

Immersed Boundary Method for Compressible Navier-Stokes Equations



By

Sadaqt Hussain
(Registration No: 00000365342)

Department of Mathematics

School of Natural Sciences

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

(2024)

Immersed Boundary Method for Compressible Navier-Stokes Equations



By

Sadaqat Hussain

(Registration No: 00000365342)

A thesis submitted to the National University of Sciences and Technology, Islamabad,

in partial fulfillment of the requirements for the degree of

Master of Science in

Mathematics

Supervisor: Dr. Muhammad Asif farooq

School of Natural Sciences

National University of Sciences and Technology (NUST)

Islamabad, Pakistan

(2024)


National University of Sciences & Technology


MS THESIS WORK

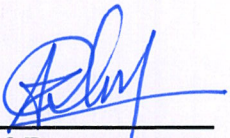
We hereby recommend that the dissertation prepared under our supervision by: Sadaqat Hussain, Regn No. 00000365342 Titled Immersed Boundary Method for Compressible Navier-Stokes Equation be Accepted in partial fulfillment of the requirements for the award of MS degree.

Examination Committee Members

1. Name: PROF. MERAJ MUSTAFA HASHMI Signature: 

2. Name: DR. RIZWAN UL HAQ Signature: 

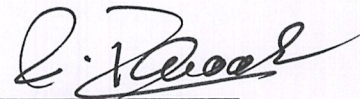
Supervisor's Name: DR. MUHAMMAD ASIF FAROOQ Signature: 


Head of Department

15-04-2024
Date

COUNTERSIGNED

Date: 15.04.2024


Dean/Principal

DEDICATION

This thesis is dedicated to My teachers , My beloved parents and My life partner for their valuable encouragement.

ACKNOWLEDGEMENTS

First and foremost, I would like to praise Allah the Almighty, the Most Gracious, and the Most Merciful for His blessing given to me during my study and in completing my thesis. May Allah's blessing goes to His last Prophet Muhammad (peace be up on him), his family and his companions.

Thank you to my supervisor, Dr. Muhammad Asif Farooq, for his patience, guidance, and support. I have benefited greatly from your wealth of knowledge and meticulous editing. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. I would like to express my gratitude and sincere thanks to Head of Department Prof. Mujeeb ur Rehman . I would also like to express the gratitude to GEC members Dr. Meraj Mustafa Hashmi and and Dr. Rizwan-ul-Haq all faculty members for their help. I am grateful for my parents,sister Fozia and brother in law chaudhary Shabbir whose constant love and support keep me motivated and confident. My accomplishments and success are because they believed in me. I would like to extend my sincere thanks to Anam chaudhary who helped me in my thesis and supported me in my work.

Finally, I sincerely owe my beloved friends Abdul Shakoor Saifi, for their suggestions, support, and motivation. Without their support I cannot complete this research work. At the end, the great thanks to those who supported me in any respect during the completion of my thesis.

Contents

LIST OF FIGURES	xi
ABSTRACT	xii
1 Introduction	1
1.1 Basic Definitions	2
1.1.1 Fluid	2
1.1.2 Computational Fluid Dynamics	2
1.1.3 Compressible Flow	2
1.1.4 Incompressible Flow	3
1.1.5 Mach Number	3
1.1.6 Viscous Fluid	3
1.1.7 Conservation Laws	3
1.1.8 Continuity Equation	4
1.1.9 Momentum Equation	4
1.1.10 Energy Equation	5
1.1.11 CFL Number	5
1.1.12 Von Neumann Number	5
1.1.13 Rankine-Hugoniot Relations	6
1.2 Skin Friction Coefficient	6
1.2.1 MUSCL Scheme	7
1.2.2 Total Variation Diminishing	7
1.2.3 Roe's Approximate Riemann Solver	7
1.2.4 Cartesian Grid Method	7
1.2.5 Body Fitted Grid Method	8
1.3 Comparing the Cell-Vertex as well as Cell-Centred Methodologies	8
1.3.1 Cell-Centred Scheme	9
1.3.2 Cell-Vertex Scheme	9
1.3.3 Boundary Conditions	9

2	Governing Equations	11
2.1	Compressible Navier-Stokes Equations	11
3	Numerical Methods	13
3.1	Basic Discretization	13
3.2	Spatial Discretization	14
3.3	Local Lax-Friedrichs Method	14
3.4	Boundary Conditions Discretization	18
3.5	Temporal Discretization	19
3.6	Analysis of a Numerical Scheme's Stability	20
3.7	Roe's Approximate Riemann Solver	21
3.8	Explanation for Simplified Ghost Point Method on Embedded Boundary in two Dimensions	24
3.9	Treatment of Ghost Points with Second Order Accuracy	26
3.10	Calculating the Shear Stress on a Body	27
4	Test Cases for Navier-Stokes Equation	28
4.1	Supersonic Flow over Circular Cylinder Employing Compressible Navier-Stokes Equation	28
5	Conclusions and Outlook	39
	BIBLIOGRAPHY	40

List of Figures

1.1	Cartesian grid method.	8
1.2	An example of Body fitted grid.	8
1.3	Domain and boundary conditions diagram [21].	9
3.1	Diagram exhibiting the Cartesian domain's numerical fluxes.	18
3.2	Domain sketch with discretized boundary conditions.	19
3.3	81x81 grid with ghost cells marked with flags. Ghost points' whereabouts are indicated by back points.	25
3.4	The configuration of the ghost points to create a solid cylinder and the selection of the fluid points as mirror points as an example.	26
3.5	A demonstration of the MUSCL ghost point computations.	27
3.6	The locations for which the shear stress on the wall was averaged.	27
4.1	Velocity u_o contours having 5000 iterations for 81×81 and 161×161 grid points for 1 cylinder.	29
4.2	Velocity v_o contours having 5000 iterations for 81×81 and 161×161 grid points for 1 cylinder.	29
4.3	Mach number contours having 5000 iterations for 81×81 and 161×161 grid points for 1 cylinder.	30
4.4	Density contours having 5000 iterations for 81×81 and 161×161 grid points for 1 cylinder.	30
4.5	Velocity u_o contours having 5000 iterations for 81×81 and 161×161 grid points for 3 cylinder.	31
4.6	Velocity v_o contours having 5000 iterations for 81×81 and 161×161 grid points for 3 cylinder.	31
4.7	Mach number contours having 5000 iterations for 81×81 and 161×161 grid points for 3 cylinder.	32
4.8	Density contours having 5000 iterations for 81×81 and 161×161 grid points for 3 cylinder.	32

4.9	A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average. . .	33
4.10	A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average. . .	33
4.11	A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average. . .	34
4.12	A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average. . .	34
4.13	A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average. . .	35
4.14	A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average. . .	35
4.15	A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average. . .	36
4.16	A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average. . .	36
4.17	A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average. . .	37
4.18	A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average. . .	37
4.19	A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average. . .	38

Abstract

For viscous compressible flow, a Cartesian grid approach has been devised to solve the Navier-Stokes equations. Both steady and unsteady fluxes have been taken into account. We treat the nearest grid points as mirror points of the ghost points using a simpler ghost point approach. In the submerged boundary, wall boundary requirements are established at the ghost spots. The method's accuracy has been examined for a number of test scenarios. We provide numerical instances of supersonic flow in a circular cylinder and compare them with analytical outcomes. Furthermore, we compute time-accurate results of the compressible Navier-Stokes problems for an incidence shock across a cylinder and compare the history with earlier studies. To demonstrate the correctness of the approach, we calculated skin friction profiles and compared the results with the body fitted method. Although it works better, the current approach is based on a simplified ghost point treatment that was previously established. The results are comparable, although not being as precise as those obtained using more advanced techniques.

Chapter 1

Introduction

In place of the experiment, computational fluid dynamics (CFD) has become more and more popular. Compared to laboratory studies, CFD is a relatively cost-effective approach for solving complicated issues. Today, CFD and lab tests are used together to examine physical issues. Using CFD and computer technologies to execute the identical task and make experiment modifications can reduce the expense of performing an experiment numerous times. The quality of the experimental data still makes it more valuable, although CFD is basically used to reduce the amount of laboratory experiments required. Additionally, the investigator can add novel concepts that, without adequate financing, would have been exceedingly challenging to duplicate in a lab. CFD is gaining importance in a variety of scientific domains, including meteorology, aerospace industry and astronomy [1].

Recently, the Cartesian grid approach has gained popularity in CFD as a way to estimate flows in or over complicated geometries [2, 3, 5, 9, 13, 12, 14]. The difference between it and body-fitted structured and unstructured grid approaches may be attributed to its quicker post processing, simpler grid creation, and reduced storage and operation counts. Building higher order techniques is another benefit of the cartesian grid approach. With the help of body-fitted technique, by adding a body necessitates rewriting the whole grid, however with the Cartesian grid technique under investigation here, this may be accomplished in a few straight forward stages. The body is fixed in a small cartesian grid instead of a body-fitted organized grid, and its impact is offset by suitable circumstances at grid places close to the body surface.

When using the cartesian grid approach at bent borders, the cut-cells on The edges are not square , which makes it difficult to apply the plan [13]. This issue is not existing in the simple ghost point method because ghost points in the solid next to the border are required to satisfy symmetry criteria with respect to the boundary. But in the process, conservatism is lost.

The findings of this study are compared to other approaches for embedding solid boundaries reported in [3, 8, 12, 14]. These techniques are more intricate and better able to reproduce the solid boundary's impact. At the embedded border, Sjögreen and Petersson [3] employed linear interpolation. To cover the computation domain, H. Luo et al. [12] used a gridless method to handle the boundary surfaces and a cartesian grid as an indicator mesh.

In a research [9], the simplified ghost point technique for the 2D compressible Euler equations was confirmed using an embedded boundary consisting of a bump with a round arch within a channel. Previously, the simplified ghost point technique and a circular cylinder were used to extend the compressible equations used by Navier-Stokes [21]. The procedure ran into some issues when the body's form caused the reflected ghost points to shift in their orientation. Nonphysical behavior at these locations may cause the solution to become awkward for small time steps or tiny grids, as well as have an impact on accuracy.

1.1 Basic Definitions

1.1.1 Fluid

A material with the capacity to flow and adapt to the form of its container is called a fluid. Fluids may change shape and flow in reaction to applied forces, in contrast to solids, which have a fixed shape and volume.

1.1.2 Computational Fluid Dynamics

Computational fluid dynamics is referred to as CFD. It is a subfield of fluid mechanics that use numerical techniques and algorithms to evaluate and resolve issues pertaining to the behavior of fluids, including gases and liquids. Through the use of computer simulations, CFD entails building a computational model of a fluid flow scenario in order to comprehend and forecast the flow patterns, velocities, pressures, temperatures, and other pertinent aspects.

1.1.3 Compressible Flow

The movement of a fluid in which the density significantly varies as a result of changes in pressure and temperature is referred to as compressible flow.i.e $\rho \neq \text{constant}$.

1.1.4 Incompressible Flow

The movement of a fluid in which the density does not varies as a result of changes in pressure and temperature is referred to as incompressible flow. i.e $\rho=\text{constant}$.

1.1.5 Mach Number

The ratio of a moving object's speed to the sound speed in the environment is represented by the Mach number, which is a dimensionless quantity. It is used to express how quickly something is moving in comparison to the speed of sound.

Mathematically, it is expressed as:

$$Ma = \frac{u}{c}, \quad (1.1)$$

here, c is the sound's speed, and the fluid speed is u . we can classify mach no as:

- Flow is subsonic, if $Ma < 0.8$.
- Flow is transonic, if $0.8 < Ma < 1.3$.
- Flow is sonic, if $Ma = 1$.
- Flow is supersonic, if $1.3 < Ma < 5.0$.
- Flow is hypersonic, if $Ma > 5.0$.

1.1.6 Viscous Fluid

Viscous fluid refers to a type of fluid characterized by its resistance to flow. This resistance is due to internal friction within the fluid, which causes adjacent layers of the fluid to move at different velocities when subjected to an external force. Viscosity is the property that quantifies this resistance to flow.

1.1.7 Conservation Laws

The continuity equation and Navier-Stokes equations, which are useful in determining the equation of motion for fluids, are examples of conservation laws. It is helpful to evaluate fluid flow, and conservation law is also used to design fluid systems in fields like aerodynamics, mechanical engineering, marine science, etc. When the mass of the system stays the same, the principle is known as the conservation of mass.

1.1.8 Continuity Equation

A key idea that describes the conservation of mass in a particular system is the continuity equation. According to this, the rate at which mass changes inside a volume is equal to the net flow of mass into or out of that volume. Following are differential and integral form of continuity equation:

Differential form

$$\frac{\partial \rho^*}{\partial t} + \nabla \cdot (\rho^* \mathbf{U}) = 0, \quad (1.2)$$

here t , ρ^* and \mathbf{U} are time, density and velocity respectively.

Integral form

$$\frac{d}{dt} \int_{vol} \rho^* dV + \oint_{surf} \rho^* \mathbf{U} ds = 0, \quad (1.3)$$

ds , denotes outward normal surface element.

continuity equation for stationary control volume will get form

$$\int_{\partial\sigma} \frac{\partial \rho^*}{\partial t} dV + \int_{\partial\sigma} \rho^* \mathbf{u} \cdot \mathbf{n} dA = 0. \quad (1.4)$$

1.1.9 Momentum Equation

According to Newton's second rule of motion, the system's overall rate of momentum change is equal to the sum of all external forces β operating on the system. Mathematically,

$$\frac{d\Omega_{sys}}{dt} = \beta, \quad (1.5)$$

we employ the Reynolds transport theorem to establish a connection between the momentum in the control volume and the total rate of momentum change in the system in following equation:

$$\frac{d\Omega_{sys}}{dt} = \frac{d}{dt} \left(\int_{\chi} \rho^* \mathbf{u} dV \right) + \int_{\partial\chi} \rho^* \mathbf{u} \mathbf{u} \cdot \mathbf{n} dA, \quad (1.6)$$

when we have stationary control volume then momentum equation get form as:

$$\int_{\chi} \frac{\partial \rho^* \mathbf{U}}{\partial t} + \int_{\partial\chi} \rho^* \mathbf{u} \mathbf{u} \cdot \mathbf{n} dA = - \int_{\partial\chi} p \mathbf{n} dA + \int_{\partial\chi} \boldsymbol{\tau} dA + \int_{\chi} \rho^* \mathbf{f} dV, \quad (1.7)$$

here χ , p , $\rho^* \mathbf{f}$ and $\boldsymbol{\tau}$ represents control volume, pressure, external force density and viscous stress tensor respectively.

1.1.10 Energy Equation

According to the first law of thermodynamics, the rate of energy change in a system is equal to the sum of the rates at which heat is added (J) and work is done on the system (W) by an outside force. Mathematically,

$$\frac{d\kappa_{sys}}{dt} = J + W, \quad (1.8)$$

system's overall energy is denoted by κ_{sys} . The system's overall energy rate is expressed in terms of control volume $\kappa_\chi = \int_\chi \rho^* E^* dV$ using the Reynolds Transport Theorem. $\rho^* E^*$ is total energy per unit volume.

$$\frac{d\kappa_{sys}}{dt} = \frac{d}{dt} \left(\int_{\partial\chi} \rho^* E^* dv \right) + \int_{\partial\chi} \rho^* E^* \mathbf{u} \cdot \mathbf{n} dA, \quad (1.9)$$

for stationary control volume energy equation gets form:

$$\int_\chi \frac{\partial \rho^* E^*}{\partial t} dV + \int_{\partial\chi} \rho^* E^* \mathbf{u} \cdot \mathbf{n} dA = - \int_{\partial\chi} p \mathbf{u} \cdot \mathbf{n} dA + \int_{\partial\chi} \boldsymbol{\tau} \cdot \mathbf{n} dA + \int_\chi \rho^* \mathbf{f} \cdot \mathbf{u} dV - \int_{\partial\chi} \mathbf{q} \cdot \mathbf{n} dA, \quad (1.10)$$

here, $E^* = e + \frac{1}{2}|\mathbf{u}|^2$.

1.1.11 CFL Number

One-dimensional Courant-Friedrichs-Levy (CFL) numbers are defined as:

$$C = u' \frac{\Delta t'}{\Delta x'}, \quad (1.11)$$

where u' stands for fluid wave speed, t' for time step, and x' for grid spacing. When governing equations are discretized in space and time, this number may be seen in both viscous and inviscid fluid.

1.1.12 Von Neumann Number

Von Neuman number can be defined as:

$$VNN = \Delta t_v \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \max \left(\frac{4\mu}{3\rho}, \frac{\gamma\mu}{\rho Pr} \right). \quad (1.12)$$

1.1.13 Rankine-Hugoniot Relations

The study of shock waves in particular uses a set of conservation equations called the Rankine-Hugoniot relations. These equations explain the link between a fluid's characteristics before and after a shock wave travels through it, including its density, pressure, velocity, and temperature. Mathematically,

Conservation of Mass

$$\rho_1 A_1 V_1 = \rho_2 A_2 V_2, \quad (1.13)$$

here ρ_1 , A_1 and V_1 are the densities of the fluid, cross-sectional area of the flow and velocities of the fluid before shock wave, respectively. ρ_2 , A_2 and V_2 the densities of the fluid, cross-sectional area of the flow and velocities of the fluid after the shock wave, respectively.

Conservation of Momentum

$$P_1 + \rho_1 V_1^2 = P_2 + \rho_2 V_2^2, \quad (1.14)$$

where, P_1 and P_2 are the pressures of the fluid before and after the shock wave, respectively.

Conservation of Energy

$$\frac{1}{2}\rho_1 V_1^2 + e_1 = \frac{1}{2}\rho_2 V_2^2 + e_2, \quad (1.15)$$

where e_1 and e_2 are the specific internal energies of the fluid before and after the shock wave, respectively.

1.2 Skin Friction Coefficient

In computational fluid dynamics (CFD) and fluid mechanics, the skin friction coefficient, commonly abbreviated as "Cf," is a dimensionless quantity that describes the frictional resistance that a fluid experiences when flowing over a solid surface. It gives crucial details on the drag forces in the flow and quantifies the shear forces that are present at the fluid-solid interface. Mathematically,

$$C_f \equiv \frac{\tau_w}{\frac{1}{2}\rho_\infty u_\infty^2}, \quad (1.16)$$

here, τ_w is local shear stress and computed as:

$$\tau_w = (\tau_{xx}n_x + \tau_{xy}n_y) n_y - (\tau_{xy}n_x + \tau_{yy}n_y) n_x, \quad (1.17)$$

where, x and y elements of unit normal vectors are n_x and n_y . Here, τ_{xy} , τ_{xx} and τ_{yy} show that viscous tensor element.

1.2.1 MUSCL Scheme

It is a FV approach that is used to improve the numerical method's order and prevent oscillations. The acronym "MUSCL [21]" which stands for Monotonic Upstream-Centered Scheme for Conservation Laws, was coined by Bram Van Leer in a seminal article.

1.2.2 Total Variation Diminishing

TVD, or total variation diminishing, is a two-level strategy. We design a grid function M for which:

$$TV(G) = \sum_{-\infty}^{\infty} |G_{i+1} - M_i|, \quad (1.18)$$

for each given collection of data G^n , the values G^{n+1} computed using the method fulfill:

$$TV(G^{n+1}) \leq TV(G^n). \quad (1.19)$$

1.2.3 Roe's Approximate Riemann Solver

A numerical technique used in computational fluid dynamics (CFD) to estimate the solution of the Riemann problem a fundamental idea in fluid dynamics is called Roe's estimate riemann solver. Solving equations at the border between two distinct fluid conditions is known as the riemann issue. In order to effectively compute the numerical fluxes across cell interfaces in a numerical simulation, Philip Roe created the approximate riemann solver.

1.2.4 Cartesian Grid Method

The Cartesian grid method is used to computationally solve partial differential equations (PDEs). Due of its straightforward programming, quicker grid creation, and reduced computing effort, this approach is quite prevalent nowadays. This approach does not allow embedded objects to pass through grid points and instead treats embedded boundaries as ghost points.

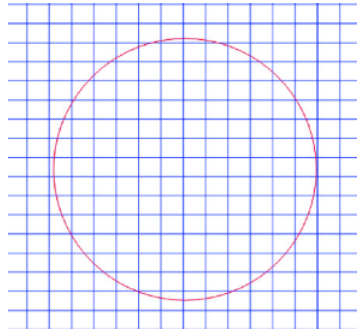


Figure 1.1: Cartesian grid method.

1.2.5 Body Fitted Grid Method

Additionally, partial differential equations are solved using it. Boundary points refer to grid locations in the framework that fits the body.

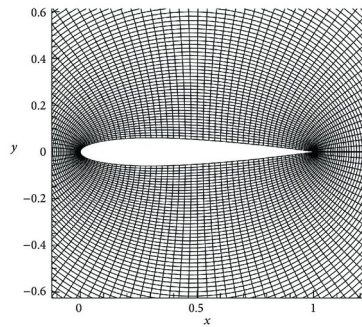


Figure 1.2: An example of Body fitted grid.

technique. Embedded objects traverse grid points in this way. Its benefit is that it may resolve difficult geometry issues.

1.3 Comparing the Cell-Vertex as well as Cell-Centred Methodologies

Our concentration depend on FVM since their property for conservation is necessary for effective shock capture. The field of computation can be made discrete using cells as well as elements, or nodes. There are two possible approaches to display the solution data on the computational domain.

- (1) Cell-Centred technique
- (2) Cell-Vertex technique

1.3.1 Cell-Centred Scheme

By using this scheme Data is positioned at the cell centroid and expressed as cell averages in cell-centered schemes. At each of the cell's faces, the fluxes are computed.

1.3.2 Cell-Vertex Scheme

In contrast to the cell-centered , the cell-vertex system places the variables at the points. The cell-vertex system is better in some circumstances, while the cell-centered design is better in others. When comparing the accurateness of cell-centered and cell-vertex schemes, the former may become inconsistent whereas the latter is first-order reliability even on distorted nodes.

1.3.3 Boundary Conditions

We discriminate between solid boundaries, supersonic outflows, and inflows as boundary conditions. Dirichlet boundary conditions for each flow variable are given for the conservatism quantities at the point of intake for supersonic supply in the x-direction.

The flow is supersonic, thus no boundary constraints need to be put in place at the exit. Boundary constraints for extrapolation are assumed. we can see in below figure

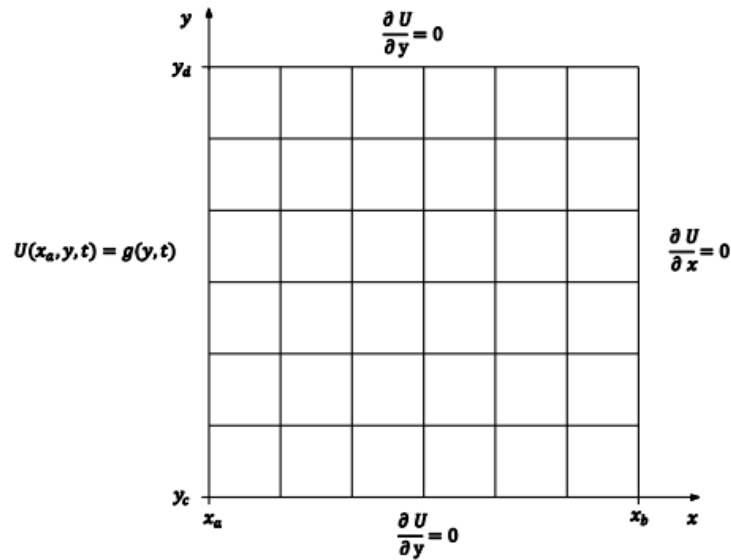


Figure 1.3: Domain and boundary conditions diagram [21].

The borders are later described as east, west, toward the north, and southward with the west edge serving as a barrier to influx.

Chapter 2

Governing Equations

2.1 Compressible Navier-Stokes Equations

Consider conservative form of compressible Navier-Stokes equations in 2D is

$$\frac{\partial Q_o}{\partial t} + \frac{\partial H_c^*}{\partial x} + \frac{\partial G_c^*}{\partial y} = \frac{\partial H_v^*}{\partial x} + \frac{\partial G_v^*}{\partial y}, \quad (2.1)$$

here,

$$Q_o = \begin{bmatrix} \rho_o \\ \rho_o u_o \\ \rho_o v_o \\ \rho_o E \end{bmatrix}, H_c^* = \begin{bmatrix} \rho_o u_o \\ \rho_o u_o^2 + p_o \\ \rho_o u_o v_o \\ (\rho_o E + p_o) u_o \end{bmatrix}, G_c^* = \begin{bmatrix} \rho_o v_o \\ \rho_o u_o v_o \\ \rho_o v_o^2 + p_o \\ (\rho_o E + p_o) v_o \end{bmatrix}, \quad (2.2)$$

$$H_v^* = \begin{bmatrix} 0 \\ 2\mu \frac{\partial u_o}{\partial x} - \frac{2}{3}\mu \left(\frac{\partial u_o}{\partial x} + \frac{\partial v_o}{\partial y} \right) \\ \mu \left(\frac{\partial v_o}{\partial x} + \frac{\partial u_o}{\partial y} \right) \\ \tau_{xx} u_o + \tau_{yx} v_o + k \frac{\partial T}{\partial x} \end{bmatrix}, G_v^* = \begin{bmatrix} \mu \left(\frac{\partial v_o}{\partial x} + \frac{\partial u_o}{\partial y} \right) \\ 2\mu \frac{\partial v_o}{\partial y} - \frac{2}{3}\mu \left(\frac{\partial u_o}{\partial x} + \frac{\partial v_o}{\partial y} \right) \\ \tau_{xy} u_o + \tau_{yy} v_o + k \frac{\partial T}{\partial y} \end{bmatrix}, \quad (2.3)$$

density is denoted by ρ_o , components of velocity in x and y directions are u_o and v_o respectively, pressure is p_o , μ denotes viscosity and E indicates energy. Tensor components are defined as:

$$\tau_{xx} = 2\mu \frac{\partial u_o}{\partial x} - \frac{2}{3}\mu \left(\frac{\partial u_o}{\partial x} + \frac{\partial v_o}{\partial y} \right), \quad (2.4)$$

$$\tau_{yy} = 2\mu \frac{\partial v_o}{\partial y} - \frac{2}{3}\mu \left(\frac{\partial u_o}{\partial x} + \frac{\partial v_o}{\partial y} \right), \quad (2.5)$$

$$\tau_{xy} = \tau_{yx} = \mu \left(\frac{\partial v_o}{\partial x} + \frac{\partial u_o}{\partial y} \right). \quad (2.6)$$

pressure P_o and temperature T for ideal gas are given as follows:

$$T = \frac{p_o}{R\rho_o}, \quad (2.7)$$

$$p_o = (\gamma - 1) \left(\rho_o E - \frac{1}{2} \rho_o (u_o^2 + v_o^2) \right), \quad (2.8)$$

here, $\gamma = \frac{c_p}{c_v}$ is ratio of specific heat for constant volume and pressure. R is constant of gas air. Using Sutherland's law, the viscosity is computed and depending on temperature:

$$\mu(t) = \mu_\infty \left(\frac{T}{T_\infty} \right)^{1.5} \frac{T_\infty + 110K}{T + 110K}. \quad (2.9)$$

Here μ_∞ and T_∞ are reference viscosity and temperature respectively. Sound's speed is calculated as

$$c^2 = \gamma \frac{p_o}{\rho_o} \quad (2.10)$$

where $\gamma = 1.4$ is ratio for specific heat and $Pr = 0.72$ for air. Prandtl number, which relates thermal conductivity to viscosity, will be used as a constant.

Chapter 3

Numerical Methods

The fundamental ideas of numerical techniques are presented in this chapter. Finite volume approaches are our main emphasis. The discretization of the fundamental equations is a crucial step. In this chapter, explicit techniques are also introduced.

3.1 Basic Discretization

There are actually three main categories of discretization techniques that could potentially be used to address actual-world issues monitored by PDEs.

- (1) Finite Element Method (FEM)
- (2) Finite Difference Method (FDM)
- (3) Finite Volume Method (FVM)

FEM is one of the effective numerical methods approaches. The variables in FEM are distributed throughout the elements. FEM has the benefit of being able to handle complicated geometry more readily than FDM because of this.

The values are placed at a grid's nodes in FDM. The benefit includes the applicability to computer programs in addition to higher level approaches. On cartesian grids, with the exception of boundary treatments, the FVM and FDM are the same for both the initial and second order discretizations.

In FVM, the amount is averaged over all of the sections. using a variable that is positioned in the center of each cell. Integrals that guarantee conservation qualities and capture discontinuities are used in the derivation. The difficulty of raising the method's order is a drawback of this discretization. Due to low order restrictions for high accuracy requirements, it is ineffective.

3.2 Spatial Discretization

The domain is the rectangular box $[-L_{x_1}, L_{x_1}] \times [-L_{y_1}, L_{y_1}]$ and $[N \times M]$ Cartesian grid having equal grid $\Delta x_1 = \frac{2L_{x_1}}{N-1}, \Delta y_1 = \frac{2L_{y_1}}{M-1}$ cartesian values for the grid point (i_1, j_1) are (x_{1i_1}, y_{1j_1}) , where $x_{1i_1} = -L_{x_1} + (i_1 - 1)\Delta x_1, i_1 = 1, 2, \dots, N$ and $y_{1j_1} = -L_{y_1} + (j_1 - 1)\Delta y_1, j_1 = 1, 2, \dots, M$. The semi-discretization of 2D compressible Navier-Stokes equation using finite difference method is:

$$\frac{d\mathbf{Q}_{o(i_1, j_1)}}{dt} = - \frac{\mathbf{H}_{i_1+\frac{1}{2}, j_1}^{*c} - \mathbf{H}_{i_1-\frac{1}{2}, j_1}^{*c}}{\Delta x_1} - \frac{\mathbf{G}_{i_1, j_1+\frac{1}{2}}^{*c} - \mathbf{G}_{i_1, j_1-\frac{1}{2}}^{*c}}{\Delta y_1} + \frac{\mathbf{H}_{i_1+\frac{1}{2}, j_1}^{*v} - \mathbf{H}_{i_1-\frac{1}{2}, j_1}^{*v}}{\Delta x_1} + \frac{\mathbf{G}_{i_1, j_1+\frac{1}{2}}^{*v} - \mathbf{G}_{i_1, j_1-\frac{1}{2}}^{*v}}{\Delta y_1}, \quad (3.1)$$

$\mathbf{Q}_{o(i_1, j_1)}$ is an approximation of \mathbf{Q}_o at (x_{1i_1}, y_{1j_1}) . Average of \mathbf{Q}_o at the cell is given by:

$$\sigma_{o(i_1, j_1)} = \left[x_{1i_1} - \frac{\Delta x_1}{2}, x_{1i_1} + \frac{\Delta x_1}{2} \right] \times \left[y_{1j_1} - \frac{\Delta y_1}{2}, y_{1j_1} + \frac{\Delta y_1}{2} \right]. \quad (3.2)$$

\mathbf{H}^{*c} and \mathbf{G}^{*c} the convective parts are the numerical fluxes For 2D compressible Navier Stokes Equation .

3.3 Local Lax-Friedrichs Method

The numerical flux for the convective parts for the local Lax-Friedrichs method is described as:

$$\mathbf{H}_{i_1+\frac{1}{2}, j_1}^{*ILF} = \frac{1}{2} [\mathbf{H}(\mathbf{Q}_{o(i_1, j_1)}) + \mathbf{H}(\mathbf{Q}_{o(i_1+1, j_1)} - \max(|u_{o(i_1+1, j_1)}| + c_{o(i_1+1, j_1)}, |u_{o(i_1, j_1)}| + c_{o(i_1, j_1)})))(\mathbf{Q}_{o(i_1+1, j_1)} - \mathbf{Q}_{o(i_1, j_1)})] \quad (3.3)$$

$$\mathbf{G}_{i_1, j_1+\frac{1}{2}}^{*ILF} = \frac{1}{2} [\mathbf{G}(\mathbf{Q}_{o(i_1, j_1)}) + \mathbf{G}(\mathbf{Q}_{o(i_1, j_1+1)} - \max(|v_{o(i_1, j_1+1)}| + c_{o(i_1, j_1+1)}, |v_{o(i_1, j_1)}| + c_{o(i_1, j_1)})))(\mathbf{Q}_{o(i_1, j_1+1)} - \mathbf{Q}_{o(i_1, j_1)})]. \quad (3.4)$$

Except in extreme cases where accuracy drops to first order, the MUSCL technique utilizing the minmod limiter yields second order precision and prevents unwanted oscillations. Here $\mathbf{Q}_{o(i_1, j_1)}$ is replaced by $\mathbf{Q}_{o(i_1+\frac{1}{2}, j_1)}^L$ and $\mathbf{Q}_{o(i_1+1, j_1)}$ is replaced by $\mathbf{Q}_{o(i_1+\frac{1}{2}, j_1)}^R$ [10].

$$\mathbf{Q}_{\circ(i_1+\frac{1}{2},j_1)}^L = \mathbf{Q}_{\circ(i_1,j_1)} + \frac{1}{2} \minmod(\mathbf{Q}_{\circ(i_1,j_1)} - \mathbf{Q}_{\circ(i_1-1,j_1)}, \mathbf{Q}_{\circ(i_1+1,j_1)} - \mathbf{Q}_{\circ(i_1,j_1)}), \quad (3.5)$$

$$\mathbf{Q}_{\circ(i_1+\frac{1}{2},j_1)}^R = \mathbf{Q}_{\circ(i_1+1,j_1)} + \frac{1}{2} \minmod(\mathbf{Q}_{\circ(i_1+2,j_1)} - \mathbf{Q}_{\circ(i_1+1,j_1)}, \mathbf{Q}_{\circ(i_1+1,j_1)} - \mathbf{Q}_{\circ(i_1,j_1)}), \quad (3.6)$$

here, mimmod limiter is defined as:

$$\minmod(m, n) = \begin{cases} m & \text{if } |m| \leq |n| \text{ and } mn > 0 \\ n & \text{if } |n| < |m| \text{ and } mn > 0 \\ 0 & \text{if } mn \leq 0 \end{cases} . \quad (3.7)$$

Now we will do the discretization of viscous terms \mathbf{H}_v^* and \mathbf{G}_v^* using central finite difference,

$$\mathbf{H}_v^* = \begin{bmatrix} 0 \\ 2\mu \frac{\partial u_{\circ}}{\partial x} - \frac{2}{3}\mu \left(\frac{\partial u_{\circ}}{\partial x} + \frac{\partial v_{\circ}}{\partial y} \right) \\ \mu \left(\frac{\partial v_{\circ}}{\partial x} + \frac{\partial u_{\circ}}{\partial y} \right) \\ \tau_{xx} u_{\circ} + \tau_{yx} v_{\circ} + k \frac{\partial T}{\partial x} \end{bmatrix}, \quad (3.8)$$

here, $H_{v,1}^* = 0$, $H_{v,2}^* = 2\mu \frac{\partial u_{\circ}}{\partial x} - \frac{2}{3}\mu \left(\frac{\partial u_{\circ}}{\partial x} + \frac{\partial v_{\circ}}{\partial y} \right)$, $H_{v,3}^* = \mu \left(\frac{\partial v_{\circ}}{\partial x} + \frac{\partial u_{\circ}}{\partial y} \right)$ and

$$H_{v,4}^* = \tau_{xx} u_{\circ} + \tau_{yx} v_{\circ} + k \frac{\partial T}{\partial x}.$$

Now, we will calculate $(H_v^*)_{i_1+\frac{1}{2},j_1}$, So

$$(H_{v,1}^*)_{i_1+\frac{1}{2},j_1} = 0. \quad (3.9)$$

$$(H_{v,2}^*)_{i_1+\frac{1}{2},j_1} = 2\mu \left(\frac{\partial u_{\circ}}{\partial x} \right)_{i_1+\frac{1}{2},j_1} - \frac{2}{3}\mu \left[\left(\frac{\partial u_{\circ}}{\partial x} \right)_{i_1+\frac{1}{2},j_1} + \left(\frac{\partial v_{\circ}}{\partial x} \right)_{i_1+\frac{1}{2},j_1} \right],$$

$$(H_{v,2}^*)_{i_1+\frac{1}{2},j_1} = 2\mu \left[\frac{u_{\circ(i_1+1,j_1)} - u_{\circ(i_1,j_1)}}{\Delta x_1} \right] - \frac{2}{3}\mu \left[\frac{u_{\circ(i_1+1,j_1)} - u_{\circ(i_1,j_1)}}{\Delta x_1} + \frac{\frac{\partial v_{\circ}}{\partial y} \Big|_{i_1+\frac{1}{2},j_1+1} + \frac{\partial v_{\circ}}{\partial y} \Big|_{i_1+\frac{1}{2},j_1-1}}{2\Delta y_1} \right],$$

$$(H_{v,2}^*)_{i_1+\frac{1}{2},j_1} = \frac{4}{3\Delta x_1} (u_{\circ(i_1+1,j_1)} - u_{\circ(i_1,j_1)}) - \frac{1}{3}\mu \left[\frac{v_{\circ(i_1+1,j_1+1)} - v_{\circ(i_1+1,j_1-1)}}{2\Delta y_1} + \frac{v_{\circ(i_1,j_1+1)} - v_{\circ(i_1,j_1-1)}}{2\Delta y_1} \right]. \quad (3.10)$$

$$(H_{v,3}^*)_{i_1+\frac{1}{2},j_1} = \mu \left(\frac{\partial v_{\circ}}{\partial x_1} + \frac{\partial u_{\circ}}{\partial y_1} \right),$$

$$(H_{v,3}^*)_{i_1+\frac{1}{2},j_1} = \mu \left[\frac{v_{\circ(i_1+1,j_1)} - v_{\circ(i_1,j_1)}}{\Delta x_1} + \frac{\mu}{2} \left[\frac{u_{\circ(i_1+1,j_1+1)} - u_{\circ(i_1+1,j_1-1)}}{2\Delta y_1} + \frac{u_{\circ(i_1,j_1+1)} - u_{\circ(i_1,j_1-1)}}{2\Delta y_1} \right] \right],$$

$$\begin{aligned} \left(H_{v,3}^*\right)_{i_1+\frac{1}{2},j_1} &= \frac{1}{4\Delta y_1} \mu \left[\left(u_{o(i_1+1,j_1+1)} - u_{o(i_1+1,j_1-1)}\right) + \left(u_{o(i_1,j_1+1)} - u_{o(i_1,j_1-1)}\right) \right] \\ &\quad + \frac{1}{\Delta x_1} \mu \left(v_{o(i_1+1,j_1)} - v_{o(i_1,j_1)}\right). \end{aligned} \quad (3.11)$$

$$\begin{aligned} \left(H_{v,4}^*\right)_{i_1+\frac{1}{2},j_1} &= 2\mu u_o \frac{\partial u_o}{\partial x} - \frac{2}{3}\mu u_o \left(\frac{\partial u_o}{\partial x} + \frac{\partial v_o}{\partial y}\right) + v_o \mu \left(\frac{\partial v_o}{\partial x} + \frac{\partial u_o}{\partial y}\right) + K \frac{\partial T}{\partial x}, \\ \left(H_{v,4}^*\right)_{i_1+\frac{1}{2},j_1} &= \frac{1}{2} \left(u_{o(i_1+1,j_1)} + u_{o(i_1,j_1)}\right) \left(H_{v,2}^*\right)_{i_1+\frac{1}{2},j_1} + \frac{1}{2} \left(v_{o(i_1+1,j_1)} + v_{o(i_1,j_1)}\right) \left(H_{v,3}^*\right)_{i_1+\frac{1}{2},j_1} + \\ &\quad \frac{1}{\Delta x_1} K \left(T_{i_1+1,j_1} - T_{i_1,j_1}\right). \end{aligned} \quad (3.12)$$

$$\left(H_{v,4}^*\right)_{i_1+\frac{1}{2},j_1} = \begin{bmatrix} 0 \\ \frac{4}{3\Delta x_1} \left(u_{o(i_1+1,j_1)} - u_{o(i_1,j_1)}\right) - \frac{1}{3}\mu \left[\frac{v_{o(i_1+1,j_1+1)} - v_{o(i_1+1,j_1-1)}}{2\Delta y_1} + \frac{v_{o(i_1,j_1+1)} - v_{o(i_1,j_1-1)}}{2\Delta y_1}\right] \\ \frac{1}{4\Delta y_1} \mu \left[\left(u_{o(i_1+1,j_1+1)} - u_{o(i_1+1,j_1-1)}\right) + \left(u_{o(i_1,j_1+1)} - u_{o(i_1,j_1-1)}\right)\right] + \frac{1}{\Delta x_1} \mu \left(v_{o(i_1+1,j_1)} - v_{o(i_1,j_1)}\right) \\ \frac{1}{2} \left(u_{o(i_1+1,j_1)} + u_{o(i_1,j_1)}\right) \left(H_{v,2}^*\right)_{(i_1+\frac{1}{2},j_1)} + \frac{1}{2} \left(v_{o(i_1+1,j_1)} + v_{o(i_1,j_1)}\right) \left(H_{v,3}^*\right)_{(i_1+\frac{1}{2},j_1)} + \frac{1}{\Delta x_1} K \left(T_{i_1+1,j_1} - T_{i_1,j_1}\right) \end{bmatrix} \quad (3.13)$$

$$\mathbf{G}_v^* = \begin{bmatrix} 0 \\ \mu \left(\frac{\partial v_o}{\partial x} + \frac{\partial u_o}{\partial y}\right) \\ 2\mu \left(\frac{\partial v_o}{\partial y}\right) - \frac{2}{3}\mu \left(\frac{\partial u_o}{\partial x} + \frac{\partial v_o}{\partial y}\right) \\ \tau_{xy} u_o + \tau_{yy} v_o + K \frac{\partial T}{\partial y} \end{bmatrix}, \quad (3.14)$$

here , $G_{v,1}^* = 0$, $G_{v,2}^* = \mu \left(\frac{\partial v_o}{\partial x} + \frac{\partial u_o}{\partial y}\right)$, $G_{v,3}^* = 2\mu \left(\frac{\partial v_o}{\partial y}\right) - \frac{2}{3}\mu \left(\frac{\partial u_o}{\partial x} + \frac{\partial v_o}{\partial y}\right)$ and

$$G_{v,4}^* = \tau_{xy} u_o + \tau_{yy} v_o + K \frac{\partial T}{\partial y}.$$

We will calculate $(G_v^*)_{i_1,j_1+\frac{1}{2}}$,

$$(G_{v,1}^*)_{i_1,j_1+\frac{1}{2}} = 0. \quad (3.15)$$

$$(G_{v,2}^*)_{i_1,j_1+\frac{1}{2}} = \mu \frac{\partial v_o}{\partial x} \Big|_{i_1,j_1+\frac{1}{2}} + \mu \frac{\partial u_o}{\partial y} \Big|_{i_1,j_1+\frac{1}{2}},$$

$$(G_{v,2}^*)_{i_1,j_1+\frac{1}{2}} = \mu \left[\frac{1}{2} \left(v_{o(i_1+1,j_1+\frac{1}{2})} + v_{o(i_1-1,j_1+\frac{1}{2})}\right)\right] + \mu \left[\frac{u_{o(i_1,j_1+1)} - u_{o(i_1,j_1)}}{\Delta y_1}\right],$$

$$(G_{v,2}^*)_{i_1,j_1+\frac{1}{2}} = \frac{1}{2} \mu \left[\frac{v_{o(i_1+1,j_1+1)} - v_{o(i_1-1,j_1+1)}}{2\Delta x_1} + \frac{v_{o(i_1+1,j_1)} - v_{o(i_1-1,j_1)}}{2\Delta x_1}\right] + \mu \left[\frac{u_{o(i_1,j_1+1)} - u_{o(i_1,j_1)}}{\Delta y_1}\right],$$

$$(G_{v,2}^*)_{i_1, j_1 + \frac{1}{2}} = \frac{1}{4\Delta x_1} \mu [v_{o(i_1+1, j_1+1)} - v_{o(i_1-1, j_1+1)} + v_{o(i_1+1, j_1)} - v_{o(i_1-1, j_1)}] + \mu \left[\frac{u_{o(i_1, j_1+1)} - u_{o(i_1, j_1)}}{\Delta y_1} \right]. \quad (3.16)$$

$$(G_{v,3}^*)_{i_1, j_1 + \frac{1}{2}} = 2\mu \frac{\partial v_o}{\partial y} \Big|_{i_1, j_1 + \frac{1}{2}} - \frac{2}{3} \mu \left(\frac{\partial u_o}{\partial x} \Big|_{i_1, j_1 + \frac{1}{2}} + \frac{\partial v_o}{\partial y} \Big|_{i_1, j_1 + \frac{1}{2}} \right),$$

$$(G_{v,3}^*)_{i_1, j_1 + \frac{1}{2}} = 2\mu \left[\frac{v_{o(i_1, j_1+1)} - v_{o(i_1, j_1)}}{\Delta y_1} \right] - \frac{2}{3} \mu \left[\frac{1}{2} \frac{u_{o(i_1+1, j_1+1)} - u_{o(i_1-1, j_1+1)}}{2\Delta x_1} + \frac{1}{2} \frac{u_{o(i_1+1, j_1)} - u_{o(i_1-1, j_1)}}{2\Delta x_1} + \frac{v_{o(i_1, j_1+1)} - v_{o(i_1, j_1)}}{\Delta y_1} \right],$$

$$(G_{v,3}^*)_{i_1, j_1 + \frac{1}{2}} = \frac{4}{3\Delta y_1} \mu (v_{o(i_1, j_1+1)} - v_{o(i_1, j_1)}) - \frac{1}{6\Delta x_1} \mu [u_{o(i_1+1, j_1+1)} - u_{o(i_1-1, j_1+1)} + (u_{o(i_1+1, j_1)} - u_{o(i_1-1, j_1)})]. \quad (3.17)$$

$$(G_{v,3}^*)_{i_1, j_1 + \frac{1}{2}} = [\tau_{xy} u_o + \tau_{yy} v_o + K \frac{\partial T}{\partial y}]_{i_1, j_1 + \frac{1}{2}},$$

$$(G_{v,4}^*)_{i_1, j_1 + \frac{1}{2}} = \mu \left(\frac{\partial v_o}{\partial x} \Big|_{i_1, j_1 + \frac{1}{2}} + \frac{\partial u_o}{\partial y} \Big|_{i_1, j_1 + \frac{1}{2}} \right) u_o \Big|_{i_1, j_1 + \frac{1}{2}} + v_o \Big|_{i_1, j_1 + \frac{1}{2}} 2\mu \frac{\partial v_o}{\partial y} \Big|_{i_1, j_1 + \frac{1}{2}} - \frac{2}{3} \mu \left(\frac{\partial u_o}{\partial x} \Big|_{i_1, j_1 + \frac{1}{2}} + \frac{\partial v_o}{\partial y} \Big|_{i_1, j_1 + \frac{1}{2}} \right) + K \frac{\partial T}{\partial y} \Big|_{i_1, j_1 + \frac{1}{2}},$$

$$(G_{v,4}^*)_{i_1, j_1 + \frac{1}{2}} = \frac{1}{2} (u_{o(i_1, j_1+1)} - u_{o(i_1, j_1)}) (G_{v,2}^*)_{i_1, j_1 + \frac{1}{2}} + \frac{1}{2} (v_{o(i_1, j_1+1)} - v_{o(i_1, j_1)}) (G_{v,3}^*)_{i_1, j_1 + \frac{1}{2}} + K \left(\frac{T_{i_1, j_1+1} - T_{i_1, j_1}}{\Delta y_1} \right), \quad (3.18)$$

hence,

$$(G_v^*)_{i_1, j_1 + \frac{1}{2}} = \begin{bmatrix} 0 \\ \frac{1}{4\Delta x_1} \mu (v_{o(i_1+1, j_1+1)} - v_{o(i_1-1, j_1+1)} + v_{o(i_1+1, j_1)} - v_{o(i_1-1, j_1)}) - \frac{1}{\Delta y} \mu [u_{o(i_1, j_1+1)} - u_{o(i_1, j_1)}] \\ \frac{4}{3\Delta y_1} \mu (v_{o(i_1, j_1+1)} - v_{o(i_1, j_1)}) + \frac{1}{6\Delta x_1} \mu [(u_{o(i_1+1, j_1+1)} - u_{o(i_1-1, j_1+1)}) + (u_{o(i_1+1, j_1)} - u_{o(i_1-1, j_1)})] \\ \frac{1}{2} (u_{o(i_1, j_1+1)} + u_{o(i_1, j_1)}) (G_{v,2}^*)_{i_1, j_1 + \frac{1}{2}} + \frac{1}{2} (v_{o(i_1, j_1+1)} + v_{o(i_1, j_1)}) (G_{v,3}^*)_{i_1, j_1 + \frac{1}{2}} + \frac{1}{\Delta y_1} K (T_{i_1, j_1+1} - T_{i_1, j_1}) \end{bmatrix} \quad (3.19)$$

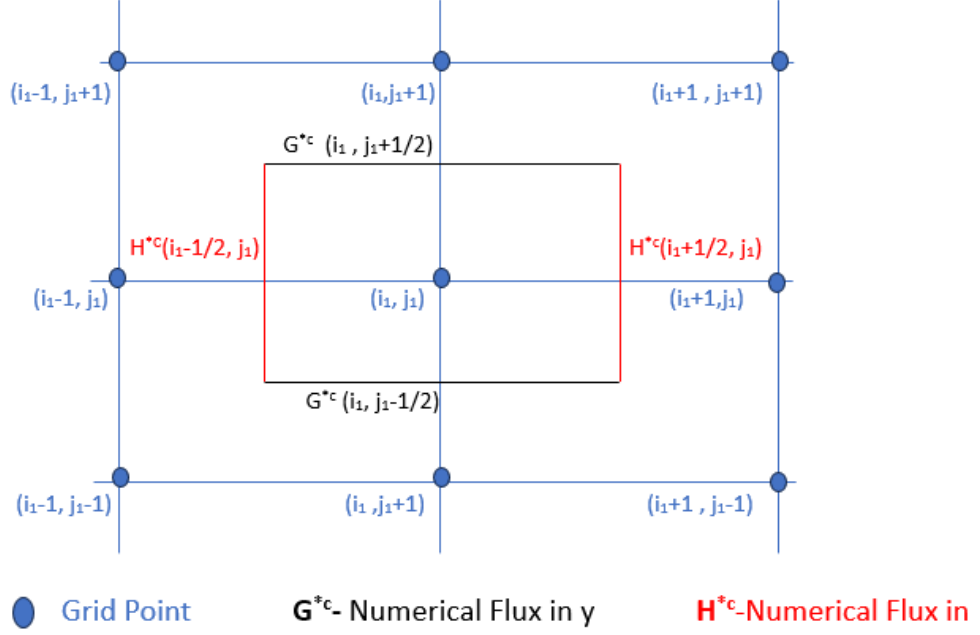


Figure 3.1: Diagram exhibiting the Cartesian domain's numerical fluxes.

3.4 Boundary Conditions Discretization

Here, first-order boundary conditions are employed since their robustness has been established. Second order conditions can also be applied, however in such scenario, each iteration should verify if the condition is physical or not. Since we are thinking about the inflow boundary condition for supersonic flow, we define variables at the inlet:

$$\mathbf{Q}_o(1, j_1) = \mathbf{Q}_{o(ref)}, \quad (3.20)$$

also,

$$\rho_o(1, j_1) = \rho_{o(ref)}, \quad p_o(1, j_1) = p_{o(ref)}, \quad u_o(1, j_1) = u_{o(ref)}, \quad v_o(1, j_1) = 0 \quad (3.21)$$

Extrapolation is employed here.

Regarding the eastward outflow boundary:

$$\mathbf{Q}_o(N, j_1) = \mathbf{Q}_o(N - 1, j_1). \quad (3.22)$$

Regarding the northward outflow boundary:

$$\mathbf{Q}_o(i_1, M) = \mathbf{Q}_o(i_1, M - 1). \quad (3.23)$$

Regarding the southward outflow boundary:

$$\mathbf{Q}_o(i_1, 1) = \mathbf{Q}_o(i_1, 2). \quad (3.24)$$

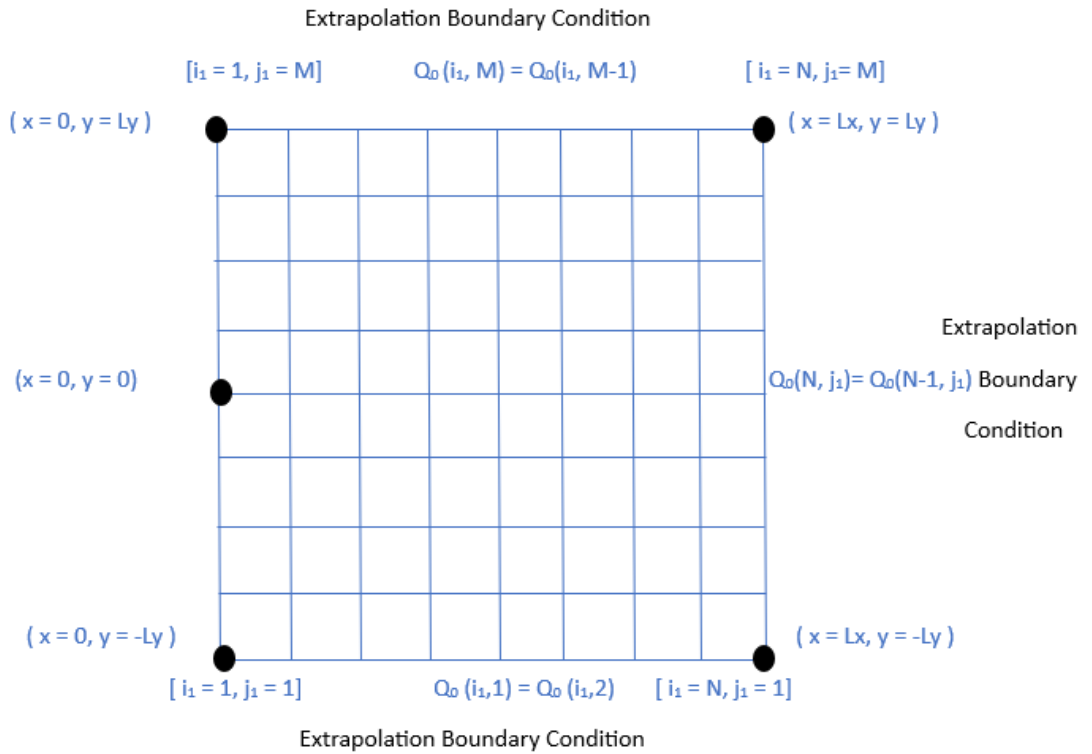


Figure 3.2: Domain sketch with discretized boundary conditions.

3.5 Temporal Discretization

The options for selecting a time-integrating technique are numerous. The machine utilized for the computation, the required precision, and the computation's runtime should all be taken into account. Multistep techniques like the Adam Bashforth approach can be utilized if the computer has a lot of fast memory [4]. Runge-Kutta methods are other techniques that use less memory but more CPU computations. The

third order Runge-Kutta (RK3) technique is a widely used option. The approach is stable enough to handle a pretty big time step, and the precision is sufficient for most purposes. Two distinct RK3 techniques have been examined.

In computational fluid dynamics, numerical techniques with less random-access memory (RAM) are crucial. To fit an issue in a cache—a high-speed memory that is closely connected to the central processor unit—minimum memory use is required. In order to fit an issue into a cache more successfully, one is frequently ready to perform an additional number of floating point operations because efficient cache utilization is such a crucial component of numerical code execution.

We compute $\frac{\partial \mathbf{Q}_o}{\partial t} = \mathbf{R}(\mathbf{Q}_o)$ for every step of time integrating approach. where \mathbf{R} is the residual, or the right side of equation (3.1), for all inside grid points and \mathbf{Q}_o is the vector containing the conservative variables.

here, residual of the 2D compressible Navier-Stokes equations is,

$$\mathbf{R}_{i_1, j_1} = -\frac{\mathbf{H}^{*c}_{i_1+\frac{1}{2}, j_1} - H^{*c}_{i_1-\frac{1}{2}, j_1}}{\Delta x_1} - \frac{\mathbf{G}^{*c}_{i_1, j_1+\frac{1}{2}} - \mathbf{G}^{*c}_{i_1, j_1-\frac{1}{2}}}{\Delta y_1} + \frac{\mathbf{H}^{*v}_{i_1+\frac{1}{2}, j_1} - \mathbf{H}^{*v}_{i_1-\frac{1}{2}, j_1}}{\Delta x_1} + \frac{\mathbf{G}^{*v}_{i_1, j_1+\frac{1}{2}} - \mathbf{G}^{*v}_{i_1, j_1-\frac{1}{2}}}{\Delta y_1}.$$

The Runge-Kutta techniques with total variation declining (TVD) provide an additional option. Since this is a strong stability preserving approach (SSP), they maintain the TVD property of the procedure [2]. The argument that TVD Runge-Kutta techniques are vital for such applications is verified when non-TVD but linearly stable Runge-Kutta temporal discretization may induce oscillations even for TVD spatial discretization [6].

Therefore, the TVD RK3 technique is utilized to ensure that the discretization's TVD property is maintained.

Thus, TVD RK3 [6] is employed and computed as:

$$\begin{aligned} \mathbf{Q}_o^{(1)} &= \mathbf{Q}_o^n + \Delta t_1 \mathbf{R}(\mathbf{Q}_o^n), \\ \mathbf{Q}_o^{(2)} &= \frac{3}{4} \mathbf{Q}_o^n + \frac{1}{4} \mathbf{Q}_o^{(1)} + \frac{1}{4} \Delta t_1 \mathbf{R}(\mathbf{Q}_o^{(1)}), \\ \mathbf{Q}_o^{(n+1)} &= \frac{1}{3} \mathbf{Q}_o^n + \frac{2}{3} \mathbf{Q}_o^{(2)} + \frac{2}{3} \Delta t_1 \mathbf{R}(\mathbf{Q}_o^{(2)}). \end{aligned} \quad (3.25)$$

3.6 Analysis of a Numerical Scheme's Stability

The stability of time-based iterations is determined by the discretization technique's characteristics and the eigenvalues of the flux vector's Jacobian matrices. Since a very long step cannot be made, Von-Neumann number (VNN) and Courant-Friedrichs-Lewy

(CFL) are employed. Given the 2D compressible Navier-Stokes equations, the following formulas are used to determine CFL and VNN:

$$CFL = \Delta t_1 \left(\frac{|u_o| + c}{\Delta x_1}, \frac{|v_o| + c}{\Delta y_1} \right), \quad (3.26)$$

$$VNN = \Delta t_1 \left(\frac{1}{\Delta x_1^2} + \frac{1}{\Delta y_1^2} \right) \max \left(\frac{4}{3} \frac{\mu}{\rho_o}, \frac{\gamma \mu}{\rho_o Pr} \right), \quad (3.27)$$

CFL is changed from 0.5 to 0.25 while VNN is fixed at 0.4. At each step, the time step sizes Δt_I and Δt_v are calculated using CFL and VNN the least of both is selected as Δt in RK3.

3.7 Roe's Approximate Riemann Solver

To compute the numerical fluxes across cell interfaces in a numerical simulation, Philip Roe created the Approximate Riemann Solver in 1981. The technique is based on employing a local Jacobian matrix to linearize the flow vector. This linearized approximation is then used to the cell interfaces to calculate the numerical fluxes. Defining the vector of conserved quantities as follow

$$\mathbf{Q}_o = \begin{bmatrix} \rho_o \\ \rho_o u_o \\ \rho_o e_T \end{bmatrix}, \quad (3.28)$$

the terms u_1, u_2 , and u_3 may be used to refer to different components of \mathbf{Q}_o . These terms stand for mass, momentum, and total energy per unit volume, respectively.

Defining the flux vector as follows

$$\mathbf{F}_o = \begin{bmatrix} \rho_o u_o \\ \rho_o u_o^2 + P_o \\ (\rho_o e_T + P_o) u_o \end{bmatrix} = \begin{bmatrix} \rho_o u_o \\ \rho_o u_o^2 + P_o \\ \rho_o H_T u_o \end{bmatrix}, \quad (3.29)$$

The terms f_1, f_2 , and f_3 may be used to refer to the component of \mathbf{F}_o . These terms stand for mass flux, momentum flux plus pressure force, and total energy flux pressure work, respectively. Despite being referred to as the flow vector, \mathbf{F}_o also takes pressure effects into account. Although they are not technically fluxes, the pressure effects on momentum and energy are frequently handled as such. However, if you'd want, you may separate the contributions from pressure and flux as:

$$\mathbf{F}_o = \begin{bmatrix} \rho_o u_o \\ \rho_o u_o^2 \\ \rho_o u_o e_T \end{bmatrix} = \begin{bmatrix} 0 \\ P_o \\ \rho_o u_o \end{bmatrix}, \quad (3.30)$$

consider conservative form of Euler equation using vector notation:

$$\frac{\partial \mathbf{Q}_o}{\partial t} + \frac{\partial \mathbf{F}_o}{\partial x} = 0. \quad (3.31)$$

One way to express the flow vector \mathbf{F}_o is as a function of the conserved values \mathbf{Q}_o . f_3 can be written explicitly as a function of u_1, u_2 , and u_3 :

$$\begin{aligned} f_3 &= (\rho_o e_T + P_o)u = [\rho_o e_T + (\gamma - 1)(\rho_o e_T - \frac{1}{2}\rho_o u_o^2)]u_o, \\ &= (\gamma \rho_o e_T - \frac{\gamma-1}{2} \frac{\rho_o^2 u_o^2}{\rho_o}) \frac{\rho_o u_o}{\rho_o}, \\ &= (\gamma u_3 - \frac{\gamma-1}{2} \frac{u_2^2}{u_1}) \frac{u_2}{u_1} = \gamma \frac{u_3 u_2}{u_1} - \frac{\gamma-1}{2} \frac{u_2^3}{u_1^2}, \end{aligned} \quad (3.32)$$

then by chain rule , $\frac{\partial \mathbf{F}_o}{\partial x} = \frac{\partial \mathbf{F}_o}{\partial \mathbf{Q}_o} \frac{\partial \mathbf{Q}_o}{\partial x}$,
where ,

$$\frac{\partial \mathbf{F}_o}{\partial \mathbf{Q}_o} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \frac{\partial f_1}{\partial u_3} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \frac{\partial f_2}{\partial u_3} \\ \frac{\partial f_3}{\partial u_1} & \frac{\partial f_3}{\partial u_2} & \frac{\partial f_3}{\partial u_3} \end{bmatrix}, \quad (3.33)$$

which is called Jacobian matrix of \mathbf{F}_o .

suppose the quasi-linear equation

$$\frac{\partial \mathbf{Q}_o}{\partial t} + \mathbf{A} \frac{\partial \mathbf{Q}_o}{\partial x} = 0, \quad (3.34)$$

here, $\mathbf{A} = \frac{\partial \mathbf{F}_o}{\partial \mathbf{Q}_o}$ is the Jacobian matrix of \mathbf{F} . Such that:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\gamma-3}{2} u_o^2 & (3-\gamma)u_o & \gamma-1 \\ -\gamma u_o e_T + (\gamma-1)u_o^3 & \gamma e_T - \frac{3}{2}(\gamma-1)u_o^2 & \gamma u_o \end{bmatrix}, \quad (3.35)$$

note that $\mathbf{A} = \mathbf{A}(\mathbf{Q}_o)$.

Define the linear problem:

$$\frac{\partial \mathbf{Q}_o}{\partial t} + \hat{\mathbf{A}} \frac{\partial \mathbf{Q}_o}{\partial x} = 0, \quad (3.36)$$

where $\hat{\mathbf{A}} = \hat{\mathbf{A}}(\mathbf{Q}_{o(L)}, \mathbf{Q}_{o(R)}) = \text{constant}$ such that:

- $\hat{\mathbf{A}}(\mathbf{Q}_{o(L)}, \mathbf{Q}_{o(R)}) \rightarrow \mathbf{A}(\mathbf{Q}_o)$ for $\mathbf{Q}_{o(L)} \rightarrow \mathbf{Q}_o$ and $\mathbf{Q}_{o(R)} \rightarrow \mathbf{Q}_o$.
- $\mathbf{F}_{o(1)}^c(\mathbf{Q}_{o(R)}) - \mathbf{F}_{o(1)}^c(\mathbf{Q}_{o(L)}) = \hat{\mathbf{A}}(\mathbf{Q}_{o(L)} - \mathbf{Q}_{o(R)})$.
- Real eigenvalues and linearly independent eigenvectors are present in $\hat{\mathbf{A}}$.

These prerequisites are met for

$$\hat{\mathbf{A}} = \mathbf{A}(\hat{Q}_\circ), \quad (3.37)$$

where the Roe-averages decide \hat{Q}_\circ ,

$$\hat{\rho}_\circ = \sqrt{\rho_{\circ(L)}\rho_{\circ(R)}}, \hat{u}_\circ = \frac{\sqrt{\rho_{\circ(L)}}\mathbf{u}_{\circ(L)} + \sqrt{\rho_{\circ(R)}}\mathbf{u}_{\circ(R)}}{\sqrt{\rho_{\circ(L)}} + \sqrt{\rho_{\circ(R)}}}, \hat{\mathbf{H}}_\circ = \frac{\sqrt{\rho_{\circ(L)}}\mathbf{H} + \sqrt{\rho_{\circ(R)}}\mathbf{H}_{\circ(R)}}{\sqrt{\rho_{\circ(L)}} + \sqrt{\rho_{\circ(R)}}}. \quad (3.38)$$

The Roe-averaged sound speed is obtained from $\hat{c}^2 = (\gamma - 1)(\hat{H}^* - \frac{1}{2}|\hat{u}_\circ|^2)$.

If there is just one shock or one contact discontinuity connecting $Q_{\circ(L)}$ and $Q_{\circ(R)}$, then Roe's approximation riemann solver is accurate.

we can diagonalize $\hat{\mathbf{A}}$ like \mathbf{A} and express equation as 3 linear equations in 2D

$$\frac{\partial \mathbf{W}_i}{\partial t} + \hat{\lambda}_i \frac{\partial \mathbf{W}_i}{\partial x} = 0, i = 1, 2, 3. \quad (3.39)$$

where $\mathbf{W} = \hat{\mathbf{R}}^{-1}\mathbf{Q}_\circ = \begin{pmatrix} -\frac{\hat{\rho}_\circ u_\circ}{2\hat{c}} + \frac{1P_\circ}{2\hat{c}^2} \\ \rho_\circ - \frac{1}{\hat{c}^2}P_\circ \\ \frac{\hat{\rho}_\circ u_\circ}{2\hat{c}} + \frac{1P_\circ}{2\hat{c}^2} \end{pmatrix}$ are the characteristics variables.

$\hat{\lambda}_{(1)} = \hat{u}_\circ - \hat{c}$, $\hat{\lambda}_{(2)} = \hat{u}_\circ$, $\hat{\lambda}_{(3)} = \hat{u}_\circ + \hat{c}$ are the constant wave of speed.

let δW_i denote the strength of the i-th wave ,i.e the i-th component of $\hat{R}^{-1}(\mathbf{Q}_{\circ(\mathbf{R})} - \mathbf{Q}_{\circ(\mathbf{L})}) = \hat{T}^{-1}(\mathbf{V}_{\circ(\mathbf{R})} - \mathbf{V}_{\circ(\mathbf{L})})$,

then exact solution of (3.36) at the interface can be expressed can be as:

$$W_i(0, t) = \begin{cases} W_{iL} = W_{iR} - \delta W_i, & \text{if } \hat{\lambda}_i > 0 \\ W_{iR} = W_{iL} + \delta W_i, & \text{if } \hat{\lambda}_i < 0, \end{cases} \quad (3.40)$$

the flux difference can be written as:

$$\mathbf{F}_{\circ(1)}^c(\mathbf{Q}_{\circ(\mathbf{R})}) - \mathbf{F}_{\circ(1)}^c(\mathbf{Q}_{\circ(\mathbf{L})}) = \hat{\mathbf{R}}\hat{\mathbf{A}}^*\hat{\mathbf{R}}^{-1}(\mathbf{Q}_{\circ(\mathbf{R})} - \mathbf{Q}_{\circ(\mathbf{L})}) = \sum_{i=1}^3 \delta W_i \hat{\lambda}_i \hat{\mathbf{r}}_i, \quad (3.41)$$

where $\hat{\mathbf{r}}_i$ is the ith right eigenvector of $\hat{\mathbf{A}}$, i.e. the ith column of $\hat{\mathbf{R}} = \mathbf{R}(\hat{Q}_\circ)$ The right eigenvector matrix of $\mathbf{R}(\mathbf{Q}_\circ)$ reads:

$$\mathbf{R} = \begin{bmatrix} 1 & 1 & 1 \\ u_\circ - c & u_\circ & u_\circ + c \\ H_\circ - u_\circ c & \frac{u_\circ^2}{2} & H_\circ + u_\circ c \end{bmatrix}, \quad (3.42)$$

So, Roe describes the flow at the contact as follows:

$$\mathbf{F}_o(0, t) = \mathbf{F}_o(\mathbf{Q}_{o(L)}) + \hat{\mathbf{A}}^-(\mathbf{Q}_{o(R)} - \mathbf{Q}_{o(L)}), \quad (3.43)$$

$$\mathbf{F}_o(0, t) = \mathbf{F}_o(\mathbf{Q}_{o(R)}) - \hat{\mathbf{A}}^+(\mathbf{Q}_{o(R)} - \mathbf{Q}_{o(L)}), \quad (3.44)$$

$$\mathbf{F}_o(0, t) = \frac{1}{2}[\mathbf{F}_o(\mathbf{Q}_{o(L)}) + \mathbf{F}_o(\mathbf{Q}_{o(R)}) - |\hat{\mathbf{A}}|(\mathbf{Q}_{o(R)} - \mathbf{Q}_{o(L)})], \quad (3.45)$$

where (3.45) is obtained by averaging (3.43) and (3.44). Here, $\hat{\mathbf{A}}^\mp = \hat{\mathbf{R}}\hat{\mathbf{A}}^{*\mp}\hat{\mathbf{R}}^{-1}$ and $|\hat{\mathbf{A}}| = \hat{\mathbf{R}}|\hat{\mathbf{A}}^*|\hat{\mathbf{R}}^{-1}$, where $\hat{\mathbf{A}}^* = \text{diag}(\hat{\lambda}_j^\mp)$ with $\hat{\lambda}_j^\mp = \frac{1}{2}(\hat{\lambda}_j \mp |\hat{\lambda}_j|)$ and $|\hat{\mathbf{A}}^*| = \text{diag}(|\hat{\lambda}_j|)$.

If the expansion fan spans the time axis, that is, if its sonic condition is either $u^* + c = 0$ or $u^* - c = 0$, then approximating the expansion fan by an expansion shock violates the entropy requirement and results in an incorrect flux at the interface. For example, using Roe's approximation riemann solver, an expansion shock at the interface will incorrectly stay unaltered rather than spreading like an expansion fan.

By adding a sonic state to a sonic expansion fan, we may impose the entropy requirement, i.e. if either $u_o - c = 0$ or $u_o + c = 0$; alternatively, if $|u_o - c|$ or $|u_o + c|$ become tiny due to numerical dissipation added to the acoustic waves.

3.8 Explanation for Simplified Ghost Point Method on Embedded Boundary in two Dimensions

For the simple ghost point treatment, the reflecting point is the point F_1 of the fluid next to the embedding border. The embedded boundary is therefore thought to be midway between ghost and fluid points. The concept is shown in figure on a 9x9 grid. The cylinder's highlighted ghost cells, generated by the algorithm, are displayed on a finer grid in figure.

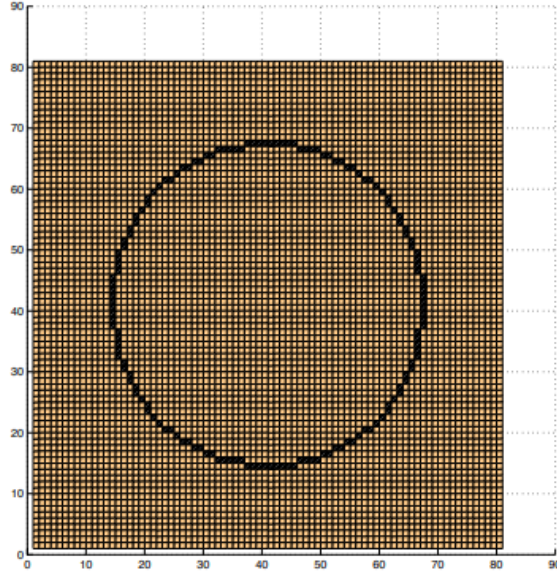


Figure 3.3: 81x81 grid with ghost cells marked with flags. Ghost points' whereabouts are indicated by back points.

When there are ghost points , the primitive variables are configured as

$$\rho_{oG_1} = \rho_{oF_1}, \quad u_{oG_1} = -u_{oF_1}, \quad v_{oG_1} = -v_{oF_1}, \quad P_{oG_1} = P_{oF_1}. \quad (3.46)$$

Considering that the embedding boundary is situated halfway between mirrored point F_1 and ghost point G_1 . The boundary criteria $u_o = v_o = 0$ and $\frac{\partial \rho_o}{\partial n} = 0$ are presumed to exist at the embedded boundary. It's important to pick the mirror points F_1 carefully. The values at the fluid points to the west of the ghost points $\mathbf{Q}_{o(i_1G_1, j_1G_1)} = \mathbf{Q}_{o(i_1G_1-1, j_1G_1)}$ are used to find the values of the ghost spots on the western edge of the domain's boundaries. similarly, we can get ghost point for southern side, east side and northern side $\mathbf{Q}_{o(i_1G_1, j_1G_1)} = \mathbf{Q}_{o(i_1G_1, j_1G_1-1)}$, $\mathbf{Q}_{o(i_1G_1, j_1G_1)} = \mathbf{Q}_{o(i_1G_1+1, j_1G_1)}$ and $\mathbf{Q}_{o(i_1G_1, j_1G_1)} = \mathbf{Q}_{o(i_1G_1, j_1G_1+1)}$, respectively.

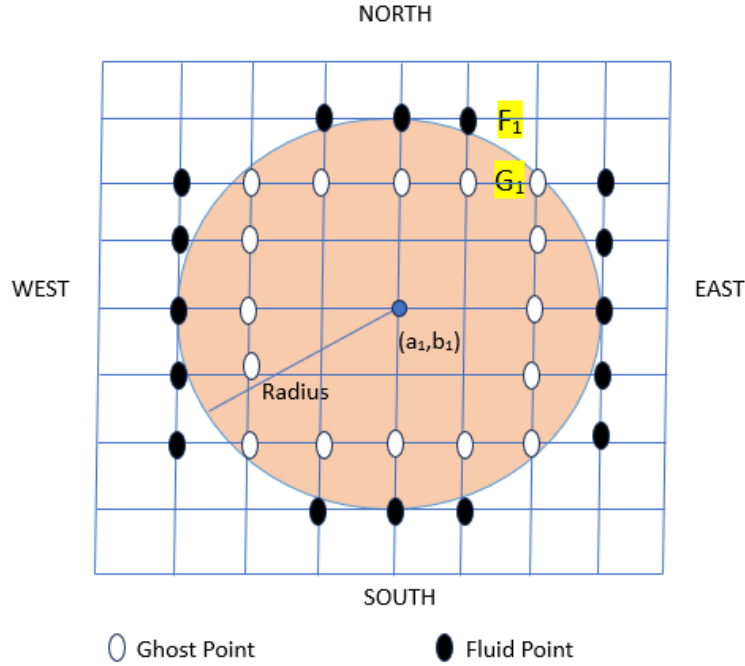


Figure 3.4: The configuration of the ghost points to create a solid cylinder and the selection of the fluid points as mirror points as an example.

The circle and the grid lines' junction at the center point (a_1, b_1) , or four locations on the grid, are precisely on the circle. These are fluid points that need to be addressed explicitly.

On the grid, the 45-degree junction lines represent four ghost locations. The closest fluid point in either the x or y directions can be used to compute a ghost point on this line. The closest fluid point in the x-direction is used in this function to compute these ghost spots. Selecting the diagonal, or closest fluid point on the intersecting line, is an additional strategy.

3.9 Treatment of Ghost Points with Second Order Accuracy

It takes a new set of ghost points, If the MUSCL's minmod limitation is used to obtain second order precision. Ghost point values need to be set at $(x_{1(i_1 \pm 2)}, y_{1(j)})$ and $(x_{1(j_1)}, y_{1(j_1 \pm 2)})$ if they are inside the solid, as the minmod limiter also utilizes the values $Q_{o(i_1 \pm 2, j_1)}$ and $Q_{o(i_1, j_1 \pm 2)}$. Primitive variables are set at the ghost spots to

$$\rho_o(G2) = \rho_o(F2), \quad u_o(G2) = -u_o(F2), \quad v_o(G2) = -v_o(F2), \quad P_o(G2) = P_o(F2), \quad (3.47)$$

accomplishes this, as seen in figure.

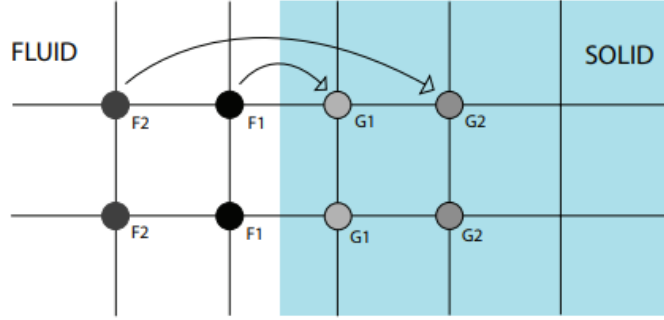


Figure 3.5: A demonstration of the MUSCL ghost point computations.

3.10 Calculating the Shear Stress on a Body

Due to its location between two points, it might be challenging to precisely identify the wall of the selected embedded body when using the basic ghost point treatment. This problem was solved by averaging the skin friction at locations close to the wall. There have been several uses near the implanted body's wall.

The averaged wall shear stress $(\tau_{w'})_{i_1, j_1}$ may be obtained by averaging the following values, where $(\tau_{w'})_{i_1, j_1}$ is the fluid point's shear stress at its closest to the wall. See in figure and equation.

$$(\tau_{w'})_{i_1, j_1} = \frac{2(\tau_{p'})_{i_1, j_1} + (\tau_{p'})_{i_1+1, j_1} + (\tau_{p'})_{i_1, j_1-1} + (\tau_{p'})_{i_1, j_1+1}}{6}, \quad (3.48)$$

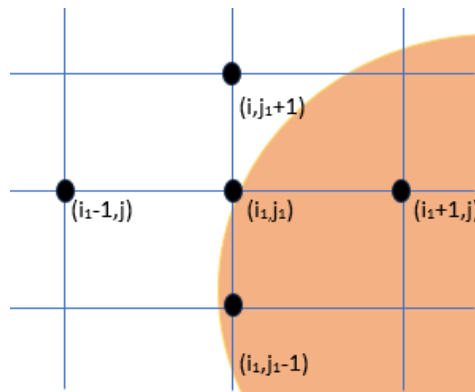


Figure 3.6: The locations for which the shear stress on the wall was averaged.

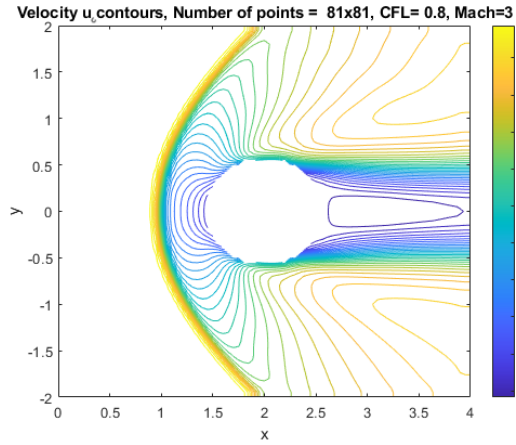
Chapter 4

Test Cases for Navier-Stokes Equation

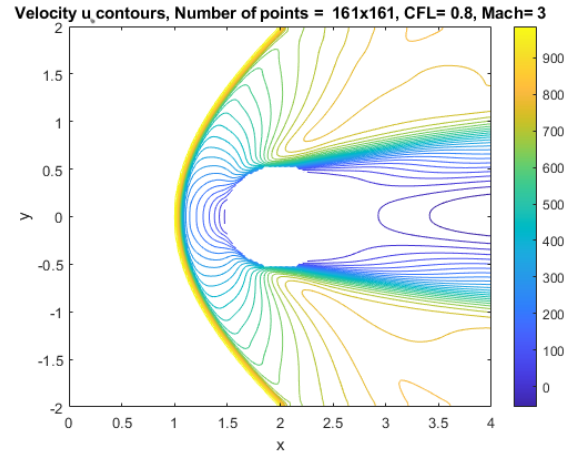
The aim of this chapter is to compare the simplified ghost point method with the well-established cartesian grid methodology for viscous movement under flow under supersonic speed settings.

4.1 Supersonic Flow over Circular Cylinder Employing Compressible Navier-Stokes Equation

In context of the compressible Navier-Stokes equation, flow at supersonic speed across a circular cylinder is examined. It has been thought about the movement across a circular cylinder at supersonic speed having $\text{Mach} = 3$ and $\text{Re} = 500$. The computational domain $[-2, 2] \times [-2, 2]$ has been discretized using grids consisting of 71×71 , 141×141 , 281×281 and 561×561 cells. The cylinder has a diameter of $D = 1$, and its center is at $(0,0)$. It is believed that the cylinder wall is adiabatic. For each test instance, the boundary conditions are as follows: supersonic outflow for all other boundaries, and supersonic inflow for boundary. $x = -2$. The MUSCL system achieves second order precision, with the exception of extrema. Here minmod is used, hence it's accuracy is second. Here we have done comparison of velocity u , v and density contours using different grids. Figures depicts the curves of the Mach number. We note a bow shock that is sharply resolved.

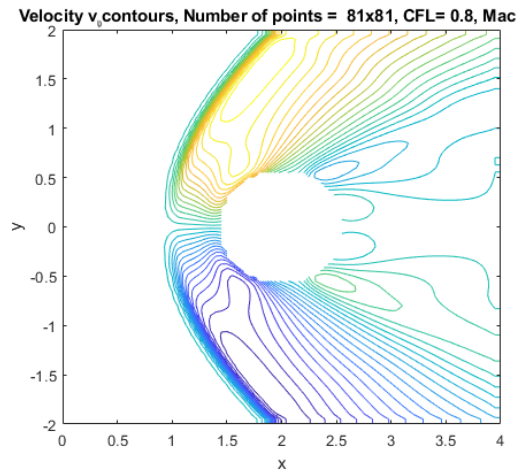


(a) Velocity u_o contours.

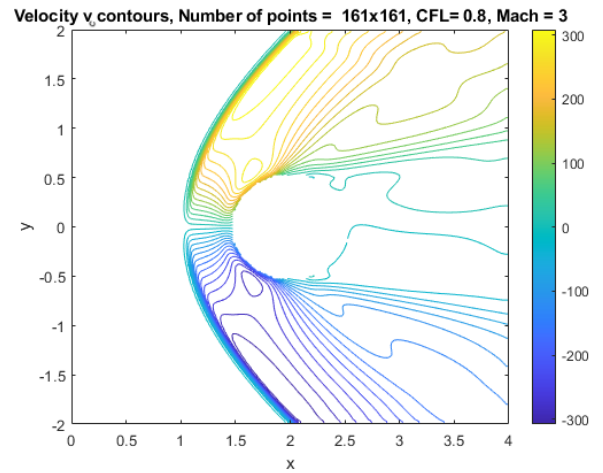


(b) Velocity u_o contours.

Figure 4.1: Velocity u_o contours having 5000 iterations for 81×81 and 161×161 grid points for 1 cylinder.

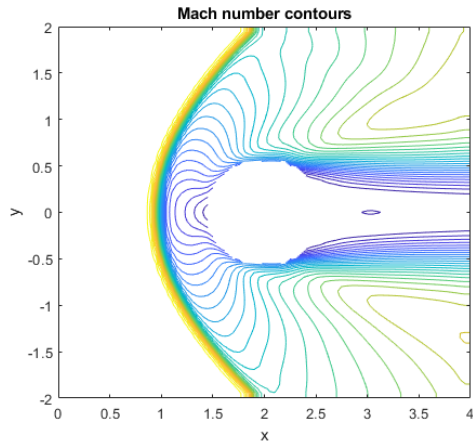


(a) Velocity v_o contours.

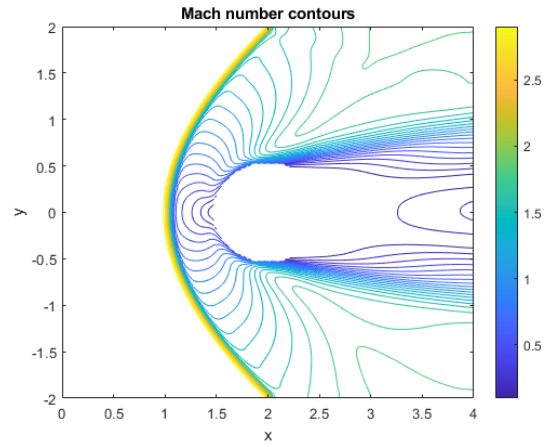


(b) Velocity v_o contours.

Figure 4.2: Velocity v_o contours having 5000 iterations for 81×81 and 161×161 grid points for 1 cylinder.

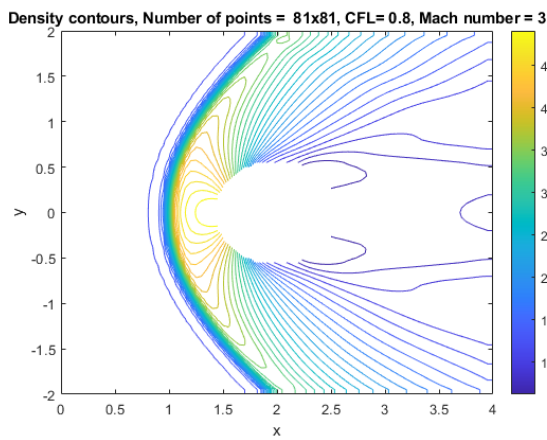


(a) Mach number contours.

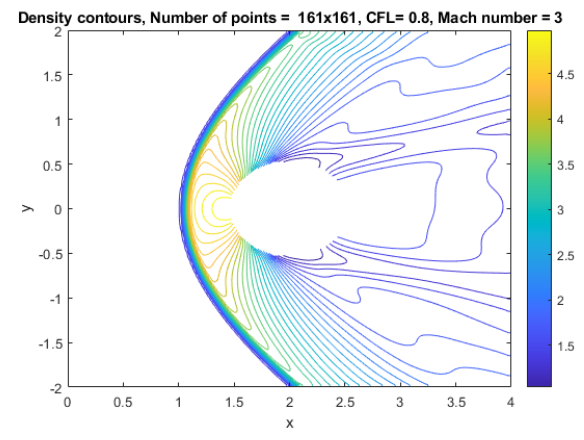


(b) Mach number contours.

Figure 4.3: Mach number contours having 5000 iterations for 81×81 and 161×161 grid points for 1 cylinder.

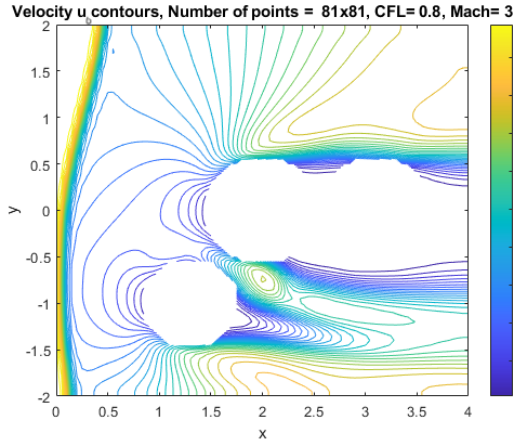


(a) Density contours.

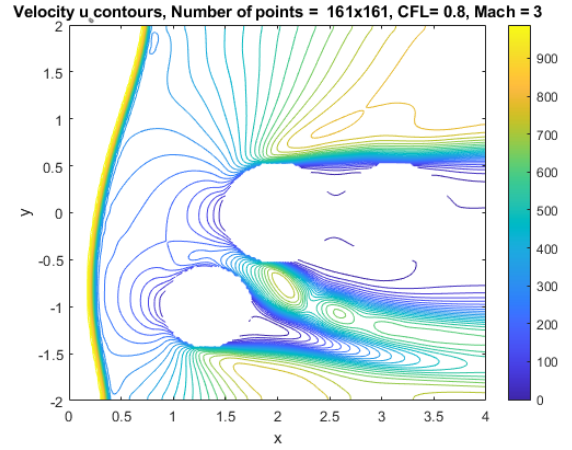


(b) Density contours.

Figure 4.4: Density contours having 5000 iterations for 81×81 and 161×161 grid points for 1 cylinder.

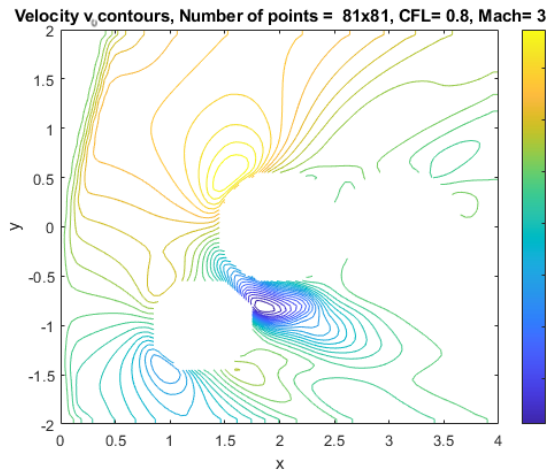


(a) Velocity u_o contours.

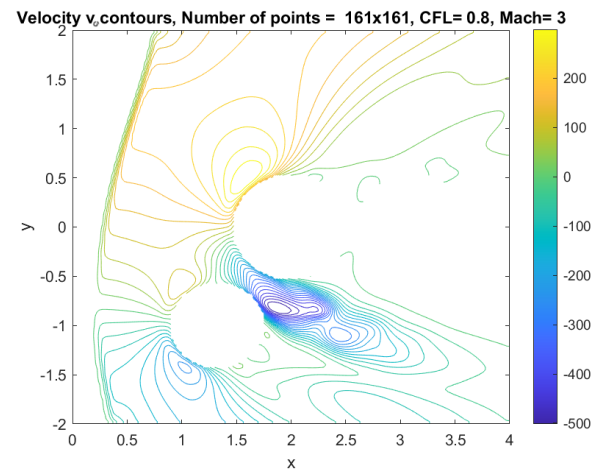


(b) Velocity u_o contours.

Figure 4.5: Velocity u_o contours having 5000 iterations for 81×81 and 161×161 grid points for 3 cylinder.

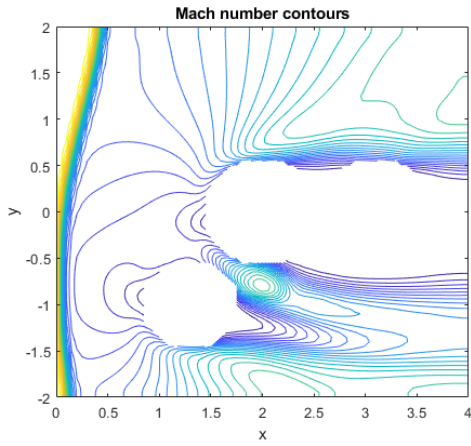


(a) Velocity v_o contours.

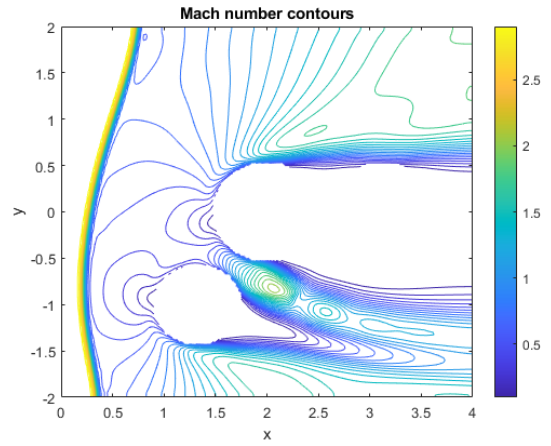


(b) Velocity v_o contours.

Figure 4.6: Velocity v_o contours having 5000 iterations for 81×81 and 161×161 grid points for 3 cylinder.

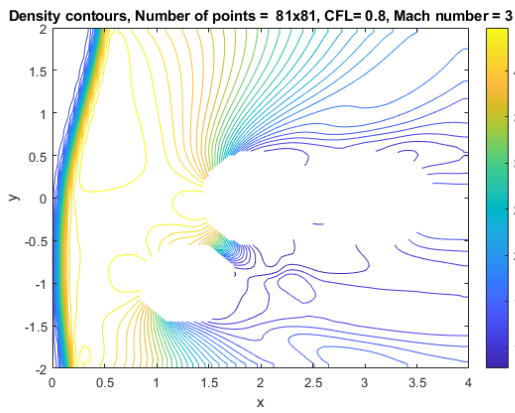


(a) Mach number contours.

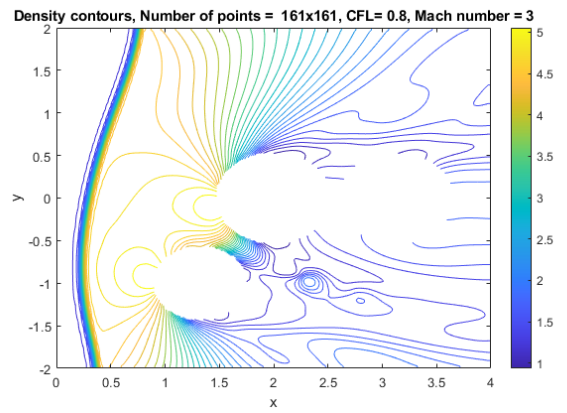


(b) Mach number contours.

Figure 4.7: Mach number contours having 5000 iterations for 81×81 and 161×161 grid points for 3 cylinder.



(a) Density contours.



(b) Density contours.

Figure 4.8: Density contours having 5000 iterations for 81×81 and 161×161 grid points for 3 cylinder.

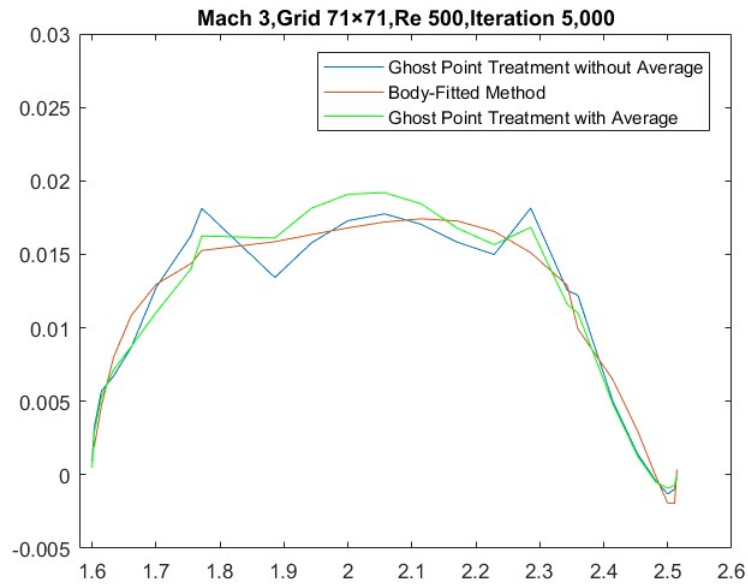


Figure 4.9: A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average.

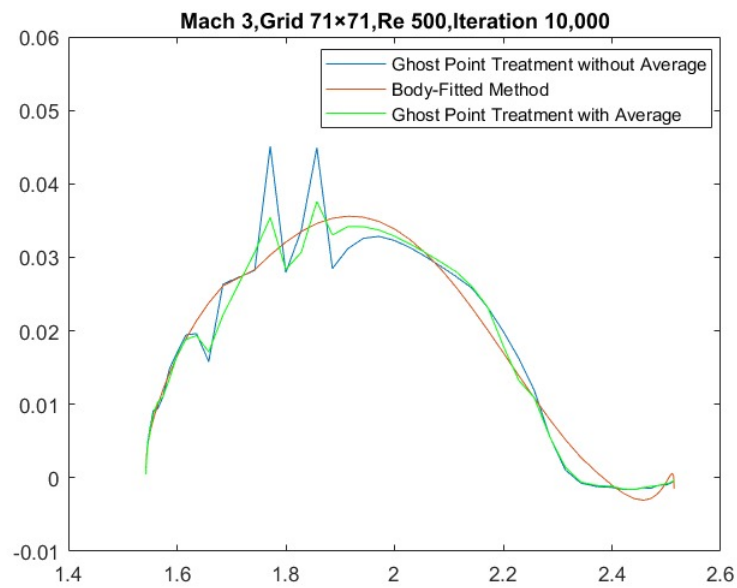


Figure 4.10: A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average.

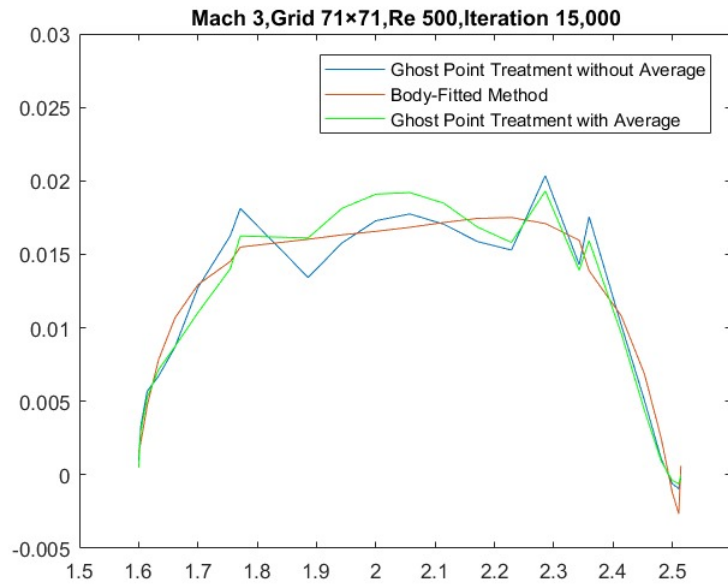


Figure 4.11: A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average.

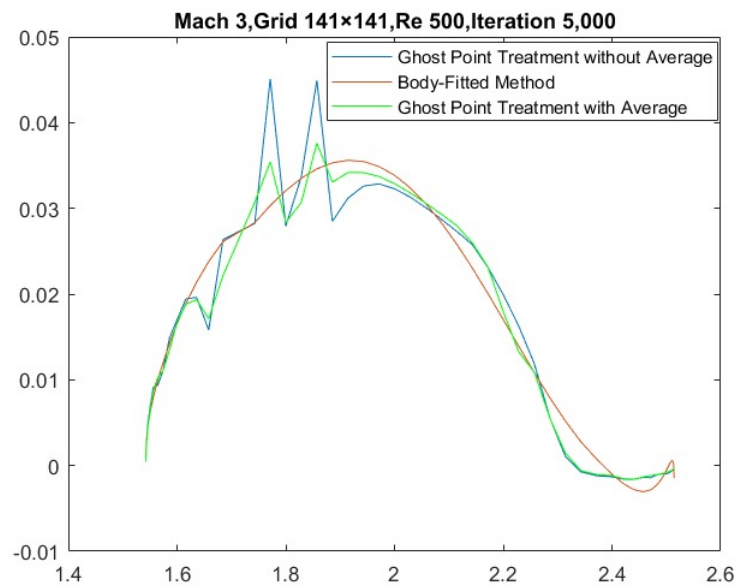


Figure 4.12: A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average.

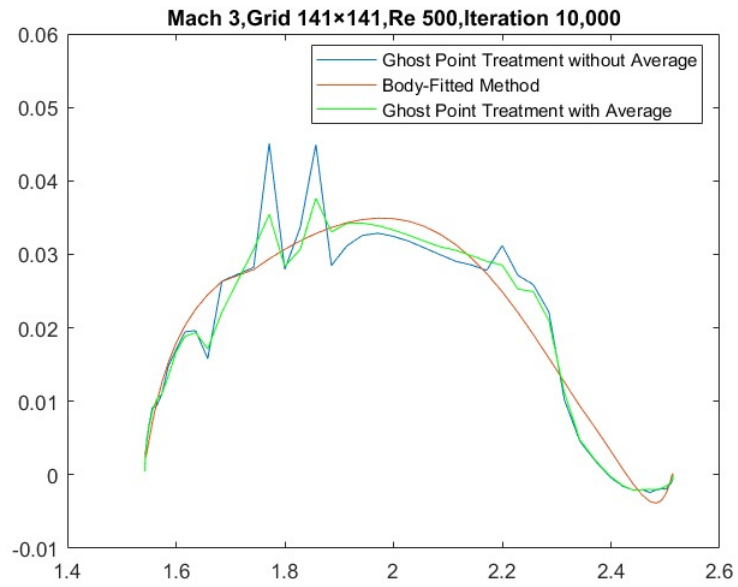


Figure 4.13: A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average.

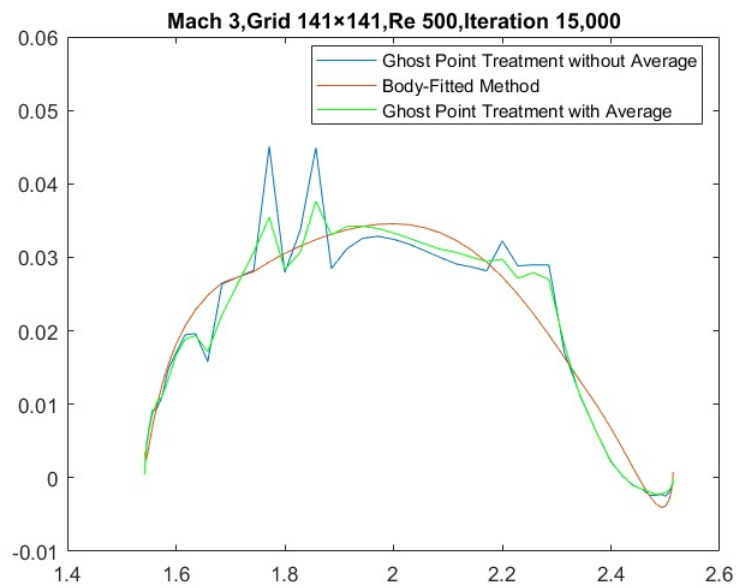


Figure 4.14: A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average.

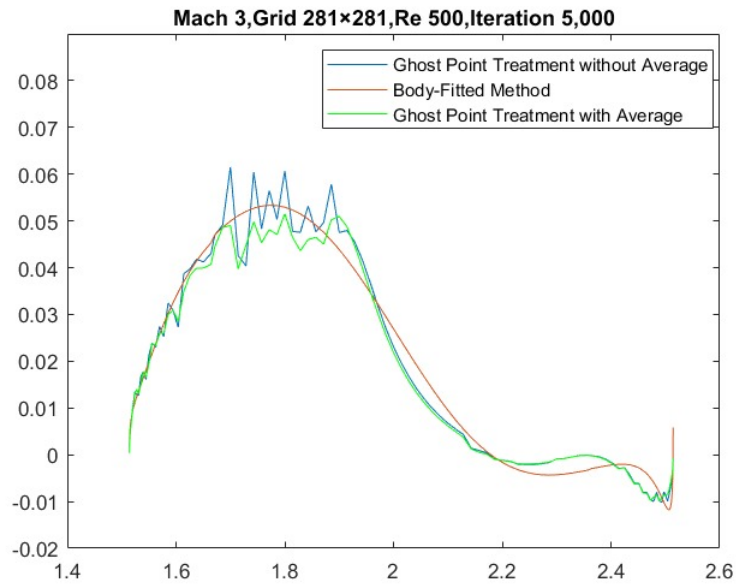


Figure 4.15: A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average.

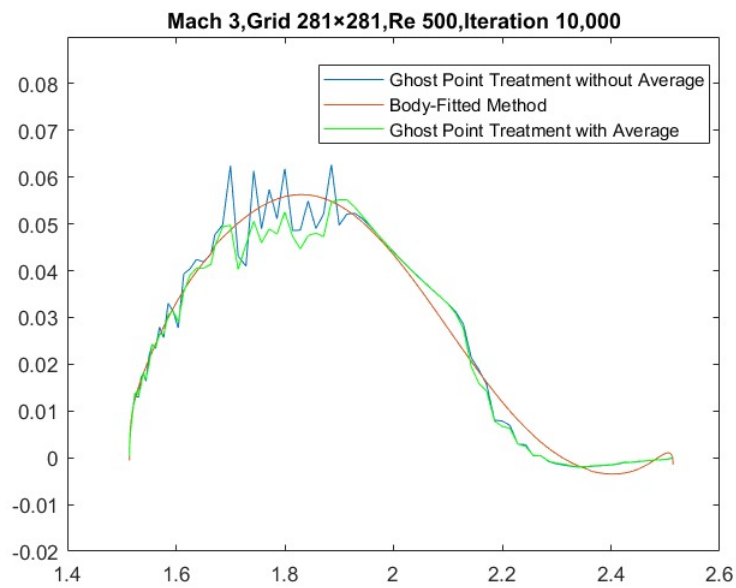


Figure 4.16: A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average.

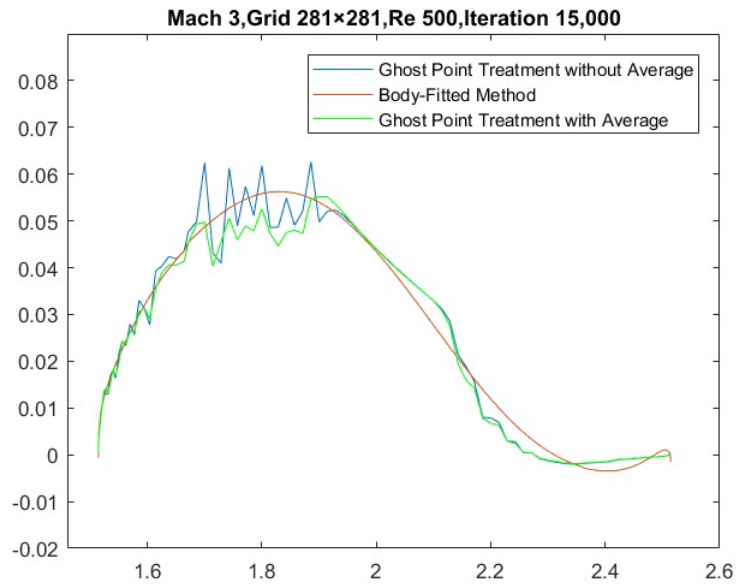


Figure 4.17: A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average.

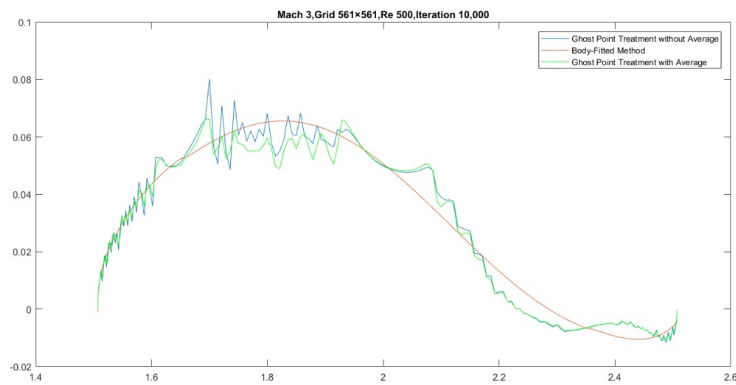


Figure 4.18: A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average.

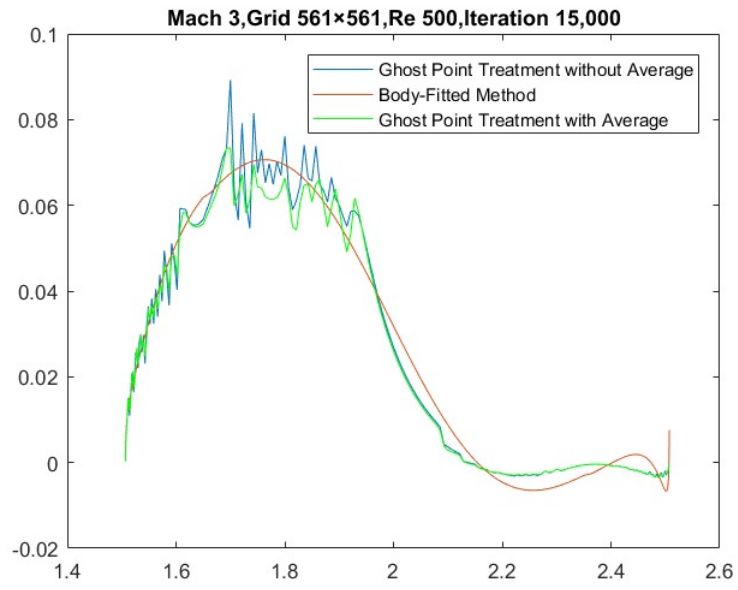


Figure 4.19: A comparison for skin friction coefficient of the body-fitted method, simplified ghost point treatment with average and without average.

Chapter 5

Conclusions and Outlook

Our consideration is to flow at supersonic speed across a circular cylinder in compressible Navier-Stokes equations. We have employed several grid locations for supersonic flow across a circular cylinder. For spatial discretization, we employed the MUSCL (Monotone Upstream-centered Schemes for Conservation Laws) scheme and the local Lax-Friedrichs approach. We used Runge-Kutta (RK3) and Total Variation Diminishing (TVD) techniques for temporal discretization.

A technique for calculating viscous compressible flows around intricate geometries is the embedded boundary approach that we have demonstrated. For two-dimensional issues, the approach is simple to apply and can yield first- or second-order precision. Using viscous flow with modest Reynolds numbers, varying The quantity of grid points, and increasing the number of iterations, the approach has been validated against thoroughly recorded steady and unsettled test situations in supersonic flow regimes. Compared to methods that utilize gridles points at borders or local grid refining , we discover further grid points are needed to obtain excellent resolution at nearness to the body and in regions with significant flow-gradients. The accuracy, resilience, and adaptability of the suggested approach have all been shown by a number of numerical tests. According to the numerical data, the simplified approach outperforms the ghost point method in terms of performance. While not as exact as other more advanced approaches, the results obtained by the simplified method are comparable.

It would be interesting to improve the approach for computing three-dimensional problems further in future studies. Enhancing the precision level is possible, and it makes sense to consider implementing local grid refinement. We can increase the Reynolds number by finding solutions for smaller sizes. if we have improved resolution in relevant regions. Further research and testing of the skin friction coefficient calculation is increasingly urgent since it has shown to be extremely sensitive to changes in the computation techniques. Finally, there is a chance to refine and test the simplified ghost point approach for other scenarios.

Bibliography

- [1] M. Farooq. "Cartesian Grid Method for Compressible Flow Simulation". Doctoral thesis at NTNU: 85 (2012)
- [2] S. Gottlieb, C.-W. Shu and E. Tadmor. "Strong Stability-Preserving High-Order
- [3] B. Sjögren, M. Kupiainen. "A Cartesian Embedded Boundary Method for the Compressible Navier-Stokes Equations". *J. Sci. Comput.* 41, pp. 94-117 (2009)
- [4] T. R. Bewley. "Numerical Renaissance". Not published yet, but available online at <http://renaissance.ucsd.edu/> (2010)
- [5] H.-O. Kreiss and N.A. Petersson. "A Second Order Accurate Embedded Boundary Method for the Wave Equation with Dirichlet Data". *SIAM J. Sci. Comput.* 27, pp. 1141-1167 (2006)
- [6] S. Gottlieb, C.-W. Shu. "Total variation Diminishing Runge-Kutta Schemes". *Mathematics of Computation* Vol. 67, No. 221, pp. 73-85 (1998)
- [7] B. Müller. "Navier-Stokes solution for laminar transonic flow over a NACA0012 airfoil". FFA Report 140 The Aeronautical Research Institute of Sweden, Stockholm (1986)
- [8] B. Sjögren and N.A. Petersson. "A Cartesian Embedded Boundary Method for Hyperbolic Conservation Laws." *Commun. Comput. Phys.*, pp. 1199-1219 (2007)
- [9] M. Farooq and B. Müller, Accuracy assessment of the Cartesian grid method for compressible inviscid flows using a simplified ghost point treatment. *Journal of Structural Mechanics*, Vol. 44, No. 3, pp.279-291 (2011)
- [10] B. van Leer. "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method". *Journal of Computational Physics* 32, pp. 101-136 (1979)
- [11] Ames Research Staff. "Equations, Tables and Charts for Compressible Flow". NACA Report 1135, (1953)

- [12] H. Luo, J.D. Baum and R. Löhner. "A hybrid Cartesian grid and gridless method for compressible flows". *Journal of Computational Physics* 214, pp. 618-632 (2006)
- [13] J.J. Quirk. "An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies". *Computers and Fluids* 23, pp. 125-142 (1994)
- [14] P. De Palma, M.D. Tullio, G. Pascazio and M. Napolitano. "An immersed-boundary method for compressible viscous flows". *Computers and Fluids* 35, pp. 693-702 (2006)
- [15] J. Zierep. "Theoretische Gasdynamik" G. Braun, Karlsruhe (1976)
- [16] J.D Anderson JR. "Fundamentals of Aerodynamics". Tata McGraw-Hill Publishing Company Limited, New Delhi, Fourth Edition (2007)
- [17] M.O. Bristeau, R. Glowinski, J. Periaux and H. Viviand. "Numerical Simulation of Compressible 48 Navier-Stokes Flows". *Notes on Numerical Fluid Mechanics* Vol 18, Vieweg, Braunschweig (1987)
- [18] R. Crockett, P. Colella and T. Marthaler. "A Cartesian grid projection method for the incompressible Euler equations in complex geometries". *SIAM J. Sci. Comput.* 18, pp 1289-1309 (1997)
- [19] R. M. Beam and R.F. Warming. "An Implicit Factored Scheme for the Compressible Navier Stokes Equations". *AIAA Journal* vol.16, no.4, pp. 393-402 (1978)
- [20] Randall J. Leveque, "Finite-volume methods for hyperbolic problems", University of Washington (2004)
- [21] A. A. Skøien. "Cartesian Grid Method for Viscous Compressible Flow". Project thesis at NTNU (2011)