UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA DE SÃO CARLOS

Vinicius Henrique Aurichio

Immersed-interface methods in the presence of shock waves

São Carlos

2019

Vinicius Henrique Aurichio

Immersed-interface methods in the presence of shock waves

Thesis presented to the Graduate Program in Physics at the Instituto de Física de São Carlos, Universidade de São Paulo, to obtain the degree of Doctor in Science.

Concentration area: Basic Physics

Advisor: Prof. Dr. Attilio Cucchieri
Co-advisor: Profa. Dra. Maria Luiza Bambozzi de Oliveira

Original version

São Carlos

2019

Aurichio, Vinicius Henrique
Immersed-interface methods in the presence of shock waves / Vinicius Henrique Aurichio; advisor Attilio Cucchieri; co-advisor Maria Luísa Bambozzi de Oliveira -- São Carlos 2019.
85 p.

Thesis (Doctorate - Graduate Program in Física Básica) -- Instituto de Física de São Carlos, Universidade de São Paulo - Brasil , 2019.

1. Immersed-interface methods. 2. Navier-Stokes equations. 3. Shock waves. I. Cucchieri, Attilio, advisor. II. de Oliveira, Maria Luísa Bambozzi, co-advisor. III. Title.

FOLHA DE APROVAÇÃO

Vinícius Henrique Auríchio

Tese apresentada ao Instituto de Física de São Carlos da Universidade de São Paulo para obtenção do título de Doutor em Ciências. Área de Concentração: Física Básica.

Aprovado(a) em: 03/05/2019

Comissão Julgadora

Dr(a). Attilio Cucchieri

Instituição: (IFSC/USP)

Dr(a). Paulo Celso Greco Junior

Instituição: (EESC/USP)

Dr(a). Leonardo Paulo Maia

Instituição: (IFSC/USP)

Dr(a). Murilo Francisco Tomé

Instituição: (ICMC/USP)

Dr(a). José Antônio Silveira Gonçalves

Instituição: (UFSCar/São Carlos)

To Krissia, who always believed I could do it.
And to our cats, Marie and Erwin, that were always looking.

ACKNOWLEDGEMENTS

I did not write this thesis alone and I must express my gratitute to those that supported me along the way. I am sure this list will be far from comprehensive and that my words will not give them the appropriate level of recognition, but I will try my best just as they did for me.

I would like to thank both of my advisors, Prof. Attilio Cucchieri and Profa. Maria Luiza Bambozzi de Oliveira for all the time they dedicated to guiding me in this PhD. I could always just walk in their rooms to discuss any topic, related to my PhD or not, and I would be welcome. It was from those conversations that everything in this thesis was born. They would proofread all my texts, even when I sent them just one day before a deadline. Their comments always transformed my OK texts in excellent texts. Thank you very much for everything.

When I first entered my room in the Physics department as a PhD candidate I immediatly made a new friend. Cesar Uliana and I could talk about anything and never get bored or tired of one another. It was also through him that I met many of the friends I made during my PhD: Renata, Paulo Eduardo, Tiago, Ágide and Carlos. They would all come to our room after lunch to drink coffee and chat, and very often that would be the best part of my day. Our coffee group kept growing to the point where we all could not fit in my small room and we had to find a larger one. That is how I got closer to José Ricardo, José Teixeira and Naidel, three more in this great group of friends. The little time we shared everyday made everything better.

From my first year as an Undergraduate student until today I have been practicing kung-fu. I have met hundreds of people during the training sessions and learned a lot about myself through all of them. I enjoyed kung-fu so much that I became the responsible professor in São Carlos, after Danilo—who was teaching me at the time—had to leave the city to work. Once I became a professor I had the honor to teach great students. Gustavo, Natalia, Michelle, Gabriel Fabiano, Gabriel Lopes, Nicolas and Sérgio Zanchetta would always learn fast and happily replace me in class, whenever I needed some extra time to write a piece of code, or send an abstract at the last minute.

About one year ago I had to move from São Carlos and work. This was a big change and, at times, it seemed to me I would not be able to finish my PhD. But I am a very lucky person after all. My immediate bosses, Henrique and Ricardo, were very understanding whenever I had to take a day-off to attend to academic activities and also whenever I arrived like a zombie at work after a sleepless night writing this thesis. Many of my co-workers have academic degrees of their own and understood what I was going

through from experience, and all of them encouraged me to proceed. José Vitor, Thais, Giulia, Gabriel, Caio, Julia, Mariana, Verna, Wesley, Afonso, Daniela San Juan, Daniela Saito, André, Priscila, Regiane, thank you very much for your support.

Marta, Thereza, André, Paulo Matias and Joseana, you are all amazing people. I know you all for more than a decade now and our friendship never gets old. André and Paulo, most of what I know about computers came from being in touch with you through these years. Marta, Thereza and Joseana, you have seem me whole and broken, and helped through every step of the way. Thank you.

Krissia, I dedicated this thesis to you and that is not nearly enough to show my gratitude towards you. You are much more than my wife. You and I know that without you, even with all the support everyone else gave me, I would not have finished this thesis. We taught each other more than I thought possible and I am sure we have still to learn a lot more together.

Finally, I acnkowledge CNPq for the financial support in my PhD scholarship.

"Look ma, no wiggles!."
Gino Moretti

ABSTRACT

AURICHIO, V. H. **Immersed-interface methods in the presence of shock waves**. 2019. 85p. Thesis (Doctor in Science) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2019.

Fluid motion has always been of great importance for humanity since much of our progress has been related to our understanding of fluid dynamics and to our control over the fluids surrounding us. In particular, the experimental techniques and the methods for numerical simulation developed during the last century allowed for great progresses both in creating new technologies and in improving old ones. Despite the great importance of experimental techniques, measuring all properties of a fluid throughout the whole domain, without intefering with the flow to be studied, is impossible. Also, building models even in scale is usually expansive. Both of these reasons have driven the development of numerical methods to the point they became an invaluable tool for fluid dynamic studies and the main tool for developing engineering solutions. If numerical methods are to be of any use, though, they have to correctly describe the problem geometry as well as capture the rich dynamics in a variety of flow situations, such as turbulence, boundary-layers and shock-waves. This thesis addresses two of these problems. In particular, I show modified versions of two immersed-interface methods to describe the geometry, simplifying their implementations with no impact to their applicability. I also introduce two methods for handling shock-waves: first aiming to minimize computational costs, then improving shock-wave resolution without increasing the number of grid points.

Keywords: Immersed-interface methods, Navier-Stokes equations, shock waves.

RESUMO

AURICHIO, V. H. **Métodos de interface imersa na presença de ondas de choque**. 2019. 85p. Tese (Doutorado em Ciências) - Instituto de Física de São Carlos, Universidade de São Paulo, São Carlos, 2019.

O movimento dos fluidos sempre foi de grande importância para a humanidade, dado que muito de nosso progresso esteve intimamente relacionado a um entendimento mais profundo de fluidodinâmica e de como controlar os flúidos ao nosso redor. Em particular, os métodos experimentais e de simulação computacional, desenvolvidos no último século, nos permitiram grandes avanços na criação de novas tecnologias e na otimização das já existentes. Apesar de sua grande importância, as dificuldades de se mensurar todas as propriedades de um flúido em todo o espaço, sem interferir com o comportamento do fluxo, além dos custos de se elaborar experimentos em tamanho real ou em escala, fez com que cada vez mais os métodos numéricos se tornassem uma importante ferramenta no estudo da fluido dinâmica e a principal ferramenta para o desenvolvimento de soluções de engenharia. Porém, para efetivamente substituir experimentos, os métodos numéricos tem que ser capazes de corretamente descrever a geometria do problema, além de capturarem todo tipo de comportamento apresentado pelos flúidos, como turbulência, camada limite e ondas de choque. Esta tese busca contribuir com dois destes desafios. Em particular, mostro versões modificadas de métodos de interface imersa para a descrição da geometria, simplificando as implementações originais sem prejudicar sua aplicabilidade. Também abordo métodos para tratar ondas de choque: primeiro buscando minimizar o esforço computacional e depois buscando aumentar a resolução do choque sem precisar refinar a malha computacional.

Palavras-chave: Métodos de interface imersa, equações de Navier-Stokes, ondas de choque.

LIST OF FIGURES

LIST OF SYMBOLS

$x \equiv x_1\, y \equiv x_2$ Cartesian coordinates

$\Delta x,\ \Delta y$ Grid spacing in x and y directions respectively

x_i Coordinate $x = x_{min} + i\Delta x$

y_i Coordinate $y = y_{min} + i\Delta y$

ϕ_i Quantity ϕ evaluated at position $x = x_i$

$\phi_{i,j}$ Quantity ϕ evaluated at position $x = x_i,\ y = y_i$

ϕ_i^n Quantity ϕ evaluated at position $x = x_i$ and time $t = t_n$

C_p Specific heat at constant pressure

C_v Specific heat at constant volume

$\gamma = C_p/C_v$ Heat capacity ratio

$\delta = (\gamma - 1)/2$ Convenient definition

κ Thermal condutivity

μ Viscosity

ρ Fluid density

$u \equiv u_1,\ v \equiv u_2$ Velocity in the x and y directions respectively

T Temperature

$e = C_v T$ Internal energy (ideal gas)

$p = (\gamma - 1)\rho e$ Pressure (ideal gas)

$E = \rho(e + (u^2 + v^2)/2)$ Total energy

$\tau_{xx} \equiv \tau_{11},\ \tau_{xy} \equiv \tau_{12},\ \tau_{yx} \equiv \tau_{21},\ \tau_{yy} \equiv \tau_{22}$ Viscous stress tensor components

$q_x \equiv q_1, q_y \equiv q_2$ Heat flux in the x and y directions respectively

$c = \sqrt{\gamma p/\rho}$ Speed of sound (ideal gas)

$S = \frac{1}{\gamma(\gamma-1)} \ln(p/\rho^\gamma)$ Entropy (ideal gas)

$Re = \rho w L/\mu$ Reynolds number for some velocity w and reference length L

$Pr = C_p\mu/\kappa$ Prandtl number

$Ma = w/c$ Mach number for some velocity w

δ_{ij} Kronecker delta

CONTENTS

1 INTRODUCTION

Fluids are fundamental in our everyday lives: our bodies are composed of about 70% water, we constantly breath air, Earth has most of its surface covered by water, and learning to manipulate water to irrigate crops enabled humans to settle and build societies.[3] For millennia, most of human understanding of fluids came from the necessity to transport them to specific areas, be it for agriculture, for human and cattle consumption, or for flood protection. This only changed when the Greek scientist Archimedes, in his work *On Floating Bodies*, introduced the law of buoyancy, also known as Archimedes' Principle. After him, various complex tools were created to manipulate fluids: water pumps were studied in Alexandria around 120 BC under Ptolemies, Roman aqueducts efficiency was studied by Sextus Julius Frontinus around 90 DC, and the study of specific weights, automatic controls, plug valves by Islamic engineers led to the development of mathematical theories of ratios and infinitesimals.

Significant progress was made on the theory of fluids when the Renaissance began. Leonardo da Vinci took rich notes while observing the movement of fluids, which foresaw the conservation of mass in one-dimensional steady flow. Galileo's disciples Benedetto Castelli and Evangelista Torricelli applied the discoveries of their master to study the motion of rivers and canals, as well as the velocity of a water jet coming from the bottom of a vessel. These works summarized years of observation and enabled further progress in the theory of fluid motion. In 1663, a treatise by Blaise Pascal on the equilibrium of liquids[4] was published after his death, effectively elevating hydrostatics to a science. Pascal developed simple proofs to the laws of the equilibrium of fluids that were amply confirmed by experiments. Also Sir Isaac Newton turned his attention to the movement of fluids. In his work *Philosophiae Naturalis Principia Mathematica*, while trying to understand what made fluids slow down, Newton concluded that the sheer stress, on an interface tangent to the direction of flow, is proportional to the velocity gradient. This particular form of the sheer stress is called Newton's viscosity law, and fluids which obey this law are called Newtonian fluids.

Leonhard Euler published the general form of the continuity equation and of the momentum equation in 1757.[5] This allowed for a full description of the movement of incompressible fluids. Let us recall that these equations were among the first partial differential equations to be written down. The last equation of fluid motion, the energy conservation equation, was derived 59 years later by Laplace and together with Euler's equations of fluid dynamics allow to fully describe the movement of inviscid fluids.

Euler's equations had an obvious flaw that made it of limited use for engineering though: it did not account for viscosity. In 1752, Jean le Rond d'Alembert showed that

Euler's equations implied that immersed objects, moving at constant speed relative to the fluid, would not experience drag, in direct contradiction with experiments and experience. This is known as d'Alambert's paradox and, until a solution was found, theoretical fluid mechanics and engineering were developed separately.

Even though Newton introduced the basic ideas of a mathematical formulation for viscous effects, it was only in 1822 that Claude-Louis Navier introduced the correct term in the momentum equation.[6] Navier derived the dissipative term from intermolecular forces, but his derivation was valid only for incompressible fluids. In 1845, George Gabriel Stokes finally derived the same equation as Navier, only this time in a way valid for compressible fluids as well. This equation, representing the momentum dynamics in a Newtonian dissipative medium, is know as the Navier-Stokes equation. When bundled with the mass conservation equation and the energy equation (including the dissipative effects as well), they are collectively known as the Navier-Stokes equations. Analytical solutions to the full equations are available for a few cases, but a general solution is not known.

Boundary-Layer theory is an interesting approach to study fluid dynamics analytically. In this method, the flow far from an immersed-body is computed using Euler's equations, since the viscous terms are negligible compared to the convective terms. Near the body, on the contrary, the flow is dominated by viscous effects and different solutions need to be found in this limit. Finally, a prescription for matching both solutions is used to compute the flow in the whole domain. Analytical solutions can only take us so far, though. To study flows in complex geometries, such as those of interest in engineering, numerical solutions had to be constructed. Various methods have been developed to generate such solutions and the impact they had in our world is impossible to overestimate. Numerical solutions to the Navier-Stokes equations have been successfully used (to mention a few applications) to improve the aerodynamics of cars and airplanes by reducing drag and improving lift, to ameliorate the hydrodynamics of boats and submarines, to design better water pumps and valves. Modified versions of the equations have also been used to study reactive flows, plasma moving under magnetic fields—such as in fusion reactors, stars and interstellar space—, granular media and non-Newtonian fluids.

Yet another challenge appears when dealing with high-speed flows: shock-waves can form. Pressure and density gradients can increase more and more as the fluid velocity gets larger, which can lead to discontinuities. These discontinuities then have to be appropriately handled both numerically and analytically. Shock-waves are found in front of high-speed projectiles and intercontinental ballistic missiles, and are the destructive drive of high-explosives. Then, it is no surprise that much of the research on shock-waves were sponsored by military branches. There are also less belligerent applications to shock-waves, though.

Shock waves are used, for example, for kidney stones treatment[7] and for some types of tendinopathy.[8]

In this thesis, I will describe methods that allow us to account for immersed solid bodies in a simulation, and the modifications demanded by the presence of shock waves in the fluid flow. To this end, I will first introduce the main numerical methods used to study fluid flows in chapter 2. Then I will address two of the many problems that arise in the numerical study of fluids. First, in chapter 3, I will show methods capable of describing complex geometries, which are at the same time simple to implement, capable of handling moving boundaries, and parallelizable. Second, in chapter 4, I will describe the many ways shock-waves can be handled, giving special attention to the shock-fitting method. In chapter 5 I show some numerical results obtained using the techniques introduced in the previous chapters and I conclude the thesis in chapter 6 by summarizing my work.

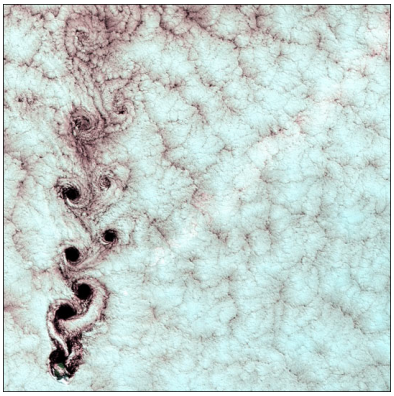
2 NUMERICAL SIMULATIONS OF FLUIDS

Developing a numerical method for fluid simulation is difficult, as will be discussed in this chapter and in the next two. In particular, the non-linear nature of the governing equations of motion—the Navier-Stokes equations—creates a coupling between small numerical errors and the desired solution which, if not appropriately treated, can lead to unstable or unphysical solutions. An example of these types of effects will be shown in (fig 28).

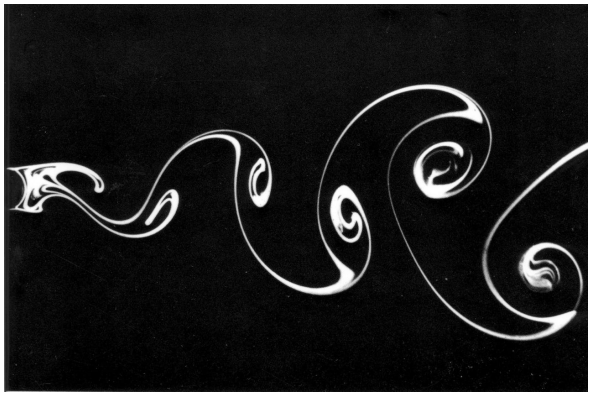
Here, I will first recall the Navier-Stokes equations and write them as two equivalent sets of differential equations. Then, I will discuss how numerical methods can be classified with respect to both their spatial and temporal discretization. I will discuss also the topic of filtering, which helps to deal with numerical errors when using a central differencing scheme for spatial discretization. In section 2.4, in order to avoid differentiating across shock-wave discontinuities, I will introduce shock detectors. Finally, in the last part of this chapter, I will present the specific methods used in this thesis: central and upwind finite-difference schemes, weighted essentially non-oscillatory method (WENO), a filter method and a shock-detector method.

2.1 Navier-Stokes equations

The Navier-Stokes equations describe the motion of heat-conducting viscous fluids. When studying these equations, it is useful to cast them in non-dimensional form.[9] This is achieved by writing any dimensional variable $\hat{\phi}$ as $\phi\phi^*$, where ϕ is an adimensional quantity, and ϕ^* is a dimensional reference value. For instance, the density ρ^* of dry air at standard temperature and pressure (STP) conditions is $1.2754 kg/m^3$. If we use this density as our reference, we can obtain the adimensional density ρ for any value of the dimensional density $\hat{\rho}$ as $\rho = \hat{\rho}/\rho^*$. The reference values can then be combined into three dimensionless constants that are characteristic of the system being considered: the Reynolds number Re, the Prandtl number Pr and the freestream Mach number Ma_∞. The Reynolds number $Re = \rho^* v^* L^*/\mu^*$ depends on the reference values of density ρ^*, velocity v^*, length L^* and viscosity μ^* and is usually interpreted as the ratio between inertial (convective) forces $(\rho^* v^*)$ and viscous (dissipative) forces (μ^*/L^*). The Prandtl number $Pr = C_p \mu^*/k$ depends on the heat capacity at constant pressure C_p, the viscosity μ^* and the thermal conductivity k, and is defined as the ratio of viscous diffusion rate to the thermal diffusion rate. The freestream Mach number $Ma_\infty = v^*/c_\infty$ is the ratio between the reference velocity v^* and the speed of sound c_∞ computed from the reference values for temperature, density and velocity. This adimensionalization procedure has two benefits: any bounds derived for the stability of the numerical methods will depend only on the characteristic constants, not



(a) Vortices near an island.



(b) Vortices near a cylinder.

Figure 1 – von Kármán vortices. Even though a mountain in an island (hundreds of meters) and a small cylinder (few centimeters) have different scales, the flow around the body in both systems exhibit the same vortex pattern, the so-called von Kármán vortex sheet. The occurrence of this effect depends mainly on the Reynolds number of the flow, one of the characteristic values obtained by combining the reference values of the system. Source: (a) EARTH OBSERVATORY [10] (b) VAN DYKE [11].

on the actual reference values; it also becomes evident that solutions of the equations are dependent only on the characteristic constants. From the second statement it follows that systems with different sizes can have identical dynamics as long as their characteristic constants are the same, as the example shown in (fig 2).

In their original version, the Navier-Stokes equations are written in terms of fluid's density ρ, the velocity components u_i, the total energy E, the pressure p, the heat-flux components q_i and the stress-tensor components τ_{ij}.[6,12] In non-dimensional form, their 2D version are written as[9]

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{2.1}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j + p\delta_{ij} - \tau_{ji}) = 0, \;\; i = 1, 2 \tag{2.2}$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x_j}(u_j E + u_j p + q_j - u_i \tau_{ij}) = 0, \tag{2.3}$$

where the implied summation over repeated indices is used. Also, the characteristic constants are incorporated to the definitions of τ_{ij} and q_i, so this form of the Navier-Stokes equations is the same for every system. The equations are in conservation form and they describe the mass conservation (eqn 2.1), the momentum conservation (eqn 2.2), and the energy conservation (eqn 2.3). Note that there are more variables than equations, so we have to combine the Navier-Stokes equations with the constitutive relations of the fluid of

interest, which in our case are

$$p = (\gamma - 1)\rho e \tag{2.4}$$

$$q_j = -\frac{\mu}{(\gamma - 1)Ma_\infty^2 RePr}\frac{\partial T}{\partial x_j} \tag{2.5}$$

$$\tau_{ij} = \frac{2\mu}{Re}S_{ij} \tag{2.6}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}, \tag{2.7}$$

where δ_{ij} is the Kronecker delta. From the above equations we see that this fluid is an ideal gas (eqn 2.4), obeys the Fourier law for heat transfer by conduction (eqn 2.5), is Newtonian (eqn 2.6), and has negligible bulk viscosity, which yields a viscous strain tensor with a simple form (eqn 2.7).

It is useful to define the vectors

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} \tag{2.8}$$

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E + p)\, u \end{bmatrix} \tag{2.9}$$

$$G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E + p)\, v \end{bmatrix} \tag{2.10}$$

$$F_D = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xx}u + \tau_{xy}v - q_x \end{bmatrix} \tag{2.11}$$

$$G_D = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yx}u + \tau_{yy}v - q_y \end{bmatrix} \tag{2.12}$$

so that (eqn 2.1), (eqn 2.2), and (eqn 2.3) can be summarized as

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = \frac{\partial F_D}{\partial x} + \frac{\partial G_D}{\partial y}. \tag{2.13}$$

The vectors F and G are called the fluid's x and y fluxes respectively, while F_D and G_D contain the dissipative terms.

When studying shock-waves in chapter 4 it will be convenient to use an alternative form of the Navier-Stokes equations. This form can be obtained from the first one by a series of variable changes. Instead of using the density ρ, the velocities u and v, and the total energy E, this form uses the sound speed c, the velocities u and v, and the entropy

S. Then, using $\delta = (\gamma - 1)/2$, the Navier-Stokes equation become[13]

$$\frac{1}{\delta}\frac{\partial c}{\partial t} + \frac{1}{\delta}u_i\frac{\partial c}{\partial x_i} - c\frac{\partial S}{\partial t} - cu_i\frac{\partial S}{\partial x_i} = \frac{2\delta c}{\gamma p}\left(\frac{\partial q_i}{\partial x_i} + \frac{\partial}{\partial x_i}\left(u_j\tau_{ij}\right)\right) \tag{2.14}$$

$$\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j} + \frac{c}{\delta}\left(\frac{\partial c}{\partial x_1} + \frac{\partial c}{\partial x_2}\right) - c^2\left(\frac{\partial S}{\partial x_1} + \frac{\partial S}{\partial x_2}\right) = \frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial x_j},\ i = 1, 2 \tag{2.15}$$

$$\frac{\partial S}{\partial t} + u_i\frac{\partial S}{\partial x_i} = \frac{1}{\gamma p}\left(\frac{\partial q_i}{\partial x_i} + \frac{\partial}{\partial x_i}\left(u_j\tau_{ij}\right)\right). \tag{2.16}$$

2.2 Discretizations

To obtain numerical solutions for the Navier-Stokes equations, presented in the last section, there are various possible approaches. For the purpose of this thesis it will suffice to consider Eulerian methods—in which we describe the fluid properties in fixed points in space—capable of simulating compressible high-speed flows. This means I will not describe Lagrangian methods (e.g. Smoothed Particle Hydrodynamics[14])—where the fluid particles are tracked individually and no grid is necessary—, nor the Lattice Boltzmann method[15] (suited for low Mach number, incompressible flows), nor spectral methods[16] (not suited for flows with discontinuities such as shock waves).

2.2.1 Truncation errors

An important concept when studying numerical approximations is the truncation error. A numerical solution obtained by any method will almost always differ from the exact solution to the problem, and so it is necessary to measure how large is the error introduced by the approximations. Besides the absolute value of the error, different discretization methods have different convergence ratios with respect to refining the discretization. To make these ideas more concrete, let $f(x)$ be an infinitely differentiable function, and $D(f, x_i) = F(f_{i-k}, \ldots, f_{i+m}),\ k, m \in \mathbf{N}$ an approximation for the first derivative, defined on a regular grid, with spacing Δx, and evaluated at $x = x_i$. The difference between the real derivative and the proposed approximation is given by

$$\left.\frac{df}{dx}\right|_{x=x_i} - D(f, x_i) = TE(D) = \sum_{m=l}^{+\infty}\alpha_m\frac{d^{m+1}f}{dx^{m+1}}(\Delta x)^m, \tag{2.17}$$

with $\alpha_m \in \mathbf{R},\ l \in \mathbf{N}$. As shown above, the truncation error $TE(D)$ of the approximation given by D can be written as a power series of the grid width Δx. Notice that, as the grid is refined, Δx goes to zero and so the approximation converges to the real value of the derivative. Moreover, the error term is dominated by $(\Delta x)^l$, the first term in the series expansion, and this determines the convergence rate of this particular approximation. It is then said that the discretization D has a truncation error of order $(\Delta x)^l$, or, in a shorthand notation, that $TE(D) = \mathcal{O}\left((\Delta x)^l\right)$. As a consequence, the discretization D itself is said to have order l.

2.2.2 Spatial discretization

For the purpose of this thesis I only consider three categories of spatial discretization: finite difference, finite volume, and finite element methods.

2.2.2.1 Finite difference (FD) methods

Finite difference methods, as the name implies, approximate the spatial derivatives by differences. The points used in the discretization must form a grid, regular or not, as in (fig 2a). The accuracy of the approximation is related to the number of points used to reconstruct the derivatives. All methods employed in this thesis belong to this category and a more detailed description will be given in section 2.5.3.

2.2.2.2 Finite volume (FV) methods

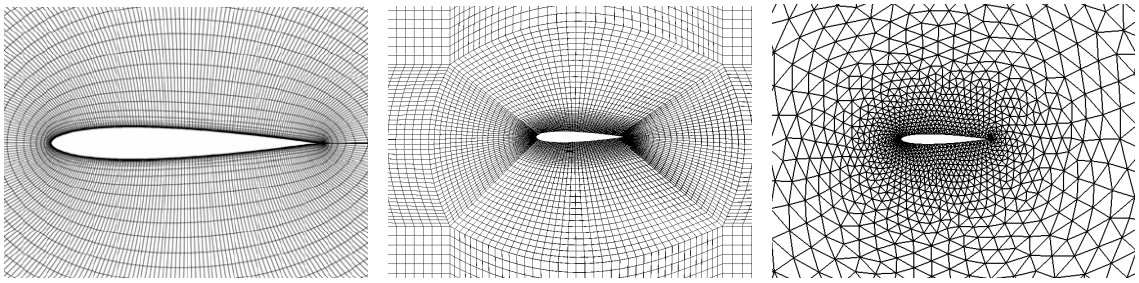
In finite volume methods, the space is partitioned in arbitrarily shaped, non-overlapping pieces called cells. Although it is possible to combine cells of different shapes, it is usual to consider one specific shape (triangles or quadrilaterals are the most common choices in 2D). This allows for much more flexible meshes as shown in (fig 2b) and (fig 2c). Instead of directly computing the derivatives as in the finite difference methods, we use the divergence theorem to pose the question differently: the rate of change of a quantity within a cell equals the net flux across its boundaries, so if we compute the fluxes at the boundaries we can obtain the variation in the flow variables. The accuracy of a FV discretization is related to the reconstruction of the fluxes at the cells interfaces.

2.2.2.3 Finite element (FE) methods

The same meshes used in FV methods can be used to perform computations using the finite element methods. Here, the pieces covering the space are called elements instead of cells. The flow-variable values are computed at some points in the element (the nodes), which allows their values at all points inside each element to be computed by interpolation. The compatibility relations between neighboring elements is given by the Navier-Stokes equations, resulting in a set of algebraic equations that must be solved at each time step. The accuracy in a FE discretization is related to the number of nodes in each element, which in turn determines the degree of the interpolating function used, as well as to the size of each element.

2.2.3 Temporal integration

Once the space is discretized using one of the previous schemes, it is necessary to choose a method to compute the time evolution of the Navier-Stokes equations. At any time t of the computation the values of all flow variables $\phi(t)$ will be known and we would like to determine their values at $t + \Delta t$, i.e. $\phi(t + \Delta t)$. There are two families of methods



(a) Structured mesh. (b) Multiblock mesh. (c) Unstructured mesh.

Figure 2 – Examples of body-fitted meshes around a NACA0012 airfoil. Structured meshes are topologically equivalent to a Cartesian grid. Multiblock meshes combine multiple structured meshes to adapt to more complex geometries. Finally, unstructured meshes partition the domain in arbitrarily-shaped non-overlapping pieces.
Source: (a) JAVADI [17], (b) MANISHA [18], (c) CHEN [19]

that we cover next. To this end, consider the differential equation

$$\frac{d\phi}{dt} = \mathcal{F}(\phi(t),\ t), \tag{2.18}$$

where $\mathcal{F}$ is an arbitrary function depending on ϕ and t.

2.2.3.1 Explicit time-integration methods

In an explicit time-integration methods, the values of $\phi(t+\Delta t)$ depend only on the values of ϕ at previous times.[20] Therefore, given $\phi(t)$ (and possibly $\phi(t-\Delta t),\ \phi(t-2\Delta t)$ etc), the values of $\phi(t+\Delta t)$ are immediately computable. The simplest explicit method is due to Euler, usually called Euler method. If we apply it to (eqn 2.18) we obtain

$$\frac{\phi(t+\Delta t)-\phi(t)}{\Delta t} = \mathcal{F}(\phi(t),\ t) \tag{2.19}$$

$$\phi(t+\Delta t) = \phi(t) + \Delta t\ \mathcal{F}(\phi(t),\ t). \tag{2.20}$$

Notice that (eqn 2.20) explicitly gives us the value of $\phi(t+\Delta t)$.

2.2.3.2 Implicit time-integration methods

Implicit time-integration methods require the solution of an equation to obtain $\phi(t+\Delta t)$.[20] This equation is sometimes linear, and it is then solvable using exact methods, such as Gaussian elimination, but it can also be non-linear and some approximation might be needed. The simplest implicit method is also due to Euler, the so-called backward Euler method, which when applied to (eqn 2.18) results in

$$\frac{\phi(t+\Delta t)-\phi(t)}{\Delta t} = \mathcal{F}(\phi(t+\Delta t),\ t+\Delta t) \tag{2.21}$$

$$\phi(t+\Delta t) = \phi(t) + \Delta t\ \mathcal{F}(\phi(t+\Delta t),\ t+\Delta t). \tag{2.22}$$

Note that, in contrast to explicit methods, it is not possible, in general, to obtain an explicit expression for $\phi(t+\Delta t)$ from (eqn 2.22).

2.3 Filtering

Finite-difference schemes with central stencils (central-difference schemes) are interesting because, as a consequence of their symmetry, there are no even derivatives in the truncation error, and so they have no numerical dissipation. This, however, leads to instabilities,[21] as there are no mechanism to control the growth of numerical errors. The errors can accumulate and the numerical solution becomes unphysical. A possible solution to this issue is to introduce filtering, a mechanism through which the highest frequencies are attenuated. It is important to realize that the appeal of a central-difference scheme is precisely its zero-dissipation nature and that filtering introduces an artificial mechanism that is similar to dissipation to control the numerical errors. This means that the filtering mechanism must be chosen carefully, otherwise there will be no advantage in using a central-difference scheme.

2.4 Shock detectors

Under certain circumstances the flow variables may develop discontinuities—such as gradient discontinuities, contact discontinuities, and shock-waves—due to the transition from subsonic to supersonic flow. It is important to detect the formation of these discontinuities early in the simulation in order to avoid taking numerical derivatives across them. Indeed, not only the numerical result would be unphysical, as the information can only travel one-way across shock-waves, but also oscillations would form, as a consequence of the Gibbs phenomenon. [22–24] Shock detectors are methods that can detect sharp variations in the function, so that in regions with sharp variations an appropriate method can then be used.

2.5 Methods used in this thesis

After this short overview of numerical discretizations, I will detail the numerical approximations used in this thesis. All the spatial discretizations are in the finite-difference category, and the time discretization used is a member of the Runge-Kutta family of methods.

2.5.1 Flux splitting

Two of the methods that follow depend on a separation of the flux $F(G)$ into two components, representing waves travelling along the positive and negative directions of

the x(y)-axis. Steger and Warming[25] described a method to compute one such separation for an ideal gas. Defining the general flux $\mathcal{F}$ as

$$\mathcal{F}\left[\lambda_1, \lambda_3, \lambda_4;\ k_1, k_2\right] = \frac{\rho}{2\gamma}\begin{bmatrix} 2(\gamma-1)\lambda_1 + \lambda_3 + \lambda_4 \\ 2(\gamma-1)\lambda_1 u + \lambda_3(u+ck_1) + \lambda_4(u-ck_1) \\ 2(\gamma-1)\lambda_1 v + \lambda_3(v+ck_2) + \lambda_4(v-ck_2) \\ (\gamma-1)\lambda_1(u^2+v^2) + \frac{\lambda_3}{2}\left[(u+ck_1)^2 + (v+ck_2)^2\right] \\ +\frac{\lambda_4}{2}\left[(u-ck_1)^2 + (v-ck_2)^2\right] + \frac{(3-\gamma)(\lambda_3+\lambda_4)}{2(\gamma-1)} \end{bmatrix}, \tag{2.23}$$

where

$$\lambda_1 = k_1 u + k_2 v \tag{2.24}$$
$$\lambda_3 = \lambda_1 + c \tag{2.25}$$
$$\lambda_4 = \lambda_1 - c, \tag{2.26}$$

and considering

$$\hat{\lambda}_1^{\pm} = \frac{\lambda_1 \pm |\lambda_1|}{2} \tag{2.27}$$
$$\hat{\lambda}_3^{\pm} = \frac{\lambda_3 \pm |\lambda_3|}{2} \tag{2.28}$$
$$\hat{\lambda}_4^{\pm} = \frac{\lambda_4 \pm |\lambda_4|}{2}, \tag{2.29}$$

then we can write

$$F^+ = \mathcal{F}\left[\lambda_1^+, \lambda_3^+, \lambda_4^+; k_1 = 1, k_2 = 0\right] \tag{2.30}$$
$$F^- = \mathcal{F}\left[\lambda_1^-, \lambda_3^-, \lambda_4^-; k_1 = 1, k_2 = 0\right] \tag{2.31}$$
$$G^+ = \mathcal{F}\left[\lambda_1^+, \lambda_3^+, \lambda_4^+; k_1 = 0, k_2 = 1\right] \tag{2.32}$$
$$G^- = \mathcal{F}\left[\lambda_1^-, \lambda_3^-, \lambda_4^-; k_1 = 0, k_2 = 1\right]. \tag{2.33}$$

One can verify that $F = F^+ + F^-$ and $G = G^+ + G^-$, where F and G are the x and y fluxes defined in (eqn 2.9) and (eqn 2.10).

2.5.2 Weighted essentially non-oscillatory method (WENO)

Introduced by Liu, Shu and Osher[26] the weighted essentially non-oscillatory method (WENO) is a popular method to compute numerical solutions of the Navier-Stokes equations, since it is capable of yielding high-order of accuracy, of handling shock-waves and it is not very difficult to implement.

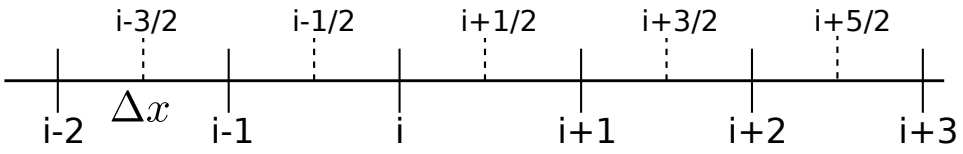


Figure 3 – Regular grid with Δx spacing. Dashed lines are at the mid-point between two consecutive grid points. The labels are simplified, so a point marked with i should be interpreted as being at position x_i.
Source: By the author.

I will illustrate the WENO procedure by considering the x derivative of one component ϕ of the F^+ flux. The first step is to define the following expression

$$\left.\frac{\partial\phi}{\partial x}\right|_{x=x_{i+1/2}} = \frac{\Phi_{i+1/2} - \Phi_{i-1/2}}{\Delta x}, \tag{2.34}$$

which is exact as long as a suitable expression for Φ is determined. We then proceed to construct three approximations for $\Phi_{i+1/2}$, obtained by interpolating values of Φ at different grid points:

$$\hat{\Phi}^{(1)}_{i+1/2} = \frac{1}{3}\phi_{i-2} - \frac{7}{6}\phi_{i-1} + \frac{11}{6}\phi_i \tag{2.35}$$

$$\hat{\Phi}^{(2)}_{i+1/2} = -\frac{1}{6}\phi_{i-1} + \frac{5}{6}\phi_i + \frac{1}{3}\phi_{i+1} \tag{2.36}$$

$$\hat{\Phi}^{(3)}_{i+1/2} = \frac{1}{3}\phi_i + \frac{5}{6}\phi_{i+1} - \frac{1}{6}\phi_{i+2}. \tag{2.37}$$

Notice that the above approximations are not symmetrical with respect to the point $x_{i+1/2}$, but show a bias towards the left. This is because F^+ corresponds to right-running waves, and so the left region is the origin of these waves. These interpolations are obtained using second degree polynomials and have third order accuracy. It is possible to combine the three of them and obtain a fifth order interpolation, and this is what the essentially non-oscillatory method (ENO) does. The WENO method adds an extra step so that interpolations are only made using regions where the flow is continuous, avoiding the degradation of the accuracy. To achieve this it is necessary to compute three smoothness indicators for the region considered, i.e.

$$\beta_1 = \frac{13}{12}\left(\phi_{i-2} - 2\phi_{i-1} + \phi_i\right)^2 + \frac{1}{4}\left(\phi_{i-2} - 4\phi_{i-1} + 3\phi_i\right)^2 \tag{2.38}$$

$$\beta_2 = \frac{13}{12}\left(\phi_{i-1} - 2\phi_i + \phi_{i+1}\right)^2 + \frac{1}{4}\left(\phi_{i-1} - \phi_{i-1}\right)^2 \tag{2.39}$$

$$\beta_3 = \frac{13}{12}\left(\phi_i - 2\phi_{i+1} + \phi_{i+2}\right)^2 + \frac{1}{4}\left(\phi_{i+2} - 4\phi_{i+1} + 3\phi_i\right)^2. \tag{2.40}$$

The next step will create ponderators that will make a particular approximation more important than the others if it is in a smoother region. These ponderators (after the next step) are identical to those obtained in the ENO method if the three regions are

equally smooth. The ponderators σ are

$$\sigma_1 = \frac{1}{10}\frac{1}{(\beta_1+\epsilon)^2} \tag{2.41}$$

$$\sigma_2 = \frac{3}{5}\frac{1}{(\beta_2+\epsilon)^2} \tag{2.42}$$

$$\sigma_3 = \frac{3}{10}\frac{1}{(\beta_3+\epsilon)^2}, \tag{2.43}$$

where ϵ is a small number (e.g. 10^{-30}) to avoid a division by zero. Also, the sigmas are normalized to create the final ponderators ω

$$\omega_1 = \frac{\sigma_1}{\sigma_1+\sigma_2+\sigma_3} \tag{2.44}$$

$$\omega_2 = \frac{\sigma_2}{\sigma_1+\sigma_2+\sigma_3} \tag{2.45}$$

$$\omega_3 = \frac{\sigma_3}{\sigma_1+\sigma_2+\sigma_3}. \tag{2.46}$$

Finally the approximated value of $\Phi_{i+1/2}$ can be obtained as

$$\Phi_{i+1/2} = \omega_1\hat{\Phi}^{(1)}_{i+1/2} + \omega_2\hat{\Phi}^{(2)}_{i+1/2} + \omega_3\hat{\Phi}^{(3)}_{i+1/2}. \tag{2.47}$$

This procedure is repeated with the adequate shift to obtain $\Phi_{i-1/2}$ and (eqn 2.34) is then used to obtain $\partial\phi/\partial x$.

A similar calculation can be employed to compute all four components of the derivative of the flux vector F^+. We must also compute the derivative of F^- following essentially the same procedure. It suffices to note that if we reverse the x-axis, the F^- waves are now running in the same direction of the reversed axis, so we can repeat exactly the same procedure. Of course, as there is no essential difference between the x and y axis, it is trivial that the same procedure applies also to $G^{\pm}$.

2.5.3 Finite differences

Sometimes the WENO method described above is not necessary or it is not practical. Simple problems do not need its adaptative behaviour, and it might be too slow for the application, because of the number of operations necessary to compute it. Finite differences methods are reasonable alternatives in these cases. I will consider two categories of points: regular and irregular. Points in the fluid bulk are in the former category, and points near interfaces are in the latter one.

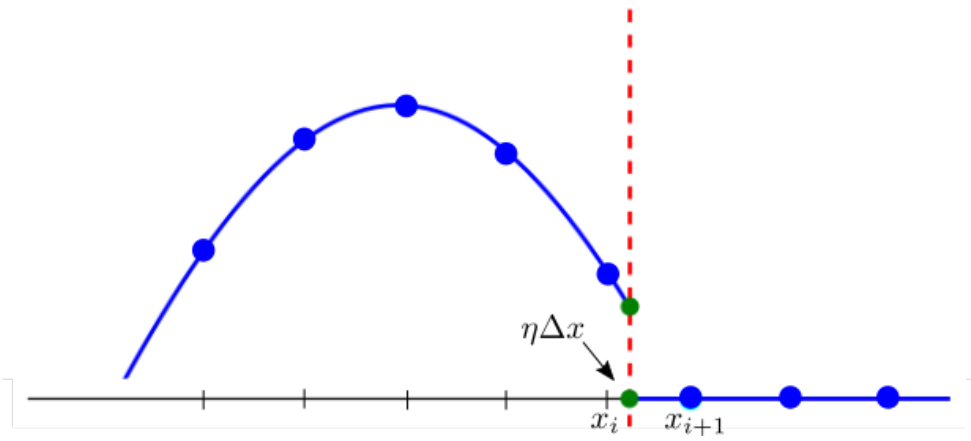


Figure 4 – The function in blue has a discontinuity at a point between x_i and x_{i+1}, marked by the dashed red line. The discontinuity is at a distance $\eta\Delta x$ from x_i. The points at either side of the discontinuity are marked green.
Source: By the author.

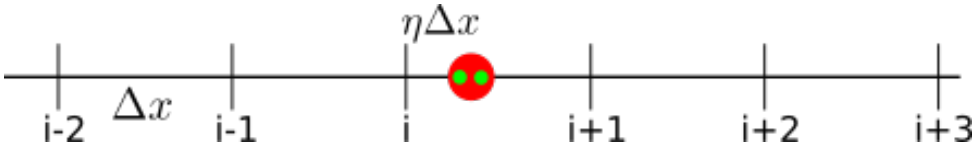


Figure 5 – A simplified view of (fig 4). The discontinuity is shown as a red dot with two green dots inside and the function is omitted. This representation will be useful later as it can be used for any type of discontinuity, such as those originating from immersed bodies and shock waves.
Source: By the author.

2.5.3.1 Regular points

For a flow variable ϕ, the second-order accurate first and second derivatives are calculated as

$$\left.\frac{\partial \phi}{\partial x}\right|_{x=x_i} = \frac{1}{2\Delta x}\left(\phi_{i+1} - \phi_{i-1}\right) + \mathcal{O}((\Delta x)^2) \tag{2.48}$$

$$\left.\frac{\partial^2 \phi}{\partial x^2}\right|_{x=x_i} = \frac{1}{(\Delta x)^2}\left(\phi_{i+1} - 2\phi_i + \phi_{i-1}\right) + \mathcal{O}((\Delta x)^2), \tag{2.49}$$

in the case of a regular point at $x = x_i$.

2.5.3.2 Irregular points

If the expressions in (eqn 2.48) and (eqn 2.49) are applied to the function in figure (fig 4) at x_{i+1} the result is non-zero in both cases, but clearly the function is constant to the right of the discontinuity, and the analytical result is zero for both derivatives. Also, the $\mathcal{O}((\Delta x)^2)$ truncation error derived for the expressions above is only valid where the function being derived is smooth. Indeed, the actual error when taking a derivative across a discontinuity is $\mathcal{O}(1)$. It is then necessary to derive special expressions to compute the derivatives near discontinuities.

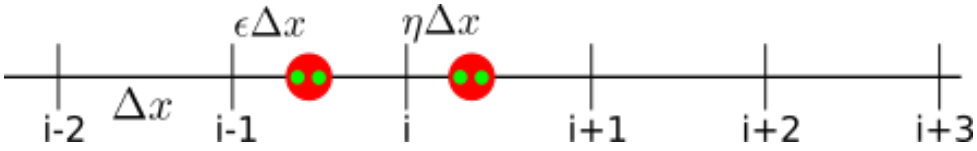


Figure 6 – Two discontinuities surrounding the point x_i. The first is at a distance $\epsilon\Delta x$ from x_{i-1}. The second is at a distance $\eta\Delta x$ from x_i.
Source: By the author.

Denoting the value to the left of the discontinuity as ϕ_d^l and the value to the right as ϕ_d^r, a first-order expression for the first derivative, at a point with a discontinuity to it's left, e.g. x_{i+1} in (fig 4), or to it's right, is given by

$$\left.\frac{\partial\phi}{\partial x}\right|_{x=x_i}^{l} = \frac{1}{(1+(1-\eta))\Delta x}\left(\phi_{i+1} - \phi_d^r\right) + \mathcal{O}(\Delta x) \tag{2.50}$$

$$\left.\frac{\partial\phi}{\partial x}\right|_{x=x_i}^{r} = \frac{1}{(1+\eta)\Delta x}\left(\phi_d^l - \phi_{i-1}\right) + \mathcal{O}(\Delta x). \tag{2.51}$$

The expression for a second derivative with a discontinuity to it's left is

$$\left.\frac{\partial^2\phi}{\partial x}\right|_{x=x_i}^{l} = \frac{2}{(2-\eta)(3-\eta)(\Delta x)^2}\left(\phi_d^r - (3-\eta)\phi_{i+1} + (2-\eta)\phi_{i+2}\right) + \mathcal{O}(\Delta x). \tag{2.52}$$

Another possibility is that two discontinuities surround the point where the derivative is being computed as in (fig 6). The expression obtained for the first derivative is then

$$\left.\frac{\partial\phi}{\partial x}\right|_{x=x_i} = \frac{1}{((1-\epsilon)+\eta)\Delta x}\left(\phi_{d_\eta}^l - \phi_{d_\epsilon}^r\right) + \mathcal{O}(\Delta x), \tag{2.53}$$

where $\phi_{d_\epsilon}^r$ is the value to the right of the discontinuity to the left of x_i, and $\phi_{d_\eta}^r$ is the value to the left of the discontinuity to the right of x_i. For stability reasons this derivative is only computed if $(1-\epsilon)+\eta > 1$, otherwise it is set to zero. The second derivative is always set to zero in this configuration.

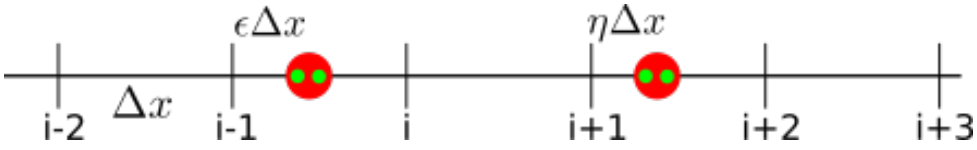


Figure 7 – Two discontinuities surrounding points x_i and x_{i+1}. This is similar to (fig 6) but with two regular points in between the discontinuities.
Source: By the author.

A final case to consider is shown in (fig 7). The first derivatives are computed using (eqn 2.50) and (eqn 2.51), but the second derivative must be computed using

$$\left.\frac{\partial^2\phi}{\partial x}\right|_{x=x_i} = \frac{4}{\left((2+\eta-\epsilon)\,\Delta x\right)^2}\left(\phi_{d_\epsilon}^r - 2\left(\frac{\eta+\epsilon}{2}\left(\phi_{i+1} - \phi_i\right)\right) + \phi_{d_\eta}\right). \tag{2.54}$$

2.5.3.3 Upwind methods

We can explore further[25] the flux-splitting method introduced in section 2.5.1. Let's consider only the flux component F^+ for a moment. As mentioned, the flux F^+ represent waves travelling from left-to-right along the x-axis. We can incorporate this physical interpretation into the numerical method by computing one-sided (or biased) derivatives of F^+, favoring the direction where the waves are coming from. This can be interpreted as a mean to enforce the correct dependency domain at a given point. Only the region from where the waves are coming (i.e. upwind from the point) will influence it. Of course the same arguments are equally applicable to the F^- component. The simplest approximation for the convective term derived from this idea is

$$\left.\frac{\partial F}{\partial x}\right|_{x=x_i} = \frac{F_i^+ - F_{i-1}^+}{\Delta x} + \frac{F_{i+1}^- - F_i^-}{\Delta x}. \tag{2.55}$$

2.5.4 Filtering

As mentioned in section 2.3, when applying FD methods based on central-differences, the numerical errors accumulate due to the lack of numerical dissipation, and high-frequency unphysical oscillations are generated. This effect is enhanced when the Reynolds number is high, as the system's dissipation is low. However, if a filtering operation is applied to the grid before each timestep, this unphysical oscillations can be controlled and a reasonable numerical solution can be obtained. Here, I chose to apply the filter introduced by Vasilyev [27] to control the oscillations. In this case, a filtered flow variable $\bar{\phi}$ is computed as

$$\bar{\phi}_i = -\frac{1}{16}\phi_{i-1} + \frac{1}{4}\phi_{i-1} + \frac{5}{8}\phi_i + \frac{1}{4}\phi_{i+1} - \frac{1}{16}\phi_{i+2} \tag{2.56}$$

$$\bar{\phi}_i = +\frac{1}{16}\phi_{i-1} + \frac{3}{4}\phi_i + \frac{3}{8}\phi_{i+1} - \frac{1}{4}\phi_{i+2} + \frac{1}{16}\phi_{i+3} \tag{2.57}$$

$$\bar{\phi}_i = +\frac{15}{16}\phi_i + \frac{1}{4}\phi_{i+1} - \frac{3}{8}\phi_{i+2} + \frac{1}{4}\phi_{i+3} - \frac{1}{16}\phi_{i+4}, \tag{2.58}$$

where (eqn 2.56) is to be used in the fluid bulk, and (eqn 2.57) and (eqn 2.58) near the domain boundaries. No special treatment is necessary near discontinuities due to shock-waves.

2.5.5 Shock detector

Two different methods were used to detect shocks. In this chapter I will only describe one of them, leaving the other to chapter 4. In 2016, Bambozzi and Pires [28] introduced a shock-detector capable of detecting discontinuities not only in the function, but in any of it's derivatives. The shock detector works by comparing the approximation obtained using all the grid points with the approximation obtained using, say, only the odd grid points. If the function is sufficiently smooth, both approximations will be very similar, but if there is a discontinuity they will be different. Let $F_{\Delta x}^{(n)}$ denote the $n-th$

derivative of function F approximated using all grid points, and $F^{(n)}_{2\Delta x}$ the same derivative approximated using only the odd numbered grid points. The shock detector* compares these approximations by computing

$$S_d = \log_2 \left(\frac{(2\Delta x)^2 \left|F^{(2)}_{2\Delta x}\right| + (2\Delta x)^3 \left|F^{(3)}_{2\Delta x}\right|}{(\Delta x)^2 \left|F^{(2)}_{\Delta x}\right| + (\Delta x)^3 \left|F^{(3)}_{\Delta x}\right|} \right). \tag{2.59}$$

The value of S_d is enough to infer the function continuity since

$$S_d \approx \begin{cases} 2, & \text{if } F \text{ is continuous up to it's 1st derivative} \\ p, & \text{if } F \text{ has a jump in it's } p < 2 \text{ derivative.} \end{cases} \tag{2.60}$$

2.5.6 Runge-Kutta integration

Runge-Kutta methods are a family of iterative methods, used to compute approximate solutions for the time-evolution of systems described by ordinary differential equations. This family includes both explicit and implicit methods. In particular, the Euler method and the backward Euler method of section 2.2.3 are examples of Runge-Kutta methods. The system time evolution in every simulation in this thesis is computed using the following third-order explicit method

$$\frac{\partial \phi}{\partial t} = \mathcal{F}(\phi(t), t) \tag{2.61}$$

$$k_1 = \mathcal{F}(\phi(t), t) \tag{2.62}$$

$$k_2 = \mathcal{F}(\phi(t) + \frac{\Delta t}{2} k_1, t + \frac{\Delta t}{2}) \tag{2.63}$$

$$k_3 = \mathcal{F}(\phi(t) - \Delta t k_1 + 2\Delta t k_2, t + \Delta t) \tag{2.64}$$

$$\phi(t + \Delta t) = \phi(t) + \frac{\Delta t}{6} \left(k_1 + 4k_2 + k_3\right). \tag{2.65}$$

Here $\mathcal{F}$ is a generic function. In particular, if we identify ϕ with U from (eqn 2.8) and use

$$\mathcal{F}(\phi(t), t) = -\frac{\partial F}{\partial x} - \frac{\partial G}{\partial y} + \frac{\partial F_D}{\partial x} + \frac{\partial G_D}{\partial y} \tag{2.66}$$

we can time-evolve the Navier-Stokes equations.

* This is not the shock-detector version used in[28], but the same results apply to this one.

3 IMMERSED-INTERFACE METHODS

In 1952, Hyman [29] introduced a method to solve elliptic equations using an arbitrary interface embedded in the domain with a Cartesian grid-based discretization. This method was further developed and is now well established in the works of Leveque[30], Li[31,32] and Zhong.[33] These fictitious domain methods present two advantages: first, numerical solvers can be made very efficient by leveraging the regularity from the underlying Cartesian grid and second, introducing new arbitrary interfaces is inexpensive in comparison with remeshing.

In 1971, Charles Peskin was studying the blood flow in the heart. He developed a numerical method [34] to simulate the presence of a moving, flexible immersed boundary to account for the presence of a leaflet in a heart valve. Peskin's method was the first work to apply the ideas from Hyman's work to fluid dynamical problems, creating a class of methods known as immersed-boundary methods (IBM). In this method, the boundary movement is coupled to the fluid movement using a distributed force that spreads the boundary influence to neighbouring points, making the interface diffuse. Other methods, such as those used by Ghias[35] and Karagiozis,[36]compute the effects of the interface at its exact position keeping the interface sharp. I will describe how these last two methods work in this chapter.

3.1 Boundary description

Consider the solid-fluid interface in (fig 8). The interface is described as series of marker points, shown as blue dots in the figure, that are later interpolated. In this thesis,

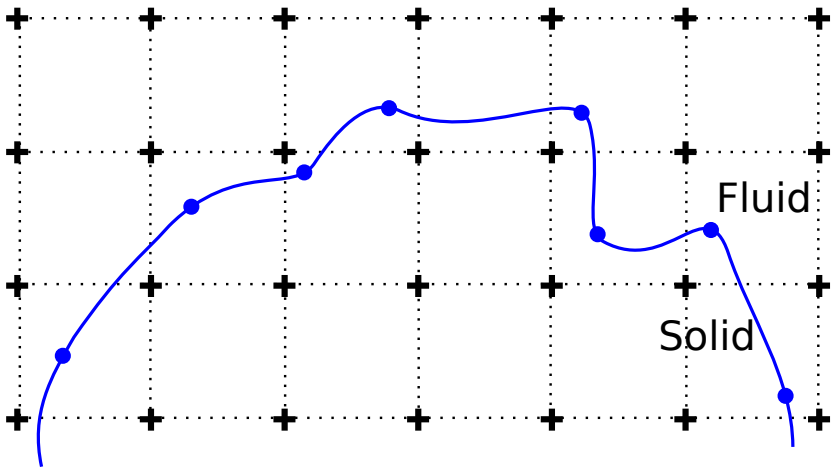


Figure 8 – A Cartesian grid with a blue curve representing the interface between a solid region and the fluid. The blue line is the result of the interpolation of the blue points.
Source: By the author.

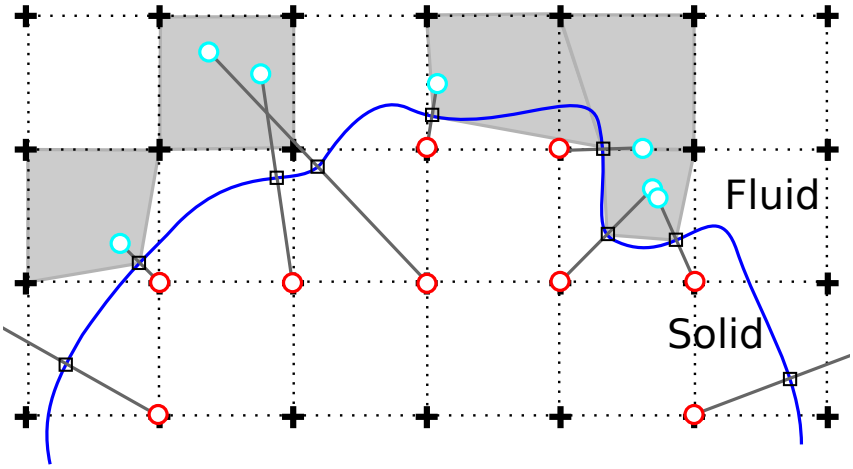


Figure 9 – The ghost-point construction. The cyan dots are the mirror image of the red dots with respect to the interface. The black squares are the midpoint between the red and cyan points. The shaded regions mark the regions in which the bilinear interpolations are constructed.
Source: By the author.

I have used cubic splines to obtain the blue curve. See ref.[37] for a detailed explanation on how splines are computed.

Both Karagiozis and Ghias methods are identical up to this point. I will now describe how each proceed from this point.

3.2 Ghias method

Ghias employs a ghost-point based method to account for immersed interfaces. The effect of the boundary is substituted by flow-variable values, attributed to some points inside the immersed bodies (the ghost points).

Before further discussing the advantages and disadvantages of this method, let's see in detail how the ghost-points values are obtained. Using (fig 9) as a reference:

- For each point inside the body with at least one neighbour in the fluid (the ghost points, marked as red circles) compute the point on the boundary that is closest to it (the boundary intercepts, marked as black squares);

- Compute the mirror image of the ghost point with respect to the boundary intercept (the image points, marked as cyan circles);

- For each image point, determine the four grid points closest to it. If one of such points is a ghost point, replace it for it's corresponding boundary intercept. The shaded quadrilateral regions are formed connecting the four points obtained for each image point in the figure;

- Compute the flow variables at the image points using a bilinear interpolation (detailed in section 3.2.1) constructed using the four points obtained in the last step;
- Use the values at the image points and their corresponding boundary intercepts to extrapolate the ghost-point values.

The method I have described has an important difference when compared to the original paper.[35] I substitute the ghost points for their boundary intercept at the third step. Ghias only makes this substitution when the neighbouring ghost point is the one corresponding to the image point, such as in the leftmost shaded region. He later uses an iterative method to solve the coupled equations. I find the method I employed easier to implement and it also produces good results.

This procedure keeps the computational grid regular everywhere making the use of irregular derivatives unnecessary, and makes it easy to use powerful methods—such as the WENO method—in combination with it. On the downside, however, computing suitable values for the ghost points needs a bilinear interpolation and an extrapolation, and there is also a significant increase in the overall computational geometry complexity. The computational geometry steps required to determine both the image points and the interpolation weights also makes the handling moving boundaries more involved.

Now that we know how to compute the flow variables at the ghost points, we can time-evolve the system. For each Runge-Kutta substep:

- Compute the flow variables at the grid points;
- Compute the fluxes at all grid points using derivatives for regular points;
- Time-evolve the solution using the computed fluxes.

Notice that it is not necessary to recompute the fluxes this time.

3.2.1 Bilinear interpolation

As mentioned at the end of chapter 2 I will describe the bilinear interpolation[35] used to compute the flow variables values at the image points. We consider a function of the form

$$\Phi(x, y) = a + bx + cy + dxy, \tag{3.1}$$

where the values of $a,\ b,\ c,\ d$ are obtained from the four image-point neighbours. Remember that some of the neighbours might be boundary-intercept points, and that at these points a Neumann boundary condition can be enforced. When that is the case, we do not have direct access to the function value at the neighbour point, but only to it's normal derivative at that point.

I will first describe how to obtain the coefficients when the function values are known at all four points. This situation arises when either all four neighbours of the image point are fluid points, or the boundary conditions being enforced are Dirichlet boundary conditions.
To this end, write

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 \\ 1 & x_4 & y_4 & x_4y_4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Phi_4 \end{bmatrix}, \tag{3.2}$$

where $x_i,\ y_i$ are the coordinates x and y of the image-point neighbours and Φ_i is the function value at that point. Once solved, (eqn 3.2) yields the four coefficients $a,\ b,\ c,\ d$ and it is possible to compute $\Phi(x, y)$ at the image point. If the boundary is static, the coordinates $x_i,\ y_i$ are fixed and the square matrix on the left hand side of (eqn 3.2) can be inverted once at the start of the computation and stored to be reused.

Let's now consider the case of Neumann boundary conditions. Suppose without loss of generality that the fourth neighbour of an image point is a boundary-intercept point and that the boundary condition

$$\vec{n} \cdot \vec{\nabla}\Phi = \Psi \tag{3.3}$$

is to be enforced. Careful manipulation of the equations will lead to the new system of equations

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 \\ 0 & n_x & n_y & n_xy_4 + n_yx_4 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \Psi \end{bmatrix} \tag{3.4}$$

where n_x, n_y are the components of the outward-normal unit vector at the boundary point. Once again, if the boundary is static the matrix can be inverted once at the beginning of the computation and stored to be reused.

3.3 Karagiozis method

Instead of constructing auxiliary points as in Ghias method, Karagiozis utilizes the boundary explicitly in his method. Karagiozis method begins marking the points where the interpolated interface crosses the grid lines as shown in (fig 10). Then, it enforces Dirichlet or Neumann boundary conditions on the flow variables at these points, and that is how the immersed interface is accounted for in the simulation.

Dirichlet boundary conditions are used to enforce the no-slip condition for viscous flows, and to fix the body temperature. Remember from (fig 5) that each discontinuity has an internal structure, so enforcing Dirichlet boundary conditions is as simple as setting

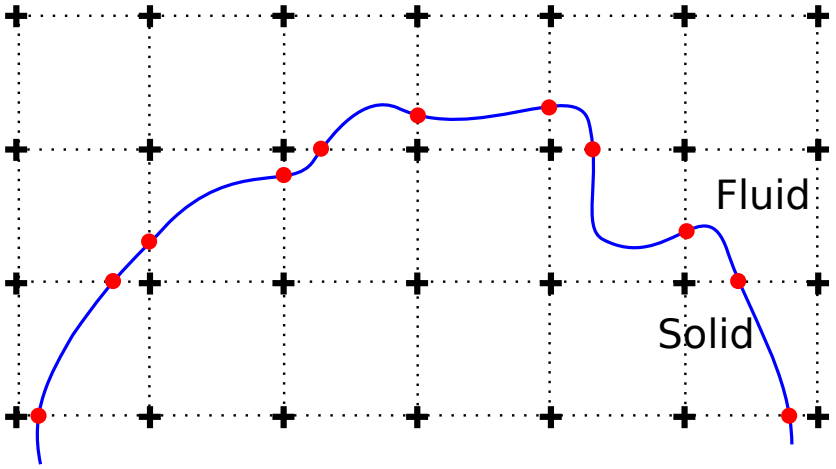


Figure 10 – The same interface as in (fig 8), but with the crossings between the interface and the grid lines marked in red. These points are the same type of discontinuity represented in (fig 5), but I have omitted the internal green dots for simplicity.
Source: By the author.

the internal values appropriately. Density values at the boundary are obtained using a linear extrapolation from the neighboring points.

Consider the discontinuity in (fig 5). To compute the density ρ to the left (ρ_d^l) of and to the right (ρ_d^r) of the discontinuity we use

$$\rho_d^l = \rho_i + (\rho_i - \rho_{i-1})\ \eta \tag{3.5}$$

$$\rho_d^r = \rho_{i+1} + (\rho_{i+1} - \rho_{i+2})\ (1 - \eta). \tag{3.6}$$

Now that we have a procedure to enforce the boundary conditions we can compute the time evolution of the system. For each Runge-Kutta substep:

- Enforce all boundary conditions;
- Compute the fluxes at all grid points using derivatives for regular points;
- Recompute the fluxes at points near immersed interfaces using derivatives for irregular points;
- Time-evolve the solution using the computed fluxes.

In this chapter I showed two methods to describe solid bodies embedded in fluid using immersed-interfaces that are suitable for numerical simulations of fluids. In the next chapter I will introduce another type of discontinuity: shock waves.

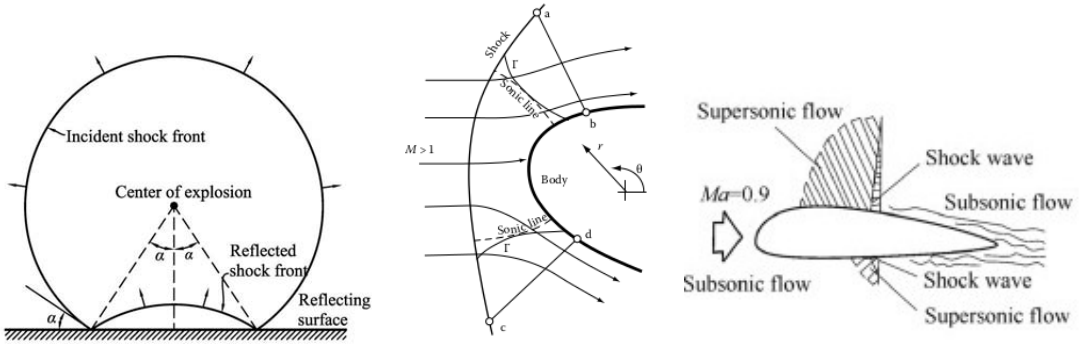
4 SHOCK WAVES

One way to classify flows is in subsonic and supersonic flows. Flows are subsonic if the magnitude of the velocities is lower then the speed of sound, and supersonic otherwise. One interesting effect can manifest in the transition between subsonic and supersonic: shock wave formation. A variety of situations can create the right circumstances for formation of shock waves. The rapid gas expansion from an exploding agent, a sudden change in the flow direction caused by a blunt body, and the recompression midway through an airplane wing are some examples, as shown in (fig 11).

As mentioned in section 2.4, when shock waves appear during a simulation they must be adequately treated. In this section, I will first introduce the Rankine-Hugoniot jump conditions, a compatibility relation between the two sides of a discontinuity. Then, I will discuss two classes of numerical methods—conservative and non-conservative—and I will combine them using the shock detector from section 2.4. In particular, the shock detector will be used to determine the regions where a conservative method must be used. Finally, I will show a completely different approach to handle shock waves, i.e. the so-called shock-fitting method.

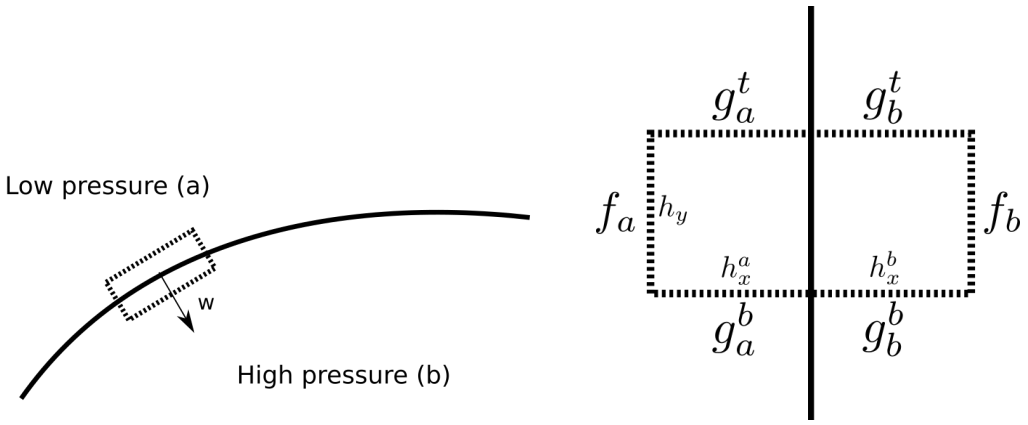
4.1 Rankine-Hugoniot jump conditions

In this section I will derive the main relation in the field of shock-waves: the Rankine-Hugoniot jump conditions. Shock-waves manifest as discontinuities in the flow variables. This is a problem for the differential form of the Navier-Stokes equations, so we must evoke the integral form of the equations. I will make a simplifying consideration by



(a) An explosion induced shock. (b) Bow shock near a blunt body. (c) A recompression shock on an airplane wing.

Figure 11 – Different causes for shock-wave formation. In every case, flow variables become discontinuous at the shock-layer.
Source: (a) SHIN [38] (b) SALAS [39] (c) WU [40]



(a) Schematic shock wave (b) Detailed view

Figure 12 – A shock wave separating high- and low-pressure regions. In the vicinity of the dashed rectangle we consider only the flow component normal to the shock interface. The shock-wave velocity w is positive in the normal direction facing the high-pressure region. In (b) we have a more detailed view of the dashed rectangle, including the sizes $h_y,\ h_x^a,\ h_x^b$ of each side.
Source: By the author.

ignoring the dissipative terms at the position of the shock. This is justified, considering that shock-waves are found in flows with a high Reynolds number and therefore the influence of the dissipative terms is relatively small. From now on in this chapter I will take the dissipative terms q_i and τ_{ij} to be zero.

Notice that all components of the Navier-Stokes equations can be written as

$$\frac{\partial \phi}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = \frac{\partial \phi}{\partial t} + \vec{\nabla} \cdot \vec{H} = 0 \tag{4.1}$$

where ϕ, f and g are the components of the vectors in (eqn 2.8), (eqn 2.9) and (eqn 2.10) respectively, and $\vec{H} = f\hat{x} + g\hat{y}$. Consider the region marked by the dashed rectangle in (fig 12) to be infinitesimal and that the only discontinuity in the flow is due to the shock wave. Using a coordinate system with the x axis normal to the shock interface, near the dashed rectangle, we write the integral form of the equation as

$$\begin{aligned} \frac{d}{dt} \iint_{rectangle} \phi \; dx \; dy &= - \iint_{rectangle} \vec{\nabla} \cdot \vec{H} \; dx \; dy \\ &= - \left[(f_b - f_a)\, h_y + \left(\left(g_a^t - g_a^b \right) h_x^a + \left(g_b^t - g_b^b \right) h_x^b \right) \right] \\ &\quad + \mathcal{O}\left(h^2 \right), \end{aligned} \tag{4.2}$$

where I employed the divergence theorem to go from the first to the second line and h, in the error term, stands for anyone of the lengths. Also, the lengths h_x^a and h_x^b are the rectangle's width to either side of the shock, and h_y it the rectangle's height. The subscripts a and b in the flux components f and g indicate if they are computed in the low- or high-pressure region, respectively. Also, the superscripts t and b in g indicate if the flux

is computed on the top or bottom side of the rectangle respectively. We now manipulate the left hand side of (eqn 4.2). If we keep the outer limits of the rectangle fixed, i.e h_y and $h_x^a + h_x^b$ are constant, but allow for the shock-wave to move we can write

$$\begin{aligned}\frac{d}{dt}\iint_{rectangle}\phi\ dx\ dy &= \frac{d}{dt}\left[\left(\phi_a h_x^a h_y + \phi_b h_x^b h_y\right) + \mathcal{O}\left(h^3\right)\right] \\ &= \left(\frac{d\phi_a}{dt}h_x^a + \phi_a\frac{dh_x^a}{dt}\right. \\ &\quad \left. + \frac{d\phi_b}{dt}h_x^b + \phi_b\frac{dh_x^b}{dt} + \mathcal{O}\left(h\right)\right) h_y.\end{aligned} \tag{4.3}$$

Also, notice that

$$\frac{dh_x^a}{dt} = -\frac{dh_x^b}{dt} = w. \tag{4.4}$$

Combining 4.2, 4.3 and 4.4, we obtain

$$\begin{aligned}w\left(\phi_a - \phi_b\right) + \frac{d\phi_a}{dt}h_x^a + \frac{d\phi_b}{dt}h_x^b &= \left(f_a - f_b\right) - \frac{g_a^t - g_a^b}{h_y}h_x^a - \frac{g_b^t - g_b^b}{h_y}h_x^b + \mathcal{O}(h) \\ &= \left(f_a - f_b\right) - \frac{dg_a}{dy}h_x^a - \frac{dg_b}{dy}h_x^b + \mathcal{O}(h),\end{aligned} \tag{4.5}$$

where we took the limit $h_y \to 0$ and used the continuity of g to identify the terms on the right hand side as derivatives. Taking the limit as h_x^a and h_x^b go to zero, we finally obtain

$$w\left(\phi_a - \phi_b\right) = f_a - f_b. \tag{4.6}$$

This is known as the Rankine-Hugoniot jump condition of a conservation law. Replacing the specific values of ϕ and f found in Euler equations, we obtain

$$w\left(\rho_a - \rho_b\right) = \rho_a u_a - \rho_b u_b \tag{4.7}$$

$$w\left(\rho_a u_a - \rho_b u_b\right) = \left(\rho_a u_a^2 + p_a\right) - \left(\rho_b u_b^2 + p_b\right) \tag{4.8}$$

$$w\left(\rho_a v_a - \rho_b v_b\right) = \rho_a u_a v_a - \rho_b u_b v_b \tag{4.9}$$

$$w\left(E_a - E_b\right) = \left(E_a + p_a\right)u_a - \left(E_b^2 + p_b\right)u_b. \tag{4.10}$$

By multiplying 4.7 by v_b and subtracting 4.9 from it, we find

$$w\rho_a\left(v_b - v_a\right) = \rho_a u_a\left(v_b - v_a\right) \tag{4.11}$$

$$v_a = v_b, \tag{4.12}$$

where we have to impose the condition 4.12 in order to make (eqn 4.11) valid in general. The combination of (eqn 4.7), (eqn 4.8), (eqn 4.10) and (eqn 4.12) are the Rankine-Hugoniot jump conditions for Euler equations. They are the fundamental relations in the shock-wave theory and will be the basis of the shock-fitting method described in section 4.3

Had we used the alternative form of the Navier-Stokes presented in 2.14, 2.15 and 2.16, we would have obtained the equivalent jump conditions[53]

$$c_b = c_a \frac{\sqrt{(\gamma M^2 - \delta)(1 + \delta M^2)}}{(1+\delta) M} \tag{4.13}$$

$$u_b = u_a + c_a \frac{1 - M^2}{(1+\delta) M} \tag{4.14}$$

$$S_b = S_a + \frac{1}{2\delta\gamma} \left[\ln \frac{\gamma M^2 - \delta}{1+\delta} - \gamma \ln \frac{(1+\delta) M^2}{1 + \delta M^2} \right] \tag{4.15}$$

$$v_b = v_a \tag{4.16}$$

$$M = \frac{u_a - w}{c_a}, \tag{4.17}$$

where M is the relative shock Mach number.

4.2 Numerical methods to handle shock waves

4.2.1 Conservative and non-conservative methods

Terminology can be confusing when using this classification. The components of the Navier-Stokes equations are conservation laws, so what is meant by a non-conservative method? Conservative methods are derived from the equations put in divergence form, while non-conservative methods are derived from the equations after the chain rule has been applied to expand the derivatives into more terms. The methods derived from the divergence form automatically guarantees the conservation of fluxes through a particular control volume, and are thus called conservative methods. On the contrary, this is not exactly guaranteed when the method is derived from the expanded equations, and that is why they are called non-conservative methods.

To compare conservative and non-conservative methods I will consider the 1-D Burgers equation, which is a limiting case of the Navier-Stokes equations, i.e.

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0 \tag{4.18}$$

$$\frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial u^2}{\partial x} = 0. \tag{4.19}$$

Analytically, (eqn 4.18) and (eqn 4.19) are equivalent. However, (eqn 4.18) is in non-conservative form, and (eqn 4.19) is in conservative form. Moreover, consider the following upwind discretizations (for $u \geq 0$)

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + u_i^n \frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \tag{4.20}$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{1}{2} \frac{(u_i^n)^2 - \left(u_{i-1}^n\right)^2}{\Delta x} = 0. \tag{4.21}$$

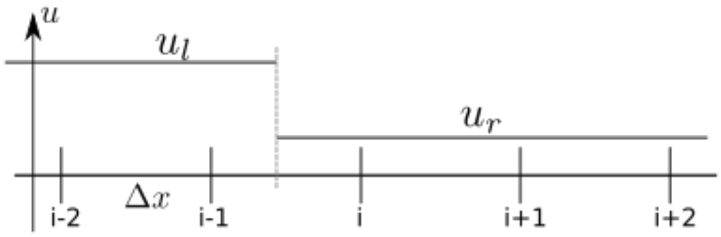


Figure 13 – Discretization of a step function. The discontinuity between x_{i-1} and x_i makes it impossible for $u^n_{i-1} = u^n_i$ to apply, even when the grid is refined.
Source: By the author.

It is clear that it is impossible to reduce one equation to the other. Let's now manipulate the second term on the left-hand side of (eqn 4.21):

$$\frac{1}{2}\frac{(u^n_i)^2 - \left(u^n_{i-1}\right)^2}{\Delta x} = \frac{1}{2}\frac{\left(u^n_i + u^n_{i-1}\right)\left(u^n_i - u^n_{i-1}\right)}{\Delta x}$$
$$= \bar{u^n_{i+\frac{1}{2}}}\frac{u^n_i - u^n_{i-1}}{\Delta x} \tag{4.22}$$
$$\bar{u^n_{i+\frac{1}{2}}} = \frac{u^n_i + u^n_{i-1}}{2}. \tag{4.23}$$

Thus, the term obtained in (eqn 4.21) equals to the second term in (eqn 4.20) if and only if $u^n_i = u^n_{i-1}$. Let us note that, for a continuous function,

$$u^n_{i-1} = u^n_i - \Delta x \left.\frac{\partial u}{\partial x}\right|_{x=x_i} + \mathcal{O}\left((\Delta x)^2\right). \tag{4.24}$$

Therefore, even though (eqn 4.20) and (eqn 4.21) are inequivalent, they will converge to the same solution in the limit $\Delta x \to 0$ if u is continuous. On the other hand, if u is discontinuous this argument is no longer valid. Indeed, looking at the simple example in (fig 13) it is clear that, even taking the continuum limit, u^n_{i-1} and u^n_i will never be equal. Analytically, the derivative in (eqn 4.24) becomes ill-defined at the discontinuity in the limit $\Delta x \to 0$, so it is not possible to justify the equality of (eqn 4.21) and (eqn 4.20).

As I have shown, in the presence of a discontinuity, discretizations (eqn 4.20) and (eqn 4.21) give different results, and so it is natural to ask which of them is the correct one in this case. To answer this question we must use another physical condition to justify a choice.* In this case, we look for a violation of the second-law of Thermodynamics. It turns out that (eqn 4.21) is the discretization that yields the physically correct solution and so we should always apply it near discontinuities. This is a general result: the physically relevant solutions of hyperbolic partial differential equations are obtained from the conservative form of the equations.[41] Another natural question to ask is where does the solution obtained using (eqn 4.20) comes from. Both solutions are in fact *weak* solutions of (eqn 4.18). For a

* This has the same rationale as evoking causality to choose the retarded potentials as the physically acceptable solution in electrodynamics.

more in-depth discussion on the topic of weak solutions, refer to the excellent book by Randal J. LeVeque.[41]

It is well known that conservative methods effectively reduce to first-order accuracy at the shock-position.[42] As a consequence, in this case, the effects of numerical dissipation are larger and they smooth out the discontinuity across a few grid points. To describe this effect, it is said that the shock is *captured* within a narrow region, motivating another name for this type of method: shock-capturing methods. As a final check, we can derive the Rankine-Hugoniot jump condition (eqn 4.6) for Burgers' equation and obtain the speed of the discontinuity:

$$w\left(u_l - u_r\right) = \frac{u_l^2 - u_r^2}{2} \tag{4.25}$$

$$w = \frac{u_l + u_r}{2}. \tag{4.26}$$

If we use (eqn 4.21) in a numerical simulation we verify that the obtained shock speed is indeed the one predicted by (eqn 4.26).

4.2.2 The role of shock detectors

In the previous section I have shown that conservative methods must be used near a shock-wave. Yet, sometimes, it can be useful to employ a non-conservative method, such as the finite-difference schemes of section 2.5.3, to take advantage of their lower computational cost. By using the shock detector discussed in section 2.4 we can selectively apply conservative methods (e.g. WENO), near shock-waves, and a finite-difference method, everywhere else. In the simulations I present in chapter 5 I also applied the filter introduced in section 2.3 to control unphysical oscillations.

The overall algorithm employed can be schematized as follows:

- Before each timestep:
  - Use shock detector to find discontinuities in the flow;
  - Filter all flow variables;
- At all Runge-Kutta substeps:
  - Compute fluxes using finite-differences discretization, in shock-free regions, and WENO, near shock waves.

4.3 Principles of shock-fitting

Shock-capturing methods became popular for the simplicity of their implementation, even though the shock is spread over a region which is orders of magnitude larger than the actual physical shock. As computational power became less of an issue nowadays, people

can refine more and more their grids in order to obtain better approximation to the real solutions. Nevertheless, using these methods the numerical representation of shock-waves will always be much wider than they physically are. In fact, the Navier-Stokes equations themselves are only valid when the fluid can be described as a continuous medium, while shock waves in air are only about $200nm$ thick[43]—comparable to one mean free path of air molecules. Thus, it would not be justifiable to use the Navier-Stokes equations in a grid thin enough to describe a shock. Moreover, shock-capturing methods (specially high-order methods such as WENO) propagate waves across the shock, carrying information from the subsonic region into the supersonic region, violating physical causality.[44] This clearly does not happen in the exact solution and it would be desirable that numerical methods preserve this property for the sake of physical correctness.

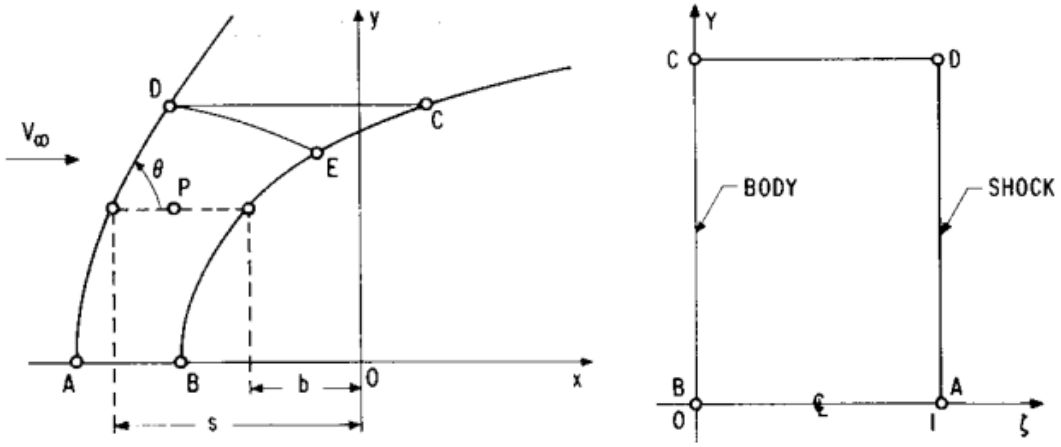
These problems, however, are not enough to dismiss shock-capturing methods. Indeed, the shock thickness is almost irrelevant for the shock movement, and the information leakage across the shock is usually small. But these issues raise the following question: is there a numerical method which treats shocks as localized discontinuities and enforces the appropriate Rankine-Hugoniot jump conditions exactly? The answer is yes. Shock-fitting methods have been around since 1960s in works by Gino Moretti[2] and they address both issues present in shock-capturing methods.

There are two types of shock-fitting methods that I will detail next: boundary shock fitting, and floating shock fitting.

4.3.1 Boundary shock-fitting

In the 1966 paper by Gino Moretti and Michael Abbett,[2] a novel technique for solving the general blunt-body problem in high-Mach number flows was presented. To this end, they consider the problem represented in panel (a) of figure (fig 14), where the segment BC represents the surface of a solid body which is immersed in a fluid with freestream velocity V_∞. The line AD represents the bow-shock that forms ahead of the body. As a first step to obtain a numerical solution, the physical domain is mapped into the computational domain $ABCD$ shown in panel (b). Then, in order to compute both the bow-shock position and the flow variables in the $ABCD$ domain at the steady state, one usually employs an iterative procedure. In the boundary shock-fitting method, at each iteration the bow-shock is considered as a supersonic inlet in which the Rankine-Hugoniot jump conditions are enforced exactly. The results obtained by Moretti and Abbett, reported in (fig 15), showed excellent agreement with the best available simulations and experiments at the time.[1]

Many authors used boundary shock fitting in different applications since its inception. For example, Kutler[45] and Marconi[46] used it to study supersonic airplanes, Zhong[47,48] studied boundary layer receptivity to perturbations, and Romick[49] considered shock-waves



(a) A blunt body in a high-Mach number flow. (b) Computational domain representation after a coordinate transformation.

Figure 14 – Panel (a) shows a solid body (segment BC) in a high-Mach number flow ($V_\infty/c_\infty > 1$). The body abscissa b is a given fixed function. The shock abscissa s is a function of time and y that we want to determine. A coordinate transformation $f:\ (x,\ y) \to (\zeta,\ Y)$ maps the physical domain in (a) into the computational domain in (b).
Source: (a) and (b) ABBETT [2]

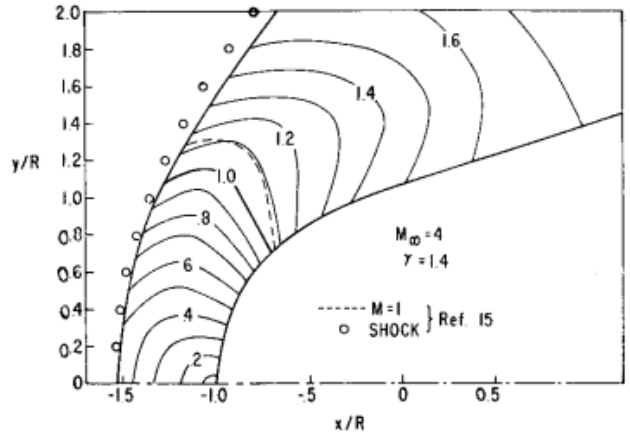


Figure 15 – Solution for the flow around a circular cylinder body obtained using the boundary shock-fitting method (the x and y coordinates are rescaled by $R = \ln \rho$). The dots and the dashed line were obtained by Belotserkivskii[1] and were used for comparison by Abbett and Moretti.[2]
Source: ABBETT [2]

emanating from high-explosives detonations. Modern implementations of this method, see Zhong[48] and Romick[49], introduced improved grid descriptions, new ways to compute shock velocities using the Rankine-Hugoniot jump conditions, and can also simulate the full time evolution of the system. Yet, they are conceptually the same as the one implemented by Abbett and Moretti in 1966. In Rawat and Zhong's paper [50] there is a comparison of various boundary shock-fitting methods.

Let us stress that boundary shock fitting solves two important problems that shock-capturing methods have. First, the shock is sharp, as there is no need for numerical dissipation across the domain boundary, and the correct jump conditions are applied exactly. Second, there is no computational power wasted on the supersonic region of the flow, where all flow variables are constant. Computations are only made where necessary.

It is not always possible to identify shock waves with a domain boundary though. A second type of shock fitting was also developed by Moretti[51] and that is the topic of the next section.

4.3.2 Floating shock-fitting

Soon after developing the boundary shock-fitting method, Moretti went further and developed methods capable of tracking the shock discontinuities. First, in 1969, he studied piston-driven flows,[51] where a gas initially at rest in a tube is suddenly compressed by a moving piston. These settings are simple enough to allow for analytical solutions which can then be compared to the numerical results. Also the resulting flow itself is relatively simple: two regions separated by a single shock-wave. This means the numerical method employed can also be relatively simple. Moretti ends this paper stating that the treatment of flows with multiple shocks will be found in another paper. Indeed, two years later he published the aptly named paper *Complicated one-dimensional flows*, [52] where he showed that his technique could handle multiple shocks, shock-shock interactions, shock-reflections and contact discontinuities without introducing any oscillations. One of his results is presented in (fig 16), where two moving pistons create colliding shock-waves.

Moretti continued to improve the shock-fitting method until he retired from academic work. The last section in this chapter presents a simplified version of his 1983 and 1987 papers on this technique.[53,54]

4.4 A shock-fitting method for integrating Euler equations

The mathematical structure of Euler equations is richer than what we have explored so far. Indeed, I mentioned that flux-splitting procedures represent waves travelling in different directions, but this fact was not further used. Next I will explore the method of characteristics for Euler equations.

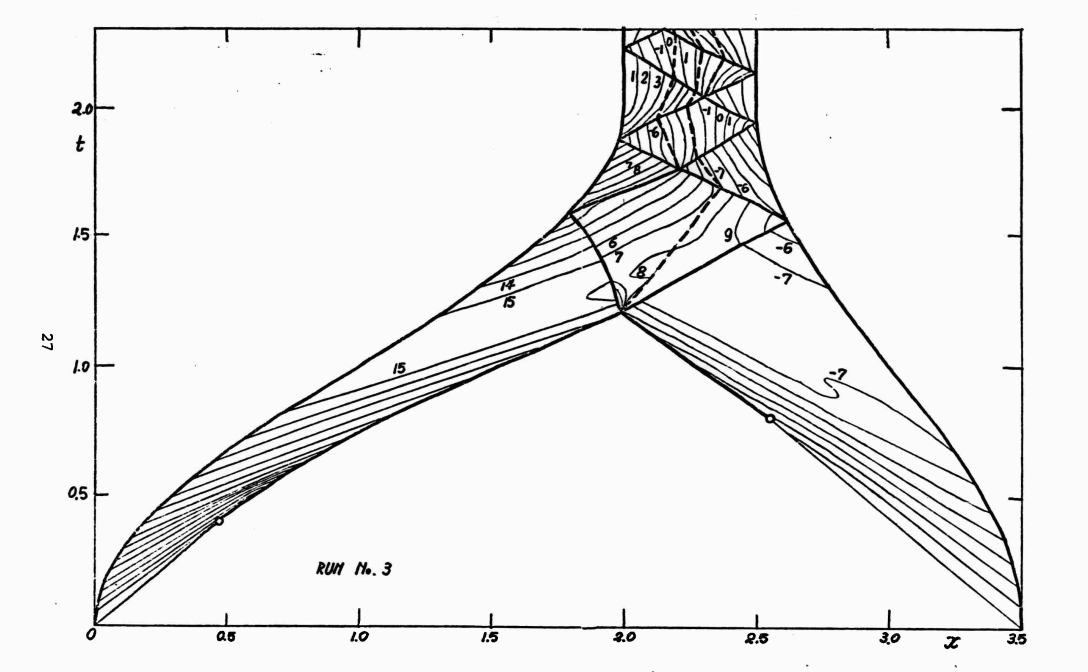


Figure 16 – Two moving pistons generating shock-waves. The thick solid lines show the pistons paths and the shock paths. The point in the space-time diagram where shock-waves are formed can be found analytically and are marked as hollow circles. The dashed lines are the contact discontinuities and the thin solid lines are lines of constant velocity.
Source: MORETTI[52]

4.4.1 Lambda scheme

The lambda scheme[55] was introduced in 1979 by Moretti as an alternative to MacCormack's method.[56] I will not use the lambda scheme in any simulations reported in chapter 5, but it is useful to understand how the mathematical structure of Euler equations can be explored to construct a numerical method.

As shown below, the lambda scheme is based on the Euler equations written in terms of Riemann variables, so I start by deriving them. Consider the one-dimensional form of (eqn 2.14), (eqn 2.15) and (eqn 2.16) with zero dissipation ($q_i = \tau_{ij} = 0$):

$$\frac{1}{\delta}\frac{\partial c}{\partial t} + \frac{1}{\delta}u\frac{\partial c}{\partial x} - c\frac{\partial S}{\partial t} - cu\frac{\partial S}{\partial x} = 0 \tag{4.27}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{c}{\delta}\frac{\partial c}{\partial x} - c^2\frac{\partial S}{\partial x} = 0 \tag{4.28}$$

$$\frac{\partial S}{\partial t} + u\frac{\partial S}{\partial x} = 0. \tag{4.29}$$

By defining

$$R_1 = \frac{c}{\delta} + u, \quad R_2 = \frac{c}{\delta} - u, \quad \lambda_1 = u + c, \quad \lambda_2 = u - c, \tag{4.30}$$

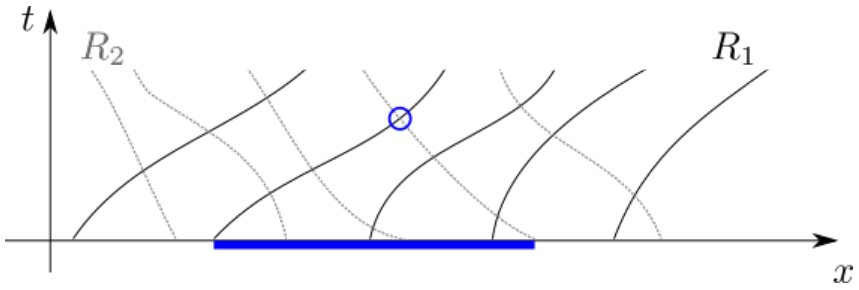


Figure 17 – Schematic representation of Riemann variables following the characteristics lines in the space-time diagram. Black solid lines represent constant R_1 and gray dashed lines represent constant R_2. The blue line at the bottom is the dependency domain of the blue circle.
Source: By the author.

we can rewrite (eqn 4.27), (eqn 4.28) and (eqn 4.29) as

$$\frac{\partial R_1}{\partial t} + \lambda_1 \left(\frac{\partial R_1}{\partial x} - c\frac{\partial S}{\partial x} \right) - c\frac{\partial S}{\partial t} = 0 \tag{4.31}$$

$$\frac{\partial R_2}{\partial t} + \lambda_2 \left(\frac{\partial R_2}{\partial x} - c\frac{\partial S}{\partial x} \right) - c\frac{\partial S}{\partial t} = 0 \tag{4.32}$$

$$\frac{\partial S}{\partial t} + u\frac{\partial S}{\partial x} = 0. \tag{4.33}$$

The R_1, R_2 in (eqn 4.30) are known as Riemann variables. To understand their meaning, let's first consider a simple case. For an isentropic flow (constant S) we obtain

$$\frac{\partial R_1}{\partial t} + \lambda_1 \frac{\partial R_1}{\partial x} = 0 \tag{4.34}$$

$$\frac{\partial R_2}{\partial t} + \lambda_2 \frac{\partial R_2}{\partial x} = 0 \tag{4.35}$$

with (eqn 4.33) being trivially satisfied. Then the Riemann variable $R_{1(2)}$ is simply being convected by the term $\partial R_{1(2)}/\partial x$. This implies that R_1 and R_2 are constant along the characteristic lines

$$\frac{dx}{dt} = \lambda_1, \quad \frac{dx}{dt} = \lambda_2, \tag{4.36}$$

respectively. In this case R_1 and R_2 are called Riemann invariants. As we started with three equations, we must also obtain three groups of characteristics. The third of them are the particle paths defined by

$$\frac{dx}{dt} = u, \tag{4.37}$$

along which S is constant (in fact, in this simple case S is constant everywhere, at all times).

Consider the blue circle in (fig 17). The values of the Riemann variables at the blue circle are entirely determined by the initial conditions at the domain region marked by the blue line. The Riemann variables are therefore the fundamental information carriers in Euler equations. This is true even when the flow is no longer isentropic. By computing

the Riemann variables $R_1,\ R_2$ at any time after the initial conditions we can reconstruct the original flow variables $c,\ u$. The paths where $R_1,\ R_2$ are constant are not obtained using the simple expressions in (eqn 4.36), but the interpretation remains the same: the Riemann variable $R_{1(2)}$ is convected by the non-linear term $\partial R_{1(2)}/\partial x - c\ \partial S/\partial x$ in the presence of the source term $c\ \partial S/\partial t$.

The core of the lambda scheme is that it accounts for the above results when discretizing the equations of motion (eqn 4.31), (eqn 4.32) and (eqn 4.33). The scheme has the form of a predictor-corrector method and so it is divided in two steps. To compute a time-step from a time-level n to $n+1$, for the case $u > 0$, the first step in the lambda scheme is computing the values f_i^n defined as

$$\begin{aligned}(f_1)_i^n = &-\frac{1}{2\Delta x}\Big(\left[(\lambda_1)_i^n + (\lambda_1)_{i-1}^n\right]\left[(R_1)_i^n - (R_1)_{i-1}^n\right] \\ &\quad - \left[(c\lambda_1)_i^n + (c\lambda_1)_{i-1}^n\right]\left[S_i^n - S_{i-1}^n\right]\Big) \qquad (4.38)\\ (f_2)_i^n = &-\frac{1}{2\Delta x}\Big(\left[(\lambda_2)_j^n + (\lambda_2)_{j'}^n\right]\left[(R_2)_j^n - (R_2)_{j'}^n\right] \\ &\quad - \left[(c\lambda_2)_j^n + (c\lambda_2)_{j'}^n\right]\left[S_j^n - S_{j'}^n\right]\Big) \qquad (4.39)\\ (f_3)_i^n = &-\frac{1}{2\Delta x}\left[u_i^n + u_{i-1}^n\right]\left[S_i^n - S_{i-1}^n\right], \qquad (4.40)\end{aligned}$$

where the sites $i,\ j,\ j'$ are such that $j = i$ if $\lambda_2 > 0$, $j = i+1$ if $\lambda_2 < 0$ and $j' = j-1$. Similar expressions are obtained when $u < 0$. We then update the variables to an intermediate time-level $n + 1/2$ using

$$S_i^{n+1/2} = S_i^n + (f_3)_i^n \frac{\Delta t}{2} \qquad (4.41)$$

$$(R_1)_i^{n+1/2} = (R_1)_i^n + \left[c_i^n\,(f_3)_i^n + (f_1)_i^n\right]\frac{\Delta t}{2} \qquad (4.42)$$

$$(R_2)_i^{n+1/2} = (R_2)_i^n + \left[c_i^n\,(f_3)_i^n + (f_2)_i^n\right]\frac{\Delta t}{2}. \qquad (4.43)$$

This allows to compute $f_i^{n+1/2}$ using the intermediate values just obtained. Next, we compute the values $F_i^{n+1/2}$ to be used in the correction step considering the relations

$$(F_1)_i^{n+1/2} = 2\,(f_1)_i^{n+1/2} - (f_1)_{i-1}^n \qquad (4.44)$$

$$(F_2)_i^{n+1/2} = 2\,(f_2)_i^{n+1/2} - (f_2)_j^n \qquad (4.45)$$

$$(F_3)_i^{n+1/2} = 2\,(f_3)_i^{n+1/2} - (f_3)_{i-1}^n, \qquad (4.46)$$

where $j = i-1$ if $\lambda_2 > 0$ and $j = i+1$ if $\lambda_2 < 0$. Finally, we use (eqn 4.41), (eqn 4.42) and (eqn 4.43) with $n + 1/2$ instead of n, and $F_i^{n+1/2}$ instead of f_i^n, to obtain the updated variables S^{n+1}, R_1^{n+1} and R_2^{n+1}.

4.4.2 Shock computations

In (fig 17) one can see that characteristics lines of constant R_1 or R_2 never crosses another line of the same family. Of course, it is inadmissible for a point in a continuous

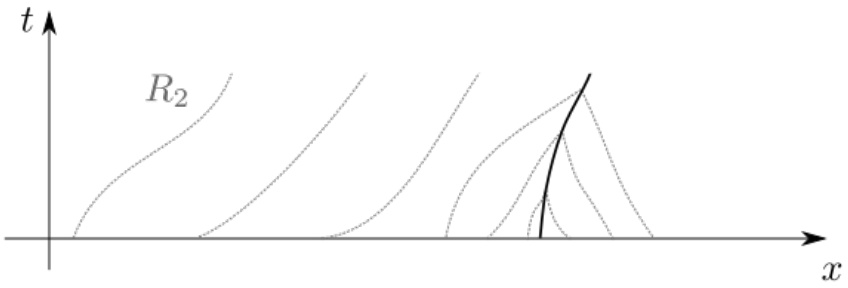


Figure 18 – Schematic representation of R_2 Riemann variable following the characteristics lines in the space-time diagram near a shock. The black solid line represent the shock path at the transition between a supersonic region (to the left) and a subsonic region (to the right).
Source: By the author.

flow to simultaneously have two different R_1 or two different R_2 values. On the contrary, at the position of a shock-wave we do have different R_1 and R_2 values at either side as shown in (fig 18) because continuity is lost. By virtue of the Rankine-Hugoniot jump conditions, the shock-wave speed and the values of the Riemann variables on both sides of the shock are related. This relation can be used to efficiently compute the shock velocity w. To this end, consider the conveniently defined dimensionless quantity

$$\Sigma \;=\; \delta\frac{R_{2b}-R_{2a}}{c_a}+1 \;=\; \delta\frac{R_{2b}+u_a}{c_a}, \tag{4.47}$$

where $\delta=(\gamma-1)/2$, u_a is the flow speed on the low-pressure side of the shock, and c_a is the speed of sound on the same side. The quantity Σ is clearly related to the jump of R_2 across the shock. Combining the Rankine-Hugoniot conditions with (eqn 4.47) we obtain another expression for Σ, i.e.

$$\Sigma \;=\; \frac{\sqrt{(\gamma M^2-\delta)\,(1+\delta M^2)}+\delta\,(M^2-1)}{(1+\delta)M}. \tag{4.48}$$

Notice now that Σ can be computed from known values, using (eqn 4.47), which can then be used as an input in (eqn 4.48). On the other hand, (eqn 4.48) has a dependency on the unknown shock-velocity w via the relative Mach number $M=(u_a-w)/c_a$ defined in (eqn 4.17). It is also useful to invert the question: what is the relative Mach number M that yields the known Σ value? Looking at the plot in (fig 19) we see that Σ is a monotonically increasing function of M, and so there is only one value of M for each Σ. Also, the function is well approximated by a line for $M>1$. The dashed line in (fig 19) is the function

$$\gamma_1=\frac{\sqrt{(16\gamma-\delta)(1+16\delta)}+11\delta-4}{12\,(1+\delta)} \tag{4.49}$$

$$\Sigma=\gamma_1\,(M-1)+1, \tag{4.50}$$

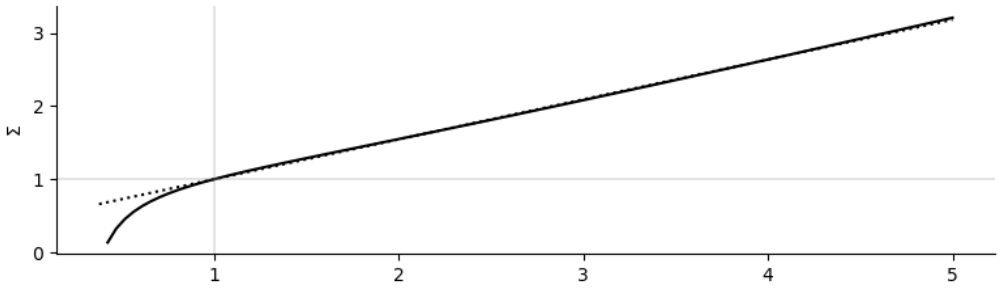


Figure 19 – The solid line is a plot of the function defined by (eqn 4.48). The function is almost linear for all $M > 1$ values. Also, $\Sigma > 1$ if and only if $M > 1$. The dashed line is the function defined in (eqn 4.50) and is used as a first guess in the iterative procedure. Here, we considered $\gamma = 7/5$ for a diatomic gas.
Source: By the author.

which coincides with the value given by (eqn 4.48) at $M = 1$ and $M = 4$.

We can now compute w with a simple iterative procedure:

(a) Compute Σ using (eqn 4.47);

(b) Obtain M from (eqn 4.50);

(c) Use M to compute Σ from (eqn 4.48) and denote it as Σ_1;

(d) Update M by adding $(\Sigma - \Sigma_1)/\gamma_1$ to it;

(e) Iterate (c) and (d) until $|\Sigma - \Sigma_1| < \epsilon$, where ϵ is the desired accuracy;

(f) Compute w from the obtained M.

Typically only five iterations of the procedure are necessary to obtain an accuracy of 10^{-6}. After w and M have been computed, the Rankine-Hugoniot jump conditions are simple to enforce as we can use (eqn 4.13), (eqn 4.14) and (eqn 4.15) to compute the values at the high-pressure side from those at the low-pressure side and the computed M. The R_2 value on the high-pressure side will be the same as before the update, but R_1 and S do not have to be preserved. Notice that in this procedure the values on the high-pressure side change the shock-speed w, but have no influence over the values on the low-pressure side, respecting the domain of dependency. A worked example using this procedure is shown in appendix A.

4.4.3 Shock detection

The method presented in section 2.4 detects shocks by comparing how well a region is described by different interpolating polynomials. This means a shock will be numerically detected shortly before being fully developed. A different method based on the ideas

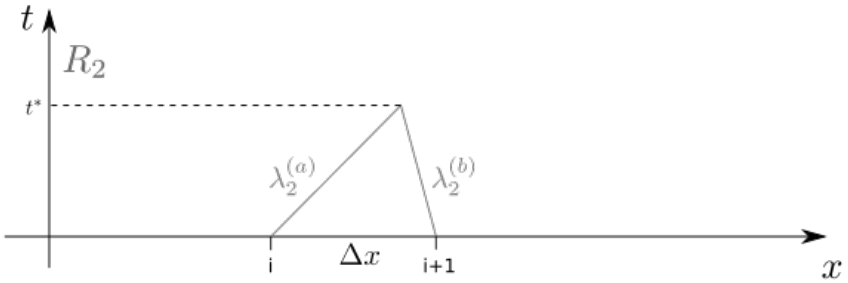


Figure 20 – Two characteristic lines carrying R_2 with slopes $\lambda_2^{(a)}$, $\lambda_2^{(b)}$ starting at points x_i, x_{i+1} respectively. After a time t^* the lines meet. If $\Delta t > 0.22t^*$ we include a tentative shock between x_i and x_{i+1} and keep it if it is strong enough.
Source: By the author.

presented in this chapter can also be used to preemptively detect shocks. The latter is not a complete substitute for the former, since it cannot detect jumps in the first derivative, but it can place incipient discontinuities more precisely on the grid.

We know that shocks are formed when the characteristics corresponding to Riemann variables of the same family coalesce, and that the slope of the Riemann variables in a space-time diagram is given by (eqn 4.36) in isentropic regions. If we now consider two neighbouring points x_i, x_{i+1} it is reasonable to assume that the entropy is approximately constant during a short period of time, justifying the use of (eqn 4.36). This, in turn, allows us to determine where the characteristic lines would intercept, if their slope really were constant. If the lines would intercept within 4 or 5 time-steps of integration, we introduce a point of discontinuity that will move according to it's calculated velocity w. This discontinuity is what is called a floating shock point. Let us note that if we introduce a shock too early the simulation will become filled with shock-waves. On the contrary, if we try to introduce it only when the characteristics intercept within 1 or 2 time-steps we could miss the shock formation and introduce non-physical oscillations. Also, one should only keep shocks that are sufficiently strong, so we opted to introduce it only if $\Sigma > 1.05$. (Recall that Σ is a dimensionless number.)

We can now describe how the shock-fitting method of Moretti uses these ideas to accurately simulate a flow with shock-waves. In each computational time-step:

- Detect new shocks using the criteria in 4.4.3;
- Remove any shocks that are weak ($\Sigma < 1.05$);
- Update the flow variables disallowing derivatives across the shocks (e.g. use the expressions for irregular points in section 2.5.3.2 or an adapted lambda-scheme[13,53]);
- Enforce the Rankine-Hugoniot jump conditions using the procedure in 4.4.2;
- Move the shock discontinuities using the computed shock-speed w.

5 RESULTS

In this chapter, I present results obtained using some of the methods discussed in this thesis. In no way this is a complete set of the possible results obtainable from the ideas presented so far, but it is, in my opinion, a succinct demonstration of the power of the methods presented.

All simulations were performed on an Intel(R) Core(TM) i5-3450 CPU @ 3.10GHz using three cores. Typical execution time for the one-dimensional simulations was 10 seconds, and for the two-dimensional simulations it was about 2 hours.

5.1 One-dimensional simulations

I will first present some results for one-dimensional simulations. For this I will use a standard test-case: Sod's shock-tube problem.[57] This problem is an instance of the Riemann problem, where two fluid regions of different densities and pressures are initially at rest separated by a membrane. When the membrane is removed the dynamics starts and, with the initial conditions of Sod's shock-tube problem shown in (fig 21), a shock-wave appears. Using Euler's equations this problem is exactly solvable through an iterative procedure and the solution at $t = 1$ can be seen in (fig 22).

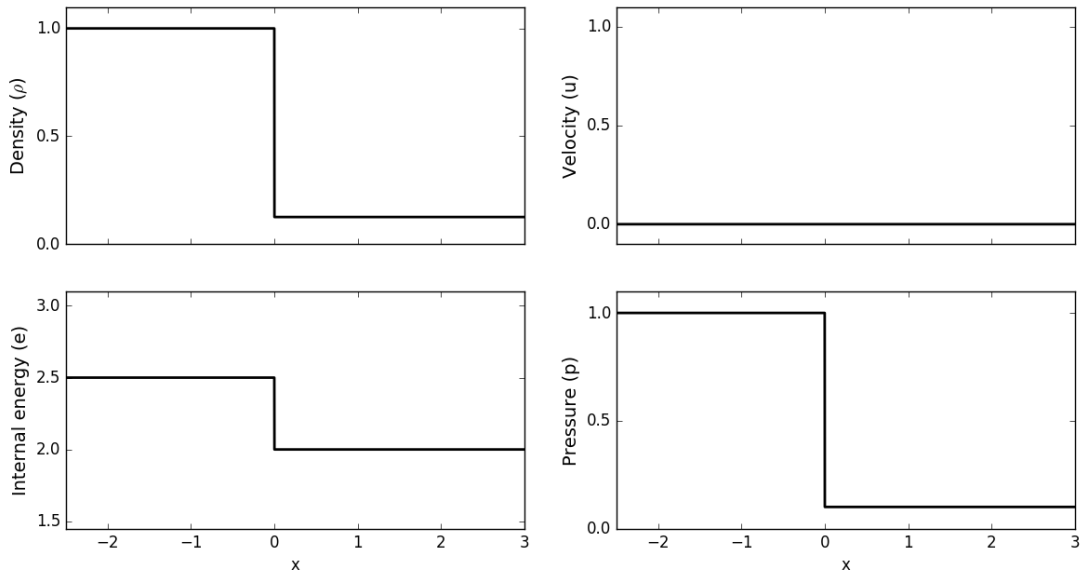


Figure 21 – Initial conditions for the standard Sod's shock-tube problem. On the left side of the membrane the fluid has $\rho_L = 1.0,\ u_L = 0,\ p_L = 1.0$, while it has $\rho_R = 0.125,\ u_R = 0,\ p_R = 0.1$ on the right side. The values of e are a direct consequence of the others quantities.
Source: By the author.

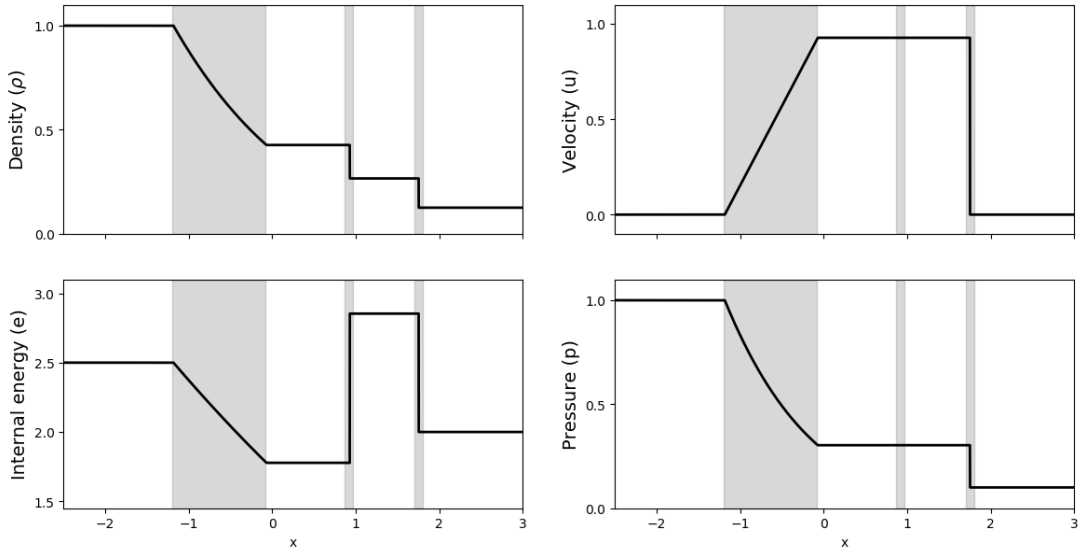


Figure 22 – Exact results for Sod's shock-tube problem at $t = 1$. Three main structures are evidenced in gray from left to right: an expansion fan, a contact discontinuity, and a shock-wave.
Source: By the author.

We can now compare the results obtained for the same initial conditions using different numerical methods presented in this thesis. In particular, I will show results obtained using three different shock-capturing methods and results obtained using the shock-fitting method of chapter 4. A grid extending from $x = -5$ to $x = 5$ with a spacing $\Delta x = 0.05$ was used in all simulations in this section.

5.1.1 Shock-capturing

In (fig 23) I show the results obtained using the upwind flux-splitting method presented in 2.5.3.3. The first-order derivatives used introduce an error proportional to the second-derivative of the flux components. This means that regions where sharp gradients suddenly appear, such as in contact discontinuities and shock-waves, are damped by the second-derivatives in the error term.

The numerical dissipation introduced by the simple first-order upwind method is large and makes it impractical for the accuracy requirements of most practical problems. The WENO method introduces numerical dissipation in a more controlled way, as it continuously switches between a fifth-order method, where the flow variables change slowly, and a first-order method, in regions where they change rapidly. The result of the simulation is show in (fig 24). In particular, the WENO simulation shows a shock-wave captured within fewer mesh points then in the results in (fig 23).

The main problem of the WENO method is that it is more computationally demanding than the simple upwind method, or than methods based on central differences. As the error term in the simple upwind method is dominated by a second-order derivative,

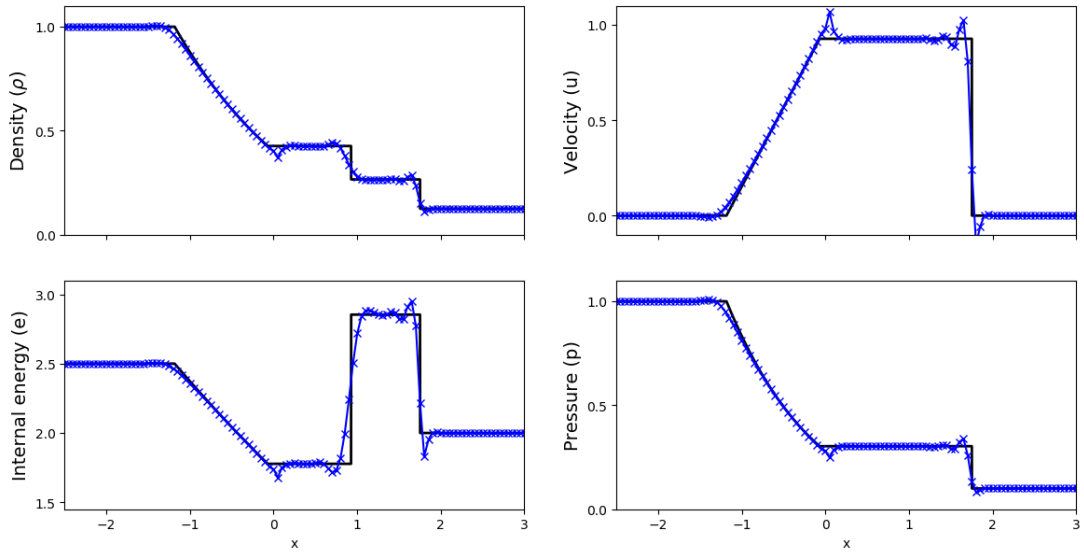


Figure 23 – Results of Sod's shock-tube problem simulated using flux-splitting technique. Some dissipative effects are evident across the contact discontinuity and the shock wave.
Source: By the author.

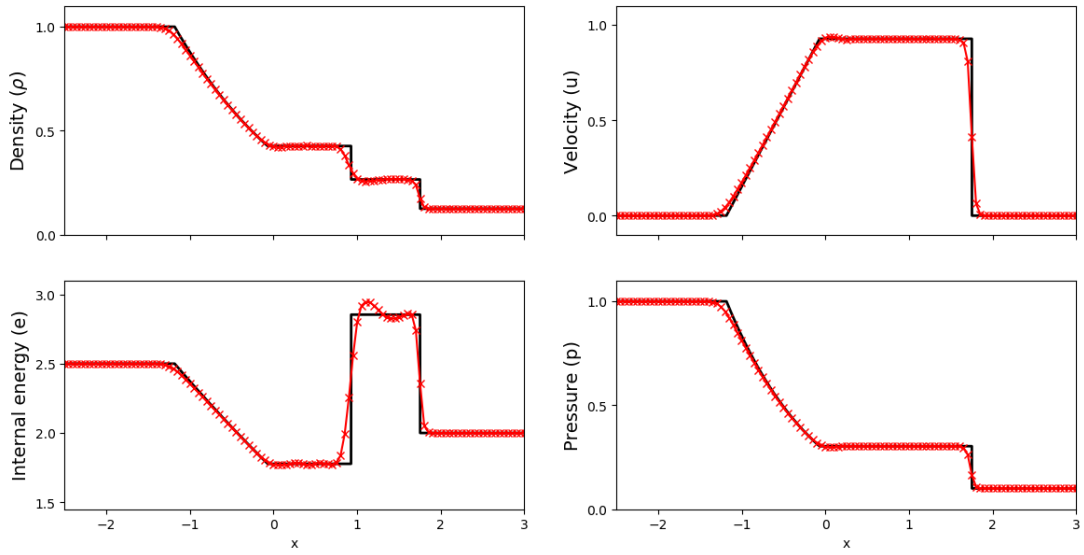


Figure 24 – Sod's shock-tube problem simulated using WENO convection scheme. The numerical solution is closer to the exact solution and the discontinuities are resolved withing fewer mesh points in comparison to (fig 23).
Source: By the author.

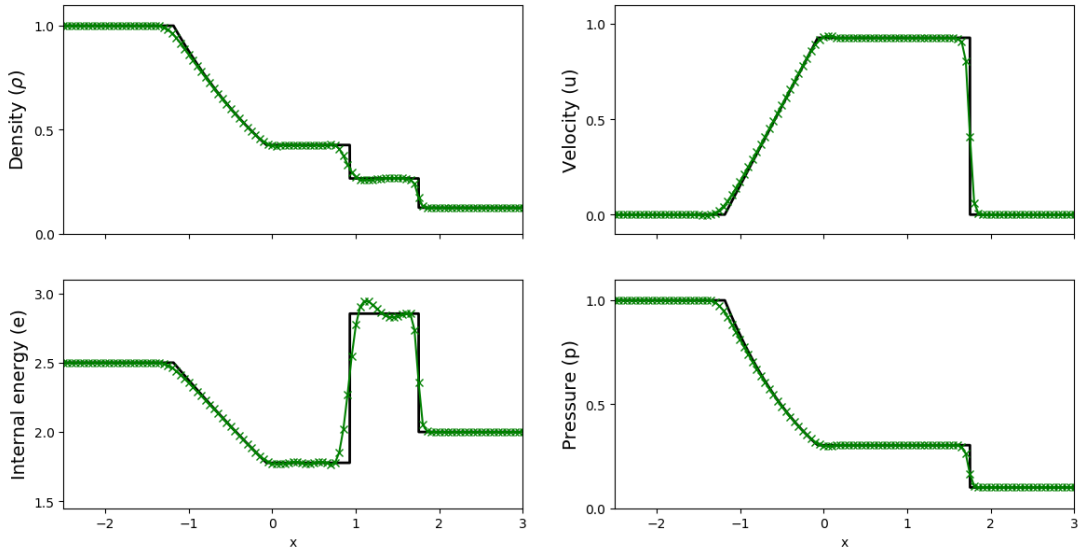


Figure 25 – Sod's shock-tube problem simulated using the hybrid method. The solution is almost identical to the one obtained using WENO method, but the use of a computationally inexpensive method in the continuous regions makes it run about four times faster then a WENO simulation.
Source: By the author.

this can be used in regions where the flow is smooth, since in this case the error is negligible. Therefore, one should use a method, such as the shock-detector method described in 2.5.5, that can determine the smoothness of a flow region in order to differentiate where the WENO method is necessary and where it is not. This can reduce execution time with little to no impact on the overall method's numerical accuracy. Let us recall that the shock detector is able to determine the maximum derivative order S_d that can be considered continuous in a region, when reconstructed using the function values at the grid points. Thus, we can use S_d as a smoothness indicator in order to decide in which regions the lower order method is acceptable. In (fig 25) we can see the results using a hybrid method, combining WENO and an upwind-biased FD scheme. Clearly they are almost identical to those obtained using only the WENO method, but were obtained in four times less time, even considering the overhead of detecting the shocks.

To see how small is the difference between the WENO and hybrid methods, and how they differ from the flux-splitting method I plot all three results in (fig 26).

5.1.2 Shock-fitting

Now I will show an example using the shock-fitting technique, described in section 4.3.2, combined with the immersed-interface method of section 3.3. This simple proof-of-concept simulation shows the compatibility of these methods. Let us stress that this is the first instance of a simulation using simultaneously both a shock-fitting and an immersed-interface approach.

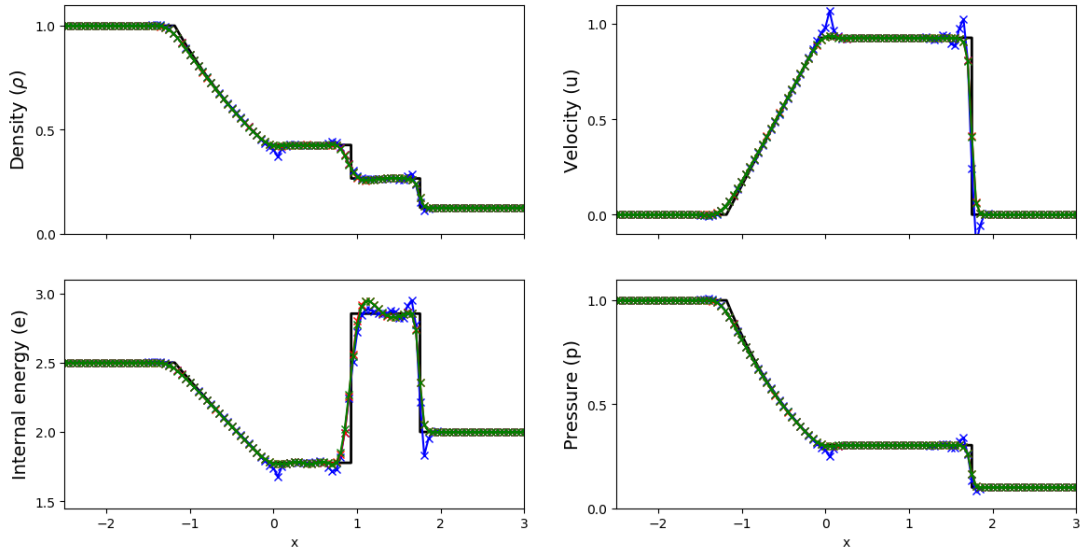


Figure 26 – The three solutions discussed are compared on the same plot. The colors are the same as in (fig 23), (fig 24) and (fig 25): flux-splitting in blue, WENO in red, and the hybrid method in green. The difference between the red and green curves is minimal.
Source: By the author.

In (fig 27) we can see the density profile of a shock, with relative Mach number $M = 2$, moving towards a fixed adiabatic wall (first panel). After that, the shock moves without spreading (second panel) and reflects from the wall (third panel). Finally, after the collision, the shock continues to move in the opposite direction, again without any numerical dissipation (fourth panel).

5.2 Two-dimensional simulations

The study of one-dimensional systems such as Sod's shock-tube in the previous section is instructive, but not of much practical use. In this section I will show how some of the ideas presented in one-dimension can be applied to two-dimensional problems. I will also show examples using both immersed-interface methods described in chapter 3. All simulations in this section use the full Navier-Stokes equations.

5.2.1 Immersed-interface methods

In many points throughout this thesis I mentioned that there are many pitfalls in solving the Navier-Stokes equations. Most times an inappropriate method will generate infinite values, or negative pressures and ultimately result in a crash, but that is not always the case. In (fig 28) we can see one such case where the simulation yields wrong results without breaking down completely. This simulation uses the Ghias method, explained in section 3.2, to describe the immersed-interface. The issue is that, when combined with a central-differencing scheme, as done here, numerical errors accumulate due to an odd-even decoupling between the pressure and velocity fields. Indeed, the pressure values

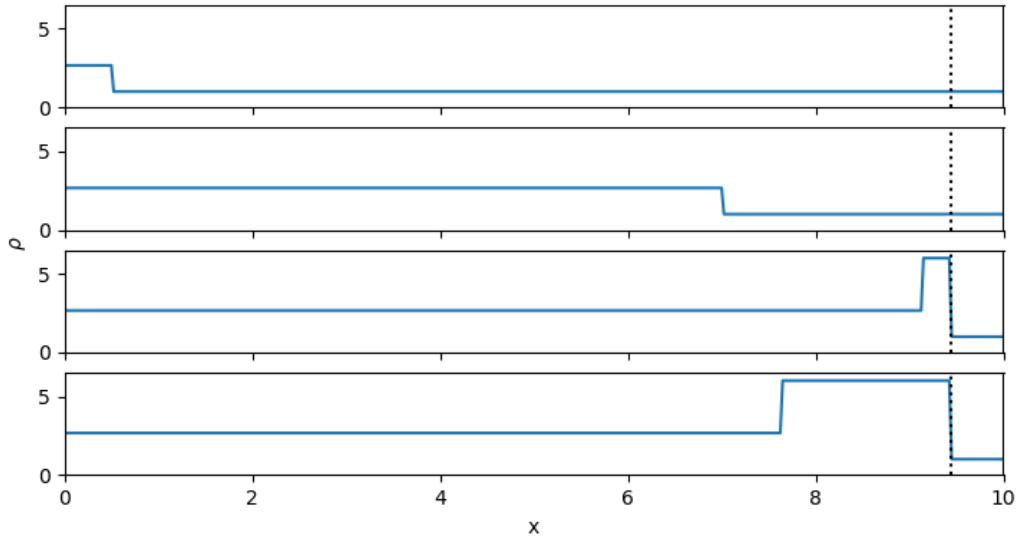


Figure 27 – Density profile of a floating-shock reflecting from a wall (black dashed line). The initial conditions are such that $\rho_l = 8/3$, $u_l = 5/4$, $p_l = 45/14$ to the left of the shock, and $\rho_r = 1$, $u_r = 0$, $p_r = 5/7$ to the right of it. From top to bottom the panels show: the system's initial conditions, the shock approaching the wall, the system moments after the shock interacts with the wall, the shock moving away from the wall. Both the shock and the wall boundaries are described by the same base data-structure and are treated identically when computing numerical derivatives.
Source: By the author.

at odd-numbered points depend on the velocity values at the even-numbered points and vice-versa. Moreover, in these settings the viscosity is too low to ensure the coupling of even and odd points through the second-order derivatives. This checkerboard pattern is a well known problem in fluid simulations and a clear indication of unphysical behaviour.

On the other hand, if we use a different convection scheme, such as WENO, we can avoid the odd-even decoupling to once again obtain a physically correct solution. Indeed, in (fig 29) we can see that, when using the WENO method, the checkerboard pattern is no longer present and the solution is indeed physically acceptable.

However, using WENO or any other convection scheme is not the only way of eliminating the unphysical oscillations. Indeed, if we apply the filter method of section 2.5.4 we recouple the values of odd- and even-numbered points, avoiding the high-frequency oscillations in the checkerboard pattern. In (fig 30) we show the result obtained using the same central-differencing scheme considered in (fig 28) combined with this filter.

It is interesting to point out that the change in convection scheme and the use of filtering did not require any changes in the immersed-interface method being used.

As a final example in this section I also simulate the same cylinder using central-differences convection and filtering, but using Karagiozis IIM, described in 3.3, for the

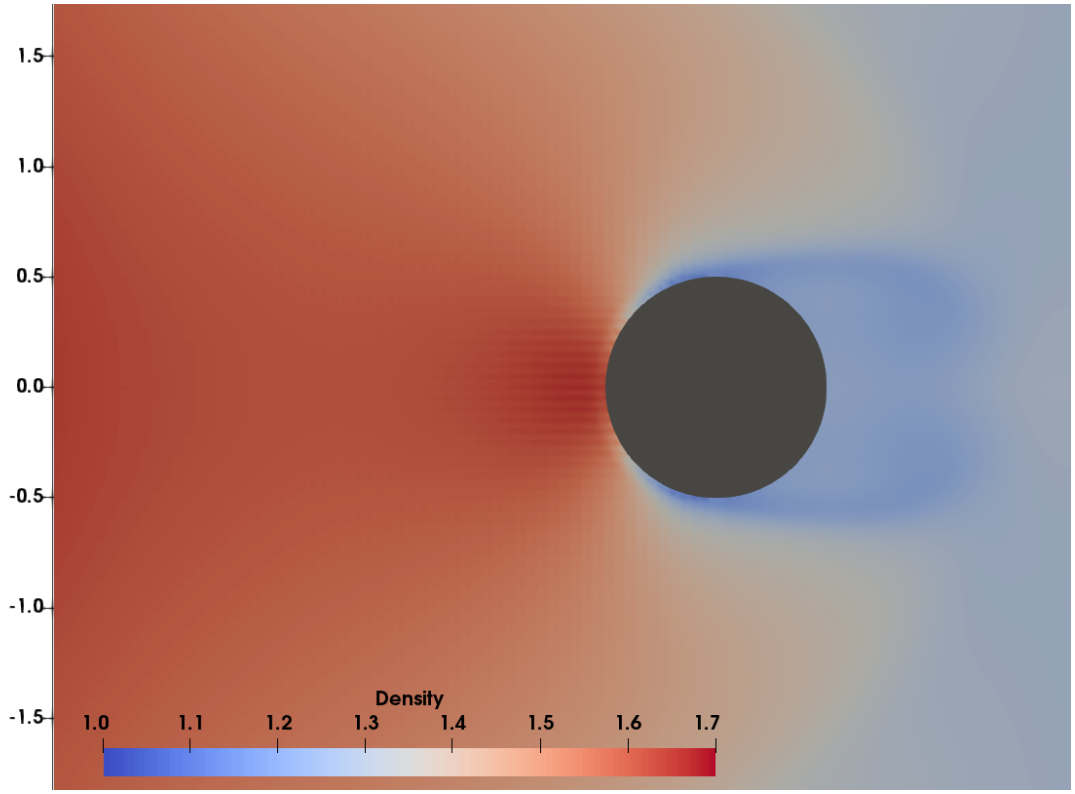


Figure 28 – Navier-Stokes simulation of a cylinder of unit diameter at Mach number $Ma = 0.5$, Reynolds number $Re = 500$, and Prandtl number $Pr = 1$, performed on a 400×200 regular Cartesian grid over the domain $[0, 10] \times [-2.5, 2.5]$. The top and bottom domain boundaries are adiabatic no-slip walls. The left domain boundary is a subsonic inlet with a parabolic velocity profile. The right domain boundary is a subsonic outlet. The cylinder is described using Ghias IIM and convection is performed using a simple central-differencing scheme for all derivatives. Time evolution is obtained using the third-order accurate Runge-Kutta scheme of section 2.5.6. The accumulation of numerical errors leads to an unphysical checkerboard pattern in the density field, visible in the center of the figure.
Source: By the author.

boundary description instead of Ghias IIM. The results in (fig 31) are not completely comparable to the others, since the Karagiozis IIM uses Dirichlet boundary conditions for the temperature whereas Ghias IIM uses Neumann boundary conditions. Nevertheless, we can see similar results to those obtained in (fig 30).

5.2.2 High-speed flows

I now extend the results of the hybrid method, used to obtain (fig 25), from one to two dimensions, as introduced by myself in a recent conference.[58] In particular, we detect shocks on each grid line and grid column as if they were one-dimensional grids. If a point has been marked as discontinuous in either direction, all convective terms are computed using the WENO method, while, if it was considered smooth, all convective terms are computed using a central-difference scheme. Dissipative terms are always computed using central differences. Similar to the one-dimensional version, this hybrid method selectively

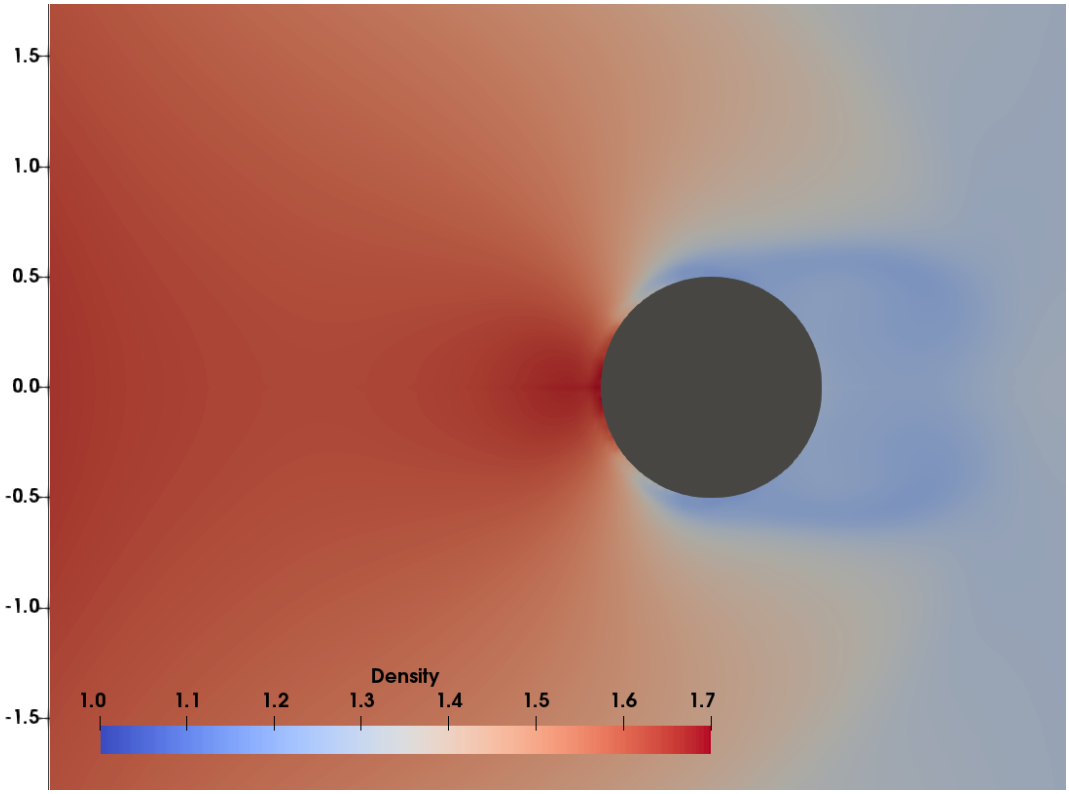


Figure 29 – Navier-Stokes simulation of a unit diameter cylinder under the same conditions used in (fig 28) except for the convective terms discretization scheme. Using WENO scheme for the convective terms controls the oscillations appearing in (fig 28) resulting in a physically acceptable solution.
Source: By the author.

employs the more expansive WENO method where it is necessary, thus lowering the time required to run the simulation. The result is reported in (fig 32)

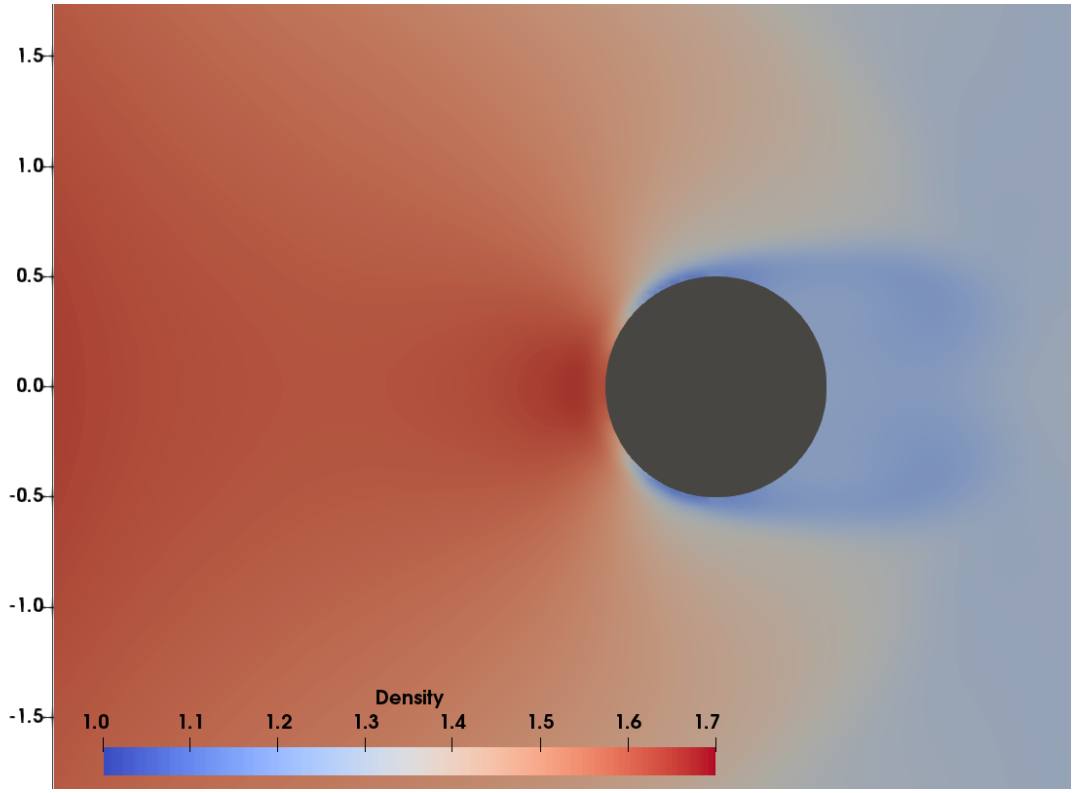


Figure 30 – Navier-Stokes simulation of a unit diameter cylinder under the same conditions used in (fig 28) adding a filtering step. The filtering smoothes the solution by damping the high-frequency components, thus eliminating the checkerboard pattern from (fig 28).

Source: By the author.

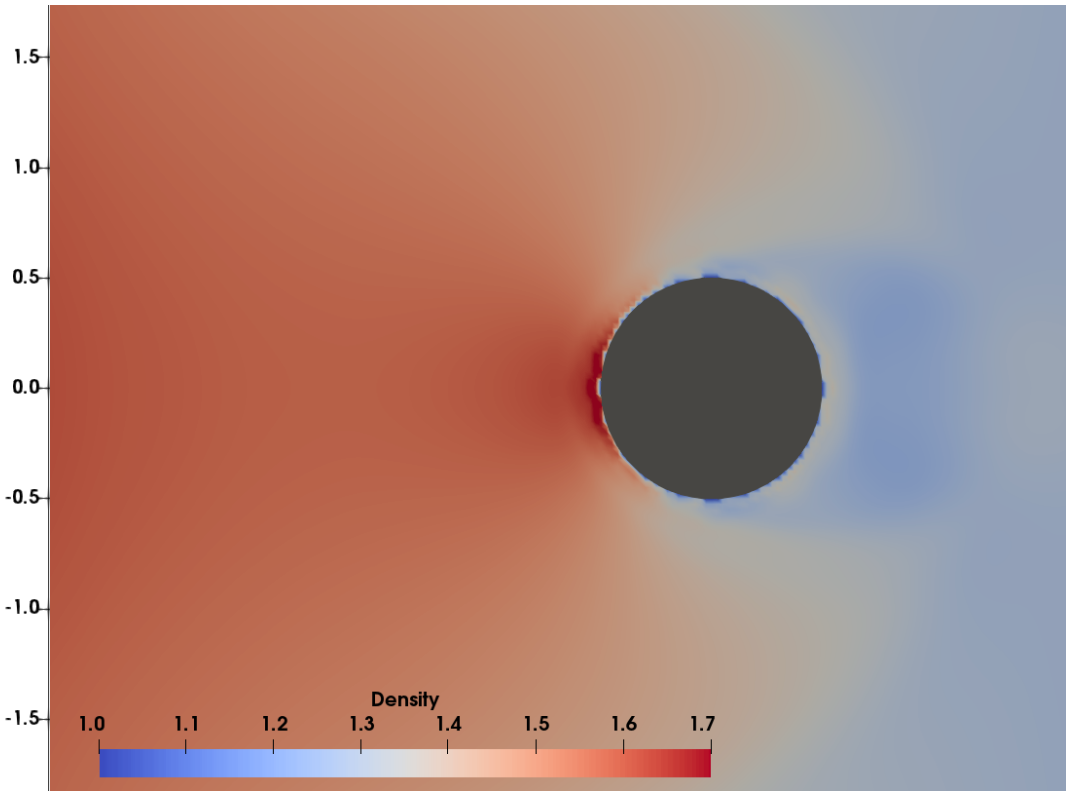


Figure 31 – Navier-Stokes simulation of a unit diameter cylinder under the same conditions considered in (fig 30) except for the immersed-interface method used to describe the cylinder. Here, the simulation was performed using Karagiozis boundary description and the results are similar but not identical to those obtained in (fig 30), due to the difference in boundary conditions at the interface. The blue jagged pattern visible near the interface is an artifact of plotting and it is not related to the simulation results.

Source: By the author.

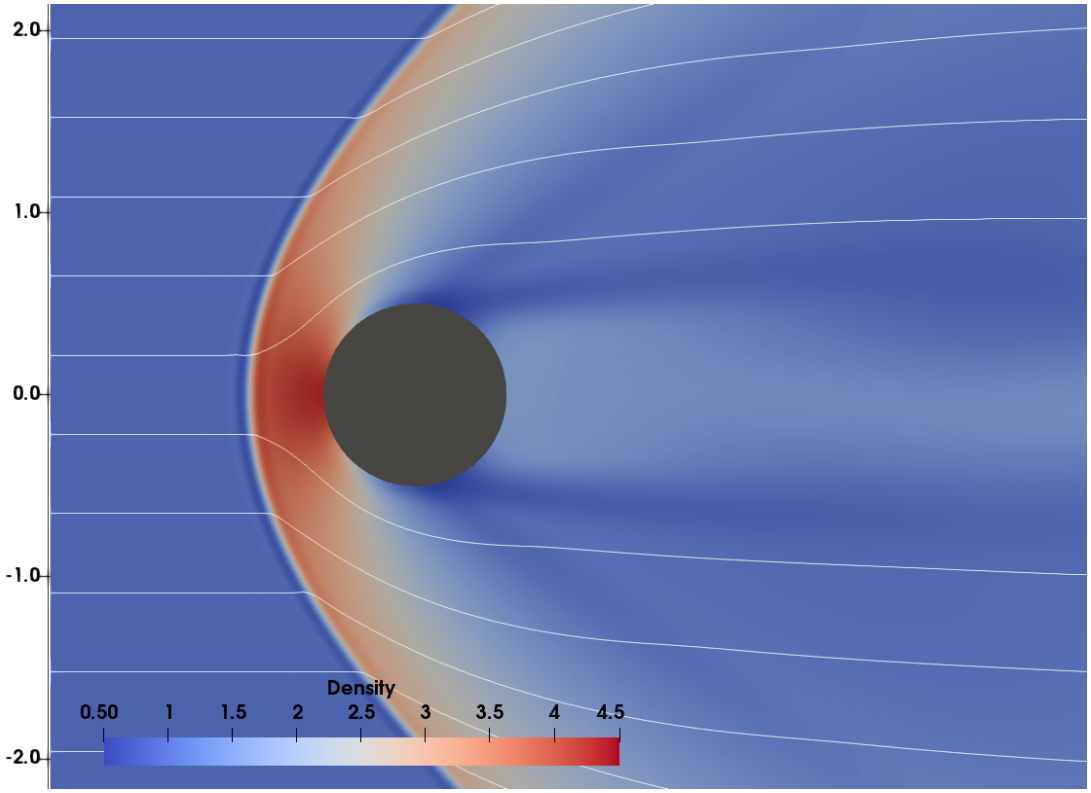


Figure 32 – Simulation of a cylinder travelling at a Mach number $Ma = 3$, Reynolds number $Re = 500$, and Prandtl number $Pr = 1$, using the hybrid method. The bow shock is well resolved and no unphysical oscillations are visible. The white lines, called streaklines, trace the path a particle would follow in the fluid.
Source: By the author.

6 CONCLUSIONS

In the present thesis I have studied immersed-interface methods and methods to handle shock-waves. The original topic of my thesis was high-performance implementations of computational fluid dynamics methods, but it changed throughout the years. Nevertheless, my study was always guided by the search for numerical methods amenable to optimizations, such as those based on Cartesian grids, which are suitable building blocks for high-performance computing. In particular, in this thesis, I proposed two versions of IIMs simplifying their original implementations. Also, I have shown how one can use a shock-detector to combine a computationally-demanding shock-capturing method with a fast non-conservative method, in order to reduce simulations run-time. Finally, I combined a shock-fitting method and an IIM. Such combination can accurately position a shock-wave on the grid and account for shock-shock and shock-wall interactions using fewer grid points. Next, I will summarise the main results and suggest ways in which this work can be continued.

The first new result in this thesis was the simulation using a hybrid method reported in (fig 25). By using a shock-detector to switch between WENO method and a FD scheme, I obtained almost identical results as the ones computed using WENO method exclusively, but in less time. I have also extended this result to two dimensions in (fig 32). There are some minor disadvantages to this method, though. First, depending on the combination of methods employed almost all points in the grid are marked as discontinuous and all benefits are lost. Second, it is not obvious what should be the shock-detector sensitivity, i.e. the value of S_d that triggers the switching, so there is a tunable parameter to be adjusted.

In this work, Ghias immersed-interface method had it's ghost-point computations simplified, eliminating the iterative procedure originally employed when the values at one ghost-point depended directly on another's value. This simplification did not impact the method's applicability compared to the original one. I showed that it can immediately be combined with both filtering methods and shock-detectors, as well as with different convection schemes. The main disadvantage of this IIM is the expansive computation required to handle moving boundaries, as the ghost-points must be determined and all image-point positions must be recomputed at each timestep. Karagiozis IIM, on the other hand, can handle moving boundaries easily, as there are no auxiliary structures, e.g. image points, that need to be recomputed due to the interface moving. The original implementation of this second IIM required an implicit computation to obtain the dissipative terms near the interface, but I showed that it suffices to use the second-order derivatives described in section 2.5.3.2.

In chapter 4 the main topic was shock waves and how they can be treated both analytically and numerically. I proceeded to show an alternative approach that imposes the Rankine-Hugoniot jump conditions exactly: the floating shock-fitting method introduced by Gino Moretti. Moretti's method is relatively unknown and I believe it deserves more attention, so I joined the efforts of his close collaborators to spread it. To give my contribution to the development of floating shock-fitting methods, I showed that the IIM created by Karagiozis is compatible with Moretti's method. Indeed, both methods use the same type of structure to mark discontinuities on the grid lines. It has been the first time a shock-fitting method was combined with an IIM. The next obvious step is to extend the results to two dimensions.

This thesis has been my contribution to the development of fluid dynamics simulation, specially by showing the viability of the shock detector of section 2.5.5 for two dimensional problems and the combination of shock-fitting and immersed interface methods in section 5.1.2.

REFERENCES

1 BELOTSERKOVSKII, O. M. **Flow past a circular cylinder with a detached shock wave**. 1959. Available from: <https://apps.dtic.mil/dtic/tr/fulltext/u2/607641.pdf>. Accessible at: 23 Jan. 2018.

2 ABBETT, M. et al. A time-dependent computational method for blunt body flows. **AIAA Journal**, v. 4, n. 12, p. 2136–2141, 1966. doi: 10.2514/3.3867.

3 HALL, A.E.; CANNELL, G.H.; LAWTON, H.W. **Agriculture in Semi-Arid Environments**. New York: Springer, 2011. (Ecological studies, 34).

4 PASCAL, B. **Traités de l'equilibre des liqueurs et de la pesanteur de la masse de l'air** . Paris: Gauthier-Uillars, 1663. 258 p.

5 EULER, L. Principes généraux du mouvement des fluides. **Mémoires de l'Académie des Sciences de Berlin**, v. 11, p. 274–315, 1757.

6 NAVIER, C. L. Memoire sur les lois du mouvement des fluides. **Mémoires de l'Académie des sciences de l'Institute de France**, v. 6, p. 389–440, 1822.

7 SACKMANN, M. et al. Shock-wave lithotripsy of gallbladder stones. **New England journal of medicine**, v. 318, n. 7, p. 393–397, 1988. doi: 10.1056/NEJM198802183180701.

8 SPEED, C. et al. Extracorporeal shock wave therapy for plantar fasciitis. a double blind randomised controlled trial. **Journal of Orthopaedic Research**, v. 21, n. 5, p. 937–940, 2003. doi: 10.1016/S0736-0266(03)00048-2.

9 PLETCHER, R. et al. **Computational fluid mechanics and heat transfer, 2nd ed**. Washington: Taylor & Francis, 1997.

10 OBSERVATORY, E. **von Karman Vortices**. 2002. Available from <https://earthobservatory.nasa.gov/images/2613/von-karman-vortices>. Accessible at: 10 Dec. 2018.

11 DYKE, M. V. **An album of fluid motion**. California: Parabolic Press, 1982.

12 STOKES, G. G. On the theories of the internal friction of fluids in motion. **Transactions of the Cambridge Philosophical Society**, VIII, p. 287–305, 1845.

13 NASUTI, F. et al. Analysis of unsteady supersonic viscous flows by a shock-fitting technique. **AIAA Journal**, v. 34, n. 7, p. 1428–1434, 1996. doi: 10.2514/3.13249.

14 GINGOLD, R. A. et al. Smoothed particle hydrodynamics: theory and application to non-spherical stars. **Monthly Notices of the Royal Astronomical Society**, v. 181, n. 3, p. 375–389, 1977. doi: 10.1093/mnras/181.3.375.

15 CHEN, S. et al. Lattice Boltzmann method for fluid flows. **Annual review of fluid mechanics**, v. 30, n. 1, p. 329–364, 1998. doi: 10.1146/annurev.fluid.30.1.329.

16 HUSSAINI, M. Y. et al. Spectral methods in fluid dynamics. **Annual Review of Fluid Mechanics**, v. 19, n. 1, p. 339–367, 1987. doi: 10.1146/annurev.fl.19.010187.002011.

17 JAVADI, A. et al. Modification of k-epsilon turbulent model using kinetic energy–preserving method. **Numerical Heat Transfer**, v. 68, n. 6, p. 554–577, 2015. doi: 10.1080/10407790.2015.1065145.

18 MANISHA, P. **Multiblock mesh**. 2014. Available from <https://manisha059.files.wordpress.com/2014/01/super-fine-mesh.png>. Accessible at: 16 Mar. 2019.

19 CHEN, X. **Unstructured mesh**. 2011. Available from <https://www.msi.umn.edu/~chenxy/images/msh4.gif>. Accessible at: 18 Mar. 2019.

20 MORTON, K. et al. **Numerical solution of Partial differential equations**. New York: Cambridge University Press, 2005.

21 ANDERSON, J. et al. **Computational fluid dynamics: an introduction**. New York: Springer Berlin Heidelberg, 2013.

22 GIBBS, J. W. Fourier's series. **Nature**, v. 59, n. 1522, p. 200, 1898. doi: 10.1038/059200b0.

23 _____. Fourier's series. **Nature**, v. 59, n. 1539, p. 606, 1899. doi: 10.1038/059606a0.

24 CARSLAW, H. S. A historical note on gibbs' phenomenon in fourier's series and integrals. **Bulletin of the American Mathematical Society**, v. 31, n. 8, p. 420–424, 1925. doi: 10.1090/S0002-9904-1925-04081-1.

25 STEGER, J. L. et al. Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods. **Journal of Computational Physics**, v. 40, p. 263–293, 1981. doi: 10.1016/0021-9991(81)90210-2.

26 LIU, X.-D. et al. Weighted essentially non-oscillatory schemes. **Journal of computational physics**, v. 115, p. 200–212, 1994. doi: 10.1006/jcph.1994.1187.

27 VASILYEV, O. V. et al. A general class of commutative filters for LES in complex geometries. **Journal of Computational Physics**, v. 146, n. 1, p. 82–104, 1998. doi: 10.1006/jcph.1998.6060.

28 OLIVEIRA, M. L. B. de et al. Analysis of a high-order finite difference detector for discontinuities. **International Journal of Computer Mathematics**, v. 94, n. 2, p. 676–689, 2016. doi: 10.1080/00207160.2015.1124100.

29 HYMAN, M. A. Non-iterative numerical solution of boundary-value problems. **Applied Scientific Research B**, Springer, v. 2, n. 1, p. 325–351, 1952. doi: 10.1007/BF02919780.

30 LEVEQUE, R. J. et al. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. **SIAM Journal on Numerical Analysis**, v. 31, n. 4, p. 1019–1044, 1994.

31 LI, Z. **The immersed interface method**: a numerical approach for partial differential equations with interfaces. 1994. 172 p. Ph. D. Thesis (Philosophy) – University of Washington, Washington, 1994.

32 LI, Z. An overview of the immersed interface method and its applications. **Taiwanese Journal of Mathematics**, v. 7, n. 1, p. 1–49, 2003. doi: 10.11650/twjm/1500407515.

33 ZHONG, X. A new high-order immersed interface method for solving elliptic equations with imbedded interface of discontinuity. **Journal of Computational Physics**, v. 225, n. 1, p. 1066–1099, 2007. doi: 10.1016/j.jcp.2007.01.017.

34 PESKIN, C. Flow patterns around heart valves: A numerical method. **Journal of Computational Physics**, v. 10, n. 2, p. 252–271, 1972. doi: 10.1016/0021-9991(72)90065-4.

35 GHIAS, R. et al. A sharp interface immersed boundary method for compressible viscous flows. **Journal of Computational Physics**, v. 225, n. 1, p. 528–553, 2007. doi: 10.1016/j.jcp.2006.12.007.

36 KARAGIOZIS, K. et al. A low numerical dissipation immersed interface method for the compressible Navier-Stokes equations. **Journal of Computational Physics**, v. 229, n. 3, p. 701–727, 2010. doi: 10.1016/j.jcp.2009.10.005.

37 WEISSTEIN, E. W. **Cubic spline**. Available from <http://mathworld.wolfram.com/CubicSpline.html>. Accessible at: 26 Jan. 2019.

38 SHIN, J. et al. Reflection coefficients and reflected scaled impulses from detonations of high explosives as a function of angle of incidence. **Journal of Structural Engineering**, v. 143, n. 7, p. 04017043, jul 2017. doi: 10.1061/(asce)st.1943-541x.0001766.

39 SALAS, M. D. **A shock-fitting primer**. Boca Raton: CRC Press, 2017. (Chapman & Hall/CRC Applied Mathematics & Nonlinear Science).

40 WU, Z. et al. Review of shock wave detection method in CFD post-processing. **Chinese Journal of Aeronautics**, v. 26, n. 3, p. 501–513, jun 2013. doi: 10.1016/j.cja.2013.05.001.

41 LEVEQUE, R. J. et al. **Numerical Methods for Conservation Laws**. Basel: Birkhäuser, 2005.

42 GOTTLIEB, S. et al. Recovering high-order accuracy in WENO computations of steady-state hyperbolic systems. **Journal of Scientific Computing**, v. 28, n. 2-3, p. 307–318, 2006. doi: 10.1007/s10915-006-9078-8.

43 TALBOT, L. et al. Measurement of shock wave thickness by the electron beam fluorescence method. **Physics of Fluids**, v. 9, n. 4, p. 633, 1966. doi: 10.1063/1.1761728.

44 MORETTI, G. Thirty-six years of shock fitting. **Computers & Fluids**, v. 31, p. 719–723, 2002. doi: 10.1016/S0045-7930(01)00072-X.

45 KUTLER, P. et al. Multishocked, three-dimensional supersonic flowfields with real gas effects. **AIAA Journal**, v. 11, n. 5, p. 657–664, 1973. doi: 10.2514/3.50505.

46 MARCONI, F. et al. Computation of three dimensional flows about aircraft configurations. **Computer & Fluids**, v. 1, n. 2-3, p. 185–195, 1973. doi: 10.1016/0045-7930(73)90017-0.

47 ZHONG, X. High-order finite-difference schemes for numerical simulation of hypersonic boundary-layer transition. **Journal of Computational Physics**, v. 144, n. 2, p. 662–709, 1998. doi: 10.1006/jcph.1998.6010.

48 _____. Leading-edge receptivity to free-stream disturbance waves for hypersonic flow over a parabola. **Journal of Fluid Mechanics**, v. 441, n. 2001, p. 315–367, 2001. doi: 10.1017/S0022112001004918.

49 ROMICK, C. M. et al. High-order shock-fitted detonation propagation in high explosives. **Journal of Computational Physics**, v. 332, p. 210–235, 2017. doi: 10.1016/j.jcp.2016.11.049.

50 RAWAT, P. S. et al. On high-order shock-fitting and front-tracking schemes for numerical simulation of shock-disturbance interactions. **Journal of Computational Physics**, v. 229, n. 19, p. 6744–6780, 2010. doi: 10.1016/j.jcp.2010.05.021.

51 MORETTI, G. **A Critical Analysis of Numerical Techniques: The Piston-Driven Inviscid Flow**. **A critical analysis of numerical techniques:** the piston-driven inviscid flow. New York, 1969. Available from: <https://apps.dtic.mil/dtic/tr/fulltext/u2/693037.pdf>. Accessible at: 23 Jan. 2019.

52 _____. **Complicated one-dimensional flows**. New York, 1971. Available from: <https://pdfs.semanticscholar.org/147b/c8d5c79c30c572aac99b31fa689904b7696f.pdf>. Accessible at: 23 Jan. 2019.

53 MORETTI, G. et al. **An improved lambda-scheme for one-dimensional flows**. Washington: National Aeronautics and Space Administration, 1983. (NASA contractor report, 3712).

54 MORETTI, G. A technique for integrating two-dimensional Euler equations. **Computers and Fluids**, v. 15, n. 1, p. 59–75, 1987. doi: 10.1016/0045-7930(87)90005-3.

55 _____. The λ-scheme. **Computers & Fluids**, v. 7, n. 3, p. 191–205, 1979. doi: 10.1016/0045-7930(79)90036-7.

56 MACCORMACK, R. **The effect of viscosity in hypervelocity impact cratering**. In: AERODYNAMIC TESTING CONFERENCE, 4., 1969. doi: 10.2514/6.1969-354.

57 SOD, G. A. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. **Journal of Computational Physics**, v. 27, n. 1, p. 1–31, 1978. doi: 10.1016/0021-9991(78)90023-2.

58 AURICHIO, V. H. et al. An immersed interface method for high mach number flows In: CONGRESSO BRASILEIRO DE FLUIDODINÂMICA COMPUTACIONAL, 2., 2018. Rio de Janeiro. **Anais...** Rio de Janeiro: Galoa, 2018. doi: 10.17648/cbcfd-83728.

Appendix

APPENDIX A – A WORKED SHOCK-COMPUTATION EXAMPLE

I will show how the procedure described in 4.4.2 is used to enforce the Rankine-Hugoniot jump conditions. We will consider the same initial conditions used in (fig 27), i.e. $\rho_a = 1,\ u_a = 0,\ p_a = 5/7$ and $\rho_b = 8/3,\ u_b = 5/4,\ p_b = 45/14$, and use (fig 33) as a reference. As before, we set $\gamma = 7/5$ which implies $\delta = (\gamma - 1)/2 = 1/5$. Using these values, we obtain $c_a = 1$ and $c_b = 3\sqrt{3}/4$. To compute Σ we first compute the Riemann variable

$$R_{2b} = \frac{c_b}{\delta} - u_b = \frac{15\sqrt{3}}{4} + \frac{5}{4}, \tag{A.1}$$

and then we obtain Σ using (eqn 4.47), i.e.

$$\Sigma \ = \ \delta\frac{R_{2b} + u_a}{c_a} = \frac{3\sqrt{3}+1}{4}. \tag{A.2}$$

We have to remember that Σ is computed in a reference frame directed from the low-pressure side a towards the high-pressure side b, so that the velocity u_b had it's sign flipped. If we use the iterative procedure in 4.4.2 to determine M we obtain $M = 2$. Indeed, substituting $M = 2$ in (eqn 4.48) we find the same value for Σ as in (eqn A.2). Notice that to obtain M we used only one quantity from the high-pressure side b, namely R_{2b}, and so any other combination of ρ_b, u_b and p_b resulting in the same value of R_{2b} would give the same value for M. Using $M = 2$ we can now obtain the shock-velocity $w = u_A - Mc_a = -2$. Again, recall that the reference frame is reversed, so the shock is moving from left to right. Finally, using (eqn 4.7), (eqn 4.8) and (eqn 4.10) we can compute the values for ρ_b, u_b and p_b from the flow-variable values in a and w. In this example, they would be the same as in the initial conditions.

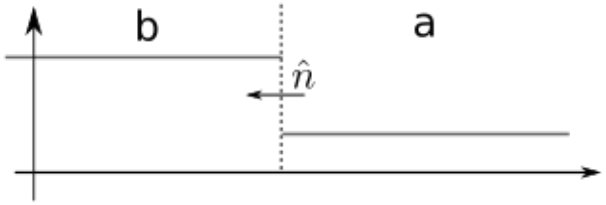


Figure 33 – Schematic view of a shock. The low-pressure side a is on the right and the high-pressure side b to the left of the shock. The reference frame for the shock computation is the normal unit vector $\hat{n}$.
Source: By the author.