THESIS

COMPUTATIONAL MODELING OF PLASMA-ASSISTED SHOCK WAVE CONTROL USING THE CARTESIAN CUT CELL METHOD

Submitted by

Elijah D. House

Department of Mechanical Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Fall 2024

Master's Committee:

Advisor: Ciprian Dumitrache

Bret Windom
Wolfgang Bangerth

ABSTRACT

COMPUTATIONAL MODELING OF PLASMA-ASSISTED SHOCK WAVE CONTROL USING THE CARTESIAN CUT CELL METHOD

To control shock waves using plasma discharges, we have developed a numerical model focusing on improving the accuracy and efficiency of simulating supersonic channel flow. Shock waves are essential in high-speed air-breathing propulsion devices such as ramjets and scramjets. The lack of turbomachinery means that shock waves compress air prior to combustion. The shocks decelerate the high-speed flow, increasing static temperature and pressure, which is necessary for efficient combustion. However, the advantage of simplicity (no moving parts) to achieve compression is counteracted by increased wave drag, total pressure losses, and flow separation inside the engine. In this context, the generation of shock wave trains (a sequence of reflected oblique and normal shocks propagating through the engine) must be appropriately managed and optimized to reduce drag and enhance thrust.

To tackle these challenges, we use the APDL-CFD code to model a Ma=2.5 supersonic flow over a 10-degree triangular wedge inside a straight channel. The wedge generates a shock wave train that is typically encountered inside the isolator of a scramjet engine. These conditions are indicative of conditions that are currently being tested in supersonic wind tunnels. The code solves the compressible Navier-Stokes equations, incorporating advective and diffusive fluxes. The advective fluxes account for mass, momentum, and energy transport, while the diffusive fluxes capture viscous stresses and thermal conduction. This formulation includes viscous dissipation and heat diffusion, ensuring accurate modeling of compressible flow behavior.

Furthermore, we enhance the APDL-CFD code with the Cartesian cut cell method, which allows the representation of complex geometries on a Cartesian mesh. This research represents geometries found in wind tunnel models and internal vehicle designs. Using the cut cell method,

the model can capture flow caused by geometries that do not conform to a Cartesian mesh, like the wedge that generates the oblique shock waves. This improves accuracy and significantly reduces computational costs, allowing for lower grid resolutions on a Cartesian mesh. The cut cell method is implemented to research the use of plasma actuators as an active control mechanism. The model investigates how varying key parameters, such as the location and temperature of the plasma, affect shock wave dynamics and the associated separation bubbles. Results show that the plasma kernel alters the flow and provides an effective way to shift the position and reduce the intensity of the shock waves inside the channel. The numerical simulations aim to optimize this control, showing that shocks can be dynamically managed with the proper plasma parameters to enhance flow stability and performance. The results demonstrate significant improvements in controlling shock waves and flow separation when plasma actuators are employed, showing potential for their use in high-speed propulsion systems such as scramjets. Moreover, incorporating the cut cell method has optimized the APDL-CFD code, making it more efficient and better suited for running rapid test simulations. The results can inform future experiments, such as those planned for the Colorado State University (CSU) wind tunnel. Overall, the research offers valuable insights into active flow control in supersonic and hypersonic vehicles for improving vehicle performance, efficiency, and reliability.

ACKNOWLEDGEMENTS

The path to this point has been challenging and not always straightforward, but I have had a fantastic support group behind me to help me navigate these challenges. I would like to first thank my advisor, Dr. Dumitrache, for his support on this project. This project was something new to both of us, so being able to work together to develop this model and code was a great experience full of learning and growth. He made me feel exceptionally welcome in the laboratory, joining the group halfway through my graduate studies. As a computationalist, seeing and learning about this group's vast variety of experiments was inspiring. It helped me gain fundamental knowledge I used to further understand the computational work I was doing.

I would also like to thank my colleagues, past and present, including Andy, Chris, Erik, Mozhdeh, Spencer, and others, for their help in navigating the challenges of graduate school. Their work is equally fascinating and inspiring. I will forever be thankful for the support and friendship they have provided me.

I am incredibly grateful for my family, including my mother, Lynn, and my brothers, Noah, Asher, and Jaden. The journey of graduate school has been difficult, to say the least. I lost my father during the first year of my graduate studies, and it has been a long road to get to this point. I know he would be proud of me for getting to this point, and I am grateful for the support of my family during this time. I would not have been able to finish this research without them.

Most importantly, none of this would have happened without the endless support from my fiancé, Marta. She motivated me to pursue this degree and for that, I am forever grateful. Her unwavering support and encouragement throughout this journey have helped push me to do better. Her love and patience have been a constant source of strength and inspiration, and I am grateful for her presence in my life.

DEDICATION

This thesis is dedicated to my father,

Bret House

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

Air-breathing propulsion refers to using the surrounding atmosphere as the oxidizer for the fuel. The quintessential example of an air-breathing hypersonic vehicle with supersonic internal flow is a scramjet (supersonic combustion ramjet) engine. These specialty engines get their name from their combustion process, where supersonic air oxidizes the fuel in the combustor. Scramjets have been shown to provide distinct advantages due to their increased range, higher speeds, faster response time, and improved maneuverability. However, considerably more research is still required to achieve the desired reliability for travel and launch applications, particularly when carrying humans onboard. Regardless of the application, the current operating envelope limits scramjet operations. Current high-speed air-breathing vehicles are typically launched at high speeds, using a rocket booster or other aircraft, before the vehicle can ignite and operate. These limitations are driving the development of new control strategies to improve the performance and reliability of hypersonic propulsion systems so they can be used in a broader range of applications, including commercial space travel and space exploration launch missions.

Scramjets differ from conventional jet engines because they have few moving parts, such as compressors or turbines. Instead, the engine relies on the geometry of the inlet to compress the air, which is then mixed with fuel and ignited to produce thrust. The air is compressed in the inlet through a series of oblique shock waves central to the scramjet operation. The oblique shock wave trains present challenges in both design and simulation. Accurately predicting the interaction of these shock waves with the vehicle's boundary layers is vital for determining performance, stability, and overall engine operation. Despite these challenges, the potential benefits of scramjets, such as increased efficiency and speed, make them an attractive option for in-atmosphere travel.

Figure 1.1 illustrates the propulsion efficiency of various aerospace propulsion systems as a function of flight Mach number. The plot highlights that there is no "one-size-fits-all" propulsion device for all flight regimes. At low Mach numbers (Ma < 2), turbojets are the most efficient propulsion systems. They leverage atmospheric air and turbomachinery to generate thermal energy through combustion, converted into thrust. However, as the Mach number increases, turbomachinery becomes increasingly inefficient due to limitations in mechanical components and increased drag losses. Beyond this regime, ramjets and scramjets become more effective since they have no moving parts, relying on supersonic or hypersonic airflow to compress the incoming air for combustion. Scramjets can achieve greater propulsion efficiency at even higher speeds than rockets, particularly in the Mach 5-10 range, by efficiently utilizing atmospheric oxygen without needing onboard oxidizers.

In summary, no propulsion technology suits the entire flight envelope. Therefore, the future of aerospace propulsion lies in combined-cycle engines, which can seamlessly transition between different modes to maintain efficiency across various flight regimes—from subsonic to hypersonic and even into space. Multi-mode propulsion concepts such as Turbine-Based Combined Cycle (TBCC) and Rocket-Based Combined Cycle (RBCC) engines reflect this need for flexibility. TBCC systems use turbofans or turbojets at low speeds and switch to ramjet or scramjet modes at higher speeds. At the same time, RBCC engines integrate air-breathing and rocket propulsion to operate across atmospheric and space environments. These tradeoffs present unique challenges, particularly in understanding and managing the complex interactions between high-speed flows and shock waves across different flight regimes. In this context, Computational Fluid Dynamics (CFD) plays a crucial role by providing a powerful tool for simulating complex flow fields, capturing intricate interactions like shock-boundary layer interactions, and enabling detailed analysis that would be difficult or impossible to achieve experimentally.

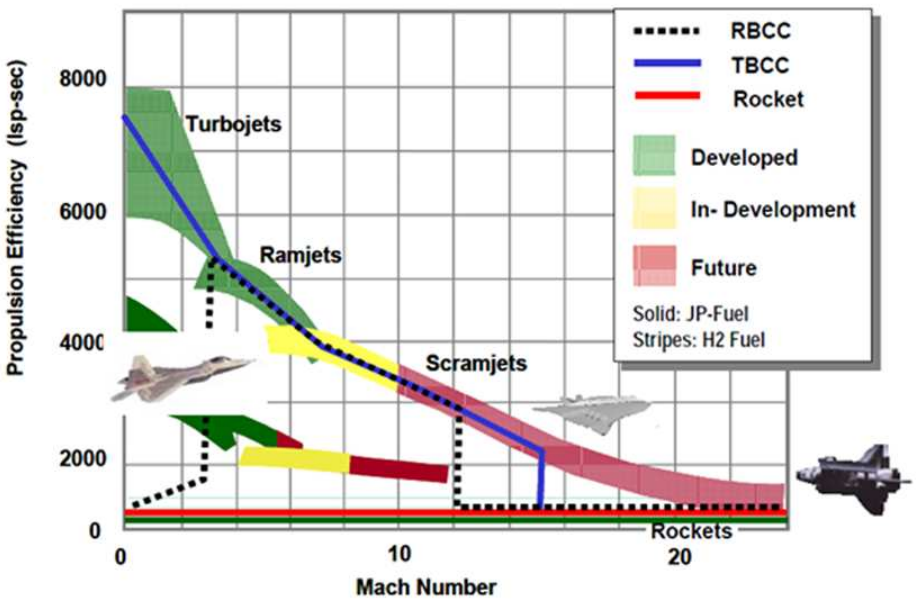

Figure 1.1: Plot of the Mach number versus the specific impulse for aerospace development [2].

Experimental validation of supersonic flow interactions presents noteworthy challenges due to extreme flow conditions and the limited availability of in-flight testing data. Ground-based facilities, such as wind tunnels, remain costly and often struggle to replicate real-flight conditions accurately. In this context, advanced computational techniques are essential to bridge these gaps, providing valuable insights and complementing experimental efforts [12]. Furthermore, this scarcity of experimental data highlights the need for advanced computational techniques that improve accuracy and reduce reliance on large-scale wind tunnel tests [13]. Moreover, CFD can inform experimental conditions, reducing the number of required tests [14]. Modeling supersonic flow within hypersonic vehicles demands significant computational resources and specialized algorithms. Nevertheless, depending on the application, assumptions can be made to simplify the model and reduce the number of unknowns. Chapter 3 further details the assumptions used in this thesis.

Numerical modeling of supersonic flows within a hypersonic vehicle poses significant challenges due to the complex physical phenomena involved, happening at a range of different time and length scales. Air-breathing vehicles that travel at hypersonic speeds, defined as Mach 5 and

above, have internal flows within the engine at supersonic speeds. To tailor these flows to reach successful combustion, the geometry must be exact and can be complex to model. The flows also experience extreme temperatures and pressure differences, forming strong shock waves, chemical reactions, and air ionization. As a result, the behavior of the flow becomes highly nonlinear, with complex interactions between fluid dynamics, thermodynamics, and chemistry. Accurately capturing these effects requires sophisticated computational models capable of resolving fine-scale flow structures while managing the wide range of time and length scales in hypersonic regimes. Additionally, the strong viscous effects near the surface lead to highly complex boundary layer interactions, further complicating the modeling effort [15–18].

Many CFD codes, including both commercial and research-based solvers like CFD++, ANSYS Fluent, Vulcan, and OpenFOAM, have been employed to simulate these flows [19–24]. While commercial solvers can provide preliminary results for design, the unique features of supersonic flows within hypersonic vehicles often require further customization of models and codes, especially in capturing detailed shock wave boundary layer interaction effects. This need has motivated the expansion of the in-house APDL-CFD code described in this thesis.

There is an expanding push to develop more efficient control techniques for the shock waves within supersonic channel flow. This is where CFD becomes a valuable tool. CFD allows for rapid iterations through parameters and different control types without performing costly flight tests and can also expose areas not seen through experimental sensors alone. This research aims to develop a CFD model capable of running rapid test cases to optimize control techniques for shock waves, informing future experiments at the CSU wind tunnel [25]. This requires a code capable of simulating high-speed flows around complex geometries and capturing the intricate dynamics of shock waves and boundary layer interactions and the effects of plasma actuators on these phenomena.

One of the critical challenges in accurately simulating supersonic flows within hypersonic engines, particularly in the context of complex geometries like scramjet inlets or isolators, is the need for precise geometric representation in computational models. Traditional CFD methods rely on

body-conforming or mapped meshes on structured or unstructured grids and are computationally expensive to generate. Cut cell methods have been developed to address this as an alternative approach for simulating flow in and around complex geometries. Cut cell methods improve resolution at the interface between solid boundaries and the fluid flow by allowing cells to be cut along the geometry boundaries. These improvements are all done while keeping the simplicity and efficiency of a Cartesian grid [26–31].

In this thesis, cut cell methods are employed to model flow through supersonic channel flow and the shock wave-boundary layer interactions present in the channel. Additionally, plasma actuators are modeled to control the oblique shock waves originating from a wedge generated through a cut cell geometry. The cut cell methods play a pivotal role in efficiently modeling complex geometries required to generate shock waves and compress air inside a supersonic channel. Further details on implementing these methods within the APDL-CFD code are provided in Chapter 3.

The central focus of this work is the active control of shock waves using plasma-assisted techniques, which can manipulate the flow field and improve system stability dynamically. By investigating these control methods, this research aims to expand the operational envelope of hypersonic propulsion systems, enhancing both performance and robustness. Integrating cut cell methods and plasma actuators expands the possibilities in modeling control methods for shock wave boundary layer interactions inside a hypersonic vehicle engine.

Subsequent sections provide a literature review on historical hypersonic CFD techniques and the modeling of complex geometries using CFD. Next, the objectives of this thesis are presented, and then an outline of the thesis structure is given.

1.2 Literature Review

This section provides a historical overview of the development of hypersonic technologies and highlights how computational fluid dynamics (CFD) has significantly contributed to this advancement. Additionally, it discusses the evolution and current status of Cartesian cut cell methods used for representing complex geometries in CFD.

1.2.1 History of CFD for Hypersonic Flows

The literature on modeling hypersonic flow control is extensive, reflecting the complex nature and interest in the problem [32, 33]. Numerous studies have developed highly customized codes and methods tailored to specific flow regimes, vehicle configurations, and control strategies. Modern models often integrate advanced CFD techniques with data from physical experiments to capture the diverse interactions between shock waves, boundary layers, and other physical and chemical phenomena occurring in hypersonic flows. CFD plays a pivotal role in aerospace and other industries, complementing experimental methods [34].

Hypersonic technologies and research have matured over 60 years of development [35], becoming a high priority for defense, commercial travel, and launch capabilities [36, 37]. However, the CFD modeling of hypersonic flows is still evolving, with its origins tracing back to the 1980s [38]. Before the advent of sophisticated CFD codes, development relied heavily on analytical methods, wind tunnel testing, and flight tests. Physical testing in hypersonic conditions is costly and dangerous. The introduction of CFD has made rapid prototyping and parametric studies, such as geometry optimization and flow control, feasible, drastically improving design efficiency. This aligns with the goal of this research: to develop a CFD model capable of running fast test cases to optimize control techniques for shock waves, thereby informing future experiments at the CSU wind tunnel [25].

Advances in computational power and numerical methods over recent decades have enabled more sophisticated simulations, particularly for hypersonic flow control [39]. The complexity of the underlying physics and the growing interest in hypersonic applications have spurred extensive research in this area [40]. In the early 1980s, Cosner et al. developed the X3D solver, a Reynolds-averaged Navier-Stokes (RANS) method that successfully informed early military aircraft designs [41]. However, at that time, CFD was inadequate for hypersonic applications due to the complexity of the flow physics, leading to a decade dominated by experimental research. In 1987, Dwoyer et al. [42] highlighted how CFD technologies could not efficiently handle the complex physics of hypersonic flow. Two years later, Povinelli [43] outlined the requirements for

developing a hypersonic flow solver, explicitly noting the differences from traditional CFD solvers. With the growth of computing power and the development of new supercomputers by the United States Department of Defense (DoD), the feasibility of using CFD for evaluating hypersonic flow became more realistic towards the 1990s and beyond [44].

A notable achievement celebrating CFD's role in hypersonic research is the National AeroSpace Plane (NASP) program, which aimed to develop a single-stage-to-orbit vehicle capable of hypersonic flight. The program, initiated in the 1980s, was a joint effort between NASA and the DoD, with the goal of developing a vehicle that could reach Mach 25. The NASP program was a significant driver of hypersonic research, leading to advancements in CFD and experimental techniques. At the time, a substantial fraction of the country's supercomputing power was dedicated to the project. The program was eventually canceled, but much of the aerodynamic design was already completed. The project left a lasting impact on the use of CFD in hypersonic research and development [45], and even helped investigate problems like aerodynamic flutter.

Another crucial application of CFD emerged in space capsule re-entry modeling, where the design and sizing of thermal protection systems (TPS) for re-entry vehicles were driven by the need to withstand extreme thermal loads. CFD simulations were employed to predict the heat transfer rates and aerodynamic forces during re-entry, which informed the design of heat shields. These simulations were crucial in developing capsules like the Apollo and Orion, significantly reducing the need for extensive physical testing [46].

In addition, CFD has been instrumental in designing complex inlet geometries for scramjets and other hypersonic vehicles. By capturing shock-shock and shock-boundary layer interactions, CFD models have optimized inlet shapes for efficient compression and reduced flow separation, particularly in cases where traditional experimental methods would have been too costly or challenging [47]. CFD has also enhanced optimization capabilities, particularly in topology optimization. Integrating topology algorithms with CFD has allowed for the exploration of a more expansive design space and the identification of optimal configurations for hypersonic vehicles, improving performance and efficiency in challenging flight regimes.

Research has recently focused on coupled problems, such as aerothermoelastic deformations in hypersonic vehicle inlets. These studies combine finite element methods (FEM) with CFD to simulate how thermal loads and aerodynamic forces deform inlet structures, affecting flow patterns and potentially leading to performance degradation [40]. This coupling of structural and fluid simulations has become a cutting-edge research area, addressing the intricate interactions between temperature, material deformation, and airflow at extreme speeds.

Recent discussions stemming from the CFD Vision 2030 study have emphasized the need to enhance the accuracy and robustness of CFD tools applied to hypersonic scenarios [48]. The study outlines future research directions aimed at increasing the use and reliability of CFD in high-risk areas, providing guidelines for researchers to identify critical topics requiring immediate attention. One such area is plasma modeling for flow control and combustion, which is a large part of this research [49]. Additionally, there is a push to increase computing power to make this research more efficient.

1.2.2 Representing Complex Geometries in CFD

One of the most expensive aspects of any CFD simulation is mesh generation. Cartesian meshes are highly advantageous for hypersonic problems due to several reasons. Their uniform structure minimizes storage requirements for grid connectivity, making them highly memory-efficient. Many multigrid and adaptive solvers perform better on Cartesian grids because of the alignment of elements and their structured nature. The regular arrangement of cells allows for using preconditioned solvers and optimized parallelization strategies. Furthermore, the alignment of grid lines with the coordinate axes minimizes numerical diffusion compared to skewed or unstructured grids, resulting in more accurate solutions for problems with aligned flow fields. Cartesian grids are also well-suited for domain decomposition techniques, enabling efficient parallel computation where each process can handle a block of the domain independently with minimal communication overhead.

However, Cartesian meshes alone cannot easily accommodate complex geometries typical in hypersonic applications. To address this limitation, Cartesian cut cell methods provide a cost-effective solution by allowing for a regular grid that accommodates complex geometries without altering the base Cartesian mesh [30,50–54]. This is achieved through the use of embedded boundaries (EB). A vital advantage of this method is that it calculates all the geometric parameters for complex geometries during preprocessing, eliminating additional costs during the CFD simulation. The technique allows for a more accurate representation without requiring a highly refined mesh or an unstructured grid by overlaying a geometry on a Cartesian mesh and cutting cells where the geometry intersects. This approach leverages the benefits of Cartesian grids and extends their applicability to complex geometries. However, this approach is not without drawbacks, the most notable being the small cell problem [30, 55], which will be discussed in more detail in Chapter 3.

Charles Peskin is often credited with developing the immersed boundary (IB) method [56], which he used to simulate blood flow in the heart. This method differs from the EB method, as IB does not use cut cells but treats the boundary with ghost cells. Nevertheless, Peskin's work helped usher further research in representing complex geometries, leading to methods like EB. Additional information on the differences between the two methods is found in Section 3.6.

A significant portion of the research for this thesis has focused on developing the cut cell method to enhance the modeling of complex geometries, particularly for generating and controlling shock waves. For instance, Ghosh [3] developed an EB technique to simulate flow over control devices such as vortex generators, bleed holes, and aeroelastic mesoflaps for supersonic flow control. The primary mechanism of the EB method is to classify cells as fluid, solid, or cut cells and then treat each type differently. Ghosh demonstrated that the cut cell technique improves accuracy around complex surfaces, showcasing not only the effectiveness of the cut cell method but also the potential for controlling supersonic flow.

Many variations of cut cell methods have been developed for a variety of applications. Causon et al. [57] used cut cells with the cell merging technique to model moving boundaries and ultimately applied it to ships moving in water using the shallow water equations. Another method

is dimensional splitting with embedded boundaries, as described by Liu et al. [58], which efficiently represents geometries on Cartesian meshes. This is similar to the method used in this thesis in which dimensional splitting is combined with the cut cell method. Berger et al. [59] utilized rotated coordinates to enhance flux reconstruction around cut cells. Their method introduces the h-box approach, which manages the irregular cells at the boundary and creates a rotated coordinate system to solve the boundary flux, allowing for the calculation of the flux normal to the wall. This technique improves stability and accuracy for one-dimensional systems and is extended to two dimensions.

The integration of multigrid methods [54] and adaptive mesh refinement (AMR) with cut cell methods has further improved simulation capabilities. Berger, one of the pioneers in this space, often combines AMR with cut cell methods [60], allowing for dynamic refinement of the grid in regions of interest, such as shock wave boundary layer interactions. The combination of AMR and cut cell methods is particularly effective in hypersonic flow simulations, where high-resolution grids are necessary for capturing the intricate details.

1.3 Objectives

The following research objectives are identified for the thesis.

1. Develop and validate a CFD model with the capabilities to simulate the flow inside a supersonic channel with complex geometries.

2. Simulate the effect of plasma injection on shock wave trains and boundary layer interactions.

3. Expand the APDL-CFD code using a Cartesian cut cell method to represent embedded boundaries.

1.4 Thesis Organization

The rest of this thesis is organized as follows. The physics of supersonic flow is given in Chapter 2, including a discussion on compressible flow, shock wave boundary layer interaction (SWBLI),

and control of oblique shock waves. Chapter 3 describes the mathematical model, governing equation, and the cut cell method used in the APDL-CFD code. Chapter 4 details the numerical modeling and plasma-assisted shock wave control results. Chapter 5 presents the improvements to the APDL-CFD code by including the cut cell method. Finally, Chapter 6 concludes the thesis with a summary of the findings and recommendations for future work.

Chapter 2

Supersonic Flow Physics

This chapter discusses the fundamental physics of supersonic flow. It begins by discussing the applications of these physics to scramjets (Supersonic Combustion Ramjet). Then, it continues with an overview of compressible flow, followed by a discussion on shock wave boundary layer interactions (SWBLI) and the control of these interactions and the shock wave trains. A discussion of the control methods in the literature is followed by a review of the plasma control method used in this research.

2.1 Introduction

It is important to distinguish between hypersonic and supersonic flow. Supersonic flow refers to speeds greater than the speed of sound, corresponding to a Mach number greater than 1. Hypersonic flow, typically defined as Mach numbers greater than 5, introduces additional complexities beyond those of supersonic flow. At Mach 5 and above, the chemistry of the flow, usually air, begins to change significantly due to the high flow energy causing ionization, resulting in the formation of free electrons and ions. These phenomena introduce thermodynamic and electrodynamic effects that are largely absent at lower Mach numbers. In real-world scenarios, a vehicle may be traveling at hypersonic speeds, but the flow inside the vehicle may remain at supersonic speeds; this is the case for scramjet engines.

The scramjet engine depicted in Figure 2.1 consists of several key components that work together to generate thrust at hypersonic speeds. Starting from the inlet, the airflow is compressed by a series of oblique shocks, including the vehicle's bow shock and the forebody compression shock. As the air enters the isolator, a critical component that manages the interaction between the inlet and combustor, the shock wave train becomes more complex, with contributions from both the external flow and the combustion process. The red shock waves in the figure represent

shocks generated from the incoming upstream airflow, while the blue shock waves arise from the combustion occurring within the isolator and combustor.

After passing through the isolator, the airflow enters the combustor, where fuel is injected and ignited in the high-speed stream. The challenge here is to achieve stable combustion despite the supersonic or hypersonic flow. Shock waves in the isolator help slow the flow enough for efficient fuel-air mixing and combustion. As combustion occurs, the temperature and pressure of the gases increase significantly, resulting in a robust flow expansion.

The heated gases then expand through the convergent-divergent nozzle, accelerating to supersonic or hypersonic speeds. This rapid expansion generates the thrust required for the scramjet to propel the vehicle forward. The engine relies on careful management of shock-boundary layer interactions and forebody compression to maintain optimal airflow throughout the system. The integration of propulsion and airframe ensures smooth flow transitions, contributing to efficient flight at high Mach numbers. The combined effect of proper compression, combustion, and expansion ultimately generates the thrust necessary for the scramjet to maintain its high-speed operation.

Unlike conventional jet engines, which have moving parts like compressors and turbines to compress incoming air, scramjets have no moving parts. Instead, they rely on the engine's geometry to compress the air as it moves at supersonic speeds. The airflow remains supersonic throughout the engine, including the combustion stage, which increases combustion efficiency.

Despite these advantages, scramjets face significant challenges. The whole process is susceptible and requires precise control. Since there are no moving parts, controlling the shock waves in the isolator that lead to the combustor is difficult. The inability to control these shock waves can lead to *unstart*, where the engine loses the oxidizer and combustion ceases.

Unstart is a failure mode in scramjets where the shock wave system within the engine becomes unstable, causing the flow to reverse direction and propagate upstream. This occurs when shock wave trains interact improperly, especially when shocks from the inlet and isolator interact. This disruption causes the flow inside the engine to become subsonic while the external airflow remains supersonic, as illustrated in Figure 2.2. When unstart occurs, the engine loses its oxidizer

source, effectively ceasing combustion and resulting in a sudden loss of thrust. This can lead to catastrophic failure of the engine, with loss of vehicle control. Consequently, understanding and controlling shock wave dynamics is critical to hypersonic propulsion system design.

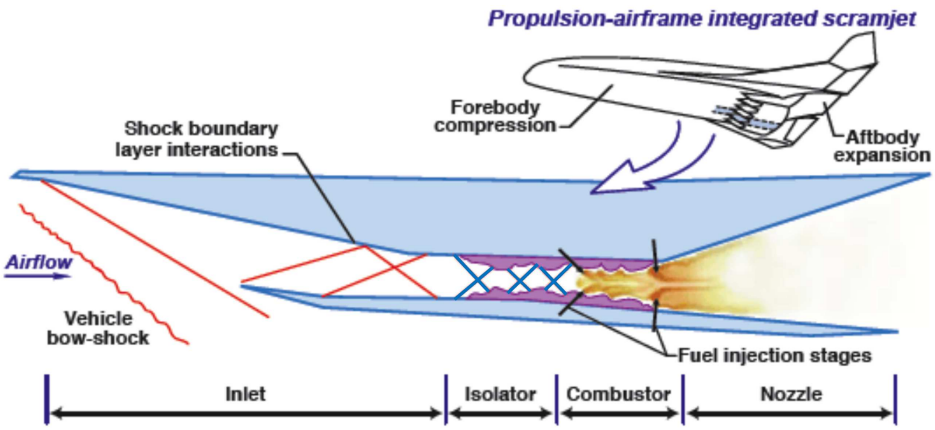

Figure 2.1: Scramjet illustration with shock wave trains in the inlet and isolator. The red shock waves represent those in the inlet from upstream airflow, while the blue shock waves originate from the combustion process in the isolator. Modified from [61].

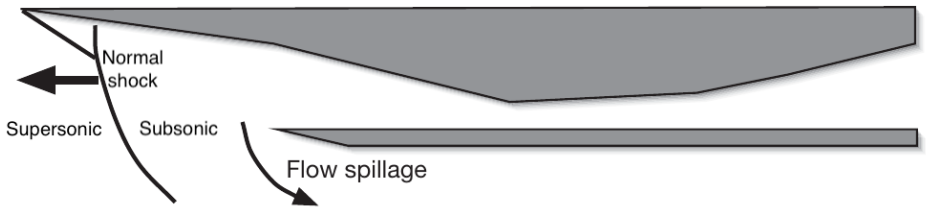

Figure 2.2: Scramjet illustration depicting unstart. The shock wave structure is disrupted, leading to a loss of compression and engine failure. Modified from [62].

Starting a scramjet is also challenging due to the high-speed requirements for proper operation. The Hyper-X is a notable example of a hypersonic vehicle with a limited operating envelope. This vehicle was attached to a rocket, launched from an aircraft, and accelerated to the required speed by the rocket. Once at speed, the scramjet would detach from the rocket and begin operation. This scenario illustrates a typical challenge with scramjets: they have a limited operating envelope due to a lack of flow control.

Another critical challenge in hypersonic flow is the significant heating of the boundary layers, especially near solid surfaces. This intense thermal load, primarily caused by the high kinetic energy of the flow, must be carefully managed to avoid material failure in hypersonic vehicles. To handle this, vehicles like scramjets often rely on ablative materials, which absorb and dissipate heat by gradually eroding. While the material specifics are outside the scope of this thesis, it is important to note that this is another significant factor in the limited operating envelope of these vehicles.

Additionally, separation bubbles, regions where the boundary layer detaches from the surface, pose significant challenges. Increased drag, loss of efficiency, and potential flow instabilities can result from improper control of separation bubbles. On the contrary, separation bubbles can be used to control oblique shock waves. A review of control techniques for separation bubbles and their interaction with oblique shock waves will be given in Section 2.2.3.

Passive and active control techniques have been developed to address these challenges in managing shock waves and separation. Passive techniques, such as fixed geometries, are built into the design and operate without adjustments during flight. While effective in certain conditions, they often struggle to adapt to varying flight regimes, limiting their operational flexibility. Active control techniques, such as movable flaps, mass injection systems, or plasma actuators, can adjust based on changes in flow conditions. These techniques offer greater flexibility, allowing hypersonic vehicles to operate across a broader range of speeds and altitudes. An in-depth review of techniques and the rationale for using plasma actuators is given in Section 2.3.

2.2 Compressible Flow

Ever since Captain Chuck Yeager broke the sound barrier in 1947, interest in supersonic flow has surged, leading to an increased focus on the study of compressible flow. This section provides a background on compressible flow, emphasizing the physics of shock waves, including normal and oblique shocks, and their interactions with boundary layers.

2.2.1 Normal Shock Waves

Normal shock waves occur when supersonic flow encounters a sudden change in flow conditions, resulting in a shock wave perpendicular to the flow direction. As the flow passes through a normal shock, the Mach number decreases from supersonic to subsonic, while static pressure, density, and temperature increase. However, despite the increase in static pressure, the total (stagnation) pressure decreases across the shock due to an increase in entropy.

The properties of the flow across a normal shock can be described by the conservation laws for mass, momentum, and energy, leading to the Rankine-Hugoniot relations. The pressure ratio across a normal shock is given by:

$$\frac{P_2}{P_1} = \frac{2\gamma Ma_1^2 - (\gamma - 1)}{\gamma + 1}, \tag{2.1}$$

where P_1 and P_2 are the static pressures before and after the shock, γ is the specific heat ratio, and Ma_1 is the upstream Mach number.

The total pressure ratio, accounting for the entropy increase and irreversible losses, is expressed as:

$$\frac{P_{t2}}{P_{t1}} = \left[\frac{(\gamma + 1)Ma_1^2}{(\gamma - 1)Ma_1^2 + 2}\right]^{\frac{\gamma}{\gamma - 1}} \left[\frac{(\gamma + 1)}{2\gamma Ma_1^2 - (\gamma - 1)}\right]^{\frac{1}{\gamma - 1}}. \tag{2.2}$$

Additional relations that describe the changes in properties across a normal shock wave are given below:

$$\frac{T_2}{T_1} = \frac{[2\gamma Ma_1^2 - (\gamma - 1)]\,[(\gamma - 1)Ma_1^2 + 2]}{(\gamma + 1)^2 Ma_1^2}, \tag{2.3}$$

$$\frac{T_{t2}}{T_{t1}} = 1, \tag{2.4}$$

$$\frac{\rho_2}{\rho_1} = \frac{(\gamma + 1)Ma_1^2}{(\gamma - 1)Ma_1^2 + 2}, \tag{2.5}$$

$$Ma_2^2 = \frac{(\gamma - 1)Ma_1^2 + 2}{2\gamma Ma_1^2 - (\gamma - 1)}, \tag{2.6}$$

where the subscripts 1 and 2 refer to upstream (before the shock) and downstream (after the shock) conditions, respectively. P_t and T_t denote the total pressure and total temperature, ρ represents the density, and Ma is the Mach number.

These equations illustrate how the total pressure decreases across a shock wave due to the increase in entropy. In contrast, oblique shock waves occur at an angle to the flow direction, and the flow downstream can remain supersonic depending on the shock angle and upstream Mach number.

2.2.2 Oblique Shock Waves

Oblique shock waves are a common feature in supersonic and hypersonic flow. They occur when supersonic flow encounters a change in flow direction, leading to changes in pressure, density, and velocity. Unlike normal shocks, which are perpendicular to the flow, oblique shocks are inclined at an angle β to the flow direction.

Oblique shocks can be observed when flow passes over a wedge or an airfoil, where the surface deflects the flow. Essentially, the particles turn into each other, creating molecular collisions that form a shock. This interaction increases static pressure and density while the velocity decreases as the flow slows across the shock.

The changes across a shock wave are abrupt. In contrast, an expansion wave causes abrupt changes in the opposite direction, decreasing static pressure and density while increasing velocity. This is because the flow accelerates around a corner. Table 2.1 provides a summary of these changes in shock waves and expansion waves. Taylor and Maccoll notably described this phenomenon in 1933 [63]. An illustration of the behavior of flow around concave and convex corners, which generate oblique shock waves and expansion waves, respectively, is presented in Figure 2.3. In Figure 2.3 (a), the flow over a concave corner generates oblique shock waves. In contrast, the flow over a convex corner in Figure 2.3 (b) creates expansion waves.

The shock angle β is related to the Mach number Ma and the deflection angle θ through the following relation:

Table 2.1: Summary of Flow Properties Across a Shock Wave

Wave Type	Mach	Pressure	Temperature	Density
Shock Wave	↓	↑	↑	↑
Expansion Wave	↑	↓	↓	↓

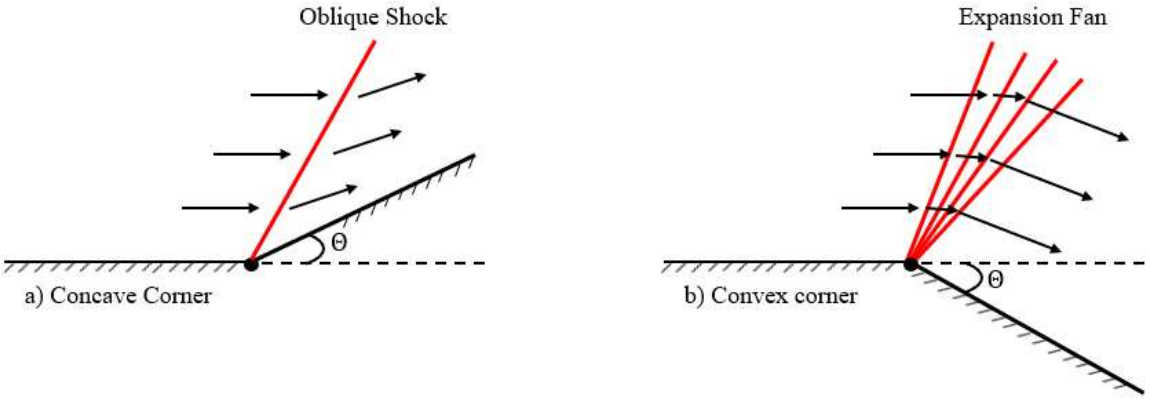

Figure 2.3: Illustration of the flow over (a) a concave corner creating oblique shock waves and (b) a convex corner creating expansion waves.

$$\tan(\theta) = 2\cot(\beta)\left[\frac{Ma^2\sin^2(\beta) - 1}{Ma^2\left(\gamma + \cos(2\beta)\right) + 2}\right], \tag{2.7}$$

where θ is the deflection angle (the angle of the wedge), β is the shock angle, Ma is the Mach number, and γ is the specific heat ratio. This equation is crucial for experimental quantification and calibration but can also be used to validate computational models. In the simulations conducted for this thesis, differentiating between the various flow regimes in supersonic flow is essential for understanding and modeling the effects of oblique shocks on the flow field. If the Mach number and ramp angle are known, the angle of the oblique shock wave can be calculated using the θ-β-Ma relation given in (2.7). Figure 2.4 gives a graphical illustration of the relationship between the deflection angle θ, shock angle β, and Mach number Ma for a wedge. The numerical schlieren in the bottom right corner of Figure 2.4 confirms that the Mach number for the numerical experiments is $Ma = 2.5$. It should be noted that for each θ and Ma, there are two solutions for β. The larger value is called the strong shock, and the smaller is the weak shock. In a solution where the flow

goes from supersonic to subsonic, that is a strong shock. Otherwise, the downstream conditions are supersonic, and it is a weak shock. The flow conditions in this case are supersonic on both sides of the shock, so the weak shock is the relevant solution.

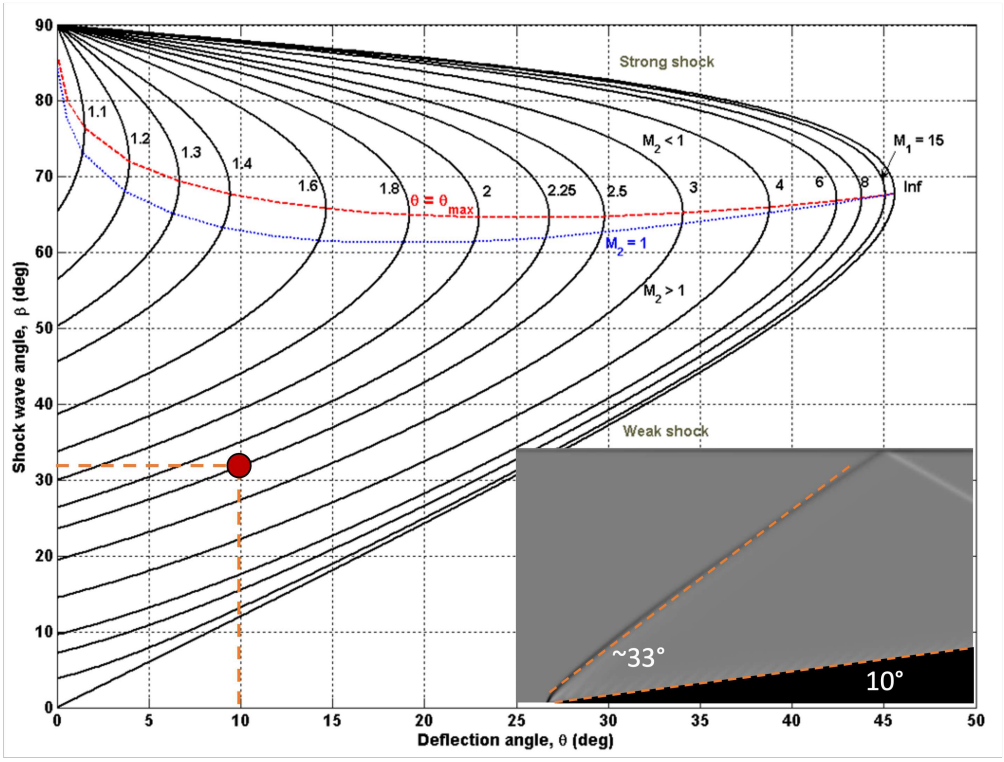

Figure 2.4: Graphical representation of the θ-β-Ma relations for a wedge, confirming that the Mach number for the numerical experiments is $Ma = 2.5$.

2.2.3 Shock Wave Boundary Layer Interaction

Separation bubbles (SB) form when the boundary layer detaches from the surface due to the local adverse pressure gradient generated by the interaction with a shock wave [18]. These bubbles can significantly affect performance by increasing drag and disrupting flow stability. In many cases SB are considered detrimental to performance, especially in external flow applications such as aerodynamics. On the contrary, separation bubbles can have favorable effects that can benefit the performance. Thus, managing these separation bubbles is essential in hypersonic vehicles.

Various factors, including flow velocity, pressure gradients, shock impacts, and surface roughness, influence the formation of separation bubbles.

The interaction between shock waves and boundary layers in supersonic flows leads to complex flow phenomena. Figure 2.5 provides a schematic representation of a SB caused by an oblique shock wave (incident shock) in a supersonic flow. From left to right, the figure illustrates the following stages: Oblique shock waves impacting the boundary layer on the wall cause an adverse pressure gradient, which is strong enough to cause boundary layer separation. The separation starts slightly upstream of the impinging shock. The incident shock wave hits the boundary layer, causing the boundary layer to thicken due to the adverse pressure gradient. As the boundary layer separates from the surface, a recirculation region forms the SB. Around the SB is a shear layer as the flow goes over the bubble. Simultaneously, the SB now acts as an object in the flow field. This causes compression waves to form into a reflected shock wave to form at the front of the separation region, which intersects with the incident shock. Then, the flow goes over the SB and expands, forming the expansion fan. Then, the flow compresses after the expansion wave and can form a second shock, the reattachment shock. The reattachment shock is not shown in the schematic in Figure 2.5 but is shown in the computational results in Figure 2.6.

In numerical simulations, the SB can be observed in finer detail. Figure 2.6 shows a simulated result from the APDL-CFD code of a SB caused by an oblique shock wave impacting the boundary layer. In this simulation, the bulk flow is $Ma = 2.5$, and a wedge, represented using the cut cell method (see Section 3.6.1), generates the incident shock. The physics of the SB follows the schematic in Figure 2.5, but there is an evident second reflected shock in the simulation. The reattachment shock wave is a result of the compression waves coalescing and forming a shock wave. However, this is much weaker than the leading edge shock.

The SB is characterized by a recirculation zone that forms when the boundary layer detaches due to the impinging shock. In this case, the oblique shock wave creates a significant adverse pressure gradient, leading to boundary layer separation and bubble formation. Within this region, there is a strong flow reversal. This simulation allows for a deeper understanding of the behavior

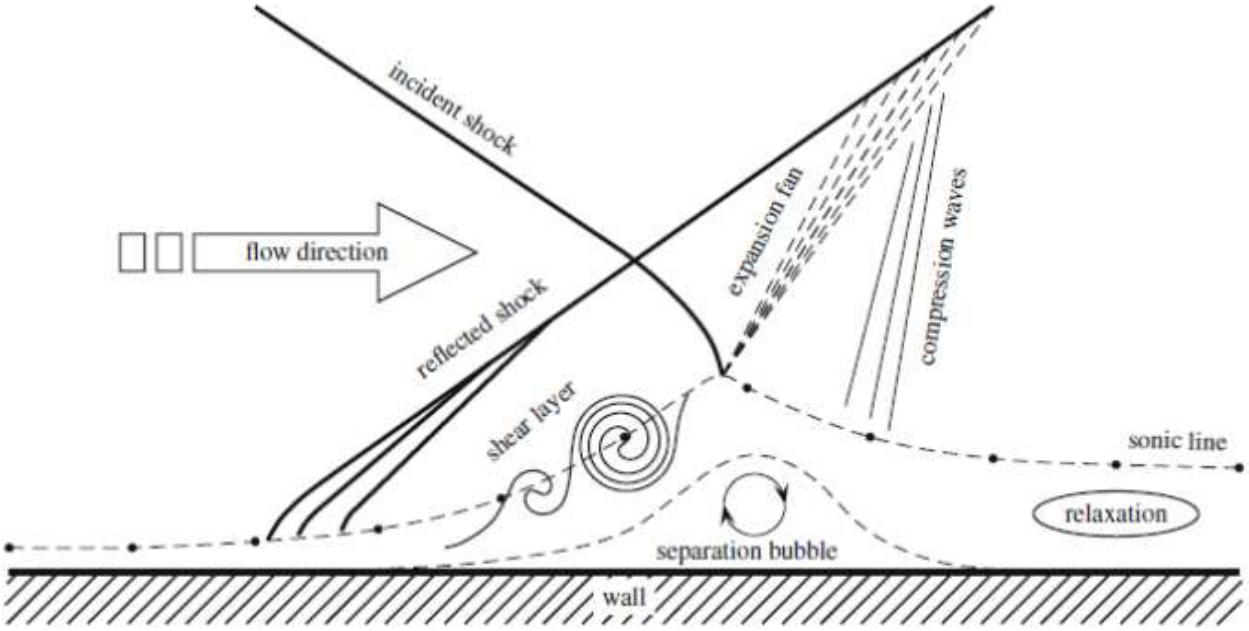

Figure 2.5: Schematic of a separation bubble caused by an oblique shock wave (incident shock) in a hypersonic flow [3].

of these bubbles and their effects on flow dynamics. The movement and growth of the SB is a crucial factor to control and will lead to the control of the shock wave train explained in more detail in Chapter 4.

The wall temperature within or upstream of the SB region is also an essential factor influencing the flow. This is especially true in the hypersonic regime, in which significant differences exist between the wall temperature and the outer-flow stagnation temperature. However, wall temperature effects and surface heating are outside the scope of this thesis.

2.3 Control of Shock Waves

In supersonic flows, controlling shock waves is critical for maintaining vehicle stability, efficiency, and performance [64–66]. If uncontrolled, shock wave-boundary layer interactions (SWBLI) can lead to adverse effects, such as increased drag and flow instabilities, potentially causing unstart. Both passive and active control techniques have been developed to mitigate these effects, each offering distinct advantages and challenges [1]. Passive control methods typically involve altering the surface geometry to influence the flow. In contrast, active control strategies introduce

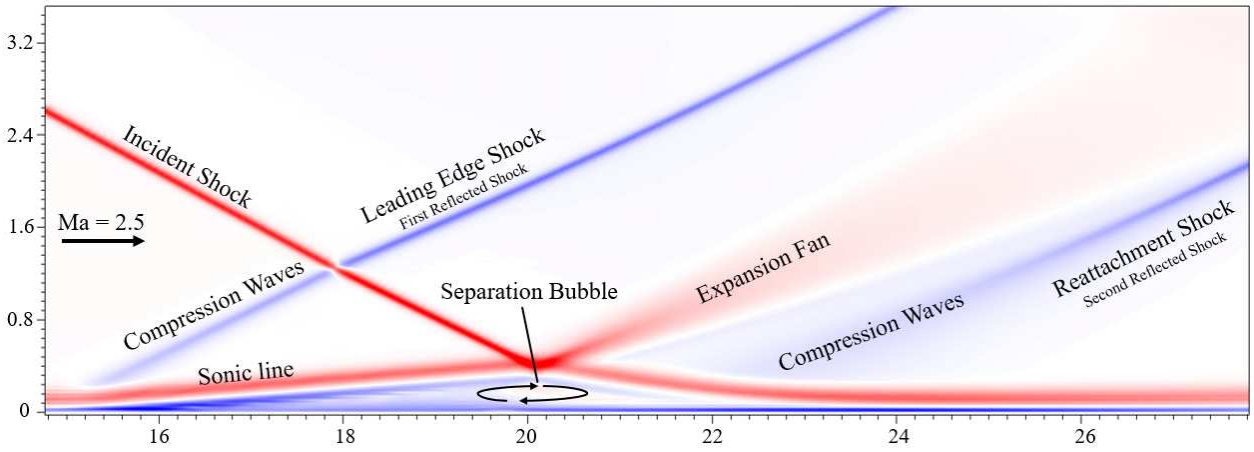

Figure 2.6: Numerical schlieren plot for $\partial\rho/\partial y$ illustrating the separation bubble observed from simulation.

external energy to modify the flow behavior more dynamically and adaptably. Understanding the trade-offs between these approaches is crucial for designing hypersonic vehicles that can operate efficiently in various conditions. Table 2.2 provides an overview of different control techniques for managing shock waves in hypersonic flows, highlighting their benefits and drawbacks.

2.3.1 Passive Control Techniques

Passive control techniques are designed to influence the flow without additional energy input. These methods are simpler due to the lack of external energy required. Common passive control techniques include micro-vortex generators (MVGs) [4, 67], porous cavities [5], surface bumps [68], slots and groves [69, 70], and splitter plates [71]. These methods aim to redistribute the flow, weaken shock waves, and suppress separation bubbles, ultimately improving vehicle performance in hypersonic regimes.

Micro-vortex generators (MVGs) and micro ramps are a widely used passive technique. MVGs introduce streamwise vortices into the boundary layer, which energizes the boundary layer and reduces separation caused by shock waves. The vortices generated by MVGs can weaken oblique shocks and change their reflection angles. However, the height of MVGs can contribute to additional drag, and the location of these ramps is fixed [72].

Table 2.2: Control Techniques for Shock Wave Management in Hypersonic Flows. Modified from [1].

Control Techniques	Benefits	Drawbacks
Porous cavity (Passive)	Splits strong shock into weaker shocks via fluid recirculation.	Thickens incoming boundary layer.
Micro-Vortex Generators (MVG) (Passive)	Suppresses separated shear layer by energizing the boundary layer with vortices.	Results in additional drag due to MVG height.
Surface bump (Passive)	Suppresses boundary layer separation by introducing a bump.	Less effective at off-design conditions.
Slots and grooves (Passive)	Improves total pressure recovery by smearing strong shock into lambda shock.	Thickens boundary layer, causing additional viscous penalty.
Splitter plate (Passive)	Reduces shock intensity by splitting strong SWBLI into weaker zones.	Exhibits additional drag and can create more shock waves.
BL bleed/suction (Passive/Active)	Suppresses boundary layer by sucking low momentum fluid out of the boundary layer.	Reduces ingested mass flow, requiring larger intake area, increasing drag and weight.
Tangential blowing (Active)	Reduces separation bubble size by energizing boundary layer with high velocity fluid.	Consumes significant pressurized air, decreasing engine efficiency.
Micro jets (Active)	Reduces shock intensity by breakdown of shock foot	Requires additional energy, reducing overall engine efficiency.
Air-Jet Vortex Generator (Active)	Creates streamwise vortices, no parasitic drag.	Consumes extra energy, decreasing engine efficiency.
Plasma jets (Active)	Improves pressure recovery, extremely tunable.	High temperatures lead to increased energy for cooling requirements.

The red lines in Figure 2.7 illustrate what happens when a micro ramp is placed in the correct location. Ideally, it is placed where the shock wave impinges on the wall. The recirculation region at the back of the ramp causes an adverse pressure gradient and will weaken and change the reflection angle of the shock wave. The challenge with these passive control techniques is that they are not easily tunable since the micro ramps are in fixed positions. This limits the operational capabilities of the control technique.

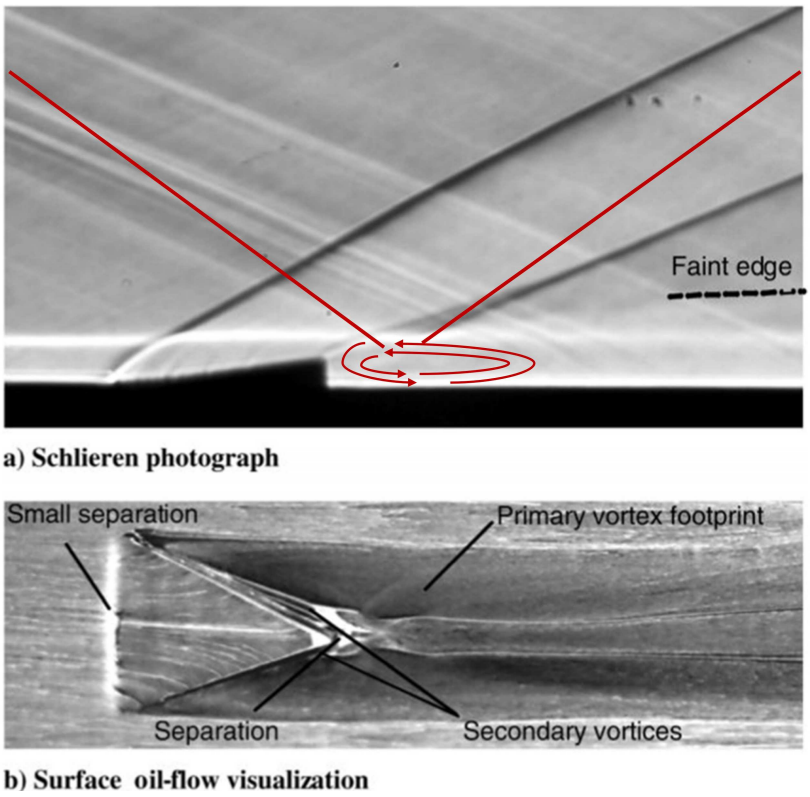

Figure 2.7: Micro-ramps are used to redistribute the flow and reduce shock intensity [4].

Another common passive control method is to use porous cavities [7], shown in Figure 2.8. These aim to reduce the impact of SWBLI by altering the shock wave structure or splitting shocks into weaker lambda shocks. Porous cavities introduce a region of high circulation, which causes localized flow disturbances that can help grow the boundary layer, but their effectiveness diminishes at off-design conditions [1, 73]. Despite the limitations of porous cavities and MVGs, these techniques are attractive due to their simplicity and the lack of an external energy requirement.

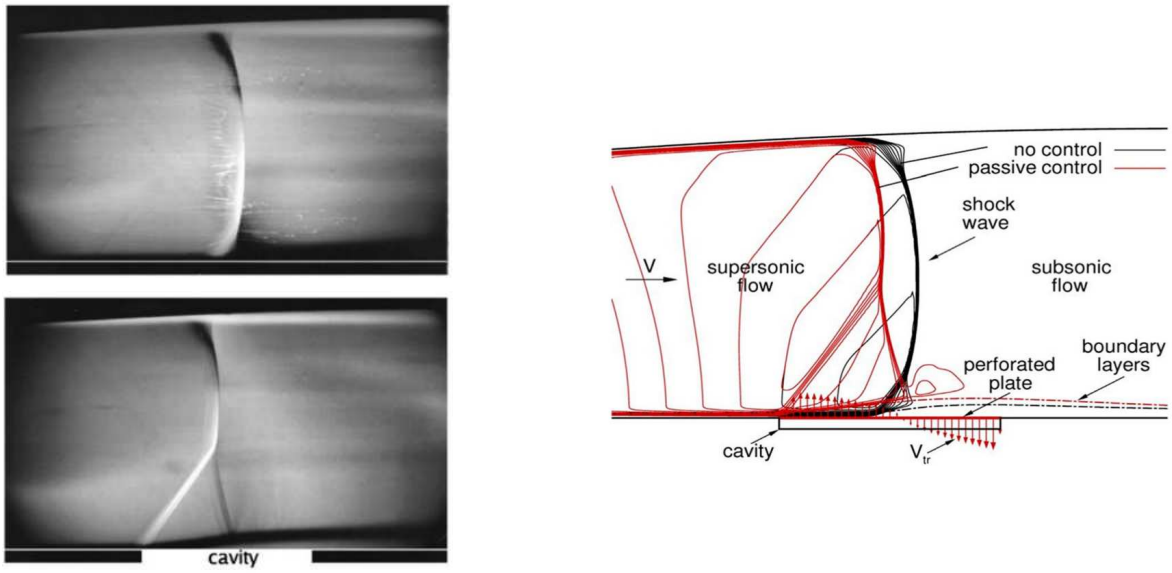

Figure 2.8: Porous cavities are used to weaken normal shocks [5]. The red shock waves on the right illustrate the splitting and weakening of the shock and the growth of the boundary layer.

2.3.2 Active Control Techniques

In contrast to passive methods, active control techniques involve external energy input to dynamically influence the flow and shock wave behavior. These techniques provide greater flexibility and adaptability to varying flow conditions, making them ideal for real-time flow management applications. Examples of active control are: micro jets [7], air-jet vortex generators [74], dynamic boundary layer (BL) bleeds [6], tangential blowing [75] and plasma jets [8]. This section will focus on two methods for brevity, followed by detailed explanations of plasma jets in the next section.

One such method of active control is boundary layer bleed [6, 76–79], where low-momentum fluid is actively removed from the boundary layer upstream of the shock interaction region. The effectiveness of this method, seen in Figure 2.9, is most evident by the absence of the separation bubble and separation shock in the bleed case. This technique reduces the strength of the shock and suppresses separation, but it comes at the cost of reduced mass flow through the intake, requiring larger intake areas and leading to increased drag and weight.

Micro jets represent another active approach, utilizing small high-pressure air jets. These are typically placed upstream of an object to change how the shock moves around the object. The jets

break down the shock foot, reduce the intensity of shock waves, and control their flow around an object. While effective, micro jets require additional energy, which can lower the overall efficiency of the propulsion system [7, 80, 81].

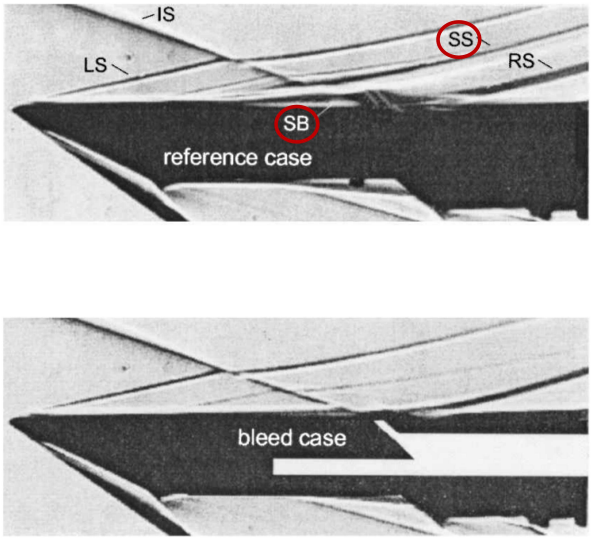

Figure 2.9: Boundary layer bleed reduces the strength of the shock by removing low-momentum fluid from the boundary layer [6].

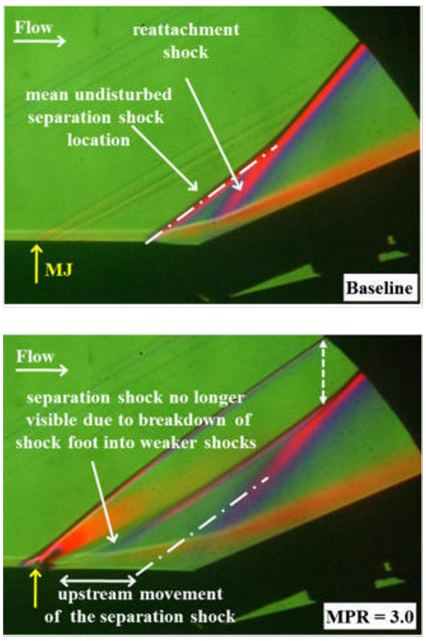

Figure 2.10: Micro jets are used to break down the shock foot and reduce shock intensity [7].

Plasma-based flow control is gaining attention for its potential in shock wave management [82–84]. Plasma jets offer a highly tunable flow control method capable of altering shock wave and boundary layer characteristics through a combination of thermodynamic, electrodynamic, and momentum injection effects. Plasma-based control also offers superior control time scales, which can be activated much faster than mechanical-based methods. Overall, active control methods, particularly plasma-based techniques, offer significant potential for improved shock wave control in hypersonic applications. Plasma actuators are given in more detail in the next section

2.3.3 Plasma Actuators

Plasma as an active device for flow control has been extensively explored [83]. Dumitrache et al. [11] demonstrated that plasma discharges could control ignition in supersonic flows and aid

in flame holding. By using nanosecond pulsed lasers, they could control the direction and intensity of the vortices originating within the plasma kernel, demonstrating advanced control capabilities. Visbal and Gaitonde [84] illustrated how plasma can control flow over airfoils, highlighting the potential of these methods for external flows. However, the focus of this thesis is on internal flow.

Plasma-assisted shock wave control has emerged as a promising active control strategy for internal flows. Zhang et al. [32] and Caruana [83] highlight the potential of plasma discharges in reducing drag, fatigue, and enhancing vehicle stability during hypersonic flight. Shahrbabaki et al. [85] numerically investigated the control of shock position and boundary layer interaction using nanosecond dielectric barrier discharge (DBD) plasma through a 3D simulation. Their findings revealed an upstream movement of the shock waves and separation bubble growth, which aligns with the results presented in this thesis in Chapter 5.

For low Mach number flows, plasma has proven effective in boundary layer acceleration, separation control, and drag reduction [86–91]. However, due to its fast activation and controllability, there has recently been an increased interest in using plasma-based control for hypersonic and supersonic controls. To prevent unstart in scramjets, experimental results show promising uses of plasma to control shocks in inlets and isolators [92, 93].

Specifically, plasma actuators have shown great promise in shock wave management. There are three main mechanisms plasma has on a flow that can be tuned to control flow: momentum injection, shocks, and chemical effects. The momentum injection effect can stabilize the boundary layer by inducing extremely high velocities. This injection induces shock waves since the plasma increases the temperature and pressure of the local flow. Significant vorticity and enhanced turbulent mixing can be induced through the interaction between the fluid and shock waves via baroclinic torque [11, 94]. Additionally, the chemical effects arise from the reactive ions and radicals produced by the discharge.

Different methods such as quasi-direct current (Quasi-DC), dielectric barrier discharge (DBD), and laser filament discharge have been studied for their effects on flow properties [8]. Choosing a plasma discharge type depends on the application and the flow conditions. Due to simplicity

and reliability, DBDs are often selected for low Mach number aerodynamics. However, these only produce a low momentum injection [95]. Alternatively, pulsed discharges, such as quasi-DC or nanosecond repetitively pulsed discharges, can generate higher electron densities and significant temperature gradients within the boundary layer. However, they are more challenging to sustain over extended periods [96]. Caraballo et al. investigated the application of localized arc filament plasma actuators (LAFPAs) for controlling shock wave-boundary layer interactions in a supersonic mixed-compression inlet [97]. Their findings indicated that placing the actuators upstream of the separation bubble increased the normalized streamwise velocity within the SWBLI region.

Leonov's group at the University of Notre Dame implemented a quasi-DC electrical discharge embedded within a ceramic surface on the tunnel wall. Electrodes were arranged in alternating polarity to produce a filamentary plasma loop connecting adjacent electrodes. Optical emission spectroscopy measurements indicated that the initial plasma filament had a diameter of approximately 2 mm and a temperature of about 6000 K [98, 99]. When exposed to supersonic flow, the plasma filament elongated downstream, significantly affecting the boundary layer along the bottom wall.

Experimental results by Elliot et al. in $Ma = 2.0$ freestream flow demonstrated that quasi-DC discharges can recover some of the total pressure loss by suppressing the strong reflected shock at the bottom of the channel and replacing it with a weaker reattachment shock [8]. The schematic of the plasma actuators set up is shown in Figure 2.11. However, this method does not enable uniform plasma generation across the spanwise direction, resulting in pronounced three-dimensional plasma effects.

Elliot's experimental results in Figure 2.12 show the ability of plasma actuators to displace shock waves upstream and downstream. This visualization demonstrates how the plasma actuator affects the intensity and position of the shock waves' second impact location. These trends align with the findings in this thesis, which focus on understanding how plasma-generated ionized gas interacts with supersonic flow to control separation and manage shock waves.

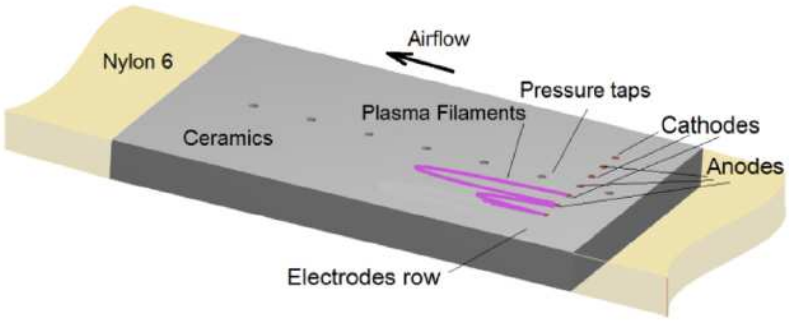

Figure 2.11: Schematic of the plasma actuators used in Elliot's Study. There are alternating cathodes and anodes to create a plasma filament.

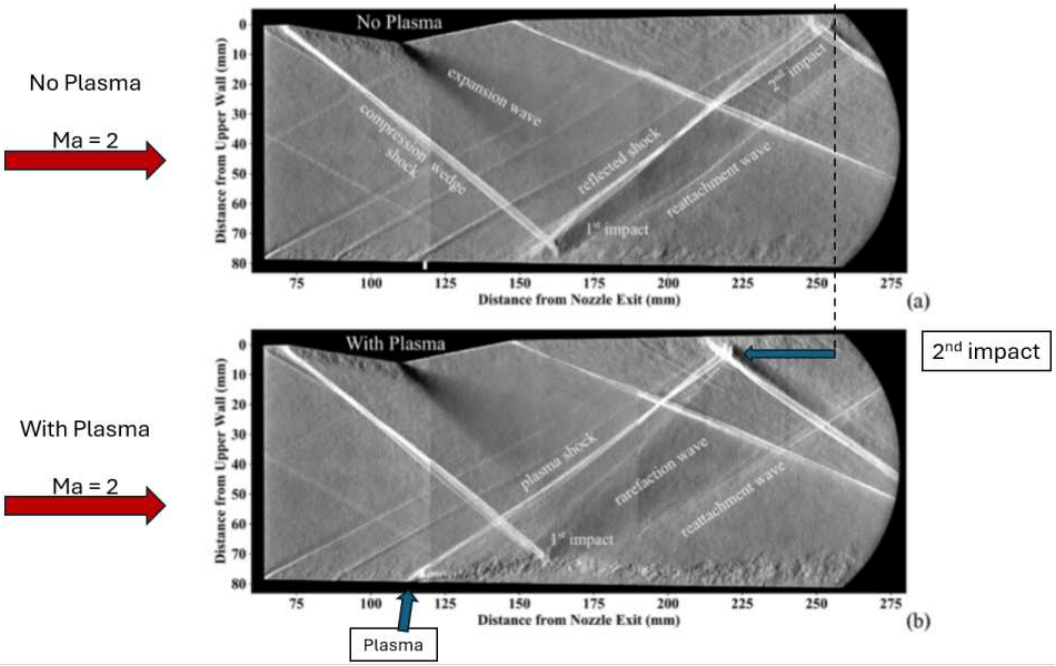

Figure 2.12: Shock wave position and intensity control using a plasma actuator placed upstream, based on experiments by Elliot et al. [8]. The second impact moves about 5 cm upstream with plasma on.

To achieve a more uniform spanwise plasma, Ruggles et al. combined the preionization effect of femtosecond laser pulses to trigger capacitive electrical discharges [9]. Using dual-mode energy deposition, the laser-triggered spark gap generated a spanwise plasma arc across their wind tunnel. Their technique demonstrated control by reducing surface pressure over wedge and compression ramp geometries by creating a low-density plasma kernel spanning the wedge. The model by Ruggles et al. is more suitable for this thesis, as the plasma is uniform across the spanwise direction of the flow. Thus, a 2D model is accurate in this case. The model Ruggles uses is displayed in Figure 2.13. However, for this investigation, the plasma is generated without the electrodes in the flow path to avoid another obstacle in the flow. The plasma is generated using pulsed nanosecond and femtosecond laser filament discharge. The dual pulse method uses an overlapping paired lasers to create a plasma filament, presented in more detail in Figure 4.3 in Chapter 4.

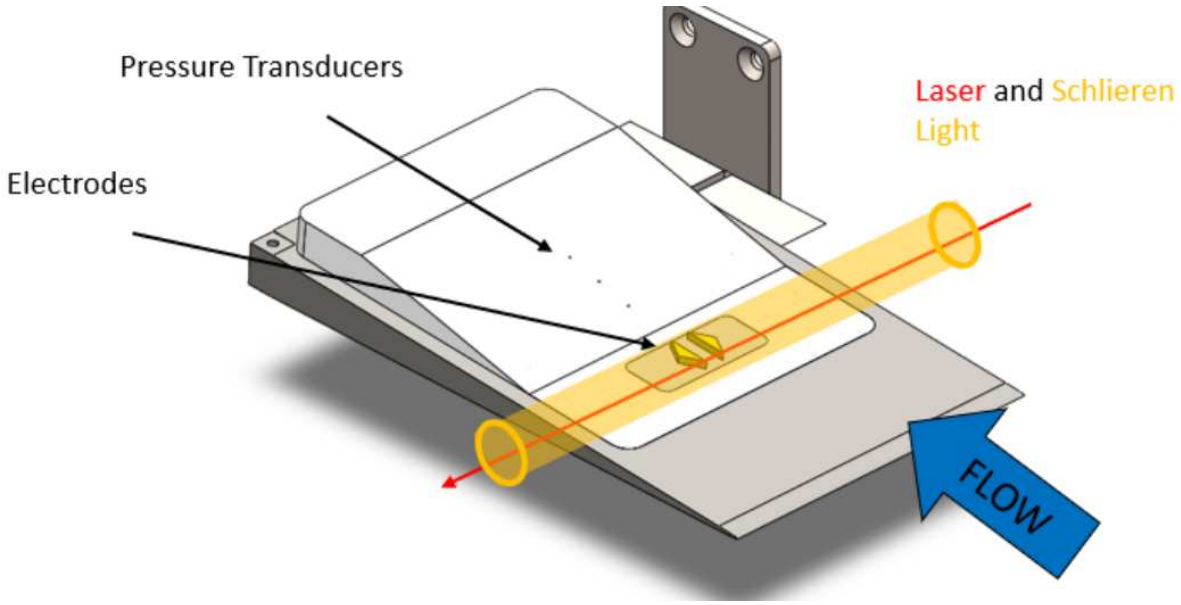

Figure 2.13: The plasma model used by Ruggles et al. [9] to generate a uniform plasma across the spanwise direction.

Chapter 3

Numerical Approach for Supersonic Modeling

In the introduction, this chapter will give an overview of computational fluid dynamics methods. Then, the mathematical model is presented with the governing equations and the numerical methods to solve them. Next, a discussion of the numerical methods used in this thesis to compute the cut cells to model complex geometries is given. The chapter will conclude with a simplified flow chart of the APDL-CFD code.

3.1 Introduction

The development of the APDL-CFD has been an 8-year project starting development for Dumitrache's dissertation [100]. Now, there are nine branches used to solve a wide variety of problems, including combustion, plasma physics, and laser modeling. The current branch built for this thesis is the embedded boundaries branch. The new version of the code includes the addition of complex geometries rather than a standard Cartesian mesh, which can now represent geometries through cut cells. Furthermore, a significant effort has been made to make the code more streamlined, documented, and user-friendly for future students and collaborators.

The primary motivation for further developing the APDL-CFD code is to investigate the bulk flow physics of supersonic flows, specifically the effect of plasma on the control of shock wave trains inside a supersonic channel.

3.2 Mathematical Model

This section describes the mathematical model developed for this thesis. The Navier-Stokes equations are a system of partial differential equations (PDEs) that describe the conservation of mass, momentum, and energy in a fluid. These equations are fundamental in computational fluid dynamics (CFD) and simulate fluid flows, including those encountered in supersonic and hypersonic regimes. To this end, the compressible Navier-Stokes equations are solved using a finite

volume method (FVM). There are a few assumptions made in the model that are important to note. Since a uniform plasma model is used across the whole spanwise direction, it is possible to model this in 2D, saving computational resources. The main challenge with modeling the supersonic flows is the extreme temperatures and chemical reactions. This model ignores chemistry and radiation of the working fluid. However, considering the high temperatures, using a calorically perfect gas would be an inaccurate approximation. Thus, the model uses a thermally perfect gas model, which accounts for the temperature dependence of the specific heat ratio.

Another assumption is that the plasma here is modeled as an area of hot, low-density gas. Previous studies indicate that the plasma kernels' effect on the bulk flow does not depend on chemistry [101, 102]. Thus, the plasma chemistry has been neglected for this study.

3.2.1 Governing Equations

The simulations in this thesis are conducted using Direct Numerical Simulation (DNS) of the 2-Dimensional compressible Navier-Stokes equations. The governing equations are written in Cartesian conservative form [103]:

$$U_t + F(U)_x^a + G(U)_y^a = F(U)_x^d + G(U)_y^d \tag{3.1}$$

where U is the vector of conservative variables:

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \tag{3.2}$$

and the advective flux vectors are given by $F(U)^a$ and $G(U)^a$ for the x and y directions, respectively. The diffusive flux vectors are denoted by $F(U)^d$ and $G(U)^d$.

$$F(U)^a = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{bmatrix}, \qquad G(U)^a = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{bmatrix}, \tag{3.3}$$

$$F(U)^d = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} + q_x \end{bmatrix}, \qquad G(U)^d = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} + q_y \end{bmatrix}. \tag{3.4}$$

The terms in the equations represent the mass, momentum, and energy conservation laws, respectively. The variables in the vector U are the density ρ, the velocity components u and v, and the total energy E. The equations can also be written in cylindrical coordinates, but for the purposes of this thesis, the Cartesian form is used. If desired a source term can be added. This doubles as a way to change the coordinate system:

$$U_t + F(U)^a_r + G(U)^a_\theta = F(U)^d_r + G(U)^d_\theta + S(U) \tag{3.5}$$

The source term, $S(U)$,

$$S(U) = -\frac{\delta}{x} \begin{bmatrix} \rho u \\ \rho u^2 - \tau_{xx} \\ \rho uv - \tau_{xy} \\ u(E+p) - u\tau_{xx} - v\tau_{xy} + q_x \end{bmatrix} \tag{3.6}$$

If $\delta = 0$, it solves a Cartesian system. Setting $\delta = 1$ solves the equations in cylindrical coordinates, and $\delta = 2$ solves the equations in spherical coordinates. In this thesis the equations are solved in Cartesian coordinates.

3.2.2 Thermodynamic Relations

There are further equations necessary to fully describe the state of the fluid. The thermodynamic properties of the fluid, such as pressure, temperature, and density, are governed by the equation of state. For a compressible flow, the ideal gas law is typically used:

$$p = \rho R T, \tag{3.7}$$

where R represents the specific gas constant and T represents temperature. This relates the pressure and density to the temperature of the fluid. These relations are crucial in closing the system of equations and ensuring that the conservation laws are satisfied. The introduction of temperature into the system requires another relation to close the system of equations. The energy equation,

$$E = \rho \left(e + \frac{u^2 + v^2}{2} \right), \tag{3.8}$$

where e is the internal energy of the fluid, and the other term in the equation is the kinetic energy of the fluid. Introducing the dependence of temperature into (3.8) gives the internal energy in terms of enthalpy,

$$E = \rho \left(h + -\frac{p}{\rho} \frac{u^2 + v^2}{2} \right), \tag{3.9}$$

where h is the enthalpy of the fluid, defined as

$$h = h_o^f + \int_{T_{ref}}^{T} c_p(T) dT, \tag{3.10}$$

where h_o^f is the heat of formation, and the integral term is the sensible enthalpy. The specific heat, c_p is independent of temperature for a calorically perfect gas and the relations between energy, tem-

perature, density, and pressure are solved by substituting (3.7) and (3.10) into (3.9). However, for thermally perfect gases, the specific heat is a function of temperature, so the temperature is solved for iteratively. At each time step, the temperature is solved for using the implicit equation [104]:

$$T = \frac{-E + \rho\left(\frac{u^2+v^2}{2}\right) + \rho h(T)}{\rho R}. \tag{3.11}$$

The enthalpy is solved using the polynomials from the Shomate interpolation polynomials from the NIST database [105]. The algorithm used to solve this equation is the Newton-Raphson method, which is an iterative method that converges to the solution by updating the guess value based on the derivative of the function. The temperature is solved for at each cell in the domain, ensuring that the energy equation is satisfied and the thermodynamic properties are accurately represented. More information on the Newton-Raphson method and iteratively solving for the temperature can be found in Section A.

3.2.3 Transport Properties

To fully describe the mathematical model, it is essential to include the transport properties that capture the physical effects of viscosity. The viscous forces in the fluid are represented by the molecular stress tensor, Π_{ij}, which is referenced in (3.6). The viscous stress tensor Π_{ij} and the deformation rate tensor D_{ij} in Cartesian coordinates are given as follows:

$$\Pi_{ij} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix}, \quad D_{ij} = \begin{bmatrix} u_x & \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) & \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \\ \frac{1}{2}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right) & v_y & \frac{1}{2}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\ \frac{1}{2}\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right) & \frac{1}{2}\left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right) & w_z \end{bmatrix} \tag{3.12}$$

In the case of a Newtonian fluid, the stress tensor Π_{ij} is directly related to the rate of strain, which represents how the fluid deforms over time. Therefore, the stress tensor can be defined by:

$$\Pi_{ij} = 2\mu D_{ij} + \left(\mu_B + \frac{2}{3}\mu\right)(\nabla \cdot \mathbf{v}). \tag{3.13}$$

Here, μ is the dynamic viscosity, and μ_B is the bulk viscosity. The term involving $(\nabla \cdot \mathbf{v})$ in (3.13) accounts for changes in the volume of the fluid, ensuring compatibility with the conservation laws for compressible flow. The above equations can easily be reduced to 2D for the applications in this thesis.

In this thesis, the Sutherland's law is used to model the temperature dependence of viscosity, which is particularly important in high-temperature flows encountered in hypersonic applications. The Sutherland's law is given by:

$$\mu(T) = \mu_0 \left(\frac{T}{T_0}\right)^{3/2} \frac{T_0 + S}{T + S}, \tag{3.14}$$

where μ_0 is the reference viscosity, T_0 is the reference temperature, and S is the Sutherland's constant. This relation captures the non-linear temperature dependence of viscosity, which is essential for accurately modeling high-temperature flows.

The heat conductivity given in (3.4) is described in Fourier's Law:

$$q = -k\nabla T. \tag{3.15}$$

There are three contributions to the thermal conductivity: the electronic, vibrational, and translational contributions. However, this can be simplified by the specific heat and thermal conductivity of the fluid through Prandtl's number. The Prandtl number is the ratio of the momentum diffusivity to the thermal diffusivity, given by:

$$\mathrm{Pr} = \frac{\mu c_p}{k}, \tag{3.16}$$

for the simulations here, $\mathrm{Pr} = 0.7$.

3.3 Numerical Methods

When written in the conservative form, (3.1) can be separated into advective and diffusive terms, making solving the flux more straightforward. When the right-hand side (RHS) equals zero (i.e., no diffusive flux), the equations become Euler's equations.

The Navier-Stokes equations are solved by separating the advective and diffusive terms. The advective terms are solved using Roe's flux differencing scheme with slope limiters [106]. The diffusive terms are approximated using a second-order central difference scheme developed by Edwards [107]. Time integration is performed using a fourth-order Runge-Kutta method (RKIV) [108]. The equations are solved using dimensional splitting, a conventional method for handling multi-dimensional problems through a series of one-dimensional sweeps. Details on dimensional splitting are provided in the following section.

3.4 Dimensional Splitting and Flux Calculations

Dimensional splitting is a numerical technique used to reduce a multi-dimensional problem into simpler easier to solve one-dimensional problems. The process involves splitting the fluxes into their x- and y-components and solving the resulting equations sequentially. Toro [109] states that the 2D advection problem in the RHS of (3.1) can be solved through dimensional splitting as follows:

$$\text{x-split problem:} \quad \begin{cases} \text{PDE:} & U_t + F(U)_x^a = 0 \quad \xrightarrow{\Delta t} \quad U^{n+\frac{1}{2}} \\ \text{ICs:} & U^n \end{cases} \tag{3.17}$$

$$\text{y-split problem:} \quad \begin{cases} \text{PDE:} & U_t + G(U)_y^a = 0 \quad \xrightarrow{\Delta t} \quad U^{n+1} \\ \text{ICs:} & U^{n+\frac{1}{2}}. \end{cases} \tag{3.18}$$

The above equations are solved using the solution at timestep n, then it is advanced to $n+1$ by solving a 1D advection problem in the x-direction, followed by a 1D advection problem in the y-

direction. Each directional solve acts as a half step, where $U^{n+1/2}$, the result from the x-direction, is used to compute the y-split problem. The solution operators for the x-split and y-split problems are represented by $\mathcal{X}(\Delta t)$ and $\mathcal{Y}(\Delta t)$, respectively, then the solution at the next time step, U^{n+1}, can be expressed as [109, 110]:

$$U^{n+1} = \mathcal{Y}(\Delta t)\mathcal{X}(\Delta t)U^n. \tag{3.19}$$

The dimensional splitting approach shown in (3.19) reduces the complexity of the numerical process since it only requires developing a single algorithm for the 1-D advection problem, which is then applied once for each spatial direction.

A similar splitting strategy can be applied to an inhomogeneous system, called Strang splitting [111], such as the case with the source term in (3.5), where the right-hand side includes both diffusive fluxes and source terms:

$$\begin{aligned}
&\text{Advection problem:} \quad \begin{cases} \text{PDE:} \quad U_t + F(U)^a_x + G(U)^a_y = 0 \quad \xrightarrow{\Delta t} \quad U^{n+1}, \\ \text{ICs:} \quad U^n \end{cases} \\
&\text{Source problem:} \quad \begin{cases} \text{PDE:} \quad U_t = D(U) \quad \xrightarrow{\Delta t} \quad U^n, \\ \text{ICs:} \quad U^{n+1} \end{cases}
\end{aligned} \tag{3.20}$$

where $D(U) = F(U)^d_x + G(U)^d_y + S(U)$. The advection problem is initially solved using dimensional splitting (as in (3.17) and (3.18)), and the resulting solution is then used as the initial condition to solve the ODE for the source term, obtained from the discretization of both the diffusive flux and the source term $D(U)$.

In general, solving the system of conservation laws described in (3.5) involves addressing three separate problems: the advection PDEs in both spatial directions and the source ODEs. If we denote the solution operator for the source ODE by $\mathcal{D}(\Delta t)$, then the complete solution at time $n+1$ is given by:

$$U^{n+1} = \mathcal{Y}(\Delta t)\mathcal{X}(\Delta t)\mathcal{D}(\Delta t)U^n. \tag{3.21}$$

The dimensional splits in (3.17), (3.18) poses an initial boundary value problem (IBVP) for which a solution must be found. The method follows Toro's approach using Riemann solvers for nonlinear conservation laws [109].

The IBVP for the x-split 2-D problem can be formulated as:

$$\begin{aligned} &\text{PDE:} \quad U_t + F(U)_x = 0 \\ &\text{IC:} \quad U(x,0) = U_0(x) \\ &\text{BCs:} \quad U(0,t) = U_L(t), \quad U(x,t) = U_R(t), \end{aligned} \tag{3.22}$$

where U_L and U_R are the left and right conservative variables at the cell face, respectively. When the IBVP is discretized over a computational domain, this causes the resulting PDEs to be converted into ODEs, which are then integrated over time. Assuming a piecewise constant solution in each cell, (3.22) becomes:

$$\frac{d\bar{U}_i}{dt} = \frac{1}{\Delta x}\left(F_{i-\frac{1}{2}} - F_{i+\frac{1}{2}}\right), \tag{3.23}$$

where $\bar{U}_i$ is the average value of the conservative variables in cell i, and Δx represents the grid spacing. The term $\left(F_{i-\frac{1}{2}} - F_{i+\frac{1}{2}}\right)$ represents the difference in intercell flux across the interfaces. Note that these fluxes are not as straight forward in the cut cell method and will be described in section Section 3.6.1. According to Godunov [112], the intercell flux can be determined by converting the initial boundary value problem (IBVP) from (3.22) into a Riemann problem, which is solved at the interface of each cell:

$$U_t + F(U)_x = 0 \tag{3.24}$$

$$U(x,0) = \begin{cases} U_L & \text{if } x < 0 \\ U_R & \text{if } x > 0 \end{cases} \tag{3.25}$$

The flux is then given by:

$$F_{i+\frac{1}{2}} = F(U_{i+\frac{1}{2}}(0)), \tag{3.26}$$

where $U_{i+\frac{1}{2}}(0)$ is the solution of (3.25). At this stage, there are no numerical approximations. Godunov's method finds an exact analytical solution for the flux, assuming a piecewise constant distribution of the initial data. Although exact, this method is not computationally efficient.

Roe introduced an alternative approach [113], where the original Riemann problem (3.25) is replaced by an approximate linearized version. This method uses the Jacobian matrix of the system of conservation laws, $A(U) = \partial F/\partial U$, to reformulate the problem as:

$$U_t + A(U)U_x = 0 \tag{3.27}$$

$$U(x,0) = \begin{cases} U_L & \text{if } x < 0 \\ U_R & \text{if } x > 0 \end{cases} \tag{3.28}$$

Roe's approach replaces the exact Jacobian $A(U)$ with a linearized version, $\tilde{A}(U)$. If the system of equations is hyperbolic, the eigenvectors must be linearly independent. This allows for determining the intercell flux by using the eigensystem of the Jacobian. Then, according to (3.26), the intercell flux is obtained as:

$$F_{i+\frac{1}{2}} = F_L + \sum_{i=1}^{n} \tilde{\alpha}_i \tilde{\lambda}_i \tilde{K}^{(0)}, \tag{3.29}$$

where $\tilde{\alpha}_i$, $\tilde{\lambda}_i$, and $\tilde{K}^{(0)}$ are the coefficients for linear independence, the eigenvalues, and the eigenvectors of the Jacobian matrix $\tilde{A}(U)$. Physically, they represent the wave strengths, the speed at which information is propagating, and the characteristic directions over which information travels within the computational domain.

One major challenge with Roe's original method is that it requires an explicit knowledge of the linearized Jacobian matrix. To address this, the code in this thesis uses a variation based on Roe and Pikes [106], which avoids this issue by adding another linearization step. The approach assumes that, if the data U_L and U_R are sufficiently close to a reference state $\bar{U}$, then the Jacobian $\tilde{A}(U)$ can be approximated as $\hat{A}(\bar{U})$. This represents the second level of approximation of the original Jacobian $A(U)$. Wave strengths are determined by solving the Riemann problem (3.28) around the reference state:

$$\Delta U = U_R - U_L = \sum_{i=1}^{n} \hat{\alpha}_i \hat{K}^{(0)} \tag{3.30}$$

The solution vector $\hat{W}$ is determined by evaluating wave strengths at the average state and solving the resulting system of non-linear algebraic equations:

$$\hat{\alpha}_i = a_i(\hat{W}), \quad \hat{\lambda}_i = \lambda_i(\hat{W}), \quad \hat{K}^{(0)} = k^{(0)}(\hat{W}) \tag{3.31}$$

where:

$$\tilde{W} = \begin{bmatrix} \tilde{\rho} \\ \tilde{u} \\ \tilde{v} \\ \tilde{a} \end{bmatrix} \tag{3.32}$$

is the vector of primitive variables obtained from the conservative variables U. The solution vector $\hat{W}$ is used to determine the wave strengths, speeds, and directions, which are then used to calculate the intercell fluxes. For further information on Riemann solvers, the reader is referred to Toro, LeVeque and Roe [109, 110, 113] for detailed explanations on the Roe-Pikes method used in this thesis.

3.5 Time Marching Method

In this thesis, a fourth-order Runge-Kutta method (RKIV) is employed for time integration. This method is chosen for its balance between computational efficiency and the ability to accurately capture the transient behavior of high-speed flows. The method involves taking multiple stages within each time step, allowing for higher accuracy in the integration process.

Once the advection flux is computed, the system of governing PDEs is reduced to a system of ODEs in time as follows:

$$\begin{cases} U_t & = F(t, U(t)) \\ U(t_0) & = U_0. \end{cases} \tag{3.33}$$

The solution from time step n to $n+1$ is advanced in the using an explicit 4th-order Runge-Kutta method [114]:

$$\begin{cases} U^{(1)} = U^{(n)} - \frac{\Delta t}{\Delta x} F(U^{(n)}) \\ U^{(2)} = U^{(n)} - \frac{1}{2}\frac{\Delta t}{\Delta x} F(U^{(1)}) \\ U^{(3)} = U^{(n)} - \frac{1}{2}\frac{\Delta t}{\Delta x} F(U^{(2)}) \\ U^{(4)} = U^{(n)} - \frac{1}{6}\frac{\Delta t}{\Delta x} \left[F(U^{(n)}) + 2F(U^{(1)}) + 2F(U^{(2)}) + F(U^{(3)}) \right]. \end{cases} \tag{3.34}$$

When used with dimensional splitting, described in Section 3.4, the algorithm described in (3.34) is applied three times. Once for each complete time step: once for the x-split, once for the y-split,

and once for the source term, as outlined in (3.17), (3.17), and if needed (3.21). For the other two problems, $F(U)$ in Equation 3.34 is replaced by $G(U)$ and $D(U)$, respectively.

The choice of the time step is a critical step to ensure the numerical stability of the explicit ODE solver. Generally, the time step depends on three factors: (1) the mesh size, (2) the speed at which information propagates through the domain (i.e., wave speed), and (3) the intercell flux calculation method. These are all related through the Courant-Friedrichs-Lewy number. Mathematically, this can be represented for a 1-D system as:

$$\Delta t \leq \frac{\Delta x}{s_{\text{max}}}, \tag{3.35}$$

where Δx is the cell size in the x-direction, and s_{max} is the maximum wave speed in the computational domain. This wave speed must be calculated at each cell to determine the next time step. In practice, the time step is calculated using the CFL:

$$\Delta t = \text{CFL} \cdot \frac{\Delta x}{s_{\text{max}}}. \tag{3.36}$$

To satisfy (3.35), the CFL number must be less than one. For 2-D simulations, (3.36) is extended to:

$$\Delta t = \text{CFL} \cdot \min\left(\frac{\Delta x}{s^x_{\text{max}}}, \frac{\Delta y}{s^y_{\text{max}}}\right). \tag{3.37}$$

The wave speed for each cell is calculated as:

$$s^x_{\text{max}} = |u| + a, \quad s^y_{\text{max}} = |v| + a, \tag{3.38}$$

where a is the speed of sound, and u and v are the velocities in the x and y directions, respectively. Physically, (3.37) indicates that the time step size must be chosen based on the minimum wave speed in both directions.

3.6 Modeling Complex Geometry

Accurate representation of complex geometries is critical in CFD simulations, particularly when dealing with irregular surfaces that do not conform to a standard Cartesian grid. Several methods exist to handle complex geometries, including structured and unstructured grids, body-fitted coordinates, and immersed boundary methods.

In this work, we utilize the cut cell method, which accurately represents complex geometries on a Cartesian grid. This method simplifies grid generation while maintaining high accuracy near the boundaries.

It must be noted that the goal of implementing the cut cell method into the APDL-CFD code is to allow lower-resolution simulations for rapid prototyping. Without the use of cut cells, the flow field exhibits noticeable distortions, creating a 'wavy' appearance along the boundaries. This effect arises because the geometry, represented on a Cartesian grid, lacks smoothness and precise alignment with the physical surface. As seen in Figure 3.1, the wedge geometry is approximated by stair-step patterns inherent to the grid, which introduce artificial irregularities in the velocity field. These grid-induced artifacts interfere with the accurate capture of boundary conditions, resulting in deviations from the expected flow behavior.

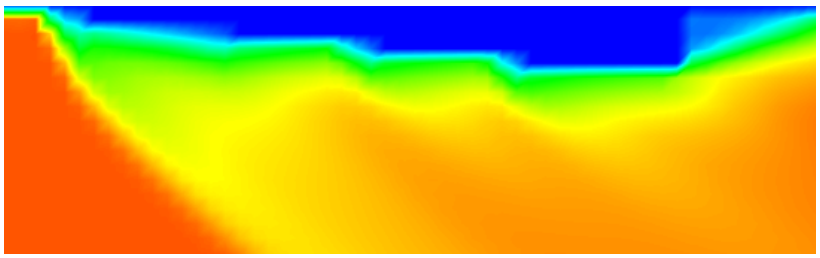

Figure 3.1: A low resolution wedge with no representation of a smooth geometry. Velocity magnitude is shown as an example.

Section 1.2 mentioned Peskin's use of immersed boundaries. These are different from the embedded boundary method used in this thesis. The difference is shown in Figure 3.2. The main takeaway from this is that the cell center in the embedded boundary is effectively moved, and the

quantities are solved at the new cell center of the fluid cell polygon. In IB, there are no cut cells; the geometry is represented by body forces and modifying the flow field via imposed boundary conditions.

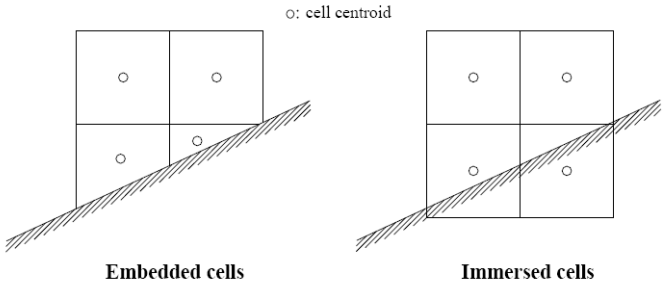

Figure 3.2: Embedded boundary (left) vs. Immersed boundary (right) [10].

3.6.1 Cut Cell Method

A cut cell method is employed to handle internal geometries that do not necessarily conform to a Cartesian grid. This approach first generates a structured Cartesian grid independently of the boundary geometry. The boundary, which is calculated separately in preprocessing, is then overlaid onto this grid, resulting in cells that are partially intersected by the boundary, known as cut cells. The geometry also fully covers cells, denoted as solid cells. The equations are still solved using the finite volume method on this structured grid, which remains Cartesian and regular everywhere except near the boundary where cut cells are present.

The cut cell method offers an alternative to body-fitted meshes, which can suffer from high aspect ratios and skewed cells near complex boundaries. However, cut cell methods introduce challenges, notably the small cell problem. The small cell problem is illustrated by the cell in Figure 3.3. The blue section is the solid, the gray is the fluid part of the cell. When a cell is mostly covered by the boundary, the remaining fluid portion can be very small, leading to numerical instability due to excessively small time steps required for stability. Many techniques to avoid this include cell merging [31] or flux redistribution [53]. For the sake of this thesis, cells below a user-defined threshold of either solid or fluid ratio are considered fully solid or fully fluid, respectively.

For example, the right cell shown in Figure 3.3 could be considered a solid cell since the cell is over 99% solid. This is just for visualization; the cut-off range is typically many orders or magnitude smaller. The small cell problem is more of a problem when the cell is primarily solid, but the same principle applies to cells that are mostly fluid.

Most cut cell methods are only first- or second-order accurate at the boundaries. However, according to Duan et al. [115], that is sufficient for high-speed aerodynamic flows. This only becomes a problem when dealing with highly turbulent flows.

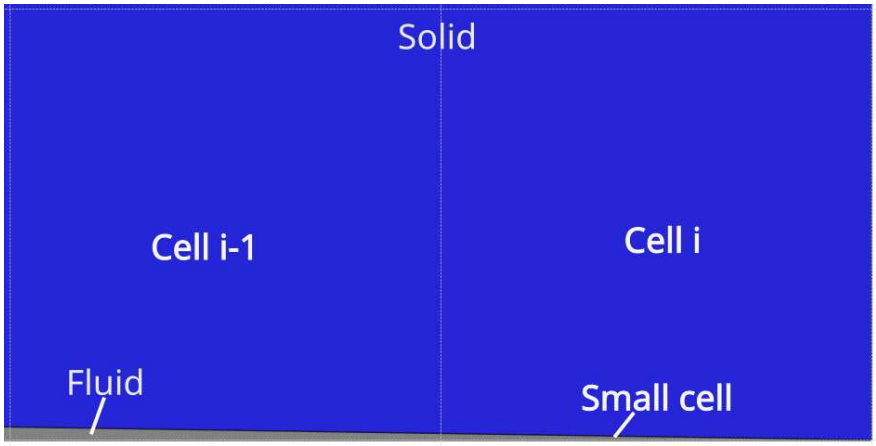

Figure 3.3: Two cut cells with the right having a small fluid portion, illustrating the small cell problem.

Cut cell methods have been extensively studied, with various approaches developed to handle the complexities introduced by the boundary geometry. The method described here is inspired by several existing techniques, notably the KBN method by Klein et al. [26] and its extension to more complex geometries by Gokhale et al. [52], who utilized methods from Meyer et al. [53] for boundary flux reconstruction. A distinguishing feature of these methods is the use of dimensional splitting, which simplifies the extension of the method to higher dimensions.

In the Cartesian cut cell method, there are three types of cells: fluid cells, cut cells, and solid cells. Fluid cells are regular grid cells not intersected by the boundary and can be solved normally. Solid cells lie entirely within the solid boundary and have no flux. In this case, solid cells are considered stationary, but if desired, creating a moving boundary requires special treatment of

these cells. Cut cells are partially cut by the boundary and require particular equations to calculate flux, which differ from those in the above sections. Accurately capturing the flow behavior near the boundary requires reconstructing the flow properties at the intersection points within the cut cells, including capturing the velocity gradient that occurs near a no-slip wall [31]. Figure 3.4 illustrates the different types of cells. The leftmost cell represents a fluid cell, the middle cell is a cut cell, and the rightmost cell is a solid cell.

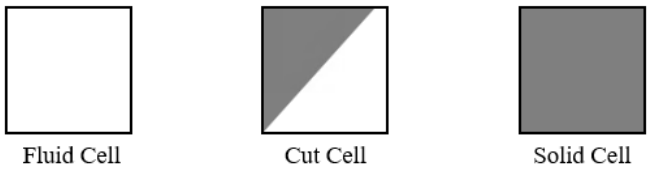

Figure 3.4: Three types of cells in the cut cell method: fluid cell (left), cut cell (middle), and solid cell (right).

The cut cells can be further divided into twelve different types based on the intersection of the boundary with the cell faces, as shown in Figure 3.5. These are divided to calculate the polygon points. The left column of cut cells has three polynomial points, the middle column has four, and the right column has five. The polynomial points are used to determine the cell fractions, which represent the proportion of the cell volume occupied by fluid (represented by the white region of the cells in the figure).

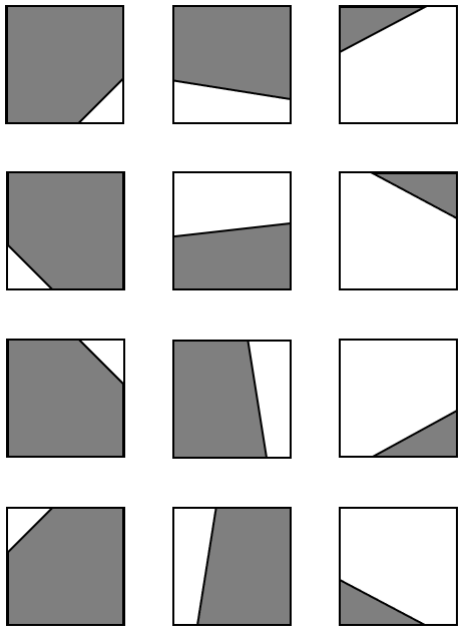

Figure 3.5: Different types of cut cells based on face intersection.

An advantage of the cut cell method is its computational efficiency. All geometric parameters are computed during preprocessing, which can be significantly faster than generating an unstructured mesh for complex geometries. The geometric parameters are listed in the order that they are calculated during preprocessing:

1. The cell intersection points, where the boundary intersects the cell faces.
2. The face fraction, β of each face in the cell. This represents the proportion of the face area that is occupied by fluid.
3. The length of the intersecting line, and the coordinates of the midpoint of this line x_m.
4. The unit normal to the boundary for each cell, $\hat{n}^b$.
5. The centroids of the fluid portion of the cut cells, denoted as x_c.
6. The area of the fluid part of the cell, A_f.
7. The area fraction, α. This is the fluid area, A_f, non-dimensionalized by the total cell area.

The coordinates of the intersections are simply identified as the first step. Next, the face fractions are calculated by β by taking the distance between the coordinate of the corner of the face in the fluid section and the coordinate of the cell intersection. This is done for each face, giving $\beta_{i-1/2}, \beta_{i+1/2}, \beta_{j-1/2}, \beta_{j+1/2}$. A simple example of this is shown in Figure 3.6. To ensure this is calculating the fluid part of the cell, a sign distance function can be used to determine if the point is inside or outside the boundary,

$$\frac{AI}{AB} = \beta = -\left(\frac{\phi_A}{\phi_B - \phi_A}\right), \tag{3.39}$$

where ϕ_A and ϕ_B are the sign distance function at points A and B, respectively.

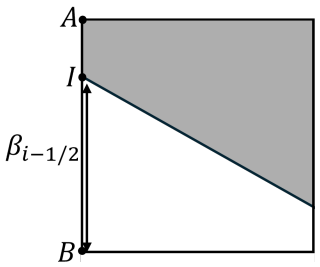

Figure 3.6: Face fractions β for a cut cell.

In 3D, this becomes a face calculation, and uses the same formula to calculate α. The alpha value is then computed by using the formula for the area of a polygon. Before the alpha is computed the polygon points are sorted in order based on the orientation that the boundary intersects the cell. Once sorted, the α is calculated using the formula:

$$\alpha = \frac{1}{2A_b}\sum_{i=1}^{n}(x_i y_{i+1} - x_{i+1} y_i), \tag{3.40}$$

where the sum term is the formula to calculate the area for an arbitrary polygon and is periodic so $n + 1 = 1$. This can also be expanded to 3D, the calculation would be for a polyhedron. For the sake of this thesis, the focus is on 2D simulations.

In the following, we describe the calculations used in the cut cell method, first in one dimension (1D), then extending to two dimensions (2D). Since the method employs dimensional splitting, it can be straightforwardly expanded to three dimensions (3D), although our focus here is on 2D simulations. Klein et al. [26] introduced a reference state at the wall, denoted as $u^0 = (\rho^0, u^0, v^0, p^0)$, representing the conservative variables at the wall. However, in this investigation, the walls are stationary, and we consider them as reflective boundaries zero velocity, so the reference states are omitted from these calculations.

For a 1D cut cell, the standard explicit update equation in finite volume methods is given by, described in more detail in Section 3.4:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+1/2}^n - F_{i-1/2}^n\right), \tag{3.41}$$

where U_i^n is the conservative variable vector at cell i at time level n, Δt is the time step, Δx is the cell width, and $F_{i\pm1/2}^n$ are the numerical fluxes at the cell interfaces.

However, near a cut cell, this equation changes to account for the reduced fluid volume within the cell:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\alpha_i \Delta x}\left(F_{i-1/2}^n - F^{b,n}\right), \tag{3.42}$$

where F^b is the flux at the boundary, and $F_{i-1/2}^n$ is the standard left flux. However, the use of $F_{i-1/2}^n$ is not stable, according to Klein et al. [26] the actual conservative update is:

$$\hat{U}_i^{n+1} = \hat{U}_i^n + \frac{\Delta t}{\alpha \Delta x}\left(F_{i-1/2}^{KBN,n} - F_{i-1/2}^n\right), \tag{3.43}$$

where $F^{KBN,n}$ is the stabilized flux computed at time step n to account for the presence of cut cells, ensuring numerical stability.

To determine the expression for the stabilized flux, we equate the right-hand sides of (3.43) and (3.42), yielding:

$$F_{i-1/2}^{KBN,n} = F^{b,n} + \alpha \left(F_{i-1/2}^{n} - F^{b,n} \right). \tag{3.44}$$

This stabilized flux is used to update both the cut cell and its neighboring cells, ensuring conservation within the scheme at each regular time step Δt. Note that (3.44) is derived from [26] and behaves consistently in the natural limits of the grid. When $\alpha \to 1$, the stabilized flux $F_{i-1/2}^{KBN,n}$ approaches the regular flux $F_{i-1/2}^{n}$. Conversely, as $\alpha \to 0$, $F_{i-1/2}^{KBN,n}$ converges to the boundary flux $F^{b,n}$. The KBN flux was further stabilized by Gokhale et al. [52] by using the wavespeed and more geometric information. The first improvement is to use a new local timestep for the cut cells. Since the timestep is based off the cell size and the wave speed, with cut cells a smaller timestep is needed:

$$\Delta t_{\text{cc}} = \text{CFL} \frac{\alpha \Delta x}{S_i}, \tag{3.45}$$

where S_i is the wave speed for the cut cell. With this modification, we replace (3.44) with:

The modified form of the stabilized flux is given by:

$$F_{i-1/2}^{KBN,n} = F^{b,n} + \frac{\Delta t}{\Delta t_{cc}} \left(F_{i-1/2}^{n} - F^{b,n} \right), \tag{3.46}$$

where the local timestep Δt_{cc} is used to stabilize the flux in the cut cell. This modification ensures that the fluxes are consistent with the local wave speed, preventing numerical instability in the presence of cut cells. This can then be used to further stabilize the cut cell flux with the Locally Preserving Flux Splitting (LPFS) method is given by:

$$F_{i-1/2}^{\text{LPFS}} = \frac{\Delta t_{\text{cc}}}{\Delta t} F_{i-1/2}^{n} + \left(1 - \frac{\Delta t_{\text{cc}}}{\Delta t} \right) F_{i-1/2}^{\text{KBN}}, \tag{3.47}$$

This formula blends the standard flux $F_{i-1/2}^{n}$ with the KBN-stabilized flux from (3.46) to maintain stability.

This is all done in 1D, but Klein et al. give the extension in 2D. This requires additional parameters, some of these parameters are represented graphically in Figure 3.7. The left figure shows

the geometric parameters used and the right shows the fluxes. It is clear from this figure that the parameters are split into two main regions at the interface between two cells:

1. The unshielded region, the portion of the cell that is not intersected by the boundary.

2. A singly shielded portion, where the region is shielded by the boundary to the right (SS,R) or left (SS,L). In this case there is only SS,R.

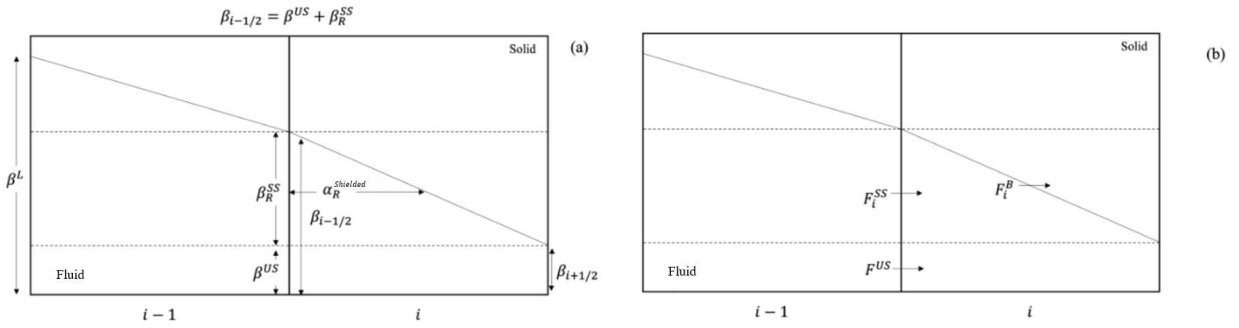

Figure 3.7: Geometric parameters for cut cell calculations. (a) Geometric parameters for cut cell calculations. (b) Fluxes in a cut cell.

The following explains the fluxes in Figure 3.7:

- For the unshielded region, the standard 1D flux (3.26) is used for F^{US} and weighted in (3.52).

- For the singly shielded regions, the flux is computed using the LPFS flux, from (3.47). These are represented as either $F^{SS,R}$ or $F^{SS,L}$, depending on the direction of the shield. α is also replaced with α_{shielded} in (3.45), which represents the distance between the center of the boundary x_m and the edge of the cell.

$$\alpha^{\text{shielded}} = \frac{\hat{\alpha}_{i-1/2,j}}{\beta_{i-1/2,j} - \beta_{i+1/2,j}}, \tag{3.48}$$

  where $\hat{\alpha}_{i-1/2,j} = \alpha_{i,j} - \beta_{i+1/2,j}$.

- The boundary flux, F^b, requires special treatment. These fluxes are done in a similar way to Meyer et al. [53]. Since the pressure and density are reflected from the boundary, it is

taken as a simple interpolation from the surrounding fluid cells. However, due to the no-slip conditions on the wall, special consideration must be taken for the velocity. This is done at an interpolation point located at a distance, P', from the surface midpoint x_m along the normal coordinate n. To determine the location of P', the function h is used:

$$h = \frac{1}{2}\sqrt{(\Delta x \cdot n_{\perp,x})^2 + (\Delta y \cdot n_{\perp,y})^2}, \tag{3.49}$$

where $n_{\perp,x}$ and $n_{\perp,y}$ are the components of the normal vector perpendicular to the boundary in the x and y directions, respectively. Then to reconstruct this value at the centroid of the cut cell, the distance from the boundary midpoint to the centroid is found:

$$\Delta h = \sqrt{(x_{c,x} - x_{m,x})^2 + (x_{c,y} - x_{m,y})^2}. \tag{3.50}$$

This value is then used to calculate the velocity at the cut cell centroid,

$$u_{\text{cut}} = \frac{\Delta h u_{P'}}{h}. \tag{3.51}$$

This takes the interpolated velocity at P', $u_{P'}$ and puts the linear interpolation of it onto the centroid of the cut cell, x_c. These velocities are then used to compute the flux at the boundary. Figure 3.8 illustrates the velocity gradient graphically.

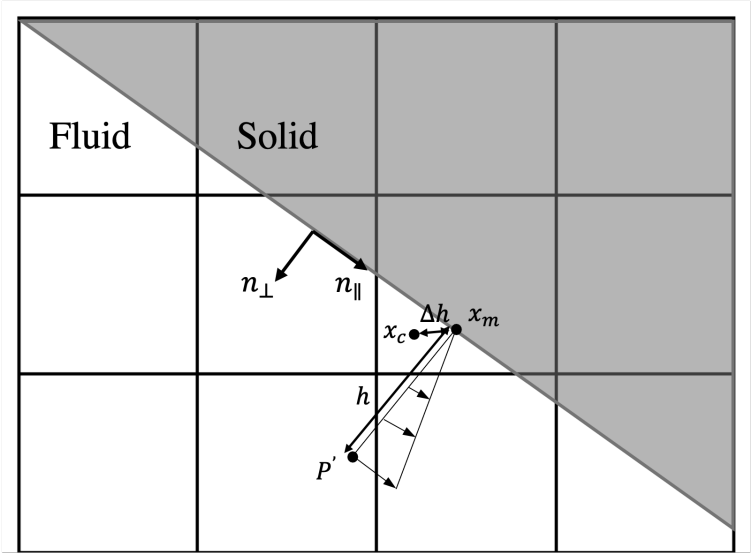

Figure 3.8: Velocity gradient near the wall. The velocity at the boundary is interpolated to the centroid of the cut cell.

Finally, the fluxes are all put into one equation to get the modified flux at the cut cell. This is done by taking a weighted average of the fluxes from the different sections, depending on the region of the cell, illustrated in Figure 3.7. The modified flux is:

$$F^{\text{modified},n} = \frac{1}{\beta_C}\left[\beta^{\text{US}} F^{\text{US},n} + \beta^{\text{SS,L}} F^{\text{SS,L},n} + \beta^{\text{SS,R}} F^{\text{SS,R},n}\right], \tag{3.52}$$

where the β coefficients are face fractions, and the superscripts refer to different flux contributions from various parts of the cell (e.g., unshielded fluxes, shielded-left fluxes, shielded-right fluxes.). For the case with the wedge we only have singly shielded to the right, so $F^{\text{SS,L},n} = 0$.

Then, (3.52) is used to update the cut cell in the x-direction, and then the y-direction. The final update equation for the cut cell is given by:

$$U_{i,j}^{n+1/2} = U_{i,j}^{n} + \frac{\Delta t}{\alpha \Delta x}\left[\beta_{i-1/2} F_{i-1/2}^{\text{modified},n} - \beta_{i+1/2} F_{i+1/2}^{\text{modified},n} - (\beta_{i-1/2} - \beta_{i+1/2}) F_i^{\text{b},n}\right], \tag{3.53}$$

$$U_{i,j}^{n+1} = U_{i,j}^{n+1/2} + \frac{\Delta t}{\alpha \Delta y} \left[\beta_{j-1/2} G_{j-1/2}^{\text{modified},n} - \beta_{j+1/2} G_{j+1/2}^{\text{modified},n} - (\beta_{j-1/2} - \beta_{j+1/2}) G_j^{\text{b},n} \right], \quad (3.54)$$

where $U_{i,j}^{n+1/2}$ is the intermediate state after the first dimensional update, and $U_{i,j}^{n+1}$ is the updated state after completing both dimensions.

In summary, the cut cell method provides an efficient and accurate way to handle complex geometries in computational fluid dynamics simulations without needing body-fitted meshes. Reconstructing the flow variables in cut cells and employing flux stabilization techniques allows for accurate geometry representation and stable numerical simulations. It should be noted that in the case of concave boundaries, meaning where the face of a cell might be shielded on both sides, the method becomes more complex. This requires splitting the shielded region into singly and double shielded regions and treating them separately. However, only convex boundaries are considered in this case, so this is not a concern.

3.6.2 Algorithm for Grid Generation

The grid generation is all done in python before the simulation is run as a preprocessing step. The grid is generated using the following steps:

Algorithm 1: Cut Cell Grid Generation Procedure

1 Generate a uniform grid over the computational domain;
2 Define the embedded geometry through a series of lines;
3 Find the intersection points of the grid lines with the embedded geometry;
4 Store all cell corners and cell intersections;
5 **foreach** *cut cell* **do**
6 Calculate face fractions $\beta_{i-1/2}$, $\beta_{i+1/2}$, $\beta_{j-1/2}$, $\beta_{j+1/2}$ by finding the distance between the cell corners and the intersecting line on the cell face;
7 Calculate additional face fractions $\beta^{SS,L}$, $\beta^{SS,R}$, $\beta^{SS,U}$, $\beta^{SS,D}$ for each cut cell;
8 Sort cell corners and intersections and determine the intersection type based on Figure 3.5, including checking the normal to determine the direction of the intersecting line;
9 Calculate the cell area, A_f for each cut cell;
10 Determine the normals of each cut cell and calculate polynomial points based on the cut cell type;
11 Calculate volume fractions α, $\alpha^{SS,R}$, $\alpha^{SS,L}$, $\alpha^{SS,U}$, $\alpha^{SS,D}$ for each cut cell;
12 **end**
13 Output all calculated parameters: β's, α's, normals, midpoints, and cut cell centroids to the CFD code;

3.6.3 Modeling of Plasma

The plasma is represented as a hot kernel of low-density, high-temperature gas. In the simulations presented in this thesis the first impact of the incident shock wave is on the bottom wall. Thus, the plasma is placed upstream of the first impact location. The location and temperature are varied to determine the effects. The plasma chemistry has been neglected for this study, following

recommendations of previous studies [101, 102]. Further details on the modeling of plasma are presented in Chapter 4.

3.6.4 Code flowchart

The following flowchart is a simplified version showing the flow of information. There are many more steps in the code, but this gives a general idea of how the code works. Starting with preprocessing in Python for geometry generation, the information is then passed to the CFD code. The CFD code then runs the simulation and outputs the results. The results are then post-processed in Python or VisIt to generate the figures and data for analysis. The orange boxes represent the main loop in the code. The validation of this code can be found in Chapter 5.

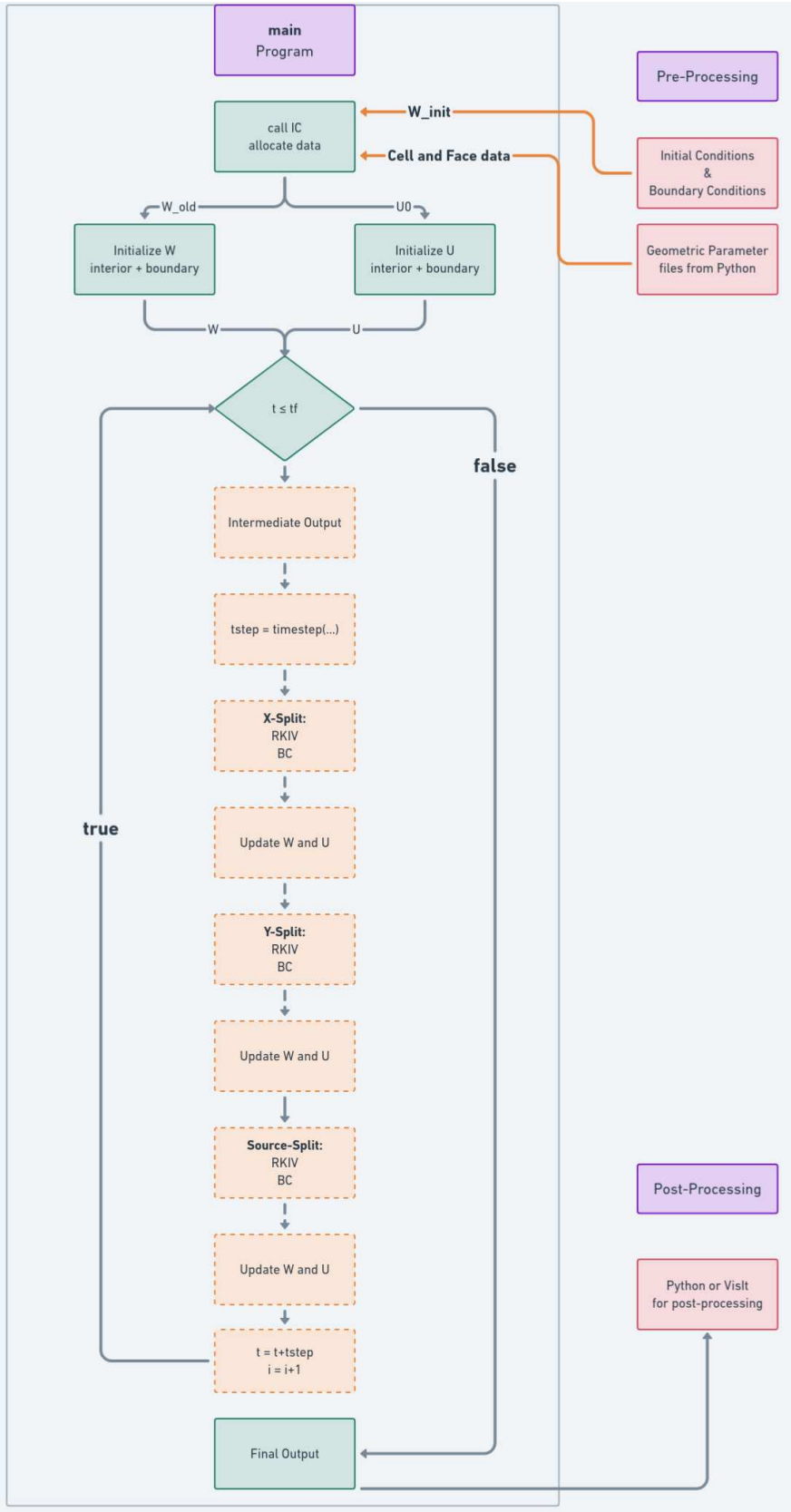

Figure 3.9: Flowchart of the entire code.

Chapter 4

Modeling of Plasma-Assisted Shock Wave Control

The following chapter will be primarily from results published in a AIAA conference paper [116] and a journal paper that is pending submission.

4.1 Introduction

This chapter investigates the use of plasma-based flow control for hypersonic applications, focusing on the impact of plasma on shock wave (SW) dynamics in a supersonic channel at Mach 2.5. Numerical simulations were performed in a straight channel with a square cross-section (10x10 cm) and a length of 50 cm, using the in-house APDL-CFD code. An oblique SW train was generated by a 10° wedge mounted on the upper wall, 5 cm downstream of the channel inlet. A simulated femto-filament laser discharge was introduced just upstream of the point where the leading shock impinges on the bottom wall.

The study explores the effects of plasma energy deposition on SW-boundary layer interactions (SWBLI), with plasma temperature varying from 1,000 to 8,000 K and its position adjusted to maximize shock displacement. The simulations reveal an upstream shift of the reflected SW by about 5 cm at the upper wall impingement point, resulting from the interaction between the plasma, boundary layer (BL), and SW. Plasma generation led to significant thickening of the bottom wall BL, causing a larger separation bubble that weakened the reflected shock. Consequently, two weaker shocks were produced at different angles compared to the original reflected shock.

Synthetic schlieren imaging and pressure measurements showcase the effects of plasma on the SW pattern. The CFD results align with experimental trends from the University of Notre Dame's SBR-50 supersonic wind tunnel, suggesting that plasma-induced SWBLI could lead to novel plasma-based shock control methods for applications such as scramjet inlets and isolators.

4.2 Problem Setup

This section will introduce the experimental and computational setup for the study.

4.2.1 Modeling the CSU Wind Tunnel

The primary goal of this work is to inform experiments that will be completed in the CSU supersonic wind tunnel, developed by Teeter et al. [25]. A figure APDL hypersonic wind tunnel is shown in Figure 4.1; the computational domain is based on the test section shown. The tunnel operates as an indraft facility whereby flow acceleration is obtained by generating a pressure differential between the atmospheric inlet air and a vacuum tank placed at the outlet. Flow conditioning is attained using an asymmetric linear converging-diverging nozzle, which ensures a uniform $Ma = 2.5$ inside the test section. Work is underway to convert the tunnel into a heated blowdown facility. More information regarding the facility can be found in Teeter et al. [25].

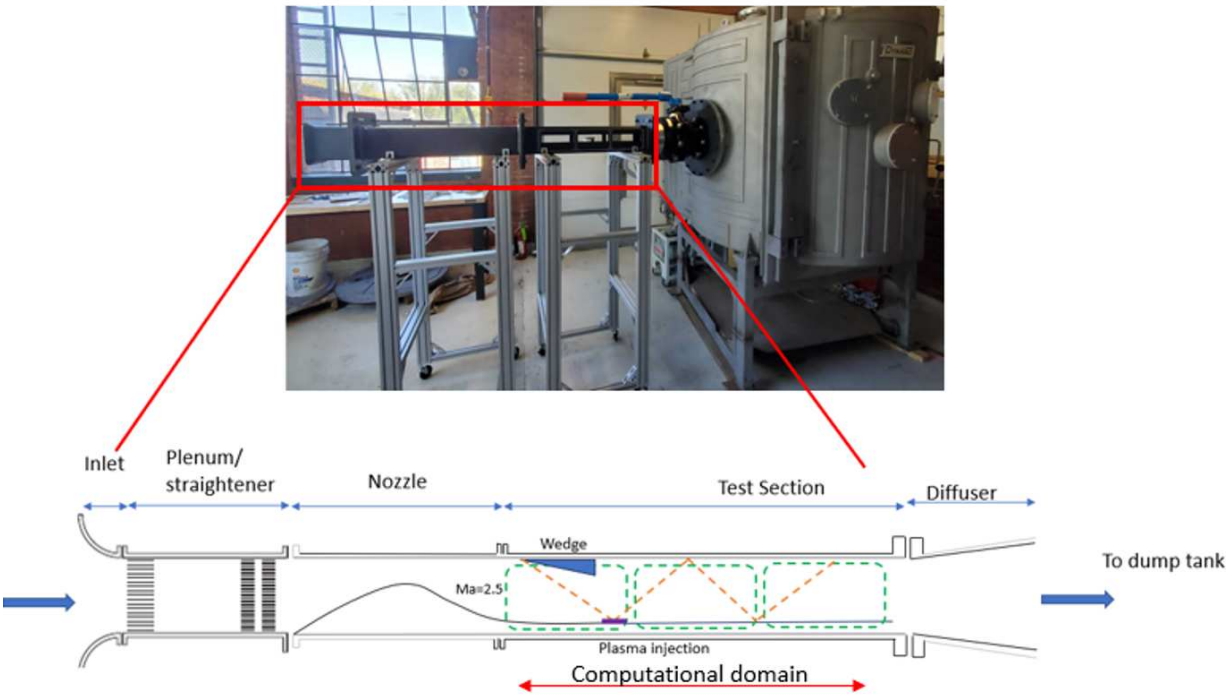

Figure 4.1: CSU Wind Tunnel photo and schematic. The wind tunnel is an indraft design with a vacuum chamber downstream from the test section and atmospheric conditions at the inlet. Significant improvements such as the addition of a valve system have been made to the wind tunnel since the time of the photo.

A summary of the specifications are given in Table 4.1

Table 4.1: Colorado State University APDL Wind Tunnel Specifications.

Test Section Size	(3–12.7) $\times$ 12.7 in
Vacuum Tank	$V = 3$ m^3; $P_0 = 0.01$ mbar
Pump System	4-stage: 60 m^3/hr
Total Pressure	$P_0 = 0.1$ MPa
Total Temperature	$T_0 = 300 - 2000$ K
Flow Rate	0.41–1.2 kg/s
Mach Number	2.5–7
Test Time	0.1–0.5 sec

4.2.2 Mathematical Model

The hypersonic flow is simulated using our direct numerical simulation (DNS) custom-built APDL-CFD code. The governing equations, written in Cartesian coordinates are the same as those in Chapter 3.

4.2.3 Computational Domain

For the study conducted here, the computational domain covers most of the test section length of the tunnel. The computational space is represented in 2-D with a 0.5 m $\times$ 0.1 m domain as shown in Figure 4.2. A supersonic inlet boundary condition specifies the left side of the domain. The inlet conditions were determined using a 1-D isentropic calculation by imposing a desired free-stream Ma=2.5 in the channel and a stagnation condition corresponding to 1 atm and 650 K. Starting at this elevated temperature reflects the high-energy conditions the flow would encounter at the entrance of a scramjet isolator at such speeds. The supersonic outlet is implemented using the non-reflective boundary treatment defined by Poinsot et al. [117], which allows the evacuation of the supersonic flow with minimal reflections back into the domain. A 10° triangular wedge is embedded inside the domain, which starts at $x = 5$ cm. The walls and the wedge are modeled with no-slip conditions.

The location of the plasma was varied between 2.5–20 cm along the bottom wall of the tunnel to better understand the impact of plasma placement on the shock wave train displacement. The plasma temperature was varied between 1000–8000 K to study the effect of the energized boundary layer on the SWBLI. The chosen temperature range represents what has been previously reported in a dual-pulse laser plasma heating technique. The plasma size and temperature were selected based on the optical emission data obtained by Houpt [99].

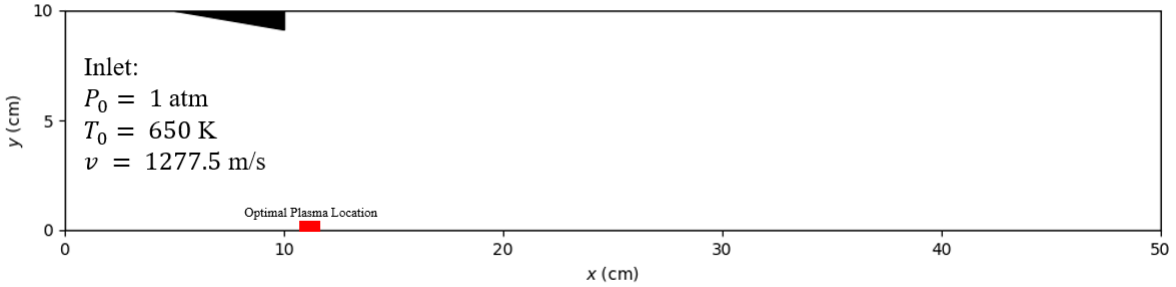

Figure 4.2: Computational domain with plasma location shown by the red block. Note that while the wedge size is to scale, the plasma location is enlarged for visibility; in reality, it measures 2.5 $\times$ 2.5 mm and the actual position is varied.

Flow interaction strongly depends on the spatial distribution of discharge energy, primarily controlled by the electrode geometry and plasma-flow coupling. In this case, the electrodes are embedded into a ceramic material on the bottom of the wind tunnel as shown previously by Leonov [8] so the discharge electrodes do not physically interfere with the computational domain. Regarding the plasma kernel, it is important to note that this study does not aim to model filament dynamics or plasma kinetics in the study. Rather, this study assumes plasma exerts a purely thermal effect on the flow. The plasma model used for the study is illustrated in Figure 4.3. The plasma is created by overlapping dual-pulse femtosecond and nanosecond lasers, creating a spanwise uniform plasma arc across the whole domain and creating accurate 2D effects.

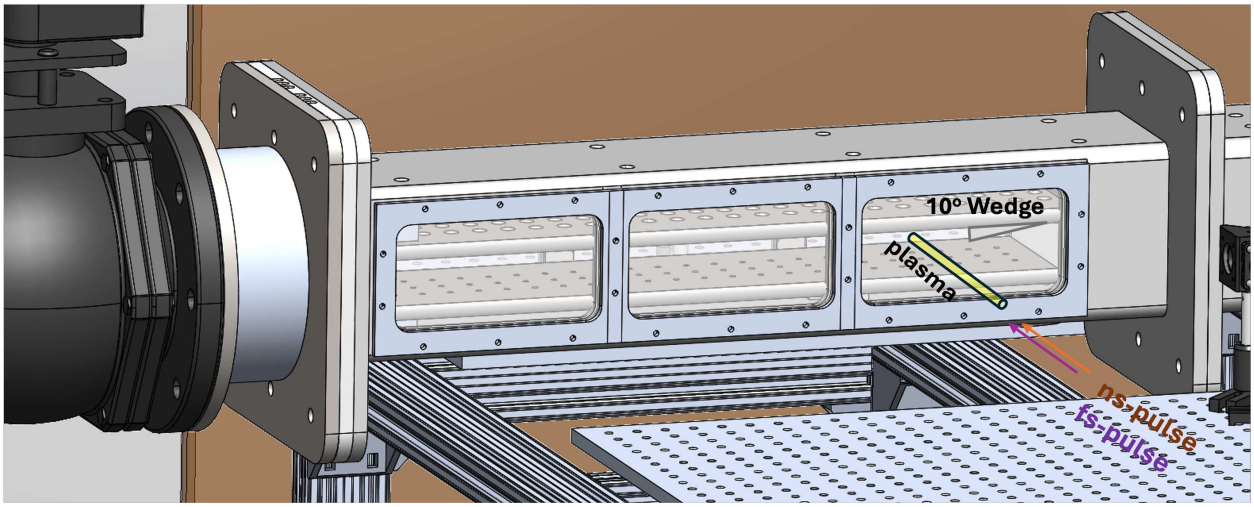

Figure 4.3: Experimental setup to create a linear spanwise plasma. This can be solved computationally by using a 2D model.

4.3 Numerical Results and Discussion

The upcoming section will detail the findings on controlling shock wave trains using a simulated femto-filament plasma discharge. It begins with an explanation of the underlying physics of shock wave train control and the interaction between shock waves and boundary layers (SWBLI), including a thorough description of the separation bubble. Following this, the study will explore the effects of varying plasma locations and follow with an analysis of the impact of different plasma temperatures.

4.3.1 Control of Shock Wave Position

Results showing the oblique shock wave train induced by the triangular wedge are presented Figure 4.4 via synthetic schlieren, modeled here simply as $\partial\rho/\partial y$, since every property is assumed constant in the spanwise direction. Shown in Figure 4.4-top is the steady-state solution in the absence of the femto-filament plasma discharge. The 10° wedge starts at $x = 5$ cm along the top wall. The first incident shock impacts the bottom wall at $x = 21$ cm, and the reflected shock creates a second impact on the top wall at $x = 38$ cm. The position of this second impact point, and how it is altered by plasma injection, is the primary focus of our analysis. Once the supersonic flow fully develops in the computational domain and the steady shock wave train is established ($t = 1.3$

ms after the start of the simulation), the plasma is turned on. As seen in Figure 4.4-bottom, a plasma kernel is placed upstream of the first impact point at $x = 12.5$ cm, and the effect is seen by the second impact position by moving the shock wave train by about 5 cm upstream in this case. Additionally, we observe a growth in the separation bubble in the plasma-energized boundary layer. Adding the plasma thickens the boundary layer. The plasma energy deposition leads to a local drop in density, which in turn leads to an acceleration of the fluid in the boundary layer and an increase in boundary layer thickness. The larger boundary layer thickness subsequently leads to the formation of a larger separation bubble. This in turn will affect how the reflected shock propagates. The plasma is stretched downstream due to the no-slip conditions on the bottom wall and the advection of the bulk flow over the kernel. The separation bubble plays an important role in shifting the second impact point on the top wall.

These results improve our previously published studies by reducing the wedge size in the channel, thereby minimizing the effects of the low-momentum wake generated at the rear of the wedge. This helps lead to a more isolated analysis of the shock wave interaction with the plasma.

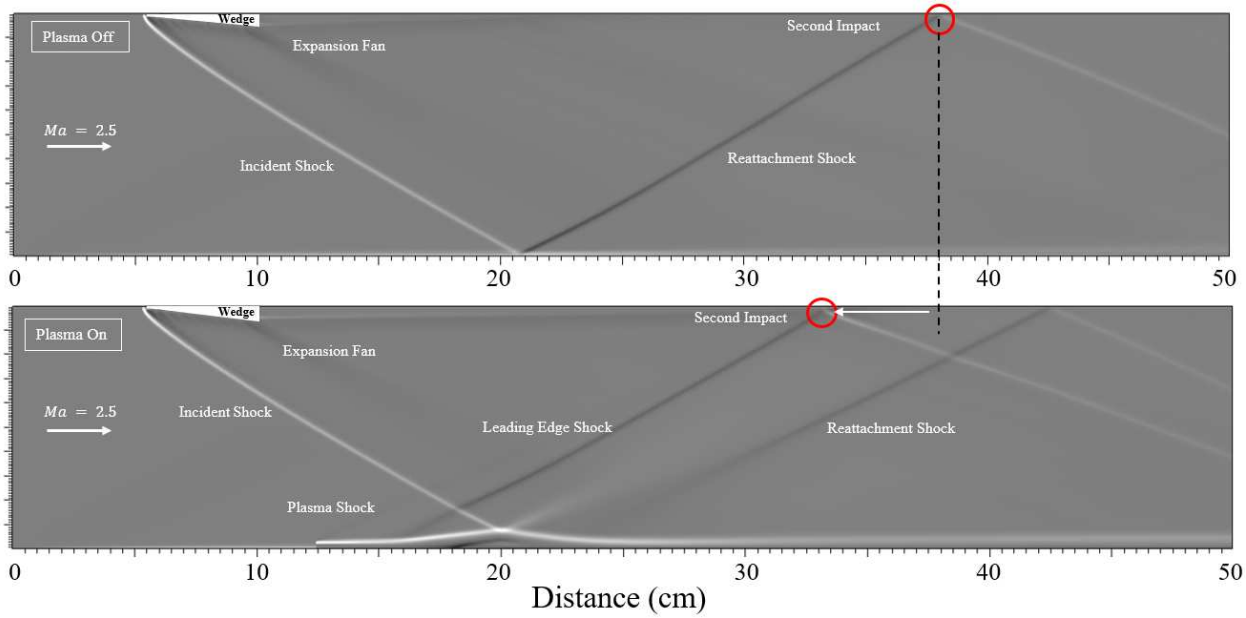

Figure 4.4: Numerical schlieren plot ($\partial\rho/\partial y$) of the SW train. Top: Plasma is turned off. Bottom: Plasma on (2.5×2.5 mm initial size,T_{plasma}=5,000K) is turned on. The image set showcases the upstream displacement of the SW train when the plasma is turned on. The position of the plasma actuator is at $x = 12.5$ cm.

The flow physics of shock wave boundary layer interaction (SWBLI) in a $Ma = 2.5$ flow without plasma are illustrated in Figure 4.5. In this configuration, an incident shock generated by an upstream wedge impinges on the boundary layer along the bottom wall, leading to the formation of a separation bubble. In an ideal scenario, the incident shock will impinge on the bottom wall causing a perfect reflection due to the impedance mismatch at the interface. However, in reality, the interaction between the shock and the fluid near the wall leads to an adverse pressure gradient which causes a local flow separation at the impact point. The separation shock disturbs the ideal reflection condition, leading to the shift upstream of the reflected shock. This phenomenon is why in Figure 4.5 below we see that the incident and leading edge shocks cross each other even in the absence of a plasma. As the separated flow interacts with the shock, a leading edge shock (first reflected shock) is formed at the separation point. Downstream of the bubble, the flow accelerates through an expansion fan before being compressed, generating a reattachment shock. This complex shock structure and the associated separated flow significantly influence the downstream pressure distribution.

When plasma actuation is applied, the flow structure undergoes a marked transformation. The separation bubble grows in size and shifts slightly upstream due to the plasma-induced body forces within the boundary layer. This displacement causes the leading edge shock to move upstream as well, resulting in a larger distance between the leading edge and reattachment shocks and the second impact to be moved upstream. In this configuration, the reattachment shock is also displaced upstream. Notably, with plasma actuation, the separation bubble height increases to approximately 8 mm, compared to 5.3 mm without plasma. The presence of plasma modifies the boundary layer and shock dynamics, resulting in a different shock structure. The separation and relative strength of these two shock waves depend on various factors, including plasma temperature, plasma size, and the plasma kernel's position relative to the first impact point.

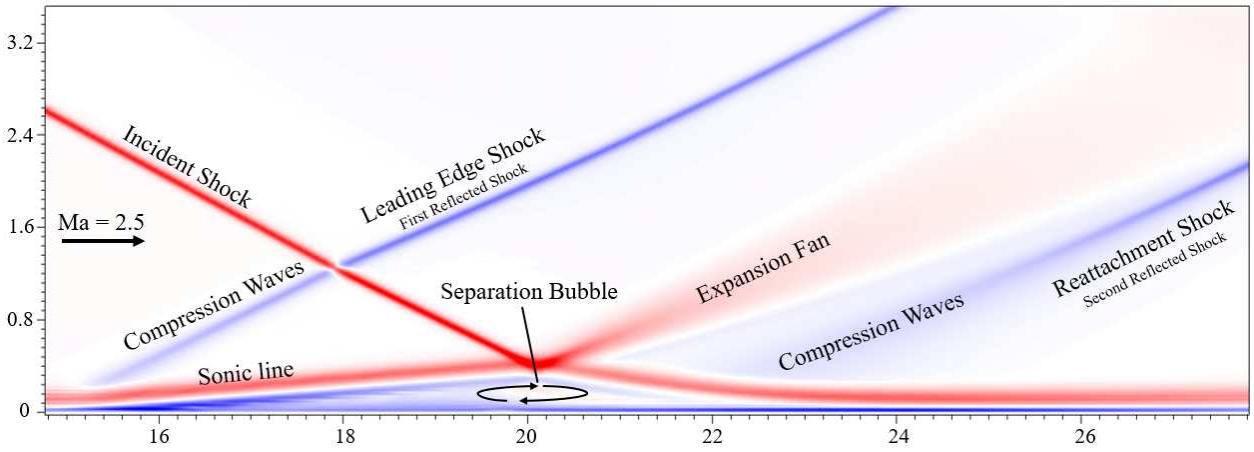

Figure 4.5: Numerical schlieren plot ($\partial\rho/\partial y$) zoomed in on the separation bubble physics shown with plasma off.

The impact of plasma on the SWBLI can be seen in the schlieren images of Figure 4.6. In the absence of plasma (top image), the boundary layer remains relatively thin, leading to a smaller separation bubble and a later reattachment shock. However, after plasma actuation (bottom image), the boundary layer thickens significantly, and the separation bubble expands upstream. The interaction between the incident shock and the plasma-activated boundary layer generates a plasma-induced shock, which emerges due to the flow's response to the low-density plasma region. Additionally, a secondary reattachment shock forms just downstream of the separation bubble, further altering the flow structure. Figure 4.6 also shows the growth of the separation bubble height, which increases from 5.3 mm to 8 mm after plasma actuation. This growth is due to the plasma's thermal effects on the boundary layer, which cause the flow to separate further upstream. The plasma-induced shock and the secondary reattachment shock are also visible in the bottom image, demonstrating the significant impact of plasma on the SWBLI.

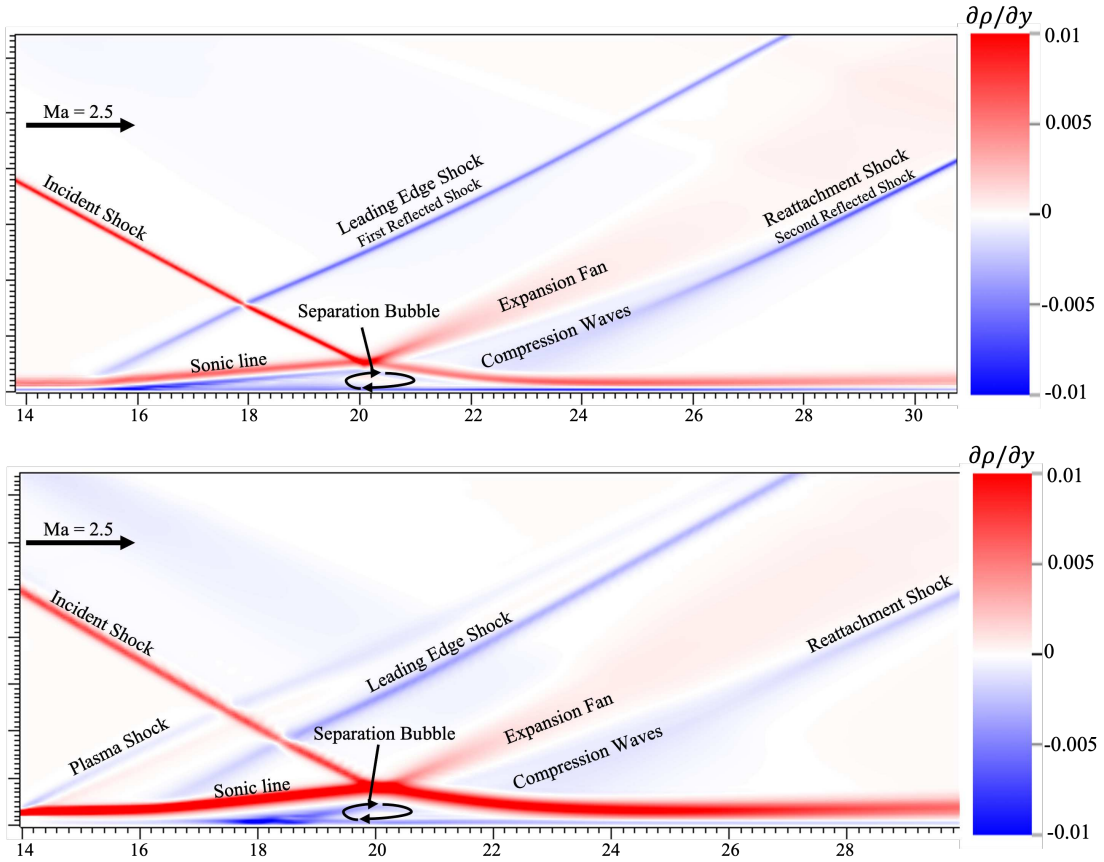

Figure 4.6: Numerical schlieren images describing the shock-boundary layer interaction at the moment of plasma actuation. Top: SWBLI with plasma off. Bottom: Shock-plasma-boundary layer interaction a few hundred microseconds after plasma actuation.

4.3.2 Plasma Location Effects

To understand the optimal plasma location for increasing the upstream displacement of the reflected shock wave's second impact point, the plasma kernel's position along the bottom wall was varied in a series of numerical simulations. Figure 4.7-Left presents the pressure traces along the top wall for five cases: one with no plasma and four with plasma placed at distances ranging from 12.5 to 20 cm from the channel inlet. The first impact of the shock on the bottom wall occurs around $x = 21$ cm, so placing the plasma at $x = 20$ cm proves ineffective, as the kernel is too close to the separation bubble to produce a significant displacement. The characteristic lines of the shock at this location are beyond the plasma's influence, leading to only a minor impact. Next,

plasma placed at $x = 17.5$ cm has a more significant effect, but it was found that moving it even further upstream would lead to a more significant effect.

The results indicate that the ideal placement for the plasma is at $x = 12.5$ cm from the inlet, where the most significant upstream displacement of the second impact point is achieved. As shown in Figure 4.7-Right, the displacement of the reflected shock's second impact depends non-linearly on the plasma location, with a peak effect observed when the plasma is located sufficiently upstream, allowing for interaction with the boundary layer before the shock impacts the bottom wall. This interaction causes the boundary layer to thicken, influencing the shock structure.

When the plasma is placed too close to the first impact, such as at $x = 20$ cm, it still weakens the shock but fails to induce a significant displacement of the second impact point. The shock wave is mitigated, but the reduced distance between the plasma and the separation bubble limits the plasma's effectiveness in altering the flow structure.

Interestingly, there is minimal change in the displacement of the second impact point when the plasma is positioned between $x = 2.5$ and $x = 12.5$ cm from the inlet. This behavior suggests that this region's boundary layer is already fully developed. The plasma further enhances the boundary layer's thickness, but the flow conditions near the separation bubble remain relatively unchanged, preventing further displacement of the reflected shock.

Table 4.2: Location of the Plasma and Displacement of the Second Impact Point.

Location (cm)	Displacement (cm)
2.5	4.15
5.0	4.40
7.5	4.40
10.0	4.45
12.5	4.80
15.0	4.40
17.5	3.30
19.0	1.80
20.0	0.20

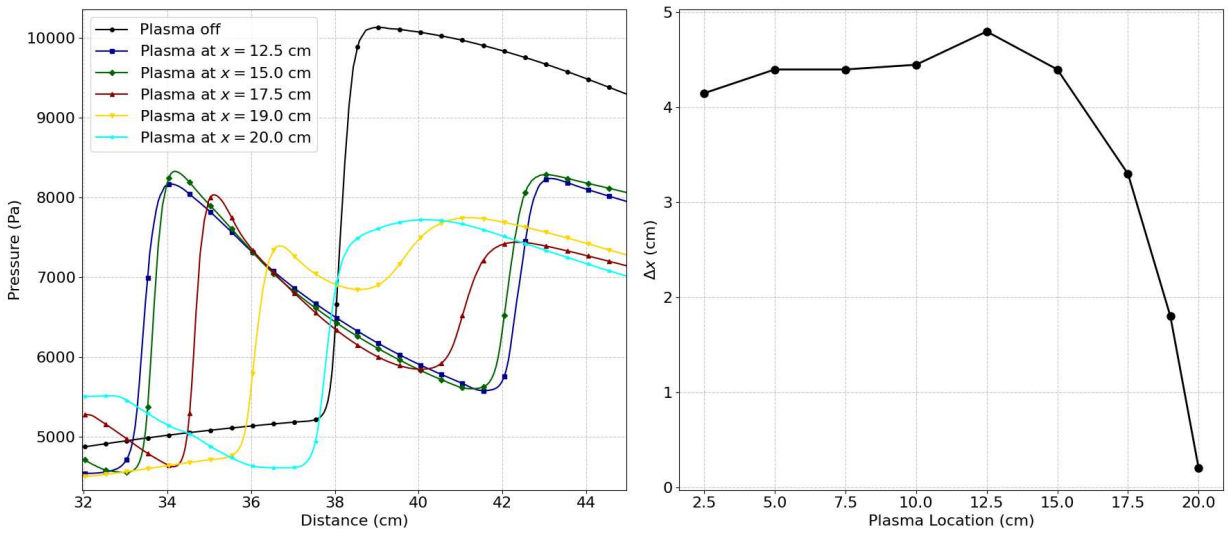

Figure 4.7: Left — Pressure distributions at the top wall for different plasma locations, locations < 12.5 omitted from the plot for clarity. Right— Plasma Location vs. Δx (Distance second impact traveled upstream). Simulations were performed at a constant plasma temperature of 5000 K.

Figure 4.8 shows a progression of shock wave interactions at three different locations: $x = 20$ cm, $x = 12.5$ cm, and $x = 2.5$ cm. At $x = 20$ cm, there is a small growth in the boundary layer. However, the effect when the plasma is placed at this location is minimal, as the plasma is far too close to significantly influence the flow. At $x = 12.5$ cm, the optimal location for this study, there is a greater impact with a larger separation bubble. At $x = 2.5$ cm, the boundary layer is fully developed since the plasma has a longer way to travel. Expectedly, the plasma has less impact than $x = 12.5$ cm but still forms a strong recirculation region. The flow in this region shows less variation compared to upstream locations, implying that the boundary layer has stabilized.

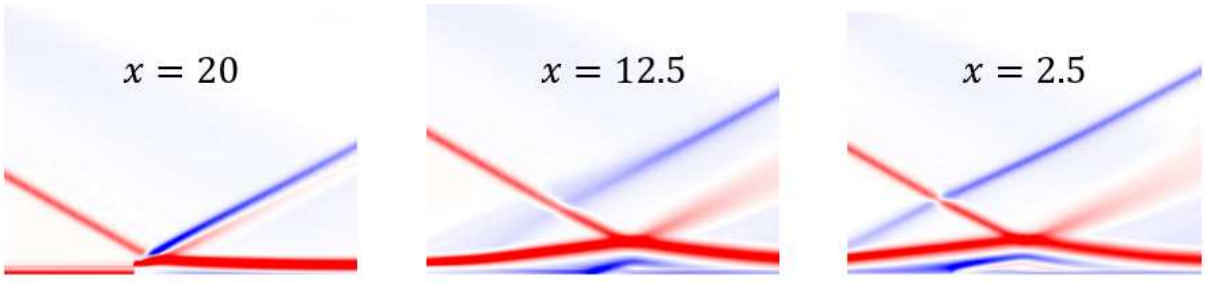

Figure 4.8: Boundary Layer formation at Plasma locations $x = 20$, $x = 12.5$ and $x = 2.5$ at $t = 2.5$ ms at a temperature of $T = 5000$ K

4.3.3 Plasma Temperature Effects

To further understand the influence of plasma energy deposition within the boundary layer on the shock wave structure, the plasma kernel temperature T_p was varied between 1,000 K and 8,000 K. The resulting pressure distributions are presented in Figure 4.9. The data, collected via simulated pressure taps mounted on the top wall, provide insight into how different plasma temperatures affect the shock wave boundary layer interaction (SWBLI).

In the baseline case, a strong leading-edge shock is observed without plasma actuation, with the first pressure peak around $x = 39$ cm. However, once the plasma is activated, the interaction dynamics change significantly. The leading-edge shock is split into two weaker shocks, with the first impact point shifting upstream by approximately 4 cm to 5 cm. The separation bubble generates this initial shock, while a weaker reattachment shock leads to the secondary peak observed at around $x = 42$ cm. This division of the shock structure demonstrates the plasma's ability to mitigate the strength of the original shock, reducing its impact on the flow. The total pressure recovery when the plasma is active is higher than in the baseline case. This suggests plasma actuation can reduce wave drag in supersonic internal flows by weakening shock wave interactions.

When analyzing the effect of different plasma temperatures, it becomes evident that higher plasma temperatures lead to more pronounced effects on shock displacement. For example, a plasma temperature of 8,000 K achieves the highest displacement of the second impact point, shifting it further downstream compared to lower temperatures. However, while higher temperatures produce more substantial effects, practical limitations in experimental setups and real-world applications make it challenging to achieve such extreme conditions. These high temperatures require unique, expensive materials. Therefore, for the remaining parts of this study, a plasma temperature of $T_p = 5,000$ K was chosen. This temperature represents a more realistic scenario for experimental validation while still providing significant control over the shock wave structure. It's also worth noting that even at the lowest temperature of $T_p = 1000$ K, the plasma still weakens the shock, though the displacement is minimal.

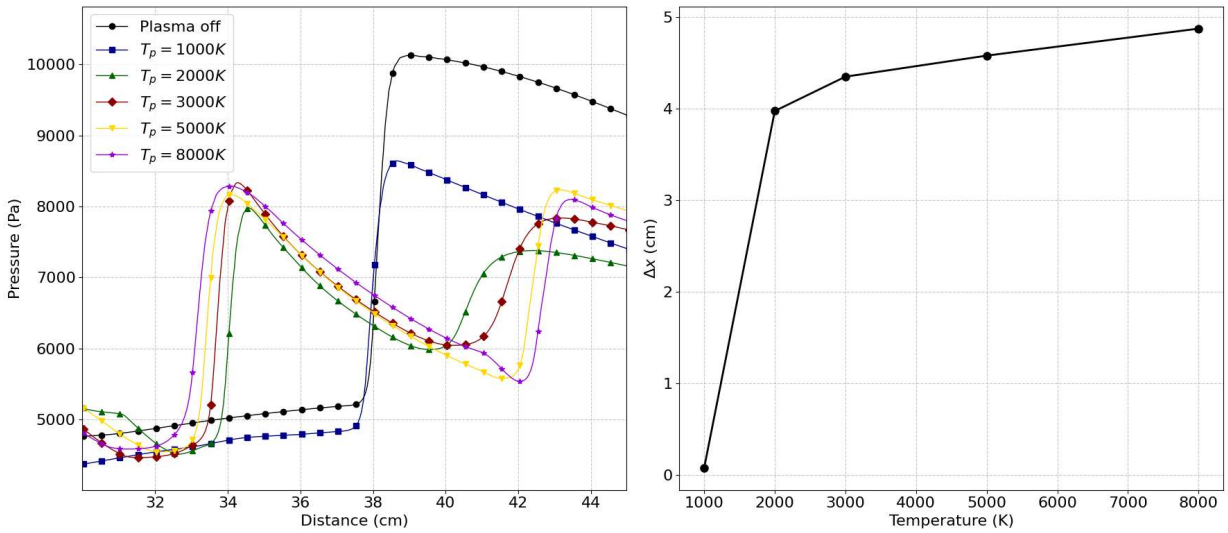

Figure 4.9: Left - Pressure distributions over the top wall. Right - Temperature vs. Δx (The change in second impact location). The results showcase the impact of plasma energy deposition on the SW train.

.

Table 4.3: Temperature of the Plasma and Displacement of the Second Impact Point.

Temperature (K)	Displacement (cm)
1000	0.07
2000	3.95
3000	4.35
5000	4.60
8000	4.88

The effect of varying plasma temperatures on shock wave-boundary layer interactions (SWBLI) is illustrated in Figure 4.10, which compares three cases with plasma temperatures of $T_p = 1000$, 2000, and 8000 K. These results demonstrate the significant impact of plasma energy on separation bubble growth and shock dynamics.

At $T_p = 1000$ K (Figure 4.10-Left), the plasma energy is modest, resulting in limited flow disruption. The temperature increase is just above the stagnation temperature of 650 K. It is insufficient to cause substantial growth in the separation bubble or significant changes in the reflected shock, as evident in Figure 4.9. Consequently, the flow remains stable, with minimal SWBLI effects.

For $T_p = 2000$ K (Figure 4.10-Middle), the higher plasma energy generates clear compression waves at the leading edge of the separation bubble and merges to form a stronger leading edge shock. An expansion wave follows, interacting with the separation bubble, leading to the formation of a distinct reattachment shock downstream. The increased separation in this case indicates a moderate influence of the plasma.

The most pronounced effect is observed at $T_p = 8000$ K (Figure 4.10-Right), where the plasma introduces substantial energy into the flow. This leads to the formation of the largest separation bubble and significant displacement between the leading edge shock and the reattachment shock. The darker recirculation region and the increased spacing between shock structures indicate more substantial SWBLI effects. Additionally, the plasma shock is faintly visible in this figure.

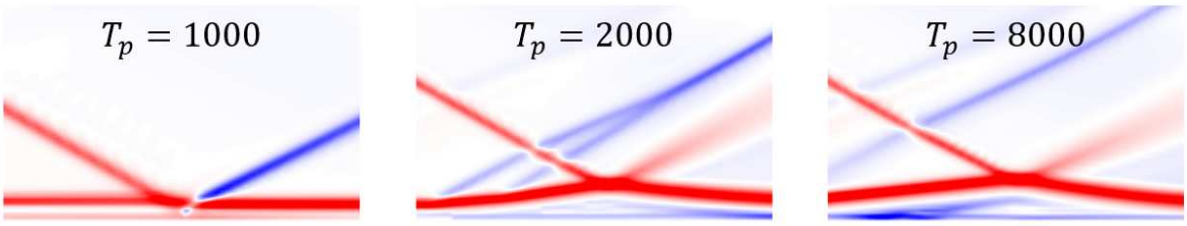

Figure 4.10: Separation bubble formation with plasma at different temperatures: $T_p = 1000$, 2000, and 8000 K from left to right, respectively.

4.4 Conclusion

In this chapter, the numerical investigation of plasma-assisted SW control was conducted. It was found through modeling that plasma kernels can control SWs if placed in the correct location within a channel. The movement of the SW train is shown most notably by visualizing the second impact point using synthetic Schlieren images, along with pressure distribution data over the top wall. The significant finding of this study is that plasma injection into the boundary layer can indeed affect the SW train in a supersonic channel. By varying the plasma energy deposition and the injection location, we observed an upstream displacement of the second impact point (located at the top wall, behind the wedge) by about 5 cm. The results suggest that plasma-assisted flow control can mitigate the adverse effects of SWs in supersonic flows and potentially reduce the total pressure losses in an inlet-isolator configuration.

Chapter 5

Numerical Results and Validation of the Cut-Cell Method

This chapter focuses on the improvements made to the APDL-CFD code by implementing the cut cell method. The first section validates the code before modifying it to ensure a stable base. The results are shown for the cut cell method and compared to results without the cut cell method to show the improvements made.

5.1 Code Validation

Before implementing the cut cell method, the code was validated using the Sod's shock tube problem [118]. The results of the shock tube problem are shown in Figure 5.1. The results show the density, and pressure profiles at $t = 0.2$ s. The results are compared to the exact solution and show good agreement.

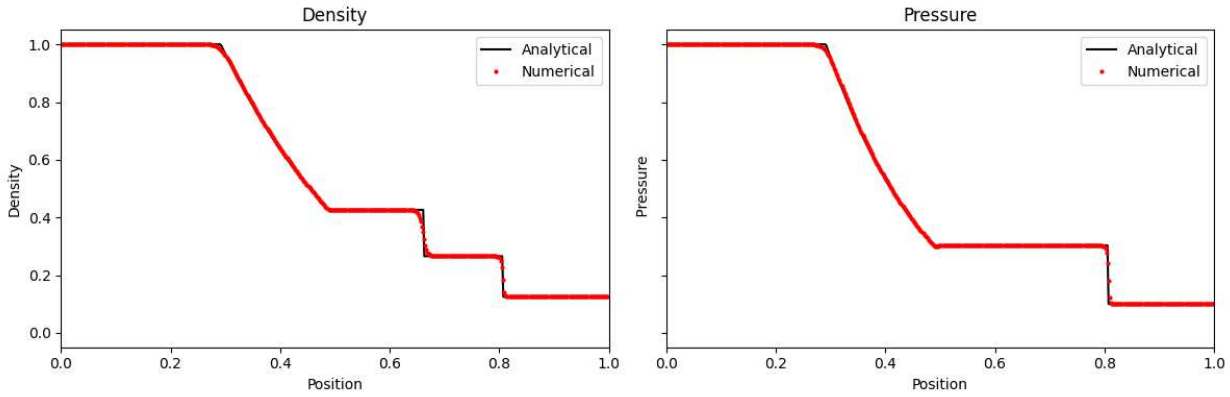

Figure 5.1: Sod's shock tube problem numerical simulation (red) and the analytical solution (black) show good agreement between the numerical results and the analytical.

The code was also validated by experimental results using a laser ignition problem, where the ignition kernel was simulated in an axisymmetric setup. The results of the laser ignition problem

are shown in Figure 5.2. Dumitrache shows the simulated kernel compared to the experimental kernel, results show good agreement between the two. The labels for Δz are the offsets from the dual-pulse laser. Readers interested in the details of the laser ignition problem can refer to Dumitrache et al. [11].

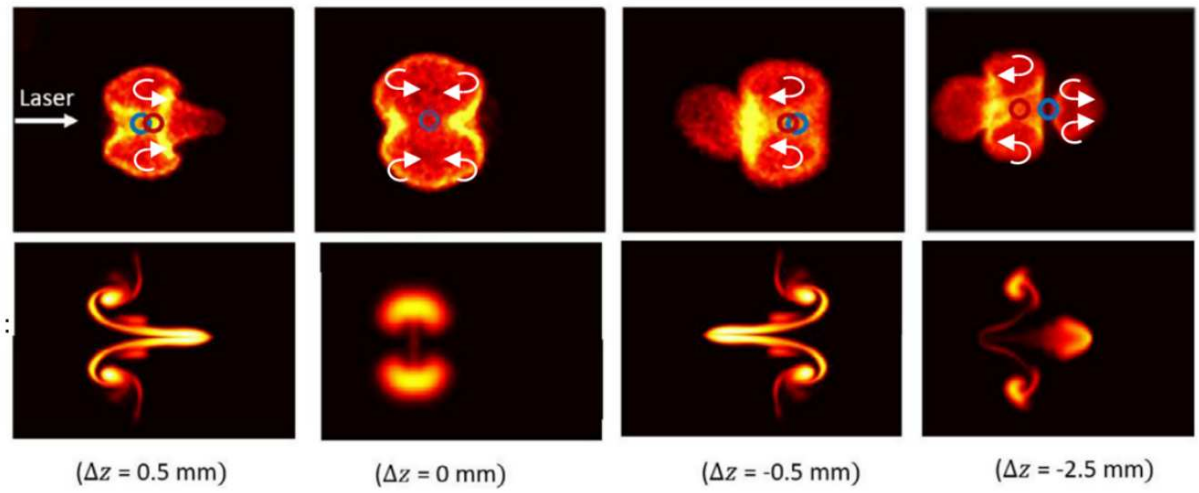

Figure 5.2: Laser ignition plasma kernel simulation (bottom) compared to experimental results (top) from [11].

5.1.1 Grid Independence

Grid independence differs for DNS from other methods, such as LES or RANS modeling. Typically, for turbulence modeling, the Kolmogorov scale is used to inform the number of cells needed to capture small-scale turbulence effects when viscosity dominates [119]. However, we are not dealing with a flow where viscosity dominates. As such, this study is not interested in modeling fine-scale turbulent effects. The Mach number is a much more critical factor for supersonic flow than the Reynolds number, as it is for turbulent flow. Since the bulk flow is the most important, a simple test was performed by comparing the 1D pressure values in the middle of the domain and taking the percent change between the different grid resolutions. The grid independence tests were run for 2E-4 seconds, just enough time for the flow to start going over the wedge, and an oblique shock wave began to form. The results are shown in Table 5.1 and Figure 5.3. The accepted percent

change when changing grid resolutions is 5% [120]. The results show that the 32000 or 50000 cell grids balance accuracy and computational resources best. The purpose is to keep the resources low so that a coarse grid can capture the desired results for the bulk flow properties. Many higher resolution grids were run, $N_{cells} > 3$ million, to get a more detailed look at the separation bubble, but for the bulk flow properties, a coarse grid can capture the desired results.

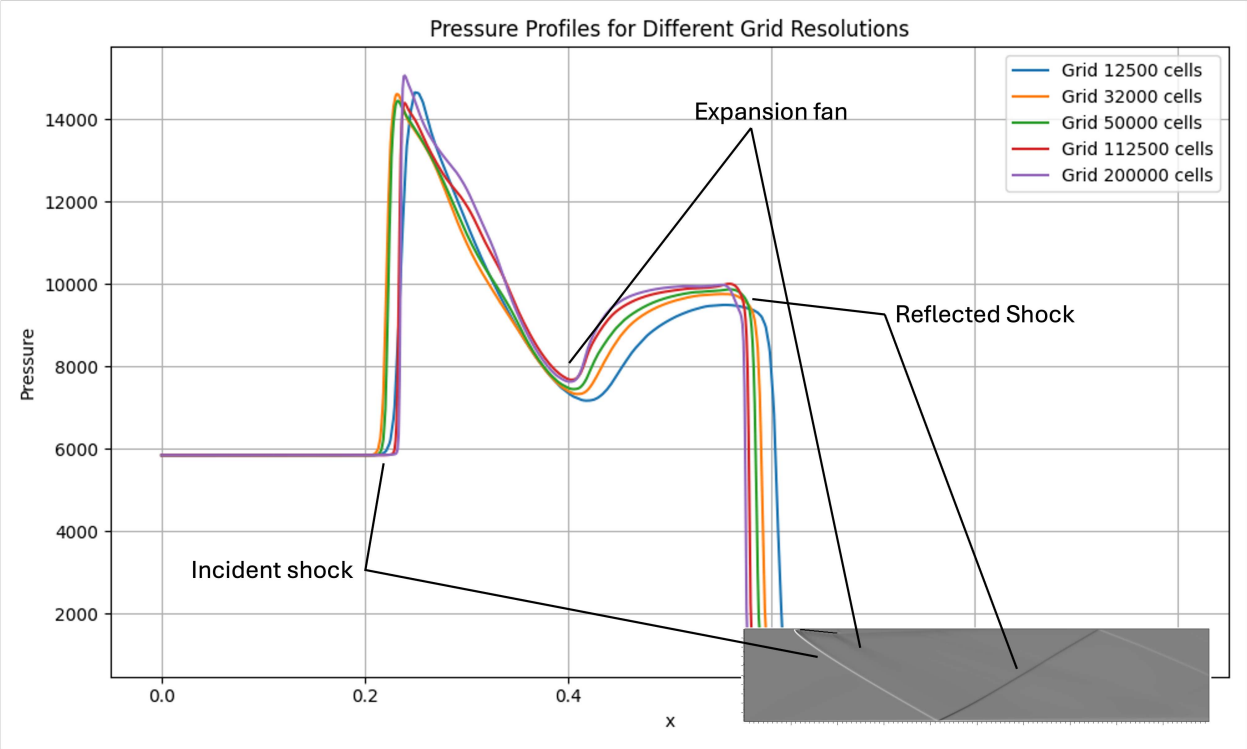

Figure 5.3: Study of grid independence for the wedge problem showing the average pressure across the y-direction.

Table 5.1: Grid independence study showing average pressure and percent change in average pressure for different grid resolutions.

Cells	Average Pressure	Percent Change (%)
12500	5394.74	—
32000	5425.36	4.53
50000	5414.51	1.53
112500	5367.79	2.55
200000	5366.63	1.18

5.2 Improvements to the APDL-CFD In-House Code

The aim of this code is to maintain simplicity while enabling rapid iterations of parametric studies by operating at a low resolution. The primary geometry tested with this code is the wedge used to generate the oblique shock waves presented in Chapter 2 and Chapter 4.

Figure 5.4 compares the velocity profile around two configurations of the wedge. The figure shows only the velocity profile, omitting the overlaying geometry to emphasize that the wedge is stationary, which is represented by the blue regions indicating zero velocity. The top image corresponds to the *blocky* wedge configuration, where the geometry is expressed solely by solid cells without the use of cut cells. In contrast, the bottom image depicts the *smooth* wedge configuration, where cut cells have been implemented to more accurately capture the wedge geometry.

The comparison between these two configurations highlights the advantages of the cut cell method. In the *blocky* wedge, the flow around the wedge is jagged because it is influenced by the stair-stepped representation of the boundary. However, in the *smooth* wedge, the velocity profile is significantly more uniform, demonstrating that the cut cell method allows for a more precise representation of geometries that do not conform to a Cartesian grid. Consequently, the smoother

velocity profile around the wedge confirms the effectiveness of the cut cell method in improving the accuracy of supersonic flow simulations involving complex geometries.

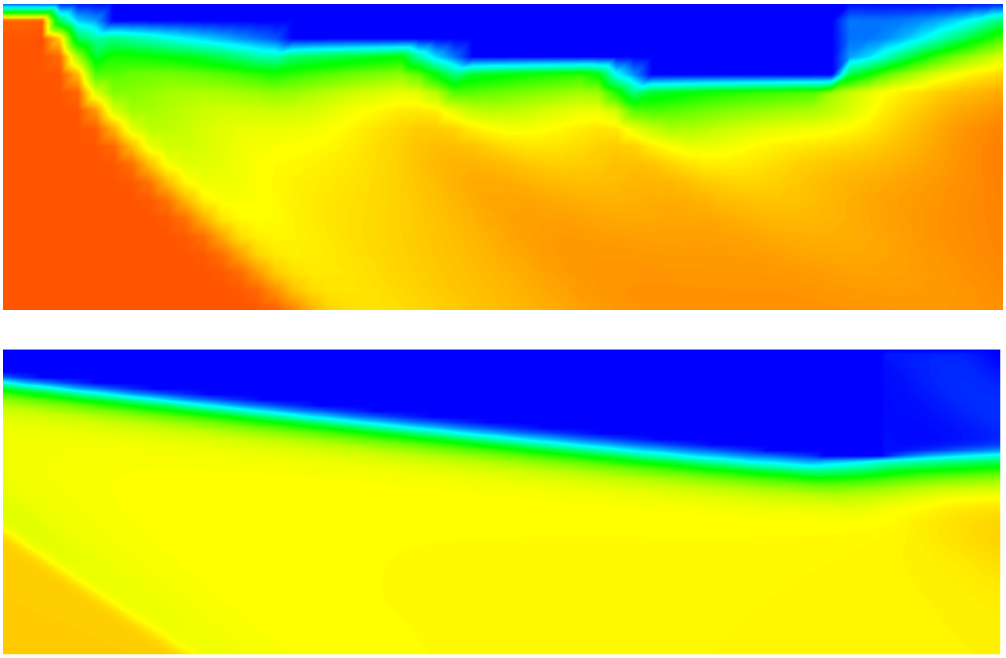

Figure 5.4: Velocity profiles near the wedge for a grid without cut cells (top) and with cut cells (bottom). Both figures are at the same grid resolution of 500×100 cells.

The difference is also shown in Figure 5.5, showing the geometric parameter α. The top figure in Figure 5.4 corresponds to the top figure in Figure 5.5, and the bottom figure in Figure 5.4 corresponds to the bottom figure in Figure 5.5. Here, the cell area fractions are shown for a 10-degree wedge. Only the solid cells are shown in the top image, where $\alpha = 0$. The bottom image shows the solid cells and the cut cells, where $1 > \alpha > 0$. The α values are used to weight the fluxes in the cut cells. The α values are calculated by the area of the cell that is in the domain of interest divided by the total area of the cell.

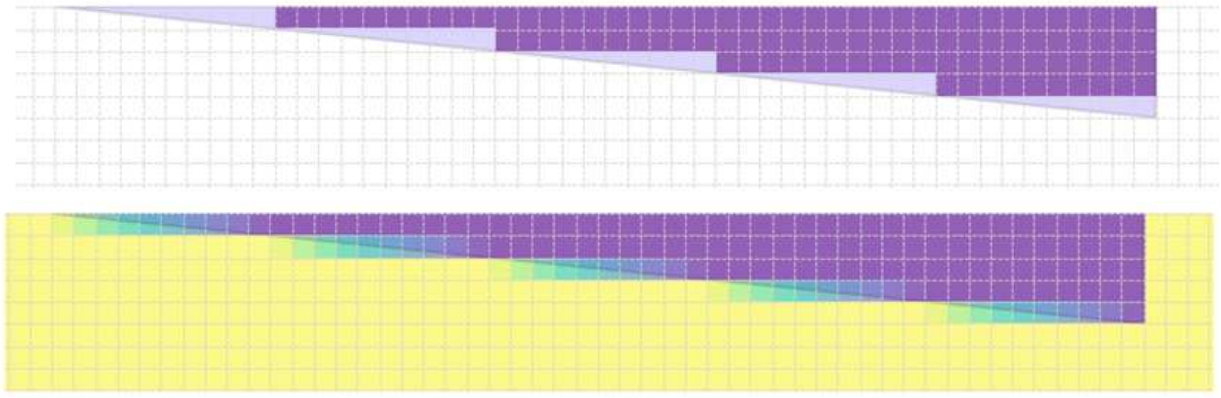

Figure 5.5: Matrix showing the α values for a 10-degree wedge (overlaid). The top shows the solid cells only, where $\alpha = 0$. The bottom shows the solid cells and the cut cells, where $1 > \alpha \geq 0$.

For the code's first iteration (level 0), the embedded boundary was defined as an area of zero velocity and constant density and pressure. Then, conditions were added to reflect the velocity. This worked well for high-resolution runs. However, the code is intended to be run at a relatively low resolution for rapid iterations to study different parameters. The effect in Figure 5.4-top is seen at low resolution. The next iteration (level 1) featured the cut cells, but they were implemented with just weighting by the β values. This resulted in a geometry that was acceptable but not complete. The final iteration of the code (level 2) was the full cut cell method, where the fluxes were reconstructed in the cut cells using the numerical methods in Chapter 3. Level 2 resulted in the smooth geometry shown in Figure 5.4-bottom.

To illustrate the code improvements by using the cut cell method, a 1D line was taken under the wedge to observe the flow properties. The lineout location is shown in Figure 5.6. Everything is taken at a constant y location.

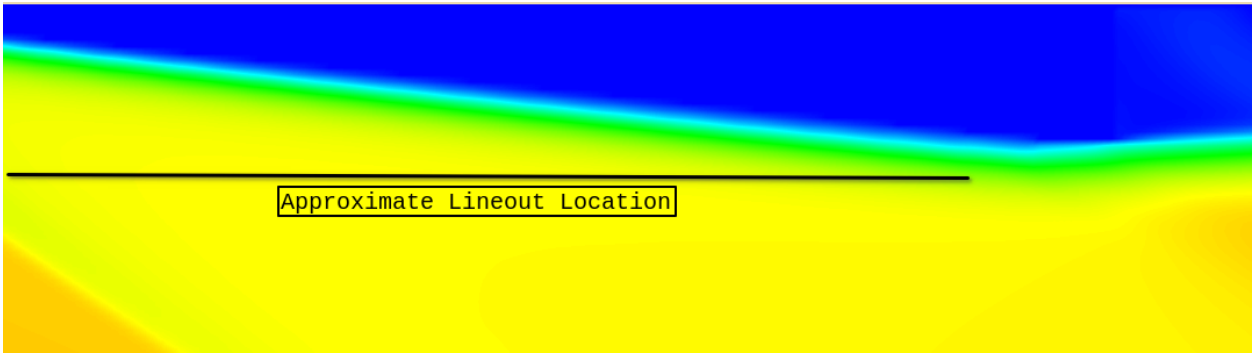

Figure 5.6: The lineout location shown under the wedge with a velocity magnitude profile.

The results at the lineout location in Figure 5.6 are shown in Figure 5.7, and the percent error of level 0 and level 1 iterations, based on the complete cut cells, are in Figure 5.8. Figure 5.7 shows the three levels of improvement of the code. The black line shows the level 0 case, which is the result with no weights or flux reconstruction around the wedge. In the level 0 case, the geometry is just made up of blocky cells, and the wedge is made up of four steps, so there are four spikes in each plot from the reflection off the steps. There is already a good improvement in the level 1 case, but the velocity is higher than expected that, close to the wedge. The spike in pressure and density caused by the initial impact with the wedge is also not smooth in the level 1 case. Level 2, represented by the blue line in Figure 5.7, shows the full cut cell method with the most accurate wedge representation.

At first glance, the level 2 case appears to have an off-by-one error, but this shift is physical. In the level 0 and 1 cases, the flow reflects off the wedge's first "step," causing the flow to reflect backward before turning to create an oblique shock. This acts as a small bow shock and pushes the flow slightly upstream. This is not physical if the goal is to model a smooth geometry, as it does not capture the smooth flow over the wedge. Level 2 also features a velocity profile that is more in line with what is expected. It is much lower than the first two levels due to the interpolation of the velocity profile near the wedge.

Furthermore, since the 1D line is taken at a constant y location, the velocity is expected to decrease the closer the line gets to the wedge. This is shown by the level 2 case, where the velocity

is lower than the level 0 and 1 cases. The increase in velocity at the end of the wedge is due to the expansion fan as the flow goes over the back of the wedge.

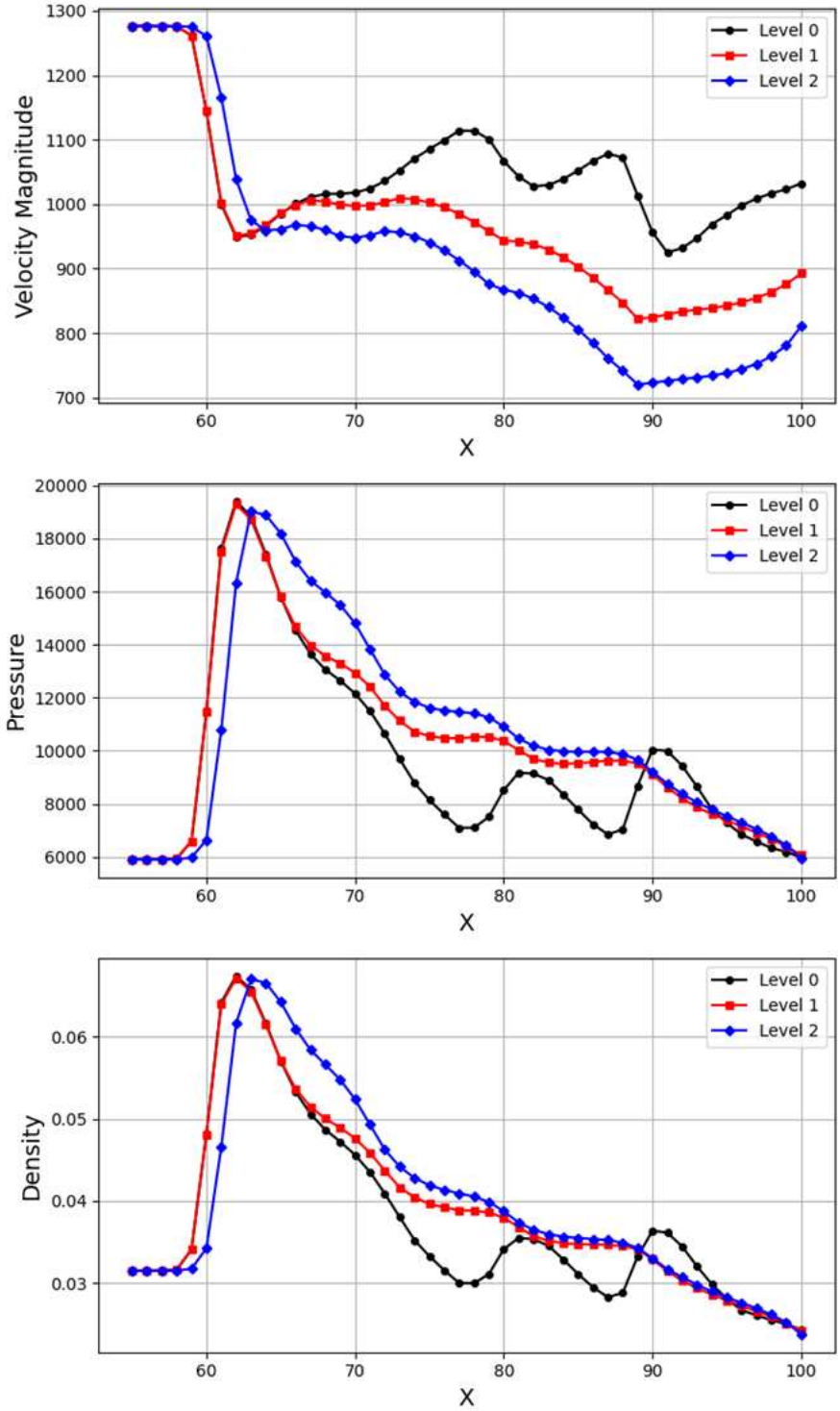

Figure 5.7: Profiles of the 1D lineout under the wedge at low resolution.

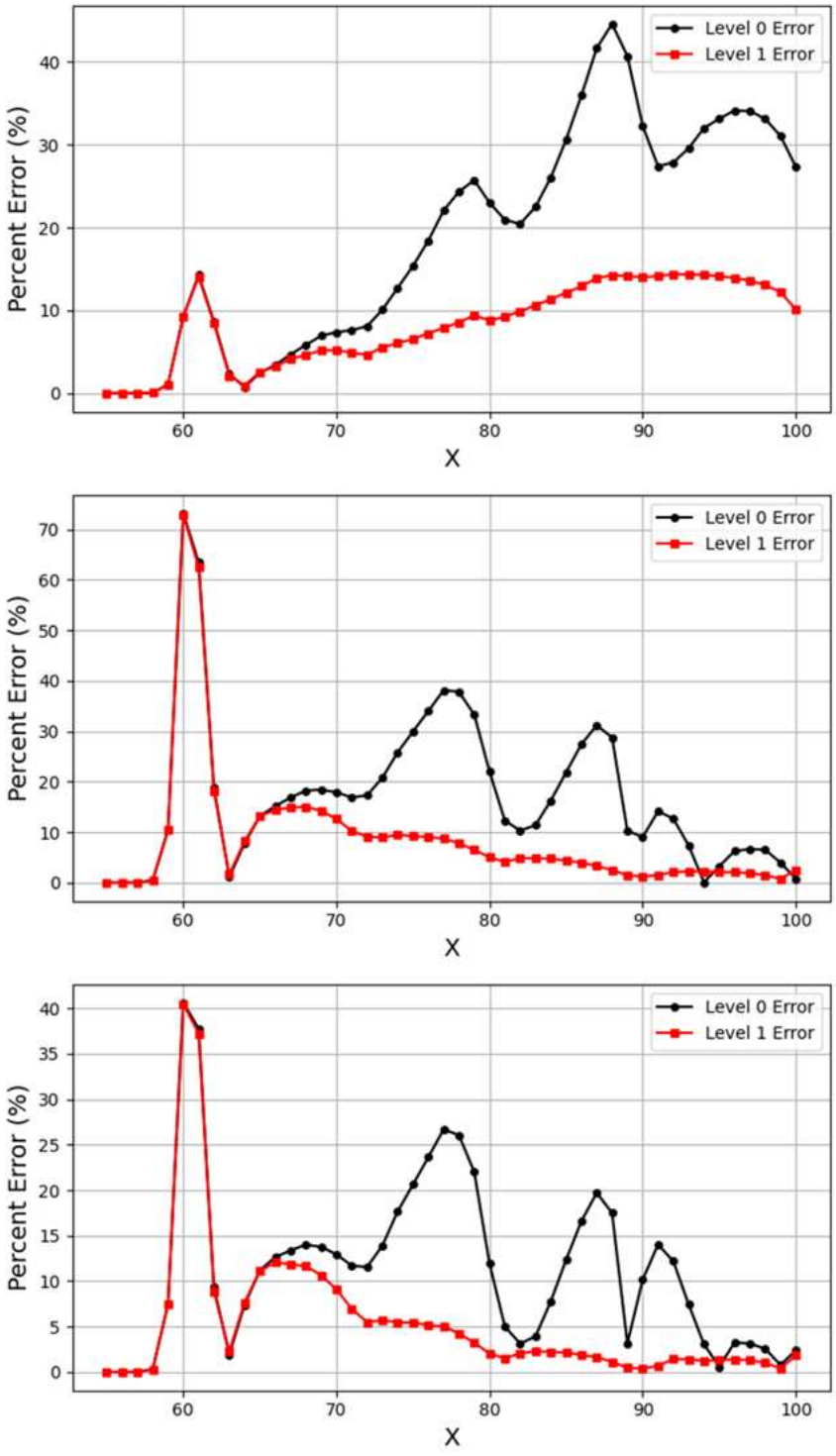

Figure 5.8: The percentage error at the 1D lineout location.

This method was also tested using a spherical geometry to compare to the analytical calculations of the standoff distance of the bow shock found by Farris and Russell [121]. The equation that relates the Mach number to the standoff distance of a bow shock from a blunt body is defined as:

$$\frac{D_{BS}}{D_{OB}} = 1 + 1.1\frac{(\gamma - 1)Ma^2 + 2}{(\gamma + 1)Ma^2}, \tag{5.1}$$

where D_{BS} and is the distance to the bow shock and D_{OB} the distance to the object origin. The $Ma = 2.5$ is known, as is $D_{OB} = 40mm$. Thus, the bow shock standoff distance can be calculated. The analytical standoff distance is calculated to be $D_{BS} = 14.2$ mm. This analytical solution will be compared to the numerical solution solved in the APDL-CFD code.

The code has the capability to run in axisymmetric for higher efficiency in symmetric flows. Thus, to test the spherical problem, the code was run in axisymmetric mode. It is important to note that the no-slip condition on the wall is turned off so that the simulation can be mirrored along that axis. The geometry is a quarter circle with a radius of 40 mm, depicted in Figure 5.9. The blue region represents the sphere. The gray cells around the sphere are the cut cells, and the α value is visualized through the cells. The red line represents the 1D lineout taken through the circle to compute the standoff distance. The second cell was taken for the measurement to avoid any issues when mirroring the geometry.

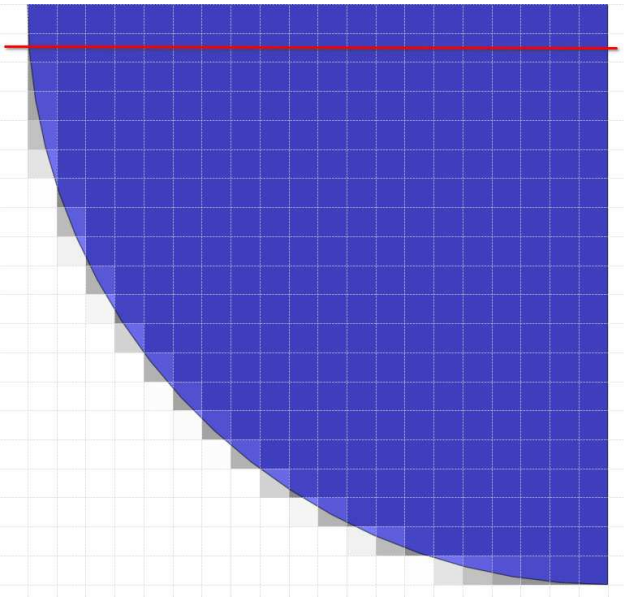

Figure 5.9: Quarter circle geometry with the α values shown for the cells inside $\alpha = 0$ and the cut cells with $1 > \alpha > 0$. The red line represents the 1D lineout location taken through in front of the geometry to compute the standoff distance.

The 1D lineout results plotted in Figure 5.10 depict the pressure and density through the bow shock and the wedge. The first spike is where the bow shock begins, and the second is the geometry, which is held at a constant pressure and density. Close to the geometry ($X \approx 186$), the "level 2" line has a smoother transition to the flow away from the boundary. The magnitude of pressure and density near the wedge is also more significant for the level 2 case, which is physical. Another indication that the cut cell method is effective here is that there is the same shift in the initial shock location from the geometry, as seen in the wedge problem.

The distance between the circular geometry's surface and the bowshock's start is $D_{BS} \approx 15$ mm in the cut cell case (level 2), and the blocky geometry (level 0) is $D_{BS} \approx 16$ mm. These values have an error of 5.6% and 12.7%, respectively. This simulation was computed at low resolution, so the error would be reduced to a higher resolution for both cases. However, these results showcase the cut cell's effectiveness at low resolution, especially when compared to the blocky geometry.

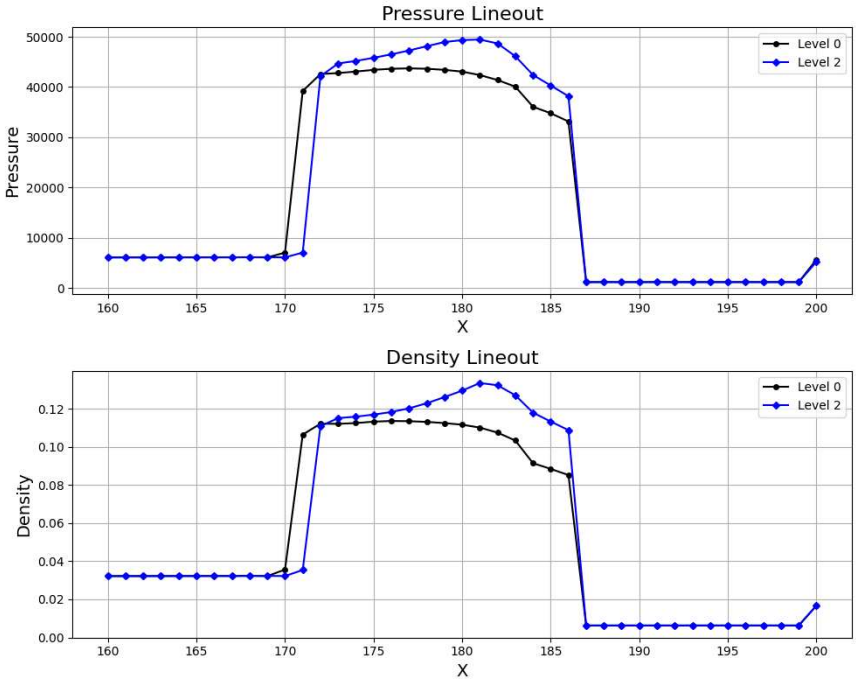

Figure 5.10: 1D line depicting the Bow shock distance for a spherical geometry.

The velocity magnitude plot around the circle is in Figure 5.11. There is a low velocity region at the front of the circle, which is expeceted due to the stagnation point. The velocity increases as the flow goes around the circle.

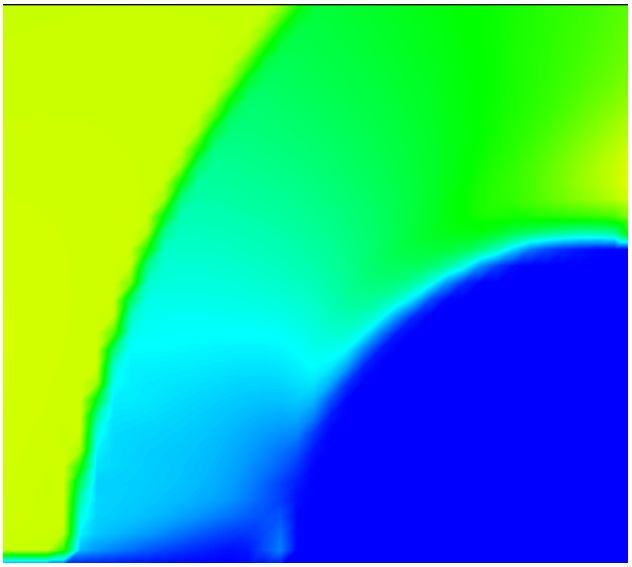

Figure 5.11: Velocity magnitude around the circle. The solid blue region is the quarter circle with the cut cell geometry.

The final figures showcase the full cut cell method for a wedge geometry for velocity magnitude, pressure and density, these are shown in Figure 5.12, Figure 5.13, and Figure 5.14, respectively. These are displayed at relatively low resolution to show the effect of the cut cell method. The results show the plasma active, as explained in Chapter 4. Note more results of the full flow field using the cut cell method are found in Chapter 4.

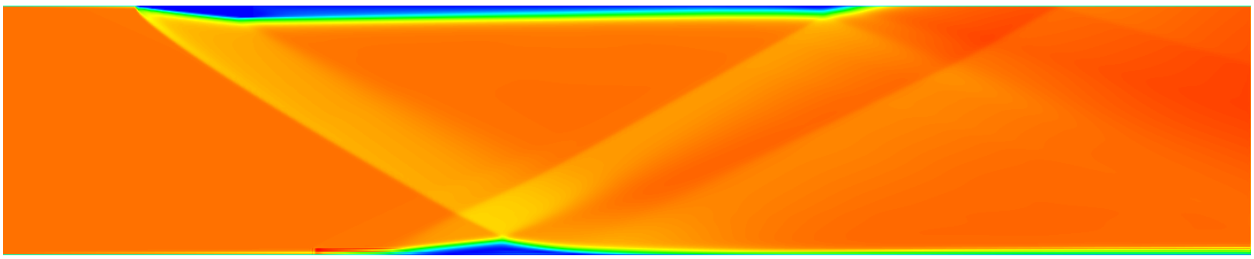

Figure 5.12: Velocity magnitude profile using the cut cell method at a grid size of 500×100 cells.

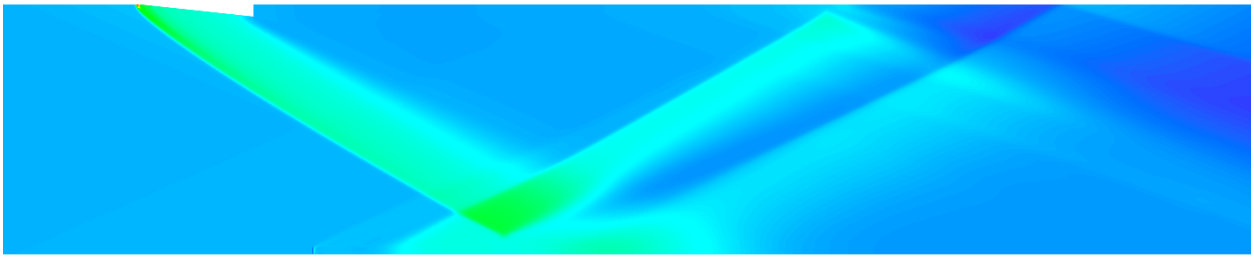

Figure 5.13: Pressure profile using the cut cell method at a grid size of 500×100 cells.

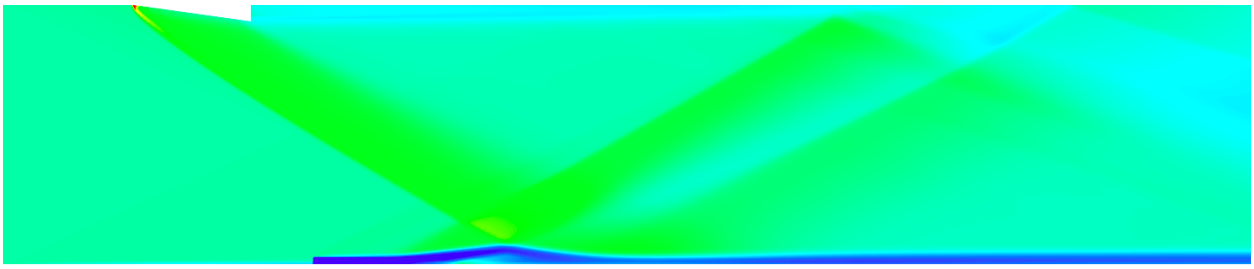

Figure 5.14: Density profile using the cut cell method at a grid size of 500×100 cells.

5.3 Conclusion

This chapter presents the numerical results obtained from the simulations and discusses the cut cell method's implications for modeling complex geometries in channel flow. First, the code was validated against a common benchmark problem of the Sod shock tube as well as experimental results. The code was also validated by testing the grid independence, proving that the bulk flow could be represented at a relatively low resolution. The results show that the cut cell method significantly improves the accuracy of the simulations, as demonstrated by the comparison between the blocky and smooth wedge configurations. The cut cell method provides a more precise representation of the wedge geometry, resulting in a smoother velocity profile around the wedge. The results of the 1D lineout under the wedge further confirm the effectiveness of the cut cell method in capturing the flow properties near complex geometries. Finally, these results allow the APDL-CFD code to be run iteratively at low resolution to gather information for experimental tests.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The primary objectives of this thesis were to (1) develop and validate a Computational Fluid Dynamics (CFD) model capable of simulating supersonic flow within channels of complex geometries; (2) simulate the effect of plasma injection on shock wave trains and boundary layer interactions; and (3) expand the APDL-CFD code using a Cartesian cut cell method to represent embedded boundaries.

Firstly, a robust CFD model was developed and validated to simulate the flow inside a supersonic channel with complex geometries. The model successfully captured essential features of supersonic flow, including shock wave formations and boundary layer behaviors over a 10-degree wedge within a straight channel. These conditions are representative of scramjet inlets and isolators. Validation was achieved through analytical solutions and experimental data comparisons, confirming the model's reliability for studying supersonic flow phenomena.

Secondly, the validated CFD model was employed to investigate the effects of plasma injection on shock wave trains and boundary layer interactions. By injecting plasma at various locations along the bottom of the channel and varying the plasma temperature, the simulations demonstrated that plasma actuators can effectively control the position and strength of shock wave trains. Plasma injection altered the flow field, modified shock wave structures, and influenced separation bubbles. Notably, the plasma effectively reduced shock strength and shifted shock wave positions and angles. These findings suggest that plasma actuators have significant potential for active flow control in hypersonic propulsion systems, offering ways to optimize performance, reduce drag, and enhance thrust.

Thirdly, to facilitate efficient simulations of complex geometries without incurring prohibitive computational costs, the APDL-CFD code was expanded using a Cartesian cut cell method. This

enhancement accurately represented embedded boundaries, such as wedges, on a Cartesian mesh. The cut cell method maintained computational efficiency while preserving accuracy, enabling rapid simulations necessary for extensive parametric studies. This advancement improved the code's capabilities and made it more suitable for investigating active control mechanisms in supersonic flows.

In summary, this research has successfully achieved its objectives by developing a validated CFD model for supersonic flows in complex geometries, demonstrating the effectiveness of plasma actuators in controlling shock wave trains and boundary layer interactions and enhancing the APDL-CFD code through the implementation of the Cartesian cut cell method. The insights gained from this study contribute to the understanding of active flow control in hypersonic propulsion systems. They can potentially inform future wind tunnel tests and the design of hypersonic vehicle engines.

6.2 Future Work

6.2.1 Adaptive Mesh Refinement (AMR) and Parallel Methods

Capturing the fine details in the boundary layer would increase the accuracy of simulating supersonic flow. One way to do this is to add inflation. This allows the boundary layer to include more cells than the bulk flow, increasing the resolution locally near the boundary. An alternative to this would be to add multigrid or adaptive mesh refinement. One branch of the APDL-CFD code is concerned with developing an AMR addition to the code. Combined with the cut cell method, this would allow for the accurate representation of geometries, increased resolution near the boundaries, and more efficient use of computational resources. This would allow the grid to be refined in areas of high interest, such as separation bubbles, and coarse in regions of low interest, such as the bulk flow. AMR would also significantly improve the accuracy of the interface between the fluid and the solid. Additionally, with block-structured AMR comes a greater ability to parallalize [122]. Implementing these changes can further improve the efficiency of the code.

6.2.2 High Performance Computing

If there is a need to add chemistry and plasma kinetics to the model to get further details, then high-performance computing (HPC) will be required. Parallelizing the code will become extremely important to increase its efficiency and allow it to run on supercomputers through message-passing interface (MPI) based codes. Parallelization of the code can be paired with the above section, AMR, to allow for the most efficient use of computational resources. Pairing AMR with HPC has been a great success in the past [123]. The challenge to consider with this dimensional splitting requires different considerations of the equations. This has been done in the YALES2 code by Moureau et al. [124]. However, This was done on an unstructured mesh, so effort needs to be made to complete this task on structured meshes [125]. Another push for more efficiency would be to adapt the code to be run on GPUs [126]. Utilizing the power of GPUs would allow for the code to run more efficiently.

6.2.3 Further Plasma Modeling

To fully resolve the physics of the flow, chemistry, and plasma kinetics should be included in the model. While the current research is not concerned with the microscopic details of plasma energy transfer and chemistry effects, it would be helpful to study this in detail in the future. Knowledge of the plasma's thermal effect on the wall is crucial in choosing and developing materials that can withstand high temperatures. Moreover, we believe that some of the shock energy would be absorbed into the plasma-energized boundary layer, increasing vibrational and electronic temperatures and a further decrease in shock intensity. This energy transfer in the plasma-activated boundary layer is not captured at this stage but might be useful to investigate. A further investigation of the fine-scale chemistry and physics can be done using a plasma model such as the one used by Zhang et al. [32] and Hooshyar [127]; the general form of the equations is provided in Appendix B.

However, the timescales required to properly resolve the chemical reactions are much smaller than the timescales of the bulk flow. This disparity greatly increases the computationally ex-

pense, as resolving the fine-scale chemistry necessitates much smaller time steps and grid size. Another branch of our code is currently being prepared, which replaces the explicit fourth-order Runge-Kutta (RKIV) time integration method with a more robust stiff ODE solver package called DVODE, developed by Lawrence Livermore National Laboratory (LLNL). This solver is better suited for handling the stiff equations arising from detailed chemistry. Implementing DVODE in the 2-D version of the code, along with parallelization, would help manage the computational demands. This development leads back to the previous discussions on Adaptive Mesh Refinement (AMR) and High-Performance Computing (HPC), emphasizing the need for advanced computational strategies to efficiently simulate the coupled flow and chemistry.

Overall, a detailed plasma model is underway in our laboratory and the goal is to combine it with this code to get a wholistic view of the process. This ongoing research will involve expanding the modeling of plasma discharge effects, using methodologies outlined by Hooshyar et al. [127].

Bibliography

[1] Tamal Jana and Mrinal Kaushik. Survey of control techniques to alleviate repercussions of shock-wave and boundary-layer interactions. *Advances in Aerodynamics*, 4(1):27, 2022.

[2] Dean Andreadis. Scramjet engines enabling the seamless integration of air & space operations. *Pratt & Whitney Space Propulsion, Hypersonics, West Palm Beach, Fl*, pages 33410–9600, 2004.

[3] Santanu Ghosh. *An immersed boundary method for simulating the effects of control devices used in mitigating shock/boundary-layer interactions*. North Carolina State University, 2010.

[4] H Babinsky, Yl Li, and CW Pitt Ford. Microramp control of supersonic oblique shock-wave/boundary-layer interactions. *AIAA journal*, 47(3):668–675, 2009.

[5] Piotr Doerffer and Oskar Szulc. Shock wave strength reduction by passive control using perforated plates. *Journal of Thermal Science*, 16:97–104, 2007.

[6] Dirk Schulte, Andreas Henckels, and Ralph Neubacher. Manipulation of shock/boundary-layer interactions in hypersonic inlets. *Journal of Propulsion and Power*, 17(3):585–590, 2001.

[7] Mohd Y Ali, Farrukh S Alvi, Rajan Kumar, C Manisankar, SB Verma, and L Venkatakrishnan. Studies on the influence of steady microactuators on shock-wave/boundary-layer interaction. *AIAA journal*, 51(12):2753–2762, 2013.

[8] Skye Elliott, Philip Lax, and Sergey B. Leonov. *Control of Shock Positions in a Supersonic Duct by Plasma Array*, page 2553. AIAA, 2022.

[9] Kyle T Ruggles, Richard B Miles, Nathan R Tichenor, and Christopher Limbach. Dual-mode energy deposition for hypersonic aerodynamic control. In *AIAA AVIATION FORUM AND ASCEND 2024*, page 4596, 2024.

[10] Robert R Arslanbekov, Vladimir I Kolobov, and Anna A Frolova. Immersed boundary method for boltzmann and navier-stokes solvers with adaptive cartesian mesh. In *AIP Conference Proceedings*, volume 1333, pages 873–877. American Institute of Physics, 2011.

[11] Ciprian Dumitrache, Arnaud Gallant, Nicolas Minesi, Sergey Stepanyan, Gabi D Stancu, and Christophe O Laux. Hydrodynamic regimes induced by nanosecond pulsed discharges in air: mechanism of vorticity generation. *Journal of Physics D: Applied Physics*, 52(36):364001, 2019.

[12] C Fureby, M Chapuis, E Fedina, and Sebastian Karl. Cfd analysis of the hyshot ii scramjet combustor. *Proceedings of the Combustion Institute*, 33(2):2399–2405, 2011.

[13] Devendra Sen, Apostolos Pesyridis, and Andrew Lenton. A scramjet compression system for hypersonic air transportation vehicle combined cycle engines. *Energies*, 11(6):1568, 2018.

[14] Kozo Fujii. Progress and future prospects of cfd in aerospace—wind tunnel and beyond. *Progress in Aerospace Sciences*, 41(6):455–470, 2005.

[15] Corin Segal. *The scramjet engine: processes and characteristics*, volume 25. Cambridge University Press, 2009.

[16] Richard Lechner, Jörn Sesterhenn, and Rainer Friedrich. Turbulent supersonic channel flow. *Journal of Turbulence*, 2(1):001, 2001.

[17] Datta V Gaitonde. Progress in shock wave/boundary layer interactions. *Progress in Aerospace Sciences*, 72:80–99, 2015.

[18] Jean Délery and Jean-Paul Dussauge. Some physical aspects of shock wave/boundary layer interactions. *Shock waves*, 19(6):453–468, 2009.

[19] Yuanshu Liu, Xin He, Rui Xue, Yuntian Zhang, Chaoqi Xu, Jianxun Du, et al. Numerical simulation of the interaction between shock train and combustion in three-dimensional m12-02 scramjet model. *International Journal of Hydrogen Energy*, 47(12):8026–8036, 2022.

[20] Rene Pecnik, Vincent Terrapon, Frank Ham, and Gianluca Iaccarino. Full system scramjet simulation. *Annual Research Briefs*, 2009.

[21] NM Sudharsan, VA Jambekhar, and V Babu. A validation study of openfoam using the supersonic flow in a mixed compression intake. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 224(6):673–679, 2010.

[22] Uriel Goldberg, Oshin Peroomian, Sukumar Chakravarthy, Balu Sekar, Uriel Goldberg, Oshin Peroomian, Sukumar Chakravarthy, and Balu Sekar. Validation of cfd++ code capability for supersonic combustor flowfields. In *33rd Joint Propulsion Conference and Exhibit*, page 3271, 1997.

[23] Brian Maicke, Timothy Barber, and Joe Majdalani. Evaluation of cfd codes for hypersonic flow modeling. In *46th AIAA/ASME/SAE/ASEE joint propulsion conference & exhibit*, page 7184, 2010.

[24] MA Keller, MJ Kloker, SV Kirilovskiy, PA Polivanov, AA Sidorenko, and AA Maslov. Study of flow control by localized volume heating in hypersonic boundary layers. *CEAS Space Journal*, 6:119–132, 2014.

[25] Spencer Teeter, Katie Plese, Rebecca Zulch, Caleigh Haid, Bret Windom, Azer P Yalin, and Ciprian Dumitrache. Development of a supersonic wind tunnel facility for scramjet testing at colorado state university. In *AIAA SCITECH 2024 Forum*, page 2129, 2024.

[26] R Klein, KR Bates, and N Nikiforakis. Well-balanced compressible cut-cell simulation of atmospheric flow. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1907):4559–4575, 2009.

[27] Ata Onur Baskaya and Stefan Hickel. A conservative cut-cell immersed boundary method for accurate simulation of hypersonic flows with gas-surface interactions. In *57th 3AF international conference on applied aerodynamics*, 2023.

[28] David M Ingram, Derek M Causon, and Clive G Mingham. Developments in cartesian cut cell methods. *Mathematics and Computers in Simulation*, 61(3-6):561–572, 2003.

[29] Lennart Schneiders, Daniel Hartmann, Matthias Meinke, and Wolfgang Schröder. An accurate moving boundary formulation in cut-cell methods. *Journal of Computational Physics*, 235:786–809, 2013.

[30] Marsha Berger and Michael Aftosmis. Progress towards a cartesian cut-cell method for viscous compressible flow. In *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 1301, 2012.

[31] Alexander O Kleb, Krzysztof Fidkowski, and Joaquim R Martins. Development of a cartesian cut-cell solver for viscous flows. In *AIAA SciTech 2023 Forum*, page 1795, 2023.

[32] Wenqing Zhang, Zhijun Zhang, Xiaowei Wang, and Tianyi Su. A review of the mathematical modeling of equilibrium and nonequilibrium hypersonic flows. *Advances in Aerodynamics*, 4(1):38, 2022.

[33] David Sziroczak and Howard Smith. A review of design issues specific to hypersonic flight vehicles. *Progress in Aerospace Sciences*, 84:1–28, 2016.

[34] Thomas A Heppenheimer. *Facing the heat barrier: a history of hypersonics*, volume 4232. NASA, 2006.

[35] David M Van Wie. Hypersonics: Past, present, and potential future. *Johns Hopkins APL Technical Digest*, 35(4):335–341, 2021.

[36] Thomas Fetterhoff, Edward Kraft, Marion Laster, and William Cockson. High-speed/hypersonic test and evaluation infrastructure capabilities study. In *14th AIAA/AHI Space Planes and Hypersonic Systems and Technologies Conference*, page 8043, 2006.

[37] Scott Berry, Roger Kimmel, and Eli Reshotko. Recommendations for hypersonic boundary layer transition flight testing. In *41st AIAA Fluid Dynamics Conference and Exhibit*, page 3415, 2011.

[38] Graham V Candler, Pramod K Subbareddy, and Joseph M Brock. Advances in computational fluid dynamics methods for hypersonic flows. *Journal of Spacecraft and Rockets*, 52(1):17–28, 2015.

[39] Thomas M Evans, Andrew Siegel, Erik W Draeger, Jack Deslippe, Marianne M Francois, Timothy C Germann, William E Hart, and Daniel F Martin. A survey of software implementations used by application codes in the exascale computing project. *The International Journal of High Performance Computing Applications*, 36(1):5–12, 2022.

[40] Mori Mani and Andrew J Dorgan. A perspective on the state of aerospace computational fluid dynamics technology. *Annual Review of Fluid Mechanics*, 55(1):431–457, 2023.

[41] R Cosner. Integrated flowfield analysis methodology for fighter inlets. In *Aircraft Design Systems and Operations Meeting*, page 3071, 1985.

[42] Douglas L Dwoyer, Paul Kutler, and Louis A Povinelli. Retooling cfd for hypersonic aircraft. *Aerospace America*, 25, 1987.

[43] Louis A Povinelli. Advanced computational techniques for hypersonic propulsion. In *International Symposium on Air Breathing Engines*, number E-4711, 1989.

[44] Arthur Rizzi and James M Luckring. Historical development and use of cfd for separated flow simulations relevant to military aircraft. *Aerospace Science and Technology*, 117:106940, 2021.

[45] Robert R Barthelemy. The national aero-space plane program-a revolutionary concept. Technical report, SAE Technical Paper, 1991.

[46] John D Anderson Jr, Mark J Lewis, Ajay P Kothari, and Stephen Corda. Hypersonic waveriders for planetary atmospheres. *Journal of Spacecraft and Rockets*, 28(4):401–410, 1991.

[47] William H Heiser and David T Pratt. *Hypersonic airbreathing propulsion*. Aiaa, 1994.

[48] Jeffrey P Slotnick, Abdollah Khodadoust, Juan Alonso, David Darmofal, William Gropp, Elizabeth Lurie, and Dimitri J Mavriplis. Cfd vision 2030 study: a path to revolutionary computational aerosciences. Technical report, 2014.

[49] Hariswaran Sitaraman, Shashank Yellapantula, Marc T. Henry de Frahan, Bruce Perry, Jon Rood, Ray Grout, and Marc Day. Adaptive mesh based combustion simulations of direct fuel injection effects in a supersonic cavity flame-holder. *Combustion and Flame*, 232:111531, 2021.

[50] M Berger. Cut cells: Meshes and solvers. In *Handbook of Numerical Analysis*, volume 18, pages 1–22. Elsevier, 2017.

[51] Michael J Aftosmis, Marsha J Berger, and John E Melton. Robust and efficient cartesian mesh generation for component-based geometry. *AIAA journal*, 36(6):952–960, 1998.

[52] Nandan Gokhale, Nikos Nikiforakis, and Rupert Klein. A dimensionally split cartesian cut cell method for hyperbolic conservation laws. *Journal of Computational Physics*, 364:186–208, 2018.

[53] Michael Meyer, A Devesa, Stefan Hickel, Xiangyu Y Hu, and Nikolaus A Adams. A conservative immersed interface method for large-eddy simulation of incompressible flows. *Journal of Computational Physics*, 229(18):6300–6317, 2010.

[54] Dharshi Devendran, Daniel Graves, Hans Johansen, and Terry Ligocki. A fourth-order cartesian grid embedded boundary method for poisson's equation. *Communications in Applied Mathematics and Computational Science*, 12(1):51–79, 2017.

[55] Peter T Brady and Daniel Livescu. Foundations for high-order, conservative cut-cell methods: stable discretizations on degenerate meshes. *Journal of Computational Physics*, 426:109794, 2021.

[56] Charles S Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220–252, 1977.

[57] D.M Causon, D.M Ingram, and C.G Mingham. A cartesian cut cell method for shallow water flows with moving boundaries. *Advances in Water Resources*, 24(8):899–911, 2001.

[58] Jiankang Liu and Zhoushun Zheng. A dimension by dimension splitting immersed interface method for heat conduction equation with interfaces. *Journal of Computational and Applied Mathematics*, 261:221–231, 2014.

[59] Marsha J Berger, Christiane Helzel, and Randall J LeVeque. H-box methods for the approximation of hyperbolic conservation laws on irregular grids. *SIAM Journal on Numerical Analysis*, 41(3):893–918, 2003.

[60] Marsha J Berger and Phillip Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of computational Physics*, 82(1):64–84, 1989.

[61] Michael K Smart. Scramjet isolators. *AVT-185 AVT/VKI Lecture Series, von Kármán Inst. RTO-EN-AVT-185-AC/323 (AVT-185) TP/375, Rhode St. Genèse, Belgium*, 2010.

[62] Oskar Szulc, Piotr Doerffer, Pawel Flaszynski, and Marianna Braza. Moving wall effect on normal shock wave–turbulent boundary layer interaction on an airfoil. *International Journal of Numerical Methods for Heat & Fluid Flow*, 34(7):2567–2600, 2024.

[63] Geoffrey Ingram Taylor and JW Maccoll. The air pressure on a cone moving at high speeds.—ii. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 139(838):298–311, 1933.

[64] Mohsen Jahanmiri. Laminar separation bubble: its structure, dynamics and control. *Chalmers University of Technology*, pages 297–303, 2011.

[65] Robin Hunt, James F Driscoll, and Mirko Gamba. Periodic forcing of a shock train in mach 2.0 flow. In *55th AIAA aerospace sciences meeting*, page 0088, 2017.

[66] A Valdivia, KB Yuceil, JL Wagner, NT Clemens, and DS Dolling. Control of supersonic inlet-isolator unstart using active and passive vortex generators. *AIAA journal*, 52(6):1207–1218, 2014.

[67] Dorbala Sai Naga Bharghava, Amit Krishnat Mali, Tamal Jana, and Mrinal Kaushik. Numerical studies on uncontrolled and controlled shock wave/boundary layer interactions in hypersonic intake. *Aerospace Systems*, 7(3):539–557, 2024.

[68] Yue Zhang, Hui-jun Tan, Shu Sun, and Cai-yan Rao. Control of cowl shock/boundary-layer interaction in hypersonic inlets by bump. *AIAA Journal*, 53(11):3492–3496, 2015.

[69] HA Holden and H Babinsky. Separated shock-boundary-layer interaction control using streamwise slots. *Journal of Aircraft*, 42(1):166–171, 2005.

[70] AN Smith, H Babinsky, JL Fulker, and PR Ashill. Shock wave/boundary-layer interaction control using streamwise slots in transonic flows. *Journal of Aircraft*, 41(3):540–546, 2004.

[71] Wen-Zhong Xie, Zhong-Ming Wu, An-Yuan Yu, and Shengmin Guo. Control of severe shock-wave/boundary-layer interactions in hypersonic inlets. *Journal of Propulsion and Power*, 34(3):614–623, 2018.

[72] G Humrutha, M Kaushik, and KP Sinhamahapatra. Shock-boundary layer interaction control using innovative micro-vortex generators in supersonic intake. In *Proceedings, 47th AIAA fluid dynamics conference, Colorado, USA*, 2017.

[73] Tamal Jana, T Thillaikumar, and Mrinal Kaushik. Assessment of cavity covered with porous surface in controlling shock/boundary-layer interactions in hypersonic intake. *International Journal of Aeronautical and Space Sciences*, pages 1–18, 2020.

[74] Ryszard Szwaba. Influence of air-jet vortex generator diameter on separation region. *Journal of Thermal Science*, 22(4):294–303, 2013.

[75] Jean M Delery. Shock wave/turbulent boundary layer interaction and its control. *Progress in Aerospace Sciences*, 22(4):209–280, 1985.

[76] MK Fukuda, WR Hingst, and E Reshotko. Bleed effects on shocka3oundary-layer interactions in supersonic mixed compression inlets. *Journal of Aircraft*, 14(2):151–156, 1977.

[77] R Sriram and G Jagadeesh. Shock tunnel experiments on control of shock induced large separation bubble using boundary layer bleed. *Aerospace Science and Technology*, 36:87–93, 2014.

[78] Dirk Schulte, Andreas Henckels, and Ulrich Wepler. Reduction of shock induced boundary layer separation in hypersonic inlets using bleed. *Aerospace science and technology*, 2(4):231–239, 1998.

[79] Neil Titchener and Holger Babinsky. Shock wave/boundary-layer interaction control using a combination of vortex generators and bleed. *AIAA journal*, 51(5):1221–1233, 2013.

[80] Rajan Kumar, Mohd Y Ali, Farrukh S Alvi, and L Venkatakrishnan. Generation and control of oblique shocks using microjets. *AIAA journal*, 49(12):2751–2759, 2011.

[81] Mohd Yousuf Ali, Farukh Alvi, C Manisankar, S Verma, and L Venkatakrishnan. Studies on the control of shock wave-boundary layer interaction using steady microactuators. In *41st AIAA Fluid Dynamics Conference and Exhibit*, page 3425, 2011.

[82] Venkat Narayanaswamy, Laxminarayan L Raja, and Noel T Clemens. Control of a shock/boundary-layer interaction by using a pulsed-plasma jet actuator. *AIAA journal*, 50(1):246–249, 2012.

[83] D Caruana. Plasmas for aerodynamic control. *Plasma Physics and Controlled Fusion*, 52(12):124045, nov 2010.

[84] Miguel Visbal and Datta Gaitonde. *Control of Vortical Flows Using Simulated Plasma Actuators*. 2006.

[85] A Nazarian Shahrbabaki, M Bazazzadeh, and R Khoshkhoo. Investigation on supersonic flow control using nanosecond dielectric barrier discharge plasma actuators. *International Journal of Aerospace Engineering*, 2021(1):2047162, 2021.

[86] Thomas C Corke, C Lon Enloe, and Stephen P Wilkinson. Dielectric barrier discharge plasma actuators for flow control. *Annual review of fluid mechanics*, 42(1):505–529, 2010.

[87] Flint O Thomas, Thomas C Corke, Muhammad Iqbal, Alexey Kozlov, and David Schatzman. Optimization of dielectric barrier discharge plasma actuators for active aerodynamic flow control. *AIAA journal*, 47(9):2169–2178, 2009.

[88] James D'Entremont, Rohan M Gejji, Prashanth Venkatesh, and Sally P Bane. Plasma control of combustion instability in a lean direct injection gas turbine combustor. In *52nd Aerospace Sciences Meeting*, page 0622, 2014.

[89] Yun Wu and Yinghong Li. Plasma flow control. In *Encyclopedia of Plasma Technology-Two Volume Set*, pages 1016–1037. CRC Press, 2016.

[90] IV Adamovich, I Choi, N Jiang, JH Kim, S Keshav, WR Lempert, E Mintusov, M Nishihara, M Samimy, and M Uddi. Plasma assisted ignition and high-speed flow control: non-thermal and thermal effects. *Plasma Sources Science and Technology*, 18(3):034018, 2009.

[91] D Opaits, D Roupassov, Svetlana Starikovskaia, Andrei Starikovskii, Ivan Zavialov, and Seyed Saddoughi. Plasma control of boundary layer using low-temperature non-equilibrium plasma of gas discharge. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*, page 1180, 2005.

[92] VI Zvegintsev. Gas-dynamic problems in off-design operation of supersonic inlets. *Thermophysics and Aeromechanics*, 24:807–834, 2017.

[93] Skye Elliott, Mitsugu Hasegawa, Hirotaka Sakaue, and Sergey Leonov. Shock-dominated flow control by plasma array: Pressure analysis including pressure-sensitive paint visualization. *Experimental Thermal and Fluid Science*, 131:110522, 2022.

[94] Ciprian Dumitrache and Azer P Yalin. Gas dynamics and vorticity generation in laser-induced breakdown of air. *Optics express*, 28(4):5835–5850, 2020.

[95] Thomas McLaughlin, C Lon Enloe, and Robert VanDyken. Mechanisms and responses of a single dielectric barrier plasma. In *41st Aerospace Sciences Meeting and Exhibit*, page 1021, 2003.

[96] DV Roupassov, AA Nikipelov, MM Nudnova, and A Yu Starikovskii. Flow separation control by plasma actuator with nanosecond pulsed-periodic discharge. *AIAA journal*, 47(1):168–185, 2009.

[97] Edgar Caraballo, Nathan Webb, Jesse Little, Jin-Hwa Kim, and Mo Samimy. Supersonic inlet flow control using plasma actuators. In *47th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 924, 2009.

[98] Philip Andrews, Philip Lax, Skye Elliott, Alexander Firsov, and Sergey Leonov. Flow characterization at heated air supersonic facility sbr-50. *Fluids*, 7(5):168, 2022.

[99] Alec Houpt, Brock Hedlund, Sergey Leonov, Timothy Ombrello, and Campbell Carter. Quasi-dc electrical discharge characterization in a supersonic flow. *Experiments in Fluids*, 58:1–17, 2017.

[100] Ciprian Dumitrache. *Novel laser ignition technique using dual-pulse pre-ionization*. PhD thesis, Colorado State University, 2017.

[101] MH Morsy and SH Chung. Numerical simulation of front lobe formation in laser-induced spark ignition of ch4/air mixtures. *Proceedings of the Combustion Institute*, 29(2):1613–1619, 2002.

[102] Maria Castela, Sergey Stepanyan, Benoit Fiorina, Axel Coussement, Olivier Gicquel, Nasser Darabiha, and Christophe O Laux. A 3-d dns and experimental study of the effect of the recirculating flow pattern inside a reactive kernel produced by nanosecond plasma discharges in a methane-air mixture. *Proceedings of the Combustion Institute*, 36(3):4095–4103, 2017.

[103] Eleuterio F Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

[104] Ronald P Fedkiw, Barry Merriman, and Stanley Osher. High accuracy numerical methods for thermally perfect gas flows with chemistry. *Journal of Computational Physics*, 132(2):175–190, 1997.

[105] EW Lemmon, MO McLinden, DG Friend, P Linstrom, and W Mallard. Nist chemistry webbook. *NIST standard reference database*, (69):20899, 2011.

[106] PL Roe and J Pike. Efficient construction and utilisation of approximate riemann solutions. In *Proc. of the sixth int'l. symposium on Computing methods in applied sciences and engineering, VI*, pages 499–518, 1985.

[107] Jack R Edwards. A low-diffusion flux-splitting scheme for navier-stokes calculations. *Computers & Fluids*, 26(6):635–659, 1997.

[108] John Charles Butcher and Gerhard Wanner. Runge-kutta methods: some historical notes. *Applied Numerical Mathematics*, 22(1-3):113–151, 1996.

[109] Eleuterio F Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. 01 1999.

[110] Randall J LeVeque and Randall J Leveque. *Numerical methods for conservation laws*, volume 214. Springer, 1992.

[111] Gilbert Strang. On the construction and comparison of difference schemes. *SIAM journal on numerical analysis*, 5(3):506–517, 1968.

[112] Sergei K Godunov and I Bohachevsky. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematičeskij sbornik*, 47(3):271–306, 1959.

[113] Philip L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.

[114] Harvard Lomax, Thomas H Pulliam, David W Zingg, and TA Kowalewski. Fundamentals of computational fluid dynamics. *Appl. Mech. Rev.*, 55(4):B61–B61, 2002.

[115] Le Duan, Xiaowen Wang, and Xiaolin Zhong. A high-order cut-cell method for numerical simulation of hypersonic boundary-layer instability with surface roughness. *Journal of computational physics*, 229(19):7207–7237, 2010.

[116] Elijah D House and Ciprian Dumitrache. Numerical modeling of plasma-assisted shock control in supersonic flow. In *AIAA AVIATION FORUM AND ASCEND 2024*, page 4594, 2024.

[117] T J Poinsot and SK Lelef. Boundary conditions for direct simulations of compressible viscous flows. *Journal of computational physics*, 101(1):104–129, 1992.

[118] Gary A Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of computational physics*, 27(1):1–31, 1978.

[119] Andrey Nikolaevich Kolmogorov. A refinement of previous hypotheses concerning the local structure of turbulence in a viscous incompressible fluid at high reynolds number. *Journal of Fluid Mechanics*, 13(1):82–85, 1962.

[120] Patrick J Roache. *Verification and validation in computational science and engineering*, volume 895. Hermosa Albuquerque, NM, 1998.

[121] MH Farris and CT Russell. Determining the standoff distance of the bow shock: Mach number dependence and use of models. *Journal of Geophysical Research: Space Physics*, 99(A9):17681–17689, 1994.

[122] Michael Aftosmis, Marsha Berger, and Gedas Adomavicius. A parallel multilevel method for adaptively refined cartesian grids with embedded boundaries. In *38th Aerospace Sciences Meeting and Exhibit*, page 808, 2000.

[123] Weiqun Zhang, Ann Almgren, Vince Beckner, John Bell, Johannes Blaschke, Cy Chan, Marcus Day, Brian Friesen, Kevin Gott, Daniel Graves, et al. Amrex: a framework for block-structured adaptive mesh refinement. *The Journal of Open Source Software*, 4(37):1370, 2019.

[124] Vincent Moureau, Pascale Domingo, and Luc Vervisch. Design of a massively parallel cfd code for complex geometries. *Comptes rendus. Mécanique*, 339(2-3):141–148, 2011.

[125] PK Senecal, KJ Richards, E Pomraning, T Yang, MZ Dai, RM McDavid, MA Patterson, S Hou, and T Shethaji. A new parallel cut-cell cartesian cfd code for rapid grid generation applied to in-cylinder diesel engine simulations. Technical report, SAE Technical Paper, 2007.

[126] Alo Roosing, OT Strickson, and Nikos Nikiforakis. Fast distance fields for fluid dynamics mesh generation on graphics hardware. *arXiv preprint arXiv:1903.00353*, 2019.

[127] Mozhdeh Hooshyar and Ciprian Dumitrache. Computational modeling of a femtosecond-initiated continuous optical discharge in air. In *AIAA AVIATION FORUM AND ASCEND 2024*, page 3901, 2024.

Appendix A

Newton-Raphson Method

The Newton-Raphson method is an iterative technique for finding the roots of a nonlinear equation. It is based on the idea of linearizing the function around an initial guess and then updating the guess based on the linear approximation. The general form of the Newton-Raphson method is given by the following equation [114]:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{A.1}$$

where x_n is the current guess, x_{n+1} is the updated guess, $f(x_n)$ is the function value at the current guess, and $f'(x_n)$ is the derivative of the function at the current guess. The method is repeated until the difference between successive guesses is below a specified tolerance. In the context of solving the system of equations in the APDL-CFD code, the Newton-Raphson method is used to update the conservative variables at each cell based on the fluxes or residuals computed in the RKIV method. The method is applied to (3.11) to update the temperature field in the code. The general form of the equation is given by:

$$T_{n+1} = T_n - \frac{f(T_n)}{f'(T_n)} \tag{A.2}$$

where

$$f(T) = T - C_1 - C_2 h(T). \tag{A.3}$$

Appendix B

Future Work Plasma Model

To get a better understanding of the plasma effects on the flow, a more detailed plasma model can be used. The general form of the equations is given by:

$$U_t + F(U)^a_x + G(u)^a_y = F(U)^d_x + G(U)^d_y + H(U) \tag{B.1}$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \\ E_v \\ E_e \\ \rho Y_i \end{bmatrix}, \qquad F(U)^a = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \\ uE_v \\ F_e \\ \rho u Y_i \end{bmatrix}, \qquad G(U)^a = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \\ vE_v \\ F_e \\ \rho v Y_i \end{bmatrix} \tag{B.2}$$

$$F(U)^d = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \\ 0 \\ -q_{ex} \\ 0 \end{bmatrix}, \quad G(U)^d = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - q_y \\ 0 \\ -q_{ey} \\ 0 \end{bmatrix} \tag{B.3}$$

$$H(U) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ Q_{vT} + Q_{eT} + Q_R \\ Q_{eV} - Q_{vT} \\ J_L - Q_{eV} - Q_{eT} - Q_i - Q_{ex} \\ G_i - L_i \end{bmatrix} \tag{B.4}$$

The terms in red are the added terms to the equations. More information is found in the work by Hooshyar [127].