A MULTIMATERIAL EULERIAN APPROACH FOR FLUID-SOLID INTERACTION

Benjamin Obadia

PHD THESIS

SCHOOL OF ENGINEERING

CRANFIELD UNIVERSITY

Cranfield University

Benjamin Obadia

A MULTIMATERIAL EULERIAN APPROACH FOR FLUID-SOLID INTERACTION

SCHOOL OF ENGINEERING

PHD THESIS

Academic Year 2011-2012

Supervisor: Dimitris Drikakis

Abstract

This thesis is devoted to understanding and modeling multimaterial interactions, and to develop accordingly a robust scheme taking into account the largest variety of those, with a particular interest in resolving solid/fluid configurations. This very general frame of studies can be tackled with numerous different approaches as several issues arise and need to be addressed before attempting any modelisation of these problems. A first questioning should be the frame of reference to be used for the materials considered. Eulerian shock-capturing schemes have advantages for modeling problems involving complex non-linear wave structures and large deformations. If originally reserved mostly to fluids components, recent work has focused on extending Eulerian schemes to other media such as solid dynamics, as long as the set of equations employed is written under a hyperbolic system of conservation laws.

Another matter of interest when dealing with multiple immiscible materials it the necessity to include some means of tracking material boundaries within a numerical scheme. Interface tracking methods based on the use of level set functions are an attractive alternative for problems with sliding interfaces since it allows discontinuous velocity profiles at the material boundaries whilst employing fixed grids. However, its intrinsic lack of variables conservation needs to be circumvented by applying an appropriate fix near the interface, where cells might comprise multiple components.

Another requirement is the ability to correctly predict the physical interaction at the interface between the materials. For that purpose, the Riemann problem corresponding to the interfacial conditions needs to be formulated and solved. This implies in turn the need of appropriate Riemann solvers; if they are largely available when the materials are identical (i.e. governed by the same set of equations), a specific Riemann solver will be developed to account for fluid/solid interaction.

Eventually, these newly developed methods will be tested on a wide range of different multimaterial problems, involving several materials undergoing large deformations. The materials used, whether modeling fluid/fluid or solid/fluid interactions, will be tested using various initial conditions from both sides of the interface, to demonstrate the robustness of the solver and its flexibility. These testcases will be carried out in 1D, 2D and 3D frames, and compared to exact solutions or other numerical experiments conducted in previous studies.

Acknowledgements

First and foremost, I would like to thank Prof. Dimitris Drikakis for both having granted me the opportunity of achieving this PhD, his supervision throughout all of it, and his clear and precise guidance over these last three years. I am extremely grateful to Phil Barton, who provided me priceless help, answering all my (numerous!) questions about his previous work on which I heavily relied. I also thank all my colleagues from the FMaCS group for their availability and their kindness in answering all the questions I could have as a newcomer: Ben Thornber, Marco Hahn, Max Starr, and all the ones I forget to mention!

I thank my officemate Konstantinos, a.k.a wolf, for our mutual emulation during the office hours. It was a pleasure to sharing my office with him, along with all our jokes and habits built over the years. I can mention as well all the friends I met here, for their constant support, and the good time which made my Cranfield experience unforgettable: Benjamin “Leroy”, Antonis “TennisPro”, the Pirate, Christos “Barbarossa”, Ioannis “El Bastardo”, Matthias “Kenny”, Yingchun “French learner”, Huijia “traditional”, and all my lovely housemates who were always up for a fun time after hard working days in the office.

And of course, I am infinitely grateful to my parents for the love they show me everyday, giving me limitless support through all the times of my PhD. It is without question that I dedicate this work to them.

Contents

List of Figures

List of Tables

Nomenclature

$\mathbf{A}$	Jacobian matrix
$\mathcal{A}$	aperture
a	sound of speed
$\mathfrak{B}$	band of cell near the interface
C	CFL number
$\mathcal{C}$	vector of characteristic variables
$D^{\pm}$	finite difference of derivative
D	detonation front velocity
$\mathcal{D}$	diagonal matrix
$\mathbf{E}(S)$	energy function of surface S
E	total energy
$\mathbf{e}$	unit vectors
$\mathbf{F}$	deformation gradient
$\mathcal{F}$	forcing term
F	flux function
$\mathbf{G}$	Finger stress tensor
G	shear modulus
$H(\phi)$	Godunov Hamiltonian
H	enthalpy
$\mathbf{H}$	transformation primitive to conservative matrix
h	grid spacing

$\mathbf{I}$	identity matrix
$\mathbf{K}, \mathbf{Q}$	orthogonal matrices
L	norm
$\mathbf{n}$	normal
P	WENO polynomial ; pressure
Q	total chemical energy
q	potential chemical energy
$\dot{q}$	reaction rate
$\mathbf{R}/\mathbf{L}$	right/Left eigenvectors
$\mathcal{R}$	auxiliary variables
$\mathcal{S}$	entropy
S	source terms
$S(\phi)$	sign function
t	time
$\mathbf{U}$	vector of conserved variables
$\mathbf{u}$	velocity vector
u	components of $\mathbf{U}$**,** velocity in x-direction
v	velocity in y-direction
V	volume
w	velocity in z-direction
$\mathbf{W}$	vector of primitive variables
x, y, z	spatial directions

Units

s	seconds
ms	milliseconds
μs	microseconds
m	meters
cm	centimeters
J	Joules
K	Kelvins
g	grams
kg	kilograms
Pa	Pascals
GPa	gigaPascals

Superscripts

C	compatibility, conserved
ign	ignition time
n	time, power law coefficient
P	plasticity
S/F	solid/fluid
T	transpose
-1	inverse
$\backsim$	Riemann problem solution
$*$	previous solution

Subscripts

i, j, k, l	cell centered indexes, going from 1 to 3 unless specified otherwise
R/L	right/left state
S/F	solid/fluid
SM	small cell
TG	target cell
0	reference/initial state
’	rotated variable

Greek letters

α	WENO coefficient, volume fraction
β	WENO coefficient, RK coefficient
ρ	density
ξ	internal energy
λ	eigenvalue; wave speed
γ	target value during reinitialisation, RK coefficient, Gamma-law gas coefficient
Γ	interface
ϕ	level set function
τ	artificial time; relaxation time
Δ	difference
∇	gradient
σ	stress tensor
Ω	acoustic tensor; computational domain
π	permutation matrix

Ψ	auxiliary variables
Υ	auxiliary variables
ϵ	small number $\ll 1$
ε	marker for material position

Acronyms

ALE	arbitrary Lagrangian-Eulerian
B-N	Baer-Nunziato
CAD	computer-aided design
CFL	Courant- Friendrichs-Lewy
CV	control volume
EOS	equation of state
ENO	essentially non-oscillatory
GFM	ghost fluid method
GNL	genuinely nonlinear fields
IC	initial condition
IVP	initial value problem
JWL	Jones Wilkins Lee
LDF	linearly degenerate fields
MGFM	modified ghost fluid method
MPWENO	monotonicity preserving weighted essentially non-oscillatory
PDE	partial differential equation
RK	Runge-Kutta
SIM	sharp interface methods

TV	total variation
TVD	total variation diminishing
UNDEX	underwater explosion
VOF	volume of fluid method
WENO	weighted essentially non-oscillatory

1

Introduction

Fluid dynamics are everywhere. From the smallest scales to the largest, it is constantly interacting with our physical world, and thus is one of the most investigated fields of modern sciences. It affects extremely diversified domains such as medical, biological, aerodynamics, magnetohydrodynamics, chemicals, or propulsion applications. The common point between all these specialties is the set of mathematical equations behind the flows concerned. Even if the formulation can differ from one to another due to some specific terms and/or simplifications, it can always be expressed under similar form, allowing crossovers between the fields. A resolution method developed for a specific application will be for instance also applied in all the similar applications.

One of the most challenging topics in fluid dynamics is multimaterial flows. Multimaterial problems have been one of the most interesting areas of research during the years. For instance, compressible two-phase flows can occur both in natural applications (geophysics or astrophysics), such as sandstorms or volcanic eruptions, and in technological applications as well, such as power plants, detonations, combustion, abrasion...The variety of problems, whether occurring at macro or micro length scales, makes it fascinating in order to further understand all the complexity of the phenomena occurring. Crucial questions arise when considering those problems, and needs to be addressed and answered in a consistent way, which is generally very problem-dependent. Indeed, regarding the materials involved and their properties generally gives a clue about the method to be used in order to capture and determine all the relevant phenomena susceptible to happen. Another kind of concern when dealing with several materials, besides the behavior of the materials themselves, is to determine with a great accuracy the location of the interface. That problem can be intrinsically present in the modeling of the mixture, which leads to so called "interface-capturing methods", or be an almost independent problem solved in conjunction with the multimaterial problem itself. These methods are called "interface-tracking methods".

1.1 Multiphase flows modeling

Multiphase flows can occur in very diversified configurations. Generally, the main division in these flows is dispersed flows in opposition to separated flows. The first one covers the case when a dispersed phase, such as particles or bubbles moves into a carrier medium, like air or water. Sandstorms, rains, snowstorms configurations can illustrate these kinds of flows. On the contrary, separated flows consist in two or more continuous materials separated by an interface (see [13, 68]).

Over the years, several approaches have been investigated to model multiphase problems. For instance, one of these approaches, developed by Drew and Passman [18], is the averaging approach. It consists in describing every phase as particles [15], evolving in a carrier fluid medium. The density of particles is represented using distribution functions, which are then integrated over time and space in order to get the solution. However, this approach shows limitations, as the kinetic theories are difficult to apply, and then preventing getting all the relevant phenomena that might be occurring in the flow fields, such as instabilities.

Another model widely used is the mixture approach. It treats the flow and the interface as a diffusive numerical mixture between two components (or more, even though the literature focuses greatly on two components problems), every single cell being thus considered as containing a certain amount of the two materials. The most famous model based on this modeling is the Baer-Nunziato (originally written in 1986), which has been then developed to take into account more complex effects, such as micro inertia or viscosity. On the other hand, others simplified models have been derived from the original B-N model [44, 58, 64, 65, 6] (for instance, the one pressure-one velocity model).

These two methods are obviously more relevant when dealing with the dispersed flows mentioned above. Nevertheless, these flows are not the ones considered in this study, and will not be, as a result, investigated any further.

One of the most relevant fields of interest when studying separated flows is the ability to correctly predict and determine the location of the interface separating the materials. The two latter approaches are in that sense “interface capturing”, which means that solving the new set of equations automatically captures the interfaces between materials. They are particularly useful when interfaces are dynamically created, for instance in cavitating flows when gas bubbles dynamically appear in a liquid. On the other hand, the numerical techniques available today cause some numerical smearing close to the interfaces when these interface-capturing methods are used.

Other classes of methods are called sharp interface methods (SIM). They assume the interface is initially known, and is then tracked in time given the flow status. Two main techniques have been developed over the years in that respect. First, the volume-of-fluid (VOF) technique, which consists in treating each computational cell as possibly containing a mixture of both fluids. The volume occupied by each fluid is represented by the volume fraction, transported with the flow. Given the volume fraction in every cell, the interface is in turn reconstructed with different levels of accuracy. This approach ensures full conservation of the physical variables (since the related equation for the volume fraction comes from a finite volume analysis), but the reconstructed interface can take a jagged shape, especially in case of large deformation.

The second interface tracking method relies on the use of functions artificially introduced, named “level-set”. It consists in using purely mathematical functions designed to represent the interface. More specifically, an isocontour of these functions is the actual interface. It allows removing the problem mentioned above for the VOF technique, i.e. the interface reconstruction, since the interface shape is directly given by the isocontour, and not by an approximate piecewise reconstruction. However, this method has also its limitations, the most straightforward one being the lack of conservation of variables since that function is purely artificial and not directly related to the physical parameter and the conserved variables. Such limitation has been the main focus of many recent developments [21, 56, 72], rendering the level-set approach still competitive and popular when dealing with separated multiphase flows. Throughout this study, that method will be favored, for its ease of extending to multi-dimensions, the direct access to geometric information, and the inherent continuity of the interfaces. Besides, the wide-spread use of level-sets in the computer graphics community means that sophisticated techniques now exist for accurately and efficiently reconstructing complex geometries from discrete sets of points that reside on the geometric surfaces (easily obtainable from CAD, see [83]).

It should be noted at this stage that much of the previous works have concentrated solely on fluid and gaseous phase. In fact, having the same set of equations from both sides of the interface makes the treatment of the interfacial conditions much easier, despite having sometimes different equations-of-state (EOS) between the fluids. Another way of understanding this trend is the will of focusing on the improvement of the interface tracking method itself, before trying to extend it to more complex problems. This explains the fresh upsurge of more complex multiphase flows, particularly the ones involving fluid-structure interaction. The more different the materials across the interface, the harder it is to cope with, since the conditions across the interface might change drastically (pressure ratios for instance), in conjunction with different behaviors and physical phenomena occurring in each material (fluid cavitation, non-linear elasticity or plastic deformations in solid materials etc...).

A way of efficiently dealing with these issues is the use of ghost cells, in order to treat each material "independently" and thus obtaining all the relevant physical phenomena provided by each model used for every material. As the single-component models are usually well investigated, it is a very convenient approach to treat these problems. Indeed, it generally allows to use all the high-order methods developed for these material models (high-order reconstruction of variables, high-order time integration for instance), and thus captures the possibly complex behaviors occurring. Originally developed for simple multi-fluid problems in [23, 45], the ghost cells technique has been successively improved in order to give a more accurate solution in case of large variables ratios across the interface. It has also been applied to multi-solid problems in [10], and some attempts have been made to treat fluid-solid problems as well [24, 46].

1.2 Fluid/Solid interaction

Fluid/solid interaction is one of the most challenging topics in the wide range of multimaterial problems. It crosses over many disciplines, from aerospace to medical applications. The related problems engulf a very important range of application, such as explosions in confined spaces, fluid-structure interactions, impacts problems etc...Environmental issues could also be named, such as tsunamis, asteroids impacts...

Here again, that coupling becomes extremely complex since both of these materials have their own properties and behavior, reflected in a mathematical way by different sets of equations. The issues encountered in such interactions are usually:

- Keeping a sharp interface even when large deformations occur (e.g. : impacts problems).
- Non-linear wave propagation in the different media.
- Accurate modeling of the constitutive properties of the solid medium under strong shock, and calculation of its elasto-plastic behavior.

A crucial question to be addressed is the formulation of the set of equations for each material, i.e. in a Lagrangian or Eulerian frame. That question is subject to many debates and is the source of very different approaches since traditionally, fluids are treated in Eulerian formulation, and solids in Lagrangian frames.

There are three possible ways of treating such problems. Both fluid and solid equations can be solved using Eulerian numerical methods; the fluid equation can be solved with Eulerian numerical methods and the solid with Lagrangian numerical methods (Arbitrary Lagrangian-Eulerian method: ALE); or both sets of equations can be solved with Lagrangian numerical methods.

The latter has been shown difficult since Lagrangian methods are particularly poor when dealing with fluids, especially when it comes to strong shocks, large deformations, or high speed flows. Therefore, it appears to be extremely unsuited to the cases we would like to investigate as described above.

The ALE method is more attractive since it allows important deformations in the fluid. It combines the traditional models of each material taken separately, and thus constitutes a more intuitive approach. However, The interface between the solid and the fluid being advected in Lagrangian fashion, large distortions are not allowed, as we notice for instance disappearance of cells at the interface [37].

A full Eulerian frame avoid these mesh distortions, consequently allowing large deformations, and thus more applications in multimaterials problems. However, although Eulerian solvers are widely used in fluid mechanics (see Toro [75]) and have been proven both robust and accurate, the Eulerian formulation for solid equations has always been challenging. In fact, as mentioned earlier, Lagrangian formulations are usually most used to describe the solid motion. Indeed, Eulerian frame requires additional laws to transport the material properties history, like elastoplastic strains that are very material-dependent. Lagrangian methods are in that respect more efficient in a sense that each element carries its corresponding history, allowing the implementation of complex constitutive equations for the material strength. However, as mentioned before, these methods behave poorly in the case of large deformation, since the interface is distorted. On the contrary, Eulerian methods, although more difficult to implement, avoid this problem, and in addition, allow to capture complex behaviors, such as non-linear wave propagation. On the other side, Eulerian methods have been well investigated for fluid dynamics problem and several high-order schemes have been developed in order to capture shocks, non-linear phenomena etc...

Recent developments have made possible the application of these existing tools by formulating the solid mechanics theory under a hyperbolic system of conservation laws. Barton and Titarev have proposed a system taking into account the non-linear waves as well as plastic deformation, using high-order methods. An exact Riemann solver has been presented [11], relying on characteristic analysis coupled with a high-order monotonicity- preserving weighted essentially non-oscillatory (MPWENO) reconstruction.

These advances make the Eulerian-Eulerian approach very competitive in our study frame. It allows coping with most of usual problems encountered when dealing with strong interactions and large deformations, more particularly in the interfacial region. As a result, the present study will focus on treating the multiphase flows in a pure Eulerian fashion, with fixed Cartesian grids.

1.3 Thesis structure

This thesis will be structured as follows. In section 2 are presented the numerical methods used in the study frame. It will include the crucial properties of hyperbolic systems of conservation laws, along with the details of the resolution methods used. It will rely on presenting characteristics analysis of the hyperbolic systems, with the relevant description of characteristic-based schemes, and high-order methods for time integration and fluxes computation, essentially Runge-Kutta and WENO reconstruction. Section 3 will be devoted to the interface tracking method, i.e. the level-set functions, and all the related concepts, like reinitialization or geometry reconstruction. In addition to the basic key-concepts, a new 3D conservative method for related level-set functions interface tracking methods will be presented, independently of any set of equations used. Section 4 will consist in applying the latter method to fluid/fluid applications, and to demonstrate its ability even in case of large deformation. Several testcases will be presented and discussed for that matter. It will also include the detailed model used for fluids, including the governing equations and the equations of state. Section 5 will focus on the development of a dedicated Riemann solver for fluid/solid interaction problems. Coupled with the earlier mentioned interface tracking method, several fluid/solid problems will be tested on very diverse configurations in order to show the robustness and capabilities of that solver, from "simple" initial value problems (IVP) to more complex 3D testcases. Finally, conclusions and future work will be presented in Section 6.

1.4 Publications and conference

- P.T Barton, B. Obadia, D. Drikakis, A conservative level-set based method for compressible solid/fluid problems on fixed grids, *J. Comp. Phys.* Vol 230, Issue 21, (published 1 September 2011), Pages 7867-7890.

- B. Obadia, P.T Barton, D. Drikakis, Shocked Fluid/Fluid and Fluid/Solid Interactions Using a Conservative Level-Set Method, ISSW 28th July 2011 in Manchester.

2

Numerical methods

Given the very diverse nature of components, a crucial concern is to use adequate models for each of them. As complex interactions will occur between the materials, the governing equations need to be expressed in a way rendering as convenient as possible the treatment of the interface. To achieve this, both sets of equations are going to be expressed in hyperbolic conservative form, and a similar characteristic analysis will be performed for each of them. Consequently, basic concepts and notions about these systems are going to be presented hereby.

2.1 Notions on hyperbolic systems and conservation laws

2.1.1 Basic definitions

A system of first order partial differential equations (1st-order PDE) is a relation involving a set of n variables depending on m independent variables $(x_1, ..., x_m)$: $\mathbf{U}$ =$(u_1(x_1, ..., x_m), ..., u_n(x_1, ..., x_m))$ and their partial derivatives, i.e. it exists a function G such that:

$$G(u_i(x_j), x_j, \partial u_i/\partial x_j) = 0 \;\; with \begin{cases} i = 1....n \\ j = 1....m \end{cases} \tag{2.1}$$

It should be noted that when dealing with physical applications and therefore the frame of that study, the so-called independent variables are generally time and space variables, i.e. m=*4* $(x_1 = x, x_2 = y, x_3 = z, x_4 = t)$.

A system of conservation laws is a particular type of PDE, which can be written in the following form (the double index refers to the Einstein notation for summing over indexes):

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial F^i(\mathbf{U})}{\partial x_i} = S \tag{2.2}$$

where $\mathbf{U}$ is the vector of considered variables (for instance: density, velocity, pressure, stress, entropy...), $F^i = (f_1, ..., f_n)^i$ are flux functions in the x_i direction, and S is a vector containing source terms. Whether dealing with fluid or solid dynamics, the conservation systems governing the relevant variables are derived upon physical considerations, like Lavoisier's principle (conservation of total mass) during a transformation. This usually motivates the governing equations to be firstly written in an integral form, which in turn allows getting, under certain assumptions, to switch to differential form, eventually leading to systems similar to the one presented above. However, it is sometimes necessary to revert to the integral form in order to get the correct solution (it is crucial when discontinuities like shocks, appear), and that matter will be discussed later when presenting Godunov's method.

System (2.2) can be rewritten as:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_i(\mathbf{U})\frac{\partial \mathbf{U}}{\partial x_i} = S \tag{2.3}$$

where $\mathbf{A}_i$ are the Jacobian of the flux functions, i.e. the following nxn matrix:

$$\mathbf{A}_i(\mathbf{U}) = \frac{\partial F^i}{\partial \mathbf{U}} = \begin{bmatrix} \partial f_1/\partial u_1 & . & . & . & \partial f_1/\partial u_n \\ . & & . & . & . \\ . & & . & & . \\ . & & . & . & . \\ \partial f_n/\partial u_1 & . & . & . & \partial f_n/\partial u_n \end{bmatrix} \tag{2.4}$$

Under the form of Equation (2.3), systems of conservation laws are classified as *quasi-linear* systems. For the sake of simplicity and understanding, the incoming concepts will be presented in one-dimensional cases, so under the following form:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} = S \tag{2.5}$$

The attributes of matrix $\mathbf{A}$ are decisive in order to get a further understanding of these systems behaviors.

First, we introduce the eigenvalues of matrix $\mathbf{A}$, which are the set of *a priori* complex numbers solution of the equation:

$$|\mathbf{A}-\lambda\mathbf{I}| = det(\mathbf{A}-\lambda\mathbf{I}) = 0 \tag{2.6}$$

where $\mathbf{I}$ is the identity matrix, and *det* the determinant function. It should be noted that there are n solutions to (2.6) since $det(\mathbf{A}-\lambda\mathbf{I})$ is a polynomial of degree n. However, these solutions might not all be distinct, in which case they have an order of multiplicity greater than one.

A system is qualified as hyperbolic if the eigenvalues of the matrix are all pure real numbers. This will actually be the case throughout the whole study (fluid and solid systems will be expressed in hyperbolic form). If the eigenvalues are distinct one from each other, the system is said *strictly* hyperbolic.

It should also be noted that as $\mathbf{A}$ depends on $\boldsymbol{U}$, each eigenvalue is not constant *a priori*, and a characteristic field is defined for each of them.

We define next the eigenvectors: an eigenvector X of a matrix $\mathbf{A}$ (associated to an eigenvalue λ), is a vector complying one of the following relation:

$$\mathbf{A}X = \lambda X \tag{2.7}$$

$$X\mathbf{A} = \lambda X \tag{2.8}$$

If condition (2.7) is satisfied, the eigenvector is qualified as a *right* eigenvector; if it satisfies (2.8), it is classified as a *left* eigenvector.

A fundamental feature of hyperbolic equations will come from the decomposition of matrix $\mathbf{A}$ in order to reduce the multi-dimensional system into a set of scalar equations. This is achieved by the diagonalisation of the matrix, which consists in the following: provided that the set of independent eigenvectors of the matrix $\mathbf{A}$ forms a vector space of dimension n, $\mathbf{A}$ is diagonalisable, meaning that there is a diagonal matrix $\boldsymbol{D}$ whose diagonal elements are the eigenvalues of $\mathbf{A}$, so that:

$$\mathbf{A} = K.\mathbf{D}.K^{-1}\ \mathit{with}\ \mathbf{D} = \begin{bmatrix} \lambda_1 & 0 & . & . & 0 \\ 0 & . & . & . & . \\ . & . & \lambda_i & . & . \\ . & . & . & . & 0 \\ 0 & . & . & 0 & \lambda_n \end{bmatrix} \tag{2.9}$$

K is the matrix which columns are the right eigenvectors of $\mathbf{A}$. As we assumed that these vectors are a base of the vector space of dimension n, the proof of invertibility of K is immediate.

Thus, the system (2.5) can now be rewritten:

$$\frac{\partial \mathbf{U}}{\partial t} + K.\mathbf{D}.K^{-1}\frac{\partial \mathbf{U}}{\partial x} = 0 \tag{2.10}$$

Here, the source term S is taken as zero, without loss of generality because the source term will be numerically treated later *via* splitting procedures (Section 5).

If it is assumed that the coefficients of $\mathbf{A}$ are evaluated at some constant state, the eigenvalues and eigenvectors are also constant, which in turn allows the next transformation of system (2.10) into:

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{D}\frac{\partial \mathbf{W}}{\partial x} = 0 \tag{2.11}$$

with $\mathbf{W} = K^{-1}\mathbf{U}$ a vector of transformed variables called *characteristic* variables. The system (2.11) is now entirely decoupled since each line of the system becomes:

$$\frac{\partial \mathbf{W}_i}{\partial t} + \lambda_i \frac{\partial \mathbf{W}_i}{\partial x} = 0 \ \ i = 1...n \tag{2.12}$$

System (2.5) has become a system of n scalar equations. System (2.12) is called the canonical form of the system and emphasizes the propagation of n characteristic variables along n characteristic curves, since (2.12) transforms directly to:

$$\frac{dx}{dt} = \lambda_i, \ \ i = 1...n \tag{2.13}$$

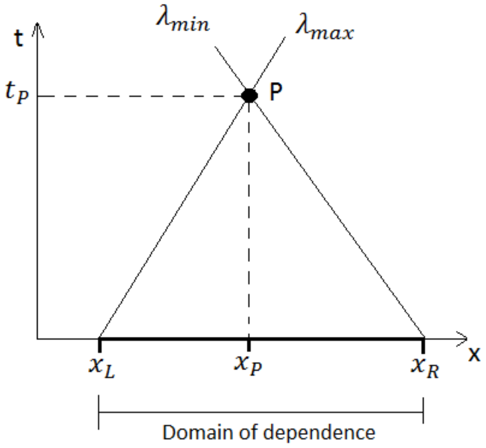


Figure 2.1: Domain of dependence of a random point P in the (x,t) plane.

As a matter of fact, we can infer that the solution consists in characteristic variables $\mathbf{W}_i$ traveling at different velocities λ_i. In the case when λ_i is constant, the characteristic curve in the (x,t) plane is a straight line. Physically, Equation (2.12) is also referred as the *advection* equation since it corresponds to the advection of the quantity $\mathbf{W}_i$ at the velocity λ_i.

It allows to understand the two following concepts:

- Domain of determinacy: for a given hyperbolic system, the variables state only depends on a domain of dependence determined by considering the lowest and fastest characteristic speeds, as illustrated on Figure 2.1, where the state value at point P can only depend from the state of the variables in the interval ($[x_L, x_R]$.
- Region of influence: A random point P and its given state can only influence a region in the (x,t) plane comprised in the cone bordered by the straight lines $(\lambda_{min}, \lambda_{max})$, as illustrated in Figure 2.2. In other words, the information contained at point P cannot travel faster than the maximum value of the characteristic speeds, and slower than the smallest one.

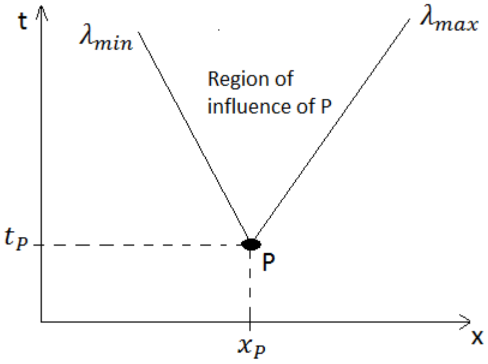


Figure 2.2: Zone of influence of a random point P in the (x,t) plane.

Now that the main concepts related to hyperbolic systems have been presented, the Riemann problem, which is an essential matter in numerical method, and its main properties, is introduced.

2.1.2 The Riemann problem

The Riemann problem is an initial value problem (IVP) consisting of two uniform conditions where the states vary discontinuously. Depending on how these uniform states are chosen, several behaviors can arise and will be hereby presented. Solution of the Riemann problem has gained significant importance in numerical schemes for systems of hyperbolic conservation laws. Indeed, the numerical methods employed will be of Godunov's type, i.e. relying on solving a Riemann problem at each intercell boundary of the computational domain.

Let us consider System (2.5) with the following initial conditions (IC):

$$\mathbf{U}_0(x,0) = \begin{cases} \mathbf{U}_L \; if \; x < 0 \\ \mathbf{U}_R \; if \; x > 0 \end{cases} \tag{2.14}$$

In a case of a linear system with constant coefficients (when the wave speeds are constants regardless of time or position), the solution is easy to grasp. If we consider a 2x2 system with 2 characteristic waves, 3 regions naturally appear in the solution of the (x,t) plane: one influenced only by the left value, one by the right, and one by the interaction of the discontinuity. This latter middle region is traditionally called the "star region" and is illustrated on Figure 2.3.

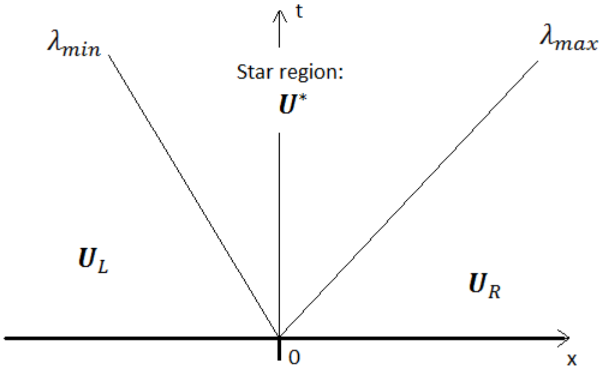


Figure 2.3: Illustration of the star region in a 2x2 system linear with constant coefficients

The goal of a Riemann solver is to precisely determine the location of the star region, and the associated values of the variables. In the particular case of Figure 2.3, the star region state is straightforward to get since the characteristic variables travel at the related characteristic speed. However, in non-linear cases, the characteristic speeds are state-dependent ($\lambda(\mathbf{U})$), so the traveling speeds coming from the right and the left are different, and leads to major phenomena: shocks and contact discontinuities, or rarefaction waves. These distortions in the wave speeds are typical of non-linear problems and will be of high matter of concern during that study.

In case of non-linearity, two types of characteristic fields must be distinguished (notation used are the same than in section 2.1.1):

- The linearly degenerate fields (LDF), satisfying:

$$\nabla\lambda_i(\mathbf{U}).K^{(i)}(\mathbf{U}) = 0, \ \ \forall\mathbf{U} \tag{2.15}$$

- The genuinely nonlinear fields (GNF), satisfying:

$$\nabla\lambda_i(\mathbf{U}).K^{(i)}(\mathbf{U}) \neq 0, \ \ \forall\mathbf{U} \tag{2.16}$$

When an i-th characteristic field is a LDF, the related wave is a **contact wave**, i.e. $\mathbf{U}_L$ and $\mathbf{U}_R$ are connected through a single jump discontinuity of speed S_i. In addition, the following properties are enforced:

1. The Rankine-Hugoniot conditions:

$$F(\mathbf{U}_R) - F(\mathbf{U}_L) = S_i(\mathbf{U}_R - \mathbf{U}_L) \tag{2.17}$$

2. The constancy of the Generalised Riemann Invariants across the wave (it is recalled that $\mathbf{W}$ is the vector of characteristic variables):

$$\frac{dw_1}{k_1^{(i)}} = \frac{dw_2}{k_2^{(i)}} = ... = \frac{dw_n}{k_n^{(i)}} \tag{2.18}$$

3. The parallel characteristic condition:

$$\lambda_i(\mathbf{U}_L) = \lambda_i(\mathbf{U}_R) = S_i \tag{2.19}$$

If the i-th characteristic field is a GNF, it can be of two types: a shockwave or a rarefaction wave. The mathematical condition related to that distinction is the following:

$$\begin{cases} |\lambda(\mathbf{U}_L)| > |\lambda(\mathbf{U}_R)| & \Longrightarrow Shock \\ |\lambda(\mathbf{U}_L)| < |\lambda(\mathbf{U}_R)| & \Longrightarrow Rarefaction \end{cases} \tag{2.20}$$

From one hand, a **shock wave** corresponds to a single jump discontinuity of speed S_i. Besides, the following properties are enforced:

1. The Rankine-Hugoniot conditions:

$$F(\mathbf{U}_R) - F(\mathbf{U}_L) = S_i(\mathbf{U}_R - \mathbf{U}_L) \tag{2.21}$$

2. The entropy condition:

$$\lambda_i(\mathbf{U}_L) > S_i > \lambda_i(\mathbf{U}_R) \tag{2.22}$$

On the other hand, a **rarefaction wave** connects the right and left states through a smooth transition, and complies the constancy of the Generalised Riemann Invariants across the wave:

$$\frac{dw_1}{k_1^{(i)}} = \frac{dw_2}{k_2^{(i)}} = ... = \frac{dw_n}{k_n^{(i)}}$$

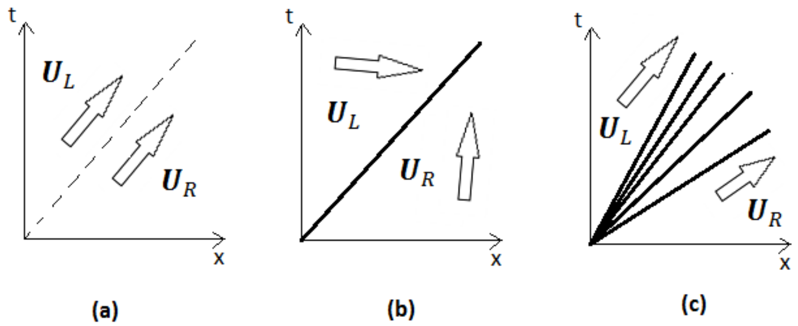


Figure 2.4: Waves pattern: (a) Contact wave, (b) Shock wave, (c) Rarefaction wave

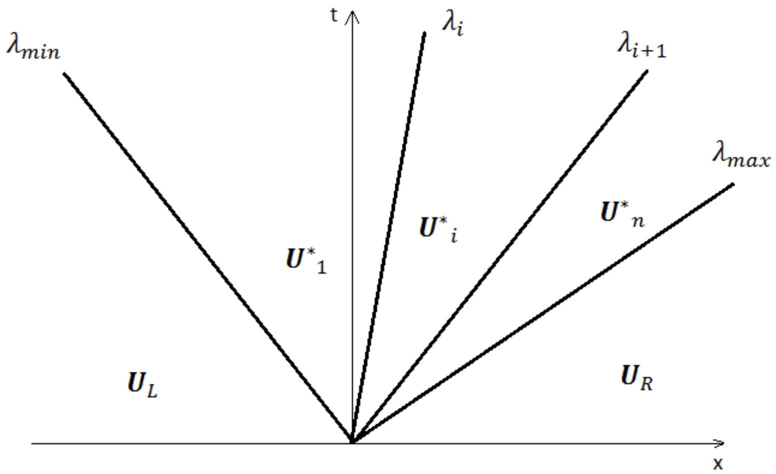


Figure 2.5: Structure of the solution for a non-linear hyperbolic system

The different types of waves are illustrated on Figure 2.4, and the complete solution form of a non-linear hyperbolic system of dimension n on Figure 2.5. The waves represented on Figure 2.5 can be rarefaction, shock, or contact waves.

In Figure 2.3, there was only one star region since it was assumed that only two traveling waves were propagating. However, when considering a random hyperbolic system of dimension n, up to n distinct characteristic waves can exist, separating n+1 different states (including $\mathbf{U}_L$ and $\mathbf{U}_R$) , with n-1 star regions. In case characteristic waves are not all distinct, the number of star regions decreases accordingly.

Further details about actual Riemann solvers will not be developed in this section, which introduces the usual concepts on hyperbolic systems. Depending on the materials that will be modeled later, different Riemann solvers will be used and described in due time. Nevertheless, it allows to understand the common methodology consisting in carrying accurate wave analysis when dealing with systems governed by hyperbolic PDE.

2.2 Finite volume method

The finite volume method (FVM) is the most common approach in computational fluid dynamics nowadays. It relies on dividing the domain into elementary control volumes (CV), commonly referred as cells. In each cell, piecewise constant data are stored. This is illustrated in Figure 2.6. The next step is to integrate the differential equation over each CV to get the solution at every time step throughout the whole mesh. It implies a discretization of the equations since the computational domain is now formed with a set of discrete and *a priori* discontinuous values. The FVM regards these discrete values as averages over finite volumes. In this manner, the discretised equations express the conservation principle for the variables inside the CV. The most compelling feature of the FVM is that the resulting solution satisfies the conservation of quantities such as mass, momentum, energy, and species. This is exactly satisfied for any CV as well as for the whole computational domain and for any number of control volumes, even with coarse grids. It is an ideal method for computing discontinuous solutions arising in many problems involving shocks, e.g. compressible flows. The CV can be rectangles, in the case of structured meshes, even if sometimes they can be of irregular shapes (random polygons) in case of unstructured meshes. Here, structured meshes will be considered, the CV being as a result segments (in 1D), rectangles (in 2D), or parallelepipeds (in 3D). It allows the convenient use of uniform spacing, i.e. each CV will be defined by its edges: $\Delta x, \Delta y$ and Δz.

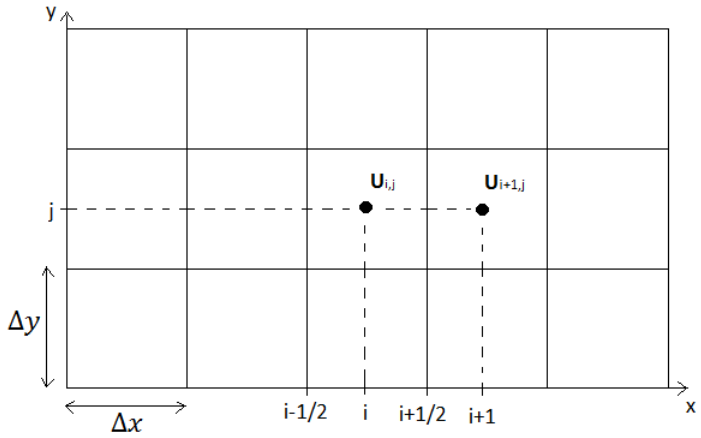


Figure 2.6: 2D discretization of the computational domain.

Traditionally, the discretization of the PDE (involving partial derivatives of variables) is achieved by finite differences and usually comes from a Taylor decomposition around every point of coordinates (i, j), that is, in one dimension (for a random function *f(x)*):

$$f(x_i + \Delta x) = f(x_i) + \sum_{k=1}^{\infty} \frac{(\Delta x)^k}{k!} f^{(k)}(x_i) \tag{2.23}$$

Practically, the index k of Equation (2.23) is chosen depending on the desired accuracy of the approximation, since that equation becomes the following at order n:

$$f(x_i + \Delta x) = f(x_i) + \sum_{k=1}^{n} \frac{(\Delta x)^k}{k!} f^{(k)}(x_i) + o((\Delta x)^n) \tag{2.24}$$

Replacing then the derivatives by the finite difference approximation in Equation (2.12) for instance allows expressing the vector of variables at time $n+1$ given the vector of variables at time n. This leads to a large variety of schemes (more details can be found in [4]) and this approach is referred as the finite difference method. However, it exploits only the differential form of the equation, which gives incorrect solution if the solution suffers discontinuities like shocks.

2.2.1 Godunov's method

To accurately capture shocks, one has to come back to the integral formulation of the equations, leading to the development of so-called conservative methods.

First of all, the cell averages are defined over finite volumes, and the cell average is now given by (1D will be assumed):

$$\mathbf{U}_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}(x, t^n) dx \tag{2.25}$$

where index i represents the spatial position of the cell, and t denotes the current time. If the hyperbolic PDE is written under differential form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial F(\mathbf{U})}{\partial x} = 0 \tag{2.26}$$

It can be integrated over the CV $((x_{i-1/2}, x_{i+1/2}); (t^n, t^{n+1}))$, on time and space, leading to:

$$(\mathbf{U}_i^{n+1} - \mathbf{U}_i^n).\Delta x + \int_{t^n}^{t^{n+1}} (F(\mathbf{U}(x_{i+1/2}, t)) - F(\mathbf{U}(x_{i-1/2}, t)))dt = 0 \tag{2.27}$$

The second term of the left hand side is the physical flux passing at the intercell. Similarly to the definition given by Equation (2.25), the average flux can be defined as:

$$F_{i+1/2} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} F(\mathbf{U}(x_{i+1/2}, t))dt \tag{2.28}$$

Finally, Equation (2.26) can be rewritten as:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \frac{\Delta t}{\Delta x}(F_{i-1/2} - F_{i+1/2}) \tag{2.29}$$

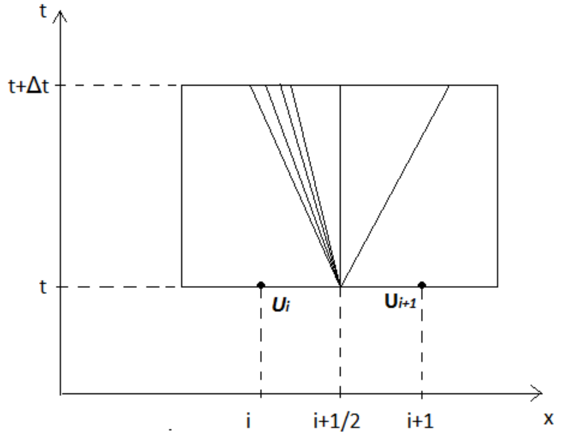


Figure 2.7: Riemann problem at the intercell.

This numerical scheme is conservative and is one of the most popular ways of solving hyperbolic PDE. It involves the numerical fluxes $F_{i+1/2}$ and $F_{i-1/2}$ which approximate the physical flux. The crucial matter is then to compute these fluxes with consistency. Godunov has been the first to propose a computation of these fluxes by solving a local Riemann problem at each intercell. His first order method consists in using a piecewise constant function to interpolate the cell center values to the cell face as seen in Figure 2.7, where the left and right conditions state of the Riemann problem are respectively $\mathbf{U}_i$ and $\mathbf{U}_{i+1}$. For the sake of illustration, both a shockwave and a rarefaction wave are represented, but one of the greatest challenges is to perfectly determine the wave pattern of the Riemann problem at the intercells. Wide ranges of exact and approximate Riemann solvers have been developed over the last decades. Roe, HLL, HLLC can be named among the most famous ones, and are largely detailed in the literature (see [75] for instance). The global method is illustrated on Figure 2.8, where an update in time is shown.

Once the solution of the Riemann problems at the intercells has been obtained, let's note them as $\overset{\frown}{\mathbf{U}}_{i+1/2}$ and $\overset{\frown}{\mathbf{U}}_{i-1/2}$, the intercell fluxes are computed as:

$$F_{i+1/2} = F(\overset{\frown}{\mathbf{U}}_{i+1/2}) \text{ and } F_{i-1/2} = F(\overset{\frown}{\mathbf{U}}_{i-1/2}) \tag{2.30}$$

and the solution is updated according to Equation (2.29).

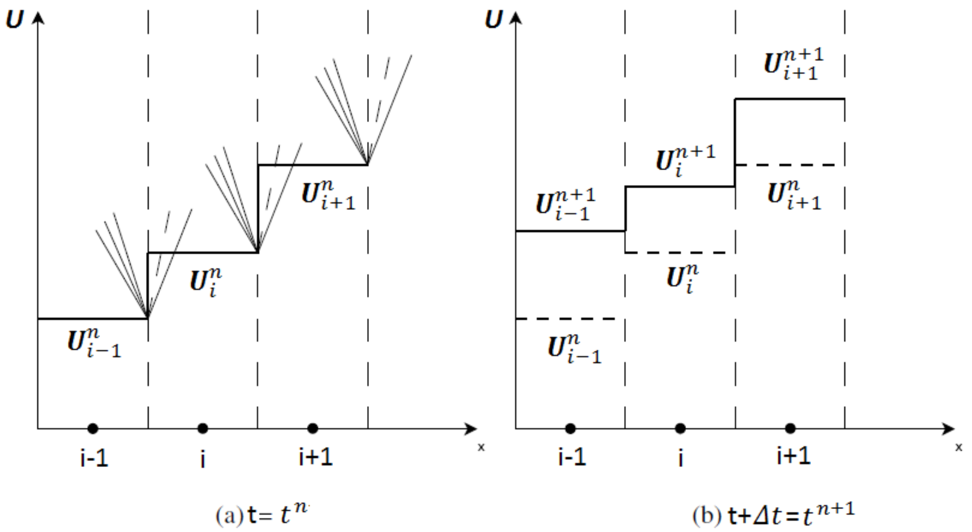


Figure 2.8: Godunov's first order method for FVM

2.2.2 Computation of timestep

The previous section introduces the basics of FVM and Godunov's method. As the method relies on solving a Riemann problem at the intercells, the wave propagation plays an important role and imposes some limitations on the time step if one wants to preserve the stability of the solution. Indeed, a simple way to visualize this is by examining Figure 2.9:

The blue wave travels at a speed that is slow enough to remain in the cell during a time step. It ensures stability since the information in one cell can only propagate in adjacent cells. However, if a wave (the red one) travels too fast, it escapes the cell domain and the solution is said to be unstable. As a corollary, the time step must be small enough to make sure that all the propagating waves remain bounded in a cell. It is thus straightforward to understand that the timestep will depend on both the dimensions of a cell (here, Δx) and the speed of the fastest way (the fastest wave being the “limiting” factor). To ensure stability, the scheme must meet the Courant-Friendrichs-Lewy condition (CFL) which states that if a wave is crossing a discrete grid, then the timestep must be less than the time for the wave to travel adjacent grid points. To reflect that condition, the CFL number, also designated by C is introduced:

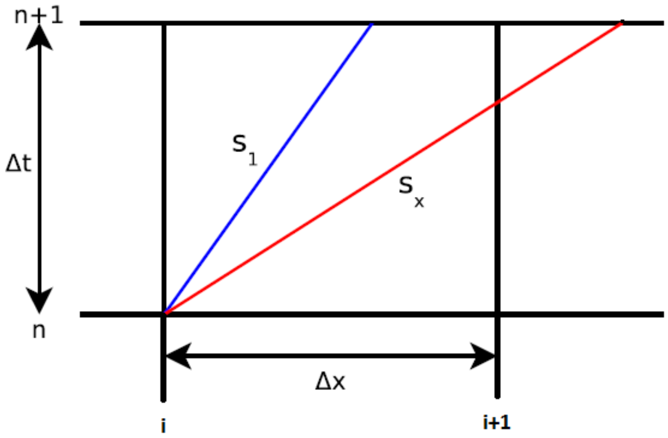


Figure 2.9: Wave speed and timestep limitation.

$$C = \frac{\Delta t}{\Delta x}.\lambda_{max} \tag{2.31}$$

The CFL number is usually fixed by the user throughout the whole computation. The limit of stability is, as explained above, when C=1. In fact, as the system is non-linear, the wave speed may vary in a non-linear manner, resulting in a faster wave than originally anticipated. In order to avoid related instabilities, it is common to take a lower CFL, that choice being very application-dependent. Once the CFL has been chosen, the grid being known in the case of fixed meshes, every timestep is computed thanks to:

$$\Delta t = \frac{C\Delta x}{\lambda_{max}} \tag{2.32}$$

It appears crucial that the wave speeds must be correctly estimated at the beginning of each time step if one wants to avoid having an unstable solution.

2.3 High Resolution Methods

In section 2.2 has been explained the first-order method originally developed by Godunov. As mentioned, it uses the piecewise constant data stored at the cell center for the IVP at the intercell, and in that sense, is first order accurate despite considered as “high resolution”. It behaves poorly, in terms of accuracy, when high gradients and strong discontinuities exist in the solution. To get a better accuracy, it appears consistent that the data taken as initial values for the Riemann problem should be different from the one at the cell center since the intercell is at a different space location. That concern led to variables reconstruction methods at the intercell of the boundary, using values obtained with a stencil of points rather than simply using the immediate adjacent cell center value. Instinctively, it might be tempting to reconstruct variables linearly (i.e. the variables at each intercell is a constant polynomial extrapolation of the stencil variables) however, it practically introduces artificial spurious oscillations in the solution even if a better accuracy is reached. Godunov showed that accuracy and stability of linear schemes are two contradictory requirements, and expressed it under the following statement (Godunov’s theorem, [28]):

“There are no monotone, linear schemes of second or higher order of accuracy”.

As a corollary, monotone schemes can only be first order accurate, which is unsatisfactory for most applications since too inaccurate. Consequently, new schemes have been developed implying non-linear differences, meaning that the variable reconstruction will depend on the character of the local solution. These high-resolution methods are aiming to reach the three following requirements (Harten, [34]):

- Provide a second order or higher order of accuracy in smooth part of the solution.
- Produce numerical solutions (relatively) free from spurious oscillations.
- Achieve a high-order resolution of discontinuities, defined as: the number of grid points capturing the discontinuity should be smaller in comparison of first-order monotone schemes.

These characteristics will be enforced by implementing certain restrictions, involving certain concepts which are described next.

2.3.1 Monotonicity Preserving

That property relates to the fact that the solution provided by the considered scheme should not be oscillatory, which is mathematically expressed as:

Whenever the data $\{\mathbf{U}_i^n\}$ is monotone at time t^n, the solution at time t^{n+1} should be monotone in the same sense.

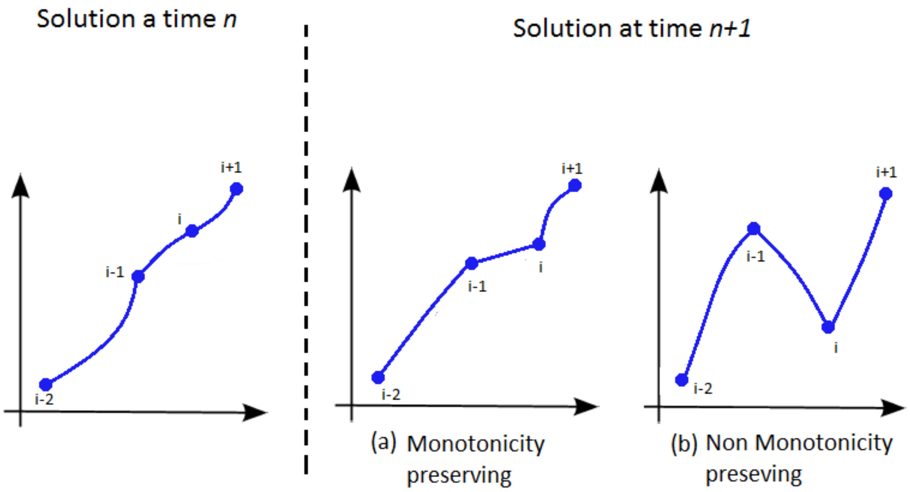


Figure 2.10: Monotonicity Preservation property.

That property is illustrated on Figure 2.10. In graph (b), the monotonicity preservation fails, since in the interval $(i-1, i)$, the solution at time *n+1* has the opposite sense of variation than at time *n* on the same interval. It shows the lack of stability which can occur from one time step to the other. If that property is satisfied nevertheless, oscillations will be avoided.

2.3.2 Total variation diminishing methods (TVD)

TVD methods are based of the total variation of the variables of the solution. The total variation of a variable $u(x)$ is defined as:

$$TV(u) = \lim_{\epsilon \to 0} \sup \frac{1}{\epsilon} \int_{-\infty}^{\infty} |u(x+\epsilon) - u(x)| dx \tag{2.33}$$

As the variable u depends *a priori* on t, the total variation is defined at a given time step, noted $TV(u^n)$. Besides, on a discretized domain, Equation (2.33) becomes, at time t^n:

$$TV(u^n) = \sum_{i=-\infty}^{+\infty} |u_{i+1}^n - u_i^n| \tag{2.34}$$

In fact, in order to get a finite TV, function u should have a constant value as approaching the infinite boundary.

Assuming a finite TV at the original time, a monotone scheme should abide the next two criteria between each time step:

- No new local extrema in the domain should be created.
- The value of a local minimum is non-decreasing, and the value of a local maximum is non-increasing.

These two conditions imply that the TV should not increase in time, and consequently:

$$TV(u^{n+1}) \leq TV(u^n) \; \forall n \tag{2.35}$$

A scheme satisfying Equation (2.35) is called TVD. The above relationship physically corresponds to the fact that the solution propagates at a speed u with unchanged shape, so that the TV should remain constant in time. It does not hold true if there are source terms in the system, since TV can thus be created physically and not only numerically. However, as mentioned in section 2.1, source terms can be treated *via* splitting methods, making the TVD definition to be relevant even when equations like (2.5) with $S \neq 0$ are considered.

2.3.3 High-order spatial reconstruction

Weighted Essentially Non-Oscillatory (WENO) schemes are a recent spatial reconstruction method allowing to reach very high-order of accuracy. The boundary-extrapolated values are obtained from cell averages by means of a high order polynomial reconstruction. They are a variant of Essentially Non-Oscillatory (ENO) schemes in which the polynomial used for the reconstruction corresponds to the smoothest slope in an adaptive stencil (which prevents the spurious oscillations appearing near discontinuities). Nowadays, ENO schemes and their variants constitute a well-established and competitive numerical framework for constructing high-order methods suited for multi-dimensional problems. Besides, it is particularly fitted when dealing with a rich solution structure, involving complex shock interaction for instance.

In opposition to ENO schemes, WENO reconstruction weights each slope regarding to some criteria (like smoothness, which will be illustrated below). The resulting convex combination gives an interpolating polynomial used to reconstruct variables at the intercell. The third and fifth order WENO schemes are going to be detailed below (see [19]):

3rd order WENO

3rd order WENO scheme is the simplest way to show how the method works. First, let us consider the two right and left linear interpolating polynomials used to get the right and left values of the Riemann problem at the intercell.

On the left:

$$P_L^i(x) = u_i + \frac{u_i - u_{i-1}}{\Delta x}(x - x_i) \tag{2.36}$$

And on the right:

$$P_R^i(x) = u_i + \frac{u_{i+1} - u_i}{\Delta x}(x - x_i) \tag{2.37}$$

The final polynomial used for the extrapolation of the right and left values is, as announced, a convex combination of the both previously defined polynomials in Equations (2.36) and (2.37) weighted so (the index i is dropped for clarity):

$$P_{WENO}(x) = \frac{\alpha_L}{\alpha_L + \alpha_R} P_L(x) + \frac{\alpha_R}{\alpha_L + \alpha_R} P_R(x) \tag{2.38}$$

where:

$$\alpha_L = \frac{C_L}{((\beta_L) + \epsilon)^2} \quad and \quad \alpha_R = \frac{C_R}{((\beta_R) + \epsilon)^2} \tag{2.39}$$

with:

- β is a smoothness indicator given by: $\beta_L = (u_i - u_{i-1})^2$ and $\beta_R = (u_{i+1} - u_i)^2$.
- ϵ is a small arbitrary parameter to avoid any division by zero, usually set to 10^{-6}.
- C_L and C_R are coefficients given by:

- if $\partial F(\mathbf{U})/\partial \mathbf{U} < 0$: $C_L = 1$ and $C_L = 1/2$ (it is recalled that F is the flux function).
- if $\partial F(\mathbf{U})/\partial \mathbf{U} > 0$: $C_L = 1/2$ and $C_L = 1$.

5th order WENO

The 5th order becomes more complicated due to an extended stencil; the left extrapolated value is defined as:

$$u_L = \frac{1}{6}[\omega_1(-u_{i+1} + 5u_i + 2u_{i-1}) + \omega_2(-u_{i-2} + 5u_{i-1} + 2u_i) + \omega_3(2u_{i-3} - 7u_{i-2} + 11u_{i-1})] \quad (2.40)$$

where $\omega_k,\ k = 1, 2, 3$ are given by:

$$\omega_k = \frac{\alpha_k}{\sum_{l=1}^{3} \alpha_l} \quad (2.41)$$

with:

$$\alpha_1 = \frac{3}{10(\beta_1 + \epsilon)^2}, \quad \alpha_1 = \frac{3}{5(\beta_2 + \epsilon)^2}, \quad \alpha_3 = \frac{1}{10(\beta_3 + \epsilon)^2} \quad (2.42)$$

Here again, ε is a small arbitrary number to avoid a null division, and β are the smoothness indicators expressed as:

$$\begin{aligned}
\beta_1 &= \frac{13}{12}(u_{i-1} - 2u_i + u_{i+1})^2 + \frac{1}{4}(3u_{i-1} - 4u_i + u_{i+1})^2 \\
\beta_2 &= \frac{13}{12}(u_{i-2} - 2u_{i-1} + u_i)^2 + \frac{1}{4}(u_{i-2} - u_i)^2 \\
\beta_3 &= \frac{13}{12}(u_{i-3} - 2u_{i-2} + u_{i-1})^2 + \frac{1}{4}(u_{i-3} - 4u_{i-2} + 3u_{i-1})^2 \quad (2.43)
\end{aligned}$$

Similarly, the right extrapolated value u_R is computed using the symmetric stencil.

Simulations carried out in [9] showed severe oscillations when the WENO schemes are applied to the conserved variables. In order to circumvent this problem, it has been proven efficient to reconstruct the characteristic variables, i.e. the vector **W** introduced in Equation (2.12) instead of the conserved variables. It should be noted that WENO-based schemes are not by default TVD nor even monotonicity-preserving, which can lead to spurious oscillations in case of very steep gradients. To overcome that issue, the modification proposed in [8] can be applied as an additional requirement to get a monotonicity-preserving scheme. These newly created schemes are referred as monotonicity-preserving WENO (MPWENO).

2.3.4 High-order time integration

Similarly to high-order methods detailed above, which concern spatial discretization, one should be concerned in achieving also a high-resolution method to deal with the time discretization. The first order explicit method directly comes from the formulation (2.29), where the flux at time n is directly added to the variable, i.e. :

$$\mathbf{U}^{n+1} = \mathbf{U}^{n} + \triangle t . Flux(\mathbf{U}^{n}) \tag{2.44}$$

This formulation is called Euler forward stepping. However, as the flux is computed only at time n, that method is only first order accurate. The idea is then to divide the time step $\triangle t$ into several stages to get a better accuracy. That process is called dual time-stepping (see [54]). One of the most famous range of methods implying explicit dual time stepping are the high-order Runge-Kutta (RK) methods where the flux is re-evaluated at each substage, guaranteeing a greater order of accuracy. Besides, it has been shown that this explicit time integration is particularly adapted to WENO spatial reconstruction, therefore the third-order TVD Runge-Kutta time integration is employed during all the study. The principle and related formula are detailed next.

The general formulation of a RK time integration at i-th order can be written under the form:

$$\mathbf{U}^{n+1} = \sum_{k=0}^{i-1} (\alpha_{i,k} \mathbf{U}^{(k)} + \beta_{i,k} \triangle t . Flux(\mathbf{U}^{k})) \tag{2.45}$$

where $u^{(k)}$ is the intermediate value of u computed at each substage k, $k = 0...i$ and with $\mathbf{U}^{(0)} = \mathbf{U}^n$ and $\mathbf{U}^{(i)} = \mathbf{U}^{n+1}$.

Each intermediate value $u^{(k)}$ is computed in the same manner:

$$\mathbf{U}^{(k)} = \sum_{l=0}^{k-1} (\alpha_{k,l}\mathbf{U}^{(l)} + \beta_{k,l}\triangle t.Flux(\mathbf{U}^{(l)})) \tag{2.46}$$

Gottlieb and Shu in [31] showed that in order for a RK time integration to be TVD compliant, the CFL number should obey the following criterion:

$$C \leq \min_{l,k} \frac{\alpha_{l,k}}{\beta_{l,k}} \tag{2.47}$$

Implementing a high-order RK integration requires re-computing the fluxes at every substage, thus making the computational cost much more expensive. However, it allows reaching up to 5th-order, even if practically, the implementation is more complicated and have a reduced stability range due to the restriction on the CFL mentioned above. In fact, the best compromise between accuracy and complexity, which is the most used in high-order computational methods, is the 3rd-order TVD RK method, explicited below:

$$\begin{aligned} \mathbf{U}^{(1)} &= \mathbf{U}^n + \triangle t.Flux(\mathbf{U}^{(n)}) \\ \mathbf{U}^{(2)} &= \frac{3}{4}\mathbf{U}^n + \frac{1}{4}\mathbf{U}^{(1)} + \frac{1}{4}\triangle t.Flux(\mathbf{U}^{(1)}) \\ \mathbf{U}^{n+1} &= \frac{1}{3}\mathbf{U}^n + \frac{2}{3}\mathbf{U}^{(2)} + \frac{2}{3}\triangle t.Flux(\mathbf{U}^{(2)}) \end{aligned} \tag{2.48}$$

These optimal weights have been determined by Gottlieb and Shu. As a result, the CFL restriction in order to ensure a TVD scheme expressed in Equation (2.47) is simply: $C \leq 1$. As a matter of fact, that supposed additional restriction is redundant since the maximum CFL number of an explicit scheme is 1 in any case, as explained in section 2.2.2.

3

Level set functions and related applications

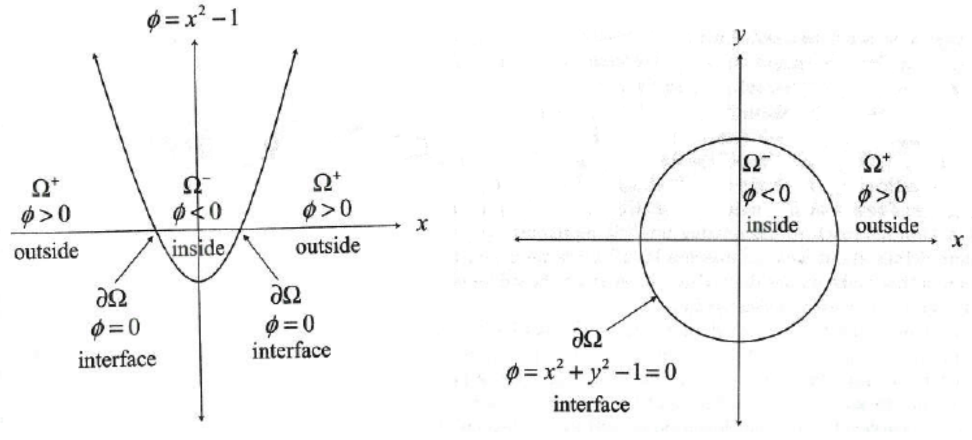
3.1 Basic concepts

3.1.1 Definition

The idea of introducing a dedicated function to track an interface between materials has first been developed by Stanley Osher and Ronald Fedkiw [23]. It consists in creating a purely geometrical function being the actual interface. Instead of being an explicit function writing down directly the points of the interface, the level-set function defines implicitly the interface as being one of its isocontour. For the sake of simplicity (and to fulfill the signed distance criteria, described later), this isocontour picked is the zero of the function, so the interface at a given time t, noted Γ, can be expressed as:

$$\Gamma(t) = \{(x, y, z) \in \Omega \mid \phi(x, y, t) = 0\} \tag{3.1}$$

where ϕ is the level-set function, and Ω the computational domain. 1D and 2D examples of level set implicit functions are represented on Figure 3.1. Level-set functions separate the computational domain in several zones, each zone corresponding to a selected material.



1D

2D

Figure 3.1: 1D and 2D examples of implicit functions

It appears that if the computational domain is of dimension n, the level-set function needs to be defined over the whole domain so its zero isocontour is of dimension n-1. Despite the apparent waste of computational time used to compute the values of ϕ throughout the domain, level-set functions provide tools which will be further used to retrieve all the relevant geometric information (in a much easier way than if the function was defined explicitly).

In addition, level-set functions satisfy another crucial property: they are signed distance functions, meaning their values in a given point in space is the distance to the interface, weighted by the signum function depending on the relative position to this interface (i.e. if the point lies at the right or the left of the interface). Mathematically, a necessary condition for a function to be a signed distance function is simply:

$$\mid \nabla\phi \mid = 1 \tag{3.2}$$

also referred as the Eikonal equation. This property is illustrated on Figure 3.2.

It is consequently appropriate to attribute a material to a sign of the level-set function. Theoretically, and in several applications, n-1 level set functions are enough to determine the interface location between n materials. Indeed, if one defines A^+ the region where $\phi(x, y, t) > 0$ and $A^-$the region where $\phi(x, y, t) < 0$,

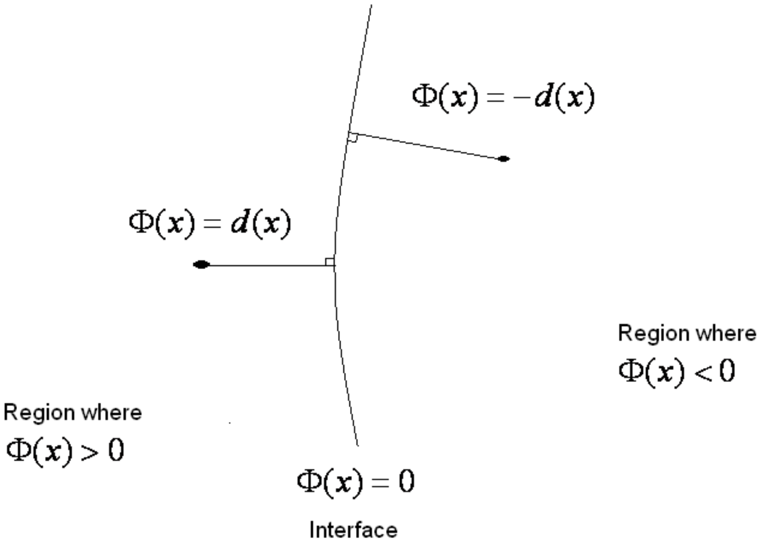


Figure 3.2: Signed distance function property

A^+ would be occupied by material 1 and A^- would be occupied by material 2. However, practically, it is more convenient to use as many level-set functions as materials involved, especially when vacuum regions can appear (break-up problems). Therefore, in every simulation carried out during this study, every material has its own associated level-set function, and lies in the region where it is positive.

Classical shapes such as circles or lines for instance are given straight away by an appropriate function, as previously shown in Figure 3.1.

In a more general case, and a random initial geometry, the level-set function will be harder to initialize, and will need a special treatment based on numerical methods, which will be detailed later.

3.1.2 Equations related

Given that the interface moves at a local velocity $\mathbf{V}$, the implicit level-set function moves accordingly, and is naturally advected by:

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} + \mathbf{V}.\nabla \phi(\mathbf{x}, t) = 0 \tag{3.3}$$

That PDE and its characteristics, along with the resolution methods available have been widely developed in Section 2. Even if theoretically only the interface moves, Equation (3.3) needs to be solved over the whole domain Ω. In fact, it only needs to be solved in a band of cell near the interface. The bandwidth varies based on the numerical method used, and on the desired level of accuracy.

A second equation is added throughout the simulations. Indeed, having the property of signed distance function is a very useful tool for the next methods. However, even if that property holds true initially, it is not enforced as the level-set function is advected, and is consequently lost over time. It has indeed been noticed that solving (3.3) only eventually results in a wave compression in parts of the solution, and a strong smearing in others. This phenomenon arise numerical difficulties when solving the advection equation, producing instabilities and oscillations, and therefore unacceptable errors in accessing the required geometry information. Consequently, a re-initialization procedure has to be applied to the level-set function. To comply with the Eikonal equation, the following equation is solved until convergence [25] (τ being an artificial time):

$$\frac{\partial \phi}{\partial t} + S(\phi).(|\nabla \phi| - 1) = 0 \tag{3.4}$$

where $S(\phi)$ is a sign function, to be defined. Reinitialization procedures have been investigated, and will be detailed in the next section.

3.2 Reinitialisation procedures

As explained in the latter section, a reinitialization procedure has to be applied. A first remark is to notice that the reinitialization does not absolutely need to be performed at every time step. Indeed, the "compression-smearing" phenomenon mentioned earlier does not occur from one time step to another. It usually requires several time steps before it becomes preoccupying and thus the reinitialization only needs to be performed every N timesteps, N being problem-dependent.

Two key-points need to be addressed when looking at Equation (3.4): the definition of $S(\phi)$ and the approximation of $|\nabla \phi|$.

$S(\phi)$ is a sign function, to keep the sign of ϕ throughout the resolution of Equation (3.4). A first definition of $S(\phi)$ is as a result simply:

$$S(\phi) = \begin{cases} 1 & if\ \phi > 0 \\ -1 & if\ \phi < 0 \end{cases} \tag{3.5}$$

If that works rather well when ϕ is relatively smooth, the interface might move incorrectly if ϕ is too steep. Numerical tests carried by Peng and Sussmann [60] demonstrate that introducing the smoother smeared function as:

$$S(\phi) = S(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + (\Delta x)^2}} \tag{3.6}$$

where ϕ_0 is the level-set function before the reinitialization procedure is applied, greatly reduces the inaccuracies.

Another remark is that the sign function depends here only on the level-set function before its reinitialization.

Concerning the discretization of $|\nabla\phi|$, the Godunov Hamiltonian is given by:

$$|\nabla\phi| \sim H(\phi) \tag{3.7}$$

with:

$$H(\phi) = H(D_x^+\phi, D_x^-\phi, D_y^+\phi, D_y^-\phi, D_z^+\phi, D_z^-\phi) \tag{3.8}$$

where $D_i^\pm$ is the finite difference (forward and backward) approximation of the derivative of ϕ with respect to the i-th direction. High-order approximation can be used in order to compute the latter, including the WENO methods described in Chapter 2.

More exactly:

$$H(a,b,c,d,e,f) = \begin{cases} \sqrt{max(a_+^2, b_-^2) + max(c_+^2, d_-^2) + max(e_+^2, f_-^2)} & if\ \phi > 0 \\ \sqrt{max(a_-^2, b_+^2) + max(c_-^2, d_+^2) + max(e_-^2, f_+^2)} & if\ \phi < 0 \end{cases} \tag{3.9}$$

where $a_+ = max(a, 0)$ and $a_- = min(a, 0)$.

After the spatial discretisation outlined above, Equation (3.4) can be temporally integrated using a high-order TVD Runge-Kutta described in Chapter 2.

These reinitialization procedures have an important drawback though: they artificially move the interface, changing the zero isocontour of the level-set function and then leading to unphysical deformations. That phenomenon is particularly well demonstrated in [60, 56] with the Zalesak testcase [81] where a fixed geometry is deformed during a uniform convection under several steps of reinitialization.

To overcome that issue which is particularly sensitive when large deformation occur, some authors proposed successively in [35, 71] then in [36] a constrained reinitialization equation, consisting in adding a forcing term $\mathbf{F}$ on the right hand side of Equation (3.4), behaving like a source term. Equation (3.4) becomes:

$$\frac{\partial \phi}{\partial t} + S(\phi).(|\nabla \phi| - 1) = \beta \mathbf{F} \tag{3.10}$$

where β is a weighting factor, set to 0.5.

That forcing term should be non-zero near the interface, and zero everywhere else. One should first identify the domain near the interface, noted $\Lambda(\supset \Gamma)$, and defined as:

$$\Lambda = \{x_{(i,j,k)} \epsilon \Omega | (\phi_{i',j,k}\phi_{i,j,k} \leq 0) \cup (\phi_{i,j',k}\phi_{i,j,k} \leq 0) \cup (\phi_{i,j,k'}\phi_{i,j,k} \leq 0)\} \tag{3.11}$$

where i', j', and k' representing the intervals (i-1,i+1), (j-1,j+1), (k-1,k+1) respectively. In other words, Λ is identified by a change of sign from a cell to another. Actually, this subset needs to be computed at every time step during the time integration of Equation (3.10), so that domain will be, at a given time n, be referred as Λ^n.

A Dirac function is subsequently defined, noted λ so that:

$$\lambda_{i,j,k} = \begin{cases} 1 & if\ x_{i,j,k} \in \Lambda^n \\ 0 & otherwise \end{cases} \tag{3.12}$$

The discrete forcing term is computed as follows:

$$\mathbf{F}^n_{i,j,k} = \lambda_{i,j,k} \mathcal{F}^n_{i,j,k} \tag{3.13}$$

where:

$$\mathcal{F}^n_{i,j,k} = \frac{1}{\Delta x}(\gamma^n_{i,j,k} - \phi^n_{i,j,k}) \tag{3.14}$$

$\gamma^n_{i,j,k}$ is a target value, and its computation will be detailed below. As a remark, the Δx factor introduced in Equation (3.14) is present to remove the inherent lengthscale of the level-set function, which scales with Δx (refer to the signed distance property). First, let us denote $S_{i,j,k}$ the set of neighboring points affiliated to every point $x_{i,j,k} \in \Lambda^n$ with a change of sign in the level-set field, so:

$$S_{i,j,k} = \left\{\alpha_{i,j,k} | \phi_{i,j,k}\phi_{\alpha_{i,j,k}} \leq 0\right\} \tag{3.15}$$

where $\alpha_{(i,j,k)} \subset [i+1 : i-1, j, k] \cup [i, j+1 : j-1, k] \cup [i, j, k+1 : k-1]$, i.e. the set of all neighboring points.

If we note $N_{i,j,k} = card(S_{i,j,k})$ the number of elements of $S_{i,j,k}$, and m_α the element number of coordinates α, the expression of the target value $\gamma^n_{i,j,k}$ can be computed according to the following:

$$r_{i,j,k} = \frac{\phi^0_{i,j,k}}{\sum\limits_{m_\alpha=1}^{N_{i,j,k}} \phi^0_\alpha} \tag{3.16}$$

where $r_{i,j,k}$ is an intermediate variable to obtain:

$$\gamma^n_{i,j,k} = r_{i,j,k} \sum_{m_\alpha=1}^{N_{i,j,k}} \phi_\alpha \tag{3.17}$$

Note that in Equation (3.16), only the values of ϕ before reinitialization (noted ϕ^0) are used, while the values in Equation (3.17) must be taken at the local time n of the reinitialization. Equation (3.14) finally becomes:

$$\mathcal{F}^n_{i,j,k} = \frac{1}{\Delta x}(r_{i,j,k} \sum_{m_\alpha=1}^{N_{i,j,k}} \phi_\alpha - \phi^n_{i,j,k}) \tag{3.18}$$

Therefore, Equation (3.10) is solved until reaching a prescribed convergence threshold. Among the several techniques available, Hartmann's method has been picked because of its good compromise between accuracy, its straightforward implementation and its ease to extend in three dimensions.

3.3 Geometry reconstruction

Another matter of importance when using level-set functions and their related applications is the ability to reconstruct arbitrary surfaces given a data set of points. If simple shapes are easy to model with level-set functions (like circles or lines) by being explicited with one (or few) functions throughout the entire domain, more complex shapes require a specific treatment. It is particularly relevant when using crossed softwares for design and simulations. Indeed, one can assume that a geometry has been created independently with a CAD software, the latter sending the geometry to the simulation software. It is assumed that the CAD software provides the location of points constituting the geometry. In order to use level-set related methods, one has to be able to reconstruct the initial level-set field, so the signed distance function whose zero isocontour represents the surface.

To achieve an accurate reconstruction, the algorithm developed by Zhao and Osher in [84, 83] is used. It consists in minimizing the gradient flow of a surface energy function. The minimal surface obtained will be smooth and accurate, i.e. higher order than a simple piecewise linear approximation. The method is fast, able to handle large topological changes, and proves very efficient for three-dimensional surfaces, see the results of [25].

The reconstruction model relies on three steps:

- Computing the distance of every cell center to the data set points
- Finding a good initial guess for the surface
- Solving the PDE of the gradient flow presented below

Let us introduce first the energy function of a surface S, denoted $\mathbf{E}(S)$. It is defined as:

$$\mathbf{E}(S) = (\int_S d^p(\mathbf{x}) ds)^{1/p}, \quad 1 \leq p \leq \infty \tag{3.19}$$

where p acts as a weighting factor, corresponding mathematically as the L^p norm of the potential on S. $d(\mathbf{x})$ is the distance from the point to the data set point.

It ensues that the gradient flow can be expressed as:

$$\frac{dS}{dt} = -(\int_S d^p(\mathbf{x})ds)^{\frac{1}{p}-1})d^{p-1}(\mathbf{x}).[\nabla d(\mathbf{x}).\mathbf{n} + \frac{1}{p}d(\mathbf{x})\kappa]\mathbf{n} \tag{3.20}$$

where $\mathbf{n}$ is the normal pointing outward of the surface, and κ its curvature. The steady state of Equation (3.20) leads to the Euler-Lagrange equation for the gradient flow:

$$d^{p-1}(\mathbf{x}).[\nabla d(\mathbf{x}).\mathbf{n} + \frac{1}{p}d(\mathbf{x})\kappa] = 0 \tag{3.21}$$

It shows a balance between the attraction $\nabla d(\mathbf{x}).\mathbf{n}$ and the surface tension $d(\mathbf{x})\kappa$. Besides, the scaling in $d(\mathbf{x})$ makes the reconstructed surface more flexible in regions where the sampling density is high and more rigid when the density is lower. Concerning the parameter p, values are usually set to 1 or 2; more details can be found in [83]. For a simpler expression in the following, we set p=1. Equation (3.20) thus becomes:

$$\frac{dS}{dt} = -[\nabla d(\mathbf{x}).\mathbf{n} + d(\mathbf{x})\kappa]\mathbf{n} \tag{3.22}$$

Transforming (3.22) into a PDE with the sought level-set field requires to write that:

$$S(t) = \{\mathbf{x}|\phi(\mathbf{x},t) = 0\} \tag{3.23}$$

So, the total derivative of ϕ can be written thanks to the partial derivative (and noticing that the zero level-set has the same motion law as the moving surface):

$$\frac{d\phi(S(t),t)}{dt} = \frac{\partial\phi}{\partial t} + \frac{dS(t)}{dt}.\nabla\phi = 0 \tag{3.24}$$

ϕ being the signed distance function, it is immediate to note that:

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \text{ and } \kappa = \nabla.(\frac{\nabla\phi}{|\nabla\phi|}) \tag{3.25}$$

Combining Equations (3.25), (3.24), (3.22) leads to the following final PDE for ϕ:

$$\frac{\partial\phi}{\partial t} = |\nabla\phi| \left[\nabla d.\frac{\nabla\phi}{|\nabla\phi|} + d\nabla.\frac{\nabla\phi}{|\nabla\phi|}\right] \tag{3.26}$$

which is the gradient flow equation for the level set formulation, to be solved until convergence as the final step of the geometry reconstruction.

As an illustration of the whole procedure, a 2D example is presented. It consists in 20 points randomly spaced in a 50x50 grid of dimensions [(0,1);(0,1)]. The geometry is obviously very rough compared to the grid size, the number of points being low, but it will demonstrate well the different steps below.

Computing distance to the data

The distance to the data set satisfies the Eikonal equation (3.2) and is initialized knowing that the values of $d(\mathbf{x})$ are zero whenever $\mathbf{x}$ belongs to the surface so:

$$|\nabla d(\mathbf{x})| = 1 \text{ and } d(\mathbf{x} \in S) = 0 \tag{3.27}$$

The following will present the algorithm in 2D for the sake of simplicity. The full 3D implementation is described in Appendix A. Assuming a uniform squared grid ($\Delta x = \Delta y$=h), Equation (3.27) becomes discretized as:

$$[(u_{i,j} - x_{min})^+]^2 + [(u_{i,j} - y_{min})^+]^2 = h^2 \tag{3.28}$$

where:

$$(a)^+ = \begin{cases} a & if\ a > 0 \\ 0 & otherwise \end{cases} \tag{3.29}$$

And:

$$x_{min} = min(u_{i+1,j}, u_{i-1,j}) \quad xy_{min} = min(u_{i,j+1}, u_{i,j-1}) \tag{3.30}$$

It appears clearly that this equation is solved by propagation of the original known information (the data set points) through the other cells and the consecutive updates at each Gauss-Seidel operation. It can be shown in [83] that Equation (3.28) converges, and that besides an exact formula can be provided for $u_{i,j}$:

$$u_{i,j} = \begin{cases} min(x_{min}, y_{min}) + h & if\ |x_{min} - y_{min}| \geq h \\ \frac{x_{min}+y_{min}+\sqrt{2h^2-(x_{min}-y_{min})^2}}{2} & if\ |x_{min} - y_{min}| < h \end{cases} \tag{3.31}$$

Each sweep at every iteration must be performed in a different order, so the updates come alternatively from the left, the right, the top and the bottom. It usually takes for sweeps (each with a different order) plus an additional arbitrary sweep, 5 sweeps in total (in 2D, 8+1=9 in 3D) to reach convergence of the values $u_{i,j}$ in Equation (3.28).

On the example mentioned above, the contours of $d(\mathbf{x})$ are represented on Figure 3.3 after the application of the latter algorithm. It has the expected shape, i.e. concentric circles around the data set points with a zero value at these.

Finding a good initial guess

An initial surface to start with must at least contain all the data set points. A good initial guess to solve the PDE (3.26) is fundamental for getting a fast convergence in one hand, and to avoid spurious oscillations resulting in a jagged shape of the reconstructed surface on the other hand. One of the simplest choices to achieve this is to take the rectangular box containing all these points, and then progressively reduce that rectangular box to a more precise geometry. It requires being able to identify the exterior points of the surface to those inside, to make sure that the kept geometry still contains all the data set points. The following details the procedure.

Each point of the domain is attributed a label, *interior* or *exterior*. Starting with the rectangular box, every point outside is considered as *exterior*, and the others as *interior*. Consequently, the firstly formed exterior region is a subset of the true exterior region to the real surface. Now, a third label is introduced, the *temporary* points. If an *interior* point has a neighbour which is an *exterior*

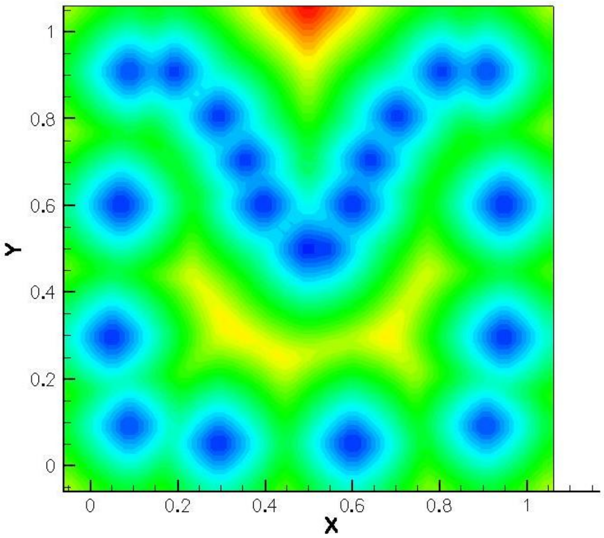


Figure 3.3: Distance to the data set points

point, it is qualified as *temporary*. Once all the *temporary* points have been identified, they are classed in an array according to their distance value. The *temporary* point with the largest distance value is outlined. If one of its *interior* neighbours has a larger distance value, that *temporary* point is taken out of the array of *temporary* points, and is considered as a final point for the boundary of our initial guess. If none of its *interior* neighbours has a larger distance value, the point is still removed from the array, but now considered as an *exterior* point, and in addition, all its interior neighbours are added to the temporary array. That procedure is repeated until all the *temporary* points left in the array have a value lesser than the grid size, which means that every boundary point is close enough to the data set. The set of boundary points forms the initial surface guess. Finally, the initial signed distance function is obtained by weighting the distance values by 1 or -1 if the cell center is an *interior* or *exterior* point, respectively.

The initial surface guess of the case set out earlier is represented on Figure 3.4. It shows a rough, but fair approximation of the true surface. Nevertheless, it constitutes a much better guess than a simple rectangular box.

Solving the PDE

Now that a decent initial condition has been computed, Equation (3.26) can be solved using the numerical tools presented in Chapter 2. High-order methods

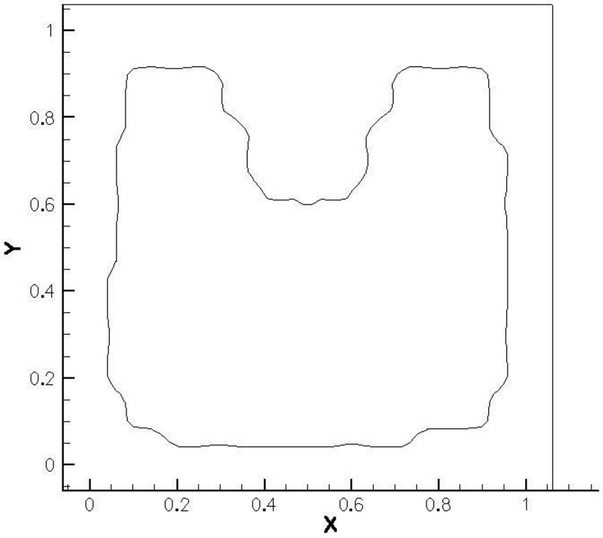


Figure 3.4: Initial surface guess

are used for both time integration and space reconstruction, e.g. WENO and TVD Runge-Kutta. Besides, as getting a signed distance function is wanted, reinitialization procedures are applied during the resolution of the PDE. The resulting reconstructed surface is smooth, so is the level-set field obtained, as shown in Figures 3.5 and 3.6.

3.4 Ghost fluid methods

3.4.1 Original Ghost Fluid Method

Overview of the technique

The original formulation of the Ghost Fluid Method (GFM) is from Fedkiw [23]. It is based on creating numerically a ghost material in a band of cells $\mathfrak{B}$ near the interface (which bandwidth is determined by the desired accuracy) through the use of ghost cells. Every point of $\mathfrak{B}$ has a real cell associated, containing the state values of the real material, and also a ghost cell, which state values corresponds to a ghost material, to be defined.

In Fedkiw's approach, the state values of the ghost cells are those taken from the real material in term of pressure and velocity, but the entropy taken is the one from the material located at the other side of the interface. That process

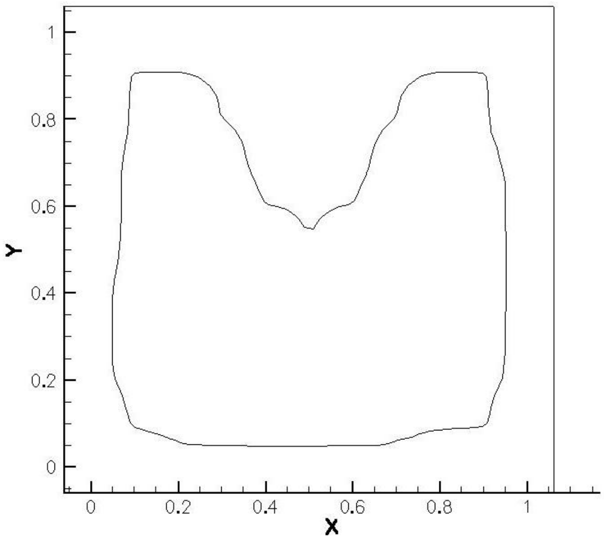


Figure 3.5: Final reconstructed surface

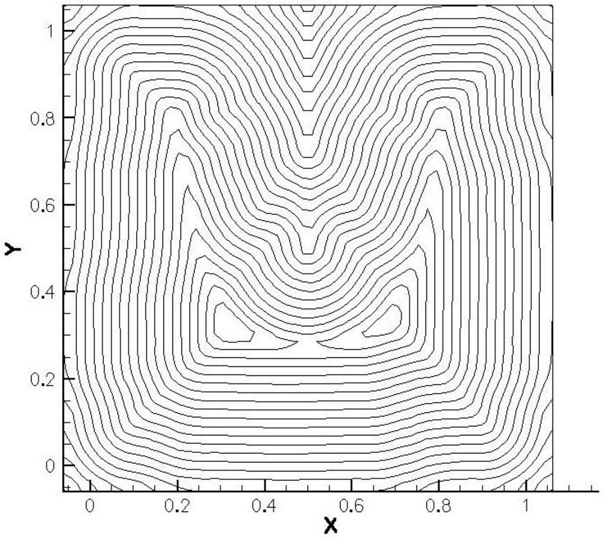


Figure 3.6: Initial Level-set field

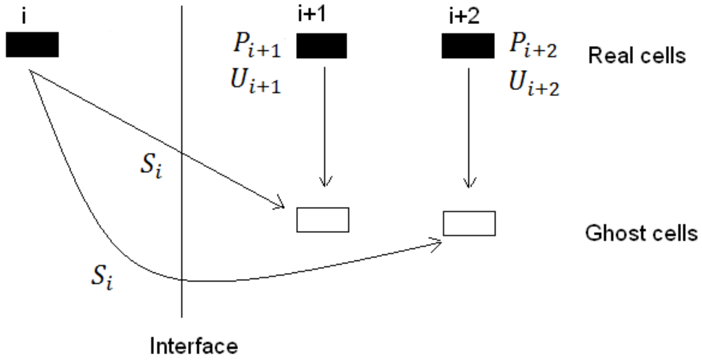


Figure 3.7: Original GFM

is illustrated in 1D on Figure 3.7. A remark can be made at that point: originally, that method has been developed for fluid dynamics, thus the notion on pressure in the cells (and also the name of the method; however, the GFM terms will be abusively used even when dealing with solid/fluid interaction). However, that notion can be extended to other materials. When dealing later with solids, the relevant variable corresponding to the fluid pressure will be the normal stresses. As a result, what follows in that section remains general and independent of the materials considered. The actual applications to real materials will be the content of Chapters 4 and 5.

Symmetrically, ghost cells are defined at node *i*, using pressure and velocity of the material lying in node *i* ($P_i^{ghost} = P_i$, $U_i^{ghost} = U_i$), and the entropy of the material lying in node *i+1* ($S_i^{ghost} = S_{i+1}$).

Fedkiw also proposed a modification when defining the ghost cells, called the entropy fix, in order to reduce overheating errors. It consists in fixing the entropy of node *i* (and so the entropy of the ghost cells at nodes *i*+1, *i*+2...) as being the entropy of node *i*-1. This leads to a slightly modified definition, as illustrated in Figure 3.8. It helps reducing numerical overheating errors that can arise at the interface. See [23] for more details.

Extension to multidimensions

The method presented above has been explicited in 1D cases. However for the extension in multidimensions, more than one velocity component are involved, and consequently, a consistent extrapolation procedure to fill the ghost cells needs to be specified. Basically, the extrapolation of the state values is carried out in

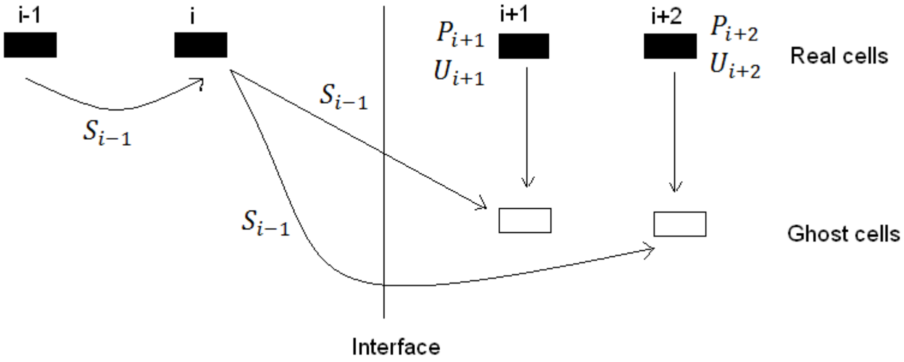


Figure 3.8: GFM with entropy fix

the normal direction. This extrapolation is done by solving the following PDE:

$$\frac{\partial u}{\partial \tau} \pm \mathbf{n}.\nabla u = 0 \tag{3.32}$$

where u is the quantity to be extrapolated, τ is a fictitious time, and $\mathbf{n}$ is the normal vector to the interface, always pointing from the negative level set field to the positive. Therefore, the sign in Equation (3.32) is determined by the direction of the extrapolation.

The normal $\mathbf{n}$ is simply defined using the level set function, as already mentioned in Equation (3.25):

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|} \tag{3.33}$$

The procedure is then identical than in the 1D case: the pressure in the ghost cell will be the one from the real fluid, so as the normal velocity, and the tangential velocity and entropy will be the ones coming from the extrapolation of the other fluid.

Once the ghost cells have been defined, the multimaterial problem is reduced to several "single component" problems that are easier to handle. Each material is indeed solved independently as one material defined in the cells where he is actually lying and in $\mathfrak{B}$ where it is defined through the ghost values.

3.4.2 Modified Ghost Fluid Method (MGFM)

Overview of the technique

In the original GFM, conditions of the ghost material at the interface were simply defined by using the pressure and velocity of one material, while having the entropy of the other one. In other terms, the ghost material is defined as having the same characteristics (pressure, velocity) as the real one, but appearing thermodynamically to be the other material lying across the interface (entropy). This simple approach is not accurate enough however in case of strong shocks impacting interfaces, due to the sudden jump of quantities. In such situations, the interface conditions must be predicted and computed separately, before defining the ghost cells. The MGFM uses the same concepts as in the GFM, in terms of using ghost cells and level-set functions. The change consists in the values input in the ghost cells. Basically, a Riemann problem shall first be solved at the interface, whose left and right ICs are respectively the values at nodes i and $i{+}1$, giving the right and left values for pressure, density, velocity and entropy at the interface. These values will then be injected in the ghost cell adjacent to the interface. That new method is illustrated in Figure 3.9, where the upper index I denotes the interface. It is interesting to note that Figure 3.9 assumes different left and right values obtained after the resolution of the Riemann problem for pressure and velocities. However, physical interfacial conditions and characteristics analysis can make these values non-distinct. For instance, between two fluids, continuity exists at the interface for both pressure and velocity, so $U_R^I = U_L^I$ and $P_R^I = P_L^I$. That method proved to be more robust and accurate in case of strong shocks and very high quantities ratios of variables, whether density, pressure or velocity.

Extension to multidimensions

Similarly to the GFM, extrapolation is needed to fill the ghost cells. The procedure is the same than the one described in section 3.4.1. Another matter of importance is that a change of referential is required to simplify the resolution of the Riemann problem. Indeed, it is preferable to set the problem in the referential whose axis corresponds to the normal and tangential direction with respect to the interface. It allows reducing the 3D Riemann problem to a 1D problem. As a result, the vector of conserved variables is firstly rotated in the latter referential, and rotated back in the Cartesian referential after resolution of the Riemann problem.

It should be noted than nor the original GFM nor the MGFM are fully

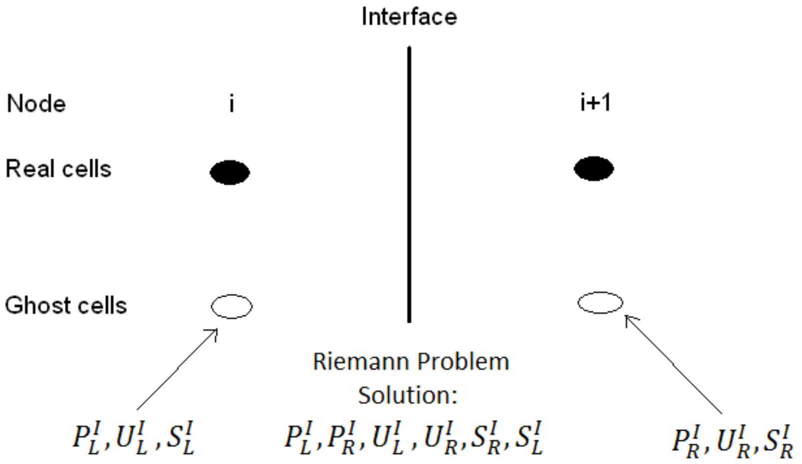


Figure 3.9: The Modified Ghost Fluid Method

conservative schemes: indeed, every single cell is considered as fully containing material 1 or material 2. Errors of conservation can therefore naturally arise into cut cells, when advecting the level set equation through the domain. The authors in [23] have discussed this problem, and shown that this error in conservation could be reduced by refining the grid. However, although this errors are not relevant in shock tube problems, in the case of strong shocks impacting interfaces, not only in terms of mass losses (which needs to be considered, depending of the application dealt with), but also in an inaccuracy about the shock and interface location [45].

3.5 The conservative level set method

To overcome the issues mentioned above concerning the lack of conservation in the GFM and MGFM, a further development needs to be carried out. It consists in treating specifically the mixed cells (being the origin of the conservation errors) in order to avoid the rough approximation of having full cells throughout the entire domain. A first attempt has been proposed by Hu [38] or Sheu [69] and successfully implemented in 2D. Wherever mixed cells are identified, a merging and redistribution procedure is applied in order to conserve the quantities present in the mixed cells. It involves the use of neighbouring cells as target cells to act as a receptacle of variables by merging it to the mixed cell. Hereby, that concept is further developed and implemented in 3D. The following presentation concerns an arbitrary PDE under the form of Equation (2.2), and is thus independent on the material concerned. For convenience, Equation (2.2) is reminded here:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial F^i(\mathbf{U})}{\partial x_i} = S \tag{3.34}$$

where $\mathbf{U}$ is the vector of considered variables (for instance: density, velocity, pressure, stress, entropy...), $F^i = (f_1, ..., f_n)^i$ are flux functions in the x_i direction, and S is a vector containing source terms.

3.5.1 Accessing the geometry information

The first step is to introduce the volume fraction of a material, i.e. the relative volume of the material with respect to the volume of each cell:

$$\alpha^{\eta}_{i,j,k}(t) = \frac{\mathcal{V}^{\eta}_{i,j,k}(t)}{V_{i,j,k}} \quad (0 \leq \alpha^{\eta} \leq 1) \tag{3.35}$$

where η is the component number (dropped later for clarity), $\mathcal{V}^{\eta}_{i,j,k}$ the volume it occupies in a given cell at a given time, and $V_{i,j,k}$ the volume of the cell, which is constant in time if the grid is non-moving ($V_{i,j,k} = \Delta x \Delta y \Delta z$ in the case of uniform grids).

Let us denote $\Gamma^{\eta}(t)$ the region occupied by the material η. Integrating Equation (3.34) over a computational cell and applying Gauss's theorem gives:

$$\begin{aligned}\int\limits_{t_1}^{t_2} dt \int\limits_{V_{i,j,k}\cap\Gamma} dxdydz \frac{\partial \mathbf{U}}{\partial t} + \int\limits_{t_1}^{t_2} dt \int\limits_{\partial V_{i,j,k}\cap\partial\Gamma} dxdydz\, F.\mathbf{n} \\ = \int\limits_{t_1}^{t_2} dt \int\limits_{V_{i,j,k}\cap\Gamma} dxdydz\, S \end{aligned} \tag{3.36}$$

where $\partial V_{i,j,k}$ denoted the boundary of the mesh element, whilst $\partial V_{i,j,k} \cap \partial\Gamma$ is an arbitrary polyhedron possibly formed through intersection of an interface plane with the computational element (physically corresponding to the shape of the material inside the cell). This is illustrated in Figure 3.10. Integration over the former includes only the cell wall fluxes, whilst the latter may contain all or some of these fluxes in addition to material interface fluxes acting on the

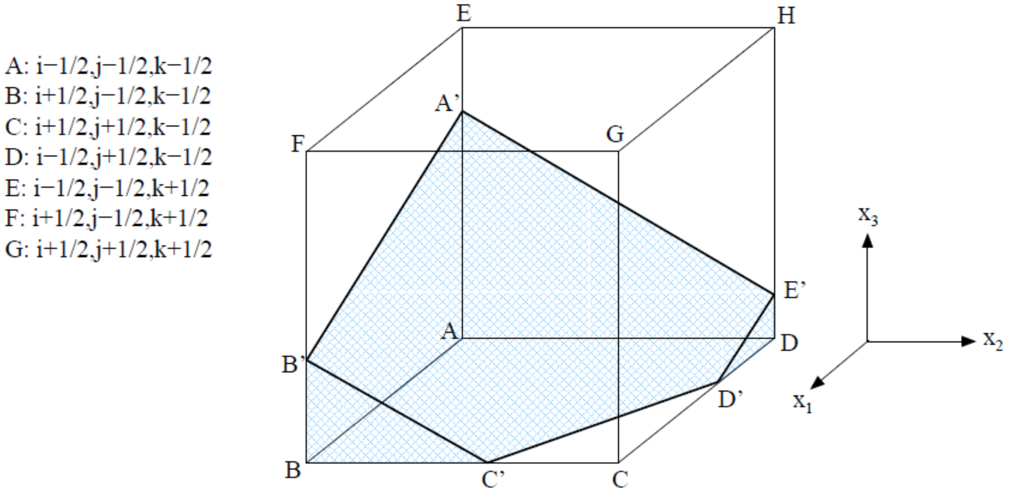


Figure 3.10: Illustration of the volume generated within a mixed cell through intersection of the interface plane with the cell volume. Primed letters indicate intersection points.

intersecting interface plane for the case of cut cells. The discrete representation of the surface integral in Equation (3.36) shaped by N_f faces can be written:

$$\int_{t_1}^{t_2} dt \int_{\partial V_{i,j,k} \cap \partial \Gamma} dxdydz\, F.\mathbf{n} = \int_{t_1}^{t_2} dt \left[\sum_{m}^{N_f} (\mathcal{A}(t)\, F.\mathbf{n})_m \right] \tag{3.37}$$

where for the m-th face $\mathbf{n}_m$ is the outwards pointing unit normal, and $\mathcal{A}_m(t)$ is the aperture: the transient area on which the respective flux applies. In [52] an accurate representation of the apertures was proposed through the construction of space-time areas, but introduced considerable additional complexity when applied to three spatial dimensions. Similarly to [38] it is found that evaluation of the apertures from data at the current time level in an explicit method is sufficient for those examples considered here. Using these results, a forward Euler representation of Equation (3.36) can be written:

$$\begin{aligned} V_{i,j,k}((\alpha\mathbf{U})^{n+1}-(\alpha\mathbf{U})^{n})_{i,j,k} &= \Delta t[-(\mathcal{A}F^{1})_{i+1/2,j,k}+(\mathcal{A}F^{1})_{i-1/2,j,k}- \\ &\quad (\mathcal{A}F^{2})_{i,j+1/2,k}+(\mathcal{A}F^{2})_{i,j-1/2,k}- \\ &\quad (\mathcal{A}F^{3})_{i,j,k+1/2}+(\mathcal{A}F^{3})_{i,j,k-1/2}- \\ &\quad (\mathcal{A}\mathbf{n}.F)_{b}+(\alpha V)_{i,j,k}S_{i,j,k}] \end{aligned} \tag{3.38}$$

where F^b is used to denote the vector of fluxes acting on the material interface and all terms on the right-hand-side are assumed to be evaluated from data at the current time level t^n. It is straightforward to see that for cells with $\alpha^n_{i,j,k} = 1$, the discretisation Equation (3.38) reduces to the standard form for a single material calculation on a Cartesian grid. It is remarked that the method assumes a sharp interface description and hence for a component η, the state at any given time will be stored only within cells with $\alpha_{i,j,k} \geq 0$. Thus within mixed cells shared by multiple components, the partial state of each will be stored rather than mixture quantities. It will be described in detail how each component is updated independently following the interface interaction calculation. So as to avoid adjusting the numerical method in the region of the interface, a band of ghost cells outside a given material's domain is defined to have an extrapolated state such to provide adequate data to complete the applied numerical stencil in the boundary regions.

Within each cell, several geometric quantities exist in Equation (3.38) that remain to be determined, specifically $\mathcal{A}_m, n_m, m = 1, ..., N_f$ and $\mathcal{V}$. It can be shown that the problem can be reduced to determining only the values of apertures at the cell faces and that all other geometric parameters are either related to these or determined in the process. For any cell where $0 < \alpha_{i,j,k} \leq 1$, the volume of material can be found using the general formula for arbitrary polyhedra [30]:

$$V = \frac{1}{3}|\sum_{m=1}^{N_f}(\overline{\mathbf{x}}.\mathbf{n})_m\mathcal{A}_m| \tag{3.39}$$

where A, n, and N_f have the same definition as before. Since the bounding sides are assumed to be planer, the vector coordinate $\overline{\mathbf{x}}_m$ can be taken as any point residing on the respective plane. For cells where $\alpha^\eta = 1$, the complexity of the problem is reduced considerably and the only faces that need be considered in Equation (3.39) are those of the cell boundary. Using the node coordinates $x_{i,j,k}$, it is thus straightforward to compute the cell boundary normals and apertures: $\mathcal{A}_{m\pm1/2} = A_{m\pm1/2}$, $m = i, j, k$ where $A_{m\pm1/2}$ is the area of the face considered. For cells which are cut by an interface plane, $0 < \alpha_{i,j,k} < 1$, the volume of material is bounded by the (possibly cut) cell face apertures, where now $\mathcal{A}_{m\pm1/2} \leq A_{m\pm1/2}$, $m = i, j, k$, and the interface plane (Figure 3.10). The area of the interface plane can be found using (from [59]):

$$\mathcal{A}_b = \sqrt{(\mathcal{A}_{i+1/2,j,k} - \mathcal{A}_{i-1/2,j,k})^2 + (\mathcal{A}_{i,j+1/2,k} - \mathcal{A}_{i,j-1/2,k})^2 + (\mathcal{A}_{i,j,k+1/2} - \mathcal{A}_{i,j,k-1/2})^2} \tag{3.40}$$

and the unit normal vector pointing outwards from the interface can be computed from:

$$\mathbf{n}_b = \frac{1}{\mathcal{A}_b} \begin{pmatrix} \mathcal{A}_{i-1/2,j,k} - \mathcal{A}_{i+1/2,j,k} \\ \mathcal{A}_{i,j-1/2,k} - \mathcal{A}_{i,j+1/2,k} \\ \mathcal{A}_{i,j,k-1/2} - \mathcal{A}_{i,j,k+1/2} \end{pmatrix} \tag{3.41}$$

Thus, it is clear that irrespective of the volume of material within a cell, the only geometric information that remains to be determined is that associated with the cell face apertures, with all other geometric quantities being determined from these. As it has already been mentioned, computing these parameters for each cell face not intersected by the interface plane is a trivial task. However, parameters of those faces cut by the interface are slightly more involved.

It highlights the need of the signed distance property of the level-set field. Indeed, the combination of distance functions can be used to define any point on the interface in a given cut cell. Hence, the required apertures in Equation (3.38) can be considered functions of the spatial coordinates and level set field:

$$\mathcal{A} = \mathcal{A}(\mathbf{x}, \phi(\mathbf{x})) \tag{3.42}$$

which can be subsequently be used to define the material volume through Equation (3.39), volume fraction through Equation (3.35) and, for cut cells, the interface geometry with Equations (3.40) and (3.41). Furthermore, the sign of the level-set functions can be used to determine the orientation of materials within the mixed cells.

For each cell edge, regardless of whether it is intersected by the interface, the coordinates of any node which is a vertice of the material volume are required. A positive value of the nodal value of the level-set function for the respective material will indicate this. Furthermore, whether or not a cell edge is intersected by an interface can be determined through the values of the level-set function at the cell nodes. The nodal values are taken to be the arithmetic mean of the surrounding cell centered values (it is reminded here that using a FVM, the values of variables are stored in cell centers):

$$\begin{aligned}
\phi_{i+1/2,j-1/2,k+1/2} &= (\phi_{i,j,k} + \phi_{i,j-1,k} + \phi_{i,j-1,k+1} + \phi_{i,j,k+1} + \\
&\quad \phi_{i+1,j,k} + \phi_{i+1,j-1,k} + \phi_{i+1,j-1,k+1} + \phi_{i+1,j,k})/8 \\
\phi_{i+1/2,j+1/2,k+1/2} &= (\phi_{i,j,k} + \phi_{i,j,k+1} + \phi_{i,j+1,k+1} + \phi_{i,j+1,k} + \\
&\quad \phi_{i+1,j,k} + \phi_{i+1,j,k+1} + \phi_{i+1,j+1,k+1} + \phi_{i+1,j+1,k})/8 \\
&\;\;. \\
&\;\;. \\
&\;\;. \\
\phi_{i-1/2,j+1/2,k-1/2} &= (\phi_{i,j,k} + \phi_{i,j+1,k-1} + \phi_{i,j,k-1} + \phi_{i,j+1,k} + \\
&\quad \phi_{i-1,j,k} + \phi_{i-1,j+1,k} + \phi_{i-1,j+1,k-1} + \phi_{i-1,j,k-1})/8
\end{aligned} \tag{3.43}$$

For each cell $I_{i,j,k}$, if all the nodal values are positive, then $\alpha_{i,j,k} = 1$ and all apertures equal the cell face areas; if all values are negative, then $\alpha_{i,j,k} = 0$ and all the apertures are zero (the cell is empty regarding the material considered). If the signs are mixed then the cell is intersected by the interface plane. Each cell face is then treated in turn. If all the nodal values of the level-set field for the cell face are found to be positive then the cell face aperture is equal to the cell face area: $\mathcal{A} = A$. If all the values are negative, then $\mathcal{A} = 0$. If there is a sign change in any values then the cell face must be intersected by the interface plane and the first requirement is to determine the coordinates of all vertices of the aperture. Using the mixed cell depicted in Figure (3.10) as an example, the positive value of the level-set field for node B will mean that the coordinate $\mathbf{x}_B$ is a vertice of the material volume. The sign change in the value of ϕ between two nodes B and C, i.e. $\phi(\mathbf{x}_B).\phi(\mathbf{x}_C) < 0$, indicates an intersection along the curve (cell edge) joining these two points. The coordinates of the intersection point along the adjoining edge that is cut by the interface can be computed as follows:

$$\mathbf{x}_{C'} = \mathbf{x}_B + \theta|\mathbf{x}_C - \mathbf{x}_B|\,\mathbf{n}_{BC} \tag{3.44}$$

where:

$$\theta = \frac{|\phi_B|}{|\phi_B| + |\phi_C|}$$

is the fraction of the cell edge occupied by the material; and $\mathbf{n}_{BC}$ is the unit normal from points B to C. It is easy to see how this procedure generalises for each node/cell-edge. Determination of coordinates for positive nodes and intersection points for the cell face is carried out in an anti-clockwise direction. The values of all vertices of the arbitrary polyhedron can then be stored for use in the computation of cell face apertures and eventually the volume fraction of the cell. Note that when a value of an intersection point is computed, this value must be stored as $\overline{\mathbf{x}} = \mathbf{x}_B$, the value of a point on the interface plane, as required in the computation of the material volume using Equation (3.39). Recall however, that the coordinate vector of only one point on the interface is required, so this needs not be stored each subsequent time an intersection is found. The apertures are assumed to be arbitrary polygons defined through intersection of the interface with the cell faces and thus the areas can be computed from:

$$A = \frac{1}{2}|\mathbf{n}.\sum_{m=1}^{N_V}\mathbf{x}_m \times \mathbf{x}_{m+1}| \tag{3.45}$$

where $x_1, x_2, ..., x_{N_V}$ are the vertices, and N_V the total numbers of these vertices. Note that $x_1 = x_{N_V}$.

Based upon the level-set field then, the computation of the cell face apertures and finally the volume fraction can be summarised as follows:

1. Compute the nodal values of the level set function using Equation (3.43).
2. Check whether the cell is full, empty, or mixed. For each mixed-cell, check if there is a change in sign across the bounding nodes. For those edges cut by the interface, compute the intersection point using Equation (3.44). Store the coordinates for all vertices with positive values of the level-set field.
3. For each cell face cut by the interface, compute the area occupied by the material using Equation (3.45). Alternatively, if all nodal values are posi-

tive, the area is simply the area of the cell face, and if all values are negative, the area is zero.

4. Compute the area of the interface using Equation (3.40), and the interface normal using Equation (3.41).
5. Compute the volume of the material using Equation (3.39), and finally the volume fraction using Equation (3.35).

It is remarked that the combination of employing level-set functions to track material boundaries and the above described reconstruction procedure within mixed cells does limit the range cell geometries that can be resolved. In regions of curvature, the act of averaging cell-centered values of the level-set functions to obtain the required nodal data smooths the boundary profile, the error introduced hence scaling with the grid size. Features such as corners and cusps proved to be problematic and could not be resolved within a single cell. However, it is mentioned that regularisation inherent in the numerical methods employed to solve the level-set advection and reinitialization equations will likely lead to the destruction of such features over a number of iterations, reducing the local curvature significantly and hence increasing the faithfulness of sub-cell interface representation as a flat plane. The event of two interfaces of the same component passing through, but not merging in, the same cell, would prove to be a singularity in the above method, potentially irrespective of the regularisation. Additional steps would be required to decompose the cell to determine the orientation of each face and overcome the ambiguity in possible scenarios (such as is done in the marching cubes algorithm [47]).

3.5.2 Cell merging and time integration

A distinct difficulty faced by any explicit interface tracking method is the appearance of small cut cells, which restrict the global maximum allowable timestep to proportionally small values in order to avoid violation of the CFL condition presented in section 2.2.2. The global timestep is taken to be the minimum of local time-steps computed for each material that occupies each cell, which in turn will be a function of the wave speeds and the cell dimensions:

$$\Delta t = C\underset{i,j,k}{min}(\underset{\eta}{min}(\Delta t^{(1)}, \Delta t^{(2)}, ..., \Delta t^{(\eta)}) \tag{3.46}$$

For each material in a given cell, the timestep $\Delta t^{(\eta)}$ will depend linearly on the volume $\mathcal{V}^{\eta}$ of material actually residing in the cell, which in turn is a function of its volume fraction α^{η}.

Hence, if $\mathcal{V}^{\eta} \rightarrow 0$ then $\Delta t^{(\eta)} \rightarrow 0$ also in order to resolve a stable state following time integration. Since this scenario will occur indefinitely in a system where the interfaces evolve through space and time, it is a critical matter to address. Ideally, the timestep will be limited to the smallest of those for cells with a volume fraction of say $\alpha \geq 0.5$. It is mentioned that this difficulty also appears in static/rigid-body-motion immersed boundary methods (see for example [41, 17, 80, 59]). In the past different methods have been proposed to overcome the problem, including cell merging techniques (see for example [80, 38, 57]), and the linear hybridisation method of conservative and non-conservative updates in [17, 52]. In order to avoid small volume fractions being taken into account necessarily in the calculation of the timestep and hence potentially yielding very small values, a cell merging technique is employed. In this, small cells, designated by the value of volume fraction (typically $\alpha < 0.5$, thus including empty cells), are merged with an adjacent cell with a larger volume fraction or indeed a volume fraction of unity, designated as target cells. Note that a target cell may be shared by a number of small cells. At the start of each timestep, all those cells that have been combined as one should assume a common state $\mathbf{U}_C^n$. A forward Euler time-integration of the combined cell with finite-volume discretisation of the resultant polyhedron can be written:

$$\mathcal{V}_C^{n+1}\mathbf{U}_C^{n+1} = \mathcal{V}^n\mathbf{U}_C^n - \Delta t\sum_{m=1}^{N_{f_C}}(\mathcal{A}F\mathbf{n})_m \tag{3.47}$$

where N_f is the number of faces forming the exterior of the combined material volume. The combined cell material volume comprises the sum of all component cell material volumes:

$$\mathcal{V}_C = \mathcal{V}_{TG} + \sum_{m=1}^{N_{SM}}\mathcal{V}_m \tag{3.48}$$

Furthermore, the sum of fluxes over the polyhedron surface is expressible in terms of the independent sum of fluxes of each component cells per Equation (3.38):

$$\sum_{m=1}^{N_{f_C}}(\mathcal{A}F\mathbf{n})_m = \sum_{m=1}^{N_{f_{TG}}}(\mathcal{A}F\mathbf{n})_m + \sum_{l=1}^{N_{SM}}\left[\sum_{m=1}^{N_{f_l}}(\mathcal{A}F\mathbf{n})_m\right] \tag{3.49}$$

i.e. a finite volume discretisation can be performed for each component cell irrespective of the others, thus including computing fluxes across cell-walls that are interior to $\mathcal{V}_C$ but not necessarily coinciding with $\partial\mathcal{V}_C$; since the unit normal for such faces are equal and opposite for the pair of component cells sharing it then such contributions will cancel in the sum Equation (3.49). Combining Equations (3.47)-(3.49) and using the fact that $U^n_{TG} = U^n_{SM} = U^n_C$ gives:

$$\begin{aligned}\left(\mathcal{V}_{TG} + \sum^{N_{SM}}\mathcal{V}_{SM}\right)^{n+1}\mathbf{U}^{n+1}_C \quad &= \quad (\mathcal{V}_{TG}\mathbf{U}_{TG})^n - \Delta t\left[\sum_{m=1}^{N_{f_{TG}}}(\mathcal{A}F\mathbf{n})_m\right] + \\ &\quad \sum^{N_{SM}}(\mathcal{V}_{SM}\mathbf{U}_{SM})^n - \Delta t\sum_{l=1}^{N_{SM}}\left[\sum_{m=1}^{N_{f_l}}(\mathcal{A}F\mathbf{n})_m\right] \qquad (3.50) \\ &= \quad (\mathcal{V}\mathbf{U})^{n+1^*}_{TG} + \sum_{m=1}^{N_{SM}}(\mathcal{V}\mathbf{U})^{n+1^*}_m \qquad (3.51)\end{aligned}$$

where the asterisk is used to denote quantities found through updating to the next time-level individually using Equation (3.38), irrespective of the size and cell pairing. This last result forms the basis of the convenient merging method proposed by Hu et al. [38] that avoids direct evaluation of Equation (3.47): the finite-volume update Equation (3.38) is computed for all cells regardless of the volume fraction to give $(\mathcal{V}\mathbf{U})^*$; following each temporal update, the solution $\mathbf{U}^{n+1}_{TG} = \mathbf{U}^{n+1}_C$ is achieved in all target cells through a correction to the $(\mathcal{V}\mathbf{U})^*_{TG}$ state which effectively cancels the fluxes computed across surfaces interior to $\mathcal{V}$ and accounts for the fluxes on $\partial\mathcal{V}$ from all associated small cells, and the change in small cell material volumes. The correction for the target cells can be found after some rearrangement of Equation (3.51):

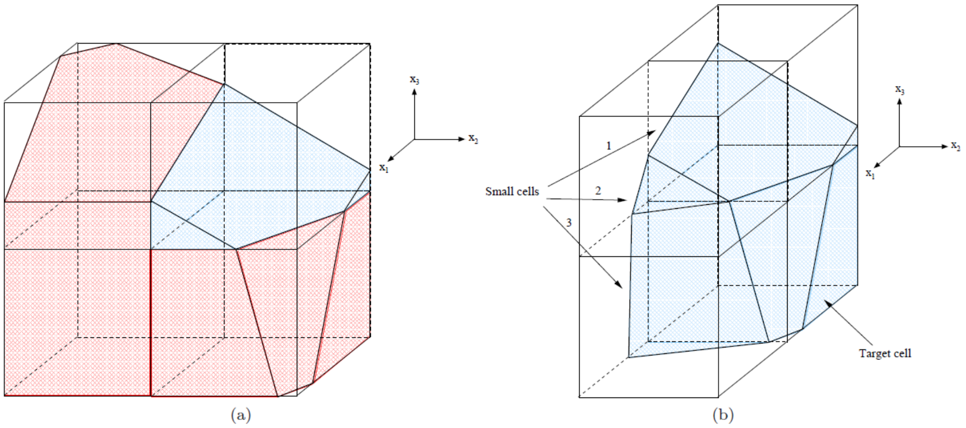


Figure 3.11: Illustration of the cell merging method, showing (a) a highlighted small cell surrounded by possible targets, (b) a three-dimensional region of large curvature (perhaps a corner) where using the position of the largest component of the interface normal for the small cell 2 would choose either cell 1 or 3, both of which are small, whereas the target cell diagonally should be chosen. The final conservative and stable cell pairing shows that the target cell has three associated small cells; in the extended mesh note that this number may be larger since there are also neighbouring empty cells.

$$(\alpha\mathbf{U})^{n+1}_{TG} = (\alpha\mathbf{U})^{n+1^*}_{TG} + \frac{\alpha^{n+1}_{TG}\sum^{N_{SM}}(\alpha\mathbf{U})^{n+1^*}_{SM} - (\alpha\mathbf{U})^{n+1^*}_{TG}\sum^{N_{SM}}\alpha_{SM}}{\left(\alpha_{TG}+\sum^{N_{SM}}\alpha_{SM}\right)^{n+1}} \tag{3.52}$$

where the result has been divided through by the mesh element volume V. Whilst a similar correction could be formulated for each of the small cells, it is more convenient to simply put $U^{n+1}_{SM} = U^{n+1}_{TG}$ following evaluation of Equation (3.52).

Strict conservation requires a consistent pairing of small and target cells. For each small cell, it is suggested in [38] that the inwards pointing unit normal to the interface, computed using the level-set field, can be used to determine a neighbouring target cell. Specifically, the target cell is chosen to be the adjacent cell in the x_i-direction, where $1 \leq i \leq 3$ is chosen to equal the index position of the component of $\mathbf{n}$ with the largest magnitude. Note that this approach generates maximal volume combined cells and thus guarantees that the time-step chosen on the basis of the dimensions and states of large cells will not violate the CFL condition. In regions of the boundary where curvature is small, no further action need be taken and typically there is a one-to-one pairing of boundary full cells and adjacent small/empty cells (Figure 3.11-a). However, for small cells in regions where the boundary has large curvature (e.g. near corners), the cells adjacent in the Cartesian coordinate directions might also be small. Such a situation is illustrated in Figure 3.11-b. Instead, the nearest full cell whose orientation agrees with the unit normal lies diagonally. In order to maintain strict conservation, each small cell should be associated with one target cell, but, as Figure 3.11-b depicts, a target cell might be shared by any number of small cells. Hence, if, when searching for a target cell, the cells adjacent in the Cartesian coordinate directions chosen by the aforementioned criteria are also small then all cells in the direct vicinity are checked for compatibility until a suitable candidate target is found. It is mentioned that this approach also becomes necessary in the event of thin ligaments forming, where the unit normals are occasionally erroneously computed and hence would reflect in an erroneous cell pairing. It is further stipulated that if a multi-stage time-integration method is employed (e.g. Runge-Kutta) then the cell pairings must remain the same for each stage. This fact can easily be shown through derivation of the modified correction that must be applied to the updated state in the target cells at the end of each time-level. For higher order accuracy time integration of conserved quantities, Equation (3.38) can be evaluated using a ν-stage explicit Runge-Kutta method, where any variant in compact form can be written:

$$\begin{aligned}(\alpha\mathbf{U})^{(i)} &= (\alpha\mathbf{U})^{n} - \frac{\Delta t}{V}\sum_{k=1}^{i}\beta_{i,k}\left(\sum^{N_f}(\mathcal{A}F\mathbf{n})^{(k-1)} + VS^{(k-1)}\right)\\ (\alpha\mathbf{U})^{n+1} &= (\alpha\mathbf{U})^{n} - \frac{\Delta t}{V}\sum_{i=1}^{\nu}\gamma_{i}\left(\sum^{N_f}(\mathcal{A}F\mathbf{n})^{(i)} + VS^{(i)}\right)\end{aligned} \tag{3.53}$$

The $\nu \times \nu$ matrix $\beta_{(i,k)}$, and vector $\gamma_i = (\gamma_1, ..., \gamma_\nu)$ contain the constant coefficients corresponding to the RK method employed. For the third-order TVD variant described in [42], the coefficients are hereby reminded:

$$\beta = \begin{pmatrix} 1 & 0 & 0 \\ 1/4 & 1/4 & 0 \\ 0 & 0 & 0 \end{pmatrix} \text{ and } \gamma = \begin{pmatrix} 1/6 \\ 1/6 \\ 2/3 \end{pmatrix} \tag{3.54}$$

The final update for a combined cell using a high-order RK integration derived in Equation (3.53) following Equation (3.50) would thus be:

$$\begin{aligned} \left(\mathcal{V}_{TG} + \sum^{N_{SM}} \mathcal{V}_{SM}\right)^{n+1} \mathbf{U}_C^{n+1} &= (\mathcal{V}_{TG}\mathbf{U}_{TG})^n - \Delta t \left[\sum_{l=1}^{\nu} \gamma_k \sum_{k=1}^{N_{f_{TG}}} (\mathcal{A}F\mathbf{n})_k\right] + \\ & \sum^{N_{SM}} (\mathcal{V}_{SM}\mathbf{U}_{SM})^n - \Delta t \sum_{i=1}^{N_{SM}} \left[\sum_{l=1}^{\nu} \gamma_k \sum_{k=1}^{N_{f_{TG}}} (\mathcal{A}F\mathbf{n})_k^{(l)}\right] \end{aligned} \tag{3.55}$$

from which it can be seen that the correction to target states is equal to Equation (3.52) only if the group of component cells does not change between t^n and t^{n+1}. Note that an equivalent correction must be invoked for the state $(\alpha\mathbf{U})_{TG}^{(.)}$ at the end of each sub-stage. The corrected target state $\mathbf{U}_{TG}$ for any stage can then be extrapolated to the associated small cells before moving to the next time-level. It is insufficient to apply the PDE extrapolation method Equation (3.32) for this purpose since the state of small cells must strictly be taken from the corresponding target. Doing otherwise and using Equation (3.52) to correct for the flux deficit will violate conservation. This caveat is further exemplified by the requirement that the cell pairings remain unchanged during the update from t^n to t^{n+1}, whereas the unit normals upon which the PDE extrapolation method is based may change between RK sub-stages. The maintenance of cell pairings is conveniently achieved using a cell flagging method similar to that proposed in [60] for use with the fast-local-level-set method. It is mentioned that the PDE method is used to extrapolate states to all cells outside the material domain that are not designated as small cells which are required for continuity of the numerical stencils used by the reconstruction procedures when determining numerical fluxes in the vicinity of the boundaries.

Summary of scheme

It is assumed that when the solution is initialised not only are the physical states known for all cells, but also the level-set fields, apertures, volume fractions, and based upon the latter the cell pairings for small cells. For each time update from t^n to t^{n+1} the numerical scheme can then be summarised as follows:

1. Solve a multi-material Riemann problem in mixed cells and compute the boundary fluxes and interface velocities using the PDE extrapolation method Equation (3.32);
2. For each component, extrapolate the interfacial velocities from mixed cells into a finite sized band of cells encompassing the interface;
3. For each component, compute the cell face fluxes for all cells irrespective of whether material resides in a cell or not, and construct the discretised terms on the right-hand-side of Equation (3.38). Also, compute the convective terms in the evolution equation Equation (3.3) for the level-set fields for each material;
4. Update both the physical variables and level-set fields in all cells to the next time-level;
5. Compute the new cell apertures and volume fractions based upon the updated level-set fields;
6. Apply the cell-merging correction for all target cells using Equation (3.52);
7. Recover the conserved variables in all target cells by dividing through by the new volume fractions and extrapolate the solution to the respective small cells. Furthermore, populate the state of cells surrounding the component interfaces that are not designated small cells using the PDE extrapolation method to ensure continuity of the numerical stencil in the boundary regions;
8. Repeat steps 1-7 for each sub-stage in the multi-level time integration method;
9. For all full, target, and small cells use the updated state as initial data in solving the source terms that remain to be evaluated in the governing models;
10. Reinitialise the level-set fields if required;

11. Reset the cell pairings, designating target cells for all those that, based upon the updated volume fraction field, are now categorised as small;

12. Repeat steps 1-11 until the desired time-level is reached.

This method is formulated independently of the set of equations used. The only restriction is the need of having a hyperbolic system of conservation laws for the latter derivation to be correct. It also allows to treat each material separately, using the same spirit as the ghost fluid methods in term of filling cells near the interface with prescribed values in order to use single-component solvers.

The two following chapters will be dedicated on applying this new 3D method to fluid/fluid and fluid/solid problems, respectively.

4

Numerical results: fluid/fluid applications

4.1 Constitutive models

4.1.1 Euler equations

The model used for fluid components is the Euler equations, for inviscid compressible flows. That simplified model obtained by simplification of the Navier-Stokes equations (NSE) assumes that the heat conductivity is null and besides, there are no viscous effects and forces associated. The three-dimensional set of equations written using conserved variables is thus given by [75]:

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho u v \\ \rho u w \\ (E+P)\,u \end{pmatrix}_x + \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + P \\ \rho v w \\ (E+P)\,v \end{pmatrix}_y + \begin{pmatrix} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + P \\ (E+P)\,w \end{pmatrix}_z = 0 \quad (4.1)$$

where ρ, u, v, w, P, E are the density, x-velocity component, y-velocity component, z-velocity component, pressure and total energy, respectively. $(\mathbf{U})_m, m = t, x, y, z$ denotes the partial derivative of $\mathbf{U}$ with respect to the m-th variable. The total energy can be re-written using the internal energy ξ as:

$$E = \rho(\xi + \frac{(u^2 + v^2 + w^2)}{2}) \quad (4.2)$$

Rewriting Equation (4.1) as a hyperbolic system of conservative laws like Equation (2.2) leads to the following formulation, by identification:

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ E \end{pmatrix}, \quad F^{(i)} = \begin{pmatrix} \rho u_i \\ u_i\rho\mathbf{u} + P\mathbf{e}_i \\ u_i(E+P) \end{pmatrix}, \quad S = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{4.3}$$

With **u** the velocity vector, and $\mathbf{e}_i$ the normal unit vector in the i-th direction.

To carry out the characteristics analysis, Equation (4.3) must in turn be expressed under the form of Equation (2.3). It implies to get the Jacobian matrix derived from F. Derivation is tedious algebra, so the expression will only be given when using a Gamma-law gas (Equations of state will be described in detail in the next section):

$$A(\mathbf{U}) = \frac{\partial F}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \tau H - u^2 - a^2 & (3-\gamma)u & -\tau v & -\tau w & \tau \\ -uv & v & u & 0 & 0 \\ -uw & w & 0 & u & 0 \\ \frac{1}{2}[(\gamma-3)H - a^2] & H - \tau u^2 & -\tau uv & -\tau uw & \gamma u \end{bmatrix} \tag{4.4}$$

where γ is the characteristic constant of the gas, $\tau = \gamma - 1$, and the enthalpy H is introduced:

$$H = \frac{(E+P)}{\rho} \tag{4.5}$$

This system is indeed hyperbolic, the eigenvalues of A being (note that three of them are non-distinct, the system is then non-*strictly* hyperbolic):

$$\lambda_1 = u - a, \quad \lambda_1 = \lambda_2 = \lambda_3 = u, \quad \lambda_5 = u + a \tag{4.6}$$

where a is the speed of sound, defined by:

$$a = \sqrt{\frac{\partial P}{\partial \rho}} \tag{4.7}$$

4.1.2 Reactive Euler equation

Some numerical testcases carried for studying solid/fluid interaction in Chapter 5 involve a reaction model for the simulation of a detonation. For that purpose, the modified Euler equations for reactive materials will be employed. It is expressed as the following (see [52]):

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ E \\ Q \end{pmatrix}, \quad F^{(i)} = \begin{pmatrix} \rho u_i \\ u_i \rho \mathbf{u} + P\mathbf{e}_i \\ u_i(E+P) \\ u_i Q \end{pmatrix}, \quad S = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \dot{Q} \end{pmatrix} \tag{4.8}$$

where Q is the total chemical potential energy, and E the total energy being now:

$$E = \rho(\xi + \frac{(u^2+v^2+w^2)}{2}) + Q \tag{4.9}$$

The potential chemical energy can be defined as:

$$q = \frac{Q}{\rho} \tag{4.10}$$

It is mentioned however that the complexity of resolving small reaction zones required for use of the classical Arrhenius models is circumvented by employing a programmed burn model.

Basically, the cells are considered containing a given amount of potential chemical energy, to be ignited at some time. Each cell is then or unreacted (the chemical energy is at its maximum value), totally reacted (all the chemical energy has been depleted and converted into internal energy), or partially reacted. Therefore, the chemical energy in a cell can be written as:

$$q = \alpha q_0 \tag{4.11}$$

With α a mass fraction of reacted material determining the state of the cell, and q_0 the initial amount of potential chemical energy.

Two remarks can be made at that point. Firstly, this reaction model only assumes transfer from chemical energy to internal energy ξ without any change

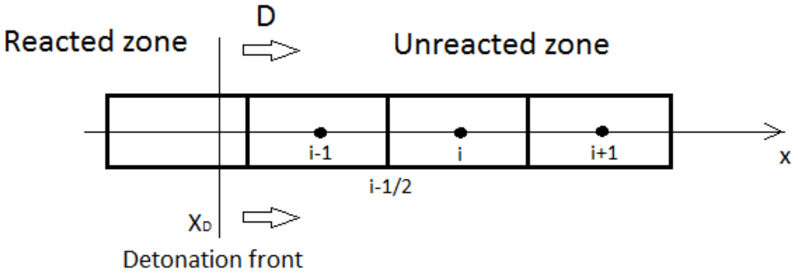


Figure 4.1: Propagation of the detonation front in 1D

of mole number, and consequently without affecting density. Secondly, that reaction occurs only this way: internal energy cannot convert back to chemical energy, i.e. the chemical energy of a computational cell can only decrease in time, and thus $0 \leq \alpha \leq 1$. Besides, it is assumed that the reaction depletes energy at a constant rate $\dot{q_0}$.

In [52], the reaction state of a cell is known by the temperature in the cell, a threshold determining if a cell is reacted or not. Another option to avoid the (possibly) tedious computation of temperature is to use the programmed-burn approach of [27] where the velocity of the detonation front (noted D) is assumed known since a constant of the material; thus, a cell is identified as reacted if the detonation front has reached the cell. That process is illustrated on Figure 4.1 where the detonation moves from the left to the right.

As a result, the time for the cell to be entirely ignited (each cell is considered as having the size Δx) is:

$$\tau = \frac{K\Delta x}{D} \tag{4.12}$$

where K is a constant multiplier controlling the diffusivity of the reaction zone, usually taken to 1. The ignition time of the cell i is immediately given by:

$$t_i^{ign} = \frac{x_{i-1/2} - x_D}{D} \tag{4.13}$$

Which corresponds to the time at which the detonation front reaches the left hand face of the given cell. The burn fraction λ at a time t can then be defined based on the predicted burn time:

$$\lambda = \begin{cases} 0 & \text{if } t < t_i^{ign} \text{ (the detonation has not reached the cell yet)} \\ \frac{(t-t_i^{ign})}{\tau} & \text{if } t_i^{ign} + \tau \geq t \geq t_i^{ign} \text{ (the detonation front has} \\ & \text{partially reacted the cell)} \\ 1 & \text{otherwise (the detonation front has entirely reacted the cell)} \end{cases} \tag{4.14}$$

At the end of a computational time step, the chemical energy in each cell is then updated accordingly:

$$q^{(n+1)} = q^{(n+1)^*} - \lambda \Delta t \dot{q_0} \tag{4.15}$$

where $q^{(n+1)^*}$ denotes the solution from the preceding update involving only the convective terms. Since, as mentioned earlier, the model employs a constant reaction rate and that the chemical energy can only decrease in time, another condition should be enforced, conveying the fact that the chemical energy cannot become negative. If from one time step to another, a cell becomes entirely depleted of chemical energy, the amount of q should then be set to 0, leading to the following restriction:

$$q^{(n+1)} = max(0, q^{(n+1)}) \tag{4.16}$$

where $q^{(n+1)}$ of the right-hand side corresponds to the one obtained with Equation (4.15).

4.2 Equation of state (EOS)

Noticing that the system (4.1) contains six unknowns for five equations (or System (4.8) has seven unknowns for six equations), another relation needs to be specified in order not to have an under-determined system. To close the system, the missing equation comes from a thermodynamical relation between state variables. That relation is dependent on the material state only, not of the flow conditions, therefore it should relate only pressure, density and internal energy. Later in the section, and in Chapter 5, three EOS will be used and are described below.

4.2.1 Gamma-law gas

The Gamma-law gas is probably the most common approximation for gases like air or helium, i.e. non-reactive gases in which intermolecular forces are neglected.

In that case, pressure, internal energy and density are related through a single constant parameter, denoted γ and defined as the ratio between the specific heats at constant pressure c_p and constant volume c_v:

$$\gamma = \frac{c_p}{c_v} \tag{4.17}$$

That ratio is shown constant if the assumptions of perfect gas are complied. It becomes therefore a parameter characteristic of the material itself.

To model atmospheric air, γ is usually taken as $\gamma = 1.4$. However, for different gases, gamma can be different, e.g. for Helium, $\gamma = 1.66$.

The EOS of a Gamma-law gas is then formulated according to the value of Gamma, under the Mie-Grüneisen form [70]:

$$\rho\xi = \frac{P}{\gamma - 1} \tag{4.18}$$

Another relevant quantity to be established is the expression of entropy for the Gamma-Law gas (extensively used in GFM and MGFM). The definition of Fedkiw [26], that defines the entropy of a Gamma-Law gas as:

$$\mathcal{S} = \frac{P}{\rho^{\gamma}} \tag{4.19}$$

4.2.2 Tait's equation

Tait's equation is an EOS proposed for compressible water. It can be expressed in a Mie-Grüneisen form [70] as:

$$\rho\xi = AP + B \tag{4.20}$$

with A and B two constants to be fixed. In most tables, A and B are usually replaced by two variables, N and P_{ref}, so that: $A = \frac{1}{N-1}$ and $B = \frac{N}{N-1}(P_{ref1} - P_{ref2}) = \frac{N}{N-1}P_{ref}$.

4.2.3 Jones–Wilkins–Lee equation

The Jones– Wilkins–Lee (JWL) EOS is used to simulate explosives (TNT, C-4...), and is therefore mostly used in detonation testcases. The equation of state is given in [40]:

$$P = A_0(1-\frac{\rho}{R_1\rho_0})\exp(\frac{-R_1\rho_0}{\rho})+B_0(1-\frac{\rho}{R_2\rho_0})\exp(\frac{-R_2\rho_0}{\rho})+\Gamma_0\rho(\xi+\xi_0) \quad (4.21)$$

where $A_0, B_0, \rho_0, \xi_0, R_1, R_2$ and Γ_0 are constant coefficients.

That EOS is widely used for its simplicity, each term of the right-hand-side of Equation (4.21) being prevailing at different pressures. For more details, see [3].

4.3 Riemann solver

For fluid/fluid cases, the multimaterial Riemann problem at the interfaced is solved by using a simple linearised primitive variable Riemann solver. In multi-dimensions, to account for the arbitrary angle of the interface with respect to the Cartesian axis, the state variables for both materials within a mixed cell must be rotated on to the orthogonal coordinate system $\vec{\mathbf{x}}'_i(\mathbf{n})$, where $\vec{\mathbf{x}}'_1 = \mathbf{n}$, and the tangential directions can be taken arbitrarily.

If $\mathbf{R}^{ROT} = (\vec{x_1}, \vec{x_2}, \vec{x_3})^T$ denotes the tensor of direction cosines of $\mathbf{x}'$ with respect to $\mathbf{x}$, then the transformation of state variables onto the interface normal can be written:

$$\begin{pmatrix} \rho' \\ \rho'\mathbf{u}' \\ E' \end{pmatrix} = \begin{pmatrix} \rho \\ \rho\mathbf{R}^{ROT}\mathbf{u} \\ E \end{pmatrix} \quad (4.22)$$

Once rotated, the vector of conserved variables is converted into the primitive variables ρ, u_n, u_t, P, and the resulting Jacobian matrix is treated as having constant coefficients. By integrating the relations given by the generalized Riemann invariants over the characteristic curves (see Chapter 2), the values at the interface are given by (see [75]):

$$\begin{cases} \tilde{P} = \frac{C_R P_L + C_L P_R + C_L C_R (u_L - u_R)}{C_L + C_R} \\ \\ \tilde{u} = \frac{C_L u_L + C_R u_R + (P_L - P_R)}{C_L + C_R} \\ \\ \tilde{\rho_L} = \rho_L + \frac{\tilde{P} - P_L}{a_L^2} \\ \\ \tilde{\rho_R} = \rho_R + \frac{\tilde{P} - P_R}{a_R^2} \end{cases} \tag{4.23}$$

where $C = \rho a$. The sound of speed a is computed thanks to the equation of state of each material. The variables are then converted back to conservative, and rotated back to the Cartesian referential.

4.4 Numerical examples

4.4.1 1D Results: Initial value problems

Initial values problems (IVP) are employed to reflect a wide range of possible applications. They allow to determine the accuracy of the numerical solver employed, e.g. its shock capturing ability, the numerical dissipation created, its robustness in case of high gradients...

Three IVP will be tested, involving different EOS with strong gradients leading to shock and rarefaction waves traveling in the different media.

First testcase

The first testcase is a stiff air-Helium testcase with a strong shock created by a pressure gradient (originally proposed by [1]). These problems, often encountered in explosions and detonations, are proven challenging for numerical methods dealing in multiphase flow, especially the GFM-based methods [23, 45]. Both materials are Gamma-law gases, with $\gamma = 1.4$ for air, and $\gamma = 1.667$ for Helium.

Stiff air is the high-pressured gas, on the left, and Helium on the right. The original interface location is at x=0.5cm. The conditions are the following:

$$(\rho, u, P, \gamma) = \begin{cases} (1, 0, 500, 1.4) & \text{Stiff air} \\ (1, 0, 0.2, 1.667) & \text{Helium} \end{cases} \tag{4.24}$$

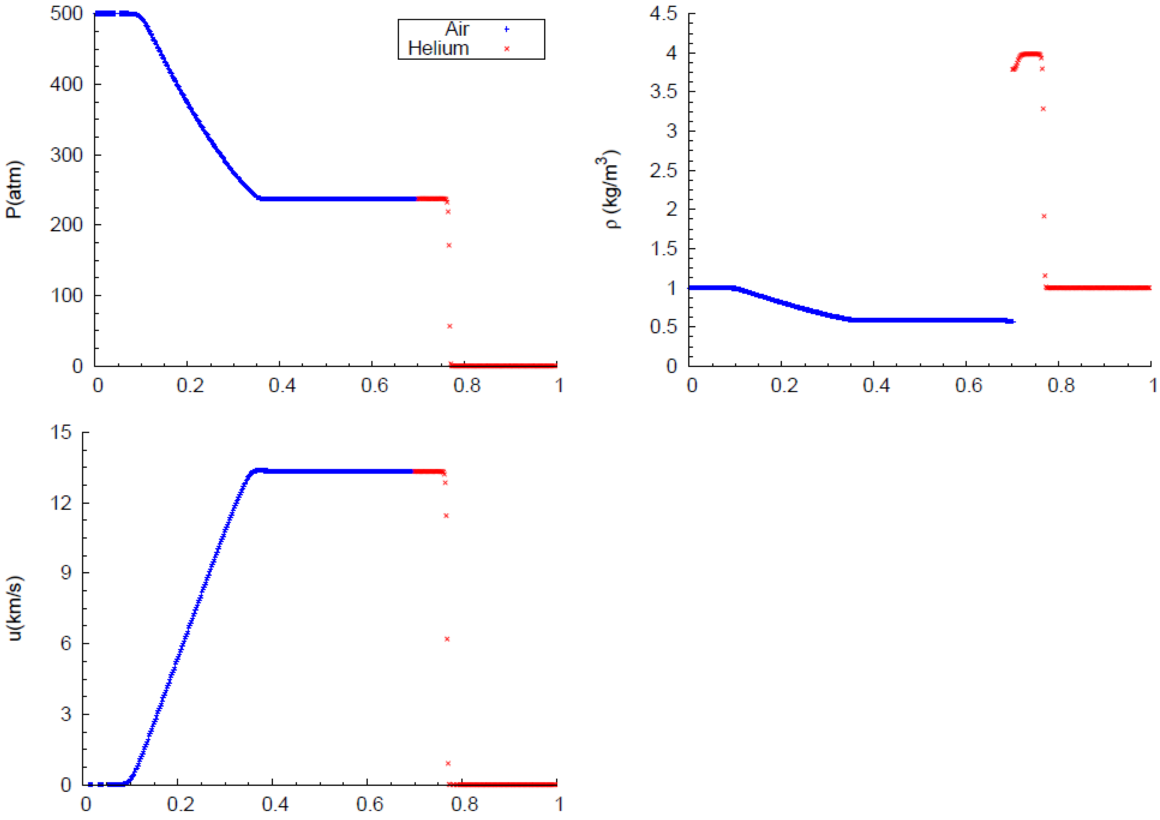


Figure 4.2: Numerical solution of the stiff air-Helium testcase at the time t = 0.015s using WENO-3 with C = 0.9, and x = 1/500 m.

i.e. a pressure ratio of 2500. Units in conditions (4.24) are $kg.m^{-3}$,$km.s^{-1}$ and bars for densities, velocities and pressures, respectively. The computed results are given at t=0.015s. The results on different grid size and reconstruction scheme are provided in Appendix C. Specifically 1st-Order, WENO-3 and monotonicity-preserving WENO-5 (MPWENO-5) [8] reconstruction methods are tested, the latter WENO methods both used in conjunction with the 3rd-Order TVD Runge-Kutta time integration method. WENO-3rd has been kept as the best compromise between accuracy and computational time. Grid sizes are 250, 500 and 1000 points.

Figure 4.2 presents the results with the 500 points grid size, a CFL number C=0.9, and the WENO-3 reconstruction. The shock is well captured (over 5 numerical cells) and the results are in good agreement with [39, 55].

Second testcase

The second testcase is a gas-water shock-tube problem, with a high pressure ratio at the interface. The initial high-pressure is located on the gas side. The initial conditions of both sides are:

$$(\rho, u, P) = \begin{cases} (1.27, 0, 8000) & \text{Air} \\ (1, 0, 1) & \text{Water} \end{cases} \tag{4.25}$$

and the interface is initially located at x=0.4cm. Air is here again a Gamma-law gas with $\gamma = 1.4$ and water is modeled with Tait's equation, described in section 4.2.2. The coefficients for Tait's model is N=7.15, and $P_{ref} = 3309.10^5 Pa$.

The settings for grid and schemes used are the same than in IVP 1. The results are plotted at t=0.00171s on Figure 4.3(to match the output time of [45]), with a 500 points grid and WENO-3 reconstruction, and a CFL number C=0.9.

The results are in very good agreement with [45] in term of wave location and accuracy. Besides, the higher order reconstruction avoids the oscillation observed in the latter numerical results.

Third testcase

The third and last testcase is a plane explosion in water, and has been proposed in [79]. An explosive modeled by a JWL EOS is in contact with water. The explosive is at high-pressure, while water is in a quiescent state. The initial conditions are:

$$(\rho, u, P) = \begin{cases} (1630, 0, 7.81 \times 10^9) & \text{JWL} \\ (1000, 0, 10^5) & \text{Water} \end{cases} \tag{4.26}$$

in SI units. The coefficients of each material are provided in Table 4.1.

The output time is $t = 0.25 \times 10^{-3}$s. The x range of the grid is [0,4]. The numerical results illustrated on Figure 4.4, with a 500 points grid and WENO-3 reconstruction, and a CFL number C=0.9. They compare well with the exact solution given in [79].

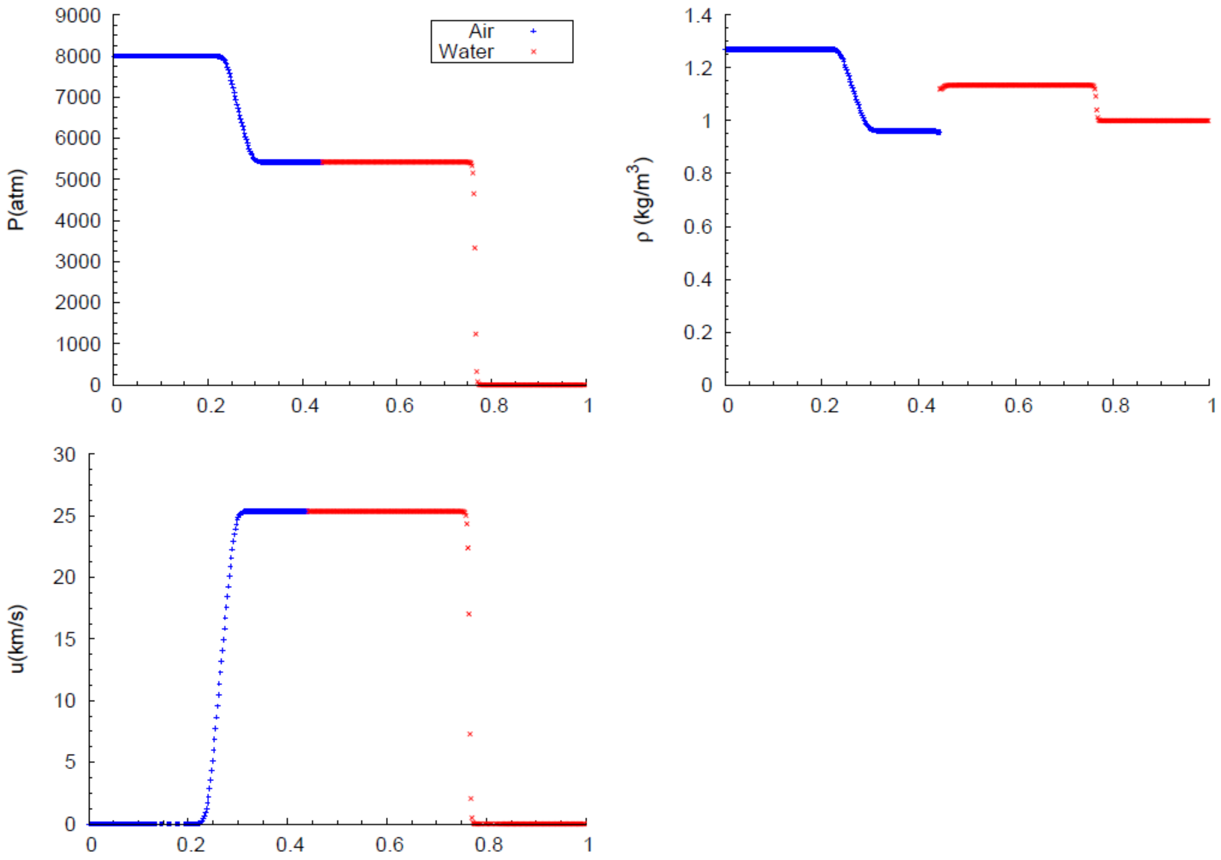


Figure 4.3: Numerical solution of the air-water testcase at the time $t = 0.00171$s using WENO-3 with $C = 0.9$, and $x = 1/500$ m.

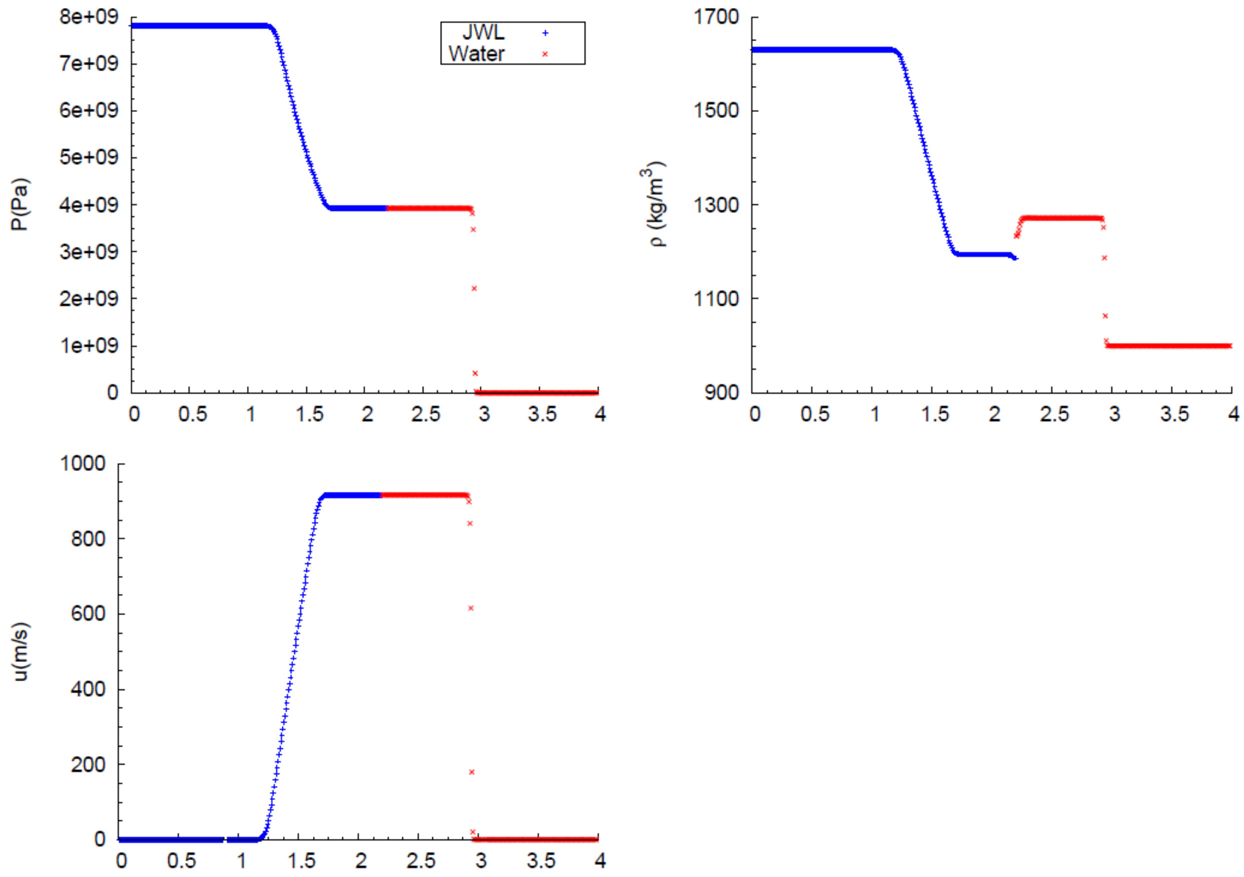


Figure 4.4: Numerical solution of the JWL-water testcase at the time t = 025ms using WENO-3 with C = 0.9, and x = 4/500 m.

Constants	Values	Units
TNT		
ρ_0	1630	$[kg\, m^{-3}]$
A_0	$5.484.10^{11}$	[Pa]
B_0	$0.09375.10^{11}$	[Pa]
R_1	4.6	-
R_2	1.35	-
ξ_0	4281.4	[J]
Γ_0	0.28	-
Water		
N	7.15	-
P_{ref}	3.309×10^9	[Pa]

Table 4.1: Constants for JWL-Water testcase

4.4.2 2D Results: Underwater explosion.

The 2D testcase considered is an underwater explosion (UNDEX) previously studied and simulated in [40, 32]. A high-pressured bubble is embedded within a water medium at quiescent state. The water medium is also in equilibrium with surrounding air, causing the water surface to deform into the air at rest. The testcase is illustrated on Figure 4.5 where the LP and HP notation denote the low-pressure region and high-pressure region, respectively.

The problem is treated as a two-components one, the LP and HP gas between the same Gamma-law gas, with $\gamma = 1.4$, the water medium being modeled again by Tait's equation, and the same parameters as IVP 2 and 3 in the latter section.

The initial conditions can be summarised as:

$$(P,u,\rho) = \begin{cases} (1,0,1.2\times 10^{-3}) & \text{for LP air} \\ (1,0,1) & \text{for water} \\ (10^4,0,1.25\times 10^{-3}) & \text{for HP air} \end{cases} \tag{4.27}$$

where the units are bar, m.s^{-1}, and g.cm^{-3} for pressure, velocity, and density respectively.

The testcase has been run over an 800x500 fine grid, of boundary [(0,4); (0,2.5)], so that $\Delta x = \Delta y = 1/200$m. Due to the fact that the bubble is close

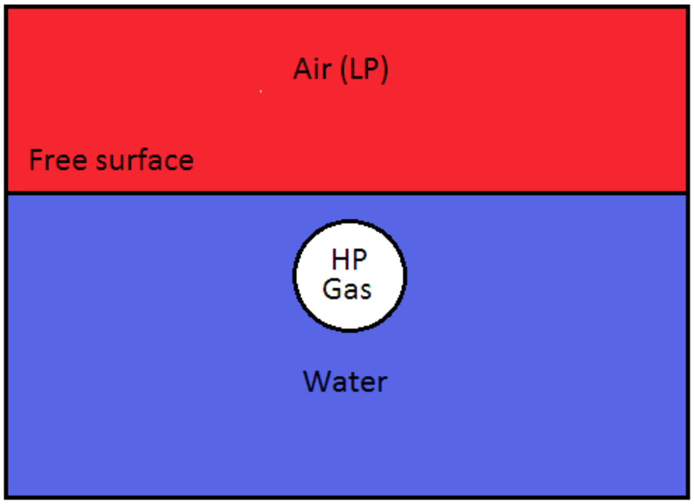


Figure 4.5: Description of the UNDEX testcase

to the free surface, it is necessary to have a decent number of points between those, in order to capture efficiently the behaviour and the deformation of the bubble. WENO-3 reconstruction coupled with 3rd-order TVD RK integration and a CFL number of C=0.6 are used for the simulation

The results are plotted at different times, from 0 to 1.5ms. The bubble deforms from its original circular shape into an oval-like shape (often referred as “egg-shape”), as observed in [40]. Figure 4.6 shows the pressure wave propagation in time, along with the deformation. A contour blanking around the bubble is used for a better illustration.

Figure 4.7 emphasizes the total deformation of bubble from t=0 to t=1.5ms by plotting the zero isocontour of the level-set field over time. Figure 4.8 shows the conservation of both mass of the materials and total energy, which is perfect, the conservation errors sticking strictly to 0%.

4.4.3 3D Results: Shocked air-Helium bubble interaction

This last testcase involves a strong impact between a helium bubble initially at a quiescent state and an air shockwave of strength Mach=1.22, resulting in a significant deformation of the bubble. The configuration is illustrated in 2D on Figure 4.9.

That testcase has been investigated by numerous authors, e.g. experimentally in [33], and numerically in [48, 62, 7]. The initial conditions can be

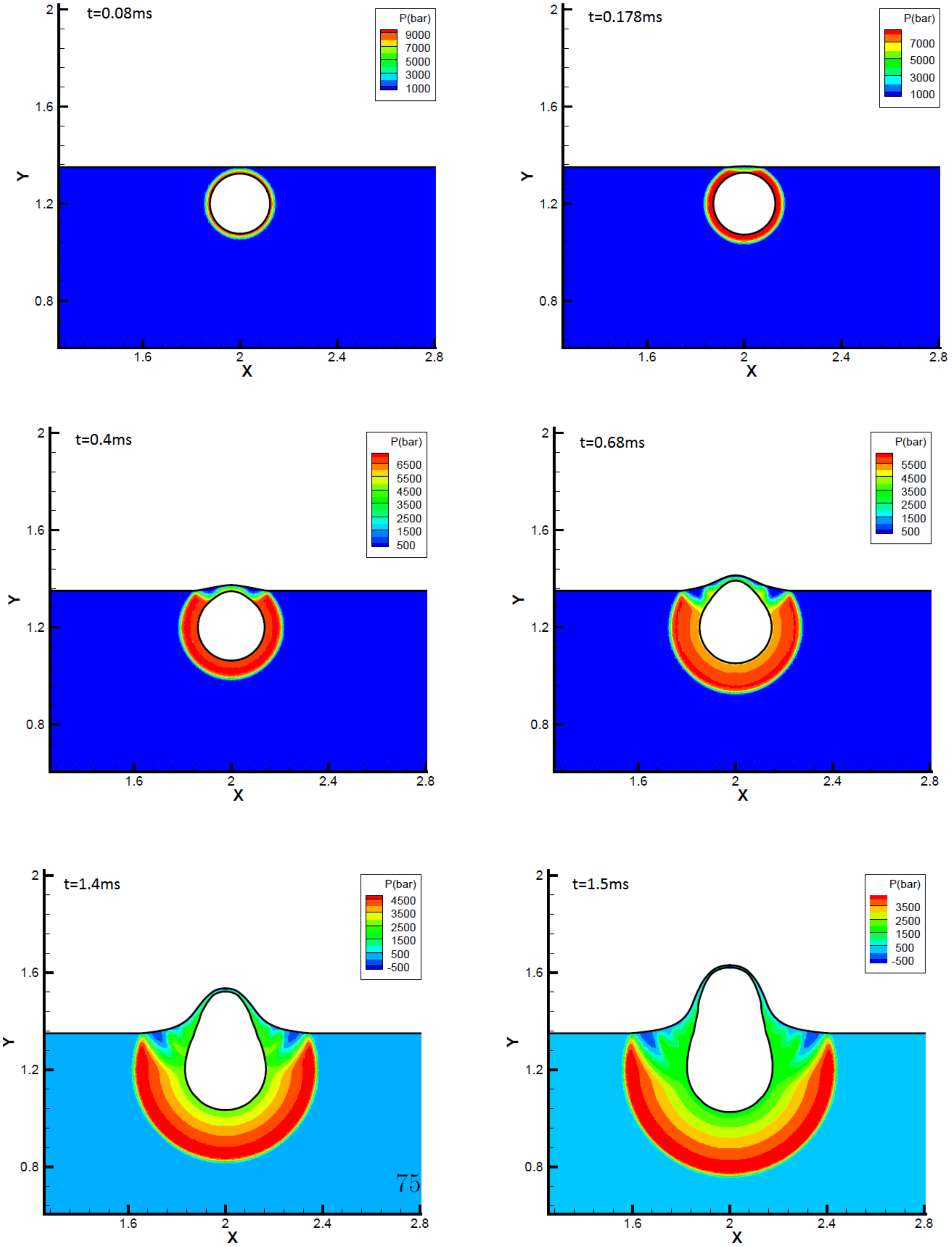


Figure 4.6: Pressure contours at different time steps for the UNDEX.

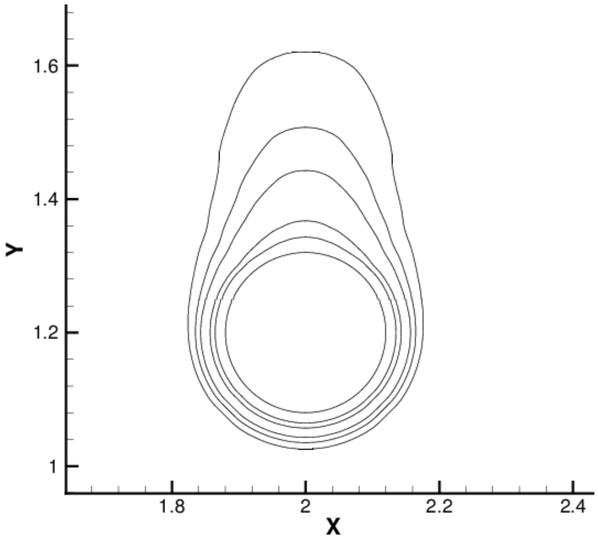


Figure 4.7: Comparison of the zero-level isocontour of the level-set field at different time instances of UNDEX example.

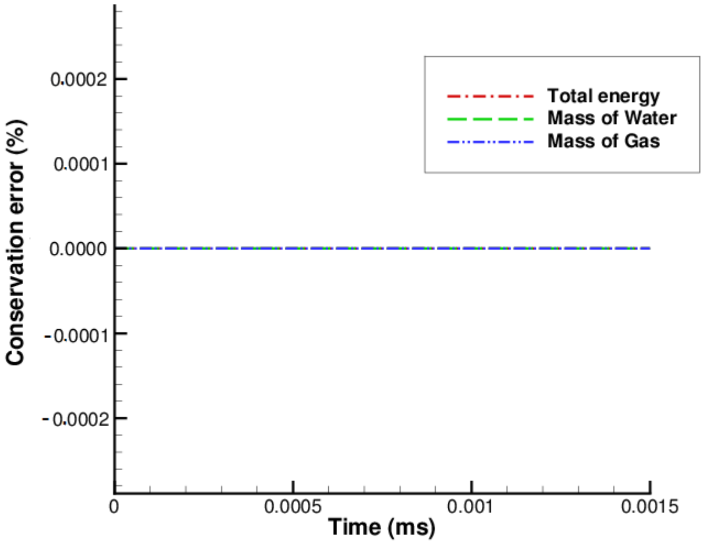


Figure 4.8: Conservation errors for the UNDEX testcase.

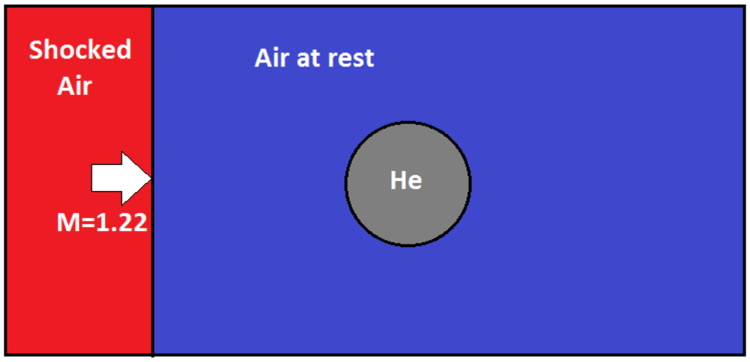


Figure 4.9: Impact air-Helium description.

summarised as follows:

$$(P,u,\rho,\gamma)=\begin{cases}(101325,0,1.225,1.4) & \text{Air at rest}\\(159060,113.516,1.686,1.4) & \text{Shocked air}\\(101325,0,0.228,1.66) & \text{Helium bubble}\end{cases} \tag{4.28}$$

where the units are in Pa, $m.s^{-1}$, $kg.m^{-3}$ for pressure, velocity and density respectively.

The numerical domain is taken from [48] and extended in 3D, consequently being [(0,0.45);(0,0.089);(0,0.089)] (in m). The grid resolution is 800x65x65, and WENO-3, RK-TVD schemes are used, with a CFL number C=0.6.

The results are presented on Figure 4.10 at different time steps. The bubble gets impacted and undergoes a large deformation. With time, the characteristic deformation, referred in the literature as “kidney shape”, appears, as illustrated on Figure 4.11 where a 2D slice (in the plane $y=0, z=0$) is extracted and plotted for comparison with the experiments carried out in the references mentioned earlier.

Eventually, the conservation errors are plotted on Figure 4.12. They are slightly more important than the 2D case, due to the numerical dissipation added by the higher number of equations solved in the z-direction, and most importantly, by the fact that the deformation is much larger than the UNDEX testcase. However, they remain extremely small, not exceeding $2.10^{-5}\%$.

N.B: Since there is an inflow/outflow, the system considered is not conservative strictly speaking. However, to account for the lack of conservation due to the

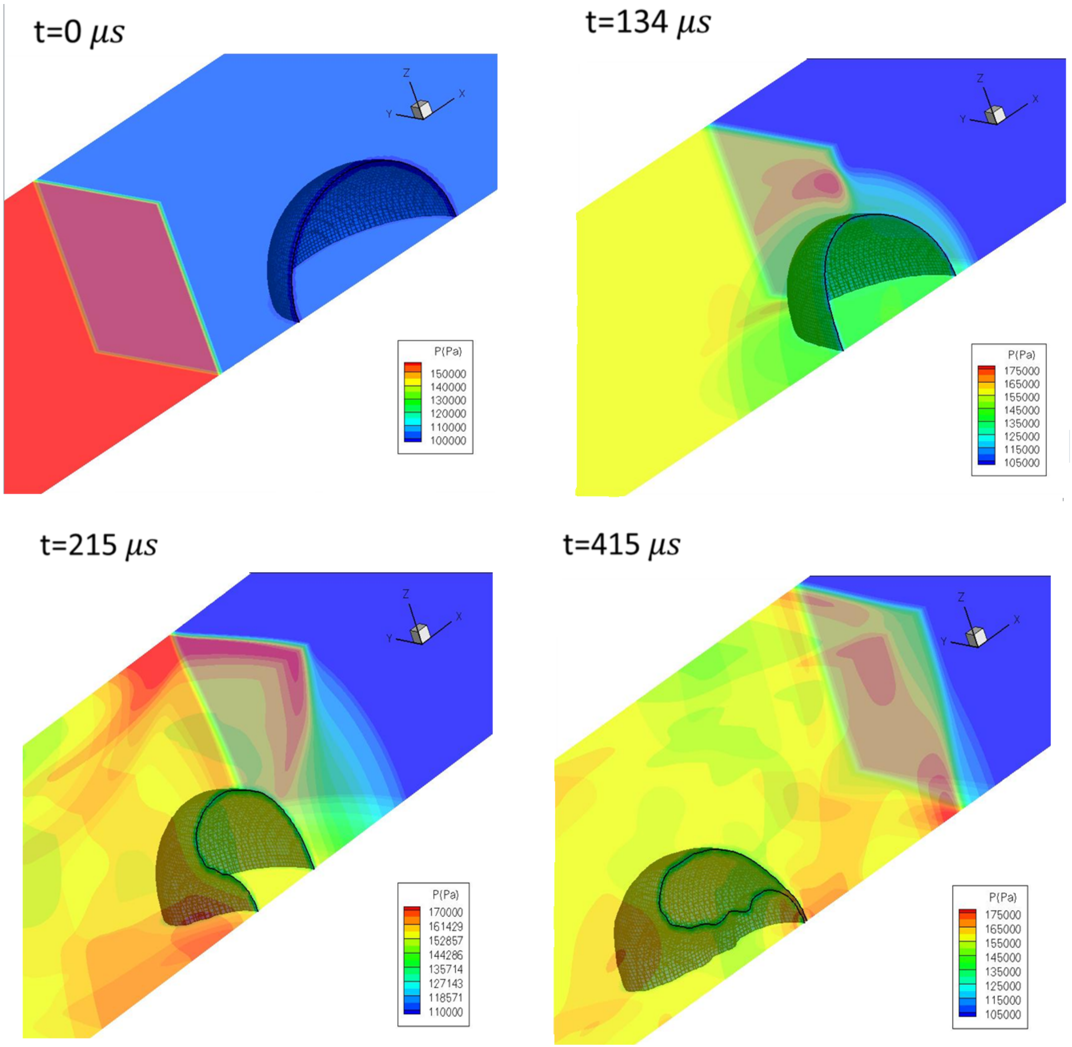
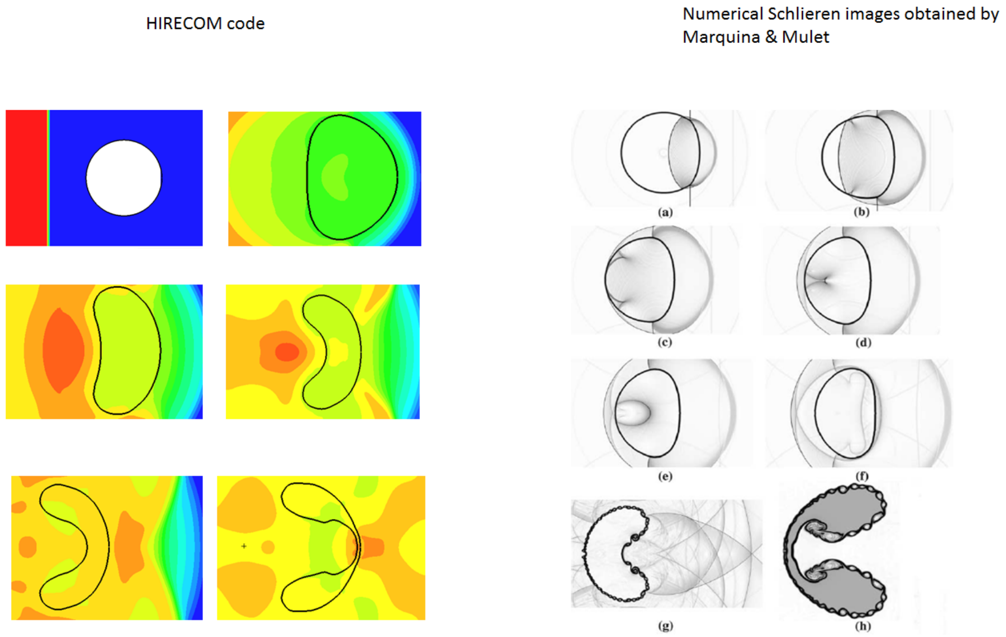


Figure 4.10: Pressure contours and deformation of the Helium bubble impact.


HIRECOM code

Numerical Schlieren images obtained by Marquina & Mulet

Figure 4.11: 2D comparison of Helium deformation of the conservative method (on the left) with the numerical experiment of [48] (on the right).

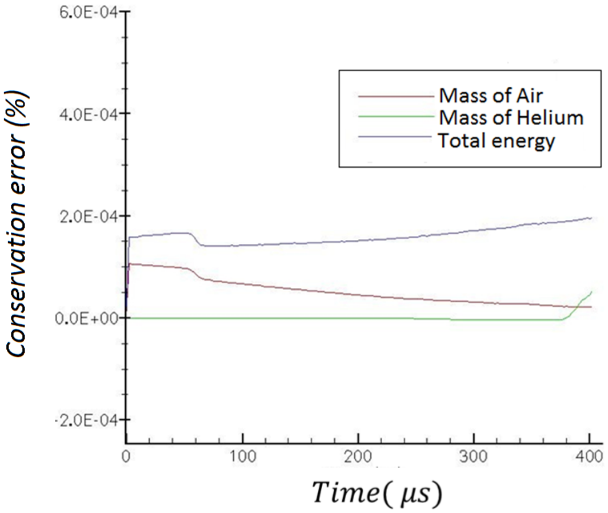


Figure 4.12: Conservation errors for the Helium impact testcase.

multimaterial interaction, the inflow/outflow values of mass and energy coming in/out of the domain respectively are subtracted from the total mass/energy of the domain in order to keep track on the energy/mass transfer at the interface between components.

5

An Eulerian solver for fluid/solid interaction

5.1 Constitutive model for solids

5.1.1 Hyperbolic system of conservation laws

For solid materials, the system of equations also needs to be expressed in hyperbolic conservative form. The model used is developed in [29] and [12] and is parametrized by 13 variables, which are the velocity vector, the elastic deformation gradient $\mathbf{F} = [\mathbf{F}_{ij}]$, and entropy $\mathcal{S}$. The vector of conserved variables and the fluxes appearing in Equation (2.2) are (the source term being detailed later):

$$\mathbf{U} = \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{F}^T e_1 \\ \rho\mathbf{F}^T e_2 \\ \rho\mathbf{F}^T e_3 \\ \rho(\xi + |\mathbf{u}|^2/2) \end{pmatrix}, \quad F^{(i)} = \begin{pmatrix} \rho\mathbf{u}u_i - \sigma\mathbf{e}_i \\ \rho\mathbf{F}^T\mathbf{e}_1 u_i - u_1\rho\mathbf{F}^T\mathbf{e}_i \\ \rho\mathbf{F}^T\mathbf{e}_2 u_i - u_2\rho\mathbf{F}^T\mathbf{e}_i \\ \rho\mathbf{F}^T\mathbf{e}_3 u_i - u_3\rho\mathbf{F}^T\mathbf{e}_i \\ \rho(E + |\mathbf{u}|^2/2)u_i - \mathbf{e}_i^T.(\sigma\mathbf{u}) \end{pmatrix} \tag{5.1}$$

where ξ is the internal energy, and the density is a function of the deformation gradients and a reference density corresponding to the unstressed state ρ_0:

$$\rho = \frac{\rho_0}{det|\mathbf{F}|} \tag{5.2}$$

and $\mathbf{e}_i$ are the Cartesian unit vectors. The solid materials shall be considered to be hyperelastic such that the Cauchy stresses are formulated in terms of the Finger strain tensor $\mathbf{G} = \mathbf{F}^{-T}\mathbf{F}^{-1}$ and the derivatives of the internal energy ξ:

$$\sigma_{ij} = -2\rho G_{ik}\frac{\partial \xi}{\partial G_{kj}} \tag{5.3}$$

Furthermore, the materials are considered to be isotropic and thus the specific internal energy function is formulated in terms of the principle invariants of $\mathbf{G}$. The specific form is taken to be the empirical one from [74]:

$$\xi(J_1, J_2, J_3, \mathcal{S}) = \frac{c_0^2}{2\alpha^2}(J_3^{\alpha/2} - 1)^2 + C_v T_0 J_3^{\gamma/2}(\exp(\mathcal{S}/C_v) - 1) + \frac{b_0^2}{2} J_3^{\beta/2}(J_1^2/3 - J_2), \tag{5.4}$$

with:

$$J_1 = tr(\mathbf{G}), \quad J_2 = \frac{1}{2}\left[(trg(\mathbf{G}))^2 - tr(\mathbf{G}^2)\right], \quad J_3 = det|\mathbf{G}| = (\rho/\rho_0)^2 \tag{5.5}$$

where c_0, b_0, T_0 are reference values of the longitudinal wave speed, shear wave speed, and temperature respectively; and α, β, γ are material dependent interpolation constants controlling the non-linear dependence of the elastic moduli on the compressibility.

For elastoplastic solids, the source term present in Equation (2.2) can actually be split in two terms (see [12]):

$$S = S^C + S^P = -\begin{pmatrix} 0 \\ u_1\beta \\ u_2\beta \\ u_3\beta \\ 0 \end{pmatrix} - \frac{1}{2G\tau}\begin{pmatrix} 0 \\ (\sigma'.\rho\mathbf{F})\mathbf{e}_1 \\ (\sigma'.\rho\mathbf{F})\mathbf{e}_2 \\ (\sigma'.\rho\mathbf{F})\mathbf{e}_3 \\ 0 \end{pmatrix} \tag{5.6}$$

with $\beta = \nabla.\rho F^T$, and:

$$S^C = -\begin{pmatrix} 0 \\ u_1\beta \\ u_2\beta \\ u_3\beta \\ 0 \end{pmatrix}, \; S^P = -\frac{1}{2G\tau}\begin{pmatrix} 0 \\ (\sigma'.\rho\mathbf{F})\mathbf{e}_1 \\ (\sigma'.\rho\mathbf{F})\mathbf{e}_2 \\ (\sigma'.\rho\mathbf{F})\mathbf{e}_3 \\ 0 \end{pmatrix} \tag{5.7}$$

S^C (named as compatibility source term) is a vector added to the equations for $\mathbf{F}$ to ensure the divergent form of the system, and can be treated as source [66]. Whilst these terms can be eliminated from the system, the equations for $\mathbf{F}$ then lose the desirable divergent form:

$$\frac{\partial \rho \mathbf{F}_{ij}}{\partial t} + \frac{\partial \rho u_k \mathbf{F}_{ij}}{\partial x_k} - \rho \mathbf{F}_{kj} \frac{\partial u_i}{\partial x_k} = -\frac{\rho}{2G\tau} \sigma_{ik} \mathbf{F}_{kj} \tag{5.8}$$

S^P (named as plasticity source term) contains terms governing inelastic deformations in a way analogous to the viscoplastic Maxwell solid model, limiting permissible values of the elastic gradients. The tensor $\sigma' = \sigma - Tr(\sigma)\mathbf{I}/3$ denotes the deviatoric stresses, $G = \rho b^2$ is the shear modulus. The relaxation time of tangential stresses τ specifies the plastic behavior, and the simplest discontinuous function $\tau = \infty$ if $\sigma_{eq} < \sigma_Y$, $\tau = 0$ if $\sigma_{eq} \geq \sigma_Y$, , where $\sigma_{eq} = \sqrt{3\sigma'_{ij}\sigma'_{ji}/2}$ is the Von-Mises stress, and σ_Y is a flow stress, would give idealised plasticity behaviour. Here viscoplastic behaviour is achieved using the following power law function [12]:

$$\tau = \tau_0 \left(\frac{\sigma_0}{\sigma_{eq}} \right)^n \tag{5.9}$$

where τ_0 is a reference time, σ_0 is a reference flow stress, and n is a material specific exponent controlling the rate dependency. Both models are hyperbolic and have complete sets of eigenvalues and eigenvectors. Their complete expression is quite tedious (see [12]), hence for sake of clarity, they will only be presented in Appendix B.

5.1.2 Numerical solvers

Since the suitability of a single chosen time-integration method may differ depending the terms appearing in the inhomogeneous systems of governing models for either material, a time-operator splitting approach is employed that approach has been mentioned in Chapter 2). This comprises a preliminary update applied to the reduced system comprising only the convective terms and selected sources; the solution is used subsequently as initial conditions for solving the system of ordinary differential equations for the remaining sources. In general, the following systems are updated consecutively:

$$\frac{d\mathbf{U}}{dt} = -\nabla.F + S^{(1)} \tag{5.10}$$

$$\frac{d\mathbf{U}}{dt} = S^{(2)} \tag{5.11}$$

where the original source vectors can be split into either $S^{(1)}$ or $S^{(2)}$ as desired. More specifically, for solids, the initial update neglects only source terms associated with limiting elastic deformations due to irreversible plastic deformations thus leaving the convective terms and those terms associated with compatibility of the deformation gradient equations. These are solved using an unsplit explicit third-order TVD Runge-Kutta (see Equation (3.53)) integration method to ensure high resolution of shocks. Following this, the source terms for inelastic deformations are solved using an implicit time integration method, which is necessary to ensure resolution of the small relaxation times in the event of high-strain rate behaviours. Since the mechanism for controlling inelastic deformations for solids is volume-preserving, the solution of the associated source terms does not influence the level-set field. Hence, the cell merging method does not change on account of using the time-operator splitting method, merely this must be used following the explicit updates and prior to the evaluation of the source terms in the next fractional step.

For solids, the choice of Riemann solver is somewhat limited owing to the complexity of the governing equations. Although several solvers were examined in [74] for the special case of non-linear elasticity, the linearised method proved to be unsuitable only in limited circumstances. The limitations of linearised solvers are well known for fluids and circumstances where instability may occur are not as unique. As a result, more robust Riemann solvers have been proposed (see [75]) and any of these could be used in place of a linearised method. Here however a linearised solver is used for both solids and fluids for consistency.

The linearised approach has been presented in Chapter 4 for fluids. In general, it assumes that the Jacobian defined in Equation (2.4) is linearised and the coefficients evaluated from the arithmetic mean of left and right cell-centered states. All matrices in the spectral decomposition of the Jacobian, $\mathbf{A} = \mathbf{R}D\mathbf{L}$, are thus also constant. An exact solution of the Riemann problem is then trivial and explicit expressions can be determined for the governing models of both solids and fluids; for the sake of clarity these are relegated to the Appendix B. High order spatial accuracy is achieved by reconstructing the cell-volume averaged data along the direction for which the solution is sought using the weighted essentially non-oscillatory (WENO) method detailed in Chapter 2. More specifically, reconstruction of characteristic variables is employed to ensure monotonicity in the event of multiple jumps in state variables occurring within a single numerical stencil; such a scenario is increasingly likely for solid materials given the complex seven waves Riemann fan.

The source term corresponding to compatibility conditions defined in Equation (5.7), associated with ensuring the divergent form of the equation set, can be included in the explicit update of the conserved variables according to Equation (3.38). The spatial derivatives within S^C are solved using a finite volume discretisation with values at the cell faces evaluated using the solution of the Riemann problem [12, 51]:

$$S^C_{i,j,k} = -\begin{pmatrix} 0 \\ u_{1_{i,j,k}}(\nabla.(\rho\mathbf{F}))_{i,j,k} \\ u_{2_{i,j,k}}(\nabla.(\rho\mathbf{F}))_{i,j,k} \\ u_{3_{i,j,k}}(\nabla.(\rho\mathbf{F}))_{i,j,k} \\ 0 \end{pmatrix} \tag{5.12}$$

with:

$$\begin{aligned}(\nabla.(\rho\mathbf{F}))_{i,j,k} \;=\; & \frac{(\rho\mathbf{F}^T)_{i+1/2}\mathbf{e}_1 - (\rho\mathbf{F}^T)_{i-1/2}\mathbf{e}_1}{\Delta x_{1_i}} + \frac{(\rho\mathbf{F}^T)_{j+1/2}\mathbf{e}_2 - (\rho\mathbf{F}^T)_{j-1/2}\mathbf{e}_2}{\Delta x_{2_j}} + \\ & \frac{(\rho\mathbf{F}^T)_{k+1/2}\mathbf{e}_3 - (\rho\mathbf{F}^T)_{k-1/2}\mathbf{e}_3}{\Delta x_{3_k}} \end{aligned} \tag{5.13}$$

As a result of the finite-volume discretisation, however, associated boundary terms will also exist. Details of the modification of the fluxes acting on the interface to include these additional terms shall be discussed in the next section.

5.2 Riemann solver for fluid/solid interaction

Within mixed cells, the states of the two contacting materials constitute a Riemann problem orientated normal to the interface. In the same way as the numerical method for single component cell-boundary fluxes, the solution of the Riemann problem can be used to accurately predict the state from which the interfacial flux can be computed. The difficulty with solving this in mixed cells is that the Riemann problem might comprise different materials, hence different equation sets, and the requirement to impose specific interfacial boundary conditions. In [10] these problems were addressed for solid/solid and solid/vacuum systems. It was shown that through careful consideration of the characteristic relations for the materials involved, a linearised method can be derived for which the interfacial boundary conditions provide the closure relations. By comparison, it is found that the solid/fluid Riemann problem solution following this approach is not dissimilar. For multi-dimensional problems, to account for the arbitrary angle of the interface with respect to the Cartesian axis, the state variables for both materials within a mixed cell must be rotated on to the orthogonal coordinate system $\vec{\mathbf{x}'_i}(\mathbf{n})$, where $\vec{\mathbf{x}'_1} = \mathbf{n}$, and the tangential directions can be taken arbitrarily.

If $\mathbf{R}^{ROT} = (\vec{x_1}, \vec{x_2}, \vec{x_3})^T$ denotes the tensor of direction cosines of $\mathbf{x}'$ with respect to $\mathbf{x}$ then the transformation of state variables onto the interface normal can be written:

$$\begin{pmatrix} \mathbf{u}' \\ \mathbf{F}' \\ \xi' \end{pmatrix} = \begin{pmatrix} \mathbf{R}^{ROT}\mathbf{u} \\ \mathbf{R}^{ROT}\mathbf{F}\mathbf{R}^{ROT^T} \\ \xi' \end{pmatrix} \tag{5.14}$$

Given the coordinates of $\mathbf{n} = (n_1, n_2, n_3)$ (computed thanks to the level-set field), the two tangential vectors are given by (see [52]):

$$\vec{\mathbf{x}'_2} = \frac{1}{\sqrt{2(1 - n_1 n_2 - n_1 n_3 - n_2 n_3)}} \begin{pmatrix} (n_2 - n_3) \\ (n_3 - n_1) \\ (n_1 - n_2) \end{pmatrix} \tag{5.15}$$

$$\vec{\mathbf{x}'_3} = \frac{1}{\sqrt{2(1 - n_1 n_2 - n_1 n_3 - n_2 n_3)}} \begin{pmatrix} (n_1(n_2 + n_3) - n_2^2 - n_3^2 \\ (n_2(n_1 + n_3) - n_1^2 - n_3^2 \\ (n_3(n_1 + n_2) - n_1^2 - n_2^2 \end{pmatrix} \tag{5.16}$$

That choice failing when $n_1 = n_2 = n_3 = \pm 1/\sqrt{3}$, it is picked only when $|n_2 + n_3| \leq |n_2 - n_3|$. In the other cases when $|n_2 + n_3| > |n_2 - n_3|$, the following choice is picked:

$$\vec{\mathbf{x}'_2} = \frac{1}{\sqrt{2(1 + n_3(n_2 - n_1) + n_1 n_3)}} \begin{pmatrix} (n_2 + n_3) \\ (n_3 - n_1) \\ -(n_1 + n_2) \end{pmatrix} \tag{5.17}$$

$$\vec{\mathbf{x}'_3} = \frac{1}{\sqrt{2(1 + n_3(n_2 - n_1) + n_1 n_3)}} \begin{pmatrix} (n_1(n_3 - n_2) - n_2^2 - n_3^2 \\ (n_2(n_1 + n_3) + n_1^2 + n_3^2 \\ (n_3(n_1 - n_2) - n_1^2 - n_3^2 \end{pmatrix} \tag{5.18}$$

The initial conditions for the Riemann problem thus become: $\mathbf{U}_L = \mathbf{U}_{M1}$ and $\mathbf{U}_R = \mathbf{U}_{M2}$ where *M1* and *M2* denote material 1 and material 2 respectively. An example of the resultant Riemann problem is depicted in Figure 5.1 (where s and f subscripts denote the solid and fluid values respectively). For the solid, the solution comprises three non-linear waves, and for the fluid one; both materials are then separated by the central contact wave. Similar multi-material Riemann problems were considered in [12] for solid/solid and solid/vacuum systems with different interfacial boundary conditions and act as a basis for solid/fluid problems. The solution for each solid in these cases was determined through consideration of the appropriate invariants for the three non-linear waves and contact (10 invariant relations), with the system closed assuming that the traction features in the specific boundary conditions, thus providing the remaining three invariant relations. From this analysis, assuming linearisation of the coefficients appearing in the invariants, in general the solution for the interfacial state for the solid can be obtained through evaluation of:

$$\begin{pmatrix} \tilde{\mathbf{u}'} \\ \tilde{\mathbf{F}'}^T \mathbf{e}'_1 \\ \tilde{\mathbf{F}'}^T \mathbf{e}'_2 \\ \tilde{\mathbf{F}'}^T \mathbf{e}'_3 \\ \tilde{\mathcal{S}'} \end{pmatrix} = \begin{pmatrix} \mathbf{u}' \\ \mathbf{F}'^T \mathbf{e}'_1 \\ \mathbf{F}'^T \mathbf{e}'_2 \\ \mathbf{F}'^T \mathbf{e}'_3 \\ \mathcal{S}' \end{pmatrix} + \frac{1}{\rho'} \begin{pmatrix} \varepsilon \mathbf{Q}^{-1} \mathcal{D}^{-1} \mathbf{Q} (\tilde{\sigma'} - \sigma') \mathbf{e}'_1 \\ (\mathbf{F}'^T \mathbf{e}'_1) \otimes (\mathbf{e}'^T_1 \Omega^{-1}) (\tilde{\sigma'} - \sigma') \mathbf{e}'_1 \\ (\mathbf{F}'^T \mathbf{e}'_1) \otimes (\mathbf{e}'^T_2 \Omega^{-1}) (\tilde{\sigma'} - \sigma') \mathbf{e}'_1 \\ (\mathbf{F}'^T \mathbf{e}'_1) \otimes (\mathbf{e}'^T_3 \Omega^{-1}) (\tilde{\sigma'} - \sigma') \mathbf{e}'_1 \\ 0 \end{pmatrix} \tag{5.19}$$

where $\varepsilon = \pm 1$ is a parameter used to distinguish the direction of wave propagation: $\epsilon = -1$ if solving for solid material on the right and $\epsilon = +1$ for solid material on the left. A tilde is used to denote the desired values at the boundary, whilst all other parameters are assumed to be evaluated from the cell centered state. The featured acoustic tensor is defined as:

$$\Omega_{ij} = \frac{1}{\rho'} \frac{\partial \sigma'_{1i}}{\partial F'_{jk}} \mathbf{F}'_{1k} = \mathbf{Q}^{-1} \mathcal{D}^2 \mathbf{Q} \tag{5.20}$$

where $\mathbf{Q}$ is an orthogonal matrix, and $\mathcal{D} = diag(\sqrt{\lambda_{ac1}}, \sqrt{\lambda_{ac2}}, \sqrt{\lambda_{ac3}})$ is the diagonal matrix of positive eigenvalues (illustrated on Figure 5.1). The solution for the state of the solid is thus uniquely determined once the traction $\tilde{\sigma'} \mathbf{e}'_1$ is specified.

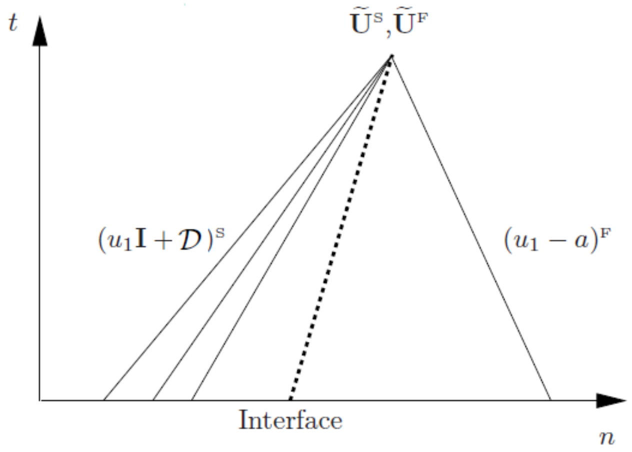


Figure 5.1: n-t plot (n indicating the direction normal to the interface) of an example solid/fluid Riemann problem solution for solid on the left in contact with a fluid on the right.

For the fluid, the Generalised Riemann invariants give the following differential relationships:

$$\begin{cases} \rho a du \pm dP = 0 & \text{along the } u \pm a - \text{curve} \\ d\rho - \frac{dP}{a^2} = 0 & \text{along the } u - \text{curve} \end{cases} \tag{5.21}$$

A similar linearisation than in Equation (5.19) and subsequent substitution through integration of Equation (5.21) (where only one of the two curves $u \pm a$ is picked depending on the side where the fluid lies) leads to the interfacial solution:

$$\begin{pmatrix} \tilde{\mathbf{u}}' \\ \tilde{\rho}' \\ \tilde{P}' \end{pmatrix} = \begin{pmatrix} \mathbf{u}' \\ \rho' \\ P' \end{pmatrix} + \begin{pmatrix} -\varepsilon(1/\rho' a')(\tilde{P}' - P')\mathbf{e}'_1 \\ (1/a'^2)(\tilde{P}' - P') \\ \tilde{P}' - P' \end{pmatrix} \tag{5.22}$$

where ε is used in the same way as Equation (5.19).

In order to close the system, closure equations need to be addressed. They physically correspond to interfacial conditions. Slip conditions are used, resulting in continuity in the normal components of velocity and traction, while tangential stresses in the solid are zero. Indeed, the fluid being inviscid, no shear stress is applied to the solid. The conditions can thus be written (where s and f subscripts denote the solid and fluid values respectively):

$$\begin{cases} \tilde{\sigma'}^{S}_{11} = -\tilde{P'}^{F} \\ \tilde{u'}^{S}_{1} = \tilde{u'}^{F}_{1} \\ \tilde{\sigma'}^{S}_{12} = \tilde{\sigma'}^{S}_{13} = 0 \end{cases} \tag{5.23}$$

The normal component of traction for the fluid of course reduces to the hydrostatic pressure since deviatoric stresses are neglected. Using the boundary conditions, the normal components of velocity in Equations (5.19) and (5.22) can be equated and rearranged to give an explicit prediction of the interfacial pressure/normal stress:

$$\tilde{\sigma'}^{S}_{11} = -\tilde{P'}^{F} = \frac{\left[\varepsilon(u'^{F}_{1} - u'^{S}_{1}) + \frac{1}{\rho'^{S}}(\mathbf{Q}^{-1}\mathcal{D}^{-1}\mathbf{Q})^{S}_{11}\sigma'^{S}_{11} + (\frac{1}{\rho'^{F}a'^{F}})P'^{F}\right]}{\left[\frac{1}{\rho'^{S}}(\mathbf{Q}^{-1}\mathcal{D}^{-1}\mathbf{Q})^{S}_{11} - (\frac{1}{\rho'^{F}a'^{F}})\right]} \tag{5.24}$$

This in turn, substituted in Equation (5.22) gives explicitly the interfacial normal velocity:

$$\tilde{u'}^{S}_{1} = \tilde{u'}^{F}_{1} = u'^{F}_{1} + \varepsilon\frac{(\tilde{P}' - P')}{\rho' a'} \tag{5.25}$$

In the examples that follow, systems will be considered that include regions of vacuum and thus solution of the solid/vacuum Riemann problem can also emerge in mixed cells. The solution for the interfacial state of solids in this case follows directly from the boundary conditions:

$$\tilde{\sigma'}\mathbf{e}'_{1} = 0 \tag{5.26}$$

Using for example a solid/fluid boundary problem, the solution for the interfacial states would comprise an evaluation of the traction Equation (5.23) as a function of known left and right states, and subsequently using the result to permit evaluation of the linearised solution Equation (5.19) or Equation (5.22) for solids and fluids respectively, at this stage independent of one-another. Note that since the solution is linearised, a single evaluation, in particular for solids, may not necessarily ensure that the vector of state variables satisfies the desired interfacial traction. Instead, the solution can be evaluated iteratively, each successive evaluation using as initial conditions the prior determined solution. Typically, only a few iterations are required to yield convergence to machine accuracy. Once the solution has been computed, one needs only rotate the resultant state back to the Cartesian coordinate system using the inverse of Equation (5.14). The state solutions in the Cartesian frame can subsequently be used to compute the numerical flux functions acting on the interface. For either material, these will correspond to the Lagrangian system, thus the convective flux terms will only contain contributions due to traction acting on the boundary plane.

For fluids the interfacial fluxes are:

$$F_{i,j,k}^{int} = -\mathcal{A}\begin{pmatrix} \tilde{P}\mathbf{n} \\ 0 \\ \tilde{P}\mathbf{u}^T.\mathbf{n} \end{pmatrix} \tag{5.27}$$

For solids the interfacial fluxes are:

$$F_{i,j,k}^{int} = -\mathcal{A}\begin{pmatrix} \tilde{\sigma}\mathbf{n} \\ \tilde{\rho}\tilde{u_1}\tilde{\mathbf{F}}^T\mathbf{n} \\ \tilde{\rho}\tilde{u_2}\tilde{\mathbf{F}}^T\mathbf{n} \\ \tilde{\rho}\tilde{u_3}\tilde{\mathbf{F}}^T\mathbf{n} \\ (\tilde{\sigma}\mathbf{n})^T.\mathbf{u} \end{pmatrix} - \mathcal{A}\begin{pmatrix} 0 \\ u_{1_{i,j,k}}\tilde{\rho}\tilde{\mathbf{F}}^T\mathbf{n} \\ u_{1_{i,j,k}}\tilde{\rho}\tilde{\mathbf{F}}^T\mathbf{n} \\ u_{1_{i,j,k}}\tilde{\rho}\tilde{\mathbf{F}}^T\mathbf{n} \\ 0 \end{pmatrix} \tag{5.28}$$

The first term on the right-hand-side of the solid boundary flux stems from the convective flux functions. The second term is a result of the spatial derivatives in the artificial vector on the right-hand-side of the equations for $\mathbf{F}$, with $u_{\eta_{i,j,k}}$ denoting the volume averaged velocity components at the center of cell *i, j, k* at the current time level. This addition is an extension of the finite volume discretisation used in computation of the divergent terms appearing in the vector denoted S^C, defined in Equation (5.7) for regular (un-cut) cells as described in the previous section.

5.3 Numerical Examples

5.3.1 1D Results: Initial value problems

Initial value problems (IVP) are chosen that reflect the range of applications that are of interest here. Specifically the case of a high-pressured gas in contact with a solid, and a stressed solid in contact with a quiescent gas. Such initial value problems allow the analysis of the shock capturing performance of the numerical scheme if the solid is assumed to remain purely elastic and the reactions of gas neglected since the state profiles are then self-similar and it is possible to determine exact solutions in some cases.

First testcase

The first IVP to be considered is a high-pressure gas in contact with an elastic solid in a quiescent state:

$$\begin{aligned}\mathbf{U}^F &= \Big\{ u = 0\,km\,s^{-1},\quad \rho = 1.84 g\,cm^{-3},\quad P = 18.9 GPa,\quad \alpha = 0\,(reacted)\\ \mathbf{U}^S &= \left\{ u = 0\,km\,s^{-1},\quad \mathbf{F} = \begin{bmatrix} 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1 \end{bmatrix},\quad S = 0 kJ\,g^{-1}K^{-1} \right.\end{aligned} \tag{5.29}$$

with the fluid taken to be on the left and the solid on the right.

The solid material is taken to be copper with the material constants for the specific internal energy appearing on Equation (5.4) described in Table 5.1.

The gas is assumed to be reacted PBX-9404, with values presented in Table 5.1, taken from [27]. It should be noted that the PBX-9404, despite being a reactive gas governed by the reactive Euler equations described in Chapter 4, the EOS associated is a simple Gamma-law gas. Solutions are found for the spatial domain [0:1] and performed in the x-direction with the initial interface located at x=0.5. The domain is discretised using the range of 250, 500, 1000 cells. The initial conditions result in a left traveling rarefaction in the fluid and a right traveling shock in the solid.

Constants	Values	Units
Copper		
ρ_0	8.93	$[g\, cm^{-3}]$
c_0	4.651	$[km\, s^{-1}]$
b_0	2.141	$[km\, s^{-1}]$
C_v	$3.9.10^{-4}$	$[kJ\, g^{-1} K^{-1}]$
T_0	300	[K]
α	1.0	-
β	3.0	-
γ	2.0	-
PBX-9404		
γ	2.85	-
ρ_0	1.84	$[g\, cm^{-3}]$
Q	5.543	$[kJ\, g^{-1}]$

Table 5.1: Values of material constants for IVP 1

Exact solutions are found through modification of the iterative exact solver for solid materials presented in [11]. Here, an initial guess of the intermediate states is used to determine initial estimates of the wave speeds. The complete solution is determinable if one has knowledge of the initial left and right states and the wave speeds. Thus, the initial guess can be used to integrate across each wave in turn to determine the intermediate states adjacent to the contact surface. If the wave speeds are correct, then the desired boundary conditions across the contact should be satisfied; any residual error then in these boundary conditions reflects in errors in the values of the wave speeds and can be subsequently used to find refined values. Modification of this method then to cases of solid/fluid coupling is straightforward given that the boundary conditions are known (Equation (5.23)) and the characteristic system is known for both materials. Details of both single component Riemann solvers are presented in Appendix B.

Variants of the high-order components of the numerical method have been applied to facilitate an assessment of the performance of each. Specifically 1st-Order, WENO-3 and monotonicity-preserving WENO-5 (MPWENO-5) [8] reconstruction methods are tested, the latter WENO methods both used in conjunction with the 3rd-Order TVD Runge-Kutta time integration method. Note that, as in [11], it was found to be necessary to supplement the WENO-5 method with monotonicity- preserving constraints in order to suppress severe oscillations occurring behind the leading longitudinal shocks arising in Riemann problems for non-linear elastic materials. The additional constraints further restrict the formal CFL number depending on the chosen coefficients [8], however it is found that C = 0.6 can safely be used before instabilities become apparent, and is for clarity employed for calculations using the 1st-Order and WENO-3 approaches also. It is mentioned that in practice however, values of C closer to unity are permissible with these. At any time, the domain occupied by each of the materials is clearly distinguishable from the large change in density between each component across the interface which exhibits no smearing as expected since the partial states are stored of both materials that are present in mixed cells, rather than mixture quantities (Figure 5.2). Both the speeds and jumps in properties across each wave are captured correctly and the numerical calculations are in excellent agreement with the exact solution. The resolution of the shock wave is good for the WENO methods, being diffused across approximately five cells in the case of MPWENO-5. Whilst this might be more than what is achieved for similar IVP using the numerical methods for single-phase problems, some loss of resolution is expected due to the local reduction in order of accuracy near the contact surface. L1-errors and convergence orders of selected variables are tabulated in Table 5.2 for different grid sizes and variants of the numerical method. It is mentioned that orders of convergence are not expected to exceed unity on account of the discontinuities present in the solution; nonetheless, the error analysis provides a further means of justifying the high-order component of the proposed scheme. In this example the upgrade from 1st-Order to the WENO-3 method leads to significant improvements on the overall accuracy, with the errors in some variables using the 1st-order method on the finest grid achievable using the coarsest grid with the WENO-3 variant. The benefits of using the more costly MPWENO-5 method in comparison to the WENO-3 variant are less significant. This apparent degradation of the WENO-5 method can be attributed to the added numerical dissipation introduced through the required monotonicity-preserving constraints.

Second testcase

The second IVP consists in having a moving, stressed solid copper in contact with a quiescent unreacted PBX-9404. The solid and fluid states are summarized below:

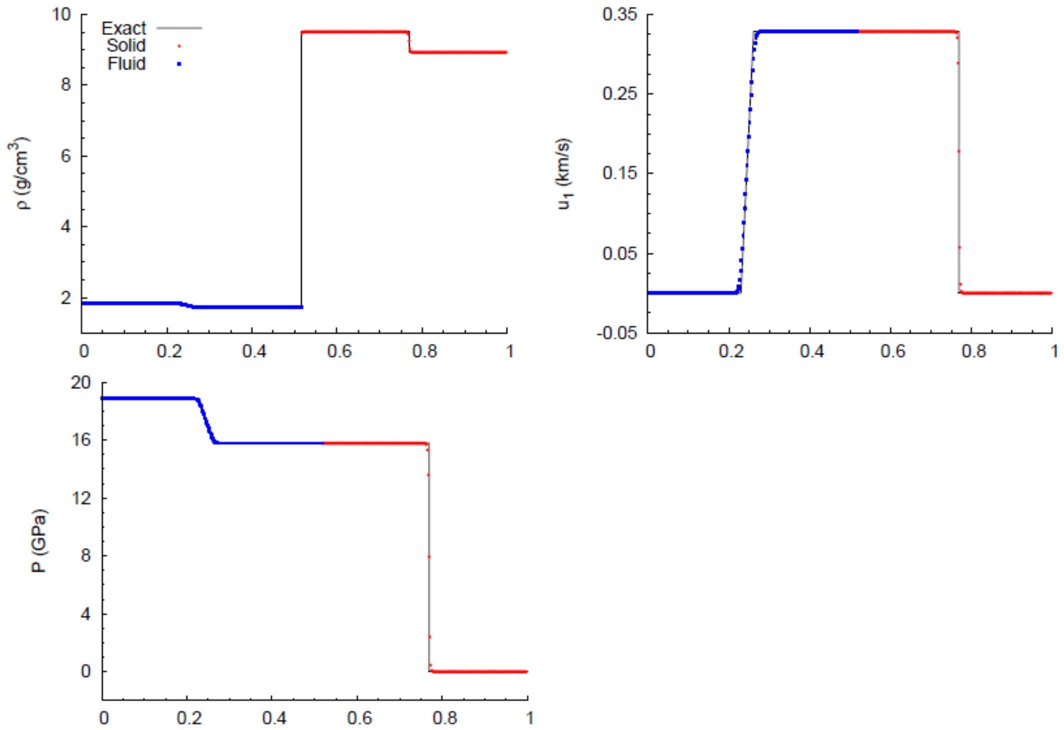


Figure 5.2: Comparison of exact (solid line) and numerical (points) solutions of the reacted PBX-9404/copper testcase at the time t = 0.5 µs using MPWENO-5 with C = 0.6, and x = 1/500 cm..

		ρ		u_1		P	
Method	N	L_1 Error	L_1 Order	L_1 Error	L_1 Order	L_1 Error	L_1 Order
1st-Order	250	$4.378{\cdot}10^{-2}$	–	$1.279{\cdot}10^{-2}$	–	$3.210{\cdot}10^{-1}$	–
	500	$2.304{\cdot}10^{-2}$	0.926	$8.069{\cdot}10^{-3}$	0.665	$1.909{\cdot}10^{-1}$	0.750
	1000	$1.201{\cdot}10^{-2}$	0.940	$4.999{\cdot}10^{-3}$	0.691	$1.089{\cdot}10^{-1}$	0.810
WENO-3	250	$3.496{\cdot}10^{-2}$	–	$3.782{\cdot}10^{-3}$	–	$9.558{\cdot}10^{-2}$	–
	500	$1.760{\cdot}10^{-2}$	0.990	$1.938{\cdot}10^{-3}$	0.965	$4.971{\cdot}10^{-2}$	0.943
	1000	$8.749{\cdot}10^{-3}$	1.009	$9.620{\cdot}10^{-4}$	1.010	$2.479{\cdot}10^{-2}$	1.004
MPWENO-5	250	$3.369{\cdot}10^{-2}$	–	$2.382{\cdot}10^{-3}$	–	$6.105{\cdot}10^{-2}$	–
	500	$1.700{\cdot}10^{-2}$	0.986	$1.252{\cdot}10^{-3}$	0.928	$3.365{\cdot}10^{-2}$	0.859
	1000	$8.420{\cdot}10^{-3}$	1.014	$5.986{\cdot}10^{-4}$	1.065	$1.593{\cdot}10^{-2}$	1.079

Table 5.2: L1 Errors and orders of convergence for Testcase 1, reacted PBX-9404 in contact with unstressed copper.

$$\mathbf{U}^F = \Big\{ u = 0\,km\,s^{-1}, \;\; \rho = 1.84 g\,cm^{-3}, \;\; P = 10^{-4} GPa, \;\; \alpha = 1\,(\text{unreacted})$$

$$\mathbf{U}^S = \left\{ \mathbf{u} = \begin{pmatrix} 2 \\ 0 \\ 0.1 \end{pmatrix} km\,s^{-1}, \;\; \mathbf{F} = \begin{bmatrix} 1 & 0 & 0 \\ -0.01 & 0.95 & 0.02 \\ -0.015 & 0 & 0.9 \end{bmatrix}, \;\; S = 0 kJ\,g^{-1} K^{-1} \right. \tag{5.30}$$

The material constants, domain, and discretisation parameters are taken to be the same as in the last case. This time however, the solid on the left of the fluid is in a state of three-dimensional deformation intended to reveal the complete set of non-dimensional waves. The initial state of the solid is the same as that used in the solid/vacuum IVP presented in [10], and since the conditions of pressure and velocity are much greater than the fluid, which also has a lower inertia than the solid, it can be expected that the subsequent resultant motion of the contact surface to be similar. The initial conditions result in a left traveling longitudinal rarefaction wave followed by two transverse shocks in the solid (the same as that for the solid/vacuum case), and a right traveling shock in the fluid. In comparison to the previous example, each of the non-linear waves is distinct and exhibits a jump in properties (Figure 5.3). Entropy errors are discernible in the density profiles of the fluid as an undershoot near the contact surface, an artifact well known to plague many multi-component methods. Whilst it is reasonable to assume that applying an entropy fix as proposed in [22] could alleviate these errors largely, such action would compromise the conservation of the scheme and is thus avoided.

Note that the severity of density errors is independent of the initial density ratio between the two components, and rather scales (apparently self-similarly) with the strength of the resultant shock. Hence similar errors can be expected to arise in gas shocked through driving a solid piston (infinite density) [22] and gas/gas Riemann problems (densities of the same order of magnitude) [23] for ghost fluid type methods. As before, the *L1*-errors and convergence orders are presented in Table 5.3 for variants of the method and similar conclusions can be drawn on the benefits of adding the reconstruction method as expected. In contrast to the previous example however, certain tangential state variables now exhibit a jump across each of the non-linear waves in the solid material. Experience shows that capturing these delicate features can prove challenging even when employing the high-resolution methods. The poor resolution is clearly visible when using first order accuracy (see Appendix D), where some non-linear

		ρ		u_2		P	
Method	N	L_1 Error	L_1 Order	L_1 Error	L_1 Order	L_1 Error	L_1 Order
1st-Order	250	$6.661{\cdot}10^{-2}$	–	$5.746{\cdot}10^{-4}$	–	$3.438{\cdot}10^{-1}$	–
	500	$3.306{\cdot}10^{-2}$	1.011	$3.756{\cdot}10^{-4}$	0.613	$1.908{\cdot}10^{-1}$	0.849
	1000	$1.687{\cdot}10^{-2}$	0.971	$2.482{\cdot}10^{-4}$	0.598	$1.123{\cdot}10^{-1}$	0.765
WENO-3	250	$4.414{\cdot}10^{-2}$	–	$3.397{\cdot}10^{-4}$	–	$1.395{\cdot}10^{-1}$	–
	500	$2.034{\cdot}10^{-2}$	1.118	$1.911{\cdot}10^{-4}$	0.830	$6.102{\cdot}10^{-2}$	1.193
	1000	$1.005{\cdot}10^{-2}$	1.017	$1.078{\cdot}10^{-4}$	0.826	$2.965{\cdot}10^{-2}$	1.041
MPWENO-5	250	$3.777{\cdot}10^{-2}$	–	$2.855{\cdot}10^{-4}$	–	$1.037{\cdot}10^{-1}$	–
	500	$1.856{\cdot}10^{-2}$	1.025	$1.523{\cdot}10^{-4}$	0.907	$5.096{\cdot}10^{-2}$	1.026
	1000	$9.088{\cdot}10^{-3}$	1.030	$8.110{\cdot}10^{-5}$	0.909	$2.673{\cdot}10^{-2}$	0.931

Table 5.3: *L1* Errors and orders of convergence for Testcase 2, deformed copper in contact with unreacted PBX-9404.

waves are not being captured. Again, it is seen that for the density and pressure, which undergo only small jumps between the left traveling longitudinal rarefaction and contact surface, the orders differ little between each of the schemes and accuracy of the WENO-3 and MPWENO-5 methods are comparable. However, when the errors and orders of the tangential velocity u_2 are considered, the benefits of the MPWENO-5 variant become more apparent, and the orders are slightly improved. From these two examples, it is clear that if the capturing of delicate tangential waves is important, then use of the MPWENO-5 method would be advised. However, this comes at an additional cost and if the importance of such features is inappreciable then the WENO-3 method is likely to suffice.

Third testcase

The third testcase consists again in a high-pressured gas in contact with a quiescent elastic copper plate. However, that testcase will involve a different EOS which is the JWL described in Chapter 4, section 4.2.3. This is to demonstrate the robustness of the solver in case of strong shocks even with more complex EOS than simple Gamma-law gases. The fluid considered is TNT, and the associated coefficients are given in Table 5.4. The exact solution has not been derived for that testcase due to more elaborated EOS, however, the results can be compared to the numerical results of [40] and [67] where that same testcase is carried out, whilst the impacted materials are modeled by a Cochran–Chan EOS, and treated as fluids.

The initial conditions are then given by:

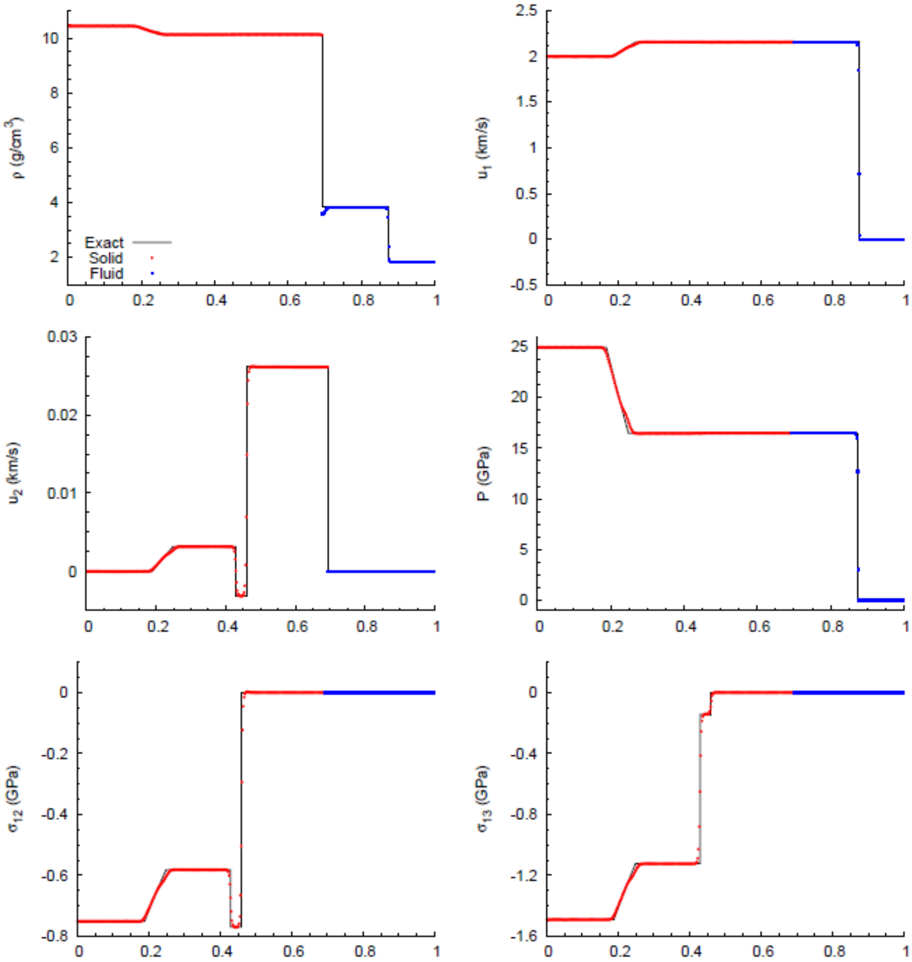


Figure 5.3: Comparison of exact (solid line) and numerical (points) solutions of the stressed copper/unreacted PBX-9404 testcase at the time t = 0.9 µs using MPWENO-5 with C = 0.6, and x = 1/500 cm.

Constants	Values	Units
TNT		
ρ_0	1.84	$[g\,cm^{-3}]$
A_0	854.5	[Gpa]
B_0	20.5	[Gpa]
R_1	4.6	-
R_2	1.35	-
e_0	1.0	[kJ]
Γ_0	0.25	-

Table 5.4: JWL constants for TNT

$$\mathbf{U}^F = \Big\{ u = 0\,km\,s^{-1}, \;\; \rho = 2.48537g\,cm^{-3}, \;\; P = 37GPa$$

$$\mathbf{U}^S = \left\{ u = 0\,km\,s^{-1}, \;\; \mathbf{F} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \;\; S = 0kJ\,g^{-1}K^{-1} \right. \qquad (5.31)$$

The domain and discretisation parameters are taken to be the same as in IVP 1 and 2. In contrast to the previous problems however, the CFL number has been set to C=0.9 to match the settings of [40]. The simulation is stopped when the real time is $t = 0.6\mu s$. Similarly to IVP 1, the solution consists in a rarefaction wave traveling into the fluid medium, while a shock wave propagates into the solid. The position of the shock is correctly captured with almost no smearing across the waves; the numerical calculations are in excellent agreement with [40] (see Figure 5.4). The differences observed in terms of shock locations and variables level come from the fact that the solid model is slightly different in terms of modelisation and related constants. Here again, WENO methods provide a significant improvement on the overall accuracy. Besides, the difference between WENO-3rd and MPWENO-5th is not worth the additional cost for the improvement shock resolution, which is why in the 2D and 3D testcases, WENO-3rd will be used.

5.3.2 2D Results: Simulation of a void collapse

A quiescent bubble of gamma law gas ($\gamma = 1.4$) is embedded within a copper matrix; both materials initially in an unstressed state. Upstream of the bubble

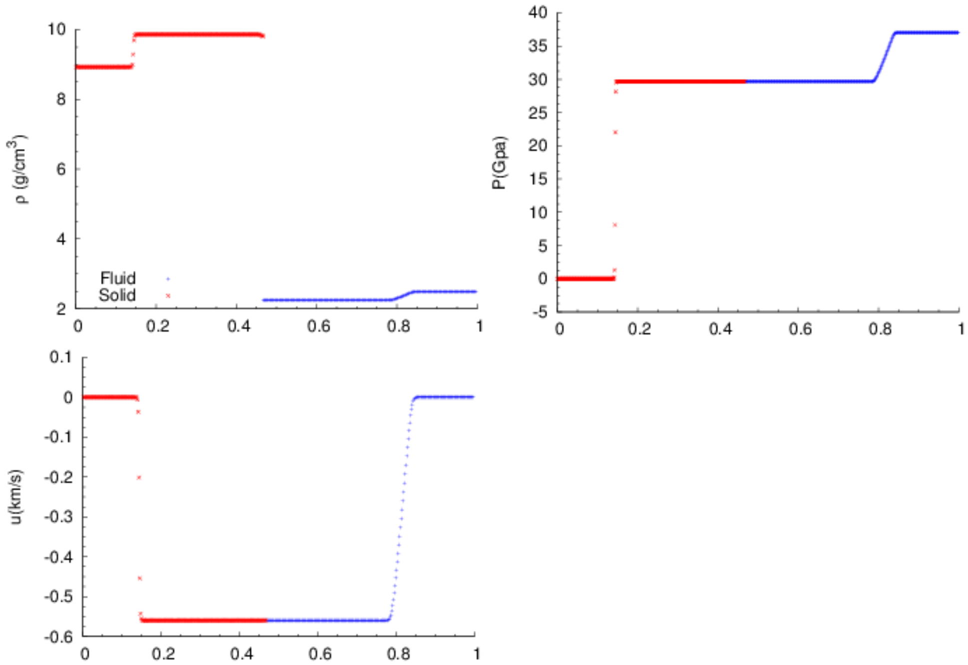


Figure 5.4: Numerical solution of the unstressed copper/TNT testcase at the time $t = 0.6$ µs using MPWENO-5 with $C = 0.9$, and $x = 1/500$ cm.

a shock is initiated in the solid that upon impacting results in the collapse of the gas bubble. The initial bubble radius is taken to be 1 cm centered at x=(4, 0) cm such that the y-axis represent a line of symmetry and the simulation conducted on this basis. A uniform grid was employed with a grid spacing $\Delta x = \Delta y = 0.02cm$. The top y=2 cm boundary is taken to be symmetric. A shock is generated in the solid by initialising the material x < 2.5 cm to have a velocity $u_x = 500ms^{-1}$. The case is illustrated on Figure 5.5. The material constants for the solid are taken to correspond to those employed in the previous one-dimensional initial value problems. In this case however, the solid is assumed to deform elastoplastically and the material parameters for the relaxation time model Equation (5.9) taken from [50] are as follows: $\tau_0 = 0.92s$, $\sigma_0 = 0.045GPa$, $n = 10.1$. The WENO-3 method is employed with C= 0.6. The example hereby is similar to the void (vacuum) collapse example in [76, 77] and the behaviours are comparable for the selected initial velocity. Indeed, the impulsive forces imparted on the bubble from the shock wave act to cause a (cylindrically) non-symmetric collapse, with overall dilation accompanied by jetting of the left hand quadrant of the bubble surface (Figure 5.6). The resultant deformations are large with the bubble area at late time a fraction of the initial state (Figure 5.7). Despite the large deformations, conservation errors do not exceed $\sim 0.1\%$ (Figure 5.8). It is likely that those that do arise can be attributed to errors generated through discretisation of the level-set equations as opposed to errors in the conservative discretisation of the governing physical equations. This also explains then the increasing level of error as the simulation progresses: the highly compressed area of gas rapidly contracts decreasing the resolution of the interface as localised curvatures increase; thus leading to increased error through regularisation in the solution of the level-set advection equation.

5.3.3 2D Results: Pipe explosion

A high-pressured gas bubble is surrounded by a quiescent water medium encircled by an aluminium pipe. That testcase has been proposed by Liu in [46]. The configuration is illustrated on Figure 5.9. This is a 3-components problem, involving both fluid/fluid and fluid/solid interactions, as well as solid-vacuum interaction. The pipe has indeed a finite thickness, and can largely deform in a vacuum region on the outside. The geometry is as follows: The pipe length is 350cm long, the bubble radius is 15cms, its distance to the pipe is 32.625 cm, and the thickness of the pipe is 19.05 cm (dimensions come from [46]). The axis x=0 is a symmetry axis. WENO-3rd is employed, with a CFL number C=0.9. The grid used is 80x250, with uniform spacing $\Delta x = \Delta y = 1.4.10^{-1}$cm.

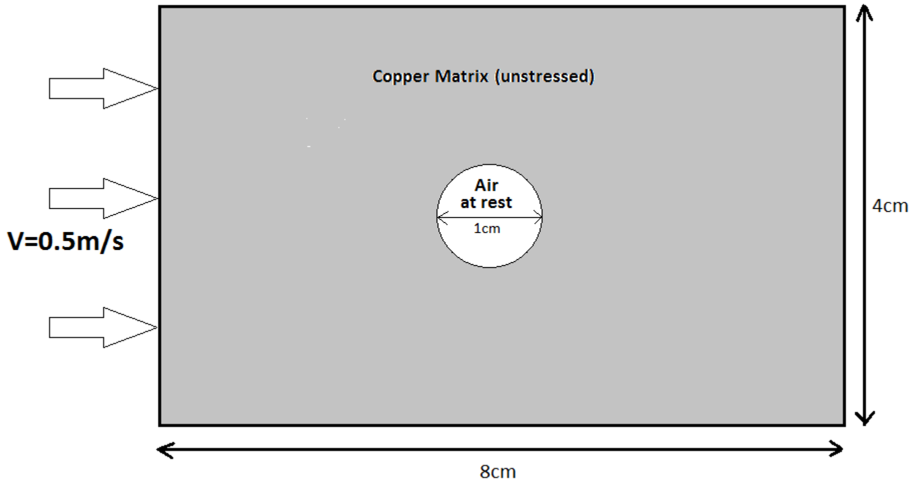


Figure 5.5: Void collapse testcase description.

The material constants are presented in Table 5.5. Air is considered as a perfect Gamma-law gas, and water is modeled with Tait's EOS, presented in Chapter 4. Aluminium is assumed to deform elastoplastically. Initial pressure of the air bubble is $P_0^{Air} = 2Gpa$ while water is at a quiescent state, i.e. $P_0^{Water} = 10^{-4}GPa$. The gas expands into the water medium and sends a shockwave which impacts the unstressed Aluminium, causing both a refracted wave back into the water and a transmitted shock in the solid which undergoes deformation. The wave propagation is very well captured (Figure) and matches the behaviour exhibited in [46], especially at the early stages, since cavitation is not taken into account, contrarily to the latter. The consecutive series of shocks suffers very little numerical dissipation. Besides, the conservation, whether mass or energy is nearly perfect ($< 10^{-4}\%$); compared to the other case, it is explained by the fact that the deformation is less significant for the solid.

5.3.4 3D Results: Simulation of a confined explosion

To demonstrate the scheme in three dimensions, the testcase of an explosion within a closed vessel proposed in [52] is considered. This comprises a copper can filled with PBX-9404, where all solid material is assumed to be in the reference unstressed state, whilst a layer of the fluid is assumed to be fully reacted and at high pressure and the remaining fluid in a quiescent unreacted state. The configuration is clearly illustrated on Figure 5.13. The geometry of the can is as

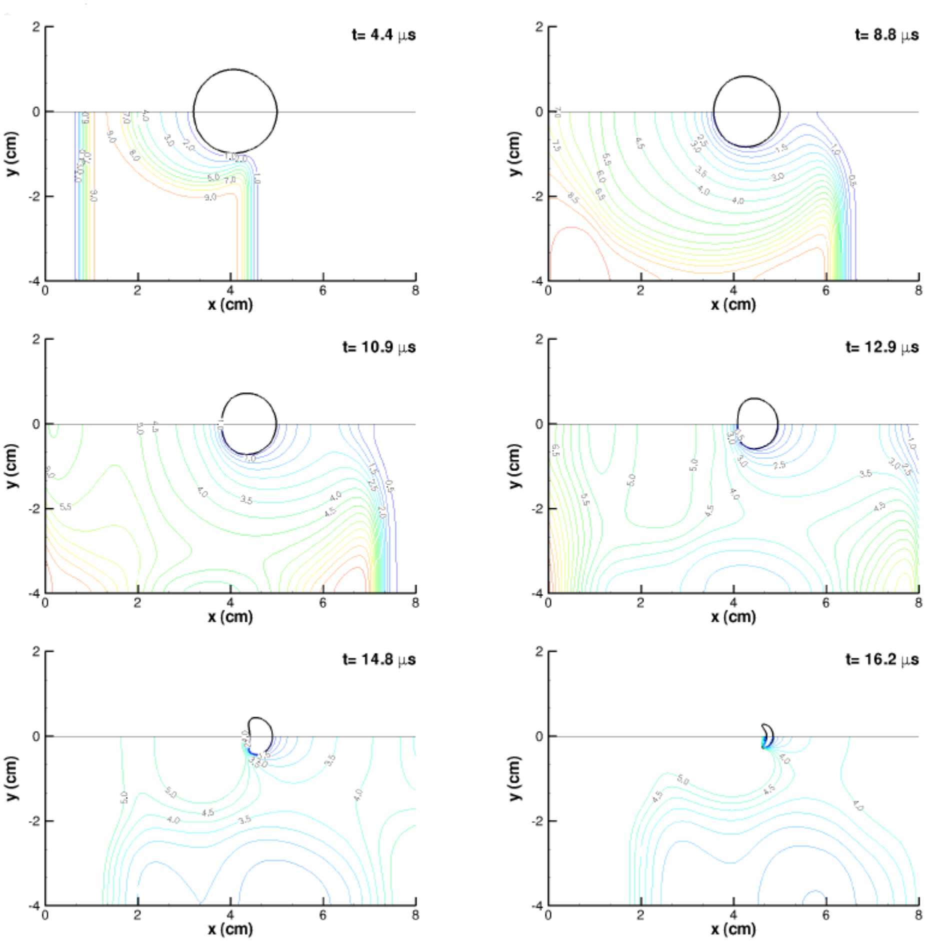


Figure 5.6: Pressure contours (bottom) and zero-level isocontour of the level-set field (top) for different time instances of the gas bubble collapse in a copper matrix example.

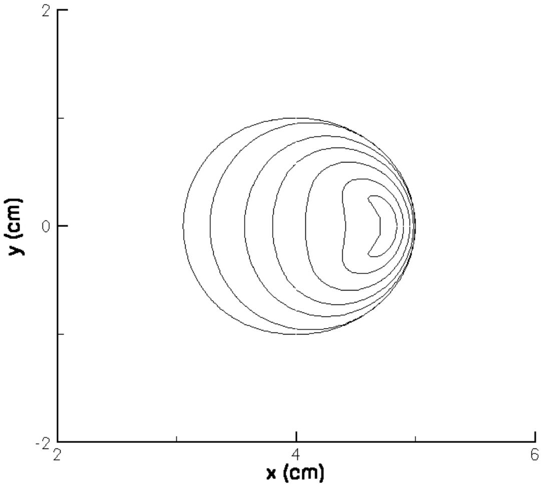


Figure 5.7: Comparison of the zero-level isocontour of the level-set field at different time instances of the gas bubble collapse in a copper matrix example.

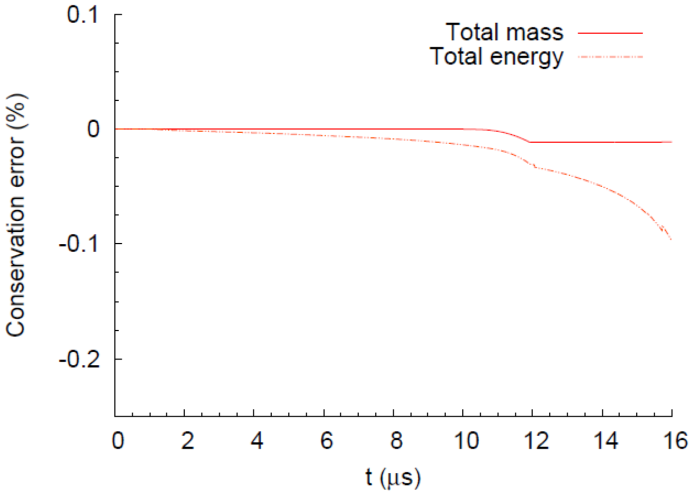


Figure 5.8: Mass and energy conservation errors for the void collapse example.

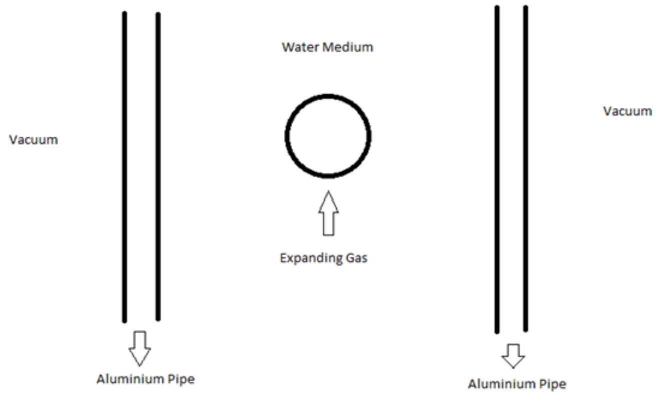


Figure 5.9: Pipe explosion testcase description.

Constants	Values	Units
Aluminium		
ρ_0	2.712	$[g\ cm^{-3}]$
c_0	6.22	$[km\ s^{-1}]$
b_0	3.16	$[km\ s^{-1}]$
C_v	9.10^{-4}	$[kJ\ g^{-1}K^{-1}]$
T_0	300	[K]
α	1.0	-
β	3.577	-
γ	2.088	-
Air		
γ	1.4	-
ρ_0	1.6	$[g\ cm^{-3}]$
P_0	2	$[GPa]$
Water		
N	4.4	-
P_{ref}	0.6	$[GPa]$

Table 5.5: Material constants for pipe explosion testcase.

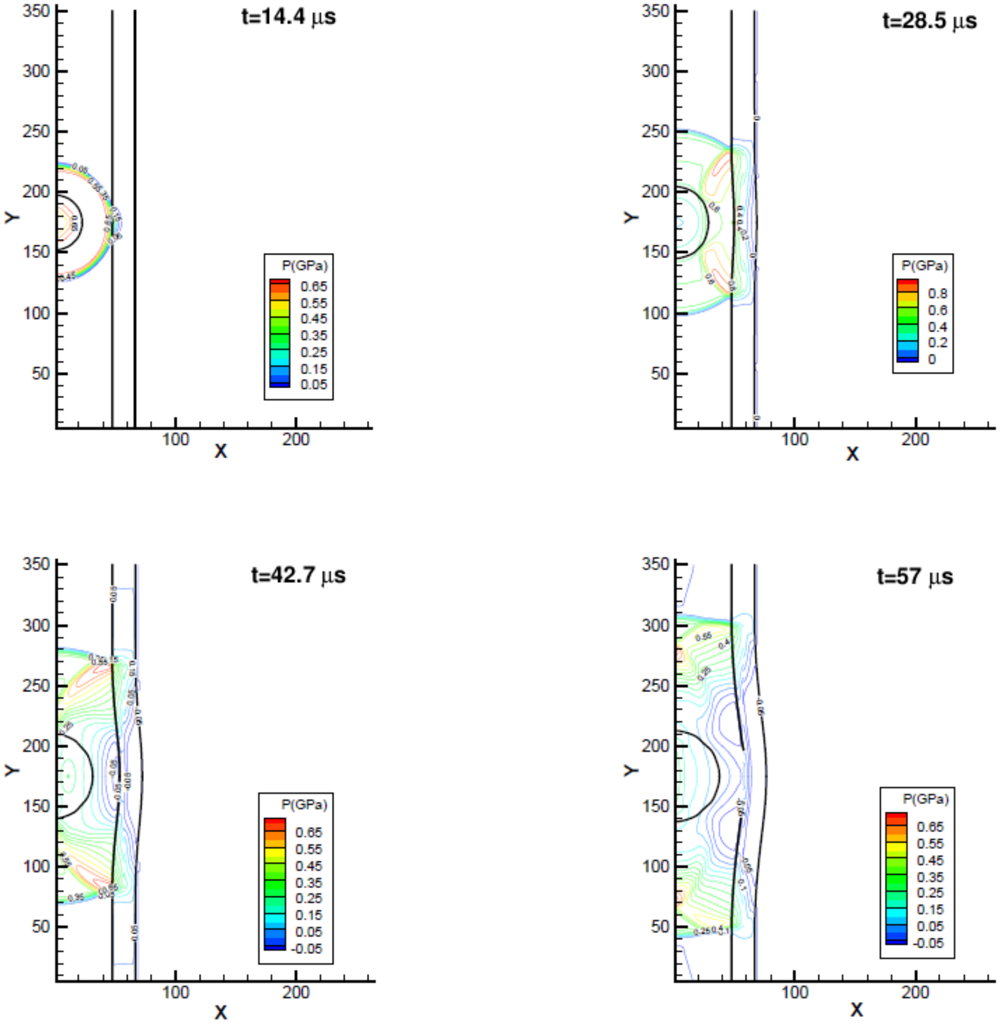


Figure 5.10: Pressure contours during the pipe deformation.

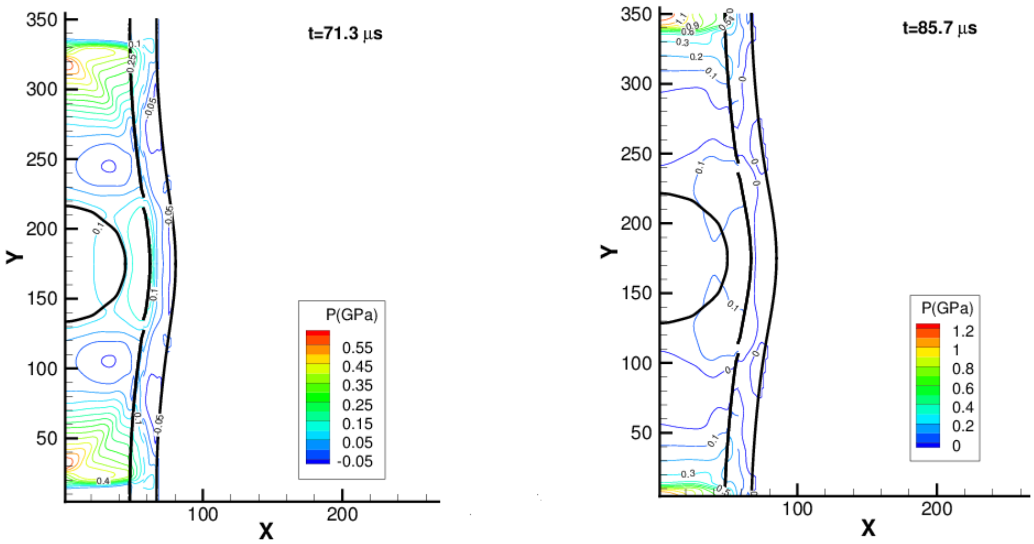


Figure 5.11: Pressure contours during the pipe deformation (continued)

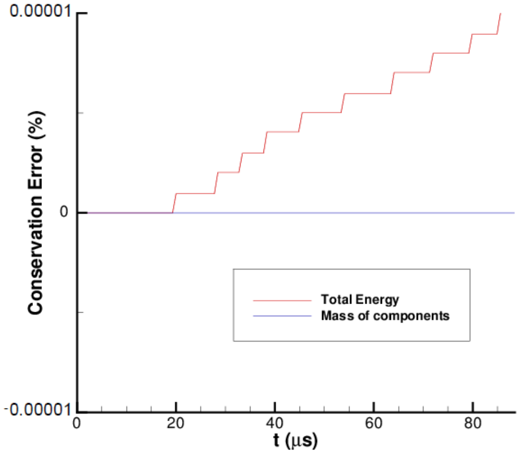


Figure 5.12: Conservation errors for the pipe explosion testcase (for each material, the mass conservation sticks strictly to 0).

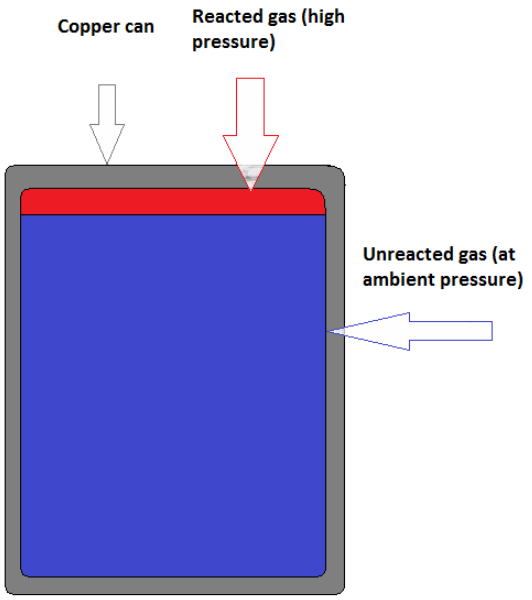


Figure 5.13: Confined can explosion description.

follows: the total length of the can was 24.6 cm, with an outside radius of 14 cm and thickness of 2 cm; the corners of the can are given a chamfer of 0.6 cm. A uniform grid was employed with a grid spacing $\Delta x = \Delta y = \Delta z = 0.2cm$. The detonation is initiated by assuming that a layer of the fluid at the top of the can with thickness 1.6 cm has the initial conditions corresponding to depleted chemical energy, $\alpha = 0$, whilst the remaining fluid is in a quiescent unreacted state. Again, the WENO-3 method is employed. Progress of the detonation is modeled using the programmed-burn described in Chapter 4, section 4.1.2. The relevant quantities to model the chemical reaction are the following: $q_0 = 5.543kJg^{-1}$ and $\dot{q}_0 = 100kJ.g^{-1}s^{-1}$.

The material constants for the fluid phase are taken to correspond to those employed in the previous one-dimensional initial value problems. For the elasto-plastic solid, the constants from the previous two-dimensional void collapse example are assumed. The initial conditions result in a detonation wave propagating down the length of the can (Figures 5.14 and 5.15) which subsequently reflects off the bottom wall leading to the formation of release waves. As expected the wall of the can is seen to displace in the regions aft of the detonation and a wave speed differential is seen across the thickness of the copper walls as a result the gradient in pressure between the solid/fluid and zero traction solid/void boundaries. At late times, the geometry exhibits large deformations in comparison to the initial profile, and the overall volume of the can has increased quite significantly under the explosive loading. Qualitatively, the results agree well with those from [52]. Conservation errors for the mass of each material and total energy of the system are again found to be small; not exceeding 0.15 %, see Figure 5.16.

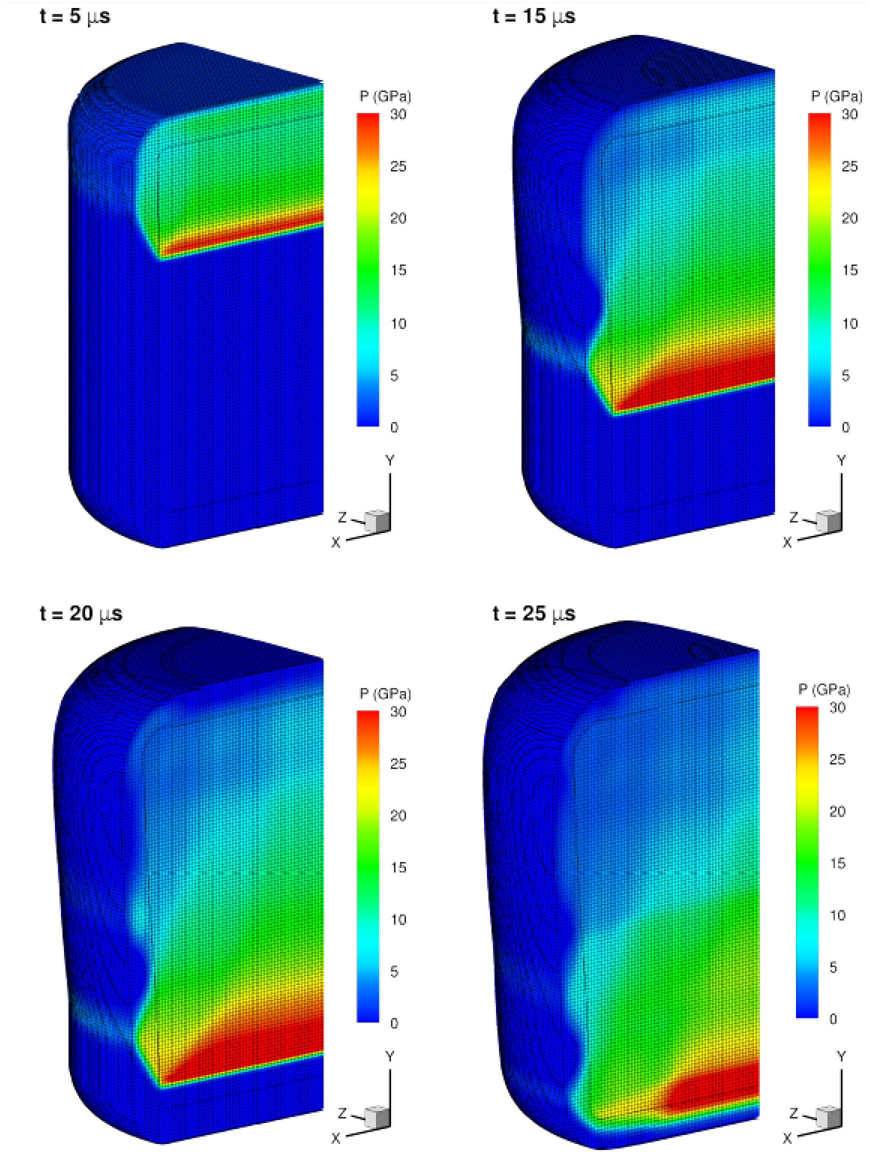


Figure 5.14: Pressure contours at the indicated times for the confined explosion example. Only those cells that contain either solid or fluid material are shown in order to highlight the geometries.

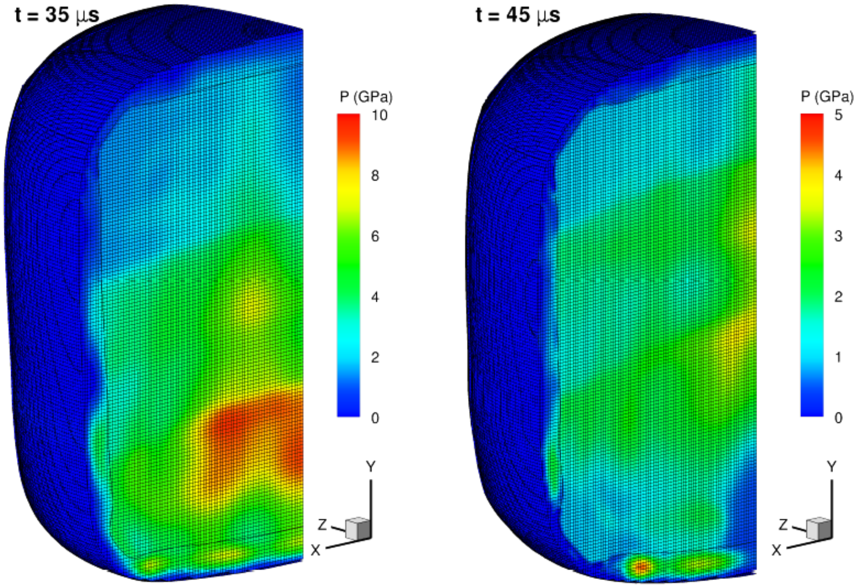


Figure 5.15: Pressure contours at the indicated times for the confined explosion example. Only those cells that contain either solid or fluid material are shown in order to highlight the geometries (continued).

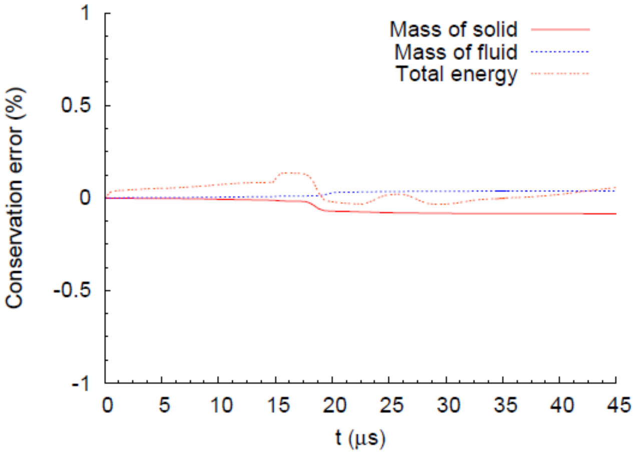


Figure 5.16: Mass and energy conservation errors for the confined explosion example.

6

Conclusions and future work

6.1 Conclusions

This thesis was devoted to proposing a numerical method capable of resolving simulations of systems comprising multiple compressible solid and fluid materials on fixed grids. For such simulations, it is paramount to maintain sharp interfaces between the immiscible components, and to resolve the complex wave patterns to the highest degree. The first challenge was to derive a robust interface tracking methods to achieve such features. Among the numerous methods available, a three-dimensional level-set based interface tracking method has been derived, valid for any kind of multimaterial problems provided that their set of equations are formulated under a hyperbolic system of conservation laws. Besides the ability of handling strong topological changes and large deformations, that method ensures full conservation even in the latter cases. This has been done by implementing a specific treatment to account for cut cells; the problem of small cells limiting the time-step is overcome by using a cell merging technique. Eventually, use of level-set functions allows an easy geometry reconstruction , particularly when dealing with three-dimensional geometries when other techniques face a higher challenge. It should also be noted that, by treating each material independently, high-resolution Godunov method for both solids and fluids are available.

Its first application has been in fluid/fluid interactions where its performance has been tested in 1D, 2D, and 3D testcases. Fluids are modeled with Euler equations, i.e. the required hyperbolic model. The 1D testcases were shock tube problems and have been carried out with various different EOS for each side of the interface, and high pressure gradients leading to strong shock interactions between the materials. The 2D and 3D testcases have proven the ability to capture the relevant physical phenomena occurring in problems with large deformations. In addition, the conservation errors have been proved negligible for these testcases.

The second stage of this thesis has been to apply the interface tracking method to solid/fluid problems each material having its set of equation. For solids and fluids, this relied upon using a hyperbolic model in conservative form. Contrary to similar alternative models, the variant employed for solids requires only thirteen constitutive transport laws to model elastoplastic deformations, decreasing the expense of what are already computationally intensive calculations in higher space dimensions As mentioned earlier, the interface tracking method had been developed regardless of the system of law considered. Nevertheless, it practically requires having a consistent Riemann solver to get an accurate state of each variable across the interface. Consequently, a dedicated Riemann solver for fluid/solid interaction has been derived for that purpose. The Riemann solver has been tested also in 1D, 2D and 3D testcases. The 1D testcases were various shock tube problems between fluids and solids in different states, with again high pressure gradients and impact velocities. The results have been proved in excellent agreement with the exact solution. Two-dimensional testcases have demonstrated the ability of the method in case of large deformations. Despite the rapidly changing in topology and increasing localised curvature of the material interface, mass and energy conservation errors were found to be negligible. Finally, the method was demonstrated in three-dimensions for the example of a confined explosion. The results agreed well with those presented elsewhere using alternative numerical methods and models for solid materials.

6.2 Future work

Much of the work represents the foundations of a more vast capability and there remains much further work to establish a general three-dimensional modeling technique that covers all physical behaviours exhibited in multiple fluid/solid interactions. Such improvements would be relying upon implementing more comprehensive models for each set of equations in order to account for more physics. Some examples are given below:

1. For solids:

    (a) Strain-hardening : One means of including work hardening is to add to the list of state variables the equivalent plastic strain [61]. Transport equations can thus be derived and added to the governing system where the mechanism for production of plastic strain is a function of the equivalent plastic strain rate, many models for which are available in the open literature. Besides, the latter methodology can be accompanied by a modification of the closure model for internal energy density to include a contribution due to strain hardening.

   (b) Thermal softening: The inclusion of thermal softening and melting within the proposed scheme does not require modifications of the bulk algorithms. In this work, the mechanism for inelastic deformations was governed by the relaxation time of tangential stresses, which here was considered merely as a function of the stress tensor components. In fact, the relaxation time would be a complex non-linear function of not only stress but also temperature. It is possible to formulate common constitutive models for flow stress that include thermal influences, such as the Johnson-Cook model [43].
   (c) Fracture mechanics: Incorporation of damage kinetics within the current model can be achieved using additional equations for the evolution of a scalar damage measure [14]; another alternative is to introduce pore volume per unit volume [63](both the latter exhibit similar form). Implications on the existing constitutive models manifest as additional terms in the closure model for internal energy density; specifically an additional energy function in terms of the chosen scalar parameter for the damage kinetics.

2. For fluids:

   (a) Viscous effects: The Euler equations have here been used and neglect both thermal conductivity and viscosity effects. Implementing such effects through additional terms in the momentum equation ([16]) notably would greatly improve the accuracy of these schemes for several applications, like hypersonics, or on the other hand, low speed flows. For another range of applications, it would be relevant to use other models that Euler equations, such as incompressible flows equations.
   (b) Reactive model: The detonation models used in Chapter 5 have relied on a simple reactive Euler model with a one-way exchange between potential chemical energy and internal energy. A more comprehensive model for explosive problems would be preferable.
   (c) Chemical reactions: For a wide range of applications, the fluid model could be enhanced in order to take into account effects such as gas dissociation or ionization (in case of hypersonic flows for instance [5]). The consecutive heating effects could be extremely relevant especially if the thermal softening of materials mentioned above is implemented in parallel.

Another field of improvement can concern the level-set method itself, more particularly when a cell contains multiple junctions, i.e. more than two materials; corners are also a source of instabilities, especially when trying to obtain the normals at the interface; this has been avoided in the testcases by rounding the corners (e.g. in section 5.3.4, where the can corners are modified by applying a chamfer). Attempts to circumvent these geometric issues has been proposed in [49], or the use of particle level set method [21, 78], where Lagrangian markers are introduced, coupled with adaptive mesh refinement (see also [56])in the under-resolved zones.

References

[1] R. Abgrall, and S. Karni, Computations of Compressible Multifluids, *J. Comp. Phys.*, Vol 169, 594-623 (2001).

[2] D Adalsteinsson, J.A Sethian, The Fast Construction of Extension Velocities in Level Set Methods, *J. Comp. Phys.* Vol 148 (1999), Issue 1, pp 2-22.

[3] A. Alia, M. Souli, High explosive simulation using multi-material formulations, *Applied Thermal Engineering* 26 (2006) 1032–1042.

[4] J. D. Anderson, Computational Fluid Dynamics: the basics with applications, 1995, Ed Mc Graw-Hill.

[5] J. D. Anderson, Hypersonic and High-Temperature Gas Dynamics Second Edition. AIAA Education Series (2006).

[6] N. Andrianov , G. Warnecke, The Riemann problem for the Baer–Nunziato two-phase flow model, *J. Comp. Phys.*, Vol 195 (2004) 434–464.

[7] A. Bagabir, D. Drikakis, Numerical experiments using high-resolution schemes for unsteady, inviscid, compressible flows, *Comput. Methods Appl. Mech. Engrg.* 193 (2004) 4675–4705.

[8] D. Balsara and C. Shu. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *J. Comp. Phys.*, Vol 160(2), 405 (2000).

[9] P. T. Barton, A novel multi-dimensional Eulerian approach to computational solid dynamics, PhD thesis 2009.

[10] P. T. Barton & D. Drikakis, An Eulerian method for multi-component problems in non-linear elasticity with sliding interfaces, *J. Comp. Phys.*,, Vol 229 (2010).

[11] P. T. Barton, D. Drikakis, E. Romenski & V. A. Titarev, Exact and approximate solutions of Riemann problems in non-linear elasticity, *J. Comp. Phys.*, Vol 228 7046 (2009).

[12] P. T. Barton, D. Drikakis & E. Romenski, An Eulerian finite-volume scheme for large elastoplastic deformations in solids, *Int. J. Num. Meth. Eng.*, Vol 81, 453 (2010).

[13] Christopher E. Brennen, Fundamentals of Multiphase Flows, Cambridge University Press (April 18, 2005).

[14] A. V. Bushman, Intense Dynamic Loading Of Condensed Matter, (CRC, 1992).

[15] J-A. Carillo and Al, Simulation of fluid and particles flows: Asymptotic preserving schemes for bubbling and flowing regimes, *J. Comp. Phys.*, Vol 227 (2008) pp. 7929-7951.

[16] P. Chassaing , Mécanique des fluides : élements d'un premier parcours, Ed Cépadues (1997).

[17] P. Colella, D. T. Graves, B. J. Keen & D. Modiano, A Cartesian grid embedded boundary method for hyperbolic conservation laws,*J. Comp. Phys.*, Vol 211, 347 (2006).

[18] D. A. Drew and S. L. Passman, Theory of multicomponent fluids, Ed Springer (1998).

[19] D. Drikakis, W. Rider, High-resolution methods for incompressible and low-speed flows, Ed Springer (2005).

[20] A. Eberle. Characteristic flux averaging approach to the solution of Euler's equations. In VKI Lecture Series, Computational Fluid Dynamics, 1987.

[21] D. Enright, R. Fedkiw, J. Ferziger & I. Mitchell, A Hybrid particle Level Set Method for Improved Interface Capturing, *J. Comp. Phys.*, Vol 183 (2002).

[22] R. P. Fedkiw, A. Marquina & B. Merriman, An Isobaric Fix for the Overheating Problem in Multimaterial Compressible Flows, *J. Comp. Phys.*, Vol 148 (1999).

[23] R. P. Fedkiw and Al, A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method), *J. Comp. Phys.*, Vol 152 (1999), pp. 457-492.

[24] R. P. Fedkiw, Coupling an Eulerian Fluid Calculation to a Lagrangian Solid Calculation with the Ghost Fluid Method, *J. Comp. Phys.*, Vol 175 (2002), pp. 200-224.

[25] R. P. Fedkiw and S. Osher, Level Set Methods and Dynamic Implicit Surfaces, Ed Springer (2003).

[26] R. P. Fedkiw and Al, An Isobaric Fix for the Overheating Problem in Multimaterial Compressible Flows, *J. Comp. Phys.*, Vol 148, Issue 2 (1999), pp 545-578.

[27] W. Fickett & L. M. Scherr, Numerical calculation of the cylinder test, LA-5906 (1975).

[28] S. K. Godunov. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. Matematicheski Sbornik 47, 271 (1959).

[29] S. K. Godunov & E. I. Romenskii, Elements of Continuum Mechanics and Conservation Laws, (Kluwer Academic/Plenum Publishers, 2003).

[30] R. N. Goldman, Area of planar polygons and volume of polyhedra, in J. Arvo (Ed.) Graphics Gems II: No. 2, (Morgan Kaufmann, 1994).

[31] S. Gottleib, C. Shu, and E. Tadmor, Strongly stability preserving high-order time discretization methods, *SIAM review*, 43:89-112, 2001.

[32] J. Grove, R. Manikoff, Anomalous reflection of shock wave at a fluid interface, *J. Fluid Mech.* 219 (1990) 313–336.

[33] J-F Haas & B. Sturtevant. Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities. *J. Fluid Mech.*, Vol 181 (1987), 41-76.

[34] A. Harten. High-resolution schemes for hyperbolic conservation laws. *J. Comp. Phys.*, Vol 135, 260 (1997).

[35] D. Hartmann and Al, Differential equation based constrained reinitialization for level set methods, *J. Comp. Phys.*,, Vol 227, Issue 14 (2008), Pages 6821-6845.

[36] D. Hartmann, The constrained reinitialization equation for level set methods, *J. Comp. Phys.*,, Vol 229, Issue 5 (2010), Pages 1514-1535.

[37] C. W. Hirt and Al, An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds, *J. Comp. Phys.*, Vol 135 (1997) pp. 203-216.

[38] X. Y. Hu and Al, A conservative interface method for compressible flows, *J. Comp. Phys.*, Vol 219 (2006) pp.553-578.

[39] X. Y. Hu, B. C. Khoo, An interface interaction method for compressible multifluids, *J. Comp. Phys.*, Vol 198 (2004) 35–64.

[40] X. Y. Hu and Al, On the HLLC Riemann solver for interface interaction in compressible multi-fluid flow, *J. Comp. Phys.* Vol 228 (2009) 6572–6589.

[41] H. Ji, F. S. Lien & E. Yee, Numerical simulation of detonation using an adaptive Cartesian cut-cell method combined with a cell-merging technique, *Comp. Fluids*, 39, 1041 (2010).

[42] G. S. Jiang & C. W. Shu, Efficient implementation of weighted ENO schemes, *J. Comp. Phys.*, Vol 126, 202 (1996).

[43] G. R. Johnson & W. H. Cook, A constitutive model and data for metals subjected to large strains, high strain rates and high temperatures, *in Proceedings of the Seventh International Symposium on Ballistics*, Hague, (1983).

[44] A .K. Kapila and Al, Two phase modeling of deflagration-to detonation transition in granular materials: Reduced equations, *Physics of Fluids*,, Vol 13, No 10 (2001).

[45] T.G Liu and Al, Ghost fluid method for strong shock impacting on material interface, *J. Comp. Phys.*, Vol 190 (2003) pp. 651-681.

[46] T.G Liu and Al, Multiphase modeling of dynamic fluid-structure interaction during close-in explosion, *International Journal for Numerical Methods in Engineering* (2008), pp 1019-1043.

[47] W. E. Lorensen & H. E. Cline, Marching Cubes: A high resolution 3D surface construction algorithm, Computer Graphics, 21, 163 (1987).

[48] A. Marquina & P. Mulet, A flux-split algorithm applied to conservative models for multicomponent compressible flows, *J. Comp. Phys.,* 185 (2003).

[49] B. Merriman, J. K. Bence, S. Osher, Motion of multiple junctions: a level set approach, *J. Comp. Phys.* Vo 112, (1994), pp 334-363.

[50] L. A. Merzhievsky, A. V. Tyagel'sky , Relation of dislocation kinetics with dynamic characteristics in modeling mechanical behaviour of materials, *Journal de Physique* C3 1988; 49:457–466.

[51] G. H. Miller & P. Colella, A High-Order Eulerian Godunov Method for Elastic-Plastic Flow in Solids, *J. Comp. Phys.*, Vol 167 (2001).

[52] G. H. Miller & P. Colella, A Conservative Three-Dimensional Eulerian Method for Coupled Solid-Fluid Shock Capturing, *J. Comp. Phys.*, Vol 183 (2002).

[53] C. Min, On reinitializing level-set functions, *J. Comp. Phys.*, Vol 229 (2010), pp. 2764-2772.

[54] J.P. Nougier, Méthodes de calcul numérique, Ed Masson (1989).

[55] R.R. Nourgaliev, T.N. Dinh, and T.G. Theofanous, Adaptive Characteristics-Based Matching for Compressible Multifluid Dynamics, *J. Comp. Phys.*, Vol 213 (2006).

[56] R. R. Nourgaliev & T. G. Theofanous, High-fidelity interface tracking in compressible flows: Unlimited anchored adaptive level set, *J. Comp. Phys.*, Vol 224 (2007).

[57] R. R. Nourgaliev, M. S. Liou, & T. G. Theofanous, Numerical prediction of interfacial instabilities: Sharp interface method (SIM), *J. Comp. Phys.*, Vol 227, 3940 (2008).

[58] M. V. Papalexandris and V. Deledicque, A conservative approximation to compressible two-phase flow models in the stiff mechanical relaxation limit, *J. Comp. Phys.*, Vol 227 (2008), pp. 9241-9270.

[59] R. B. Pember, J. B. Bell, P. Collela, W. Y. Crutchfield & M. L. Welcome, An adaptive Cartesian grid method for unsteady compressible flow in irregular regions, *J. Comp. Phys.*, Vol 120, 278 (1995).

[60] D. Peng and Al, A PDE-Based Fast Local Level Set Method, *J. Comp. Phys.*, Vol 155 (1999) pp. 410-438.

[61] B. J. Plohr & D. H. Sharp, A conservative Eulerian formulation of the equations for elastic flow, *Adv. Appl. Math.*, Vol 9, 418 (1988).

[62] J. Quirk & S. Karni, On the dynamics of a shock-bubble interaction, *J. Fluid Mech*, vol 318 (1996), pp. 129-163.

[63] E. I. Romenski, Deformation model for brittle materials and the structure of failure waves, *J. App. Mech. Tech. Phys.*, 48, 437 (2007).

[64] E. Romenski and E. F. Toro, Compressible Two-Phase Flows: Two-Pressure Models and Numerical Methods, tech. report, Isaac Newton Institute for Mathematical Sciences, 2004.

[65] E. Romenski and E. F. Toro, Conservative Hyperbolic Model for Compressible Two-Phase Flow with Different Phase Pressures and Temperatures, *Quart. Appl. Math.* 65 (2007), pp. 259–279.

[66] E. Romenski, Thermodynamics and hyperbolic systems of balance laws in continuum mechanics, in Godunov Methods: Theory and Applications, edited by E. F. Toro. (Kluwer Academic/Plenum Publishers, 2001).

[67] R. Saurel, R. Abgrall, A multiphase Godunov method for compressible multifluid and multiphase flows, *J. Comp. Phys.*, Vol 150 (1999) 425–467.

[68] R. Saurel, F. Petitpas, R. A. Berry, Simple and efficient relaxation methods for interfaces separating compressible fluids, cavitating flows and shocks in multiphase mixtures, *J. Comp. Phys.*, Vol 228 (2009) 1678–1712.

[69] T. Sheu , C.H. Yu, P.H. Chiu, Development of a dispersively accurate conservative level set scheme for capturing interface in two-phase flows, *J. Comp. Phys.*, Vol 228 (2009) 661–686.

[70] K.-M Shyue, A Fluid Mixture Type Algorithm for Compressible Multicomponent Flow with Mie-Grüneisen Equation of State, *J. Comp. Phys.*, Vol 171 (2001), pp. 678-707.

[71] M.B. Sun, Z.G. Wang, X. S. Bai, Assessment and modification of sub-cell fix method for re-initialization of level-set distance function, *Int. J. Numer. Meth. Fluids* 2010; 62: 211-236.

[72] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *J. Comp. Phys.*, Vol 162 (2000), pp. 301-337.

[73] H.S. Tang and F. Sotiropoulos, A Second-Order Godunov Method for Wave Problems in Coupled Solid-Water-Gas Systems, *J. Comp. Phys.*, Vol 151 (1999), pp. 790-815.

[74] V. A. Titarev, E. Romenski & E. F. Toro, MUSTA-type upwind fluxes for non-linear elasticity, *Int. J. Numer. Meth. Eng.*, 73 897 (2008).

[75] E.F Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics: a practical introduction, Ed Springer (1997).

[76] L.B. Tran, H.S. Udaykumar, Simulation of void collapse in an energetic material, *J. Power and Propulsion*, V22, N5, pp. 947 (2006).

[77] L.B. Tran, H.S. Udaykumar, A particle-level set-based sharp interface Cartesian grid method for impact, penetration, and void collapse, *J. Comp. Phys.*, Vol 193 (2004), Issue 2, 20, pp. 469-510.

[78] Z. Wang, J. Yang , F. Stern, ,An improved particle correction procedure for the particle level set method, *J. Comp. Phys.*, Vol 228 (2009), pp5819-5837.

[79] A. B. Wardlaw, Underwater explosion testcases, IHTR 2069 (1998).

[80] G. Yang, D. M. Causon & D. M. Ingram, Calculation of compressible flows about complex moving geometries using a three-dimensional Cartesian cut cell method, *Int. J. Numer. Meth. Eng.*, 33 1121 (2000).

[81] S. T. Zalesak. Fully multidimensional Flux-corrected transport. *J. Comp. Phys.*, 31:335{362, 1979.

[82] H.K. Zhao, T. Chan, B. Merriman, and S. Osher, A variational level set approach to multiphase motion, *J. Comp. Phys.*, Vol127 (1996), pp. 179-195.

[83] H.K. Zhao, S. Osher, B. Merriman & M. Kang, Implicit and Nonparametric shape reconstruction from unorganized data using a variational level set method, Comput. Vis. Image Und., 80 295 (2000).

[84] H.-K Zhao, H.-K., Osher, S. Fedkiw, R., ‘‘Fast Surface Reconstruction using the Level Set Method,’’ Proc. First IEEE Workshop on Variational and Level Set Methods, in conjunction with Proc. ICCV ’01, IEEE, 2001, pp. 194-202.

Appendixes

Appendix A: 3D implementation of the geometry reconstruction

The 2D implementation has been detailed in Chapter 3. The 3D implementation uses the same methodology, with Equation (3.28) becoming:

$$\underbrace{[(u_{i,j,k} - x_{min})^+]^2}_{x^+} + \underbrace{[(u_{i,j,k} - y_{min})^+]^2}_{y^+} + \underbrace{[(u_{i,j,k} - z_{min})^+]^2}_{z^+} = h^2 \tag{6.1}$$

provided that $\Delta x = \Delta y = \Delta z = h$, and that $(a)^+$ is defined in Equation (3.29). x^+, y^+ and z^+ are auxiliary variables to be used next. The following explicit relation for $u_{i,j,k}$ expressed in 2D by Equation (3.31) becomes then:

$$u_{i,j,k} = \begin{cases} x_{min} + h \\ \quad \text{if } x^+ \neq 0 \cup y^+ = 0 \cup z^+ = 0 \\ y_{min} + h \\ \quad \text{if } x^+ = 0 \cup y^+ \neq 0 \cup z^+ = 0 \\ z_{min} + h \\ \quad \text{if } x^+ = 0 \cup y^+ = 0 \cup z^+ \neq 0 \\ max(\frac{2(x_{min}+y_{min})+\sqrt{D}}{4}, \frac{2(x_{min}+y_{min})-\sqrt{D}}{4}) \\ \quad \text{with } D = 4(x_{min}+y_{min})^2 - 8(x_{min}^2 + y_{min}^2 - h^2) \\ \quad \text{if } x^+ \neq 0 \cup y^+ \neq 0 \cup z^+ = 0 \\ max(\frac{2(x_{min}+z_{min})+\sqrt{D}}{4}, \frac{2(x_{min}+z_{min})-\sqrt{D}}{4}) \\ \quad \text{with } D = 4(x_{min}+z_{min})^2 - 8(x_{min}^2 + z_{min}^2 - h^2) \\ \quad \text{if } x^+ \neq 0 \cup y^+ = 0 \cup z^+ \neq 0 \end{cases}$$

$$u_{i,j,k} = \begin{cases} max(\frac{2(y_{min}+z_{min})+\sqrt{D}}{4}, \frac{2(y_{min}+z_{min})-\sqrt{D}}{4}) \\ \quad \text{with } D = 4(y_{min}+z_{min})^2 - 8(y_{min}^2 + z_{min}^2 - h^2) \\ \quad \text{if } x^+ = 0 \cup y^+ \neq 0 \cup z^+ \neq 0 \\ max(\frac{2(y_{min}+y_{min}+z_{min})+\sqrt{D}}{6}, \frac{2(y_{min}+y_{min}+z_{min})-\sqrt{D}}{6}) \\ \quad \text{with } D = 4(x_{min}+y_{min}+z_{min})^2 - 12(x_{min}^2 + y_{min}^2 + z_{min}^2 - h^2) \\ \quad \text{if } x^+ \neq 0 \cup y^+ \neq 0 \cup z^+ \neq 0 \end{cases} \tag{6.2}$$

By alternatively looping these successive updates throughout the computational domain, distance to the original data set of points is obtained.

Finding the initial guess for the surface does not require a specific 3D treatment compared to the one detailed in section 3.3.

Solving the PDE however requires the evaluation of the curvature (second term of the right hand side of Equation (3.26)). It is performed by extension of the method proposed in [82]. The curvature term is approximated by using:

$$\nabla . \left(\frac{\nabla\phi}{|\nabla\phi|}\right) = \left(\frac{\phi_x}{|\nabla\phi|}\right)_x + \left(\frac{\phi_y}{|\nabla\phi|}\right)_y + \left(\frac{\phi_z}{|\nabla\phi|}\right)_z \tag{6.3}$$

so, at first order:

$$\begin{aligned} \nabla . \left(\frac{\nabla\phi}{|\nabla\phi|}\right)_{i,j,k} &= \left[\frac{\left(\frac{\phi_x}{|\nabla\phi|}\right)_{i+1/2,j,k} - \left(\frac{\phi_x}{|\nabla\phi|}\right)_{i-1/2,j,k}}{h}\right] + \left[\frac{\left(\frac{\phi_y}{|\nabla\phi|}\right)_{i,j+1/2,k} - \left(\frac{\phi_y}{|\nabla\phi|}\right)_{i,j-1/2,k}}{h}\right] \\ &\quad + \left[\frac{\left(\frac{\phi_z}{|\nabla\phi|}\right)_{i,j,k+1/2} - \left(\frac{\phi_z}{|\nabla\phi|}\right)_{i,j,k-1/2}}{h}\right] \end{aligned} \tag{6.4}$$

where:

$$\left(\frac{\phi_x}{|\nabla\phi|}\right)_{i+1/2,j,k} =$$

$$\frac{(\phi_{i+1,j,k}-\phi_{i,j,k})/h}{\sqrt{\begin{array}{c}\left[\frac{(\phi_{i+1,j,k}-\phi_{i,j,k})}{h}\right]^2+\left[\frac{1}{2}\left(\frac{(\phi_{i,j+1,k}-\phi_{i,j-1,k})}{2h}+\frac{(\phi_{i+1,j+1,k}-\phi_{i+1,j-1,k})}{2h}\right)\right]^2+\\ \left[\frac{1}{2}\left(\frac{(\phi_{i,j,k+1}-\phi_{i,j,k-1})}{2h}+\frac{(\phi_{i+1,j,k+1}-\phi_{i+1,j,k-1})}{2h}\right)\right]^2\end{array}}}$$

$$\left(\frac{\phi_x}{|\nabla\phi|}\right)_{i-1/2,j,k} =$$

$$\frac{(\phi_{i,j,k}-\phi_{i-1,j,k})/h}{\sqrt{\begin{array}{c}\left[\frac{(\phi_{i-1,j,k}-\phi_{i,j,k})}{h}\right]^2+\left[\frac{1}{2}\left(\frac{(\phi_{i,j+1,k}-\phi_{i,j-1,k})}{2h}+\frac{(\phi_{i-1,j+1,k}-\phi_{i-1,j-1,k})}{2h}\right)\right]^2+\\ \left[\frac{1}{2}\left(\frac{(\phi_{i,j,k+1}-\phi_{i,j,k-1})}{2h}+\frac{(\phi_{i-1,j,k+1}-\phi_{i-1,j,k-1})}{2h}\right)\right]^2\end{array}}}$$

$$\left(\frac{\phi_x}{|\nabla\phi|}\right)_{i,j+1/2,k} =$$

$$\frac{(\phi_{i,j+1,k}-\phi_{i,j,k})/h}{\sqrt{\begin{array}{c}\left[\frac{(\phi_{i,j+1,k}-\phi_{i,j,k})}{h}\right]^2+\left[\frac{1}{2}\left(\frac{(\phi_{i+1,j+1,k}-\phi_{i-1,j+1,k})}{2h}+\frac{(\phi_{i+1,j,k}-\phi_{i-1,j,k})}{2h}\right)\right]^2+\\ \left[\frac{1}{2}\left(\frac{(\phi_{i,j+1,k+1}-\phi_{i,j+1,k-1})}{2h}+\frac{(\phi_{i,j,k+1}-\phi_{i,j,k-1})}{2h}\right)\right]^2\end{array}}}$$

$$\left(\frac{\phi_x}{|\nabla\phi|}\right)_{i,j-1/2,k} =$$

$$\frac{(\phi_{i,j,k}-\phi_{i,j-1,k})/h}{\sqrt{\begin{array}{c}\left[\frac{(\phi_{i,j-1,k}-\phi_{i,j,k})}{h}\right]^2+\left[\frac{1}{2}\left(\frac{(\phi_{i+1,j,k}-\phi_{i-1,j,k})}{2h}+\frac{(\phi_{i+1,j-1,k}-\phi_{i-1,j-1,k})}{2h}\right)\right]^2+\\ \left[\frac{1}{2}\left(\frac{(\phi_{i,j-1,k+1}-\phi_{i,j-1,k-1})}{2h}+\frac{(\phi_{i,j,k+1}-\phi_{i,j,k-1})}{2h}\right)\right]^2\end{array}}}$$

$$\left(\frac{\phi_x}{|\nabla\phi|}\right)_{i,j,k+1/2} =$$

$$\frac{(\phi_{i,j,k+1}-\phi_{i,j,k})/h}{\sqrt{\begin{array}{c}\left[\frac{(\phi_{i,j,k+1}-\phi_{i,j,k})}{h}\right]^2+\left[\frac{1}{2}\left(\frac{(\phi_{i+1,j,k}-\phi_{i-1,j,k})}{2h}+\frac{(\phi_{i+1,j,k+1}-\phi_{i-1,j,k+1})}{2h}\right)\right]^2+\\ \left[\frac{1}{2}\left(\frac{(\phi_{i,j+1,k}-\phi_{i,j-1,k})}{2h}+\frac{(\phi_{i,j+1,k+1}-\phi_{i,j-1,k+1})}{2h}\right)\right]^2\end{array}}}$$

$$\left(\frac{\phi_x}{|\nabla\phi|}\right)_{i,j,k-1/2} = \frac{(\phi_{i,j,k}-\phi_{i,j,k-1})/h}{\sqrt{\begin{array}{c}\left[\frac{(\phi_{i,j,k-1}-\phi_{i,j,k})}{h}\right]^2 + \left[\frac{1}{2}\left(\frac{(\phi_{i+1,j,k}-\phi_{i-1,j,k})}{2h} + \frac{(\phi_{i+1,j,k-1}-\phi_{i-1,j,k-1})}{2h}\right)\right]^2 + \\ \left[\frac{1}{2}\left(\frac{(\phi_{i,j+1,k}-\phi_{i,j-1,k})}{2h} + \frac{(\phi_{i,j+1,k-1}-\phi_{i,j-1,k-1})}{2h}\right)\right]^2\end{array}}}$$

which allows to solve Equation (3.26) until reaching convergence.

Appendix B: Solid equations characteristics and Riemann solver

The incoming equations complete Chapter 5 and come mostly from [9, 12]. Hence, all the notations used in Chapter 5 will be re-used.

Primitive variable formulation

In Equation (5.1), the system of conservation laws is expressed in terms of conserved variables. However, the linearisation proposed in Equation (5.19) is expressed in primitive variables. Below is detailed the characteristic decomposition in primitive variables.

Introducing the vector of primitive variables $\mathbf{W} = (u, \mathbf{F}^T\mathbf{e}_1, \mathbf{F}^T\mathbf{e}_2, \mathbf{F}^T\mathbf{e}_3, \mathcal{S})$, Equation (5.1) can be rewritten in a quasi-linear form:

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{A}^\alpha \frac{\partial \mathbf{W}}{\partial x_\alpha} = -S^p \tag{6.5}$$

with $\mathbf{A}^\alpha$ the Jacobian matrix defined in Chapter 2 (Equation (2.4), and which can be explicited here as:

$$\mathbf{A}^\alpha = \begin{pmatrix} u_\alpha I & -A^{\alpha 1} & -A^{\alpha 2} & -A^{\alpha 3} & -B^\alpha \\ -\mathbf{F}^T D_{\alpha 1} & u_\alpha I & 0 & 0 & 0 \\ -\mathbf{F}^T D_{\alpha 2} & 0 & u_\alpha I & 0 & 0 \\ -\mathbf{F}^T D_{\alpha 3} & 0 & 0 & u_\alpha I & 0 \\ 0 & 0 & 0 & 0 & u_\alpha \end{pmatrix} \tag{6.6}$$

where $D_{ij} = \mathbf{e}_i \otimes \mathbf{e}_j^T$ represents the unit dyads and the coefficients are given by:

$$A_{ij}^{\alpha\beta} = \frac{1}{\rho}\frac{\partial \sigma_{\alpha i}}{\partial \mathbf{F}_{\beta j}}, \quad B_i^\alpha = \frac{1}{\rho}\frac{\partial \sigma_{\alpha i}}{\partial \mathcal{S}} \tag{6.7}$$

Th vector of source terms is:

$$S^p = \frac{1}{2\mathbf{G}\tau}\begin{pmatrix} 0 \\ (\sigma'\mathbf{F})\mathbf{e}_1 \\ (\sigma'\mathbf{F})\mathbf{e}_2 \\ (\sigma'\mathbf{F})\mathbf{e}_3 \\ -\frac{1}{\rho T}\sum_{i,k=1}^{3}\sigma'_{ik}\sigma_{ik} \end{pmatrix} \tag{6.8}$$

In what follows next, only the axis $\overrightarrow{x_1}$ will be considered, for both the eigenvectors analysis, transformations associated, and solution of the Riemann problem. The complete expressions are provided in [9], and as the changes of coordinates only involves minor changes in the next expressions, therefore they are not being entirely rewritten for the sake of clarity.

It is firstly reminded that the acoustic tensor is defined as:

$$\Omega_{ij} = (e_i^T A_{1j}).(\mathbf{F}^T e_1) = \mathbf{Q}^{-1}\mathcal{D}^2\mathbf{Q}, \quad i,j = 1,2,3 \tag{6.9}$$

where $\mathbf{Q}$ is an orthogonal matrix, and $\mathcal{D} = diag(\sqrt{\lambda_{ac1}}, \sqrt{\lambda_{ac2}}, \sqrt{\lambda_{ac3}})$, $\lambda_{ac1} \geq \lambda_{ac2} \geq \lambda_{ac3}$ is the diagonal matrix of positive eigenvalues (its existence is guaranteed by the fact that System (6.5) is hyperbolic).

The diagonal matrix of eigenvalues is thus given by:

$$\Lambda = diag(u_1\mathbf{I} - \mathcal{D}, u_1\mathbf{I}, u_1\mathbf{I}, u_1, u_1\mathbf{I} + \pi D\pi) \tag{6.10}$$

where $\mathbf{I}$ is the identity 3x3 matrix, and π the permutation matrix:

$$\pi = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{6.11}$$

We provide here the set of left and right eigenvectors for the Jacobian $\mathbf{A}^\alpha$ (as here $\alpha = 1$, that index is dropped for clarity).

The matrix of left eigenvectors is:

$$\mathbf{L} = (l_1, l_2, l_3, l_4, l_5)^T \tag{6.12}$$

with:

$$
\begin{aligned}
l_1 &= (\mathcal{D}\mathbf{Q}, \mathbf{Q}A_{11}, \mathbf{Q}A_{11}, \mathbf{Q}A_{11}, \mathbf{Q}A_{11}, B_1)\\
l_2 &= (0, \frac{1}{\mathbf{F}_{11}}(\mathbf{F}_{12}D_{11} + \mathbf{F}_{13}D_{21}) - D_{12} - D_{23}, \frac{1}{\mathbf{F}_{11}}(\mathbf{F}_{12}D_{31}) - D_{32}, 0, 0)\\
l_3 &= (0, 0, \frac{1}{\mathbf{F}_{11}}(\mathbf{F}_{13}D_{11}) - D_{13}, \frac{1}{\mathbf{F}_{11}}(\mathbf{F}_{12}D_{21} + \mathbf{F}_{13}D_{31}) - D_{22} - D_{33}, 0)\\
l_4 &= (0, 0, 0, \frac{1}{\mathbf{F}_{11}}(\mathbf{F}_{12}D_{21} + \mathbf{F}_{13}D_{31}) - D_{22} - D_{33}, 0)\\
l_5 &= (0, 0, 0, 0, 1) \qquad (6.13)
\end{aligned}
$$

Assuming that the right eigenvectors are orthonormal to the left, i.e. $\mathbf{RL} = \mathbf{I}$, the matrix of right eigenvectors is given by:

$$\mathbf{R} = (r_1, r_2, r_3, r_4, r_5) \qquad (6.14)$$

with:

$$\begin{aligned}
r_1 = (&\frac{1}{2}\mathbf{Q}^{-1}\mathcal{D}^{-1}, \frac{1}{2}(\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_1^T\mathbf{Q}^{-1}\mathbf{D}^{-2}),\\
&\frac{1}{2}(\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_2^T\mathbf{Q}^{-1}\mathcal{D}^{-2}), \frac{1}{2}(\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_3^T\mathbf{Q}^{-1}\mathcal{D}^{-2}), 0)^T\\
\\
r_2 = (&0, (\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_1^T\Psi_1) - D_{21} - D_{32}, (\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_2^T\Psi_1) - D_{23},\\
&(\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_3^T\Psi_1), 0)^T\\
\\
r_3 = (&0, (\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_1^T\Psi_2), (\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_2^T\Psi_2) - D_{31},\\
&(\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_{31}^T\Psi_2) - D_{22} - D_{33}, 0)^T,\\
\\
r_4 = (&0, (\mathbf{e}_1^T\Omega^{-1}B_1)\mathbf{F}^T\mathbf{e}_1, (\mathbf{e}_2^T\Omega^{-1}B_1)\mathbf{F}^T\mathbf{e}_1, (\mathbf{e}_3^T\Omega^{-1}B_1)\mathbf{F}^T\mathbf{e}_1)^T\\
\\
r_5 = (&\frac{1}{2}\mathbf{Q}^{-1}\mathcal{D}^{-1}\pi, -\frac{1}{2}((\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_1^T\mathbf{Q}^{-1}\mathcal{D}^{-2}))\pi,\\
&-\frac{1}{2}((\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_2^T\mathbf{Q}^{-1}\mathcal{D}^{-2}))\pi, -\frac{1}{2}((\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_3^T\mathbf{Q}^{-1}\mathcal{D}^{-2}))\pi, 0)^T
\end{aligned}\tag{6.15}$$

and:

$$\begin{aligned}
\Psi_1 &= \Omega^{-1}(A_{11}D_{21} + A_{11}D_{32} + A_{12}D_{23})\\
\Psi_2 &= \Omega^{-1}(A_{12}D_{31} + A_{13}D_{22} + A_{13}D_{33})
\end{aligned}\tag{6.16}$$

Transformation to primitive variables to conserved variables

To maintain high-order accuracy in the numerical methods, it is necessary to revert to conserved variables (noted **U**) to express the invariants (see Chapter 2). Partial derivatives of primitive variables can be expressed in term of partial derivatives of conserved variables according to the matrix **H**, defied as:

$$\mathbf{H} = \frac{\partial \mathbf{W}}{\partial \mathbf{U}} = \frac{1}{2\rho}\begin{pmatrix} -2\mathbf{I} & u\otimes(\mathbf{e}_1^T\mathbf{F}^{-T}) & u\otimes(\mathbf{e}_2^T\mathbf{F}^{-T}) & u\otimes(\mathbf{e}_3^T\mathbf{F}^{-T}) & 0 \\ 0 & (\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_1^T\mathbf{F}^{-T})-2\mathbf{I} & (\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_2^T\mathbf{F}^{-T}) & (\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_3^T\mathbf{F}^{-T}) & 0 \\ 0 & (\mathbf{F}^T\mathbf{e}_2)\otimes(\mathbf{e}_1^T\mathbf{F}^{-T}) & (\mathbf{F}^T\mathbf{e}_2)\otimes(\mathbf{e}_2^T\mathbf{F}^{-T})-2\mathbf{I} & (\mathbf{F}^T\mathbf{e}_2)\otimes(\mathbf{e}_3^T\mathbf{F}^{-T}) & 0 \\ 0 & (\mathbf{F}^T\mathbf{e}_3)\otimes(\mathbf{e}_1^T\mathbf{F}^{-T}) & (\mathbf{F}^T\mathbf{e}_3)\otimes(\mathbf{e}_2^T\mathbf{F}^{-T}) & (\mathbf{F}^T\mathbf{e}_3)\otimes(\mathbf{e}_3^T\mathbf{F}^{-T})-2\mathbf{I} & 0 \\ 2\frac{d\mathcal{S}}{d\xi}u^T & -2\frac{d\mathcal{S}}{d\mathbf{e}_1^T\mathbf{F}}-\mathbf{e}_1^T\mathbf{F}^{-T}\Psi_3 & -2\frac{d\mathcal{S}}{d\mathbf{e}_2^T\mathbf{F}}-\mathbf{e}_2^T\mathbf{F}^{-T}\Psi_3 & -2\frac{d\mathcal{S}}{d\mathbf{e}_3^T\mathbf{F}}-\mathbf{e}_3^T\mathbf{F}^{-T}\Psi_3 & 2\frac{d\mathcal{S}}{d\xi} \end{pmatrix} \tag{6.17}$$

with:

$$\Psi_3 = \frac{d\mathcal{S}}{d\xi}\left(\frac{1}{2}|\mathbf{u}|^2 - \xi\right) - \sum_{i,j=1}^{3}\frac{d\mathcal{S}}{d\mathbf{F}_{ij}}\mathbf{F}_{ij} \tag{6.18}$$

Next, the matrices of eigenvectors, **L** and **R** explicited in Equations (6.13) and (6.15) respectively, of the primitive Jacobian A, can be transformed to those of the conservative Jacobian $A^c = H^{-1}AH$ according to (it is recalled that $E = (|u|^2 + \xi)$, the total energy):

$$\mathbf{R}^c = H^{-1}\mathbf{R}, \quad \mathbf{L}^c = \mathbf{L}H \tag{6.19}$$

The inverse of **H** is given by:

$$\mathbf{H}^{-1} = \frac{\partial \mathbf{U}}{\partial \mathbf{W}} = -\rho\begin{pmatrix} -\mathbf{I} & u\otimes(\mathbf{e}_1^T\mathbf{F}^{-T}) & u\otimes(\mathbf{e}_2^T\mathbf{F}^{-T}) & u\otimes(\mathbf{e}_3^T\mathbf{F}^{-T}) & 0 \\ 0 & (\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_1^T\mathbf{F}^{-T})-\mathbf{I} & (\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_2^T\mathbf{F}^{-T}) & (\mathbf{F}^T\mathbf{e}_1)\otimes(\mathbf{e}_3^T\mathbf{F}^{-T}) & 0 \\ 0 & (\mathbf{F}^T\mathbf{e}_2)\otimes(\mathbf{e}_1^T\mathbf{F}^{-T}) & (\mathbf{F}^T\mathbf{e}_2)\otimes(\mathbf{e}_2^T\mathbf{F}^{-T})-\mathbf{I} & (\mathbf{F}^T\mathbf{e}_2)\otimes(\mathbf{e}_3^T\mathbf{F}^{-T}) & 0 \\ 0 & (\mathbf{F}^T\mathbf{e}_3)\otimes(\mathbf{e}_1^T\mathbf{F}^{-T}) & (\mathbf{F}^T\mathbf{e}_3)\otimes(\mathbf{e}_2^T\mathbf{F}^{-T}) & (\mathbf{F}^T\mathbf{e}_3)\otimes(\mathbf{e}_3^T\mathbf{F}^{-T})-\mathbf{I} & 0 \\ -u^T & \frac{d\xi}{d\mathbf{e}_1^T\mathbf{F}}-\mathbf{e}_1^T\mathbf{F}^{-T}E & \frac{d\xi}{d\mathbf{e}_2^T\mathbf{F}}-\mathbf{e}_2^T\mathbf{F}^{-T}E & \frac{d\xi}{d\mathbf{e}_3^T\mathbf{F}}-\mathbf{e}_3^T\mathbf{F}^{-T}E & -\frac{d\xi}{d\mathcal{S}} \end{pmatrix} \tag{6.20}$$

The Riemann solver and characteristic variables

Provided the above, i.e. the vectors of variables, and the associated eigenvectors, it is easy to determine the vector of characteristic variables (see Section 2.1.1). The derivation leads to the following formulation for the characteristic variable vector $\mathcal{C} = (\mathcal{C}_1, ..., \mathcal{C}_{13})$, and its evaluation for the Riemann solver:

$$
\begin{aligned}
\mathcal{C}_1 &= \mathbf{Q}_{11}\Sigma_1^{(1)} + \mathbf{Q}_{12}\Sigma_2^{(1)} + \mathbf{Q}_{13}\Sigma_3^{(1)} + \mathcal{D}_{11}(\mathbf{Q}_{11}\overline{\mathbf{F}}_{11}^{(1)} + \mathbf{Q}_{12}\overline{\mathbf{F}}_{12}^{(1)} + \mathbf{Q}_{13}\overline{\mathbf{F}}_{13}^{(1)})\\
\mathcal{C}_2 &= \mathbf{Q}_{21}\Sigma_1^{(2)} + \mathbf{Q}_{22}\Sigma_2^{(2)} + \mathbf{Q}_{23}\Sigma_3^{(2)} + \mathcal{D}_{22}(\mathbf{Q}_{21}\overline{\mathbf{F}}_{11}^{(2)} + \mathbf{Q}_{22}\overline{\mathbf{F}}_{12}^{(2)} + \mathbf{Q}_{23}\overline{\mathbf{F}}_{13}^{(2)})\\
\mathcal{C}_3 &= \mathbf{Q}_{31}\Sigma_1^{(3)} + \mathbf{Q}_{32}\Sigma_2^{(3)} + \mathbf{Q}_{33}\Sigma_3^{(3)} + \mathcal{D}_{33}(\mathbf{Q}_{31}\overline{\mathbf{F}}_{11}^{(3)} + \mathbf{Q}_{32}\overline{\mathbf{F}}_{12}^{(3)} + \mathbf{Q}_{33}\overline{\mathbf{F}}_{13}^{(3)})\\
\mathcal{C}_4 &= -\mathbf{Q}_{11}\Sigma_1^{(4)} + \mathbf{Q}_{12}\Sigma_2^{(4)} + \mathbf{Q}_{13}\Sigma_3^{(4)} + \mathcal{D}_{11}(\mathbf{Q}_{11}\overline{\mathbf{F}}_{11}^{(4)} + \mathbf{Q}_{12}\overline{\mathbf{F}}_{12}^{(4)} + \mathbf{Q}_{13}\overline{\mathbf{F}}_{13}^{(4)})\\
\mathcal{C}_5 &= -\mathbf{Q}_{21}\Sigma_1^{(5)} + \mathbf{Q}_{22}\Sigma_2^{(5)} + \mathbf{Q}_{23}\Sigma_3^{(5)} + \mathcal{D}_{22}(\mathbf{Q}_{21}\overline{\mathbf{F}}_{11}^{(5)} + \mathbf{Q}_{22}\overline{\mathbf{F}}_{12}^{(5)} + \mathbf{Q}_{23}\overline{\mathbf{F}}_{13}^{(5)})\\
\mathcal{C}_6 &= -\mathbf{Q}_{31}\Sigma_1^{(6)} + \mathbf{Q}_{32}\Sigma_2^{(6)} + \mathbf{Q}_{33}\Sigma_3^{(6)} + \mathcal{D}_{33}(\mathbf{Q}_{31}\overline{\mathbf{F}}_{11}^{(6)} + \mathbf{Q}_{32}\overline{\mathbf{F}}_{12}^{(6)} + \mathbf{Q}_{33}\overline{\mathbf{F}}_{13}^{(6)})\\
\mathcal{C}_7 &= \frac{\mathbf{F}_{12}}{\mathbf{F}_{11}}\overline{\mathbf{F}}_{12}^{(0)} - \overline{\mathbf{F}}_{12}^{(0)}\\
\mathcal{C}_8 &= \frac{\mathbf{F}_{13}}{\mathbf{F}_{11}}\overline{\mathbf{F}}_{11}^{(0)} - \overline{\mathbf{F}}_{13}^{(0)}\\
\mathcal{C}_9 &= \frac{\mathbf{F}_{12}}{\mathbf{F}_{11}}\overline{\mathbf{F}}_{21}^{(0)} - \overline{\mathbf{F}}_{22}^{(0)}\\
\mathcal{C}_{10} &= \frac{\mathbf{F}_{123}}{\mathbf{F}_{11}}\overline{\mathbf{F}}_{21}^{(0)} - \overline{\mathbf{F}}_{23}^{(0)}\\
\mathcal{C}_{11} &= \frac{\mathbf{F}_{12}}{\mathbf{F}_{11}}\overline{\mathbf{F}}_{31}^{(0)} - \overline{\mathbf{F}}_{32}^{(0)}\\
\mathcal{C}_{12} &= \frac{\mathbf{F}_{13}}{\mathbf{F}_{11}}\overline{\mathbf{F}}_{31}^{(0)} - \overline{\mathbf{F}}_{33}^{(0)}\\
\mathcal{C}_{13} &= \overline{S}^{(0)}
\end{aligned}
\tag{6.21}
$$

where:

$$\begin{aligned}
\Sigma_i^{(.)} &= \sum_{j,k=1}^{3} A_{ik}^{1j}\overline{\mathbf{F}}_{jk}^{(.)} + B_i\mathcal{S}^{(.)} \\
\overline{u_i}^{(.)} &= (\rho u_i)^{(.)} - u_i\hat{\rho} \\
\overline{\mathbf{F}}_{ij}^{(.)} &= (\rho\mathbf{F})_{ij}^{(.)} - \mathbf{F}_{ij}\hat{\rho} \\
\overline{S}^{(.)} &= \frac{d\mathcal{S}}{d\xi}[(\rho E)^{(.)} - (u_1(\rho u_1)^{(.)} + u_2(\rho u_2)^{(.)} + u_3(\rho u_3)^{(.)}) + |\mathbf{u}|/2 - \xi] + \\
&\qquad \sum_{i,j=1}^{3}\frac{d\mathcal{S}}{d\mathbf{F}_{ij}}((\rho\mathbf{F})_{ij}^{(.)} - \mathbf{F}_{ij}\hat{\rho}) \\
\hat{\rho} &= \frac{1}{2}\sum_{i,j=1}^{3}\mathbf{F}_{ij}^{-T}(\rho\mathbf{F})_{ij}^{(.)} \qquad (6.22)
\end{aligned}$$

The numbers in parenthesis indicate the waves along which the associated quantity is evaluated. The eigenvalues are enumerated as follows:

$$\begin{aligned}
&\lambda^{(1)} = u_1 - D_{11}, \quad \lambda^{(2)} = u_1 - D_{22}, \quad \lambda^{(3)} = u_1 - D_{33}, \\
&\lambda^{(4)} = u_1 + D_{11}, \quad \lambda^{(5)} = u_1 + D_{22}, \quad \lambda^{(6)} = u_1 + D_{33}, \\
&\lambda^{(0)} = u_1 \qquad (6.23)
\end{aligned}$$

For each evaluation, the choice of whether to equate the quantities $\mathcal{C}^{(.)}$ to either left or right values can be achieved easily using:

$$\mathcal{C}^{(i)} = \left(\frac{1}{2} + \psi_i\right)\mathcal{C}^L + \left(\frac{1}{2} - \psi_i\right)\mathcal{C}^R \qquad (6.24)$$

with:

$$\psi_i = \frac{1}{2}\frac{\lambda^{(i)}}{|\lambda^{(i)}| + \epsilon} \qquad (6.25)$$

where $\mathcal{C}^L$ and $\mathcal{C}^R$ represent the left and right states adjacent to the boundary found using a reconstruction method, and ϵ a small number to prevent division by zero.

The Riemann problem solution can eventually be written explicitly as follows:

$$\begin{pmatrix} \widetilde{\rho \mathbf{u}} \\ \widetilde{\rho \mathbf{F}}^T \mathbf{e}_1 \\ \widetilde{\rho \mathbf{F}}^T \mathbf{e}_2 \\ \widetilde{\rho \mathbf{F}}^T \mathbf{e}_3 \\ \widetilde{\rho E} \end{pmatrix} = \mathcal{R}^1 \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{F}^T \mathbf{e}_1 \\ \rho \mathbf{F}^T \mathbf{e}_2 \\ \rho \mathbf{F}^T \mathbf{e}_3 \\ \rho E \end{pmatrix} + \begin{pmatrix} \mathbf{u}.\mathcal{R}^2 \\ \mathbf{f}^T(\mathcal{R}^3 + \mathcal{R}^4)\mathbf{e}_1 - \mathcal{C}_7 \mathbf{e}_2 - \mathcal{C}_8 \mathbf{e}_3 \\ \mathbf{f}^T(\mathcal{R}^3 + \mathcal{R}^4)\mathbf{e}_2 - \mathcal{C}_9 \mathbf{e}_2 - \mathcal{C}_{10} \mathbf{e}_3 \\ \mathbf{f}^T(\mathcal{R}^3 + \mathcal{R}^4)\mathbf{e}_3 - \mathcal{C}_{11} \mathbf{e}_2 - \mathcal{C}_{12} \mathbf{e}_3 \\ \mathcal{R}^5 \end{pmatrix} \tag{6.26}$$

where the auxiliary variables $\mathcal{R}^1, \mathcal{R}^2, \mathcal{R}^3, \mathcal{R}^4, \mathcal{R}^5$ are defined as:

$$\begin{aligned}
\mathcal{R}^1 &= -(\mathcal{R}^3 + \mathcal{R}^4)\mathbf{e}_1 + \\
&\quad \mathbf{F}_{12}^{-T}\mathcal{C}_7 + \mathbf{F}_{13}^{-T}\mathcal{C}_8 + \mathbf{F}_{22}^{-T}\mathcal{C}_9 + \mathbf{F}_{23}^{-T}\mathcal{C}_{10} + \mathbf{F}_{32}^{-T}\mathcal{C}_{11} + \mathbf{F}_{33}^{-T}\mathcal{C}_{12} \\
\mathcal{R}^2_i &= \frac{1}{2}[(\mathcal{D}\mathbf{Q})^{-1}_{i1}(\mathcal{C}_1 + \mathcal{C}_4) + (\mathcal{D}\mathbf{Q})^{-1}_{i2}(\mathcal{C}_2 + \mathcal{C}_5) + (\mathcal{D}\mathbf{Q})^{-1}_{i3}(\mathcal{C}_3 + \mathcal{C}_6)] \\
\mathcal{R}^3_i &= \frac{1}{2}[(\mathcal{D}^2\mathbf{Q})^{-1}_{i1}(\mathcal{C}_1 - \mathcal{C}_4) + (\mathcal{D}^2\mathbf{Q})^{-1}_{i2}(\mathcal{C}_2 - \mathcal{C}_5) + (\mathcal{D}^2\mathbf{Q})^{-1}_{i3}(\mathcal{C}_3 - \mathcal{C}_6)] \\
\mathcal{R}^4_i &= \Upsilon^1_{i1}\mathcal{C}_7 + \Upsilon^2_{i1}\mathcal{C}_8 + \Upsilon^1_{i12}\mathcal{C}_9 + \Upsilon^2_{i2}\mathcal{C}_{10} + \Upsilon^1_{i3}\mathcal{C}_{11} + \Upsilon^2_{i3}\mathcal{C}_{12} - \Upsilon^1_{i}\mathcal{C}_{13} \\
\mathcal{R}^5 &= \mathbf{u}.\mathcal{R}^2 - T\left[\left(\frac{dF}{d\mathbf{F}}\mathbf{f}\right).(\mathcal{R}^2 + \mathcal{R}^3) - \right. \\
&\quad \left. \frac{d\mathcal{S}}{d\mathbf{F}_{12}}\mathcal{C}_7 - \frac{d\mathcal{S}}{d\mathbf{F}_{13}}\mathcal{C}_8 - \frac{d\mathcal{S}}{d\mathbf{F}_{22}}\mathcal{C}_9 - \frac{d\mathcal{S}}{d\mathbf{F}_{23}}\mathcal{C}_{10} - \frac{d\mathcal{S}}{d\mathbf{F}_{32}}\mathcal{C}_{11} - \frac{d\mathcal{S}}{d\mathbf{F}_{33}}\mathcal{C}_{12} - \mathcal{C}_{13}\right]
\end{aligned} \tag{6.27}$$

with:

$$\begin{aligned}
\mathbf{f} &= \mathbf{e}_1^T\mathbf{F} \\
\Upsilon^1_{ij} &= \Omega^{-1}_{i1}A^{1j}_{12} + \Omega^{-1}_{i2}A^{1j}_{22} + \Omega^{-1}_{i3}A^{1j}_{32} \\
\Upsilon^2_{ij} &= \Omega^{-1}_{i1}A^{1j}_{13} + \Omega^{-1}_{i2}A^{1j}_{23} + \Omega^{-1}_{i3}A^{1j}_{33} \\
\Upsilon^3_{i} &= \Omega^{-1}_{i1}B_1 + \Omega^{-1}_{i2}B_2 + \Omega^{-1}_{i3}B_3
\end{aligned} \tag{6.28}$$

Appendix C: Complete 1D results for fluid/fluid interaction.

IVP 1: Air-Helium shock tube

Below are presented all the graphs relative to different grid size and reconstruction methods used, i.e. 1st Order, WENO-3, and MPWENO-5.

1st Order

1. 250 points:

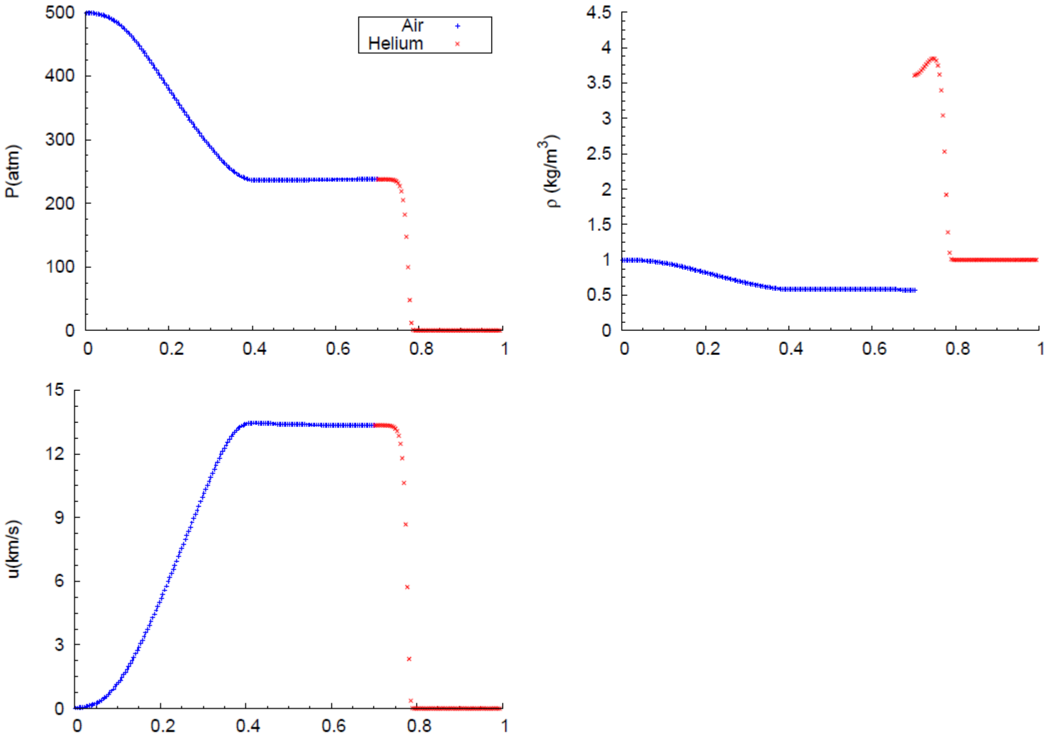


Figure 6.1: Air Helium IVP, 1st-Order and 250 points grid size.

2. 500 points

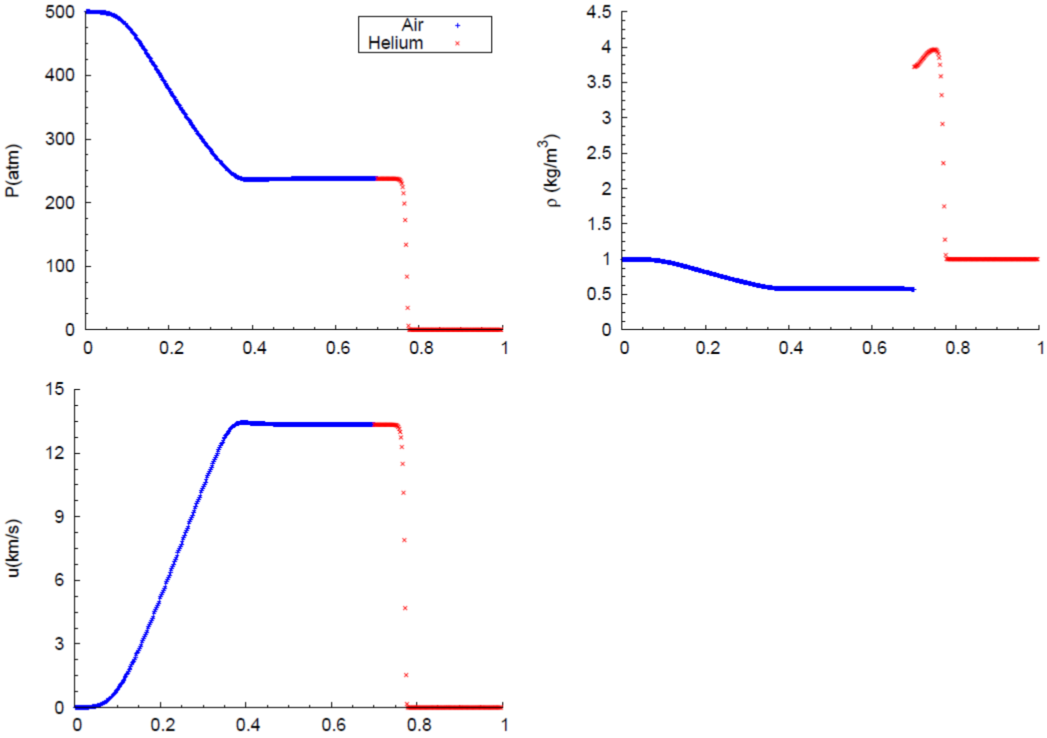


Figure 6.2: Air Helium IVP, 1st-Order and 500 points grid size.

3. 1000 points

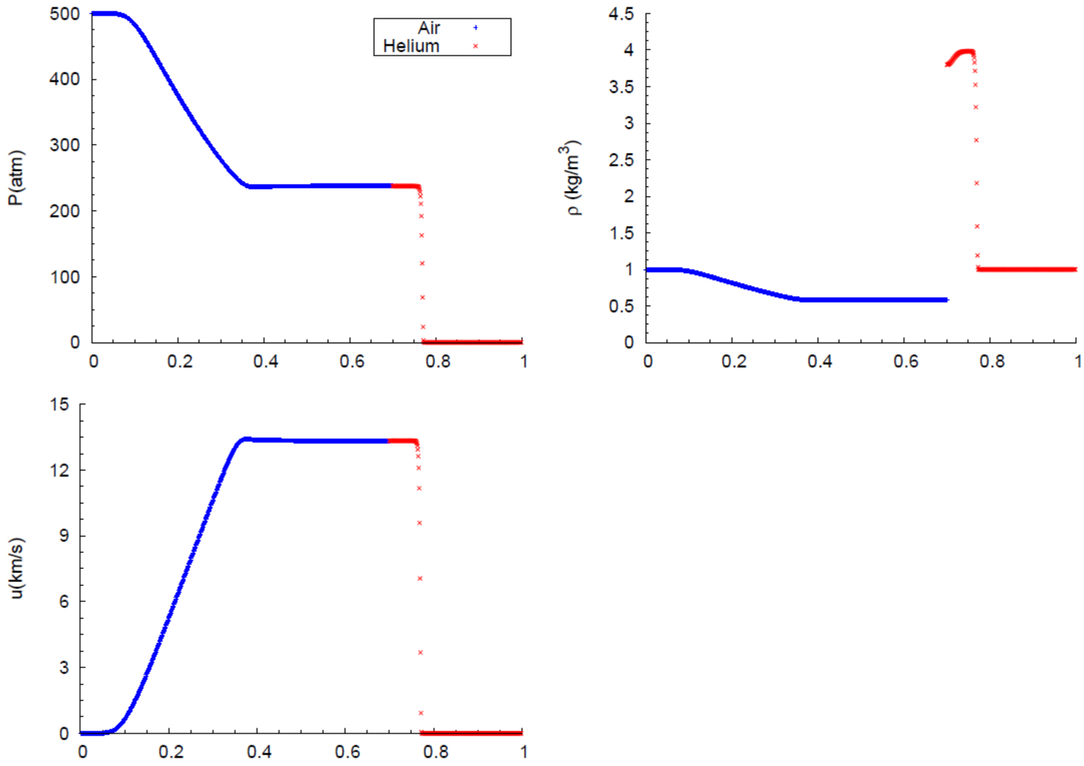


Figure 6.3: Air Helium IVP, 1st-Order and 1000 points grid size.

WENO-3

1. 250 points

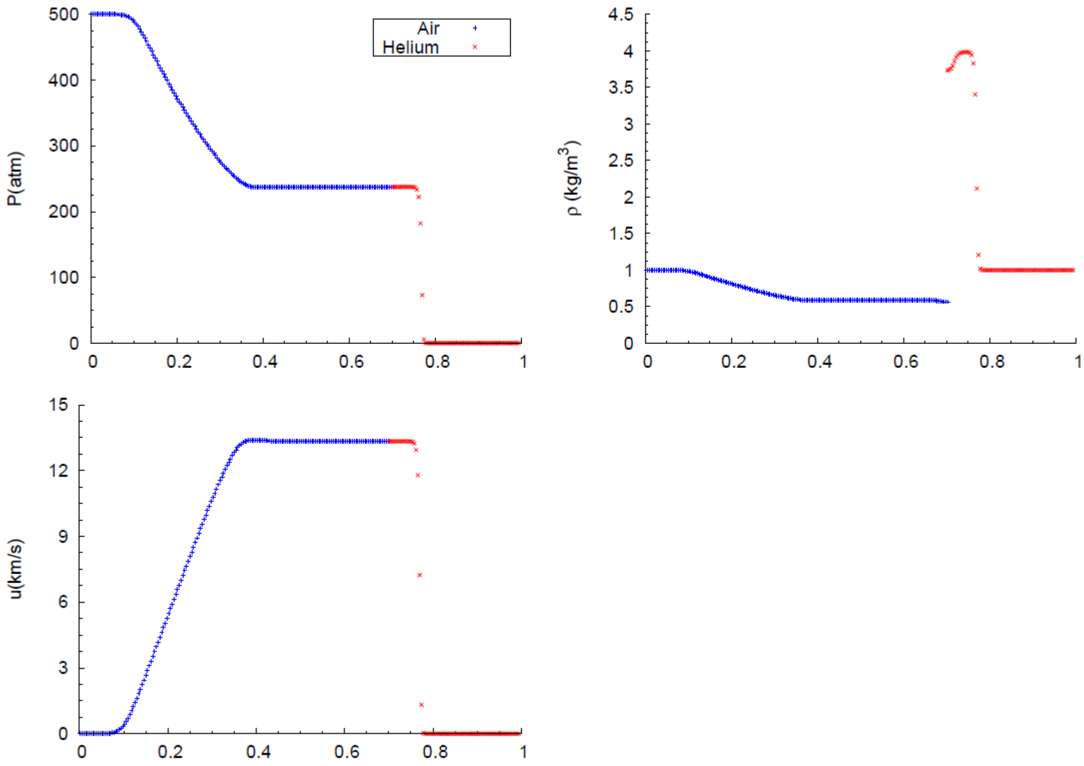


Figure 6.4: Air Helium IVP, WENO-3 and 250 points grid size.

2. 500 points

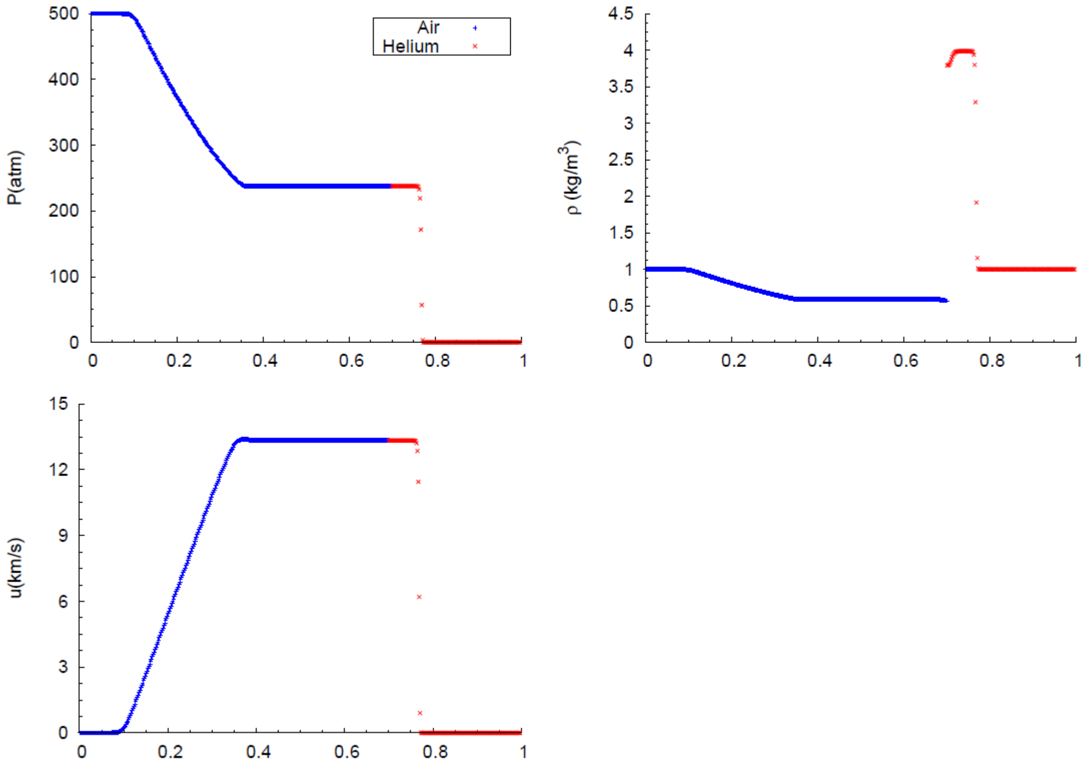


Figure 6.5: Air Helium IVP, WENO-3 and 500 points grid size.

3. 1000 points

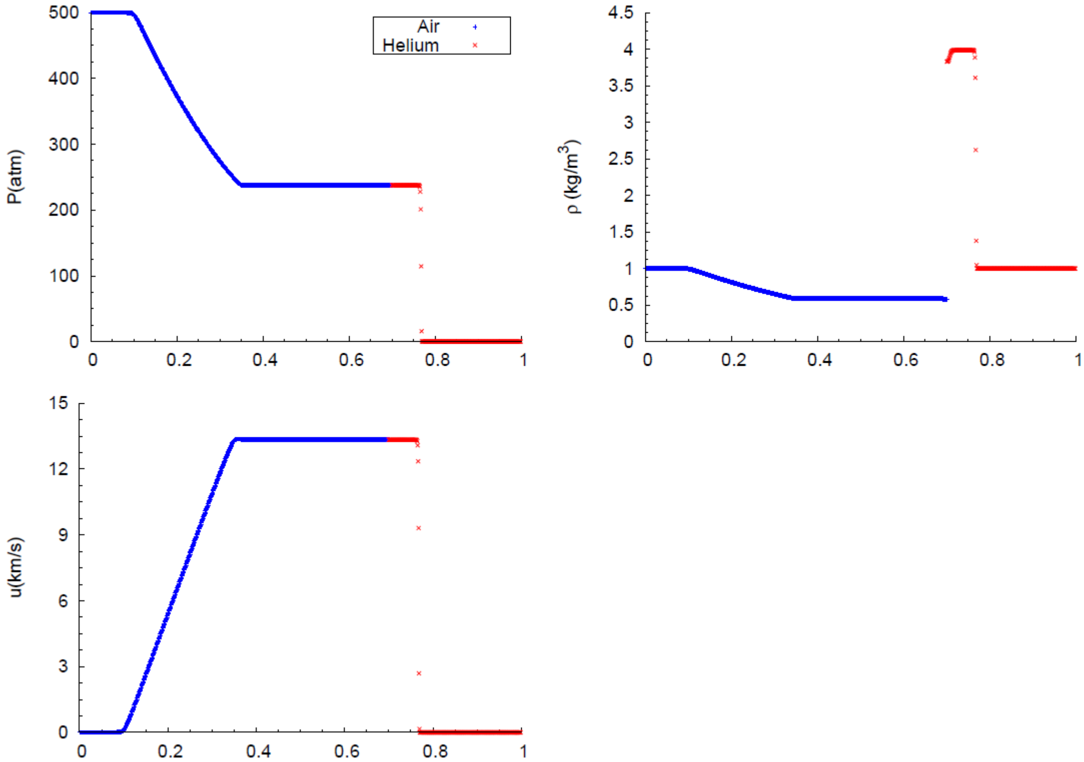


Figure 6.6: Air Helium IVP, WENO-3 and 1000 points grid size.

MPWENO-5

1. 250 points

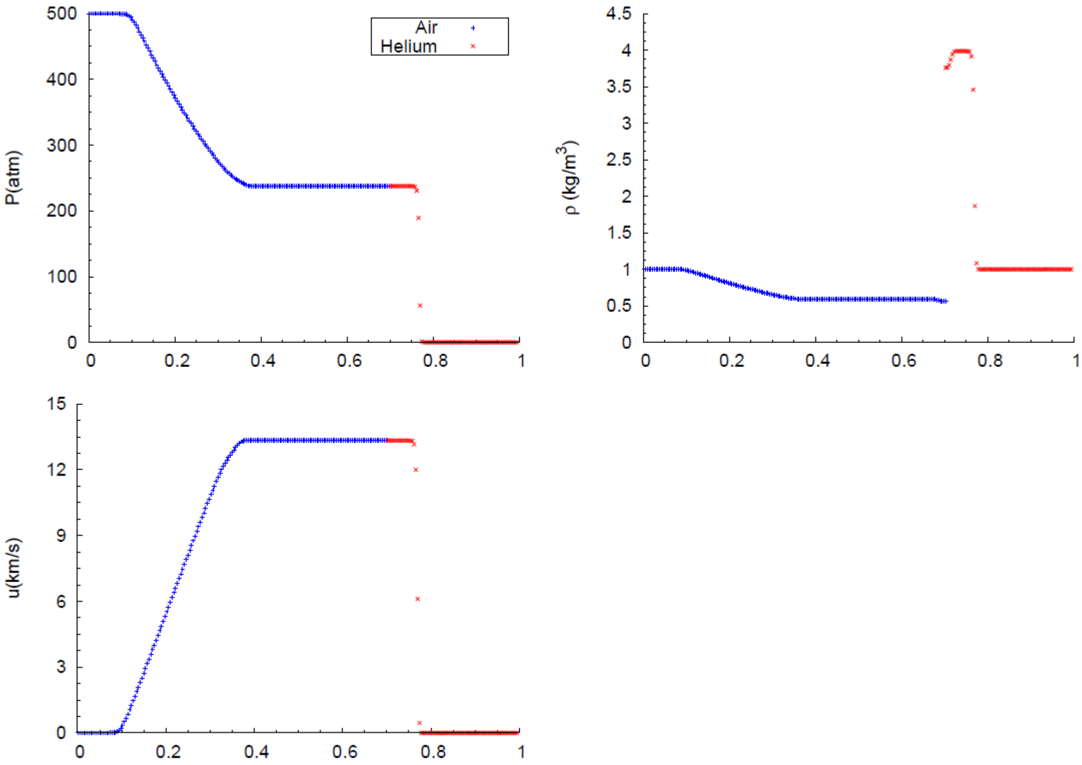


Figure 6.7: Air Helium IVP, MPWENO-5 and 250 points grid size.

2. 500 points

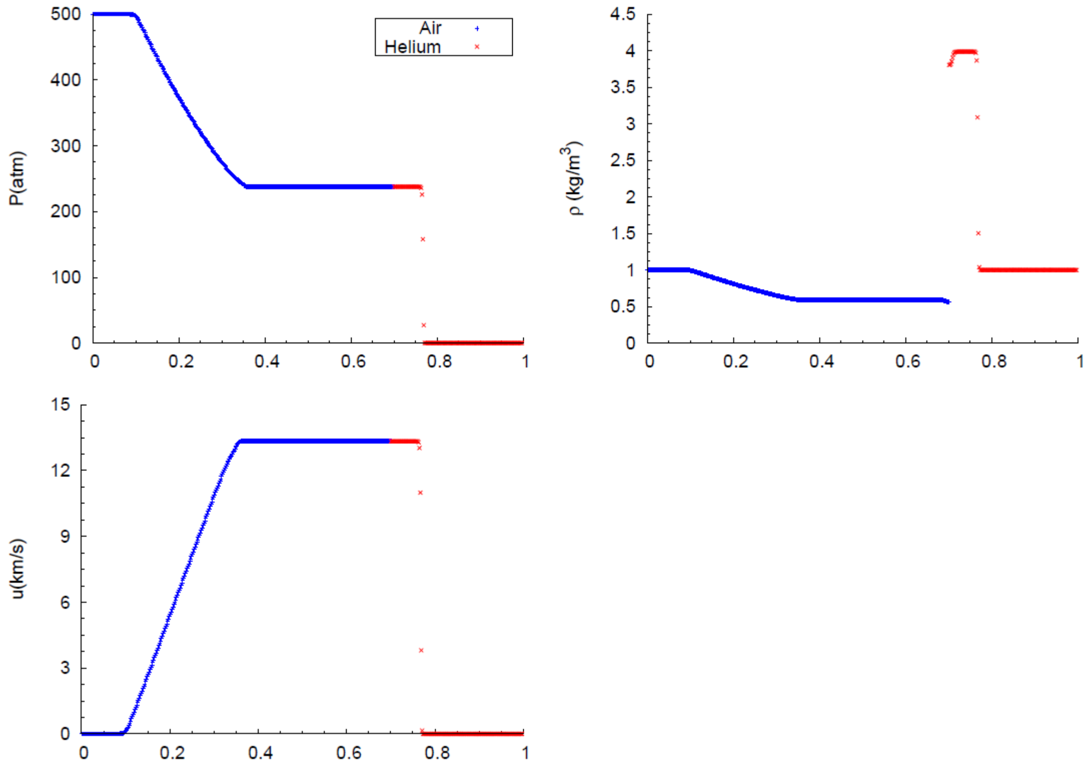


Figure 6.8: Air Helium IVP, MPWENO-5 and 500 points grid size.

3. 1000 points

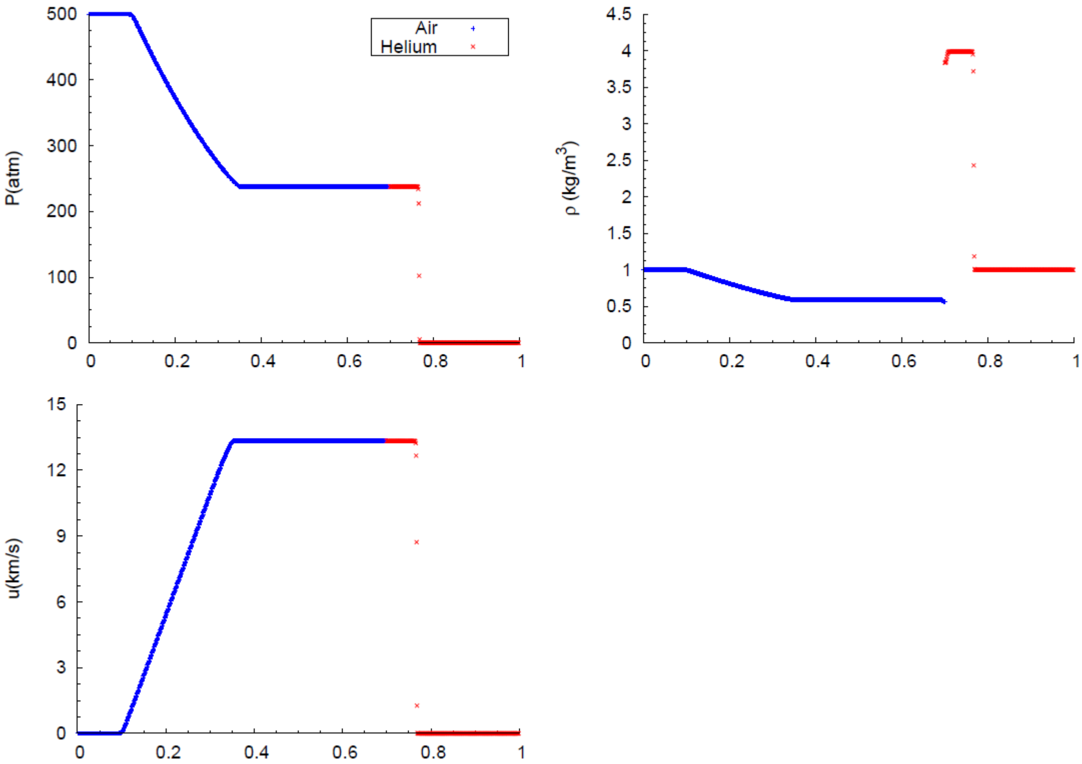


Figure 6.9: Air Helium IVP, MPWENO-5 and 1000 points grid size.

IVP 2: Air-Water shock tube

Below are presented all the graphs relative to different grid size and reconstruction methods used, i.e. 1st Order, WENO-3, and MPWENO-5.

1st Order

1. 250 points:

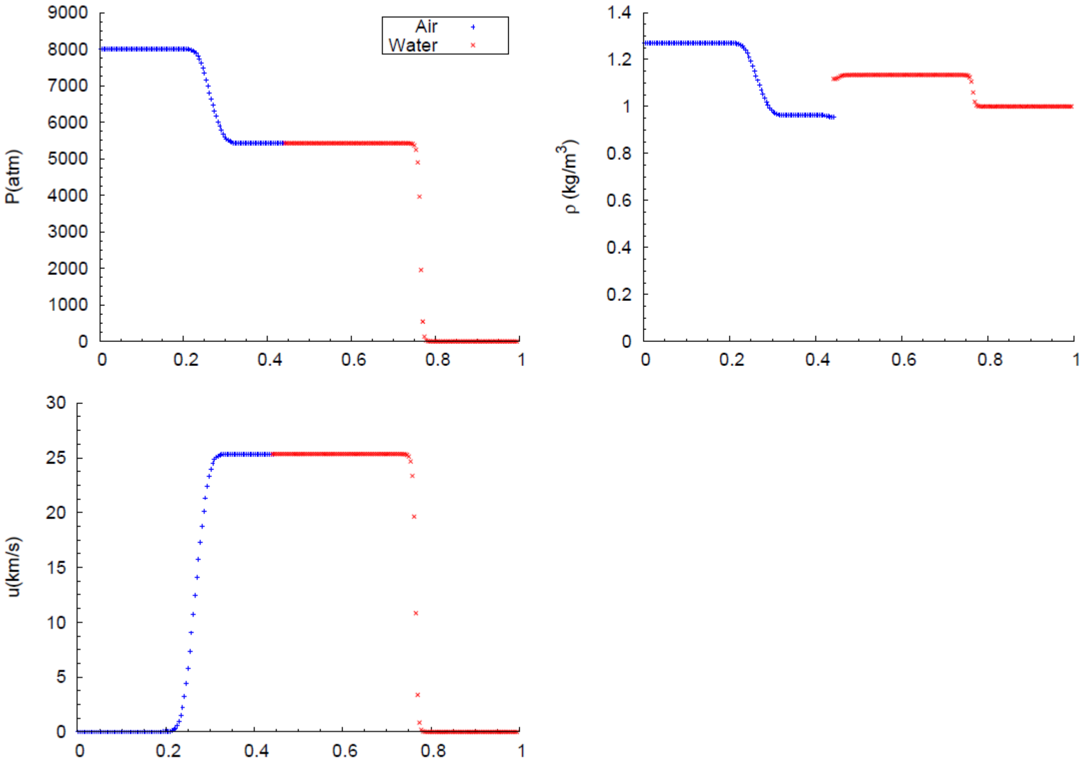


Figure 6.10: Air Water IVP, 1st-Order and 250 points grid size.

2. 500 points

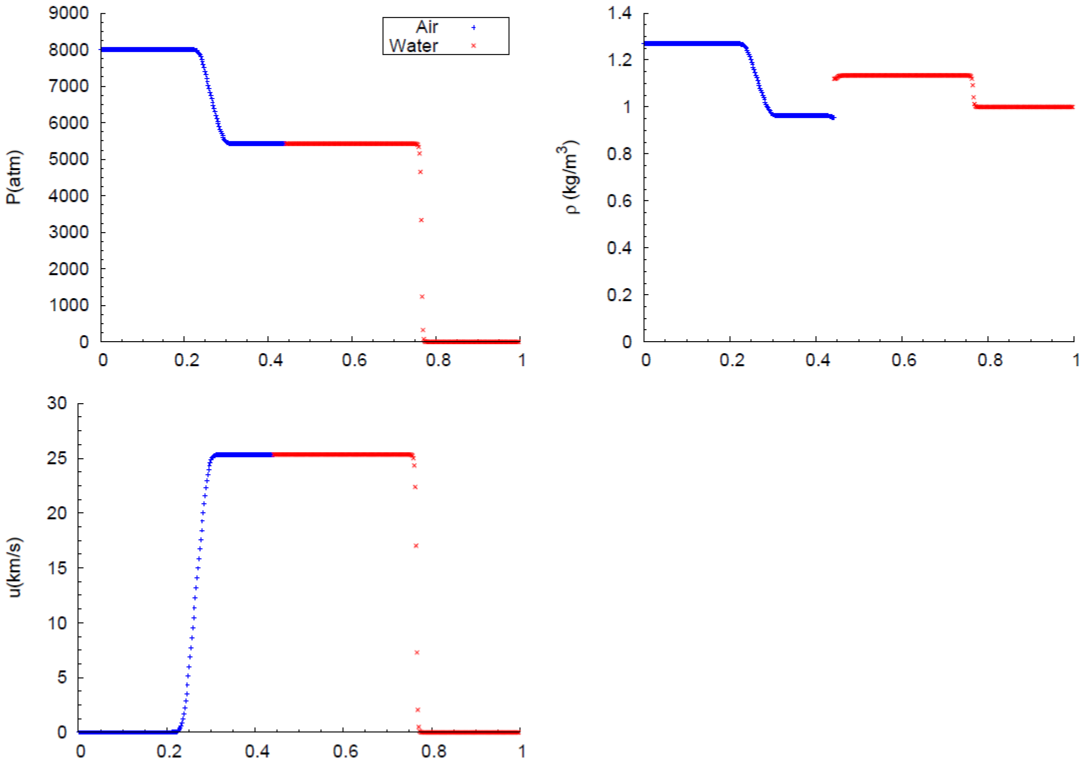


Figure 6.11: Air Water IVP, 1st-Order and 500 points grid size.

3. 1000 points

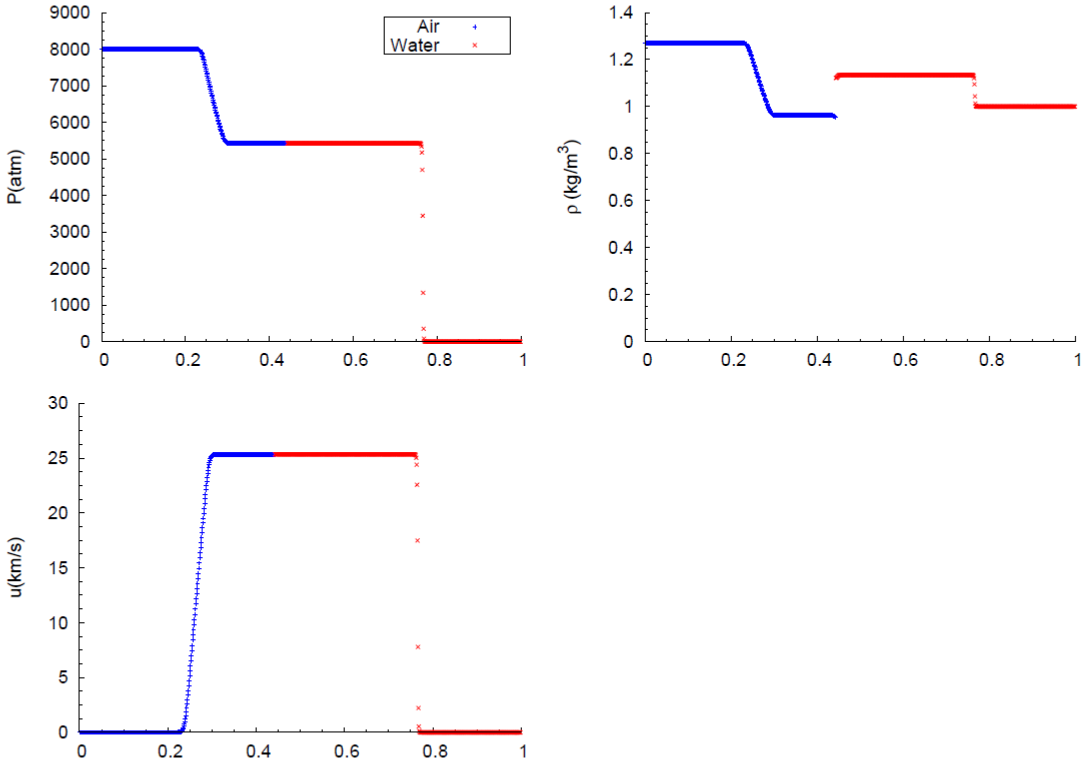


Figure 6.12: Air Water IVP, 1st-Order and 1000 points grid size.

WENO-3

1. 250 points

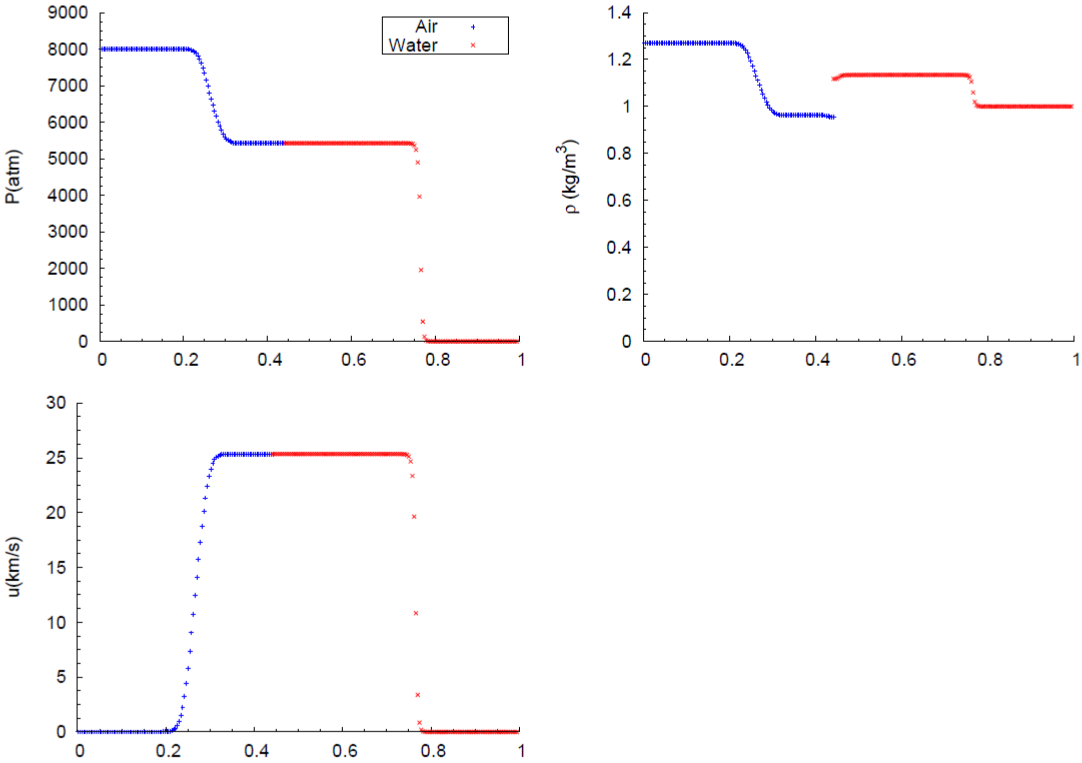


Figure 6.13: Air Water IVP, WENO-3 and 250 points grid size.

2. 500 points

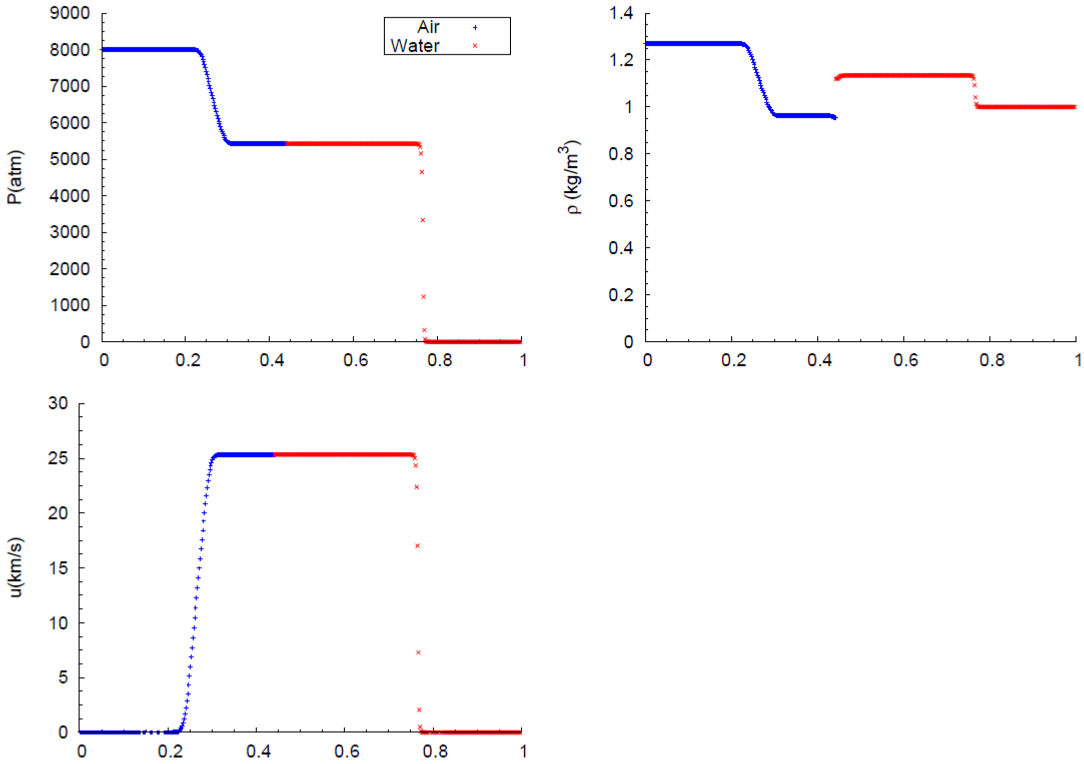


Figure 6.14: Air Water IVP, WENO-3 and 500 points grid size.

3. 1000 points

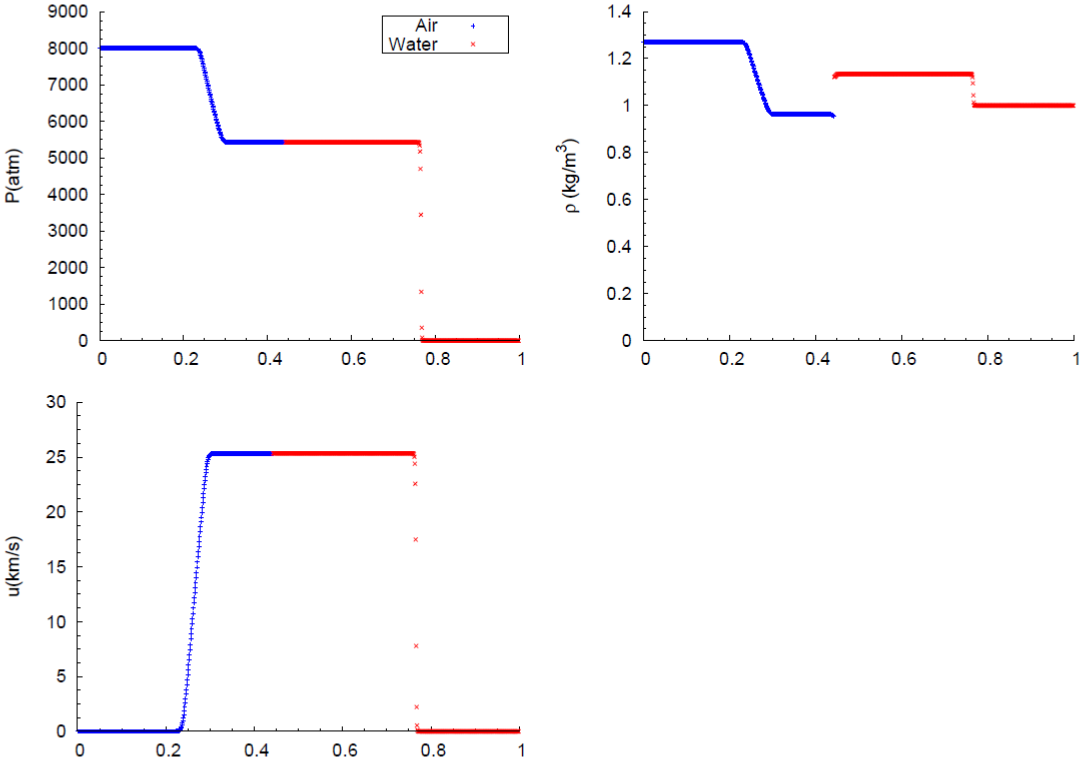


Figure 6.15: Air Water IVP, WENO-3 and 1000 points grid size.

MPWENO-5

1. 250 points

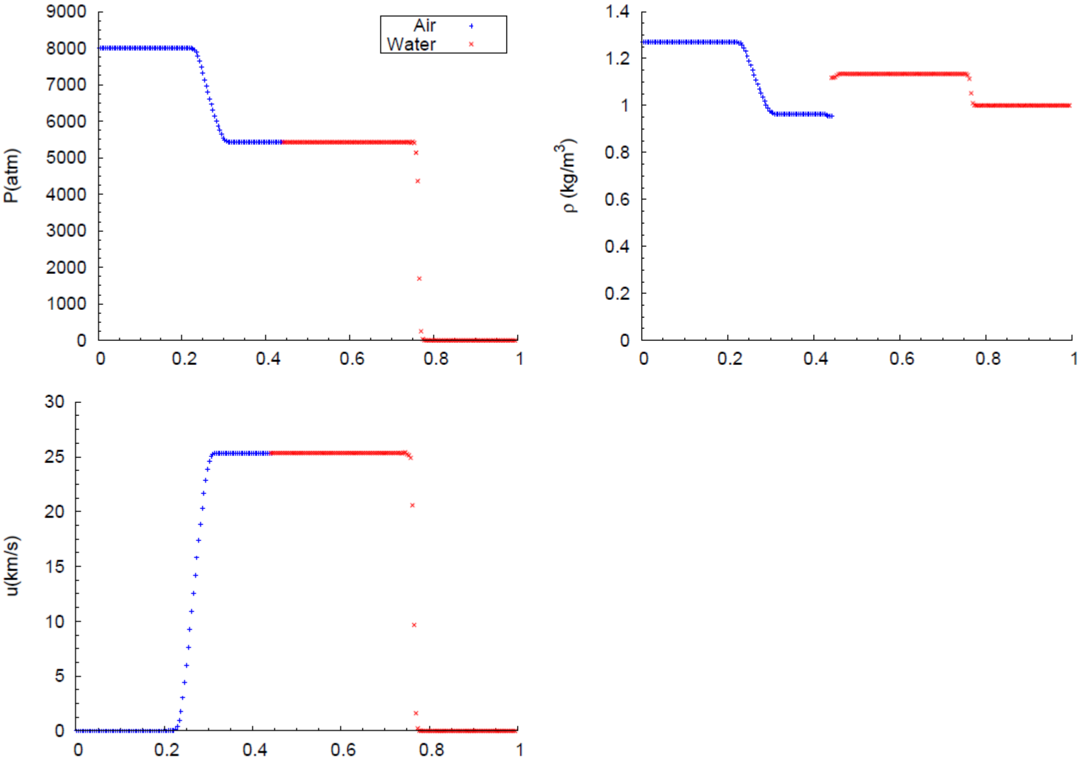


Figure 6.16: Air Water IVP, MPWENO-5 and 250 points grid size.

2. 500 points

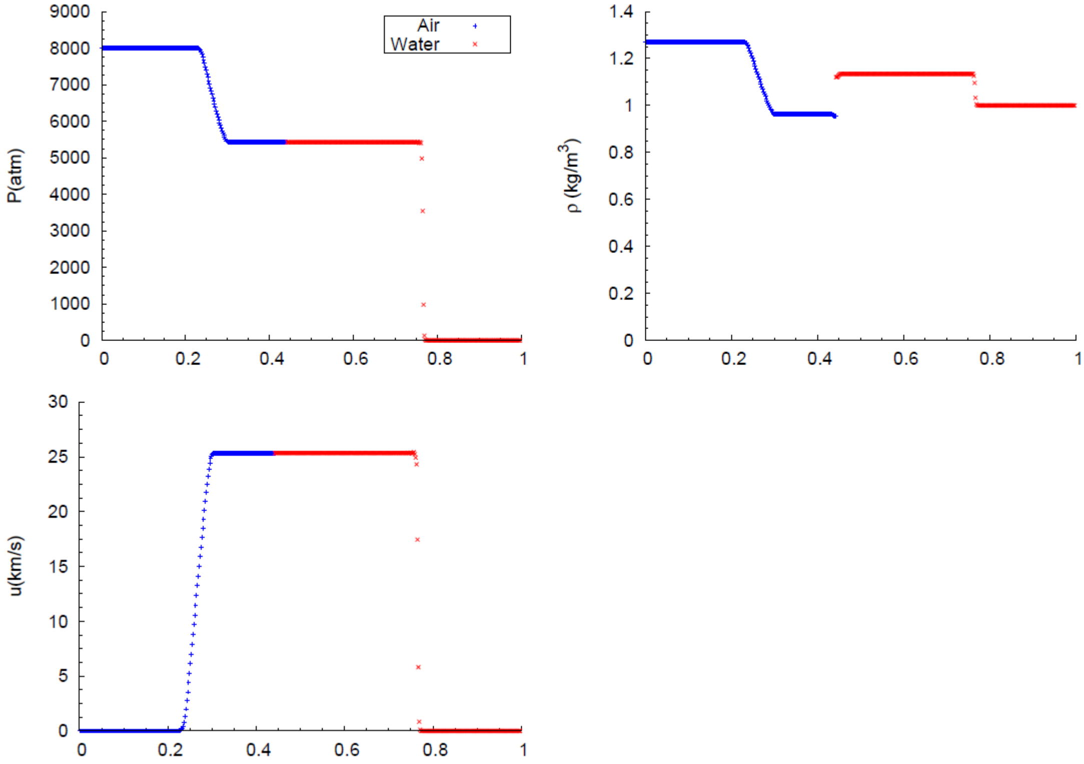


Figure 6.17: Air Water IVP, MPWENO-5 and 500 points grid size.

3. 1000 points

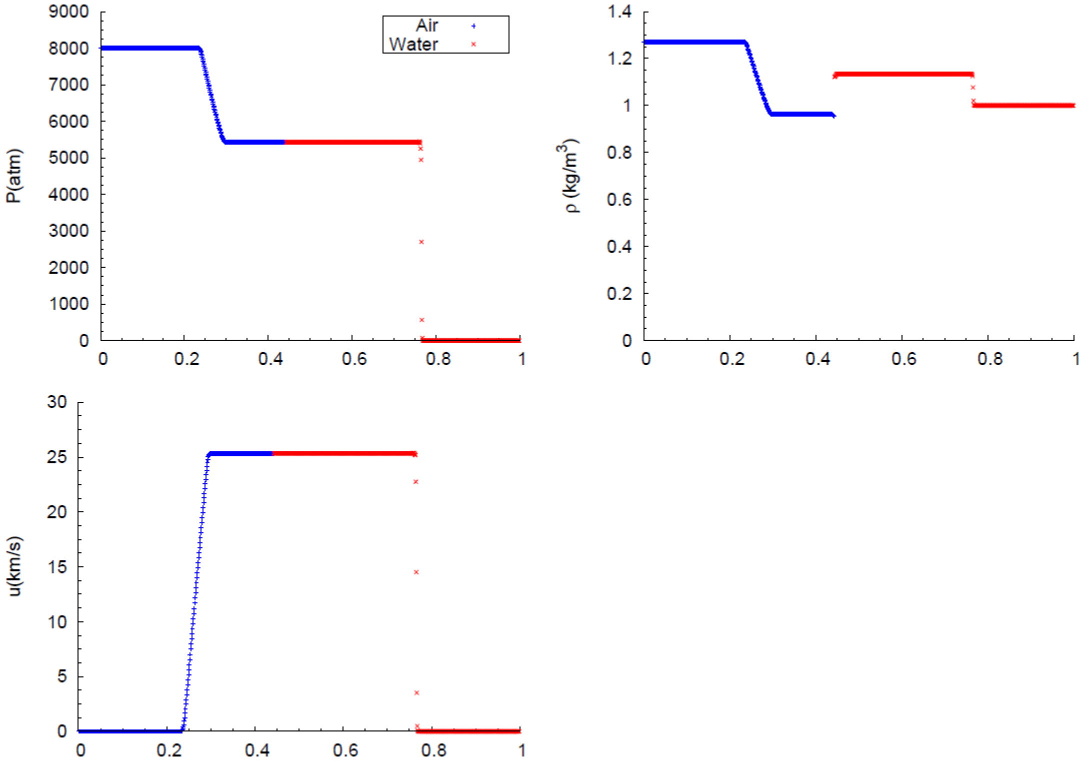


Figure 6.18: Air Water IVP, MPWENO-5 and 1000 points grid size.

IVP 3: JWL-Water shock tube

Below are presented all the graphs relative to different grid size and reconstruction methods used, i.e. 1st Order, WENO-3, and MPWENO-5.

1st Order

1. 250 points:

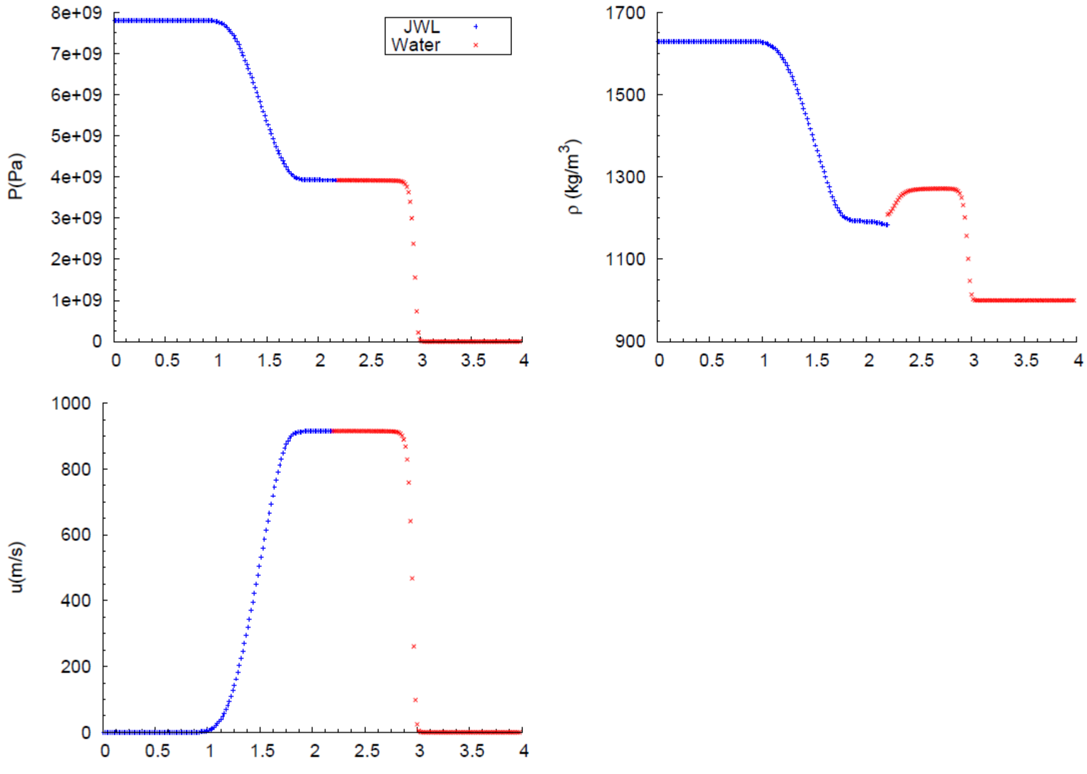


Figure 6.19: JWL Water IVP, 1st-Order and 250 points grid size.

2. 500 points

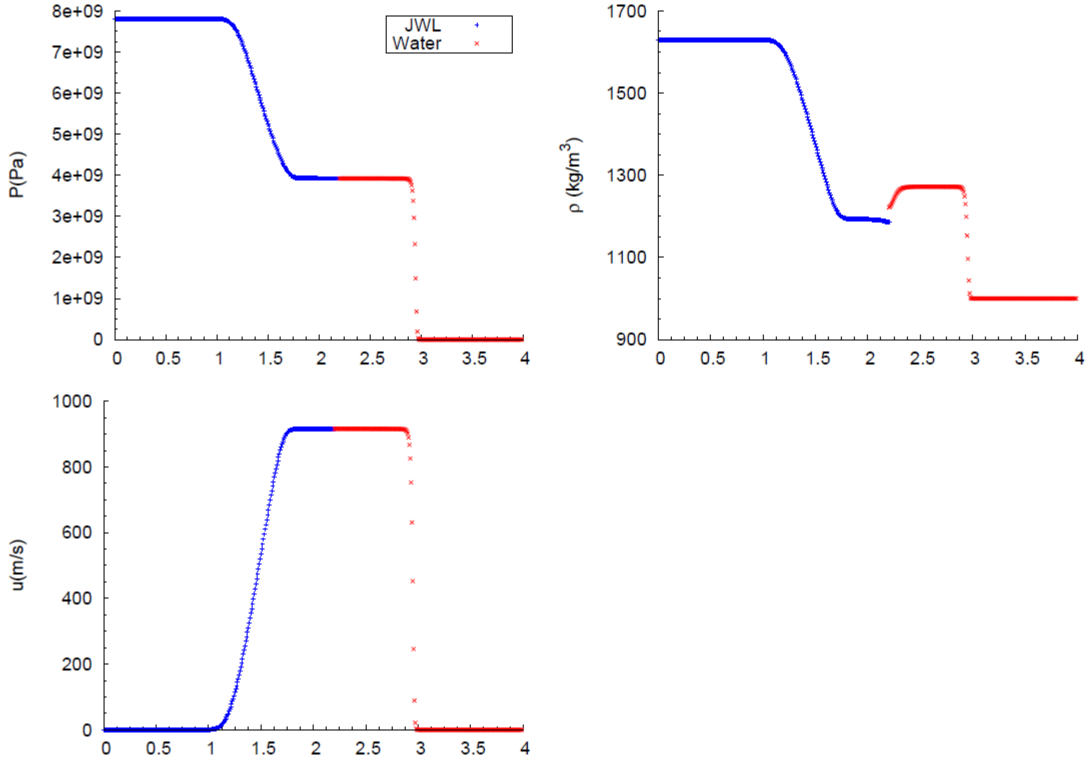


Figure 6.20: JWL Water IVP, 1st-Order and 500 points grid size.

3. 1000 points

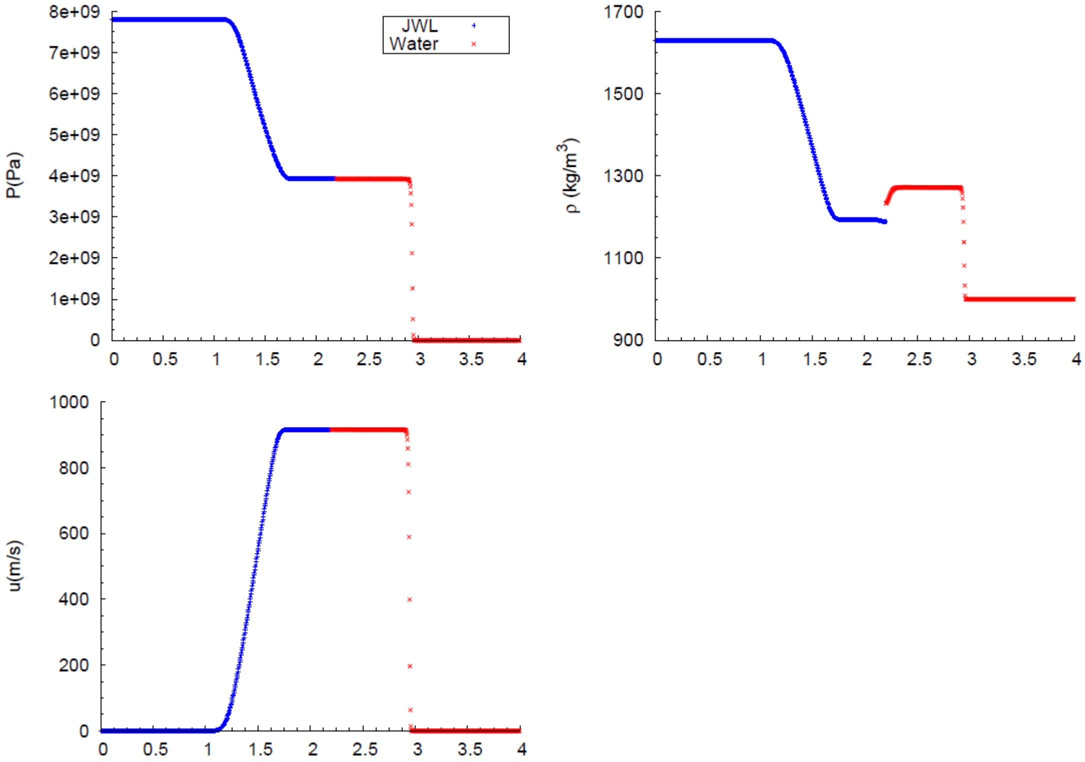


Figure 6.21: JWL Water IVP, 1st-Order and 1000 points grid size.

WENO-3

1. 250 points

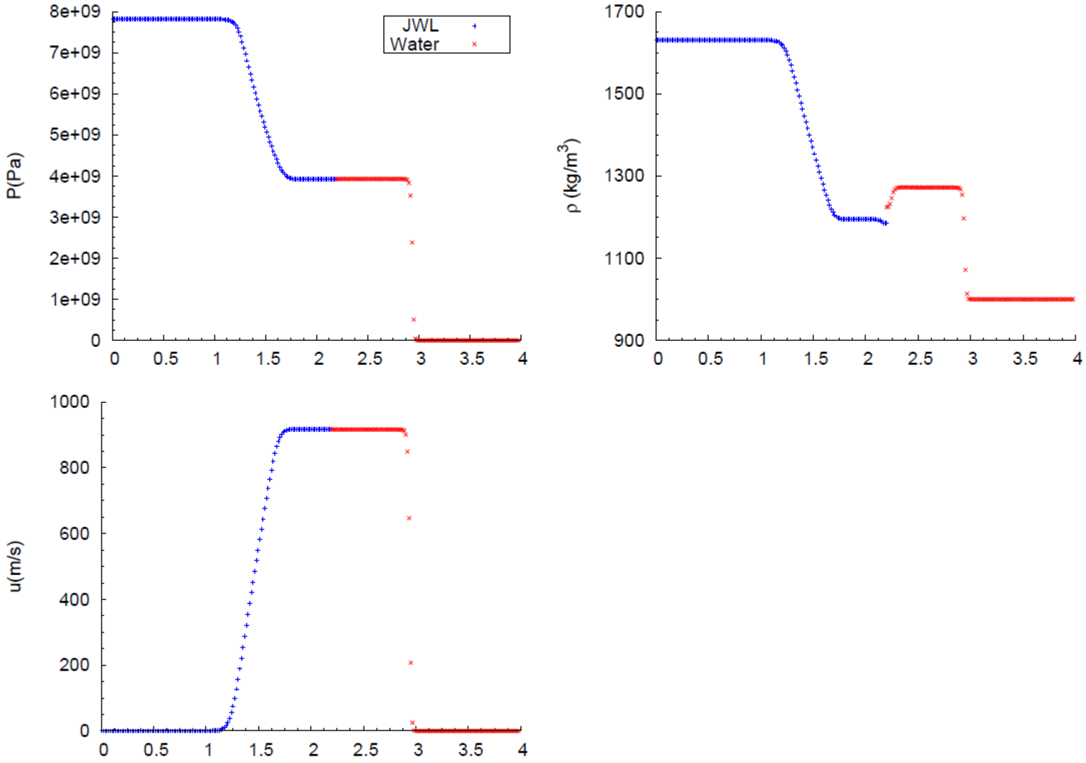


Figure 6.22: JWL Water IVP, WENO-3 and 250 points grid size.

2. 500 points

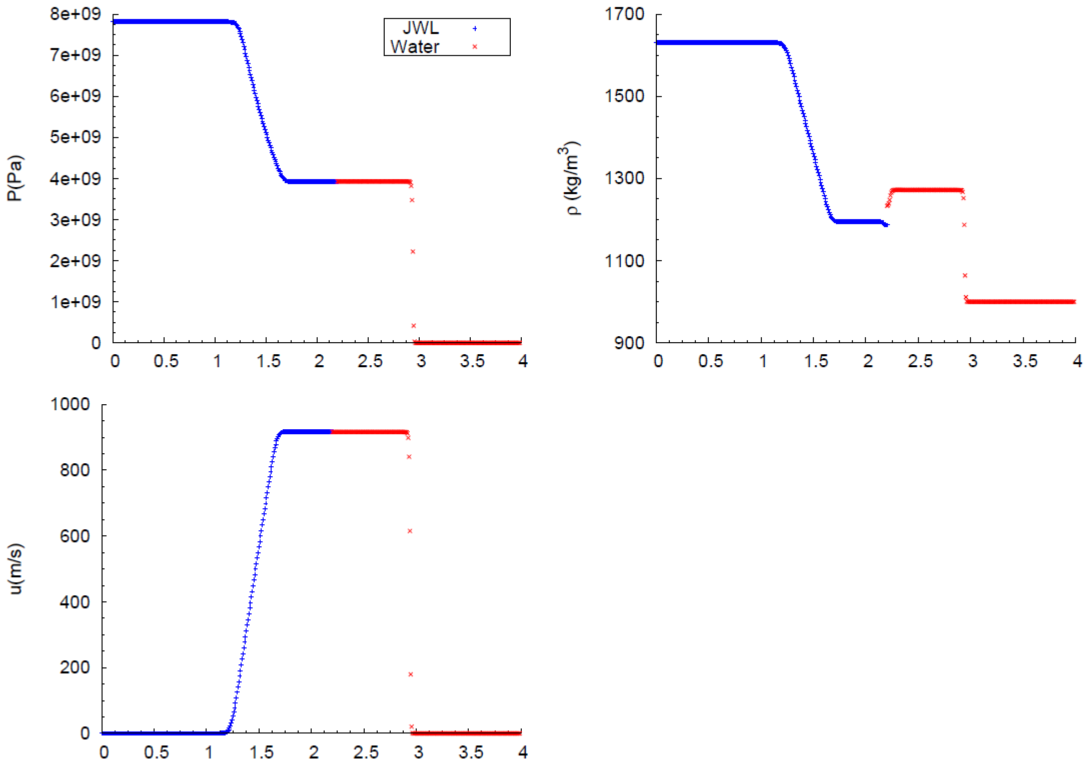


Figure 6.23: JWL Water IVP, WENO-3 and 500 points grid size.

3. 1000 points

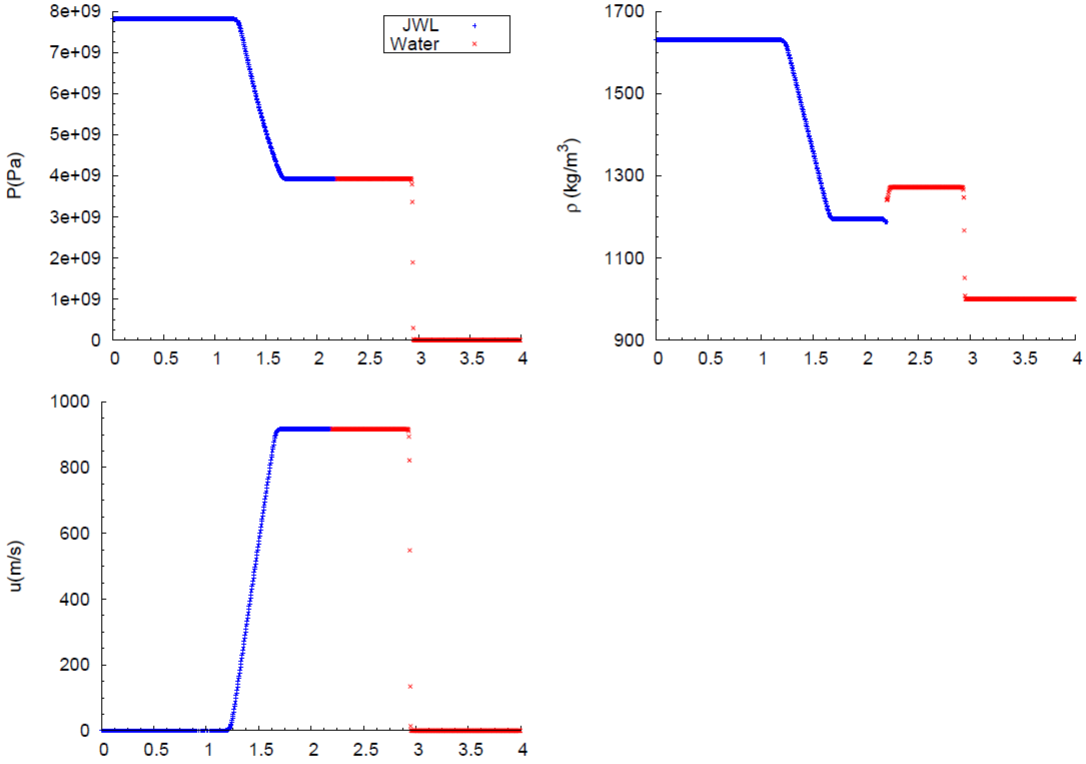


Figure 6.24: JWL Water IVP, WENO-3 and 1000 points grid size.

MPWENO-5

1. 250 points

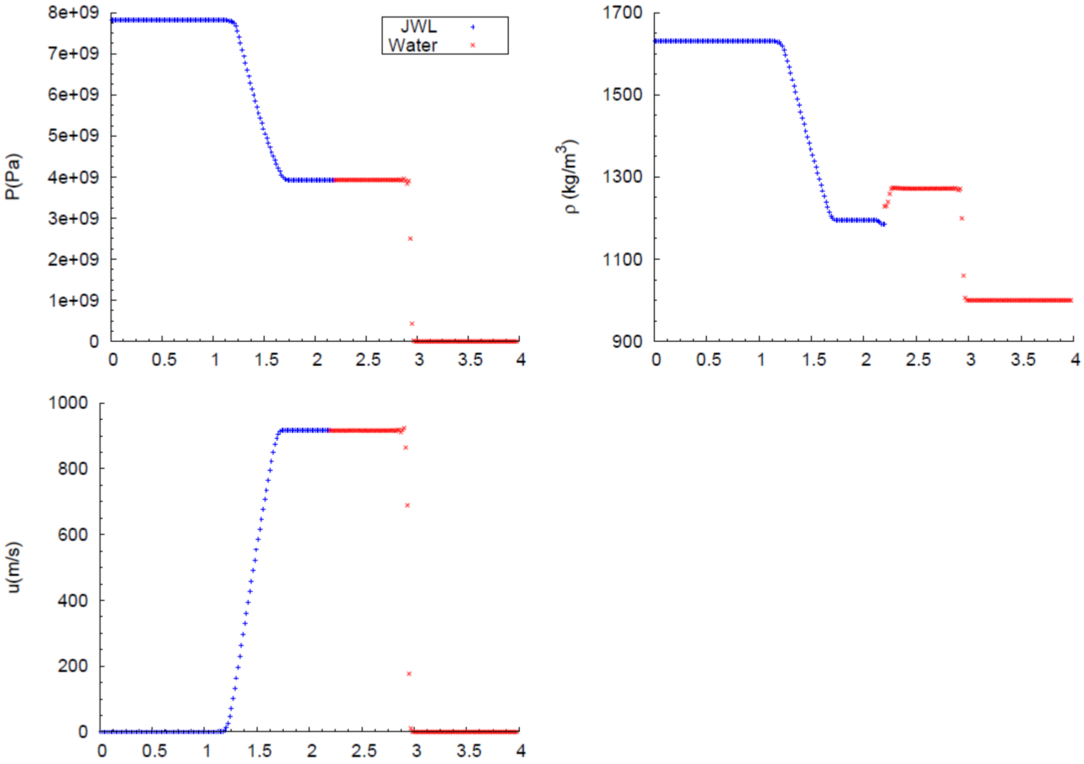


Figure 6.25: JWL Water IVP, MPWENO-5 and 250 points grid size.

2. 500 points

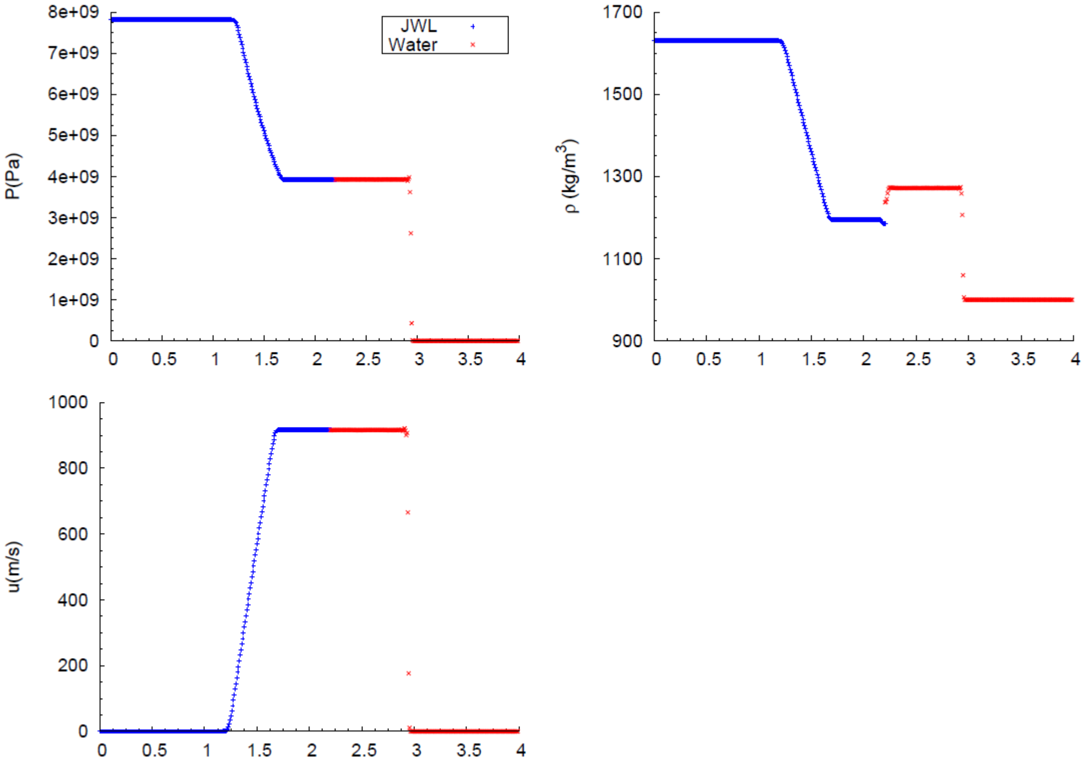


Figure 6.26: JWL Water IVP, MPWENO-5 and 500 points grid size.

3. 1000 points

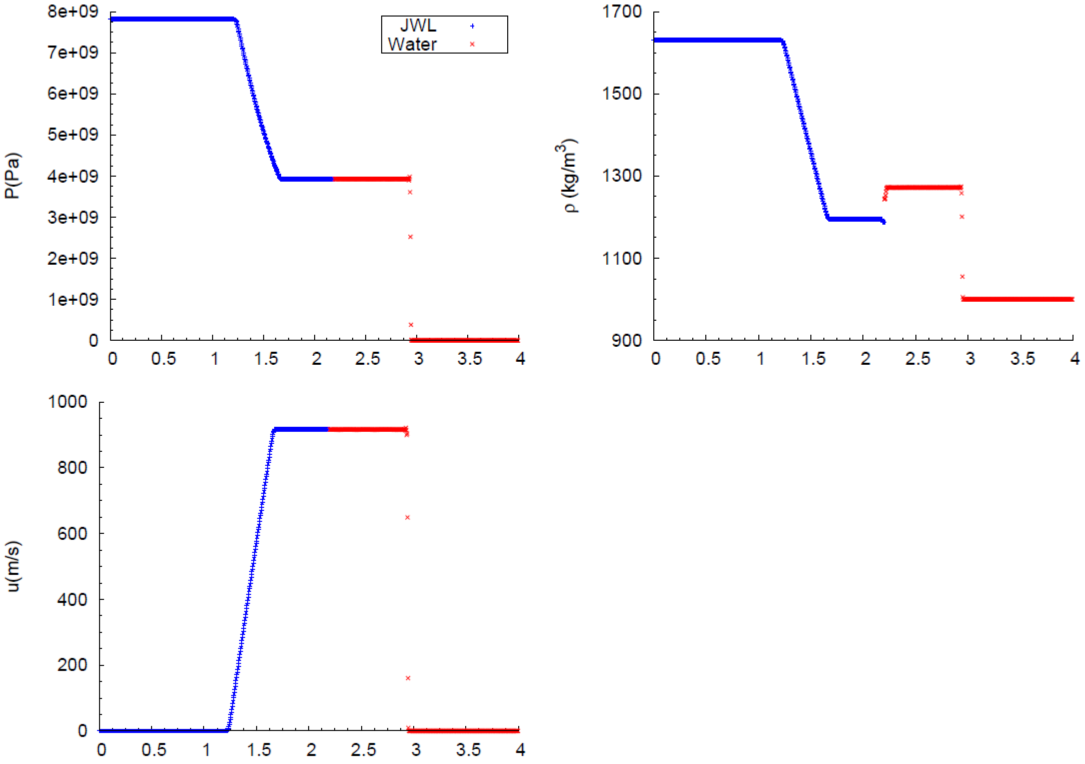


Figure 6.27: JWL Water IVP, MPWENO-5 and 1000 points grid size.

Appendix D: Complete 1D results for solid/fluid interaction.

IVP 1: Reacted PBX-9404 with unstressed copper

Below are presented all the graphs relative to different grid size and reconstruction methods used, i.e. 1st Order, WENO-3, and MPWENO-5.

1st Order:

1. 250 points:

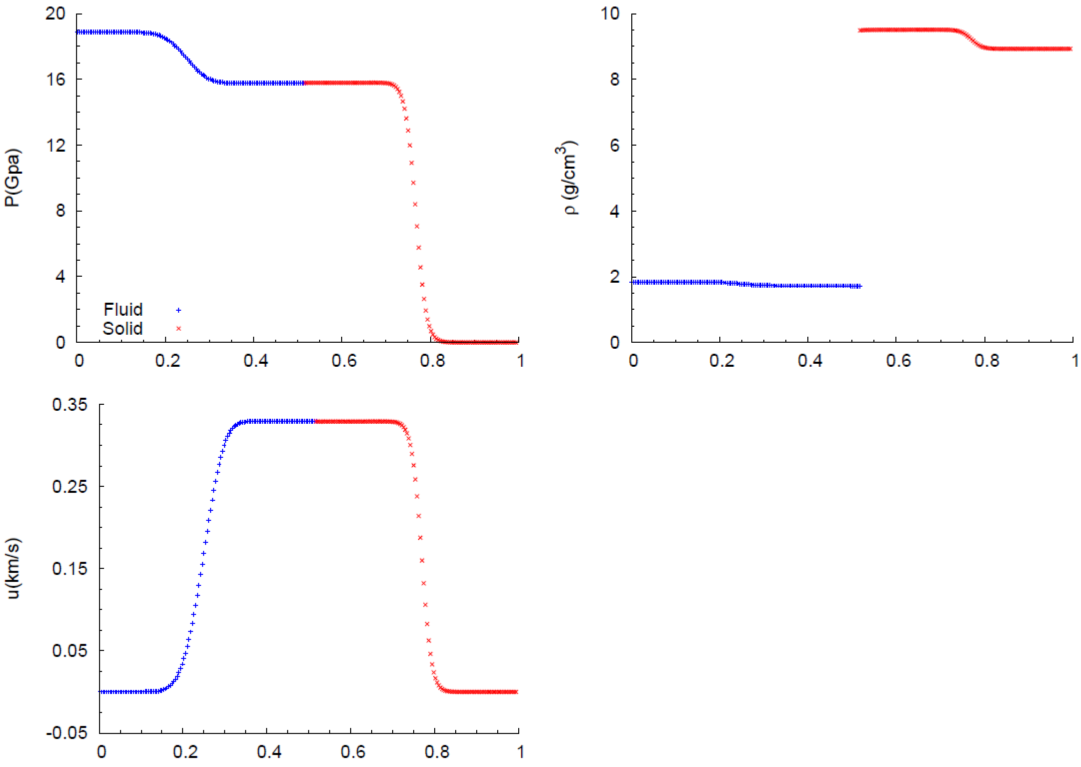


Figure 6.28: Reacted PBX-9404 with unstressed copper IVP, 1st-Order and 250 points grid size.

2. 500 points

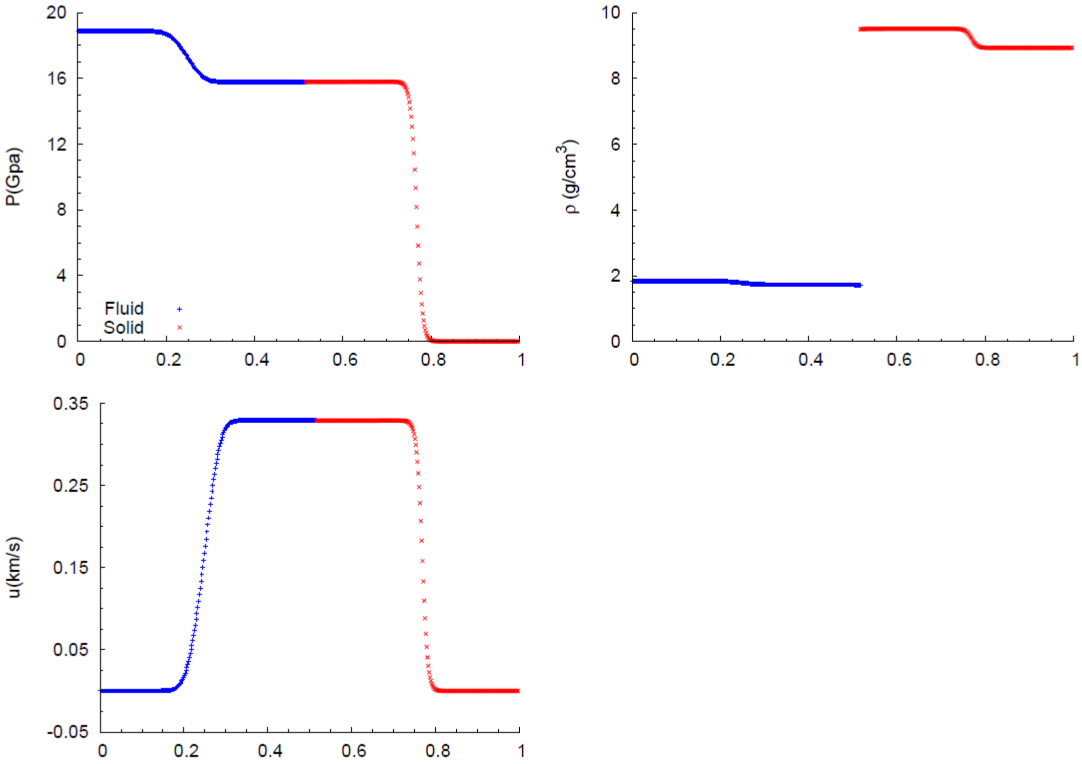


Figure 6.29: Reacted PBX-9404 with unstressed copper IVP, 1st-Order and 500 points grid size.

3. 1000 points

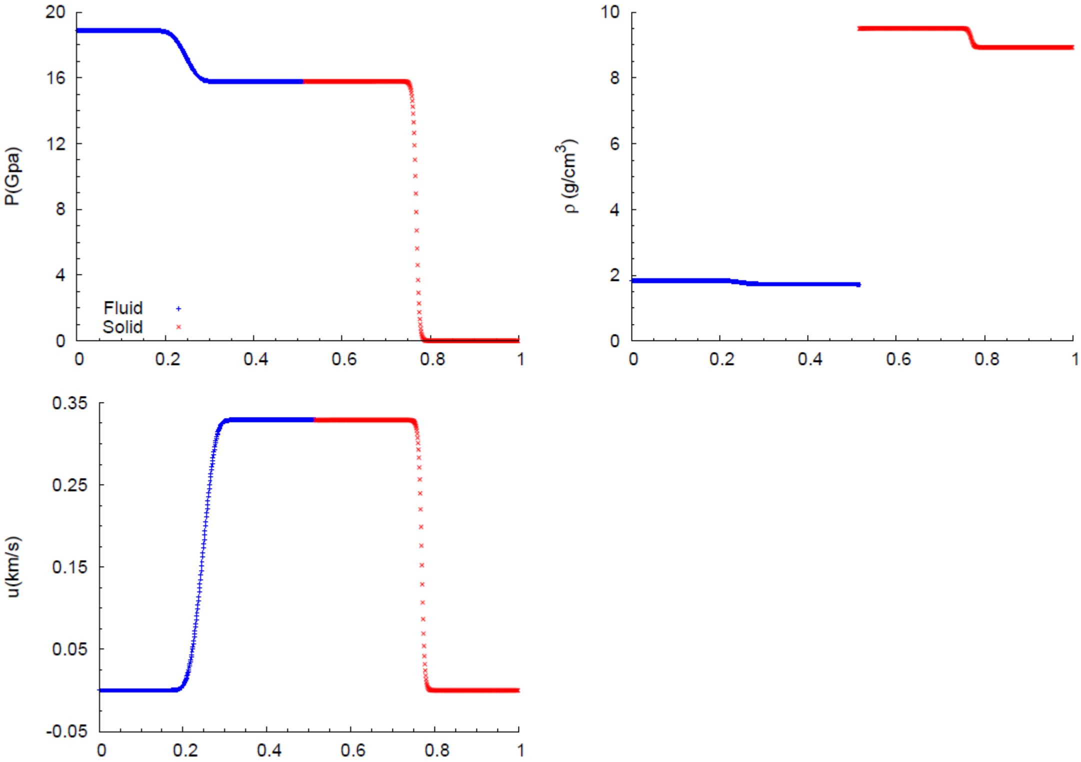


Figure 6.30: Reacted PBX-9404 with unstressed copper IVP, 1st-Order and 1000 points grid size.

WENO-3

1. 250 points

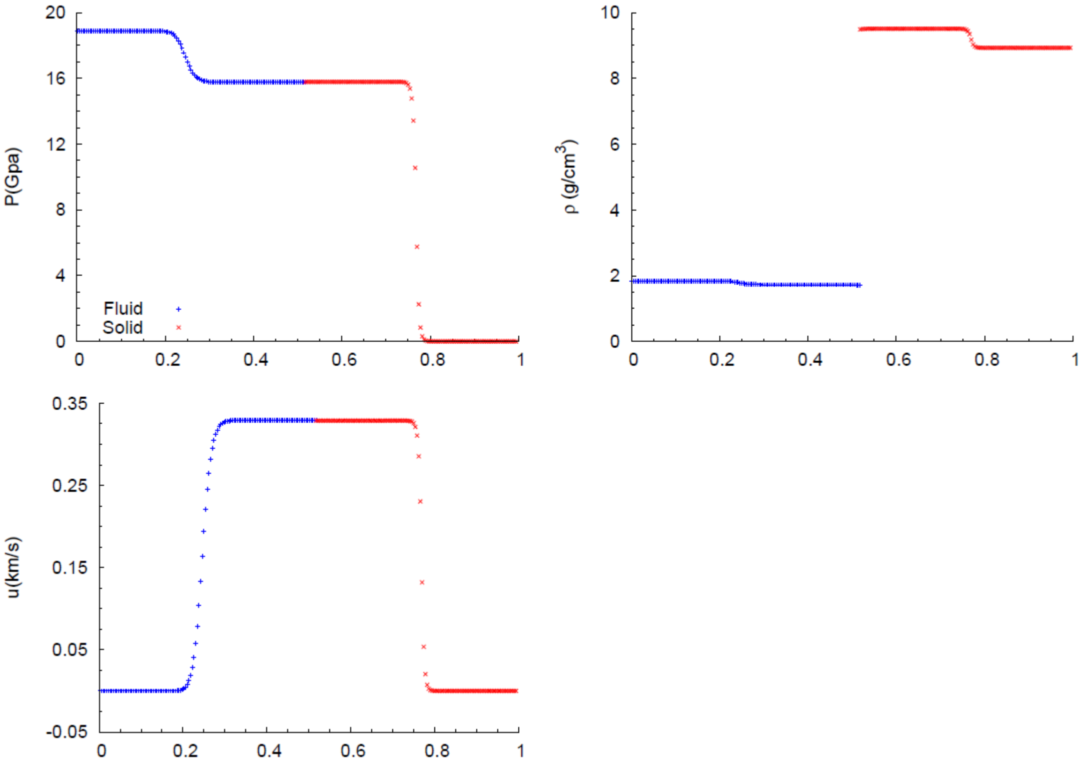


Figure 6.31: Reacted PBX-9404 with unstressed copper IVP, WENO-3 and 250 points grid size.

2. 500 points

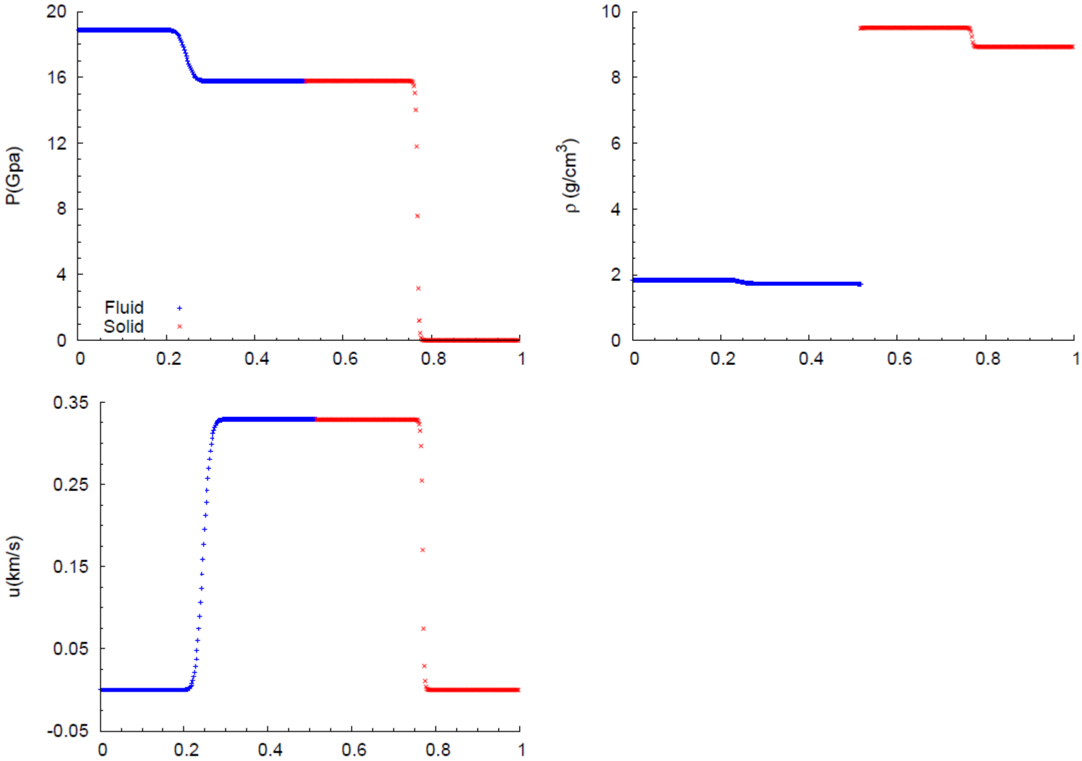


Figure 6.32: Reacted PBX-9404 with unstressed copper IVP, WENO-3 and 500 points grid size.

3. 1000 points

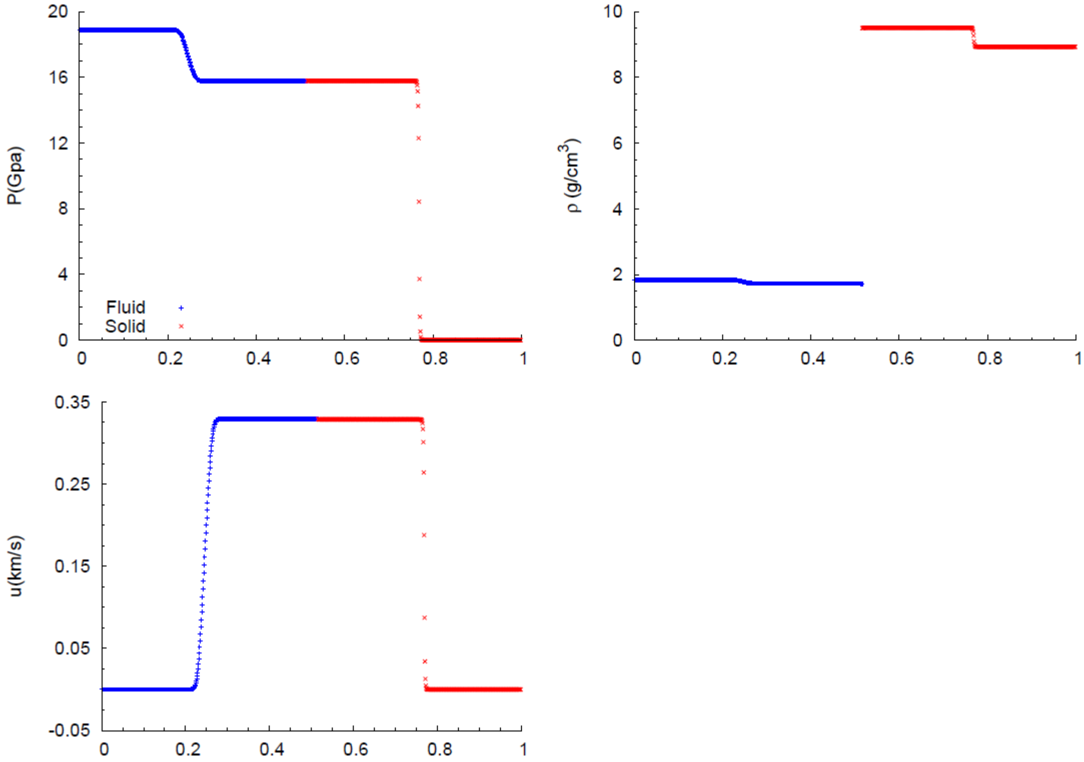


Figure 6.33: Reacted PBX-9404 with unstressed copper IVP, WENO-3 and 1000 points grid size.

MPWENO-5

1. 250 points

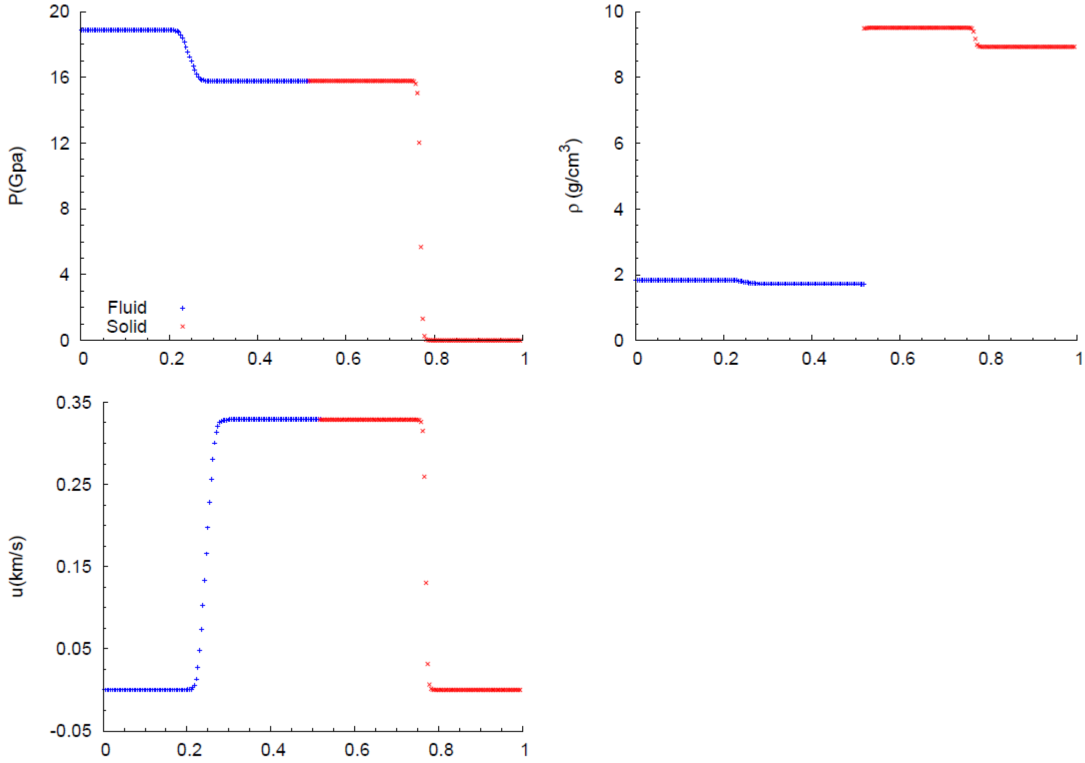


Figure 6.34: Reacted PBX-9404 with unstressed copper IVP, MPWENO-5 and 250 points grid size.

2. 500 points

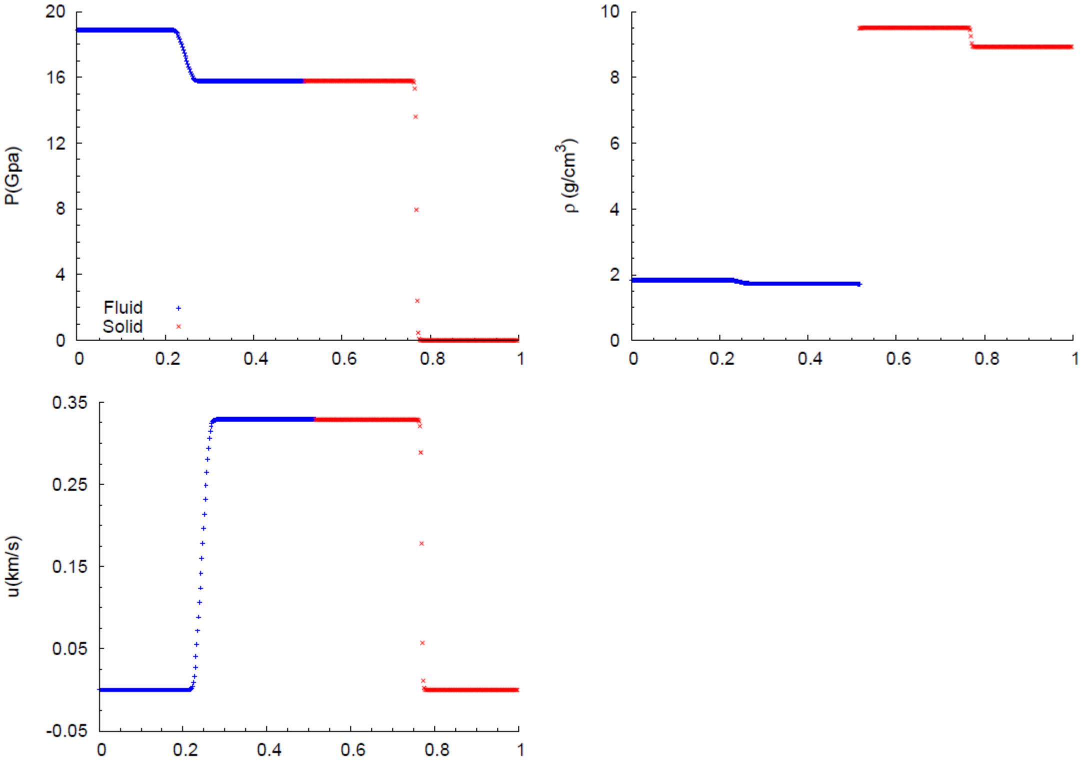


Figure 6.35: Reacted PBX-9404 with unstressed copper IVP, MPWENO-5 and 500 points grid size.

3. 1000 points

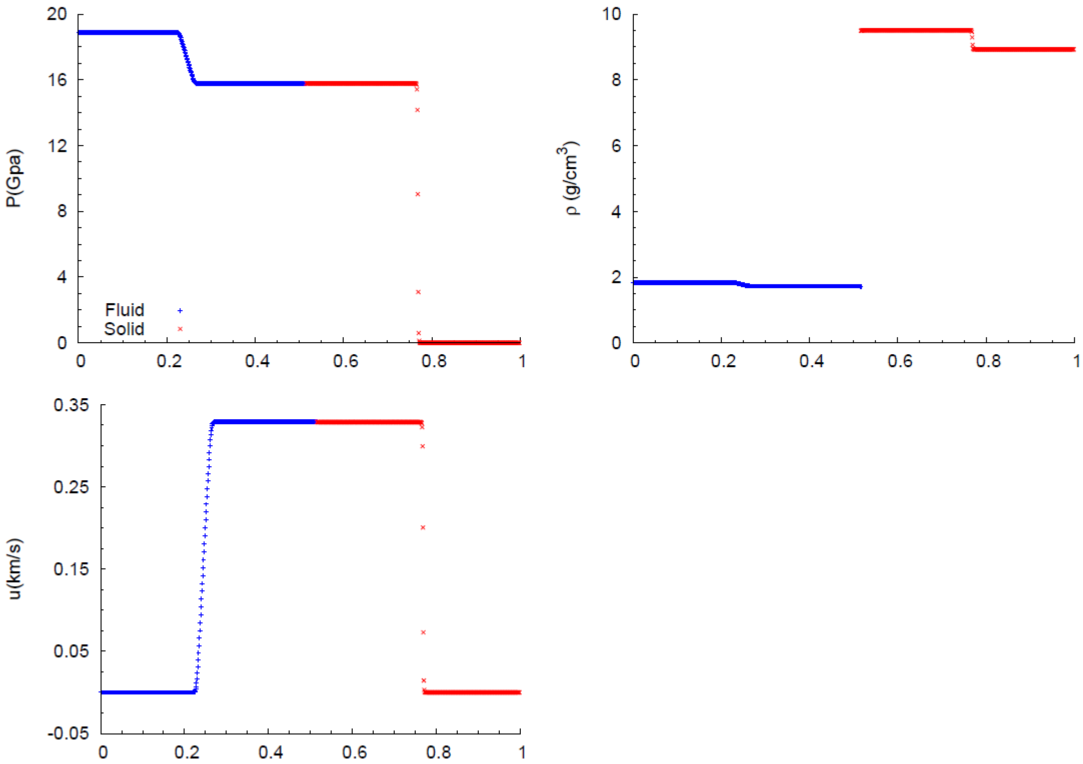


Figure 6.36: Reacted PBX-9404 with unstressed copper IVP, MPWENO-5 and 1000 points grid size.

IVP 2: Unreacted PBX-9404 with stressed copper

Below are presented all the graphs relative to different grid size and reconstruction methods used, i.e. 1st Order, WENO-3, and MPWENO-5.

1st Order:

1. 250 points:

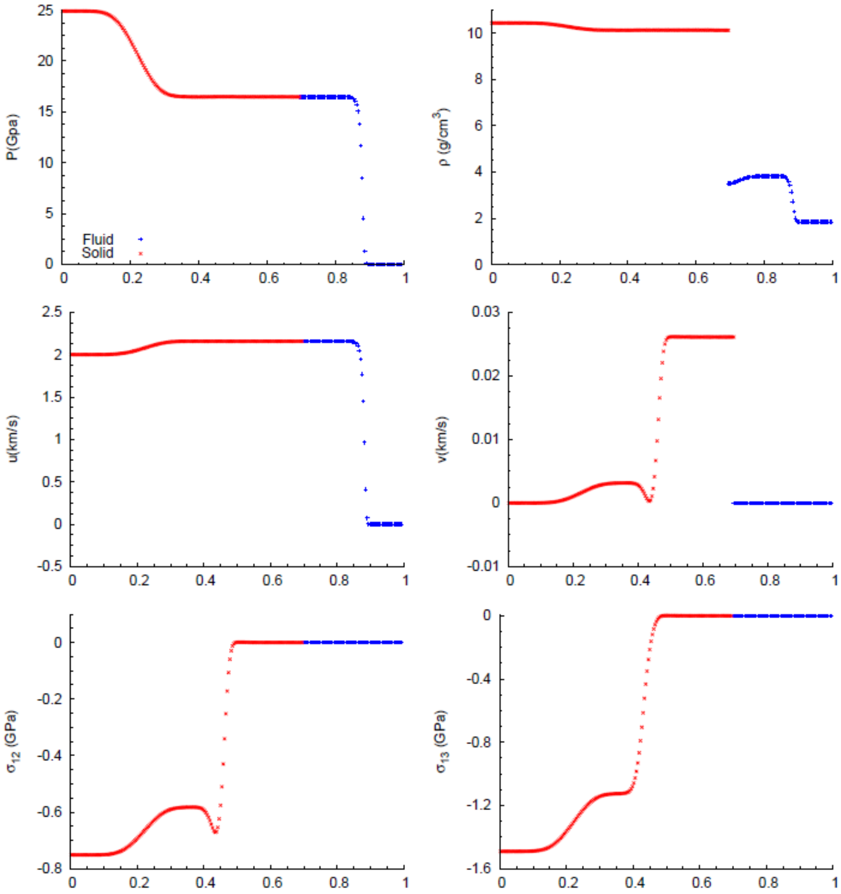


Figure 6.37: Unreacted PBX-9404 with stressed copper IVP, 1st-Order and 250 points grid size.

2. 500 points

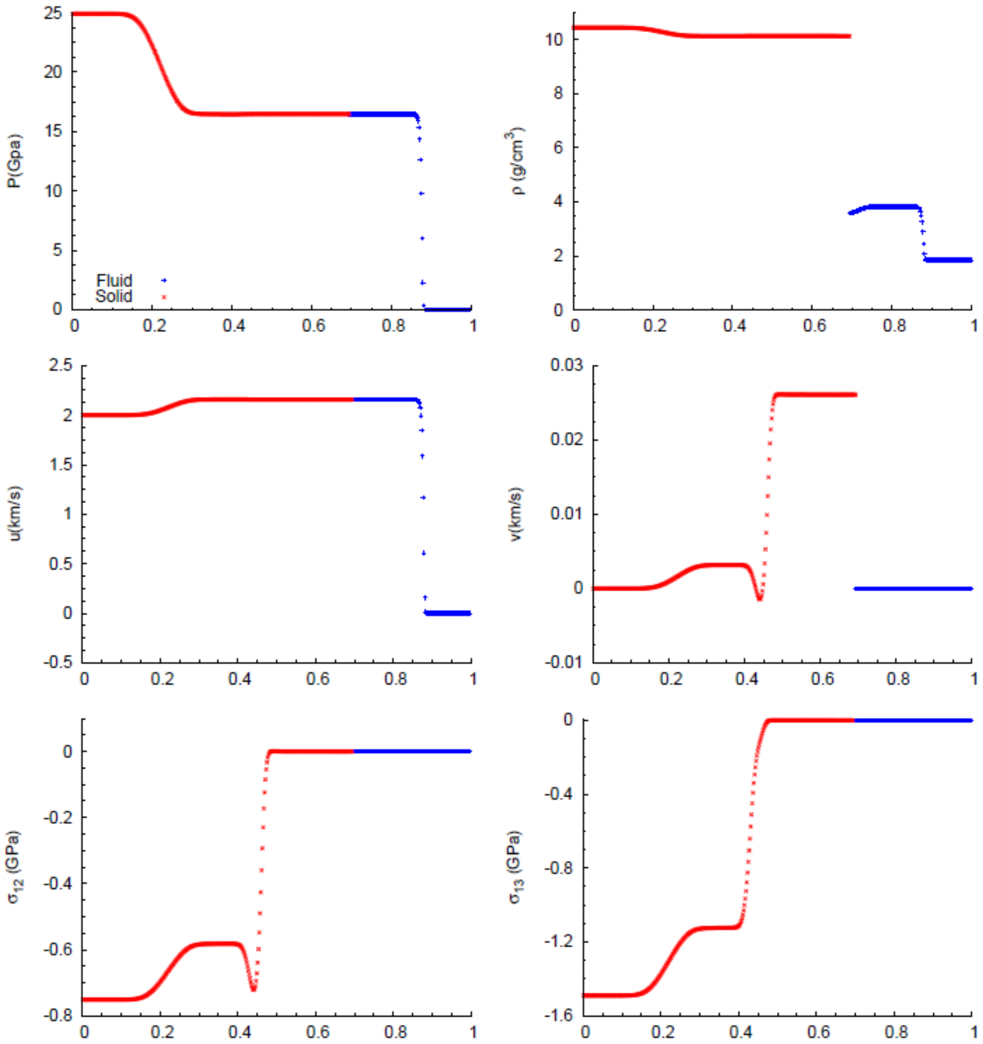


Figure 6.38: Unreacted PBX-9404 with stressed copper IVP, 1st-Order and 500 points grid size.

3. 1000 points

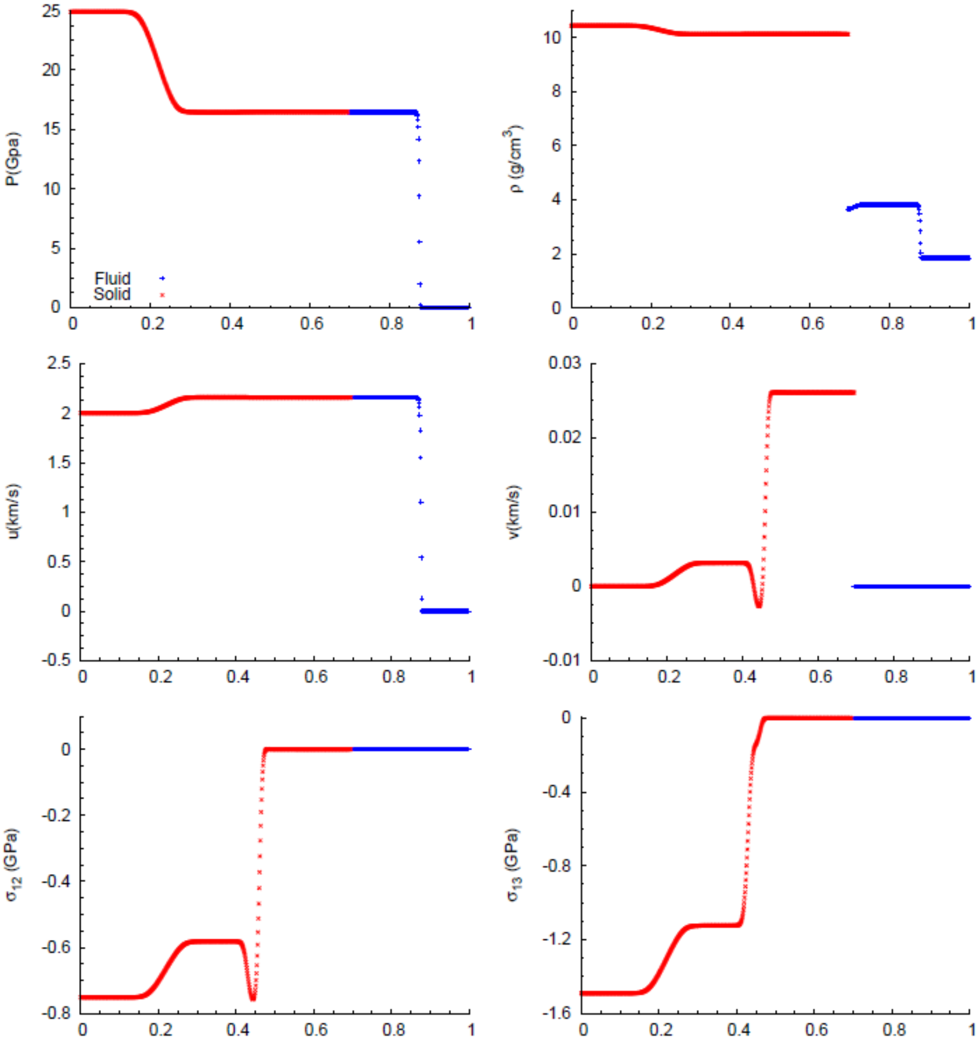


Figure 6.39: Unreacted PBX-9404 with stressed copper IVP, 1st-Order and 1000 points grid size.

WENO-3

1. 250 points

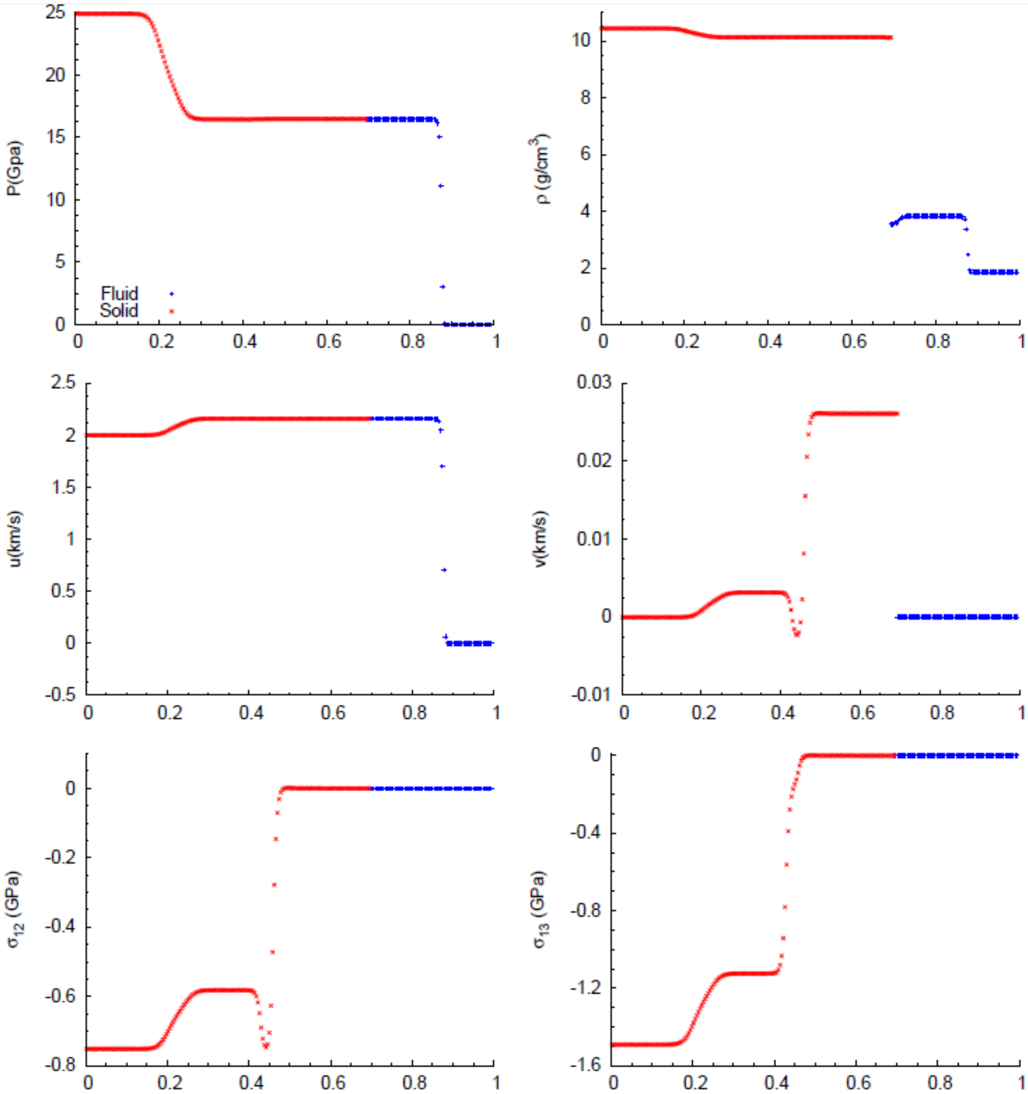


Figure 6.40: Unreacted PBX-9404 with stressed copper IVP, WENO-3 and 250 points grid size.

2. 500 points

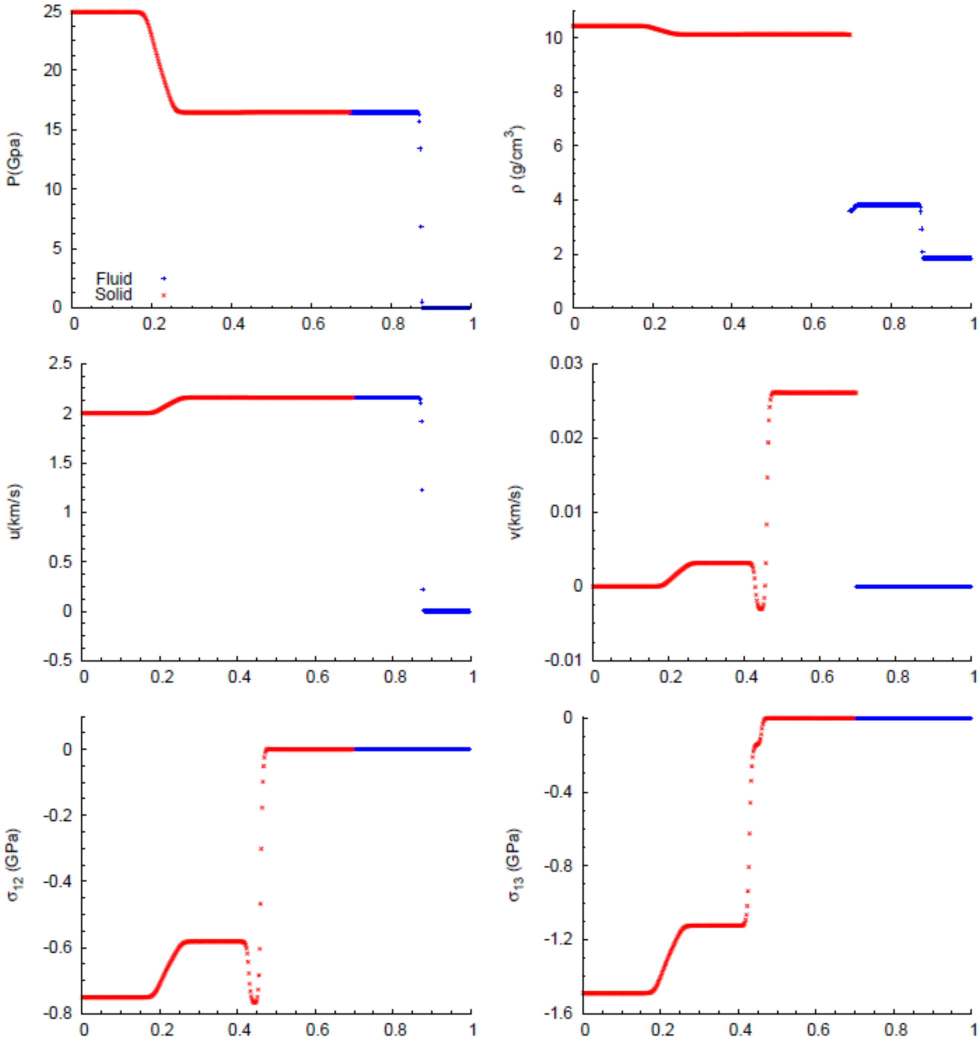


Figure 6.41: Unreacted PBX-9404 with stressed copper IVP, WENO-3 and 500 points grid size.

3. 1000 points

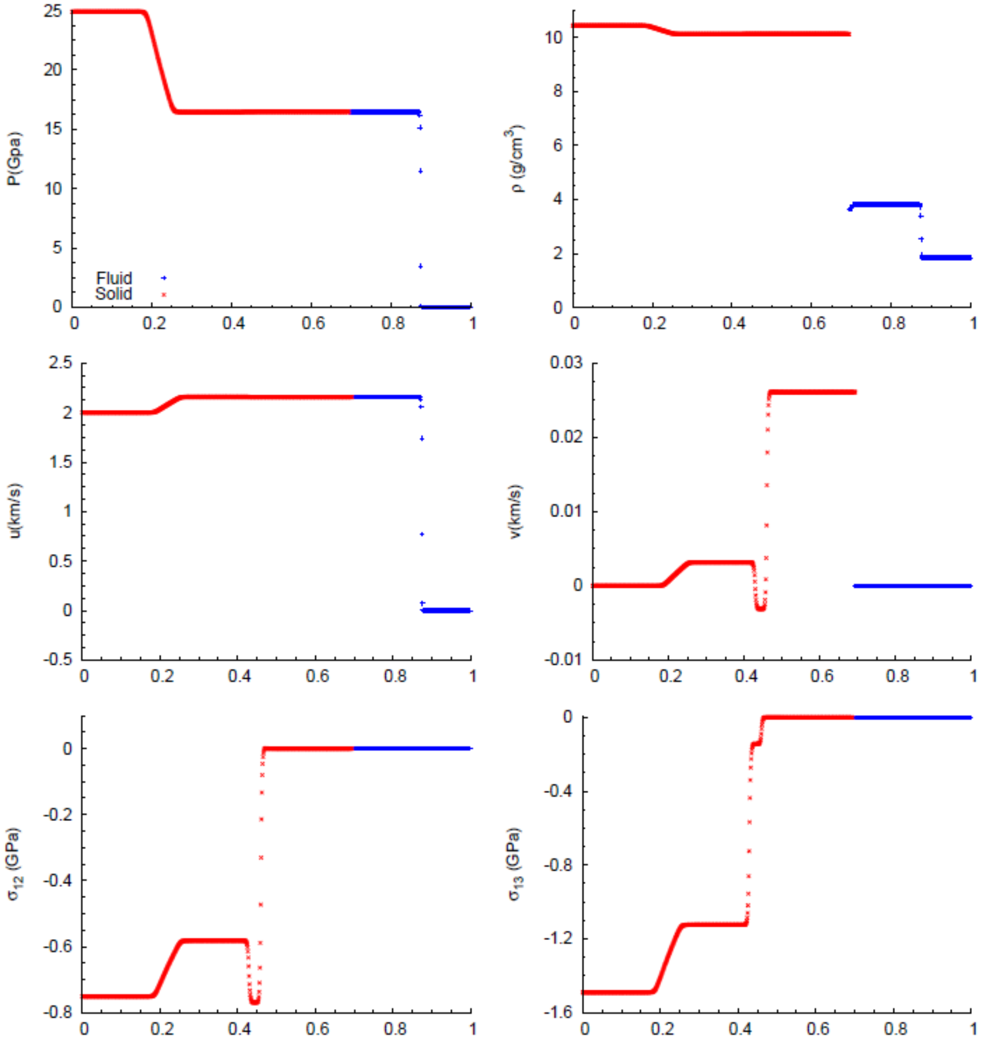


Figure 6.42: Unreacted PBX-9404 with stressed copper IVP, WENO-3 and 1000 points grid size.

MPWENO-5

1. 250 points

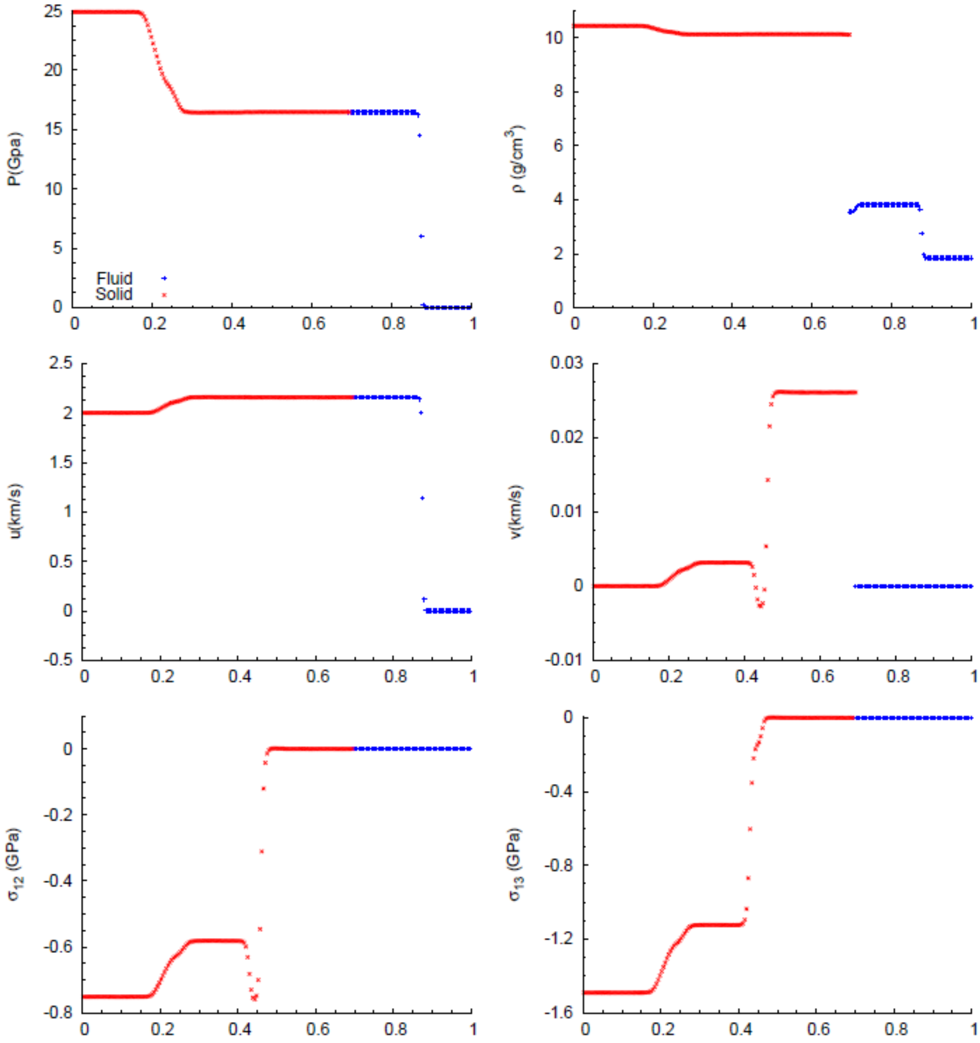


Figure 6.43: Unreacted PBX-9404 with stressed copper IVP, MPWENO-5 and 250 points grid size.

2. 500 points

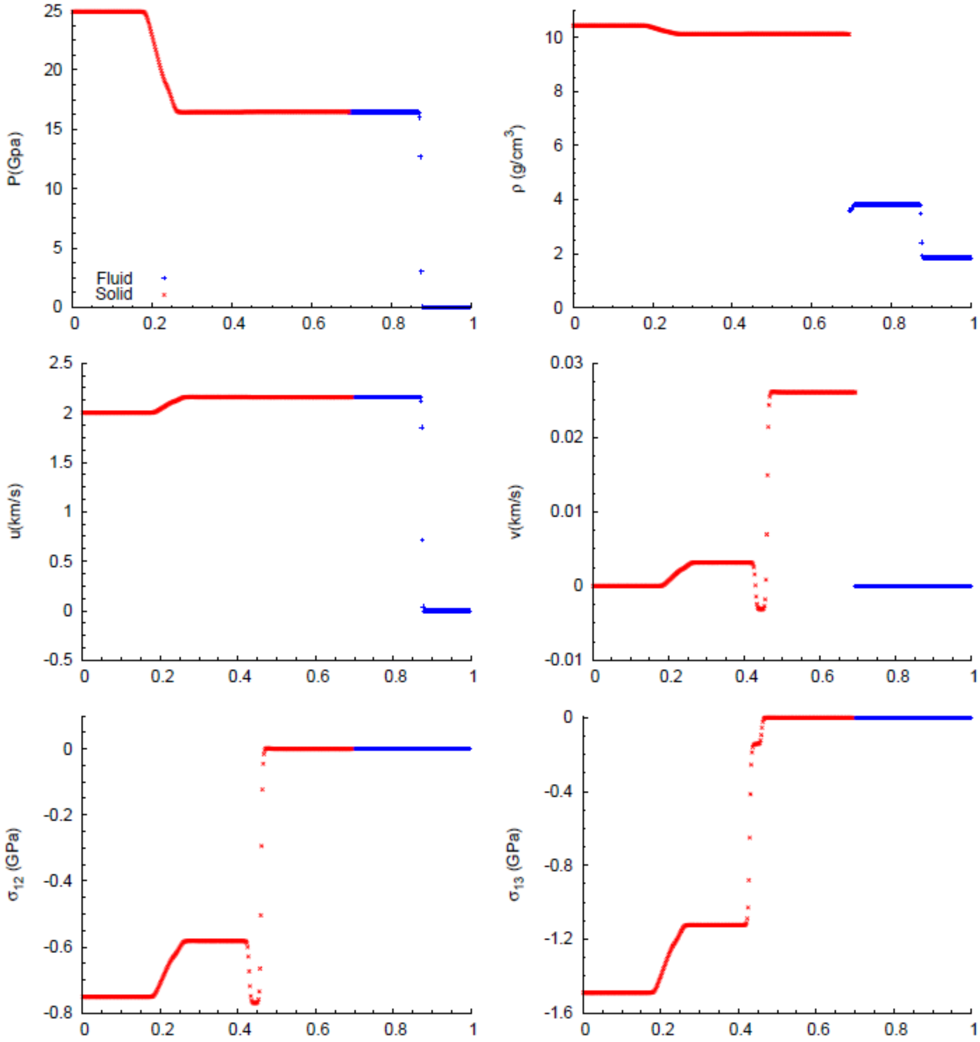


Figure 6.44: Unreacted PBX-9404 with stressed copper IVP, MPWENO-5 and 500 points grid size.

3. 1000 points

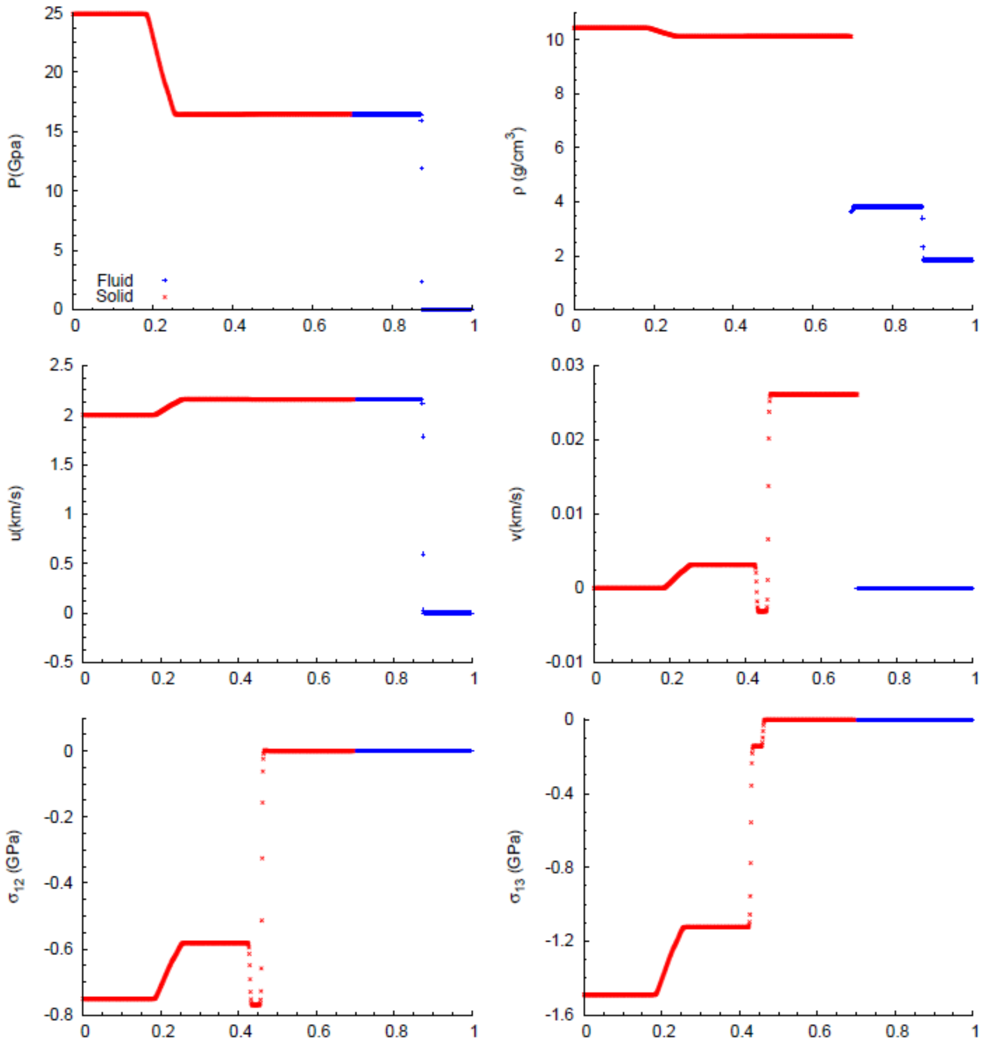


Figure 6.45: Unreacted PBX-9404 with stressed copper IVP, MPWENO-5 and 1000 points grid size.

IVP 3: TNT with unstressed copper

Below are presented all the graphs relative to different grid size and reconstruction methods used, i.e. 1st Order, WENO-3, and MPWENO-5.

1st Order:

1. 250 points:

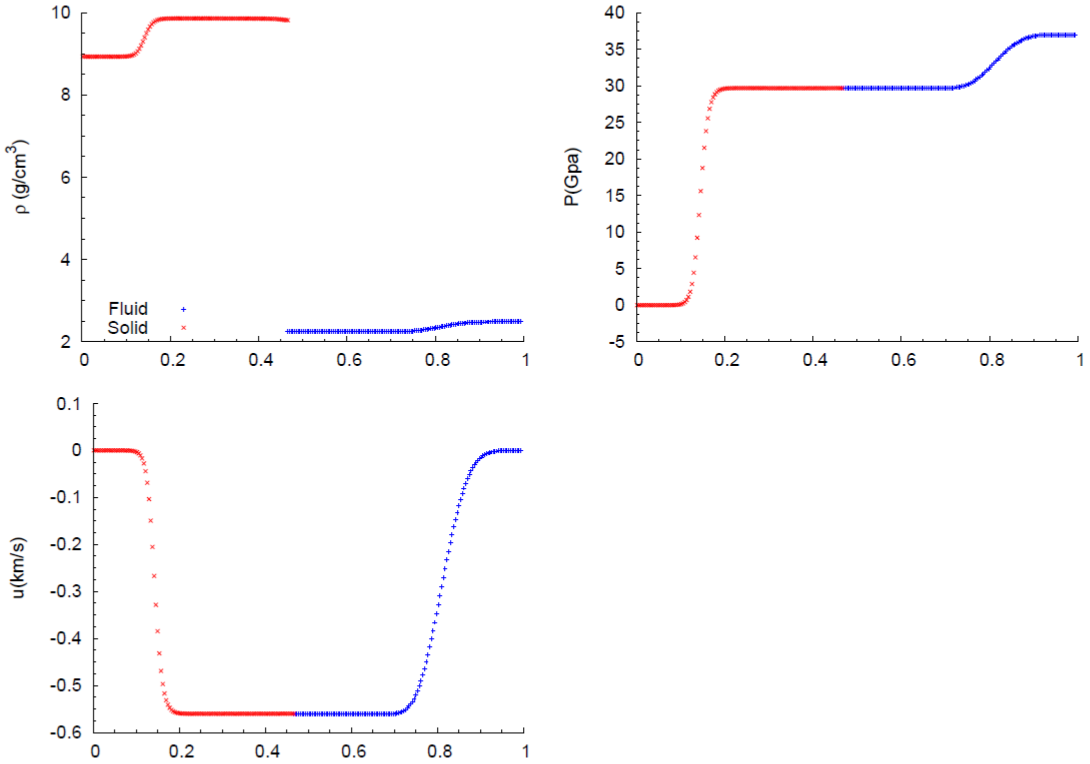


Figure 6.46: TNT Copper IVP, 1st-Order and 250 points grid size.

2. 500 points

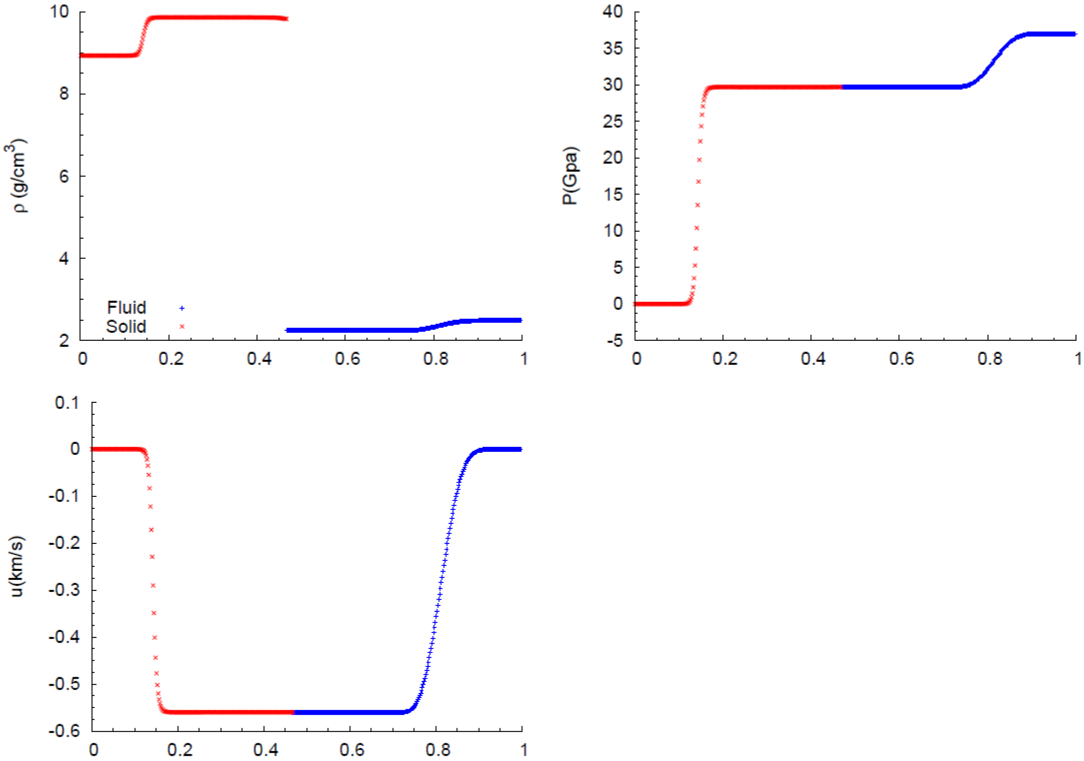


Figure 6.47: TNT Copper IVP, 1st-Order and 500 points grid size.

3. 1000 points

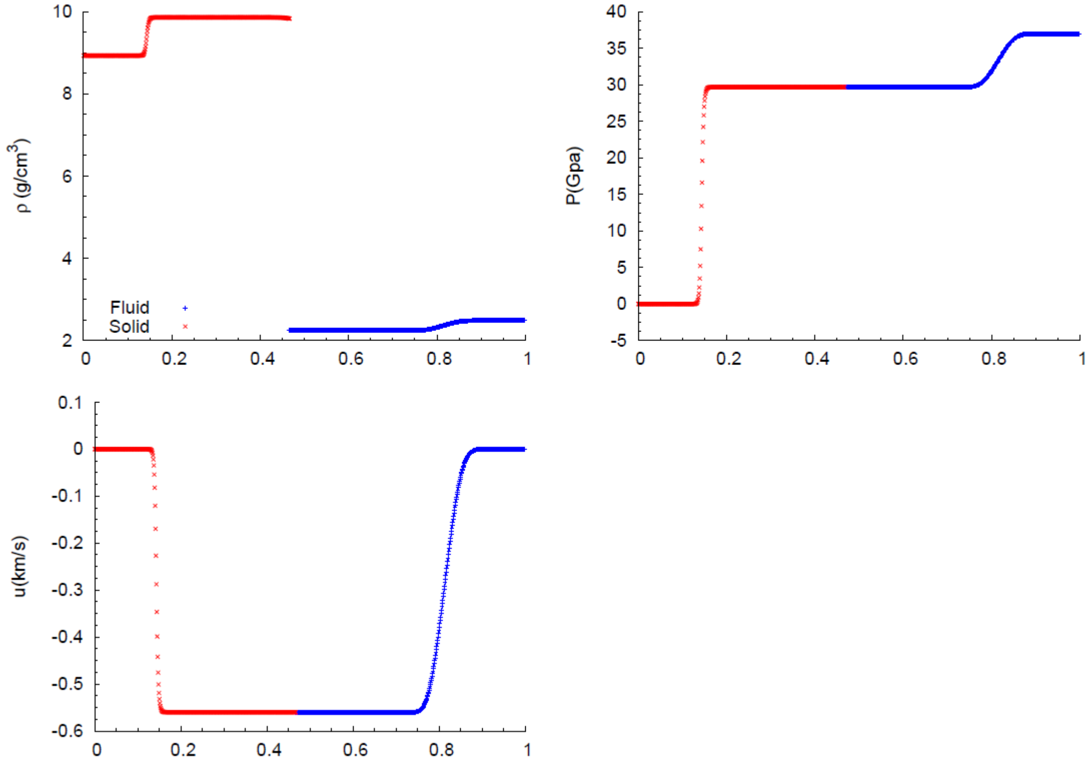


Figure 6.48: TNT Copper IVP, 1st-Order and 1000 points grid size.

WENO-3

1. 250 points

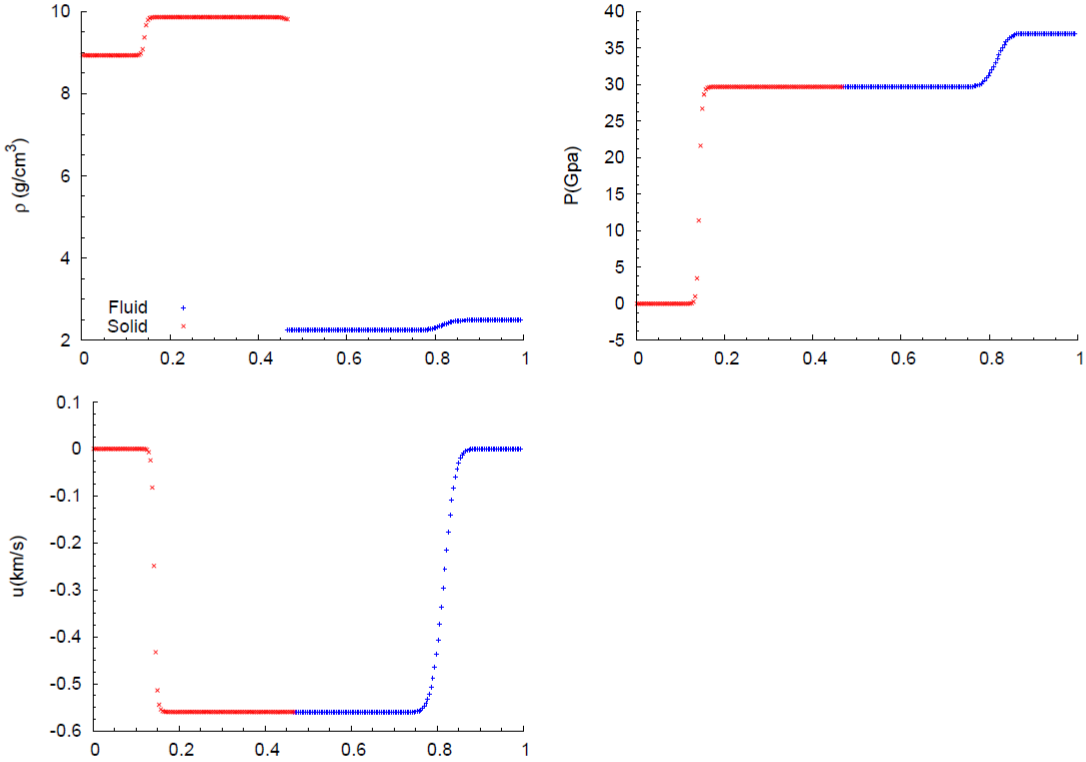


Figure 6.49: TNT Copper IVP, WENO-3 and 250 points grid size.

2. 500 points

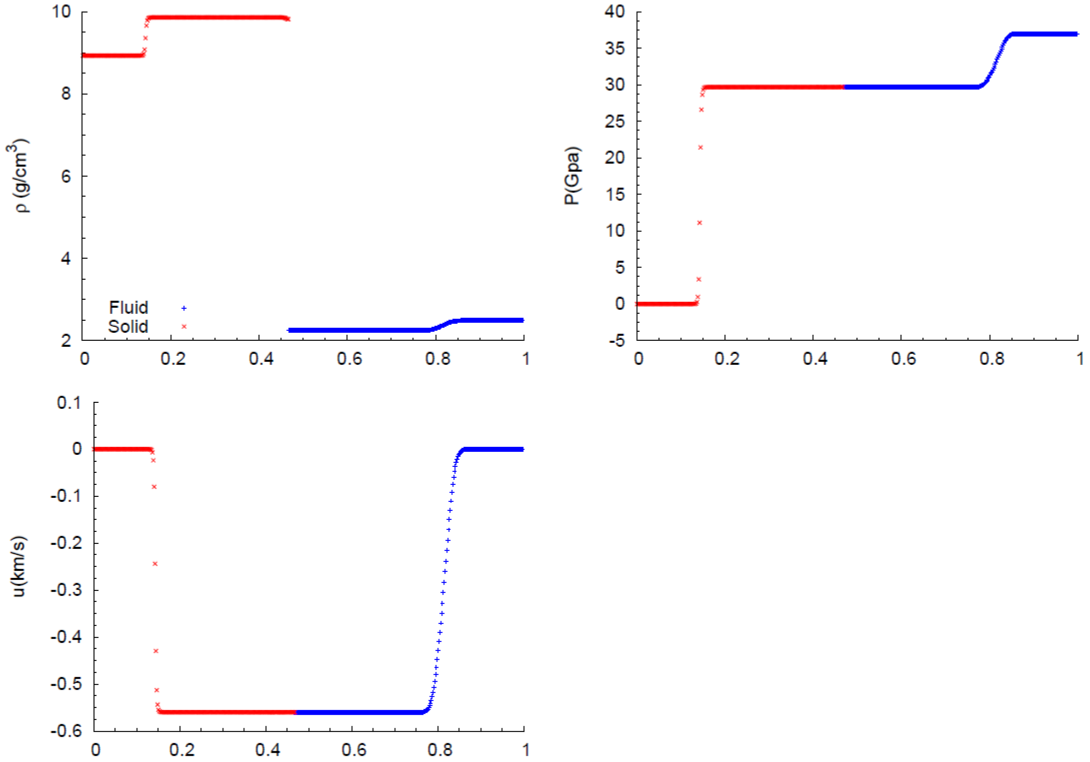


Figure 6.50: TNT Copper IVP, WENO-3 and 500 points grid size.

3. 1000 points

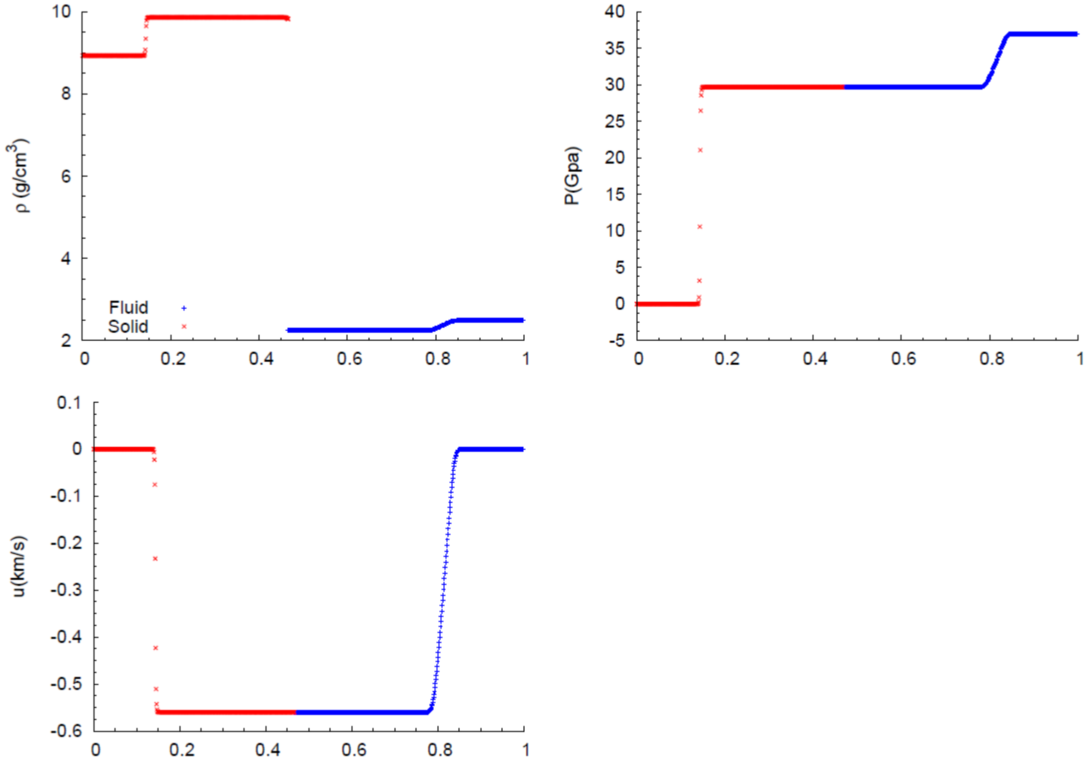


Figure 6.51: TNT Copper IVP, WENO-3 and 1000 points grid size.

MPWENO-5

1. 250 points

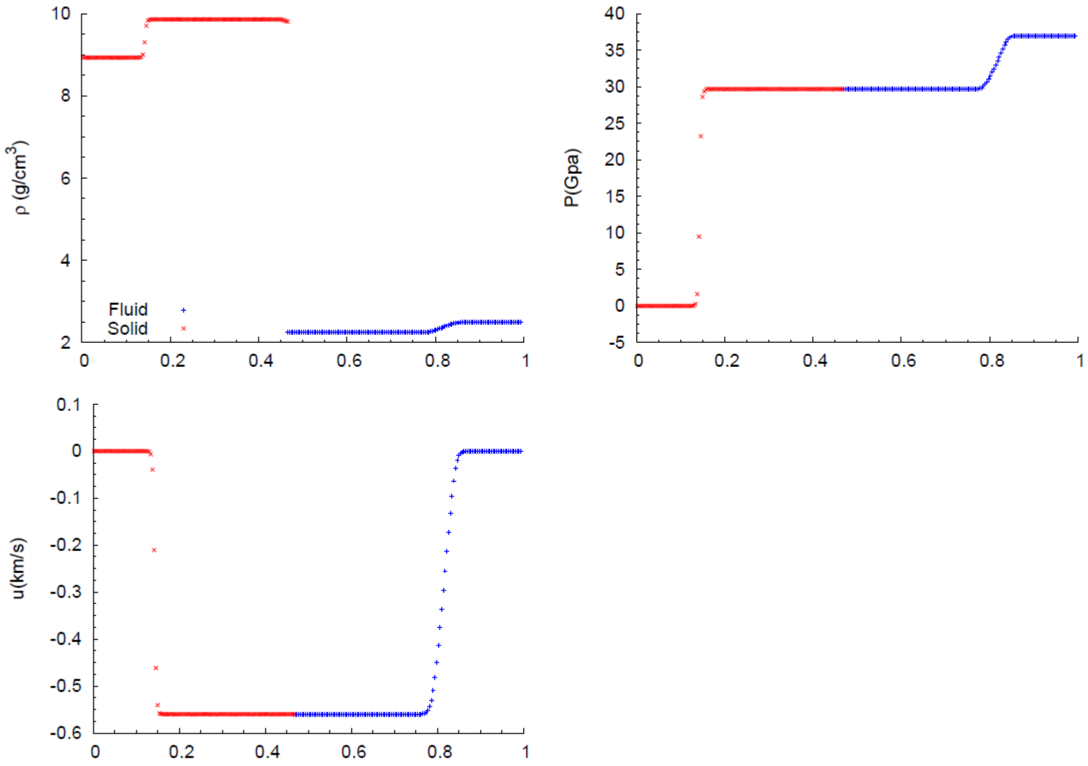


Figure 6.52: TNT Copper IVP, MPWENO-5 and 250 points grid size.

2. 500 points

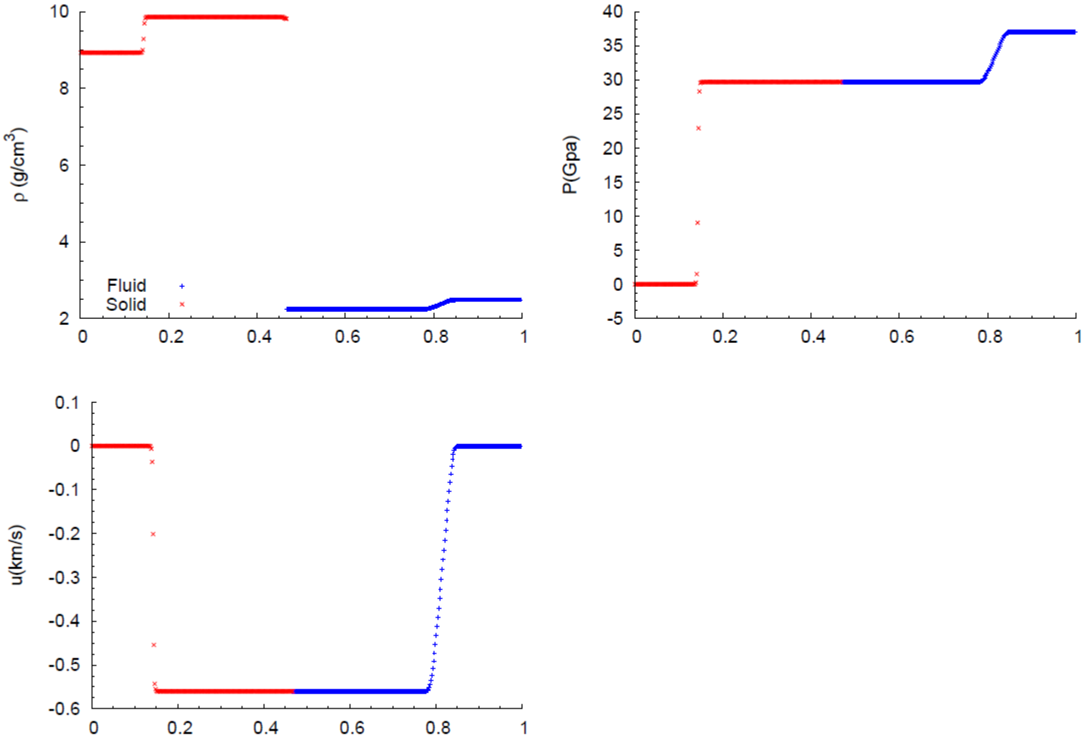


Figure 6.53: TNT Copper IVP, MPWENO-5 and 500 points grid size.

3. 1000 points

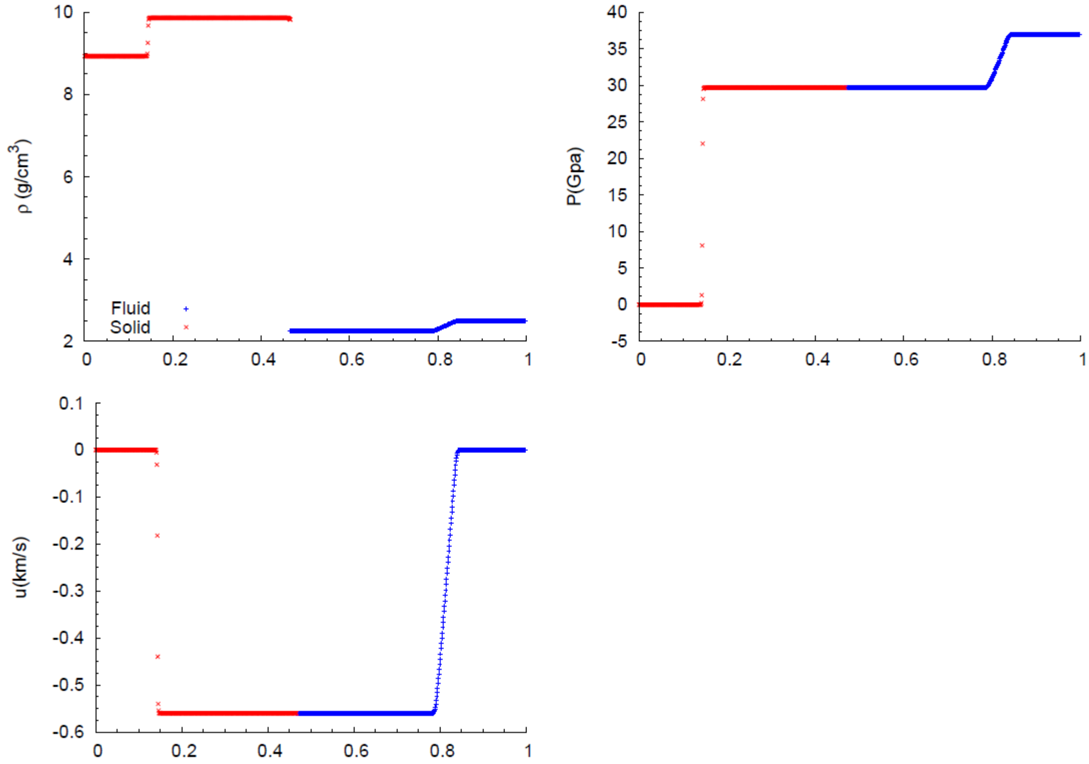


Figure 6.54: TNT Copper IVP, MPWENO-5 and 1000 points grid size.