

POLITECNICO DI TORINO

Master's Degree in Aerospace Engineering



**Politecnico
di Torino**



**von KARMAN INSTITUTE
FOR FLUID DYNAMICS**

Master's Degree Thesis

Development of boundary conditions for Lattice Boltzmann Method applied to non-lattice conforming geometries

Supervisors

Prof. Sandra PIERACCINI

Prof. Thierry MAGIN

Candidate

Andrea COLTRARO

JULY 2023

Abstract

This master's thesis focuses on the Lattice Boltzmann Method, an alternative to conventional Computational Fluid Dynamics, which provides significant computational cost savings by employing a regular lattice independent of the geometry considered instead of a structured or unstructured mesh which itself may prove to be a labor and CPU-intensive complex task, especially for complex geometries. The computational efficiency of the LBM, the simplicity of the algorithm, and the ease with which this method allows complex geometries to be dealt with promise numerous advantages in complex industrial flow applications even in the supersonic and hypersonic ranges.

This work is a follow-up of the LBMHYPE project, an ESA TRP research project involving a partnership between the von Karman Institute for Fluid Dynamics, École Polytechnique, Université Paris Sud Saclay and CENAERO. The aim of the LBMHYPE project is to extend LBM to supersonic and hypersonic regimes since, at the state of the art, LBM has been limited to the incompressible and weakly compressible range. The major innovation proposed in the LBMHYPE project to simulate compressible flows is the use of innovative vectorial scheme methods, in which each conservation equation is solved by a dedicated Lattice Boltzmann scheme.

A practical challenge encountered in LBM which seriously degraded the solution was the application of slip-wall boundary conditions to geometries that do not conform with the Cartesian lattice, such as an arbitrarily inclined wedge surface of a flow over a wedge.

The contribution of this work involves implementing a new type of LBM boundary condition, called Bouzidi Bounce Back with Normal, within the Python library "pylbn" which was developed in the course of the LBMHYPE project.

This type of boundary condition is an extension of the frequently applied Bouzidi-type LBM boundary condition, which is in turn an extension of the better-known LBM boundary condition, namely, the Bounce-Back boundary condition by adding a corrective term, which is a numerical flux, to Bouzidi's relationships. This treatment takes into account the normal and tangential directions of the wall surface and accurately resolves the geometrical orientation of the slip wall and its normal.

The Bouzidi Bounce Back with Normal boundary conditions successfully solved the problems faced in the LBMHYPE project regarding the imposition of the slip condition to non-Cartesian coordinate conforming geometries. The results compare more favorably with the analytical and US3D code results than the results obtained in the LBMHYPE project. Furthermore, these conditions have demonstrated good

performance in non-rectilinear geometries. In fact, in the last part of this thesis, BCs were tested in the classic problem of a compressible flow around a 2D cylinder.

Possible future developments of the present project might involve addressing the numerical diffusion and extension of the LBM vectorial scheme to the Navier-Stokes equations. The LBM method development should also target where it is particularly powerful, in cases involving complex geometries and multiple physics such as multi-phase flows.

Table of Contents

List of Tables	VII
List of Figures	VIII
Acronyms	XIV
1 Introduction	2
1.1 Conventional CFD	5
1.2 Lattice Boltzmann Method	6
1.3 Contents of each chapter of this thesis	8
2 The Boltzmann Equation	11
2.1 Kinetic Theory	11
2.2 Distribution Function	13
2.2.1 Maxwell distribution function	14
2.2.2 Boltzmann Distribution	16
2.3 The Boltzmann Equation	18
2.3.1 Boltzmann Transport Equation	18
2.3.2 The BGK Approximation	20
2.3.3 Multi Relaxation Times schemes	22
3 Lattice Boltzmann Method	25
3.1 Literature Review for LBM	25
3.2 LBM for compressible applications	27
3.2.1 Finite Difference Lattice Boltzmann Method	28
3.2.2 Finite Volume Methods & Lattice Boltzmann	30
3.2.3 Implicit Lattice Boltzmann Methods	31
3.2.4 Lattice Boltzmann & VLES	32
3.3 The lattice Boltzmann approach	33
3.4 Vectorial Schemes	37
3.4.1 Lattice Boltzmann schemes for compressible flows	38

3.4.2	LBM vectorial scheme D_1Q_{222}	39
3.4.3	LBM vectorial scheme D_2Q_{4444}	41
4	Boundary conditions for the LBM	46
4.1	Literature Review of BCs	46
4.2	Bounce-Back Boundary Conditions	47
4.2.1	Implementation on Python of Bounce Back BCs	50
4.3	Bouzidi Bounce-Back Boundary Conditions	53
4.3.1	Implementation on Python of Bouzidi BCs	56
5	Implementation of BCs for general geometries	58
5.1	Interpretation of <i>f-functions</i> as flux	58
5.1.1	Euler Equation	61
5.2	Bounce-Back with Normal	66
5.3	Bouzidi Bounce-Back with Normal	68
5.3.1	Implementation on Python of Bouzidi Bounce-Back with Normal Boundary Conditions	69
6	Supersonic Wedge	74
6.1	Theory of oblique shock	74
6.1.1	Oblique shock	76
6.1.2	Bow shock	77
6.1.3	Analytical solution	78
6.2	Analytical Solution of the Wedge	80
6.3	Wedge US3D Solution	84
6.3.1	US3D Introduction	84
6.3.2	CFD simulation input parameters	85
6.3.3	US3D Wedge Results	88
6.4	Wedge LBM Solution	95
6.4.1	LBM scheme used for the wedge	96
6.4.2	Bouzidi BCs Solution	98
6.4.3	Bouzidi Bounce Back with Normal BCs Solution	105
7	Supersonic cylindrical-nosed body	123
7.1	Bow shock	124
7.1.1	Theoretical background	125
7.1.2	Billig's Approximation	127
7.2	Test case Cylinder	129
7.2.1	Analytical solution normal shock	130
7.3	LBM Solution	133
7.3.1	Bouzidi BCs Solution	134
7.3.2	Bouzidi Bounce Back with Normal BCs Solution	141

7.4	Conclusion	148
8	Conclusion	150
8.1	Achievements	150
8.1.1	Wedge	151
8.1.2	Supersonic cylindrical-nosed body	151
8.2	Future developments	152
A	Euler equations	154
B	Python code for new boundary conditions	156
C	Wedge Results	161
C.1	US3D Results	161
D	Second-order Equivalent Equations	164
E	Supersonic cylindrical-nosed body results	171
E.1	Bouzidi solution	171
E.2	Bouzidi Bounce-Back with Normal solution	172
	Bibliography	174

List of Tables

6.1	Wedge Parameters - Free Stream	81
6.2	Analytical Solution - Downstream shock field	84
6.3	US3D Solution - Downstream shock field	93
6.4	Relative (or absolute) errors between solutions: analytical - US3D	95
6.5	Input parameters in the LBM simulation	95
6.6	Results - LBM - Bouzidi	103
6.7	Errors between LBM Bouzidi - Analytic solutions	104
6.8	Errors between LBM Bouzidi - US3D solutions	105
6.9	Results - LBM - Bouzidi Bounce Back with Normal	113
6.10	Errors between LBM Bouzidi Bounce Back with Normal - Analytic solutions	114
6.11	Errors between LBM Bouzidi Bounce Back with Normal - US3D solutions	114
6.12	Absolute errors in ratios between LBM Bouzidi Bounce Back with Normal - Analytic solutions	116
6.13	Absolute errors in ratios between LBM Bouzidi Bounce Back with Normal - US3D solutions	116
6.14	Results - LBM - Bouzidi Bounce Back with Normal - $\lambda = 11$; $s = 1.6$; $\Delta x = 0.0025$	120
6.15	Errors between LBM Bouzidi Bounce Back with Normal - Analytic solutions - $\lambda = 11$; $s = 1.6$; $\Delta x = 0.0025$	120
7.1	Cylinder Parameters - Free Stream	130
7.2	Analytical Solution - Downstream normal shock - $y = 2$	132
7.3	Input parameters in the LBM simulation	133
7.4	Results - LBM - Bouzidi	140
7.5	Errors between LBM Bouzidi - Analytic solutions	140
7.6	Results - LBM - Bouzidi Bounce Back with Normal	147
7.7	Errors between LBM Bouzidi Bounce Back with Normal - Analytic solutions	147

List of Figures

1.1	An example of the three different scales used to describe matter (i.e. macroscopic, mesoscopic, and microscopic scales).	3
2.1	The probability distribution function for N_2 gas as a function of molecular velocity, c	16
2.2	Distribution function for N_2 , H_2 and CO_2 at room temperature as a function of molecular velocity, c	16
3.1	Sketch of a lattice Boltzmann one time step	34
4.1	Bounce-back boundary conditions, No-Slip case	48
4.2	Bounce-back boundary conditions, Slip case	49
4.3	Details of the collision process near the boundary for different distances between the boundary node and the wall.	53
5.1	Single lattice of the LBM with the four particle distribution functions moving along the four directions	59
5.2	Frame of reference (ξ, η) turned by a generic angle θ	63
5.3	Direction normal to wall	64
5.4	D_2Q_4 scheme, a convention adopted to indicate neighboring points .	66
5.5	Trivial case in which we have $\theta = \pi$	67
5.6	Case of a rebound treated with Bouzidi Bounce Back with Normal .	68
6.1	Oblique shock waves produced by a scramjet-powered hypersonic aircraft [116]	75
6.2	Shock waves on X-15 flight vehicle [118]	75
6.3	Supersonic flow over a wedge	76
6.4	Supersonic flow over a wedge - Strong and weak shock waves	77
6.5	Supersonic flow over a wedge - Detached bow shock	77
6.6	$\theta - \beta - Mach$ curve	79
6.7	Schematic description of the flow domain and angle definitions. . . .	80
6.8	Supersonic flow over a wedge - Analytical Solution	82

6.9	201 x 201 Uniform, Cartesian Computational Grid US3D	86
6.10	CFD Mach number field for the Mach 2.5 supersonic wedge on the 201x201 grid	88
6.11	Discontinuity of Mach number for various coordinates y	89
6.12	Numerical approximation of the shock wave obtained with US3D . .	89
6.13	Mach values for $x = 0.5 m$ - US3D	90
6.14	Pressure values for $x = 0.5 m$ - US3D	90
6.15	Density values for $x = 0.5 m$ - US3D	91
6.16	Normal wall velocity - US3D	91
6.17	Tangential wall velocity - US3D	92
6.18	Mach wall - US3D	92
6.19	Distance between shocks [m]. Solutions: analytical - US3D	94
6.20	Mach Number Field - LBM Bouzidi Solution	98
6.21	Leading Edge - LBM Bouzidi Solution	99
6.22	The different shock waves considered - LBM (Bouzidi) - Analytical - US3D	99
6.23	Density Field - LBM Bouzidi Solution	100
6.24	Pressure Field - LBM Bouzidi Solution	100
6.25	Momentum-x Field - LBM Bouzidi Solution	100
6.26	Momentum-y Field - LBM Bouzidi Solution	100
6.27	$ V $ Field - LBM Bouzidi Solution	100
6.28	Speed of Sound Field - LBM Bouzidi Solution	100
6.29	Normal Velocity Field - LBM Bouzidi Solution	101
6.30	Tangential Velocity Field - LBM Bouzidi Solution	101
6.31	Normal Velocity Outlet - LBM Bouzidi Solution	101
6.32	Tangential Velocity Outlet - LBM Bouzidi Solution	101
6.33	Mach Outlet - LBM Bouzidi Solution	102
6.34	Energy Outlet - LBM Bouzidi Solution	102
6.35	Density Outlet - LBM Bouzidi Solution	102
6.36	Pressure Outlet - LBM Bouzidi Solution	102
6.37	Velocity-x Outlet - LBM Bouzidi Solution	102
6.38	Velocity-y Outlet - LBM Bouzidi Solution	102
6.39	Mach Number Field - LBM Bouzidi Bounce Back with Normal Solution	106
6.40	Leading Edge - LBM Bouzidi Bounce Back with Normal Solution .	107
6.41	The different shock waves considered - LBM (Bouzidi with Normal) - Analytical - US3D	108
6.42	Density Field - LBM Bouzidi with Normal Solution	109
6.43	Pressure Field - LBM Bouzidi with Normal Solution	109
6.44	Momentum-x Field - LBM Bouzidi with Normal Solution	109
6.45	Momentum-y Field - LBM Bouzidi with Normal Solution	109
6.46	$ V $ Field - LBM Bouzidi with Normal Solution	109

6.47	Speed of Sound Field - LBM Bouzidi with Normal Solution	109
6.48	Normal Velocity Field - LBM Bouzidi with Normal Solution	110
6.49	Tangential Velocity Field - LBM Bouzidi with Normal Solution	110
6.50	Normal Velocity Outlet - LBM Bouzidi with Normal Solution	110
6.51	Tangential Velocity Outlet - LBM Bouzidi with Normal Solution	110
6.52	Mach Outlet - LBM Bouzidi with Normal Solution	111
6.53	Energy Outlet - LBM Bouzidi with Normal Solution	111
6.54	Density Outlet - LBM Bouzidi with Normal Solution	111
6.55	Pressure Outlet - LBM Bouzidi with Normal Solution	111
6.56	Velocity-x Outlet - LBM Bouzidi with Normal Solution	112
6.57	Velocity-y Outlet - LBM Bouzidi with Normal Solution	112
6.58	Mach Ratio Outlet - LBM Bouzidi with Normal Solution	115
6.59	Energy Ratio Outlet - LBM Bouzidi with Normal Solution	115
6.60	Density Ratio Outlet - LBM Bouzidi with Normal Solution	115
6.61	Pressure Ratio Outlet - LBM Bouzidi with Normal Solution	115
6.62	Mach Number Field - LBM Bouzidi Bounce Back with Normal Solution - $\lambda = 11$; $s = 1.6$; $\Delta x = 0.0025$	118
6.63	Mach Number Ratio Outlet - LBM Bouzidi Bounce Back with Normal Solution - $\lambda = 11$; $s = 1.6$; $\Delta x = 0.0025$	119
6.64	Mach Number Field - LBM Bouzidi Bounce Back with Normal Solution - $\lambda = 15$; $s = 1.6$; $\Delta x = 0.005$	121
6.65	Mach Number Outlet - LBM Bouzidi Bounce Back with Normal Solution - $\lambda = 15$; $s = 1.6$; $\Delta x = 0.005$	121
7.1	3D graphic representation of the reentry of the Dragon capsule (SpaceX)	124
7.2	Supersonic flow over a wedge - Detached bow shock	125
7.3	Flow over a blunt body traveling at supersonic speed.	126
7.4	Comparison of shock wave shapes for flow across cylinders computed and measured from references [132], [133], [134], [135], [136], and [137]	128
7.5	Schematic description of the flow domain	129
7.6	Billig's approximation	133
7.7	Mach Number Field - LBM Bouzidi Solution	134
7.8	The different shock waves considered - LBM (Bouzidi) - Analytical (Billig)	135
7.9	Velocity u_x Field - LBM Bouzidi Solution	136
7.10	Velocity u_y Field - LBM Bouzidi Solution	136
7.11	Velocity u_x for $x = 2$ m - LBM Bouzidi Solution	136
7.12	Velocity u_y for $x = 2$ m - LBM Bouzidi Solution	136
7.13	Mach for $x = 2$ m - LBM Bouzidi Solution	137
7.14	Energy for $x = 2$ m - LBM Bouzidi Solution	137

7.15	Density for $x = 2$ m - LBM Bouzidi Solution	137
7.16	Pressure for $x = 2$ m - LBM Bouzidi Solution	137
7.17	Mach number trend for $y = 2$	138
7.18	Density trend for $y = 2$	139
7.19	Pressure trend for $y = 2$	139
7.20	Energy trend for $y = 2$	139
7.21	Temperature trend for $y = 2$	139
7.22	Total Pressure trend for $y = 2$	139
7.23	u_x trend for $y = 2$	139
7.24	Mach Number Field - LBM Bouzidi Bounce Back with Normal Solution	141
7.25	The different shock waves considered - LBM (Bouzidi Bounce Back with Normal) - Analytical (Billig)	142
7.26	Velocity u_x Field - LBM Bouzidi Bounce Back with Normal Solution	143
7.27	Velocity u_y Field - LBM Bouzidi Bounce Back with Normal Solution	143
7.28	Velocity u_x for $x = 2$ m - LBM Bouzidi Bounce Back with Normal Solution	143
7.29	Velocity u_y for $x = 2$ m - LBM Bouzidi Bounce Back with Normal Solution	143
7.30	Mach for $x = 2$ m - LBM Bouzidi Bounce Back with Normal Solution	144
7.31	Energy for $x = 2$ m - LBM Bouzidi Bounce Back with Normal Solution	144
7.32	Density for $x = 2$ m - LBM Bouzidi Bounce Back with Normal Solution	144
7.33	Pressure for $x = 2$ m - LBM Bouzidi Bounce Back with Normal Solution	144
7.34	Mach number trend for $y = 2$	145
7.35	Density trend for $y = 2$	146
7.36	Pressure trend for $y = 2$	146
7.37	Energy trend for $y = 2$	146
7.38	Temperature trend for $y = 2$	146
7.39	Total Pressure trend for $y = 2$	146
7.40	u_x trend for $y = 2$	146
7.41	Comparison between the two solutions LBM and Billig	148
C.1	CFD pressure field for the Mach 2.5 supersonic wedge on the 201x201 grid	161
C.2	CFD density field for the Mach 2.5 supersonic wedge on the 201x201 grid	162
C.3	Velocity $ V $ values for $x = 0.5$ m - US3D	162
C.4	Energy E values for $x = 0.5$ m - US3D	162
C.5	Velocity u_x values for $x = 0.5$ m - US3D	163
C.6	Velocity u_y values for $x = 0.5$ m - US3D	163
C.7	Momentum Q_x values for $x = 0.5$ m - US3D	163

C.8	Momentum Q_y values for $x = 0.5 m$ - US3D	163
C.9	Speed of Sound c values for $x = 0.5 m$ - US3D	163
C.10	Temperature T values for $x = 0.5 m$ - US3D	163
E.1	Density Field - LBM Bouzidi Solution	171
E.2	Pressure Field - LBM Bouzidi Solution	171
E.3	Momentum-x Field - LBM Bouzidi Solution	172
E.4	Momentum-y Field - LBM Bouzidi Solution	172
E.5	$ V $ Field - LBM Bouzidi Solution	172
E.6	Speed of Sound Field - LBM Bouzidi Solution	172
E.7	Density Field - LBM Bouzidi Bounce Back with Normal Solution .	173
E.8	Pressure Field - LBM Bouzidi Bounce Back with Normal Solution .	173
E.9	Momentum-x Field - LBM Bouzidi Bounce Back with Normal Solution	173
E.10	Momentum-y Field - LBM Bouzidi Bounce Back with Normal Solution	173
E.11	$ V $ Field - LBM Bouzidi Bounce Back with Normal Solution	173
E.12	Speed of Sound Field - LBM Bouzidi Bounce Back with Normal Solution	173

Acronyms

BCs

Boundary Conditions

BE

Boltzmann Equations

BGK

Bhatnagar-Gross-Krook

CE

Chapman-Enskog

CFD

Computational Fluid Dynamics

CPU

Central Processing Unit

DBM

Discrete Boltzmann Method

DDF

Double Distribution Function

DOF

Degrees of Freedom

DSMC

Direct Simulation Monte Carlo

ELBM

Entropic Lattice Boltzmann Method

ESA TRP

ESA Technology Research Programme

FCHC

Face-Centered Hyper-Cubic

FDLBM

Finite Difference Lattice Boltzmann Method

FE

Finite Element

FV

Finite Volume

LBM

Lattice Boltzmann Method

LB

Lattice Boltzmann

LBFS

Lattice Boltzmann Flux Solver

LGA

Lattice Gas Automata

MRT

Multiple-Relaxation-Time

NS

Navier-Stokes

SOA

State Of the Art

US3D

Unstructured 3D

VLES

Very Large Eddy Simulation

WENO

Weighted Essentially Non-Oscillatory

WENO-LBFS

Weighted Essentially Non-Oscillatory Lattice Boltzmann Flux Solver

Chapter 1

Introduction

In many industrial, technical, biological, and environmental processes, an accurate estimation of the interaction between fluids and structures is crucial. Therefore, in the past two decades, research has intensified in this field. New progress in computing power and the creation of new numerical approaches have made it possible to approach the study of fluid dynamics problems by directly solving governing equations.

The term *Computational Fluid Dynamics* (CFD) refers to this subfield of fluid dynamics. The description of the fluid dynamics system, in this case, is more detailed than the experimental one because we are capable of estimating approximations of the velocity and pressure at each location in the discretized region. Also, compared to the empirical technique, studying the fluid dynamics problem using CFD is definitely less expensive and time-consuming.

Three different points of view can be used to study and analyze CFD. These three levels are shown in Figure 1.1.

- The Navier-Stokes equations [1], which regulate the fluid at a **macroscopic** scale, are solved in the first method, often known as the traditional approach to CFD. In Figure 1.1, the control volume considered contains a continuum of particles so a study at this level will give an idea of the average level.
- The Lattice Boltzmann (LB) approach [2], it is based on Boltzmann's kinetic equation [3], which has a **mesoscopic** point of view rather than the Navier-Stokes continuum assumption. In Figure 1.1, the mesoscopic technique uses a control volume that is located in the area around $x = (x, y, z)$.

- The behavior at the macroscopic level is determined by the dynamics of a system of molecules, which is how the problem is approached at the lowest level, from a **microscopic** molecular point of view, as shown in Figure 1.1,

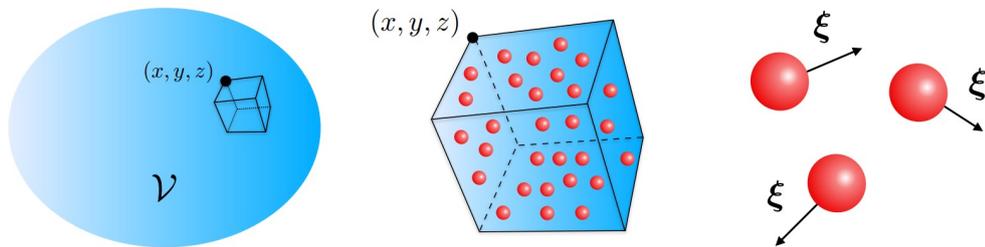


Figure 1.1: An example of the three different scales used to describe matter (i.e. macroscopic, mesoscopic, and microscopic scales).

For industrial applications, the Lattice Boltzmann Method (LBM) demonstrates a number of interesting benefits (simple algorithm, simplicity of dealing with complicated geometries, high parallel processing efficiency, provides for multi-physics issues, etc.); as a result, it is a potential numerical method for simulating complex fluid flow. However, most existing LBM applications only apply to incompressible flows. Since LBM is based on a bottom-up multiscale approach where the macroscopic quantities and their governing equations do not directly appear in the LBM scheme ¹, this could cause the limitation that occurs at the state of the art of LBM, although there is no certainty about this yet, so the causes of the low compressibility limitation are yet unknown.

There have been several methods proposed for the extension of LBM to compressible flows during the past few years. The goal of the ESA TRP research project LBMHYPE was to determine whether Lattice Boltzmann Methods was practical for hypersonic applications [4], or even possible. The LBMHYPE initiative conducted additional analysis of the potential and constraints of pure LBM schemes to model highly compressible flow characteristics. The physics that interact in highly compressible flows are very complex (shock waves, rarefaction, thermal exchanges, reactive multi-species, etc.). Even with the most advanced numerical techniques, simulating such intricately interconnected processes is incredibly difficult. In fact, this project's objective was to propose a first tool that can simulate the Euler equations in the high-Mach range using a Lattice Boltzmann scheme [5].

The proposed hyperbolic LBM schemes in [6] demonstrate promising capabilities both in terms of accuracy and computational cost. The idea is based on an

¹In other words, it requires applying sophisticated mathematical and numerical techniques to examine the links between the (microscopic) LBM scheme and the (macroscopic) flow behavior.

innovative vectorial scheme method in which each conservation equation of the Euler system is solved by a dedicated LB scheme. Once identified and developed in the first phase of the LBMHYPE project promising 1D and 2D LBM schemes for highly-compressible inviscid flows, in the second phase of the project, the focus was on studying the results provided by the LBM, comparing what was obtained with state-of-the-art CFD methods (US3D) for the defined set of 1D and 2D test cases.

Concerning the 1D tests performed, LBM's accuracy appears to be comparable to the majority of the standard CFD methods. It could be improved further by making specific improvements to locally smooth the spurious oscillations, such as the non-oscillating LBM system presented in [7].

Dealing with the two-dimensional tests performed, on which we will focus in this thesis, the LB approach is superior in terms of CPU, programming, and mesh creation compared to one of the top hypersonic codes in the world. However, optimal results were only obtained in the supersonic Forward Facing Step test case, a two-dimensional example introduced by Emery [8], where he examined various schemes. The best results were obtained among the 2D test cases for this case due to the fact that the geometry conformed to the Cartesian lattice. A second test case investigated in the LBMHYPE project was a supersonic wedge flow. Since the wedge was a slanted surface that did not conform with the Cartesian LBM lattice, the accuracy of the scheme was degraded. The application of the slip wall boundary condition on such a slanted surface is the focus of the present work. Since the scheme that has been explored in-depth, D_2Q_{4444} only takes into account horizontal and vertical velocities, there is a further difficulty in this case, so the contributions to the horizontal and vertical velocities must be approximations in order to account for the slip boundary condition.

In spite of the LBMHYPE project being able to reproduce the shock angle with a small discrepancy value and the computational time being much quicker with the LBM technique than with US3D, the absence of non-aligned slip-wall BC is degrading the solution, hiding the accuracy of the LBM developed in the project, as we will see in sections 6.4.2 and 7.3.1 of this thesis. The aim of the work that will be presented in this thesis is to fix the boundary conditions to take into consideration the geometric inclination of the surfaces, hence improving the accuracy of the method in a wider range of applications to be more competitive in industrial applications.

In developing this master thesis, a great deal of help was given by Prof. Loic Gouarin of École Polytechnique for the support in developing the new code, and in understanding the Python package `pylbm` [4], by Prof. Benjamin Graille of Université Paris-Saclay, who helped me in the theoretical understanding of the new boundary conditions implemented in this work and finally Doc. Zuheyr Alsalihi of von Karman Institute for Fluid Dynamics.

1.1 Conventional CFD

About the conventional method, only a few particular instances allow for the closed-form solution of the Navier-Stokes equations. In all practical applications, these equations are mostly solved using finite element (FE) or finite volume (FV) techniques which are very common but have a number of limitations and there is one that constitutes the strong point of the LBM: the cost. The major drawback of the CFD is the cost, due to the following reasons:

- The first aspect that increases the computational cost of a CFD simulation is certainly the spatial resolution. It turns out to be obvious that higher spatial resolution requires a finer subdivision of the computational grid, which implies an increase in the number of computational points. More computational points imply greater computational complexity and require more computational resources.
- Similar to the previous point, the temporal resolution also leads to an increase in computational cost. Some CFD simulations require high temporal resolution to correctly capture transient or unsteady phenomena. The higher temporal resolution requires more integration steps over time, i.e., a significant increase in computational cost.
- Another aspect that increases computational cost may be the physical model. The use of more complex physical models, such as advanced turbulence models or real transport models, may require more computational power. These models often require more complex differential equations to solve, increasing the overall computational complexity. Without going into complex turbulence models, since the convective term is non-local and non-linear, every traditional CFD solver must handle it, which requires a significant amount of computational work².

²This term is sometimes ignored, as in the case of the potential flow hypothesis. As is known, this theory rules the macroscopic behavior of fluids under the assumptions of irrotational flow and inviscid fluid, which represents an ideal condition that is typically not applicable in real-world situations.

- Even in the case of complex geometries, a lot of CPU can be spent. If the geometry of the simulation domain is complex, for example with intricate shapes or curved surfaces, it may be necessary to use a more detailed computational grid to represent it accurately. This increases the number of computational points and, consequently, the computational cost. Demanding meshes are required to study the interaction with solid bodies (i.e., prism layer near the boundary layers), while an even more striking case, and close to what will be dealt with in this thesis is the mesh refinement, which is necessary in case of supersonic CFD simulations in the proximity of shock waves, which again necessitates significant computing costs.
- It is also logical that the size of the simulation domain affects the computational cost. The number of computational points increases with the size of the domain, i.e., increasing the computational cost.
- Another aspect is the numerical method to be used, and the order of accuracy associated with it. More accurate methods require more computational power to obtain accurate results.
- Finally, the simulation time required also affects CPU hours. If long-term CFD simulations are to be run or a large number of parametric simulations are to be performed, this will increase the overall computational cost.

These limitations recommend using a different strategy to solve the CFD challenge.

1.2 Lattice Boltzmann Method

The Lattice Boltzmann Method, a new approach to CFD, has been introduced and refined during the past 25 years. The Navier-Stokes equations are no longer the focus of this new theory; instead, the so-called Lattice Boltzmann equation (LBE), which makes use of statistical particle distribution functions, is the focus. Macroscopic quantities such as density and momentum can be defined using these particle distribution functions. It can be demonstrated that in the continuous limit, these macroscopic quantities satisfy the Navier-Stokes equations [9]. Compared to the Navier-Stokes equations, the LBM provides a lower scale of description, but a higher level of description when compared to molecular dynamics, which directly accounts for particle collisions. Therefore, the Lattice-Boltzmann approach is referred to as a mesoscopic model to distinguish it from the microscopic size of molecular dynamics and the macroscopic scale of the Navier-Stokes equations. The similarity with the kinetic theory leads to the assumption that the numerical solution approaches the solution of the Navier-Stokes equations when the discrete

Knudsen number (the ratio of the mesh size over a reference length of the simulation) decreases to zero.

The Lattice Gas Automata (LGA) [10], [11], [12] developed during the 1970s and 1980s provided the basis for the LBM in significant part. In 1988, McNamara & Zanetti [13] developed the first Lattice Boltzmann model to address the LGA's primary flaws. Thus, the LGA and LBM both started out with the same concept, if the system's domain is discretized with enough symmetry, it is possible to define simplified dynamics that can recover the macroscopic Navier-Stokes in the continuum limit and only requires that mass and momentum be conserved during the collision. The Bhatnagar-Gross-Krook (BGK) with just one relaxation parameter, which was previously used to simulate collision in the Boltzmann Equation (BE) [14], [15], was the collision term employed in [13]. This is the first indication of a link between the BE and the LBE. Despite being developed at the particle level, the LBM's principal application focuses on macroscopic phenomena. A multi-scale method known as Chapman-Enskog (CE) [9] expansion is used to get the macroscopic equations. The study [16] demonstrates how the Euler equations can be reconstructed on a fast convective time scale according to the Chapman-Enskog expansion. Another indication of the relationship with the Boltzmann equation is the fact that the CE approach was initially created to extract the macroscopic equations for the Boltzmann equation (BE) in [9]. In [17] there is an example of a CE for the LBE. He and Luo explicitly state this relationship in [18], where they demonstrate how the LBE can be viewed as a discretization of the BE. Strong theoretical roots for LBM have been established by this work, which somehow were missing in the first formulation derived from the LGA. Only historical connections remain between the LBM and LGA now, and [18]'s interpretation is the one that is most frequently used.

Since its debut, the LBM has been the focus of ongoing study and development. Its success is primarily due to:

- The amazing computational efficiency
- The ease of managing complex geometry
- The simplicity of the code
- The ease of parallelizing

One disadvantage, however, of the LBM is that it is not simple to control the simulations and analyze the results.

Nowadays, research is focusing on the development and understanding of three LBM-related problems:

- Appropriate boundary conditions
- New more flexible and stable collision terms
- Multi-phase models

In contrast to conventional CFD, in LBM the imposition of the boundary conditions is carried out directly on the particle distribution functions. The so-called Bounce-Back technique [19], which is pretty simple and one of the reasons LBM works well with complex geometries, is typically used to implement the wall boundary conditions. Bouzidi et al.'s [20] boundary conditions are a generalization of bounce-back boundary conditions, allowing the placement of the boundary between two grid layers at any distance, rather than just halfway as in the '*conventional*' Bounce-Back.

However, as we will see in this thesis work, this simple implementation brings undesirable effects with the scheme that was developed within the LBMHYPE project. Moreover, more complex schemes have been put out [21]. For pressure and velocity boundary conditions, several techniques are proposed in [22], [23], [24] and [25]. The Multiple-Relaxation-Time (MRT) collision model has been developed to overcome various issues with the BGK collision term [26], [27] [28]. By modifying the model's parameters, it is possible to increase stability and get rid of some undesirable BGK model side effects, like the dependence of the simulation geometry on the relaxation parameter.

1.3 Contents of each chapter of this thesis

In summary, these are the contents of each chapter of this master's thesis:

- **Chapter 2:** In this introductory chapter, the theoretical background necessary to fully understand LBM is introduced, then the kinetic theory will be discussed, the particle distribution function will be defined, to finally deal with the Boltzmann equation and collision operator approximations.
- **Chapter 3:** In this introductory chapter, LBM is discussed, first introducing an in-depth theoretical background on LBM, focusing on the compressible field. From section 3.3, LBM will be treated, with the innovative approach taken in this thesis, namely Vectorial Schemes, described in section 3.4.

- **Chapter 4:** In chapter four the main topic of this master's thesis, namely the application of boundary conditions, is introduced. Sections 4.2 and 4.3 discuss both the well-known boundary conditions of Bounce Back and Bouzidi, which are the foundations of the Bouzidi Bounce Back with Normal boundary conditions.
- **Chapter 5:** This is the main chapter of the entire thesis, as it deals with the newly implemented boundary conditions. After introducing the new interpretation of particle distribution functions as flux, it is explained how to add this corrective factor so that at the wall the slip condition is imposed, regardless of the geometry considered.
- **Chapter 6:** This is the first chapter in which the results obtained in this thesis are shown; in fact, the solution obtained with the LBM is shown, imposing the Bouzidi Bounce Back with Normal condition on the surface of a wedge in an inviscid supersonic current.
- **Chapter 7:** The second results chapter deals with the LBM solution of a cylindrical-nosed body in an inviscid supersonic current. This simulation, more than showing the accuracy of the method, aims to demonstrate that the Bouzidi Bounce Back with Normal boundary conditions is also applicable in a non-rectilinear surface.
- **Chapter 8:** The concluding chapter summarizes what has been achieved in this thesis, but more importantly it deals with possible future developments of the pylbm project.

Chapter 2

The Boltzmann Equation

Before moving on to **LBM**, it is essential to be familiar with the terms and ideas of kinetic theory. According to the kinetic theory of gases, a gas is composed of a lot of tiny particles (atoms or molecules) moving randomly and interacting consistently with each other. By taking into account the molecular composition and movement of gases, the kinetic theory describes the macroscopic characteristics of gases, such as pressure, temperature, and volume. Boltzmann's equation is the basis of the kinetic theory, which will be dealt with following the approach presented in [19].

2.1 Kinetic Theory

The molecules and sub-molecules are the basic elements of all-natural materials. These molecules can be pictured as solid spheres that are moving *randomly* in a free region. Mass, momentum, and energy conservation are satisfied. Thus, it is possible to use Newton's second law (on the conservation of momentum), which establishes that the rate at which momentum changes is equal to the net applied force.

$$F = \frac{d(mc)}{dt} \quad (2.1)$$

where \mathbf{m} is the particle's mass, \mathbf{c} is its velocity vector, \mathbf{t} is the time, and \mathbf{F} stands for the intermolecular and external forces.

The equation can be made simpler for a constant mass as follows:

$$F = m \frac{d(c)}{dt} = ma \quad (2.2)$$

where \mathbf{a} is the acceleration vector.

After the application of an external force, \mathbf{F} , to a particle of mass \mathbf{m} , the particle's location changes from \mathbf{r} to $\mathbf{r} + \mathbf{c}dt$, and its velocity changes from \mathbf{c} to $\mathbf{c} + \mathbf{F}\frac{dt}{m}$. Without an external force, the particle freely moves from one place to another without modifying its direction or speed, assuming no collision occurs. As the internal energy of the system rises (for example, by heating the system), the magnitude of the particle velocity increases and the interaction between the particles grows. The domain walls are constantly being hit by the particles (molecules). In a macroscopic sense, pressure is the force these actions produce per unit area. From this basic framework, we can see that there is a relationship between temperature and pressure. As temperature rises, which also means molecules' kinetic energies rise, we expect that the probability of particles hitting the domain's wall rises.

Let's assume that there is a single particle on a box of length \mathbf{L} continuously hitting the box's wall while moving at a speed of \mathbf{c}_x (in the x direction). If the collision is perfectly elastic, the force a particle applies to one end is equal to the rate of change of the momentum. If Δt is the time between collisions, knowing both the size of the box and the velocity of the particle, it is not difficult to estimate the time between two consecutive collisions in the wall box ($\Delta t = \frac{2\mathbf{L}}{\mathbf{c}_x}$).

$$F\Delta t = mc_x - (-mc_x) = 2mc_x \rightarrow F = m\frac{c_x^2}{L} \quad (2.3)$$

Results are generalizable to \mathbf{N} particles. \mathbf{N} particles generate a total force that is proportional to $\mathbf{F} \propto \mathbf{N}\frac{mc^2}{L}$. Typically, $\mathbf{c}^2 = \mathbf{c}_x^2 + \mathbf{c}_y^2 + \mathbf{c}_z^2$; where \mathbf{c}_x , \mathbf{c}_y , and \mathbf{c}_z are the respective velocity components in the \mathbf{x} , \mathbf{y} , and \mathbf{z} directions. It is reasonable to believe that these elements are equal, hence $\mathbf{c}^2 = 3\mathbf{c}_x^2$ follows.

Since the definition of pressure is a force per unit area that is perpendicular to the force vector $\mathbf{P} = \frac{\mathbf{F}}{\mathbf{A}}$, the pressure that \mathbf{N} particles impose on the ends of the tube is equal to:

$$P = N\frac{mc^2}{3LA} = N\frac{mc^2}{3V} \quad (2.4)$$

where \mathbf{V} is the volume, which is equal to $\mathbf{V} = \mathbf{L}\mathbf{A}$.

This easy description of molecular motion connects the kinetic energy to the macroscopic pressure.

$$P = \frac{N}{V}\frac{mc^2}{2}\frac{2}{3} = \frac{2}{3}nE_K \quad (2.5)$$

where \mathbf{n} is the number of molecules per unit volume and \mathbf{E}_K is the kinetic energy. We ignored the effects of molecule interaction and molecular size in this straightforward model (the ideal gas model). However, the results are surprisingly accurate for gas at ambient temperature. The particle in real systems has volume, and there is particle collision.

Finally, to add the link between pressure and temperature, the following relationship is well-known for a perfect gas:

$$PV = nRT \quad (2.6)$$

where n is number of moles ¹ and R is the gas constant. From equation 2.5 and 2.6, introducing Boltzmann constant ($k = \frac{R}{N_A} = 1.38 \times 10^{-23} \text{ J/K}$), it is possible to obtain an expression for the kinetic energy as a function of temperature.

$$E_k = \frac{3}{2}kT \quad (2.7)$$

2.2 Distribution Function

Despite Newton's second law (i.e. the governing equation) being well-known, Maxwell (1831–79) observed in 1859 that dealing with a large number of molecules makes it challenging to formulate an equation. A macroscopic system cannot follow the path of every molecule, so the concept of averaging was introduced. According to Maxwell, it is not crucial the knowledge of each molecule's position and speed at every moment in time.

The distribution function, which measures the percentage of molecules in a container with velocities that fall within a certain range at a given instant in time, is a key factor in evaluating the impact of molecules. When the molecules of a gas that has a wide range of velocities collide, the fast molecules give the slow molecules momentum. The collision has the effect of conserving momentum. The distribution function for gas at thermal equilibrium, when the gas is dispersed uniformly throughout the domain, is not a function of time; the only unknown is the velocity distribution function.

In a gas of N particles, the amount of particles with x-direction velocities between \mathbf{c}_x and $\mathbf{c}_x + d\mathbf{c}_x$ is equal to $Nf(\mathbf{c}_x)d\mathbf{c}_x$, where the percentage of particles with velocities between \mathbf{c}_x and $\mathbf{c}_x + d\mathbf{c}_x$ in the x-direction is represented by the function $f(\mathbf{c}_x)$. The probability distribution function can be defined in a similar manner for the y and z-direction. Then, the probability for the velocity to lie down between \mathbf{c}_x and $\mathbf{c}_x + d\mathbf{c}_x$, \mathbf{c}_y and $\mathbf{c}_y + d\mathbf{c}_y$ and \mathbf{c}_z and $\mathbf{c}_z + d\mathbf{c}_z$, will be $Nf(\mathbf{c}_x)f(\mathbf{c}_y)f(\mathbf{c}_z)d\mathbf{c}_x d\mathbf{c}_y d\mathbf{c}_z$ ²

As long as any direction can be x, y, or z, the distribution function should depend only on the particle speed and not on the direction:

$$f(c_x)f(c_y)f(c_z) = \Phi(c_x^2 + c_y^2 + c_z^2) \quad (2.8)$$

¹ $n = N/N_A$, where N_A is Avogadro's number

²It is crucial to note that if that equation is integrated (summed) over all possible values of the velocities the result will be the total number of particles N .

where Φ is an unknown function that needs to be found. The distribution function's value ought to be positive (between zero and unity) ³.

There are two types of functions that could have the property of equation 2.8: exponential and logarithmic. It could be demonstrated that the distribution function's correct form should be such as:

$$f(c_x) = Ae^{-Bc_x^2} \quad (2.9)$$

where \mathbf{A} and \mathbf{B} are constants.

From the properties of exponential functions, choosing this form for the f-functions, it follows that the distribution in terms of the particle speed c is obtained by multiplying the probability distributions for the three directions. It is crucial to consider how particles are distributed in velocity space, a three-dimensional space (\mathbf{c}_x , \mathbf{c}_y , and \mathbf{c}_z), where each particle is represented by a point with coordinates that match its velocity.

2.2.1 Maxwell distribution function

Since a particle moving at speed \mathbf{c} has kinetic energy $\frac{1}{2}mc^2$, we may utilize the probability distribution function to determine the average kinetic energy per particle by using the following formula:

$$\overline{\frac{1}{2}mc^2} = \frac{\int_0^\infty \frac{1}{2}mc^2 f(c)dc}{\int_0^\infty f(c)dc} \quad (2.10)$$

where the total energy is the numerator, while the total number of particles is the denominator.

When the value of $f(c)$ is substituted according to 2.9 the integrals become:

$$\overline{\frac{1}{2}mc^2} = \frac{3m}{4B} \quad (2.11)$$

Since the link between kinetic energy and temperature is known from 2.7:

$$\overline{\frac{1}{2}mc^2} = \frac{3}{2}kT = \frac{3m}{4B} \quad (2.12)$$

Then it is possible to obtain the value of the constant $\mathbf{B} = \frac{m}{2kT}$. Since we factored out the number of particles \mathbf{N} in our definition of $\mathbf{f}(\mathbf{c})$, the constant of proportionality is obtained by integrating over all speeds and setting the result equal to one.

The final result in terms of the distribution function is:

$$f(c) = 4\pi \left(\frac{m}{2\pi kT} \right)^{3/2} c^2 e^{-\frac{mc^2}{2kT}} \quad (2.13)$$

³As a result, velocity is squared in 2.8 to prevent negative magnitude

This equation is the so-called Maxwell-Boltzmann distribution function 2.13, whose rises parabolically from zero for slow speeds, achieves a maximum value, and then falls exponentially [29]. As the temperature rises, the maximum's location moves to the right (i.e. the number of molecules with high velocities rises as the temperature rises). The integral of 2.13 from 0 to ∞ is always one, by definition. Since particle velocities change constantly across a vast range, the probability of finding a particle that has a specific velocity is zero. Finding the likelihood of a particle or particles within a range of velocities rather than at a particular velocity is an important question. The equation 2.13 must therefore be integrated within that range of velocity. Determining the probability of a particle or particles within a range of velocities rather than at a particular velocity is an interesting matter, for that reason 2.13 must be integrated within that range of velocity.

These are some characteristic speeds obtainable from the Maxwell distribution function 2.13:

- The speed that is most probable is equal to

$$\sqrt{\frac{2kT}{m}} \quad (2.14)$$

This equation can be achieved by setting the distribution function's derivative with respect to velocity to zero and then solving for velocity.

- The average speed is equivalent to

$$\langle c \rangle = \sqrt{\frac{8kT}{\pi m}} \quad (2.15)$$

This is the average weighted speed. The distribution function can be integrated from zero to infinity to obtain, as

$$\langle c \rangle = \int_0^{\infty} c f(c) dc \quad (2.16)$$

- The root-mean-average speed is equal to

$$\langle c \rangle^2 = \int_0^{\infty} c^2 f(c) dc = \frac{3kT}{m} \quad (2.17)$$

The mean average speed is equal to $(c_x^2 + c_y^2 + c_z^2)^{1/2}$ and average speed is equal to $\frac{c_x^2 + c_y^2 + c_z^2}{3}$.

- The root mean squared speed of a molecule is

$$\langle c \rangle = \sqrt{\langle c^2 \rangle} = \sqrt{\frac{3kT}{m}} \quad (2.18)$$

In general, lighter gases (lower molecular weight) have faster molecular speeds than heavier gases. The probability function for H_2 (2 kg/kmole), N_2 (28 kg/kmole), and CO_2 (44 kg/kmole) is shown in Figure 2.2. As we can notice, most N_2 molecules move at around 500 m/s at ambient temperature, while most hydrogen molecules move at about 1,600 m/s.

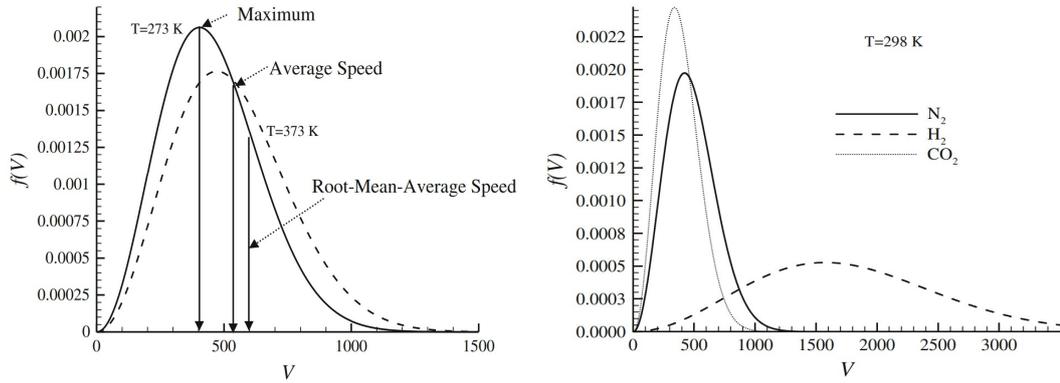


Figure 2.1: The probability distribution function for N_2 gas as a function of molecular velocity, c **Figure 2.2:** Distribution function for N_2 , H_2 and CO_2 at room temperature as a function of molecular velocity, c

2.2.2 Boltzmann Distribution

The theory developed by Ludwig Boltzmann [30], an Austrian physicist of the 19th century, allowed for dealing with the distribution of energy in complex thermodynamic systems. Previous to Boltzmann, the distribution of energy in thermodynamic systems had been described by James Clerk Maxwell through the Maxwell-Boltzmann distribution, as just seen in the previous paragraph, but this description was limited to systems consisting of a few atoms or molecules.

Boltzmann generalized the Maxwell distribution for systems of arbitrary size and furthermore was the first to realize the deep connection between the thermodynamic concept of entropy and the statistical analysis of possible states of a large system. In particular, Boltzmann demonstrated that the increase in entropy of a system over time is a consequence that the system tends to reach the state of maximum disorder possible, i.e. the one corresponding to the largest number of possible microscopic arrangements. In other words, Boltzmann showed that the disorder

and entropy of a system are closely related to the distribution of energy among its microscopic components and that the system tends to evolve in such a way as to maximize the number of possible microscopic configurations, i.e. the state of maximum disorder possible. This theory is one of the foundations of statistical physics and has important applications in numerous fields, from thermodynamics to chemistry to biology. According to Boltzmann, macroscopic values corresponding to thermal equilibrium have a far higher number of microscopic states available for a given energy.

By the relation $S = k \log(W)$, Boltzmann presented a link between a system's thermodynamic entropy \mathbf{S} and the number \mathbf{W} of microscopic states that are available to it at a given energy \mathbf{E} .

The probability of being in a specific state for energy \mathbf{E} is proportional to $e^{-\frac{E}{kT}}$ for any system large or small in thermal equilibrium at temperature T , according to Boltzmann.

$$f(E) = Ae^{-\frac{E}{kT}} \quad (2.19)$$

which is the Boltzmann Distribution.

Let us look at the kinetic energy of molecules moving in the x direction:

$$E = \frac{1}{2}mc_x^2 \quad (2.20)$$

The probability function integrated for all values of velocity (from minus to plus infinity) for a normalized probability function should be one.

$$\int_{-\infty}^{\infty} Ae^{-\frac{mc_x^2}{2kT}} dc = 1 \quad (2.21)$$

The probability of finding velocity \mathbf{c}_x is

$$f(c_x) = \sqrt{\frac{m}{2\pi kT}} e^{-\frac{mc_x^2}{2kT}} \quad (2.22)$$

The probability of the three-dimensional velocity (\mathbf{c}), which is $c^2 = c_x^2 + c_y^2 + c_z^2$, is given by the multiple of each function $\mathbf{f}(\mathbf{c}_i)$:

$$f(c) = f(c_x) \cdot f(c_y) \cdot f(c_z) = \left[\sqrt{\frac{m}{2\pi kT}} \right]^3 e^{-\frac{mc^2}{2kT}} \quad (2.23)$$

It should be emphasized that the above equation 2.23 ignores that there are numerous ways to increase velocity. This distribution function must be multiplied by the factor $4\pi c^2$ (which is the surface area of a sphere in the phase space) in order to take into consideration the density of velocity states that are available to particles when moving from this expression to the Maxwell speed distribution. In

other words, the Maxwell distribution function 2.13 is included, in fact, equation 2.23 is produced when the Maxwell distribution function 2.13 is integrated across a sphere's surface in phase space.

At equilibrium, an ideal gas has a particular distribution function (Maxwell distribution function). Maxwell, however, made no indication of how balance is obtained. This was one of Boltzmann's innovative contributions, upon which the **LBM** is based.

2.3 The Boltzmann Equation

The development of statistical mechanics explains and makes predictions about how the microscopic properties influence the macroscopic one, such as viscosity, thermal conductivity, and diffusion coefficient. That is one of Ludwig Eduard Boltzmann's (1844–1906) greatest accomplishments. Instead of tracking every particle as in molecular dynamic simulations, the distribution function ⁴ is used in **LBM**. The technique significantly reduces the computational load on the CPU. The next two sections 2.3.1 and 2.3.2, deal with the Boltzmann Equation and the BGK approximation through the approach presented in [19].

2.3.1 Boltzmann Transport Equation

As indicated in the paragraph 2.2, the distribution function $\mathbf{f}(\mathbf{r}, \mathbf{c}, \mathbf{t})$ can be used to explain the statistical description of a system. A gas molecule of unit mass will have its position change from \mathbf{r} and $\mathbf{r}+d\mathbf{r}$ and its velocity change from \mathbf{c} and $\mathbf{c}+d\mathbf{c}$ in response to an external force \mathbf{F} .

If there are no molecular collisions, the number of molecules $\mathbf{f}(\mathbf{r}, \mathbf{c}, \mathbf{t})$, before the application of the external force is equal to the number of molecules following the disturbance $\mathbf{f}(\mathbf{r} + \mathbf{c} \cdot d\mathbf{t}, \mathbf{c} + \mathbf{F} \cdot d\mathbf{t}, \mathbf{t} + d\mathbf{t})$.

$$\mathbf{f}(r + c \cdot dt, c + F \cdot dt, t + dt)dr \cdot dc - \mathbf{f}(r, c, t)dr \cdot dc = \mathbf{0} \quad (2.24)$$

However, there will be a difference in the number of particles in the interval $d\mathbf{r} \cdot d\mathbf{c}$ if there are molecular collisions. The collision operator $\mathbf{\Omega}$ is the rate of change between the distribution function's final and starting status.

⁴Probability of finding particles within a specified range of velocities at a certain location at a given time, or in other words the number of molecules at time t that are positioned between r and $r+dr$ and have velocities between c and $c+Fdt$

Thus, the following 2.25 is how the equation for the evolution of the number of molecules can be expressed:

$$\mathbf{f}(r + c \cdot dt, c + F \cdot dt, t + dt)dr \cdot dc - \mathbf{f}(r, c, t)dr \cdot dc = \mathbf{\Omega}(f)dr \cdot dc \cdot dt \quad (2.25)$$

By dividing the previous equation by $d\mathbf{t} \cdot d\mathbf{r} \cdot d\mathbf{c}$ and as $d\mathbf{t} \rightarrow \mathbf{0}$, the result is the following:

$$\frac{df}{dt} = \mathbf{\Omega}(f) \quad (2.26)$$

According to the equation above 2.26, the distribution function's overall rate of change is equal to the rate of collision. Considering that \mathbf{f} is a function of \mathbf{r} , \mathbf{c} , and \mathbf{t} , we can expand the total rate of change as:

$$\frac{d\mathbf{f}}{dt} = \frac{\partial \mathbf{f}}{\partial r} \frac{dr}{dt} + \frac{\partial \mathbf{f}}{\partial c} \frac{dc}{dt} + \frac{\partial \mathbf{f}}{\partial t} = \frac{\partial \mathbf{f}}{\partial r} c + \frac{\partial \mathbf{f}}{\partial c} a + \frac{\partial \mathbf{f}}{\partial t} \quad (2.27)$$

In 2.27 the vector \mathbf{r} has been written in the three-dimensional Cartesian coordinate system as $\mathbf{r} = x\hat{\mathbf{i}} + y\hat{\mathbf{j}} + z\hat{\mathbf{k}}$, where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ are unit vectors along the x , y , and z axes, respectively. Furthermore, \mathbf{a} is the acceleration and can be linked to force \mathbf{F} through Newton's second law.

The Boltzmann transport equation 2.26 can be expressed as follows:

$$\frac{\partial \mathbf{f}}{\partial t} + \frac{\partial \mathbf{f}}{\partial r} \cdot c + \frac{F}{m} \cdot \frac{\partial \mathbf{f}}{\partial c} = \mathbf{\Omega} \quad (2.28)$$

where, in order to solve, we must identify the $\mathbf{\Omega}$, which is a function of \mathbf{f} .

The Boltzmann equation for a system with no outside force is expressed as:

$$\frac{\partial \mathbf{f}}{\partial t} + c \cdot \nabla \mathbf{f} = \mathbf{\Omega} \quad (2.29)$$

If $\mathbf{\Omega}$ is explicitly known, the relation above 2.29 is an advection equation with a source term ($\mathbf{\Omega}$) or an advection equation with a response term, and it can be solved precisely along the characteristic lines that are tangent to the vector \mathbf{c} . The issue is that 2.29 is an integro-differential equation, which is challenging to solve. Furthermore, $\mathbf{\Omega}$ is a function of \mathbf{f} .

Following is the relationship between the aforementioned equation 2.29 and macroscopic elements like fluid density ρ , fluid velocity vector \mathbf{u} , and internal energy e ⁵,

⁵The internal energy can be represented using the kinetic theory (2.1) as $e = \frac{3}{2m}kT$, i.e. the mass unit version of the relation 2.7

i.e the equations of conservation of mass, momentum, and energy.

$$\begin{aligned}\boldsymbol{\rho}(r, t) &= \int m f(r, c, t) dc \\ \boldsymbol{\rho}(r, t) \mathbf{u}(r, t) &= \int m c f(r, c, t) dc \\ \boldsymbol{\rho}(r, t) \mathbf{e}(r, t) &= \frac{1}{2} \int m u_a^2 f(r, c, t) dc\end{aligned}\tag{2.30}$$

where \mathbf{m} is the molecular mass and \mathbf{u}_a the particle velocity relative to the fluid velocity ($\mathbf{u}_a = \mathbf{c} - \mathbf{u}$).

2.3.2 The BGK Approximation

The Boltzmann equation is challenging because the collision term is complicated to compute. It is unlikely that two body crashes will have a significant impact on the values of many measured quantities [31]. As a result, it is possible to roughly approximate the collision operator with a simple operator without materially changing the outcome of the solution. In 1954, Bhatnagar, Gross, and Krook (**BGK**) presented a simplified model for the collision operator [32]. The replacement for the collision operator is the following:

$$\boldsymbol{\Omega} = \omega(f^{eq} - f) = \frac{1}{\tau}(f^{eq} - f)\tag{2.31}$$

The relaxation factor is τ , and the coefficient $\omega = \frac{1}{\tau}$ is known as the collision frequency. The Maxwell-Boltzmann distribution function, denoted by \mathbf{f}^{eq} , stands for the local equilibrium distribution function. The Boltzmann equation 2.29 (without external forces) can be approximated by the BGK approximation as follows:

$$\frac{\partial f}{\partial t} + c \cdot \nabla f = \frac{1}{\tau}(f^{eq} - f)\tag{2.32}$$

The aforementioned equation 2.32 is discretized and presumed to be true along a set of directions in the Lattice Boltzmann approach. In light of this, the discrete Boltzmann equation can be expressed as follows:

$$\frac{\partial f_i}{\partial t} + c_i \nabla f_i = \frac{1}{\tau}(f_i^{eq} - f_i)\tag{2.33}$$

In CFD simulations, the aforementioned equation takes the place of the Navier-Stokes equation and is the foundation of the lattice Boltzmann method. Boltzmann equation can be used to derive the Navier-Stokes equation.

Regarding equation 2.33, we can say the following:

- The equation is a linear partial differential equation
- The equation has a source term and resembles an advection equation
- The advection is represented by the right-hand side of the equation (streaming)
- The collision process, or source term, is represented by the left-hand side

Equation 2.33 can be discretized as follow:

$$f_i(r + c_i\Delta t, t + \Delta t) = f_i(r, t) + \frac{\Delta t}{\tau} [f_i^{eq}(r, t) - f_i(r, t)] \quad (2.34)$$

The kind of issue that needs to be resolved is determined by the local equilibrium distribution function with a relaxation time. It is important to mention that by simply defining a different equilibrium distribution function and source term, this equation can be used for a variety of physical phenomena, simply by considering an external force. It is simple to add a source term (force term) to the equation above.

It is possible to solve the partial differential equation 2.33 by using finite difference or finite volume. This method was employed by several writers to address fluid dynamics issues on irregular grids. The solution domain for LBM must be partitioned into lattices. The fictitious particles (i.e. the particle distribution function) are located at each node of the lattice. Some of these particles move to nearby nodes in the directions that have been predetermined. The configuration of the lattice determines the number of directions and linkages.

Despite BGK (Bhatnagar-Gross-Krook) model being the most common way of discretizing the collision operator ⁶ in the LBM, this model has certain limitations due to its approximations, which make it not suitable for supersonic flows:

- Linearization of collision frequencies: in the BGK model, a linear collision frequency is assumed, which can lead to inaccuracies in simulations of supersonic flows, where collision frequencies can vary substantially.
- Isothermal approximation: the BGK model assumes that the kinetic energy of particles is isothermally distributed. This approximation may become inaccurate for supersonic flows, where there may be significant variations in the local temperature.

⁶The operator that describes how the probability distributions of particles change due to interactions between them

- Numerical stability: the BGK model may suffer from numerical instability in the presence of supersonic flows, especially when using high-resolution discretization schemes.

In order to overcome these problems, the Multiple-Relaxation-Time (MRT) method can be used to discretize the collision operator. The MRT method has certain advantages over the BGK model, especially for supersonic flows, which will be analyzed in the following section.

2.3.3 Multi Relaxation Times shemes

The MRT (Multiple-Relaxation-Time) is a multiple-relaxation-time Boltzmann equation, introduced in [26], and used in [33] to simulate fluid flows in three dimensions. In comparison to the BGK model, the MRT lattice Boltzmann equation is a more sophisticated variant of the lattice Boltzmann equation. The MRT model contains various configurable relaxation times, which gives it additional flexibility in the creation of equilibrium functions and significantly increases numerical stability. Certainly, this greater sophistication and flexibility of the method brings some advantages over the BGK model, especially for supersonic flows:

- Greater control over collision frequencies: the MRT method allows different collision frequencies to be set for different modes (i.e. for different components of the particle probability distributions). This allows greater accuracy in modeling collisions in supersonic flows.
- Better approximation of thermodynamic properties: the MRT method can better capture local temperature and density variations in supersonic flows, as it does not rely on the isothermal approximation of the BGK model. In fact, it overcomes some limitations of the Bhatnagar-Gross-Krook equation, such as the fixed Prandtl number and the ratio between the kinematic and bulk viscosities.
- Increased numerical stability: the MRT method tends to be more numerically stable than the BGK model, especially for supersonic flows and high-resolution discretization schemes. The higher stability is due to the way the MRT method allows the different modes to be handled separately, which gives better control over the numerical dissipation and accuracy of the solution.

This was the method implemented to deal with the collision operator, and basically consists of the following three elements:

- A regular D-dimensional lattice and a selection of discrete velocities connecting each lattice site to some of its neighbors i.e. a discrete phase space. The set of velocity distribution functions that are defined on each node of the lattice is the basic component of the theory.
- A collision matrix and the equilibrium distribution functions. The local conserved quantities are functions of the equilibrium distribution functions.
- The discrete time evolution equation.

This model, described in [26] has been expanded to the family of vectorial schemes [5] [6] which will be discussed in the next chapter 3.4. This model is easily adaptable to many sorts of schemes and exhibits surprising features when used in shock wave tests. The vectorial scheme model will be the one that has been implemented in the open-source software PyLBM [4], i.e. what will be modified to improve the handling of boundary conditions.

In conclusion, the MRT method is a better choice than the BGK model for discretizing the collision operator in the Lattice Boltzmann Method when dealing with supersonic flows, as it offers greater accuracy, a better approximation of thermodynamic properties, and greater numerical stability.

Chapter 3

Lattice Boltzmann Method

The Lattice Boltzmann Method's main goal is to establish kinetic models that are as simple as possible. This simplified model is built so that the macroscopic averaged characteristics obey the appropriate macroscopic equations, and incorporate the fundamental physics of microscopic phenomena. This approach is much different from traditional numerical techniques based on the discretization of mesoscopic kinetic equations and macroscopic models. The collective movement of several microscopic particles in a fluid produces its macroscopic dynamics, which are not susceptible to underline microscopic physics. This is why the simplified kinetic-type approaches are appropriate for simulating the macroscopic fluid flows. To follow each fluid particle as in molecular dynamics simulations, the entire Boltzmann equations need extremely powerful computing power. One can avoid solving challenging kinetic equations like the full Boltzmann equation by creating a simplified version of the kinetic equation.

To sum up, the Lattice Boltzmann Method it's a numerical scheme that is mimicking somehow the Boltzmann Equation (kinetic equation), but you are not solving directly the Boltzmann Equation. In fact, the general Boltzmann equation is one level higher than even Navier-Stokes, as the rarefied effect is taken into account.

3.1 Literature Review for LBM

The LBM can be interpreted as a finite difference scheme for the discrete velocity distribution function's kinetic equation, in fact, it can be seen as a numerical method that uses pseudo-particles made to move over a cartesian mesh. When these arrive on a spatial cell, these fictitious particles—which are actually real numbers—interact with one another locally.

The first time the concept of simulating fluid flows using the simplified kinetic equation, in that previous case with just a single particle speed, was in 1964 for the study of shock structures [34]. The idea of adopting a multispeed discrete particle velocities model to examine shock-wave structures was in 1990 [35].

As mentioned in the introductory chapter, however, the ancestor of today's LBM is the LGA, in which, like each of these first models, space and time were continuous, despite the particle velocity being discretized in the distribution function. Hardy & al. proposed the full discrete particle velocity model in 1976 [36] to examine the transport characteristics of fluids. In this model, space and time are discretized on a square lattice. Frisch & al. identified the significance of the lattice's symmetry for the recovery of the Navier Stokes equation in 1986 [12]. For the first time, they were able to start with the Lattice Gas Automata on a hexagonal lattice and obtain the right Navier-Stokes equation. The cellular automaton model [37] [38] and the 3D model employing the four-dimensional Face Centered Hyper Cubic (FCHC) lattice [39] are the main ideas in the development of LBM that are contemporaneous with [12].

The primary characteristic of the LBM is that it ignores individual particle motion and particle-particle correlations in the kinetic equations [13], replacing the particle occupation variables ¹ with a single particle distribution function, which is the real variable. By using this method, the LBM's statistical noise is removed. The LBM has the advantage of maintaining the locality in the kinetic approach, which is necessary for parallelism since its kinetic form is still the same as the LGA form.

In the article [40] of 1989, an important simplification of the LBM is proposed, in fact, Higuera & Jimenez linearized the collision operator by supposing that the distribution is fairly close to the local equilibrium state. The same author also suggested a linearly stable collision operator technique [41]. The relaxation period towards the local equilibrium is used in a specific simple linearized form of the collision operator using a single-time relaxation. The already discussed Bhatnagar-Gross-Krook (BGK) collision operator [42], which was independently proposed by other authors [43] [44], is the name of the most common relaxation term. The local equilibrium distribution is used in this Lattice BGK (LBGK) model to recover the Navier-Stokes macroscopic equations [45] [46]. The computations are more effective and the transport coefficient can be changed when the lattice BGK model is used. As already discussed in section 2.3.3, the MRT model will be used in this thesis to deal with the collision operator, due to the range of advantages it offers, especially in the range with a Mach number greater than one.

¹The LGA uses boolean variables to specify the particle positions.

3.2 LBM for compressible applications

At SOA, when dealing with flow simulations with high Reynolds numbers and/or moderate to high Mach numbers, LBM faces stability problems. In order to extend LBM to the compressible (and eventually thermal) case there are two main approaches that can be taken:

- The first involves preserving the common "Collide & Stream" algorithm while expanding the number of free parameters and changing the velocity discretization and discrete equilibrium state.
- The second solution focuses on comparing several numerical approaches for the LBE's numerical discretization, therefore this approach is based only on a numerical understanding of stability constraints, assuming that the method used to obtain the LBE has not much impact on stability problems.

It results in rationally believing that the best approach that can be implemented is a middle way that combines both solutions. However, the problems faced with LBM in the compressible range are not only numerical in origin, the set values for the heat capacity ratio γ and the Prandtl number ($Pr = 1$)² are two additional non-numerical restrictions that belong to ordinary LBM.

- According to the first restriction, the thermal diffusivity coefficient cannot be freely set while the dynamic viscosity is fixed, and vice versa.
- The second limitation is closely related to the number of internal degrees of freedom of gas molecules. Indeed from the γ relationship, if only translational DOFs are considered, as in the case of conventional LBM, this limits the field of use of the method, applicable accurately to monoatomic gases only. Rotational DOFs must be included at least when defining the internal energy or the total energy in order to reproduce realistic flows, such as air.

It should be emphasized that these restrictions are more related to the future of the pylbm software [4], as only the Euler equation can be simulated at the moment, so the literature review on compressible and thermal LBM is done for illustrative purposes only. Although not strictly related to our test case, based on what was introduced, three types of LBMs—multispeed (or high-order), double distribution function, and hybrid LBMs exist in the literature for the simulation of thermal and

²Where γ is defined by the relationship $\gamma = 1 + \frac{2}{DOF}$, where DOF are the number of gas molecule internal degrees of freedom (i.e. DOF), while as for the Prandtl number, it is the ratio between dynamic and thermal diffusivity coefficients (i.e. $Pr = \frac{\mu C_p}{k}$), where μ is the dynamic viscosity, C_p is the specific heat at constant pressure, while k is the thermal conductivity

completely compressible flows with changing Prandtl number and heat capacity ratio.

The goal of the ESA TRP research project LBMHYPE is to figure out whether Lattice Boltzmann Methods are practical for applications involving hypersonic flows. The LBMHYPE work proposes carrying out a more in-depth analysis of the potential and limits of pure LBM schemes to model highly compressible flow behaviors. The feasibility study itself — can we utilize LBM to simulate high compressible flows? — was the focus of the first phase of the project. A review of the literature was conducted in the first part of the project in order to find suitable LBM models that could simulate highly compressible flows. The following section, therefore, aims to show some proposals from the literature for dealing with the supersonic regime also with the LBM, in particular, the approach of Professor Benjamin Graille and Professor François Dubois, given in the TN1.1 - Feasibility Assessment report of the LBMHYPE project was followed.

3.2.1 Finite Difference Lattice Boltzmann Method

The Finite Difference Lattice Boltzmann Method (FDLBM) involves discretizing a Boltzmann equation. A linear collision operator is used to simplify the Boltzmann equation in time, space, and velocities. Lastly, the equilibrium distribution function values are fixed in order to conserve the mass, momentum, and total energy of the particles.

The streaming operator is discretized using a finite differences scheme (such as the upwind or 3d-order upwind scheme), which is how it differs from the classical LBM. The kinetic velocities are fixed as multiples of λ in the LBM streaming operator, which makes it exact. This approach allows the selection of circular or spherical stencils, but it also increases errors during the streaming phase and makes the process tougher.

Single distribution function

An initial example of the application of LBM in the supersonic field was proposed in [47], [48]. However, a problem with numerical instability prevents FDLBM use in real systems with a Mach number greater than 1.

A research group created an innovative numerical approach known as the *Discrete Boltzmann Method* (DBM). The method used by the researchers to recover the complete Navier-Stokes equations is to directly approximate the Boltzmann equation using a simpler collisional operator. In [49], as done in [47], the stability of the Boltzmann equation's spatial discretization is examined in order to suggest a discrete velocity Boltzmann approach. In order to increase the von Neumann stability, this scheme includes a finite differences scheme with an additional dissipation

factor. What is crucial, however, as the same idea was then used in developing the *vectorial schemes* that were later implemented in `pylbn` [4], is that the same group of researchers also proposed an MRT variant of the FDLBM using the conventional MRT-LBM [50]. This concept was further developed in [51] and [52], which offers a discussion of the approach and applications that emphasize compressible flows. For high-speed compressible flows ranging from continuum to transition, always the same research group offer a generic framework in [53], [54] to develop trans-scale discrete Boltzmann models (DBMs). Two shock tubes were successfully simulated, demonstrating the model's robustness, accuracy, and applicability to compressible flows with strong shock wave interaction.

Double distribution function

The multispeed method is a direct increase in conventional second-order isothermal and weakly compressible LBMs. Adding discrete velocities to the LBE and accounting for high-order velocity terms for defining the discrete equilibrium state are the two key elements of this method. Two separate distribution functions are used in the double distribution function (DDF) technique, one for the temperature field and the other for the flow field, so through the evolution of a second distribution function for the energy, the DDF approach seeks to introduce the idea of temperature fluctuations. This particular type of LBM can thus separate the aerodynamic and thermodynamic relaxation processes, allowing for the independent selection of the dynamic and thermal diffusivity coefficients.

In [55], the first DDF model was developed. Since its development, this model has received a lot of attention due to its exceptional numerical stability and Pr number independence. This model nevertheless has delicate gradient factors that involve the temporal and spatial derivatives of the macroscopic flow variables, which could cause extra mistakes and degrade the numerical stability. In order to partially solve this problem, a total energy distribution function was added in [56] to replace the internal energy distribution function in their alternative version. The flow field is unaffected by the temperature field in this decoupling model.

For the 2D compressible Navier-Stokes equations, the research group in [57] suggested a coupled finite difference (FD) DDF model with an adaptable specific-heat ratio and Pr number. In [58] is adopted a D_3Q_{25} DVBE (Double Distribution Velocity Boltzmann Equation) model to solve the Sod shock tube and a regular shock reflection.

3.2.2 Finite Volume Methods & Lattice Boltzmann

During the literature analysis carried out, the following approaches were found, which are two finite volumes techniques that include kinetic aspects: the first approximation Riemann solver uses a one-dimensional lattice Boltzmann solver, while the second discretizes the velocity space.

Lattice Boltzmann Flux Solver

This approach that will be discussed in this section, as well as citing some interesting results in the supersonic field with the LBM, will offer insight into the implementation of boundary conditions in section 5.1, the focus of this thesis.

The lattice Boltzmann method has been developed for compressible applications by an Asian research team. They focused on new lattice Boltzmann schemes in [59], which are based on a new equilibrium distribution function that imitates the Maxwellian equilibrium with internal degrees of freedom. The Boltzmann equation's discretization, specifically the equilibrium distribution function, is the main foundation of the methodology. The proposed test case is a simulation of the Sod tube. The Asian research team then took another way adopting an LBM-based flux solver (finite volumes approach), as the method is founded on the creation of a circular equilibrium distribution function. Article [60] summarizes all of their work in this field. By simulating inviscid compressible flows with test problems in 1D and 3D, they compare the results of three LBM-based flux solvers. These proposals accomplish effectively, particularly for the Sod tube, the Shu-Osher shock tube, supersonic flow via a channel with a ramp segment, and the forward-facing step. This team's most recent contribution in this field is [61]. In this study, a stable Lattice Boltzmann Flux Solver (LBFS) has been proposed for the simulation of complicated hypersonic fluxes. In LBFS, the Navier-Stokes equations are solved using the finite volume method. The viscous flux is solved using the central difference approach, while the inviscid flux across the cell interface is reconstructed using the one-dimensional Lattice Boltzmann (D_1Q_4) model. Li et al. provide the WENO-LBFS method's results for compressible flows in [62], in which they evaluate various approaches and suggest a hybrid one that can efficiently capture shock waves without oscillations and calculate the solution of smooth regions with accuracy.

Unified Gas-Kinetic Scheme

This approach was presented in 2001 [63], and it uses discrete particle simulations when the flow is rarefied and it is based on a finite volume discretization. It appears to be a significant alternative to the Direct Simulation Monte Carlo (DSMC) approach, especially when a transitional regime is located close to another section

of the flow where the thermodynamical equilibrium is established. Using a unified gas kinetic scheme, in [64] is shown a study on the Knudsen number around a reentry vehicle with wings ($M = 4$, $Re = 6 \cdot 10^4$). Using conventional high-order interpolation (Weighted Essentially Non Oscillatory methods), this method can be expanded to higher precision. The authors of [65] present a wide range of test cases, in which this method is tested. More theoretical research and numerical analysis are the main topics of other contributions [66] [67] [68].

3.2.3 Implicit Lattice Boltzmann Methods

The *entropic lattice Boltzmann method* is an implicit Lattice Boltzmann technique that involves the minimization of entropy. The contribution [69] presents the ELBM, which can be applied also to compressible flows.

The particles-on-demand method instead, could be a solution to overcome all of the lattice Boltzmann method's physical restrictions.

Entropic lattice Boltzmann method

The authors offer a lattice Boltzmann model in [70] that, in their perspective, covers the full range of fluid flows, from low Mach weakly compressible flows to transonic and supersonic flows. Three key improvements to the LBM scheme are made to get over the low Mach number limit:

- A multi-speed lattice accurately developed
- A precise evaluation of the equilibrium
- An entropic relaxation for the collision

The basis for their work is that the Navier-Stokes equations can be recovered if the Maxwell-Boltzmann distribution function is approximated accurately. The main rule is to select a lattice whose discrete equilibria approach as close as it is possible to the moments of the Maxwell-Boltzmann distribution function.

Particles on demand

The lattice Boltzmann schemes' basic structure is quite similar to that of the particle methods. Based on this notion and the connection to relaxation schemes, different research teams start to build implicit schemes. In [71], a research team proposed an implicit approach to a scalar, one-dimensional hyperbolic problem using D_1Q_2 and D_2Q_4 as foundations. A Riemann problem for a barotropic gas is resolved in the reference [71]. This group's aim consists of handling the magnetohydrodynamics equations. Using the same collision operator as the Entropic Lattice Boltzmann

approach, a lattice Boltzmann strategy is provided in [72]. In order to satisfy the set of moments, a distribution of particles is generated from the specified moments. But the transport velocities are of type $c_j = v + \lambda_j$ ³ rather than using the lattice's fixed velocities as in conventional LBM. The D_2Q_9 scheme is used to offer an amazing test case for the thermal gas dynamics equations in [72] at very high Mach numbers.

In [72], authors pretend that only fluid velocities near zero and temperatures close to a reference temperature can be employed with the conventional LBM. On the other hand, the *particles on demand* method's possible flow velocities and temperatures are limited only by the lattice speed selection.

The algorithms presented in [71], [72] are of significant importance despite their implicit nature, for their innovative way to increase stability. On the other hand, we must consider the higher computational cost.

3.2.4 Lattice Boltzmann & VLES

In [73] it is proposed a Very Large Eddy Simulation (VLES) method based on the LBM to simulate high Reynolds number turbulent flows, a scheme used to study the interaction between a reflected oblique shock and a supersonic turbulent boundary layer.

As with the DDF-LBM, the hybrid LBM is based on a two-equation approach, but the energy equation is in this case resolved using conventional numerical techniques (finite volume, finite difference, etc.). A hybrid scheme was proposed in the publication [74] to simulate the flows past a wedge and flows past an airfoil, which was an earlier work using the same idea as [73].

³Where \mathbf{v} is the velocity of the flow, while λ_j is the lattice velocity

3.3 The lattice Boltzmann approach

As already seen in the previous chapter 2, the equation that is solved with LBM is the Boltzmann Equation 2.29, which can be rewritten as a function of Knudsen number Kn :

$$\frac{\partial f(t, x, c)}{\partial t} + c \frac{\partial f(t, x, c)}{\partial x} = \frac{1}{Kn} \Omega(f) \quad (3.1)$$

where \mathbf{t} , \mathbf{x} , and \mathbf{c} are the time, space, and velocity variables. In this partial differential equation 3.1, the unknown quantity is the function \mathbf{f} of the three variables $(\mathbf{t}, \mathbf{x}, \mathbf{c})$.

The LBM can be thought of as a numerical method involving fictitious particles (i.e. the particle distribution function \mathbf{f}) forced to move on a cartesian lattice. These pseudo-particles are scalars (i.e. numbers) and they interact locally with each other during the collision phase, also known as the relaxation phase when arriving on a lattice node.

These are the elements that compose a lattice Boltzmann scheme:

- a **cartesian spatial mesh** $L = \Delta x \in \mathbb{Z}^d$, where d is the dimension of the space, while Δx is the uniform spatial step.
- a **time step** Δt , which can be linked to the uniform spatial step Δx through the parameter $\lambda = \frac{\Delta x}{\Delta t}$ which is the velocity of the lattice.
- a **stencil of velocities** $\{c_0, \dots, c_{q-1}\} \in \lambda \mathbb{Z}^d$, where q is the length of this stencil (i.e the number of velocities used in the method).
We can denote f_j the particle distribution function of the particles that move with the velocity c_j where $0 \leq j < q$.
- an **invertible matrix** \mathbf{M} of size $q \times q$ that defines the moments.
- the **equilibrium** value of the particle distribution functions.
- the **relaxation parameters**.

The lattice Boltzmann scheme makes the quantities $f_j(x, t)$ ⁴ change in accordance with some straightforward criteria. One time step of the scheme splits into four sub-steps, as we can see in the figure 3.1

⁴where $0 \leq j < q$, $x \in L$ the cartesian spatial mesh and the time step $t \in \Delta \mathbb{N}$

- **f2m**, based on the particle distribution functions $f_j(x, t)$, calculate the moments m_k $0 \leq k < q$

$$m_k(x, t) = \sum_{j=0}^{q-1} M_{kj} f_j(x, t) \quad 0 \leq k < q \quad (3.2)$$

- **R** relaxes the moments toward the equilibrium

$$m_k^*(x, t) = (1 - s_k) m_k(x, t) + s_k m_k^{eq}(x, t) \quad 0 \leq k < q \quad (3.3)$$

where s_k is the relaxation parameter associated to the k_{th} moment.

- **m2f**, from the moments, calculate the particle distribution functions

$$f_j^*(x, t) = \sum_{k=0}^{q-1} M_{jk}^{-1} m_k^*(x, t) \quad 0 \leq j < q \quad (3.4)$$

- **T** transport the particle distribution functions according to their velocities

$$f_j(x, t + \Delta t) = f_j^*(x - c_j \Delta t, t) \quad 0 \leq j < q \quad (3.5)$$

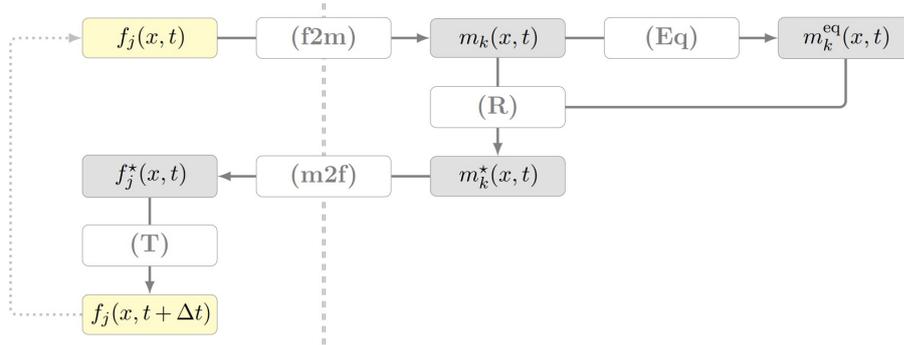


Figure 3.1: Sketch of a lattice Boltzmann one time step

There are two categories of moments m_k , $0 \leq k < q$.

- The first n moments $0 \leq n < q - 1$ are “conserved” during the relaxation phase:

$$m_k^{eq} = m_k \quad 0 \leq k < n \quad (3.6)$$

- While for the remaining moments m_k^{eq} $n \leq k < q$, their equilibrium value is a function of the conserved moments, which is known *a priori*.

$$m_k^{eq} = m_k^{eq}(m_0, \dots, m_{n-1}) \quad n \leq k < q \quad (3.7)$$

Therefore, the Boltzmann equation 3.1 can be discretized using LBM by separating the transport and collision operators. The collision operator is approximated by an explicit Euler scheme, while the transport phase is exact according to the theory of the characteristics.

The following list of advantages of this algorithm can be summed up:

- The transport step is exact, in fact, all that needs to be done during this phase is to move the data from one point to another.
- All the other steps are local in space, which suggests that parallelization can be quite successful.
- The algorithm is fully explicit, so there is no need to save big matrices or handle non-linear problems.

The disadvantages are as follows, in summary:

- Finding a lattice Boltzmann scheme that simulates a certain partial differential equation is difficult. It is possible to determine which equation is simulated for a given scheme, but the opposite is challenging.
- There is no established and unified framework for the numerical analysis of such a system, and concepts like consistency, stability, and convergence are not well-defined.
- The numerical stability is not completely understood; even if each step is stable on its own, the overall system may develop spurious instabilities.
- It can be challenging to offer precise boundary conditions since they depend on the incoming particle distribution functions rather than the moments.

The macroscopic conservative equations on the invariant moments of the collision operator Ω result from an asymptotic evolution of the solution, according to the tiny parameter Kn . These conserved moments are the kinetic velocity-dependent moments that remain constant throughout a collision. The important aspect of this theory is that the derived macroscopic equations depend only on these conserved moments and not on the precise expression of the collision operator. Moments of the microscopic particle distribution function can be used to determine the macroscopic fluid density, momentum, and internal energy. In fact, from the 2.30 relations, these can be generalized depending on the discretization that was set out

to be performed (i.e., the number of velocities chosen in the method)

$$\begin{aligned}
 \boldsymbol{\rho}(x, t) &= \sum_{k=0}^{q-1} f_k(x, t) \\
 \mathbf{Q}(x, t) &= \sum_{k=0}^{q-1} \lambda_k \mathbf{f}_k(x, t) \\
 \mathbf{E}(x, t) &= \sum_{k=0}^{q-1} (\lambda_k - \mathbf{u}(x, t))^2 \mathbf{f}_k(x, t) / \boldsymbol{\rho}(x, t)
 \end{aligned} \tag{3.8}$$

Given these properties that are provided by the numerical method, any other properties of the fluid can be easily derived. For example, given momentum (which can be decomposed along Cartesian axes), it is possible to derive velocities.

$$\begin{aligned}
 u_x &= \frac{Q_x}{\rho} \\
 u_y &= \frac{Q_y}{\rho} \\
 |V| &= \sqrt{u_x^2 + u_y^2}
 \end{aligned} \tag{3.9}$$

Or, from the relations of gas-dynamics, the pressure can also be derived:

$$p = (\gamma - 1) \left[E - \frac{1}{2} \frac{Q_x^2 + Q_y^2}{\rho} \right] \tag{3.10}$$

As well as the speed of sound:

$$c = \sqrt{\gamma(\gamma - 1) \frac{E - \frac{1}{2}\rho(u_x^2 + u_y^2)}{\rho}} \tag{3.11}$$

Then the Mach number can be obtained:

$$\text{Mach} = \frac{|V|}{c} \tag{3.12}$$

Then the total pressure can also be derived, remembering that we are assuming a compressible situation:

$$P_{TOT} = p \left(1 + \frac{\gamma - 1}{2} \text{Mach}^2 \right)^{\frac{\gamma}{\gamma - 1}} \tag{3.13}$$

Finally, although we will then go on to investigate an isothermal test case 6.2 and 7.2, the temperature can also be derived through the perfect gas equation:

$$T = \frac{p}{\rho R} \tag{3.14}$$

3.4 Vectorial Schemes

According to the analysis carried out by Professor Benjamin Graille and Professor François Dubois, reported in section 3.2 of this chapter on the feasibility of treating the supersonic regime through the LBM. The following elements can be used to create a Lattice Boltzmann scheme, in order to simulate compressible flows:

- A promising approach is to use vectorial schemes (one distribution function for each scalar conserved moment or a double distribution function)
- Re-centering the velocities (using relative velocity schemes or changing the kinetic velocities) might increase stability.
- The intensity of internal energy might be considered through an adjustment of the moment's matrix

As already discussed in section 2.3.3, the MRT model will be used in this thesis to deal with the collision operator, MRT concept which was furthermore extended to the family of vectorial schemes [5] [6].

Focusing more on Professor Graille's article [6], introduces a new lattice Boltzmann scheme named $D_1Q_2^n$, which is intended to mimic one-dimensional hyperbolic conservative fluid dynamics equations, particularly those written in terms of conservation law. These include, in particular, the equations derived from the kinetic theory of gases (i.e. Euler's equations) [75]. The [6] methodology treats each of the system's equations independently while minimizing the necessity of the Boltzmann equation. Usually, in order to increase the dimension of the system (i.e. the number of conservation equations) conserved moments of the one-dimensional hyperbolic system with higher velocities are introduced, with two consequences:

- The stencil of the velocities is extended, as a result, it will be more challenging to apply the boundary conditions;
- Adding new velocities modifies the scheme, necessitating a re-evaluation of all prior studies of stability and accuracy.

The well-known and simplest lattice Boltzmann system, the D_1Q_2 (one spatial dimension and two discrete velocities), is replicated for each of the n conserved moments in the proposed scheme [6], represented as $D_1Q_2^n$. This scheme is related to a specific discretization of the relaxation method that separates the linear hyperbolic part (which is dealt with using a Lax-Friedrichs discretization [76]) from the relaxation part (which is dealt with using an explicit Euler solver).

As a consequence, it is simple to apply what has been implemented from the study of the scalar equation to the system of n -equations. In addition, since the boundary conditions are given in terms of particle distribution functions within the context of lattice Boltzmann schemes, the decoupling of these functions greatly simplifies the selection of the incoming particle distribution functions on the boundaries in order to satisfy the boundary conditions.

Given stability conditions, the scheme is defined within the framework of MRT approach, which leads to the benefits already discussed. The relaxation method presented by Jin and Xin [77] was analyzed in the paper by Professor Graille [6] in order to connect the scheme with an explicit finite differences approximation. A stiff source term and the relaxation approach were developed by [77] to convert a non-linear hyperbolic system of dimension n into a linear hyperbolic system of dimension $2n$.

3.4.1 Lattice Boltzmann schemes for compressible flows

In this section, we are going to analyze several schemes to simulate Euler equations that make up a classical hyperbolic system. The following *vectorial schemes* will be discussed to show numeric artifacts like oscillations and diffusion. It is important to point out that within the LBMHYPE project, as in [6], the proposed schemes are written into d’Humières’ framework [78]; The Taylor expansion approach is used to provide the corresponding equations up to the second order [79] [80]. As summed up in [81], the lattice Boltzmann scheme’s accuracy⁵ can be increased numerically by using the Taylor expansion method. According to this method, it is possible to determine the equivalent partial differential equation of the LB scheme up to the second order of accuracy using the only parameter Δt as infinitesimal variable⁶. The Taylor expansion approach allows the development of an explicit formula, in this way, it is easier to tune the parameters of the LB scheme and obtain a higher-order accuracy. Through a trade-off between stability and precision, the residual terms of second order are examined in accordance with the scheme’s free parameters. As we will see in section 3.4.3, often figuring out how to set parameters is not at all simple, and often the trivial solution is unstable.

Back to *vectorial schemes*, they can be obtained by combining several equations. In this way, it is possible to emulate a set of equations like Euler equations.

⁵As well as the accuracy of any numerical method for simulating partial differential equations associated with conservation laws of physics

⁶In addition to requiring that the lattice speed λ be constant

In 1D the system reads:

$$\begin{cases} \partial_t \rho + \partial_x(\rho u) = 0, \\ \partial_t(\rho u) + \partial_x(\rho u^2 + p) = 0, \\ \partial_t(E) + \partial_x((E + p)u) = 0, \end{cases} \quad (3.15)$$

where ρ is the density, \mathbf{u} the flow velocity, $\mathbf{q} = \rho \mathbf{u}$ the momentum, \mathbf{p} the pressure and \mathbf{e} the total energy. All these quantities are both time-space-dependent. The pressure \mathbf{p} and the total energy \mathbf{E} are linked by the relation

$$E = \frac{1}{2} \rho u^2 + \frac{p}{\gamma - 1} \quad (3.16)$$

where γ is the heat capacity ratio.

In 2D the system reads:

$$\begin{cases} \partial_t \rho + \partial_x(\rho u_x) + \partial_y(\rho u_y) = 0, \\ \partial_t(\rho u_x) + \partial_x(\rho u_x^2 + p) + \partial_y(\rho u_x u_y) = 0, \\ \partial_t(\rho u_y) + \partial_x(\rho u_x u_y) + \partial_y(\rho u_y^2 + p) = 0, \\ \partial_t(E) + \partial_x((E + p)u_x) + \partial_y((E + p)u_y) = 0, \end{cases} \quad (3.17)$$

where ρ is the density, $\mathbf{u} = (u_x, u_y)$ the flow velocity, $\mathbf{q} = \rho \mathbf{u}$ the momentum, \mathbf{p} the pressure and \mathbf{E} the total energy. All these quantities are both time-space-dependent. The pressure \mathbf{p} and the total energy \mathbf{E} are linked by the relation

$$E = \frac{1}{2} \rho (u_x^2 + u_y^2) + \frac{p}{\gamma - 1} \quad (3.18)$$

where γ is the heat capacity ratio.

3.4.2 LBM vectorial scheme D_1Q_{222}

To begin dealing with vectorial schemes, let us start with the simplest one that is possible to implement, although a more advanced method (i.e., one that takes into account more speed) will be used in the test cases that will be analyzed in 6 and 7.

The simplest 1D scheme that can model the 1D Euler equations is the D_1Q_{222} vectorial scheme. Three coupled D_1Q_2 schemes are used, one for each of the one-dimensional Euler conservation equations. The velocity stencil is the smallest stencil in 1D and consists of only 2 velocities $(1, -1)$. The conserved moments are ρ , \mathbf{q} , and \mathbf{E} .

In order to build an invertible matrix $\mathbf{M}_{D_1Q_{222}}$, we must first define the vectors of the six moments $\mathbf{m} = (\rho, m_0, q, m_1, E, m_2)$ and the vectors of the six particle distribution functions $\mathbf{f} = (f_0, \dots, f_5)$.

$$M_{D_1 Q_{222}} = \begin{pmatrix} M_{D_1 Q_2} & 0 & 0 \\ 0 & M_{D_1 Q_2} & 0 \\ 0 & 0 & M_{D_1 Q_2} \end{pmatrix}, \quad M_{D_1 Q_2} = \begin{pmatrix} 1 & 1 \\ \lambda & -\lambda \end{pmatrix}, \quad (3.19)$$

where $\lambda = \Delta x / \Delta t$ is the lattice speed.

Moments $\mathbf{m} = (\rho, m_0, q, m_1, E, m_2)$, particle distribution function $\mathbf{f}_{0\dots 5}$, and the M-matrix $\mathbf{M}_{D_1 Q_{222}}$ are linked by the following relationship:

$$m = M_{D_1 Q_{222}} f \quad (3.20)$$

During the relaxation phase, the non-conserved moments are modified in accordance with

$$m_0^* = m_0 + s_\rho (m_0^{\text{eq}} - m_0), \quad (3.21)$$

$$m_1^* = m_1 + s_u (m_1^{\text{eq}} - m_1), \quad (3.22)$$

$$m_2^* = m_2 + s_p (m_2^{\text{eq}} - m_2), \quad (3.23)$$

where s_ρ, s_u, s_p are the three relaxation parameters which must be chosen between 0 and 2 for stability reasons.

The equilibrium value of the non-conserved moments employed during the relaxation phase must also be defined. As mentioned earlier, what was introduced in [6] will be used:

$$m_0^{\text{eq}} = \rho u, \quad m_1^{\text{eq}} = \rho u^2 + p, \quad m_2^{\text{eq}} = (E + p)u. \quad (3.24)$$

Last but not least, defining the temperature \mathbf{T} by $\mathbf{T} = p / (\rho R) = c^2 / (\gamma R)$ with \mathbf{R} the specific gas constant, it is possible to obtain the corresponding equations up to the second order using the expansion of Dubois [79] [80]:

$$\begin{cases} \partial_t \rho + \partial_x(\rho u) = \partial_x(B_{\rho,\rho} \partial_x \rho + B_{\rho,u} \partial_x u + B_{\rho,T} \partial_x T) + \mathcal{O}(\Delta t^2), \\ \partial_t(\rho u) + \partial_x(\rho u^2 + p) = \partial_x(B_{u,\rho} \partial_x \rho + B_{u,u} \partial_x u + B_{u,T} \partial_x T) + \mathcal{O}(\Delta t^2), \\ \partial_t E + \partial_x((E + p)u) = \partial_x(B_{T,\rho} \partial_x \rho + B_{T,u} \partial_x u + B_{T,T} \partial_x T) + \mathcal{O}(\Delta t^2), \end{cases} \quad (3.25)$$

where the coefficients $\mathbf{B}_{\alpha,\beta}$, $\alpha, \beta \in \{\rho, u, T\}$, are the following:

$$\begin{aligned}
 B_{\rho,\rho} &= \left(\lambda^2 - u^2 - \frac{c^2}{\gamma} \right) \sigma_\rho \Delta t, \\
 B_{\rho,u} &= -2\rho u \sigma_\rho \Delta t, \\
 B_{\rho,T} &= -\rho R \sigma_\rho \Delta t, \\
 B_{u,\rho} &= \left(\lambda^2 - u^2 - \frac{c^2}{\gamma} \right) u \sigma_u \Delta t, \\
 B_{u,u} &= \left(\lambda^2 - 3u^2 - c^2 \right) \rho \sigma_u \Delta t, \\
 B_{u,T} &= -3\rho u R \sigma_u \Delta t, \\
 B_{T,\rho} &= \left(\frac{u^2}{2} (\lambda^2 - u^2) + \frac{c^2}{\gamma(\gamma-1)} \left(\lambda^2 - \frac{1}{2} (5\gamma-3) u^2 - c^2 \right) \right) \sigma_p \Delta t, \\
 B_{T,u} &= \left(\lambda^2 - 2u^2 - \frac{\gamma+1}{\gamma-1} c^2 \right) \rho u \sigma_p \Delta t, \\
 B_{T,T} &= \frac{\rho R}{\gamma-1} \left(\lambda^2 - \frac{1}{2} (5\gamma-3) u^2 - c^2 \right) \sigma_p \Delta t,
 \end{aligned} \tag{3.26}$$

with the Henon parameters σ_ρ , σ_u , and σ_p defined by

$$\sigma_\rho = \frac{1}{s_\rho} - \frac{1}{2}, \quad \sigma_u = \frac{1}{s_u} - \frac{1}{2}, \quad \sigma_p = \frac{1}{s_p} - \frac{1}{2}. \tag{3.27}$$

What was presented by using this scheme in the LBMHYPE project is that this D_1Q_{222} is consistent with the Euler equations up to the second order only for $s_\rho = s_u = s_p = 2$, but this choice of configurations is frequently unstable. The complexity of the numerical diffusion depends on the hydrodynamic parameters.

From these complex relationships obtained for the simplest existing vectorial scheme, we can already observe the non-linear interaction between the main parameters of the LBM, namely the diffusivity σ of the scheme (related to the relaxation parameters via 3.27) and the lattice velocity λ . For example, it can be seen that λ , which has to satisfy a CFL-type condition to ensure the stability of the scheme⁷ is also involved in the numerical diffusion (i.e. the property under σ) and vice versa.

3.4.3 LBM vectorial scheme D_2Q_{4444}

By increasing the size of the system, it is necessary to take into consideration more speed in the stencil, so let's start to see the method that will be used to solve the wedge first 6 and the cylinder then 7.

⁷It should not be forgotten that we are using an explicit scheme.

The simplest 2D vectorial system that can simulate the 2D Euler equations is D_2Q_{4444} , in which four coupled D_2Q_4 schemes are used, one for each of the two-dimensional (2D) Euler conservation equations. The velocity stencil is the smallest stencil in 2D and only contains 4 velocities ($((1,0), (0,1), (-1,0), (0,-1))$). The conserved moments are ρ , q_x , q_y , and E .

In this case, since the complexity and size of the system to be solved are increasing, the vector of moments consists of sixteen elements:

$$m = (\rho, m_{\rho,x}, m_{\rho,y}, m_{\rho,2}, q_x, m_{q_x,x}, m_{q_x,y}, m_{q_x,2}, \\ q_y, m_{q_y,x}, m_{q_y,y}, m_{q_y,2}, E, m_{E,x}, m_{E,y}, m_{E,2})$$

As well, the vector of particle distribution functions is composed of sixteen elements $f = (f_0, \dots, f_{15})$.

In analogy with the vectorial scheme seen previously 3.4.2, there is a relationship between moments and particle distribution functions through the invertible matrix $M_{D_2Q_{4444}}$:

$$m = M_{D_2Q_{4444}} f \quad (3.28)$$

where the matrix $M_{D_2Q_{4444}}$ is defined as follows:

$$M_{D_2Q_{4444}} = \begin{pmatrix} M_{D_2Q_4} & 0 & 0 & 0 \\ 0 & M_{D_2Q_4} & 0 & 0 \\ 0 & 0 & M_{D_2Q_4} & 0 \\ 0 & 0 & 0 & M_{D_2Q_4} \end{pmatrix}, \quad (3.29)$$

where the $M_{D_2Q_4}$ sub-matrix is defined as follows, as a function of the lattice speed $\lambda = \Delta x / \Delta t$:

$$M_{D_2Q_4} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ \lambda & 0 & -\lambda & 0 \\ 0 & \lambda & 0 & -\lambda \\ \lambda^2 & -\lambda^2 & \lambda^2 & -\lambda^2 \end{pmatrix}, \quad (3.30)$$

During the relaxation phase, the non-conserved moments are modified according to:

$$m_{\rho,\xi}^* = m_{\rho,\xi} + s_\rho(m_{\rho,\xi}^{\text{eq}} - m_{\rho,\xi}), \quad m_{\rho,2}^* = m_{\rho,2} + s_{\rho,2}(m_{\rho,2}^{\text{eq}} - m_{\rho,2}), \quad (3.31)$$

$$m_{q_\zeta,\xi}^* = m_{q_\zeta,\xi} + s_u(m_{q_\zeta,\xi}^{\text{eq}} - m_{q_\zeta,\xi}), \quad m_{q_\zeta,2}^* = m_{q_\zeta,2} + s_{u,2}(m_{q_\zeta,2}^{\text{eq}} - m_{q_\zeta,2}), \quad (3.32)$$

$$m_{E,\xi}^* = m_{E,\xi} + s_p(m_{E,\xi}^{\text{eq}} - m_{E,\xi}), \quad m_{E,2}^* = m_{E,2} + s_{p,2}(m_{E,2}^{\text{eq}} - m_{E,2}), \quad (3.33)$$

for $\zeta, \xi \in \{x, y\}$, where s_ρ , s_u , s_p are the three relaxation parameters which must be chosen between 0 and 2 for stability reasons.

Exactly as in the scheme presented before, it is then necessary to define the equilibrium value of non-conserved moments adopted during the relaxation phase:

$$\begin{aligned}
 m_{\rho,\xi}^{\text{eq}} &= \rho u_\xi, & m_{\rho,2}^{\text{eq}} &= \rho(u_x^2 - u_y^2), \\
 m_{q_\zeta,\xi}^{\text{eq}} &= \rho u_\zeta u_\xi + p \delta_{\zeta,\xi}, & m_{q_\zeta,2}^{\text{eq}} &= 0, \\
 m_{E,\xi}^{\text{eq}} &= (E + p)u_\xi, & m_{E,2}^{\text{eq}} &= 0,
 \end{aligned} \tag{3.34}$$

where $\delta_{\zeta,\xi} = 1$ if $\zeta = \xi$ and $\delta_{\zeta,\xi} = 0$ if $\zeta \neq \xi$.

Last but not least, defining the temperature \mathbf{T} by $\mathbf{T} = p/(\rho R) = c^2/(\gamma R)$ with \mathbf{R} the specific gas constant, it is possible to obtain the corresponding equations up to the second order using the expansion of Dubois [79] [80]:

$$\left\{ \begin{aligned}
 \partial_t \rho + \partial_x(\rho u_x) + \partial_y(\rho u_y) &= \sigma_\rho \Delta t \left(\partial_x \theta_x^\rho + \partial_y \theta_y^\rho \right) + \mathcal{O}(\Delta t^2), \\
 \partial_t(\rho u_x) + \partial_x(\rho u_x^2 + p) + \partial_y(\rho u_x u_y) &= \sigma_u \Delta t \left(\partial_x \theta_x^{u_x} + \partial_y \theta_y^{u_x} \right) + \mathcal{O}(\Delta t^2), \\
 \partial_t(\rho u_y) + \partial_y(\rho u_y^2 + p) + \partial_x(\rho u_y u_x) &= \sigma_u \Delta t \left(\partial_x \theta_x^{u_y} + \partial_y \theta_y^{u_y} \right) + \mathcal{O}(\Delta t^2), \\
 \partial_t E + \partial_x((E + p)u_x) + \partial_y((E + p)u_y) &= \sigma_E \Delta t \left(\partial_x \theta_x^E + \partial_y \theta_y^E \right) + \mathcal{O}(\Delta t^2),
 \end{aligned} \right. \tag{3.35}$$

where the various parameters present are given by the following relationships:

$$\begin{aligned}
 \theta_x^\rho &= \left(\frac{1}{2} \lambda^2 - \frac{1}{\gamma} c^2 - \frac{1}{2} (u_x^2 + u_y^2) \right) \partial_x \rho - u_x u_u \partial_y \rho - \rho R \partial_x T \\
 &\quad - \rho u_x \partial_x u_x - \rho u_y \partial_y u_x - \rho u_x \partial_y u_y - \rho u_y \partial_x u_y, \tag{3.36}
 \end{aligned}$$

$$\begin{aligned}
 \theta_y^\rho &= \left(\frac{1}{2} \lambda^2 - \frac{1}{\gamma} c^2 - \frac{1}{2} (u_x^2 + u_y^2) \right) \partial_y \rho - u_x u_u \partial_x \rho - \rho R \partial_y T \\
 &\quad - \rho u_y \partial_y u_y - \rho u_x \partial_x u_y - \rho u_y \partial_x u_x - \rho u_x \partial_y u_x, \tag{3.37}
 \end{aligned}$$

$$\begin{aligned}
 \theta_x^{u_x} &= \left(\frac{1}{2} \lambda^2 - \frac{2\gamma-1}{\gamma(\gamma-1)} c^2 - u_x^2 \right) u_x \partial_x \rho - \left(u_x^2 + \frac{1}{\gamma(\gamma-1)} c^2 \right) u_y \partial_y \rho - \frac{2\gamma-1}{\gamma(\gamma-1)} \rho u_x R \partial_x T \\
 &\quad - \frac{1}{\gamma(\gamma-1)} \rho u_y R \partial_y T + \rho \left(\frac{1}{2} \lambda^2 - 3u_x^2 \right) \partial_x u_x - 2\rho u_x u_y \partial_y u_x - \rho u_x^2 \partial_y u_y, \tag{3.38}
 \end{aligned}$$

$$\begin{aligned}
 \theta_y^{u_x} &= \left(\frac{1}{2} \lambda^2 - \frac{1}{\gamma} c^2 - u_y^2 \right) u_x \partial_y \rho - \left(u_x^2 + \frac{1}{\gamma} c^2 \right) u_y \partial_x \rho - \frac{1}{\gamma} \rho u_y R \partial_x T - \frac{1}{\gamma} \rho u_x R \partial_y T \\
 &\quad + \rho \left(\frac{1}{2} \lambda^2 - u_y^2 \right) \partial_y u_x - 2\rho u_x u_y \partial_x u_x - \rho u_x^2 \partial_x u_y - 2\rho u_x u_y \partial_y u_y, \tag{3.39}
 \end{aligned}$$

$$\begin{aligned}
 \theta_x^{u_y} &= \left(\frac{1}{2} \lambda^2 - \frac{1}{\gamma} c^2 - u_x^2 \right) u_y \partial_x \rho - \left(u_y^2 + \frac{1}{\gamma} c^2 \right) u_x \partial_y \rho - \frac{1}{\gamma} \rho u_x R \partial_y T - \frac{1}{\gamma} \rho u_y R \partial_x T \\
 &\quad + \rho \left(\frac{1}{2} \lambda^2 - u_x^2 \right) \partial_x u_y - 2\rho u_y u_x \partial_y u_y - \rho u_y^2 \partial_y u_x - 2\rho u_y u_x \partial_x u_x, \tag{3.40}
 \end{aligned}$$

$$\begin{aligned} \theta_y^{u_y} = & \left(\frac{1}{2} \lambda^2 - \frac{2\gamma-1}{\gamma(\gamma-1)} c^2 - u_y^2 \right) u_y \partial_y \rho - \left(u_y^2 + \frac{1}{\gamma(\gamma-1)} c^2 \right) u_x \partial_x \rho - \frac{2\gamma-1}{\gamma(\gamma-1)} \rho u_y R \partial_y T \\ & - \frac{1}{\gamma(\gamma-1)} \rho u_x R \partial_x T + \rho \left(\frac{1}{2} \lambda^2 - 3u_y^2 \right) \partial_y u_y - 2\rho u_y u_x \partial_x u_y - \rho u_y^2 \partial_x u_x, \end{aligned} \quad (3.41)$$

$$\begin{aligned} \theta_x^E = & \left(\frac{1}{2} u_x^4 + \frac{1}{2} u_x^2 u_y^2 + \frac{3\gamma^2-4\gamma+3}{2\gamma(\gamma-1)} u_x^2 c^2 + \frac{1}{2\gamma} u_y^2 c^2 + \frac{1}{\gamma(\gamma-1)} c^4 \right) \partial_x \rho \\ & + \left(\frac{1}{2} u_x^2 + \frac{1}{2} u_y^2 + \frac{\gamma^2-\gamma+1}{\gamma(\gamma-1)^2} c^2 \right) u_x u_y \partial_y \rho + \rho u_x (2u_x^2 + u_y^2 + \frac{\gamma}{(\gamma-1)^2} c^2) \partial_x u_x + \rho u_y u_x^2 \partial_x u_y \\ & + \rho \left(\frac{1}{2} u_x^3 + \frac{3}{2} u_x u_y^2 + \frac{3}{2} u_x^2 u_y + \frac{1}{2} u_y^3 + \frac{1}{\gamma-1} c^2 u_y + \frac{\gamma}{\gamma-1} c^2 u_x \right) \partial_y u_y \\ & + \rho R \left(\frac{3\gamma^2-4\gamma+3}{2(\gamma-1)^2} u_x^2 + \frac{1}{2} u_y^2 + \frac{1}{\gamma-1} c^2 \right) \partial_x T + \rho R \frac{\gamma^2-\gamma+1}{(\gamma-1)^2} u_x u_y \partial_y T, \end{aligned} \quad (3.42)$$

$$\begin{aligned} \theta_y^E = & \left(\frac{1}{2} u_y^4 + \frac{1}{2} u_y^2 u_x^2 + \frac{3\gamma^2-4\gamma+3}{2\gamma(\gamma-1)} u_y^2 c^2 + \frac{1}{2\gamma} u_x^2 c^2 + \frac{1}{\gamma(\gamma-1)} c^4 \right) \partial_y \rho \\ & + \left(\frac{1}{2} u_y^2 + \frac{1}{2} u_x^2 + \frac{\gamma^2-\gamma+1}{\gamma(\gamma-1)^2} c^2 \right) u_y u_x \partial_x \rho + \rho u_y (2u_y^2 + u_x^2 + \frac{\gamma}{(\gamma-1)^2} c^2) \partial_y u_y + \rho u_x u_y^2 \partial_y u_x \\ & + \rho \left(\frac{1}{2} u_y^3 + \frac{3}{2} u_y u_x^2 + \frac{3}{2} u_y^2 u_x + \frac{1}{2} u_x^3 + \frac{1}{\gamma-1} c^2 u_x + \frac{\gamma}{\gamma-1} c^2 u_y \right) \partial_x u_x \\ & + \rho R \left(\frac{3\gamma^2-4\gamma+3}{2(\gamma-1)^2} u_y^2 + \frac{1}{2} u_x^2 + \frac{1}{\gamma-1} c^2 \right) \partial_y T + \rho R \frac{\gamma^2-\gamma+1}{(\gamma-1)^2} u_y u_x \partial_x T. \end{aligned} \quad (3.43)$$

These complex formulas suggest that selecting the equilibrium value of the last moment in order to generate a null second-order operator is not practical. Even though the D_2Q_{4444} has more degrees of freedom in terms of equilibrium options, the conclusion is the same as it was for the D_1Q_{222} : the scheme is consistent with the Euler equation up to the second order only for $s_\rho = s_u = s_p = 2$, but this choice of parameters is unstable.

Chapter 4

Boundary conditions for the LBM

Unlike conventional CFD, it is challenging to implement boundary conditions in the lattice Boltzmann approach since the required hydrodynamic macro quantities imposed at the boundary and the variables of the algorithm, the so-called particle distributions, do not associate directly with each other. For instance, when trying to implement Dirichlet boundary conditions, at the boundary nodes, the velocity cannot be directly imposed, unlike conventional CFD, but it is necessary to set the particle distributions in order to have satisfied the required boundary velocity.

4.1 Literature Review of BCs

The traditional method employed in lattice Boltzmann simulations to implement the non-slip boundary condition is the bounce-back rule [82] [83]. Although it is very simple, it turns out to be an insufficiently accurate method, as demonstrated in [84] [85] [86] [87].

Another difficulty that must be considered is the imposition of the slip condition, which is more difficult to set than the no-slip condition. This is a problem that was dealt with in the LBMHYPE project, but several issues with the new implementation have been solved, as will be analyzed in Chapters 6 and 7. Other proposals for satisfying the slip condition can be found in the literature ([24], [23], [85], [88], [89], [90], [91]).

Subsequently, great effort was put by the research into developing boundary methods that were more precise than the bounce-back rule. Early boundary treatments were variants of the bounce-back rule and were only applicable to extremely basic regular boundary geometries, like flat walls. Various examples in the literature of these simple improvements to the bounce-back rule can be found

in [22] [23] [24] [89] [92] [93] [90] [94].

Following that, more research is put into finding approaches that deal with curved boundaries. For instance, [95] [96] proposed a first method that also considers non-rectilinear geometries, the algorithm, nevertheless, might be unstable.

To solve these problems of algorithm instability, [91] [97] proposed an improvement to [95] and [96], by including an additional neighboring node and therefore expanding the stability region, but without resolving completely the problem of the prior approach.

A different link-based method using linear interpolation with one or two neighbor nodes was proposed by Bouzidi et al. in 2001 [20], which can also be generalized, as shown in [98]. This type of boundary condition, being the basis of the new type of boundary condition implemented in this thesis, will be discussed in detail in section 5.3. In fact, the *Bouzidi Bounce Back with Normal* boundary conditions are nothing but an extension of the Bouzidi-type, to which a correction term is added to accurately implement the slip conditions, regardless of the orientation of the wall with respect to the lattice, since the normal direction (of the wall) is taken into account. By increasing the accuracy of the solution, there is also a slight increase in computational cost as it is necessary to have knowledge of the velocity and/or density at the node where the boundary scheme is established (i.e. this method also depends significantly on certain flow properties).

4.2 Bounce-Back Boundary Conditions

The LBM has the advantage that fixed no-slip boundaries along the grid's Cartesian directions can be implemented easily and efficiently while keeping the same level of accuracy of the method.

If the boundary exactly lies between the grid nodes, the most basic approach, known as the bounce-back method, is possible to implement. The concept is represented in Figure 4.1 and can be explained through the viewpoint of a particle. If a grid node adjacent was already inside the solid boundary (i.e. from picture 4.1, $\underline{x}_F + \Delta t \cdot \underline{\xi}_{i,b} = \underline{x}_{B,n}$), for directions $\underline{\xi}_{i,b}$ towards the boundary node, a propagating particle would basically "bounce back" to its original location. The particle would even be reflected in the transverse direction for a rough surface (i.e. pointing at a transverse node $\underline{x}_{B,tr}$ in picture 4.1).

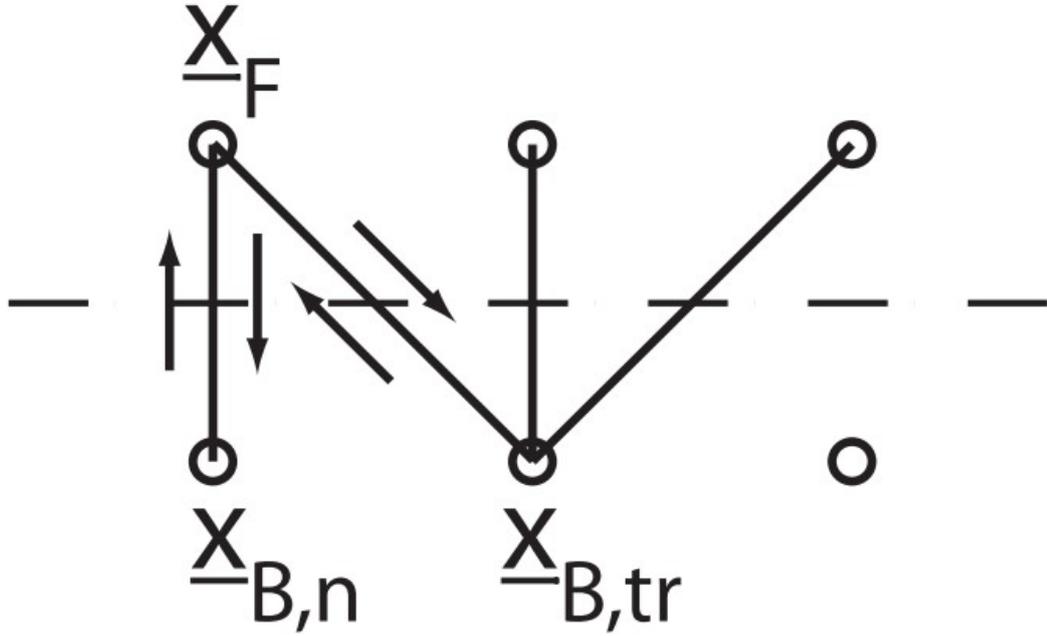


Figure 4.1: Bounce-back boundary conditions, No-Slip case

Since it is a simple reflection, the bounce-back rule for no-slip boundaries is given by the following relationship:

$$f_{\bar{i},\bar{b}}(t + \Delta t, \underline{x}_F) = f_{i,b}^*(t, \underline{x}_F) \quad (4.1)$$

where $f_{\bar{i},\bar{b}}$ is the opposite distribution to $f_{i,b}$ and the superscript "*" identifies the post-collision state.

Since of more interest in the topic of this thesis, the Bounce-Back boundary condition can also be employed to implement the slip condition, in such case a perfectly reflecting wall, as seen in figure 4.2, behaves as a free slip boundary.

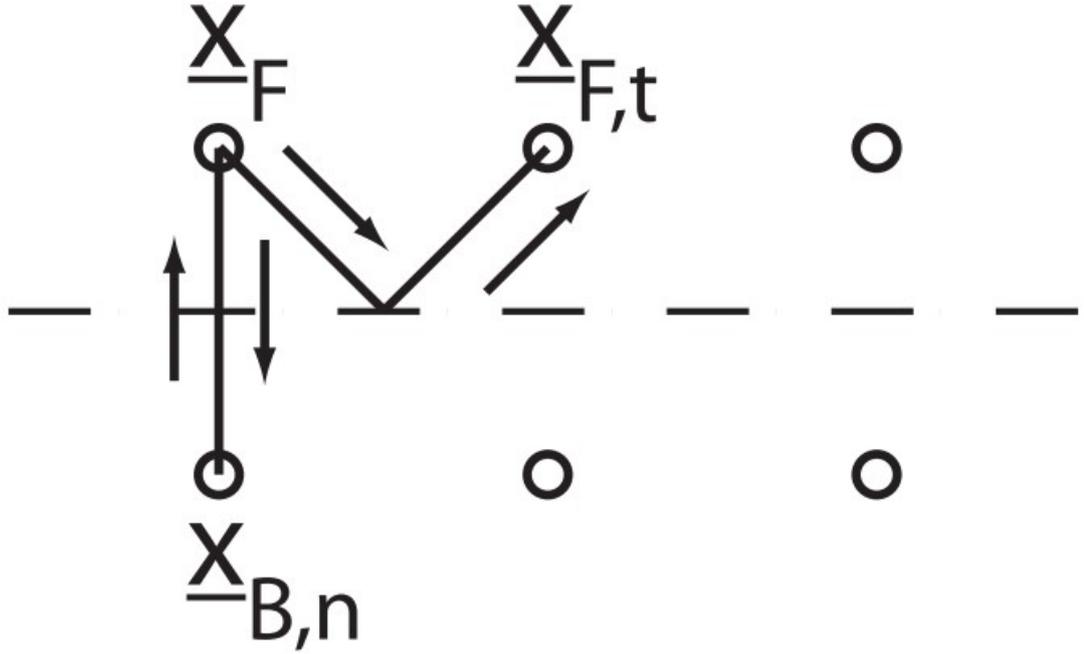


Figure 4.2: Bounce-back boundary conditions, Slip case

Different transverse directions are set in the direction of the boundary, having a tangential influence on the nearby fluid node $\underline{x}_{F,t}$:

$$f_{i,b}^-(t + \Delta t, \underline{x}_{F,t}) = f_{i,b}^*(t, \underline{x}_F) \quad (4.2)$$

A "2-time step" approach is frequently used, in which distributions are moved in the direction of the boundary node, the following time step without colliding, turned around and shifted back. However, the approach is not mass conservative like the original bounce-back approach.

The research presented in [86] demonstrates that bounce-back is second-order accurate in space. An example supporting this can be found in the reference [87], in which is presented that, in a pressure-driven Poiseuille flow, this straightforward bounce-back approach only achieved second-order precision when the boundary nodes were placed half a grid spacing from the solid wall ¹. Whenever the mid-link condition is not satisfied (for instance with moving / curved or sloped boundaries) the accuracy of the simple bounce-back rule is only first-order.

¹The mid-link condition is the formal name for this particular case.

According to the study mentioned in [99], the simplest bounce-back technique has a little slip velocity inaccuracy, this is also the reason for the error in the results obtained within the LBMHYPE project, reported in sections 6.4.2 and 7.3.1. This hidden error, as demonstrated by [86] [87] is evident when the Bounce-Back scheme is compared with the standard Chapman–Enskog analysis. The contribution made to research by [87] is also to predict, from a theoretical point of view, the value of the slip velocity for the BGK collision model.

In the bibliography [100] and [101], instead, it is shown that, in the case of the two-relaxation-time (TRT) collision model, the unwanted slip-velocity was due to the relaxation rates. The important contribution of these two articles is that a method for eliminating slip-velocity is also presented, a concept that was also extended to the multi-relaxation-time (MRT) collision model in [102]. In the theoretical framework, [102], the hidden mistakes are revealed by recasting the bounce-back distribution using the space Taylor expansion, what we deduce is that with the Bounce Back boundary condition, it is not possible to obtain a numerical solution to achieve accurate up to the second order.

The inaccuracies we will see in sections 6.4.2 and 7.3.1, however, are not only due to what was dealt with in [99], since in [103], in order to solve this problem, it is proposed a method that eliminates non-equilibrium error throughout the collision cycle (also known as the "magic" collision number for MRT collision).

4.2.1 Implementation on Python of Bounce Back BCs

Before moving on to the specific implementation of the boundary condition, there is a main class in the `boundary.py` script of `pylbm`, namely *Boundary*, which represents the boundary conditions of a lattice Boltzmann simulation in the `pylbm` library [4].

Boundary class

This class has three inputs basically, a domain object that represents the domain in which the simulation takes place, a generator object that instead implements the appropriate conditions for each stencil element, and finally, the *'dico'* dictionary, which also describes the boundary conditions to be implemented through the keys *'method'* (which specifies the type of BCs, so Bounce Back or Bouzidi for example) and *'value_bc'* which specifies, as the name already suggests, the value you have in the boundary. The *Boundary* class also initializes two attributes, the first is another dictionary that contains the lists of spatial indices and distances for each label (*bv_per_label*), the second is a list of methods for BCs instances (*methods*). Finally, in this class, there is also a method for figuring out which Bcs you need to apply, the *"compute_boundary"* method, which takes a distribution function and a list of boundary conditions as inputs, applies the boundary conditions to the

distribution function. This method is part of the Boundary class.

BoundaryMethod class

Once we have seen very briefly how the algorithm figures out where to apply the desired boundary condition, it is good to describe the key elements that are common to any boundary condition that one wants to apply to a given surface. The 'mother' class of any boundary condition in pylbm is *BoundaryMethod*. That class includes several methods, mainly to perform the following operations: updating the equilibrium distribution function, computing the distribution function at the equilibrium on the boundary, and obtaining the point indices required to compute the boundary condition.

The attributes of the class provide information about both the boundary conditions and the domain over which the simulation takes place. The main attributes of the *BoundaryMethod* class are as follows:

- **feq:** a Numpy ndarray (n-dimensional array) representing the distribution function's equilibrium values on the boundary
- **rhs:** a Numpy ndarray that represents the extra terms needed to fix the boundary values
- **distance:** a Numpy ndarray representing the distance to the boundary (required for some boundary condition algorithms, such as Bouzidi or Bouzidi Bounce Back with Normal)
- **istore:** a Numpy ndarray representing the indices of points where the boundary condition is applied
- **ilabel:** a Numpy ndarray representing the label of the boundary
- **iload:** a Numpy list of indices of points needed to compute the boundary condition
- **value_bc:** a dictionary of the prescribed values on the boundary

BoundaryMethod also has a number of functions implemented, the most important of which are as follows:

- **__init__:** initializes the attributes of the class.
- **fix_iloal:** transposes *iload* and *istore* and converts them to contiguous arrays of int32 type.

- **prepare_rhs:** compute the distribution function at the equilibrium with the value on the boundary.
- **update_feq:** update the *feq* attribute of the *BoundaryMethod* object, which represents the distribution function's equilibrium values on the boundary. This method, in the original version of *pylbn*, is used only for time-dependent BCs which have a direct dependence on simulation. Since the Bouzidi Bounce Back with Normal also have a dependence on the simulation ², it will be exploited such a method already present within the library.
- **__get_istore_iloal_symb:** generate symbolic variables for the *istore* and *iloal* attributes of the *BoundaryMethod* object.
- **__get_rhs_dist_symb:** generate symbolic variables for the *rhs* and *distance* attributes of the *BoundaryMethod* object.
- **update:** update the distribution functions with the boundary conditions specified by the *BoundaryMethod* object.
- **__get_args:** retrieve the necessary arguments for updating the distribution functions with the boundary conditions specified by the *BoundaryMethod* object.
- **move2gpu:** move the arrays needed to compute the boundary conditions on the GPU memory.

Bounce Back class

The *BounceBack* class is a subclass of the *BoundaryMethod* class in the *pylbn* library and it is the implementation of a boundary condition of type bounce-back. In analogy with previous classes, this one will be simply explained in broad terms, as it is not the subject of this thesis, and implementing a well-known boundary condition. For any further information, please refer to the bibliographic entry [4]. Being a subclass of *BoundaryMethod*, the *BounceBack* class inherits many functions of the parent class, and for that reason has only three functions:

- **set_iloal:** computes the indices that are needed (symmetric velocities and space indices) for these particular BCs.
- **set_rhs:** computes and sets the additional terms needed to fix the boundary values.

²in fact, to evaluate p^* 5.17 it is necessary to know the value of the moments, thus the physical parameters

- **generate:** generates the numerical code for the boundary condition.

4.3 Bouzidi Bounce-Back Boundary Conditions

Bouzidi's boundary conditions are presented in [20], which focused on the lattice Boltzmann equation LBE's velocity boundary condition for curved boundaries by combining the "bounce-back" strategy with spatial interpolations of the first or second order. These boundary conditions are clearly also applicable in the case of fixed geometry, as in our case, and are of particular interest since the presented boundary condition are a simple, robust, efficient, and accurate scheme. It has been proved numerically to achieve second-order accurate velocity and first-order accurate pressure.

For boundaries parallel with any of the discrete velocities of the model, previous boundary conditions ("bounce-back" or "specular reflection") were demonstrated to be adequate [16]. For general boundaries, it was discovered that taking into account some sort of average over a region that included several lattice nodes, coming up with mean boundary conditions, was necessary [84] [86] [104]. Physically, this seems as if the dynamics' relaxation behavior behaves as a low-pass spatial frequency filter which removes the specifics of the smooth boundary representation by a series of linear steps drawn on the base lattice.

In [20], a straightforward approach using interpolations and the intuitive concept of "bounce-back" is proposed for dealing with boundaries of arbitrary shape in the LBE method. In terms of accuracy and simplicity of the algorithm, it is compatible with the idea of LBE approaches (i.e. the use of a regular lattice) and most importantly it is not computationally expensive. This is different from conventional CFD approaches, which employ nonuniform meshes designed to match solid boundaries. There are some studies in the literature that aim to reproduce these concepts, applied to LBM [105] [106].

Focusing now more on how these types of BCs actually work, as reported in [20], to handle the advection step when boundaries are present, based on the one-dimensional example shown in Figure 4.3, Bouzidi proposes an intuitive method.

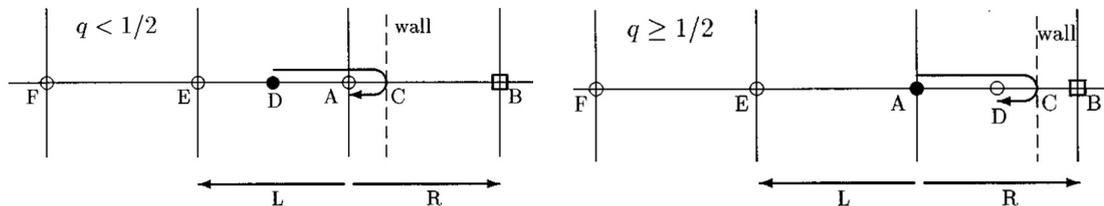


Figure 4.3: Details of the collision process near the boundary for different distances between the boundary node and the wall.

In particular in Figure 4.3:

- On the left, there are fluid nodes (i.e. A, D, E, and F), with A being the final one close to the wall.
- On the right, there are solid nodes, so the only one important for the simulation is just the first one (i.e. B).
- Last but not least, C is the node that identifies the wall.

The following relation 4.3 identifies the dimensionless location of the wall.

$$q = \frac{|AC|}{|AB|} \tag{4.3}$$

Assuming that each particle distribution function moves at a speed of one unit, except when q is equal to 0, $1/2$, or 1, a particle leaving A and reflecting on the wall (a "bounce-back" situation) won't arrive at a fluid node after traveling a total distance of 1. This implies that the population of particles at A with velocity -1 (represented by the vector with the label L in 4.3) is unknown after the collision step.

- For $q < \frac{1}{2}$, Bouzidi et al. set up the population of fictitious particles at location D that will move to location A after bouncing back on the wall at location C using the information known from the fluid.
- For $q \geq \frac{1}{2}$, the unknown quantities at A will be determined using information related to the particle departing A and arriving in D as well as the new (post-advection) situation at fluid nodes E (and F).

Bouzidi's boundary conditions thus apply linear or quadratic interpolation formulas involving values at two or three nodes in both situations.

This first one-dimensional example was shown simply to make the basic concept easier to understand, but in order to perform accurate calculations in two (or three) dimensions, one needs to first identify all lattice links that cross the solid boundary.

Assume that \mathbf{r}_l is a fluid node and $\mathbf{r}_l + \mathbf{c}_i$ is a solid node. Let's denote the opposite of the speed \mathbf{c}_i by $\mathbf{c}_{i'}$ (i.e. $\mathbf{c}_{i'} = -\mathbf{c}_i$). Using linear interpolation, the relationship for Bouzidi's boundary conditions is as follows³:

³Instability is the reason for employing two distinct expressions depending on the value of q , as discussed in [107].

$$f_{i'}(\mathbf{r}_l, t + 1) = 2qf_i^c(\mathbf{r}_l, t) + (1 - 2q)f_i^c(\mathbf{r}_l - \mathbf{c}_i, t) \quad q < \frac{1}{2} \quad (4.4)$$

$$f_{i'}(\mathbf{r}_l, t + 1) = \frac{1}{2q}f_i^c(\mathbf{r}_l, t) + \frac{2q - 1}{2q}f_i(\mathbf{r}_l, t) \quad q \geq \frac{1}{2} \quad (4.5)$$

The parameter f^c , in the right-hand-side of the equations 4.4 and 4.5, are taken after collision and before propagation. The left-hand side $f(\cdot, t + 1)$ indicates the values of a particle distribution function following a collision and following propagation (i.e. following a complete LBM time step) ⁴.

Using quadratic interpolation instead, the relationship for Bouzidi's boundary conditions is as follows:

$$f_{i'}(\mathbf{r}_l, t + 1) = q(2q + 1)f_i^c(\mathbf{r}_l, t) + (1 + 2q)(1 - 2q)f_i^c(\mathbf{r}_l - \mathbf{c}_i, t) - q(1 - 2q)f_i^c(\mathbf{r}_l - 2\mathbf{c}_i, t) \quad q < \frac{1}{2} \quad (4.6)$$

$$f_{i'}(\mathbf{r}_l, t + 1) = \frac{1}{q(2q + 1)}f_i^c(\mathbf{r}_l, t) + \frac{(2q - 1)}{q}f_{i'}^c(\mathbf{r}_l, t) + \frac{(1 - 2q)}{(1 + 2q)}f_{i'}^c(\mathbf{r}_l - \mathbf{c}_i, t) \quad q \geq \frac{1}{2} \quad (4.7)$$

It is interesting to note that the relation 4.6 turns out to be an upwind interpolation, in contrast to the relation 4.7 which turns out to be a downwind interpolation.

Finally, to be complete, it should be pointed out that the classical Bounce Back approach is included in Bouzidi's relation, in case of zero-order interpolation, the relation already seen above 4.1 is obtained, which is written with the new notation adopted by Bouzidi becomes the following⁵:

$$f_{i'}(\mathbf{r}_l, t + 1) = f_i^c(\mathbf{r}_l, t) \quad (4.8)$$

A first consideration that can be done about these boundary conditions is that the suggested technique uses interpolations, which makes it more stable than some existing schemes that employ extrapolations [90] [96].

Another consideration that can be pointed out is that in equations 4.4, 4.5, 4.6, and 4.7, the change in q is continuous. In the specific case where $q = \frac{1}{2}$,

⁴In other words, this indicates that the only distinction between the dynamics at nodes near a boundary and at bulk nodes (i.e., the points that are not directly affected in the boundary conditions, i.e., the domain nodes) is the propagation step.

⁵Obviously in this case, there is no longer a dependence on the dimensionless parameter q .

the interpolation formulas 4.4, 4.5, 4.6, and 4.7 are reduced to the "bounce-back" approach.

Although it is one of the boundary conditions implemented on `pylbm` [4], and already has its advantages in terms of accuracy and simplicity, there are improvements to this implementation in the literature. For example, a research team extended Bouzidi's rule in [108] by adding an additional on-wall node and implemented a uniform approach that could be used in any fluid-solid interaction situation.

4.3.1 Implementation on Python of Bouzidi BCs

The *BouzidiBounceBack* class is a subclass of the *BoundaryMethod* class in the `pylbm` library and it is the implementation of a boundary condition of type Bouzidi Bounce-Back. In analogy with previous classes, this one will be simply explained in broad terms, as it is not the subject of this thesis. For any further information, please refer to the reference [4]. Being a subclass of *BoundaryMethod*, the *BouzidiBounceBack* class inherits many functions of the parent class:

- **set_iloat:** computes the indices that are needed (symmetric velocities and space indices) for these particular BCs. In the case of this implementation, unlike the basic Bounce-Back, there will be two different *iloat*, since, as seen from relation 4.4 and 4.5, two different relations must be implemented depending on the distance between the boundary node and the wall.
- **__get_args:** returns a dictionary of arguments used by the code generator to produce the boundary condition's numerical code.
- **set_rhs:** computes and sets the additional terms needed to fix the boundary values.
- **generate:** generates the numerical code for the boundary condition.

Chapter 5

Implementation of BCs for general geometries

Applying Bounce-Back or Bouzidi boundary conditions to the wedge problem (or the cylinder problem), which will be discussed in sections 6.4.2 (and 7.3.1) of this thesis, leads to inaccurate results for a series of reasons that will be discussed later. Therefore, in order to create an improvement to Bouzidi's boundary conditions which can provide an optimal result not only with geometries that conform to the LBM lattice, Graille and Dubois developed this new type of boundary condition. One of the main problems encountered in the study of supersonic flow over a wedge is that the slip condition is applied as a no-slip boundary condition, which is inappropriate when solving the Euler equations. Therefore, this new type of boundary condition has the task of appropriately applying the slip condition, which is more complicated to apply than the no-slip condition. That is because whereas the no-slip boundary condition sets the velocity on the wall to zero, the slip condition requires quantifying the tangential velocity on the wall surface.

5.1 Interpretation of *f-functions* as flux

Graille and Dubois applied the conventional finite-volume scheme to a LB lattice node, not forgetting, however, that in the case of LBM, the information is contained in the just point. The framework is to exploit the analogy between the four sides of a D_2Q_4 scheme lattice cell and the four particle distribution functions, for that reason, this idea is applicable to the LB D_2Q_4 scheme only.

As shown in Figure 5.1, there are four possible directions for the particle distribution function, taking the following notation for propagation directions:

- f_1 is for the particle distribution that is moving toward the right
- f_2 is for the particle distribution that is moving toward the top
- f_3 is for the particle distribution that is moving toward the left
- f_4 is for the particle distribution that is moving toward the bottom

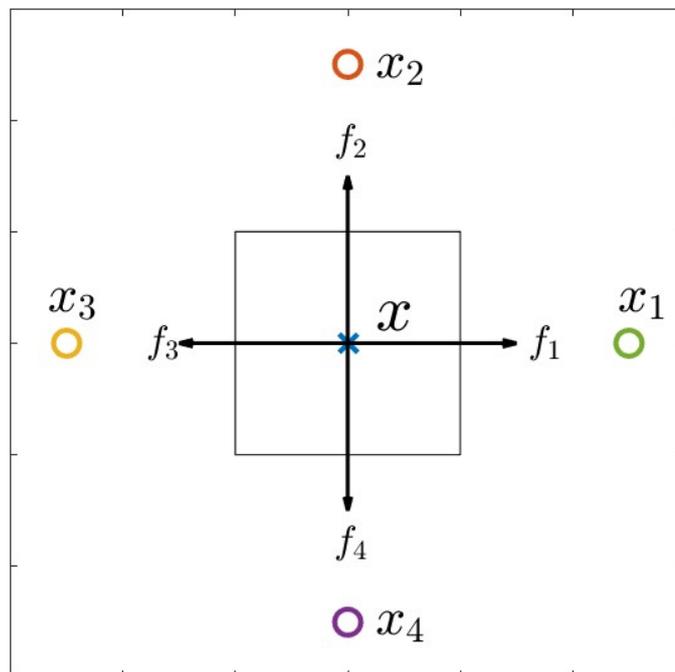


Figure 5.1: Single lattice of the LBM with the four particle distribution functions moving along the four directions

Let us consider a generic vectorial scheme, thus for each conserved variable (φ_i), we have a set of four particle distributions. Assume for simplicity that there is only one conserved variable (so we are solving a single equation, e.g., conservation of mass in Euler). Therefore, having a single φ , it is possible to say that around a mesh point x , we have for the conserved moment the following relations:

$$\varphi(x, t) = \sum_{j=1}^4 f_j^*(x, t) = f_1^*(x, t) + f_2^*(x, t) + f_3^*(x, t) + f_4^*(x, t) \quad (5.1)$$

Supposing we are evaluating particle distribution functions at point x at instant $t + \Delta t$, in other words, we will have four particle distribution functions coming from the four neighboring nodes x_1, x_2, x_3 , and x_4 respectively. We then evaluate the four particle distribution functions after the advection, i.e. after the transport and before the relaxation (it is without the superscript $*$, which stands for after relaxation phase (so post-collision)).

$$\begin{aligned} f_1(x, t + \Delta t) &= f_1^*(x_3, t) \\ f_2(x, t + \Delta t) &= f_2^*(x_4, t) \\ f_3(x, t + \Delta t) &= f_3^*(x_1, t) \\ f_4(x, t + \Delta t) &= f_4^*(x_2, t) \end{aligned} \quad (5.2)$$

In 5.2, we simply applied the last step of the LBM (i.e. transport)¹. Then substituting the relations 5.2 into 5.1 at the next time step, we obtain the following relation:

$$\varphi(x, t + \Delta t) = \sum_{j=1}^4 f_j(x, t + \Delta t) \quad (5.3)$$

$$= f_1(x, t + \Delta t) + f_2(x, t + \Delta t) + f_3(x, t + \Delta t) + f_4(x, t + \Delta t) \quad (5.4)$$

$$= f_1^*(x_3, t) + f_2^*(x_4, t) + f_3^*(x_1, t) + f_4^*(x_2, t) \quad (5.5)$$

At this point we make a difference between 5.5 and 5.1, yielding a relationship that reminds a finite volume scheme:

$$\begin{aligned} \varphi(x, t + \Delta t) - \varphi(x, t) &= [f_1^*(x_3) - f_3^*(x)] + [f_2^*(x_4) - f_4^*(x)] + \\ &\quad [f_3^*(x_1) - f_1^*(x)] + [f_4^*(x_2) - f_2^*(x)] \end{aligned} \quad (5.6)$$

¹For example, $f_4(x, t + \Delta t)$ after advection, will be equal to the particle distribution function associated with direction 4, evaluated, however, at the point above x (i.e x_2), before the transport phase (i.e. post-collision).

At this point, we come to the main aspect of this framework proposed by Profs. Graille and Dubois, which is defining a numerical flux through each of the four sides of the lattice unit. Therefore we define the four number streams as follows:

$$\begin{cases} \Phi_3 = \lambda \left(f_3^*(x) - f_1^*(x_3) \right) \\ \Phi_4 = \lambda \left(f_4^*(x) - f_2^*(x_4) \right) \\ \Phi_1 = \lambda \left(f_1^*(x) - f_3^*(x_1) \right) \\ \Phi_2 = \lambda \left(f_2^*(x) - f_4^*(x_2) \right) \end{cases} \quad (5.7)$$

Therefore, by substituting the relations just defined 5.7 into 5.6, we obtain the following more compact relation:

$$\frac{1}{\Delta t} \left[\varphi(x, t + \Delta t) - \varphi(x, t) \right] + \frac{1}{\Delta x} \sum_{j=1}^4 \Phi_j = 0 \quad (5.8)$$

The relationship 5.8 therefore allows the calculation of the numerical flux Φ_j between x and x_j , being an indirect link between the conservative variables and this numerical flux, which depends on the particle distribution functions. Based on this flux, new boundary conditions will be implemented to satisfy the slip condition in a case where the boundary surface does not cross the grid points. The basic idea is to compute the numerical flux that the wall must impose on the external fluid in order to have a zero velocity along the normal direction.

5.1.1 Euler Equation

After having seen the basic idea, let us look at how to apply this concept to the problem we want to solve, i.e. the two-dimensional Euler equations:

$$\frac{\partial}{\partial t} \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{Bmatrix} + \frac{\partial}{\partial x} \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uE + pu \end{Bmatrix} + \frac{\partial}{\partial y} \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vE + pv \end{Bmatrix} = 0 \quad (5.9)$$

To describe the fluid's behavior, as can be seen in equations 5.9, we have chosen Euler's equations written in a conservative form, i.e. as a function of density, momentum, and total energy. Euler equations written in conservative variables generally express the variations of these quantities and the fluxes of these quantities through a control volume. In 5.9, the relationships are written as a system of partial derivative equations, where the conserved quantities are the independent variables and the fluxes are the functions of the partial derivatives of the conserved quantities.

Since the problem we have to solve deals with geometries that do not conform to the typical LBM regular lattice, we will have to apply the boundary conditions in a rotated frame of reference. For this reason, we write the system 5.9, in a generic reference system (ξ, η) , but first, we should point out that the momentum $J = (\rho u, \rho v)$ and its fluxes $\Phi_{xx}, \Phi_{xy}, \Phi_{yx}, \Phi_{yy} = (\rho u^2 + p, \rho uv, \rho uv, \rho v^2 + p)$ are invariant by rotation ².

We need to point out the difference between the two similar notations that are being adopted, in fact, we have the momentum fluxes $\Phi_{xx}, \Phi_{xy}, \Phi_{yx}, \Phi_{yy}$, and the numerical fluxes $\Phi_1, \Phi_2, \Phi_3, \Phi_4$ 5.7.

- In the first case, we refer to momentum fluxes, i.e. physical fluxes, through vector notation which in the case of Euler 2D having four equations (mass, momentum x, momentum y, and energy), there is a flux for each equation.
- In the case of 5.7 relations, there are always four fluxes, one for each of the directions with which the Boltzmann equation has been discretized (i.e. 4, since we are using D_2Q_4).

Therefore, since vectorial schemes will be adopted, and the basic scheme D_2Q_4 , that solve a single conservation equation, is associated with four numerical fluxes, there will be a total of $4 \times 4 = 16$ numerical fluxes to describe the four-momentum fluxes.

We then write the equations 5.9, effectively as a function of the conservative variables $\{\rho; q_x = \rho u; q_y = \rho v; \rho E\}$ and the fluxes associated with them:

$$\begin{cases} \partial_t \rho + \partial_x q_x + \partial_y q_y = 0 \\ \partial_t q_x + \partial_x \Phi_{xx} + \partial_y \Phi_{xy} = 0 \\ \partial_t q_y + \partial_x \Phi_{xy} + \partial_y \Phi_{yy} = 0 \\ \partial_t E + \partial_x \psi_x + \partial_y \psi_y = 0 \end{cases} \quad (5.10)$$

For the momentum conservation equations we can have the following vectorial notation:

$$\partial_t q + \nabla \cdot \Phi = 0 \quad (5.11)$$

defining vectorial fluxes through a symmetric matrix:

$$\Phi = \begin{Bmatrix} \Phi_{xx}, \Phi_{xy} \\ \Phi_{yx}, \Phi_{yy} \end{Bmatrix} \quad (5.12)$$

²If a quantity's value or shape does not change when the system is rotated around a certain point or axis, in physics and mathematics it is said to be invariant by rotation. These invariants by rotation are significant because they characterize properties independently of the system's orientation.

In order to rotate the inertial frame of reference of equations 5.10, we define the new generic frame of reference (ξ, η) .

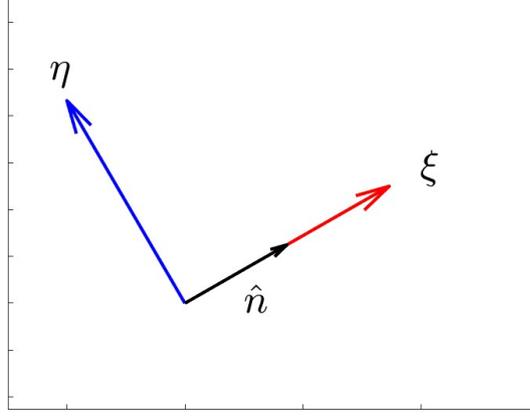


Figure 5.2: Frame of reference (ξ, η) turned by a generic angle θ

In a rotation of angle Θ which allows to point in a direction n 5.2 the coordinates (u_n, u_t) in the frame (ξ, η) are written:

$$\begin{cases} u_n = u \cos(\Theta) + v \sin(\Theta) \\ u_\tau = -u \sin(\Theta) + v \cos(\Theta) \end{cases} \quad (5.13)$$

This operation is necessary to separate the tangential coordinates and the normal coordinates. Then the Euler equations can be written in the new frame as follow:

$$\frac{\partial}{\partial t} \begin{Bmatrix} \rho \\ \rho u_n \\ \rho u_\tau \\ \rho E \end{Bmatrix} + \frac{\partial}{\partial \xi} \begin{Bmatrix} \rho u_n \\ \rho u_n^2 + p \\ \rho u_n u_\tau \\ \rho u_n E + p u_n \end{Bmatrix} + \frac{\partial}{\partial \eta} \begin{Bmatrix} \rho u_\tau \\ \rho u_n u_\tau \\ \rho u_\tau^2 + p \\ \rho u_\tau E + p u_\tau \end{Bmatrix} = 0 \quad (5.14)$$

In this way, similarly to 5.12, momentum flux can be expressed in the new reference system

$$\begin{aligned} \Phi_{nn} &= \rho u_n^2 + p \\ \Phi_{n\tau} &= \Phi_{\tau n} = \rho u_n u_\tau \\ \Phi_{\tau\tau} &= \rho u_\tau^2 + p \end{aligned}$$

After this long introduction on how the concept of flux can be introduced, we can now move on to explain operationally how to exploit what we have just seen to apply the slip condition.

From a purely physical point of view, it is known that a wall (or slip) condition is expressed by the condition:

$$(\vec{u} \cdot \hat{n}) = u_n = 0 \quad (5.15)$$

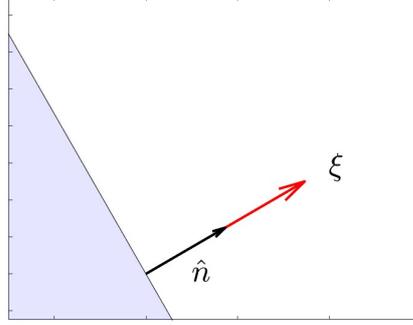


Figure 5.3: Direction normal to wall

From Euler's equations 5.14, by extracting only what happens along the normal direction (i.e. ξ), it will be seen that the only non-zero contribution is from the momentum flux in the ξ direction.

$$\frac{\partial}{\partial \xi} \begin{Bmatrix} \rho u_n \\ \rho u_n^2 + p \\ \rho u_n u_\tau \\ \rho u_n E + p u_n \end{Bmatrix} \rightarrow \Phi_n = \begin{Bmatrix} 0 \\ p^* \\ 0 \\ 0 \end{Bmatrix} \quad (5.16)$$

From 5.16, it can be seen that only pressure ' p ' (called p^*) remains in the relationship, relating to the momentum flux contribution in the ξ direction (i.e. normal). To calculate this pressure, from literature [109], Dubois, dealing with the partial Riemann problem proposes the relation:

$$p + \rho c u_n = p^* \quad (5.17)$$

with ρ density, p the pressure, c the speed of sound, and u_n the normal speed. All these values are not taken at the wall, also because the velocity u_n would be zero by hypothesis, but at the node in the domain closest to the wall.

The relationship 5.17 to evaluate the quantity p^* is based on a framework by Dubois [109] and was derived from the acoustics equation, considering a Riemann problem orthogonal to the wall.

What is crucial to emphasize is that, unlike conventional CFD, in imposing the slip condition (i.e. $u_n = 0$), the idea is to compute the pressure p^* in order to fix

$u_n = 0$. In other words, we compute the pressure p^* that the wall imposes on the fluid for the result in terms of normal velocity in the wall to be zero.

What can be deduced from the 5.17 is that in implementing the boundary conditions, unlike the classical Bounce-Back or Bouzidi, it will be necessary to know the values of the fluid's macroscopic properties at each time step, which will certainly lead to a higher computational cost, but will produce satisfactory results as we will see in 6.4.3 and 7.3.2. To discover macroscopic information concerning the flows that the LBE approach simulates, it is necessary to consider that the single-particle distribution function's information can be provided directly by the LBE approach. In fact, as seen in Chapter 3.3, it is simple to determine macroscopic flow information, such as flow velocity and local pressure distributions, based on the distribution function.

In the case of a boundary, the fluid-boundary interaction affects the momentum transfer of all the particles that come into contact with the boundary, which as a consequence changes the distribution function at the boundary. The fluid-boundary interaction is described in the LBE approach in order to produce the desired macroscopic boundary conditions for the appropriate hydrodynamic scenarios. Realizing the locations of the effective boundary for specific fluid-boundary interactions where the desired macroscopic boundary conditions are satisfied is crucial when studying the LBE boundary conditions [84] [86] [104].

Returning to focus on the relationship 5.16, since the scheme to be adopted (D_2Q_4) only has velocities along Cartesian directions, it is necessary to calculate the Cartesian components of the normal flux Φ_n . Through algebraic steps, we obtain the following relationship:

$$\frac{\partial}{\partial t} \begin{Bmatrix} \rho u \\ \rho v \end{Bmatrix} + \frac{\partial}{\partial \xi} \begin{Bmatrix} \rho u u_n + p \cos \Theta \\ \rho v u_n + p \sin \Theta \end{Bmatrix} + \frac{\partial}{\partial \eta} \begin{Bmatrix} \rho u u_\tau - p \sin \Theta \\ \rho v u_\tau + p \cos \Theta \end{Bmatrix} = 0 \quad (5.18)$$

At this step, considering only the normal component locally to the wall (i.e. $n_\xi = 1$ and $n_\eta = 0$), setting the condition of no fluid crossings the wall $u_n = 0$, we obtain that the momentum flux in the normal direction has Cartesian components:

$$p^* \begin{Bmatrix} \cos \Theta \\ \sin \Theta \end{Bmatrix} = p^* \hat{n} \quad (5.19)$$

So the four components for the Euler equation of the normal flux Φ_n are written as follows:

$$\tilde{\Phi}_n = \begin{Bmatrix} 0 \\ p^* \cos \Theta \\ p^* \sin \Theta \\ 0 \end{Bmatrix} \quad (5.20)$$

with p^* computed as 5.17 and $\hat{n} = (\cos(\theta), \sin(\theta))$.

Finally, returned to a Cartesian reference system, it is possible to connect the four relations 5.7 which compute the numerical flux, with the relation of $\tilde{\Phi}_n$ 5.20 just written in the hypothesis where $u_n = 0$, with p^* calculated as 5.17. In this way, it is possible to calculate the physical pressure that the wall must impose on the external flow to fulfill the slip condition. This idea can be used to improve both the Bounce Back and Bouzidi boundary conditions.

5.2 Bounce-Back with Normal

Let us then look at how to extend the boundary conditions of Bounce-Back, through the normal flux just defined. First, however, in order for it to be possible to generalize the relationships, we adopt a convention to denote neighboring points differently. Therefore, the convention adopted is as follows for the 4 neighboring points:

$$\begin{cases} \sigma(1) = 3 \\ \sigma(2) = 4 \\ \sigma(3) = 1 \\ \sigma(4) = 2 \end{cases} \quad (5.21)$$

Convention 5.21 is based on the directions from which the particle distribution functions came at the previous time instant 5.4.

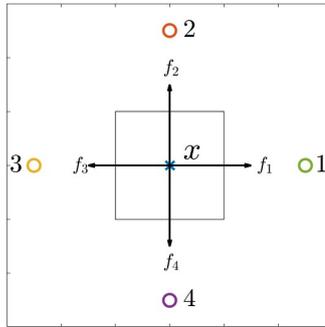


Figure 5.4: D_2Q_4 scheme, a convention adopted to indicate neighboring points

To apply the innovative concepts introduced to the classical Bounce-Back boundary condition, we define the external fluid normal as follows:

$$\hat{n} = \begin{cases} \cos(\theta) \\ \sin(\theta) \end{cases} \quad (5.22)$$

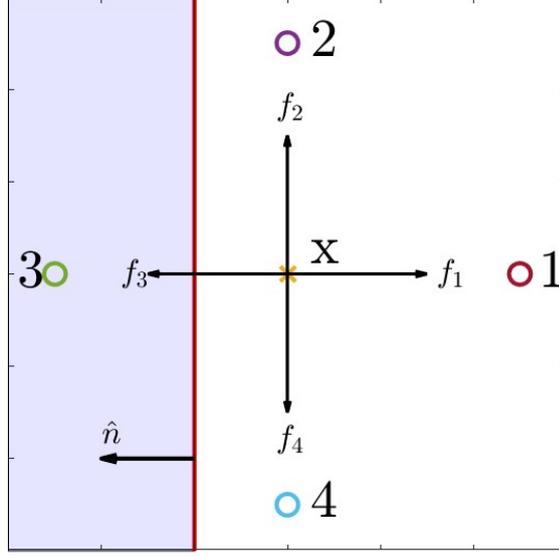


Figure 5.5: Trivial case in which we have $\theta = \pi$

Let us rewrite the relation 5.20, where the wall flux of the Euler equations is calculated with:

$$\Phi_n = \begin{pmatrix} 0 \\ p^* \cos(\theta) \\ p^* \sin(\theta) \\ 0 \end{pmatrix} \quad (5.23)$$

where, as discussed in relation 5.17, we have the pressure equal to $p^* = p + \rho c(u_n)$, which corresponds to a flux Φ , as introduced previously 5.7. We then write the numerical flux, using the new convention adopted 5.21:

$$\lambda \left[f_{\sigma(j)}^*(x) - f_j^*(x_{\sigma(j)}) \right] = \Phi_{\sigma(j)} = \Phi_n \quad (5.24)$$

By arranging the previous relation 5.24, we obtain a Bounce-Back boundary condition, to which, however, the additional flux term Φ_n was added.

$$f_j^*(x_{\sigma(j)}) = f_{\sigma(j)}^*(x) - \frac{1}{\lambda} \Phi_n \quad (5.25)$$

This new type of boundary condition was presented only on a theoretical level and was not implemented in the Python pylbm [4] package, as it was preferred to directly implement the Bouzidi Bounce Back with Normal.

5.3 Bouzidi Bounce-Back with Normal

Similarly, the concept of Bounce Back boundary conditions has been extended, it is possible to extend the Bouzidi boundary conditions, described in detail in section 4.3. Then we write Bouzidi's classical relationship 4.4, using the new convention adopted 5.21:

$$f_j^*(x_{\sigma(j)}) = 2\xi f_{\sigma(j)}^*(x) + \left(1 + 2\xi\right) f_{\sigma(j)}^*(x_j) \quad 0 \leq \xi \leq \frac{1}{2} \quad (5.26)$$

$$f_j^*(x_{\sigma(j)}) = \frac{1}{2\xi} f_{\sigma(j)}^*(x) + \left(1 - \frac{1}{2\xi}\right) f_j^*(x) \quad \frac{1}{2} \leq \xi \leq 1 \quad (5.27)$$

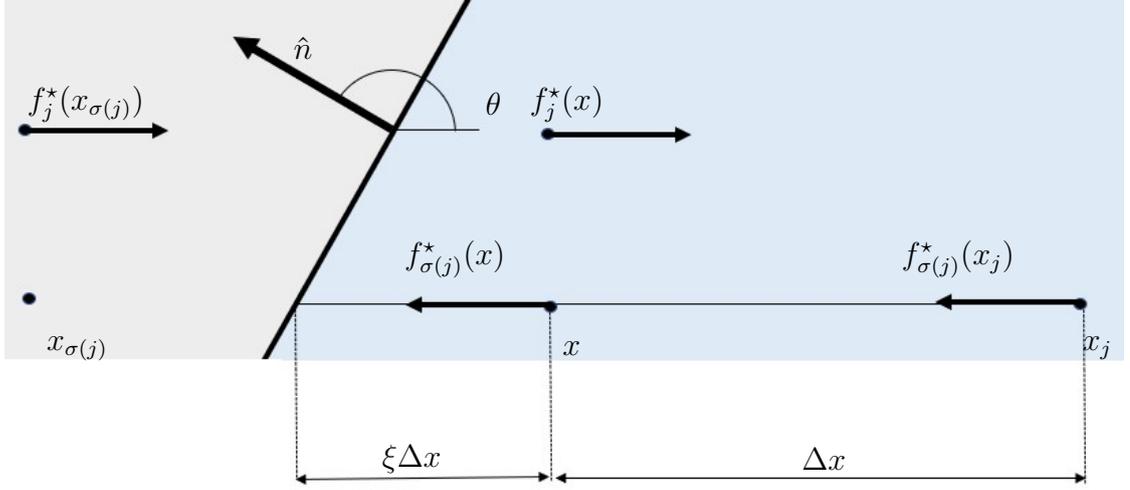


Figure 5.6: Case of a rebound treated with Bouzidi Bounce Back with Normal

Similar to what has been done with simple Bounce-Back BCs, we can apply a condition of type $f_j^*(x_{\sigma(j)}) = f_{\sigma(j)}^*(x) + \Phi$, to Bouzidi conditions.

$$f_j^*(x_{\sigma(j)}) = \left[2\xi f_{\sigma(j)}^*(x) + \left(1 + 2\xi\right) f_{\sigma(j)}^*(x_j) \right] - \frac{1}{\lambda} \Phi_n \quad 0 \leq \xi \leq \frac{1}{2} \quad (5.28)$$

$$f_j^*(x_{\sigma(j)}) = \frac{1}{2\xi} \left[f_{\sigma(j)}^*(x) - \frac{1}{\lambda} \Phi_n \right] + \left(1 - \frac{1}{2\xi}\right) f_j^*(x) \quad \frac{1}{2} \leq \xi \leq 1 \quad (5.29)$$

where the normal wall flux is calculated using the Euler equations:

$$\Phi_n = \begin{pmatrix} 0 \\ p^* \cos \Theta \\ p^* \sin \Theta \\ 0 \end{pmatrix} \quad (5.30)$$

5.3.1 Implementation on Python of Bouzidi Bounce-Back with Normal Boundary Conditions

After dealing with the Bouzidi-Bounce-Back with Normal boundary conditions from a theoretical point of view, we now proceed to discuss the implementation of this new type within the Python library `pylbn` [4].

The new implementation took its design cue from the already existing and briefly discussed in section 4.3.1 of the last chapter `BouzidiBounceBack` class, which, in fact, is the mother class of the newly implemented subclass (i.e. **BouzidiBounceBackWithNormalX** and **BouzidiBounceBackWithNormalY**), which will be discussed in this section and whose code is given in full in Appendix B.

BouzidiBounceBackWithNormalX

Being a subclass of `BouzidiBounceBack`, the `BouzidiBounceBackWithNormalX` class inherits many functions of the parent class:

- **update_feq**: As mentioned in paragraph 4.2.1, related to the Boundary Class, in this new implementation of boundary conditions, it is necessary to know values from the simulation and update them at each time step to "correct" BCs from time to time. The `update_feq` function then has the task of updating the equilibrium distribution function `feq`.
- **update_flux**: This is the main function within class `BouzidiBounceBackWithNormalX`, as it is responsible for calculating the corrective flux to be added to the Bouzidi boundary conditions to obtain slip conditions. Going on to analyze the code in detail, as is possible to see in line 6 of the reported part of the code below, one of the first things to do is to compute the indices where the boundary conditions are applied. As it is reported then in the code first we extract the values given in `istore`³, which are shifted by one cell so as to obtain the first node within the geometry, thus outside the domain. Having extracted the moments from the simulation in row 3 from Listing 5.1, it is possible to calculate the values of the variables associated with each of the conserved moments (i.e. ρ , Q_x , Q_y , and E). Computed density, momentum along x and y, and energy, it is easy to obtain the components of velocity 3.9 and pressure 3.10, so as to calculate the pressure p^* 5.17 needed to calculate the flux. Finally, the following function implements a different equation (5.28 or 5.29) depending on the distance between the boundary node and the wall, how to compute the additional term for the Bouzidi-Bounce-Back with Normal boundary condition.

³a Numpy ndarray representing the indices of points where the boundary condition is applied

Listing 5.1: Function `update_flux` of class `BouzidiBounceBackWithNormalX` in the `boundary.py` script implemented in the `pylbn` library

```

1 def update_flux(self, simulation):
2     gamma = 1.4
3     m = simulation.m_halo
4     k = self.istore[:, 0]
5     v = self.stencil.get_all_velocities()
6     self.flux_indices = self.istore[:, 1:] + v[k]
7
8     RHO = m[rho][self.flux_indices[:, 0], self.flux_indices[:, 1]]
9     QX = m[qx][self.flux_indices[:, 0], self.flux_indices[:, 1]]
10    QY = m[qy][self.flux_indices[:, 0], self.flux_indices[:, 1]]
11    E_ = m[E][self.flux_indices[:, 0], self.flux_indices[:, 1]]
12    ux = QX/RHO
13    uy = QY/RHO
14    p = (gamma - 1) * (E_ - .5 * (QX**2 + QY**2)/RHO)
15
16    p_star = p + np.sqrt(gamma * p * RHO) * (ux*self.normal[:, 0]
17        + uy*self.normal[:, 1])
18    phi_n = p_star*self.normal[:, 1]
19
20    mask = self.distance < .5
21    not_mask = np.logical_not(mask)
22
23    self.flux = np.zeros_like(phi_n)
24    self.flux[mask] = phi_n[mask] / lambda
25    self.flux[not_mask] = phi_n[not_mask]
26        / (2 * self.distance[not_mask] * lambda)

```

- **`__get_args`:** returns a dictionary of arguments used by the code generator to produce the boundary condition's numerical code. In this case, unlike previous implementations of this function, the *flux* parameter will also be present (in addition to the already present *istore*, *rhs*, and *dist*)
- **`generate`:** This part of the code generates a C++ code using symbolic manipulation and code generation libraries of `pylbn`. It is appropriate to focus on the last part of the code, i.e. line 46, where the C++ code is generated using the symbolic expressions. Briefly what line 46 of the code below does is to use a for loop, which iterates over the boundary cells and assigns the values of the *rhs* and *flux* of the Bouzidi Bounce-Back with Normal X boundary condition to the cells.

Listing 5.2: Function generate of class `BouzidiBounceBackWithNormalX` in the `boundary.py` script implemented in the `pylbm` library

```

27 def generate(self, order):
28     from pylbm.generator import For
29     from pylbm.symbolic import nx, ny, nz, indexed, ix
30
31     ns = int(self.stencil.nv_ptr[-1])
32     dim = self.stencil.dim
33
34     istore, iload, ncond = self._get_istore_iloading_symb(dim)
35     rhs, dist, _ = self._get_rhs_dist_symb(ncond)
36     flux = IndexedBase('flux', [ncond])
37
38     idx = Idx(ix, (0, ncond))
39     fstore = indexed('f', [ns, nx, ny, nz], index=[istore[idx, k]
40         for k in range(dim + 1)], priority=sorder)
41     fload0 = indexed('fcopy', [ns, nx, ny, nz], index=[iload[0][idx, k]
42         for k in range(dim + 1)], priority=sorder)
43     fload1 = indexed('fcopy', [ns, nx, ny, nz], index=[iload[1][idx, k]
44         for k in range(dim + 1)], priority=sorder)
45
46     self.generator.add_routine(('Bouzidi_bounce_back_normalX',
47         For(idx, Eq(fstore, dist[idx] * fload0 + (1 - dist[idx])
48             * fload1 + rhs[idx] - flux[idx]))))

```

BouzidiBounceBackWithNormalY

Last but not least, let's look at class `BouzidiBounceBackWithNormalY`, which being a subclass of `BouzidiBounceBackWithNormalX`, inherits many functions of the parent class and implements the bounce-back boundary condition on the y-axis. In fact, as was seen in 5.20 relationship, this boundary condition is applied only for momentum; in fact, it will be applied only for the second and third equations of the Euler system to be solved by D_2Q_{4444} vectorial scheme. Inheriting all the functions of `BouzidiBounceBackWithNormalX`, in that case, there are only two specific functions implemented in class `BouzidiBounceBackWithNormalY`:

- **update_flux:** Just as in the previous case in the x-direction, this function calculates the flux term, which is the amount of mass, momentum, and energy transferred through the boundary. So similar to the previous case, the `update_flux` function uses the halo cells to first determine the fluid characteristics (density, velocity, and pressure) at the boundary cells. The contribution from the normal component of velocity is then added to the pressure to determine the pressure at the boundary of cells. Utilizing the pressure and the normal vector, it then determines the flux. The only difference with the implementation in the x-direction is in the definition of the normal, as in the first case

one had defined the normal as `self.normal[:, 0]`, whereas in this case, one has `self.normal[:, 1]`, so the normal component used for the definition of ϕ_n is appropriately different.

- **generate:** Using the symbolic variables defined in the parent class, the `generate` method creates the code for the bounce-back boundary condition, in a way entirely analogous to the previous boundary conditions implemented. The equation for the y-axis is the same as for the x-axis, but the array indices have been changed. In summary, the two generated functions implement the same basic logic but are specialized to apply Bouzidi bounce in different directions, along the x, and y-axes respectively.

Chapter 6

Supersonic Wedge

In this chapter, we finally proceed to examine the numerical results obtained in both the LBMHYE project and the results achieved with the implementation of the new boundary conditions introduced in Section 5.3.

6.1 Theory of oblique shock

In supersonic flow, the oblique shock wave is a typical aerodynamic phenomenon. Through its full understanding and study, it can bring plenty of benefits to many areas of aerospace engineering, just because of its frequency in the supersonic flow [110] [111] [112] [113] [114] [115]. Examples of aerodynamic phenomena that can be improved by studying oblique shock waves are the following:

- Decreasing wave drag.
- Optimizing shock waves of the supersonic inlet in off-design conditions.
- Reducing pressure loss.
- From an aeroacoustic point of view, it is possible to weak the sonic boom of the supersonic vehicle.
- Inducing shock waves in the aero-engine nozzle to obtain thrust vector control and modifying shock wave symmetry to achieve flight control.
- Controlling shock waves of the wave rider ¹.

¹An aeronautical vehicle known as a "wave rider" is one that is designed to benefit from shock waves produced when traveling at supersonic or hypersonic speeds.

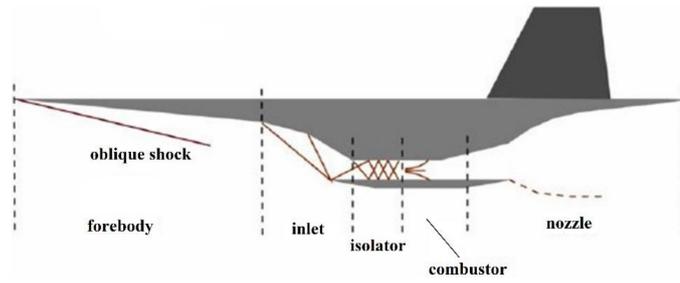


Figure 6.1: Oblique shock waves produced by a scramjet-powered hypersonic aircraft [116]

There are a wide variety of textbooks in the literature dealing with supersonic aerodynamics, those referred to in this brief introduction that is intended to be provided are [117], [118], [119] and [120].

As seen in Figure 6.2, when an object moves at supersonic speed, the flow is not aware of its presence ahead, therefore, the flow is "forced" to compress. The interaction of the supersonic moving object with the surrounding fluid causes the formation of a shock wave in the supersonic flow (i.e. in front of the object, a zone of high pressure known as a "shock wave" develops.). What happens is that the object, moving at a supersonic speed, pushes the fluid to accumulate and compress in a narrow area in front of it. As a direct consequence of this gas accumulation, there is a strong increase in pressure. The shock wave, however, brings not only a discontinuity in pressure, in fact, there is also a bump in temperature, density, and clearly in velocity (not only as an absolute value, but also in the direction).

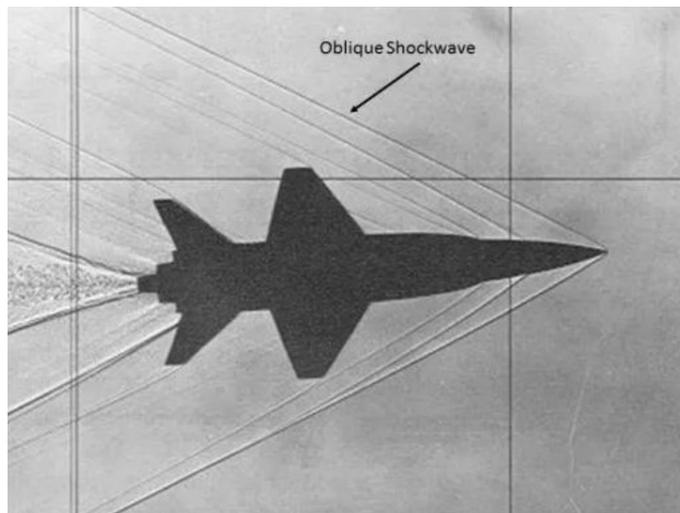


Figure 6.2: Shock waves on X-15 flight vehicle [118]

6.1.1 Oblique shock

Oblique shock refers to a shock wave that is still attached to the body and deviates from the flow direction by any arbitrary angle. The supersonic flow over a concave corner is shown in Figure 6.3 and it is the problem that will be solved in this thesis using LBM. At M_1 , the supersonic flow encounters an upward-facing corner with a turning angle of θ , as the supersonic flow approaches the corner, there will be a sudden compression process and a significant pressure gradient, with the consequent formation of a shock wave attached, emerging from the summit of the concave corner. The downstream Mach number changes to M_2 , and the oblique shock wave angle is β . It is important to note that there are two potential downstream flow

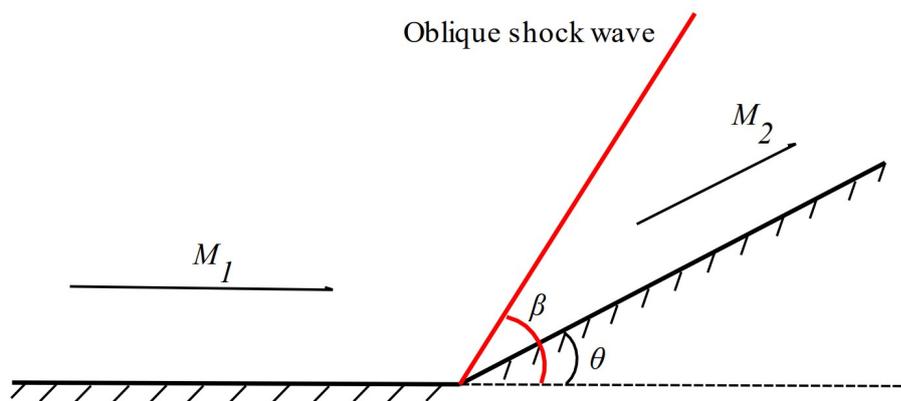


Figure 6.3: Supersonic flow over a wedge

conditions, one brought on by a weak shock and the other by a strong shock, for a given upstream Mach number M_1 and deflection angle (Figure 6.4), and this can also be seen from the relationship 6.2. The downstream pressure affects whether a weak or strong shock solution occurs. In the case of weak shock waves, the wave angle β and entropy rise are reduced. While the downstream Mach number is subsonic in the case of a strong shock, it is supersonic in the case of a weak shock.

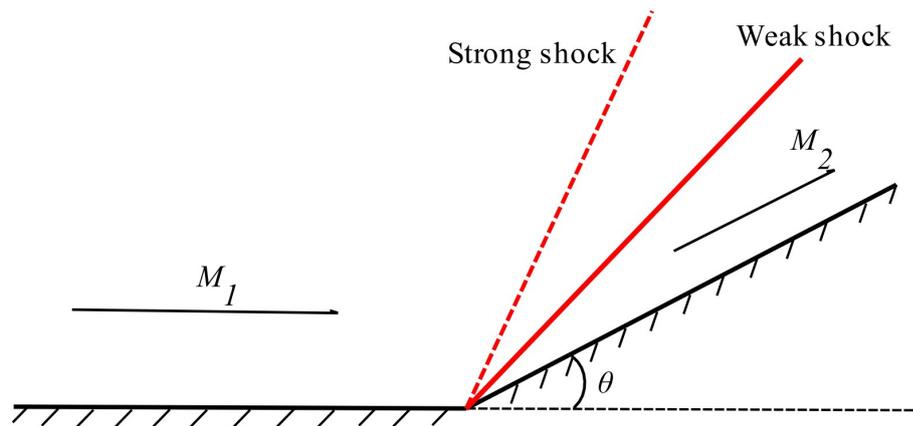


Figure 6.4: Supersonic flow over a wedge - Strong and weak shock waves

6.1.2 Bow shock

There is a maximum flow deflection angle θ_{max} that an oblique shock can handle for a given upstream Mach number M_1 . If the deflection angle θ is greater than θ_{max} , the maximum flow deflection angle, the assumptions concerning the oblique shock then do not succeed, and a curved bow shock generates in front of the body. This bow shock has a curved shape and is detached from the body.

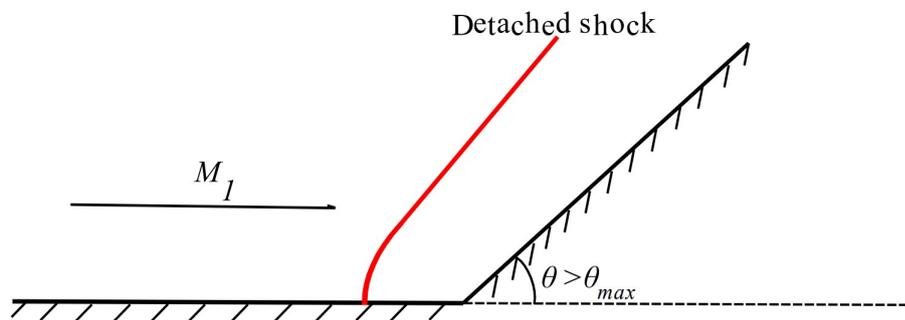


Figure 6.5: Supersonic flow over a wedge - Detached bow shock

Bow shocks are common near blunt bodies [121] [122] [123] for the reason that was just discussed, which is a too-great deflection angle. For example, as will be analyzed in the next chapter 7, in the case of a circular surface hit by a supersonic flow, the deflection angle, locally is 90° , so a detached shock. The flow velocity decreases from supersonic upstream to subsonic downstream as a result of the non-isentropic thermodynamic change across a bow shock. The bow shock causes a vehicle traveling at supersonic speeds to face a significant rise in drag, which is why, in the case of engine intakes, it is necessary to keep the shock attached (i.e.

adopt sharp geometry).

6.1.3 Analytical solution

The relationships for evaluating downstream fluid properties are derived directly from the Rankine-Hugoniot equations [124] [125]. Rankine-Hugoniot relations describe the conditions that develop after a discontinuity. They are derived from the conservation of mass, momentum, and energy. To obtain the Rankine-Hugoniot equations, however, there are some fundamental assumptions; in fact, we consider a regular, one-dimensional flow subject to the Euler equations. These fundamental equations of fluid dynamics are applicable to a normal shock, however, these equations are valid, in the case of an oblique shock, considering the normal component of Mach number to the shock. From these considerations, it is possible to obtain a relation expressing the number of mach downstream of the shock wave:

$$M_2^2 = \frac{1}{\sin^2(\beta - \theta)} \left[\frac{2 + (\gamma - 1)M_1^2 \sin^2 \beta}{2\gamma M_1^2 \sin^2 \beta - (\gamma - 1)} \right] \quad (6.1)$$

The relation 6.1 expresses $M_2 = f(\theta, \beta, M_1)$, where, however, the angle β is still an unknown quantity of the problem.

To overcome this problem, there is a link between the quantities $\theta - \beta - M_1$, which can be derived through geometric considerations, and the known identities from the equations of Rankine-Hugoniot:

$$\tan \theta = \frac{2}{\tan \beta} \left[\frac{M_1^2 (\sin \beta)^2 - 1}{M_1^2 [\gamma + \cos(2\beta)] + 2} \right] \quad (6.2)$$

Hence through the 6.2 relation, it is possible to derive the β angle that the shock wave forms with the upstream direction, so from 6.1 the parameter M_2 can be obtained ².

Plotting the 6.2 relation in a Cartesian plane $\theta - \beta$, using the Mach number M_1 as a parameter, we get the curve shown in Figure 6.6, on which a number of considerations can be made that prove what we have already seen in the previous two paragraphs.

²Then, known M_2 , it can be derived with Rankine-Hugoniot equations all the field downstream of the shock.

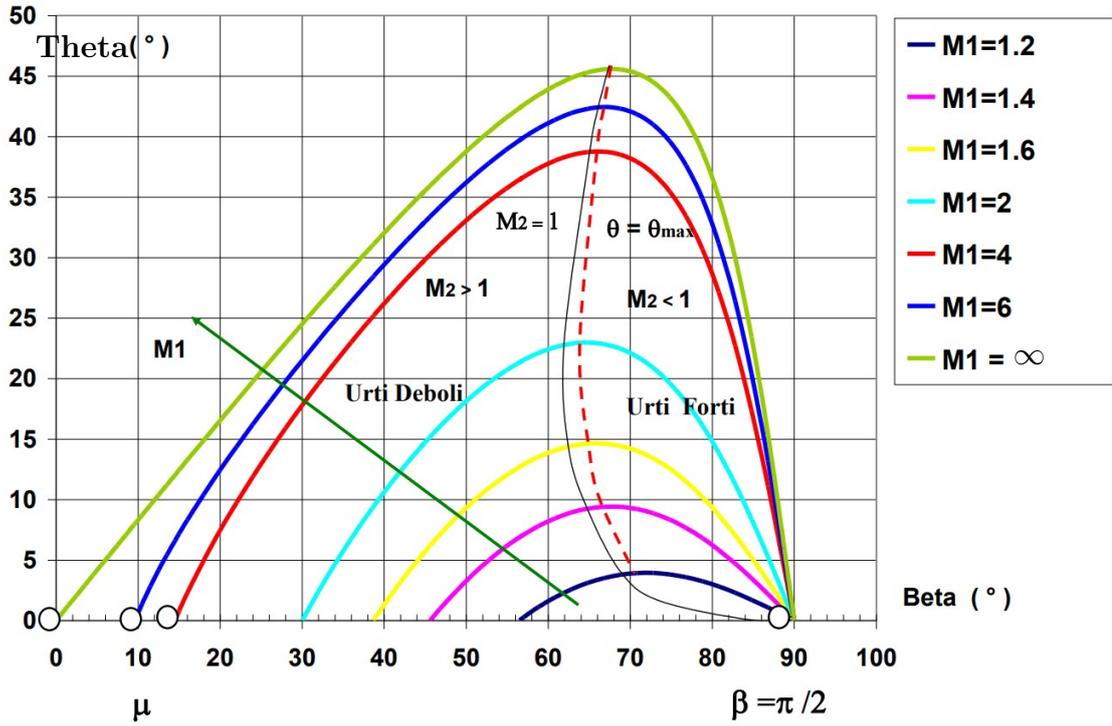


Figure 6.6: $\theta - \beta - Mach$ curve

Once we set a value of M_1 a certain curve in the $\theta - \beta$ plane is defined.

- Each curve has two intersections with the horizontal axis (i.e., zeros of the 6.2 function)
 - For the first intersection, the relationship 6.2 reduces to simple $\frac{1}{\tan(\beta)} = 0$ so it is obtained that $\beta = \frac{\pi}{2}$ (i.e., there is a normal shock) and the flow downstream of the shock is aligned with the flow upstream (i.e. $\theta = 0$).
 - For the second intersection, on the other hand, it follows that:

$$M_1^2(\sin(\beta))^2 = 0 \rightarrow (\sin(\beta))^2 = \frac{1}{M_1^2} \rightarrow \sin(\beta) = \frac{1}{M_1} \rightarrow \beta = \mu$$

So it is considered a Mach line, in fact, the current does not incur any deflection (i.e. $\theta = 0$).

- A maximum θ_{max} of the $\theta - \beta - Mach$ curve is observed, so above this value there is no longer an oblique shock, but rather a detached shock, as we have already seen in the previous section.

- Another concept discussed earlier is the existence of two solutions, namely a weak shock (thus a supersonic Mach M_2 number) in the left part of the curve and a strong shock (thus a subsonic Mach M_2 number) in the right part of the curve.
- In the $\theta - \beta - Mach$ curve, it is possible to identify the two characteristic curves, which are almost overlapping, but separate.
 - The points that identify all the maxima of the various curves (i.e. θ_{max}). In Figure 6.6 marked as a red dotted line.
 - The isomach line $M_2 = 1$, that is, the deflection θ such that there is the sonic condition downstream of the shock. In Figure 6.6 marked as a black line, very close to the previous one.

6.2 Analytical Solution of the Wedge

As mentioned several times within the thesis, the first test case chosen to evaluate the effectiveness of the new boundary conditions is the typical wedge invested by a supersonic flow. The chosen test case has the input Mach number equal to $M_1 =$

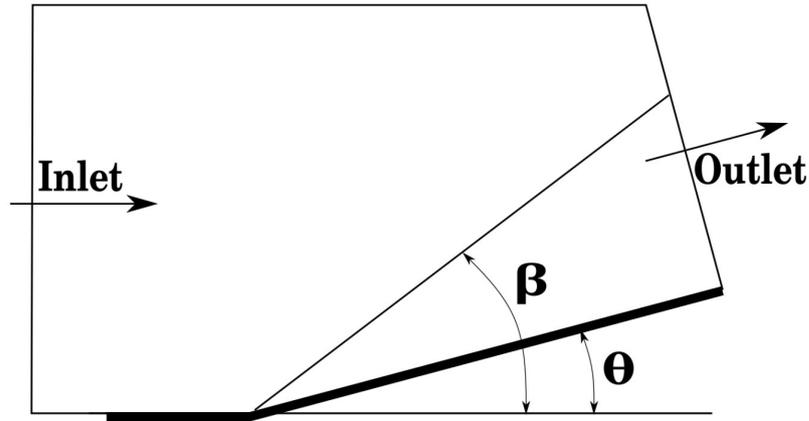


Figure 6.7: Schematic description of the flow domain and angle definitions.

2.5 and the θ angle equal to 15° as key parameters. The other parameters set in the free stream are shown in the table 6.1. The parameters set in 6.1 are set in such a way that the effects of temperature are negligible in the test case, in fact, although it is not physically meaningful, we are treating an isothermal supersonic wedge, so we focus only on the numerical aspect.

$Mach_1$	=	2,5	
p_1	=	1	[Pa]
T_1	=	8,59-05	[K]
Qx_1	=	3,5	[kg/m ² s]
Qy_1	=	0	[kg/m ² s]
ρ_1	=	1,4	[kg/m ³]
Ux_1	=	2,5	[m/s]
Uy_1	=	0	[m/s]
E_1	=	6,875	[J]
$ V _1$	=	2,5	[m/s]
c_1	=	1	[m/s]
p_{TOT1}	=	17.086	[Pa]
$u_{\perp 1}$	=	0.647	[m/s]
$u_{\parallel 1}$	=	2.415	[m/s]

Table 6.1: Wedge Parameters - Free Stream

Once the input parameters have been defined, from the relation 6.2, known θ and M_1 , we derive a β angle equal to $\beta = \mathbf{36.94}^\circ$ under the assumption of weak shock, as it is the most common in nature ³. From a visual point of view, the solution is shown in Figure 6.8.

Knowing the value of the β angle, from the relation 6.1, we obtain a value of Mach number equal to $\mathbf{M_2 = 1.87}$. From the equations of Rankine-Hugoniot, and from the equation of state of gases ⁴the entire field downstream of the shock can be obtained. All the main values of the downstream field are given below and summarized in Table 6.2.

- For the density field we employ the corresponding equation of Rankine-Hugoniot:

$$\rho_2 = \frac{(\gamma + 1)M_1^2 \sin^2 \beta}{(\gamma - 1)M_1^2 \sin^2 \beta + 2} \rho_1 = 2.613[\text{kg/m}^3] \quad (6.3)$$

- For the pressure field we employ the corresponding equation of Rankine-Hugoniot:

$$p_2 = 1 + \frac{2\gamma}{\gamma + 1}(M_1^2 \sin^2 \beta - 1)p_1 = 2.468[\text{Pa}] \quad (6.4)$$

³The strong impact solution would involve an angle $\beta = \mathbf{83.06}^\circ$, but this will not be considered in our case.

⁴In fact 6.6 relationship is proved by the equation of state, in fact, $\frac{T_2}{T_1} = \frac{p_2}{p_1} \frac{\rho_1}{\rho_2}$.

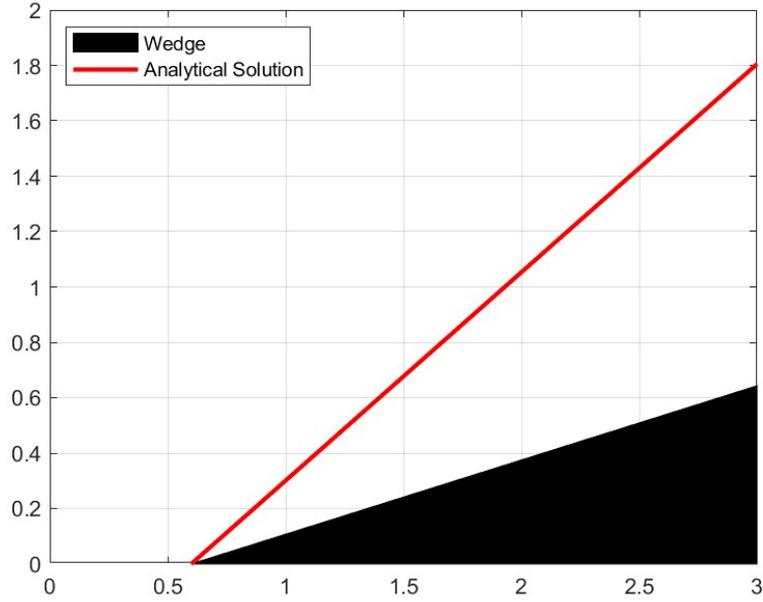


Figure 6.8: Supersonic flow over a wedge - Analytical Solution

- For the temperature field we employ the corresponding equation of Rankine-Hugoniot, proved by the equation of state, in fact, $\frac{T_2}{T_1} = \frac{p_2}{p_1} \frac{\rho_1}{\rho_2}$:

$$T_2 = \left[1 + \frac{2\gamma}{\gamma+1} (M_1^2 \sin^2 \beta - 1) \right] \left[\frac{(\gamma+1)M_1^2 \sin^2 \beta}{(\gamma-1)M_1^2 \sin^2 \beta + 2} \right]^{-1} T_1 \quad (6.5)$$

$$\rightarrow T_2 = 1,14E - 01[K]$$

- For the total pressure field we employ the corresponding equation of Rankine-Hugoniot:

$$\frac{p_{TOT2}}{p_{TOT1}} = \left(1 + \frac{2\gamma}{\gamma+1} (M_1^2 \sin^2 \beta - 1) \right)^{-\frac{1}{\gamma-1}} \left(\frac{(\gamma+1)M_1^2 \sin^2 \beta}{2 + (\gamma-1)M_1^2 \sin^2 \beta} \right)^{\frac{\gamma}{\gamma-1}} \quad (6.6)$$

$$\rightarrow p_{TOT2} = 15.872[Pa]$$

- From the definition of the speed of sound, given the pressure and temperature, it is possible to obtain c_2 of the analytical solution.

$$c_2 = \sqrt{\gamma \frac{p_2}{\rho_2}} = 1.150[m/s] \quad (6.7)$$

- Known the Mach number and the speed of sound, it follows that the absolute value of the speed will be:

$$|V|_2 = M_2 c_2 = 2.154[m/s] \quad (6.8)$$

- Then we evaluate the energy in the field downstream of the shock:

$$E_2 = \frac{1}{2} \rho_2 |V|_2^2 + \frac{p_2}{\gamma - 1} = 8.983[J] \quad (6.9)$$

- We calculate the velocity components of the downstream field:

$$Ux_2 = |V|_2 \cos \theta = 2.081[m/s] \quad (6.10)$$

$$Uy_2 = |V|_2 \sin \theta = 0.558[m/s] \quad (6.11)$$

- Knowing also the components of the velocities, we calculate the momentum in the two directions, since those will be two starting parameters for the LBM, being two of the conserved moments of the scheme used:

$$Qx_2 = Ux_2 \rho_2 = 5.437[kg/m^2s] \quad (6.12)$$

$$Qy_2 = Uy_2 \rho_2 = 1.457[kg/m^2s] \quad (6.13)$$

- Under the imposed assumptions, the flux is adiabatic (i.e., $\left(\frac{T_{o2}}{T_{o1}}\right) = 1$). The entropy drop therefore can be calculated according to the following relationship:

$$\Delta s = -\frac{R}{M} \log \left(\frac{p_{o2}}{p_{o1}} \right) = 21.12[J/K] \quad (6.14)$$

- Finally, we evaluate the normal and tangential components of the velocity since those will be critical in understanding whether the new implementation of BCs is efficient or not:

$$u_{\perp 2} = Ux_2 \sin \theta - Uy_2 \cos \theta = 0[m/s] \quad (6.15)$$

$$u_{\parallel 1} = Ux_2 \cos \theta - Uy_2 \sin \theta = 1.866[m/s] \quad (6.16)$$

$Mach_{2\text{Analytical}}$	=	1.874	
$\beta_{\text{Analytical}}$	=	36.945	[°]
$p_{2\text{Analytical}}$	=	2.468	[Pa]
$T_{2\text{Analytical}}$	=	1,14E-04	[K]
$Qx_{2\text{Analytical}}$	=	5.437	[kg/m ² s]
$Qy_{2\text{Analytical}}$	=	1.457	[kg/m ² s]
$\rho_{2\text{Analytical}}$	=	2.613	[kg/m ³]
$Ux_{2\text{Analytical}}$	=	2.081	[m/s]
$Uy_{2\text{Analytical}}$	=	0.558	[m/s]
$E_{2\text{Analytical}}$	=	12.232	[J]
$ V _{2\text{Analytical}}$	=	2.154	[m/s]
$c_{2\text{Analytical}}$	=	1.150	[m/s]
$PTOT_{2\text{Analytical}}$	=	15.872	[Pa]
$u_{\perp 2\text{Analytical}}$	=	0.000	[m/s]
$u_{\parallel 2\text{Analytical}}$	=	1.866	[m/s]
$\Delta s_{\text{Analytical}}$	=	21.128	[J/K]

Table 6.2: Analytical Solution - Downstream shock field

6.3 Wedge US3D Solution

In order to fully assess the wedge test case, in addition to the analytical solution, a traditional CFD code is used. One of the greatest CFD codes in the world, US3D, created by Graham V. Candler of the University of Tennessee, has been selected for this purpose.

6.3.1 US3D Introduction

Let's very briefly go over the main aspects of this CFD code. It employs unstructured grids and has many sophisticated numerical capabilities and physical models for multi-physics issues, it is based on NASA's well-known DPLR system.

US3D code is written in the Fortran 90 language and is distributed commercially by Candler's VirtusAero <https://virtusaero.com/us3d/>. Being very popular and widespread globally, it is very user-friendly and well-documented. Unstructured grids are utilized by US3D, while the proprietary mesh generation software developed by ICEMCFD is used to create the meshes.

The following are described in great detail in [126]:

- The main capabilities of the code.
- The physical modeling approaches.

- The different types of numerical flux functions.
- Time integration approaches.
- The parallelization strategy.
- A comparisons between US3D and the NASA DPLR code.
- Several simulations illustrate some features of the code.

6.3.2 CFD simulation input parameters

The US3D code is the main comparison used to evaluate the solutions provided by pylbm. In this section, we are going to look at the CFD parameters set in the analysis.

Computational Domain

Since our case is inviscid (i.e. Euler equations), as such, no wall stretching is required. Regarding the choice of mesh, in general, two different approaches can be implemented when comparing two results obtained by different methods:

- Using the best meshes available for comparison, such as adaptive mesh.
- Comparing using the same mesh, or a square lattice as LBM doesn't use a mesh.

The approach that has been taken in the LBMHYPE project is to compare the two solutions, adopting the same grid. This aspect was also chosen because, even with a uniform cartesian grid, mesh generation costs an effort that **LBM** method does not have, in this way, we are going to save the computational cost of the US3D mesh.

For the CFD analysis, a 1m x 1m computational domain was considered, shown in Figure 6.9 as well as the grid. The wedge is 0.5 m long, as is the symmetry plane in front of it.

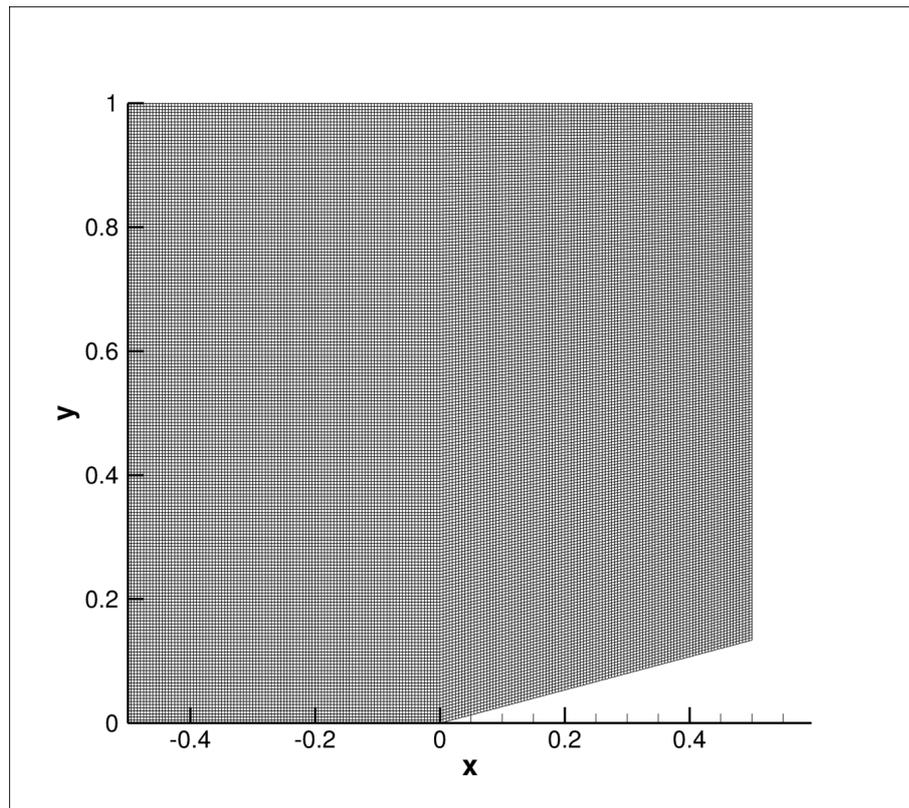


Figure 6.9: 201 x 201 Uniform, Cartesian Computational Grid US3D

Boundary Condition

Let us then look at the various typologies of BCs applied to each face of the domain:

- All of the independent flow variables are fixed at the supersonic inlet. So in this case the CFD boundary condition is the standard characteristic-based boundary condition.
- All variables are extrapolated from the inner domain at the supersonic outflow. Also, in this case, the CFD boundary condition is the standard characteristic-based boundary condition.
- Likewise supersonic outflow is set for the top surface.
- The symmetry boundary condition is set for the symmetry plane in front of the wedge ⁵.

⁵i.e. The velocity component and the normal-to-the-wall derivatives of all the dependent

- The following boundary conditions are imposed on the wall:
 - Since we are solving an inviscid flow (i.e. Euler equations), a slip condition is imposed on the wall, which states that the Dirichlet condition, $\vec{u} = \vec{U}$ on the wall reduces to $\vec{U} \cdot \hat{n} = 0$, the velocity component normal to the wall is zero.
 - On the wall, Neumann's general condition for stresses is also imposed ⁶. Since the shear stress is zero for inviscid flows, the normal derivate of pressure at the wall is zero.
 - Either the Dirichlet-type ($T = T_{wall}$) or Neumann-type ($\nabla T \cdot \hat{n} = g$, where g is a given heat flux function) boundary condition is the standard CFD boundary condition for temperature on the wall. Assuming adiabatic flow in our scenario, $\nabla T \cdot \hat{n} = 0$ on the wall is required ⁷.

These considerations suggest that the boundary conditions for adiabatic flows, which occur in this situation, and the symmetry conditions for inviscid fluids are the same.

CFD numerical scheme

For spatial discretization, the **DPLR** (Data Parallel Line Relaxation) approach with Modified Steger-Warming flux-splitting is used, employing second-order MUSCL without further numerical dissipation on the variables ρ, u, v, p . In the context of the shock wave, not employing any dissipation might not be the ideal strategy, therefore a Ducros dissipation model, which is the default and recommended one, is available by the US3D code.

variables disappear.

⁶ $(\mathcal{S}(\vec{u}) - p\vec{I})\hat{n} = \vec{S}_n$ condition is set, where \vec{u} is the velocity vector, \vec{U} velocity of the wall, \vec{S}_n is a given vector field, \mathcal{S} the shear stress tensor, reduces to $((\mathcal{S}(\vec{u}) - p\vec{I})\hat{n}) \times \hat{n} = 0$ on the wall, as well (i.e. the normal component of the stress tensor vanishes).

⁷Applying the equation for the perfect gas state for temperature, density, and pressure $p = \rho\mathcal{R}T$ follows that the derivative of density normal to the wall is also zero, as the normal derivatives of pressure and temperature.

6.3.3 US3D Wedge Results

The solution obtained by US3D is shown in Figure 6.10 in terms of Mach number. Other images have been reported in Appendix C.

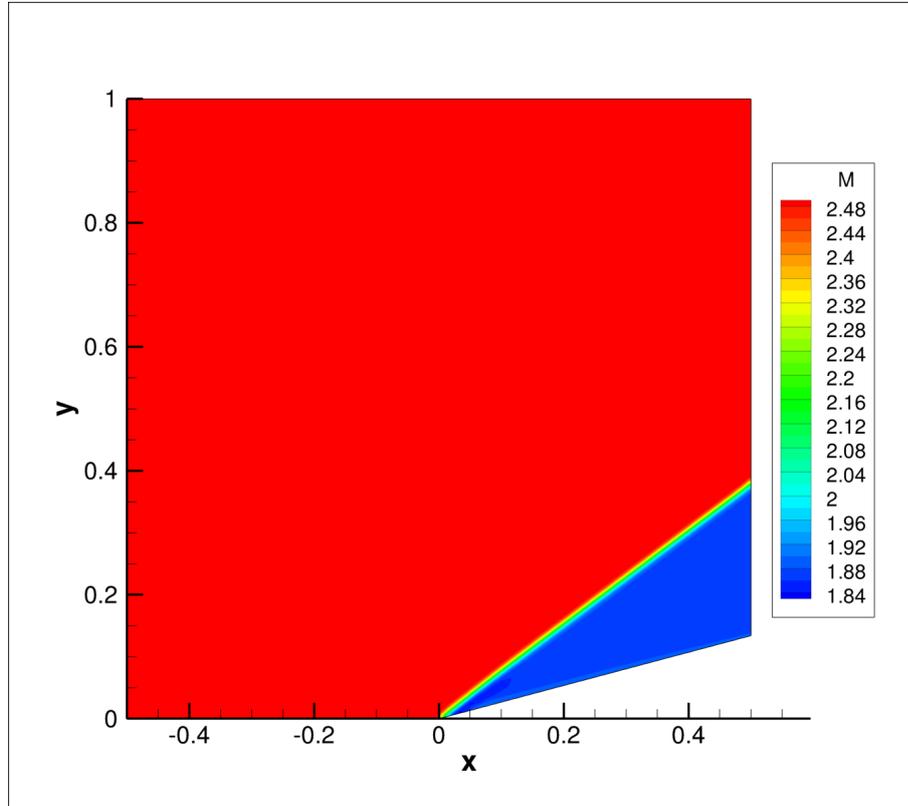


Figure 6.10: CFD Mach number field for the Mach 2.5 supersonic wedge on the 201x201 grid

Numerical approximation of the β angle.

To evaluate the value of the shock wave angle (β), all Mach number values were extracted for given y-coordinates ($y = 0.01$ m, $y = 0.05$ m, $y = 0.15$ m, $y = 0.20$ m, $y = 0.30$ m), then the corresponding x-coordinate at which the discontinuity in Mach number occurs was evaluated. In Figure 6.11, the various discontinuities in Mach number for various y-coordinates imported from the CFD simulation are shown. Already from Figure 6.11, it is possible to visually understand the corresponding x-coordinates.

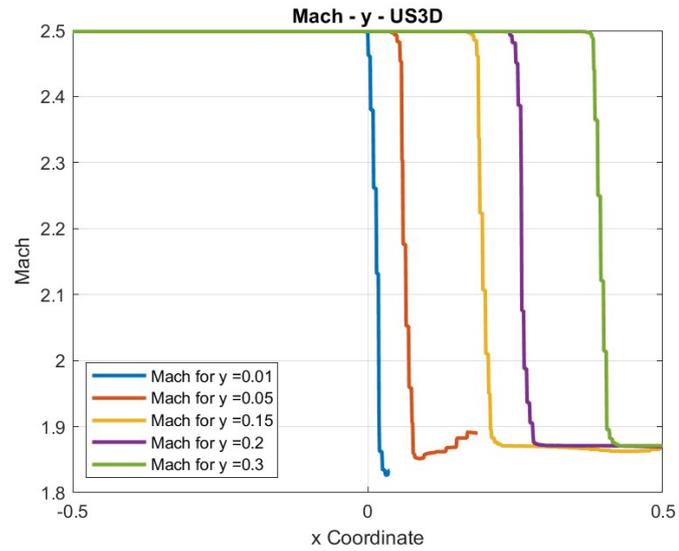


Figure 6.11: Discontinuity of Mach number for various coordinates y

What is obtained are therefore the coordinates $x = 0.012\text{ m}$, $x = 0.063\text{ m}$, $x = 0.201\text{ m}$, $x = 0.264\text{ m}$ and $x = 0.401\text{ m}$. At that step by simple linear interpolation, it is possible to compute the line that best fits the interpolating points, then from the angular coefficient of that line males an angle $\beta = 36.657^\circ$. The numerical approximation of the position of the shock wave is shown in Figure 6.12

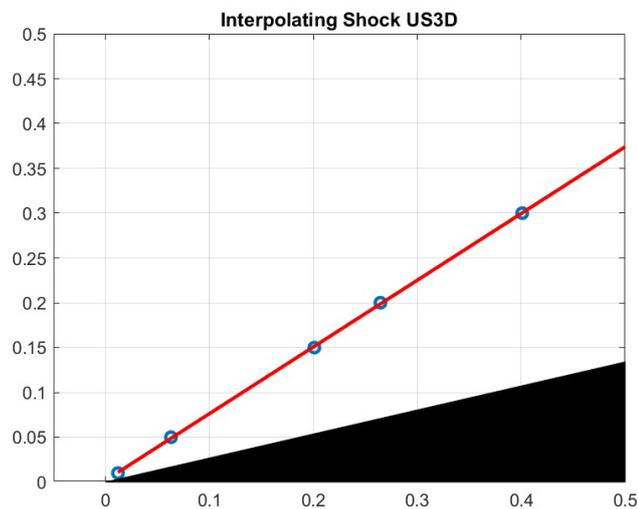


Figure 6.12: Numerical approximation of the shock wave obtained with US3D

Downstream field

To analyze the field downstream of the shock, the values of all fluid dynamics variables were extracted for a given x ($x = 0.5$ m, which corresponds to the outlet in the US3D simulation), from y_{max} , which in the considered domain is equal to 2, to y such that we have the wall ($y = 0.134$ m). Then the various fluid dynamics parameters were graphed. Mach number, pressure, and density have been reported in this section (6.13, 6.14, and 6.15). The remaining parameters (energy, temperature, velocity, sound velocity, and momentum) are given in Appendix C.

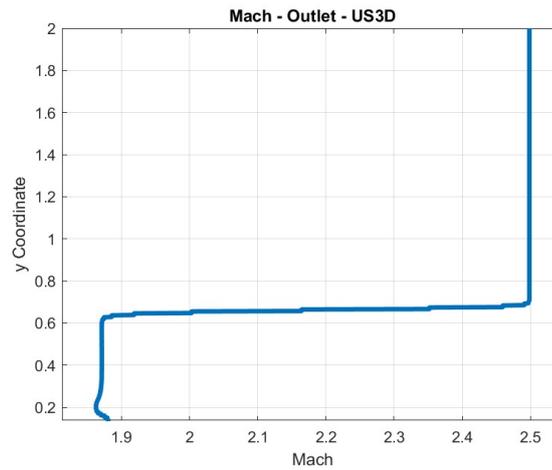


Figure 6.13: Mach values for $x = 0.5$ m - US3D

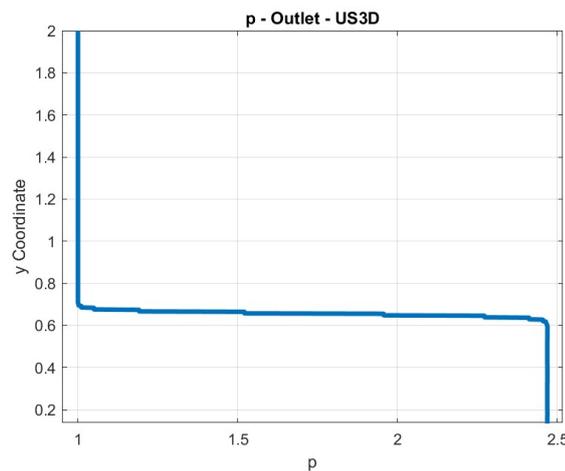


Figure 6.14: Pressure values for $x = 0.5$ m - US3D

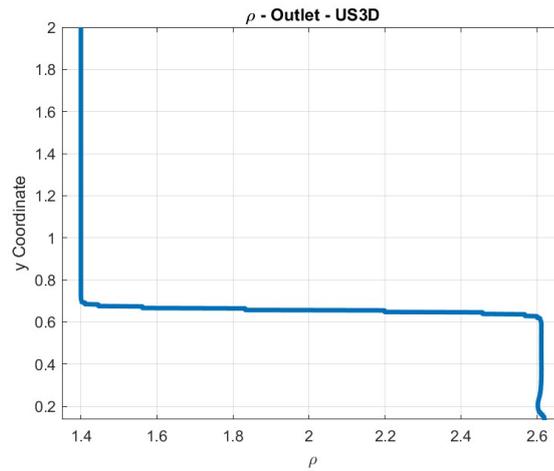


Figure 6.15: Density values for $x = 0.5 \text{ m}$ - US3D

Wall values

Since it will be critical for the new boundary conditions for the LBM to evaluate that indeed the wall velocity is zero. Let us also go on to graph the trend of the Normal Velocity, Tangential Velocity, and Mach at the wall.

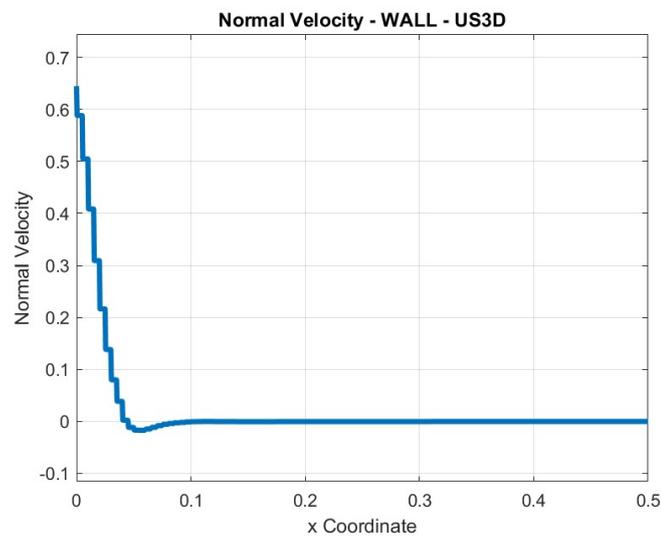


Figure 6.16: Normal wall velocity - US3D

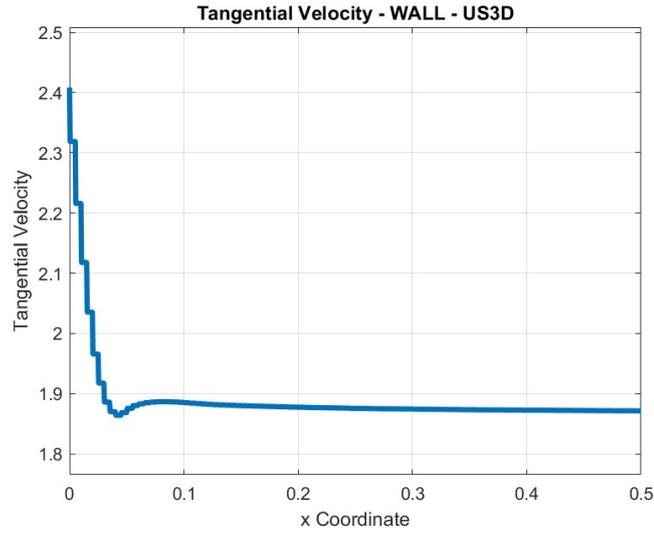


Figure 6.17: Tangential wall velocity - US3D

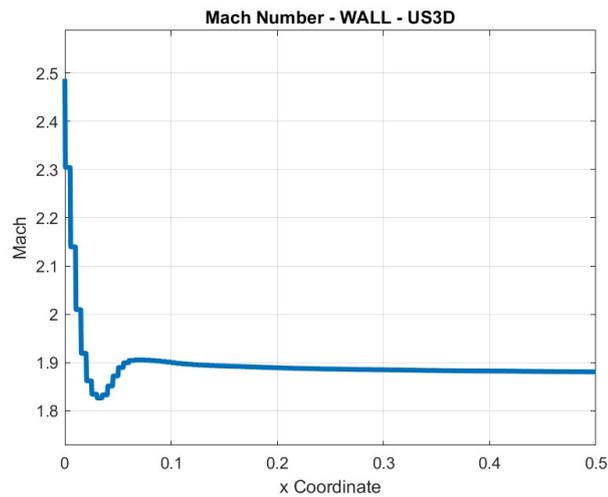


Figure 6.18: Mach wall - US3D

What is observed from Figures 6.16, 6.17 and 6.18, is the trend of u_{\perp} , u_{\parallel} and Mach at the wall, which is constant, as indeed is expected, except at the leading edge of the wedge, due to the proximity of the shock wave. Finally, it is important to note that the slip condition is satisfied, being $u_{\perp} = 0$ at the wall, while the tangential velocity is not. The fluid dynamics parameters downstream of the shock were finally averaged and reported in Table 6.3

$Mach_{2_{US3D}}$	=	1.870	
β_{US3D}	=	36.657	[°]
$p_{2_{US3D}}$	=	2.469	[Pa]
$T_{2_{US3D}}$	=	1,14E-04	[K]
$Qx_{2_{US3D}}$	=	5.428	[kg/m ² s]
$Qy_{2_{US3D}}$	=	1.454	[kg/m ² s]
$\rho_{2_{US3D}}$	=	2.610	[kg/m ³]
$Ux_{2_{US3D}}$	=	2.080	[m/s]
$Uy_{2_{US3D}}$	=	0.557	[m/s]
$E_{2_{US3D}}$	=	12.221	[J]
$ V _{2_{US3D}}$	=	2.153	[m/s]
$c_{2_{US3D}}$	=	1.151	[m/s]
$PTOT_{2_{US3D}}$	=	15.805	[Pa]
$u_{\perp 2_{US3D}}$	=	0.000	[m/s]
$u_{\parallel 2_{US3D}}$	=	1.865	[m/s]
Δs_{US3D}	=	22.348	[J/K]

Table 6.3: US3D Solution - Downstream shock field

US3D Error

What can be seen from this paragraph is that the solution obtained with US3D, despite having used a regular mesh, is extremely accurate. In this regard, in Figure 6.19 the distance that is between the two shock waves (analytical - US3D) is shown, which is of the order of magnitude of millimeters.

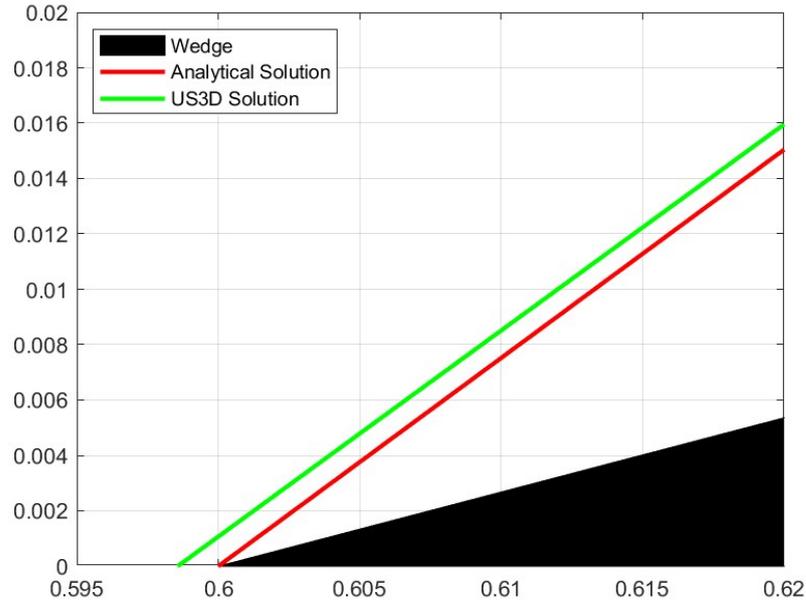


Figure 6.19: Distance between shocks [m]. Solutions: analytical - US3D

Table 6.4 shows the relative errors between the downstream values obtained analytically (i.e., the values given in Table 6.2) and those obtained with US3D (i.e., the values given in Table 6.3). Therefore, it can be noticed that the relative error is always under 1%, emphasizing the accuracy of the method. In Table 6.4, for the values of temperature, and normal speed (both having a null value), the absolute error is reported. Finally, the parameter 6.4 expresses the distance between the leading edge (where from the theoretical point of view the shock wave starts) and the starting point of the shock wave obtained with US3D. This value is tiny, and it is important as in the solution we will see in the next section 6.4.2 (LBM - Bouzidi BCs) it is a relevant value, while in the solution discussed in section 6.4.3 (LBM - Bouzidi Bounce Back with Normal), this parameter returns to millimetric values, emphasizing the efficiency of the new boundary conditions implemented in this thesis.

Error $_{Mach_{US3D}}$	=	0.175	%
Error $_{\beta_{US3D}}$	=	0.778	%
δ_{US3D}	=	1.4368	[mm]
Error $_{p_{US3D}}$	=	0.043	%
Error $_{T_{US3D}}$	=	1,75E-07	[K]
Error $_{Qx_{US3D}}$	=	0.162	%
Error $_{Qy_{US3D}}$	=	0.184	%
Error $_{\rho_{US3D}}$	=	0.111	%
Error $_{u_{US3D}}$	=	0.052	%
Error $_{uy_{US3D}}$	=	0.073	%
Error $_{E_{US3D}}$	=	0.086	%
Error $_{ V _{US3D}}$	=	0.053	%
Error $_{c_{US3D}}$	=	0.077	%
Error $_{p_{TOT}_{US3D}}$	=	0.425	%
Error $_{u_{\perp}_{US3D}}$	=	1,16E-04	[m/s]
Error $_{u_{\parallel}_{US3D}}$	=	0.050	%
Error $_{\Delta s_{US3D}}$	=	5.775	%

Table 6.4: Relative (or absolute) errors between solutions: analytical - US3D

6.4 Wedge LBM Solution

A slightly different domain, than that used for the CFD simulation, was considered in the LBM solution. So let us consider a 3m x 2m domain, in which the wedge is positioned on the long side, at a distance of 0.6 m from the origin of the frame reference system. The input parameters of the LBM simulation are given in Table 6.5, and are the results of a trade-off between stability and accuracy of the simulation. In addition, those are set such that there is an optimal solution with the new boundary conditions.

$s =$	1,6
$\lambda =$	11
$\Delta x =$	0,005
$\sigma =$	0.125
$\Delta t =$	4,545E-04

Table 6.5: Input parameters in the LBM simulation

6.4.1 LBM scheme used for the wedge

The LBM scheme used for the wedge is the D_2Q_{4444} , as described in section 3.4.3. Summarizing what has already been said, D_2Q_{4444} is a vectorial scheme, in which four distinct particle distribution functions are exploited, one for each conserved moment (i.e. the mass, the momentum in x, and y-directions, and the total energy). Each particle distribution function is discretized with four velocities: $(\lambda, 0)$, $(0, \lambda)$, $(-\lambda, 0)$, and $(0, -\lambda)$ where λ is the lattice velocity. This scheme was chosen mainly because of its solidity and robustness, despite its simplicity⁸. However, the structure of the numerical diffusion cannot be modified to fit the physical diffusion operator of Navier-Stokes.

Parameters

From a theoretical point of view there are seven parameters that can be set in the simulation:

- The lattice velocity λ . This first fundamental parameter of LBM is defined as the ratio of space step to time step ($\lambda = \frac{\Delta x}{\Delta t}$). Mainly this velocity must satisfy a CFL-type condition to ensure the stability of the scheme. However, underlining the nonlinear dependence between LBM parameters, λ also plays a role in numerical diffusion (the higher the lattice velocity, the higher the numerical diffusion). Based on this, in choosing the parameter λ it is necessary for it to be the largest of all physical velocities in the problem for stability reasons, but at the same time be as small as possible in order to minimize the numerical diffusion.
- The three relaxation parameters of first-order s_ρ , s_u , and s_p . These three parameters are involved in numerical diffusion (relaxation parameters are linked to numerical diffusion via Henon's relation 3.27), thus to the accuracy of the method⁹. Furthermore, the direct influence of s_ρ , s_u , and s_p is in the relaxation towards equilibrium for the three first-order non-conserved moments of the scheme.

To sum up, increasing the values of the parameters s_ρ , s_u , and s_p decreases the numerical diffusion while decreasing improves the stability, as seen in Section 3.4.

⁸This scheme behaves essentially like the D_1Q_2

⁹From the 3.27 relationships it can be seen that these parameters should take real values between 0 and 2. In the case where the relaxation parameter is equal to 2, there would be a second-order scheme, but highly unstable, especially in correspondence of discontinuities. A study of how to damp such oscillations is presented in [7].

- The three relaxation parameters of second-order $s_{\rho 2}$, $s_{u 2}$, and $s_{p 2}$. Unlike previous relaxation parameters, $s_{\rho 2}$, $s_{u 2}$, and $s_{p 2}$ do not play a role in the second-order numerical diffusion but just in the stability. Clearly, these relaxation parameters are involved in the relaxation towards equilibrium for the three second-order non-conserved moments of the scheme.

A good guideline for setting these parameters can be through the corresponding equations up to the second order, obtained using the expansion of Dubois [79] [80], even if those give only an asymptotic representation of the scheme. These equations are shown in Appendix D.

In order to set the parameters, the main considerations made are as follows:

- the lattice velocity λ has to be large enough to ensure that the second-order operator is positive;
- For each of the relaxation parameters $s \in \{s_{\rho}, s_u, s_p\}$, defining Hennon's parameter as $\sigma = \frac{1}{s} - \frac{1}{2}$, by writing the corresponding equations up to the second order the amplitude of the associated numerical diffusion is given by $\Delta x \lambda \sigma$, thus a function not only of λ and σ but also of the space step.
- The relaxation parameters of first-order should be chosen as close to 2 as possible to minimize the effect of the numerical diffusion while remaining a stable scheme.
- A good choice for the relaxation parameters of second-order is often to take these three parameters equal to the first-order associated relaxation parameters (i.e. $s_{\rho 2} = s_{\rho}$, $s_{u 2} = s_u$, and $s_{p 2} = s_p$).

The parameters chosen are then shown in Table 6.5, so all 6 relaxation parameters were set equal to 1.6 for simplicity, but also to provide the same contribution to numerical diffusion to all four equations. The value of 1.6 is the maximum that could be imposed, considering the other two parameters as well; providing a lower value of σ_{ρ} , for example, would have resulted in a higher numerical diffusion in the corresponding equation, and would also have degraded all the other variables in which density plays a role (such as momentum or the speed of sound, thus indirectly the Mach number). Regarding lattice speed, the value $\lambda = 11$ was set, while it was fixed $\Delta x = 0.005$.

6.4.2 Bouzidi BCs Solution

In this section, we are going to look at the wedge solution by imposing Bouzidi 4.3 conditions on the wedge surface ¹⁰.

Figure 6.20 shows the field of Mach number obtained with the LBM simulation. Already from this figure, we can see the problems faced in the LBMHYPE project. In fact, a detached shock from the leading edge of the wedge is observed. Also, the no-slip condition on the wall had to be applied instead of the slip condition, in fact, it is observed a Mach number that tends to zero globally at the wall (i.e., no-slip condition). From Figures 6.23 - 6.28, the fields of several fluid dynamics variables are shown, in which the same problems found in 6.20 can be observed.

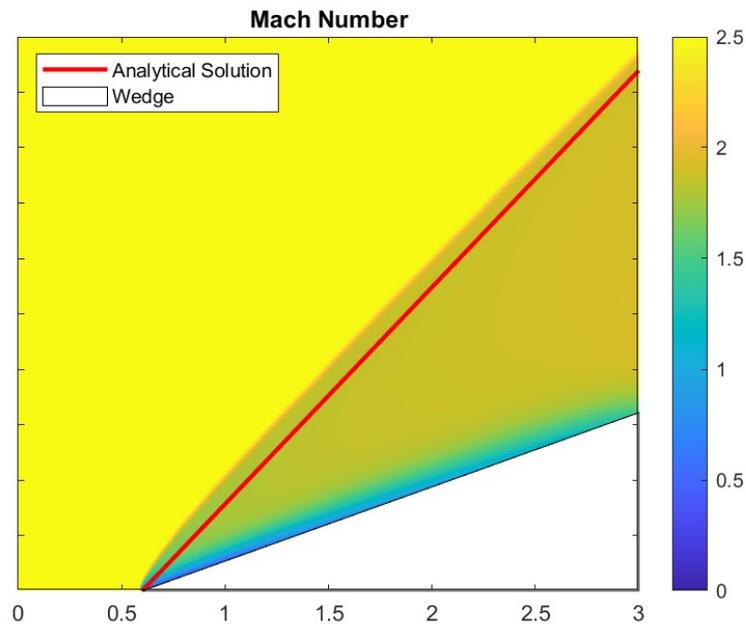


Figure 6.20: Mach Number Field - LBM Bouzidi Solution

A zoom of the leading edge of the wedge is shown in Figure 6.21 in order to show the distance between the theoretical position of the shock wave (i.e., the leading edge itself) and the numerically obtained position. This distance, called δ is equal to 27 mm as listed in the summary table of results 6.7.

To compute the angle β numerically, similar to what was done with the US3D solution, a collection of points where the discontinuity was found were extracted.

¹⁰The solution obtained with simple Bounce-Back 4.2 was not reported as it was superfluous and worse than Bouzidi's solution.

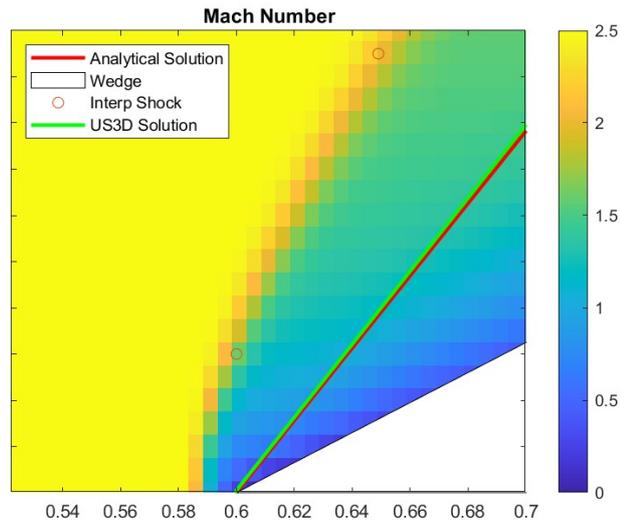


Figure 6.21: Leading Edge - LBM Bouzidi Solution

Then β angle was determined to be 37.8° by interpolation, as shown in Table 6.6. Figure 6.22 shows the three different shock waves considered, namely the one calculated with the LBM, the analytical one, and the one computed with US3D.

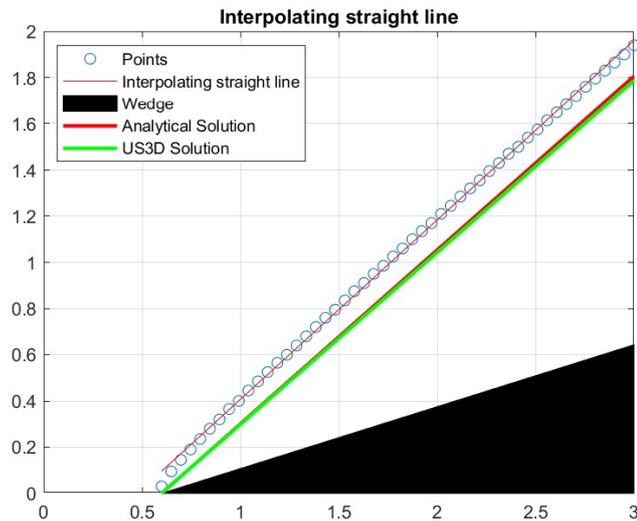


Figure 6.22: The different shock waves considered - LBM (Bouzidi) - Analytical - US3D

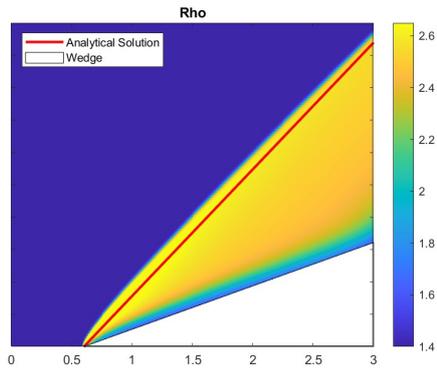


Figure 6.23: Density Field - LBM Bouzidi Solution

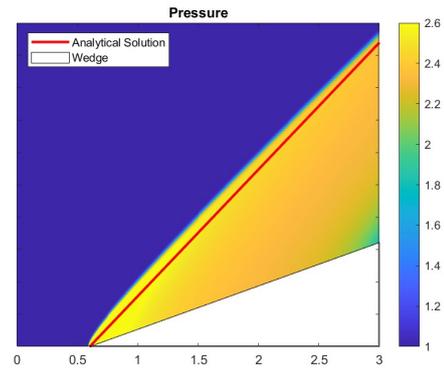


Figure 6.24: Pressure Field - LBM Bouzidi Solution

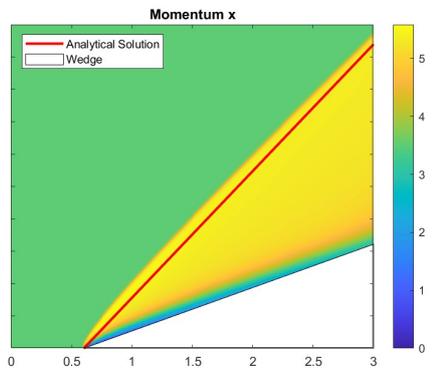


Figure 6.25: Momentum-x Field - LBM Bouzidi Solution

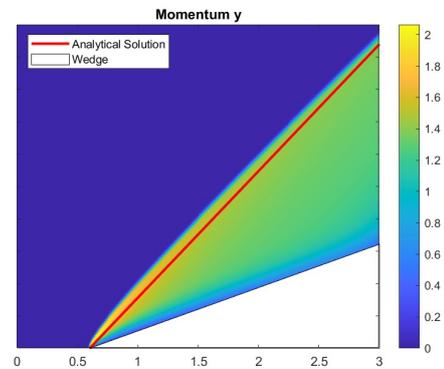


Figure 6.26: Momentum-y Field - LBM Bouzidi Solution

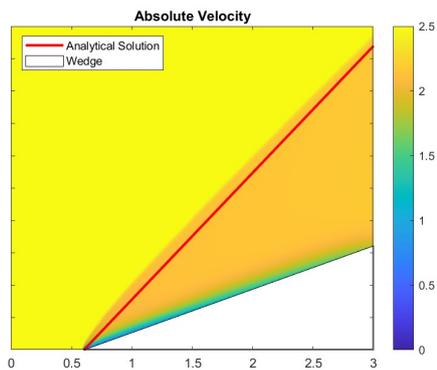


Figure 6.27: $|V|$ Field - LBM Bouzidi Solution

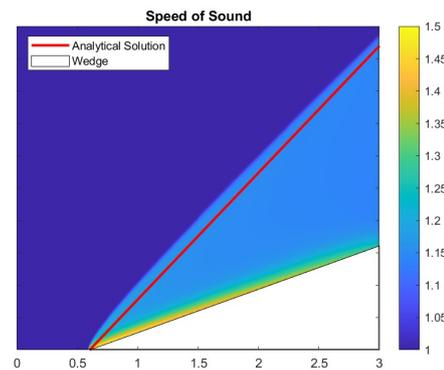


Figure 6.28: Speed of Sound Field - LBM Bouzidi Solution

In order to check what kind of boundary condition is applied, tangential and normal wall velocities are shown in Figure 6.29 and 6.30. However, the best way to

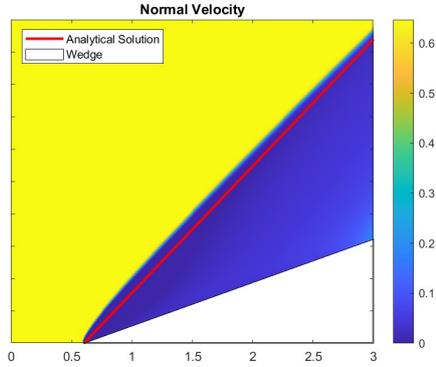


Figure 6.29: Normal Velocity Field - LBM Bouzidi Solution

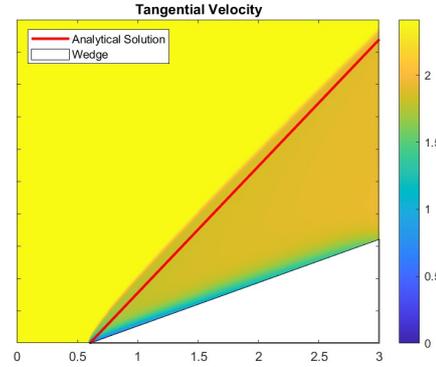


Figure 6.30: Tangential Velocity Field - LBM Bouzidi Solution

verify that the wall velocity tends to zero (both u_{\perp} and u_{\parallel}), we considered a shear section at the outlet of the domain (i.e. $x = 0.5$ m¹¹), and graph the trend of both velocities as the y -coordinate varies, as shown in Figures 6.31 and 6.32. The same

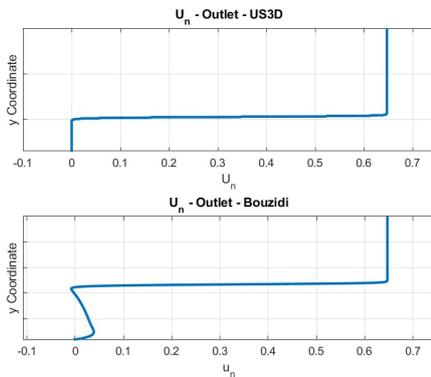


Figure 6.31: Normal Velocity Outlet - LBM Bouzidi Solution

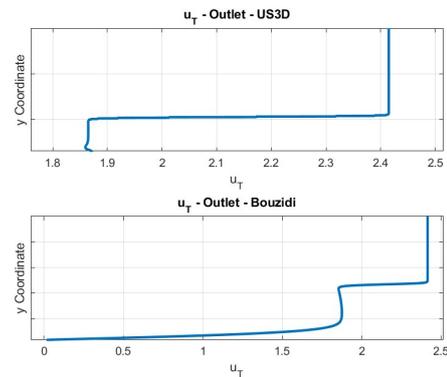


Figure 6.32: Tangential Velocity Outlet - LBM Bouzidi Solution

procedure was carried out for all other fluid dynamics variables, so from Figure 6.33 to 6.38 the trends at the domain outlet (i.e., $x = 0.5$ m) as the y -coordinate changes are shown.

¹¹If we were to consider the LBM simulation, the domain outlet is located for $x = 3$ m, despite that the outlet entry refers to the US3D simulation whose outlet is for $x = 0.5$ m.

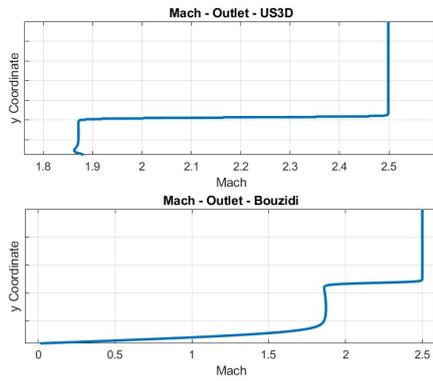


Figure 6.33: Mach Outlet - LBM Bouzidi Solution

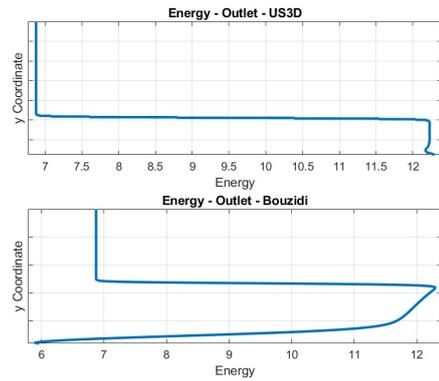


Figure 6.34: Energy Outlet - LBM Bouzidi Solution

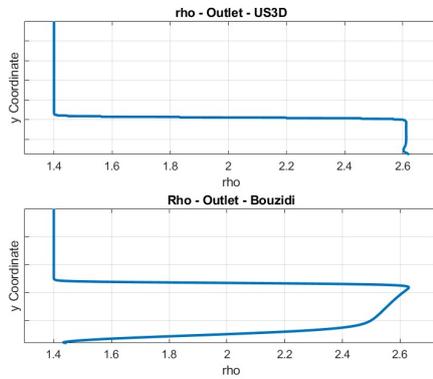


Figure 6.35: Density Outlet - LBM Bouzidi Solution

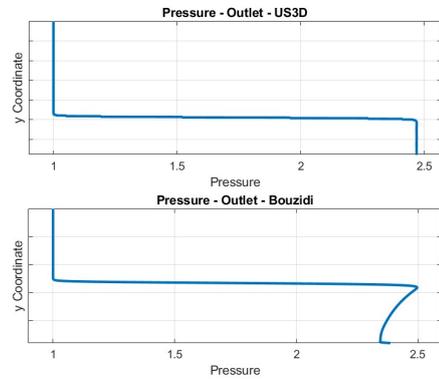


Figure 6.36: Pressure Outlet - LBM Bouzidi Solution

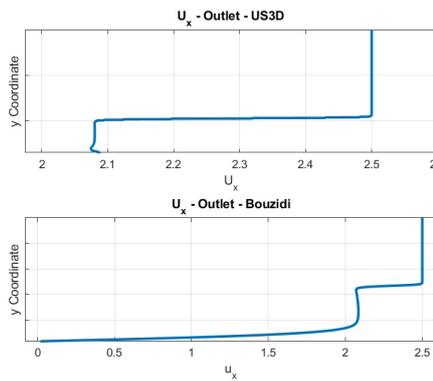


Figure 6.37: Velocity-x Outlet - LBM Bouzidi Solution

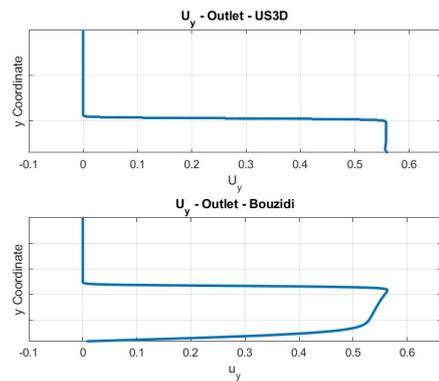


Figure 6.38: Velocity-y Outlet - LBM Bouzidi Solution

In the figures on the previous page, it is clear that the boundary condition is incorrectly applied, as the occurrence of a boundary layer is clear, so the velocity on the wall is not zero in the Euler equations.

Once the values for each y were extracted, the average value for all fluid dynamics parameters was calculated. A sample of values located downstream of the shock wave but far enough from the wedge wall (so as to avoid the boundary layer) was considered in the averaging calculation. The values obtained are shown in Table 6.6.

$Mach_{\text{outlet}}$	=	1.851	
β	=	37.840	[$^{\circ}$]
p_{outlet}	=	2.375	[Pa]
Qx_{outlet}	=	5.163	[kg/m ² s]
Qy_{outlet}	=	1.314	[kg/m ² s]
ρ_{outlet}	=	2.491	[kg/m ³]
Ux_{outlet}	=	2.072	[m/s]
Uy_{outlet}	=	0.527	[m/s]
E_{outlet}	=	11.634	[J]
$ V _{\text{outlet}}$	=	2.138	[m/s]
c_{outlet}	=	1.155	[m/s]
$P_{TOT_{\text{outlet}}}$	=	14.769	[Pa]
$u_{\perp_{\text{outlet}}}$	=	0.027	[m/s]
$u_{\parallel_{\text{outlet}}}$	=	1.865	[m/s]
Δs	=	41.785	[J/K]

Table 6.6: Results - LBM - Bouzidi

Table 6.7 shows the relative percentage errors between the analytical solution and the LBM solution. In contrast, Table 6.8 shows the relative percentage errors between US3D and the LBM solution. It can be seen from both tables that the LBM solution obtained with Bouzidi's conditions is inadequate. The error of greatest magnitude is regarding entropy, whose value calculated with the LBM is completely wrong as already observed in Table 6.6, so the actual value of the relative error was not even reported. The main problem is due to the implementation of the boundary conditions on the sloped wedge wall. The reasons for this inadequacy are due to the geometry considered. The adopted scheme, D_2Q_{4444} , only includes horizontal and vertical velocities. However, since the normal on the wedge wall surface is in the diagonal direction, it is necessary to approximate the contributions to the horizontal and vertical velocities in order to adjust for the slip boundary condition.

Since the slip condition is erroneously implemented (indeed a no-slip condition is applied), there is the occurrence of the boundary layer, which is not expected in the case where Euler equations are being solved. Due to the presence of the boundary layer, the effective body is not a sharp wedge but a streamlined shape, so the shock is detached from the wedge.

As part of the LBMHYPE project, additional simulations were carried out by fine-tuning the computational grid, thus obtaining better results. Clearly, this is not the correct way to follow as there is a limit where the step size can be reduced without numerical problems.

$\text{Error}_{Mach_{LBM}}$	=	1.202	%
$\text{Error}_{\beta_{LBM}}$	=	2.423	%
δ_{LBM}	=	27.684	[mm]
$\text{Error}_{p_{LBM}}$	=	3.756	%
$\text{Error}_{Qx_{LBM}}$	=	5.039	%
$\text{Error}_{Qy_{LBM}}$	=	9.831	%
$\text{Error}_{\rho_{LBM}}$	=	4.664	%
$\text{Error}_{ux_{LBM}}$	=	0.405	%
$\text{Error}_{uy_{LBM}}$	=	5.440	%
$\text{Error}_{E_{LBM}}$	=	4.883	%
$\text{Error}_{ V _{LBM}}$	=	0.734	%
$\text{Error}_{c_{LBM}}$	=	0.481	%
$\text{Error}_{p_{TOT}_{LBM}}$	=	6.952	%
$\text{Error}_{u_{\perp}_{LBM}}$	=	0.027	[m/s]
$\text{Error}_{u_{\parallel}_{LBM}}$	=	0.015	%

Table 6.7: Errors between LBM Bouzidi - Analytic solutions

The LBM approach permits the shock angle to be obtained with a low diffusion value, but the lack of non-aligned slip-wall BC makes the solution less accurate and hides the very great potential of the method.

$\text{Error}_{Mach_{LBM}}$	=	1.029	%
$\text{Error}_{\beta_{LBM}}$	=	3.226	%
$\text{Error}_{p_{LBM}}$	=	3.797	%
$\text{Error}_{Qx_{LBM}}$	=	4.884	%
$\text{Error}_{Qy_{LBM}}$	=	9.665	%
$\text{Error}_{\rho_{LBM}}$	=	4.558	%
$\text{Error}_{ux_{LBM}}$	=	0.353	%
$\text{Error}_{uy_{LBM}}$	=	5.371	%
$\text{Error}_{E_{LBM}}$	=	4.801	%
$\text{Error}_{ V _{LBM}}$	=	0.681	%
$\text{Error}_{c_{LBM}}$	=	0.404	%
$\text{Error}_{p_{TOT}_{LBM}}$	=	6.555	%
$\text{Error}_{u\perp_{LBM}}$	=	0.027	[m/s]
$\text{Error}_{u\parallel_{LBM}}$	=	0.035	%

Table 6.8: Errors between LBM Bouzidi - US3D solutions

6.4.3 Bouzidi Bounce Back with Normal BCs Solution

This section will instead look at the results that are obtained with the improvement introduced in this thesis of Bouzidi BCs 5.3. The same as the previous simulation were set as input parameters 6.5 in order to make a better comparison.

Figure 6.39 shows the Mach number obtained in the domain by applying the new boundary conditions. It is already evident from this image that the main problems encountered with the previous simulation have been solved. In fact, the shock seems to be attached to the leading edge of the wedge, the shock wave fits the analytical solution much better, and lastly, at the wall, there is no Mach = 0 (so the slip condition seems to be correctly applied). The main clearly visible problem is the position of the shock wave, which is lower than in the analytical - US3D case. What can be observed, however, is that the distance increases with length, so it is plausible that this problem is due to the diffusion of the scheme. We must always remember that we are using a first-order accuracy scheme for Euler's equations¹². In addition, a slight wall proximity effect can be observed.

Focusing initially on the positive aspects of the obtained solution, a zoom of the leading edge is shown in Figure 6.40. In this case, the estimated distance between the LE and the position of the shock wave is 1.8 mm, thus comparable with the 1.4

¹²From a purely theoretical point of view, as mentioned earlier, the adopted scheme has an accuracy between the first and second order. However, the second order is achieved only at $s=2$, for which, however, although it has minimal diffusion, the solution is unstable.

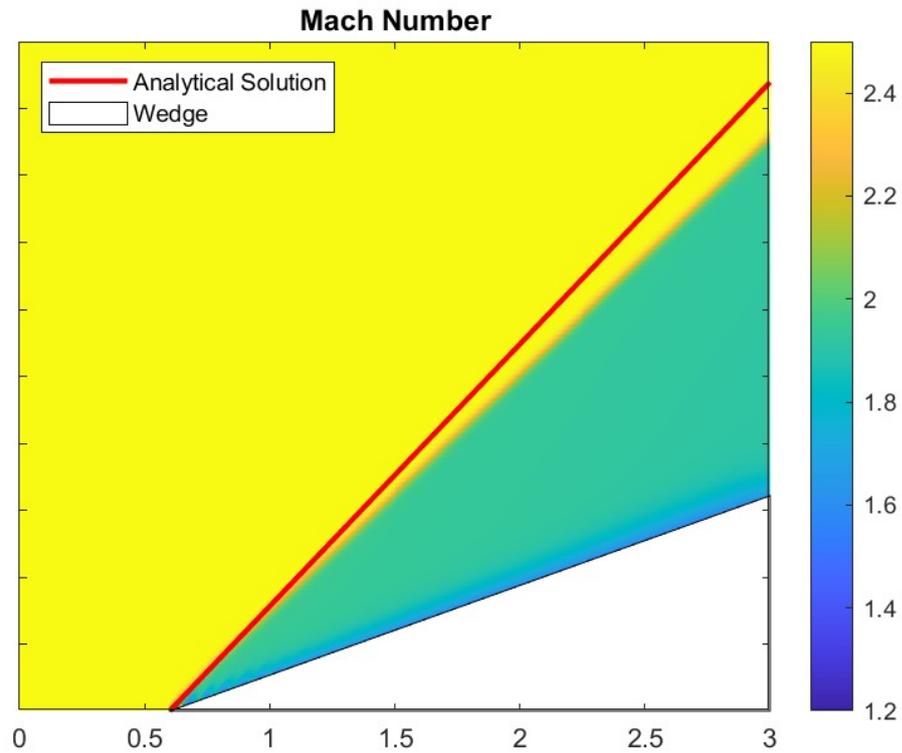


Figure 6.39: Mach Number Field - LBM Bouzidi Bounce Back with Normal Solution

mm obtained with US3D. The fact that the error is of the same order of magnitude as obtained with US3D is already something sensational, especially considering that in the previous solution, a δ of 27 mm was obtained. This type of error is unavoidable; in fact, it is also present in US3D, since to discretize an oblique surface, no matter how fine is the discretization adopted, a series of steps are used. Therefore, in a supersonic flow, even a small step size is an important perturbation, that is why a curved shock is observed near the leading edge. In that case, however, the distance between the leading edge and the shock is very small.

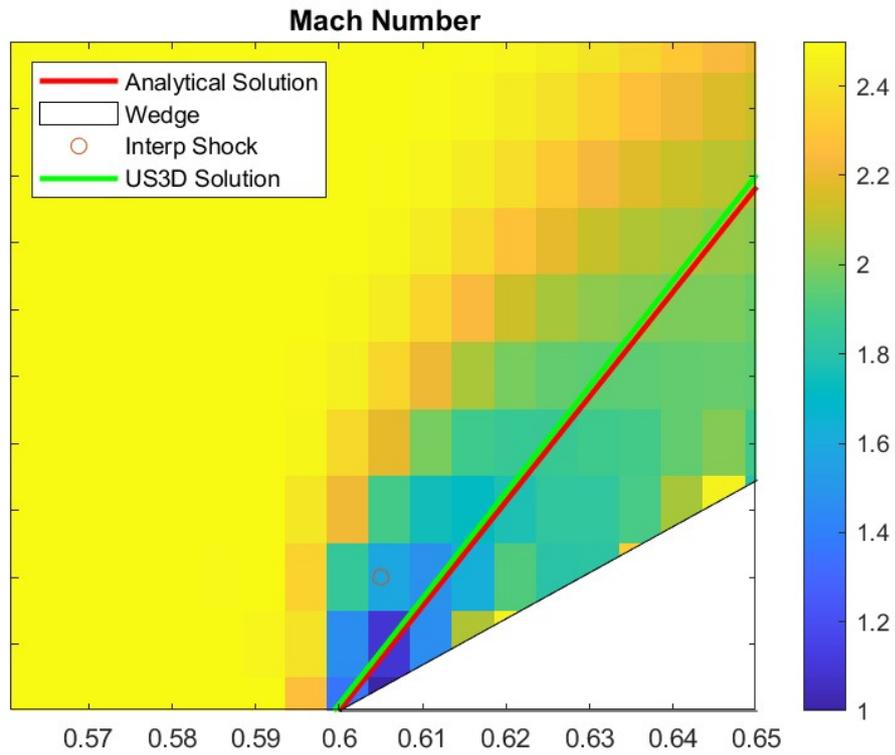


Figure 6.40: Leading Edge - LBM Bouzidi Bounce Back with Normal Solution

To compute the angle β numerically, similar to what was done with the US3D and Bouzidi's solutions, a collection of points where the discontinuity was found was extracted. Then, from the interpolating line an angle β equal to 36.9° was determined, as shown in Table 6.9. Figure 6.41 shows the three different shock waves considered, namely the one calculated with the LBM, the analytical one, and the one computed with US3D. In this case, however, the solution obtained with the LBM is much more accurate than previously obtained, in Figure 6.22 the distance between the analytical solution and the LBM solution is much greater than in Figure 6.41.

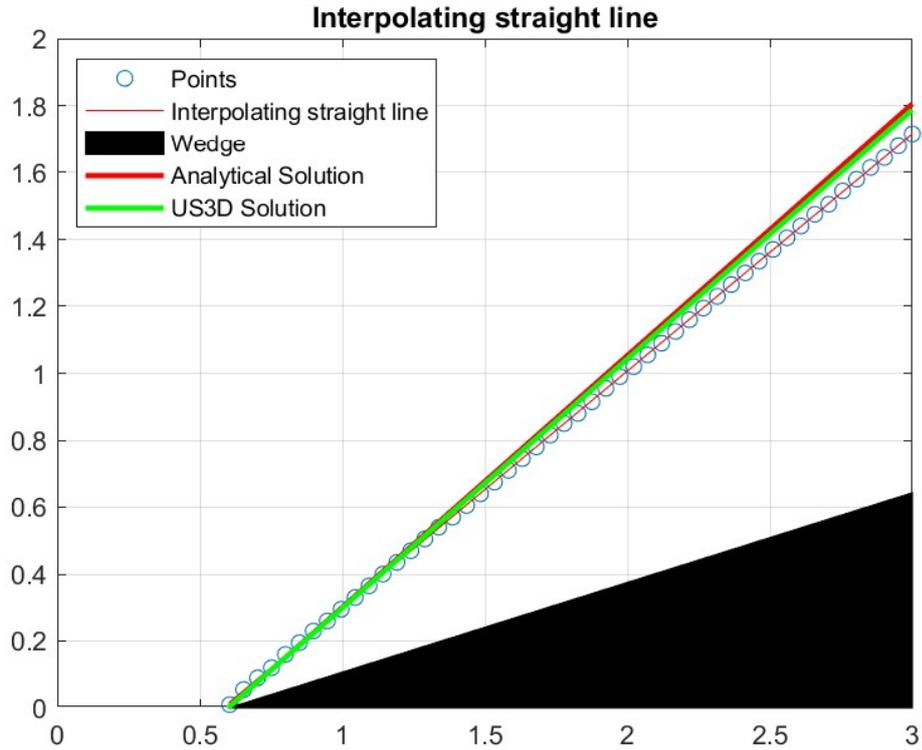


Figure 6.41: The different shock waves considered - LBM (Bouzidi with Normal)
- Analytical - US3D

From Figures 6.42 - 6.47, the fields of several fluid dynamics variables are shown, in which the same pros and cons found in 6.39 can be observed. In fact, if on the one hand, we observe a shock wave that is attached, straight, and fits the other two solutions more accurately than the previous case. On the other hand, there is a slight influence of the wall on the solution; in fact, although the slip condition is correctly applied, a slight numerical dissipation phenomenon is observed at the wall. If the parameter λ were increased (i.e. increasing numerical diffusion), this effect would be much more pronounced, as will be analyzed in section 6.4.3. However, by setting the λ parameter small enough, this effect alleviates.

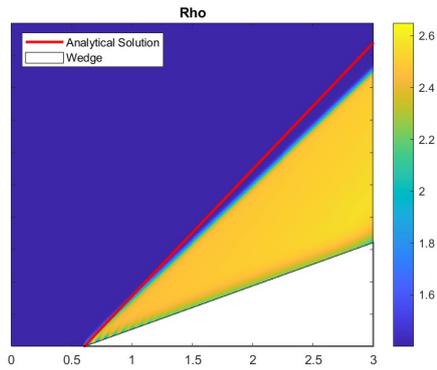


Figure 6.42: Density Field - LBM Bouzidi with Normal Solution

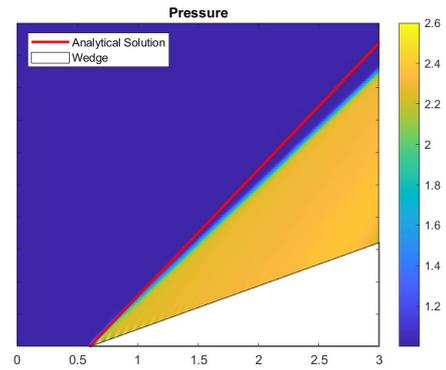


Figure 6.43: Pressure Field - LBM Bouzidi with Normal Solution

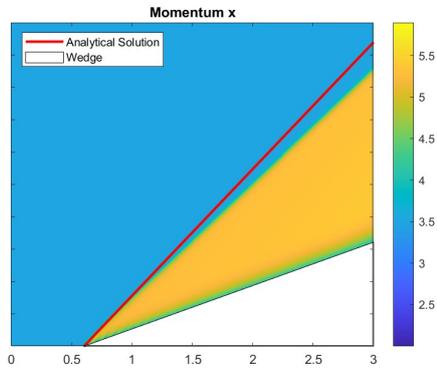


Figure 6.44: Momentum-x Field - LBM Bouzidi with Normal Solution

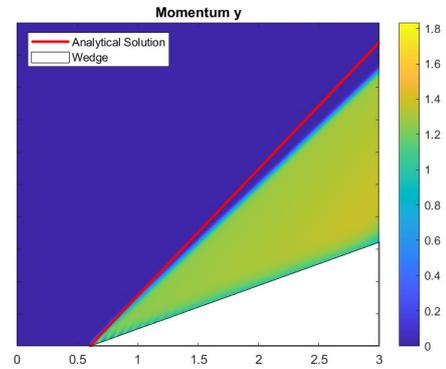


Figure 6.45: Momentum-y Field - LBM Bouzidi with Normal Solution

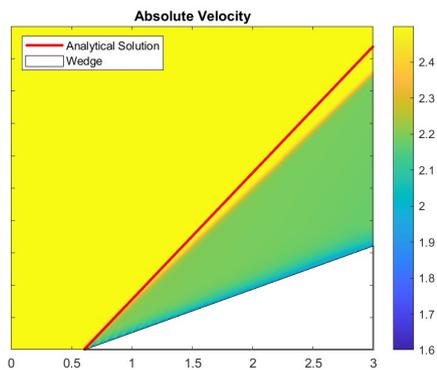


Figure 6.46: $|V|$ Field - LBM Bouzidi with Normal Solution

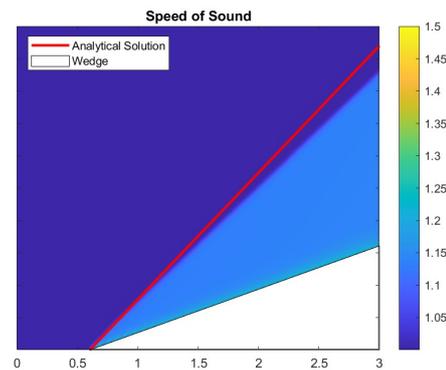


Figure 6.47: Speed of Sound Field - LBM Bouzidi with Normal Solution

Also in this *Bouzidi Bounce Back with Normal* boundary condition analysis, we check the values of normal and tangential wall velocity, in order to check what kind of boundary condition is applied. Therefore tangential and normal wall velocities are shown in Figure 6.48 and 6.49. However, the best way to verify that only the

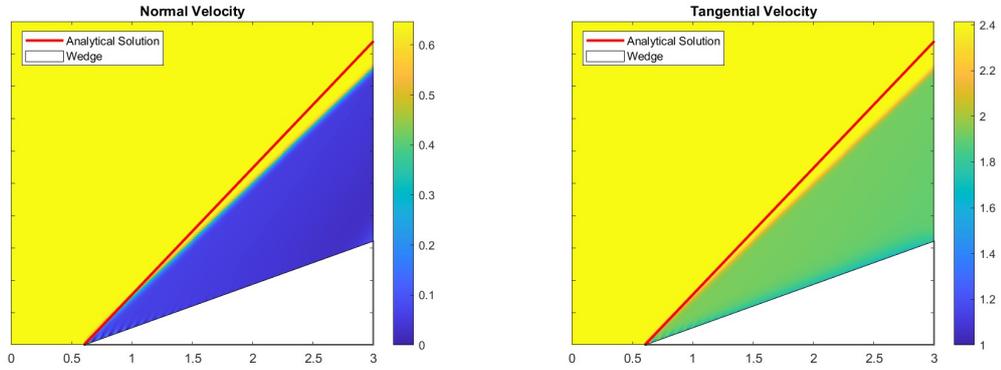


Figure 6.48: Normal Velocity Field - **Figure 6.49:** Tangential Velocity Field
 LBM Bouzidi with Normal Solution - LBM Bouzidi with Normal Solution

wall velocity u_{\perp} tends to zero, we considered a shear section at the outlet of the domain (i.e. $x = 0.5$ m, for the same consideration made in the previous LBM analysis is valid. In fact, the outlet entry refers to the US3D simulation whose outlet is for $x = 0.5$ m.), and graph the trend of both velocities as the y -coordinate varies, as shown in Figures 6.50 and 6.51. Unlike the previous section, while the normal velocity is zero, the tangential velocity is not, so the slip condition is applied successfully in this case. Thus, the Bouzidi Bounce Back with Normal BCs met expectations by correctly imposing the wall boundary condition.

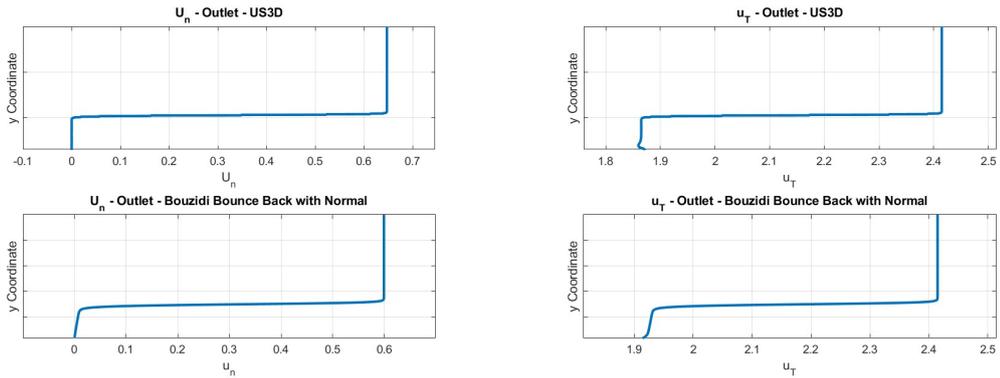


Figure 6.50: Normal Velocity Outlet - **Figure 6.51:** Tangential Velocity Out-
 let - LBM Bouzidi with Normal Solution

The same procedure was carried out for all other fluid dynamics variables, so from Figure 6.52 to 6.57 the trends at the domain outlet (i.e. $x = 0.5$ m) as a function of the y-coordinate is shown. The considerations that can be made reflect what has been seen in Figures 6.39 and from 6.42 to 6.49. In fact, it is evident that the shock is captured perfectly in a few cells, similar to the trend found with US3D. This accuracy is due to the fact that the relaxation parameter was specifically chosen to have a low numerical diffusion. The obvious problem is the numerical dispersion of values downstream of the shock wave. From an analytical point of view, one would need to find constant values of all fluid dynamics properties downstream of the discontinuity, as is the case with CFD simulation. What is observed in the graphs for the LBM simulations, however, is that the value of the fluid dynamics parameters are not perfectly constant downstream of the shock.

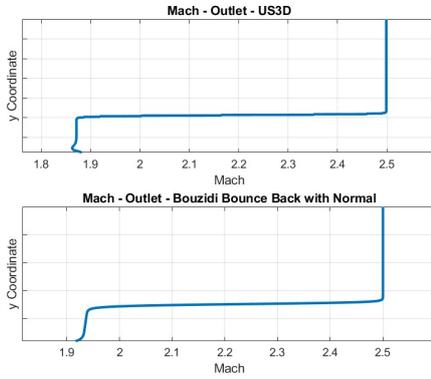


Figure 6.52: Mach Outlet - LBM Bouzidi with Normal Solution

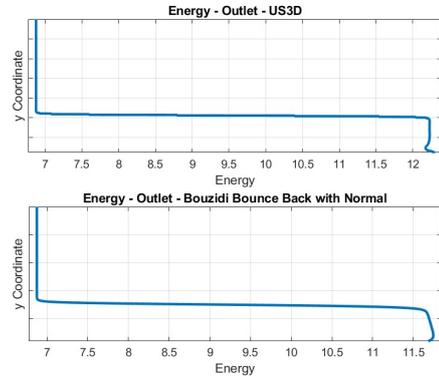


Figure 6.53: Energy Outlet - LBM Bouzidi with Normal Solution

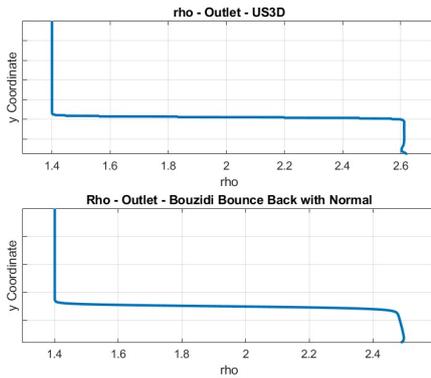


Figure 6.54: Density Outlet - LBM Bouzidi with Normal Solution

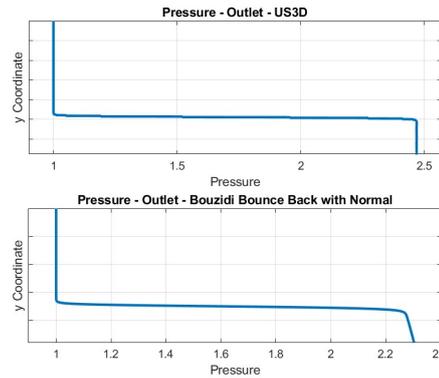


Figure 6.55: Pressure Outlet - LBM Bouzidi with Normal Solution

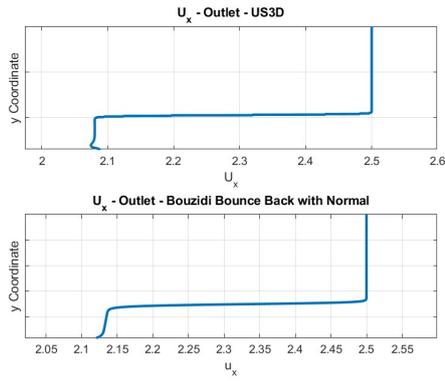


Figure 6.56: Velocity-x Outlet - LBM Bouzidi with Normal Solution

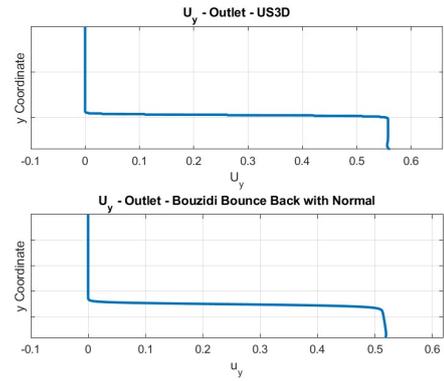


Figure 6.57: Velocity-y Outlet - LBM Bouzidi with Normal Solution

The main problem in the pictures from 6.42 to 6.49 is the imperfect match between the downstream shock values obtained with US3D and the LBM values. This error is more pronounced in some variables than in others, as will be discussed in Tables 6.10 and 6.11. The cause of the error is the first-order accuracy of the adopted vectorial scheme, as well as the ever-present and necessary diffusion in the LBM.

Before evaluating how to alleviate this undesirable effect, by considering values downstream of the shock but not too close to the wall, arithmetic averages of all fluid dynamics parameters are given in Table 6.9.

$Mach_{\text{outlet}}$	=	1.936	
β	=	35.926	[°]
p_{outlet}	=	2.280	[Pa]
Qx_{outlet}	=	5.298	[kg/m ² s]
Qy_{outlet}	=	1.277	[kg/m ² s]
ρ_{outlet}	=	2.482	[kg/m ³]
Ux_{outlet}	=	2.134	[m/s]
Uy_{outlet}	=	0.514	[m/s]
E_{outlet}	=	11.681	[J]
$ V _{\text{outlet}}$	=	2.195	[m/s]
c_{outlet}	=	1.134	[m/s]
$P_{TOT_{\text{outlet}}}$	=	16.151	[Pa]
$u_{\perp_{\text{outlet}}}$	=	0.008	[m/s]
$u_{\parallel_{\text{outlet}}}$	=	1.928	[m/s]
Δs	=	15.812	[J/K]

Table 6.9: Results - LBM - Bouzidi Bounce Back with Normal

Similar to the previous case where Bouzidi conditions were used, these values were compared to the analytical solution (Table 6.10), and those obtained with US3D (Table 6.11). Hence, with the new boundary conditions, the error in calculating the β angle is 3.8% (compared to the 2.4% obtained previously) a value that is much further than the 0.7% error obtained with US3D. This increase in error from the previous case should be carefully observed. At first glance, it would seem to have worsened this aspect. However, this problem in the shock calculation can be attributed to the method and not to the boundary conditions. Underestimating the value of β is something plausible using a method that adopts diffusion. In the previous case, however, an overestimated value of the angle formed by the shock wave was obtained. Personally, I think the presence of the boundary layer in the solution obtained with Bouzidi's boundary conditions altered this measurement. The presence of the boundary layer causes the body invested by the upstream flux to have a higher angle than 15° (i.e., it is as if the wedge has a few more degrees due to the presence of the boundary layer). This alteration in geometry causes the beta angle to be greater from a fluid dynamics point of view, so this effect goes against the underestimation of the angle typical of LBM.

Moving on to the pros of this result, the best result is the distance obtained between the leading edge and the position of the shock wave. From an analytical point of view this distance is zero, but even US3D, for the reasons already seen above, estimates a distance of about 1.4 mm. The previous LBM solution estimated a distance of 27 mm, i.e., a bow shock, which although a physically possible

solution, is not the correct one for the imposed initial conditions. In the current LBM solution, on the other hand, 1.8 mm is estimated, so an error of the same order of magnitude as US3D, and this is remarkable for a method that is only first-order accurate.

Regarding the macroscopic properties of the fluid, while there are fairly small errors, even close to 1%, there are some fluid-dynamics properties that have rather high errors as can be seen in Tables 6.10 and 6.11.

$\text{Error}_{Mach_{LBM}}$	=	3.337	%
$\text{Error}_{\beta_{LBM}}$	=	3.806	%
δ_{LBM}	=	1.814	[mm]
$\text{Error}_{p_{LBM}}$	=	7.607	%
$\text{Error}_{\rho_{LBM}}$	=	5.006	%
$\text{Error}_{E_{LBM}}$	=	4.504	%
$\text{Error}_{ V _{LBM}}$	=	1.910	%
$\text{Error}_{c_{LBM}}$	=	1.380	%
$\text{Error}_{p_{TOT_{LBM}}}$	=	1.758	%
$\text{Error}_{u_{\perp_{LBM}}}$	=	0.008	[m/s]
$\text{Error}_{u_{ _{LBM}}}$	=	3.368	%
$\text{Error}_{\Delta s_{LBM}}$	=	25.161	%

Table 6.10: Errors between LBM Bouzidi Bounce Back with Normal - Analytic solutions

$\text{Error}_{Mach_{LBM}}$	=	3.519	%
$\text{Error}_{\beta_{LBM}}$	=	3.631	%
$\text{Error}_{p_{LBM}}$	=	7.646	%
$\text{Error}_{\rho_{LBM}}$	=	4.900	%
$\text{Error}_{E_{LBM}}$	=	4.421	%
$\text{Error}_{ V _{LBM}}$	=	1.964	%
$\text{Error}_{c_{LBM}}$	=	1.456	%
$\text{Error}_{p_{TOT_{LBM}}}$	=	2.192	%
$\text{Error}_{u_{\perp_{LBM}}}$	=	0.008	[m/s]
$\text{Error}_{u_{ _{LBM}}}$	=	3.420	%

Table 6.11: Errors between LBM Bouzidi Bounce Back with Normal - US3D solutions

In this regard, instead of analyzing absolute quantities and evaluating the corresponding relative percentage errors, ratios of quantities upstream and downstream of the shock are often used in the supersonic field. Therefore, $\frac{Mach_2}{Mach_1}$, $\frac{E_2}{E_1}$, $\frac{\rho_2}{\rho_1}$ and $\frac{p_2}{p_1}$ ratios are represented in figures 6.58 to 6.61, and also in this case, the graphs are compared against the respective ratios obtained with US3D. Clearly, the error that occurred in in Tables 6.10 and 6.11 is also reflected in the graphs below, however, such graphs are more general, being dimensionless.

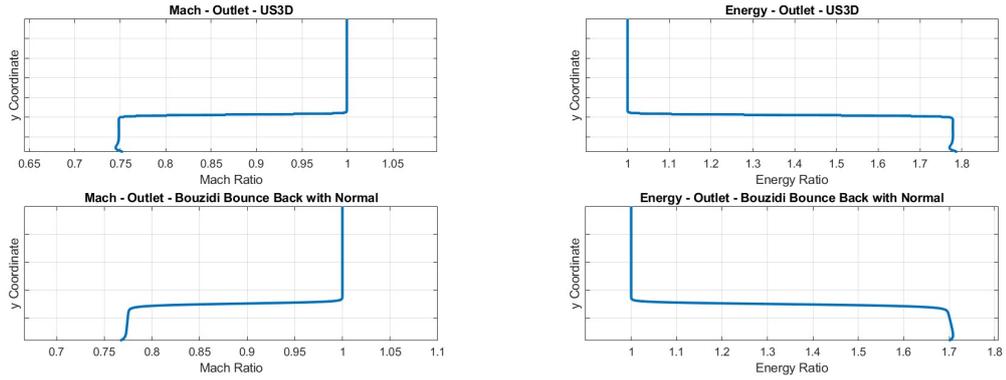


Figure 6.58: Mach Ratio Outlet - **Figure 6.59:** Energy Ratio Outlet -
LBM Bouzidi with Normal Solution LBM Bouzidi with Normal Solution

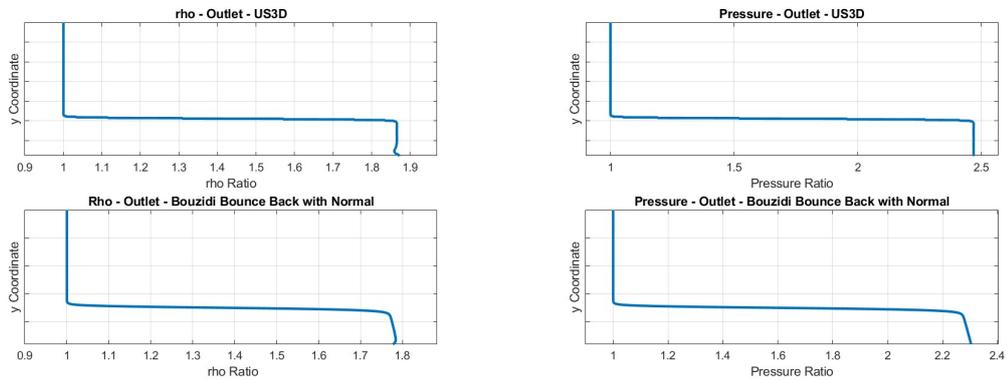


Figure 6.60: Density Ratio Outlet - **Figure 6.61:** Pressure Ratio Outlet -
LBM Bouzidi with Normal Solution LBM Bouzidi with Normal Solution

In Tables 6.12 and 6.13 absolute errors between the ratios of fluid dynamics quantities obtained with LBM are shown, compared with US3D and the analytical solution, respectively. Calculating the respective relative errors yields values around 2%.

$\text{Error}_{Mach_{LBM}}$	=	0.025
$\text{Error}_{p_{LBM}}$	=	0.188
$\text{Error}_{\rho_{LBM}}$	=	0.093
$\text{Error}_{E_{LBM}}$	=	0.080
$\text{Error}_{ V _{LBM}}$	=	0.016
$\text{Error}_{c_{LBM}}$	=	0.016
$\text{Error}_{\text{PTOT}_{LBM}}$	=	0.016

Table 6.12: Absolute errors in ratios between LBM Bouzidi Bounce Back with Normal - Analytic solutions

$\text{Error}_{Mach_{LBM}}$	=	0.026
$\text{Error}_{p_{LBM}}$	=	0.189
$\text{Error}_{\rho_{LBM}}$	=	0.091
$\text{Error}_{E_{LBM}}$	=	0.079
$\text{Error}_{ V _{LBM}}$	=	0.017
$\text{Error}_{c_{LBM}}$	=	1.432
$\text{Error}_{\text{PTOT}_{LBM}}$	=	0.020

Table 6.13: Absolute errors in ratios between LBM Bouzidi Bounce Back with Normal - US3D solutions

To sum up, the implementation of the new boundary conditions has brought many benefits and solved several problems:

- Correct application of Slip Condition:** In the solution obtained in the LBMHYPE project, by applying Bouzidi's boundary condition with the purpose of applying the slip condition, a No-slip condition had been imposed on the wall, which is not appropriate in the case where the Euler equations are being solved. The main problem was due to the geometry considered (i.e., the wedge) did not conform with the regular lattice typical of the LBM, and this led to an error in the imposition of the slip condition that had not been observed in the case of geometries conformed with the lattice (e.g., the Forward Facing Step). Through the implementation within the python library, pylbm [4], of the boundary conditions named Bouzidi Bounce Back with Normal, the slip boundary condition was correctly applied. In fact, as seen in Figures 6.50 and 6.51, while the velocity u_{\perp} approaches zero, the velocity u_{\parallel} has a value imposed by the fluid, unlike the previous case in which u_{\parallel} also tended to zero, like many fluid-dynamics quantities (in fact, by imposing the no-slip condition is observed the well-known phenomenon of the boundary layer).

- **Distance δ very close to the theoretical value zero:** In the solution obtained in the LBMHYPE project, by applying the Bouzidi boundary condition, the boundary layer phenomenon was observed. As a result, the body invested by the supersonic flow no longer has the rectilinear shape of the wedge but becomes a streamlined body (similar to a blunt body), and thus the shock wave is detached (the distance of the shock wave from the leading edge was estimated to be 27 mm). Since with the imposition of the boundary conditions Bouzidi Bounce Back with Normal, no boundary layer is observed, in that case, the shape involved by the supersonic flow is the correct one. This led to a drastic reduction in the effective distance between the shock wave and the leading edge, estimating a distance of 1.8 mm. The result is of the same order of magnitude as that obtained with US3D.

On the other hand, however, there are not only pros to the implementation of the new boundary condition but also cons to the solution. However, the defects found in the solution obtained with Bouzidi Bounce Back with normal boundary conditions are due to the method, and not to the new boundary condition. In fact, as much as D_2Q_{4444} is a simple and robust scheme, it has a rigid numerical diffusion and this induces a solution of first-order for Euler equations. Therefore, the β angle is not calculated with an infinitesimal error as in the US3D solution, and similar reasoning applies to all other fluid dynamics quantities downstream of the shock wave.

Reduction of numerical diffusion

To achieve a better solution, it is necessary to reduce the numerical diffusion of the scheme. As previously mentioned, by writing the corresponding equations up to the second order (obtained using the expansion of Dubois [79] [80]) the amplitude of the associated numerical diffusion is given by $\Delta x \lambda \sigma$, thus a function not only of λ and σ but also of the space step. With the parameters set in Table 6.5 this diffusivity parameter is equal to **0.0069**. So in order to decrease numerical diffusion, it is obvious that it is necessary to try to reduce these three parameters.

- **Decrease of λ :** Decreasing the lattice velocity greatly decreases the diffusion of the scheme, however, it must be remembered that this parameter plays a key role in CFL-type stability. Despite what might be expected, decreasing the lambda parameter observed a program crash not due to the stability of the scheme, but rather due to the imposition of boundary conditions. In particular, a mathematical error occurs, namely a negative square root in the calculation of p^* pressure, which is required for the imposition of the new boundary conditions, and this leads to a program crash. The value $\lambda = 11$ was set to avoid this type of error, so the limitation is not due to the stability

of the scheme. Therefore, numerical diffusion cannot be reduced through the parameter λ .

- **Decrease of σ :** Decreasing sigma means increasing the value of the relaxation parameter since $\sigma = \frac{1}{s} - \frac{1}{2}$. From the Hennon relationship just written, it is clear that the relaxation parameter can take values $s \in (0, 2]$. The ideal case would be to set $s = 2$, so as to have sigma = 0, thus a solution without numerical diffusion. In that case, also a second-order accurate solution would be achieved. The problem with this solution is the strong instability at discontinuities, with the occurrence of spurious oscillations also typical of conventional CFD. The set value of $s = 1.6$ (i.e., $\sigma = 0.125$) is the highest value such that a solution with no spurious instabilities has been obtained. Therefore, numerical diffusion cannot be reduced through the parameter σ .
- **Decrease of Δx :** Last but not least, to reduce the diffusivity of the scheme, it is also possible to play with the space step. Clearly, it is possible to reduce this parameter, but keeping in mind that it is always necessary to satisfy the CFL condition, but also not to increase the computational cost.

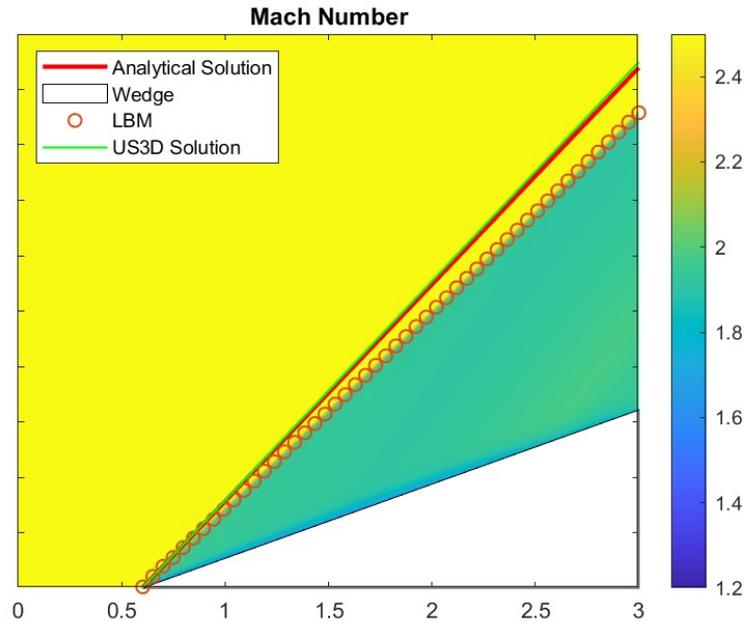


Figure 6.62: Mach Number Field - LBM Bouzidi Bounce Back with Normal Solution - $\lambda = 11$; $s = 1.6$; $\Delta x = 0.0025$

Based on what we have seen, the only way to reduce numerical diffusion is through the space step. Figure 6.62 shows the Mach number field obtained with a space step halved from the previous case (i.e. $\Delta x = 0.0025$), keeping λ and σ unchanged for the reasons just mentioned. As observed, the shock wave has a very good approximation up to one and a half meters from the leading edge and then departs from the analytical solution. So the problem of calculating β is not solved by simply halving the diffusivity, but it would be necessary to reduce the values further, which is not possible.

Concerning the values downstream of the shock wave, to verify that the accuracy of these is actually improved, let us diagram the trend of Mach number (as the ratio of upstream to downstream value) in a shear section for $x = 0.5$ m from the leading edge of the wedge. From Figure 6.63, it can be seen that the error is smaller than

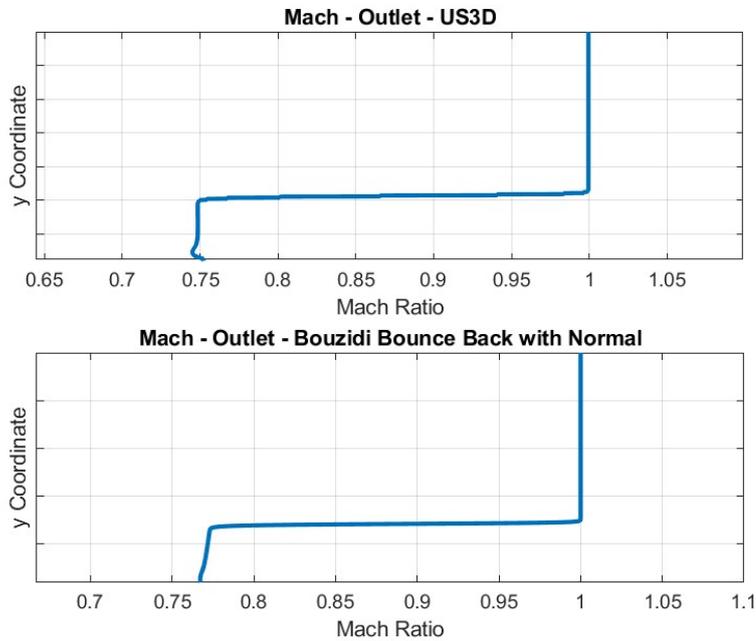


Figure 6.63: Mach Number Ratio Outlet - LBM Bouzidi Bounce Back with Normal Solution - $\lambda = 11$; $s = 1.6$; $\Delta x = 0.0025$

in the previous case (in fact, we have reduced the diffusivity and increased the thickness of the computational grid). However always due to numerical diffusion, there is an error with respect to US3D. Table 6.14 shows the results obtained with the halved Δx .

$Mach_{\text{outlet}}$	=	1.914	
β	=	36.580	[°]
p_{outlet}	=	2.362	[Pa]
Qx_{outlet}	=	5.384	[kg/m ² s]
Qy_{outlet}	=	1.362	[kg/m ² s]
ρ_{outlet}	=	2.547	[kg/m ³]
Ux_{outlet}	=	2.114	[m/s]
Uy_{outlet}	=	0.535	[m/s]
E_{outlet}	=	11.962	[J]
$ V _{\text{outlet}}$	=	2.181	[m/s]
c_{outlet}	=	1.140	[m/s]
$P_{TOT_{\text{outlet}}}$	=	16.169	[Pa]
$u_{\perp_{\text{outlet}}}$	=	0.031	[m/s]
$u_{\parallel_{\text{outlet}}}$	=	1.904	[m/s]

Table 6.14: Results - LBM - Bouzidi Bounce Back with Normal - $\lambda = 11$; $s = 1.6$; $\Delta x = 0.0025$

Table 6.15 shows the relative errors obtained with halved Δx , compared with the analytical solution. What is observed is a reduction in error, which in this case is close to 2 percent for most of the fluid dynamics variables, with the exception of pressure. Thus, the halving of diffusivity led to the desired improvement. Clearly, the error is always not at US3D levels, but the D_2Q_{4444} scheme is not as accurate as CFD's SOA.

$\text{Error}_{Mach_{LBM}}$	=	2.149	%
$\text{Error}_{\beta_{LBM}}$	=	3.395	%
δ_{LBM}	=	1.525	[mm]
$\text{Error}_{p_{LBM}}$	=	4.266	%
$\text{Error}_{\rho_{LBM}}$	=	2.548	%
$\text{Error}_{E_{LBM}}$	=	2.206	%
$\text{Error}_{ V _{LBM}}$	=	1.243	%
$\text{Error}_{c_{LBM}}$	=	0.886	%
$\text{Error}_{P_{TOT_{LBM}}}$	=	1.870	%
$\text{Error}_{u_{\perp_{LBM}}}$	=	0.031	[m/s]
$\text{Error}_{u_{\parallel_{LBM}}}$	=	2.051	%

Table 6.15: Errors between LBM Bouzidi Bounce Back with Normal - Analytic solutions - $\lambda = 11$; $s = 1.6$; $\Delta x = 0.0025$

Effect of λ on the solution

This small concluding paragraph aims to show an undesirable effect that the parameter λ might have on the accuracy of the solution. In fact, in addition to playing a key role in the stability of both the scheme and the new boundary conditions, in addition to playing a role in the diffusivity of the scheme, if too high a value is set, the formation of a boundary layer-like would be observed, as can be seen in Figure 6.64 or even better in Figure 6.65.

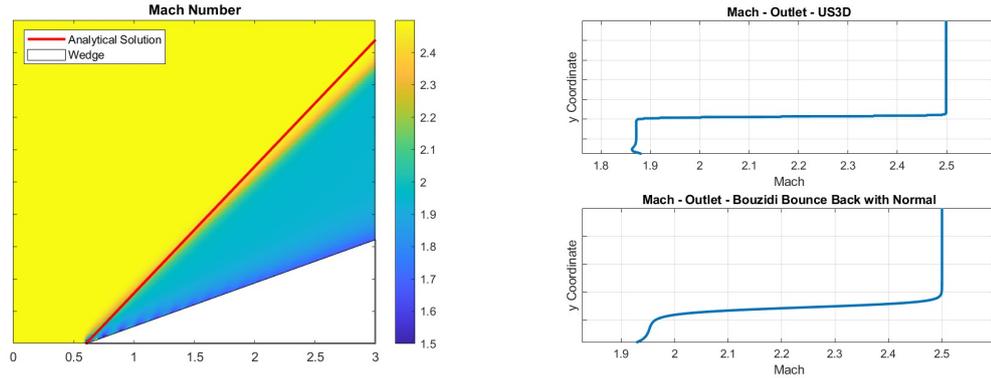


Figure 6.64: Mach Number Field - **Figure 6.65:** Mach Number Outlet -
 LBM Bouzidi Bounce Back with Normal Solution - $\lambda = 15$; $s = 1.6$; $\Delta x = 0.005$ LBM Bouzidi Bounce Back with Normal
 Solution - $\lambda = 15$; $s = 1.6$; $\Delta x = 0.005$

This effect is due to the following reason, as λ increases, the numerical flux correction introduced with the Bouzidi Bounce Back with Normal boundary conditions decreases (as is clearly observed from the 5.28 and 5.29 relations). Then the solution approaches the classical Bouzidi boundary condition (in the limit for $\lambda \rightarrow \infty$ coincides with the LBM solution with Bouzidi boundary conditions). Hence, the same phenomenon encountered in the LBMHYPE project is found, i.e., a wall boundary layer.

Chapter 7

Supersonic cylindrical-nosed body

The Bouzidi Bounce-Back with Normal boundary conditions is independent of the geometry used. The flux corrector term is evaluated with respect to the local normal to the geometry so they are general and independent of geometry. Having successfully tested the Bouzidi Bounce Back boundary conditions (within the limitations seen in the previous chapter regarding the diffusion of the scheme), in this section, we verify that these boundary conditions are applicable to any geometry. To this end, this chapter will study the problem of a supersonic flow over a blunt body with the same upstream properties as the previous test case. The test case chosen is the infinitely long cylinder, as it is simpler than the sphere since it does not have three-dimensional effects, so it is two-dimensional ¹. The choice of such a test case is first of all due to the non-rectilinear geometry, so as to test the BCs on a curved surface. The resolution of the blunt body is crucial to the future of the project since it is a particularly important shape in hypersonic aerodynamics, in which all vehicles have blunt noses to reduce aerodynamic heating.

¹In addition, with the pylbm package you can solve problems up to two dimensions.

7.1 Bow shock

Bow shocks are common near blunt bodies [121] [122] [123] such as SpaceX's Dragon reentry vehicle, shown in Figure 7.1. In a blunt-nosed body, there is always a detached bow shock with a subsonic pocket behind it and a subsequent expansion of the flow to supersonic speed. Moving at a far distance from the vehicle, the shock vanishes and its inclination with respect to the free stream direction approaches asymptotically the Mach angle (i.e. $\mu_\infty = \arcsin(\frac{1}{M_\infty})$).

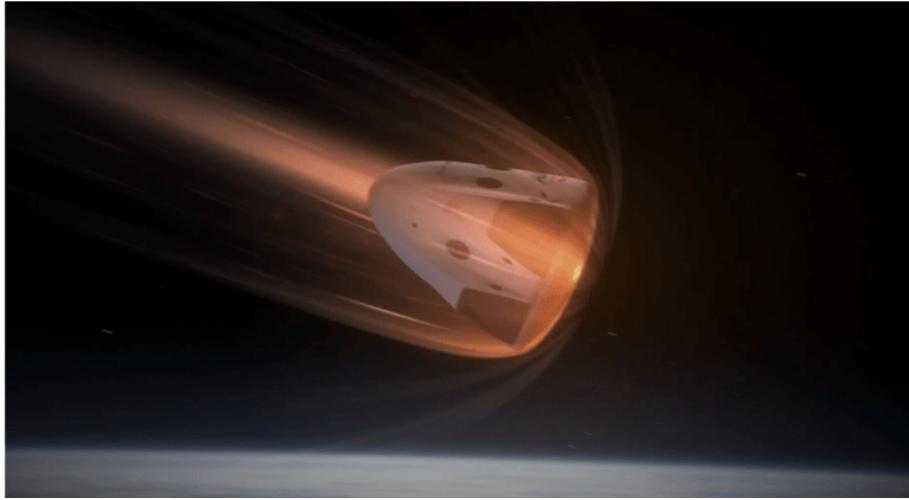


Figure 7.1: 3D graphic representation of the reentry of the Dragon capsule (SpaceX)

As mentioned in the chapter on wedge there is a maximum flow deflection angle θ_{max} that an oblique shock can handle for a given upstream Mach number M_1 . If the deflection angle θ is greater than θ_{max} , the maximum flow deflection angle, the assumptions concerning the oblique shock then fail, and a curved bow shock generates in front of the body. This bow shock has a curved shape and is detached from the body.

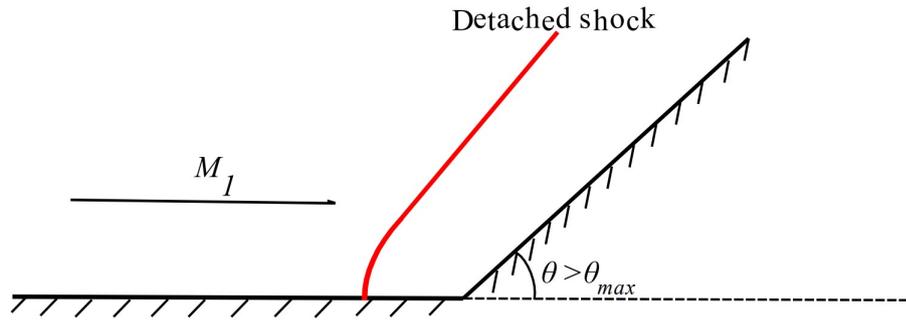


Figure 7.2: Supersonic flow over a wedge - Detached bow shock

In the issue of a circular surface hit by a supersonic flow, the deflection angle locally is 90° , so it is obvious that the shock will be detached. The flow velocity decreases from supersonic upstream to subsonic downstream as a result of the non-isentropic thermodynamic change across a bow shock.

7.1.1 Theoretical background

In order to better analyze the flow field we are studying, let us consider the supersonic flow over a body with a blunt nose, as shown in Figure 7.3, following the procedure proposed by [127]. As it is well evident, in front of this body, a strong curved bow shock wave emerges. The distance δ separates the shock from the nose (δ known as standoff distance). Focusing more on Figure 7.3, at point **a**, the wave's upstream flow is a normal shock, in fact, immediately downstream of it there is a subsonic pocket, as the shock wave is strong. The shock wave weakens and curves as it moves away from the center, finally turning into a Mach wave at great distances from the body (point **e** in 7.3). In addition, between points **a** and **e**, for an upstream Mach number M_1 , the curved shock passes through every scenario allowed for oblique shocks (i.e. all the possibilities that are feasible in the graph 6.6).

- **Point a:** Normal shock, so to calculate the value immediately downstream of the shock wave it is possible to use the relationships seen in the previous chapter of Rankine-Hugoniot [124] [125], which are only locally valid.
- **Point b:** An oblique shock occurs, very close to the normal shock wave. As a result, the shock wave is still strong, in fact downstream of the shock it is within the subsonic pocket.
- **Point c:** It marks the separation between strong and weak solutions; the highest deflection θ_{max} occurs in the streamline through point c.

- **Point c'**: The shock leads the flow to become sonic. Above this point the flow downstream of the shock is supersonic.
- **Point d**: Weak oblique shock
- **Point e**: Mach line

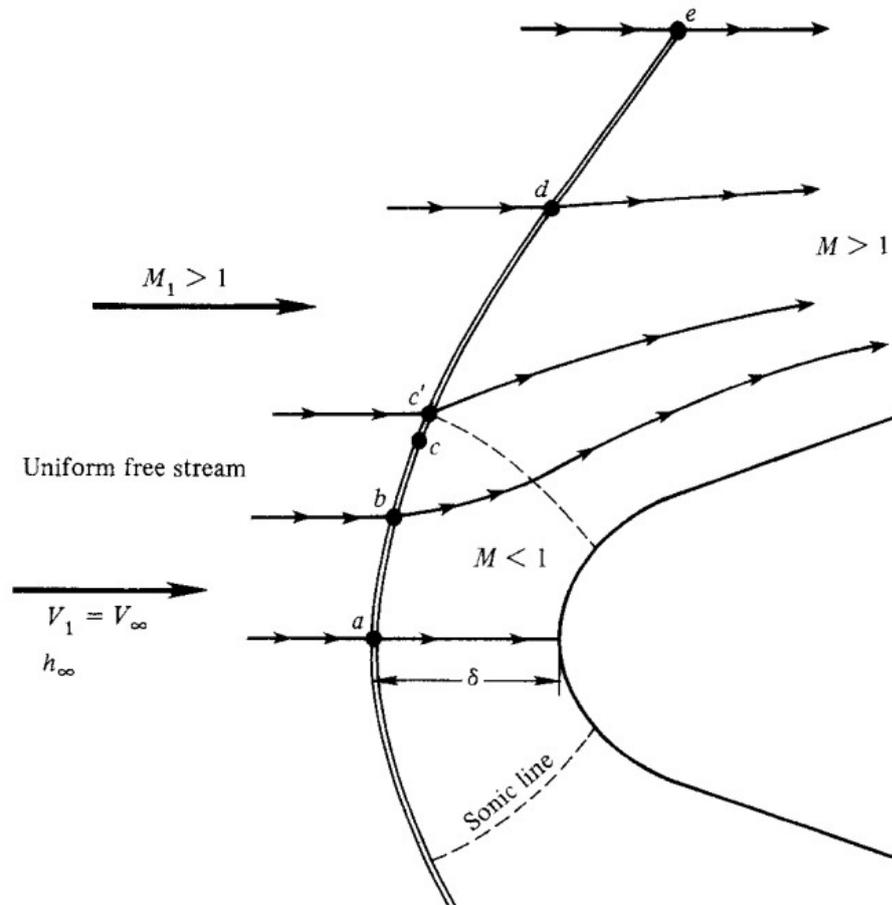


Figure 7.3: Flow over a blunt body traveling at supersonic speed.

It can be seen from Figure 7.3 that a mixed subsonic-supersonic flowfield occurs between the blunt body and its curving bow shock ². Since there is a mixed subsonic-supersonic field was very challenging to solve in the 1950s and 1960s when this problem was first faced.

²The sonic line is the imaginary dividing line between these two regions (when $M = 1$)

This flowfield's solution is not simple. The Mach number, as well as the size and shape of the body, influence the detached shock wave's shape, its detachment distance δ , and the entire flowfield.

7.1.2 Billig's Approximation

The blunt body problem has been solved for the first time through characteristics. Nevertheless, in the literature, there are very simple but effective methods based on the geometry observed and the shock wave's sonic point available [128]. Even so, it would be interesting to find out if there are additional simplifications that would produce results that closely matched those of the experiments. In this regard there is an interesting empirical solution in the literature to estimate the position of the shock wave, and also the shape, by knowing the input Mach number and the radius of the object invested by the flow, known as Billig's solution [129]. Since it is an empirical correlation it has limitations on validity, i.e., it can be applied for perfect gases, thus for a Mach number not exceeding 8, so the effects of aerothermodynamics are not significant. This approach makes the assumption that the detached shock wave is an asymptotic hyperbola to the freestream Mach angle (i.e. $\mu_\infty = \arcsin(\frac{1}{M_\infty})$)³. The equation for the shock's coordinates is the following:

$$x = R + \delta - R_c \cotan^2 \mu_\infty \left[\left(1 + \frac{y^2 \tan^2 \mu_\infty}{R_c^2} \right)^{1/2} - 1 \right] \quad (7.1)$$

In Billig's original work, expressions for the standoff distance δ and the vertex radius of curvature R_c are derived experimentally to determine the hyperbola's shape. In our case, to estimate these two parameters in equation 7.1 we use the proposal presented in [131], in which the experimental results for standoff distance from various sources were correlated in case you are studying the cylinder as⁴:

$$\frac{\delta}{R} = 0.386 \cdot \exp\left(\frac{4.67}{M_{in}^2}\right) \quad (7.2)$$

While in the case of the relationships for the vertex radius of curvature, they came to the following relation⁵:

$$\frac{R_c}{R} = 1.386 \cdot \exp\left(\frac{1.8}{(M_{in} - 1)^{0.75}}\right) \quad (7.3)$$

³In [130] is presented the case of detached shock for the wedge

⁴In the case where the case of the sphere is being studied, that is, in the case where the three-dimensional effects are included, the empirical relation 7.2 becomes $\frac{\delta}{R} = 0.143 \cdot \exp(\frac{3.24}{M_{in}^2})$

⁵Again, if a sphere is considered, in order to consider three-dimensional effects, the 7.3 relation becomes $\frac{R_c}{R} = 1.143 \cdot \exp(\frac{0.54}{(M_{in}-1)^{1.2}})$

Figure 7.4 is shown a comparison of shock wave shapes for flow across cylinders computed and measured from references [132], [133], [134], [135], [136], and [137].

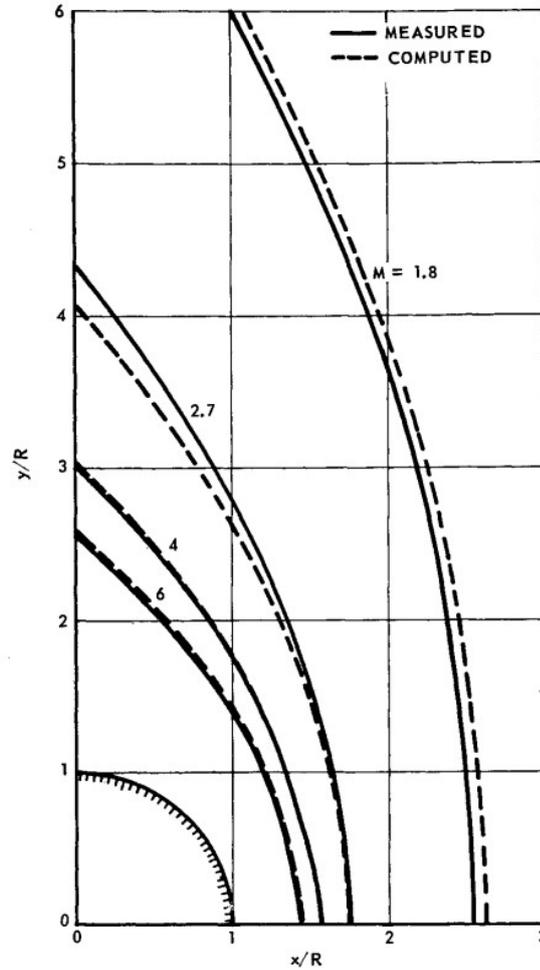


Figure 7.4: Comparison of shock wave shapes for flow across cylinders computed and measured from references [132], [133], [134], [135], [136], and [137]

The close agreement between data and theory for both spherical-nosed and cylindrical-nosed bodies serves as proof that this strategy is effective. It would be expected that the results would agree with theories for a perfect gas with $\gamma = 1.4$ since the correlations that led to equations 7.1-7.3 were based on tests conducted at relatively low temperatures. The shock-wave shape will only be significantly affected by real gas effects at Mach values higher than or equal to 8.

This solution is not exact, but it gives a fairly accurate estimate of the location of the shock, and at SOA it is used as a first approximation. For example, it is very useful to have an idea of the position of the shock wave before launching a CFD simulation, so that it is possible to know *a priori* where it will be necessary to refine the mesh.

7.2 Test case Cylinder

The case study chosen to test the newly implemented boundary conditions is the simplest bidimensional test case that adopts a non-rectilinear geometry. Thus we solve in this chapter with the LBM a supersonic flow investing the section of a cylinder (not a sphere to avoid 3D effects) whose radius is 0.5 m. Figure 7.5 schematizes the chosen test case, in order to visualize the size of the domain set as well.

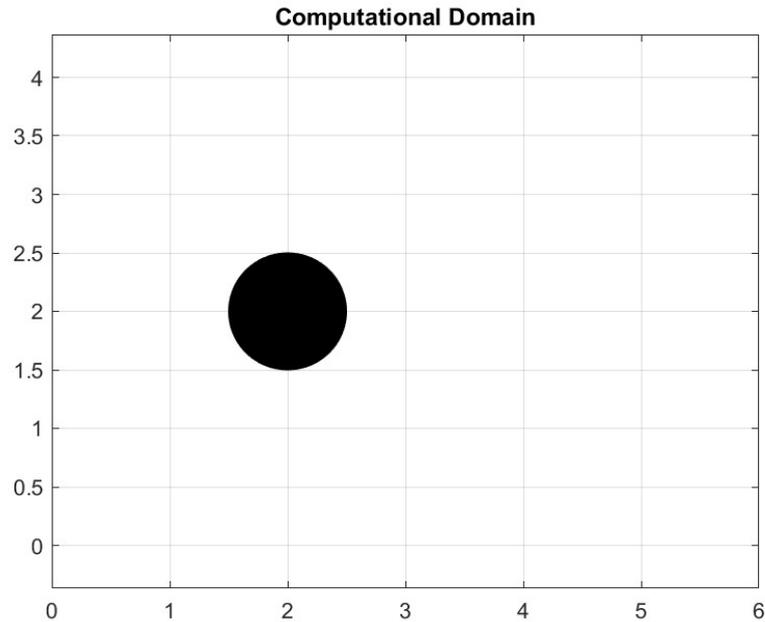


Figure 7.5: Schematic description of the flow domain

The chosen test case has the input Mach number equal to $M_1 = 2.5$ and the radius equal to 0.5 m as key parameters. The other parameters set in the freestream are shown in the table 7.1. The parameters set in 7.1 are set to focus only on the numerical aspect i.e., the same approach is being taken as that used for the wedge.

$Mach_1$	=	2,5	
p_1	=	1	[Pa]
T_1	=	8.5E-05	[K]
Qx_1	=	3,5	[kg/m ² s]
Qy_1	=	0	[kg/m ² s]
ρ_1	=	1,4	[kg/m ³]
Ux_1	=	2,5	[m/s]
Uy_1	=	0	[m/s]
E_1	=	6,875	[J]
$ V _1$	=	2,5	[m/s]
c_1	=	1	[m/s]
P_{TOT1}	=	17.086	[Pa]

Table 7.1: Cylinder Parameters - Free Stream

7.2.1 Analytical solution normal shock

Regarding analytical resolution, the solution of the oblique shock (plane and attached) is no longer valid for the calculation of the field, but immediately downstream of the curved shock, the field can still be calculated with the results of the oblique shock theory. Thus just downstream of the discontinuity, the relationships for evaluating fluid properties are derived directly from the Rankine-Hugoniot equations [124] [125]. Past that location, the flow evolves differently and the streamlines follow a curved path because it must still connect with the direction of the downstream wall to satisfy the slip condition.

Therefore, the analytical values found downstream of the shock for $y = 2$ are reported, such that the shock is normal (i.e., $\beta = 90$). In this situation, the relations are exactly those of Rankine-Hugoniot for the normal shock. Exactly as in the previous case, once the Mach number is obtained, from the equation of state of gases ⁶the entire field downstream of the shock can be obtained. All the main values of the downstream field are given below and summarized in Table 7.2.

- For the Mach number we employ the corresponding equation of Rankine-Hugoniot:

$$M_2^2 = \frac{1 + \left[\frac{\gamma-1}{2}\right] M_1^2}{\gamma M_1^2 - \frac{\gamma-1}{2}} \rightarrow M_2 = 0.513 \quad (7.4)$$

⁶In fact 6.6 relationship is proved by the equation of state, in fact, $\frac{T_2}{T_1} = \frac{p_2}{p_1} \frac{\rho_1}{\rho_2}$.

- For the density we employ the corresponding equation of Rankine-Hugoniot:

$$\frac{\rho_2}{\rho_1} = \frac{\frac{\gamma+1 p_2}{\gamma-1 p_1} + 1}{\frac{\gamma+1}{\gamma-1} + \frac{p_2}{p_1}} \rightarrow \rho_2 = 4.667[kg/m^3] \quad (7.5)$$

- For the pressure we employ the corresponding equation of Rankine-Hugoniot:

$$\frac{p_2}{p_1} = \frac{\frac{\gamma+1 \rho_2}{\gamma-1 \rho_1} - 1}{\frac{\gamma+1}{\gamma-1} - \frac{\rho_2}{\rho_1}} \rightarrow p_2 = 7.125[Pa] \quad (7.6)$$

- For the temperature field we employ the corresponding equation of Rankine-Hugoniot, proved by the equation of state, in fact, $\frac{T_2}{T_1} = \frac{p_2}{p_1} \frac{\rho_1}{\rho_2}$:

$$\begin{aligned} \frac{T_2}{T_1} &= \left[1 + \frac{2\gamma}{\gamma+1} (M_1^2 - 1) \right] \left[\frac{2 + (\gamma-1)M_1^2}{(\gamma+1)M_1^2} \right] \\ &\rightarrow T_2 = 1,84E - 04[K] \end{aligned} \quad (7.7)$$

- For the total pressure field we employ the corresponding equation of Rankine-Hugoniot:

$$\begin{aligned} \frac{p_{TOT2}}{p_{TOT1}} &= \left(1 + \frac{2\gamma}{\gamma+1} (M_1^2 - 1) \right)^{-\frac{1}{(\gamma-1)}} \left(\frac{(\gamma+1)M_1^2}{2 + (\gamma-1)M_1^2} \right)^{\frac{\gamma}{\gamma-1}} \\ &\rightarrow p_{TOT2} = 8.526[Pa] \end{aligned} \quad (7.8)$$

- From the definition of the speed of sound, given the pressure and temperature, it is possible to obtain c_2 of the analytical solution.

$$c_2 = \sqrt{\gamma \frac{p_2}{\rho_2}} = 1.462[m/s] \quad (7.9)$$

- Known the Mach number and the speed of sound, it follows that the absolute value of the speed will be:

$$|V|_2 = M_2 c_2 = 0.750[m/s] \quad (7.10)$$

- Then we evaluate the energy in the field downstream of the shock:

$$E_2 = \frac{1}{2} \rho_2 |V|_2^2 + \frac{p_2}{\gamma-1} = 19.125[J] \quad (7.11)$$

- Knowing also the components of the velocities, we calculate the momentum in the two directions, since those will be two starting parameters for the LBM, being two of the conserved moments of the scheme used:

$$Qx_2 = Ux_2\rho_2 = 3.5[kg/m^2s] \quad (7.12)$$

$$Qy_2 = UY_2\rho_2 = 0[kg/m^2s] \quad (7.13)$$

- Under the imposed assumptions, the flux is adiabatic (i.e., $\left(\frac{T_{o2}}{T_{o1}}\right) = 1$). The entropy drop therefore can be calculated according to the following relationship:

$$\Delta s = -\frac{R}{M} \log\left(\frac{p_{o2}}{p_{o1}}\right) = 199.284[J/K] \quad (7.14)$$

$Mach_{2Analytical}$	=	0.513	
$p_{2Analytical}$	=	7.125	[Pa]
$T_{2Analytical}$	=	1,84E-04	[K]
$\rho_{2Analytical}$	=	4.667	[kg/m ³]
$E_{2Analytical}$	=	19.125	[J]
$ V _{2Analytical}$	=	0.750	[m/s]
$c_{2Analytical}$	=	1.462	[m/s]
$p_{TOT2Analytical}$	=	8.526	[Pa]
$\Delta s_{Analytical}$	=	199.284	[J/K]

Table 7.2: Analytical Solution - Downstream normal shock - $y = 2$

In order to get an initial idea about the position of the shock wave, we use the previously mentioned Billig approximation. Using the 7.2 relationship, a standoff distance of **0.41 m** is estimated with Billig, while as for the radius of curvature through the relation 7.3 a value of **2.61 m** is obtained. Figure 7.6 shows the analytical representation of the solution, obtained from equation 7.1.

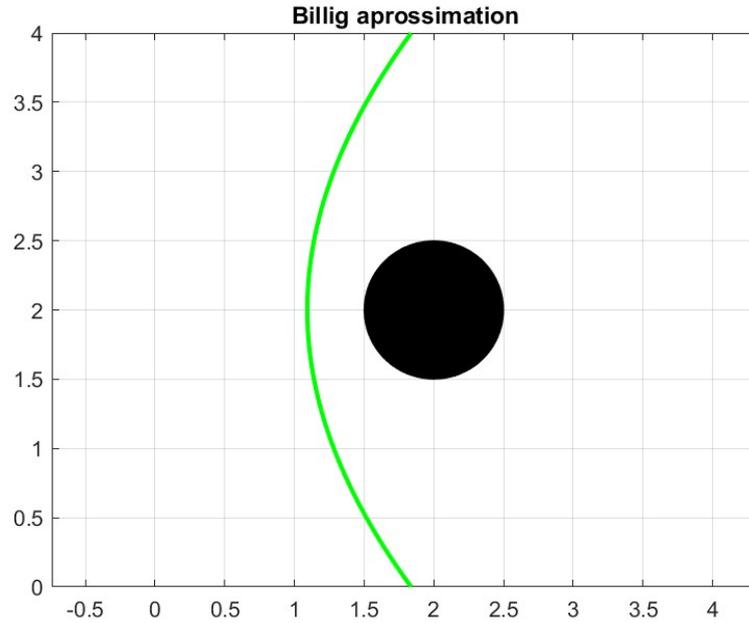


Figure 7.6: Billig's approximation

7.3 LBM Solution

In the LBM simulation, we consider a computational domain as represented in Figure 7.5, then let us consider a 6m x 4m domain, in which the section of the cylinder has its center at the point (2,2) and a radius of 0.5 m. The input parameters of the LBM simulation are given in Table 7.3, and are the results of a trade-off between stability and accuracy of the simulation. In addition, those are set such that there is an optimal solution with the new boundary conditions.

$\sigma =$	0.167
$s =$	1,5
$\lambda =$	11
$\Delta x =$	0,005
$\Delta t =$	4,5E-4

Table 7.3: Input parameters in the LBM simulation

The same considerations made in Section 6.4.1 of the previous chapter apply to parameter selection.

7.3.1 Bouzidi BCs Solution

In this section, we are going to look at the cylindrical-nosed body solution by imposing Bouzidi 4.3 conditions on the curved surface ⁷.

Figure 7.7 shows the field of Mach number obtained with the LBM simulation. As well as the analysis performed on the wedge with Bouzidi's boundary conditions, already from this figure, we can see the problems faced in the LBMHYPE project. In fact, the slip condition is misapplied, in fact, it is observed a Mach number that tends to zero globally at the wall (i.e., no-slip condition). From Figures E.1 - E.6, the fields of several fluid dynamic variables are shown, in which the same problems found in 7.7 can be observed.

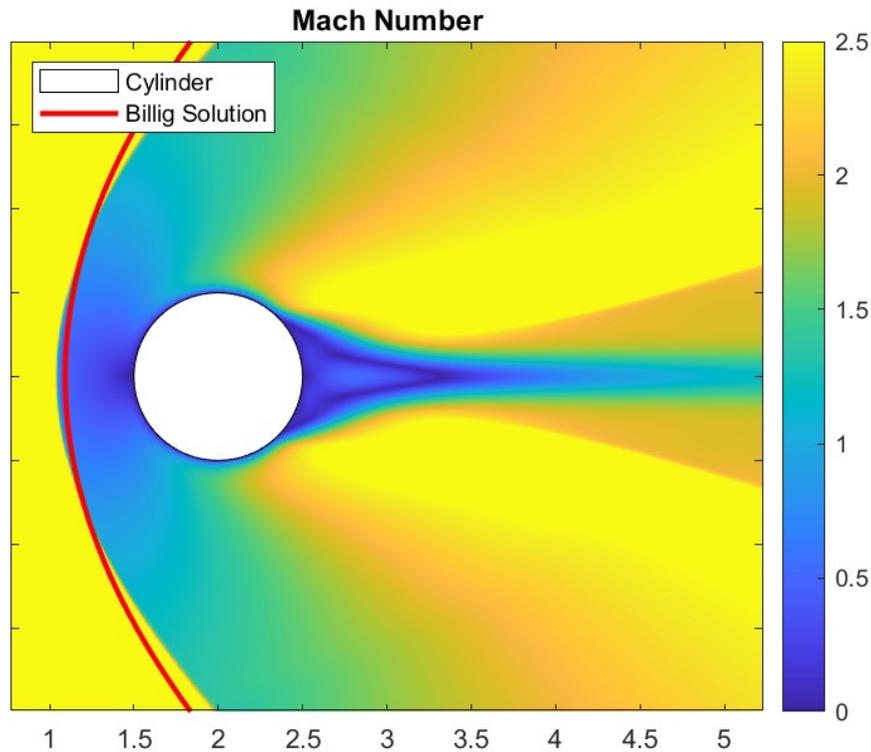


Figure 7.7: Mach Number Field - LBM Bouzidi Solution

Figure 7.8 shows the comparison between the shock position obtained with LBM and Billig's analytical solution. What can be observed from the figure is that

⁷Also in this case, the solution obtained with simple Bounce-Back 4.2 was not reported as it was superfluous and worse accurate than Bouzidi's solution.

the LBM solution fits Billig's solution quite well, which, however, is an empirical relationship that is not exact. Moreover, the difference obtained between the two solutions is comparable with the errors found in [132], [133], [134], [135], [136], and [137] between the experimental data and Billig's 7.1 relation for the set Mach number. As the Mach number increases, the Billig error decreases, as can also be guessed from Figure 7.4.

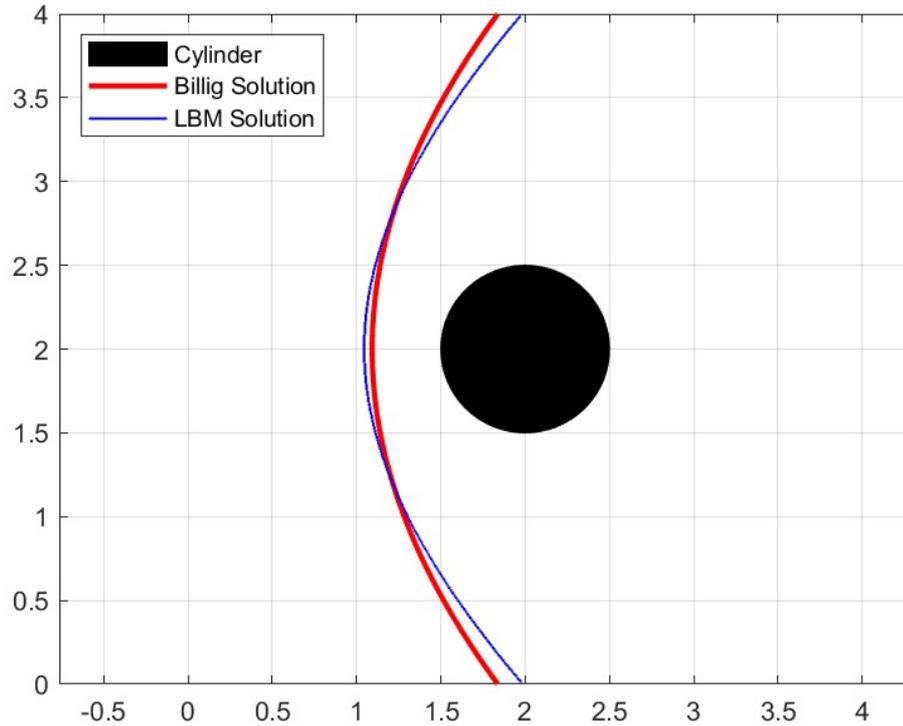


Figure 7.8: The different shock waves considered - LBM (Bouzidi) - Analytical (Billig)

The problem found in the wedge was the incorrect imposition of the slip condition at the wall. In order to check what kind of boundary condition is applied, Figures 7.9 and 7.10 show the velocities parallel to the x-axis and y-axis, respectively. From the two images, it can be seen that both have an infinitesimal value at the wall, so it can be guessed that even in this case, Bouzidi's boundary condition is not appropriate to impose the slip condition at the wall.

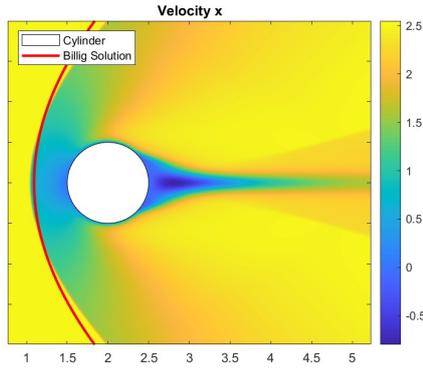


Figure 7.9: Velocity u_x Field - LBM Bouzidi Solution

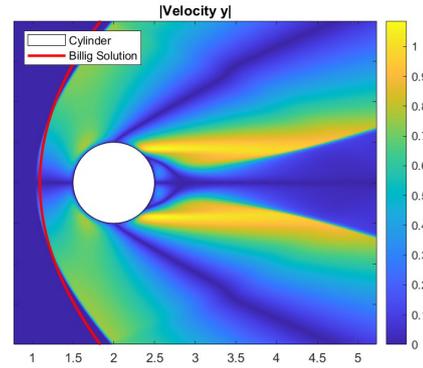


Figure 7.10: Velocity u_y Field - LBM Bouzidi Solution

However, as done in the previous wedge test case, the best way to verify that the wall velocity tends to zero (both u_x and u_y), we considered a shear section positioned at $x = 2$ m, so as to verify that the velocity tends to zero at the upper extreme of the geometry. Figures 7.11 and 7.12 graph the trend of both velocities as the y -coordinate varies. It is clear from the two figures that the velocity has a drop to a zero value at the wall, so again, as in the case of the wedge solved with Bouzidi, the occurrence of the boundary layer is observed.

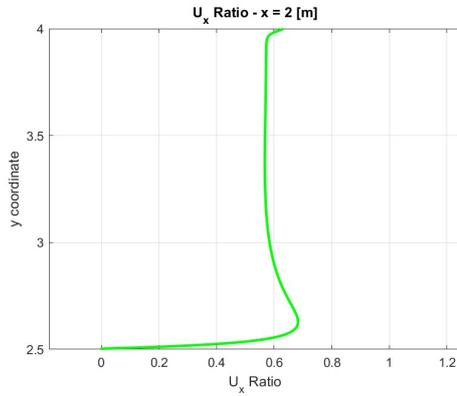


Figure 7.11: Velocity u_x for $x = 2$ m - LBM Bouzidi Solution

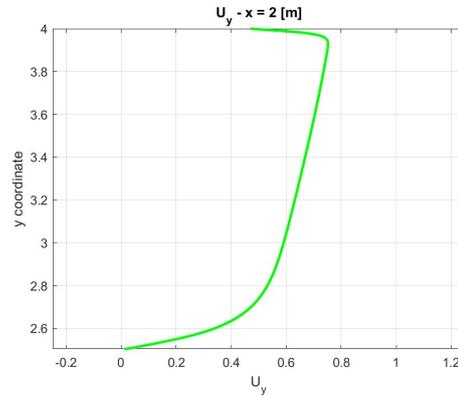


Figure 7.12: Velocity u_y for $x = 2$ m - LBM Bouzidi Solution

The same procedure was carried out for all other fluid dynamics variables, so from Figure 7.13 to 7.16 the trends for $x = 2$ m as the y -coordinate changes are shown.

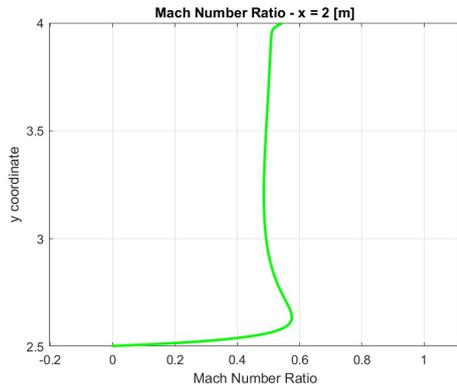


Figure 7.13: Mach for $x = 2$ m - LBM Bouzidi Solution

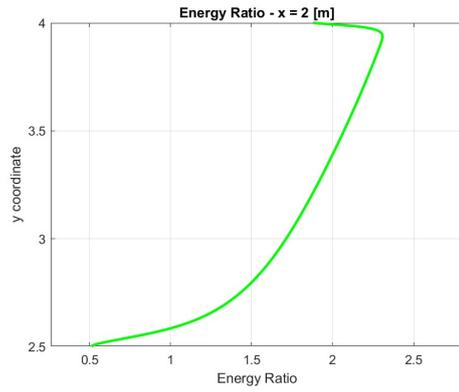


Figure 7.14: Energy for $x = 2$ m - LBM Bouzidi Solution

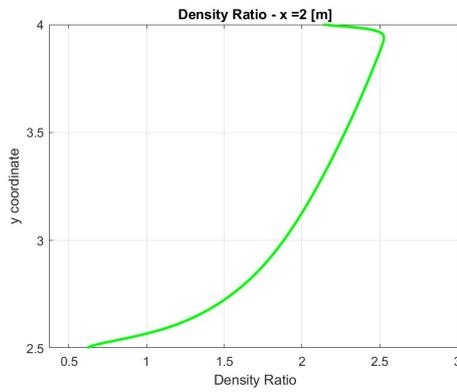


Figure 7.15: Density for $x = 2$ m - LBM Bouzidi Solution

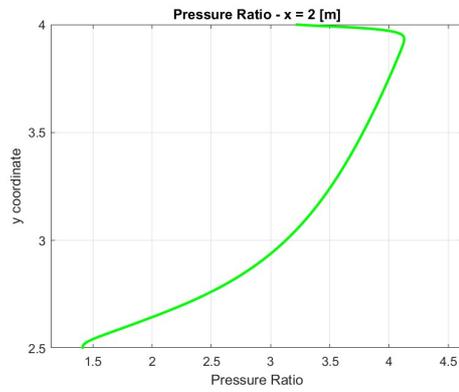


Figure 7.16: Pressure for $x = 2$ m - LBM Bouzidi Solution

From the figures on the previous page, it is clear that the boundary condition is incorrectly applied, as the occurrence of a boundary layer is clear, which does not exist by solving the Euler equations.

In this example, to evaluate the accuracy of the obtained solution, and to make a comparison with the analytical solution we graph the trend of the fluid dynamics variables for $y = 2m$ (i.e., a shear section passing through the stagnation point). The trends of the main fluid dynamics variables ratios as a function of the x coordinate are then shown in Figures 7.17 to 7.23.

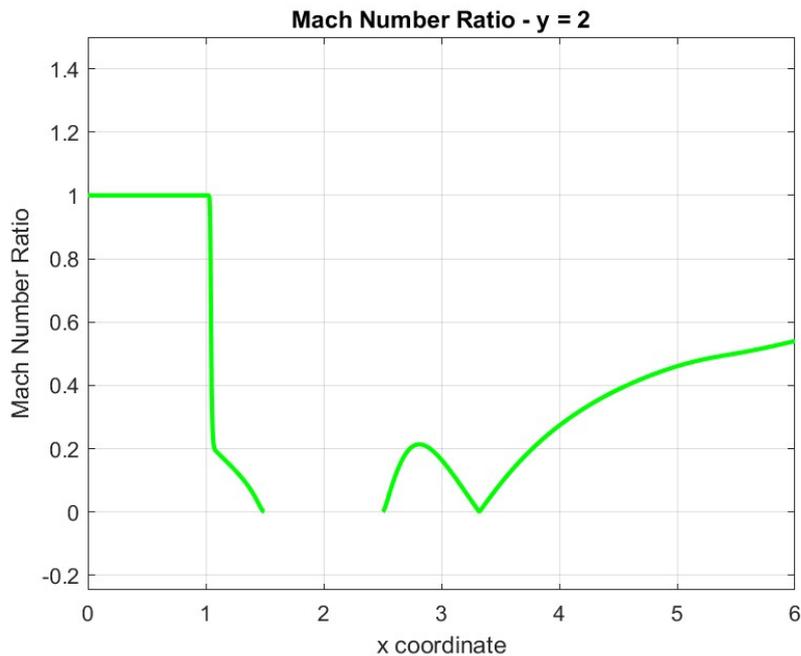


Figure 7.17: Mach number trend for $y = 2$

From Figure 7.17, related to the Mach number, the discontinuity is well evident, as well as how the velocity tends to zero as you approach the geometry (in fact you are approaching a point of isentropic stagnation). Instead, the figures on the next page show typical trends found in fluid dynamics variables downstream of a strong shock wave. In fact, a discontinuous increase in density, pressure, temperature, and energy is observed; in contrast, a drop in total pressure is present.

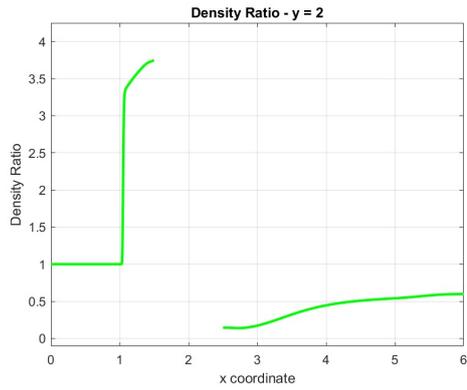


Figure 7.18: Density trend for $y = 2$

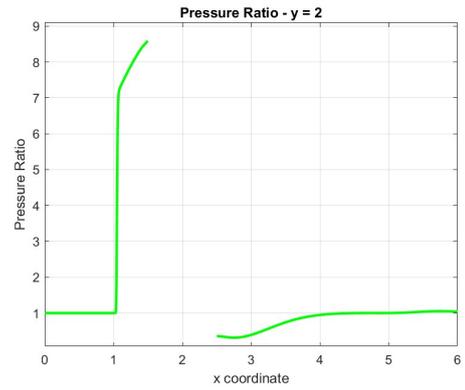


Figure 7.19: Pressure trend for $y = 2$

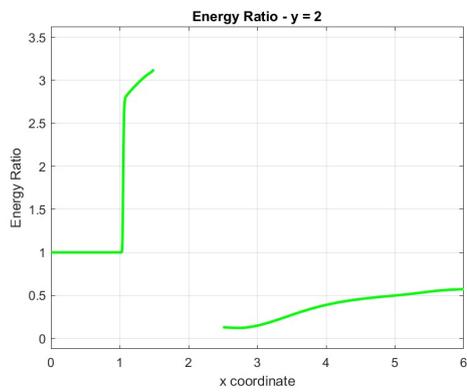


Figure 7.20: Energy trend for $y = 2$

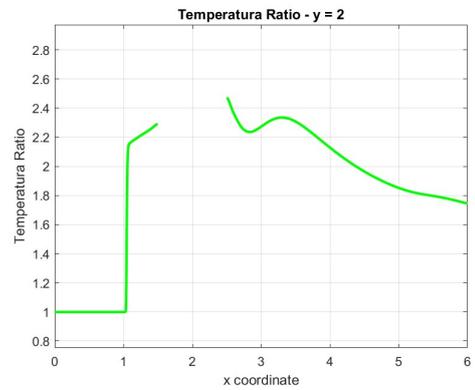


Figure 7.21: Temperature trend for $y = 2$

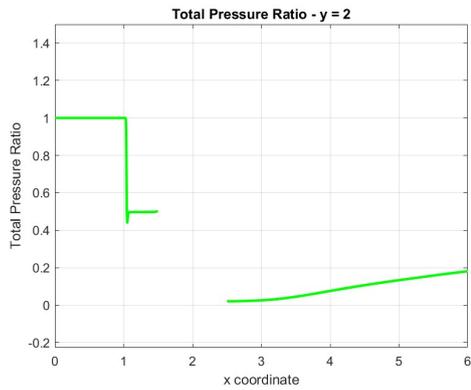


Figure 7.22: Total Pressure trend for $y = 2$

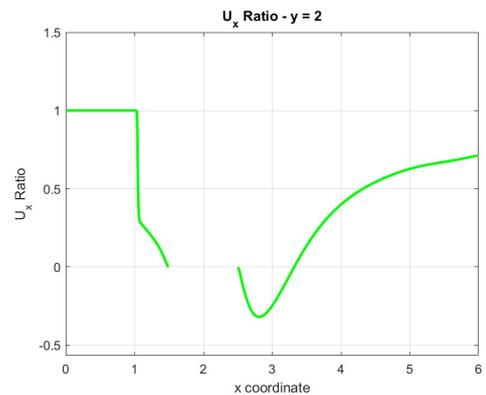


Figure 7.23: u_x trend for $y = 2$

The plotted graphs in Figures 7.17 to 7.23 are essentials for evaluating the post-shock values of all fluid dynamics variables. In fact, as already mentioned above, the analytical solution is only valid in proximity to the shock wave. In this regard, Table 7.4 shows the post-shock values evaluated with the LBM (applying Bouzidi wall boundary conditions).

$Mach_{\text{Post Shock}}$	=	0.512	
$p_{\text{Post Shock}}$	=	7.027	[Pa]
$\rho_{\text{Post Shock}}$	=	4.602	[kg/m ³]
$E_{\text{Post Shock}}$	=	18.857	[J]
$ V _{\text{Post Shock}}$	=	0.749	[m/s]
$c_{\text{Post Shock}}$	=	1.462	[m/s]
$P_{TOT_{\text{Post Shock}}}$	=	8.404	[Pa]
$T_{\text{Post Shock}}$	=	1.84E-04	[K]
Δs	=	203.421	[J/K]

Table 7.4: Results - LBM - Bouzidi

Table 7.5 shows the relative percentage of errors between the analytical solution and the LBM solution. The results obtained downstream of the shock show a very good correlation between the analytical values and the LBM values, with very limited errors around 1%.

$\text{Error}_{Mach_{LBM}}$	=	0.163	%
$\text{Error}_{p_{LBM}}$	=	1.377	%
$\text{Error}_{T_{LBM}}$	=	1,00E-05	[K]
$\text{Error}_{\rho_{LBM}}$	=	1.382	%
$\text{Error}_{E_{LBM}}$	=	1.399	%
$\text{Error}_{ V _{LBM}}$	=	0.160	%
$\text{Error}_{c_{LBM}}$	=	0.003	%
$\text{Error}_{p_{TOT_{LBM}}}$	=	1.433	%
$\text{Error}_{\Delta s}$	=	2.076	%

Table 7.5: Errors between LBM Bouzidi - Analytic solutions

Finally, regarding the standoff distance, with the LBM we estimate a δ value of **0.4550 m**. This value differs by 0.04 m from the value expected by Billig, which is an empirical approximation, that estimated a δ of **0.4074 m**.

7.3.2 Bouzidi Bounce Back with Normal BCs Solution

In this section, we are going to look at the cylindrical-nosed body solution by imposing *Bouzidi Bounce Back with Normal* 5.3 conditions on the curved surface. Figure 7.24 shows the field of Mach number obtained with the LBM simulation. Unlike the solution seen in the previous section (LBM with Bouzidi BCs imposed at the wall), in this example the slip condition seems to be applied appropriately, not observing the boundary layer near the surface. So it follows, that the new boundary conditions *Bouzidi Bounce Back with Normal* are also applicable to non-rectilinear geometries. This is due to the generality of calculating the correction factor (the flux discussed in the section 5.1), which depends on the local normal, which while in the wedge test case was constant, in this example changes as a function of position.

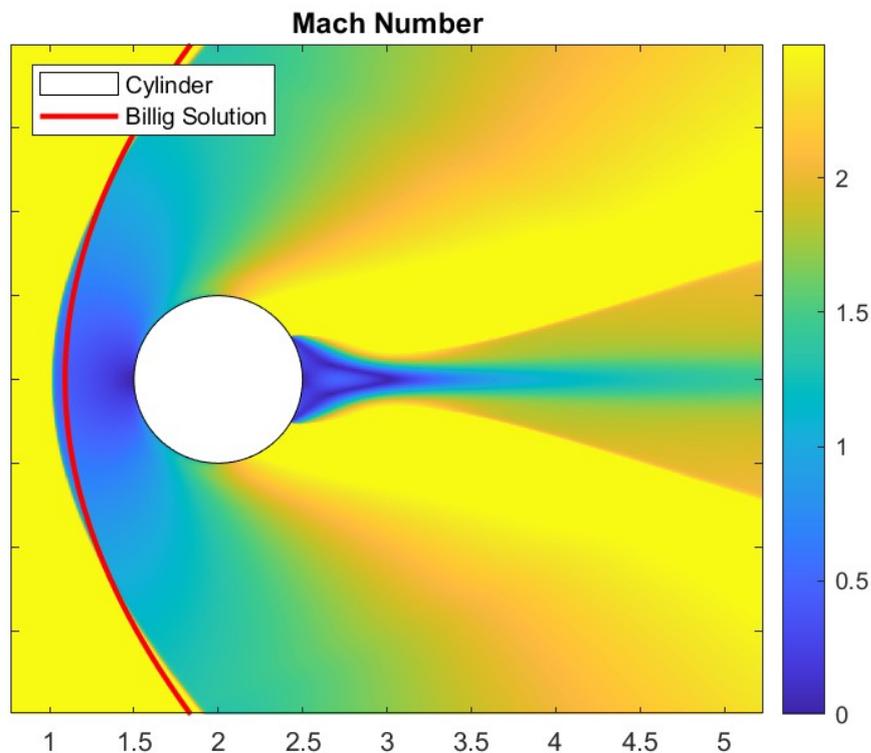


Figure 7.24: Mach Number Field - LBM Bouzidi Bounce Back with Normal Solution

In Appendix E from Figures E.7 - E.12, the fields of several fluid dynamic variables are shown.

Figure 7.25 shows the comparison between the shock position obtained with LBM and Billig's analytical solution. The conclusions that can be taken are similar to those seen with the application of Bouzidi's boundary conditions, in fact, the LBM solution fits Billig's solution quite well and the difference obtained between the two solutions is comparable with the errors found in [132], [133], [134], [135], [136], and [137] between the experimental data and Billig's 7.1 relation for the set Mach number.

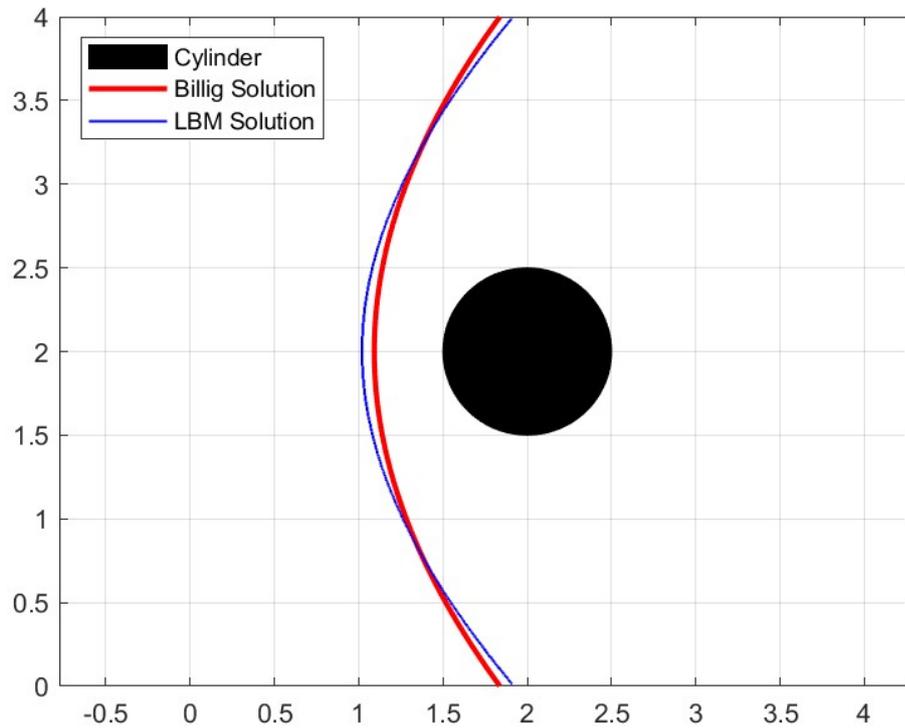


Figure 7.25: The different shock waves considered - LBM (Bouzidi Bounce Back with Normal) - Analytical (Billig)

The problem that has been found in Bouzidi is the incorrect slip condition, as already seen several times in this thesis. In order to check what kind of boundary condition is applied, Figures 7.26 and 7.27 show the velocities parallel to the x-axis and y-axis, respectively. From the two images, it can be seen that there is no boundary layer, as was already observed in Figure 7.24 on the Mach number.

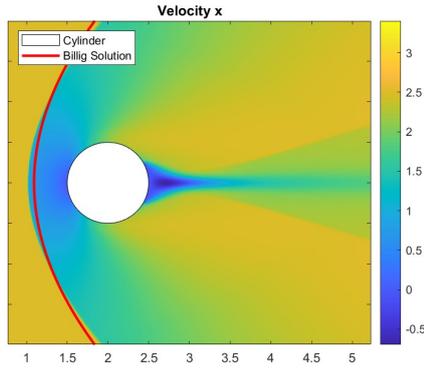


Figure 7.26: Velocity u_x Field - LBM Bouzidi Bounce Back with Normal Solution

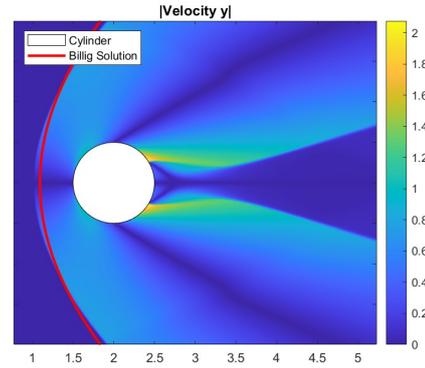


Figure 7.27: Velocity u_y Field - LBM Bouzidi Bounce Back with Normal Solution

However, as done in the previous section 7.3.1, the best way to verify that only the wall velocity u_y tends to zero, we considered a shear section positioned at $x = 2$ m, to verify that the velocity tends to zero at the upper extreme of the geometry. Figures 7.28 and 7.29 graph the trend of both velocities as the y -coordinate varies. In this example, therefore, it is evident how the tangential velocity is determined by the flux, and not imposed equal to zero by the boundary condition. Conversely, the normal velocity, which in this case is u_y , considering the upper extreme of the geometry, is zero. Therefore, the slip-condition is correctly applied, confirming that the boundary conditions of *Bouzidi Bounce Back with Normal* are feasible regardless of geometry.

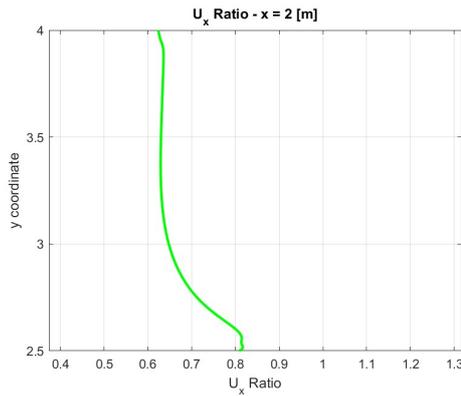


Figure 7.28: Velocity u_x for $x = 2$ m - LBM Bouzidi Bounce Back with Normal Solution

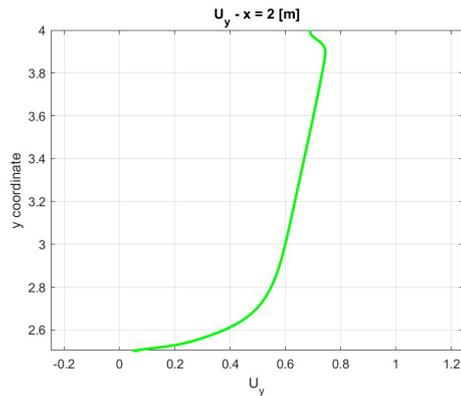


Figure 7.29: Velocity u_y for $x = 2$ m - LBM Bouzidi Bounce Back with Normal Solution

The same procedure was carried out for all other fluid dynamics variables, so from Figure 7.30 to 7.33 the trends for $x = 2$ m as the y-coordinate changes are shown.

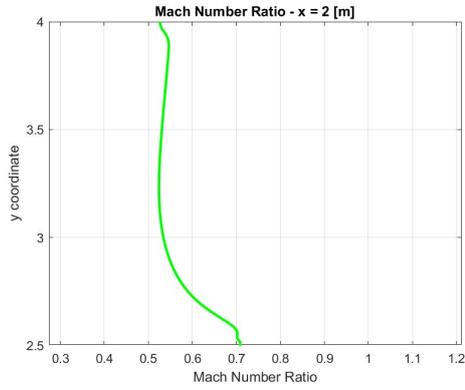


Figure 7.30: Mach for $x = 2$ m - LBM Bouzidi Bounce Back with Normal Solution

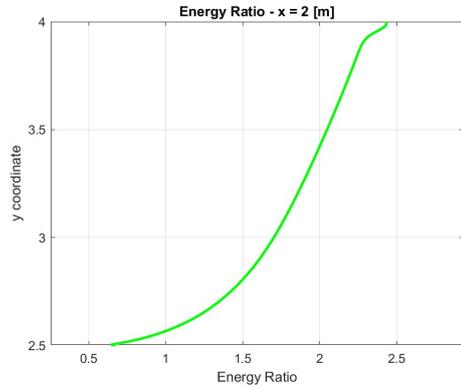


Figure 7.31: Energy for $x = 2$ m - LBM Bouzidi Bounce Back with Normal Solution

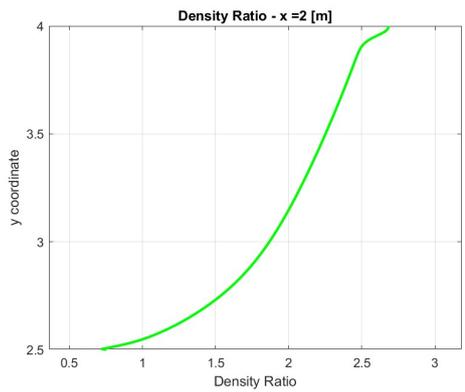


Figure 7.32: Density for $x = 2$ m - LBM Bouzidi Bounce Back with Normal Solution

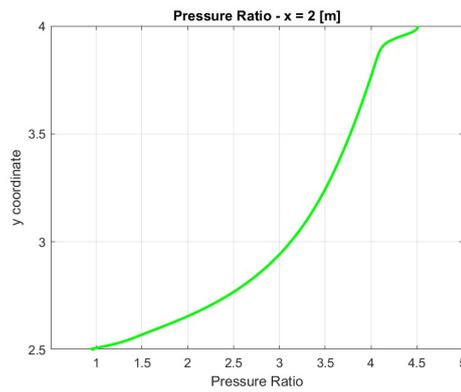


Figure 7.33: Pressure for $x = 2$ m - LBM Bouzidi Bounce Back with Normal Solution

Also in this example to evaluate the accuracy of the obtained solution, and to make a comparison with the analytical solution we graph the trend of the fluid dynamics variables for $y = 2\text{m}$ (i.e., a shear section passing through the stagnation point). The trends of the main fluid dynamics variables ratios as a function of the x coordinate are then shown in Figures 7.34 to 7.40.

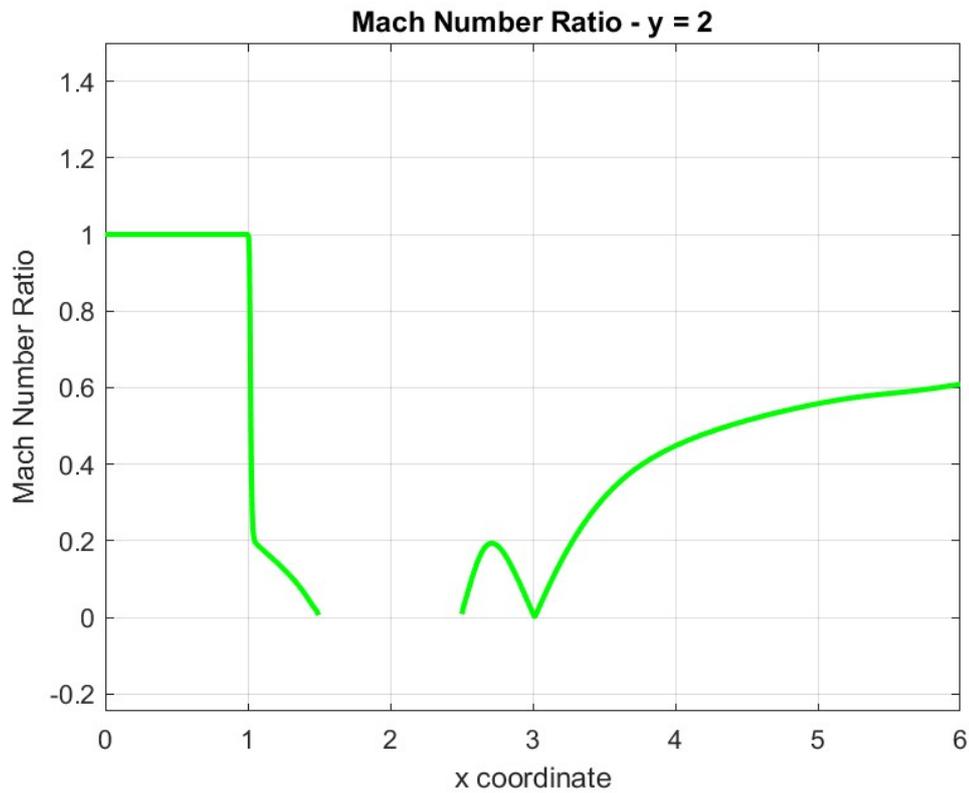


Figure 7.34: Mach number trend for $y = 2$

From Figure 7.34, related to the Mach number, the discontinuity is well evident, as well as how the velocity tends to zero as you approach the geometry (in fact you are approaching a point of isentropic stagnation). Instead, the figures on the next page show typical trends found in fluid dynamics variables downstream of a strong shock wave. In fact, a discontinuous increase in density, pressure, temperature, and energy is observed; in contrast, a drop in total pressure is present.

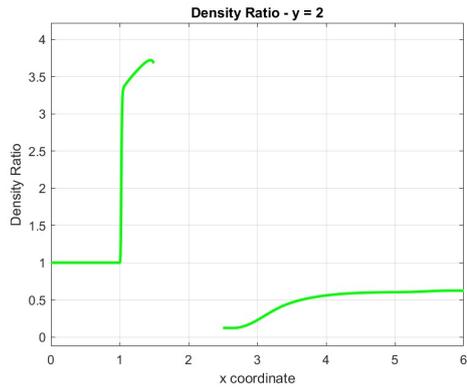


Figure 7.35: Density trend for $y = 2$

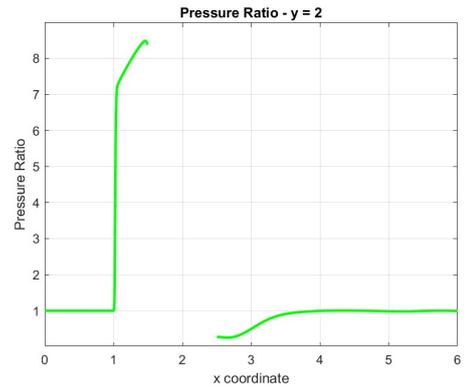


Figure 7.36: Pressure trend for $y = 2$

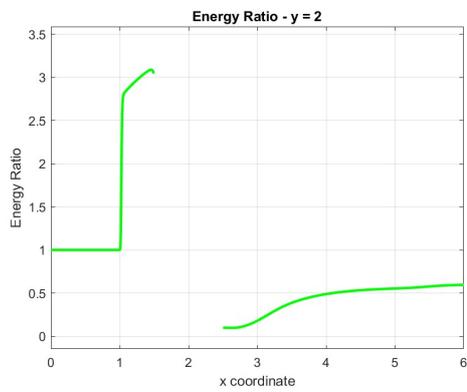


Figure 7.37: Energy trend for $y = 2$

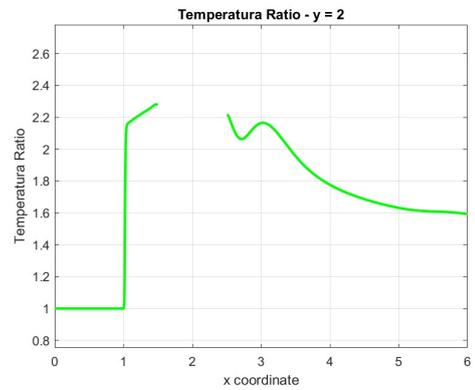


Figure 7.38: Temperature trend for $y = 2$

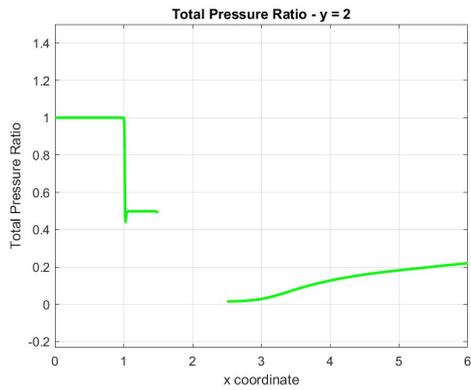


Figure 7.39: Total Pressure trend for $y = 2$

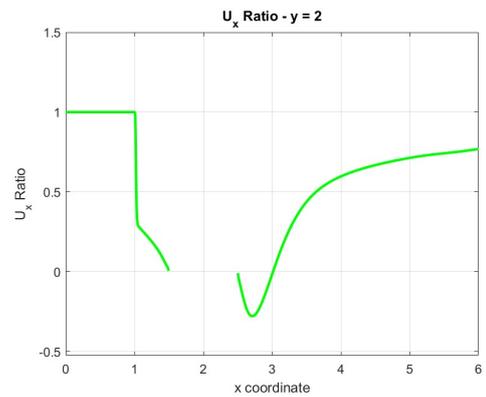


Figure 7.40: u_x trend for $y = 2$

The plotted graphs in Figures 7.34 to 7.40 are critical for evaluating the post-shock values of all fluid dynamics variables. In fact, as already mentioned, the analytical solution is only valid in proximity to the shock wave. In this regard, Table 7.6 shows the post-shock values evaluated with the LBM (applying Bouzidi Bounce Back with Normal boundary conditions).

$Mach_{\text{Post Shock}}$	=	0.508	
$p_{\text{Post Shock}}$	=	7.067	[Pa]
$\rho_{\text{Post Shock}}$	=	4.617	[kg/m ³]
$E_{\text{Post Shock}}$	=	18.942	[J]
$ V _{\text{Post Shock}}$	=	0.743	[m/s]
$c_{\text{Post Shock}}$	=	1.464	[m/s]
$P_{TOT_{\text{Post Shock}}}$	=	8.426	[Pa]
$T_{\text{Post Shock}}$	=	1.8E-04	[K]
Δs	=	202.665	[J/K]

Table 7.6: Results - LBM - Bouzidi Bounce Back with Normal

Table 7.7 shows the relative percentage of errors between the analytical solution and the LBM solution. The results obtained downstream of the shock show a very good correlation between the analytical values and the LBM values, with very limited errors around 1%. From a relative error point of view, there is a slight improvement over the LBM (Bouzidi) solution, but the big improvement is the correct imposition of slip-condition at the wall.

$\text{Error}_{Mach_{LBM}}$	=	1.030	%
$\text{Error}_{p_{LBM}}$	=	0.817	%
$\text{Error}_{T_{LBM}}$	=	4,73E-04	[K]
$\text{Error}_{\rho_{LBM}}$	=	1.072	%
$\text{Error}_{E_{LBM}}$	=	0.957	%
$\text{Error}_{ V _{LBM}}$	=	0.903	%
$\text{Error}_{c_{LBM}}$	=	0.129	%
$\text{Error}_{P_{TOT_{LBM}}}$	=	1.173	%
$\text{Error}_{\Delta s}$	=	1.697	%

Table 7.7: Errors between LBM Bouzidi Bounce Back with Normal - Analytic solutions

Finally, regarding the standoff distance, with the LBM we estimate a δ value of **0.4800 m**. This value differs by 0.07 m from the value expected by Billig which estimated a δ of **0.4074 m**. Thus for the standoff distance, the solution obtained

with Bouzidi Bounce Backs with Normal BCs differs more from Billig's empirical solution than the solution obtained with Bouzidi, as also observed from Figure 7.41.

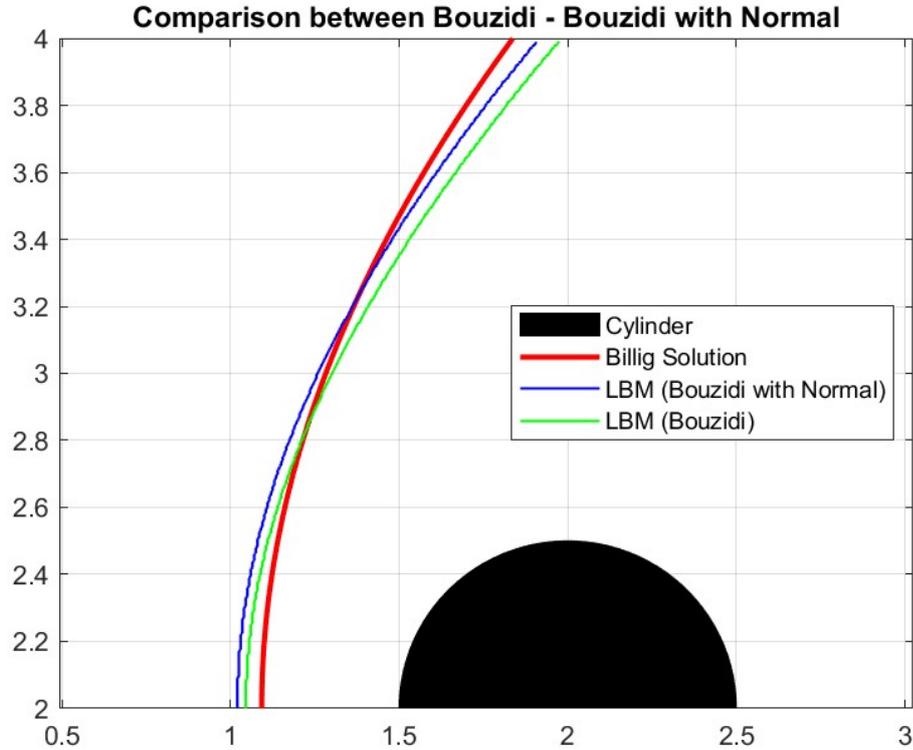


Figure 7.41: Comparison between the two solutions LBM and Billig

7.4 Conclusion

The analysis reported in this last chapter is undoubtedly incomplete, a CFD simulation performed with US3D would also be needed for this test case to better evaluate what was obtained with the LBM. However, the purpose of the LBM analysis on the cylindrical-nosed body was to test the boundary conditions of Bouzidi Bounce Back with Normal in curved geometries. From this point of view, the simulation has to be considered a success, as it appropriately imposed the slip-wall boundary condition.

Chapter 8

Conclusion

This concluding chapter aims to evaluate the results obtained, both the positive, and negative aspects, of both simulations performed. Also evaluating the possible improvements and insights that can be made.

Finally, this chapter also aims to present the future perspectives of the pylbm project, in order to make the software capable of dealing with increasingly complex problems and to make it industrially competitive.

8.1 Achievements

In this thesis, it was shown that the Lattice Boltzmann Method is a method with much potential for the future. Its success is primarily due to:

- The amazing computational efficiency
- The ease of managing complex geometry
- The simplicity of the code
- The ease of parallelizing

The disadvantage addressed in this thesis, however, is the difficulty one has in controlling and analyzing the simulation and then the results. This complexity comes from the great advantage that LBM uses to solve fluid dynamics, namely the particle distribution functions. The main problem that has been addressed in this thesis is the imposition of boundary conditions, which are more complex to impose, compared to conventional CFD, because of the particle distribution functions. Dealing with a boundary condition through particle distribution functions also makes it more complex to understand how to impose the correct boundary condition, precisely because the slip condition, in our case, is imposed indirectly.

The problem encountered in the LBMHYPE project was due to the incorrect imposition of the slip condition. However, this problem was only observed in non-Lattice conforming geometries, i.e., in any geometry that is not aligned with the Cartesian axes, this problem was encountered, such as the wedge.

In this thesis, through the implementation of Bouzidi Bounce Back with Normal boundary conditions, it was possible to solve such problem of the incorrect wall boundary condition for nonconforming geometries with Cartesian axes.

8.1.1 Wedge

In the wedge test case, the improvements brought by the new boundary conditions are incredible. Since the slip condition is correctly imposed, the boundary layer phenomenon is not observed, and the distance between the LE and the shock occurrence is very small, almost comparable to the US3D result. However, due to the high numerical diffusion of the scheme, the solution is not as accurate as that obtained by US3D but consistent with a first-order method. However, we must not forget the formidable strength of the LBM; while the US3D solution is of superior accuracy, the computational speed of the LBM is sensational; a good solution can be obtained in a couple of minutes.

8.1.2 Supersonic cylindrical-nosed body

In this thesis, in the last chapter, a simulation was performed in a blunt body, to verify the applicability of Bouzidi Bounce Back with Normal in a curvilinear geometry. In this regard, the new boundary conditions met theoretical expectations. Considering the generality of the definition of *corrective flux* discussed in Section 5.1 as a function of the normal direction at the wall, the simulation was a success. In fact, by applying the Bouzidi boundary condition, the same problem seen with the wedge in the LBMHYPE project is encountered, whereas, through the imposition of the Bouzidi Bounce Back with Normal at the wall, the slip condition is correctly imposed.

Regarding this simulation, it should be pointed out that a more in-depth analysis would be needed, again making a comparison with the US3D solution.

8.2 Future developments

Nowadays, research is focusing on the development and understanding of multiple aspects of the LBM, such as more flexible and stable collision terms or multi-phase models.

Focusing only on pylbm, however, there are several aspects that can be developed being a very ambitious project.

- **Increase the accuracy of the method:** A first aspect that can be improved in this project is the accuracy of the method. The adopted scheme has a high numerical diffusion, which can be removed by, for example, increasing the relaxation parameter s . As mentioned, however, increasing this parameter results in an unstable scheme, with the occurrence of spurious oscillations at discontinuities. A possible solution would be to adopt specific improvements to locally smooth the spurious oscillations, such as the non-oscillating LBM system presented in [7] for one-dimensional test cases. So a future improvement of pylbm could extend these schemes to solve two-dimensional problems as well, then implement these WENO schemes for D_2Q_{4444} as well.
- **Extension to Navier-Stokes equation:** The main future focus will be to extend pylbm to 2D viscous flows at high Mach numbers. In order to solve 2D Navier-Stokes problems will be necessary to better understand the numerical viscosity of the scheme, in particular, it will be challenging to deal with the viscosity in each LB discretization system. The diffusive scaling must be chosen in order to be consistent with a parabolic system, such as the complete Navier-Stokes system.
- **Hypersonic regime:** Once it is extended to Navier-Stokes, a further complication will be the extension to the hypersonic regime. The numerical limitations given by the compressible field, which have already been resolved, are added to the limitations already mentioned in section 3.2. The set values for the heat capacity ratio γ and the Prandtl number ($Pr = 1$) are two additional non-numerical restrictions that belong to LBM.
 - According to the first restriction, the thermal diffusivity coefficient cannot be freely set while the dynamic viscosity is fixed, and vice versa.
 - The second limitation is closely related to the number of internal degrees of freedom of gas molecules. Indeed from the γ relationship, if only translational DOFs are considered, as in the case of conventional LBM, this limits the field of use of the method, applicable accurately to monoatomic gases only. Rotational DOFs must be included at least when defining the internal energy or the total energy in order to reproduce realistic flows.

- **Multi-phase models:** Another crucial aspect of hypersonic after high temperatures is dealing with multiphase flows, which are a problem to be dealt with not only with pylbm but also with LBM in general, in fact, much research is focused on this aspect.
- **Treatment of any external sources:** The pylbm's handling of source terms, such as heat sink/sources in the energy equation, is yet another area to be developed.
- **Velocity stencil extension in 2D and 3D:** With validations on academic test cases in 1D and 2D, the entire vectorial schemes have demonstrated a good potential to mimic compressible flows with high Mach numbers. The study on vectorial schemes will continue by increasing the stencil of velocities, both in 2D but especially in 3D.
 - In 2D, the stencil can be extended by considering up to 9 velocities (i.e., D_2Q_9). This on the one hand would result in better accuracy, especially for turbulent flows, not well taken into account with only 4 velocities, but on the other hand, would result in an extension of the conditions Bouzidi Bounce Back with Normal. In fact, the new boundary conditions implemented in this thesis are applicable to D_2Q_4 only, as they exploit the analogy between the four cell faces and the four velocities. Therefore, it is necessary to generalize this concept. One idea would be to change the shape of the elementary latex cell, but this would lead to problems in the future, extension in 3D especially, or another option would be to properly consider the components of the corrective fluxes in the Bouzidi Bounce Back with Normal BCs.
 - The real future development of pylbm, in order to consider geometries more and more like reality, and the engineering industry is the extension to the third dimension. In 3D, the stencil can be extended by considering seven, nineteen, and twenty-seven velocities schemes (i.e., D_3Q_7 , D_3Q_{19} or D_3Q_{27}). In 3D these schemes will need to be tested and compared in terms of quality and computational cost, but more importantly in terms of stability.

Appendix A

Euler equations

The extensive studies conducted by numerous scientists between the 18th and 19th centuries led to the development of equations that describe the motion of fluids. The first to derive a macroscopic set of equations for the specific case of inviscid fluid flows was Euler in 1755 [138]. Euler's theory is based on two key principles:

- **Conservation of mass**

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (\text{A.1})$$

- **Conservation of momentum**

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}^2) = \rho \mathbf{g} - \nabla p \quad (\text{A.2})$$

where respectively ∇ is the physical space's gradient operator, ρ the density, $\rho \mathbf{u}$ the momentum, $\rho \mathbf{g}$ the external force, p the pressure, and $\rho \mathbf{u}^2$ the second-order tensor linked to the momentum convection. Assuming by hypothesis that the external force is known, the system of Euler equations A.1 and A.2 contains three unknowns (density ρ , momentum $\rho \mathbf{u}$, and pressure p), so the system is open. The system can be solved, by providing an evolutionary constitutive law for the pressure¹. In his [138] proposal, Euler suggested that pressure should just depend on density $p = p(\rho)$ (i.e. barotropic fluid) to obtain a set of equations that are well posed. However, Euler was only able to analyze the motion of non-viscous (inviscid) and non-heat-conducting flows in his study.

The absence of shear stresses in the momentum conservation equation A.2 is related to the restriction of non-viscous flows. While the non-heat-conducting flow restriction results from two main causes: firstly, the equations provided above do not

¹Clearly, like any system of PDEs, boundary and initial conditions are required

account for the energy conservation principle, and secondly, according to Euler's theory, pressure is not related to temperature through the equation of state.

Navier made the first adaptation to the Euler momentum equation for incompressible flows in 1821 [139]. Stokes modified the compressible momentum equation in 1845 using the framework of fluid mechanics [140], obtaining the Navier-Stokes equation's contemporary version.

Appendix B

Python code for new boundary conditions

This appendix shows the two classes implemented within the pylbm library. Both scripts implement the Bouzidi Bounce Back With Normal boundary conditions (in the x- and y-direction, respectively). Both classes were added to the library's existing ones in the *boundary.py* script, which can be found in the pylbm library [4].

Listing B.1: BouzidiBounceBackWithNormalX class

```
49 class BouzidiBounceBackWithNormalX(pylbm.bc.BouzidiBounceBack):
50     def update_feq(self, simulation):
51         super(BouzidiBounceBackWithNormalX,
52               self).update_feq(simulation)
53         self.update_flux(simulation)
54
55     def update_flux(self, simulation):
56         gamma = 1.4
57
58         m = simulation.m_halo
59
60         k = self.istore[:, 0]
61         v = self.stencil.get_all_velocities()
62
63         self.flux_indices = self.istore[:, 1:] + v[k]
64
65         RHO = m[rho][self.flux_indices[:, 0], self.flux_indices[:, 1]]
66         QX = m[qx][self.flux_indices[:, 0], self.flux_indices[:, 1]]
67         QY = m[qy][self.flux_indices[:, 0], self.flux_indices[:, 1]]
68         E_ = m[E][self.flux_indices[:, 0], self.flux_indices[:, 1]]
69
70
```

```

71     ux = QX/RHO
72     uy = QY/RHO
73
74     p = (gamma - 1) * (E_ - .5 * (QX**2 + QY**2)/RHO)
75
76     p_star = p + np.sqrt(gamma * p * RHO) * (ux*self.normal[:, 0]
77         + uy*self.normal[:, 1])
78
79     phi_n = p_star*self.normal[:, 0]
80
81     mask = self.distance < .5
82
83     not_mask = np.logical_not(mask)
84
85     self.flux = np.zeros_like(phi_n)
86
87     self.flux[mask] = phi_n[mask] / lambda
88     self.flux[not_mask] = phi_n[not_mask]
89         / (2 * self.distance[not_mask] * lambda)
90
91     def _get_args(self, ff):
92         dim = len(ff.nspace)
93
94         nx = ff.nspace[0]
95
96         if dim > 1:
97             ny = ff.nspace[1]
98
99         if dim > 2:
100             nz = ff.nspace[2]
101
102         f = ff.array
103         fcopy = ff.array.copy()
104
105         for i in range(len(self.iloader)):
106             exec('iloader{i} = self.iloader[{i}]'.format(i=i))
107
108         istore = self.istore
109         rhs = self.rhs
110         flux = self.flux
111
112         if hasattr(self, 's'):
113             dist = self.s
114
115         ncond = istore.shape[0]
116         return locals()
117
118
119

```

```

120
121 def generate(self, sorder):
122     from pylbm.generator import For
123     from pylbm.symbolic import nx, ny, nz, indexed, ix
124
125     ns = int(self.stencil.nv_ptr[-1])
126
127     dim = self.stencil.dim
128
129     istore, iload, ncond = self._get_istore_iloading_symb(dim)
130     rhs, dist, _ = self._get_rhs_dist_symb(ncond)
131     flux = IndexedBase('flux', [ncond])
132
133     idx = Idx(ix, (0, ncond))
134
135     fstore = indexed('f', [ns, nx, ny, nz], index=[istore[idx, k]
136             for k in range(dim + 1)], priority=sorder)
137
138     fload0 = indexed('fcopy', [ns, nx, ny, nz], index=[iload[0][idx, k]
139             for k in range(dim + 1)], priority=sorder)
140
141     fload1 = indexed('fcopy', [ns, nx, ny, nz], index=[iload[1][idx, k]
142             for k in range(dim + 1)], priority=sorder)
143
144     self.generator.add_routine(('Bouzidi_bounce_back_normalX',
145             For(idx, Eq(fstore, dist[idx]* fload0 + (1 - dist[idx])
146             * fload1 + rhs[idx] - flux[idx])))
147
148
149 @property
150 def function(self):
151     return self.generator.module.Bouzidi_bounce_back_normalX

```

Listing B.2: BouzidiBounceBackWithNormalY class

```

152 class BouzidiBounceBackWithNormalY(BouzidiBounceBackWithNormalX):
153     def update_flux(self, simulation):
154         gamma = 1.4
155
156         m = simulation.m_halo
157
158         k = self.istore[:, 0]
159         v = self.stencil.get_all_velocities()
160
161         self.flux_indices = self.istore[:, 1:] + v[k]
162
163         RHO = m[rho][self.flux_indices[:, 0], self.flux_indices[:, 1]]
164
165         QX = m[qx][self.flux_indices[:, 0], self.flux_indices[:, 1]]
166
167         QY = m[qy][self.flux_indices[:, 0], self.flux_indices[:, 1]]
168
169         E_ = m[E][self.flux_indices[:, 0], self.flux_indices[:, 1]]
170
171         ux = QX/RHO
172         uy = QY/RHO
173
174         p = (gamma - 1) * (E_ - .5 * (QX**2 + QY**2)/RHO)
175
176
177         p_star = p + np.sqrt(gamma * p * RHO) * (ux*self.normal[:, 0]
178             + uy*self.normal[:, 1])
179
180         phi_n = p_star*self.normal[:, 1]
181
182         mask = self.distance < .5
183         not_mask = np.logical_not(mask)
184
185         self.flux = np.zeros_like(phi_n)
186
187         self.flux[mask] = phi_n[mask] / lambda
188
189         self.flux[not_mask] = phi_n[not_mask]
190             / (2 * self.distance[not_mask] * lambda)
191
192
193
194
195
196
197
198
199

```

```

200 def generate(self, sorder):
201     from pylbm.generator import For
202     from pylbm.symbolic import nx, ny, nz, indexed, ix
203
204     ns = int(self.stencil.nv_ptr[-1])
205     dim = self.stencil.dim
206
207     istore, iload, ncond = self._get_istore_iloading_symb(dim)
208     rhs, dist, _ = self._get_rhs_dist_symb(ncond)
209     flux = IndexedBase('flux', [ncond])
210
211
212     idx = Idx(ix, (0, ncond))
213     fstore = indexed('f', [ns, nx, ny, nz], index=[istore[idx, k]
214             for k in range(dim + 1)], priority=sorder)
215     fload0 = indexed('fcopy', [ns, nx, ny, nz], index=[iload[0][idx, k]
216             for k in range(dim + 1)], priority=sorder)
217     fload1 = indexed('fcopy', [ns, nx, ny, nz], index=[iload[1][idx, k]
218             for k in range(dim + 1)], priority=sorder)
219
220
221     self.generator.add_routine(('Bouzidi_bounce_back_normalY',
222             For(idx, Eq(fstore, dist[idx] * fload0 + (1 - dist[idx])
223             * fload1 + rhs[idx] - flux[idx])))
224
225     @property
226     def function(self):
227         return self.generator.module.Bouzidi_bounce_back_normalY

```

Appendix C

Wedge Results

This appendix provides pictures and tables of the results obtained.

C.1 US3D Results

This section provides pictures and tables of the results obtained with US3D.

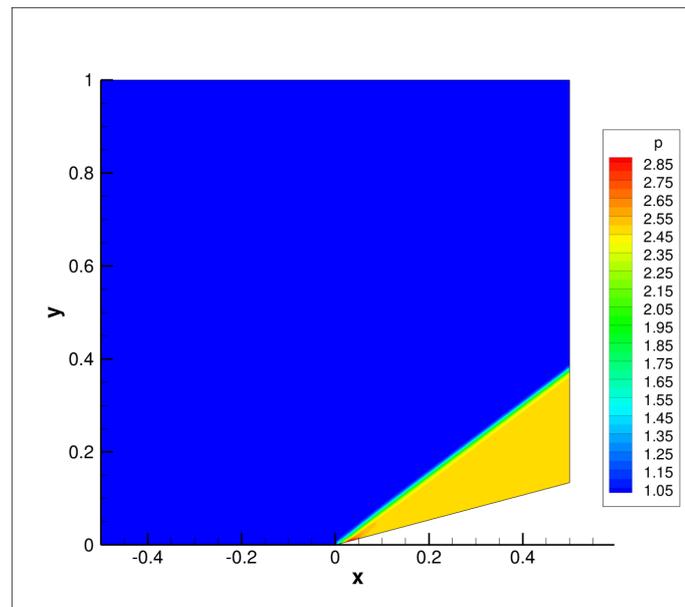


Figure C.1: CFD pressure field for the Mach 2.5 supersonic wedge on the 201x201 grid

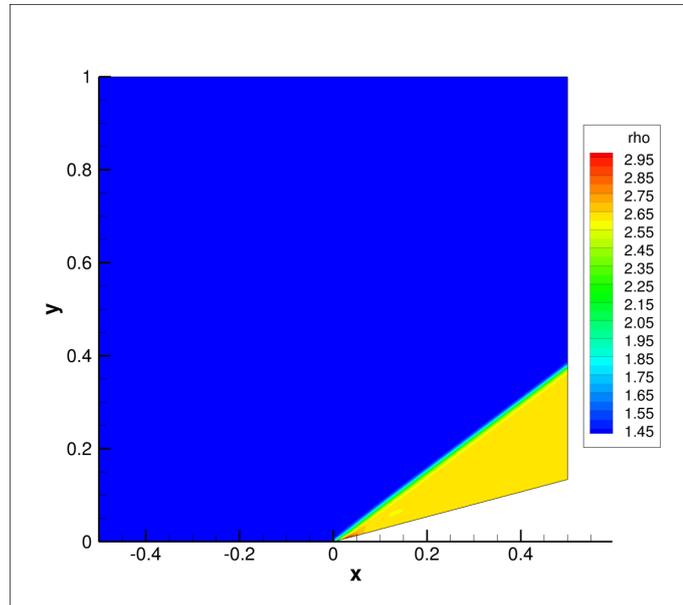


Figure C.2: CFD density field for the Mach 2.5 supersonic wedge on the 201x201 grid

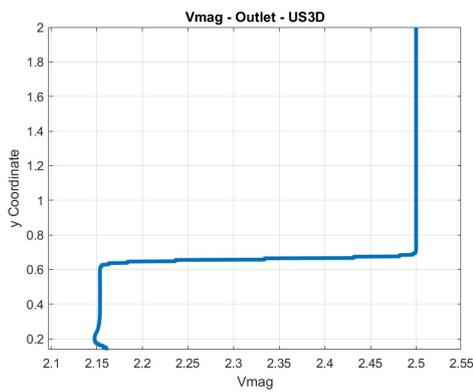


Figure C.3: Velocity $|V|$ values for $x = 0.5 \text{ m}$ - US3D

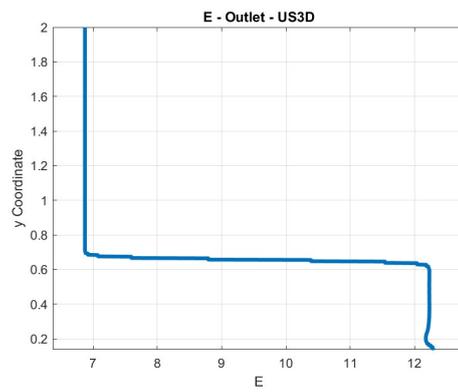


Figure C.4: Energy E values for $x = 0.5 \text{ m}$ - US3D

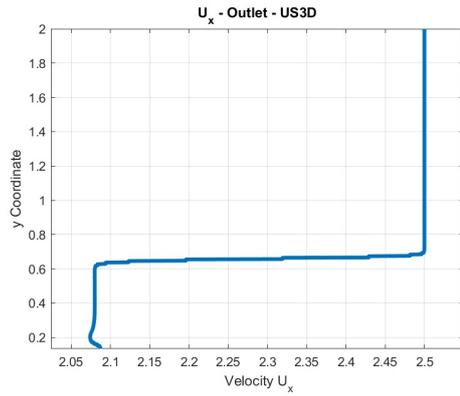


Figure C.5: Velocity u_x values for $x = 0.5 \text{ m}$ - US3D

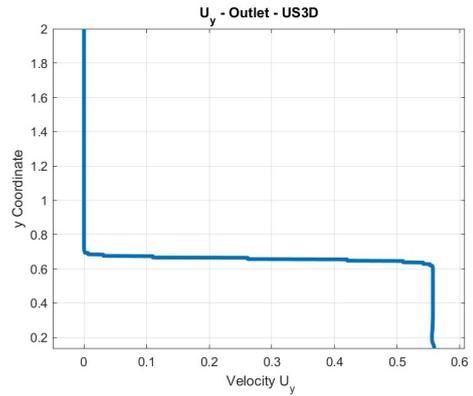


Figure C.6: Velocity u_y values for $x = 0.5 \text{ m}$ - US3D

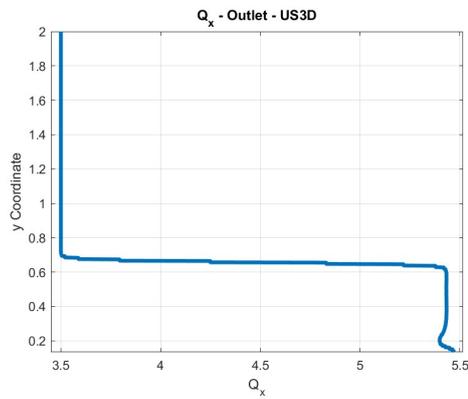


Figure C.7: Momentum Q_x values for $x = 0.5 \text{ m}$ - US3D

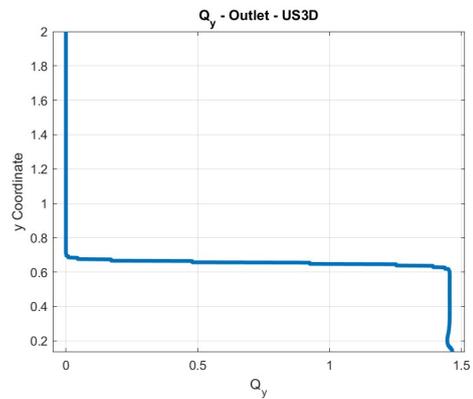


Figure C.8: Momentum Q_y values for $x = 0.5 \text{ m}$ - US3D

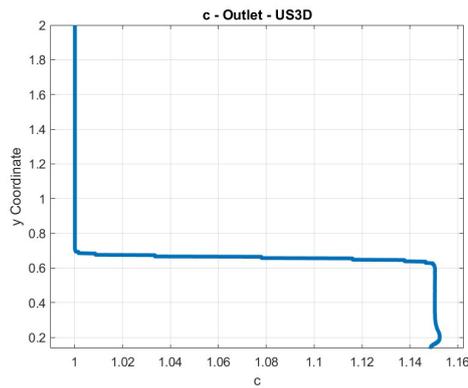


Figure C.9: Speed of Sound c values for $x = 0.5 \text{ m}$ - US3D

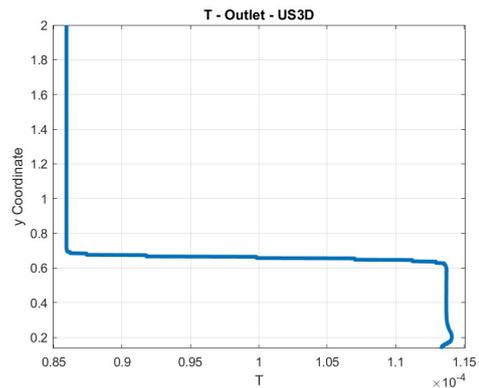


Figure C.10: Temperature T values for $x = 0.5 \text{ m}$ - US3D

Appendix D

Second-order Equivalent Equations

The corresponding equations up to the second order, obtained using the expansion of Dubois [79] [80] are shown in this appendix. The equivalent PDE equation is as follows:

$$\frac{d}{dx}F_x + \frac{d}{dy}F_y + \frac{d}{dt}U = \frac{\partial}{\partial x}B_{xx}\frac{d}{dx}U + \frac{\partial}{\partial x}B_{xy}\frac{d}{dy}U + \frac{\partial}{\partial y}B_{yx}\frac{d}{dx}U + \frac{\partial}{\partial y}B_{yy}\frac{d}{dy}U \quad (\text{D.1})$$

For each of the 4 conserved moments (ρ , q_x, q_y , and E) there will be for the first order the following relations to calculate the coefficients of D.1:

$$\begin{aligned} F_x^\rho &= q_x \\ F_x^{q_x} &= \frac{2E\gamma\rho - 2E\rho - \gamma q_x^2 - \gamma q_y^2 + 3q_x^2 + q_y^2}{2\rho} \\ F_x^{q_y} &= \frac{q_x q_y}{\rho} \\ F_x^E &= \frac{q_x (2E\gamma\rho - \gamma q_x^2 - \gamma q_y^2 + q_x^2 + q_y^2)}{2\rho^2} \\ F_y^\rho &= q_y \\ F_y^{q_x} &= \frac{q_x q_y}{\rho} \\ F_y^{q_y} &= \frac{2E\gamma\rho - 2E\rho - \gamma q_x^2 - \gamma q_y^2 + q_x^2 + 3q_y^2}{2\rho} \\ F_y^E &= \frac{q_y (2E\gamma\rho - \gamma q_x^2 - \gamma q_y^2 + q_x^2 + q_y^2)}{2\rho^2} \end{aligned} \quad (\text{D.2})$$

As for the second order, the following equations are obtained:

$$\begin{aligned}
 B_{xx}^{\rho,\rho} &= -\Delta_t \left(\frac{\gamma q_x^2 \sigma_\rho}{2\rho^2} + \frac{\gamma q_y^2 \sigma_\rho}{2\rho^2} - \frac{\lambda^2 \sigma_\rho}{2} - \frac{q_x^2 \sigma_\rho}{\rho^2} - \frac{q_y^2 \sigma_\rho}{\rho^2} \right) \\
 B_{zx}^{\rho,q_x} &= -\Delta_t \left(-\frac{\gamma q_x \sigma_\rho}{\rho} + \frac{2q_x \sigma_\rho}{\rho} \right) \\
 B_{xx}^{\rho,\omega_\mu} &= -\Delta_t \left(-\frac{\gamma q_y \sigma_\rho}{\rho} + \frac{2q_y \sigma_\rho}{\rho} \right) \\
 B_{xx}^{\rho,E} &= -\Delta_t (\gamma \sigma_\rho - \sigma_\rho) \\
 B_{xz}^{q_x,q_2} &= -\Delta_t \left(-\frac{3\gamma q_x q_y \sigma_u}{\rho^2} + \frac{3q_x q_y \sigma_u}{\rho^2} \right) \\
 B_{xx}^{q_z,B} &= -\Delta_t \left(\frac{3\gamma q_z \sigma_u}{\rho} - \frac{3q_z \sigma_u}{\rho} \right) \\
 B_{xx}^{h_1,\rho} &= -\Delta_t \left(\frac{\gamma q_x^2 q_y \sigma_u}{2\rho^3} + \frac{\gamma q_y^3 \sigma_u}{2\rho^3} - \frac{5q_z^2 q_y \sigma_u}{2\rho^3} - \frac{q_y^3 \sigma_u}{2\rho^3} \right) \\
 B_{zz}^{q_y,q_x} &= -\Delta_t \left(-\frac{\gamma q_x q_y \sigma_u}{\rho^2} + \frac{3q_x q_y \sigma_u}{\rho^2} \right) \\
 B_{xx}^{\beta_1,\omega_u} &= -\Delta_t \left(-\frac{\gamma q_y^2 \sigma_u}{\rho^2} - \frac{\lambda^2 \sigma_u}{2} + \frac{q_x^2 \sigma_u}{\rho^2} + \frac{q_y^2 \sigma_u}{\rho^2} \right) \\
 B_{xx}^{q_x^2 E} &= -\Delta_t \left(\frac{\gamma q_y \sigma_u}{\rho} - \frac{q_y \sigma_u}{\rho} \right) \\
 B_{xx}^{E,q_x} &= -\Delta_t \left(\frac{2E\gamma q_x \sigma_p}{\rho^2} - \frac{7\gamma q_z^3 \sigma_p}{2\rho^3} - \frac{3\gamma q_z q_y^2 \sigma_p}{2\rho^3} + \frac{7q_x^3 \sigma_p}{2\rho^3} + \frac{3q_x q_y^2 \sigma_p}{2\rho^3} \right) \\
 B_{xy}^{\rho,\rho} &= \frac{\Delta_t q_x q_y \sigma_\rho}{\rho^2} \\
 B_{xy}^{\rho,q_x} &= -\frac{\Delta_t q_y \sigma_\rho}{\rho} \\
 B_{xy}^{\rho,\omega_y} &= -\frac{\Delta_t q_x \sigma_\rho}{\rho} \\
 B_{xy}^{\rho,E} &= 0 \\
 B_{xy}^{q_{xy},\rho} &= -\Delta_t \left(-\frac{E\gamma^2 q_y \sigma_u}{\rho^2} + \frac{E\gamma q_y \sigma_u}{\rho^2} + \frac{\gamma^2 q_z^2 q_y \sigma_u}{2\rho^3} + \frac{\gamma^2 q_y^3 \sigma_u}{2\rho^3} - \frac{5q_x^2 q_y \sigma_u}{2\rho^3} - \frac{q_y^3 \sigma_u}{2\rho^3} \right) \\
 B_{xy}^{q_z,q_z} &= -\Delta_t \left(-\frac{\gamma q_x q_y \sigma_u}{\rho^2} + \frac{3q_x q_y \sigma_u}{\rho^2} \right) \\
 B_{xy}^{q_z,E} &= -\Delta_t \left(\frac{\gamma q_y \sigma_u}{\rho} - \frac{q_y \sigma_u}{\rho} \right)
 \end{aligned}$$

$$\begin{aligned}
 B_{xy}^{tn,\rho} &= -\Delta_t \left(\frac{\gamma q_x^3 \sigma_u}{2\rho^3} + \frac{\gamma q_z q_y^2 \sigma_u}{2\rho^3} - \frac{q_x^3 \sigma_u}{2\rho^3} - \frac{5q_x q_y^2 \sigma_u}{2\rho^3} \right) \\
 B_{xy}^{q_y, q_x} &= -\Delta_t \left(-\frac{\gamma q_x^2 \sigma_u}{\rho^2} + \frac{q_x^2 \sigma_u}{\rho^2} + \frac{q_y^2 \sigma_u}{\rho^2} \right) \\
 B_{1y}^{q_y, q_u} &= -\Delta_t \left(-\frac{\gamma q_x q_y \sigma_u}{\rho^2} + \frac{3q_x q_y \sigma_u}{\rho^2} \right) \\
 B_{xy}^{q_x, E} &= -\Delta_t \left(\frac{\gamma q_x \sigma_u}{\rho} - \frac{q_x \sigma_u}{\rho} \right) \\
 B_{xy}^{E, q_z} &= -\Delta_t \left(\frac{E \gamma q_y \sigma_p}{\rho^2} - \frac{5\gamma q_z^2 q_y \sigma_p}{2\rho^3} - \frac{\gamma q_y^3 \sigma_p}{2\rho^3} + \frac{5q_x^2 q_y \sigma_p}{2\rho^3} + \frac{q_y^3 \sigma_p}{2\rho^3} \right) \\
 B_{xy}^{E, y} &= -\Delta_t \left(\frac{E \gamma^2 q_x \sigma_p}{\rho^2} - \frac{\gamma^2 q_z^3 \sigma_p}{2\rho^3} - \frac{\gamma^2 q_x q_y^2 \sigma_p}{2\rho^3} - \frac{2\gamma q_z q_y^2 \sigma_p}{\rho^3} + \frac{q_x^3 \sigma_p}{2\rho^3} + \frac{5q_x q_y^2 \sigma_p}{2\rho^3} \right) \\
 B_{xy}^{E, E} &= -\Delta_t \left(\frac{2\gamma q_z q_y \sigma_p}{\rho^2} - \frac{q_z q_y \sigma_p}{\rho^2} \right) \\
 B_{yz}^{\rho, \rho} &= \frac{\Delta_t q_x q_y \sigma_\rho}{\rho^2} \\
 B_{3x}^{\rho, q_x} &= -\frac{\Delta_t q_y \sigma_\rho}{\rho} \\
 B_{y\alpha}^{\rho, \omega_2} &= -\frac{\Delta_t q_z \sigma_\rho}{\rho} \\
 B_{yx}^{\rho, E} &= 0 \\
 B_{yx}^{q_z, \rho} &= -\Delta_t \left(\frac{\gamma q_x^2 q_y \sigma_u}{2\rho^3} + \frac{\gamma q_y^3 \sigma_u}{2\rho^3} - \frac{5q_z^2 q_y \sigma_u}{2\rho^3} - \frac{q_y^3 \sigma_u}{2\rho^3} \right) \\
 B_{yx}^{q_x, q_x} &= -\Delta_t \left(-\frac{\gamma q_x q_y \sigma_u}{\rho^2} + \frac{3q_x q_y \sigma_u}{\rho^2} \right) \\
 B_{yx}^{q_x, \omega_u} &= -\Delta_t \left(-\frac{\gamma q_y^2 \sigma_u}{\rho^2} + \frac{q_z^2 \sigma_u}{\rho^2} + \frac{q_y^2 \sigma_u}{\rho^2} \right) \\
 B_{yr}^{q_y, E} &= -\Delta_t \left(\frac{\gamma q_y \sigma_u}{\rho} - \frac{q_y \sigma_u}{\rho} \right) \\
 B_{yz}^{q_2, \rho} &= -\Delta_t \left(-\frac{E \gamma^2 q_x \sigma_u}{\rho^2} + \frac{E \gamma q_x \sigma_u}{\rho^2} + \frac{\gamma^2 q_z^3 \sigma_u}{2\rho^3} + \frac{\gamma^2 q_x q_y^2 \sigma_u}{2\rho^3} - \frac{q_x^3 \sigma_u}{2\rho^3} - \frac{5q_x q_y^2 \sigma_u}{2\rho^3} \right) \\
 B_{yz}^{q_h, q_u} &= -\Delta_t \left(-\frac{\gamma q_x q_y \sigma_u}{\rho^2} + \frac{3q_x q_y \sigma_u}{\rho^2} \right) \\
 B_{yz}^{q_y, E} &= -\Delta_t \left(\frac{\gamma q_x \sigma_u}{\rho} - \frac{q_x \sigma_u}{\rho} \right)
 \end{aligned}$$

$$\begin{aligned}
 B_{yx}^{E,q_z} &= -\Delta_t \left(\frac{E\gamma^2 q_y \sigma_p}{\rho^2} - \frac{\gamma^2 q_z^2 q_y \sigma_p}{2\rho^3} - \frac{\gamma^2 q_y^3 \sigma_p}{2\rho^3} - \frac{2\gamma q_z^2 q_y \sigma_p}{\rho^3} + \frac{5q_z^2 q_y \sigma_p}{2\rho^3} + \frac{q_y^3 \sigma_p}{2\rho^3} \right) \\
 B_{yz}^{E,\omega_z} &= -\Delta_t \left(\frac{E\gamma q_z \sigma_p}{\rho^2} - \frac{\gamma q_x^3 \sigma_p}{2\rho^3} - \frac{5\gamma q_x q_z^2 \sigma_p}{2\rho^3} + \frac{q_x^3 \sigma_p}{2\rho^3} + \frac{5q_z q_y^2 \sigma_p}{2\rho^3} \right) \\
 B_{yx}^{E,E} &= -\Delta_t \left(\frac{2\gamma q_z q_y \sigma_p}{\rho^2} - \frac{q_z q_y \sigma_p}{\rho^2} \right) \\
 B_{yy}^{\rho,\rho} &= -\Delta_t \left(\frac{\gamma q_x^2 \sigma_\rho}{2\rho^2} + \frac{\gamma q_y^2 \sigma_\rho}{2\rho^2} - \frac{\lambda^2 \sigma_\rho}{2} - \frac{q_x^2 \sigma_\rho}{\rho^2} - \frac{q_y^2 \sigma_\rho}{\rho^2} \right) \\
 B_{yy}^{\rho,q_x} &= -\Delta_t \left(-\frac{\gamma q_x \sigma_\rho}{\rho} + \frac{2q_x \sigma_\rho}{\rho} \right) \\
 B_{yy}^{\rho,\omega_y} &= -\Delta_t \left(-\frac{\gamma q_y \sigma_\rho}{\rho} + \frac{2q_y \sigma_\rho}{\rho} \right) \\
 B_{yy}^{\rho,E} &= -\Delta_t (\gamma \sigma_\rho - \sigma_\rho) \\
 B_{yy}^{q_y y, \rho} &= -\Delta_t \left(\frac{\gamma q_x^3 \sigma_u}{2\rho^3} + \frac{\gamma q_z q_y^2 \sigma_u}{2\rho^3} - \frac{q_z^3 \sigma_u}{2\rho^3} - \frac{5q_x q_y^2 \sigma_u}{2\rho^3} \right) \\
 B_{yy^2}^{q_z, q_z} &= -\Delta_t \left(-\frac{\gamma q_z^2 \sigma_u}{\rho^2} - \frac{\lambda^2 \sigma_u}{2} + \frac{q_x^2 \sigma_u}{\rho^2} + \frac{q_y^2 \sigma_u}{\rho^2} \right) \\
 B_{yy}^{q_x, \omega_u} &= -\Delta_t \left(-\frac{\gamma q_x q_y \sigma_u}{\rho^2} + \frac{3q_x q_y \sigma_u}{\rho^2} \right) \\
 B_{yy}^{q_z, B_y} &= -\Delta_t \left(\frac{\gamma q_x \sigma_u}{\rho} - \frac{q_x \sigma_u}{\rho} \right) \\
 B_{yy}^{q_y y, q_x} &= -\Delta_t \left(-\frac{3\gamma q_x q_y \sigma_u}{\rho^2} + \frac{3q_x q_y \sigma_u}{\rho^2} \right) \\
 B_{yy}^{q_y, E} &= -\Delta_t \left(\frac{3\gamma q_y \sigma_u}{\rho} - \frac{3q_y \sigma_u}{\rho} \right) \\
 B_{yy}^{E, \omega_y} &= -\Delta_t \left(\frac{2E\gamma q_y \sigma_p}{\rho^2} - \frac{3\gamma q_1^2 q_y \sigma_p}{2\rho^3} - \frac{7\gamma q_y^3 \sigma_p}{2\rho^3} + \frac{3q_z^2 q_y \sigma_p}{2\rho^3} + \frac{7q_y^3 \sigma_p}{2\rho^3} \right)
 \end{aligned}$$

$$B_{xz}^{q_x, \rho} = -\Delta_t \left(-\frac{E\gamma^2 q_x \sigma_u}{\rho^2} + \frac{E\gamma q_x \sigma_u}{\rho^2} + \frac{\gamma^2 q_x^3 \sigma_u}{2\rho^3} + \frac{\gamma^2 q_x q_y^2 \sigma_u}{2\rho^3} + \frac{\gamma q_z^3 \sigma_u}{\rho^3} \right. \\ \left. + \frac{\gamma q_x q_y^2 \sigma_u}{\rho^3} - \frac{7q_x^3 \sigma_u}{2\rho^3} - \frac{3q_x q_y^2 \sigma_u}{2\rho^3} \right)$$

$$B_{xx}^{q_u q_x} = -\Delta_t \left(\left(\frac{E\gamma^2 \sigma_u}{\rho} - \frac{E\gamma \sigma_u}{\rho} - \frac{\gamma^2 q_x^2 \sigma_u}{2\rho^2} - \frac{\gamma^2 q_y^2 \sigma_u}{2\rho^2} - \frac{5\gamma q_x^2 \sigma_u}{2\rho^2} \right. \right. \\ \left. \left. + \frac{\gamma q_y^2 \sigma_u}{2\rho^2} - \frac{\lambda^2 \sigma_u}{2} + \frac{6q_z^2 \sigma_u}{\rho^2} \right)$$

$$B_{xx}^{E_s, \rho} = -\Delta_t \left(-\frac{E\gamma^2 q_x^2 \sigma_p}{2\rho^3} + \frac{E\gamma^2 q_y^2 \sigma_p}{2\rho^3} - \frac{3E\gamma q_x^2 \sigma_p}{2\rho^3} - \frac{E\gamma q_y^2 \sigma_p}{2\rho^3} + \frac{\gamma^2 q_x^4 \sigma_p}{4\rho^4} - \frac{\gamma^2 q_y^4 \sigma_p}{4\rho^4} \right. \\ \left. + \frac{2\gamma q_x^4 \sigma_p}{\rho^4} + \frac{5\gamma q_x^2 q_y^2 \sigma_p}{2\rho^4} + \frac{\gamma q_y^4 \sigma_p}{2\rho^4} - \frac{9q_x^4 \sigma_p}{4\rho^4} - \frac{5q_x^2 q_y^2 \sigma_p}{2\rho^4} - \frac{q_y^4 \sigma_p}{4\rho^4} \right)$$

$$B_{xx}^{E, q_p} = -\Delta_t \left(-\frac{E\gamma^2 q_y \sigma_p}{\rho^2} + \frac{E\gamma q_y \sigma_p}{\rho^2} + \frac{\gamma^2 q_x^2 q_y \sigma_p}{2\rho^3} + \frac{\gamma^2 q_y^3 \sigma_p}{2\rho^3} - \frac{3\gamma q_x^2 q_y \sigma_p}{\rho^3} \right. \\ \left. - \frac{\gamma q_y^3 \sigma_p}{\rho^3} + \frac{5q_x^2 q_y \sigma_p}{2\rho^3} + \frac{q_y^3 \sigma_p}{2\rho^3} \right)$$

$$B_{xx}^{E, E} = -\Delta_t \left(\frac{E\gamma^2 \sigma_p}{\rho} - \frac{E\gamma \sigma_p}{\rho} - \frac{\gamma^2 q_z^2 \sigma_p}{2\rho^2} - \frac{\gamma^2 q_y^2 \sigma_p}{2\rho^2} + \frac{3\gamma q_x^2 \sigma_p}{\rho^2} + \frac{\gamma q_y^2 \sigma_p}{\rho^2} - \frac{\lambda^2 \sigma_p}{2} \right. \\ \left. - \frac{3q_x^2 \sigma_p}{2\rho^2} - \frac{q_y^2 \sigma_p}{2\rho^2} \right)$$

$$B_{xy}^{q_z, \sigma_v} = -\Delta_t \left(\frac{E\gamma^2 \sigma_u}{\rho} - \frac{E\gamma \sigma_u}{\rho} - \frac{\gamma^2 q_x^2 \sigma_u}{2\rho^2} - \frac{\gamma^2 q_y^2 \sigma_u}{2\rho^2} + \frac{\gamma q_x^2 \sigma_u}{2\rho^2} \right. \\ \left. - \frac{\gamma q_y^2 \sigma_u}{2\rho^2} + \frac{q_z^2 \sigma_u}{\rho^2} + \frac{q_y^2 \sigma_u}{\rho^2} \right)$$

$$B_{xy}^{E,\rho} = -\Delta_t \left(-\frac{E\gamma^2 q_z q_y \sigma_p}{\rho^3} - \frac{E\gamma q_x q_y \sigma_p}{\rho^3} + \frac{\gamma^2 q_z^3 q_y \sigma_p}{2\rho^4} + \frac{\gamma^2 q_z q_y^3 \sigma_p}{2\rho^4} + \frac{3\gamma q_x^3 q_y \sigma_p}{2\rho^4} \right. \\ \left. + \frac{3\gamma q_z q_y^3 \sigma_p}{2\rho^4} - \frac{2q_x^3 q_y \sigma_p}{\rho^4} - \frac{2q_z q_y^3 \sigma_p}{\rho^4} \right)$$

$$B_{yx}^{q_y, q_z} = -\Delta_t \left(\frac{E\gamma^2 \sigma_u}{\rho} - \frac{E\gamma \sigma_u}{\rho} - \frac{\gamma^2 q_x^2 \sigma_u}{2\rho^2} - \frac{\gamma^2 q_y^2 \sigma_u}{2\rho^2} - \frac{\gamma q_x^2 \sigma_u}{2\rho^2} + \frac{\gamma q_y^2 \sigma_u}{2\rho^2} \right. \\ \left. + \frac{q_x^2 \sigma_u}{\rho^2} + \frac{q_y^2 \sigma_u}{\rho^2} \right)$$

$$B_{yz}^{E,\rho} = -\Delta_t \left(-\frac{E\gamma^2 q_x q_y \sigma_p}{\rho^3} - \frac{E\gamma q_x q_y \sigma_p}{\rho^3} + \frac{\gamma^2 q_z^3 q_y \sigma_p}{2\rho^4} + \frac{\gamma^2 q_x q_y^3 \sigma_p}{2\rho^4} \right. \\ \left. + \frac{3\gamma q_x^3 q_y \sigma_p}{2\rho^4} + \frac{3\gamma q_z q_y^3 \sigma_p}{2\rho^4} - \frac{2q_x^3 q_y \sigma_p}{\rho^4} - \frac{2q_x q_y^3 \sigma_p}{\rho^4} \right)$$

$$B_{yy}^{q_y, \rho} = -\Delta_t \left(-\frac{E\gamma^2 q_y \sigma_u}{\rho^2} + \frac{E\gamma q_y \sigma_u}{\rho^2} + \frac{\gamma^2 q_x^2 q_y \sigma_u}{2\rho^3} + \frac{\gamma^2 q_y^3 \sigma_u}{2\rho^3} \right. \\ \left. + \frac{\gamma q_x^2 q_y \sigma_u}{\rho^3} + \frac{\gamma q_y^3 \sigma_u}{\rho^3} - \frac{3q_x^2 q_y \sigma_u}{2\rho^3} - \frac{7q_y^3 \sigma_u}{2\rho^3} \right)$$

$$B_{yg}^{h, \omega^2} = -\Delta_t \left(\frac{E\gamma^2 \sigma_u}{\rho} - \frac{E\gamma \sigma_u}{\rho} - \frac{\gamma^2 q_x^2 \sigma_u}{2\rho^2} - \frac{\gamma^2 q_y^2 \sigma_u}{2\rho^2} + \frac{\gamma q_x^2 \sigma_u}{2\rho^2} - \frac{5\gamma q_y^2 \sigma_u}{2\rho^2} \right. \\ \left. - \frac{\lambda^2 \sigma_u}{2} + \frac{6q_y^2 \sigma_u}{\rho^2} \right)$$

$$B_{yy}^{E,\rho} = -\Delta_t \left(\frac{E\gamma^2 q_x^2 \sigma_p}{2\rho^3} - \frac{E\gamma^2 q_y^2 \sigma_p}{2\rho^3} - \frac{E\gamma q_z^2 \sigma_p}{2\rho^3} - \frac{3E\gamma q_y^2 \sigma_p}{2\rho^3} - \frac{\gamma^2 q_x^4 \sigma_p}{4\rho^4} + \frac{\gamma^2 q_y^4 \sigma_p}{4\rho^4} \right. \\ \left. + \frac{\gamma q_z^4 \sigma_p}{2\rho^4} + \frac{5\gamma q_z^2 q_y^2 \sigma_p}{2\rho^4} + \frac{2\gamma q_y^4 \sigma_p}{\rho^4} - \frac{q_x^4 \sigma_p}{4\rho^4} - \frac{5q_x^2 q_y^2 \sigma_p}{2\rho^4} - \frac{9q_y^4 \sigma_p}{4\rho^4} \right)$$

$$B_{yy}^{E,q_x} = -\Delta_t \left(-\frac{E\gamma^2 q_x \sigma_p}{\rho^2} + \frac{E\gamma q_x \sigma_p}{\rho^2} + \frac{\gamma^2 q_x^3 \sigma_p}{2\rho^3} + \frac{\gamma^2 q_x q_y^2 \sigma_p}{2\rho^3} - \frac{\gamma q_x^3 \sigma_p}{\rho^3} - \frac{3\gamma q_x q_y^2 \sigma_p}{\rho^3} \right. \\ \left. + \frac{q_z^3 \sigma_p}{2\rho^3} + \frac{5q_z q_y^2 \sigma_p}{2\rho^3} \right)$$

$$B_{yy}^{E,E} = -\Delta_t \left(\frac{E\gamma^2 \sigma_p}{\rho} - \frac{E\gamma \sigma_p}{\rho} - \frac{\gamma^2 q_x^2 \sigma_p}{2\rho^2} - \frac{\gamma^2 q_y^2 \sigma_p}{2\rho^2} + \frac{\gamma q_x^2 \sigma_p}{\rho^2} + \frac{3\gamma q_y^2 \sigma_p}{\rho^2} \right. \\ \left. - \frac{\lambda^2 \sigma_p}{2} - \frac{q_x^2 \sigma_p}{2\rho^2} - \frac{3q_y^2 \sigma_p}{2\rho^2} \right)$$

Appendix E

Supersonic cylindrical-nosed body results

This appendix provides pictures of the results obtained for the test case of the cylindrical-nosed body.

E.1 Bouzidi solution

The figures below show the fields of several fluid dynamics variables, obtained with LBM, by imposing Bouzidi's boundary conditions on the wall.

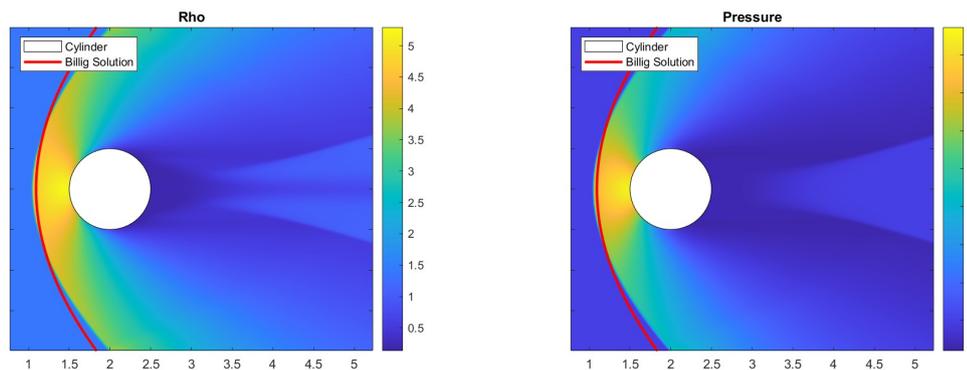


Figure E.1: Density Field - LBM Bouzidi Solution

Figure E.2: Pressure Field - LBM Bouzidi Solution

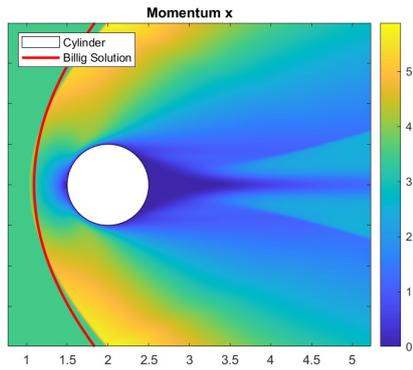


Figure E.3: Momentum-x Field - LBM Bouzidi Solution

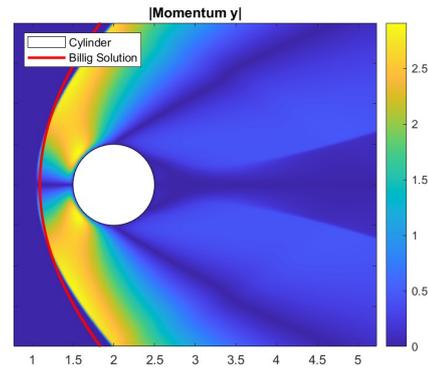


Figure E.4: Momentum-y Field - LBM Bouzidi Solution

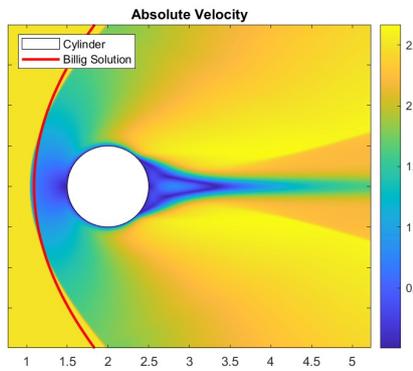


Figure E.5: $|V|$ Field - LBM Bouzidi Solution

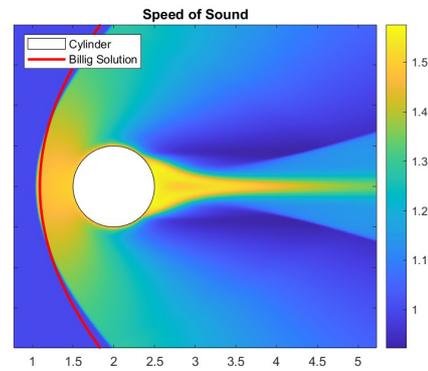


Figure E.6: Speed of Sound Field - LBM Bouzidi Solution

E.2 Bouzidi Bounce-Back with Normal solution

The figures below show the fields of several fluid dynamics variables, obtained with LBM, by imposing Bouzidi Bounce-Back with Normal boundary conditions on the wall.

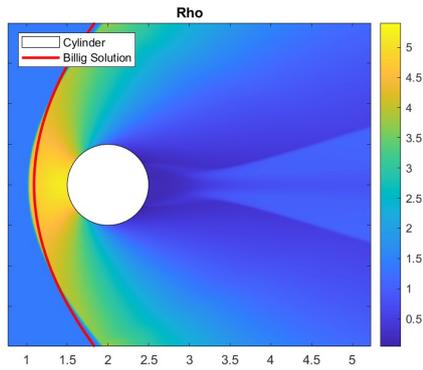


Figure E.7: Density Field - LBM Bouzidi Bounce Back with Normal Solution

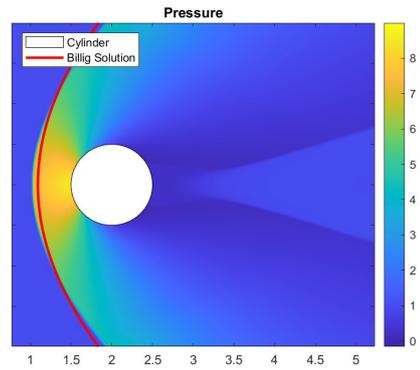


Figure E.8: Pressure Field - LBM Bouzidi Bounce Back with Normal Solution

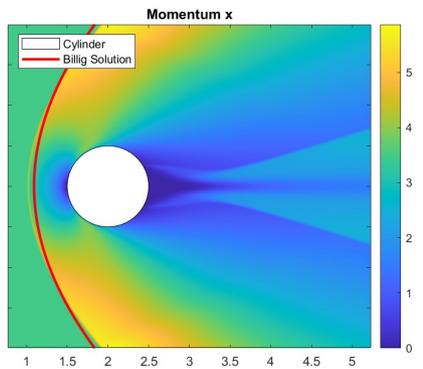


Figure E.9: Momentum-x Field - LBM Bouzidi Bounce Back with Normal Solution

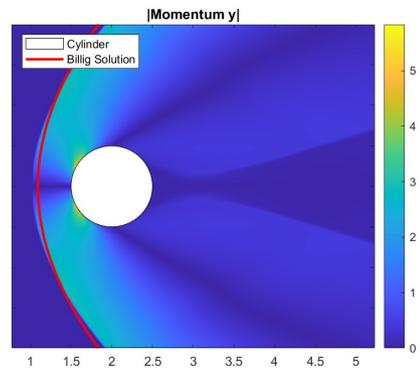


Figure E.10: Momentum-y Field - LBM Bouzidi Bounce Back with Normal Solution

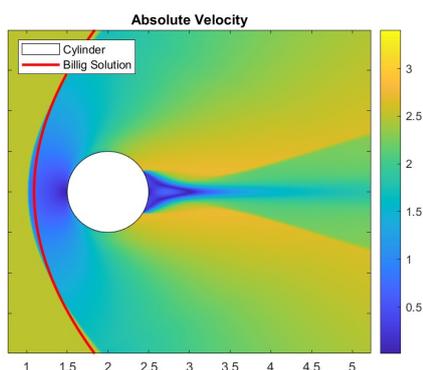


Figure E.11: $|V|$ Field - LBM Bouzidi Bounce Back with Normal Solution

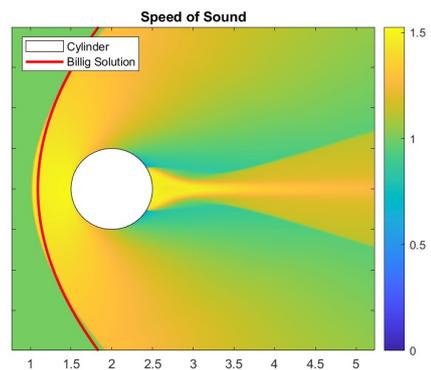


Figure E.12: Speed of Sound Field - LBM Bouzidi Bounce Back with Normal Solution

Bibliography

- [1] Roger Temam. *The Navier-Stokes Equations: Theory and Numerical Analysis*. CHEL Series. American Mathematical Society, 1977 (cit. on p. 2).
- [2] Sauro Succi. *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond*. Oxford, UK: Clarendon Press, 2001 (cit. on p. 2).
- [3] Ludwig Boltzmann. *Lectures on Gas Theory*. Dover Publications, 1995 (cit. on p. 2).
- [4] L. Gouarin and B. Graille. *pylbm*. Version 0.3.2. June 20, 2018. URL: <https://pylbm.readthedocs.io> (cit. on pp. 3, 5, 23, 27, 29, 50, 52, 56, 67, 69, 116, 156).
- [5] François Dubois, Tony Fevrier, and Benjamin Graille. «Lattice Boltzmann schemes with relative velocities». In: *Communications in Computational Physics* 17.4 (2015), pp. 1088–1112. URL: <http://arxiv.org/abs/1312.3297> (cit. on pp. 3, 23, 37).
- [6] Benjamin Graille. «Approximation of mono-dimensional hyperbolic systems: a lattice Boltzmann scheme as a relaxation method». In: *Journal of Computational Physics* 266 (2014), pp. 74–88. URL: <https://hal.archives-ouvertes.fr/hal-00919835> (visited on 04/17/2018) (cit. on pp. 3, 23, 37, 38, 40).
- [7] Mauro Canana. «Development of Non-Oscillatory Lattice Boltzmann Method for Flows with Shock waves». In: *ESA FAR Conference*. Italy, 2019 (cit. on pp. 4, 96, 152).
- [8] Ashley F Emery. «An evaluation of several differencing methods for inviscid fluid flow problems». In: *Journal of Computational Physics* 2.3 (1968), pp. 306–331 (cit. on p. 4).
- [9] S. Chapman and T. G. Cowling. *The Mathematical Theory of Nonuniform Gases*. Cambridge: Cambridge University Press, 1970 (cit. on pp. 6, 7).
- [10] Stephen Olfram. *Theory and Applications of Cellular Automata*. Singapore: World Scientific, 1986 (cit. on p. 7).

- [11] Y. Hardy, Y. Pomeau, and O. de Pazzis. «Time evolution of a two-dimensional model system. I. Invariant states and time correlation functions». In: *Journal of Mathematical Physics* 14.12 (1973), pp. 1746–1759 (cit. on p. 7).
- [12] U. Frisch, B. Hasslacher, and Y. Pomeau. «Lattice-gas automata for the Navier-Stokes equation». In: *Physical Review Letters* 56.15 (1986), pp. 1505–1508 (cit. on pp. 7, 26).
- [13] G. R. McNamara and G. Zanetti. «Use of the Boltzmann equation to simulate lattice-gas automata». In: *Physical Review Letters* 61.20 (1988), pp. 2332–2335 (cit. on pp. 7, 26).
- [14] L. Boltzmann. *Lectures on Gas Theory*. Berkeley: University of California Press, 1964 (cit. on p. 7).
- [15] C. Cercignani. *The Boltzmann Equation and Its Applications*. New York: Springer-Verlag, 1988 (cit. on p. 7).
- [16] U. Frisch, D. d’Humières, B. Hasslacher, P. Lallemand, Y. Pomeau, and J. P. Rivet. «Lattice gas hydrodynamics in two and three dimensions». In: *Complex Sys.* 1 (1987), pp. 649–707 (cit. on pp. 7, 53).
- [17] X. He and L.-S. Luo. «Lattice Boltzmann Model for the Incompressible Navier-Stokes Equation». In: *Journal of Statistical Physics* 88.3-4 (1997), pp. 927–944 (cit. on p. 7).
- [18] X. He and L.-S. Luo. «Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation». In: *Physical Review E* 56.6 (1997), pp. 6811–6817 (cit. on p. 7).
- [19] A.A. Mohamad. *Lattice Boltzmann Method, Fundamentals and Engineering Applications with Computer Codes*. London: Springer-Verlag, 2011 (cit. on pp. 8, 11, 18).
- [20] M. Bouzidi, M. Firdaouss, and P. Lallemand. «Momentum transfer of a Boltzmann-lattice fluid with boundaries». In: *Phys. Fluids* 13.11 (2001), pp. 3452–3459 (cit. on pp. 8, 47, 53).
- [21] D. d’Humières and I. Ginzburg. *Multi-reflection boundary conditions for lattice Boltzmann models*. Tech. rep. Fraunhofer-Institut für Techno- und Wirtschaftsmathematik, 2002 (cit. on p. 8).
- [22] Q. Zou and X. He. «On pressure and velocity boundary conditions for the lattice Boltzmann BGK model». In: *Physics of Fluids* 9 (1997), pp. 1591–1598 (cit. on pp. 8, 47).
- [23] P.A. Skordos. «Initial and boundary-conditions for the lattice Boltzmann method». In: *Physical Review E* 48.6 (1993), pp. 4823–4842 (cit. on pp. 8, 46, 47).

-
- [24] T. Inamuro, M. Yoshino, and F. Ogino. «Non-slip boundary-condition for lattice Boltzmann simulations». In: *Physics of Fluids* 7.12 (1995), pp. 2928–2930 (cit. on pp. 8, 46, 47).
- [25] J. Latt. «Hydrodynamic Limit of Lattice Boltzmann Equations». PhD thesis. University of Geneve, 2007 (cit. on p. 8).
- [26] D. d’Humières. «Generalized lattice Boltzmann equations». In: *Rarefied gas dynamics: theory and simulations*. Ed. by B.D. Shizgal and D.P. Weaver. Vol. 159. Progress in Aeronautics and Astronautics. 1992, pp. 450–458 (cit. on pp. 8, 22, 23).
- [27] D. d’Humières, I. Ginzburg, M. Krafczyk, P. Lallemand, and L.-S. Luo. «Multiple-relaxation-time lattice Boltzmann models in three dimensions». In: *Philosophical Transactions of the Royal Society of London A* 360 (2001), pp. 437–451 (cit. on p. 8).
- [28] I. Ginzburg, F. Verhaeghe, and D. d’Humières. «Study of simple hydrodynamic solutions with the two-relaxation-times lattice Boltzmann scheme». In: *Communications in Computational Physics* 3 (2008), pp. 519–581 (cit. on p. 8).
- [29] James Clerk Maxwell. «On the dynamical theory of gases». In: *Philosophical Transactions of the Royal Society of London* 157 (1867), pp. 49–88 (cit. on p. 15).
- [30] Ludwig Boltzmann. «Weitere studien über das wärmeleichgewicht unter gasmolekülen». In: *Berichte der Mathematisch-Physikalischen Klasse der Königlich Bayerischen Akademie der Wissenschaften* 66 (1872), pp. 275–370 (cit. on p. 16).
- [31] Carlo Cercignani and Robert E. Street. *Mathematical Methods In Kinetic Theory*. eng. Vol. 23. 10. 1990 (cit. on p. 20).
- [32] P. L. Bhatnagar, E. P. Gross, and M. Krook. «A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems». In: *Phys. Rev.* 94 (3 May 1954), pp. 511–525. DOI: 10.1103/PhysRev.94.511. URL: <https://link.aps.org/doi/10.1103/PhysRev.94.511> (cit. on p. 20).
- [33] Dominique d’Humières. «Multiple-relaxation-time lattice Boltzmann models in three dimensions». In: *Phil. Trans. R. Soc. Lond. A* 360 (2002), pp. 1089–1105. DOI: 10.1098/rsta.2001.0955. URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2001.0955> (cit. on p. 22).
- [34] James E Broadwell. «A study of weak and strong shock waves in a rarefied gas». In: *Journal of Fluid Mechanics* 19.1 (1964), pp. 1–24 (cit. on p. 26).

-
- [35] Takaji Inamuro and Bradford Sturtevant. «Numerical study of discrete-velocity gases». In: *Physics of Fluids A* 2.12 (1990), pp. 2196–2203 (cit. on p. 26).
- [36] J. Hardy, O. de Pazzis, and Y. Pomeau. «Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions». In: *Physical Review A* 13.5 (1976), pp. 1949–1958 (cit. on p. 26).
- [37] Stephen Wolfram. «Theory and applications of cellular automata». In: *Physica D: Nonlinear Phenomena* 10.1-2 (1984), pp. 1–35 (cit. on p. 26).
- [38] D. d’Humières and P. Lallemand. «Lattice gas automata for fluid mechanics». In: *Physica* 140A.2 (1990), pp. 326–335 (cit. on p. 26).
- [39] D. d’Humières, P. Lallemand, L.-S. Luo, and U. Frisch. «Lattice gas models for 3D hydrodynamics». In: *Europhysics Letters* 2.4 (1986), pp. 291–297 (cit. on p. 26).
- [40] FJ Higuera and J. Jimenez. «Boltzmann approach to lattice gas simulations». In: *Europhysics Letters* 9.7 (1989), pp. 663–668 (cit. on p. 26).
- [41] FJ Higuera, S. Succi, and R. Benzi. «Lattice gas dynamics with enhanced collisions». In: *Europhysics Letters* 9.4 (1989), pp. 345–349 (cit. on p. 26).
- [42] P. L. Bhatnagar, E. P. Gross, and M. Krook. «A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems». In: *Physical Review* 94.3 (1954), pp. 511–525 (cit. on p. 26).
- [43] YH Qian. «Lattice gas and lattice kinetic theory applied to the Navier-Stokes equations». PhD thesis. Université Pierre et Marie Curie, Paris, 1990 (cit. on p. 26).
- [44] S. Chen, H. D. Chen, D. Martinez, and W. Matthaeus. «Lattice Boltzmann model for simulation of magnetohydrodynamics». In: *Physical Review Letters* 67.26 (1991), pp. 3776–3779 (cit. on p. 26).
- [45] YH Qian, D. d’Humières, and P. Lallemand. «Lattice BGK models for Navier-Stokes equation». In: *Europhysics Letters* 17.6 (1992), pp. 479–484 (cit. on p. 26).
- [46] H. Chen, S. Chen, and W. H. Matthaeus. «Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method». In: *Physical Review A* 45.8 (1992), R5339–R5342 (cit. on p. 26).
- [47] T. Kataoka and M. Tsutahara. «Lattice Boltzmann method for the compressible Euler equations». In: *Physical Review E* 69.5 (May 2004), p. 056702 (cit. on p. 28).

- [48] M. Watari and M. Tsutahara. «Supersonic flow simulations by a three-dimensional multispeed thermal model of the finite difference lattice Boltzmann method». In: *Physica A: Statistical Mechanics and its Applications* 364 (May 2005), pp. 129–144 (cit. on p. 28).
- [49] X. F. Pan, A. Xu, G. Zhang, and S. Jiang. «Lattice Boltzmann approach to high-speed compressible flows». In: *International Journal of Modern Physics C* 18.11 (Nov. 2007), pp. 1747–1764 (cit. on p. 28).
- [50] F. Chen, A. Xu, G. Zhang, Y. Li, and S. Succi. «Multiple-relaxation-time lattice Boltzmann approach to compressible flows with flexible specific-heat ratio and Prandtl number». In: *EPL (Europhysics Letters)* 90.5 (June 2010), p. 54003 (cit. on p. 29).
- [51] A.-G. Xu, G.-C. Zhang, Y.-B. Gan, F. Chen, and X.-J. Yu. «Lattice Boltzmann modeling and simulation of compressible flows». In: *Frontiers of Physics* 7.5 (Oct. 2012), pp. 582–600 (cit. on p. 29).
- [52] Y. Gan, A. Xu, G. Zhang, and Y. Yang. «Lattice BGK kinetic model for high speed compressible flows: Hydrodynamic and nonequilibrium behaviors». In: *EPL (Europhysics Letters)* 103.2 (July 2013), p. 24003 (cit. on p. 29).
- [53] A. Xu, G. Zhang, and Y. Zhang. «Discrete Boltzmann modeling of compressible flows». In: *Kinetic Theory*. Dec. 2017. DOI: 10.1007/978-3-319-59958-9_25 (cit. on p. 29).
- [54] Y. Gan, A. Xu, G. Zhang, Y. Zhang, and S. Succi. «Discrete Boltzmann transscale modeling of high-speed compressible flows». In: *Physical Review E* 97.5 (May 2018), p. 053312. DOI: 10.1103/PhysRevE.97.053312 (cit. on p. 29).
- [55] X. He, S. Chen, and GD Doolen. «A novel thermal model for the lattice Boltzmann method in incompressible limit». In: *Journal of Computational Physics* 146.1 (Oct. 1998), pp. 282–300 (cit. on p. 29).
- [56] Z. Guo, C. Zheng, B. Shi, and TS Zhao. «Thermal lattice Boltzmann equation for low Mach number flows: Decoupling model». In: *Physical Review E* 75.3 (Mar. 2007), p. 036704 (cit. on p. 29).
- [57] Q. Li, YL He, Y. Wang, and WQ Tao. «Coupled double-distribution-function lattice Boltzmann method for the compressible Navier-Stokes equations». In: *Physical Review E* 76.5 (Nov. 2007), p. 056705 (cit. on p. 29).
- [58] R.-F. Qiu, Y.-C. You, C.-X. Zhu, R.-Q. Chen, and J.-F. Zhu. «Three-dimensional coupled double-distribution-function lattice Boltzmann models for compressible Navier–Stokes equations». In: *Pramana-Journal of Physics* 89.6 (Dec. 2017) (cit. on p. 29).

- [59] Q. Kun. «DEVELOPMENT OF LATTICE BOLTZMANN METHOD FOR COMPRESSIBLE FLOWS». PhD thesis. Northwestern Polytechnical University, China, 2008 (cit. on p. 30).
- [60] L. M. Yang, C. Shu, J. Wu, Y. Wang, and Y. Sun. «Comparative study of 1D, 2D and 3D simplified gas kinetic schemes for simulation of inviscid compressible flows». In: *Applied Mathematical Modelling* 43 (Mar. 2017), pp. 85–109 (cit. on p. 30).
- [61] Z. Meng, L. Yang, C. Shu, F. Hu, D. Wang, and W. Zhang. «Development of lattice boltzmann flux solver for simulation of hypersonic flow past flight vehicles». In: *Journal of Physics: Conference Series* 1053 (July 2018), p. 012073 (cit. on p. 30).
- [62] Y. Li, X.-D. Niu, H.-Z. Yuan, Adnan, and X. Li. «A numerical study for WENO scheme-based on different lattice boltzmann flux solver for compressible flows». In: *Acta Mechanica Sinica* 34.6 (Apr. 2018), pp. 995–1014 (cit. on p. 30).
- [63] Kun Xu. «A gas-kinetic BGK scheme for the Navier–Stokes equations and its connection with artificial dissipation and Godunov method». In: *Journal of Computational Physics* 171.1 (2001), pp. 289–335 (cit. on p. 30).
- [64] Kun Xu and Chao Liu. «A paradigm for modeling and computation of gas dynamics». In: *Physics of Fluids* 29.2 (2017), p. 026101 (cit. on p. 31).
- [65] Liang Pan, Kun Xu, Qibing Li, and Jianxun Li. «An efficient and accurate two-stage fourth-order gas-kinetic scheme for the Euler and Navier–Stokes equations». In: *Journal of Computational Physics* 326 (2016), pp. 197–221 (cit. on p. 31).
- [66] Xingyu Ji, Fengshan Zhao, Wei Shyy, and Kun Xu. «A family of high-order gas-kinetic schemes and its comparison with Riemann solver based high-order methods». In: *Journal of Computational Physics* 356 (2018), pp. 150–173 (cit. on p. 31).
- [67] Shuai Tao, Heng Zhang, Zhaoli Guo, and Lian-Ping Wang. «A combined immersed boundary and discrete unified gas kinetic scheme for particle–fluid flows». In: *Journal of Computational Physics* 375 (2018), pp. 498–518 (cit. on p. 31).
- [68] Zhaoli Guo, Kun Xu, and Rongshan Wang. «Discrete unified gas kinetic scheme for all Knudsen number flows: Low-speed isothermal case». In: *Physical Review E* 88.3 (2013), p. 033305 (cit. on p. 31).
- [69] Shyam S Chikatamarla and Iliya V Karlin. «Entropy and galilean invariance of lattice boltzmann theories». In: *Physical Review Letters* 97.19 (2006), p. 190601 (cit. on p. 31).

- [70] Nicola Frapolli, Shyam S Chikatamarla, and Iliya V Karlin. «Entropic lattice Boltzmann model for compressible flows». In: *Physical Review E* 92.6 (2015), p. 061301 (cit. on p. 31).
- [71] David Coulette, Emmanuel Franck, Philippe Helluy, Michel Mehrenberger, and Laurent Navoret. «High-order implicit palindromic discontinuous Galerkin method for kinetic-relaxation approximation». In: *Journal of Computational Physics* 355 (2018), pp. 49–74 (cit. on pp. 31, 32).
- [72] Benedikt Dorschner, Franz Bösch, and Iliya V Karlin. «Particles on demand for kinetic theory». In: *Physical Review Letters* 121.13 (2018), p. 130602 (cit. on p. 32).
- [73] Aditya Jammalamadaka, Yongle Li, Pradeep Gopalakrishnan, Rui Zhang, and Haibo Chen. «Simulation of a canonical shock wave-boundary layer interaction using an LBM-VLES approach». In: *2018 Fluid Dynamics Conference*. American Institute of Aeronautics & Astronautics. 2018 (cit. on p. 32).
- [74] Xiaojun Nie, Xiaowen Shan, and Haibo Chen. «A lattice-Boltzmann/finite-difference hybrid simulation of transonic flow». In: *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics & Astronautics. 2009 (cit. on p. 32).
- [75] Vincent Giovangigli. *Multicomponent Flow Modeling*. Erratum at <http://cmap.polytechnique.fr/~giovangi>. Boston: Birkhäuser, 1999 (cit. on p. 37).
- [76] Peter D Lax and Kurt O Friedrichs. «Some numerical calculations for the nonlinear wave equations». In: *Journal of the Society for Industrial and Applied Mathematics* 2.2 (1954), pp. 168–184 (cit. on p. 37).
- [77] Shi Jin and Zhouping Xin. «The relaxation schemes for systems of conservation laws in arbitrary space dimensions». In: *Communications on Pure and Applied Mathematics* 48.3 (1995), pp. 235–277 (cit. on p. 38).
- [78] D. d’Humière. «Generalized lattice-Boltzmann equations». In: *Rarefied Gas Dynamics: Theory and Simulation, in: Prog. Astronaut. Aeronaut., vol. 159, AIAA* (1992), pp. 450–458 (cit. on p. 38).
- [79] F. Dubois. «Une introduction au schéma de Boltzmann sur réseau». In: *ESAIM Proc.* 18 (2007), pp. 181–215 (cit. on pp. 38, 40, 43, 97, 117, 164).
- [80] F. Dubois. «Equivalent partial differential equations of a lattice Boltzmann scheme». In: *Comput. Math. Appl.* 55 (2008), pp. 1441–1449 (cit. on pp. 38, 40, 43, 97, 117, 164).

- [81] François Dubois and Pierre Lallemand. «Towards higher order lattice Boltzmann schemes». In: *Journal of Statistical Mechanics: Theory and Experiment* 2009.6 (2009), P06006. DOI: 10.1088/1742-5468/2009/06/P06006 (cit. on p. 38).
- [82] S. Wolfram. «Cellular automaton fluids 1: Basic theory». In: *J. Stat. Phys.* 45 (1986), p. 471 (cit. on p. 46).
- [83] P. Lavalée, J. P. Boon, and A. Noullez. «Boundaries in lattice gas flows». In: *Physica D* 47 (1991), p. 233 (cit. on p. 46).
- [84] R. Cornubert, D. d’Humières, and D. Levermore. «A knudsen layer theory for lattice gases». In: *Physica D* 47 (1991), pp. 241–259 (cit. on pp. 46, 53, 65).
- [85] D. P. Ziegler. «Boundary conditions for lattice Boltzmann simulations». In: *J. Stat. Phys.* 71 (1993), pp. 1171–1177 (cit. on p. 46).
- [86] I. Ginzbourg and P. M. Adler. «Boundary flow condition analysis for three-dimensional lattice Boltzmann model». In: *J. Phys. II France* 4 (1994), pp. 191–214 (cit. on pp. 46, 49, 50, 53, 65).
- [87] Xiaoyi He, Q. Zou, L.-S. Luo, and M. Dembo. «Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice Boltzmann BGK model». In: *J. of Stat. Phys.* 87 (1997), pp. 115–136 (cit. on pp. 46, 49, 50).
- [88] Oskar Behrend. «Solid boundaries in particle suspension simulations via lattice Boltzmann method». In: *Physical Review E* 52.2 (1995), p. 1164 (cit. on p. 46).
- [89] D. R. Noble, S. Chen, J. G. Georgiadis, and R. O. Buckius. «A consistent hydrodynamic boundary condition for the lattice Boltzmann method». In: *Phys. fluids A* 7 (1995), pp. 203–209 (cit. on pp. 46, 47).
- [90] S. Chen, D. Martínez, and R. Mei. «On boundary conditions in lattice Boltzmann methods». In: *Phys. Fluids* 8.9 (1996), pp. 2527–2536 (cit. on pp. 46, 47, 55).
- [91] R. Mei, L.-S. Luo, and W. Shyy. «An accurate curved boundary treatment in the lattice Boltzmann method». In: *J. Comp. Phys.* 155 (1999), pp. 307–330 (cit. on pp. 46, 47).
- [92] D. R. Noble, J. G. Georgiadis, and R. O. Buckius. «Direct assessment of lattice Boltzmann hydrodynamics and boundary conditions for recirculating flows». In: *J. Stat. Phys.* 81 (1995), pp. 17–33 (cit. on p. 47).
- [93] R. S. Maier, R. S. Bernard, and D. W. Grunau. «Boundary conditions for the lattice Boltzmann method». In: *Phys. Fluids* 8.7 (1996), pp. 1788–1801 (cit. on p. 47).

- [94] Q. Zou, S. Hou, and G. D. Doolen. «Analytical solutions of the lattice Boltzmann BGK model». In: *J. Stat. Phys.* 81 (1995), p. 319 (cit. on p. 47).
- [95] O. Filippova and D. Hänel. «Boundary-fitting and local grid refinement for lattice-BGK models». In: *Int. J. Mod. Phys. C* 9 (1998), pp. 1271–1279 (cit. on p. 47).
- [96] O. Filippova and D. Hänel. «Grid refinement for lattice-BGK models». In: *J. Comp. Phys.* 147 (1998), pp. 219–228 (cit. on pp. 47, 55).
- [97] R. Mei, W. Shyy, D. Yu, and L.-S. Luo. «Lattice Boltzmann method for 3-D flows with curved boundary». In: *J. Comp. Phys.* 161 (2000), pp. 680–699 (cit. on p. 47).
- [98] I. Ginzbourg and d’Humières. «The multireflection boundary conditions for lattice Boltzmann models». In: *Physical Review E* 68 (2003), p. 066614 (cit. on p. 47).
- [99] Simon Bennet. «A Lattice Boltzmann Model for Diffusion of Binary Gas Mixtures». PhD Thesis. University of Cambridge, 2010 (cit. on p. 50).
- [100] Irina Ginzburg, Frederic Verhaeghe, and Dominique d’Humières. «Two-relaxation-time lattice Boltzmann scheme: About parametrization, velocity, pressure and mixed boundary conditions». In: *Communications in Computational Physics* 3.2 (2008), pp. 427–478 (cit. on p. 50).
- [101] Guilherme Silva, Laurent Talon, and Irina Ginzburg. «Low-and high-order accurate boundary conditions: From Stokes to Darcy porous flow modeled with standard and improved Brinkman lattice Boltzmann schemes». In: *Journal of Computational Physics* 335 (2017), pp. 50–83 (cit. on p. 50).
- [102] François Dubois, Pierre Lallemand, and Mounir M Tekitek. «Taylor expansion method for analyzing bounce-back boundary conditions for lattice Boltzmann method». In: *ESAIM: Proceedings and Surveys* 52 (2015), pp. 25–46 (cit. on p. 50).
- [103] Dominique d’Humières and Irina Ginzburg. «Viscosity independent numerical errors for Lattice Boltzmann models: From recurrence equations to "magic" collision numbers». In: *Computers & Mathematics with Applications* 58.4 (2009), pp. 823–840 (cit. on p. 50).
- [104] I Ginzburg and D d’Humières. «Local second-order boundary methods for lattice Boltzmann models». In: *Journal of Statistical Physics* 84.3-4 (1996), pp. 927–947 (cit. on pp. 53, 65).
- [105] Shiyi Chen and Gary D Doolen. «Lattice Boltzmann method for fluid flows». In: *Annual Review of Fluid Mechanics* 30 (1998), pp. 329–364 (cit. on p. 53).

- [106] IV Karlin, S Succi, and SA Orszag. «Lattice Boltzmann method for irregular grids». In: *Physical Review Letters* 82.26 (1999), pp. 5245–5248 (cit. on p. 53).
- [107] P Lallemand and L-S Luo. «Theory of the lattice Boltzmann method: Dispersion, dissipation, isotropy, Galilean invariance and stability». In: *Physical Review E* 61.6 (2000), pp. 6546–6562 (cit. on p. 54).
- [108] D Yu, R Mei, L-S Luo, and W Shyy. «Viscous flow computations with the method of lattice Boltzmann equation». In: *Progress in Aerospace Sciences* 39.5 (2003), pp. 329–367 (cit. on p. 56).
- [109] François Dubois. «Boundary conditions for gas dynamics». In: *Absorbing Boundaries and Layers, Domain Decomposition Methods: Applications to Large Scale Computations*. Nova Science Publishers, Inc., 2001, pp. 16–77 (cit. on p. 64).
- [110] Huazhen Yan, Fan Liu, Jinglei Xu, and Yafeng Xue. «Study of Oblique Shock Wave Control by Surface Arc Discharge Plasma». In: *AIAA Journal* 56.2 (2017), pp. 532–541 (cit. on p. 74).
- [111] John M Delery. «Shock Wave/Turbulent Boundary Layer Interaction and Its Control». In: *Progress in Aerospace Sciences* 22.4 (1985), pp. 209–280 (cit. on p. 74).
- [112] Tian Gan, Yue Wu, Zhiwei Sun, Dongdong Jin, Haixia Song, and Meng Jia. «Shock Wave Boundary Layer Interaction Controlled by Surface Arc Plasma Actuators». In: *Physics of Fluids* 30.5 (2018), p. 055107 (cit. on p. 74).
- [113] P. J. K. Bruce and S. P. Colliss. «Review of Research into Shock Control Bumps». In: *Shock Waves* 25.5 (2015), pp. 451–471 (cit. on p. 74).
- [114] Nicholas Titchener and Holger Babinsky. «Shock Wave/Boundary-Layer Interaction Control Using a Combination of Vortex Generators and Bleed». In: *AIAA Journal* 51.5 (2013), pp. 1221–1233 (cit. on p. 74).
- [115] Wang Bo, Li Weidong, Zhang Yuxin, Fan Xiaoqiang, and Wu Chao. «Experimental Investigation of the Micro-Ramp Based Shock Wave and Turbulent Boundary Layer Interaction Control». In: *Physics of Fluids* 24.5 (2012), p. 055110 (cit. on p. 74).
- [116] Bjoern Reinartz and Matthias Behr. «Computational Design Study of a 3D Hypersonic Intake for Scramjet Demonstrator Testing». In: *High Performance Computing in Science and Engineering '10*. 2011 (cit. on p. 75).
- [117] John David Anderson. *Fundamentals of Aerodynamics*. Boston: McGraw-Hill, 2004 (cit. on p. 75).
- [118] I B Zel'dovich. «Physics of Shock Waves and High-Temperature Hydrodynamic Phenomena». In: *Physics Today* 23.2 (1970), p. 74 (cit. on p. 75).

- [119] Richard Courant and Kurt Otto Friedrichs. *Supersonic Flow and Shock Waves*. Beijing: World Books Publishing Corporation, 2006 (cit. on p. 75).
- [120] Ascher H Shapiro. *The Dynamics and Thermodynamics of Compressible Fluid Flow*. Malabar, Fla.: R.E. Krieger, 1987 (cit. on p. 75).
- [121] R. C. Mehta. «Numerical Simulation of Base Pressure and Drag of Space Reentry Capsules at High Speed». In: *Hypersonic Vehicles*. Cham: Springer, 2019, pp. - (cit. on pp. 77, 124).
- [122] P. R. Garabedian and H. M. Lieberstein. *Examples of Detached Bow Shock Waves in Hypersonic Flow*. New York: Institute of the Aeronautical Sciences, 1958 (cit. on pp. 77, 124).
- [123] P. V. Marrone. *Inviscid, Nonequilibrium Flow Behind Bow and Normal Shock Waves*. Tech. rep. Buffalo: Cornell Aeronautical Laboratory, 1964 (cit. on pp. 77, 124).
- [124] W. J. M. Rankine. «On the thermodynamic theory of waves of finite longitudinal disturbances». In: *Phil. Trans. Roy. Soc. London* 160 (1870), p. 277 (cit. on pp. 78, 125, 130).
- [125] H. Hugoniot. «Propagation des Mouvements dans les Corps et spécialement dans les Gaz Parfaits». In: *Journal de l'École Polytechnique* 57 (1887). Also published in *Journal de l'École Polytechnique*, vol. 58, p. 1, 1889, p. 3 (cit. on pp. 78, 125, 130).
- [126] Graham V. Candler, Heath B. Johnson, Ioannis Nompelis, Pramod K. Subbareddy, Travis W. Drayna, Vladimyr Gidzak, and Michael D. Barnhardt. «Development of the US3D code for advanced compressible and reacting flow simulations». English (US). In: *53rd AIAA Aerospace Sciences Meeting*. 53rd AIAA Aerospace Sciences Meeting. American Institute of Aeronautics and Astronautics Inc, AIAA, 2015. ISBN: 9781624103438. DOI: 10.2514/6.2015-1893 (cit. on p. 84).
- [127] John D. Anderson. *Modern Compressible Flow: With Historical Perspective*. 3rd. McGraw-Hill Series in Aeronautical and Aerospace Engineering. McGraw-Hill, 2003. ISBN: 0072424435 (cit. on p. 125).
- [128] E. S. Love. *A Re-examination of the Use of Simple Concepts for Predicting the Shape and Location of Detached Shock Waves*. Tech. rep. TN 4170. National Advisory Committee for Aeronautics, Dec. 1957 (cit. on p. 127).
- [129] Frederick S. Billig. «Shock-Wave Shapes around Spherical and Cylindrical-Nosed Bodies». In: *Applied Physics Laboratory* (Apr. 1967). Received April 18, 1967. Supervisor, Hypersonic Ramjet Project. Associate Fellow AIAA. (cit. on p. 127).

- [130] W. E. Moeckel. *Approximate Method for Predicting Form and Location of Detached Shock Waves Ahead of Plane or Axially Symmetric Bodies*. Tech. rep. TN 1921. National Advisory Committee for Aeronautics, 1949 (cit. on p. 127).
- [131] A. Ambrosio and A. Wortman. «Stagnation Point Shock Detachment Distance for Flow Around Spheres and Cylinders». In: *ARS Journal* 32 (1962), p. 281 (cit. on p. 127).
- [132] C.-S. Kim. «Experimental Studies of Supersonic Flow Past a Circular Cylinder». In: *J. Phys. Soc. Japan* 11 (Apr. 1956), pp. 439–445 (cit. on pp. 128, 135, 142).
- [133] J.W. Heberle, G.P. Wood, and P.B. Gooderum. *Data on Shape and Location of Detached Shock Waves on Cones and Spheres*. Tech. rep. TN 2000. NACA, 1950 (cit. on pp. 128, 135, 142).
- [134] W.E. Moekel. *Experimental Investigation of Supersonic Flow with Detached Shock Waves for Mach Numbers between 1.8 and 2.9*. Tech. rep. RME50D05. NACA, July 1950 (cit. on pp. 128, 135, 142).
- [135] R.E. Oliver. *An Experimental Investigation of Flow over Simple Blunt Bodies at a Nominal Mach Number of 5.8*. GALCIT Memo 26. Guggenheim Aeronautical Laboratory, California Institute of Technology, June 1955 (cit. on pp. 128, 135, 142).
- [136] D.H. Crawford and W.D. McCauley. *Investigation of the Laminar Aerodynamic Heat-Transfer Characteristics of a Hemisphere-Cylinder in the Langley 11-Inch Hypersonic Tunnel at a Mach Number of 6.8*. TN 3706. NACA, 1956 (cit. on pp. 128, 135, 142).
- [137] L. Lees and T. Kubota. «Inviscid Hypersonic Flow over Blunt-Nosed Slender Bodies». In: *Journal of Aeronautical Sciences* 24.3 (Mar. 1957), pp. 195–202 (cit. on pp. 128, 135, 142).
- [138] L. Euler. «Principes généraux du mouvement des fluides». In: *Mémoires de l'Académie Royale des Sciences de Berlin* (1755) (cit. on p. 154).
- [139] C. Navier. «Mémoire sur les lois du mouvement des fluides». In: *Mémoires de l'Académie des Sciences de l'Institut de France* 6 (1823), pp. 389–416 (cit. on p. 155).
- [140] G. Stokes. «On the theories of the internal friction of fluids in motion, and of the equilibrium and motion of elastic fluids». In: *Transactions of the Cambridge Philosophical Society* 8 (1845), p. 287 (cit. on p. 155).