $\Omega = [0, 1]^2$. A multi-index is introduced

$$\underline{l} = (l_1, l_2) \in \mathbb{N}^2$$

representing the level of refinement of the grid $\Omega_{\underline{l}}$ on Ω . The mesh size of the grid

$$h_{\underline{l}} = (h_{l_1}, h_{l_2}) = (2^{-l_1}, 2^{-l_2})$$

is different in each direction, but constant along each direction. For a given index \underline{l} , the grid $\Omega_{\underline{l}}$ is composed of $(2^{l_1} + 1) \times (2^{l_2} + 1)$ points labelled

$$x_{\underline{l}, \underline{j}} = (x_{l_1, j_1}, \dots, x_{l_2, j_2}, \dots)$$

where $j_k = 0, \dots, 2_k^l$. For such a grid, the associated space $V_{\underline{l}}$ of piecewise bilinear functions is defined

$$V_{\underline{l}} = \text{span} \left\{ \phi_{\underline{l}, \underline{j}} \quad / \quad j_t = 0, \dots, 2^{l_t}, t = 1, 2 \right\}$$

where for any $\underline{x} = (x_1, x_2)$, $\phi_{\underline{l}, \underline{j}}$ is the product

$$\phi_{\underline{l}, \underline{j}}(\underline{x}) = \phi_{l_1, j_1}(x_1) \times \phi_{l_2, j_2}(x_2)$$

of the usual one-dimensional linear hat functions $\phi_{l,j}$ with support $[x_{l,j} - h_l, x_{l,j} + h_l] \cap [0, 1]$.

A hierarchical difference space $W_{\underline{l}}$ is defined via

$$W_{\underline{l}} = V_{\underline{l}} \setminus (V_{(l_1-1, l_2)} \oplus V_{(l_1, l_2-1)})$$

where $V_{\underline{l}} = 0$ if either l_1 or l_2 is equal to -1 . Figure 4.4 shows the first steps of the construction of such a hierarchical difference space in one dimension. It is then possible to rewrite $W_{\underline{l}}$ more explicitly as

$$W_{\underline{l}} = \text{span} \left\{ \phi_{\underline{l}, \underline{j}}, \quad \underline{j} \in B_{\underline{l}} \right\}$$

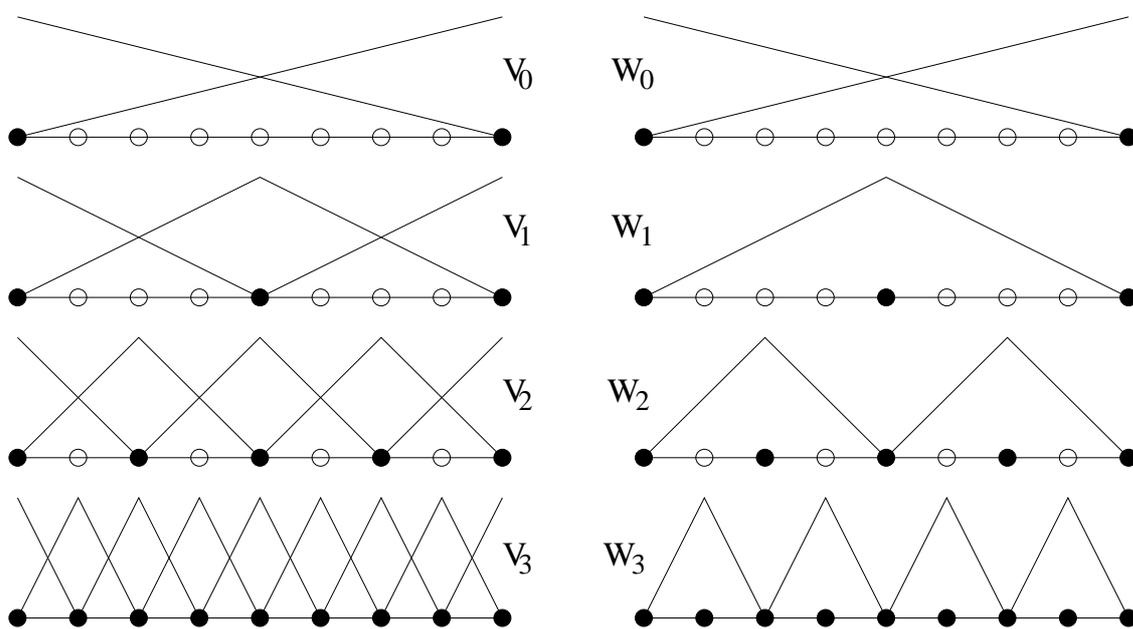


Figure 4.4: Construction of the one-dimensional hierarchical basis (on the right) in comparison to the conventional nodal basis (on the left).

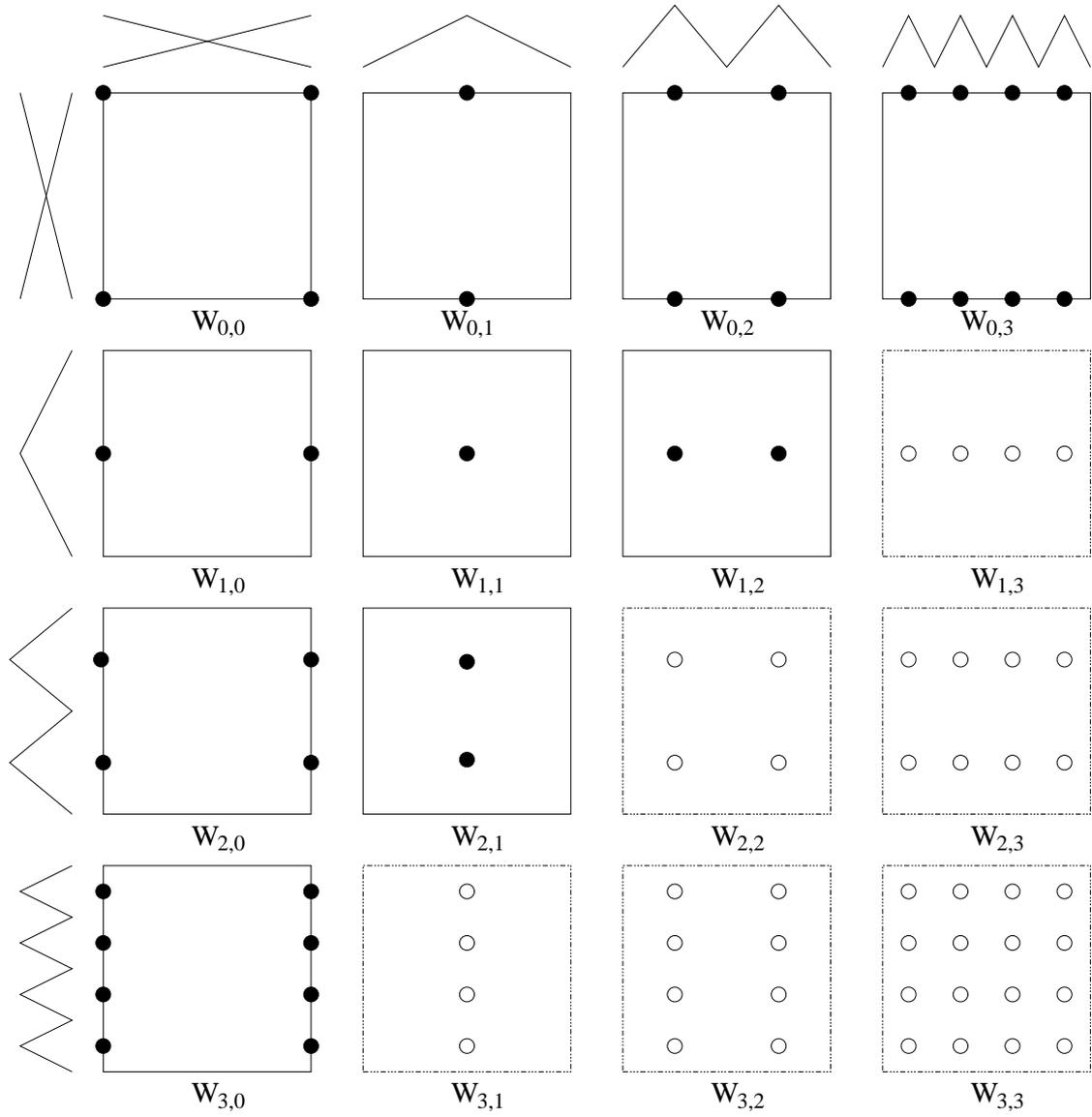


Figure 4.5: Construction of the two-dimensional hierarchical basis based on the second-order tensor product of one-dimensional hierarchical basis.

where

$$B_{\underline{l}} = \left\{ \underline{j} \in \mathbb{N}^2 \left| \begin{array}{ll} j_t = 1, \dots, 2^{l_t} - 1, & j_t \text{ odd, } t = 1, 2, \text{ if } l_t > 0 \\ j_t = 0, 1, & t = 1, 2, \text{ if } l_t = 0 \end{array} \right. \right\}$$

Observe that the supports of the basis functions $\phi_{\underline{l}, \underline{j}}$ which span $W_{\underline{l}}$ are disjunct. These hierarchical difference spaces now allow the definition of a multilevel subspace decomposition and

$$V_n = V_{(n,n)} = \bigoplus_{l_1=0}^n \bigoplus_{l_2=0}^n W_{\underline{l}} = \bigoplus_{|\underline{l}|_{\infty} < n} W_{\underline{l}} \quad (4.15)$$

and, more generally

$$V_{\underline{n}} = \bigoplus_{l_1=0}^{n_1} \bigoplus_{l_2=0}^{n_2} W_{\underline{l}} = \bigoplus_{\underline{l} \leq \underline{n}} W_{\underline{l}} \quad (4.16)$$

where the symbol \leq is an element-wise relation. Figure 4.5 shows the hierarchical basis functions for a two-dimensional domain. Any function $f_n \in V_n$ can now be represented as

$$f_n(\underline{x}) = \sum_{|\underline{l}|_{\infty} \leq n} \sum_{\underline{j} \in B_{\underline{l}}} \alpha_{\underline{l}, \underline{j}} \cdot \phi_{\underline{l}, \underline{j}}(\underline{x})$$

and any function $f_{\underline{n}} \in V_{\underline{n}}$ as

$$f_{\underline{n}}(\underline{x}) = \sum_{\underline{l} \leq \underline{n}} \sum_{\underline{j} \in B_{\underline{l}}} \alpha_{\underline{l}, \underline{j}} \cdot \phi_{\underline{l}, \underline{j}}(\underline{x})$$

where $\alpha_{\underline{l}, \underline{j}} \in \mathbb{R}$ are the coefficients of the representation in the hierarchical tensor

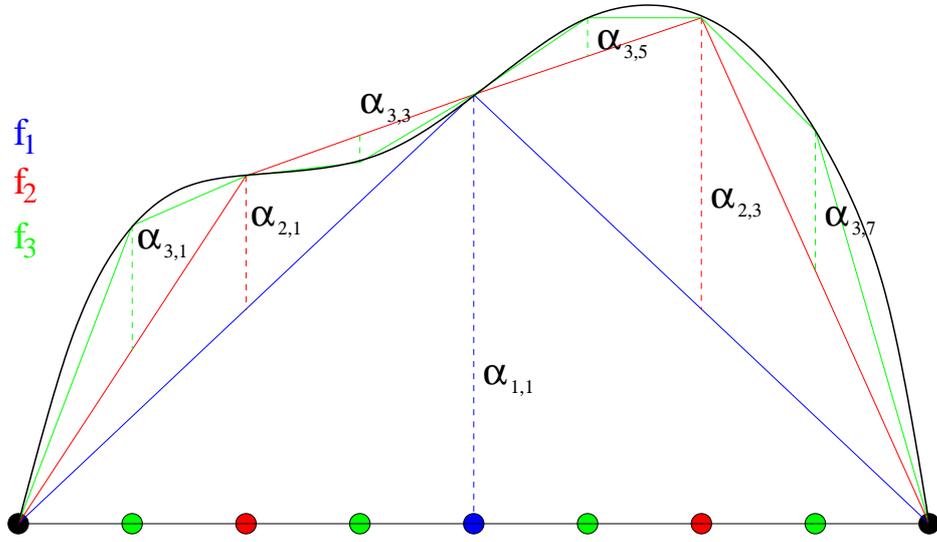


Figure 4.6: Example of a one-dimensional function and its approximations on different hierarchical basis.

product basis. The function

$$g_{\underline{l}}(\underline{x}) = \sum_{\underline{j} \in B_{\underline{l}}} \alpha_{\underline{l}, \underline{j}} \cdot \phi_{\underline{l}, \underline{j}}(\underline{x}) \quad (4.17)$$

is called the hierarchical component of f , belongs to $W_{\underline{l}}$ and allows to express $f_{\underline{n}}(\underline{x})$ as

$$f_{\underline{n}}(\underline{x}) = \sum_{\underline{l} \leq \underline{n}} g_{\underline{l}}(\underline{x})$$

See Figure 4.6 for an example of the projection of a function f_n on V_n and its representation (in one dimension).

It is now possible to build an approximation $f_n \in V_n$ of a function f using bilinear

interpolation. In one dimension, the hierarchical coefficients $\alpha_{l,j}$ are given by

$$\alpha_{l,j} = f(x_{l,j}) - \frac{f(x_{l,j-1}) + f(x_{l,j+1})}{2}$$

as shown in Figure 4.6. The second term of the right hand side actually corresponds to quantities that can be computed at the lower level of refinement $l - 1$. For that reason, when building a more and more refined interpolation of f , the $\alpha_{l,j}$ are called hierarchical surpluses as they specify what must be added to the hierarchical representation from the previous level $l - 1$.

So far, a new representation of functions in V_n has been presented. The following paragraph focuses on the bounds of a hierarchical component which will motivate the use of sparse grids. The H_{mix}^2 -norm is defined as

$$\|f\|_{H_{\text{mix}}^2}^2 = \sum_{\substack{0 \leq k_1 \leq 2 \\ 0 \leq k_2 \leq 2}} \left| \frac{\partial^{k_1+k_2} f}{\partial x_1^{k_1} \partial x_2^{k_2}} \right|_2^2$$

and the Sobolev space H_{mix}^2 can be defined with the above norm

$$H_{\text{mix}}^2 = \left\{ f : \Omega \rightarrow \mathbb{R}, \quad \|f\|_{H_{\text{mix}}^2} < \infty \right\}.$$

Notice that the norm includes not only all the derivatives up to second order, but also some mixed derivatives of third and fourth order.

The hierarchical components $g_{\mathbf{l}}$, as defined by (4.17) of a function $f \in H_{\text{mix}}^2$ satisfy

$$\|g_{\mathbf{l}}\|_2 \leq C 2^{-2(l_1+l_2)} |f|_{H_{\text{mix}}^2}$$

where $|f|_{H_{\text{mix}}^2}$ designates the semi-norm of f in H_{mix}^2

$$|f|_{H_{\text{mix}}^2} = \sum_{\substack{0 \leq k_1 \leq 2 \\ 0 \leq k_2 \leq 2 \\ k_1 + k_2 \geq 2}} \left| \frac{\partial^{k_1+k_2} f}{\partial x_1^{k_1} \partial x_2^{k_2}} \right|_2^2$$

This means that as the refinement level is increased, the L_2 -norm of the hierarchical components decrease and their contribution to the function representation decrease as well. Based on this bound on the hierarchical components, sparse grids were introduced as some of the less important hierarchical components are neglected. Instead of considering $V_n = \oplus_{|\underline{l}|_\infty \leq n} W_{\underline{l}}$, a new sparse grid function space of refinement level n is introduced $V_n^s = \oplus_{|\underline{l}|_1 \leq n} W_{\underline{l}}$. In Fig. 4.5, the sparse grid function space of refinement level 3 corresponds to the grids with the filled dots. Only the grids in the upper diagonal are considered. The refinement from level n to level $n+1$ then consists in adding the diagonal grids corresponding to $l_1 + l_2 = n+1$. Note that $V_n^s \subset V_n$ and that any function $f_n \in V_n^s$ can be represented by

$$f_n(\underline{x}) = \sum_{|\underline{l}|_1 \leq n} \sum_{\underline{j} \in B_{\underline{l}}} \alpha_{\underline{l}, \underline{j}} \cdot \phi_{\underline{l}, \underline{j}}(\underline{x})$$

It can then be shown that the number of inner grid points of grids corresponding to this approximation space V_n^s is $\mathcal{O}(h_n^{-1} \log(h_n^{-1}))$ and that the interpolation error of a function $f \in H_{\text{mix}}^2$ in the sparse grid space V_n^s is as follows

$$\|f - f_n^s\|_\infty = \mathcal{O}(h_n^2 \log(h_n^{-1}))$$

In comparison, a conventional bivariate grid method requires $\mathcal{O}(h_n^{-2})$ grid points and its accuracy is $\mathcal{O}(h_n^2)$. Thus, for the same level of refinement, a conventional grid

provides better accuracy but requires more grid points. Depending on the smoothness of the function to tabulate, for the same number of grid points, a sparse grid may provide better accuracy than its conventional counterpart. Similar results and conclusions hold for higher dimensional functions as well. In fact, the gain offered by sparse grids increases with the number of dimensions. This explains the use of sparse grids for high-dimensional problems where the curse of dimensionality prevents standard methods to be used.

Sparse grids also present the advantage of defining an automatic stopping criterion to the level of refinement. Until that criterion is met, the level of refinement of the sparse grid is increased and only a limited number of computations is done due to the hierarchical nature of the shape functions associated with the sparse grid. The stopping criterion is expressed in terms of the hierarchical surpluses, the level of refinement is increased until the following is satisfied

$$\max_{j \in B_l} \alpha_j < \max(\epsilon_{\text{rel}}(f_{\text{max}} - f_{\text{min}}), \epsilon_{\text{abs}})$$

where ϵ_{rel} and ϵ_{abs} are relative and absolute accuracy criteria respectively.

The approach presented so far does not allow for any adaptive refinement, in the sense that the refinement procedure is predetermined. As explained earlier, refinement from one level to the next consists in adding the new “diagonal” hierarchical spaces to the approximation space. A localized refinement of the sparse grid as often done on a full grid is not simple. Instead, a dimensional adaptive refinement can be considered. The conventional sparse grid approach treats all dimensions equally, meaning that the number of grid points is the same in each coordinate direction. However, not all variables have the same importance in a problem, which translates into different levels of importance for the dimensions of the sparse grid. Unfortunately,

the importance of each variable with respect to the other is not known a priori and an “exhaustive” exploration of the objective function can become too expensive. A specific dimension-adaptivity refinement was introduced in [85], so that conventional sparse grids of the form V_n and their hierarchical decompositions (4.15) are replaced by dimension-adaptive sparse grids of the form $V_{\underline{n}}$ and their hierarchical decompositions (4.16). To build the new sparse grid, instead of adding the basis functions of all the “diagonal” hierarchical spaces, the new algorithm adds only the basis functions of only one hierarchical space to the existing grid. In order to choose which one of these hierarchical space to consider, estimations of the error of the potential hierarchical spaces are computed. In the two-dimensional case, for each index (k_1, k_2) corresponding to functions already in the sparse grid, the errors at the levels $(k_1 + 1, k_2)$ and $(k_1, k_2 + 1)$ are computed. Only the functions corresponding to the largest estimated error are added to the sparse grid. However, one caveat of this greedy dimension adaptivity is the possible underestimate of the real error with respect to some dimensions, leading to a loss of the global convergence property of sparse grid interpolants. To correct this default, a degree of dimensional adaptivity specified by the user is introduced. It allows to gradually shift emphasis between greedy dimensional adaptive refinement and conventional refinement.

4.3.4 Application to an Analytical Function

The tabulation of an analytical function by sparse grids is considered in this section.

The bivariate function $f(x, y)$ is given by

$$f(x, y) = \left(\frac{5}{\pi}x - \frac{5.1}{4\pi^2}x^2 + y - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x + 10$$

and the domain of interest is $[-5, 10] \times [0 : 15]$. A plot of the function is given by Fig. 4.7. The function is then tabulated using the sparse grid method. A number of 1473 grid points were necessary to reach a maximum absolute error of 0.007 and a maximum relative error of 4.3×10^{-4} . An approximate solution is reconstructed by interpolation on the sparse grid and is shown in Fig. 4.8. The absolute and relative errors on the whole domain are also plotted in Fig. 4.9 and 4.10. The sparse grid method with linear shape functions seems adapted to the interpolation of functions for relative accuracies of the order of 10^{-3} .

4.3.5 Application to the JWL EOS

In the present work, the sparse grid is used to tabulate the Riemann invariants of the JWL equation of state and to allow for its linear interpolation. To ensure that the interpolation accuracy estimation above holds a sufficient condition is that the Riemann invariant belongs to the Sobolev space H_{mix}^2 . Since the Riemann invariant is given by (4.14) and since the speed of sound is given by the square root of Eq. (2.15), the Riemann invariant is C^∞ as long as

$$c^2(\rho, s) = \frac{(\omega + 1)s\rho^{\omega+1} + (\omega + 1) \sum_{i=1}^2 A_i e^{-\frac{R_i \rho_0}{\rho}} - f(\rho) + \rho f'(\rho)}{\rho}$$

is positive. In the physical domain of interest which considers only densities above 1.0kg.m^{-3} in practice, the above quantity is always positive and thus the Riemann invariant and its derivatives are all bounded and integrable on the domain of interest. Hence, the Riemann invariant belongs to H_{mix}^2 and the above sparse grid framework is applicable to the tabulation of the Riemann invariant (4.14).

The domain of interest is determined by $(\rho, s) \in [3.0, 1630.0] \times [-10^6, 10^{10}]$ (in

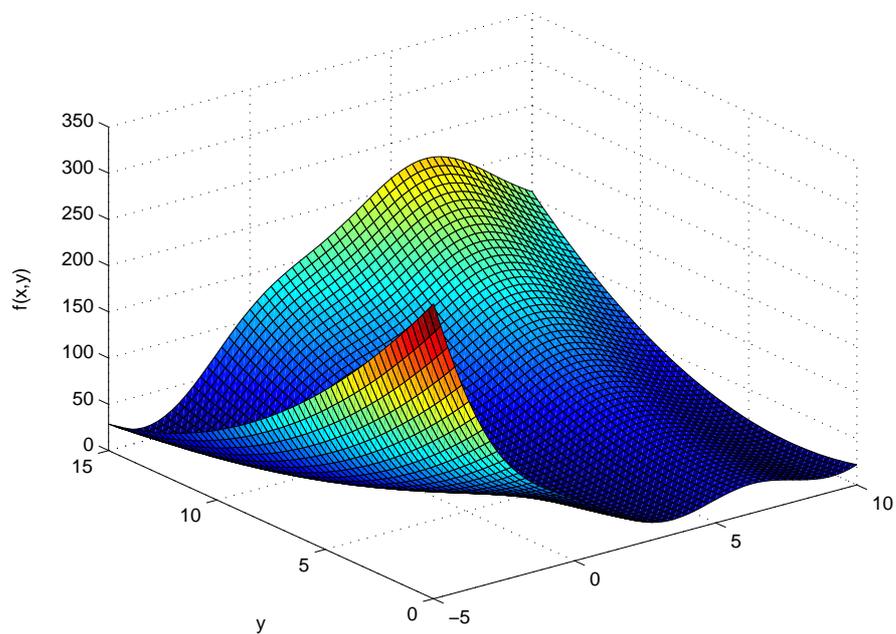


Figure 4.7: Analytical example: solution obtained from the analytical formula

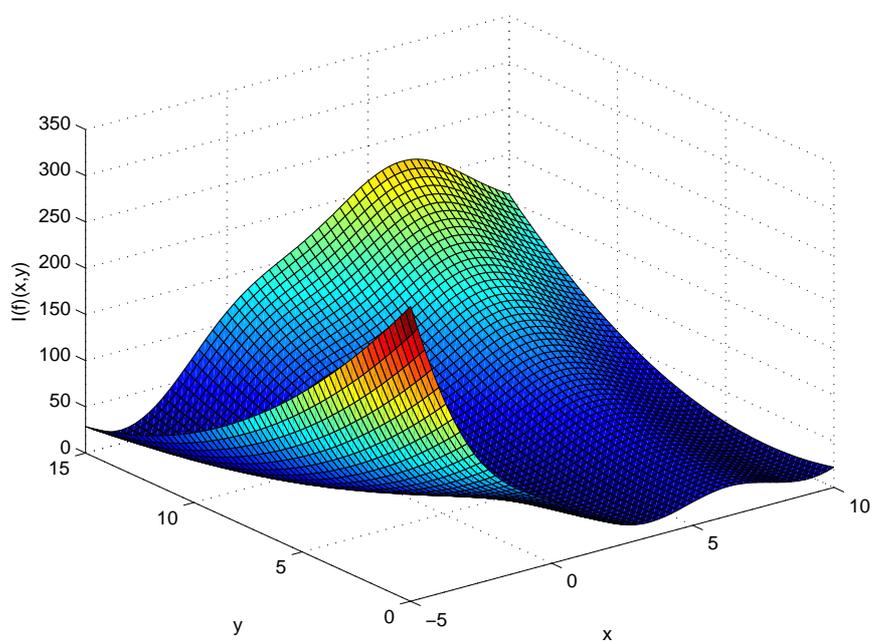


Figure 4.8: Analytical example: solution obtained from the sparse grid by interpolation

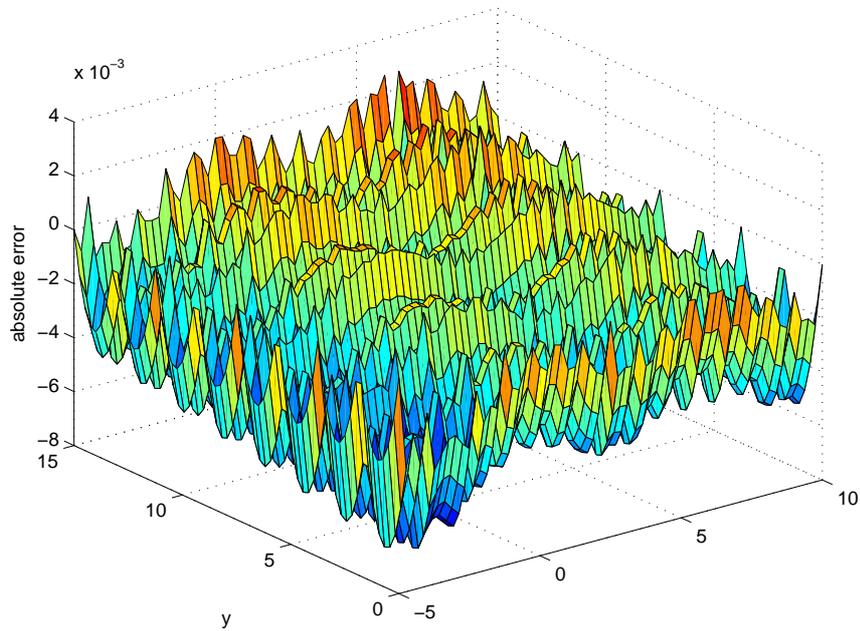


Figure 4.9: Analytical example: absolute errors

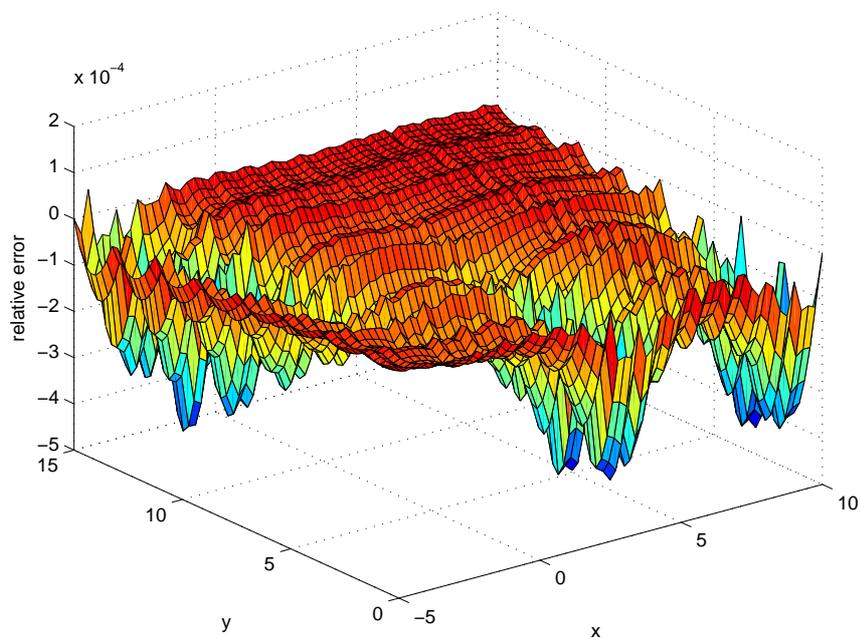


Figure 4.10: Analytical example: relative errors

kg.m³ and J.K⁻¹ respectively) if one considers explosions with initial pressures in the detonated gas between 10⁵ and 10¹⁰ Pa. The accuracy of the tabulations with respect to the total number of nodes in the sparse grids is presented in Table 4.3.5. It was

number of grid points	refinement level	relative accuracy	absolute accuracy
1089	8 × 7	3.68 × 10 ⁻²	5.21 × 10 ⁴
10753	11 × 10	1.03 × 10 ⁻²	1.45 × 10 ⁴
102401	13 × 13	1.90 × 10 ⁻³	2.63 × 10 ³
589825	15 × 15	4.21 × 10 ⁻⁴	5.95 × 10 ²

Table 4.3: Accuracy of the sparse grid tabulation of the Riemann invariant for the JWL EOS with respect to the number of grid points and the level of refinement in each direction. The range of the tabulation is $(\rho, s) \in [3.0, 1630.0] \times [-10^6, 10^{10}]$.

found that using the ODE-integration approach or the sparse grid interpolation in the resolution of a (single- or two-phase) Riemann problem with a JWL rarefaction led to identical solutions within 0.1% differences at most, when the sparse grid had a relative accuracy of 0.001. Also, using a sparse grid was found to decrease the computational time of the solution of the Riemann problem by approximately 80% over the use of an ODE-integration-based approach.

The use of sparse grids for equations of state with no isentropic relation can be expected to be more advantageous in terms of computational time over ODE-integration approaches, memory gain over full grids as long as the equation of state satisfies some smoothness properties. However, this has not been tested in the present work.

4.4 Second-Order Reconstruction

Second-order spatial accuracy is achieved through a Monotonic Upwind Scheme for Conservation Laws (MUSCL) approach for the interior nodes of a fluid. At the

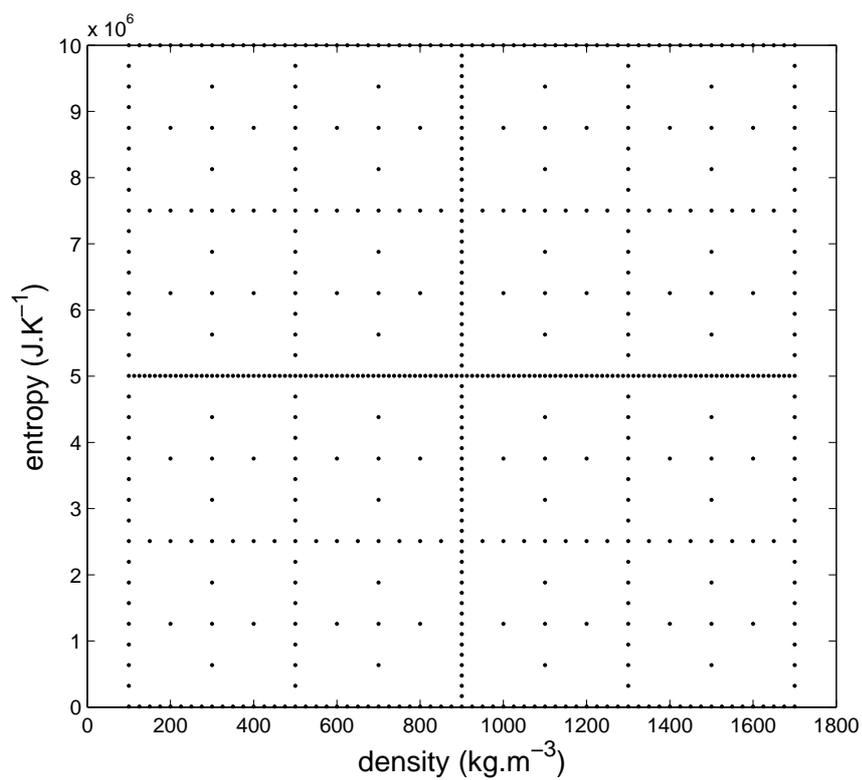


Figure 4.11: Example of a two-dimensional sparse grid for the tabulation of the JWL Riemann invariant in the (ρ, s) -range $[100, 1700] \times [10^3, 10^7]$

interface between two fluids, the corresponding approach would lead to the violation of the important concept of this method. The interfacial numerical flux is such that the discretization stencil of a node does not include, except through the use of the exact Riemann solver, nodes belonging to the other fluid. Thus, special care must be taken to reconstruct the state variables at the interface. Indeed, contrary to what is shown in section 4.6, the inadequate reconstruction of the variables at the interface can lead to the loss of the contact preserving property of the above algorithm.

Depending on the point of view adopted, different reconstructions can be envisioned. In a finite volume setting, fluxes are computed at the boundaries of the control volumes. Thus, if such a point of view is adopted, similarly to (3.4)-(3.5), the reconstruction of the variables at the interface is performed as

$$\begin{aligned}\mathbf{U}_{ij} &= \mathbf{U}_i + \frac{1}{2} (\nabla \mathbf{U})_{ij} \cdot \overrightarrow{x_i x_j} \\ \mathbf{U}_{ji} &= \mathbf{U}_j + \frac{1}{2} (\nabla \mathbf{U})_{ji} \cdot \overrightarrow{x_j x_i}\end{aligned}$$

If the fact that the interface does not actually coincide with the boundaries of control volumes is considered, the following reconstruction must be considered

$$\begin{aligned}\mathbf{U}_{ij} &= \mathbf{U}_i + (\nabla \mathbf{U})_{ij} \cdot \overrightarrow{x_i x_{ij}} \\ \mathbf{U}_{ji} &= \mathbf{U}_j + (\nabla \mathbf{U})_{ji} \cdot \overrightarrow{x_j x_{ji}}\end{aligned}$$

where, in that case, the distance of point i (j) to the interface along the edge $i - j$ is

computed as follows

$$\begin{aligned}\overrightarrow{x_i x_{ij}} &= \frac{|\phi_i|}{|\phi_i| + |\phi_j|} \overrightarrow{x_i x_j} \\ \overrightarrow{x_j x_{ji}} &= \frac{|\phi_j|}{|\phi_i| + |\phi_j|} \overrightarrow{x_j x_i}\end{aligned}$$

which assumes that the interface has locally no curvature. Numerical tests showed that the choice of either reconstruction does not significantly change solutions.

In both these reconstructions, the gradients $(\nabla \mathbf{U})_i$ and $(\nabla \mathbf{U})_j$ are computed by solving a least-square problem similarly to what was done in [2], except that for a given node only neighbors from the same side of the interface are considered

$$\mathbf{U}_j - \mathbf{U}_i = (\nabla \mathbf{U})_i \cdot \overrightarrow{x_i x_j} \quad \text{for } j \in \mathcal{N}(i, \phi_i) = \{k \in \mathcal{N}(i), \phi_i \phi_k > 0\}$$

If the number of nodes is not sufficient to have a well-posed least-square problem, the gradient is simply set to zero. This can happen when the flow features have roughly the same size as the mesh, meaning that the flow features may not be well resolved in that area.

Finally, the Van Albada limiter is used to make sure that the slopes on both sides of the interface have the same sign and lead to reconstructed states that lie in between the two original states.

4.5 Time Discretization and Complete Algorithm

Several issues need to be recognized for the time integration of the two-phase flow equations. Both the Euler equations and the level set equation must be time-integrated simultaneously. The level set values have a direct impact on the computation of the

fluxes of the Euler equations, since they determine first if the computation of the fluxes is between two nodes in a same fluid or separated by an interface, and second, which equation(s) of state must be used in the computation of these fluxes. However, while the two sets of equations are highly dependent on the solution of the other, the Euler equations and the level set equation do not have to be seen as forming one single set of equations and both monolithic and staggered time-integrations can be considered. Both explicit and implicit schemes can be considered to integrate the fluid equations of motion. While the first ones provide greater accuracy, the second ones provide better stability properties. In the context of developing numerical tools to simulate implosion problems, other requirements must be considered. In particular, when cracking of the structure occurs, numerical methods for the time-integration of the structure equations of motion do not allow for the use of an implicit scheme, and the time step is further decreased compared to the one when no cracking occurs. This limitation of the time step in the structure time-integration led to the development of explicit time-integrations scheme for the fluid equations of motion. As seen in the previous chapter, time-integration was considered on moving grids within the ALE framework. It is also in that framework that the time-integration scheme for multiphase flow problem was developed. From an accuracy point of view, the time-integration is required to be second-order even at the interface to match the time accuracy in the rest of the flow. While it seems obvious how to apply the ideas presented in section 4.2 to advance the Euler equations and the level set equation with a forward Euler scheme, the extension to an explicit multi-stage scheme is not trivial.

To this effect, the complete interfacial treatment for the Euler equations and the

physical variables of the flow is first presented. Then, two different explicit time-integration schemes based on second-order Runge-Kutta schemes are proposed.

In the present section, $\mathcal{N}(i)$ denotes the set of neighboring nodes to node i , $\mathcal{N}_+^n(i) = \{j \in \mathcal{N}(i) \mid \phi_i^n \phi_j^n > 0\}$ denotes the set of neighboring nodes to node i that are in the same fluid as node i , and $\mathcal{N}_-^n(i) = \{j \in \mathcal{N}(i) \mid \phi_i^n \phi_j^n < 0\}$ denotes the set of neighboring nodes to node i that are in a different fluid than node i .

4.5.1 Interfacial Treatment

It is assumed that at time t^n , state variable values U and level set values ϕ are available at all nodes i of the computational domain. At each pair of neighboring nodes $i - j$ where $j \in \mathcal{N}_-^n(i)$, the numerical fluxes are computed as follows:

- computation of the normal of the material interface using the level set

$$\mathbf{n}_\phi = \frac{1}{2} \left((\nabla \phi)_i + (\nabla \phi)_j \right)$$

- decomposition of the velocities at both nodes in normal and tangential components $u_{nk} = (\mathbf{u}_k \cdot \mathbf{n}_\phi)$ and $\mathbf{u}_{tk} = \mathbf{u}_k - u_{nk} \mathbf{n}_\phi$ where $k \in \{i, j\}$.
- solving for the constant interfacial states

$$(\rho_i^{\mathcal{R}}, u_n^{\mathcal{R}}, p^{\mathcal{R}})$$

$$(\rho_j^{\mathcal{R}}, u_n^{\mathcal{R}}, p^{\mathcal{R}})$$

of the one-dimensional Riemann problem as described in section 4.2.1 with initial conditions

$$(\rho_i, u_{ni}, p_i)$$

$$(\rho_j, u_{nj}, p_j)$$

- reconstruction of the three-dimensional interfacial states

$$\mathbf{V}_i^{\mathcal{R}} = (\rho_i^{\mathcal{R}}, u^{\mathcal{R}} \mathbf{n}_\phi + \mathbf{u}_{ti}, p^{\mathcal{R}})^T$$

$$\mathbf{V}_j^{\mathcal{R}} = (\rho_j^{\mathcal{R}}, u^{\mathcal{R}} \mathbf{n}_\phi + \mathbf{u}_{tj}, p^{\mathcal{R}})^T$$

- computation of the interfacial fluxes

$$\Phi_{ij} = \Phi(\mathbf{U}_i, \mathbf{U}_i^{\mathcal{R}}, EOS_i, \nu_{ij}, \kappa_{ij}) = \tilde{\Phi}(\mathbf{U}_i, \mathbf{U}_j, \nu_{ij}, \kappa_{ij}, \phi_i, \phi_j)$$

$$\Phi_{ji} = \Phi(\mathbf{U}_j, \mathbf{U}_j^{\mathcal{R}}, EOS_j, \nu_{ji}, \kappa_{ji}) = \tilde{\Phi}(\mathbf{U}_j, \mathbf{U}_i, \nu_{ji}, \kappa_{ji}, \phi_j, \phi_i)$$

In order to advance all flow variables, the general flow chart is given by:

- for each edge (i, j) , the fluxes are computed either through the above routine or via the usual Roe flux
- the Euler equations are time-integrated from t^n to t^{n+1} using the previously computed fluxes
- at the end of the iteration, if any node i has changed fluids in the time-step – meaning if $\phi_i^n \phi_i^{n+1}$ is negative –, the value \mathbf{U}_i^{n+1} is updated by the constant extrapolation from a three-dimensional interfacial state or from a neighboring node state

4.5.2 Second-Order Runge-Kutta (I)

An explicit second-order Runge-Kutta scheme is considered to integrate both the Euler equations and the level set equation. The scheme is monolithic as the two sets of equations are integrated as one single set.

Consider the solutions \mathbf{U}^n and ϕ^n at time t^n for all nodes of the computational domain. Intermediate states $\mathbf{U}^{(1)}$ are computed for all nodes according to

$$\Omega_i^{n+1} \mathbf{U}_i^{(1)} = \Omega_i^n \mathbf{U}_i^n - \Delta t \mathbf{f}_i(\mathbf{U}^n, \nu, \kappa, \phi^n)$$

where ν and κ are specified by Eqs. (3.1)-(3.2), and the fluxes are given either by the above procedure if two neighboring nodes are in different fluids or by Eq.(3.3) otherwise, allowing to write the general form:

$$\mathbf{f}_i(\mathbf{U}, \nu, \kappa, \phi) = \sum_{j \in \mathcal{N}_-^n(i)} \tilde{\Phi}(\mathbf{U}_i, \mathbf{U}_j, EOS_i, \nu_{ij}, \kappa_{ij}, \phi_i, \phi_j) + \sum_{j \in \mathcal{N}_+^n(i)} \Phi(\mathbf{U}_i, \mathbf{U}_j, \nu_{ij}, \kappa_{ij})$$

Similarly, the level set equation is integrated

$$\Omega_i^{n+1} (\rho\phi)_i^{(1)} = \Omega_i^n (\rho\phi)_i^n - \Delta t \psi_i(\phi^n, \nu, \kappa, \mathbf{U}^n)$$

where ψ_i denotes the numerical flux to approximate the level set equation flux at node i . In the second step of the Runge-Kutta scheme, intermediate states $\mathbf{U}^{(2)}$ are computed for all nodes according to

$$\Omega_i^{n+1} \mathbf{U}_i^{(2)} = \Omega_i^n \mathbf{U}_i^n - \frac{\Delta t}{2} (\mathbf{f}_i(\mathbf{U}^{(1)}, \nu, \kappa, \phi^n) + \mathbf{f}_i(\mathbf{U}^n, \nu, \kappa, \phi^n))$$

Slightly differently, the level set equation is integrated by

$$\Omega_i^{n+1}(\rho\phi)_i^{n+1} = \Omega_i^n(\rho\phi)_i^n - \frac{\Delta t}{2} (\psi_i(\phi^{(1)}, \nu, \kappa, \mathbf{U}^{(1)}) + \psi_i(\phi^n, \nu, \kappa, \mathbf{U}^n))$$

Finally, the phase-changing node update algorithm presented in section 4.2.4 is used to update all nodes i such that $\phi_i^n \phi_i^{n+1} < 0$:

$$\mathbf{U}_i^{n+1} = \frac{1}{\sum_{k \in \mathcal{N}(i)} w_k} \sum_{k \in \mathcal{N}(i)} w_k \mathbf{U}_k^*$$

where \mathbf{U}_k^* is given by Eq. (4.9) and $w_k = \max\left(0, \frac{\overrightarrow{x_k x_i}}{|x_k x_i|} \cdot \frac{\mathbf{u}_k^n}{|\mathbf{u}_k^n|}\right)$.

Note that in the computation of $\mathbf{U}_i^{(2)}$, the first flux term considers not the position of the level set after the first step of the scheme, but the one at the beginning of the time iteration. This modification from a conventional second-order Runge-Kutta scheme is done to avoid summing up two fluxes for two different fluids in the middle of the time-step. If a conventional time-integration was considered, this could happen in two cases when $\Phi_i^n > 0$, $\Phi_i^{(1)} < 0$, $\Phi_i^{n+1} < 0$ or when $\Phi_i^n > 0$, $\Phi_i^{(1)} < 0$, $\Phi_i^{n+1} > 0$, in which cases the first flux term would be based on an equation of state corresponding to positive level set while the second one would be based on the other equation of state. While this ‘‘inconsistency’’ of the two flux terms would be eventually discarded since the value of \mathbf{U}_i^{n+1} would be overwritten by an extrapolation \mathbf{U}_k^* in the first case, that ‘‘inconsistency’’ could significantly affect the results in the second case. In addition, by considering only one level set configuration for the computation of the Euler fluxes, the phase-changing node update needs only be performed at the end of the iteration. While the Euler equations are time-integrated on the configuration given at the beginning of the iteration, the level set equation takes fully benefit of

the multi-stage Runge-Kutta scheme.

4.5.3 Second-Order Runge-Kutta (II)

While the previous scheme integrates the two sets of equations in a monolithic fashion, a staggered version can be developed in an attempt to increase the stability of the overall scheme. Indeed, if a staggered implicit scheme were considered to integrate the Euler equations and the level set equation, the procedure would consist in advancing the fluid solution by one time step knowing the level set at the beginning of the time step and then in advancing the level set solution using the fluid solution at the end of the time step. The same notations are used as previously. Consider the solutions \mathbf{U}^n and ϕ^n at time t^n for all nodes of the computational domain. Intermediate states $\mathbf{U}^{(1)}$ are computed for all nodes according to

$$\Omega_i^{n+1} \mathbf{U}_i^{(1)} = \Omega_i^n \mathbf{U}_i^n - \Delta t \mathbf{f}_i(\mathbf{U}^n, \nu, \kappa, \phi^n)$$

with

$$\mathbf{f}_i(\mathbf{U}, \nu, \kappa, \phi) = \sum_{j \in \mathcal{N}_-^n(i)} \tilde{\Phi}(\mathbf{U}_i, \mathbf{U}_j, EOS_i, \nu_{ij}, \kappa_{ij}, \phi_i, \phi_j) + \sum_{j \in \mathcal{N}_+^n(i)} \Phi(\mathbf{U}_i, \mathbf{U}_j, \nu_{ij}, \kappa_{ij})$$

The level set equation is integrated with the knowledge of $\mathbf{U}_i^{(1)}$

$$\Omega_i^{n+1} (\rho\phi)_i^{(1)} = \Omega_i^n (\rho\phi)_i^n - \Delta t \psi_i(\phi^n, \nu, \kappa, \mathbf{U}^{(1)})$$

Intermediate states $\mathbf{U}^{(2)}$ are computed for all nodes according to

$$\Omega_i^{n+1} \mathbf{U}_i^{(2)} = \Omega_i^n \mathbf{U}_i^n - \frac{\Delta t}{2} (\mathbf{f}_i(\mathbf{U}^{(1)}, \nu, \kappa, \phi^n) + \mathbf{f}_i(\mathbf{U}^n, \nu, \kappa, \phi^n))$$

similarly to what was done in the previous scheme. The level set equation is then integrated

$$\Omega_i^{n+1} (\rho\phi)_i^{n+1} = \Omega_i^n (\rho\phi)_i^n - \frac{\Delta t}{2} (\psi_i(\phi^{(1)}, \nu, \kappa, \mathbf{U}^{(2)}) + \psi_i(\phi^n, \nu, \kappa, \mathbf{U}^{(1)}))$$

Again, the phase-changing node update algorithm presented in section 4.2.4 is used only at the end of the time iteration. The present scheme is only a variation of the previous one and thus presents similar characteristics, except for the expected increase in stability.

Finally, note that both schemes are based on an ALE formulation and that they satisfy their DGCL. This means that the constant solution is preserved in both cases. This property plays a crucial role in satisfying the contact-preserving property for both schemes.

4.6 Contact-Preserving Property

Consider a material front with the contact conditions $u_L^n = u_R^n = u$, $p_L^n = p_R^n = p$, but $\rho_L^n \neq \rho_R^n$, where the subscripts L and R designate the left and right sides of the front. The nodes of the computational mesh can be divided in two groups: one where each node has as all its neighbors in the same fluid medium as itself, and one where each node has at least one neighboring node that lies in a different fluid medium than itself.

Consider a node i in the first group of nodes. Since all its neighbors are in the same fluid medium as itself, it follows that

$$\forall j \in \mathcal{N}(i) \quad \mathbf{U}_j^n = \mathbf{U}_i^n$$

Hence,

$$f_i(\mathbf{U}^n) = \sum_{j \in \mathcal{N}(i)} |\partial\Omega_{ij}| \Phi(\mathbf{U}_i^n, \mathbf{U}_i^n, \nu_{ij}, \kappa_{ij})$$

In the developed framework of time-integrators on moving grids, constant quantities are integrated exactly if the considered time-integrator satisfies the DGCL. Thus,

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n$$

which implies that the state of contact is preserved.

Next, consider a node i in the second group of nodes described above, and consider a neighboring node j that belongs to a different fluid medium than that of node i . During the solution of the two-phase Riemann problem, the densities, normal velocities, and pressures at these nodes are input to the exact Riemann solver. If the normal velocities are computed using the normal to the material interface, the exact Riemann solver delivers the input itself as the solution. In this case, $\rho_{I_L}^n = \rho_L^n$, $\rho_{I_R}^n = \rho_R^n$, $u_I^n = u$, and $p_I^n = p$. Consequently, $\mathbf{U}_i^{\mathcal{R}^n} = \mathbf{U}_i^n$, $\mathbf{U}_j^{\mathcal{R}^n} = \mathbf{U}_j^n$,

$$\Phi_{ij} = \Phi(\mathbf{U}_i^n, \mathbf{U}_i^n, \text{EOS}_i, \nu_{ij}, \kappa_{ij})$$

$$\Phi_{ji} = \Phi(\mathbf{U}_j^n, \mathbf{U}_j^n, \text{EOS}_j, \nu_{ji}, \kappa_{ij})$$

and therefore

$$F_i(\mathbf{U}) = \sum_{j \in \mathcal{N}(i)} |\partial\Omega_{ij}| \Phi_{ij}(\mathbf{U}_i, \mathbf{U}_i, \nu_{ij}, \kappa_{ij}) = 0$$

and

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n$$

for the same reasons as in the previous case. This concludes the proof that as long as the input to the exact Riemann solver is computed using the normal to the material interface — and not the normal to the control volumes — the proposed method is contact preserving.

Finally, given that the material interface moves in time, a node i on one side of the material interface at time t^n can become on the other side of this interface at t^{n+1} . To preserve the structure of the solution of the contact problem at t^{n+1} , \mathbf{U}_i^{n+1} needs to be properly updated. This is done in Step (5) of the method where \mathbf{U}_i^{n+1} is overwritten by $\mathbf{U}_j^{\mathcal{R}^n} = \mathbf{U}_j^n$ in order to preserve the state of contact.

4.7 Mass Conservation

The fluxes between two nodes i and j satisfy the conservative property

$$\Phi(\mathbf{U}_i, \mathbf{U}_j, \nu_{ij}, \kappa_{ij}) = \Phi(\mathbf{U}_j, \mathbf{U}_i, -\nu_{ij}, -\kappa_{ij})$$

everywhere in the computational domain except at the material interface between two different fluids. This explains the lack of conservation properties of total quantities, that is the sum of quantities from both fluids. The computation of the interfacial fluxes relies heavily on the solution of the Riemann problem. Since the values used

for both interfacial fluxes may be significantly different, an important lack of conservation is not impossible. However, as noticed by Fedkiw in [73], the interfacial flux computation and hence the loss of conservation take place on a space with one lower dimension than the one of the computational domain and conservation errors can be expected to be fairly small. The variations of the total quantities should only be due to the boundary fluxes as no source terms are considered in the present model. Thus, conservation of these total quantities is measured by

$$\frac{\sum_{\Omega_i \in \Omega} \Omega_i \mathbf{U}_i^n}{\sum_{\Omega_i \in \Omega} \Omega_i \mathbf{U}_i^0 - \sum_{k=1}^n \Delta t^k \Phi_{\Gamma}^k}$$

where Φ_{Γ}^k represents the sum of the fluxes at the domain boundaries between time steps $k - 1$ and k . For a numerically conservative scheme in total quantities, this ratio should always be equal to 1. However, the state vectors \mathbf{U}_i do not represent an average of both fluids in mixed cells and the volumes occupied by each fluid are not tracked at the subgrid level. As a result, the above ratio can never be constantly equal to one, even for a perfect moving contact discontinuity. As seen in the next section, the ratio oscillates around the value 1 as the contact moves and partially covers and uncovers nodes.

4.8 One-Dimensional Two-Phase Flow Benchmark Problems

The present method is tested on academic one-dimensional Riemann problems for which an exact solution is known. The length of the tube is unity in the x-direction. Even though the problems are one-dimensional by nature, three-dimensional meshes

are considered with 201 lengthwise grid points, unless specified otherwise. The semi-discretization of the equations is given in chapters 3 and 4. No low Mach preconditioner is used for these tests. The time-integration scheme considered for these tests is given in section 4.5.3. The CFL, as well as the initial conditions and the final time of the simulation, is specified separately for each problem. The following notation is adopted to specify the initial conditions of the problem

$$(\rho, u, p, \text{EOS}) \tag{4.18}$$

where ρ , u and p designate density, velocity and pressure, while EOS refers to the equation of state considered and can be $\text{PG}(\gamma)$ for a perfect gas EOS, $\text{SG}(\gamma, p_{sg})$ for a stiffened gas EOS, $\text{Tait}(k_1, k_2, \rho_0, p_0)$ for a barotropic fluid following Tait EOS and $\text{JWL}(\omega, A_1, R_1, A_2, R_2, \rho_0)$ for a JWL EOS. The parameters of each EOS have been described in section 2.3.

In order to compare the present method to existing ones, the Ghost Fluid Method (GFM) [73] and the Ghost Fluid Method for the Poor (GFMP) [65] are considered. Both have been briefly introduced in section 4.1.

PG-PG Interaction

This case considers the so-called Sod shock-tube problem between perfect gas with different specific heat ratios, which has been widely used to test compressible multiphase flow algorithms in the literature [73, 64, 65]. The initial conditions and the EOS are given by

$$(\rho, u, p, \text{EOS}) = \begin{cases} (1.0, 0.0, 1.0, \text{PG}(1.4)) & \text{if } x < 0.5 \\ (0.125, 0.0, 0.1, \text{PG}(1.2)) & \text{if } x > 0.5 \end{cases}$$

The equations are time integrated until the final time 0.2 with a CFL condition of 0.9. As shown by Fig. 4.12, the computed solutions at the material interface are shown to be in very good agreement with the analytical solution. In particular, pressure and velocity are continuous at the interface. Both the material interface and the shock are properly located. As expected from our method, the material interface is sharp whereas the shock is smeared. The solution is most degraded after the shock, where an overshoot of the velocity and an undershoot of the pressure are present. As mentioned earlier, numerical study of the mass conservation show that the algorithm does not conserve mass. At the initial interaction between the two fluids, the total mass increases by less than 1%. Once the contact is established, the total mass oscillates around a mean value due to the covering and uncovering of new nodes by both fluids as mentioned in section 4.7.

PG-PG Stiff Interaction

This case considers a stiffer shock-tube problem. While the EOS are both perfect gas with different specific heat ratio, the initial pressure jump is increased from the previous case. The initial conditions and the EOS are given by

$$(\rho, u, p, \text{EOS}) = \begin{cases} (1.0, 0.0, 500.0, \text{PG}(1.4)) & \text{if } x < 0.5 \\ (1.0, 0.0, 0.2, \text{PG}(1.667)) & \text{if } x > 0.5 \end{cases}$$

The equations are time integrated until the final time 0.015 with a CFL condition of 0.9. The overall solution given by the present method is again in good agreement with the analytical solution (see Fig. 4.13) even though small oscillations can be observed. Obviously though, the right interfacial state is more difficult to capture than in the previous case, as the density is not constant between the material interface and the

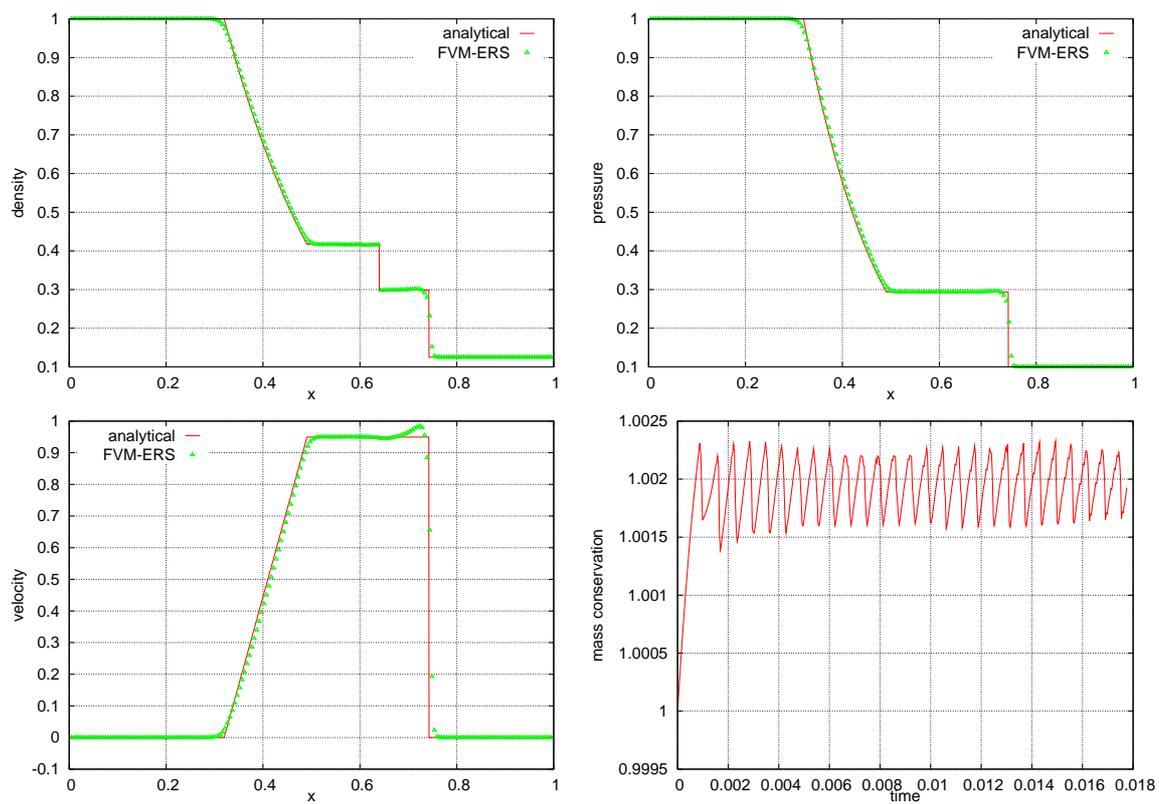


Figure 4.12: Solutions of the Sod shock-tube problem given by the FVM-ERS in comparison to the exact solution at $t = 0.2$ as well as mass conservation of the problem.

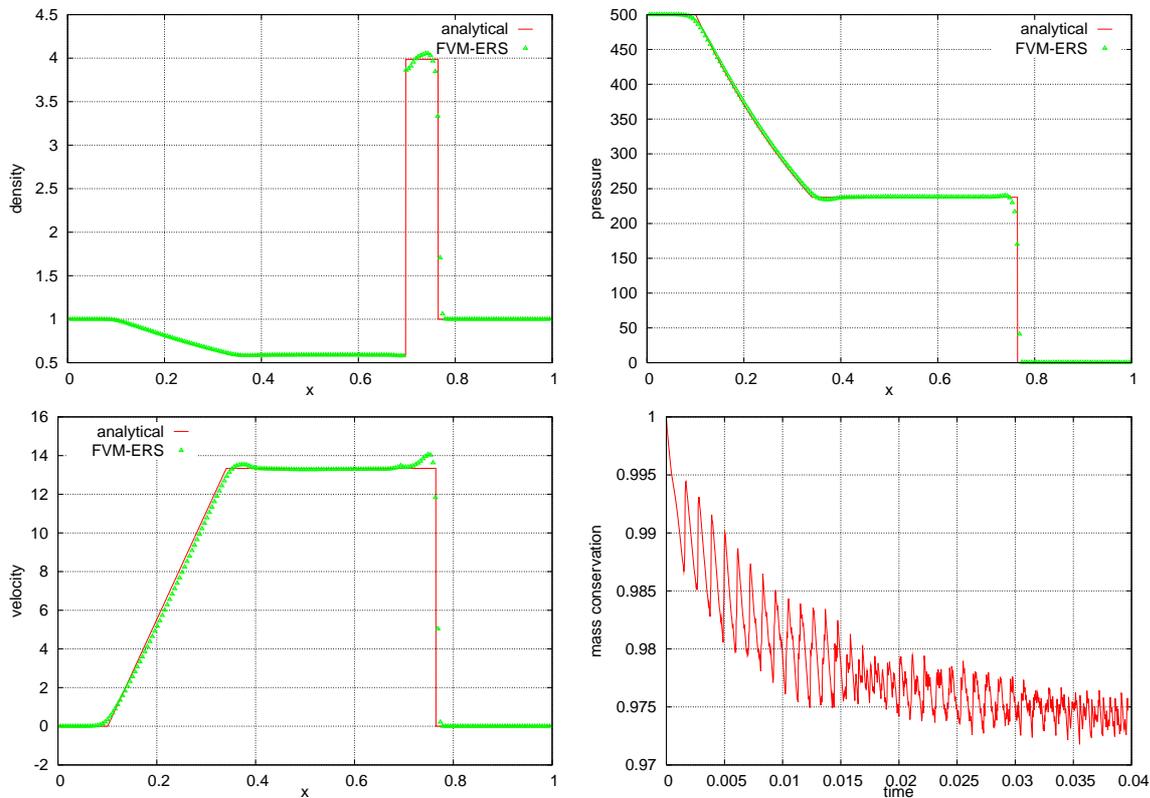


Figure 4.13: Solutions of the shock-tube problem given by the FVM-ERS in comparison to the exact solution at $t = 0.015$ as well as mass conservation of the problem.

shock. Again both locations of the shock and the material interface are correct. Similar to the previous case, a small velocity jump develops at the shock. The total mass loss of the order of 2.5% is more severe.

PG-PG Shock-Interface Interaction

This case considers the case of a reflection-less shock, which was also discussed in [86, 92]. In this case, the membrane is positioned at $x = 0.2$ and separates two

perfect gases whose initial states and EOS constants are

$$(\rho, u, p, \text{EOS}) = \begin{cases} (3.2, 9.43499279, 100.0, \text{PG}(5/3)) & \text{if } x < 0.2 \\ (1.0, 0.0, 1.0, \text{PG}(1.2)) & \text{if } x > 0.2 \end{cases}$$

This problem is easier than the previous one from the density ratio viewpoint. However, it is more challenging than the previous problem from the following viewpoint. The exact solution of this problem consists of a shock wave and a contact discontinuity only that propagate to the right side of the material interface. On the other hand, most if not all numerical methods applied to the solution of this problem can be expected to generate a non-physical reflection at the material interface and therefore produce a solution containing also a wave that propagates to the left of the material interface. This is because of so-called start-up errors that is, errors due to the unavoidable inexact representation of the initial conditions. In this sense, this problem allows to evaluate the sensitivity of a computational method to imperfect initial data.

The CFL number is set to 0.8. The results obtained at $t = 0.06$ are reported in Fig. 4.14. Some small amplitude oscillations can be observed in the computed solutions. They are due to the reflection at the material interface. As mentioned in [86], this spurious reflection is difficult to remove. The numerical solution delivered here by the FVMERS appears to be more accurate than that reported in [86] and comparable to that reported in [92].

PG-SG Interaction

Here, a series of shock tube problems is considered to illustrate the limits of the GFM and GFMP for multi-fluid problems with a strong interfacial contact discontinuity, and highlight the superior performance of the FVM-ERS for such problems.

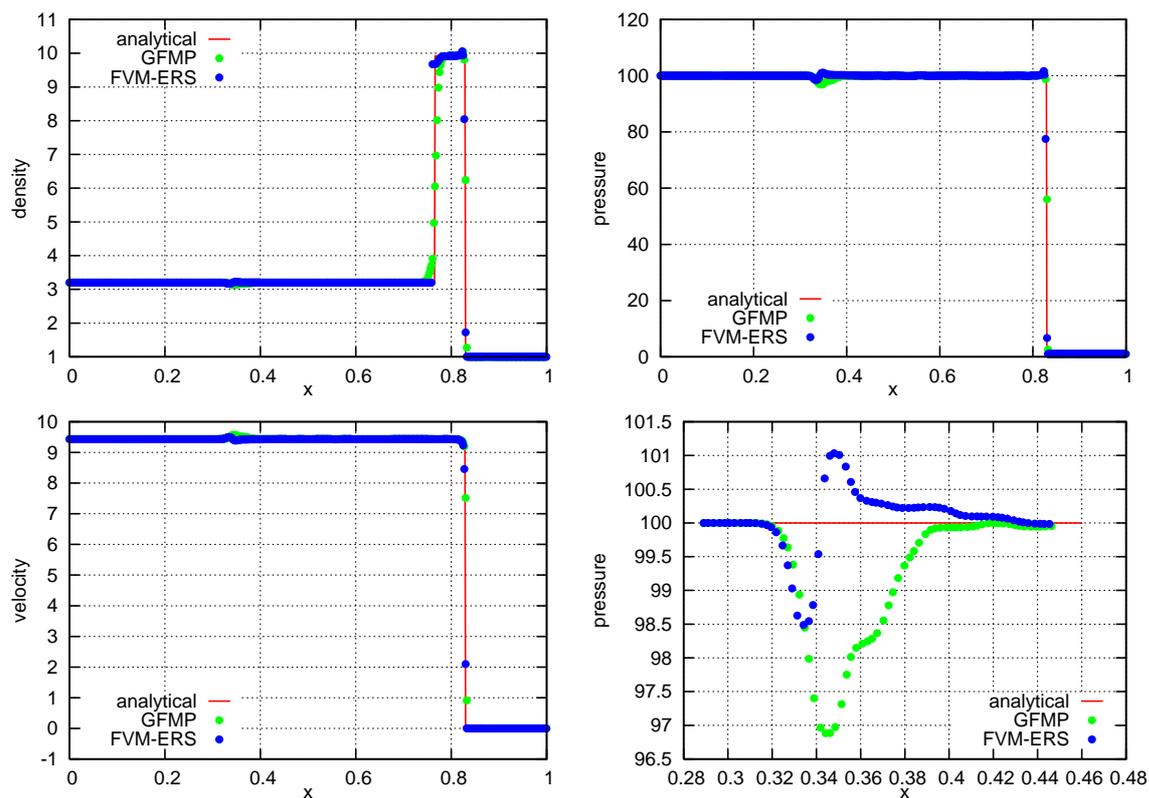


Figure 4.14: Reflection-less gaseous shock problem: variations of the density, pressure, and velocity at $t = 0.06$ along the shock tube (FVM-ERS, $\Delta x = 1/201$) — Zoom on the main oscillation is shown for the pressure field at the bottom right part of the figure.

The first shock tube problem discussed herein was also considered in [65]. In this problem, the membrane is positioned at $x = 0.3$. The fluid at the left side of this membrane is a perfect gas. The fluid at the right side of the membrane is water and is modeled as a stiffened gas. The initial states of both fluids and the constants of their EOSs are

$$(\rho, u, p, \text{EOS}) = \begin{cases} (50.0, 0.0, 10^5, \text{PG}(1.4)) & \text{if } x < 0.3 \\ (1000.0, 0.0, 10^9, \text{SG}(4.4, 6.0 \cdot 10^8)) & \text{if } x > 0.3 \end{cases}$$

and therefore

$$\frac{\rho_L}{\rho_R} = 20.$$

Two meshes are considered in this case. In addition to the mesh with 201 grid points, another mesh with 801 grid points is also considered for these computations. On each mesh, two computations are performed: the first one using the GFMP and the second one using the FVM-ERS. The CFL number is set to 0.8. The results at $t = 2.4 \times 10^{-4}$ are reported in Fig. 4.15 ($\Delta x = 1/201$) and Fig. 4.16 ($\Delta x = 1/801$). On the mesh with 801 grid points in the x direction, both of the GFMP and FVM-ERS perform well. However, the FVM-ERS predicts a sharper density jump close to the material interface. On the coarser mesh with 201 grid points in the x direction, only the FVM-ERS captures the density plateau between the shock and the contact surface. This underscores the superior performance of the FVM-ERS for such problems.

Next, variants of the above problem with an increasingly higher density ratio are considered by decreasing the initial value of the density of the perfect gas. All other parameters of the above shock tube problem are kept unchanged. The GFM, GFMP, and FVM-ERS are applied to the solution of these problems on both generated meshes

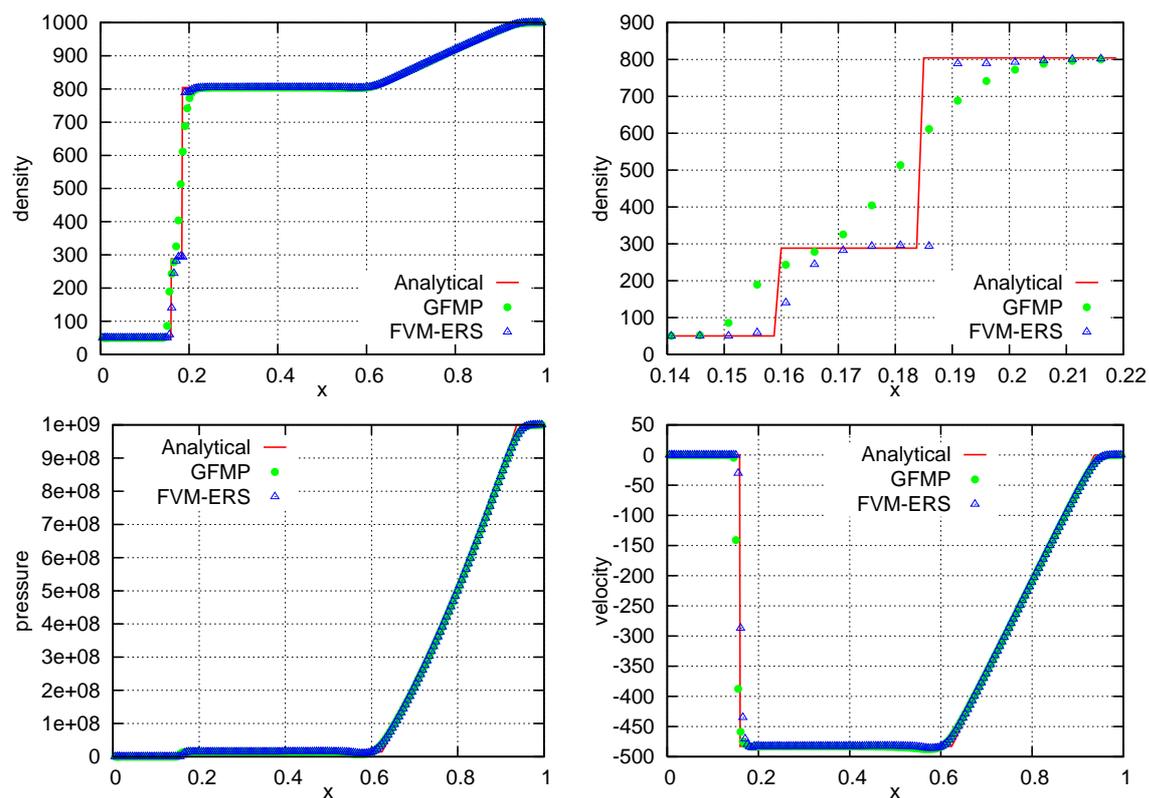


Figure 4.15: Perfect gas - stiffened gas: variations of the density, pressure, and velocity at $t = 2.4 \times 10^{-4}$ along the length of the shock-tube ($\Delta x = 1/201$) — Zoom on the “plateau” region is shown for the density field at the top right part of the figure.

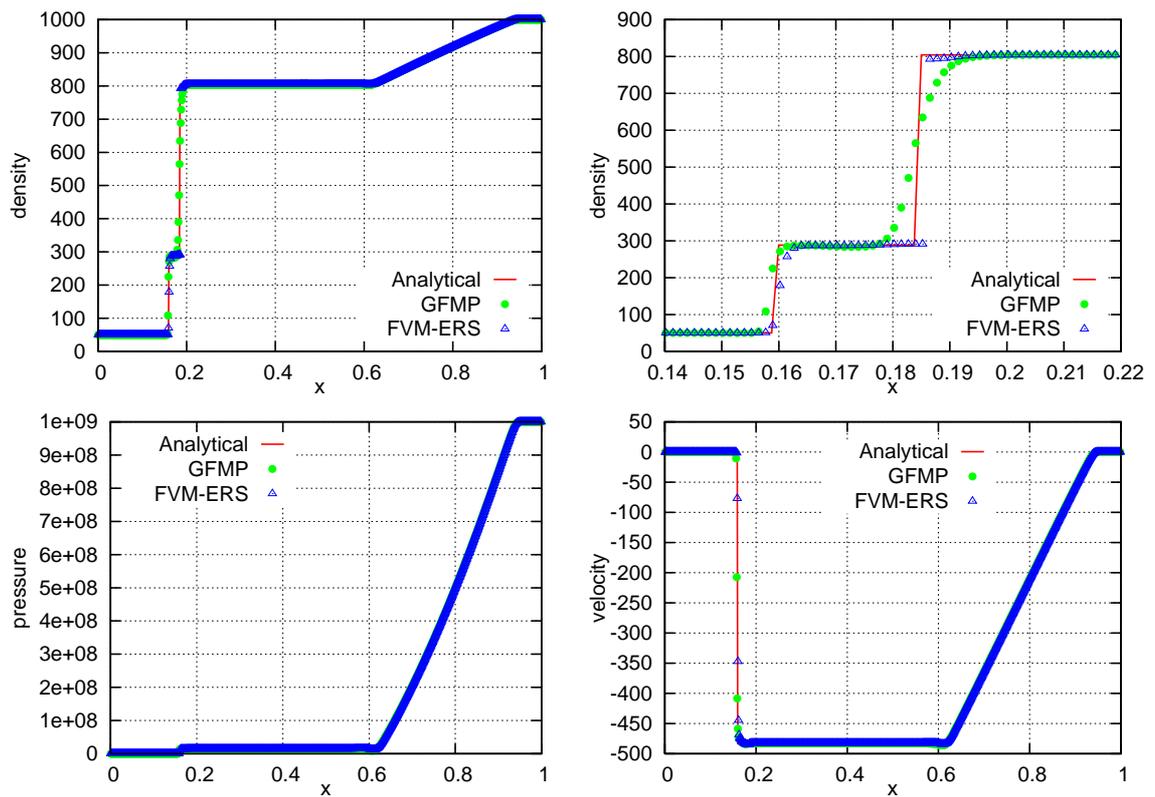


Figure 4.16: Perfect gas - stiffened gas: variations of the density, pressure, and velocity at $t = 2.4 \times 10^{-4}$ along the length of the shock-tube ($\Delta x = 1/801$) — Zoom on the “plateau” region is shown for the density field at the top right part of the figure.

density ratio	GFM Lax-Friedrichs flux CFL = 0.1	GFMP Lax-Friedrichs flux CFL = 0.1	FVM-ERS Roe flux CFL = 0.8
$1000/50 = 20$	succeeds	succeeds	succeeds
$1000/40 = 25$	fails (negative pressure, $\Delta x = 1/201$ and $\Delta x = 1/801$)	succeeds	succeeds
$1000/10 = 100$	fails (negative pressure, $\Delta x = 1/201$ and $\Delta x = 1/801$)	succeeds	succeeds
$1000/5 = 200$	fails (negative pressure, $\Delta x = 1/201$ and $\Delta x = 1/801$)	fails (negative pressure, $\Delta x = 1/201$ and $\Delta x = 1/801$)	succeeds

Table 4.4: Perfect gas - stiffened gas: limits of the GFM and GFMP and advantage of the FVM-ERS for problems with a strong interfacial contact discontinuity.

in the time-interval $[0, 1.2 \times 10^{-4}]$. For this purpose, all three methods are equipped with the RK4 time-integrator. However, the GFM and GFMP are equipped in this case with the Lax-Friedrichs flux scheme characterized by the positivity property, whereas the standard Roe flux is used in the FVM-ERS computations. The outcomes of the performed simulations are characterized in Table 4.4 below where “succeeds” means that the simulation terminates successfully and produces the correct results, and “fails” means that the computations fail during the simulation — typically, early on and because of encountered negative pressure values.

The reader can observe that despite using a flux with the positivity property, the GFM fails to solve all instances of the considered problem with a density ratio higher or equal to 25, even when the CFL number is set as low as 0.1. The GFMP equipped

with the same flux scheme also fails as soon as the density ratio exceeds the value of 200, even when the CFL number is reduced to 0.1. On the other hand, the FVM-ERS equipped with the standard Roe flux successfully solves all considered instances of the problem on both coarse and fine grids. This highlights the limits of the GFM and GFMP for problems with a strong interfacial contact discontinuity, even when equipped with a flux scheme with the positivity property, and the robustness of the FVM-ERS for such problems. It also supports the explanation that for multi-fluid problems with a large discontinuity of the density at the material interface, using a flux scheme with the positivity property does not seem to be as crucial as using an appropriate discretization scheme that does not cross the material interface.

Gas-Water Interaction(II)

Here, a stiffer model problem with two different EOSs for modeling gas and water is considered. The membrane is positioned at $x = 0.3$. The initial conditions for both fluid media are more relevant than previously to underwater implosions where the ratio of densities at the material interface is approximately 1,000 as the initial conditions are given by

$$(\rho, u, p) = \begin{cases} (1.0, 0.0, 10^5) & \text{if } x < 0.3 \\ (1000.0, 0.0, 10^7) & \text{if } x > 0.3 \end{cases}$$

Two computations are performed which differ by the modeling of the water, while the modeling of the air remains the same and is given by a perfect gas EOS. In the first one, the water is modeled by Tait's EOS, while in the second an equivalent stiffened gas EOS is considered. Hence, the first computation is given by the above initial

conditions and

$$\text{EOS} = \begin{cases} \text{PG}(1.4) & \text{if } x < 0.3 \\ \text{Tait}(7.15, 2.07 \cdot 10^9, 1000.0, 10^7) & \text{if } x > 0.3 \end{cases}$$

while the second one is completed by

$$\text{EOS} = \begin{cases} \text{PG}(1.4) & \text{if } x < 0.3 \\ \text{SG}(7.15, \frac{2.07 \cdot 10^9}{7.15}) & \text{if } x > 0.3 \end{cases}$$

Note that the analytical solution of the above problem is the same whether the water is modeled as a barotropic fluid or as a stiffened gas. The structure of this solution consists of a shock wave travelling in the air, a contact discontinuity, and a rarefaction wave propagating in the water. This is consistent with both the physics of the problem and the chosen models. Indeed, as mentioned earlier, a stiffened gas behaves during an isentropic transformation like a barotropic fluid modeled by Tait's EOS.

For this problem, the GFM and the GFMP fail early on in the simulation because of the presence of a strong contact discontinuity. On the other hand, the present method operating at $\text{CFL} = 0.8$ delivers in both cases excellent results, as shown in Fig. 4.17 for $t = 4 \times 10^{-4}$.

JWL-SG Interaction

The following problem considers a one-dimensional underwater explosion, which creates a shock in the water and an expansion in the gas. Water is modeled with the stiffened gas EOS, while the TNT gaseous products are modeled with the JWL EOS.

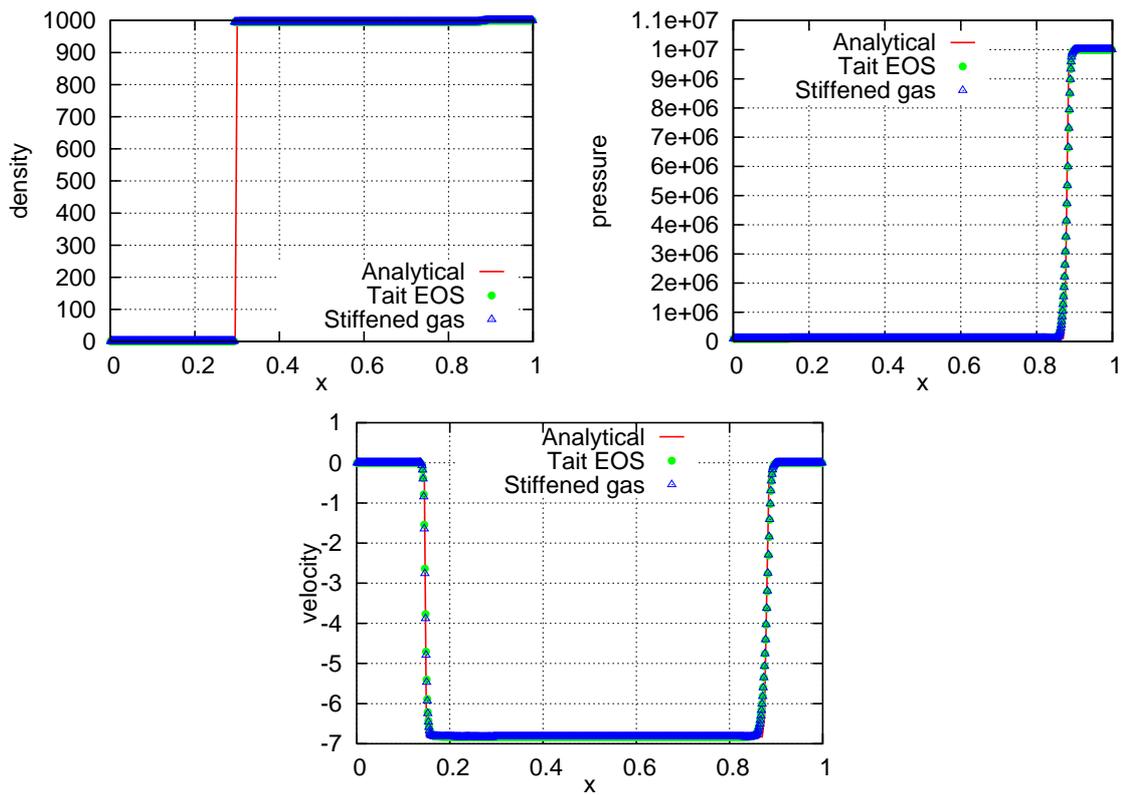


Figure 4.17: Air (perfect gas) - water (barotropic fluid/stiffened gas): variations of the density, pressure, and velocity at $t = 4.0 \times 10^{-4}$ along the length of the shock-tube (FVM-ERS, $\Delta x = 1/201$).

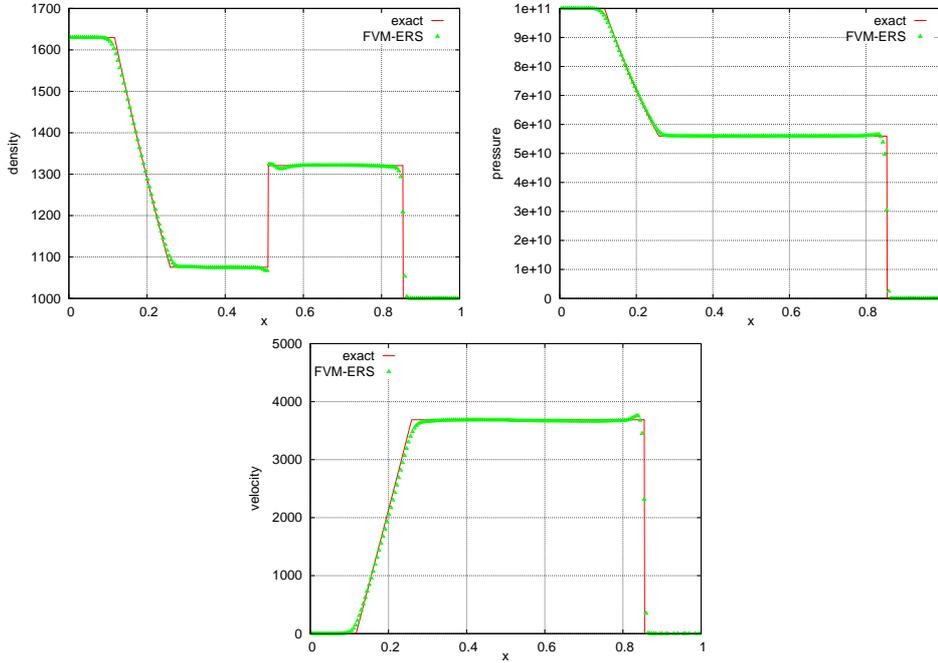


Figure 4.18: Underwater explosion: variations of the density, pressure, and velocity at $t = 3.0 \times 10^{-5}$ along the length of the shock-tube.

The initial conditions and the EOSs parameters are given by

$$(\rho, u, p, \text{EOS}) = \begin{cases} (1630.0, 0.0, 10^{11}, \text{JWL}(\omega, A_1, R_1, A_2, R_2, \rho_0)) & \text{if } x < 0.4 \\ (1000.0, 0.0, 10^5, \text{SG}(7.15, 3.309 \times 10^8)) & \text{if } x > 0.4 \end{cases}$$

where $\omega, A_1, R_1, A_2, R_2, \rho_0$ are given by Table 2.1. The CFL number is set to 0.9. The results at $t = 3 \times 10^{-5}$ are reported in Fig. 4.18 and show that the overall solution given by the present method is again in good agreement with the analytical solution. Both the locations of the shock and the material interface are correct. This example shows the capability of the developed method for a more complicated equation of state than the ones considered earlier.

Chapter 5

Fluid-Structure Coupling

5.1 Introduction

As mentioned in chapter 3, several numerical methods are available for the simulations of fluid flows on deforming domains with their own advantages and inconveniences. As mentioned in the first chapter, the focus of this dissertation is on the first instants of the implosion problems, where structural deformations are limited and cracking of the structure does not occur. For this reason, the integration of the fluid equations of motion is considered on moving meshes and more precisely, the ALE (Arbitrary-Lagrangian-Eulerian) formulation is adopted in this work. In addition to that, dynamic FSI (fluid-structure interaction) problems require the simultaneous integration of the structural equations. This is why the coupled problem, formed of the fluid and structural equations and their transmission conditions, is often seen as a two-field problem. However, the moving mesh is most often viewed as a pseudo-structural system with its own dynamics [48]. Therefore, the coupled system can be formulated as a three- rather than two-field problem [53]: the fluid, the fluid mesh

and the structure.

For simple problems, a monolithic scheme can be used to time-integrate the corresponding semi-discrete equations. In that case, the structure equations of motion are usually assumed to be linear and are efficiently recast in first-order form so that the fluid and structural equations of motion can be combined into a single system of first-order semi-discrete equations ([81] for example). The system is then solved by a single time-integrator. This approach has the advantage of simpler mathematical analysis than partitioned approaches – loosely- or strongly-coupled staggered solution procedures. In particular, the order of time-accuracy of the chosen time integrator is usually recovered in simulations. However, for more complex systems, solving the three-field equations with a monolithic scheme can become computationally challenging, partially explaining their limited use to simple and academic problems. From a practical point of view, these schemes are software-wise unmanageable and from a scientific point of view, they do not recognize the different mathematical properties of each field of equations as shown in chapter 2 and hence the different numerical algorithms tailored for each field (chapter 3).

For these reasons, partitioned procedures and staggered algorithms [79] are often used to solve the system of semi-discrete equations. Unlike in monolithic schemes, the set of equations of each subsystem is time-discretized with a method tailored to its mathematical model. Advantages of such a strategy are simplified explicit/implicit treatments, facilitated subcycling, easy load balancing, smaller computational complexity per time-step – as two smaller systems are integrated instead of a single larger one, – and software modularity – as replacements are easier when better mathematical models and methods emerge in the fluid and structure computational mechanics disciplines. The simplest and most popular form of staggering between the different

subsystems is often referred to as the conventional serial staggered (CSS) procedure. Its generic cycle consists of four steps: first transfer the motion of the interface boundary of the structure to the fluid mesh and update the position of the moving mesh accordingly, second advance the fluid system using a given flow time-integrator, third convert the new fluid pressure into a structural load, fourth advance the structural system under the flow-pressure-induced load using a given structural time-integrator. This time-integration procedure is most often referred to in the literature as a loosely-coupled solver. Its time-accuracy is in general at least one order lower than the ones of the underlying flow and structure time-integrators. Furthermore, its stability limit may be more restrictive than the ones of the flow and/or structure time-integrators. These deficiencies can be overcome with carefully designed sub-iterations for each cycle of the CSS procedure. In that case, the new staggered algorithm is a strongly-coupled method, but the strong coupling nature of the scheme comes at an increased computational cost. Lately, the development of sub-iteration free provable second-order time-accurate loosely-coupled staggered schemes have alleviated some of the shortcomings of the original CSS procedure [15]. These schemes used second-order time-accurate midpoint rule for the structural solver and second-order time-accurate three-point-backward-difference scheme for the flow solver. Even though, the stability of these staggered schemes have never been addressed for nonlinear equations, successful simulations of compressible flows around an aeroelastic structure have been performed.

The response problem of the submarine structures with respect to the flow patterns of the air, water and detonation of highly explosive products and the problems encountered in aeroelasticity present similarities in the general mathematical modeling of both the structure and the fluid. However, a few differences must be recognized.

On one side, a linear behavior of the structure cannot be assumed as both geometric and material nonlinearities must be considered, in particular plasticity. On the other side, the fluid behavior is given by various equations of state and undergoes different flow regimes with both very low- and high-Mach regions. Numerical tools have been chosen and/or developed in the previous chapters to model such complex phenomena. In particular, explicit time-integrators were considered for accuracy requirements as nonlinear phenomena are expected. The failure analysis of underwater imploding structures now requires the simultaneous integration of all subsystems. The use of explicit time-integrators is thus considered in the same framework as the one proposed in [15], where both structural and flow time-integrators were implicit. The present chapter is therefore dedicated to the development and/or application of some appropriate fluid-structure interaction algorithms in order to solve implosion problems.

To this end, the chapter is organized as follows. Section 5.2 presents the mesh motion necessary to model fluid flows in the ALE framework. Sections 5.3 and 5.4 define the three-field model and its semi-discretization respectively. The time-integration of the complete loosely-coupled system of the previously obtained ODEs is then studied in section 5.5 and provable second-order time-accurate time-integrators are obtained. Finally, the theoretical order of time-accuracy is confirmed by numerical tests in the section 5.6.

5.2 Mesh Motion

Fluid-structure interaction phenomena can lead to large deformations of the fluid domain which cannot be ignored. Except for purely Eulerian methods mentioned

earlier, the computational grid must be recomputed to some extent. In order to do that, several options are available, but the use of dynamic meshes is the most common as it allows to preserve the connectivity of the mesh. In this case, the deformed mesh must satisfy the following conditions:

- the vertices at the fluid-structure interface must be located at prescribed positions
- the mesh must remain valid, in the sense that no cross-over can be allowed
- the mesh quality should be such that the overall accuracy of the simulation remains reasonable.

A popular method for generating a dynamic mesh is to assimilate the given grid with a continuous or pseudo-structural system. At the continuous level, the considered mechanical system is often assumed to have a quasi-static behavior and is thus represented by

$$\operatorname{div}(E : \epsilon(\mathbf{x}_m - \mathbf{x}_m(0))) = 0 \text{ in } \Omega_f(t)$$

where $\mathbf{x}_m - \mathbf{x}_m(0)$ denotes the displacement field of the fictitious structure, ϵ its strain tensor and E its tensor of elasticities. It is also subject to the boundary condition

$$\mathbf{x}_m - \mathbf{x}_m(0) = \bar{u} \text{ on } \Gamma(t)$$

where \bar{u} is a prescribed displacement. As a result, the positions of the interior nodes are determined from the ones of the boundary points lying on $\Gamma(t)$.

Another approach consists in assimilating the fluid mesh with a discrete structure directly, rather than a continuous elastic medium. Several methods have been proposed including the lineal spring analogy [48], the torsional spring analogy [9] and the ball vertex analogy [19].

In the lineal spring analogy, each edge of the mesh is assimilated with a lineal spring and the mesh itself is a network of lineal springs. The stiffness of each lineal spring is inversely proportional to the length of the edge it is supported by. Heuristically, as two connected vertices get closer to one another, the stiffening of the corresponding fictitious spring increases, thus preventing the two vertices from colliding. However, it does not prevent a vertex from crossing an edge or a face of an element and thus, this mesh motion algorithm was shown to fail for large deformations.

A solution to the specific problem encountered in the lineal spring analogy is presented in [10, 9] for two-dimensional and three-dimensional problems respectively. In addition to the lineal springs considered above, a torsional spring is also located at each vertex of an element. The torsional spring is such that its stiffness increases as the vertex gets closer to another vertex or an edge or a face of that same element. This method is parameter-free, insensitive to the size of the mesh elements and has been shown to be more robust than the lineal spring analogy when large deformations of the computational domain occur.

More recently, [19] proposed another method to circumvent the shortcomings of the lineal spring method. Similarly to the torsional spring analogy, this method avoids the appearance of collapse mechanisms for the mesh elements. To this end, new linear springs are inserted between a vertex and its perpendicular projections on the opposing faces, hence bounding this vertex to its polyhedral ball. In some cases, this method proved more robust than the one based on the torsional spring analogy.

In all cases, the fluid-mesh motion requires solving an equation of the form

$$\begin{aligned} \rho_m \frac{\partial^2 \mathbf{x}_m}{\partial t^2} - \nabla_\xi \cdot \sigma_m(\mathbf{x}_m - \mathbf{x}_m(0)) &= 0 \text{ in } \Omega_f(t) \\ \mathbf{x}_m - \mathbf{x}_m(0) &= \mathbf{u}_s \text{ on } \Gamma(t) \\ \frac{\partial \mathbf{x}_m}{\partial t} &= \frac{\partial \mathbf{u}_s}{\partial t} \text{ on } \Gamma(t) \end{aligned}$$

5.3 Three-Field Problem

A three-field formulation is used to describe the fluid-structure interaction problem considered in the present work. It was introduced by [53] in order to describe a large class of FSI and nonlinear aeroelastic problems in the time-domain. In this formulation, an Arbitrary-Lagrangian-Eulerian form of the flow equations, as described in chapter 2, is used to account for a fluid-mesh motion induced by potentially large displacements of the structure. The three-field equations governing the problem are given by

$$\left. \frac{\partial J\mathbf{U}}{\partial t} \right|_\xi + J \nabla_x \cdot \mathbf{F}(\mathbf{U}, \dot{\mathbf{x}}) = 0 \quad \text{in } \Omega_f(t) \quad (5.1)$$

$$\rho_s \frac{\partial^2 \mathbf{u}_s}{\partial t^2} - \nabla_x \cdot \sigma_s(\mathbf{u}_s, \dot{\mathbf{u}}_s) = \mathbf{f}_{\text{ext}}(t) \quad \text{in } \Omega_s(t) \quad (5.2)$$

$$\left. \rho_m \frac{\partial^2 \mathbf{x}_m}{\partial t^2} \right|_\xi - \nabla_\xi \cdot \sigma_m(\mathbf{x}_m - \mathbf{x}_m(0)) = 0 \quad \text{in } \Omega_f(t) \quad (5.3)$$

where the same notations as in chapter 2 have been used for the first and second equations. The third equation models the dynamics of the fluid-mesh motion by assimilating it with a fictitious or pseudo-structure subsystem as mentioned in the previous section. In this last equation, the subscript m designates quantities associated with the fluid-mesh motion and ρ_m and σ_m are fictitious density and stress

tensor of the fluid-mesh respectively, where as \mathbf{x}_m and $\mathbf{x}_m(0)$ are the instantaneous position and the given initial position of the dynamic fluid mesh respectively.

In addition to satisfying their intrinsic boundary conditions, the first two equations are coupled at the fluid-structure interface $\Gamma(t) = \partial\Omega_f(t) \cap \partial\Omega_s(t)$ by the following transmission conditions

$$\mathbf{u} \cdot \mathbf{n}_\Gamma = \dot{\mathbf{u}}_s \cdot \mathbf{n}_\Gamma \quad (5.4)$$

$$-p \mathbf{n}_\Gamma = \sigma_s(\mathbf{u}_s, \dot{\mathbf{u}}_s) \cdot \mathbf{n}_\Gamma \quad (5.5)$$

Moreover, compatibility conditions must also be satisfied on $\Gamma(t)$

$$\mathbf{x}_m - \mathbf{x}_m(0) = \mathbf{u}_s \quad (5.6)$$

$$\frac{\partial \mathbf{x}_m}{\partial t} = \frac{\partial \mathbf{u}_s}{\partial t} \quad (5.7)$$

in order to avoid the appearance of gaps between the fluid mesh and the structure.

5.4 Semi-discretization of the Three-field Equations

The semi-discretization of the fluid and structure equations have been studied separately in chapter 3. However, the numerical analysis of the fluid-structure interaction in these frameworks requires not only the semi-discretization of the fluid-mesh equations, but also the discretization of the transmission and compatibility equations (5.4)-(5.7). This analysis is the object of the present section.

5.4.1 Semi-Discretization of the Mesh Equation

The vector of fluid grid point displacements $\mathbf{x}_m(t) - \mathbf{x}_m(0)$ is partitioned between the fluid grid points lying on the fluid-structure interface $\Gamma(t)$ whose values are somehow imposed by the deformations of the structure, and the remaining fluid grid points

$$\mathbf{x}_m(t) - \mathbf{x}_m(0) = \begin{pmatrix} \mathbf{x}_\Omega(t) - \mathbf{x}_\Omega(0) \\ \mathbf{x}_\Gamma(t) - \mathbf{x}_\Gamma(0) \end{pmatrix} \quad (5.8)$$

where, with a slight abuse of notation, the subscript $\Omega(t)$ designates the set of nodes that do not lie on the interface $\Gamma(t)$. For simplicity, the fluid-mesh motion can be computed in a quasi-static fashion, in which the finite element semi-discretization of Eq. (5.3) can be written in velocity form as

$$\mathbf{K}_{\Omega\Omega}(t)\dot{\mathbf{x}}_\Omega(t) + \mathbf{K}_{\Omega\Gamma}(t)\dot{\mathbf{x}}_\Gamma(t) = 0 \quad (5.9)$$

where $\mathbf{K}(t)$ is a fictitious finite element stiffness matrix constructed by one of the methods presented in section 5.2. The time dependence on $\mathbf{K}(t)$ is to reflect that it is usually updated at each time-instance of a coupled FSI computation. The velocity form characterizing Eq. (5.9) lends itself to an incremental approach for updating the dynamic fluid-mesh once the interface nodal displacements are known

$$\mathbf{x}_\Omega(t) = \mathbf{x}_\Omega(\tau) - \int_\tau^t \mathbf{K}_{\Omega\Omega}^{-1}(\eta)\mathbf{K}_{\Omega\Gamma}(\eta)\dot{\mathbf{x}}_\Gamma(\eta)d\eta, \quad \forall \tau \in \mathbb{R} \quad (5.10)$$

$$\dot{\mathbf{x}}_\Omega(t) = -\mathbf{K}_{\Omega\Omega}^{-1}(t)\mathbf{K}_{\Omega\Gamma}(t)\dot{\mathbf{x}}_\Gamma(t) \quad (5.11)$$

where $\mathbf{K}_{\Omega\Omega}^{-1}(t)$ is the inverse of $\mathbf{K}_{\Omega\Omega}(t)$. This incremental strategy is consistent with the idea of updating $\mathbf{K}(t)$ at each time-instance and leads to a more robust fluid-mesh-motion algorithm than its absolute counterpart.

5.4.2 Compatibility Equations

The interface displacements and velocities of the fictitious structure that represent the mesh are given by the interface displacements and velocities of the real structure through compatibility equations (5.6)-(5.7). The development of different simulation softwares for the fluid and the structure comes from the different mathematical properties of both systems and from the different precision requirements for each system. Therefore, the fluid and structure meshes have independent representations of the fluid-structure interface, which results in general in computations where the fluid and structure meshes are non-matching, since the mesh element types and/or mesh sizes can be different. This means that nor the fluid and structure nodes, neither the normals to the fluid interface boundary and to the structure interface boundary coincide. The mortar method [8] applied to the discretization of (5.6)-(5.7) and presented by [11] is one way of computing the fluid displacements and velocities from the ones of the structure. However, for practical mesh resolutions and computational costs, the mortar method is discarded for consistent interpolation based schemes, which are easy to implement and perform well in aeroelastic simulations even though they may not have the mathematical optimality properties as the mortar method [11]. In the present work, the compatibility conditions (5.6)-(5.7) are discretized according to the consistent interpolation based scheme proposed in [57]. The fluid nodes at the fluid mesh interface boundary are projected on the structure mesh interface boundary along directions given by the normals to the structure mesh interface boundary.

This is done in a preprocessing step. During the simulation, the nodal displacements and velocities are first computed at the projected locations on the structure mesh interface boundary using the semi-discretization of the structure equations and then transferred onto the fluid mesh nodes. Since the “matching” of fluid mesh nodes to the structure mesh interface boundary is done in a preprocessing step, it is independent of time. This leads to

$$\mathbf{x}_\Gamma(t) = \mathbf{x}_\Gamma(\tau) + \mathbf{T}_\Gamma(\mathbf{u}_\Gamma(t) - \mathbf{u}_\Gamma(\tau)), \quad \forall \tau \in \mathbb{R} \quad (5.12)$$

$$\dot{\mathbf{x}}_\Gamma(t) = \mathbf{T}_\Gamma \dot{\mathbf{u}}_\Gamma(t) \quad (5.13)$$

where \mathbf{T}_Γ is the identity matrix \mathbf{I} if the fluid and structure sides of the fluid-structure interface have matching discrete interfaces $\Gamma_f(t)$ and $\Gamma_s(t)$, respectively, and otherwise \mathbf{T}_Γ is a suitable time-independent transfer matrix.

In order to generalize Eqs. (5.12)-(5.13) to all the vertices of the fluid mesh, the following operator is introduced

$$\mathbf{T}(t) = \begin{pmatrix} -\mathbf{K}_{\Omega\Omega}^{-1}(t)\mathbf{K}_{\Omega\Gamma}(t) \\ \mathbf{I} \end{pmatrix} \mathbf{T}_\Gamma \quad (5.14)$$

allowing to write

$$\begin{pmatrix} \mathbf{x}_\Omega(t) \\ \mathbf{x}_\Gamma(t) \end{pmatrix} = \begin{pmatrix} \mathbf{x}_\Omega(\tau) \\ \mathbf{x}_\Gamma(\tau) \end{pmatrix} + \int_\tau^t \mathbf{T}(\eta) \dot{\mathbf{u}}_\Gamma(\eta) d\eta, \quad \forall \tau \in \mathbb{R}. \quad (5.15)$$

Eqs. (5.12)-(5.13) are then generalized as follows

$$\mathbf{x}(t) = \mathbf{x}(\tau) + \int_{\tau}^t \mathbf{T}(\eta) \dot{\mathbf{u}}_{\Gamma}(\eta) d\eta, \quad \forall \tau \in \mathbb{R}, \quad (5.16)$$

$$\dot{\mathbf{x}}(t) = \mathbf{T}(t) \dot{\mathbf{u}}_{\Gamma}(t) \quad (5.17)$$

5.4.3 Transmission Conditions and Load Transfer

In the ALE-formulated fluid-structure interaction problems, two classes of methods are most often used to discretize the transmission conditions. The non-conservative consistent interpolation approach matches nodes of one mesh to associate points of the other mesh, by projection for instance. The values to be transferred are evaluated at the associate points using the semi-discretization of the second mesh and are then directly transferred to the nodes of the first mesh. This can be done for both transmission conditions: transfer of the structural motion to the fluid and transfer of the pressure induced load to the structure. The conservative approach requires that the energy leaving one subsystem is gained by the other one. This can be expressed by the equality of the virtual works of both fluid and structure in the finite element framework as studied in [11]. The mathematical condition to enforce conservation leads to a certain formulation of the load transfer once a specific formulation of the transfer of the structural motion to the fluid is chosen. The conservative method has been successfully used in various aeroelastic computations. However, the physical problem considered in the present work differs from most aeroelastic computations as violent phenomena occur and drive the responses of both subsystems in the first case while energy transfers over several cycles dominate the second type of problems. In flutter and limit cycle oscillations problems for aircrafts, the numerical addition or removal of energy at the interface can significantly change the simulated response,

as a flutter mode can be simulated when none should truly appear. Nevertheless, a conservative approach is considered for the simulation of collapsing structures of the present work.

As mentioned in the previous section, the discretization of the compatibility equations used a consistent interpolation based scheme. It is natural to use the same discretization for the first transmission equation (5.4). Consequently, as explained in [11], in order to have conservation at the fluid-structure interface, the nodal values of the pressure induced force on the structure must be computed by

$$\mathbf{f}_{\text{ae}}^s(p(\mathbf{U}(t)), \mathbf{x}_\Gamma(t)) = U_\Gamma^T \mathbf{f}_{\text{ae}}^f(p(\mathbf{U}(t)), \mathbf{x}_\Gamma(t))$$

where $\mathbf{f}_{\text{ae}}^f(p(\mathbf{U}(t)), \mathbf{x}_\Gamma(t))$ denotes the vector of the pressure-induced nodal forces from the fluid mesh.

5.5 Time Discretization

5.5.1 The GSS Procedure

The resolution of nonlinear fluid-structure interaction problems on moving grids requires the simultaneous time integration of the three-field equations. Among the loosely-coupled partitioned procedures for the resolution of such problems, the Generalized Serial Staggered (GSS) procedure provides a framework to formally design second-order time-accurate schemes as was done in [15]. The GSS procedure is inherited from the more widely used Conventional Serial Staggered (CSS) procedure. However, while the CSS procedure is in general only first-order time-accurate even though structural and flow time-integrators may have higher-order time-accuracy,

the GSS procedure allows to make adequate use of structural displacement predictors and structural force correctors to increase its time-accuracy to second order. Here, it is assumed that both fluid and structure time-integration schemes have the same time-step Δt . The cycles of the GSS procedure can be summarized as follows:

1. predict the motion at t^{n+1} of the wet boundary of the structure, \mathbf{u}_Γ^{n+1P} , transfer it to the fluid subsystem, and update the corresponding predicted position \mathbf{x}^{n+1P} of the fluid dynamic mesh as follows:

$$\mathbf{x}^{n+1P} = \mathbf{x}^{nP} + \overline{\mathbf{T}}(\mathbf{u}_\Gamma^{n+1P} - \mathbf{u}_\Gamma^{n-1P}),$$

where $\overline{\mathbf{T}}$ is a mean value of $\mathbf{T}(t)$ for $t \in [t^n, t^{n+1}]$ and is to be determined later.

2. advance the fluid subsystem to t^{n+1} and compute a new pressure field p^{n+1} .
3. compute a corrected aerodynamic force $\mathbf{f}_{\text{ae}}^{n+1C}$ and transfer it to the structure.
4. advance the structure subsystem to t^{n+1} by solving the structure equations for \mathbf{u}^{n+1} .

A schematic of the GSS procedure is given in Fig. 5.1.

5.5.2 Fluid Time Discretization

The second-order Runge-Kutta 2 and 3-Point-Backward-Difference time-integrators on moving grids have been overviewed in section 3.4 where the mesh motion was a given data of the problem. In the following, the fact that the mesh motion is not known a priori is taken into account in the numerical analysis of the time integration.

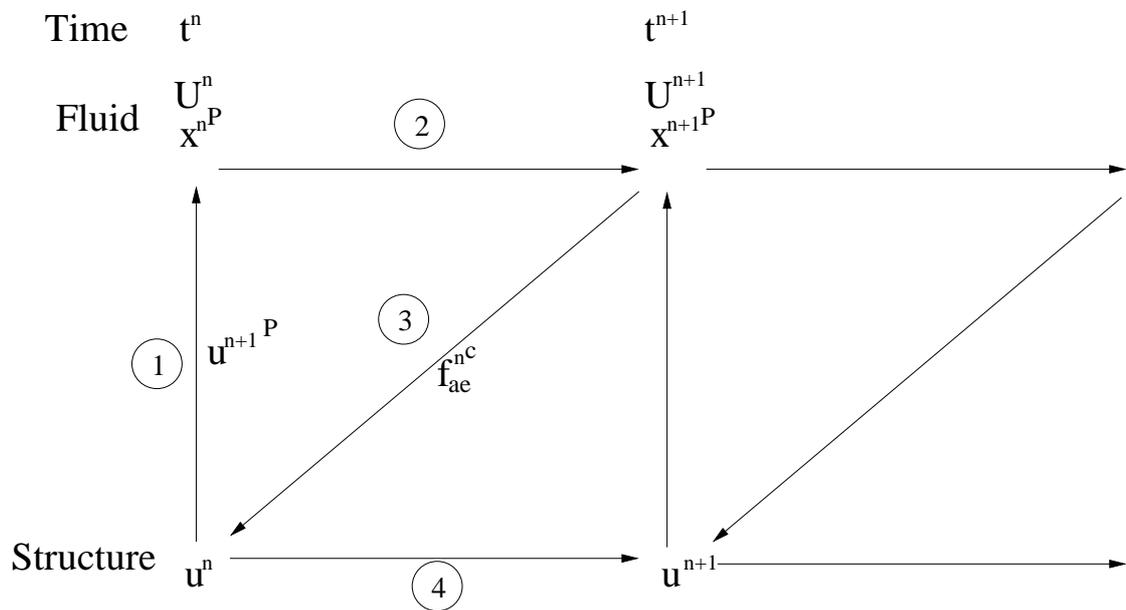


Figure 5.1: The GSS procedure

Runge-Kutta 2

The local truncation error of the Runge-Kutta 2 time integration scheme is

$$\begin{aligned} \Psi_{\mathbf{U}}(t^{n+1}) &= \Omega(\mathbf{x}^P(t^{n+1}))\mathbf{U}(t^{n+1}) - \Omega(\mathbf{x}^P(t^n))\mathbf{U}(t^n) \\ &\quad + \frac{\Delta t}{2} (\Phi(\mathbf{U}(t^n), \mathbf{x}^P(t^{n+1}), \mathbf{x}^P(t^n)) + \Phi(\mathbf{U}^*, \mathbf{x}^P(t^{n+1}), \mathbf{x}^P(t^n))) \end{aligned}$$

where \mathbf{U}^* is defined by

$$\Omega(\mathbf{x}^P(t^{n+1}))\mathbf{U}^* = \Omega(\mathbf{x}^P(t^n))\mathbf{U}(t^n) - \Delta t \Phi(\mathbf{U}(t^n), \mathbf{x}^P(t^{n+1}), \mathbf{x}^P(t^n)).$$

The predicted positions of the mesh are given by

$$\mathbf{x}^P(0, \Delta t) = \mathbf{x}(0) \quad (5.18)$$

$$\mathbf{x}^P(t, \Delta t) = \mathbf{x}^P(t - \Delta t) + \overline{\mathbf{T}}(t - \Delta t, t - 2\Delta t, \dots)(\mathbf{u}_\Gamma^P(t) - \mathbf{u}_\Gamma^P(t - \Delta t)) \quad (5.19)$$

$$\begin{aligned} \mathbf{u}_\Gamma^P(t, \Delta t) &= \mathbf{u}_\Gamma^P(t - \Delta t) + \alpha_0 \Delta t \dot{\mathbf{u}}_\Gamma^P(t - \Delta t) \\ &\quad + \alpha_1 \Delta t (\dot{\mathbf{u}}_\Gamma^P(t - \Delta t) - \dot{\mathbf{u}}_\Gamma^P(t - 2\Delta t)). \end{aligned} \quad (5.20)$$

where the dependence of \mathbf{x}^P and \mathbf{u}_Γ^P on Δt have only been expressed explicitly on the left-hand side terms and where the dependence of $\overline{\mathbf{T}}$ with respect to the time integration path has been explicitly stated and thus $\overline{\mathbf{T}}$ depends on all previous iterations. A lemma given by [75] analyzes the order of time-accuracy of the fluid computations performed by the GSS procedure, that is when the motion of the mesh is not known a priori.

Lemma 1 *If all functions characterizing the semi-discrete fluid, structure, and fluid-mesh-motion equations are sufficiently smooth, and the instantaneous position and predicted position of the dynamic fluid-mesh satisfy*

$$\exists q \in \mathbb{R} / \forall t \in [0, T], \mathbf{x}^P(t) - \mathbf{x}(t) = \mathcal{O}(\Delta t^q) \quad (5.21)$$

the local truncation error $\Psi_{\mathbf{U}}$ of the generalized serial staggered (GSS) procedure satisfies

$$\Psi_{\mathbf{U}}(t^{n+1}) = \Delta t \mathcal{O}(\|\mathbf{x}^P(t^{n+1}) - \mathbf{x}(t^{n+1})\|) + \mathcal{O}(\Delta t^3) \quad (5.22)$$

This means that the scheme is globally second-order time-accurate if \mathbf{x}^P is a first-order prediction of $\mathbf{x}(\cdot)$.

3-Point-Backward-Difference

The local truncation error of the 3-Point-Backward-Difference time integration scheme is

$$\begin{aligned}\Psi_{\mathbf{U}}(t^{n+1}) &= \frac{3}{2}\Omega(\mathbf{x}^P(t^{n+1}))\mathbf{U}(t^{n+1}) - 2\Omega(\mathbf{x}^P(t^n))\mathbf{U}(t^n) + \frac{1}{2}\Omega(\mathbf{x}^P(t^{n-1}))\mathbf{U}(t^{n-1}) \\ &+ \Delta t\Phi(\mathbf{U}(t^{n+1}), \mathbf{x}^P(t^{n+1}), \mathbf{x}^P(t^n))\end{aligned}$$

where \mathbf{x}^P is given by Eq. (5.19). A lemma given by [15] analyzes the order of time-accuracy of the fluid computations performed by the GSS procedure, that is when the motion of the mesh is not known a priori.

Lemma 2 *If all functions characterizing the semi-discrete fluid, structure, and fluid-mesh-motion equations are sufficiently smooth, and the instantaneous position and predicted position of the dynamic fluid-mesh satisfy*

$$\exists q \in \mathbb{R} / \forall t \in [0, T], \mathbf{x}^P(t) - \mathbf{x}(t) = \mathcal{O}(\Delta t^q) \quad (5.23)$$

the local truncation error $\Psi_{\mathbf{U}}$ of the generalized serial staggered (GSS) procedure satisfies

$$\Psi_{\mathbf{U}}(t^{n+1}) = \Delta t \sum_{k=-1}^1 \mathcal{O}(\|\mathbf{x}^P(t^{n+k}) - \mathbf{x}(t^{n+k})\|) + \mathcal{O}(\Delta t^3) \quad (5.24)$$

This means that the scheme is globally second-order time-accurate if \mathbf{x}^P is a first-order prediction of $\mathbf{x}(\cdot)$.

5.5.3 Structure Time Discretization

The second-order explicit central difference and implicit midpoint rule time-integrators of the structural equations have been overviewed in section 3.4 where the external forces were known exactly. In the following, the fact that the external forces from the fluid to the structure are not known a priori is taken into account in the numerical analysis of the time integration. In order to do that, at the continuous level, the external forces are split in two terms

$$\mathbf{f}_{\text{ext}}(t) + \mathbf{f}_{\text{ae}}(p(\mathbf{U}(t)), \mathbf{x}_{\Gamma}(t)) \quad (5.25)$$

where the second term denotes the vector of aerodynamic forces applied from the fluid to the structure and the first term denotes the vector of other external forces. The aerodynamic forces depend on the pressure field $p(\mathbf{U}(t))$ at the interface boundary $\Gamma(t)$. Similarly, at the discrete level, two terms $\mathbf{f}_{\text{ext}}^n$ and $\mathbf{f}_{\text{ae}}^{nC}$ represent the approximations of their continuous counterpart. Note that the superscript C is used to emphasize the fact that this term is actually a corrected aerodynamic force vector.

Explicit Central Difference

The discretization of the structure subsystem by the central difference explicit time integrator is

$$M \frac{\dot{\mathbf{u}}^{n+\frac{1}{2}} - \dot{\mathbf{u}}^{n-\frac{1}{2}}}{\Delta t} + C \frac{\dot{\mathbf{u}}^{n-\frac{1}{2}} + \dot{\mathbf{u}}^{n+\frac{1}{2}}}{2} + \mathbf{f}_{\text{int}}(\mathbf{u}^n) = \mathbf{f}_{\text{ext}}^n + \mathbf{f}_{\text{ae}}^{nC}$$

and the algorithm to update the state of the solid is:

$$\begin{aligned}\dot{\mathbf{u}}^{n+\frac{1}{2}} &= \left(M + \frac{\Delta t C}{2}\right)^{-1} \left(\Delta t (\mathbf{f}_{\text{ext}}^n - \mathbf{f}_{\text{int}}(\mathbf{u}^n)) + \left(M - \frac{\Delta t C}{2}\right) \dot{\mathbf{u}}^{n-\frac{1}{2}}\right) \\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \Delta t \dot{\mathbf{u}}^{n+\frac{1}{2}}\end{aligned}$$

In the remainder of this analysis, the new matrix \tilde{M} is introduced as

$$\tilde{M} = M + \frac{\Delta t C}{2}$$

velocity The truncation error $\Psi_{\dot{\mathbf{u}}}$ of the velocity of the structure subsystem in the GSS procedure (see section 5.5.1) is

$$\begin{aligned}\Psi_{\dot{\mathbf{u}}}(t^{n+\frac{1}{2}}) &= \dot{\mathbf{u}}(t^{n+\frac{1}{2}}) - \dot{\mathbf{u}}^{n+\frac{1}{2}} \\ \tilde{M}\Psi_{\dot{\mathbf{u}}}(t^{n+\frac{1}{2}}) &= \left(M + \frac{\Delta t}{2}C\right) \dot{\mathbf{u}}(t^{n+\frac{1}{2}}) \\ &\quad - \Delta t \left(\mathbf{f}_{\text{ext}}(t^n) + \mathbf{f}_{\text{ae}}^{n_C} - \mathbf{f}_{\text{int}}(\mathbf{u}(t^n))\right) - \left(M - \frac{\Delta t}{2}C\right) \dot{\mathbf{u}}(t^{n-\frac{1}{2}})\end{aligned}$$

Since the exact equation is satisfied at t^n ,

$$\begin{aligned}\tilde{M}\Psi_{\dot{\mathbf{u}}}(t^{n+\frac{1}{2}}) &= \left(M + \frac{\Delta t}{2}C\right) \dot{\mathbf{u}}(t^{n+\frac{1}{2}}) - \left(M - \frac{\Delta t}{2}C\right) \dot{\mathbf{u}}(t^{n-\frac{1}{2}}) \\ &\quad - \Delta t (M\ddot{\mathbf{u}}(t^n) + C\dot{\mathbf{u}}(t^n)) - \Delta t \left(\mathbf{f}_{\text{ae}}^{n_C} - \mathbf{f}_{\text{ae}}(t^n)\right)\end{aligned}$$

A third-order Taylor expansion around t^n for both velocity terms lead to

$$\begin{aligned}\tilde{M}\Psi_{\dot{\mathbf{u}}}(t^{n+\frac{1}{2}}) &= \Delta t M \ddot{\mathbf{u}}(t^n) + \Delta t C \dot{\mathbf{u}}(t^n) + \mathcal{O}(\Delta t^3) \\ &\quad - \Delta t (M\ddot{\mathbf{u}}(t^n) + C\dot{\mathbf{u}}(t^n)) - \Delta t \left(\mathbf{f}_{\text{ae}}^{n_C} - \mathbf{f}_{\text{ae}}(t^n)\right)\end{aligned}$$

Hence,

$$\Psi_{\dot{\mathbf{u}}}(t^{n+\frac{1}{2}}) = \Delta t \mathcal{O} \left(\|\mathbf{f}_{\text{ae}}(t^n) - \mathbf{f}_{\text{ae}}^{n_C}\| \right) + \mathcal{O}(\Delta t^3)$$

displacement The truncation error Ψ_u of the displacement of the structure subsystem in the GSS procedure (see section 5.5.1) is

$$\begin{aligned} \Psi_{\mathbf{u}}(t^{n+1}) &= \mathbf{u}(t^{n+1}) - \mathbf{u}^{n+1} \\ &= \mathbf{u}(t^{n+1}) - \mathbf{u}(t^n) - \Delta t \dot{\mathbf{u}}^{n+\frac{1}{2}} \\ &= \mathbf{u}(t^{n+1}) - \mathbf{u}(t^n) - \Delta t \dot{\mathbf{u}}(t^{n+\frac{1}{2}}) + \Delta t^2 \mathcal{O} \left(\|\mathbf{f}_{\text{ae}}(t^n) - \mathbf{f}_{\text{ae}}^{n_C}\| \right) + \mathcal{O}(\Delta t^3) \end{aligned}$$

A third-order Taylor expansion around $t^{n+\frac{1}{2}}$ for both displacement terms lead to

$$\begin{aligned} \Psi_{\mathbf{u}}(t^{n+1}) &= \Delta t \dot{\mathbf{u}}(t^{n+\frac{1}{2}}) + \mathcal{O}(\Delta t^3) - \Delta t \dot{\mathbf{u}}(t^{n+\frac{1}{2}}) + \Delta t^2 \mathcal{O} \left(\|\mathbf{f}_{\text{ae}}(t^n) - \mathbf{f}_{\text{ae}}^{n_C}\| \right) + \mathcal{O}(\Delta t^3) \\ &= \Delta t^2 \mathcal{O} \left(\|\mathbf{f}_{\text{ae}}(t^n) - \mathbf{f}_{\text{ae}}^{n_C}\| \right) + \mathcal{O}(\Delta t^3) \end{aligned} \quad (5.26)$$

Hence, the scheme computes a globally second-order time-accurate solution if $\mathbf{f}_{\text{ae}}^{n_C}$ is a first-order approximation of $\mathbf{f}_{\text{ae}}(t^n)$.

Implicit Midpoint Rule

In the case of a midpoint rule time integrator, it was shown that

$$\Psi_{\dot{\mathbf{u}}}(t^{n+1}) = \Delta t \left(\mathcal{O} \left(\|\mathbf{f}_{\text{ae}}(t^{n+1}) - \mathbf{f}_{\text{ae}}^{n+1_C}\| \right) + \mathcal{O} \left(\|\mathbf{f}_{\text{ae}}(t^n) - \mathbf{f}_{\text{ae}}^{n_C}\| \right) \right) + \mathcal{O}(\Delta t^3) \quad (5.27)$$

$$\Psi_{\mathbf{u}}(t^{n+1}) = \Delta t \left(\mathcal{O} \left(\|\mathbf{f}_{\text{ae}}(t^{n+1}) - \mathbf{f}_{\text{ae}}^{n+1_C}\| \right) + \mathcal{O} \left(\|\mathbf{f}_{\text{ae}}(t^n) - \mathbf{f}_{\text{ae}}^{n_C}\| \right) \right) + \mathcal{O}(\Delta t^3) \quad (5.28)$$

Thus this midpoint rule scheme computes globally second-order time-accurate displacement and velocity fields if \mathbf{f}_{ae}^{nC} is a first-order approximation of $\mathbf{f}_{ae}(t^n)$.

5.5.4 Second-Order Time-Accurate Loosely-Coupled Solution Algorithms

The study of second-order time-accurate loosely-coupled solutions algorithms where the structure time-integrator was the midpoint rule and the fluid time-integrator was the three-point backward-difference scheme was already done in [15]. Their use of implicit time-integrators was justified by the study of stability problems in aeroelasticity. In the present section, the focus is put on the use of the central difference explicit structural time-integrator and the explicit Runge-Kutta flow time-integrator as the response of the structure to violent phenomena in the fluids is studied. These two time-integrators are crucial to the eventual simulations of failure analysis of underwater air-filled structures since the multiphase flow solver has been developed with the Runge-Kutta scheme and the crack propagation algorithm has been implemented in XFEM using an explicit central difference scheme. It is reminded that explicit schemes were chosen because of the nonlinear behaviour of both the fluids and the structure and because of the accuracy requirements to model implosion problems. To this end, a fundamental result from [15] is recalled, as it allows to build the second-order time-accurate loosely-coupled solution algorithms in that article.

Lemma 3 *If the aerodynamic force corrector is at least a first-order approximation of $\mathbf{f}_{ae}(t^n)$, that is*

$$\mathbf{f}_{ae}^{nC} = \mathbf{f}_{ae}(t^n) + \mathcal{O}(\Delta t^2) \quad (5.29)$$

and if the structure predictor \mathbf{u}^P and the matrix $\bar{\mathbf{T}}$ characterizing the fluid-mesh-motion algorithm satisfy

$$\forall m, \bar{\mathbf{T}}(\mathbf{u}_\Gamma^P(t^m) - \mathbf{u}_\Gamma^P(t^{m-1})) = \int_{t^{m-1}}^{t^m} \mathbf{T}(\eta) \dot{\mathbf{u}}_\Gamma(\eta) d\eta + \mathcal{O}(\Delta t^3) \quad (5.30)$$

then the GSS procedure is second-order time-accurate.

This exact same lemma still applies in our case. The only difference comes from the time-integration schemes, but the truncation error of their solutions are similar. As a result, the proof provided in [15] is still valid with the adequate afore-proven truncation errors of both the fluid and the structure subsystems and the result still holds. For the sake of clarity, the proof is recalled in Appendix A.

As proven in [15], it is possible to design loosely-coupled and second-order time-accurate algorithm for the solution of the coupled system of ordinary differential equations governing nonlinear fluid-structure interaction problems. In particular, a midpoint rule structure time integrator and a three-point-backward-difference fluid time integrator are used in the algorithm presented therein. The remainder of this section derives other loosely-coupled second-order time-accurate algorithm with different types of time-integrators for both the fluid and the structure.

Proposition 1 *If \mathbf{f}_{ae} is a sufficiently smooth function of pressure and of the position of the structure, if pressure and position of the structure are sufficiently smooth functions of time, then the GSS procedure equipped with:*

1. *the ALE version ERK-2 of the Runge-Kutta 2 scheme for time-integrating the fluid subsystem with an offset equal to half-a-time-step*

$$\Omega^{n+\frac{1}{2}} \mathbf{U}^* = \Omega^{n-\frac{1}{2}} \mathbf{U}^{n-\frac{1}{2}} - \Delta t \Phi(\mathbf{U}^{n-\frac{1}{2}}, \mathbf{x}^{n+\frac{1}{2}P}, \mathbf{x}^{n-\frac{1}{2}P})$$

$$\Omega^{n+\frac{1}{2}}\mathbf{U}^{n+\frac{1}{2}} = \Omega^{n-\frac{1}{2}}\mathbf{U}^{n-\frac{1}{2}} - \frac{\Delta t}{2} \left(\Phi(\mathbf{U}^{n-\frac{1}{2}}, \mathbf{x}^{n+\frac{1}{2}P}, \mathbf{x}^{n-\frac{1}{2}P}) + \Phi(\mathbf{U}^*, \mathbf{x}^{n+\frac{1}{2}P}, \mathbf{x}^{n-\frac{1}{2}P}) \right)$$

2. the central difference scheme for time-integrating the structure subsystem with equilibrium at t^n

$$M\ddot{\mathbf{u}}^n + \mathbf{f}_{int}(\mathbf{u}^n, \dot{\mathbf{u}}^n) = \mathbf{f}_{ae}^{nC} + \mathbf{f}_{ext}^n$$

and the solution computed at t^{n+1} ,

3. the numerical scheme

$$\mathbf{x}^{n+\frac{1}{2}P} = \mathbf{x}^{n-\frac{1}{2}P} + \overline{\mathbf{T}}(\mathbf{u}_\Gamma^{n+\frac{1}{2}P} - \mathbf{u}_\Gamma^{n-\frac{1}{2}P})$$

with $\overline{\mathbf{T}} = \mathbf{T}^n$ for updating the position of the dynamic fluid-mesh,

4. the following second-order structure predictor

$$\mathbf{u}_\Gamma^{n+\frac{1}{2}P} = \mathbf{u}_\Gamma^n + \frac{7\Delta t}{8}\dot{\mathbf{u}}_\Gamma^{n-\frac{1}{2}} - \frac{3\Delta t}{8}\dot{\mathbf{u}}_\Gamma^{n-\frac{3}{2}},$$

which can be rewritten $\mathbf{u}_\Gamma^{n+\frac{1}{2}P} = \mathbf{u}_\Gamma^n + \frac{\Delta t}{2}\dot{\mathbf{u}}_\Gamma^{n-\frac{1}{2}} + \frac{3\Delta t}{8}(\dot{\mathbf{u}}_\Gamma^{n-\frac{1}{2}} - \dot{\mathbf{u}}_\Gamma^{n-\frac{3}{2}})$,

5. and the aerodynamic force vector

$$\mathbf{f}_{ae}^{nC} = \frac{\mathbf{f}_{ae}(p^{n-\frac{1}{2}}, \mathbf{x}_\Gamma^{n-\frac{1}{2}P}) + \mathbf{f}_{ae}(p^{n+\frac{1}{2}}, \mathbf{x}_\Gamma^{n+\frac{1}{2}P})}{2}$$

is second-order time-accurate.

Proof. The Taylor expansion around $\mathbf{x}_\Gamma(t^{n+\frac{1}{2}})$ and $\mathbf{x}_\Gamma(t^{n-\frac{1}{2}})$ of the chosen expression for \mathbf{f}_{ae}^{nC} is done as follows.

First consider that, for $e \in \{-1, 1\}$

$$\mathbf{f}_{\text{ae}}(p(t^{n+\frac{\epsilon}{2}}), \mathbf{x}_{\Gamma}^P(t^{n+\frac{\epsilon}{2}})) = \mathbf{f}_{\text{ae}}(p(t^{n+\frac{\epsilon}{2}}), \mathbf{x}_{\Gamma}(t^{n+\frac{\epsilon}{2}})) + \mathcal{O}(\|\mathbf{x}_{\Gamma}^P(t^{n+\frac{\epsilon}{2}}) - \mathbf{x}_{\Gamma}(t^{n+\frac{\epsilon}{2}})\|).$$

Then, the Taylor expansion of $\mathbf{f}_{\text{ae}}(p(t^{n+\frac{\epsilon}{2}}), \mathbf{x}_{\Gamma}(t^{n+\frac{\epsilon}{2}}))$ around $(p(t^n), \mathbf{x}_{\Gamma}(t^n))$ is

$$\begin{aligned} \mathbf{f}_{\text{ae}}(p(t^{n+\frac{\epsilon}{2}}), \mathbf{x}_{\Gamma}(t^{n+\frac{\epsilon}{2}})) &= \mathbf{f}_{\text{ae}}(p(t^n), \mathbf{x}_{\Gamma}(t^n)) + \frac{\partial \mathbf{f}_{\text{ae}}}{\partial p}(p(t^n), \mathbf{x}_{\Gamma}(t^n)) (p(t^{n+\frac{\epsilon}{2}}) - p(t^n)) \\ &+ \frac{\partial \mathbf{f}_{\text{ae}}}{\partial \mathbf{x}}(p(t^n), \mathbf{x}_{\Gamma}(t^n)) (\mathbf{x}_{\Gamma}(t^{n+\frac{\epsilon}{2}}) - \mathbf{x}_{\Gamma}(t^n)) \\ &+ \mathcal{O}(\|p(t^{n+\frac{\epsilon}{2}}) - p(t^n)\|^2 + \|\mathbf{x}_{\Gamma}(t^{n+\frac{\epsilon}{2}}) - \mathbf{x}_{\Gamma}(t^n)\|^2) \end{aligned}$$

and using the Taylor series of $p(t^{n+\frac{\epsilon}{2}})$ and $\mathbf{x}_{\Gamma}(t^{n+\frac{\epsilon}{2}})$ around t^n

$$g(t^{n+\frac{\epsilon}{2}}) - g(t^n) = \frac{\partial g}{\partial t}(t^n) (t^{n+\frac{\epsilon}{2}} - t^n) + \mathcal{O}(\Delta t^2) \quad \text{where } g(\cdot) \in \{p(\cdot), \mathbf{x}_{\Gamma}(\cdot)\},$$

this leads to

$$\begin{aligned} \mathbf{f}_{\text{ae}}(p(t^{n+\frac{\epsilon}{2}}), \mathbf{x}_{\Gamma}(t^{n+\frac{\epsilon}{2}})) &= \mathbf{f}_{\text{ae}}(p(t^n), \mathbf{x}_{\Gamma}(t^n)) + \frac{\partial \mathbf{f}_{\text{ae}}}{\partial p}(p(t^n), \mathbf{x}_{\Gamma}(t^n)) \left(\frac{\partial p}{\partial t}(t^n) (t^{n+\frac{\epsilon}{2}} - t^n) \right) \\ &+ \frac{\partial \mathbf{f}_{\text{ae}}}{\partial \mathbf{x}}(p(t^n), \mathbf{x}_{\Gamma}(t^n)) \left(\frac{\partial \mathbf{x}_{\Gamma}}{\partial t}(t^n) (t^{n+\frac{\epsilon}{2}} - t^n) \right) \\ &+ \mathcal{O}(\Delta t^2). \end{aligned}$$

Thus, the Taylor expansion of the chosen expression for $\mathbf{f}_{\text{ae}}^{n_C}$ is

$$\begin{aligned} \mathbf{f}_{\text{ae}}(p(t^n), \mathbf{x}_\Gamma(t^n)) &+ \frac{1}{2} \frac{\partial \mathbf{f}_{\text{ae}}}{\partial p}(p(t^n), \mathbf{x}_\Gamma(t^n)) \left(\frac{\partial p}{\partial t}(t^n) (t^{n+\frac{1}{2}} - 2t^n + t^{n-\frac{1}{2}}) \right) \\ &+ \frac{1}{2} \frac{\partial \mathbf{f}_{\text{ae}}}{\partial \mathbf{x}}(p(t^n), \mathbf{x}_\Gamma(t^n)) \left(\frac{\partial \mathbf{x}_\Gamma}{\partial t}(t^n) (t^{n+\frac{1}{2}} - 2t^n + t^{n-\frac{1}{2}}) \right) \\ &+ \mathcal{O} \left(\|\mathbf{x}_\Gamma^P(t^{n-\frac{1}{2}}) - \mathbf{x}_\Gamma(t^{n-\frac{1}{2}})\| + \|\mathbf{x}_\Gamma^P(t^{n+\frac{1}{2}}) - \mathbf{x}_\Gamma(t^{n+\frac{1}{2}})\| \right) \\ &+ \mathcal{O}(\Delta t^2) \end{aligned}$$

which, using the assumption that the time stepping is uniform, simplifies to

$$\mathbf{f}_{\text{ae}}(p(t^n), \mathbf{x}_\Gamma(t^n)) + \mathcal{O} \left(\|\mathbf{x}_\Gamma^P(t^{n-\frac{1}{2}}) - \mathbf{x}_\Gamma(t^{n-\frac{1}{2}})\| + \|\mathbf{x}_\Gamma^P(t^{n+\frac{1}{2}}) - \mathbf{x}_\Gamma(t^{n+\frac{1}{2}})\| + \Delta t^2 \right)$$

Since

$$\begin{aligned} \mathbf{x}_\Gamma(t) &= \mathbf{x}_\Gamma(\tau) + \mathbf{T}_\Gamma (\mathbf{u}_\Gamma(t) - \mathbf{u}_\Gamma(\tau)), \quad \forall \tau \in R, \\ \mathbf{x}^P(t, \Delta t) &= \mathbf{x}^P(t - \Delta t) + \overline{\mathbf{T}}(t - \Delta t, t - 2\Delta t, \dots)(\mathbf{u}_\Gamma^P(t) - \mathbf{u}_\Gamma^P(t - \Delta t)) \end{aligned}$$

and using the fact that \mathbf{T}_Γ is a time-independent matrix, this can also be written as

$$\mathbf{f}_{\text{ae}}(p(t^n), \mathbf{x}_\Gamma(t^n)) + \mathcal{O} \left(\|\mathbf{u}_\Gamma^P(t^{n-\frac{1}{2}}) - \mathbf{u}_\Gamma(t^{n-\frac{1}{2}})\| + \|\mathbf{u}_\Gamma^P(t^{n+\frac{1}{2}}) - \mathbf{u}_\Gamma(t^{n+\frac{1}{2}})\| + \Delta t^2 \right).$$

Since the structural predictor is more than first-order accurate (as proven in the next paragraph), it follows that

$$\frac{\mathbf{f}_{\text{ae}}(p(t^{n-\frac{1}{2}}), \mathbf{x}_\Gamma^P(t^{n-\frac{1}{2}})) + \mathbf{f}_{\text{ae}}(p(t^{n+\frac{1}{2}}), \mathbf{x}_\Gamma^P(t^{n+\frac{1}{2}}))}{2} = \mathbf{f}_{\text{ae}}(p(t^n), \mathbf{x}_\Gamma(t^n)) + \mathcal{O}(\Delta t^2)$$

which proves the first part of the proposition 1.

Before proving the second part of the proposition, the accuracy of the predictor for the displacement is verified. For clarity, the subscript Γ and the superscript P are momentarily dropped.

$$\begin{aligned}
\mathbf{u}^{n+\frac{1}{2}} &= \mathbf{u}(t^n) + \frac{7\Delta t}{8}\dot{\mathbf{u}}(t^{n-\frac{1}{2}}) - \frac{3\Delta t}{8}\dot{\mathbf{u}}(t^{n-\frac{3}{2}}) \\
&= \mathbf{u}(t^{n+\frac{1}{2}}) - \frac{1}{2}\Delta t\dot{\mathbf{u}}(t^{n+\frac{1}{2}}) + \frac{1}{8}\Delta t^2\ddot{\mathbf{u}}(t^{n+\frac{1}{2}}) \\
&+ \frac{7}{8}\Delta t \left(\dot{\mathbf{u}}(t^{n+\frac{1}{2}}) - \Delta t\ddot{\mathbf{u}}(t^{n+\frac{1}{2}}) \right) - \frac{3}{8}\Delta t \left(\dot{\mathbf{u}}(t^{n+\frac{1}{2}}) - 2\Delta t\ddot{\mathbf{u}}(t^{n+\frac{1}{2}}) \right) + \mathcal{O}(\Delta t^3) \\
&= \mathbf{u}(t^{n+\frac{1}{2}}) + \mathcal{O}(\Delta t^3)
\end{aligned}$$

which proves that the structural predictor is second-order accurate.

The proof to the second part of proposition 1 is as follows.

Taylor series and integration lead to

$$\int_{t^{m-\frac{1}{2}}}^{t^{m+\frac{1}{2}}} \mathbf{T}(\eta)\dot{\mathbf{u}}_{\Gamma}(\eta)d\eta = \mathbf{T}(t^m)\dot{\mathbf{u}}_{\Gamma}(t^m)\Delta t + \mathcal{O}(\Delta t^3)$$

Moreover, assuming that the exact solutions $\mathbf{u}_{\Gamma}(t^{m+\frac{1}{2}})$ and $\mathbf{u}_{\Gamma}(t^{m-\frac{1}{2}})$ are known, Taylor expansions around t^m provides

$$\begin{aligned}
\bar{\mathbf{T}} &= \mathbf{T}(t^m) + \mathcal{O}(\Delta t) \\
\mathbf{u}_{\Gamma}^P(t^{m+\frac{1}{2}}) - \mathbf{u}_{\Gamma}^P(t^{m-\frac{1}{2}}) &= \mathbf{u}_{\Gamma}(t^{m+\frac{1}{2}}) - \mathbf{u}_{\Gamma}(t^{m-\frac{1}{2}}) + \mathcal{O}(\Delta t^3) \\
&= \mathbf{u}_{\Gamma}(t^m) + \frac{\Delta t}{2}\dot{\mathbf{u}}_{\Gamma}(t^m) + \frac{\Delta t^2}{8}\ddot{\mathbf{u}}_{\Gamma}(t^m) \\
&\quad - \mathbf{u}_{\Gamma}(t^m) + \frac{\Delta t}{2}\dot{\mathbf{u}}_{\Gamma}(t^m) - \frac{\Delta t^2}{8}\ddot{\mathbf{u}}_{\Gamma}(t^m) + \mathcal{O}(\Delta t^3) \\
&= \Delta t\dot{\mathbf{u}}_{\Gamma}(t^m) + \mathcal{O}(\Delta t^3).
\end{aligned}$$

Thus,

$$\forall m, \quad \overline{\mathbf{T}} \left(\mathbf{u}_\Gamma^P(t^{m+\frac{1}{2}}) - \mathbf{u}_\Gamma^P(t^{m-\frac{1}{2}}) \right) = \int_{t^{m-\frac{1}{2}}}^{t^{m+\frac{1}{2}}} \mathbf{T}(\eta) \dot{\mathbf{u}}_\Gamma(\eta) d\eta + \mathcal{O}(\Delta t^3)$$

which completes the proof.

Proposition 2 *If \mathbf{f}_{ae} is a sufficiently smooth function of pressure and of the position of the structure, if pressure and position of the structure are sufficiently smooth functions of time, then the GSS procedure equipped with:*

1. *the ALE version of the three-point-backward-difference scheme for time-integrating the fluid subsystem with an offset equal to half-a-time-step*

$$\frac{3}{2}\Omega^{n+\frac{1}{2}}\mathbf{U}^{n+\frac{1}{2}} - 2\Omega^{n-\frac{1}{2}}\mathbf{U}^{n-\frac{1}{2}} + \frac{1}{2}\Omega^{n-\frac{3}{2}}\mathbf{U}^{n-\frac{3}{2}} + \Delta t \Phi(\mathbf{U}^{n+\frac{1}{2}}, \mathbf{x}^{n+\frac{1}{2}P}, \mathbf{x}^{n-\frac{1}{2}P}, \mathbf{x}^{n-\frac{3}{2}P}) = 0$$

2. *the central difference scheme for time-integrating the structure subsystem with equilibrium at t^n*

$$M\ddot{\mathbf{u}}^n + \mathbf{f}_{int}(\mathbf{u}^n, \dot{\mathbf{u}}^n) = \mathbf{f}_{ae}^C + \mathbf{f}_{ext}^n$$

and the solution computed at t^{n+1} ,

3. *the numerical scheme*

$$\mathbf{x}^{n+\frac{1}{2}P} = \mathbf{x}^{n-\frac{1}{2}P} + \overline{\mathbf{T}}(\mathbf{u}_\Gamma^{n+\frac{1}{2}P} - \mathbf{u}_\Gamma^{n-\frac{1}{2}P})$$

with $\overline{\mathbf{T}} = \mathbf{T}^n$ for updating the position of the dynamic fluid-mesh,

4. the following second-order structure predictor

$$\mathbf{u}_\Gamma^{n+\frac{1}{2}P} = \mathbf{u}_\Gamma^n + \frac{7\Delta t}{8} \dot{\mathbf{u}}_\Gamma^{n-\frac{1}{2}} - \frac{3\Delta t}{8} \dot{\mathbf{u}}_\Gamma^{n-\frac{3}{2}},$$

which can be rewritten $\mathbf{u}_\Gamma^{n+\frac{1}{2}P} = \mathbf{u}_\Gamma^n + \frac{\Delta t}{2} \dot{\mathbf{u}}_\Gamma^{n-\frac{1}{2}} + \frac{3\Delta t}{8} \left(\dot{\mathbf{u}}_\Gamma^{n-\frac{1}{2}} - \dot{\mathbf{u}}_\Gamma^{n-\frac{3}{2}} \right)$,

5. and the aerodynamic force vector

$$\mathbf{f}_{ae}^{nC} = \frac{\mathbf{f}_{ae}(p^{n-\frac{1}{2}}, \mathbf{x}_\Gamma^{n-\frac{1}{2}P}) + \mathbf{f}_{ae}(p^{n+\frac{1}{2}}, \mathbf{x}_\Gamma^{n+\frac{1}{2}P})}{2}$$

is second-order time-accurate.

Proof. This proposition is very similar to the previous one. Only the fluid time-integration has been modified, which appears only in the proof of the lemma 3 through the truncation error of its solution. Hence, the proof is the same.

Proposition 3 *If \mathbf{f}_{ae} is a sufficiently smooth function of pressure and of the position of the structure, if pressure and position of the structure are sufficiently smooth functions of time, then the GSS procedure equipped with:*

1. the ALE version ERK-2 of the Runge-Kutta 2 scheme for time-integrating the fluid subsystem with an offset equal to half-a-time-step

$$\Omega^{n+\frac{1}{2}} \mathbf{U}^* = \Omega^{n-\frac{1}{2}} \mathbf{U}^{n-\frac{1}{2}} - \Delta t \Phi(\mathbf{U}^{n-\frac{1}{2}}, \mathbf{x}^{n+\frac{1}{2}P}, \mathbf{x}^{n-\frac{1}{2}P})$$

$$\Omega^{n+\frac{1}{2}} \mathbf{U}^{n+\frac{1}{2}} = \Omega^{n-\frac{1}{2}} \mathbf{U}^{n-\frac{1}{2}} - \frac{\Delta t}{2} \left(\Phi(\mathbf{U}^{n-\frac{1}{2}}, \mathbf{x}^{n+\frac{1}{2}P}, \mathbf{x}^{n-\frac{1}{2}P}) + \Phi(\mathbf{U}^*, \mathbf{x}^{n+\frac{1}{2}P}, \mathbf{x}^{n-\frac{1}{2}P}) \right)$$

2. the midpoint rule for time-integrating the structure subsystem with equilibrium at $t^{n+\frac{1}{2}}$

$$M\ddot{\mathbf{u}}^{n+\frac{1}{2}} + \mathbf{f}_{int}(\mathbf{u}^{n+\frac{1}{2}}, \dot{\mathbf{u}}^{n+\frac{1}{2}}) = \mathbf{f}_{ae}^{n+\frac{1}{2}C} + \mathbf{f}_{ext}^{n+\frac{1}{2}}$$

and the solution computed at t^{n+1} via $\mathbf{u}^{n+1} = 2\mathbf{u}^{n+\frac{1}{2}} - \mathbf{u}^n$,

3. the numerical scheme

$$\mathbf{x}^{n+\frac{1}{2}P} = \mathbf{x}^{n-\frac{1}{2}P} + \overline{\mathbf{T}}(\mathbf{u}_\Gamma^{n+\frac{1}{2}P} - \mathbf{u}_\Gamma^{n-\frac{1}{2}P})$$

with $\overline{\mathbf{T}} = \mathbf{T}^n$ for updating the position of the dynamic fluid-mesh,

4. the following second-order structure predictor

$$\mathbf{u}_\Gamma^{n+\frac{1}{2}P} = \mathbf{u}_\Gamma^n + \frac{\Delta t}{2}\dot{\mathbf{u}}_\Gamma^n + \frac{\Delta t}{8}(\dot{\mathbf{u}}_\Gamma^n - \dot{\mathbf{u}}_\Gamma^{n-1}),$$

5. and the aerodynamic force vector

$$\mathbf{f}_{ae}^{n+\frac{1}{2}C} = \mathbf{f}_{ae}(p^{n+\frac{1}{2}}, \mathbf{x}_\Gamma^{n+\frac{1}{2}P})$$

is second-order time-accurate.

Proof. The only difference between this algorithm and the second algorithm proposed in [15] is the fluid time-integration scheme. For the same reasons that the proof between propositions 1 and 2 is the same, the proof of the above proposition is the same as the one given in [15].

5.6 Numerical Accuracy Study

The purpose of this section is to validate the accuracy properties of the scheme presented in the previous section. To this end, this section considers the simulation of a fluid-structure interaction where an aluminum cylinder containing a fluid is also surrounded by another fluid. Subscripts i and o respectively refer to quantities for the fluid inside and for the fluid outside the cylinder. Both fluids are modeled by perfect gas with a specific heat ratio $\gamma_i = \gamma_o = 1.4$. The fluids are initially at rest ($u_i = u_o = 0 \text{ m.s}^{-1}$) and have identical densities of $\rho_i = \rho_o = 13.0 \text{ kg.m}^{-3}$. A pressure difference $p_o - p_i = 10^8 - 10^7 = 9 \times 10^7 \text{ Pa}$ between the two fluids creates a displacement of the structure, which in turn modifies the flow field. The structure is modeled with an elastic material of density $\rho = 2660.0 \text{ kg.m}^{-3}$, thickness $h = 1.651 \text{ mm}$, Young modulus $E = 71.0 \times 10^9 \text{ Pa}$ and Poisson ratio $\nu = 0.3$. The cylinder has diameter $D = 0.1 \text{ m}$ and length $L = 0.3 \text{ m}$. Unless specified otherwise, the structure presents no geometrical nonlinearities. The finite volume fluid mesh is composed of 1,660,961 tetrahedra (296,583 nodes) and the finite element structure mesh is composed of 4806 3-node shell elements (Fig. 5.2).

First, the variation of the order of accuracy with respect to the structural predictor and the corrected force is studied. In this case, geometric nonlinearities are not modeled. Five different schemes are used. The different schemes are as follows. Scheme A is the actual scheme presented in proposition 1: the Runge-Kutta 2 fluid time-integrator and the explicit Central Difference structure time-integrator are combined with a structural predictor

$$\mathbf{u}_\Gamma^{n+\frac{1}{2}P} = \mathbf{u}_\Gamma^n + \frac{7\Delta t}{8} \dot{\mathbf{u}}_\Gamma^{n-\frac{1}{2}} - \frac{3\Delta t}{8} \ddot{\mathbf{u}}_\Gamma^{n-\frac{3}{2}}$$

	simu #1	simu #2	simu #3	simu #4	reference
structure	3e-8	1.5e-8	7.5e-9	3.75e-9	3.75e-10

Table 5.1: Time steps for the different simulations

and the corrected force

$$\mathbf{f}_{\text{ae}}^{n^C} = \frac{\mathbf{f}_{\text{ae}}(p^{n-\frac{1}{2}}, \mathbf{x}_{\Gamma}^{n-\frac{1}{2}P}) + \mathbf{f}_{\text{ae}}(p^{n+\frac{1}{2}}, \mathbf{x}_{\Gamma}^{n+\frac{1}{2}P})}{2}.$$

Scheme B is the same as scheme A except that the structural predictor is given by

$$\mathbf{u}_{\Gamma}^{n+\frac{1}{2}P} = \mathbf{u}_{\Gamma}^n + \frac{\Delta t}{2} \dot{\mathbf{u}}_{\Gamma}^{n-\frac{1}{2}} + \frac{\Delta t}{4} \left(\dot{\mathbf{u}}_{\Gamma}^{n-\frac{1}{2}} - \dot{\mathbf{u}}_{\Gamma}^{n-\frac{3}{2}} \right).$$

Scheme C is the same as scheme A except that the structural predictor is given by

$$\mathbf{u}_{\Gamma}^{n+\frac{1}{2}P} = \mathbf{u}_{\Gamma}^n + \frac{\Delta t}{5} \dot{\mathbf{u}}_{\Gamma}^{n-\frac{1}{2}}.$$

Scheme D is the same as scheme A except that the corrected force is given by

$$\mathbf{f}_{\text{ae}}^{n^C} = \mathbf{f}_{\text{ae}}(p^{n-\frac{1}{2}}, \mathbf{x}_{\Gamma}^{n-\frac{1}{2}P}).$$

Scheme E is the same as scheme A with no prediction and no correction, that is

$$\mathbf{u}_{\Gamma}^{n+\frac{1}{2}P} = \mathbf{u}_{\Gamma}^n \quad \text{and} \quad \mathbf{f}_{\text{ae}}^{n^C} = \mathbf{f}_{\text{ae}}(p^{n-\frac{1}{2}}, \mathbf{x}_{\Gamma}^{n-\frac{1}{2}P}).$$

For each scheme, 4 simulations with different time steps were performed, as specified in columns 2 through 5 in table 5.1. An additional simulation was performed with Scheme A and with a very refined time-step (last column of table 5.1) and was used as an exact numerical solution to compute errors.

The errors of the solutions for the structure and the fluid are presented in Fig. 5.3 and Fig. 5.4 respectively. On both figures, the slopes of an exact first-order time-accurate solution and of an exact second-order time-accurate one are given for reference. They are respectively called “slope1” and “slope2”. The error computed for the structure subsystem is the L^2 -norm of the displacements only. The error computed for the fluid subsystem is the L^2 -norm of all conservative variables. Only schemes A and B are second-order accurate. Scheme A has orders of 1.93 in the structure and 1.85 in the fluid, while scheme B has orders of 1.90 in the structure and 2.07 in the fluid. The difference between the two schemes comes from the order of approximation of the structural predictor. In order to numerically obtain a second-order time-accurate solution, a first-order approximation of the structural predictor seems sufficient. However, as the order of approximation of the predictor decreases more, the second-order time-accuracy of the scheme is lost. This is shown by scheme C which is slightly less than first-order accurate with exact order 0.86 in the structure and 0.96 in the fluid. Scheme D and scheme E both have a different corrected force compared to scheme A. Both schemes are first-order time-accurate in the fluid. However, scheme E recovers second-order time-accuracy in the structure even though the combined orders of approximation of the predictor and of the corrector are the worst of all schemes. Scheme D is first-order accurate in both the structure and the fluid with slopes 0.81 and 0.86. Scheme E returns first-order accurate fluid solutions with exact order 1.02 and second-order accurate structural displacement solutions with exact order 1.85.

The same problem with geometric nonlinearities in the structure is now considered with scheme A of proposition 1 to time integrate the equations. Again solutions of both the fluid and the structure are close to being second-order time-accurate as

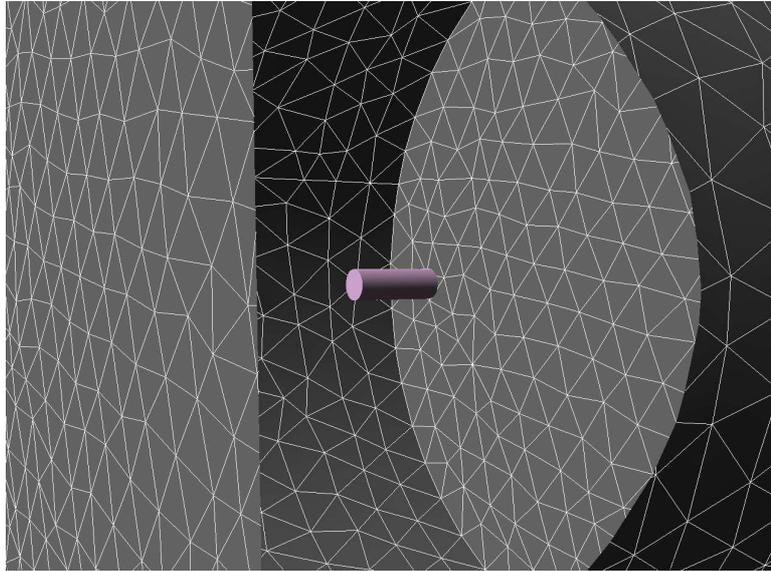


Figure 5.2: Mesh

shown by Fig. 5.5 and 5.6. Similar to the linear case, the order of time-accuracy of the structure solution decreases as the time-step approaches 10^{-9} second. This may indicate that the structure solution is close to being converged. The presence of geometric nonlinearities does not seem to modify the order of accuracy of the method.

The same problem with geometric nonlinearities is considered once more. The scheme F of proposition 2 is used to time-integrate the sets of equations. The fluid equations are integrated with the three-point backward difference scheme and the structure equations are integrated with the explicit central difference scheme. All other components of the GSS procedure are the same as for scheme A. A larger time-step is used to time-integrate the equations since the restriction of the implicit scheme remains stable for CFL values larger than one. As expected, the solution is close to being second-order time-accurate (see Fig. 5.7 and 5.8). The order of accuracy of the structure solution is 1.99 while the one of the fluid solution is 1.87.

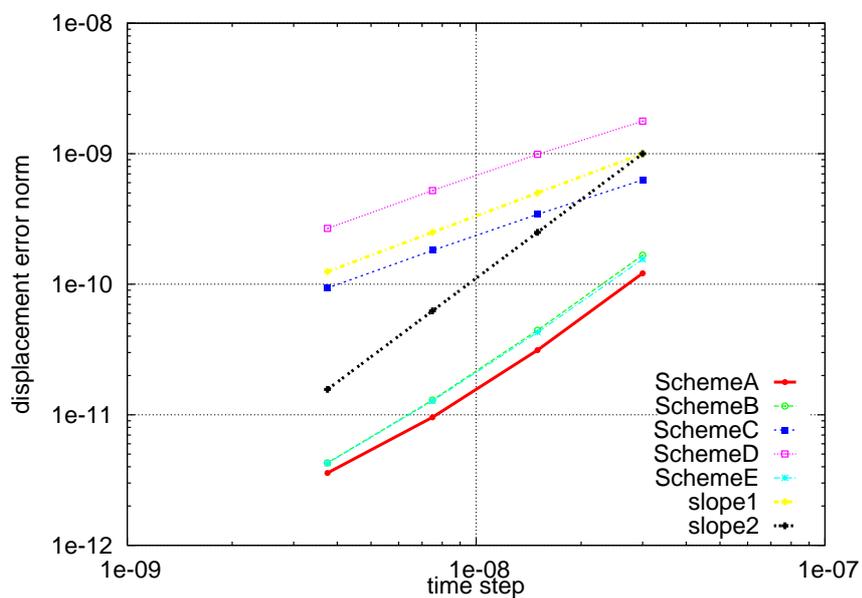


Figure 5.3: Displacement errors for the structure subsystem using schemes A through E (linear structural model).

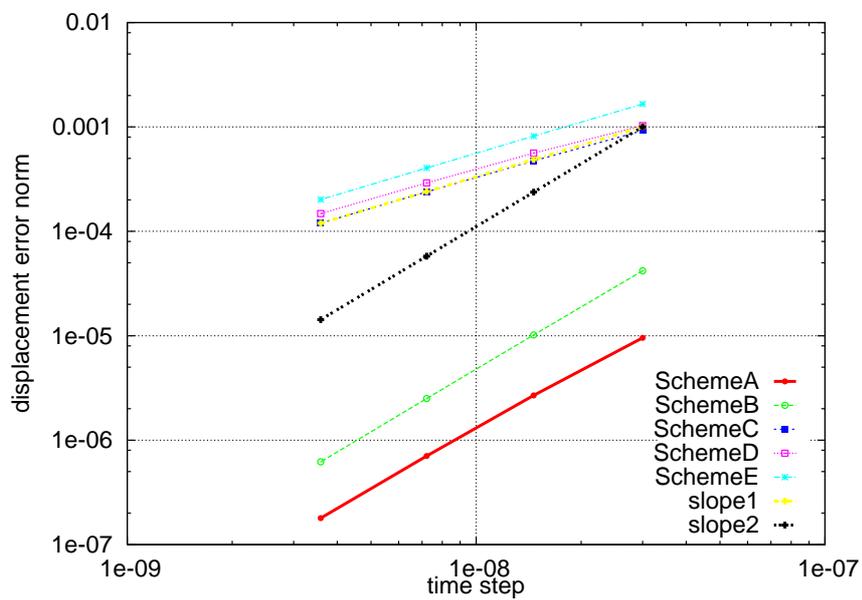


Figure 5.4: Total state errors for the fluid subsystem using schemes A through E (linear structural model).

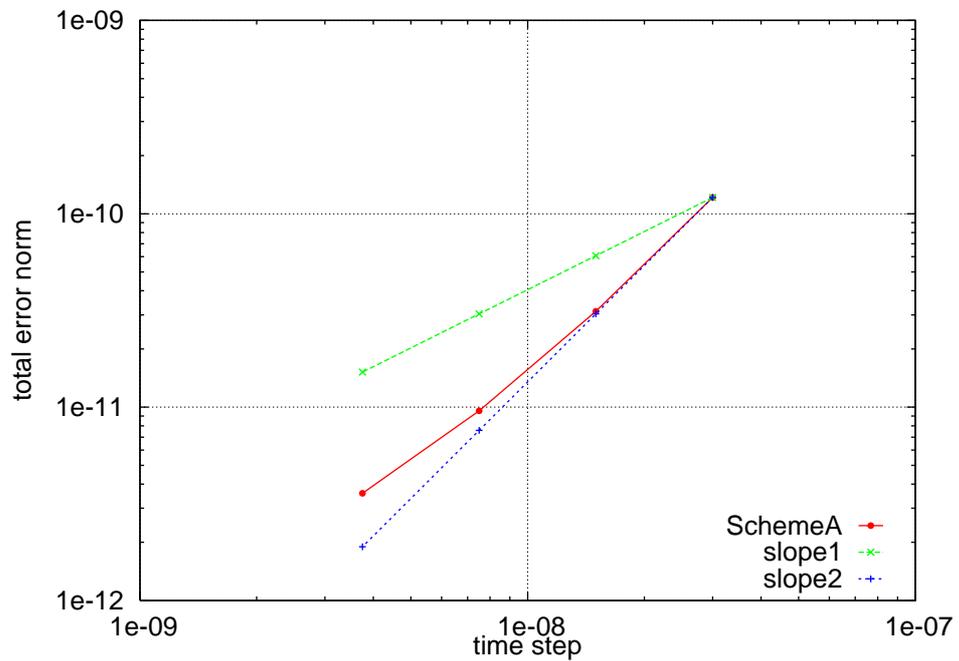


Figure 5.5: Displacement errors for the structure subsystem with scheme A (nonlinear structural model)

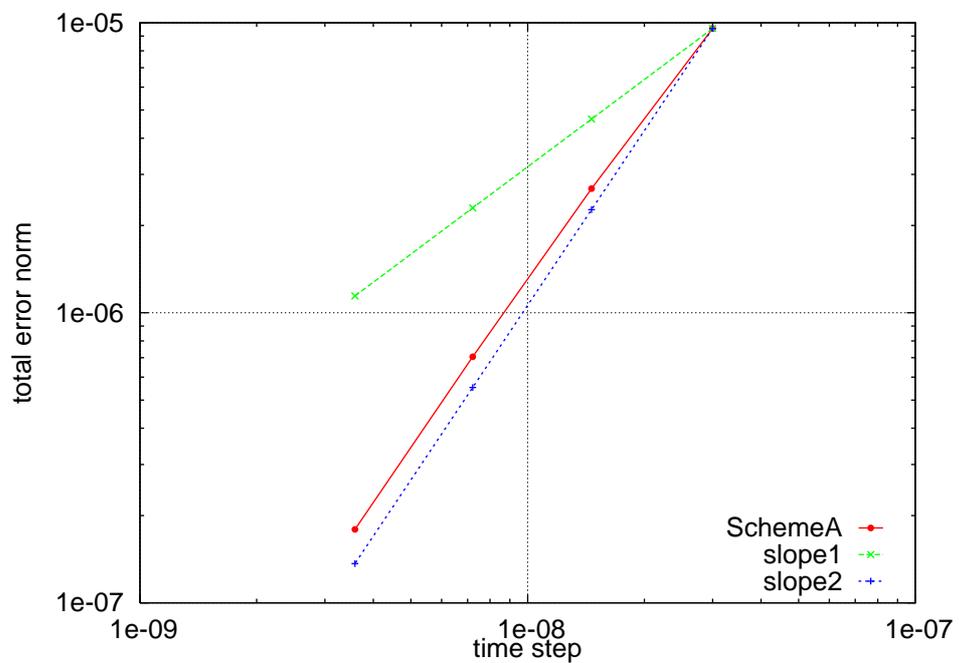


Figure 5.6: Total state errors for the fluid subsystem with scheme A (nonlinear structural model)

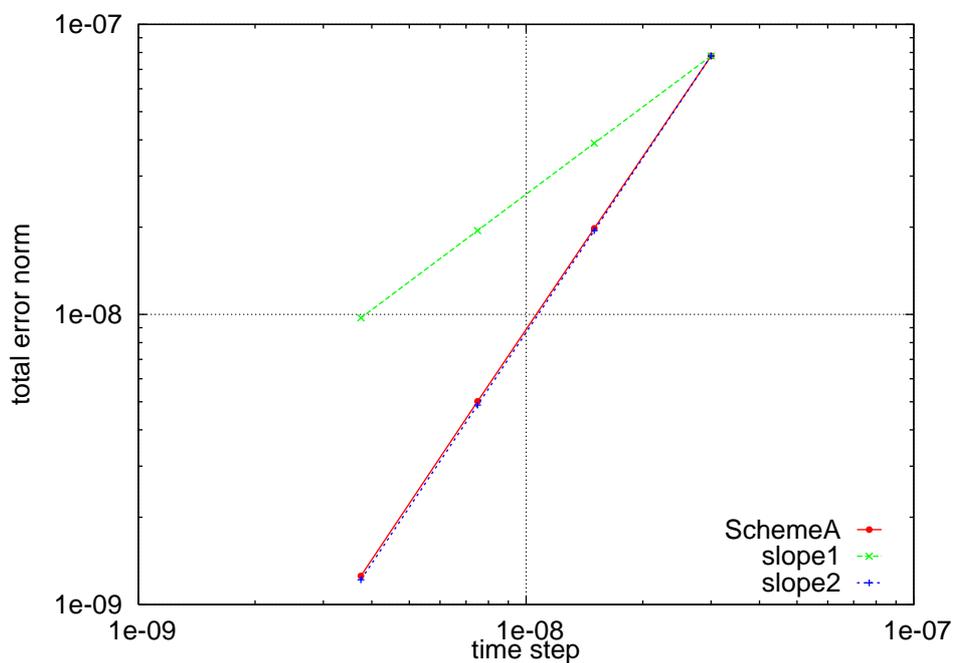


Figure 5.7: Displacement errors for the structure subsystem using scheme F (nonlinear structural model)

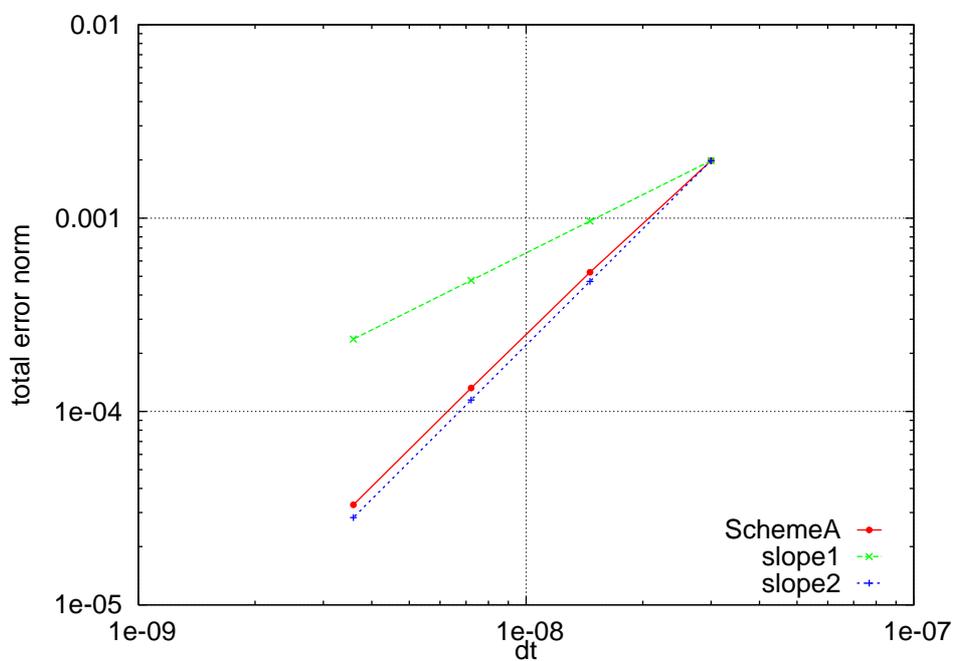


Figure 5.8: Total state errors for the fluid subsystem using scheme F (nonlinear structural model)

Chapter 6

Applications to Underwater

Implosions

The previous chapters put forth elements of a computational framework for the study of compressible multiphase flows and fluid-structure interactions on unstructured meshes. In particular, a compressible multiphase flow solver using the exact solutions of Riemann problems was developed while new second-order time-accurate staggered loosely-coupled fluid-mesh-structure solvers were designed in the ALE framework. The objective of this chapter is to assess the possibilities of using the current framework for studying underwater implosions by comparing the results obtained by both solvers with experimental data.

6.1 Collapsing Bubble

The FVM-ERS was implemented in the AERO-F flow code [62, 14]. Here, it is applied to the three-dimensional simulation of the implosion of an air-filled and submerged glass sphere. The parameters of this simulation correspond to the experiments and

test data recently reported in [82].

In the experimental setup described in [82], an air-filled glass sphere was submerged in a pressure vessel filled with water. The implosion of the glass sphere was initiated either by a critical hydrostatic pressure, or by the actuation of a piston at the bottom of the sphere. The test stand consisted of an aluminum base plate and a 7.62 cm diameter pipe standing vertically. A glass sphere with an outer radius of 3.81 cm was placed on top of the pipe. Four implosion experiments were performed with an initial hydrostatic pressure of 6.996 MPa, and an initial pressure inside the glass sphere of 101.3 kPa. Three dynamic pressure sensors were installed at 10.16 cm from the center of the sphere, at the same height, and in three directions 120° apart. For these four experiments, the recorded pressure time-histories (see Fig. 6.1) reveal pressure drops of 1.6 MPa and pressure peaks ranging between 25.8 and 27.2 MPa (variations of the order of 5%). A secondary peak can also be observed in Fig. 6.1; however, its amplitude and position in time have a greater variability than the pressure drop and primary peak.

Using the two-dimensional axisymmetric computational domain shown in Fig. 6.2, various numerical simulations were also performed by the author of [82] using the DYSMAS code [47]. In these simulations, both fluid media were assumed to be inviscid. Water was modeled by Tillotsons EOS as described in [82] and air by the perfect gas EOS. The initial conditions were set to

$$\begin{aligned}\rho_w &= 1000.0 \text{ kg/m}^3, u_w = 0 \text{ m/s}, p_w = 6.996 \text{ MPa} \\ \rho_a &= 1.3 \text{ kg/m}^3, u_a = 0 \text{ m/s}, p_a = 101.3 \text{ kPa}\end{aligned}$$

where the subscripts w and a designate water and air, respectively. An element

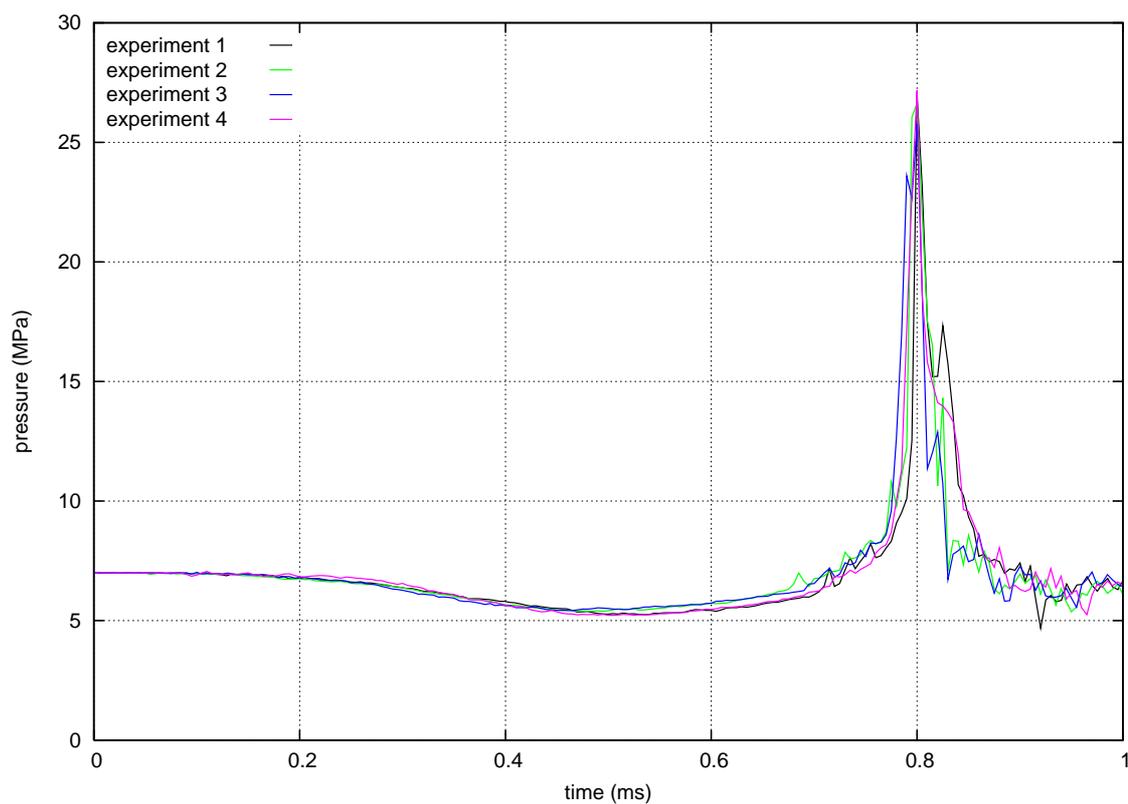


Figure 6.1: Pressure time-histories recorded at one of the sensors during the four experiments described in [82].

deletion technique for prescribing the removal of the glass material was also applied at various speeds. However, only the case where the glass was assumed to have disappeared at $t = 0$ (infinite element deletion speed) is reported here (for the sake of comparison with this papers results where no such technique was used). In general, DYSMAS predicted a pressure drop of almost 3.0 MPa and a primary pressure peak of 38.4 MPa at the sensor locations. It also predicted a secondary pressure peak and a pressure dip after both pressure peaks of approximately 4.5 MPa (see Fig. 6.3).

As mentioned at the beginning of this section, the AERO-F code equipped with the proposed FVMERS is also applied here to the simulation of the implosion experiment described above. Because the main purpose of this simulation is the verification of a three-dimensional code, a three-dimensional computational domain covering a 20° -slice of the cylindrical pressure vessel is chosen for this purpose. All other dimensions of this computational domain and corresponding non-reflecting boundaries are chosen to be the same as those used in [82]. This domain is discretized by a grid with 794,254 nodes, 4,484,412 tetrahedra, and a mesh density similar to that used for the numerical simulations reported in [82]. Symmetry boundary conditions are applied on the lateral boundaries of this computational domain. The water is modeled by the stiffened gas EOS with SG(7.15, 2.89×10^8 Pa) where the same notations as in chapter 4 are used to describe the different EOS. The air is modeled by the perfect gas EOS with PG(1.4). Both media are also assumed to be inviscid. The following initial conditions, which are consistent with those of the experiments reported in [82], are adopted. At $t = 0$, the air is assumed to occupy the same volume as the sphere of glass before it breaks, and to be still at a uniform pressure of 101.3 kPa and a uniform density of 1.3kg/m^3 . The initial hydrostatic pressure of the water surrounding this air bubble is assumed to be equal to 6.996 MPa at the depth of the center of the air

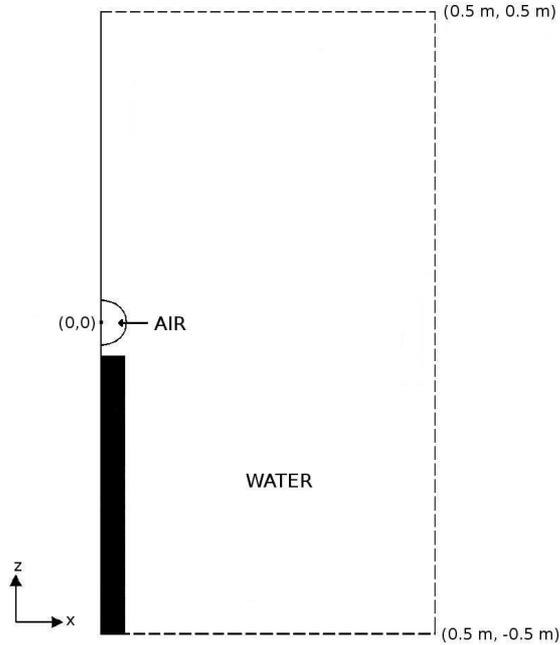


Figure 6.2: Schematic of the implosion experiment reported in [82] and corresponding computational domain (dashed lines represent the non-reflecting boundaries of the two-dimensional computational domain adopted in [82]).

bubble; its initial density is set to 1000.0kg/m^3 all over the computational domain. The AERO-F simulation is performed using a second-order space-accurate FVMERS and the second-order RungeKutta time-integrator. The CFL number is fixed to 0.5 and the computation is performed until reaching the physical time of 0.6495 ms.

Fig. 6.3 reports the pressure time-history predicted by AERO-F (equipped with the FVMERS) and compares it to: (a) that predicted by the DYSMAS code, and (b) those recorded during experiment 3 and experiment 4. The focus on the results of these two experiments is only because they envelop the results of the other two experiments, and reporting only these keeps Fig. 6.3 readable. (Note that as done in [82], time was shifted in this figure so that the pressure peak is reached at $t =$

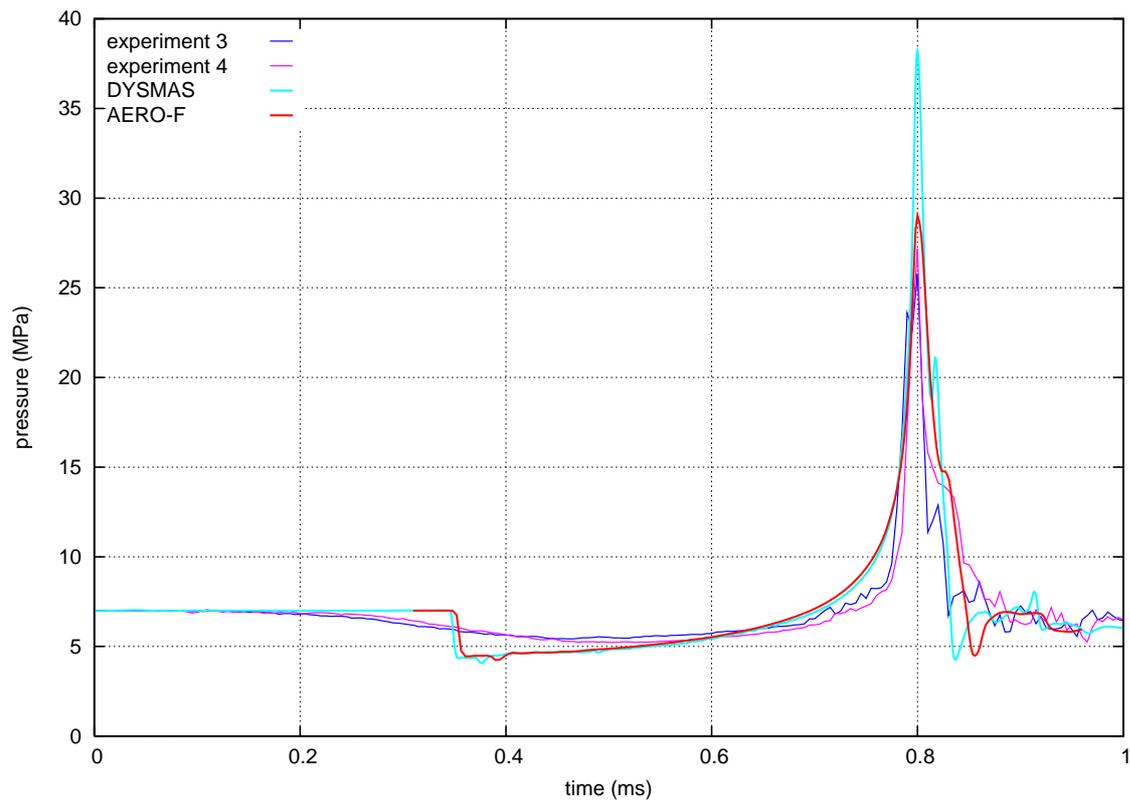


Figure 6.3: Comparison of the pressure time-histories predicted by AERO-F and DYSMAS as well as their corresponding test data.

0.8 ms). The FVMERS is shown to reproduce the recorded pressure signal fairly accurately. A first pressure drop of almost 3 MPa is predicted at $t = 0.354$ ms, slightly later than that predicted by DYSMAS ($t = 0.350$ ms). The primary pressure peak predicted by AERO-F is 29.0 MPa: it is closer to the measured pressure peak value (25.8 – 27.2 MPa) than that predicted by DYSMAS (38.4 MPa). On the other hand, the secondary peak predicted by AERO-F is flattened. The lowest pressure level predicted by AERO-F (4.5 MPa) is comparable to that predicted by DYSMAS. After this lowest pressure level is reached, both codes correctly predict similar rises to the initial pressure level.

Finally, it is noted that the implosion experiment described herein cannot be simulated by a one-dimensional spherical model. Indeed, Fig. 6.4 which displays the contour plots of the density field computed by AERO-F at $t = 0.87$ ms reveals that after some point during its collapse, the bubble is no longer spherical (until after its rebound). As mentioned already in [71] and subsequently studied both analytically and experimentally afterward, this feature is characteristic to the collapse of bubbles in the vicinity of a rigid wall, as the wall boundary generates a velocity differential between the far end and the close end of the bubble, leading to a re-entrant jet toward the wall. A similar bubble shape is found in the experimental bubble collapse near a wall by [59].

6.2 Collapsing Cylinders

The staggered loosely-coupled procedure to integrate structure and fluid equations simultaneously was implemented in the AERO-F flow code [62, 14] and the XFEM

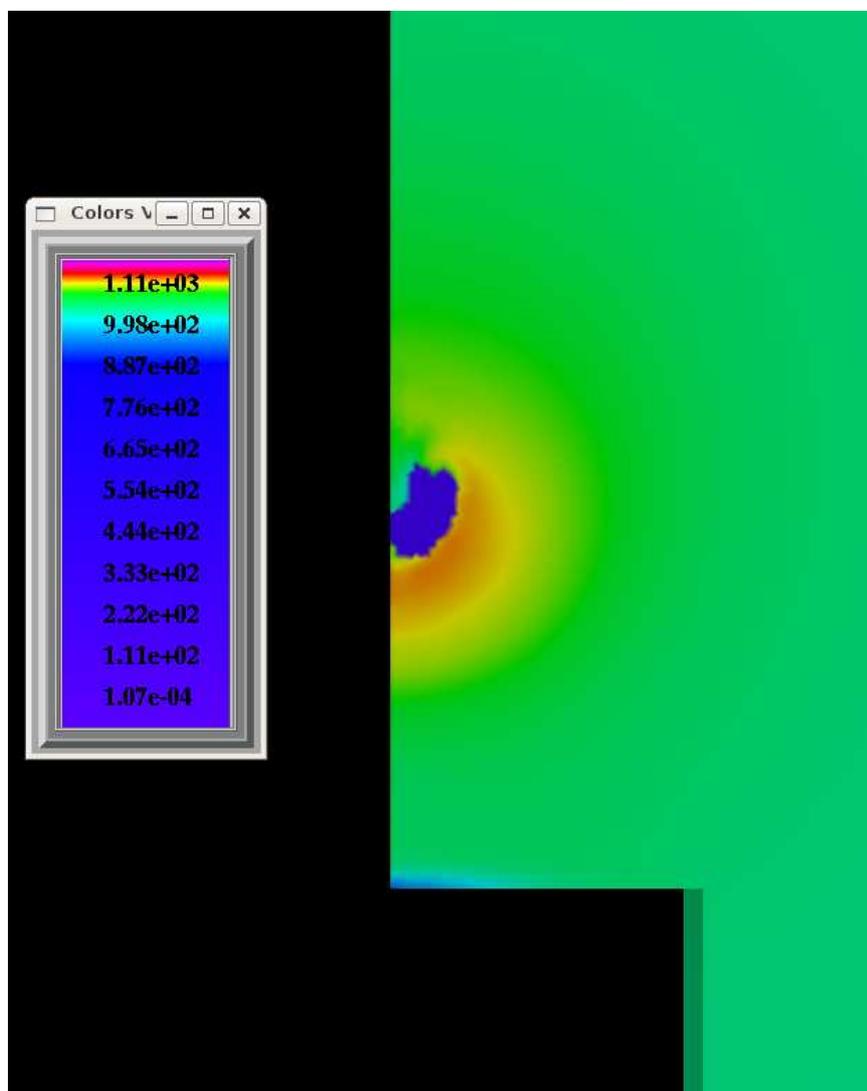


Figure 6.4: Density contour plots (in kg/m^3) predicted by AERO-F equipped with the FVM-ERS in the vicinity of the air bubble during its collapse.

structural solver [32] equipped with a pinball-based contact algorithm. This procedure is applied to the three-dimensional simulation of the collapse of an air-filled aluminum cylinder submerged in water. The parameters of this simulation correspond to experiments performed simultaneously at the University of Texas at Austin.

In the two experiments considered here, an aluminum cylinder is filled with air at atmospheric pressure and is submerged in a pressurized tank filled with water. The cylinder and the pressure sensors are maintained in the middle of the tank by an arrangement of bars fixed to the ground. The cylinder is attached to the setup through its end caps as shown in Fig. 6.5. The pressure sensors are attached to the same test stand and are positioned at roughly the same distance from the center of the cylinder at the level of the middle cross section of the cylinder. The material properties of the cylinder are characterized by a density of 2.599×10^{-4} lbs.in⁻⁴.s⁻², a Young modulus of 1.008×10^7 psi, a Poisson ratio of 0.3, a yield stress of 4.008×10^4 psi and a hardening modulus of 9.200×10^4 psi. The pressure in the tank is increased until the cylinder collapses. The dynamic collapse of the cylinder essentially happens under constant pressure.

Simulations of this type of tests are quite difficult to set up since there is no obvious initial state for the numerical simulation. During the pressurization of the tank but before the collapse, the cylinder undergoes deformations due to the increasing hydrostatic pressure and mostly retains its cylindrical shape. Only the inevitable geometric and material imperfections of the cylinder trigger the instabilities that lead to its collapse. No obvious initial state can be defined to start the numerical simulation of such a phenomenon. One possible initiation of the collapse is given in [45] which consists in using a material structure, called an indenter, to initially bend the cylinder. However, no guarantee is given that a simulation with a different

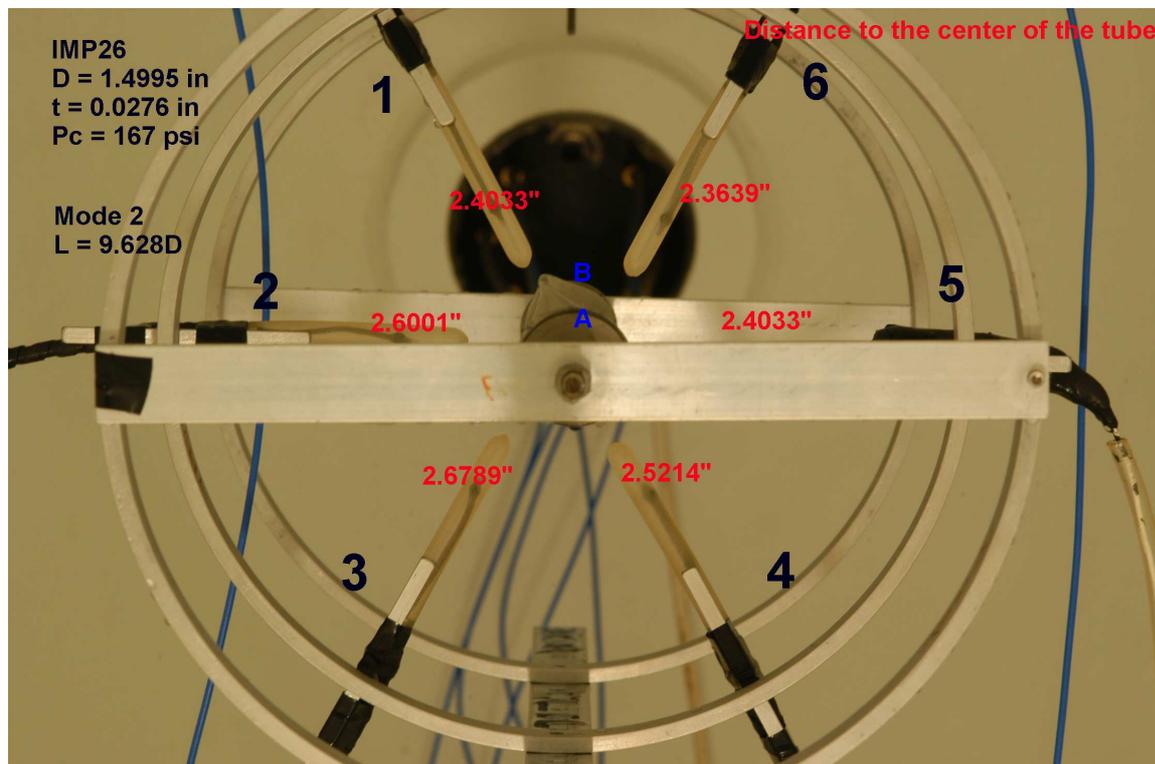


Figure 6.5: Experimental setup for the observation of the mode-2 collapse of an aluminum cylinder

initialization procedure will lead to similar results.

The remainder of the section is organized as follows. First, a study of the influence of the triggering mechanism is done on a two-dimensional cylinder. After assessing the differences between them, three-dimensional simulations of the collapse of cylinders under different conditions are performed and results are compared to experimental data.

6.2.1 Influence of the Initialization

The objective of this section is to study the variations of the pressure signals at given locations with respect to different initialization procedures. This is done on pseudo-two-dimensional meshes representing an infinite cylinder and the fluids surrounding it. The air inside the cylinder is modeled by a perfect gas with specific heat ratio of 1.4 while the water surrounding the cylinder is modeled by a stiffened gas of the form SG(4.4, 6.0×10^8). The cylinder has the same material properties as the ones in the experiments and a J_2 -flow theory plasticity model is used to describe its material behavior. Depending on its geometry, the collapse of a three-dimensional cylinder can lead to different shapes corresponding to different modes. The modal cross-sectional shapes are given by

$$r = r_0(1 - \alpha \cos(n\theta))$$

where r is the distance from the center of the original cylinder to the structure in the θ -direction, r_0 is the radius of the original cylinder, α measures the deformation of the actual structure with respect to the original cylinder and n is the mode number. In the present case of a very long cylinder, the collapse mode is 2. The same collapse pressure as the one given in the first experiment is used since both correspond to large aspect ratio, have the same collapse mode and have the same material properties.

The eight following initialization procedures are considered:

- a sudden critical pressure of 184.5 psi is initially applied to an imperfect cylinder with a 1%-mode 2 deformation,
- the pressure is uniformly and constantly increased from a value equal to the pressure inside the cylinder until the critical pressure of 184.5 psi is reached

while the initial cylinder has a 1%-mode 2 deformation,

- a sudden critical pressure of 184.5 psi is initially applied to an imperfect cylinder with a 1%-mode 4 deformation,
- the critical pressure of 184.5 psi is initially applied to a perfect cylinder while a structure indents the cylinder at a velocity of 100 in.s^{-1} ,
- the steady state of the perfect cylinder under the critical pressure of 184.5 psi is computed, and the dynamic simulation is initialized with a critical pressure of 184.5 psi and a sudden 1%-mode 2 deformation of the steady-state computed cylinder,
- the steady state of the perfect cylinder under the critical pressure of 184.5 psi is computed, and the dynamic simulation is initialized with a critical pressure of 184.5 psi and a sudden 1%-mode 4 deformation of the steady-state computed cylinder,
- the steady state of the perfect cylinder under the critical pressure of 184.5 psi is computed, and the dynamic simulation is initialized with a critical pressure of 184.5 psi while a structure indents the steady-state computed cylinder at a velocity of 100 in.s^{-1} ,
- the steady state of the perfect cylinder under the critical pressure of 184.5 psi is computed, and the dynamic simulation is initialized with a critical pressure of 184.5 psi while the steady-state structure is suddenly weakened by increasing the Young modulus of one of its element by one percent.

The comparison between the different initialization procedures is based on the final shape of the structure and the pressure signals recorded at a distance of 2.5 inches

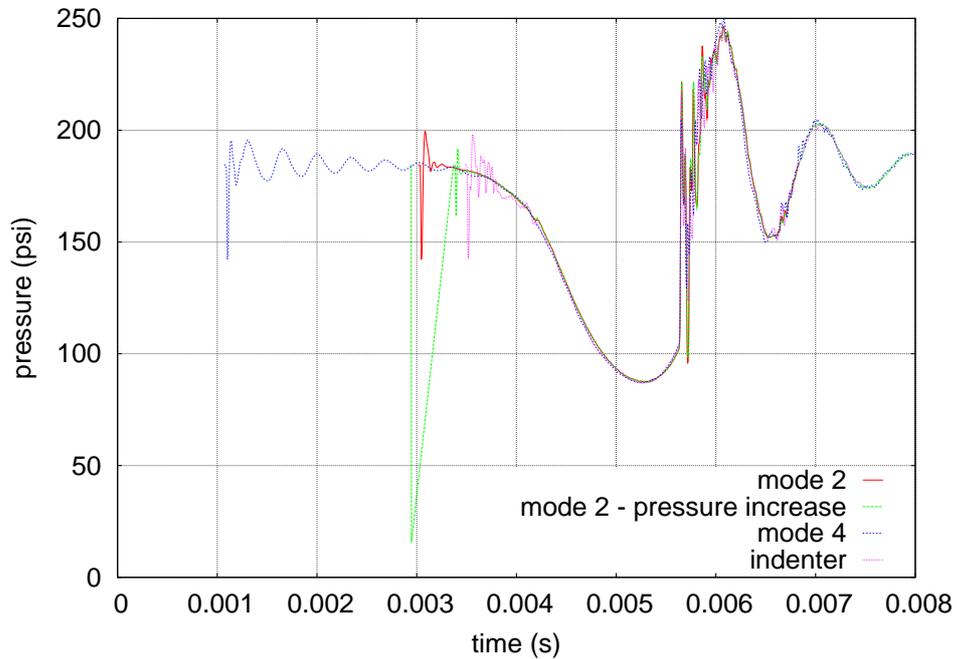


Figure 6.6: Pressure signals in the vicinity of a cylinder collapsing in mode 2 at the critical pressure of 184.5 psi. The curves correspond to different initializations of the simulation: (a) the critical pressure is initially applied to an imperfect cylinder with a 1%-mode 2 deformation, (b) pressure is increased until the critical pressure is reached while the cylinder has an initial 1%-mode 2 deformation, (c) the critical pressure is initially applied to an imperfect cylinder with a 1%-mode 4 deformation, (d) the critical pressure is initially applied to a perfect cylinder being deformed by an indenter.

from the center of the initial structure. All computations led to a collapse of the cylinder in mode 2, even with initialization procedures where an initial mode 4 deformation was applied to the structure. This shows that the solution is not sensitive to the mode of the initial deformation. Pressure signals are depicted in Fig. 6.6 for simulations using one of the first four initialization procedures where no original steady state is computed and in Fig. 6.7 for simulations using one of the other initialization procedures. Note that pressure signals were shifted in time so that all primary

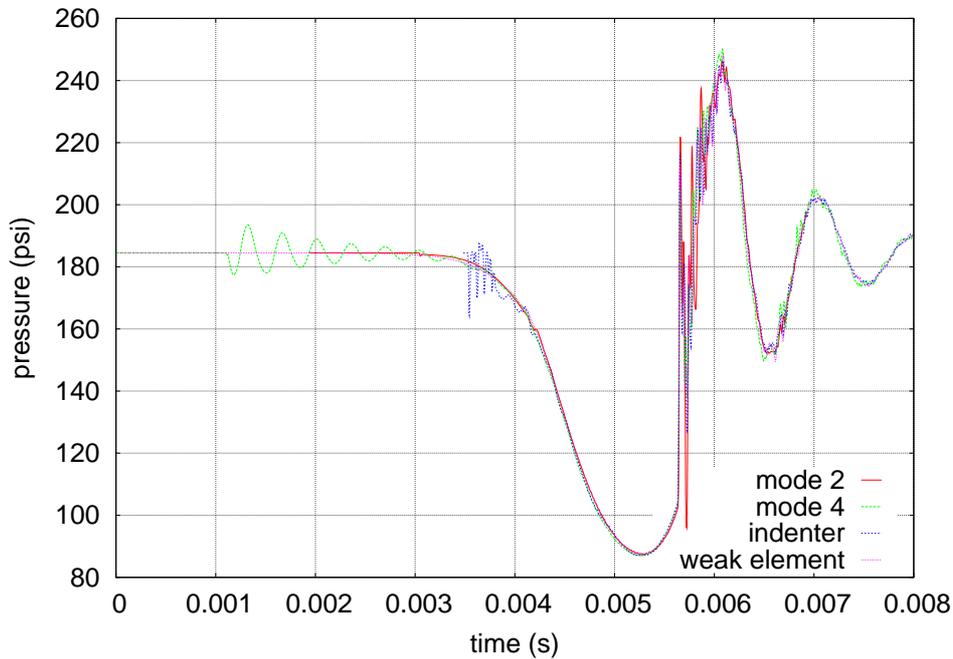


Figure 6.7: Pressure signals in the vicinity of a cylinder collapsing in mode 2 at the critical pressure of 184.5 psi. The curves correspond to different initializations of the simulation. All simulations start from a steady-state of the flow and the structure. Collapse of the cylinder is triggered by: (a) a sudden 1%-mode 2 deformation, (b) a sudden 1%-mode 4 deformation, (c) a deformation due to an indenter, (d) a sudden weakness of one of the element

pressure peaks are aligned to facilitate comparisons. Except for the early pressure oscillations, the observed signals are essentially the same. Clearly, weakening elements of a steady-state cylinder provides the smoothest transition at the beginning of the collapse, but the computational cost for achieving this smooth transition in three-dimensional simulations can become non-negligible while the overall pressure signal are all similar. Initialization procedures using a mode 4 deformation are obviously not optimal for a mode 2 collapse. The use of an indenter leads to the longest lasting oscillations and requires the specification of the speed of the indenter which can influence the pressure signal [46]. The sudden mode 2 deformation leads to large

pressure oscillations that rescind quickly. However, their amplitude can be obviated by increasing the pressure until the critical pressure instead of applying it suddenly at the initial time.

The initialization procedures proposed above all lead to the same pressure signals after a certain time and it is not obvious how to choose one of these procedures based on the quality of the signals. Instead, computational costs and ease of use suggest that a mode 2 deformation of the structure combined with an increase of the pressure until the critical pressure is reached should be considered in the simulation of collapsing cylinders.

6.2.2 Collapse of Three-Dimensional Cylinders

Two three-dimensional simulations of collapsing cylinders are considered in the remainder of this section corresponding to two different experiments where one cylinder collapsed in mode 2 while the other collapsed in mode 4.

Mode 2 Collapse

The cylinder used in the case of the mode 2 collapse has length $L = 14.44$ inches, external diameter $D = 1.4995$ inches, and thickness $t = 0.0276$ inch. It is closed by two rigid end caps preventing the cylinder to deform at its ends. Fig. 6.8 is a schematic drawing of the considered cylinder. The cylinder is maintained at the center of the tank by an arrangement of bars fixed to the ground. Pressure sensors are positioned at approximately the same distance $d = 2.55$ inches to the center of the cylinder. For this experiment, the recorded pressure time-histories reveal pressure drops of approximately 45 psi followed by an uprising of approximately 8 psi before the two collapsing walls of the cylinder come into contact. At the contact, a very sharp

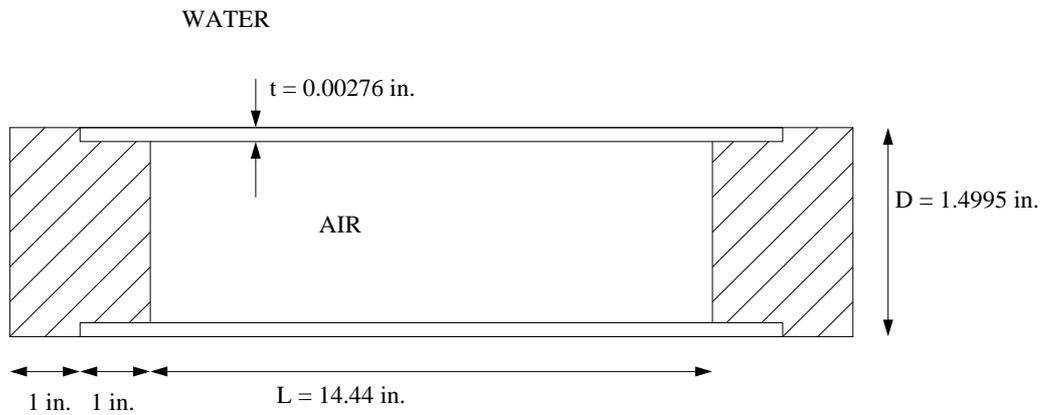


Figure 6.8: Mode 2 collapse: schematic drawing of the cylinder (striped area correspond to the end caps).

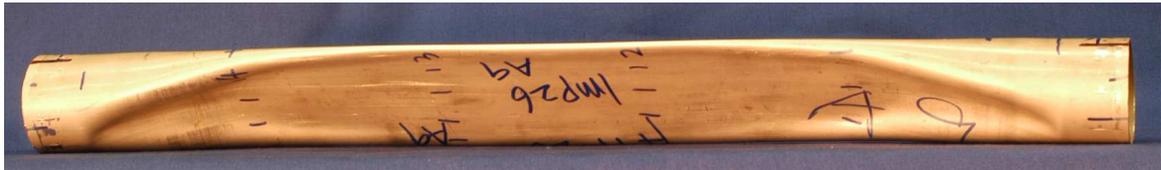


Figure 6.9: Photography of the cylinder after a mode-2 collapse. No crack is observed.

pressure rise to 254.5 psi is observed. It is followed by a second broader pressure peak with maximal value of 354.5 psi. Successive secondary peaks can then be observed and pressure signals then oscillates around the initial hydrostatic pressure. Fig. 6.9 shows that the tube flattened and did not crack.

For the simulation of this collapse, only a half cylinder in the length direction is considered and the fluid computational domain is a rectangular box that extends from the symmetry plane of the half cylinder to a distance of 16.44 inches in the direction of the cylinder, and from the symmetry axis of the cylinder to a distance of at least 20 inches in the radial direction. The structure mesh contains 12880 quad shell elements and the fluid mesh contains 2,034,067 nodes and 12,027,679 tetrahedra. Symmetry

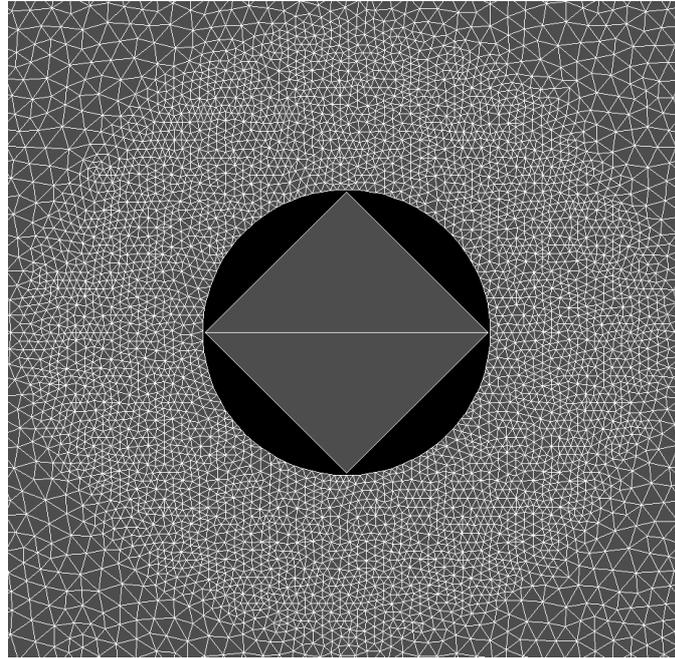


Figure 6.10: Mode 2 collapse: the coarseness of the mesh inside the cylinder is necessary to allow for mesh motion until contact

boundary conditions are applied to both the fluid and structure systems at the plane passing through the middle cross section of the cylinder. Zero-displacement and zero-velocity boundary conditions are applied at the end cap of the cylinder. Non-reflecting boundary conditions are applied at the other boundaries in the fluid computational domain. Material descriptions of aluminum, air and water have already been given above. As mentioned in the previous section, the initialization procedure consists in deforming the structure with a mode 2 deformation of amplitude 1% of the cylinder radius and in increasing the water pressure from 14.5 psi (initial pressure inside the cylinder) to the critical pressure of 184.5 psi while the air pressure inside the cylinder is maintained at 14.5 psi. During the pressure increase, only the structure is allowed to deform while both fluids remain still with densities of $9.357255 \times 10^{-5} \text{ lbs.in}^{-4} \cdot \text{s}^{-2}$

and $9.357255 \times 10^{-8} \text{ lbs.in}^{-4}.\text{s}^{-2}$ in the water and the air respectively. The fluid system is solved for only when the critical pressure is reached. A second-order space-accurate Roe scheme is used in the fluid and linear finite elements are considered in the structure. Two simulations are performed using the second-order time-accurate loosely-coupled solver of proposition 2, with two different sizes for the balls of the contact pinball algorithm. Time steps in both fluids and structure are the same and are dictated by a CFL in the structure system of 0.5. The computation are performed until reaching the final time of 2.73 ms.

Due to the compression of the volume inside the cylinder, it was necessary to use a very coarse mesh to model the air inside it in order to avoid mesh motion failures. The symmetry plane of the fluid mesh is shown in Fig.6.10. This allows to account for the compressibility of the air inside the cylinder, but it also shows the limits of mesh motion algorithms are reached in simulations characterized by contacts. Fig. 6.11 shows that the structural deformations of the cylinder at the end of the computation correspond to a mode 2 collapse. Fig. 6.12 reports the pressure time-histories

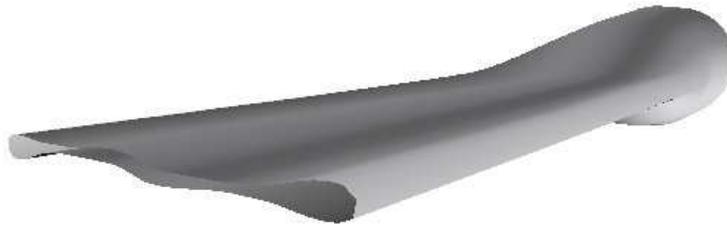


Figure 6.11: Mode 2 collapse: final deformation of the cylinder.

predicted by the simulation platform for both simulations at the position of one of the pressure sensors in comparison to the corresponding experimental pressure time-history. (Note that time was shifted such that the highest pressure peaks are aligned

for ease of comparison.) In both cases, the pressure drop is accurately described as it correlates well with the experimental data both in duration and in amplitude. In the first case, three consecutive narrow pressure peaks are observed in the simulation while a single one of higher amplitude can be observed experimentally. In the second case, the three pressure peaks are not as obvious but are present. In the first case, a broader pressure peak with maximal value of approximately 280 psi can be observed while in the second case, the corresponding pressure peak has a maximal value of 350 psi. In both cases, the peaks are well aligned with the experimental one, but only the second simulation provides an accurate prediction of the amplitude of the peak, thus providing important data on the sensitivity of the simulations with respect to the choice of parameters of the pinball contact algorithm. Successive secondary pressure peaks are not observed in the simulations.

Mode 4 Collapse

The cylinder used in the case of the mode 4 collapse has length $L = 3.0$ inches, external diameter $D = 1.4995$ inches, and thickness $t = 0.0277$ inch. It is closed by two rigid end caps preventing the cylinder to deform at its ends. Fig. 6.13 is a schematic drawing of the considered cylinder. The experimental setup of this case is the same as the one presented for the mode 2 collapse case. For this experiment, the recorded pressure time-histories (see Fig. 6.14) reveal a greater variability with respect to the location of the sensors. However, features of the pressure signals are similar with a pressure drop with minimal value of 115 psi attained around time $t = 0.38$ millisecond and a broad primary pressure whose maximal values varies between 1000 and 1150 psi and extending between times $t = 0.67$ and $t = 0.8$ millisecond. The greatest variability can be observed between times $t = 0.38$ and $t = 0.67$ millisecond,

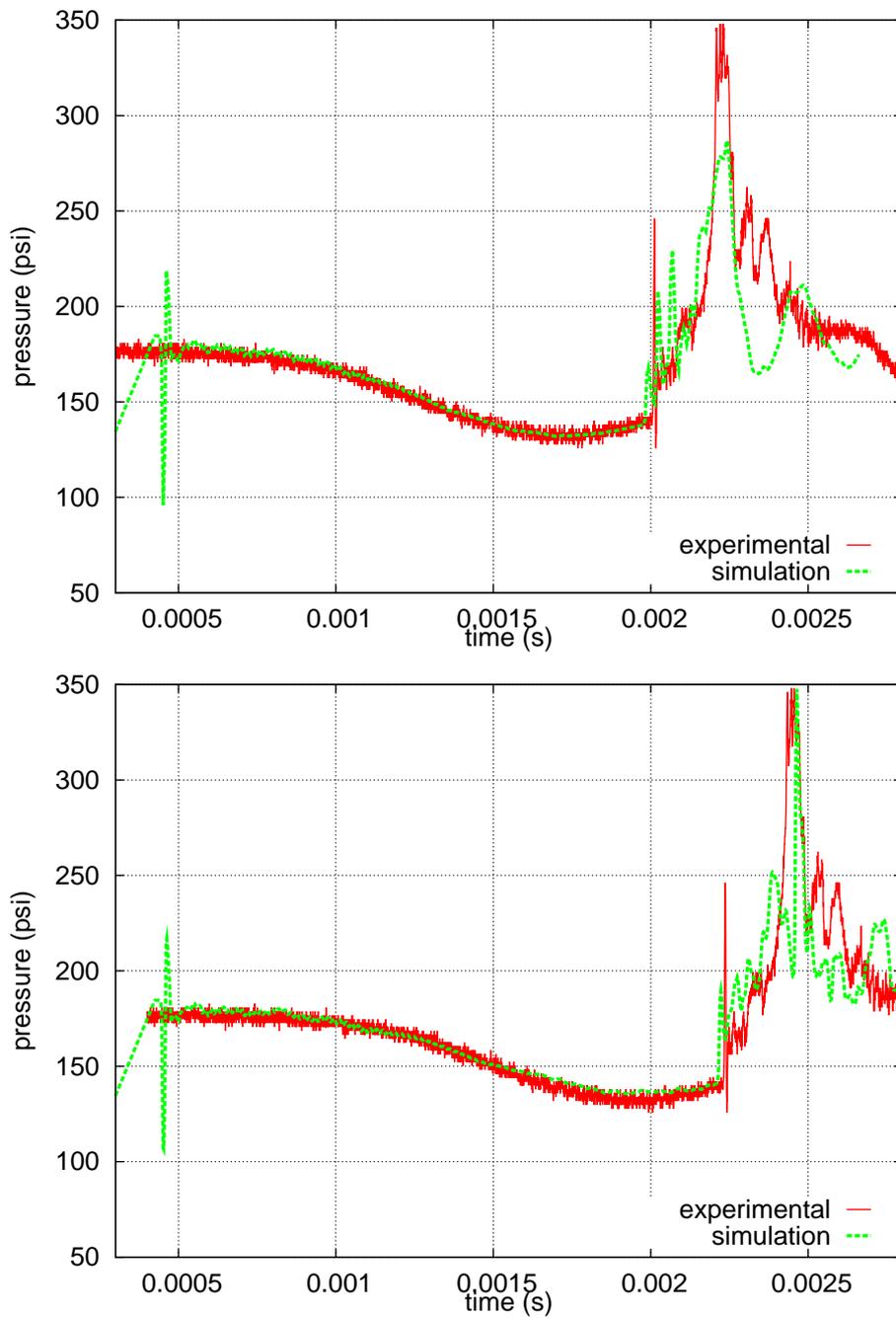


Figure 6.12: Mode 2 collapse: comparison of numerical and experimental pressure time-histories for two different setups of the contact algorithm in the structural solver.

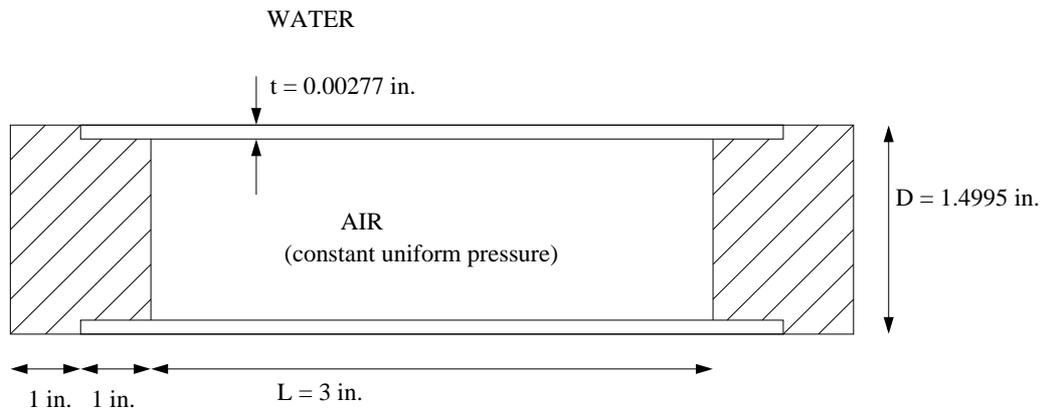


Figure 6.13: Mode 4 collapse: schematic drawing of the cylinder (striped area correspond to the end caps).

as pressure oscillations of amplitude 200 psi are reported by certain sensors and not by others. After the broad primary peak, all pressure signals oscillates around the initial hydrostatic pressure. Fig. 6.15 show that the tube collapsed in mode 4 and that cracking did occur near the end caps. The cracks observed on the cylinder vary greatly in size.

Similar to the previous simulation, only a half cylinder in the length direction is considered and the fluid computational domain is a rectangular box that extends from the symmetry plane of the half cylinder to a distance of 16.44 inches in the direction of the cylinder, and from the symmetry axis of the cylinder to a distance of at least 20 inches in radial directions. The structure mesh contains 756 quad shell elements and the fluid mesh contains 2,072,871 nodes and 12,261,808 tetrahedra. Symmetry boundary conditions are applied to both the fluid and structure systems at the plane passing through the middle cross section of the cylinder. Zero-displacement and zero-velocity boundary conditions are applied at the end cap of the cylinder. Non-reflecting boundary conditions are applied at the other boundaries in the fluid computational

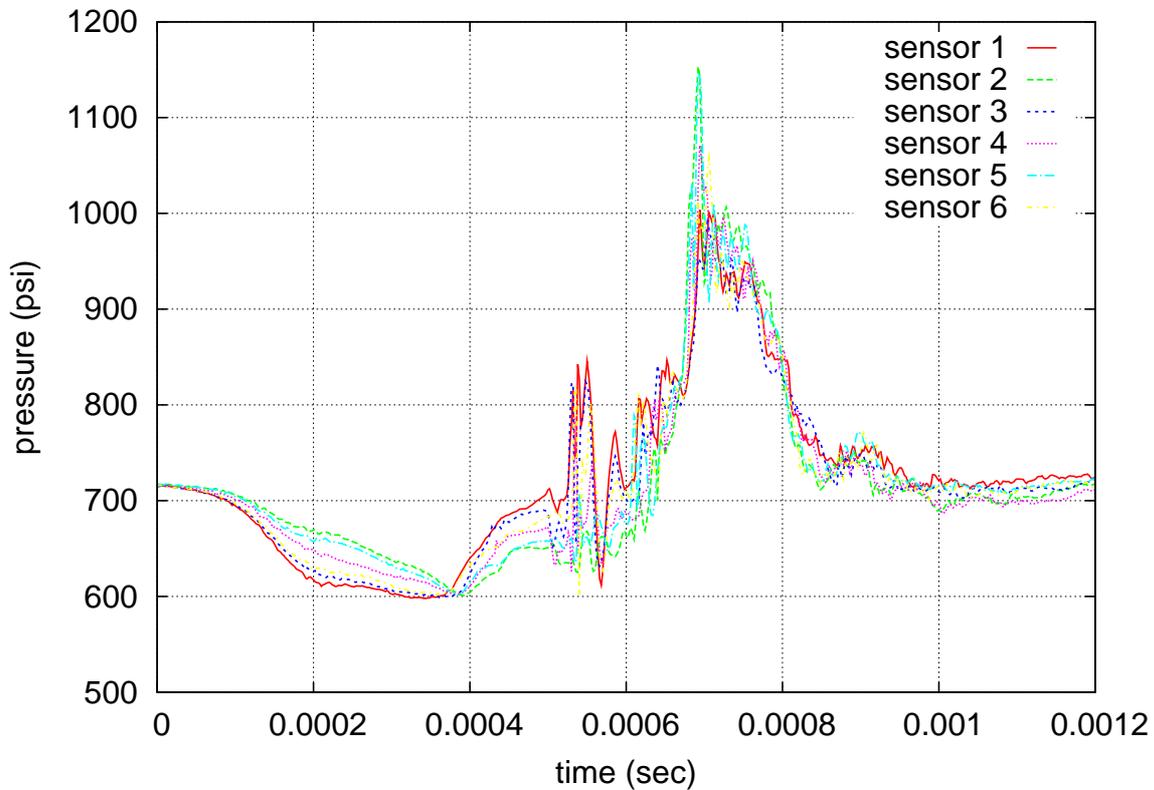


Figure 6.14: Mode 4 collapse: experimental pressure time-histories.

domain. Due to difficulties with the mesh motion, only the surrounding water is modeled, not the air inside the tube. Instead, a uniform constant pressure of 14.5 psi is applied. The same material descriptions of aluminum and water as given above are used. Note also that no crack model has been used in the present simulation. The initialization procedure is similar to the one used previously and consists in deforming the structure with a mode 4 deformation of amplitude 1% of the cylinder radius and in increasing the water pressure from 14.5 psi (pressure inside the cylinder) to the critical pressure of 729.5 psi. During the pressure increase, only the structure is allowed to deform while the fluid remains still with density of $9.357255 \times 10^{-5} \text{ lbs.in}^{-4} \cdot \text{s}^{-2}$. The fluid system is solved for only when the critical pressure is reached. A second-order



Figure 6.15: Photography of the cylinder after a mode-4 collapse. Cracks of different sizes can be observed.

space-accurate Roe scheme is used in the fluid and linear finite elements are considered in the structure. The simulation is performed using the second-order time-accurate loosely-coupled solver of proposition 2. Time steps in both fluids and structure are the same and are dictated by a CFL in the structure of 0.5. The computation is performed until reaching the final time of 1.2 ms.

It should first be recognized that the following problem reaches the limits of the present ALE computational framework, even more so than the previous simulation. A simulation with a fluid mesh inside the cylinder to model the air has not been possible, due to the complex compression of that part of the mesh as the tube collapses in mode 4. However, even without an interior fluid mesh, avoiding mesh motion issues such as the creation of negative volumes proved difficult. The mesh used for this simulation led to the deformations shown in Fig. 6.17. Mesh elements near the collapsing structure are highly stretched while other mesh elements remain mostly

unstretched. As a consequence, even though the simulation was possible, the accuracy of the results is expected to suffer.

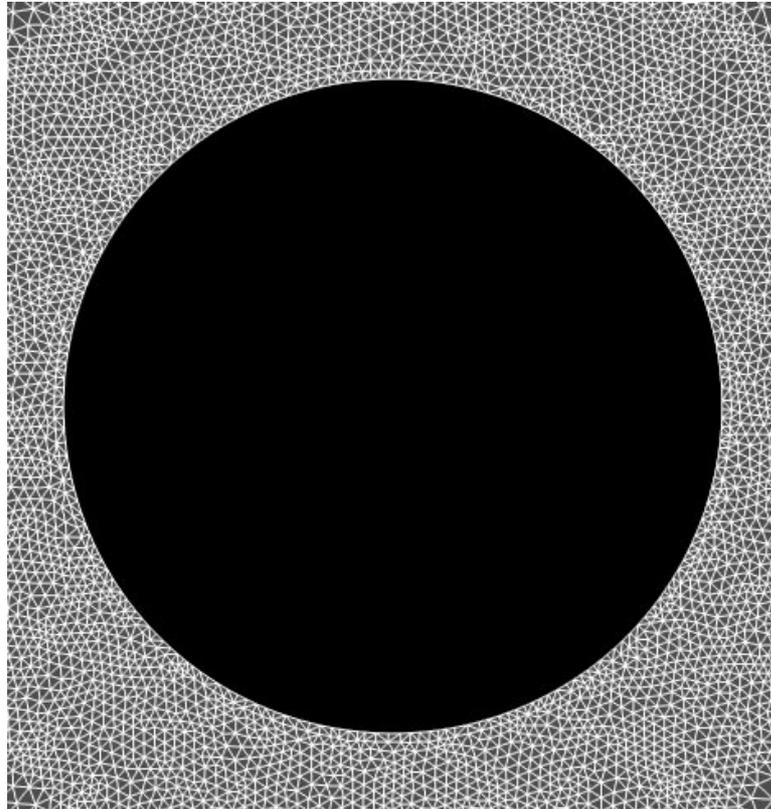


Figure 6.16: Mode 4 collapse: initial fluid mesh in the symmetry plane.

The simulation of the mode 4 collapse of the cylinder led to the pressure time-histories of various points in space depicted in Fig. 6.18. Whereas the experiment showed great variability in the recorded pressure signals, the computation led to very similar pressure signals. In order to compare the numerical and experimental pressure time-histories, two characteristic experimental ones and a single numerical one are plotted in Fig. 6.19. Time was shifted for the numerical pressure signal such that minimal pressure values coincide in the pressure drop phase. It can be observed that

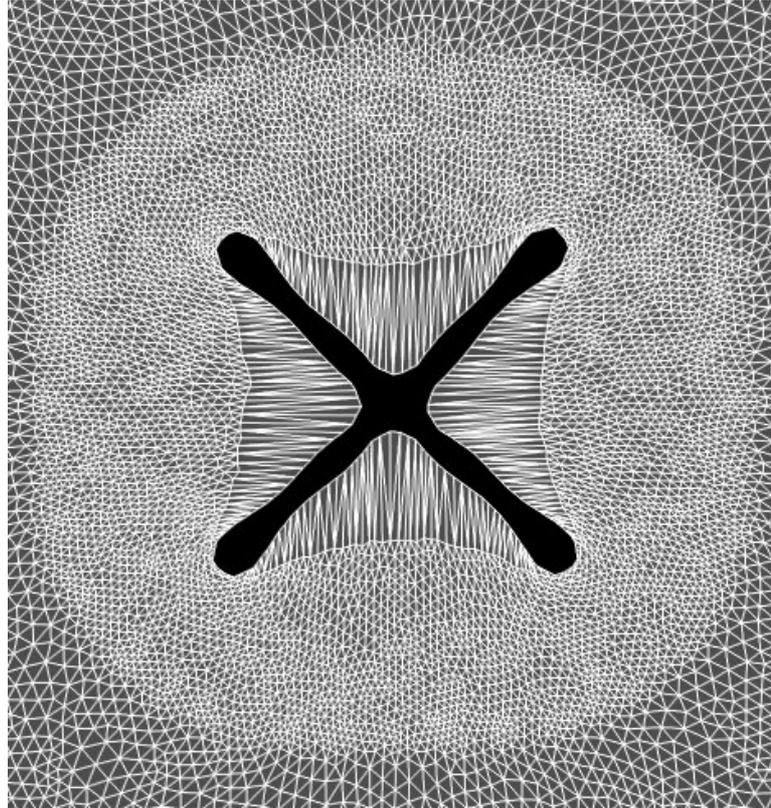


Figure 6.17: Mode 4 collapse: fluid mesh deformations at the contact in the symmetry plane.

the pressure drop given by the simulation until time $t = 0.38$ millisecond matches the one recorded experimentally by the first sensor, except for the early oscillations due to the initialization procedure. The subsequent rise of the pressure also matches experimental results until time $t = 0.44$ millisecond. For subsequent times, the pressure signals are completely different. The pressure given by the simulation rises much faster than the one given by experimental data such that the maximal pressure value is reached about 0.1 millisecond earlier. However, the maximal simulated pressure value is 1150 psi, very similar to the maximal pressure recorded by the second sensor. Eventually, all pressure signals oscillate around the initial hydrostatic pressure.

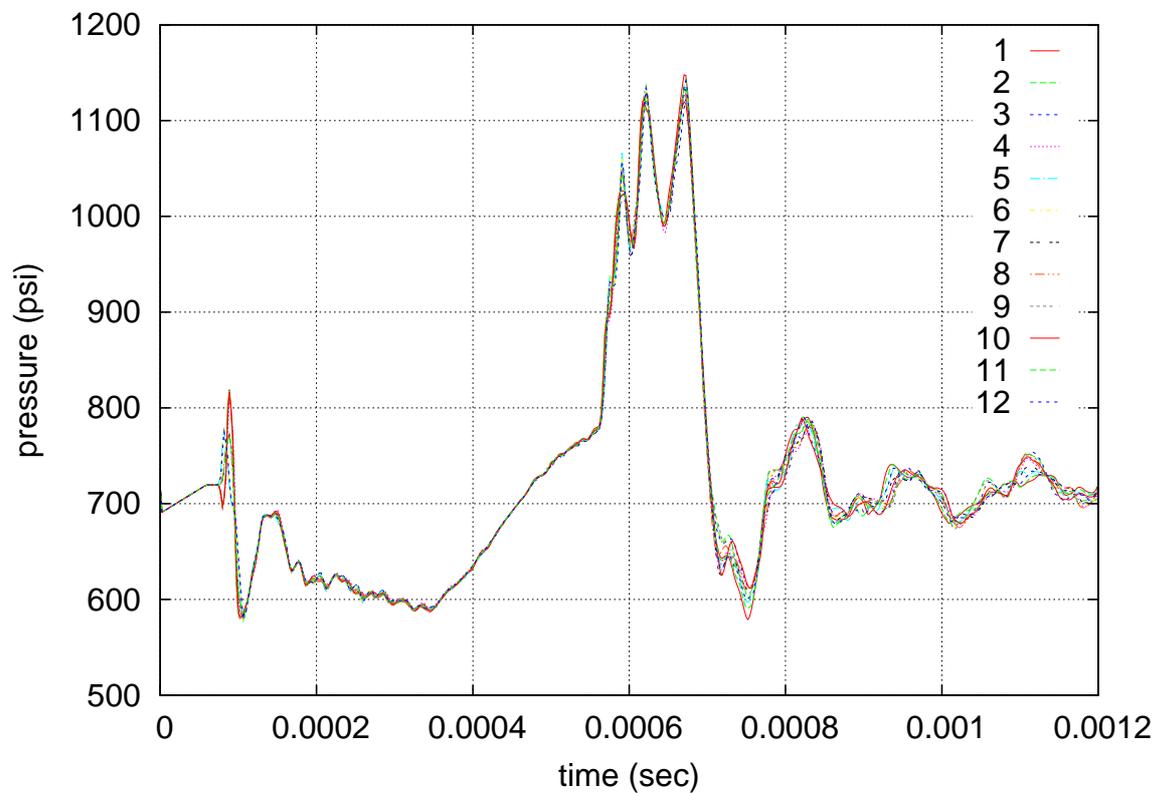


Figure 6.18: Mode 4 collapse: numerical pressure time-histories recorded at different locations around the structure including at the locations of the experimental sensors.

This particular case of collapsing cylinder proves difficult to analyze due to the large number of parameters. First, the model does not take into account that the air inside the cylinder is being compressed and that cracking can occur in the structure, allowing a subsequent mixing of air and water. The localized stretching of the mesh near the collapsing walls are expected to degrade the accuracy of the solution as gradients normal to the walls cannot be captured appropriately.

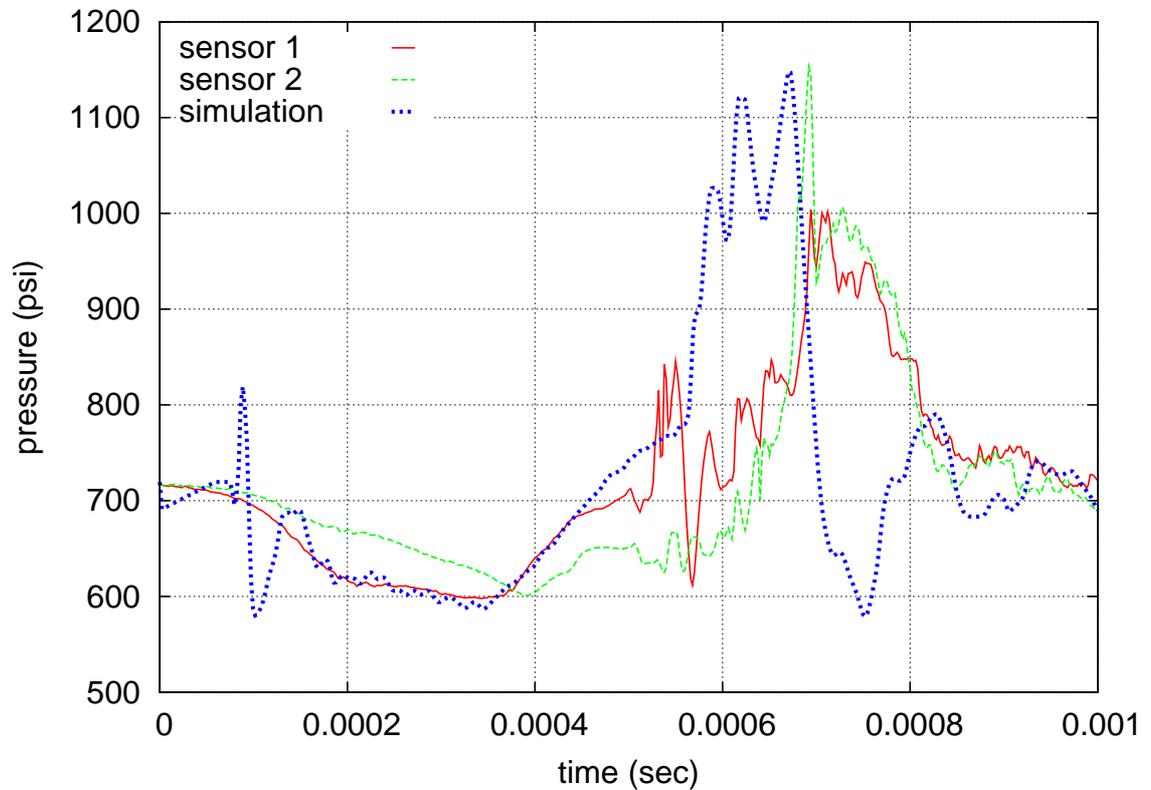


Figure 6.19: Mode 4 collapse: comparison of numerical and experimental pressure time-histories.

6.3 Summary

This chapter used the computational framework developed in the previous chapters to simulate different implosion problems. The implosion of an underwater air bubble was considered. The multiphase flow solver proved robust and accurate to recover experimental data. Most importantly, the pressure peak induced by the implosion representing the potential threat to nearby structures is reproduced accurately. The collapses of two cylinders with different characteristics were then considered. While the ALE computational framework developed in chapter 5 reproduces accurately the behavior of the fluid and the structure before contact of opposing cylinder walls, the

pressure histories obtained after the contact differ more or less from the experimental data showing the limits of the present computational framework in such particular cases. As shown with the simulations of the first collapsing cylinder, the results are fairly sensitive to the setup of the contact algorithm. In general, several possible issues need to be considered carefully. First, the mesh distortions can become too important to accurately reproduce the fluid behavior and thus, the structure behavior as well. However, the effects of the contact algorithm on the pressure histories should be considered as well for the simulation of the second collapsing cylinder. Finally, the modeling of the failure of the cylinder with the apparition of cracks may be necessary to properly capture the pressure waves in the fluids.

Chapter 7

Conclusions and Perspectives for Future Work

This chapter summarizes the work conducted in the present research and provides perspectives for future work.

7.1 Summary and Conclusions

In this dissertation, a novel compressible multiphase flow scheme was developed and the issues concerning its computational costs were addressed. The scheme was tested on academic and experimental problems. A provably second-order time-accurate staggered loosely-coupled fluid-structure procedure was developed in the Arbitrary-Lagrangian-Eulerian framework. Its time-accuracy was numerically verified. Finally, an assessment of conducting numerical simulations of imploding cylinders in the proposed computational framework was achieved.

The novel compressible multiphase flow scheme was developed for three-dimensional

unstructured meshes. It is robust, quasi-conservative and contact-preserving. A level set is used to numerically advect the material interface between any two fluids. At the interface, a locally one-dimensional two-phase Riemann problem is considered to provide adequate numerical fluxes for nodes on both sides. The algorithm was tested on academic shock-tube problems. The simulation of the collapse of an underwater air bubble and the comparison of its results with experimental data served as validation to the proposed scheme. It was also shown that the computational costs of exactly solving a two-phase Riemann problem are not always prohibitive for implosion problems. However, the use of certain equations of state as the Jones-Wilkins-Lee equation of state in the exact Riemann solver can lead to unacceptable costs. In order to alleviate them, a tabulation and interpolation procedure using sparse grids was proposed. Preliminary tests suggest that the above procedure will allow for simulations with such equations of state at an affordable computational cost.

The use of an explicit central difference scheme for the time-integration of the structure equations of motion when cracking happens has led to the necessary development of a new second-order time-accurate staggered loosely-coupled procedure. It was shown that it is possible to time-integrate the fluid equations of motion with either an explicit second-order Runge-Kutta scheme or an implicit three-point-backward-difference scheme. It was also shown that the procedure proposed in [15] can be applied with an explicit second-order Runge-Kutta scheme to time-integrate the fluid equations of motion. After verification of the order of its time-accuracy, the new procedure was applied to the simulations of the collapses of two cylinders. Given the difficulties of clearly defining an initial state for these simulations, several initialization procedures were compared. Finally, comparison of the numerical results with experimental results assessed that these implosion problems lie at the limits of the

possibilities of the proposed computational framework, in part due to the inability of the mesh motion to preserve the required accuracy to capture the important flow features. However, this computational framework is able to reproduce the behavior of the fluids and the structures in the first stages of the collapse of the cylinders and numerical experiments confirmed the need for a new computational framework for implosion problems once cracking and large deformations are present.

7.2 Perspectives for Future Work

The compressible multiphase flow scheme has shown great promises for the study of imploding bubbles. It was shown that the sparse grid-based tabulation and interpolation procedure should allow for three-dimensional simulations involving the Jones-Wilkins-Lee equation of state for gaseous products of high explosives at a reasonable computational cost. The development of an implicit scheme for compressible multiphase flows would further alleviate the overall computational costs of the present scheme for each simulation.

The simulations of collapsing cylinders have led to the conclusion that such fluid-structure interaction phenomena where contact, large deformations and crack initiation and propagation are possible, are difficult to simulate with the current computational framework as was originally expected. However, the simulations also showed that this computational framework was adequate for the simulations of the first stages of implosion problems, thus fulfilling the requirements of the MURI project. For the later stages of implosion problems, two approaches can be considered. The first would consist in improving the mesh motion algorithm to allow for better management of

the deformations encountered in imploding structures. However, this will only further push back the limits of the current framework. Therefore, a new computational framework based on purely Eulerian grids should be developed to allow arbitrarily large motions of the structure. To this end, an embedded method using pseudo-Riemann problems between a fluid and a structure is currently being developed. In addition, the incorporation of the cracking algorithm offered by the XFEM structural solver should allow for a better modeling of the interaction at the air-cylinder-water interfaces.

Appendix A

Second-Order Accuracy of the GSS Procedure

The present appendix gives the proof of lemma 3 (chapter 5) on the second-order accuracy of the Generalized Staggered Serial procedure under certain conditions. The lemma is first reminded.

Lemma 4 *If the aerodynamic force corrector is at least second-order time-accurate, that is*

$$\mathbf{f}_{ae}^{nC} = \mathbf{f}_{ae}(t^n) + \mathcal{O}(\Delta t^2) \quad (\text{A.1})$$

and if the structure predictor \mathbf{u}^P and the matrix $\bar{\mathbf{T}}$ characterizing the fluid-mesh-motion algorithm satisfy

$$\forall m, \bar{\mathbf{T}}(\mathbf{u}_\Gamma^P(t^m) - \mathbf{u}_\Gamma^P(t^{m-1})) = \int_{t^{m-1}}^{t^m} \mathbf{T}(\eta) \dot{\mathbf{u}}_\Gamma(\eta) d\eta + \mathcal{O}(\Delta t^3) \quad (\text{A.2})$$

then the GSS procedure 5.5.1 is second-order time-accurate.

The proof given in [15] is reproduced here, the only difference being that the local truncation errors are the one given in Sections 5.5.2 and 5.5.3.

Proof From Eq. 5.26 and Eq. A.1, it follows that

$$\Psi_{\dot{\mathbf{u}}}(t^{n+\frac{1}{2}}) = \mathcal{O}(\Delta t^2),$$

$$\Psi_{\mathbf{u}}(t^{n+1}) = \mathcal{O}(\Delta t^3)$$

in the case of the explicit central difference time-integrator and,

$$\Psi_{\dot{\mathbf{u}}}(t^{n+\frac{1}{2}}) = \mathcal{O}(\Delta t^3),$$

$$\Psi_{\mathbf{u}}(t^{n+1}) = \mathcal{O}(\Delta t^3)$$

in the case of the implicit midpoint rule time-integrator. This shows that the solution of the structure subsystem is second-order time-accurate.

From Eq. 5.19, it follows that

$$\mathbf{x}^P(t^{n+k}) = \mathbf{x}^P(t^{n+k-1}) + \overline{\mathbf{T}}(\mathbf{u}_\Gamma^P(t^{n+k}) - \mathbf{u}_\Gamma^P(t^{n+k-1})) \quad (\text{A.3})$$

and from Eq. 5.16, it follows that

$$\mathbf{x}(t^{n+k}) = \mathbf{x}(t^{n+k-1}) + \int_{t^{n+k-1}}^{t^{n+k}} \mathbf{T}(\eta) \dot{\mathbf{u}}_\Gamma(\eta) d\eta. \quad (\text{A.4})$$

Subtracting the latter equation to the former one gives

$$\mathbf{x}^P(t^{n+k}) - \mathbf{x}(t^{n+k}) = \mathbf{x}^P(t^{n+k-1}) - \mathbf{x}(t^{n+k-1}) \quad (\text{A.5})$$

$$+ \overline{\mathbf{T}}(\mathbf{u}_\Gamma^P(t^{n+k}) - \mathbf{u}_\Gamma^P(t^{n+k-1})) \quad (\text{A.6})$$

$$- \int_{t^{n+k-1}}^{t^{n+k}} \mathbf{T}(\eta) \dot{\mathbf{u}}_\Gamma(\eta) d\eta. \quad (\text{A.7})$$

By recursion, and noting that $\mathbf{x}^P(0) = \mathbf{x}(0)$, the above result is transformed into

$$\mathbf{x}^P(t^{n+k}) - \mathbf{x}(t^{n+k}) = \sum_{m=1}^{n+k} \left(+\overline{\mathbf{T}}(\mathbf{u}_\Gamma^P(t^m) - \mathbf{u}_\Gamma^P(t^{m-1})) - \int_{t^{m-1}}^{t^m} \mathbf{T}(\eta) \dot{\mathbf{u}}_\Gamma(\eta) d\eta \right). \quad (\text{A.8})$$

Hence, if the condition (A.2) is satisfied, then

$$\mathbf{x}^P(t^{n+k}) - \mathbf{x}(t^{n+k}) = \sum_{m=1}^{n+k} \mathcal{O}(\Delta t^3) = (n+k)\mathcal{O}(\Delta t^3) = \frac{t^{n+k}}{\Delta t} \mathcal{O}(\Delta t^3) = \mathcal{O}(\Delta t^2) \quad (\text{A.9})$$

From Lemma 1 and the above estimation, it follows that

$$\Psi_{\mathbf{U}}(t^{n+1}) = \mathcal{O}(\Delta t^3)$$

which shows that the solution of the fluid subsystem is also second-order time-accurate.

Appendix B

One-Dimensional Two-Phase Riemann Problem

At each time-step, the one-dimensional two-phase Riemann problem (4.1)-(4.2) is constructed along each edge i - j that crosses the material interface which is designated here by the subscript I . Two different algorithms are presented in this section to solve for the two constant interfacial states of the problem. The first algorithm reduces the problem to an explicit expression of the normal velocity at the material interface, u_I , as a function of the pressure at this material interface, p_I , and a non-linear equation in p_I . In the present work, two-phase Riemann problem where each fluid is either modeled by a perfect gas, a stiffened gas or a barotropic liquid with Tait EOS. The second algorithm does not reduce the problem to a non-linear equation. Instead a two-by-two system which expresses the equality of pressure p_I and velocity u_I at the interface must be solved. This algorithm is used, whenever one of the two fluids is modeled by a JWL EOS.

B.1 Non-Linear Equation Form

For example, when both media on the left and right sides of the material interface are modeled as stiffened gases, the local Riemann problem can be written as

$$\begin{aligned}
 u_I &= \frac{1}{2}(u_L + u_R) \\
 &+ \frac{1}{2}(\mathcal{R}_R^{SG}(p_I; p_R, \rho_R) - \mathcal{R}_L^{SG}(p_I; p_L, \rho_L)) \\
 \mathcal{R}(p_I; u_L, p_L, \rho_L, u_R, p_R, \rho_R) &= \mathcal{R}_L^{SG}(p_I; p_L, \rho_L) + \mathcal{R}_R^{SG}(p_I; p_R, \rho_R) \\
 &+ u_R - u_L = 0
 \end{aligned} \tag{B.1}$$

where the subscripts L and R designate the left and right sides of the material interface, respectively, \mathcal{R}_L^{SG} and \mathcal{R}_R^{SG} are two vector functions that depend on the structure of the wave solution at the left and right sides of the contact discontinuity (see Fig. 2.2), and a ";" is used to separate the unknown variables from known quantities.

When both media on the left and right sides of the material interface are modeled by Tait's EOS, the Riemann problem can be written as

$$\begin{aligned}
 u_I &= \frac{1}{2}(u_L + u_R) \\
 &+ \frac{1}{2}(\mathcal{R}_R^{Tait}(p_I; \rho_R) - \mathcal{R}_L^{Tait}(p_I; \rho_L)) \\
 \mathcal{R}(p_I; u_L, \rho_L, u_R, \rho_R) &= \mathcal{R}_L^{Tait}(p_I; \rho_L) + \mathcal{R}_R^{Tait}(p_I; \rho_R) \\
 &+ u_R - u_L = 0
 \end{aligned} \tag{B.2}$$

where \mathcal{R}_L^{Tait} and \mathcal{R}_R^{Tait} are two vector functions that depend on the structure of the wave solution at the left and right sides of the contact discontinuity (see Fig. 2.2). Analytical expressions for \mathcal{R}_L^{SG} , \mathcal{R}_R^{SG} , \mathcal{R}_L^{Tait} , and \mathcal{R}_R^* can be obtained from the Rankine-Hugoniot jump conditions for shocks and the isentropic relations for rarefactions [56].

For the sake of completeness, these are given in Section B.1.1 — Section B.1.4 of this appendix. Once Eq. (B.1) is solved for p_I — for example, using Newton's method — the computation of other interface quantities such as u_I , ρ_{I_L} and ρ_{I_R} becomes straightforward.

B.1.1 Shock Wave Relations for a Stiffened Gas

The shock wave relations for a stiffened gas can be written in terms of the unknown value of the pressure at the material interface, p_I , as follows

$$\mathcal{R}_K^{SG}(p_I; p_K, \rho_K) = \left(\sqrt{\frac{a_K}{\bar{p}_I + b_K}} \right) (p_I - p_K) \quad (\text{B.3})$$

where the subscript K designates either the medium at the left (L) or that at the right (R) of the material interface,

$$a_K = \frac{2}{(\gamma_K + 1)\rho_K} \quad b_K = \left(\frac{\gamma_K - 1}{\gamma_K + 1} \right) \bar{p}_K \quad \bar{p}_K = p_K + p_{sg,K} \quad (\text{B.4})$$

and γ_K and $p_{sg,K}$ have been defined in chapter 2 and correspond to the EOS on the K side of the interface as indicated by the subscript.

The pressure derivative of \mathcal{R}_K^{SG} at the material interface is given by

$$\mathcal{R}_K^{SG'}(p_I; p_K, \rho_K) = \frac{d\mathcal{R}_K^{SG}}{dp_I} = -\frac{a_K}{2(\bar{p}_I + b_K)^2} \quad (\text{B.5})$$

B.1.2 Expansion Wave Relations for a Stiffened Gas

The expansion wave relations for a stiffened gas are given by

$$\mathcal{R}_K^{SG}(p_I; p_K, \rho_K) = \left(\frac{2c_K}{\gamma_K - 1} \right) \left(\left(\frac{\bar{p}_I}{\bar{p}_K} \right)^{\frac{\gamma_K - 1}{2\gamma_K}} - 1 \right) \quad (\text{B.6})$$

and

$$\mathcal{R}_K^{SG'}(p_I; p_K, \rho_K) = \left(\frac{c_K}{\gamma_K \bar{p}_K^{\frac{\gamma_K - 1}{2\gamma_K}}} \right) \left(\frac{\bar{p}_I}{\bar{p}_K} \right)^{-\left(\frac{\gamma_K + 1}{2\gamma_K}\right)} \quad (\text{B.7})$$

where c_K denotes as before the speed of sound (see Eq. (2.15)) on the side K of the interface.

B.1.3 Shock Wave Relations for Tait's EOS

For Tait's EOS, the shock wave relations can be written as

$$\mathcal{R}_K^{Tait}(p_I; \rho_K) = \sqrt{\frac{(p_I - p_K)(\rho_I - \rho_K)}{\rho_K \rho_I}} \quad (\text{B.8})$$

and

$$\begin{aligned} \mathcal{R}_K^{Tait'}(p_I; \rho_K) &= \frac{d\mathcal{R}_K^{Tait}}{dp_I} \\ &= \left(\frac{1}{2\mathcal{R}_K^{Tait}} \right) \left(\frac{\rho_I(\rho_I - \rho_K) + (p_I - p_K)\rho_K \rho_I'}{\rho_K \rho_I^2} \right) \end{aligned} \quad (\text{B.9})$$

where

$$\rho_I = \left(\frac{p_I - \eta_K}{\alpha_K} \right)^{\beta_K - 1} \quad \rho_I' = \frac{\rho_I^{1 - \beta_K}}{\alpha_K \beta_K} \quad p_K = \eta_K + \alpha_K \rho_K^{\beta_K} \quad (\text{B.10})$$

α_K and β_K and η_K have been defined in chapter 2 and correspond to the EOS on the side K of the interface.

B.1.4 Expansion Wave Relations for Tait's EOS

On the other hand, the expansion wave relations governing a medium modeled by Tait's EOS are given by

$$\mathcal{R}_K^{Tait}(p_I; \rho_K) = \left(\frac{2c_K}{\beta_K - 1} \right) \left(\left(\frac{\bar{p}_I}{\bar{p}_K} \right)^{\frac{\beta_K - 1}{2\beta_K}} - 1 \right) \quad (\text{B.11})$$

and

$$\mathcal{R}_K^{Tait'}(p_I; \rho_K) = \frac{c_K}{\beta_K \bar{p}_K} \left(\frac{\bar{p}_I}{\bar{p}_K} \right)^{-\frac{\beta_K + 1}{2\beta_K}} \quad (\text{B.12})$$

B.1.5 Local Solution by Newton's Method

The application of Newton's method to the solution of the local non-linear equation (B.1) for the interface pressure, p_I , by Newton's method generates the following sequence of iterate values of p_I

$$p_I^{(m+1)} = p_I^{(m)} - \frac{\mathcal{R}(p_I^{(m)})}{\mathcal{R}'(p_I^{(m)})} \quad (\text{B.13})$$

where m designates the Newton iteration. In this work, convergence of the sequence (B.13) is declared when

$$\frac{2|p_I^{(m+1)} - p_I^{(m)}|}{p_I^{(m+1)} + p_I^{(m)}} < \epsilon \quad (\text{B.14})$$

where ϵ is a specified tolerance.

B.2 System Form

The methodology developed in [50] is applied here when one of the two fluids is modeled by a JWL equation of state. As mentioned in chapter 4, the two-phase Riemann problem can be formulated as follows

$$\begin{cases} u_{IL}(v_{IL}; \mathbf{U}_L, EOS_L) = u_{IR}(v_{IR}; \mathbf{U}_R, EOS_R) \\ p_{IL}(v_{IL}; \mathbf{U}_L, EOS_L) = p_{IR}(v_{IR}; \mathbf{U}_R, EOS_R) \end{cases}$$

where v is the inverse of the density. In the next sections, the formulae to express the Rankine-Hugoniot conditions and the conditions given by the Riemann invariants are given for a general equation of state of the form

$$p = \omega\rho\epsilon + f(\rho)$$

Thus, the u_{IL} , u_{IR} , p_{IL} , p_{IR} are applicable for the stiffened gas EOS and the JWL EOS.

B.2.1 Shockwave Relation

$$\begin{aligned} p_{IK} &= \frac{\frac{\omega+1}{\omega}v_K p_K + \frac{1}{\omega} \left(v f\left(\frac{1}{v}\right) - v_K f\left(\frac{1}{v_K}\right) \right) - \frac{1}{2}(v_K + v)p_K}{\frac{\omega+1}{\omega}v - \frac{1}{2}(v_K + v)} \\ u_{IK} &= u_K \mp \sqrt{(p_K - p_{IK})(v - v_K)} \end{aligned}$$

$$\begin{aligned}\frac{dp_{IK}}{dv} &= \frac{\frac{1}{\omega} \left(f\left(\frac{1}{v}\right) - \frac{1}{v} f'\left(\frac{1}{v}\right) \right) - \frac{p_K}{2} - \left(\frac{\omega+1}{\omega} - \frac{1}{2} \right) p_{IK}}{\frac{\omega+1}{\omega} v - \frac{1}{2}(v_K + v)} \\ \frac{du_{IK}}{dv} &= \pm \frac{(v - v_K) \frac{dp_{IK}}{dv} + p_{IK} - p_K}{2\sqrt{(p_K - p_{IK})(v - v_K)}}\end{aligned}$$

B.2.2 Expansion Wave Relation

$$\begin{aligned}p_{IK} &= p_K - \int_{v_K}^v \frac{c^2(s_K, v')}{v'^2} dv' \\ u_{IK} &= u_K \mp \int_{v_K}^v \frac{c(s_K, v')}{v'} dv'\end{aligned}$$

$$\begin{aligned}\frac{dp_{IK}}{dv} &= \frac{c^2(s_K, v)}{v^2} \\ \frac{du_{IK}}{dv} &= \mp \frac{c(s_K, v)}{v}\end{aligned}$$

B.2.3 Local Solution by Newton's Method

In order to apply Newton's method, the system is cast under the form

$$F(v_{IL}, v_{IR}) = F_{IL}(v_{IL}; \mathbf{U}_L, EOS_L) - F_{IR}(v_{IR}; \mathbf{U}_R, EOS_R).$$

The following sequence of iterative values of (v_{IR}, v_{IL}) is generated

$$\begin{pmatrix} \Delta v_{IL}^{(k+1)} \\ \Delta v_{IR}^{(k+1)} \end{pmatrix} = -J^{-1} \left(v_{IL}^{(k)}, v_{IR}^{(k)} \right) F \left(v_{IL}^{(k)}, v_{IR}^{(k)} \right)$$

where $J(v_{IL}, v_{IR})$ is the jacobian of $F(v_{IL}, v_{IR})$. where k designates the Newton iteration. In this work, convergence of the sequence (B.15) is declared when both conditions

$$\frac{2|v_{IL}^{(k+1)} - v_{IL}^{(k)}|}{v_{IL}^{(k+1)} + v_{IL}^{(k)}} < \epsilon \quad (\text{B.15})$$

$$\frac{2|v_{IR}^{(k+1)} - v_{IR}^{(k)}|}{v_{IR}^{(k+1)} + v_{IR}^{(k)}} < \epsilon \quad (\text{B.16})$$

are satisfied and where ϵ is a specified tolerance. At the end of each Newton iteration, it is necessary to check that the densities are positive. Thus, if the inverse of the densities

$$v_{IL}^{(k)} + \Delta v_{IL}^{(k+1)} \quad \text{or} \quad v_{IR}^{(k)} + \Delta v_{IR}^{(k+1)}$$

were to be negative, their values are replaced by

$$v_{IL}^{(k+1)} = \frac{v_{IL}^{(k)}}{2} \quad \text{and} \quad v_{IR}^{(k+1)} = \frac{v_{IR}^{(k)}}{2}.$$

It was also found that in the case of a Riemann problem involving a stiffened gas and a JWL fluid, it was found that relaxation was necessary to ensure convergence of the Riemann problem

$$v_{IK}^{(k+1)} = v_{IK}^{(k)} + \frac{\Delta v_{IK}^{(k+1)}}{\alpha}$$

with $\alpha > 1$. Typically, a value of 5 works in all numerical cases that have been tested and that are of interest to the present problem. This convergence problem does not appear when two different JWL fluids with somewhat similar coefficient values are considered.

Bibliography

- [1] Harten A. and Hyman J.M. Self-adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of Computational Physics*, 50:235–269, 1983.
- [2] Haselbacher A. and Blazek J. Accurate and efficient discretization of Navier-Stokes equations on mixed grids. *AIAA journal*, 38:2094–2102, 2000.
- [3] Klimke A. and Wohlmuth B. Spinter:piecewise multilinear hierarchical sparse grid interpolation in Matlab. *ACM Transactions on Mathematical Software*, 31:561–579, 2005.
- [4] Murrone A. *Modèles bi-fluides à six et sept équations pour les écoulements diphasiques à faible nombre de Mach*. PhD thesis, Université de Provence, Aix-Marseille, France, 2003.
- [5] Wardlaw A.B.Jr and Luton J.A. Fluid-structure interaction mechanisms for close-in explosions. *Shock & Vibration*, 7:265–275, 2000.
- [6] Koobus B. and Farhat C. Second-order time-accurate and geometrically conservative implicit schemes for flow computations on unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering*, 170:103–129, 1999.

- [7] Dobratz B.M. and Crawford P.C. LLNL explosives handbook. properties of chemical explosives and explosive simulants. Technical Report A572272, Lawrence Livermore National Laboratories, 1985.
- [8] Bernardi C, Maday Y., and Patera Y.T. A new nonconforming approach to domain decomposition: the mortar element method. Paris, France, 1990. College de France Seminar.
- [9] Degand C. and Farhat C. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers and Structures*, 80:305–316, 2002.
- [10] Farhat C., Degand C., Koobus B., and Lesoinne M. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163:231–245, 1998.
- [11] Farhat C., Lesoinne M., and Le Tallec P. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157:95–114, 1998.
- [12] Farhat C. and Geuzaine P. Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids. *Computer Methods in Applied Mechanics and Engineering*, 193:4073–4095, 2004.
- [13] Farhat C., Geuzaine P., and Grandmont C. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *Journal of Computational Physics*, 174:669–694, 2001.

- [14] Farhat C., Geuzaine P., and Brown G. Application of a three-field nonlinear fluid-structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter. *Computers and Fluids*, 32:3–29, 2003.
- [15] Farhat C., van der Zee K. G., and Geuzaine P. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195:1973–2001, 2006.
- [16] Viozat C. Implicit upwind schemes for low mach number compressible flows. Technical Report 3094, INRIA, 1997.
- [17] Viozat C. and Guillard H. On the behaviour of upwind schemes in the low mach number limit. Technical Report 3160, INRIA, 1997.
- [18] Chen C.-Y., Shiuan J.-H., and Lan I-F. The equation of state of detonation products obtained from cylinder expansion test. *Propellants, Explosives, Pyrotechnics*, 19:9–14, 1994.
- [19] Bottasso C.L., Detomi D., and Serra R. The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering*, 194:4244–4264, 2005.
- [20] Hirt C.W. and Nichols B.D. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.
- [21] Nguyen D., Gibou D., and Fedkiw R. A fully conservative ghost fluid method & stiff detonation waves. In *12th Int. Detonation Symposium*, San Diego , California, USA, 2002. ONR.

- [22] Darmofal D.L. and Siu K. A robust multigrid algorithm for the Euler equations with local preconditioning and semi-coarsening. *Journal of Computational Physics*, 151:728–756, 1999.
- [23] Godlewski E. and Raviart P.-A. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer, Berlin, Germany, 1995.
- [24] Johnsen E. *Numerical simulations of non-spherical bubble collapse*. PhD thesis, California Institute of Technology, Pasadena, California, USA, 2007.
- [25] Johnsen E. and Colonius T. Implementation of WENO schemes in compressible multicomponent flow problems. *Journal of Computational Physics*, 219:715–732, 2006.
- [26] Sinibaldi E., Beux F., and Salvetti M.V. A preconditioned implicit Roe’s scheme for barotropic flows: Towards simulation of cavitation phenomena. Technical Report 4891, INRIA, 2003.
- [27] Turkel E. Preconditioned methods for solving the incompressible and low-speed compressible equations. *Journal of Computational Physics*, 72:277–298, 1987.
- [28] Turkel E. Preconditioning techniques in computational fluid dynamics. *Annual Reviews of Fluid Mechanics*, 31:385–416, 1999.
- [29] Toro E.F. *Riemann solvers and numerical methods for fluid dynamics - A practical introduction*. Springer-Verlag, Berlin, Germany, 1997.
- [30] Mut F., Buscaglia G.C., and Dari E.A. New mass-conserving algorithm for level set redistancing on unstructured meshes. *Journal of Applied Mechanics*, 73:1011–1016, 2006.

- [31] Harlow F.H. and Amsden A.A. Fluid dynamics. Technical Report 4700, Los Alamos National Laboratories, 1971.
- [32] Zi G. and Belytschko T. New crack-tip elements for XFEM and applications to cohesive cracks. *International Journal for Numerical Methods in Engineering*, 57:2221–2240, 2003.
- [33] Hulbert G.M. and Chung J. Explicit time integration algorithms for structural dynamics with optimal numerical dissipation. *Computer Methods in Applied Mechanics and Engineering*, 137:175–188, 1996.
- [34] Bethe H. The theory of shock waves for an arbitrary equation of state. Technical Report PB-32189, Clearinghouse for Federal Scientific and Technical Information, U.S. Department of Commerce, 1942.
- [35] Hamashima H., Kato I., and Itoh S. Determination of JWL parameters for non-ideal explosive. In *Shock Compression of Condensed Matter*, pages 331–334. American Physical Society, 2004. Topical Group on Shock Compression of Condensed Matter.
- [36] Weyl H. Shock waves in arbitrary fluids. *Communications on Pure and Applied Mathematics*, 2:103, 1949.
- [37] Bungartz H.J. and Griebel M. Sparse grids. *Acta Numerica*, 13:147–269, 2004.
- [38] Hilber H.M. and Hughes T.J.R. Collocation, dissipation and 'overshoot' for time integration algorithms in structural dynamics. *Earthquake Engineering and Structural Dynamics*, 6:99–118, 1978.

- [39] Donea J., Giuliani S., and Halleux J.P. An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33:689–723, 1982.
- [40] Glimm J., Marchesin D., and McBryan O. A numerical method for 2-phase flow with an unstable interface. *Journal of Computational Physics*, 39:179–200, 1981.
- [41] Lubliner J. *Plasticity Theory*. MacMillan, New York, New York, USA, 1990.
- [42] Ghidaglia J.-M. and Pascal F. The normal flux boundary method at the boundary for multidimensional finite volume approximations in CFD. *European Journal of Mechanics - B/Fluids*, 24:1–17, 2005.
- [43] Cocchi J.-P. and Saurel R. A Riemann problem based method for the resolution of compressible multimaterial flows. *Journal of Computational Physics*, 137:265–298, 1997.
- [44] Sethian J.A. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge, United Kingdom, 1999.
- [45] Ambrico J.M. Computational modeling of the implosion of metal cylinders with a focus on structural behavior. Technical Report 06-020, NUWC, 2003.
- [46] Ambrico J.M. Personal communication. 2007.
- [47] Wardlaw A.B. Jr, McKeown R., and Chen H. Implementation and application of the P-a equation of state in the dysmas code. Technical Report 95/107, NSWCCD, 1996.
- [48] Batina J.T. Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA journal*, 28:1381–1388, 1990.

- [49] Fezoui L. and Larrouturou B. *Nonlinear Hyperbolic Problems*, volume 1402 of *Lecture Notes in Mathematics*, chapter On the equations of multi-component perfect or real gas inviscid flow, pages 69–98. Springer-Verlag, Berlin, Germany, 1989.
- [50] Quartapelle L., Castelletti L., Guardone A., and Quaranta G. Solution of the Riemann problem of classical gasdynamics. *Journal of Computational Physics*, 190:118–140, 2003.
- [51] Roe P. L. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.
- [52] Landau L.D. and Lifshitz E.M. *Fluid Mechanics*. Pergamon Press, Oxford, England, 1959.
- [53] Lesoinne M. and Farhat C. Stability analysis of dynamic meshes for transient aeroelastic computations. Orlando, Florida, July 1993. 11th AIAA Computational Fluid Dynamics Conference, AIAA. Paper 93-3325.
- [54] Lesoinne M. and Farhat C. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer Methods in Applied Mechanics and Engineering*, 134:71–90, 1996.
- [55] Sussman M., Smereka P., and Osher S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.

- [56] Ivings M.J., Causon D.M., and Toro E.F. On Riemann solvers for compressible liquids. *International Journal for Numerical Methods in Fluids*, 28:395–418, 1998.
- [57] Maman N. and Farhat C. Matching fluid and structure meshes for aeroelastic computations: a parallel approach. *Computers & Structures*, 54:779–785, 1995.
- [58] Moiseev N.Ya. and Mukhamadieva T.A. Newton’s method as applied to the Riemann problem for media with general equations of state. *Computational Mathematics and Mathematical Physics*, 48:1039–1047, 2008.
- [59] Lindau O. and Lauterborn W. Cinematographic observation of the collapse and rebound of a laser-produced cavitation bubble near a wall. *Journal of Fluid Mechanics*, 479:327–348, 2003.
- [60] Colella P. and Glaz H.M. Efficient solution algorithms for the Riemann problem for real gases. *Journal of Computational Physics*, 59:264–289, 1985.
- [61] Geuzaine P., Grandmont C., and Farhat C. Design and analysis of ALE schemes with provable second-order time-accuracy for inviscid and viscous flow simulations. *Journal of Computational Physics*, 191:206–227, 2003.
- [62] Geuzaine P., Brown G., Harris C., and Farhat C. Aeroelastic dynamic analysis of a full F-16 configuration for various flight conditions. *AIAA journal*, 41:363–371, 2003.
- [63] Abgrall R. Generalisation of the roe scheme for the computation of mixture of perfect gases. *La Recherche Aérospatiale*, 6:31–43, 1988.

- [64] Abgrall R. How to prevent pressure oscillations in multicomponent flow calculations: A quasi conservative approach. *Journal of Computational Physics*, 125:150–160, 1996.
- [65] Abgrall R. and Karni S. Computations of compressible multifluids. *Journal of Computational Physics*, 169:594–623, 2001.
- [66] Fedkiw R., Marquina A., and Merriman B. An isobaric fix for the overheating problem in multimaterial compressible flows. *Journal of Computational Physics*, 148:545–578, 1999.
- [67] Menikoff R. and Plohr B.J. The Riemann problem for fluid flow of real materials. *Reviews of Modern Physics*, 61:75–130, 1989.
- [68] Saurel R., Larini M., and Loraud J.-C. Exact and approximate Riemann solvers for real gases. *Journal of Computational Physics*, 112:126–137, 1994.
- [69] Saurel R. and Abgrall R. A simple method for compressible multifluid flows. *SIAM Journal on Scientific Computing*, 21:1115–1145, 1999.
- [70] Whaley R. and Wilkins J.R. III. U.S. Navy general specification for the design, construction, and repairing of diving and hyperbaric equipment. Technical Report TS500-AU-SPN-010, NAVSEA, 2006.
- [71] Cole R.H. *Underwater Explosions*. Princeton University Press, Princeton, New Jersey, USA, 1948.
- [72] Leveque R.J. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, Cambridge, United Kingdom, 2002.

- [73] Fedkiw R.P., Aslam T., Merriman B., and Osher S. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics*, 152:457–492, 1999.
- [74] Smith R.W. AUSM(ALE): A geometrically conservative arbitrary Lagrangian-Eulerian flux splitting scheme. *Journal of Computational Physics*, 150:268–286, 1999.
- [75] Brogniez S., Rajasekharan A., and Farhat C. Provably stable and time accurate extension of explicit Runge-Kutta schemes for CFD computations on moving grids. *in preparation*, 2009.
- [76] Gottlieb S., Shu C.-W., and Tadmor E. Strong stability-preserving high-order time discretization methods. *SIAM review*, 43:89–112, 2001.
- [77] Karni S. Multicomponent flow calculations by a consistent primitive algorithm. *Journal of Computational Physics*, 112:31–43, 1994.
- [78] Osher S. and Fedkiw R. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, Berlin, Germany, 2003.
- [79] Piperno S., Farhat C., and Larrouturou B. Partitioned procedures for the transient solution of coupled aeroelastic problems. *Computer Methods in Applied Mechanics and Engineering*, 124:11–79, 1995.
- [80] Timoshenko S. *Strength of Materials*. R.E. Krieger, Huntington, New York, USA, 1976.

- [81] Morton S.A., Melville R.B., and Visbal M.R. Accuracy and coupling issues of aeroelastic Navier-Stokes solutions of deforming meshes. *Journal of Aircraft*, 35:798–805, 1998.
- [82] Turner S.E. Underwater implosion of glass spheres. *Journal of the Acoustical Society of America*, 121:844–852, 2007.
- [83] Belytschko T. *Computational Methods for Transient Analysis*, chapter An Overview of Semidiscretization and Time Integration Procedures, pages 1–66. North-Holland, Netherlands, 1983.
- [84] Belytschko T., Liu W.K., and Moran B. *Nonlinear Finite Elements for Continua and Structures*. Wiley, Hoboken, New Jersey, USA, 2000.
- [85] Gerstner T. and Griebel M. Dimension-adaptive tensor-product quadrature. *Computing*, 71:65–87, 2003.
- [86] Liu T.G., Khoo B.C., and Wang C.W. The ghost fluid method for compressible gas-water simulation. *Journal of Computational Physics*, 204:193–221, 2005.
- [87] Hughes T.J.R. *Computational Methods for Transient Analysis*, chapter Analysis of Transient Algorithms with Particular Reference to Stability Behaviour, pages 67–155. North-Holland, Netherlands, 1983.
- [88] Hughes T.J.R. *The Finite Element Method. Linear Static and Dynamic Finite Element Analysis*. Prentice Hall, Upper Saddle River, New Jersey, USA, 1987.
- [89] Hughes T.J.R., Pister K.S., and Taylor R.L. Implicit-explicit finite-elements in non-linear transient analysis. *Computer Methods in Applied Mechanics and Engineering*, 17:159–182, 1979.

- [90] van Albada G.D., van Leer B., and Roberts W.W. Jr. A comparative study of computational methods in cosmic gas dynamics. *Astronomy and Astrophysics*, 108:76–84, 1982.
- [91] van Leer B. Towards the ultimate conservative difference scheme. V - A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32:101–136, 1979.
- [92] Liu T. G. Wang C. W. and Khoo B. C. A real ghost fluid method for the simulation of multimediuim compressible flow. *SIAM Journal on Scientific Computing*, 28:278–302, 2006.
- [93] Cai X.C., Keyes D.E., and Venkatakrishnan V. Newton-Krylov-Schwartz: an implicit solver for CFD. In *Proceedings of the Eighth International Conference on Domain Decomposition Methods*, pages 387–400. Wiley, 1995.
- [94] Hu X.Y., Khoo B.C., Adams N.A., and Huang F.L. A conservative interface method for compressible flows. *Journal of Computational Physics*, 219:553–578, 2006.