Chair of Aerodynamics and Fluid Mechanics
Department of Mechanical Engineering
Technical University of Munich

Application and Reformulation of a Compressible Multi-Component Levelset Method for the Simulation of Boundary-dominated Flow Configurations

Anwendung und Anpassung einer kompressiblen Mehrkomponenten Levelset-Methode für die Simulation randbedingungsdominierter Strömungszustände

Scientific work for obtaining the academic degree

Master of Science (M.Sc.)

at the Department of Mechanical Engineering of the Technical University of Munich

Supervisor Dr.-Ing Stefan Adami
Chair of Aerodynamics and Fluid Mechanics

Advisor Nico Fleischmann, M.Sc.
Chair of Aerodynamics and Fluid Mechanics

Submitted by Luis Rafael Piqueras Andreu

Submitted on September 30, 2020 in Garching

Abstract

The ALPACA compressible multiphase solver, developed by the Nanoshock workgroup of the Chair of Aerodynamics and Fluid Mechanics, is employed for flow configurations involving different fluid phases which can with each other at supersonic speeds. It employs a levelset interface approach which allows a conservative treatment of the fluid interface.

The present work deals with the proposal of reformulating and modifying the usage of the levelset interface method in order to allow for the simulation of boundary-dominated flow configurations involving a fluid phase and rigid body boundaries. Therefore, the necessary changes are implemented in the code and the validation of the new method is achieved through the calculation of several cases from literature and through qualitative and quantitative comparison of the obtained results.

Zusammenfassung

Der ALPACA Solver für kompressible Mehrphasenströmungen ist von der Nanoshock-Arbeitsgruppe des Lehrstuhls für Aerodynamik und Strömungsmechanik entwickelt worden. Er wird für Strömungszustände mit verschiedenen Fluidphasen angewandt, die miteinander interagieren. Darüber hinaus können ÜBerschallphänomene, wie z.B. Verdichtungstöße oder laufende Wellen, simuliert werden. Die Levelset-Methode für die Phasengrenze erlaubt eine konservative Behandlung der Grenze zwischen den verschiedenen Fluiden.

Die vorhandene Arbeit beschäftigt sich mit der Umformulierung und Veränderung der Nutzung der Levelset-Phasengrenzmethode, um die Berechnung randbedingungsdominierter Strömungzustände zu ermöglichen, die sowohl eine Fluidphase als auch eine Festkörperrandbedingung umfassen. Danach werden die nötigen Veränderungen in den Code implementiert. Die Validierung der neuen Methode erfolgt durch die Berechnung mehrerer Strömungszustände aus verschiedenen Literaturquellen und durch quantitativen und qualitativen Vergleich der erhaltenen Ergebnisse.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem statement

The following work deals with the implementation of a levelset interface method for compressible multicomponent flow configurations and its reformulation for different cases involving solid bodies as boundaries rather than a pure multiphase fluid problem.

For this project, the ALPACA solver, which is in active development by the Nanoshock workgroup of the Chair of Aerodynamics and Fluid Mechanics will be expanded and its behavior will be analyzed in order to ensure a proper implementation of the new feature.

This extension was proposed withing the workgroup itself, as the ALPACA solver was not capable of solving flow configurations involving rigid body boundaries and it presented itself as a useful extension, as this solver has already been used for research involving compressible multiphase flows [3], [9], [10], [16].

1.2 Objectives

The main objective of this project is broadening the scope of the ALPACA solver, so that in can be used for compressible flow calculations involving complex flow boundaries with rigid bodies. Implementation within the available codebase and subsequent validation of such flow configurations are the required steps in order to achieve such objective.

Dividing that broad statement in smaller parts, the different objectives of the present work can be divided as follows:

- 1: Acquiring theoretical knowledge about the employed methods used in the provided solver code. As ALPACA is an existing program, it is necessary to understand the involved theoretical foundations already implemented in the code, so that it is possible to introduce extensions to the solver down the line while taking into account all the previous work done by the workgroup.

- 2: Getting familiar with the actual codebase and calculating some introductory cases. Similarly to the first objective, once the theoretical bases are known, knowing the way the provided code is structured is a must in order to be able to expand it properly in the next steps. The calculation of the first introductory cases will also serve to know the different available options for case setup as an end user.

- 3: Modifying the provided code to implement a standardized solid body definition. This is the main step concerning code extension and reformulation. Those additions to the codebase should allow for the calculation of flow configurations involving rigid body boundaries as a user of the program and performance or stability regressions in the solver should be avoided. This is further discussed in Section 3.2.

- 4: Calculation of the cases of interest for validation of the modified code. Even if the extended code is apparently correct and verified, all numerical simulation software requires extensive validation for further use of its results. As such, a number of reference cases will be simulated and then compared to the available literature to ensure that the extended ALPACA solver can be reliably used for the desired use cases.

1.3 State of the art

The analysis and calculation of compressible flows at high speeds has been studied through different approaches, but the handling of a material interface has proven to be complex [7], either in the case of multi-phase problems or for flow cases involving a rigid body boundary.

Smeared boundaries for multi-component problems, such as the Volume-Of-Fluid method, and a body-fitted grid for complex boundaries are the most common approaches, as indicated in [7]. The level-set interface method employed in the ALPACA solver allows to define a non-smeared boundary which can be applied for both problem types and also attempts to obtain reduced conservation errors.

According to the current usage of the ALPACA solver, the level-set interface technique has been successfully used for different problems involving single-phase and multi-phase flow cases, employing the levelset interface method as proposed by [7].

In detail, the ALPACA solver has already been used for flow configurations involving complex interactions involving shocks, turbulent structures and multiphase interfaces [3]. The numerical schemes available in the code have allowed for very low-dissipation calculations with the introduction of methods to keep symmetry in the calculations.

Other use cases for the ALPACA solver involve simulation of bubble collapse near phase interfaces [9], complex modeling of liquid-solid phase transitions [10] and calculations involving shock-induced droplet breakup [16].

Those fields of study are already a matter of interest for practical applications as, for example, bubble breakup plays a significant role in some treatments of modern medicine [9] and droplet breakup may be used for internal combustion engines or supersonic aircraft aerodynamics [16]. Despite that, the implementation for complex boundaries involving rigid bodies has not been completely implemented yet in the program at hand and only an approximation though a multi-phase calculation has been achieved, which requires further work in order to implement the level-set technique for all flow cases it had originally been proposed.

Some relevant use cases for an implemented rigid body interface would be the analysis of the propagation of shock waves through obstacles or complex arrays of them, as studied in [2], and getting further knowledge about physical phenomena that appear in those flow configurations, like shock-vortex interactions.

Other examples would be capturing flow details like shock bifurcation against an obstacle, as mentioned by Zhang in [18].

Chapter 2

Theoretical foundations

The following chapter details the main theoretical aspects of this work, especially concerning the numerical methods employed in the ALPACA solver. This is of particular interest, as choosing the correct numerical schemes will be important in order to implement the rigid body interface properly and to ensure that the new interface is solved adequately.

2.1 HLLC Riemann Solver

The HLLC Riemann Solver is a modification of the approximate HLL Riemann Solver which focuses on solving the excessive smearing of contact waves present in the HLL Riemann Solver for intermediate waves [13]. This solver was first proposed by Toro, Spruce and Speares [14].

Detailed information of this solver will be given in the following section, as it is used by ALPACA for all the studied flow configurations in this work.

2.1.1 Euler equations

The governing equations of the cases which will be analyzed in the present work are the Euler equations. For a three-dimensional inviscid compressible flow, the Euler equations take the form [3]:

$$\boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x + \boldsymbol{G}(\boldsymbol{U})_y + \boldsymbol{H}(\boldsymbol{U})_z = 0 \tag{2.1}$$

With $\boldsymbol{U}$ being the density of the conserved quantities, such a mass, momentum amb total energy. The fluxes are then defined as given in [3]:

$$\boldsymbol{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{bmatrix}, \quad \boldsymbol{H} = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{bmatrix} \tag{2.2}$$

Then, an equation of state is needed to close the system of equations.

2.1.2 Godunov procedure

As indicated by [13], considering an Initial Boundary Value Problem consisting of:

$$\left.\begin{array}{lll} \text{PDEs} & : & \boldsymbol{U}_t + \boldsymbol{F}(U)_x = \boldsymbol{0} \\ \text{ICs} & : & \boldsymbol{U}(x,0) = \boldsymbol{U}^{(0)}(x) \\ \text{BCs} & : & \boldsymbol{U}(0,t) = \boldsymbol{U}_1(t), \quad \boldsymbol{U}(L,t) = \boldsymbol{U}_r(t) \end{array}\right\} \tag{2.3}$$

An explicit formula can be used to solve the system of equations for $0 \leq x \leq L$:

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n - \frac{\Delta t}{\Delta x}\left[\boldsymbol{F}_{i+\frac{1}{2}} - \boldsymbol{F}_{i-\frac{1}{2}}\right] \tag{2.4}$$

Defining $\boldsymbol{F}_{i+\frac{1}{2}}$ as the Godunov intercell numerical flux, whose definition is:

$$\boldsymbol{F}_{i+\frac{1}{2}} = \boldsymbol{F}(\boldsymbol{U}_{i+\frac{1}{2}}(0)) \tag{2.5}$$

And $\boldsymbol{U}_{i+\frac{1}{2}}$ is the exact similarity solution of the Riemann problem evaluated at $x/t = 0$.
Then, considering the conserved variables and the fluxes of the Riemann problem for the three-dimensional Euler equations:

$$\boldsymbol{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ u(E+p) \end{bmatrix} \tag{2.6}$$

As such, the initial data for each side of the Riemann problem at $x/t = 0$ for the Godunov flux is considered to be:

$$\boldsymbol{W}_L = \begin{bmatrix} \rho_L \\ u_L \\ v_L \\ w_L \\ p_L \end{bmatrix}, \quad \boldsymbol{W}_R = \begin{bmatrix} \rho_R \\ u_R \\ v_R \\ w_R \\ p_R \end{bmatrix} \tag{2.7}$$

Finally, the objective of the Riemann solvers presented below will be obtaining an approximation to the flux $F_{i+\frac{1}{2}}$ in order to solve the Riemann problem.

Figure 2.1 illustrates the temporal and spatial solution of a linear Riemann problem, showing that the number of different states and waves is predetermined by the size m of the system of equations. This will prove important when introducing the approximate Riemann solvers in the next section.

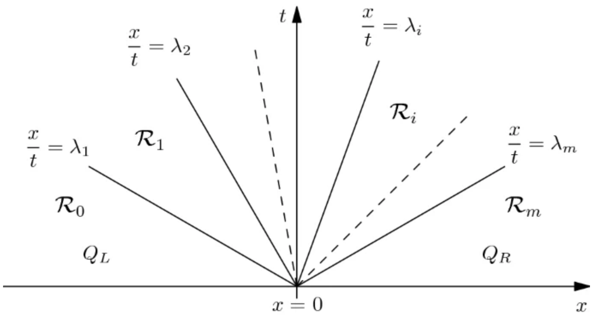
t
$\frac{x}{t} = \lambda_2$
$\frac{x}{t} = \lambda_i$
$\frac{x}{t} = \lambda_1$
$\mathcal{R}_1$
$\mathcal{R}_i$
$\frac{x}{t} = \lambda_m$
$\mathcal{R}_0$
$\mathcal{R}_m$
Q_L
Q_R
$x = 0$
x

Figure 2.1: Wave pattern corresponding to a general linear Riemann problem. [12]

2.1.3 HLLC Scheme

Before describing the HLLC Riemann Solver, it is helpful to first introduce the HLL Riemann Solver, which was proposed by Harten, Lax and van Leer [4] and [6].

The structure of this approximate Riemann solver is as follows:

$$\tilde{U}(x,t) = \begin{cases} \boldsymbol{U}_L & if \ \frac{x}{t} \leq S_L, \\ \boldsymbol{U}^{hll} & if \ S_L \leq \frac{x}{t} \leq S_R, \\ \boldsymbol{U}_R & if \ \frac{x}{t} \geq S_R, \end{cases} \tag{2.8}$$

Considering $\boldsymbol{U}^{hll} = \frac{S_R \boldsymbol{U}_R - S_L \boldsymbol{U}_L + F_L - F_R}{S_R - S_L}$. S_R and S_L are the fastest signal velocities which cause a perturbation in the initial data states (as described in [13]), and are assumed to be known beforehand.

This constitutes an approximate Riemann solver, as the problem has been reduced to only three different constant states which are separated by two waves. The effects of this simplification can be visualized in Figure 2.2 when compared to the more complex figure 2.1.

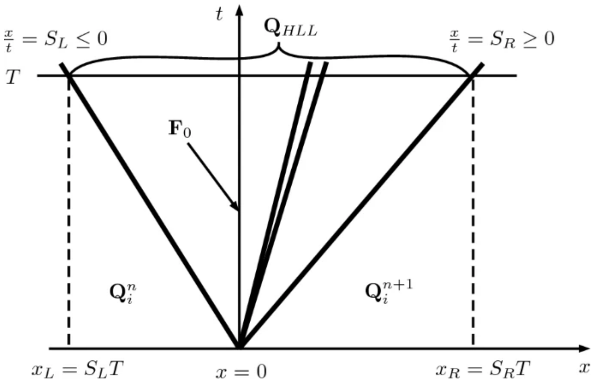

Figure 2.2: Wave pattern corresponding a Riemann problem solved with HLL. [12]

Then, the corresponding flux $\boldsymbol{F}^{hll}$ can be calculated through:

$$\boldsymbol{F}^{hll}_{i+\frac{1}{2}} = \begin{cases} \boldsymbol{F}_L & if \ 0 \leq S_L, \\ \frac{S_R \boldsymbol{F}_L - S_L \boldsymbol{F}_R + S_L S_R (\boldsymbol{U}_R - \boldsymbol{U}_L)}{S_R - S_L} & if \ S_L \leq 0 \leq S_R, \\ \boldsymbol{F}_R & if \ 0 \geq S_R, \end{cases} \tag{2.9}$$

The only necessary tool remaining is a way of computing S_R and S_L, which then allow for an approximate Godunov method that will converge to the weak solution provided by the conservation laws. However, this HLL Riemann solver present a deficiency in the handling of material interfaces, contact discontinuities or other kinds of intermediate waves [13].

This deficiency can be overcome through the restoration of missing waves (as indicated above, this scheme only considers the presence of two waves). As such, the HLLC approximate Riemann solver was proposed by Toro, Spruce and Speares [14], which recovers all inherent waves of the underlying Euler governing equations.

Once the HLL scheme has been properly explained and its shortcomings have been identified, it is now possible to introduce the HLLC Riemann solver which is the one used throughout all the numerical simulations of the present work.

The main difference resides in the fact that the two previously used waves, known by their signal speeds S_L and S_R, are now complemented with a middle wave of speed S^* [13]. Therefore, all the integrands between S_L and S_R can now be split to include the existence of S^*. As such, the following relation holds true:

$$\left(\frac{S^* - S_L}{S_R - S_L}\right) \boldsymbol{U}_{*L} + \left(\frac{S^R - S^*}{S_R - S_L}\right) \boldsymbol{U}_{*R} = \boldsymbol{U}^{hll} \tag{2.10}$$

As such, the definitions for the state vectors and the respective fluxes in the HLLC Riemann solver are similar to the HLL ones, but splitting the middle solution in order to include the aforementioned third wave.

$$\tilde{U}(x,t) = \begin{cases} \boldsymbol{U}_L & if \ \frac{x}{t} \leq S_L, \\ \boldsymbol{U}_{*L} & if \ S_L \leq \frac{x}{t} \leq S_*, \\ \boldsymbol{U}_{*R} & if \ S_* \leq \frac{x}{t} \geq S_R, \\ \boldsymbol{U}_R & if \ \frac{x}{t} \geq S_R, \end{cases} \tag{2.11}$$

The resulting flux is then:

$$\boldsymbol{F}^{hllc}_{i+\frac{1}{2}} = \begin{cases} \boldsymbol{F}_L & if \ 0 \leq S_L, \\ \boldsymbol{F}_{*L} & if \ S_L \leq 0 \leq S_*, \\ \boldsymbol{F}_{*R} & if \ S_* \leq 0 \leq S_R, \\ \boldsymbol{F}_R & if \ 0 \geq S_R, \end{cases} \tag{2.12}$$

However, the new intermediate fluxes F_{*L} and F_{*R} are still unknown. They are determined through the use of a control volume as indicated in figure 2.3. This figure also visualizes the difference between the HLL and the HLLC solvers, as the further division of the central domain and the appearance of an additional wave can be easily distinguished.

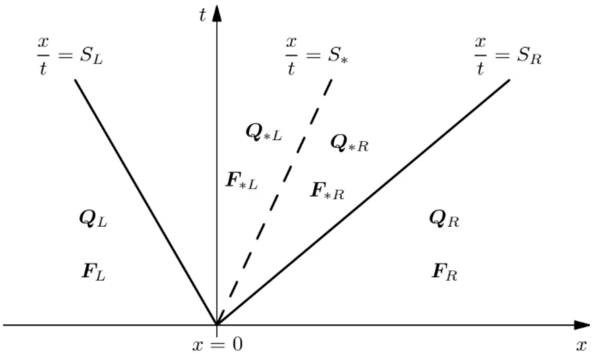

Figure 2.3: Wave pattern corresponding a Riemann problem solved with HLLC. [12]

The results of the intermediate fluxes are described as follows, as given by [13]:

$$\boldsymbol{F}_{*L} = \boldsymbol{F}_L + S_L(\boldsymbol{U}_{*L} - \boldsymbol{U}_L), \tag{2.13}$$

$$\boldsymbol{F}_{*R} = \boldsymbol{F}_R + S_R(\boldsymbol{U}_{*R} - \boldsymbol{U}_R), \tag{2.14}$$

Through the procedure explained above and other implementation details beyond the scope of this work, it should be noted that HLLC Riemann solver is already implemented in ALPACA to allow the calculation of compressible fluid problems.

2.2 WENO Numerical Scheme

The ALPACA solver employed in the current work applies a WENO5 (Weighted Essentially Non-Oscillatory) scheme for reconstruction of the states in the Riemann solver. The particular details of the implementation of the method are given in [8].

As indicated in [8], WENO schemes originate from the ENO schemes introduced previously by Harten [5], however, the WENO spatial reconstruction schemes will be presented directly in the following paragraphs.

For a given conservation law:

$$u_t + f(u_x) = 0 \tag{2.15}$$

The WENO schemes discretize the spatial derivatives of the equation, considering that the domain is divided into intervals Δx, the discretization is:

$$L = -\frac{1}{\Delta x}\left(\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}\right) \tag{2.16}$$

With $\hat{f}_{j+\frac{1}{2}}$ being the numerical flux. Usual ENO schemes approximates the flux by choosing the "smoothest" stencil from a list of r candidates, depending on the order of the method (in the present case, WENO5 indicates a fifth order WENO scheme).

Then, considering any chosen stencil of the form $S_k = \left(x_{j+k-r+1}, x_{j+k-r+2}, ..., x_{j+k}\right)$, the resulting approximation for the flux employing an ENO scheme is [8]:

$$\hat{f}_{j+\frac{1}{2}} = q_k^r\left(f_{j+k-r+1}, ..., f_{j+k}\right) \tag{2.17}$$

With the definition of g_r^k being:

$$q_r^k\left(g_0, ..., g_{r-1}\right) = \sum_{l=0}^{r-1} a_{k,l}^r g_l \tag{2.18}$$

This general ENO scheme is advantageous, as the criterion for choosing the "smoothest" stencil directly removes undesirable contributions from discontinuous stencil candidates, which would lead to oscillations when higher order stencils are applied. Despite that, it is not optimal for continuous regions, as the combination of several continuous stencils can provide a higher order solution than the one provided by just the "smoothest" stencil. Therefore, the WENO scheme comes into play by assigning each candidate S_k a weight ω_k, as indicated in [8], and ensuring that discontinuous stencils have a zero weight assigned. An example where this zero weight would be necessary is near shocks or other discontinuities in the flow.

Then, the approximation of the flow through the WENO scheme is:

$$\hat{f}_{j+\frac{1}{2}} = \sum_{k=0}^{r-1} \omega_k q_k^r\left(f_{j+k-r+1}, ..., f_{j+k}\right) \tag{2.19}$$

With the particular value of each weight ω_k chosen accordingly.

As such, the employed WENO5 scheme is a high-order discretization method with generally beneficial properties for the numerical simulations [8], as they are esentially non-oscillatory and are smooth and $r+1$-order accurate (compared to a r-order ENO scheme). Also, as indicated in the concluding remarks of [8], they are robust for shock calculations and produce only negligible overshoots that are usually not problematic.

2.3 Levelset Interface Method

The ALPACA solver employed in all calculations of the present work relies on a levelset interface method to define the interfaces between each fluid phase for a given case definition.

As indicated by [7], there are several different approaches to study the dynamics of fluids at high speeds. However, the treatment of a material interface has proven difficult for numerical simulations. There are two different kinds of interface-related problems. On the one hand, there are the multi-fluid cases where the separation between each fluid is represented by the interface. On the other hand, there are complex boundary problems where the interface represents the boundary between the fluid and a solid body.

2.3.1 General considerations

As shown by the capabilities already present in the ALPACA solver and the particular intent of this work, the levelset interface method can be generalized for any inviscid compressible fluid problem involving any particular conservation equations [7], such as:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F} = 0. \tag{2.20}$$

Considering $\boldsymbol{U}$ as any conserved quantity and $\boldsymbol{F}$ as the respective fluxes.

Additionally, the domain Ω can be split by an interface $\Lambda(t)$ into two-subdomains ($\Omega^1(t)$ and $\Omega^2(t)$). As indicated in [7], this then allows the solution of multiphase problems and the evolution of the interface can be calculated through a Riemann problem.

Then, the treatment of the interface can be different depending on whether two fluids are involved or a rigid complex boundary is present.

If two fluids are involved, the Riemann problem is described as [7]:

$$R(\boldsymbol{U}_{fluid1}, \boldsymbol{U}_{fluid2}) = 0 \quad on \quad \Gamma(t) \tag{2.21}$$

If a fluid and a solid are involved, the one-sided Riemann problem is then [7]:

$$R(\boldsymbol{U}_{fluid}, \boldsymbol{v}_r g) = 0 \quad on \quad \Gamma(t) \tag{2.22}$$

2.3.2 Definition of the interface

The main particularity of the levelset approach is that the interface is defined by the so-called levelset function, as specified by [7].

That levelset function is described as a signed distance function $\phi(x, y, t)$ with the property $\nabla|\phi| = 1$, such that the physical position of the interface dividing the domain into two subdomains is determined by the value zero of the levelset function ϕ. Therefore, the region of the domain corresponding to positive values of the signed distance function will belong to one subdomain, whereas the other subdomain will be defined by the region where the levelset value is negative.

The value of the levelset function is then advected in time using the following advection equation, as shown in [7]:

$$\phi_t + u\phi_x + v\phi_y = 0 \tag{2.23}$$

However, as indicated by Min [11], the advection procedure can distort the slope of the levelset function close to the interface and a reinitialization that adequately replaces the levelset function is desirable.

The reinitialization of the levelset after each step is accomplished through a signed distance function. The signed distance property is afterwards restores by a reinitialization process after each time integration. Further details about the applied method can be found in [11]. For the present work, only the relevant parts that are necessary for the reformulation in the code, detailed in Chapter 3, are given.

2.4 Min Levelset Reconstructor

The employed ALPACA solver provides several different methods to reconstruct the levelset interface after each iteration. For the cases of interest of the present work, the reinitializer of Min [11] provides the best results, as it keeps the position of the rigid body interface constant throughout the calculation and is able to accept and preserve sharp edges (as it will be shown in the results chapter of the work).

2.4.1 Application for the levelset function

In particular, the levelset reconstructor of Min [11] is employed in the nearfield of the interface, whereas no further treatment is necessary for the levelset function far from the interface. The need for reinitialization of the levelset is also mentioned in [7].

The definition for the equation that serves the purpose of reinitializing the levelset function is initially considered to be the following Eikonal equation:

$$\begin{cases} |\nabla(\phi)| & = 1 \\ sgn(\phi) & = sgn(\phi^0) \end{cases} \tag{2.24}$$

However, Min indicates in [11] that the following time-dependant Eikonal equation is more efficient as a levelset reconstructor, as levelset definitions between subsequent timesteps are bound to be very similar to each other. As such, the levelset reinitializer equation is:

$$\begin{cases} \phi_t + sgn(\phi^0)(|\nabla\phi| - 1) & = 0 \\ \phi(x,0) & = \phi^0(x) \end{cases} \tag{2.25}$$

This function serves then to advect the signed distance value and the reinitialized levelset function can now be used for further computations as part of a multiphase solver.

Chapter 3

Code Implementations

The following chapter explains the employed approach which allows for the levelset interface method to be modified in order to simulate a rigid body boundary. First, a prior analysis will be performed. Its results then motivate the changes done to the code, which are described afterwards.

Additionally, the detailed steps of the code modification, including new material classes, will also be explained and discussed for the purpose of understanding how the finalized code can be used by an end user of the program.

3.1 Analysis of Relevant Parameters

The main factor in a stationary solid body boundary is defined by the fact that the interface does not move, i.e., the interface velocity should be zero for all cases.

Therefore, the relationships which define the interface velocity in the implemented code are studied concerning the main variables involved in the calculation of such velocity.

3.1.1 Code analysis

In the ALPACA source code, the file involved with the calculation of the interface velocity is listed under `/src/interface_interaction/interface_riemann_solver/` in the file `linearized_interface_riemann_solver.cpp`. The resulting interface velocity is defined as:

$$v_{int} = \frac{Z_l \cdot v_{n,l} + Z_r \cdot v_{n,r} + p_l + p_r + \Delta p}{Z_l + Z_r} \tag{3.1}$$

Some cleanup has been done to present the above equation, as numerical precision concerns are not an issue for the current analysis. Z_l and Z_r are defined as the impedances of the fluids on the left and the right sides of the interface, respectively. v_{int}, $v_{n,l}$ and $v_{n,r}$ are the resulting interface velocity and the normal velocities of the fluids on both sides of the interface. Lastly, p_l, p_r and Δp are the pressures of the left and right fluids and a pressure differential due to surface tension, which is not relevant in the present study.

With the fluid impedance being defined as:

$$Z = \rho \cdot c \tag{3.2}$$

With ρ being the fluid density and c being the speed of sound in the fluid, it is now possible to rewrite the equation 3.2 in the following manner:

$$\begin{aligned}
Z &= \rho \cdot c \\
Z &= \rho \cdot \sqrt{\gamma R T} \\
Z &= \rho \cdot \sqrt{\gamma \frac{p}{\rho}} \\
Z &= \sqrt{\gamma \rho p}
\end{aligned} \tag{3.3}$$

Once the equation has been rewritten in order to show the relevant parameters, it can also be established from equation 3.1 that the objective is achieving a minimum interface velocity. Considering the right fluid as the future rigid body boundary for the sake of simplicity, the relevant relationship to minimize interface velocity is having a material with a very high impedance Z_r, which will increase the denominator of the v_{int} term, while having a very small $v_{n,r}$ to cancel out the same term in the numerator.

As shown through the transformations in equation 3.3, very high values of impedance can potentially be obtained by increasing the density and the pressure of the phase which will represent the solid as much as possible. However, arbitrarily increasing the pressure of a fluid is inadvisable, as it would lead to unacceptably small time steps. In addition to that, the minimal $v_{n,r}$ can be indicated easily in the case definition through its inputfile. This ensures that, for values of the denominator tending to infinity,the interface velocity will then tend to zero.

The analysis conducted above allows for a first approximation to treating one phase of the simulation as a solid boundary through the modification of the material properties. However, a more robust approach is needed to accomplish predictable and consistent results in the subsequent simulations. This can be carried out with an appropriate code implementation which will then turn the rigid body calculation into a process which is transparent to the user and which can be used only through changes in the case definition within its corresponding inputfile.

3.2 Implementation details

The next necessary step for the adequate application of a rigid body boundary is modifying the provided ALPACA source code to implement the changes and new definitions for the new rigid body interface. This is achieved through the definition of a new material class, called *StiffenedSolid* to respect the naming convention of other material classes found in the provided codebase.

Further particular information about the implementation will be given below, as additional changes are necessary in order to ensure that the solver is correctly employing the added methods for the computation of new solutions.

3.2.1 Material definition

As indicated above, a new material class with the name *StiffenedSolid* is created within the source code of the program. This new material is defined in the directory `/src/materials/` through their corresponding *C++* header and source files, as all other already present materials have been defined.

For the sake of code consistency, the aforementioned files have a nearly identical structured as other material definition files, for example, the *StiffenedGasSafe* one. The `.h` header file includes all relevant methods of the *StiffenedSolid* class, including defining it as a material and the methods to define some additional properties, such as the adiabatic index γ or the background pressure. Then, the `.cpp` source file includes the definition of the methods mentioned in the header file in the same manner as the other materials already available in the solver.

In addition to that, a new file has been added in the directory `/src/user_specifications/`. This new file receives the name `solid_setup.h` and serves as a new class including the internal initialization values needed for the *StiffenedSolid* material. This implementation as a class allows for further future extension, as different initialization values might be useful for other uses of the program. However, only the *Standard* initialization is currently included.

Description	Symbol	Value
Density	ρ	5.0e10
Pressure	p	1.0
X Velocity	u	0.0
Y Velocity	v	0.0
Z Velocity	w	0.0

Table 3.1: Internal properties for initialization of a *StiffenedSolid* material.

Table 3.1 shows the particular values introduced in the `solid_setup.h` file. As shown in the previous analysis, a very high density impedes the movement of the interface, coupled with zero velocity values. The high pressure values mentioned in the previous analysis are not apparent in this initial configuration, as it is specified through the background pressure parameter directly in the solid material implementation with a value of $1.0e6$.

3.2.2 Class implementation

Once the new *StiffenedSolid* class has been defined, the next step is ensuring that the solver code is able to take advantage of it for the computation of future solutions. As such, other files need to be modified in order for the new class to be completely implemented.

The modular codebase constituting the ALPACA solver software allows for most modifications to be restricted to the `src/materials/` directory, as most external functions are able to work with no explicit mention of the material type. In particular, `material_names.h` and `material_manager.cpp` experienced the most changes.

As indicated in its instructions, the `material_names.h` file allows for each material to have a unique identifier to ensure the correct treatment of each one. The *StiffenedSolid* material has been assigned the 600-range identifiers. Additionally, the modifications needed in `material_manager.cpp` are fairly straightforward, as they are just extensions to add the new material in an analogous manner to the other ones.

Beyond the directory used for the definitions of the materials, some small modifications were done elsewhere in the code. For example, under `/src/input_output/`, the source file `inputfile_parser.cpp` had to be changed to ensure that the inputfiles accepted and used the new *StiffenedSolid* material type. The practical usage of this change is discussed in the following section.

3.2.3 Code usage

Once all the code has been properly implemented and the ALPACA program has been extended in order to accept a material with such properties that a rigid body interface can be modeled, the end user of the program should be able to make use of those new capabilities in a way that is similar to other configurations already present in the software.

During compile time, some options can be configured in `/src/user_specifications/`. The files `riemann_solver_settings.h` and `numerical_setup.h` should specify HLLC as the Riemann solver and Min as the levelset reinitializer, as only this particular solver configuration has been extensively tested throughout this work and proven to be acceptably reliable.

As for the runtime, the user of the code only needs to specify the use of a *StiffenedSolid* for one of the phases of the calculation. In order to remain consistency between the fluid and solid specifications, the complete initial parameters and material properties are still present in the inputfile description, but the user should only write in "dummy" values which will we ignored, as the settings for the solid phase are determined internally.

Chapter 4

Results and validation

This chapter focuses on the obtained results with the implemented code. For their validation, the cases are selected from reference literature and the obtained results are compared. The numerical setup of each case will be detailed separately.

Unless otherwise stated, all figures shown in the following sections are obtained through Paraview with a schlieren-type postprocessing that eases the visualization of the changes in density throughout the simulation domain.

4.1 Forward-facing step

The forward-facing step is a typical flow configuration related to channel flows. This particular case is derived from the reference source [1]. It proves to be a useful point of validation and comparison, as [1] also uses an HLLC Riemann solver. In addition to that, the sharp edges present in this flow configuration are a potential boundary feature which can be difficult to properly define as a rigid body boundary employing the existing levelset definition of the ALPACA solver. The same configuration is also part of the work produced by Woodward in [17] and, therefore, it also proves to be an interesting point of comparison.

4.1.1 Setup

The setup of this case, including boundary and initial conditions, is chosen analogous to that presented in [1] and [17], the particular conditions being further detailed in the latter one. No concrete units of measurement are given in this case, so all magnitudes will be referred as *units* in this case.

Then, the properties for the gas and the initial conditions are chosen as given in Table 4.1. In this case, all initial conditions (including the Mach number) apply to the whole domain.

Description	Symbol	Value
Adiabatic index	γ	1.4
Initial density	ρ	1.4
Initial pressure	p	1.0
Mach number	Ma	3.0

Table 4.1: Setup for the forward-facing step.

Concerning the physical dimensions of the computation domain, the channel is considered to be 3 units long and 1 unit high. In addition to that, the forward-facing step is 0.2 units high and it is located 0.6

units away from the left boundary and it spans the whole remaining length of the channel. The resulting geometry is shown in Figure 4.1.

The left boundary is an inlet that supplies flow with the same state as the initial condition throughout the domain and the right boundary is an outlet. All other domain boundaries are considered as slip walls through the use of a symmetry condition.

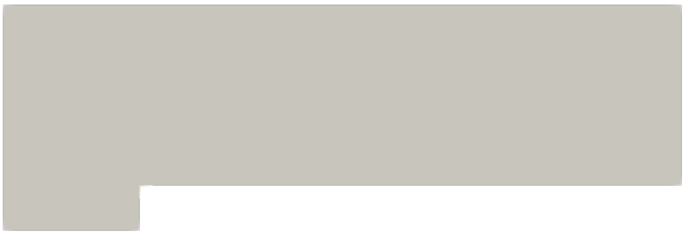

Figure 4.1: Initial state of the forward-facing step domain.

With respect to the internal ALPACA solver configurations, the step is modeled trough the *StiffenedSolid* class implemented in the present work with a Courant-Friedrichs-Levy Number of $CFL = 0.4$, a value that provided numerical stability through the full simulation time. Concerning the employed mesh resolution, the domain is divided in 30 blocks in X-direction and 10 blocks in Y-direction, with a multi-resolution parameter of 1 in the inputfile definition. Each block has an original resolution of $20x20$ cells.

4.1.2 Results

Figure 4.2 shows the immediate appearance of a bow shock slightly in front of the step that propagates through the channel. Subsequent figures, such as 4.3 and 4.4, show how the initial bow shock starts reflecting after reaching both ends of the channel and an apparently steady pattern of reflected shock waves is present for higher times, such as in figures 4.5 and 4.6, however, [17] indicates that the flow at $t = 4\,s$ is still not steady and that completely stationary results require simulation times as high as $12\,s$.

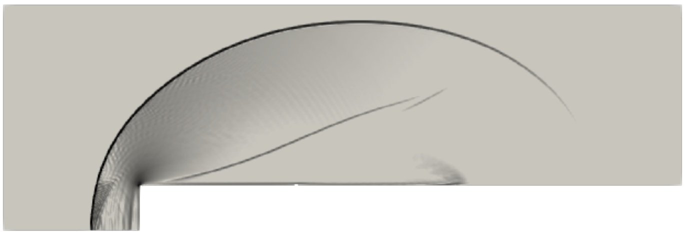

Figure 4.2: Forward-facing step at $t = 0.5s$. Schlieren contours from 3.0 to $3.7e2$.

As indicated by Woodward in [17], the edge of the forward-facing step constitutes the center of a rarefaction fan and is a singular point that might require special treatment. This is due to being a sharp edge boundary that might cause a singularity. However, no apparent spurious results appear as a result of that point and the behavior corresponds with the expected expansion waves.

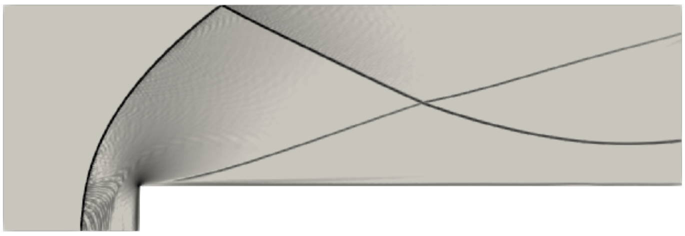

Figure 4.3: Forward-facing step at $t = 1.0s$. Schlieren contours from 3.0 to $3.7e2$.

Once the initial shock waves have been reflected against the surfaces and an oblique shock wave reflection pattern is established after the step. The formation of a complex shock wave interaction between the wave originating from the edge of the step and the reflected wave in the lower part of the channel is apparent in Figures 4.4 to 4.5. This particular phenomenon may be caused by the singular point at the edge of the step, which affects the cells downstream of it. Therefore, this shock pattern is attributed to the influence of the singular point.

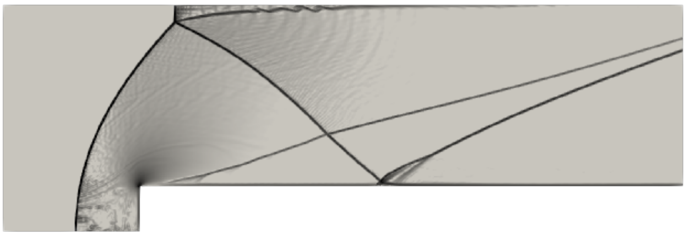

Figure 4.4: Forward-facing step at $t = 2.0s$. Schlieren contours from 3.0 to $3.7e2$.

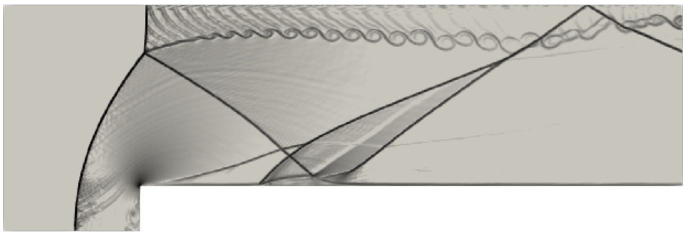

Figure 4.5: Forward-facing step at $t = 3.5s$. Schlieren contours from 3.0 to $3.7e2$.

For higher simulation times, as in figures 4.5 and 4.6, a Kelvin-Helmholtz instability is apparent along the contact line in the upper part of the flow behind the contact point of the front shock waves.

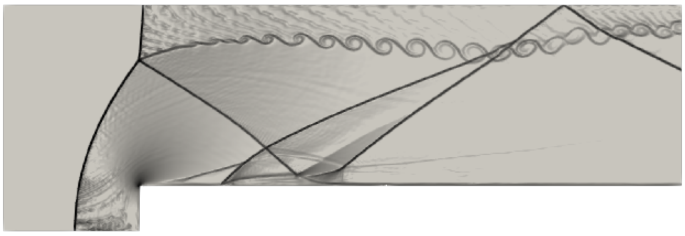

Figure 4.6: Forward-facing step at $t = 4.0s$. Schlieren contours from 3.0 to $3.7e2$.

4.1.3 Comparison

Once the obtained results have been shown, it is now possible to compare them to the reference literature and discuss their validity.

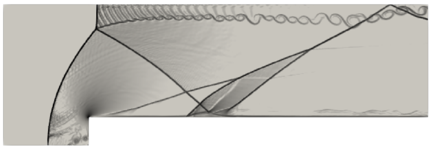

Figure 4.7: Forward-facing step at $t = 3s$ (present). Schlieren contours from 3.0 to $3.7e2$.

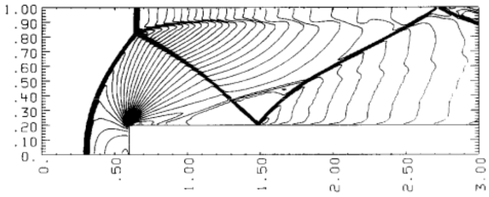

Figure 4.8: Forward-facing step at $t = 3s$ (from [17]).

The above Figures 4.7 and 4.8 show the forward-facing step for $t = 3s$. It is possible to see that the obtained results are fairly similar, for example, the main shock reflection pattern has the same shape. The appearance of an oblique shock that originates from the upper edge of the step and crosses the first reflected wave is also apparent. The Kelvin-Helmholtz instability in the upper part of the channel is also hinted in 4.8, but the simulation through ALPACA shows it more clearly, as it will be discussed later.

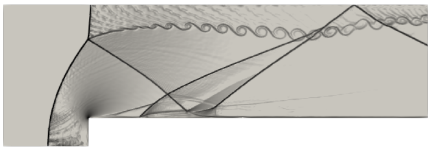

Figure 4.9: Forward-facing step at $t = 4s$ (present). Schlieren contours from 3.0 to $3.7e2$.

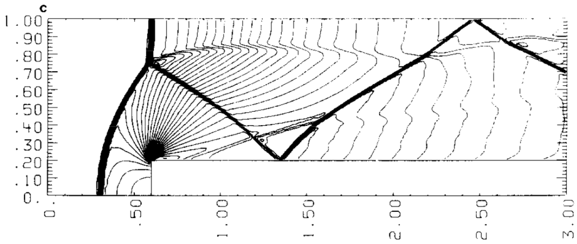

Figure 4.10: Forward-facing step at $t = 4s$ (from [17]).

Figures 4.9 and 4.10 show the same case for $t = 4s$. An equivalent image that is used as the source of comparison can be found in Fig. 7 of [1]. Once again, the reflected wave pattern is very similar. As both ALPACA and [1] use the HLLC Riemann solver, the similarity of both solutions shows that the new implementation of rigid-body boundaries in ALPACA is well behaved. However, the finer details of the solution, such as the other appearing oblique waves or the Kelvin-Helmholtz instability cannot be completely identified in Figure 4.10. The reason for those differences will be discussed below.

As such, it is possible to say that the obtained results show a good correlation with the reference literature and that the employed method for the modeling of rigid body interfaces in ALPACA is adequate for this boundary-dominated case configuration. However, it must be emphasized that the results obtained through ALPACA seem to resolve some particular fluid interactions with a higher definition than the calculations from the reference sources. An important factor lies simply in the fact that both articles [1] and [17] had much smaller computational resources at their disposal due to their age, as such, the calculations with ALPACA are able to use a much finer mesh. The particular differences between the calculations also lie in distinct numerical treatment, such as the usage of a different solver compared to [17] and a finer mesh with higher order discretization schemes compared to [1].

4.2 Muzzle blast

This case is also derived from the reference source [1] and it is also mentioned as a calibration case in [8]. Moreover, studies were conducted by Wang in [15]. The main reasons for the interest in this flow configuration reside in being able to define the rigid body boundary correctly and in applying the Rankine-Hugoniot equations properly for the initial conditions.

4.2.1 Setup

The aforementioned source does not provide the complete set of necessary information about their setup and the employed initial conditions. Despite that, the provided information is sufficient, as they indicate that a weak shock of $Ma = 1.76$ is placed at the right edge of the lower subchannel and the corresponding Rankine-Hugoniot equations are employed for the flow downstream of the shock. The detailed conditions of the flow below the subchannel are given in Table 4.2. As for the rest of the flow, it is assumed to be initially stationary with unit density and unit pressure for the sake of simplicity. It is not possible to exclude minor setup differences, but they should not affect the qualitative validity of the results. No particular units of measurement are given in this case, so a generic "*units*" will be used to refer to magnitudes.

Description	Symbol	Value
Adiabatic index	γ	1.4
Initial density	ρ	2.295
Initial pressure	p	3.447
X Velocity	u	1.175

Table 4.2: Setup for the muzzle blast behind the shock.

Geometrically, the lower subchannel is separated from the rest of the domain through a lip whose length is half of the length of the whole domain, which has a total length of 1.2 units. Then, the lower subchannel is 0.15 units high (the whole domain being 0.6 units high) and the lip has the same height as the lower subchannel.

Concerning the employed mesh resolution, the domain is divided in 6 blocks in X-direction and 3 blocks in Y-direction, with a multi-resolution parameter of 3 in the inputfile definition. Each block has an original resolution of $20x20$ cells.

Considering the boundary conditions, the lower one is a symmetry condition representing a reflective wall, and the upper, left and right ones are zero gradient boundary conditions that allow the shock waves to leave the domain. However, there is one significant difference in the treatment of the problem in the ALPACA solution and in the reference literature. They consider the muzzle to be an axisymmetric problem, but it is treated as a usual 2D case in the ALPACA simulations. This might be a cause for some difference in the solutions.

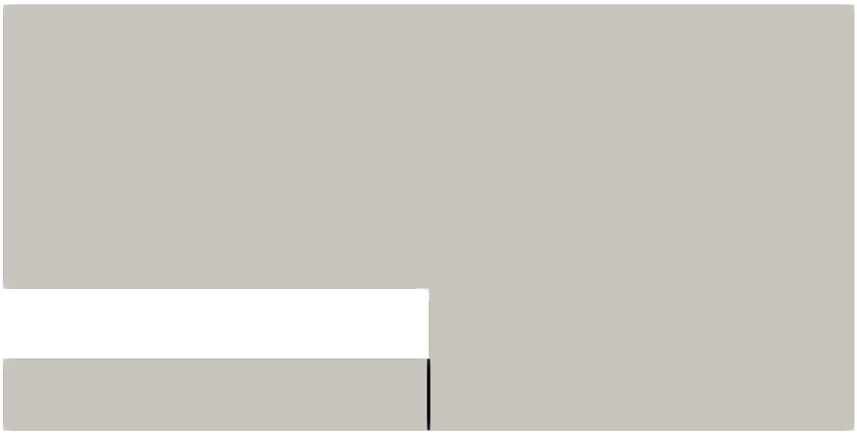

Figure 4.11: Muzzle blast at $t = 0s$.

Figure 4.11 shows the problem geometry and the initial placement of the shock wave below the lip is clearly apparent.

4.2.2 Results

Figures 4.12 through 4.16 show the temporal evolution of the muzzle blast. It can be easily identified that the initial shock beneath the lower subchannel propagates as a deflected forward propagating wave and leaves the computation domain. Some perturbations downstream of the muzzle appear after the initial shock and they also propagate.

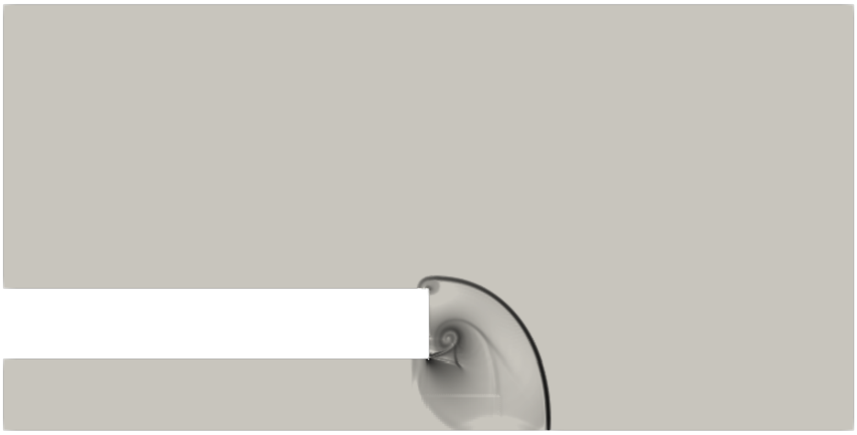

Figure 4.12: Muzzle blast at $t = 0.08s$. Schlieren contours from 3.0 to $4.9e2$.

It can be seen that the initial shock wave is propagating with a nearly circular form and the first perturbations downstream of it start being apparent, as shown in Figure 4.12. A vortex can be easily recognized close to the lower edge of the lip, as was predicted by [1].

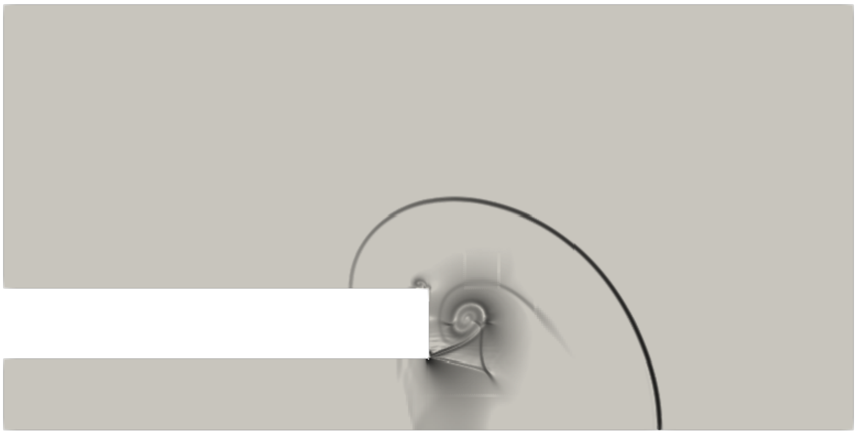

Figure 4.13: Muzzle blast at $t = 0.16s$. Schlieren contours from 3.0 to $4.9e2$.

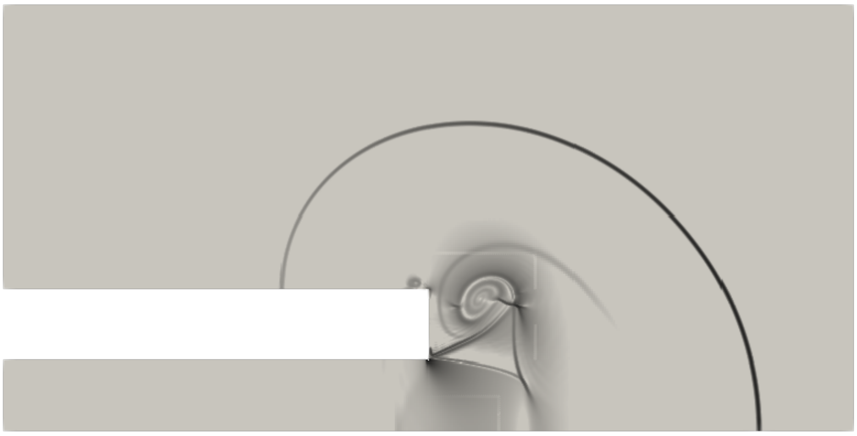

Figure 4.14: Muzzle blast at $t = 0.24s$. Schlieren contours from 3.0 to $4.9e2$.

Then, as shown in Figures 4.13 and 4.14, the initial shock keeps propagating downstream with a nearly circular form and it is able to enclose the whole section, even reaching the upper side of the lip. The previously identified vortex grows further and it is accompanied by another compression shock beneath it (especially apparent in 4.14). A horizontal compression wave also appears and it joins the lower edge of the lip and the vortex.

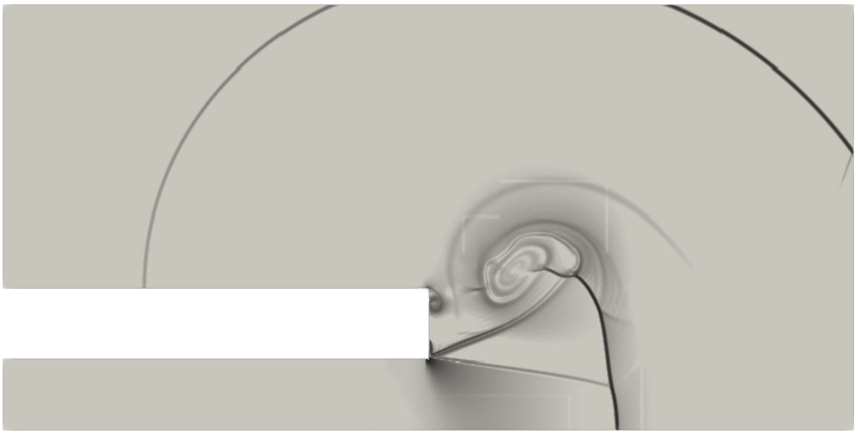

Figure 4.15: Muzzle blast at $t = 0.40s$. Schlieren contours from 3.0 to $4.9e2$.

Figure 4.15 shows further propagation of the initial shock wave and it has partially left the computation domain. The vortex shown initially to the left of the lip keeps expanding as a vortex shock. As indicated by [1], the shocks join forming a triple point downstream of the lower subchannel.

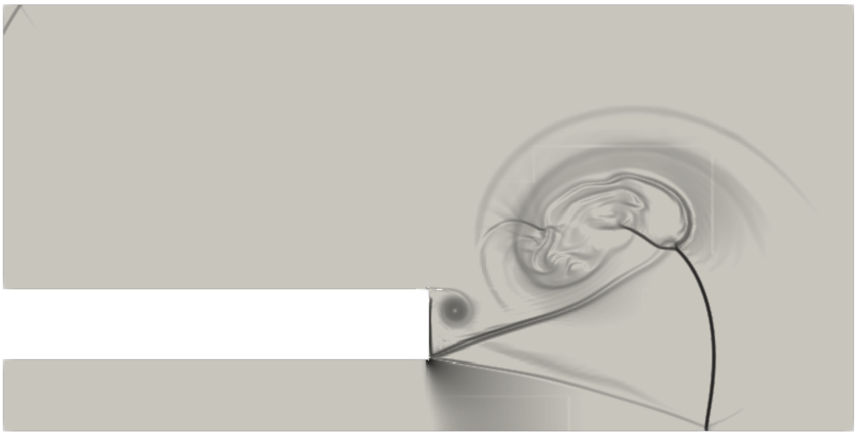

Figure 4.16: Muzzle blast at $t = 0.64s$. Schlieren contours from 3.0 to $4.9e2$.

Finally, the initial shock wave propagates completely beyond the computation domain and leaves it, whereas the formed perturbations behind it keep expanding downstream, as shown in Figure 4.16.

4.2.3 Comparison

Before proceeding to compare the results, it must be noted that, due to the partially incomplete description of the setup, a deviation in the simulation time between the obtained results and that of the reference literature is present. However, the figures that are going to be compared are taken at similar times strictly concerning the occurring physical phenomena, as the particular simulation time is not necessarily a relevant factor that influences the results by themselves.

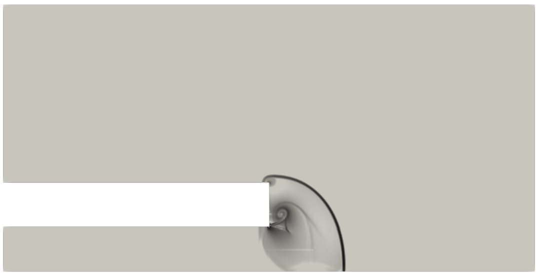

Figure 4.17: Muzzle blast at $t = 0.08s$ (present). Schlieren contours from 3.0 to $4.9e2$.

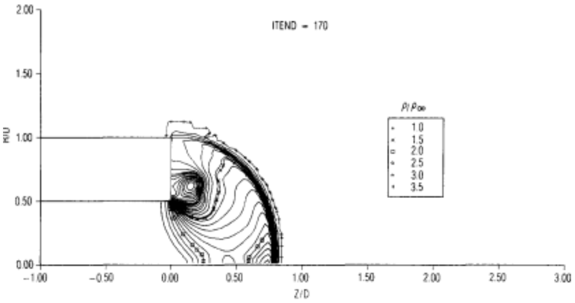

Figure 4.18: Muzzle blast at similar time (from [15]).

In Figures 4.17 and 4.18, the obtained schlieren images look similar, and equivalent phenomena can be identified. The equivalent Figure by Batten in [1] (Fig. 8) is also very similar, but it is not shown to avoid repetition. It can be seen that the initial shock propagates in the same manner, being initially connected to the upper edge of the lip and adopting a round shape while propagating. The initial stages of the vortex and the start of a compression wave are also visible in both figures.

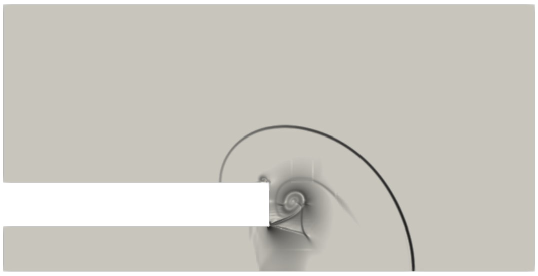

Figure 4.19: Muzzle blast at $t = 0.16s$ (present). Schlieren contours from 3.0 to $4.9e2$.

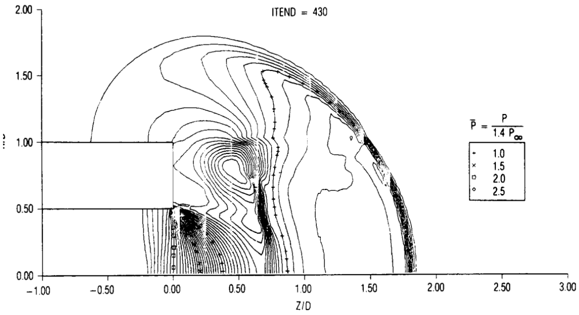

Figure 4.20: Muzzle blast at similar time (from [15]).

Once the initial shock has propagated beyond the lip, it is possible to see that the density contours are still similar in Figures 4.19 and 4.20. An equivalent image that is used as the source of comparison can be found in Fig. 9 of [1]. However, a first difference is apparent between them. In the case calculated by Batten in [1] (Figure 4.20, the initial shock apparently dissipates and does not reach the upper surface, whereas it does in Figure 4.19. This is caused by the much coarser mesh resolution in the upper left part of the computational domain in the case of [1]; this fact is shown in Fig. 9-left of the mentioned article.

As far as the other physical phenomena are concerned, both simulations still show a high correlation, because both the expansion of the vortex and the appearance of the compression shock beneath it are present.

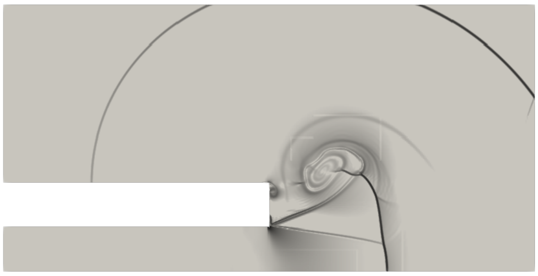

Figure 4.21: Forward-facing step at $t = 4s$ (present). Schlieren contours from 3.0 to $4.9e2$.

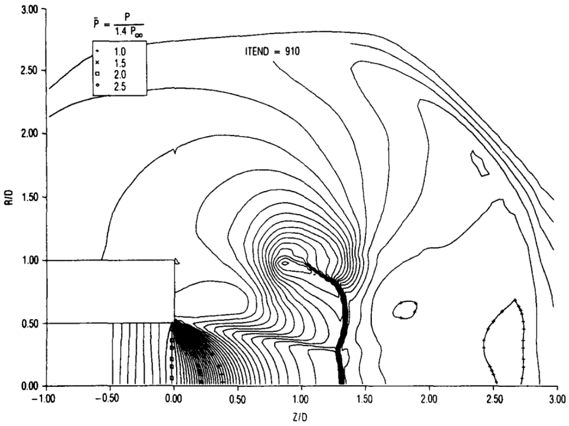

Figure 4.22: Muzzle blast at similar time (from [15]).

Finally, Figures 4.21 and 4.22 show that further evolution still shows a good correlation between the obtained results and the reference sources, but some difference are already apparent. An equivalent image that is used as the source of comparison can be found in Fig. 10 of [1]. In both cases, the initial shock continues expanding and has already partially left the domain, however, the calculations done in [1] still show that the shock is dissipated in the left half of the domain.

As for the particular new shocks formed close to the lip, the way they propagate is quite different. It can be noted that a small new vortex is formed close to the upper edge of the lip in Figure 4.21 (which grows further, as shown in Figure 4.16), but this is not present in Figure 4.22, which can be attributed to the coarser mesh resolution used in its case. Additionally, even though the main vortex shock and the compression waves (including the triple point) are present in both cases, Figure 4.21 shows that those phenomena propagate upwards, whereas in Figure 4.22, they propagate perpendicular to the main axis or even slightly downwards.

This is caused by the fact that the flow configuration is treated as an axisymmetric problem by Batten in [1]. These differences are also present and increased for higher simulation times, as the ALPACA solution keeps propagating upwards, as shown in Figure 4.16.

Once the obtained results have been compared with the reference literature, it is possible to say that the new rigid body implementation is also able to properly resolve this kind of boundary-dominated flow configurations, as the all the relevant appearing physical phenomena have been reproduced and no instability within the rigid body has been found. However, there are some significant deviations compared to the reference articles.

On the one hand, they can be mainly attributed to the fact that the reference case is an axisymmetric calculation, whereas it is treated as a usual 2D problem in the ALAPCA solution. This caused the difference in the propagation of the perturbations behind the initial shock, where they stay much closer to the axis in the reference calculations and it also hinders the analysis slightly, as the *Mach disk* mentioned by [1] cannot be reproduced through a non-axisymmetric 2D calculation, even if the obtained figures are similar at first glance. On the other hand, they are also caused by the different numerical treatment of the discretization and solution, but especially by the much higher mesh resolution employed in the calculations using the ALPACA solver.

4.3 Single triangle obstacle

The next flow configurations, which all involve a single obstacle with a specific geometrical shape, are derived from [2].

This particular case, involving a triangle, can show that the current levelset implementation is able to solve rigid body boundaries involving sharp angles as part of the interface.

4.3.1 Setup

The setup of this case replicates the calculations done by Chaudhuri in [2] as faithfully as possible. If some information about the setup provided by the literature is not directly stated, the dimensions and the orders of magnitude of the source results are analyzed and then applied to the new setup, so that the results obtained by the new calculations stay similar to the source. As the reference source provides concrete SI-units for its calculations, they will also be employed in this case.

A shock wave of Mach number $Ma = 1.4$ is placed in front of the obstacle and downstream of that wave the corresponding conditions are calculated through the Rankine-Hugoniot equations. The particular state of the flow is described in Table 4.3.

Description	Symbol	Value
Adiabatic index [-]	γ	1.4
Initial density [*kg/m*^{3}]	ρ	2.07
Initial pressure [*Pa*]	p	2.15e5
X Velocity [*m/s*]	u	194.45

Table 4.3: Setup for the single triangle obstacle behind the shock.

For the flow that is undisturbed at the beginning of the simulations, standard atmospheric conditions assuming a density $\rho = 1.225\frac{kg}{m^3}$ and a pressure $p = 101325Pa$ are employed.

Geometrically, the channel width is chosen so that there is an effective flow area of $\varepsilon = 0.25$, which corresponds that only 25% of the total channel width is free at the narrowest section and the remaining 75% is occupied by the obstacle. As indicated in [2], the size of the triangle is determined by its side length, which is $8.8mm$.

Concerning the employed mesh resolution, the domain is divided in 40 blocks in X-direction and 5 blocks in Y-direction, with a multi-resolution parameter of 3 in the inputfile definition. Each block has an original resolution of $20x20$ cells. This setup applies for all single obstacle cases.

Concerning the boundary conditions, the upper and lower boundaries are symmetry conditions that represents a reflective wall and the left and right boundaries are zero gradient boundary conditions that allow them to have inflow or outflow.

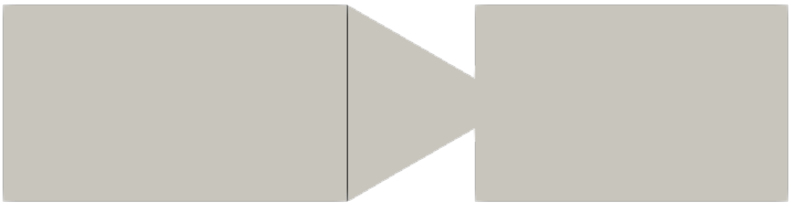

Figure 4.23: Single triangle obstacle at $t = 0s$.

Figure 4.23 shows the problem geometry and the initial placement of the shock wave in front of the obstacle, which will then propagate downstream.

4.3.2 Results

All the presented figures are given with their corresponding output timestep, so that the amount of elapsed simulation time between each one can be more easily compared. The timestep between each output interval is given as $5\mu s$.

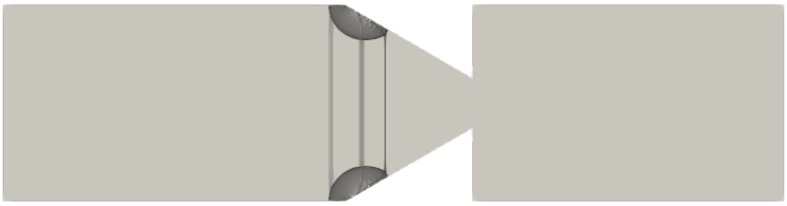

Figure 4.24: Single triangle obstacle (timestep 1). Schlieren contours from 3.0 to $4.6e4$.

Once the shock wave starts to reach the first edge of the triangle, as shown in Figure 4.24, a first pair of reflected shocks (also indicated in [2] as RS) with a circular shape are readily apparent. The initial shock wave continues propagating through the passage (as notes by Chaudhuri in [2] as the transmitted shock TS). Other two compression waves can be seen behind the initial shock wave, the left one propagating upstream, whereas the right one propagates downstream.

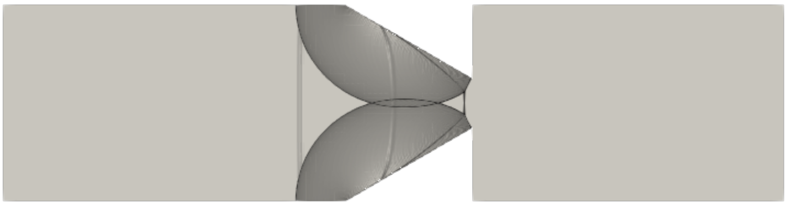

Figure 4.25: Single triangle obstacle (timestep 3). Schlieren contours from 3.0 to $4.6e4$.

After that, it can be seen in Figure 4.25 the reflected shocks keep growing in size and the initial shock wave keeps propagating forward and getting shorter in order to continue through the narrowing passage. The initial shock wave is no longer directly attached to the obstacle walls, but it now has a normal component attached to the edges of the circular reflected shocks and an oblique component located completely within the region of the circular reflected shocks. The point of connection of all those waves forms a triple point.

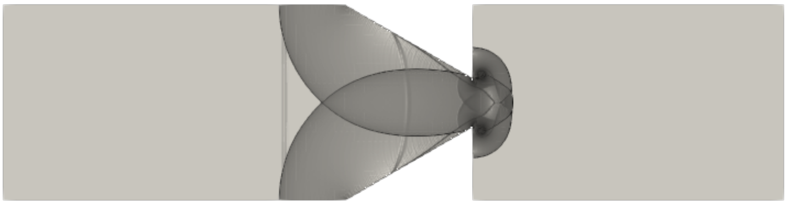

Figure 4.26: Single triangle obstacle (timestep 4). Schlieren contours from 3.0 to $4.6e4$.

Figure 4.26 shows the state of the flow once the initial shock wave (or transmitted shock) has already reached beyond the minimum free section caused by the obstacle. A complex wave reflection pattern starts to form close the rear edges of the triangle and the initial shock now adopts a semicircular shock form, rather than its initial normal shock form. Additionally, two vortices can be seen next to the rear edges of the triangle.

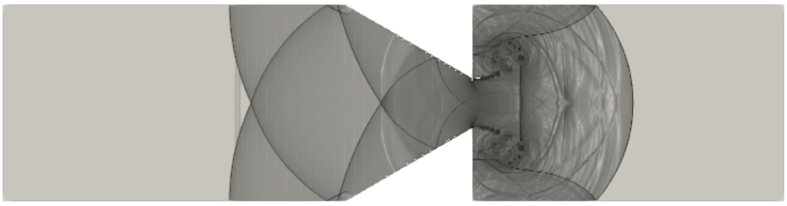

Figure 4.27: Single triangle obstacle (timestep 7). Schlieren contours from 3.0 to $4.6e4$.

The complex wave reflection pattern continues evolving in Figure 4.27. Wave interactions and reflections are also apparent in front of the obstacle, as the original reflected shocks keep propagating upstream. Downstream of the triangle, the initial shock wave keeps moving forward and the waves shown in Figure 4.26 are also propagating and being reflected. The mentioned vortices have moved downstream as well, and a normal shock wave is now located between them (referred in [2] as secondary shock SS).

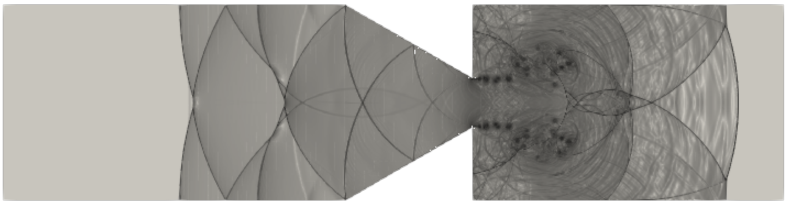

Figure 4.28: Single triangle obstacle (timestep 10). Schlieren contours from 3.0 to $4.6e4$.

Finally, Figure 4.28 shows the state of the flow for a higher elapsed simulation time. The aforementioned secondary shock is no longer immediately apparent and both the upstream and downstream shock wave interactions keep propagating. A trail of vortices that originates from the rear edges of the triangle at the narrowest section is visible.

4.3.3 Comparison

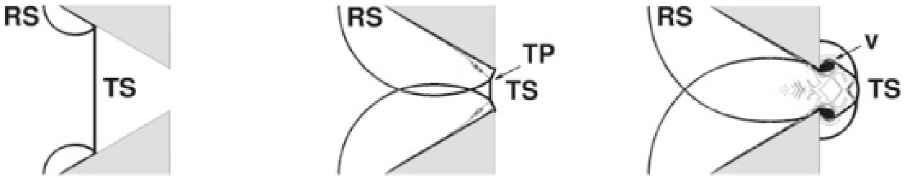

Figure 4.29: Initial wave diffraction for single triangle (from [2]).

Figure 4.29 shows the temporal evolution of the problem as calculated in [2]. From left to right, they correspond to Figures 4.24, 4.25 and 4.26, respectively. It is possible to see that the physical phenomena marked in Figure 4.29 are also reproduced in the results obtained through the new rigid body boundary method of the ALPACA solver. The only apparent difference is in Figure 4.24, where two additional normal waves are apparent. This can be attributed to the higher resolution of the ALPACA solution, whereas the figure from the reference literature is a simplified illustration that could discard physical phenomena not relevant for the analysis.

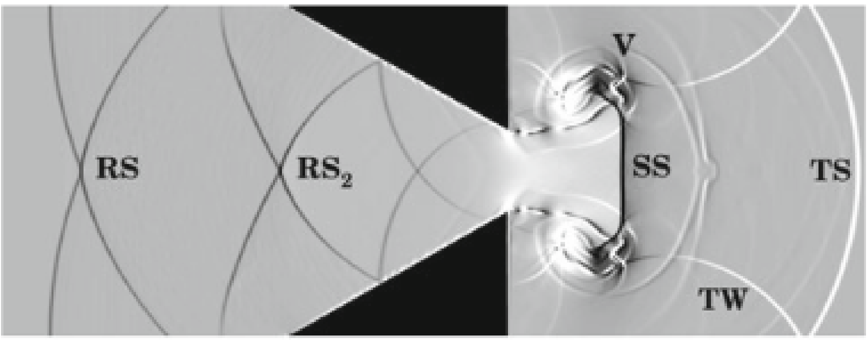

Figure 4.30: Numerical schlieren for single triangle at later stage (from [2]).

Figure 4.30 corresponds to Figure 4.27 of the results section. As with the previous comparison, the same qualitative conclusions can be drawn: All the marked relevant physical phenomena in the figure of the reference article are reproduced and the shock interaction pattern is very similar as well. Again, the main differences are caused by the propagation of the two additional initial shock waves. As this is a numerical schlieren picture of the actual simulation conducted in [2], the difference could be attributed to slight differences in the use of the Rankine-Hugoniot equations when placing the initial conditions, which could lead to some slightly different behavior of the initial shock wave. Finally, it is possible to confirm that the new employed method to represent rigid body boundaries can be used for boundary-dominated flow configurations of this kind, where strong boundary-wave interactions are present and complex shock wave interaction patterns need to be captured with a high detail. No abnormal behavior has been caused by the rigid body interface and the small differences could be attributed to factors external to the own solver programming.

4.4 Single cylinder obstacle

As indicated above, this flow configuration is referenced from [2] and is very similar to the case presented in Section 4.3. However, this particular problem involving a cylinder can also be found in the reference literature [18], with some slight differences.

These cases shall show whether the implemented method can properly keep the prescribed rigid body boundary and whether the computed results can be considered valid.

4.4.1 Setup

Unless otherwise stated, the setup of this case is the same as the one presented in Section 4.3.1.

The only relevant difference lies in the fact that the placed obstacle is different, as it is a cylinder in this case, rather than a triangle. For the triangle, the length of the sides was $8.8mm$. For the cylinder, that measure corresponds to its diameter (which is $8.8mm$ long, accordingly). All other initial and boundary conditions are left unchanged.

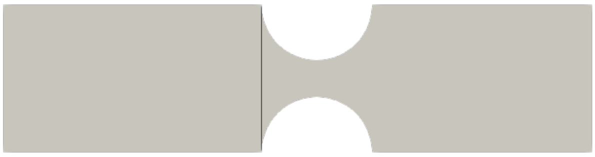

Figure 4.31: Single cylinder obstacle at $t = 0s$.

Figure 4.31 shows the problem geometry and the initial placement of the shock wave in front of the obstacle, which will then propagate downstream.

4.4.2 Results

Similarly to the former case, the presented figures are given with their corresponding output timestep, so that the amount of elapsed simulation time between each one can be more easily compared. The timestep between each output interval is given as $5\mu s$.

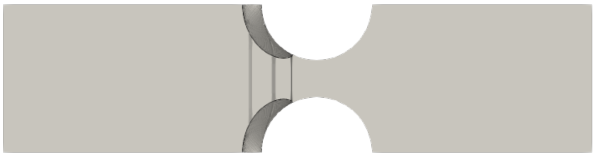

Figure 4.32: Single cylinder obstacle (timestep 1). Schlieren contours from 3.0 to $4.6e4$.

Figure 4.32 shows the evolution of the flow right after the initial shock wave has encountered the obstacle. As such, it is possible to see the beginning of the first reflected wave (as indicated by Chaudhuri in [2]) and the initial shock continues propagating downstream.

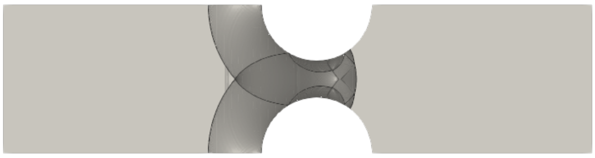

Figure 4.33: Single cylinder obstacle (timestep 3). Schlieren contours from 3.0 to $4.6e4$.

After that, Figure 4.33 shows the further upstream propagation of the reflected shocks. In addition to that, the initial shock wave has progressed beyond the minimum free section of the channel and it is no longer normal, as it starts adopting a semicircular shock form.

In addition to those previously mentioned phenomena, a new kind of wave starts appearing, it is referred in [2] as transverse wave. Those transverse waves appear close to the narrowest section of the channel and are shocks with a rounded form that originate from the cylinder obstacle and start propagating in the transverse direction of the channel, neither upstream nor downstream.

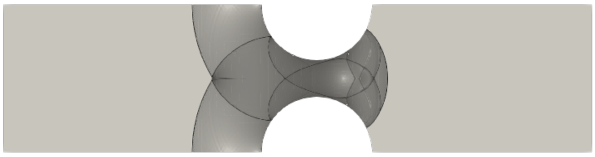

Figure 4.34: Single cylinder obstacle (timestep 4). Schlieren contours from 3.0 to $4.6e4$.

Then, it is possible to see in Figure 4.34 that all waves keep propagating in their respective direction. Considering the transverse waves, even though the main propagation direction of them is perpendicular to the main axis of the channel, they also propagate so that they get wider both upstream and downstream. In particular, Figure 4.34 shows the state of the transverse waves right before they interact with the cylinder obstacle.

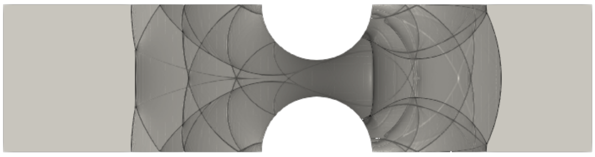

Figure 4.35: Single cylinder obstacle (timestep 8). Schlieren contours from 3.0 to $4.6e4$.

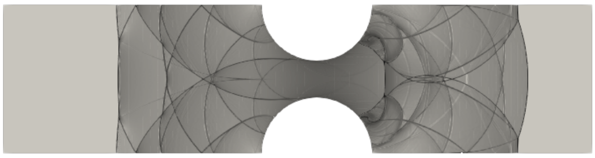

Figure 4.36: Single cylinder obstacle (timestep 9). Schlieren contours from 3.0 to $4.6e4$.

Finally, Figures 4.35 and 4.36 show the flow in a more advanced temporal state. The interaction of all waves results in a complex wave diffraction pattern both upstream and downstream of the obstacle. The transverse wave are no longer clearly identifiable, however, a secondary normal shock has appeared downstream of the obstacle and two vortices are visible at the end of that normal shock. They are especially apparent in Figure 4.36.

4.4.3 Comparison

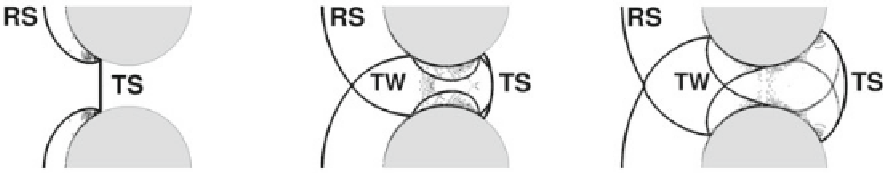

Figure 4.37: Initial wave diffraction for single cylinder (from [2]).

Figure 4.37 shows the temporal evolution of the flow according to reference source [2]. The different stages correspond to Figures 4.32, 4.33 and 4.34, respectively. It is possible to see that all physical phenomena concerning the shock wave interaction marked in Figure 4.37 are clearly present in the results obtained with the ALPACA solver and the rigid body boundary implementation. As such, the correlation between both calculations is quite high.

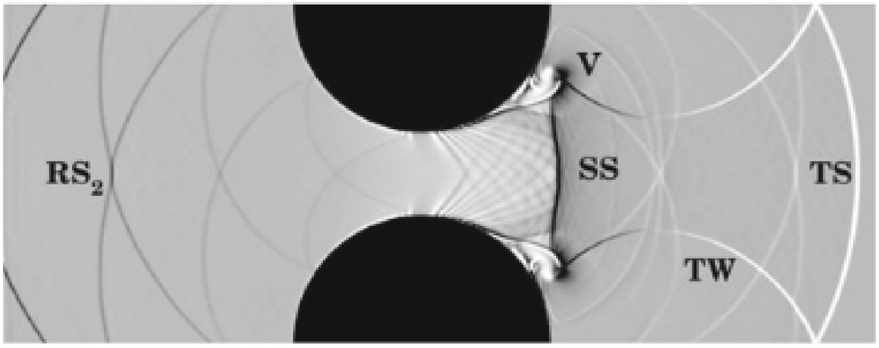

Figure 4.38: Numerical schlieren for single cylinder at later stage (from [2]).

Additionally, Figure 4.38 shows the state of the flow domain for a higher simulation time and should be compared with Figures 4.35 and 4.36. Again, all marked shock waves and the interactions, including the vortices, are present in the calculation conducted with the ALPACA solver. The principal apparent difference is the lack of a defined expansion fan after the narrowest section and before the secondary shock in the ALPACA calculation, however, this might be attributed to a different treatment of the schlieren image, where smaller density differences may appear smeared rather than distinct due to the chosen scale.

Finally, it is possible to say that the new rigid body boundary definition is able to solve this other obstacle configuration while capturing all the relevant physical phenomena concerning the shock waves and their interactions.

4.5 Single rectangle obstacle

As previously stated, this case configuration is sourced from [2], but with a different kind of obstacle placed within the channel.

4.5.1 Setup

As with the cylinder obstacle, the setup of this case is the same as the one presented in Section 4.3.1.

The only change is due to the fact that the placed obstacle is different, as it is a rectangle in this case, rather than a triangle. For the triangle, the length of the sides was $8.8mm$. For the rectangle, that measure corresponds to a side length of $8.8mm$ (therefore, the rectangle is more accurately defined as a square). All other initial and boundary conditions are left unchanged.

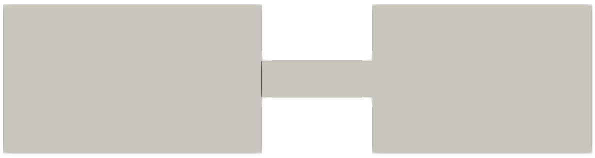

Figure 4.39: Single rectangle obstacle at $t = 0s$.

Figure 4.39 shows the problem geometry and the initial placement of the shock wave in front of the obstacle, which will then propagate downstream.

4.5.2 Results

Like in the former case, the presented figures are given with their corresponding output timestep, so that the amount of elapsed simulation time between each one can be more easily compared. The timestep between each output interval is given as $5\mu s$.

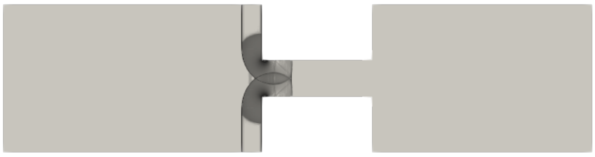

Figure 4.40: Single rectangle obstacle (timestep 1). Schlieren contours from 3.0 to $4.6e4$.

Figure 4.40 displays the state of the flow right after the first interaction of the initial shock wave with the obstacle. The reflected shocks are primarily vertical upstream of the rectangle, but they curve along the edge of said obstacle. Right by that edge, an increase of density is present, which indicates the development of a vortex. The initial shock keeps propagating downstream.

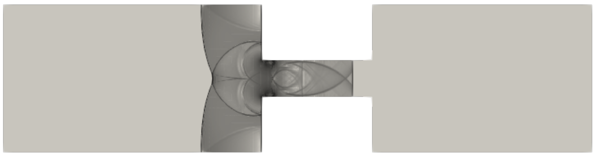

Figure 4.41: Single rectangle obstacle (timestep 3). Schlieren contours from 3.0 to $4.6e4$.

The aforementioned vortices are now more apparent in Figure 4.41 and they are growing in the downstream direction. Meanwhile, the initial shock keeps it normal shock wave form due to the form of the channel created by the rectangle.

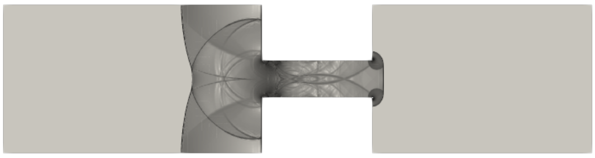

Figure 4.42: Single rectangle obstacle (timestep 4). Schlieren contours from 3.0 to $4.6e4$.

Once the initial shock propagates beyond the narrow channel in Figure 4.42, it starts to adopt the form of a forward propagating wave and two new vortices start forming on its sides. In addition to that, an oblique shock pattern is formed between the two upstream vortices, as indicated in the reference article [2] with the notation secondary shock.

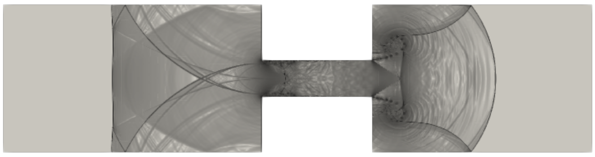

Figure 4.43: Single rectangle obstacle (timestep 8). Schlieren contours from 3.0 to $4.6e4$.

After that, the main difference apparent in Figure 4.43 is the further propagation of the new vortices that appeared first in Figure 4.42 besides the initial shock. In this case, the vortices do not propagate as quickly as the initial shock, but a secondary shock has now formed between them. This means that there are now two secondary shocks in the flow, as the one caused by the two upstream vortices is still present.

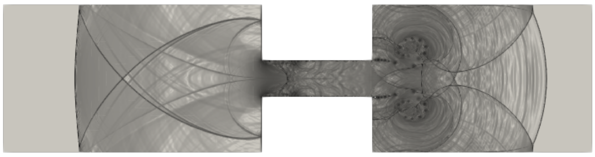

Figure 4.44: Single rectangle obstacle (timestep 10). Schlieren contours from 3.0 to $4.6e4$.

Then, Figure 4.44 shows further evolution of the flow domain. The initial shock and the first reflected shocks keep propagating downstream and upstream, respectively, and the vortices with their related secondary shocks are also propagating further, but they are not as distinct in the schlieren image as they were initially.

4.5.3 Comparison

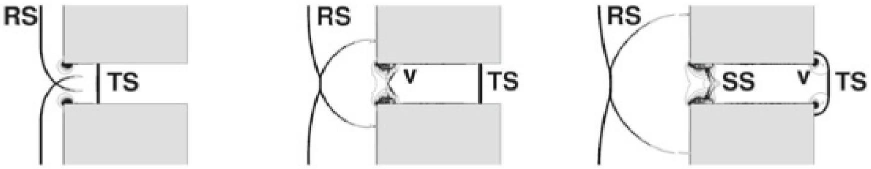

Figure 4.45: Initial wave diffraction for single rectangle (from [2]).

The aforementioned Figures 4.40, 4.41 and 4.26 are to be compared with Figure 4.45 of the reference literature by Chaudhuri [2]. As with the previously analyzed obstacle configurations, all indicated shock waves are present in the solution provided by ALPACA with the new rigid body boundary method.

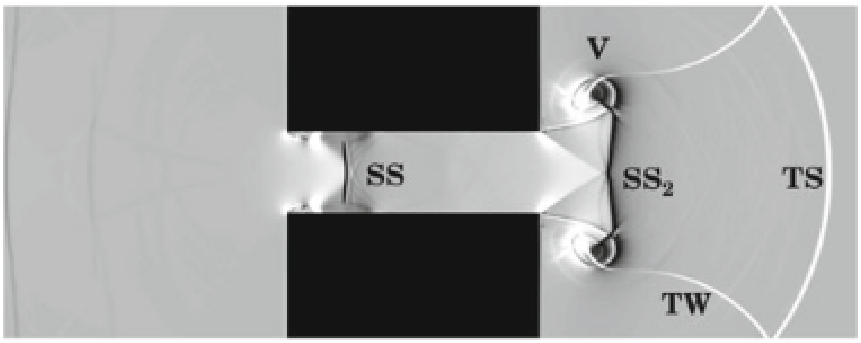

Figure 4.46: Numerical schlieren for single rectangle at later stage (from [2]).

Then, Figure 4.46 shows a similar state of the flow as Figure 4.43, which was obtained with ALPACA. Once again, both results are quantitatively similar and both represent the same physical phenomena.

As a conclusion for this flow configuration, the new rigid body boundary definition is also able to work properly with solid obstacles involving more than one sharp edge, as both the entrance of the channel and the exit involve right angles. The solver is able to calculate and capture the interaction between the solid body and the shock waves with accuracy.

4.6 Single reverse triangle obstacle

This is the last case derived from the single obstacles calculated in [2], in this case, the rectangle is placed rotated 180 degrees compared to the one in Section 4.3.

4.6.1 Setup

As with the former obstacle flow configurations, the setup of this case is the same as the one presented in Section 4.3.1.

The only change is due to the fact that the placed obstacle is rotated 180 degrees compared to the one defined in Section 4.3.1, as such, the flow encounters the base of the triangle first, rather than the corner. All other initial and boundary conditions are left unchanged.

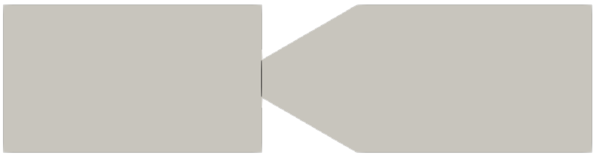

Figure 4.47: Single reverse triangle obstacle at $t = 0s$.

Figure 4.47 shows the problem geometry and the initial placement of the shock wave in front of the obstacle, which will then propagate downstream.

4.6.2 Results

Similarly to the former case, the presented figures are given with their corresponding output timestep, so that the amount of elapsed simulation time between each one can be more easily compared. The timestep between each output interval is given as $5\mu s$.

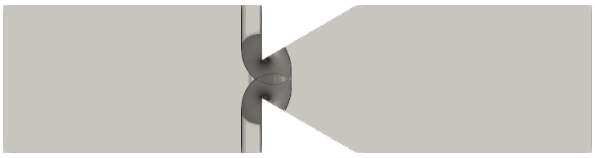

Figure 4.48: Single reverse triangle obstacle (timestep 1). Schlieren contours from 3.0 to $4.6e4$.

For the initial instant of the reverse triangle obstacle configuration shown in Figure 4.48, it should actually be noted that the initial evolution of the flow is quite similar to that obtained with a rectangle in 4.40. Both the reflected shocks which curve along the frontal edge of the obstacle and the start of the vortices behind the edge are present. The main difference is the fact that the initial shock starts curving right after going through the narrowest section of the channel, as it is not uniform like the one present in the case of the rectangle.

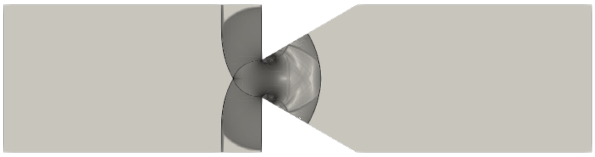

Figure 4.49: Single reverse triangle obstacle (timestep 2). Schlieren contours from 3.0 to $4.6e4$.

Then, Figure 4.41 shows the expected evolution of the flow. The initial shock keeps propagating with the form of a bow shock and the reflected shocks move upstream while the vortices propagate downstream and grow in size.

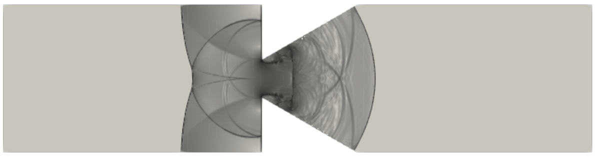

Figure 4.50: Single reverse triangle obstacle (timestep 4). Schlieren contours from 3.0 to $4.6e4$.

For Figure 4.50, the further growth of the vortices results in the appearance of a secondary shock between them, which is clearly distinguishable in this case, as the secondary shock is clearly recognizable as a normal shock wave.

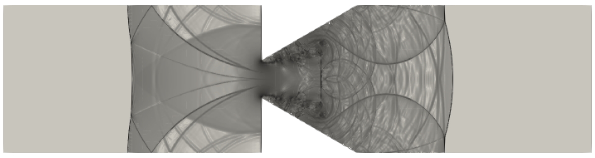

Figure 4.51: Single reverse triangle obstacle (timestep 7). Schlieren contours from 3.0 to $4.6e4$.

The vortices and the secondary shock keep propagating downstream in Figure 4.51. As the vortices move both downstream and away from the center line of the channel, the secondary shock simultaneously grows in size. The initial shock and the upstream reflected shocks are still propagating and there is a growing complex shock interaction pattern behind them. It should be noted that, as this is supersonic flow, that all perturbations are enclosed between the initial wave and the reflected shocks and the flow beyond those points remains undisturbed.

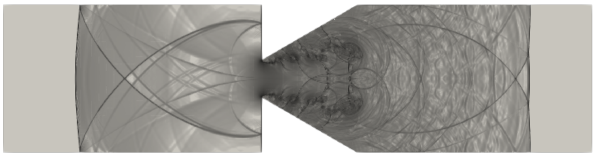

Figure 4.52: Single reverse triangle obstacle (timestep 10). Schlieren contours from 3.0 to $4.6e4$.

For Figure 4.52, the waves keep propagating downstream and upstream and the secondary shock is no longer as apparent, as other shocks are interacting with it. A particular point of interest is the apparent vortex shedding originating from the edges of the minimum free section of the channel, as a trail of vortices is easily identifiable.

4.6.3 Comparison

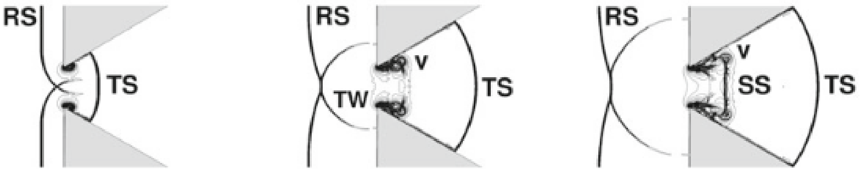

Figure 4.53: Initial wave diffraction for single reverse triangle (from [2]).

Figure 4.53 is to be compared, from left to right, with Figures 4.48, 4.25 and 4.26. As with the previous similar flow configurations, all relevant marked physical phenomena concerning shock waves are captured in the solution obtained by ALPACA.

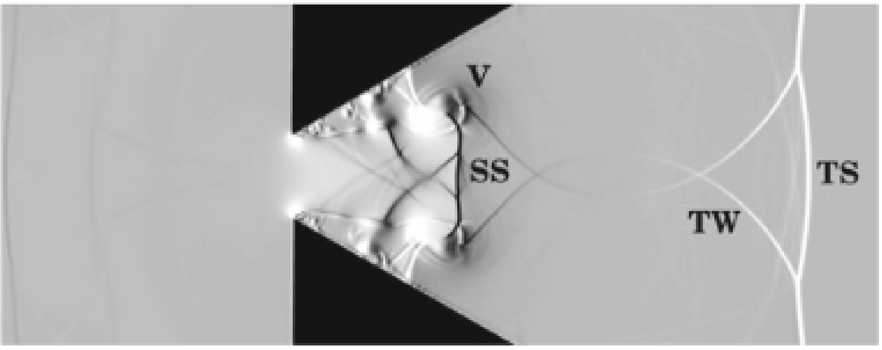

Figure 4.54: Numerical schlieren for single reverse triangle at later stage (from [2]).

Then, Figure 4.54 shows a similar simulation time as Figure 4.51. In this case, the main point of interest is the fact that the secondary shock and its accompanying vortices are properly reproduced. In addition to that, the aforementioned vortex shedding is also present in both cases, which means that this phenomenon has been calculated and captured properly.

Finally, it is possible to say that the ALPACA solver, taking advantage of the new rigid body method, is able to solve all the proposed boundary-dominated flow configurations properly and no undesirable behavior has been produced in the solid boundary interface. In addition to that, it has been shown that the solver is able to reproduce all the relevant shock phenomena.

4.7 Arrays of obstacles

Once all the simple obstacle rows have been solved and analyzed, it is now possible to increase the complexity of the problems and add another row of obstacles to the flow cases.

Those simulations are a selection of flow configurations that also originate from [2] and they will be presented in a summarized manner, as the setup and the expected results are very similar to the already analyzed simple obstacle configurations.

4.7.1 Triangles

This flow configuration is roughly equivalent to the one with only one row of obstacles, as the same initial and boundary conditions are applied. The only differences lie in the addition of more rows of obstacles and the use of a longer simulation time in order to make sure that the initial shock wave reaches the whole array. The mesh resolution configuration is the same as in the previous single obstacle cases, but the multi-resolution parameter has been reduced to 2.

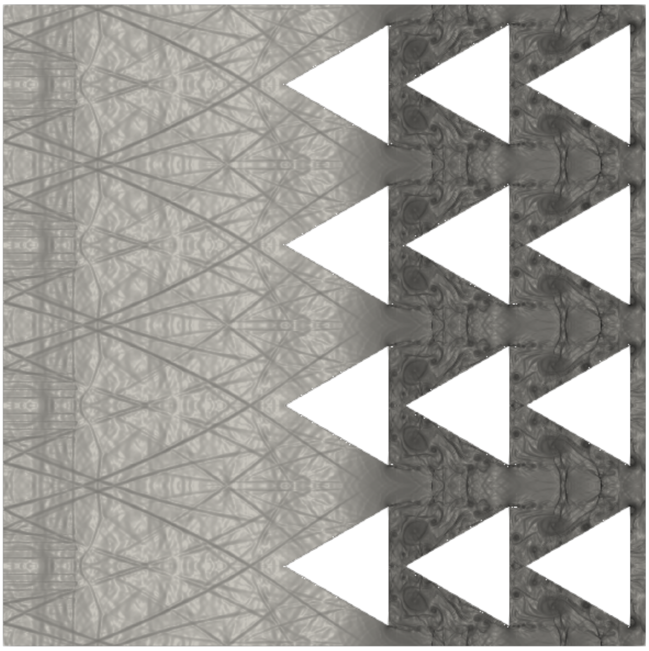

Figure 4.55: Array of triangle obstacles (final timestep). Schlieren contours from 3.0 to $5.6e4$.

Figure 4.55 shows the numerical schlieren results for the array of obstacles. It is possible to see that a complex wave interaction pattern has formed in upstream of the obstacles. Behind the first two rows of obstacles, a similar behavior is found. Big vortices are formed behind each obstacle and smaller vortices are shed from the rear edges of the triangles, which continue after each row and also produce several secondary shocks in a similar manner to the single row obstacle.

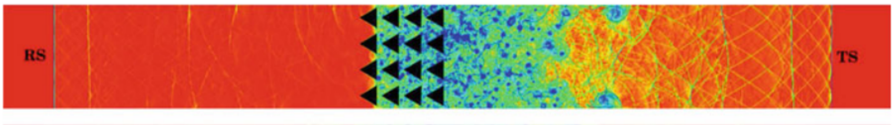

Figure 4.56: Numerical schlieren for array of triangle obstacles (from [2]).

Despite some differences between the results and the geometry shown in Figures 4.56 and 4.55, some approximate comparisons can be drawn. Both the complex shock interaction structure upstream of the array and the appearance of vortices behind the obstacles are present in both calculations. However, there are localized phenomena which appear different and are not immediately explainable, such as the loss of symmetry in the results obtained by [2]. It should be noted that keeping symmetry for the solutions has already been a part of the development process of ALAPCA, as shown by [3].

As such, it is possible to say that the new rigid body definition is able to give reasonable results for this array of obstacles with no unexpected behavior due to the solid boundary, but a quantitative validation would require more extensive testing.

4.7.2 Cylinders

As in the previous case, this new flow configuration is the same as the one with a single row of cylinders, save for the addition of a new row and the total simulation time. The mesh configuration is the same as the one employed the single obstacle flow configurations.

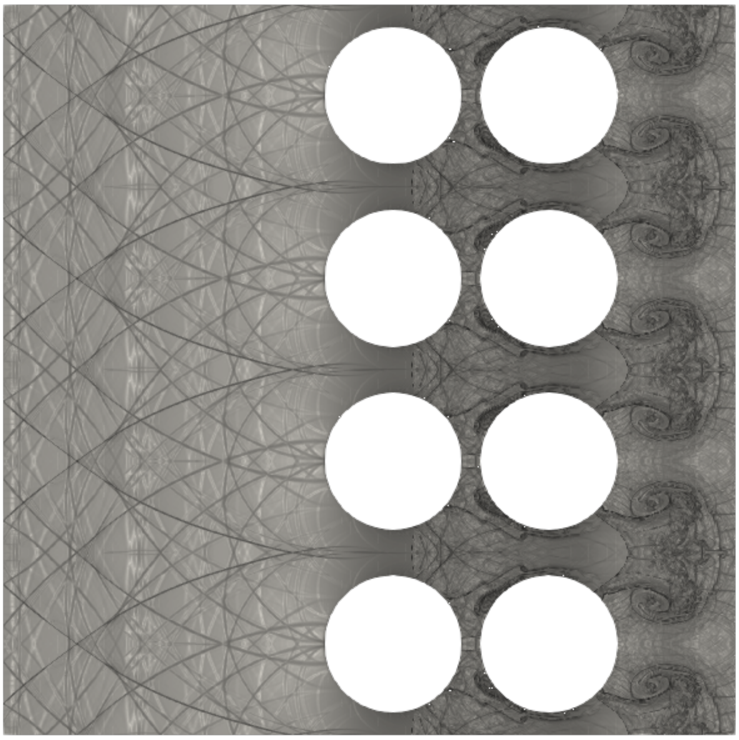

Figure 4.57: Array of cylinder obstacles (final timestep). Schlieren contours from 3.0 to $5.6e4$.

Figure 4.57 shows the existence of another complex wave interaction pattern upstream of the array of obstacles. In addition to that, big vortices with a corresponding secondary shock are also present behind the second row of cylinders. Behind the first row, it is possible to see that the wave pattern behind it is influenced by the presence of the second row, as the shocks appear to be deviated away from the second row of cylinders.

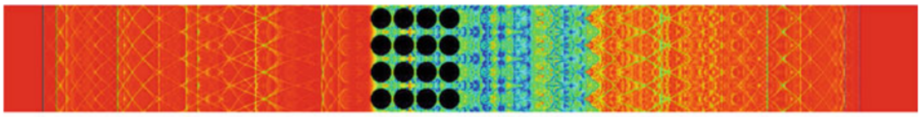

Figure 4.58: Numerical schlieren for array of cylinder obstacles (from [2]).

Using Figure 4.58 to establish a comparison with the results obtained with ALPACA, no obvious conclusions can be drawn other than the fact that symmetry has been preserved in the results obtained by Chaudhuri [2] in this case. In addition to that, the upstream wave pattern appears to be similar.

In conclusion, it is not possible to confirm the validity of the obtained results due to the differences in setup between both cases. However, it has once again been verified that the new rigid body method has been well-behaved for this flow case, and further investigation of this flow configuration is possible with the current implementation.

4.7.3 Reverse triangles

Once again, this case has the same initial setup as the single-row one, other than the addition of a new row of obstacles and a higher simulation time. As with the other arrays, the mesh configuration has been kept equal to the one used for the single obstacle cases.

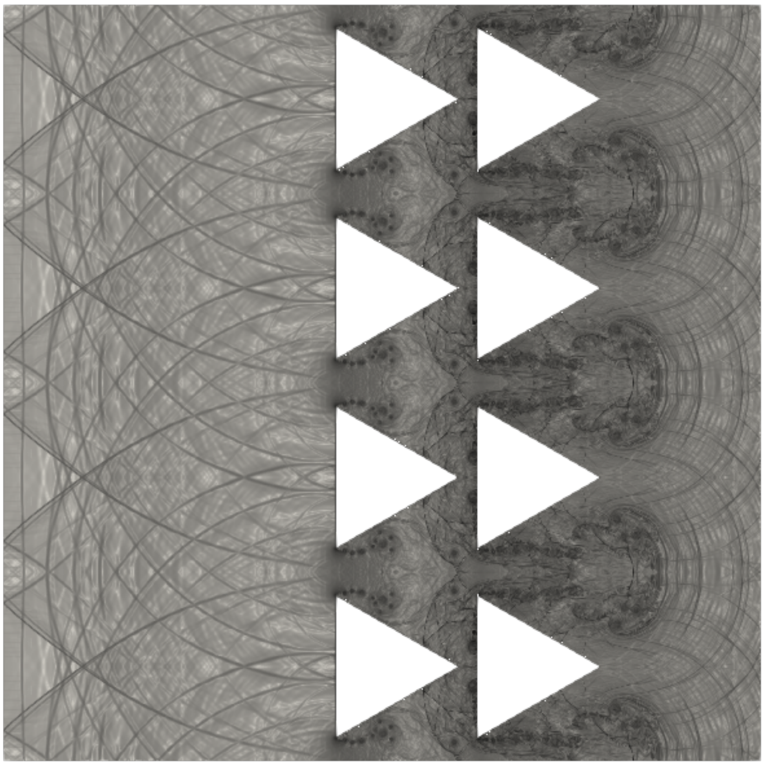

Figure 4.59: Array of reverse triangle obstacles (final timestep).

In this case, Figure 4.59 shows a behavior which can be compared with the aforementioned array of forward-facing triangles in Figure 4.55. The upstream shock interaction of the reverse-facing triangles is quite different as the one present in the other case, which can be explained by the fact that the leading edge of the obstacles is very different (comparing a sharp edge to a wide, flat side). In addition to that, the big vortices between each row of obstacles are not present. However, the vortex shedding from the edges with their corresponding secondary shocks is relatively similar for both cases.

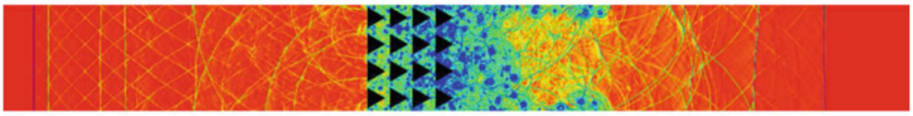

Figure 4.60: Numerical schlieren for array of reverse triangle obstacles (from [2]).

As for the results obtained by Chaudhuri in [2] shown in Figure 4.60, it is again difficult to draw a meaningful comparison. The upstream shock interaction is similar for both cases, as the shocks curve in a similar fashion. However, symmetry is again lost in the results of [2] and the particular interactions between each row of obstacles cannot be directly compared.

Therefore, the obtained results are not able to be validated for this particular flow configuration, even though the solver has behaved well and has shown apparently reasonable shock interactions. In addition to that, the new solid boundary implementation has been able to be implemented for this particular problem and no errors have been introduced because of it, so further investigation for the validation of this case is possible.

4.8 De Laval Nozzle

In addition to the already shown flow configurations, which all originate from a reference article, some additional problems are also studied. These new cases deviate slightly from the typical range of application of ALPACA, and they intend to show whether this solver can potentially be useful for other kinds of problems than those originally thought thanks to the new rigid body implementation.

In this case, a de Laval Nozzle (convergent-divergent) will be analyzed. Only a simple first approximation case will be calculated, as the objective of the current problem is not the full validation of the code, but just showing the possibility of using this solver for further applications.

4.8.1 Setup

The setup of this case is initially not dissimilar to that of a channel flow. Table 4.4 shows the particular initial conditions used for this convergent-divergent nozzle. As with other cases throughout this work, the choice of the density and the pressure are done for the sake of simplicity when determining the Mach number and do not represent any particular realistic condition.

Description	Symbol	Value
Adiabatic index [-]	γ	1.4
Initial density [kg/m^3]	ρ	1.4
Initial pressure [Pa]	p	1.0
Mach number [-]	Ma	0.8

Table 4.4: Setup for the NACA 0012 airfoil.

The domain has been divided in 100 blocks in X-direction and 2 blocks in Y-direction. The multi-resolution parameter has a value of 1. Considering that the block size is chosen as $0.003m$, the simulation domain is $300mm$ long and up $6mm$ wide at the inlet and outlet.

As for the geometry and the boundary conditions, the shape of the nozzle is done through an interpolating polynomial which gives this convergent-divergent nozzle a symmetric section with respect to its throat. The employed polynomial is defined by the zero level of the contour function $\phi(x,y) = -y+0.007+(0.177778\cdot x)\cdot(-0.3+x)$.

This is not usual for de Laval nozzles applied in actual situations, but this choice simplifies the geometry and is adequate for this initial study. The left boundary condition is an inlet which provides flow with the same conditions as the initial flow, whereas the right boudary condition is a zero gradient outlet. The lower boundary is a symmetry plane, which is then used to represent the complete nozzle and the upper part is actually bounded through the solid body interface and represent the wall of the nozzle.

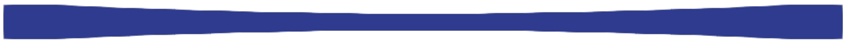

Figure 4.61: Initial state of the de Laval Nozzle.

Figure 4.61 shows the initial geometry of the de Laval Nozzle. The narrowest section and the convergent-divergent geometry can be easily seen.

4.8.2 Results

The obtained results are only intended to show the qualitative tendencies in the simulation, and no concrete numerical results need to be extracted from them.

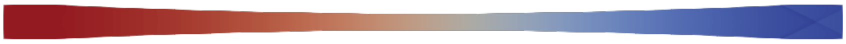

Figure 4.62: Density contours of the de Laval Nozzle. Contour values from $3.2e-1$ (blue) to 2.2 (red).

The density contours shown in Figure 4.62 only provide the information that the flow is steadily expanded throughout the nozzle, which is in line with the expected behavior, as a nozzle is supposed to accelerate and expand a given fluid.

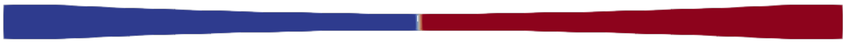

Figure 4.63: Detailed Mach number contours of the de Laval Nozzle. Contour values from 0.99 (blue) to 1.01 (red).

Figure 4.63 shows the Mach contours in the simulation domain, but the scale has been adjusted in a way such that all the gradient in the color scale happens at $Ma = 1$. The fact that the section containing sonic flow is located at the throat of the nozzle (the minimum section) can be seen.

4.8.3 Discussion

The results obtained below are not an exhaustive analysis or validation of a de Laval nozzle using the ALPACA solver and the new rigid body definition, but they show that such a section can be calculated by the solver and the solid body interface is treated properly.

In addition to that, the obtained results are consistent with what is expected for a convergent-divergent nozzle, as the flow is continuously expanded throughout the nozzle and the sonic conditions are reached precisely at the throat section of the nozzle. As such, this new approach serves to expand the scope of the ALPACA solver, as further calculations and analyses can be done to explore the validation of similar flow configurations.

4.9 NACA 0012 Airfoil

The following flow configuration diverges from the other ones which have already been analyzed. Rather than simulating some variations of the channel flow, either implementing obstacles in it or solving other kinds of problems of internal aerodynamics, this uses the ALPACA solver for an atypical flow configuration considered its initial uses.

A NACA 0012 is placed against compressible flow within the transonic region. This case is of particular interest, as it will show qualitatively whether the current solver with the new rigid body boundary configuration can show potentially accurate results for different flow configurations outside its initial scope. As the NACA 4-Series airfoils are well known and its expected behavior is also predictable, this case shall provide a good initial approach to external aerodynamics using the ALPACA solver.

4.9.1 Setup

Even if the studied problem is at first glance significantly different to the other ones, the initial setup and configuration does not differ significantly, as the particular geometry of the rigid body boundary is strictly defined by the definition of the levelset function.

The whole domain is initiated with the same flow conditions, which are defined in Table 4.5.

Description	Symbol	Value
Adiabatic index [-]	γ	1.4
Initial density [kg/m^3]	ρ	1.4
Initial pressure [Pa]	p	1.0
Mach number [-]	Ma	0.8

Table 4.5: Setup for the NACA 0012 airfoil.

It is possible to see that the properties of the employed flow are chosen for the sake of simplicity when defining the Mach number and do not represent any particular atmospheric condition. However, this should not affect the accuracy of the obtained results.

The geometry has been chosen as half of a NACA 0012 airfoil with a total chord length of 2 units. The shape of the airfoil is given through the zero level of the contour function $\phi(x,y) = -1 * (-y + 5 \cdot 0.12 \cdot (0.2969 \cdot sqrt(x-1) - 0.1260 \cdot (x-1) - 0.3516 \cdot (x-1)^2 + 0.2843 \cdot (x-1)^3 - 0.1015 \cdot (x-1)^4))$.

Concerning the mesh resolution, The domain is divided in 30 blocks in X-direction and 10 blocks in Y-direction, with each block having a resolution of $20x20$ cells. The multi-resolution parameter is chosen as 2.

Considering the boundary conditions, the left one is a fixed value condition which represents an inlet of flow with the same properties as the initial condition one. The lower boundary condition is a symmetry plane in order to represent the lower half of the airfoil and both the upper and the right boundary conditions are zero gradient boundaries which allow for the flow to leave the computation domain.

4.9.2 Results

All provided results are given for a long enough simulation time so that steady state flow is achieved, namely, a simulation time of $t = 10s$ has elapsed. As such, the transient starting behavior of the airfoil will not be part of the analysis. As the NACA 0012 airfoil is symmetric, the problem has been treated as such and only the upper surface will be mentioned throughout the analysis in order to avoid repetition.

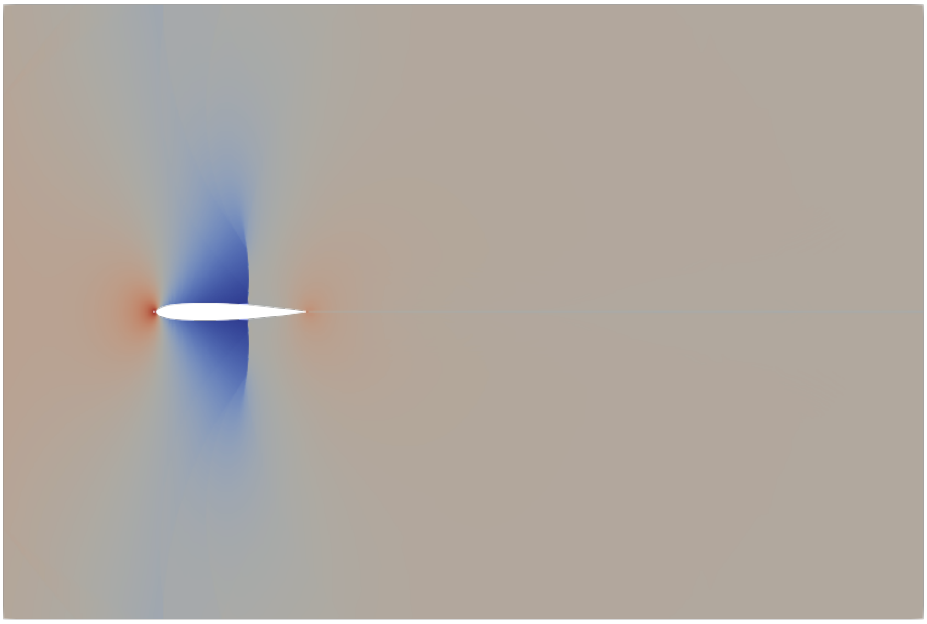

Figure 4.64: Pressure contours for the NACA 0012 airfoil. Contour values from $9e-1$ (blue) to 1.9 (red).

Figure 4.64 shows the behavior of the pressure field around the airfoil. It is possible to see a maximum pressure point at the leading edge of the airfoil, which corresponds to the stagnation point. There is also a drop in pressure along the upper surface of the airfoil until a point at approximately 3/4 of the chord where the pressure suddenly increases. This corresponds to a shock wave on the surface of the airfoil and, as such, it can easily be seen that the airfoil is operating in transonic flow.

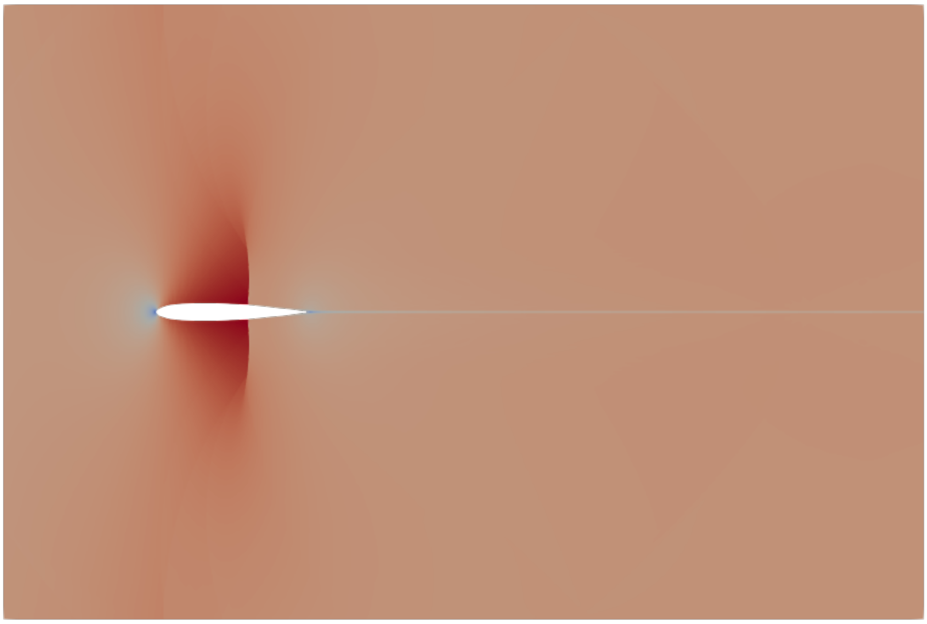

Figure 4.65: Mach contours for the NACA 0012 airfoil. Contour values from $4.1e-2$ (blue) to 1.2 (red).

The analysis of Figure 4.65, which shows the velocity field with the Mach number, is complementary to the pressure contours. In this case, the stagnation point is revealed with a point of zero velocity at the leading edge of the airfoil and the shock wave on top of the airfoil is revealed through the sudden drop in velocity on the airfoil, as the velocity reaches supersonic speeds on the airfoil and then decreases to subsonic speeds after the shock wave.

4.9.3 Discussion

The obtained results resemble the expected behavior of a NACA 0012 airfoil under transonic speeds, as the fact that it is symmetric and the lack of advanced camber geometry as in the case of supercritical airfoils allow for a straightforward interpretation of the results.

However, compared to usual calculations obtained with commercial CFD solver which include turbulence modeling and boundary layer treatment, the obtained results are incomplete. Even though the transonic flow behavior has apparently been properly calculated, the lack of boundary layer models do not allow for a complete analysis of the results. The presence of an elevated adverse pressure gradient due to the appearance of a shock wave does not result in any kind of boundary layer growth or detachment, as the ALPACA solver is not originally intended for these kinds of aerodynamics problems and the current rigid body boundary implementation is actually modeled as a slip wall with no boundary layer presence. As such, the results can be considered to be incomplete, even if the calculated solution corresponds with the expected behavior for this kind of solver.

As such, it is possible to see that the employed solver is able to solve external compressible aerodynamics problems with the new rigid body boundary definition. However, the present wall modeling does not allow for any boundary layer treatment, which limits the usefulness of this solver compared to available commercial solutions, academic research purposes notwithstanding.

Chapter 5

Conclusion

In the present work, the ALPACA solver provided by the Nanoshock group of the Chair of Aerodynamics and Fluid Mechanics is reformulated and extended beyond its original scope of application, which was compressible multi-component flow simulations, for the purpose of also accommodating the calculation of boundary-dominated flows involving a rigid body interface. This extension is of particular interest, as it opens a wide scope of new fluid cases to be analyzed, as numerical fluid simulations involving a rigid body boundary and a compressible fluid can be particularly useful for some applications, as mentioned in [2].

Additionally, it has been shown that the obtained results have been able to be validated compared to the calculations of the reference papers, showing a reasonable correlation between the provided and the obtained results. However, further improvements to the employed method are still possible, particularly in the case definition procedure for end users and perhaps additional testing for particular boundary-dominated flow configurations might be needed.

First, the cases involving the forward-facing step and the muzzle blast problem have been recreated from their original reference papers. These cases involve a rigid body boundary, but they can be grouped together as both have the rigid body partially attached to the outer boundaries of the domain, with no floating rigid body completely immersed withing the fluid.

The obtained results show that the refactored code can solve the flow configurations with no numerical stability concerns and the shape of the obstacle is kept. The obtained results provide a good correlation compared to the reference papers and the deviations are not necessarily attributable to the new rigid body definition itself.

Secondly, several similar cases involving first an obstacle within the fluid and an array of those obstacles later are also calculated. These cases present the additional complexity of having an immersed, enclosed rigid body obstacle within the domain also involving sharp acute edges for some configurations. In general, the obtained results can be qualitatively comparable to those of the reference literature.

In addition to that, a De Laval nozzle and a NACA 0012 airfoil have been also analyzed employing the new code extensions. In this case, no reference papers are provided, but the quantitative validation of the calculations resided in the available theoretical knowledge concerning both flow configurations. Considering the limitations due to the scope of the available comparable solutions, the obtained results seem to be adequate also for these additional flow configurations.

Lastly, it is possible to affirm that the realized work shows the possibility of calculating boundary-dominated flow configurations employing a levelset interface method in a compressible multi-component solver.

Bibliography

[1] Batten, P., Clarke, N., Lambert, C., and Causon, D. M. “On the choice of wavespeeds for the HLLC Riemann solver”. In: *SIAM Journal on Scientific Computing* 18.6 (1997), pp. 1553–1570.

[2] Chaudhuri, A., Hadjadj, A., Sadot, O., and Ben-Dor, G. “Numerical study of shock-wave mitigation through matrices of solid obstacles”. In: *Shock Waves* 23.1 (2013), pp. 91–101.

[3] Fleischmann, N., Adami, S., and Adams, N. A. “Numerical symmetry-preserving techniques for low-dissipation shock-capturing schemes”. In: *Computers & Fluids* 189 (2019), pp. 94–107.

[4] Harten, A. “High resolution schemes for hyperbolic conservation laws”. In: *Journal of computational physics* 135.2 (1997), pp. 260–278.

[5] Harten, A., Engquist, B., Osher, S., and Chakravarthy, S. R. “Uniformly high order accurate essentially non-oscillatory schemes, III”. In: *Upwind and high-resolution schemes*. Springer, 1987, pp. 218–290.

[6] Harten, A., Lax, P. D., and Leer, B. v. “On upstream differencing and Godunov-type schemes for hyperbolic conservation laws”. In: *SIAM review* 25.1 (1983), pp. 35–61.

[7] Hu, X. Y., Khoo, B., Adams, N. A., and Huang, F. “A conservative interface method for compressible flows”. In: *Journal of Computational Physics* 219.2 (2006), pp. 553–578.

[8] Jiang, G.-S. and Shu, C.-W. “Efficient implementation of weighted ENO schemes”. In: *Journal of computational physics* 126.1 (1996), pp. 202–228.

[9] Kaiser, J., Adami, S., and Adams, N. “Three-dimensional direct numerical simulation of shock-induced bubble collapse near gelatin”. In: *11th International Symposium on Turbulence and Shear Flow Phenomena, TSFP 2019*. 2019.

[10] Kaiser, J., Adami, S., Akhatov, I., and Adams, N. “A semi-implicit conservative sharp-interface method for liquid-solid phase transition”. In: *International Journal of Heat and Mass Transfer* 155 (2020), p. 119800.

[11] Min, C. “On reinitializing level set functions”. In: *Journal of computational physics* 229.8 (2010), pp. 2764–2772.

[12] Toro, E. F. “The HLLC Riemann solver”. In: *Shock Waves* 29 (2019), pp. 1065–1082.

[13] Toro, E. F. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Berlin Heidelberg: Springer Science & Business Media, 2013. ISBN: 978-3-540-49834-6.

[14] Toro, E. F., Spruce, M., and Speares, W. “Restoration of the contact surface in the HLL-Riemann solver”. In: *Shock waves* 4.1 (1994), pp. 25–34.

[15] Wang, J. and Widhopf, G. “Numerical simulation of blast flowfields using a high resolution TVD finite volume scheme”. In: *Computers & Fluids* 18.1 (1990), pp. 103–137.

[16] Winter, J., Kaiser, J., Adami, S., and Adams, N. “Numerical investigation of 3D drop-breakup mechanisms using a sharp interface level-set method”. In: *11th International Symposium on Turbulence and Shear Flow Phenomena, TSFP 2019*. 2019.

[17] Woodward, P. and Colella, P. “The numerical simulation of two-dimensional fluid flow with strong shocks”. In: *Journal of computational physics* 54.1 (1984), pp. 115–173.

[18] Zhang, Y., Fang, X., Zou, J., Shi, X., Ma, Z., and Zheng, Y. “Numerical simulations of shock/obstacle interactions using an improved ghost-cell immersed boundary method”. In: *Computers & Fluids* 182 (2019), pp. 128–143.

Disclaimer

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

Garching, September 30, 2020

(Signature)
Luis Rafael Piqueras Andreu