DESIGN AND MULTIFIDELITY ANALYSIS OF DUAL MODE SCRAMJET COMPRESSION SYSTEM USING COUPLED NPSS AND FLUENT SIMULATION

by

NANDAKUMAR VIJAYAKUMAR

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN AEROSPACE ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2014

Acknowledgements

It was a great priviledge to be supervised by Dr. Donald Wilson and I would like to express my sincere gratitude for his invaluable guidance, immense patience, constant encouragement and support for my thesis. I am honoured to work for Dr. Frank Lu and grateful for his insights and vast knowledge which added lots of credits to address this research. I feel fortunate and could not imagine better supervising advisors to provide such a possible support. I would also like to thank Dr. Dora Musielak for her valuable interest, support and encouragement in my thesis, which helped me to gain a lot of confidence. A special thanks to Dr. Brian Dennis for his assistance in being a part of the thesis defense committee.

I would like to appreciate the inspitation and motivation of my friends Razina Zainudeen, Pengkai Ru and Siddarth Chinthamani who never failed to lift me during my hard times. I would like to include special thanks to Mrs. Debi Barton for her unconditional support within the department and making me feel as a part of her happy family. I would like to convey a note of thanks to my fellow ARCians for making the research competitive and enjoyable.

Most importantly, it would have been impossible to persive through my master’s without the love and support of my family. A very special acknowledgement goes to Monisha Mohan, for her forbearance and encouragement during the final, critical months of my thesis. Thanks to my aunt Rekha and her family for spending their valuable time in making me feel comfortable in USA.

April 17, 2014

Abstract

DESIGN AND MULTIFIDELITY ANALYSIS OF DUAL MODE SCRAMJET COMPRESSION SYSTEM USING COUPLED NPSS AND FLUENT SIMULATION

Nandakumar Vijayakumar, M.S.
The University of Texas at Arlington, 2014

Supervising Professor: Donald. R. Wilson and Frank. K. Lu

Hypersonic airbreathing engines mark a potential future development of the aerospace industry and immense efforts have been taken in gaining knowledge in them for the past decades .The physical phenomenon occurring at the hypersonic flow regime makes the design and performance prediction of a scramjet engine hard. Though cutting-edge simulation tools fight their way toward accurate prediction of the environment, the time consumed by the entire process in designing and analyzing a scramjet engine and its component may be exorbitant. A multi-fidelity approach for designing a scramjet with a cruising Mach number of 6 is detailed in this research where high-order simulations are applied according to the physics involved in the component. Two state-of-the-art simulation tools were used to take the aerodynamic and propulsion disciplines into account for realistic prediction of the individual components as well as the entire scramjet. The specific goal of this research is to create a virtual environment to design and analyze a hypersonic, two-dimensional, planar inlet and isolator to check its operability for a dual-mode scramjet engine. The dual mode scramjet engine starts at a Mach number of 3.5 where it operates as a ramjet and accelerates to Mach 6 to be operated as a scramjet engine. The intercomponent interaction between the compression components with the rest of the engine is studied by varying the fidelity of the numerical simulation according to the complexity of the situation. Efforts have been taken to track the transition Mach number as it switches from ramjet to scramjet. A complete scramjet assembly was built using the Numerical Propulsion Simulation System (NPSS) and the performance of the engine was evaluated for various scenarios. Different numerical techniques were opted for varying the fidelity of the analysis with the highest fidelity consisting of 2D RANS CFD simulation. The interaction between the NPSS elements with the CFD solver is governed by the top-level assembly solver of NPSS. The importance of intercomponent interactions are discussed.The methodology used in this research for design and analysis, should add up to provide an efficient way for estimating the design and off-design operating modes of a dual mode scramjet engine.

Table of Contents

List of Illustrations

List of Tables

List of Symbols and Abbreviations

A_i	Area at Station i
$\mathrm{A}.S$	Properties after a Shockwave
$\mathrm{B}.S$	Properties before a Shockwave
C_f	Skin Friction Coefficient
C_p	Specific Heat at Constant Pressure
D	Distance Perpendicular from the x-y Plane, Equivalent to z
$\mathrm{D_H}$	Hydraulic Diameter
f	Fuel-to-Air Ratio
f_{st}	Stoichiometric Fuel-to-Air Ratio
H	Distance Perpendicular from the Axial Direction, Equivalent to y
$\mathrm{h_{PR}}$	Heat Of Reaction
H	Height
IT	Ignition Temperature
L	Length
M_i	Mach Number at Station i
$\dot{m}$	Mass Flow Rate
NST	Normal Shock Train
OST	Oblique Shock Train
P_i	Static Pressure at Station i

P_{ti}	Total Pressure at Station i
R	Perfect Gas Constant
Re	Reynolds Number
$(S_i$-$S_0)/\,C_p$	Dimensionless Entropy Increase Between Station i and the Inlet
Sa_i	Stream Thrust Function at Station i
S	Specific Fuel Consumption
$S.W$	Shockwave
T_i	Static Temperature at Station i
T_0	Reference Temperature
T_{ti}	Stagnation Temperature at Station i
u	Axial Veloticy along x Direction
V_i	Axial Velocity at Station i
X	Axial Location
Y	Distance Perpendicular from the Axial Direction
Z	Distance Perpendicular from the x-y Plane
π_c	Total Pressure Ratio
θ	Empirical Constant Based on Mode of Fuel Injection and Mixing
β	Oblique Shockwave angle
η_b	Burner Efficiency
η_c	Inlet Compression System Efficiency
η_e	Expansion System Efficiency
η_{KE}	Kinetic Energy Efficiency

η_o	Overall Efficiency
η_p	Propulsive Efficiency
η_{th}	Thermal Efficiency
$\emptyset$	Fuel Equivalence Ratio
δ	Turn Angle of the flow
θ/H	Ratio of Boundary Layer Momentum Thickness to Duct Height
ρ_i	Density at station i
ψ	Ratio of Burner Entry Temperature to Freestream Temperature
γ	Ratio of Specific Heat

Subscripts

b	Burner Component
c	Compression Component
e	Expansion Component
i	Station Designation

Chapter 1

Introduction

This chapter describes the basic cycle for the operation of the ramjet, scramjet and the dual-mode scramjet and these engines' applications.

1.1 Ramjet and Scramjet Engine Operation

The maximum cycle temperature is one of the most important factors for efficient operation of an airbreathing engine. Above Mach 3, the compression of air by rotatory turbine machinery becomes inefficient due to the enormous amount of heat produced in the process as well as the limitations fixed by the maximum temperature accepted by the turbine [1]. At such a high Mach number, compression of the incoming flow can be achieved by changing the area within the inlet and diffuser to attain suitable pressure to be fed into the combustion chamber and at this point the ramjet engine takes over. The ramjet is a remarkable invention in airbreathing engines, attributed to René Lorin of France in 1913 [1]. The incoming flow is initially compressed by one or more oblique shock waves generated by the forebody of the vehicle or of the diffuser, transforming the supersonic flow to a subsonic one through a terminal normal shockwave and further decelerating the flow by a divergent duct [2]. The combustion is initiated by fuel addition and the hot pressurized subsonic flow accelerates back to supersonic flow with the aid of a convergent-divergent nozzle and exits into the atmosphere. The thrust is produced because the high temperature exhaust flow has more velocity and momentum than the incoming flow.

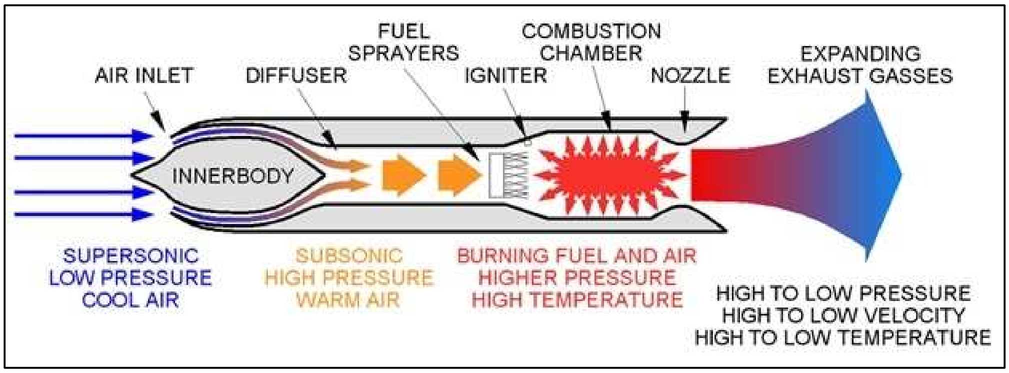

FUEL SPRAYERS
COMBUSTION CHAMBER
AIR INLET
DIFFUSER
IGNITER
NOZZLE
EXPANDING EXHAUST GASSES
INNERBODY
SUPERSONIC LOW PRESSURE COOL AIR
SUBSONIC HIGH PRESSURE WARM AIR
BURNING FUEL AND AIR HIGHER PRESSURE HIGH TEMPERATURE
HIGH TO LOW PRESSURE HIGH TO LOW VELOCITY HIGH TO LOW TEMPERATURE

Figure 1-1 Two-Dimensional Schematic Diagram of a Ramjet Engine [2].

The conventional rule of thumb defines that the hypersonic flow regime to be flow at a Mach number higher than 5 [3]. At hypersonic velocity, the reduction in the kinetic energy due to the deceleration of the flow within the engine flow path is converted into internal energy as proposed by the conservation of energy [2]. When the flight Mach number reaches around 6, certain physical phenomena such as excessive performance losses due to the normal shock wave system, high pressure for practical structural design, excessive wall heat transfer rates and loses due to dissociation occur which makes deceleration of the flow to subsonic speed inefficient [2]. Further acceleration beyond this point can become feasible by replacing the ramjet by a supersonic combustion ramjet or scramjet. In a scramjet, the flow is supersonic throughout the engine flow path including the combustion chamber due to the absence of the diffuser as well as the convergent divergent nozzle. In order to achieve stable combustion, the mixing between the fuel and air should be quick with most of it within a reasonable length of the combustion chamber.

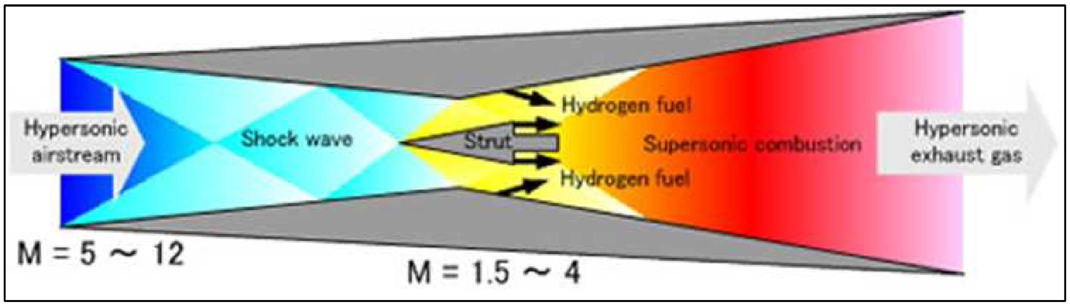


Figure 1-2 Two-Dimensional Schematic Diagram of a Scramjet Engine [2].

1.2 Dual-Mode Scramjet Operation

The conflicting requirements of high cycle thermal efficiency and the occurrence of dissociation of the working fluid at excessively high static temperature dictate that the combustion process must be subsonic for a flight Mach number less than about 5 and supersonic for greater than about 7 [2]. The ramjet engine has two throats in its flow path, one is the diffuser at the end of the inlet to stabilize the terminal shock and the other is the convergent- divergent nozzle for accelerating the flow to supersonic velocity after subsonic combustion. Around Mach 6, a normal shock will be formed at the lip of the inlet which leads to a poor total pressure recovery through the inlet [4]. To avoid carrying two different engines for ramjet and scramjet operation, it would be desirable to operate the engine duct in either ramjet or scramjet mode using only the no-throat geometry of the scramjet [2]. The possibility of dual-mode combustion was first proposed by Curran and Stull in 1963 [2]. The components of a dual-mode scramjet or DMScramjet are similar to that of a pure scramjet with an addition of a constant area duct (isolator) in front of the combustion chamber. The isolator plays a vital role of containing the precombustion shock train when the pressure in the combustion chamber exceeds the pressure produced by the compression of the inlet. The DMScramjet operates in three modes, namely, shock-free scramjet mode, ramjet mode and oblique shock train scramjet mode.

The mode in which the engine operates is influenced by the flight conditions, inlet compression or inlet exit Mach number, overall engine fuel/air ratio, and supersonic combustor area ratio [5]. In the ramjet mode, the flow is thermally choked within the combustion chamber raising the pressure at the entry of the combustion chamber which sets a normal shock train in the isolator [2]. The ramjet mode occurs in a low flight Mach number with high fuel to air ratio and the combustion process is similar to that of a conventional subsonic combustion ramjet, however, the flow accelerates to a sonic or supersonic speed before exiting the diverging area combustor, eliminating the requirement for a geometric throat [5]. A variable area combustion chamber is required in order to get the relevant combination of area and temperature distribution to choke the flow.

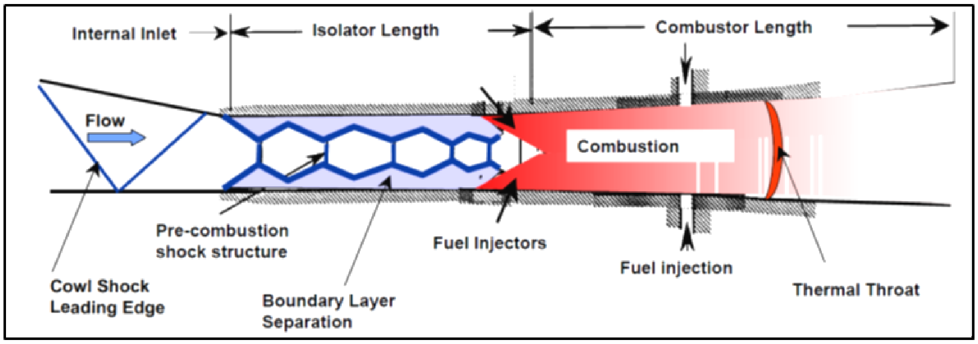


Figure 1-3 Schematic of a Ramjet Mode Operation of a DMScramjet Engine [11].

The oblique shock train scramjet mode is caused by a sudden increase in the pressure due to fuel combustion which would separate the boundary layer along the adjacent wall. It creates a confined core flow in the isolator which results in a weak oblique shock train. Usually an engine operates in this mode in a moderate flight Mach number around or above Mach 5 [5].

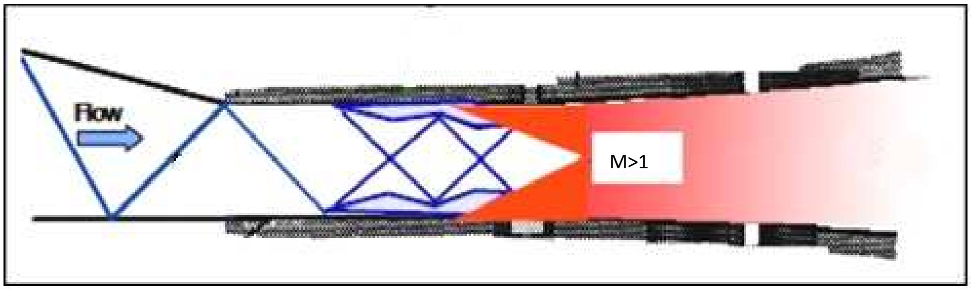


Figure 1-4 Schematic of a Scramjet Mode Operation of with Oblique Shock Train in a DMScramjet Engine [11].

In the shock-free mode, the isolator is free of a shock train since the pressure raise in the combustion chamber is relieved by the area increase in the combustion chamber. The shock free isolator mode usually occurs at high flight Mach number and low fuel/air ratio. But the isolator may contain reflected shocks from the inlet which are not cancelled at the shoulder of the inlet throat without the separation of boundary layer. The reflected shocks travel throughout the flow path of the engine leading to additional distortions.

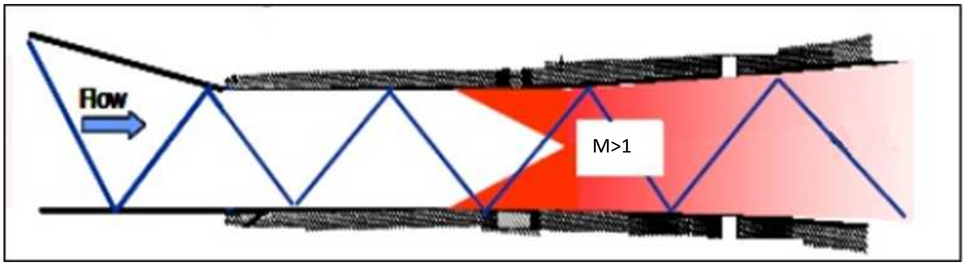


Figure 1-5 Schematic of Shocktrain Free Mode in a DMScramjet Engine [11].

All the three modes of operation of the DMScramjet are investigated in the current research work and will be explained in the respective sections of this paper.

1.3 Application of DMScramjet Engines

The type of fuel selected for a scramjet decides on its application . In general, either hydrogen or hydrocarbon fuel can be used. Possible applications for scramjets include hypersonic cruise vehicles, hypervelocity missiles, and airbreathing boosters for space applications [5]. Hydrogen fuels are generally considered for high-speed engines for access to space, whereas hydrocarbon fuels are the general choice for potential military and commercial aerospace applications operating for flight Mach numbers up to 6 [5]. Making use of the enhanced cooling capabilities of endothermic hydrocarbon fuels can increase the maximum speed for hydrocarbon-fueled missiles and vehicles upto Mach 8.

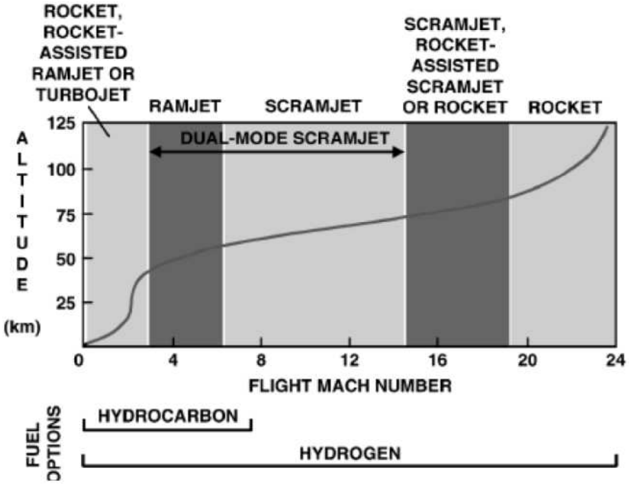


Figure 1-6 Mach Number Operating Envelope for Different Propulsion Systems [5].

From figure 1.6, we can see that for a turbine-based combined cycle engine (TBCC), the turbojet engine can initiate the propulsion up to around Mach 3-4 and then transition to a DMScramjet engine. The upper limit of the DMScramjet is near Mach 8 with a transition from ramjet mode to scramjet mode roughly around Mach 6.

This research aims at obtaining a similar TBCC with a cruising Mach number of 6 where the focus is only on developing the DMScramjet part of the engine. The following chapters provide an insight on its progress towards the goal.

Chapter 2

Goals and Background Survey

This chapter explains the purpose and objectives which are aimed to be achieved at the end of this research.

2.1 Motivation

There has been persistent interest in the field of propulsion to create a “virtual test cell” for multidisciplinary simulation to cut down the cost of large-scale hardware tests [7]. When simulating a complex flow field, especially in the hypersonic regime, multiple disciplines are involved in the process. Though high fidelity simulation is required to capture the entire physics occurring in the domain of the system, fast and simple results are essential for preliminary analysis of the system which ensures its possibility and identifies the problems one can expect during the subsequent, detailed design process. The recent trend has been to apply advanced two- and three-dimensional numerical techniques to engine components, to understand the details of their operation in isolation. But these advanced component analysis techniques often do not systematically account for inter-component interactions [6]. By multistage isolated analysis, eventually the interactions that occur between components can be masked by the limited communication between teams or the codes which perform the individual numerical analysis [8]. This can pose serious problems for today’s highly integrated propulsion systems where multidisciplinary issues can adversely impact the overall system performance [8]. Usually, these problems are found late in the design process when the initial simulations are tested under realistic operating conditions and if several design-test cycles occur, the new product is usually behind schedule and over-budget [8]. This motivates the objective of this research which describes the efforts directed at creating a system which allows individual component models to be coupled to create a full engine simulation. An appropriate simulation tool should be used in order to take most of the phenomena occurring within the domain into account, to provide reasonable results and should be decided based on individual component analysis. The preliminary goal of this research, as mentioned earlier, is to design and analyze a compression system for a dual mode scramjet engine which has its best efficiency at flight Mach number of 6. The scramjet compression system is one of the regions that involves complex fluid flow phenomena such as shock-boundary layer interaction; boundary layer separation bubbles, shock-shock interaction and inlet unstart. Thus it is often desirable to use a high-fidelity fluid simulation for the inlet and the isolator parts. This clearly sketches the overall objective of this research which would be to design and simulate a generic scramjet inlet and isolator, using Numerical Propulsion System Simulation (NPSS) coupled with two-dimensional computational fluid dynamics (CFD) code and to merge the combustion chamber and exhaust nozzle components, which will be created as a part of this research, to calculate a realistic overall performance prediction of a DMScramjet engine.

2.2 Literature Survey on DMScramjets

The development of scramjets has been explained in detail by many authors in their respective work [2, 5, and 9]. Only works that are relevant to the current research are elaborated in this section. From the 1950s through today, many programs have had the objective of developing and demonstrating hydrogen and hydrocarbon-fueled scramjet propulsion [2]. The key to using the DMScramjet at and above Mach 6 even though a conventional ramjet (CRJ) can operate successfully at the same range was absorbed from the work of Dugger on the relative performance of a kerosene-fueled CRJ which showed that the scramjet’s performance would exceed the performance of the CRJ in the speed range of Mach 6 to 8 [2]. The National Aero-Space Plane (NASP) program which was one of the most influential programs in modern scramjet development was established in 1986 to develop and fly a synergistically integrated low-speed accelerator, ramjet and scramjet propulsion system which operated with hydrogen fuel. NASA modified the original engine design from the NASP program and to develop the power plant of the successful X-43A vehicle that flew at Mach 7 (5,000 miles/hour) in March 2004 as part of the Hyper-X program [10,11]. From then, there has been an active research in employing hydrocarbon fueled scramjet engines. The U.S. Air Force/NASA and Pratt & Whitney ground tested the first uncooled hydrocarbon-fueled scramjet engine at simulated flight Mach numbers of 4.5 and 6.5 [11,12]. Further development of this engine led to the ground demonstration of a liquid JP7 hydrocarbon-fueled scramjet, which was constructed from flight-weight (nickel-based alloys) fuel-cooled structures with the potential for satisfying requirements of future operational engines such as for powering missiles, aircraft, and access-to-space vehicles at sustained hypersonic speeds [11,13]. The Defense Advanced Research Projects Agency (DARPA)/U.S. Navy and Boeing/Aerojet/JHU have also ground demonstrated a JP10 hydrocarbon-fueled dual combustion ramjet, which was constructed from non-flight weight materials (primarily nickel alloys) and intended exclusively for hypersonic missiles [11, 14]. The recent proposal by Lockheed Martin and Aerojet-Rocketdyne for the SR 72 was an important inspiration in designing a DMScramjet engine to cruise at Mach 6 and be able to take off and land from a runway [15]. The SR 72 is planned to be a TBCC with two engines and around a length of 100 feet. It is planned to use a turbine engine to ramjet/scramjet engine takeover speed at Mach 3-3.5 with thrust augmentation playing a vital role in pushing the upper limits of the turbine engines [15]. Having discussed enough about the scramjet engines which has been referred in this research, the following text explains the tools selected to achieve the objective of this research and discusses about the previous work done regarding the multi fidelity simulations.

2.3 Numerical Propulsion System Simulation

The Numerical Propulsion System Simulation (NPSS) code was created through a joint effort by the gas turbine industry, NASA and universities to develop a state-of-the-art aircraft engine cycle analysis simulation tool [16]. The vision of NPSS is to create a "numerical test cell" enabling full engine simulations overnight on cost-effective computing platforms [17]. Written in the computer language C++, NPSS is an object-oriented framework allowing the gas turbine engine analyst considerable flexibility in cycle conceptual design and performance estimation [16]. It is capable of performing multi-disciplinary simulations by combining the main stream propulsion calculations along with aerodynamics, fluid dynamics, structures and heat transfer. Furthermore, the tool was written with the assumption that most users would desire to easily add their own unique objects and calculations without the burden of modifying the source code [16]. It facilitates both steady state as well as transient simulations. A key feature of NPSS is the ability to link tools at various levels of analysis detail, or multi-fidelity, thus allowing the user to transition from traditional 0-dimensional engine cycle analyses to 1-dimensional mean line design tools and 3-dimensional CFD tools. This feature is named “zooming” and provides several advantages in design and development of an engine [21]. It enables a potential component design to be more fully and rapidly evaluated in the context of an engine system which provides accurate component boundary conditions (pressures, temperatures, flows, etc.) and an integrated result to the component designer [21]. This improves the fidelity of the result from current practice since the impact of the component on the system is accounted for in the analysis. The engine system simulation becomes more predictive since component performance characteristics can be based on physics-based, 1st principle analysis codes and increases design confidence prior to commitment to hardware [21]. It also permits the resolution and fidelity of the engine model to be tailored to match the analysis requirements. Zooming reduces the computing resource requirements since high fidelity analysis is applied only to the components of interest where the increased fidelity and/or resolution are required to support a particular analysis [21]. This flexibility allows the user to vary the detail of analysis depending upon the nature of the design problem and the availability of validated analysis tools [18]. The zooming feature plays a vital role in this research. The ability to communicate with other codes makes NPSS unique for a user who wishes to perform a multifidelity or multi-disciplinary simulation.

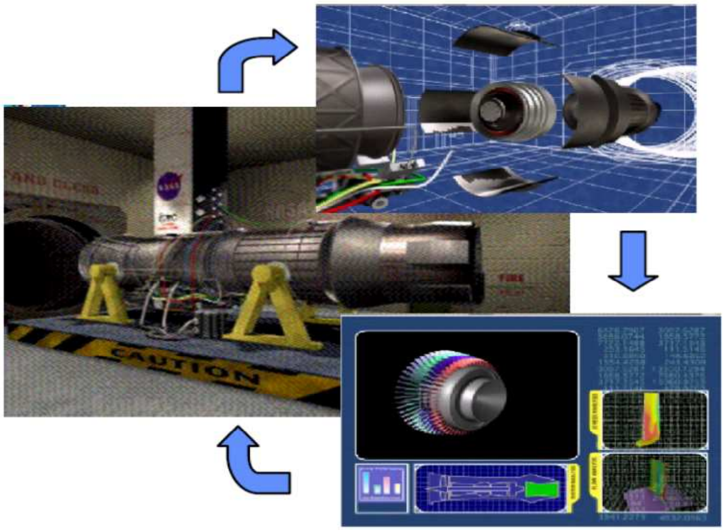


Figure 2-1 Different Levels of Component and Sub Component Based Simulation in NPSS [18].

2.3.1 Survey of Previous Work in NPSS

The features of NPSS can be categorized into three divisions namely engineering application models, simulation environment and high performance computing environment [19]. The current research falls under the first category which focuses on component integration to achieve large subsystem and system simulations, to perform multidisciplinary coupling to capture critical interactions among the disciplines and to obtain multifidelity analysis tailored to the complexity of the analysis under study [19]. Similar work done earlier which falls under this subdivision is mentioned in this section.

The National Cycle Program (NCP) using the NPSS environment was the preliminary version of NPSS [17]. NCP marked the first step toward NPSS and is its initial framework supported the aero thermodynamic system simulation process for the full life cycle of an engine, which added to a $50 million/year savings to industry through improved engineering productivity [17]. An immediate research using the NPSS was conducted to demonstrate the capability to analyze the aerodynamic performance of the complete low pressure subsystem (LPS) of the Energy Efficient Engine (EEE) [6]. The analysis evaluated the impact of steady aerodynamic interaction effects between the components of the LPS at design and off-design operating conditions. A 2D aerodynamic simulation was performed using a Navier Stokes solver called ADPAC and results from the CFD run were coupled to NPSS were the rest of the engine was analyzed. The highlight of this analysis was the knowledge of component aero/mechanical interactions gained from simulation runs which were unknown till the hardware testing before the advent of NPSS. A successful multifidelity zooming capacity of NPSS was demonstrated a couple of years later by integrating a zero-dimensional engine analysis code with a one dimensional mean line compressor analysis code [21]. This research carved the pathway to various levels of fidelity analysis using NPSS zooming feature. A benchmark research using NPSS was performed for a high fidelity, three-dimensional simulation of the GE90 [20]. A full engine, three-dimensional flow analysis was achieved as a result of this research, where the compressor and turbine parts were simulated using the APNASA turbo machinery flow code and the combustor component was simulated using National Combustor Code [20]. It is a breakthrough to analyze a complete turbine engine in three-dimensional space from scratch with components near the cycle conditions within 10 hours and 20 minutes as reported [20]. Another successful NPSS zooming was performed for a turbine engine with three levels of fidelity, zero-dimensional calculations carried out by NPSS, one-dimensional calculations are taken care by mean line analysis codes namely STGSTK and BRSTK for compression systems and AXOD for turbine components [8]. A CFD code named APNASA resulted in a detailed, three-dimensional analysis and the scripting language in NPSS was used to automate the communication between the codes [8].

The research works mentioned in the above sections provides some valuable insights on the path to be taken in order to achieve the goals of the current research.

Chapter 3

Theory and Simulation Setup

3.1 Role of a Scramjet Inlet

Having said that one of the preliminary goals of this research is to design a compression system for a DMScramjet which could start at a Mach number of 3.5 and reach a cruising Mach number of 6, the next question is that on what basis is the inlet going to be selected and designed. Scramjet inlets are a critical component and their design has important effects on the overall performance of the engine [23, 24]. The inlet plays the role of a vital component in achieving the possibility of the required mission, especially in the DMScramjet engine [15]. The performance of such inlet can be separated into two key parameters; capability, or how much compression is performed, and the efficiency of compression [23].

3.1.1 Level of Compression for a Scramjet Inlet

Choosing the level of compression to be performed is a critical decision. Too much compression can place extra system constraints on the inlet such as the need for variable geometry or boundary layer bleed, leading to large losses and external drag [23]. Overall efficiency of a scramjet engine has a great influence by the engine thermodynamic cycle efficiency, which in turn is controlled by the maximum cycle static temperature ratio. A higher static temperature ratio results in higher thermodynamic efficiency. The main criterion which should be satisfied during the inlet design is to meet the minimum auto ignition temperature of the fuel injected at the end of the compression system. Unsatisfactory temperature profile may result in inadequate ignition and combustion of fuel. Prior knowledge about the combustion chamber and the fuel to be used is of high importance. On the other hand, the static temperature entering the combustion chamber cannot be increased indefinitely, but must be limited to a value that prevents excessive dissociation of the exhaust flow which gradually destroys the benefit of higher thermodynamic cycle efficiency [2]. At or near the stoichiometric ratio, the maximum allowable compression temperature is always found to be in a narrow range of 1440 – 1670 K to prevent the dissociation of oxygen [2, 25]. The level of compression is also limited to, in terms of pressure, within the allowable range of 0.5 to 10 atm (50 to 1000 kPa) up to Mach 6 [2, 25]. The lower limit of this range is to support combustion and the upper limit is constrained by the structural and weight limits of the vehicle [25]. Maintaining a high vehicle dynamic pressure is required for good compression efficiency and cycle temperature ratio [2]. A constant flight dynamic pressure trajectory of 47,880 Pa was selected and calculated with the help of the Trajectory program available in the Heiser and Pratt software “HAP” (Hypersonic Airbreathing Propulsion) for this research [2, 25].

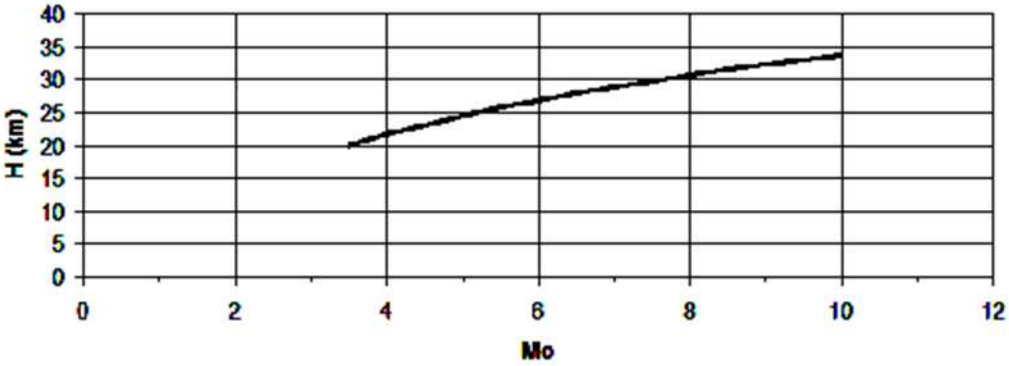


Figure 3-1 Constant Dynamic Pressure (=47,880 N/   ) Trajectory for Scramjet [2, 25]

For a DMScramjet engine, one important condition is that during the scramjet mode, the Mach number entering the combustion chamber should be high enough to sustain supersonic combustion [2]. Inlet "unstart" and shock/boundary layer separation are also to be considered with equal importance. During inlet unstart, a normal shock forms near the lip of the inlet and adjusts the flow to become subsonic when entering the combustor, making supersonic combustion impossible. For a TBCC, having good starting characteristics at ramjet/scramjet take-over speeds (Mach 4 to 5) and operating over a large Mach number range and flight conditions is important for successful ground takeoff and landing [15, 25]. The adiabatic compression efficiency of the inlet depends upon the total pressure ratio of the compression [2]. The total pressure loss occurs due to oblique shocks and should be handled carefully while selecting the turning angle of the inlet. Throughout this research, the station reference numbers are in convention with the Heiser and Pratt referencing system [2].

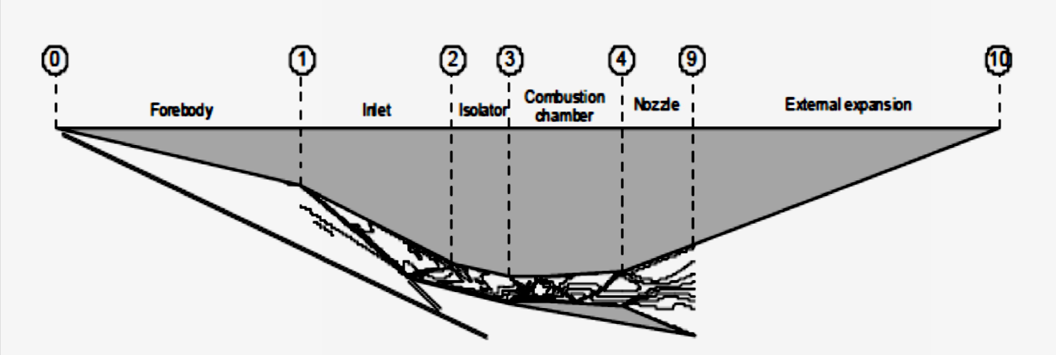


Figure 3-2 Scramjet Station Referencing System [2, 23]

3.1.2 Types and Previous Work on Scramjet Inlets

Scramjet compression systems can be broadly categorized into three sub divisions; namely external compression system, mixed compression system and internal compression system [2]. As the name suggests, the external compression system compresses the incoming flow with oblique shock waves that are external to the engine.

These inlet configurations have large cowl drag, as the flow entering the combustor is at a large angle relative to the free stream flow; however, external compression inlets are self-starting and spill flow when operated below the design Mach number [23]. In mixed compression system, the compression is divided between the shocks formed by the external surface of the inlet and the reflected shock inside the cowl region of the engine [2]. These inlets are typically longer than external compression configurations, but they also spill flow when operated below the design Mach number [23]. The angle of the external cowl relative to the free stream can be made very small to minimize external drag depending on the level of internal compression. However, mixed compression inlets may need variable geometry in order to start [23]. In internal compression systems, shockwaves are necessary to compress the free stream air and are constrained to be kept internal to the engine. This type of inlet can be shorter than a mixed compression inlet, but it does not allow easy integration with a vehicle [23]. It maintains full capture at Mach numbers lower than the design point, but its most significant limitation is that extensive variable geometry is always required for it to start [23]. In some inlet designs, the shockwaves that perform the compression may be completely or partially replaced with isentropic compression waves, but they still fall under either of the three categories. A lot of knowledge about the previous work done in hypersonic inlets with their merits and demerits can be gained by referring to the works of Bussey [24], Smart [23] and Molina [26] .The conclusions drawn from these papers are mentioned in this section to provide information on selection of the inlet configuration for the current research. In the research published by Molina [26], a baseline rectangular inlet was compared with two innovative inlets and a jaws and a scoop inlet for a hydrocarbon-fueled engine, and their advantages over one another were discussed. From that research, it can be learned that flow field distortions that occur due to shock boundary layer interactions in the innovative inlets are similar to those found in the baseline rectangular inlet [26]. The average temperature at the exit plane of the innovative inlets is lower relative to the rectangular inlet [26]. Both innovative inlets (jaws and scoop) underperform during off-design conditions when compared to the rectangular configuration [26]. The innovative inlets had their own advantages over the baseline rectangular inlet in ways such as high mass flow capture, on-design performance of the jaws inlet showing higher efficiency and less pressure losses etc. which lead to further investigation of the advantages of the different types of inlets without conclusion. Bussey [24] stated that rectangular or two-dimensional inlets have reduced boundary layer losses because of their shorter lengths. In Smart [23], five successful scramjet engines were studied in which three of them were mixed compression, fixed geometry inlets. The results from the above mentioned research work made the choice clear; which was to design a mixed compression fixed geometry inlet with a regular, two-dimensional rectangular geometry because of its advantage in good structural strength and multiple air vehicle integrations and design simplicity [26].

3.2 Theory of DMScramjet Combustion and Expansion Unit

The supersonic combustion chamber is subjected to large variations in the flow environment when installed in a DMScramjet. The combustion chamber of a scramjet engine must be able to integrate with the other flow-path components such as the inlet and the nozzle over a broad range of pressure, temperature, and flow-regime conditions [1]. At the low range of the flight regime, the heat deposition in the combustion chamber is relatively large compared with the incoming flow energy; hence the heat deposition substantially reduces the air speed and a large pressure rise is experienced with possible flow separations leading to normal shock and oblique shock trains [1]. When the flight Mach number increases significantly, the heat addition in the combustion chamber is less when compared to the gas enthalpy [1]. Two types of combustion chambers are common in hypersonic airbreathing engines namely constant area and constant pressure combustion chamber [2]. As their name suggests, they differ by variation of area. Usually, the constant pressure combustion chamber is preferred over the constant area combustion chamber to maintain the flow Mach number at the desired levels by choosing an appropriate combination of   and   [31]. Especially for the DMScramjet engine, thermal occlusion is relieved by the expanding area of the combustion chamber. Hence a constant pressure combustion chamber is designed and assessed in this research.

The function of the expansion system is to provide acceleration of the flow from the burner exit static pressure to local atmospheric static pressure over the entire range of vehicle operation in a controllable and reliable manner with maximum achievable performance [2]. Though the requirements of successful operation of the expansion unit are not as complex as the compression unit, still they have their own difficulties and can have an enormous effect on the overall performance of the engine [2]. High-speed vehicles will require a highly integrated propulsion system and airframe for structural integrity and their expansion system will encounter large variations in back pressure over the flight regime [51]. This makes the application of variable geometry nozzles impractical because of mechanical limitations that constrict extreme variations in expansion ratio [51]. The single expansion ramp nozzle (SERN) is chosen for this research due to its advantages over its potential competition which would be the axisymmetric nozzle [51]. The advantages include significant reduction of overall weight of the scramjet engine when SERN is used instead of an axisymmetric nozzle [2, 51] and minimum skin friction due to a short lower cowl [51]. High-speed SERNs are designed with a large expansion ratio necessary for maximum performance at high speeds and altitudes. The methodology of design and analysis of the expansion unit will be explained in their respective sections.

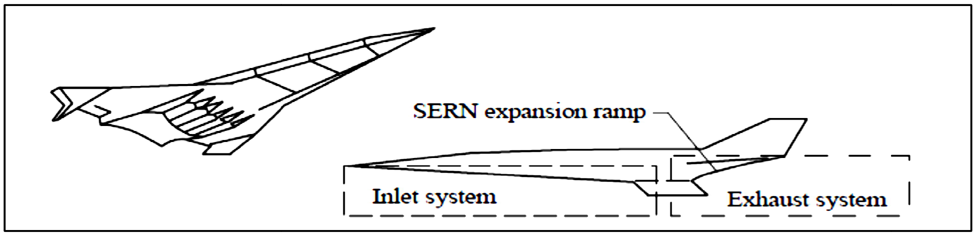


Figure 3-3 Schematic Representation of SERN on a High-Speed Vehicle [51]

3.3 Simulation Environment

3.3.1 Multi Fidelity Approach

One of the important objectives of this research is to develop a multi-fidelity simulation framework that enables fidelity of the analysis to be varied within the system depending upon the nature of the problem to be solved and the accuracy of the available models and methods. One such approach for accomplishing the multi-fidelity simulation capability is the Numerical Propulsion System Simulation code developed under the leadership of the NASA Glenn Research Center [18]. The multi-fidelity capability in the NPSS is achieved from the object-oriented nature of the C++ language with which it is written that allows the problem to be decomposed from the system-level assembly into subsystems, components, and parts, distributing the calculations to various levels [18]. The connectors between the various elements, or objects, of the system are defined in terms of the physical attributes of the elements (e.g. flow ports, bleed ports, shaft connections, etc.) which are the way the components in an engine communicate with each other [18]. The boundary conditions for each component and sub component which require higher fidelity simulation are passed on in the form of averaged data through data ports [18]. The levels of fidelity to evaluate the performance of the scramjet engine are based upon the classic tools which are used to analyze the propulsion system components and the dimensions of the simulation. The main requirement of building a successful multifidelity simulation system is to ensure that conservation of mass, momentum, and energy are maintained across the component interfaces of the system [18]

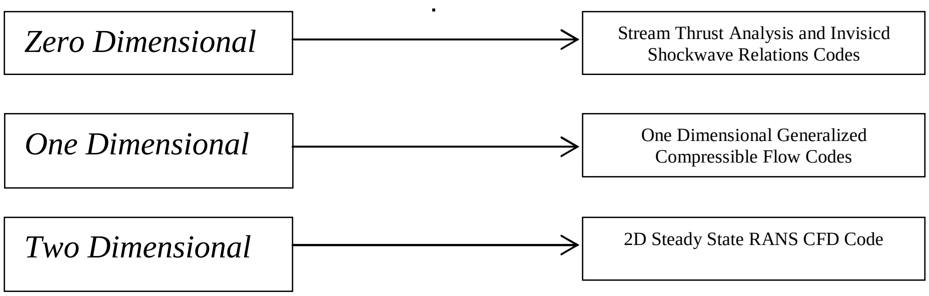

Zero Dimensional
Stream Thrust Analysis and Invisicd Shockwave Relations Codes
One Dimensional
One Dimensional Generalized Compressible Flow Codes
Two Dimensional
2D Steady State RANS CFD Code

Figure 3-4 Levels of Fidelity – Block Diagram.

3.3.2 Stream Thrust Analysis

Stream thrust analysis is a simplified method to access the performance of a scramjet engine [30]. It can determine both the installed and the uninstalled thrust produced by a scramjet engine by performing a finite control volume analysis and estimating its performance based on the study of the changes in stream thrust of the engine gas flow as it passes through the engine [30]. This method is based on a technique that conserves the fluxes of mass, momentum and energy on strategically placed control volumes to determine the propulsive forces on the vehicle [23]. The heart of this method lies in the fact that at very high speeds the area ratio of the exhaust nozzle is large enough such that the state condition of the exit flow is normally located in the non-dissociation region of the Mollier diagram [30]. This analysis has been proved to be accurate and close enough in comparison with more sophisticated computerized calculations for assessing the engine performance up to speeds of about Mach 15, which is very well within the operational limits of this research [30]. This method was originally initiated to determine the fuel specific impulse and specific thrust of the engine by evaluating the stream thrust of the working gas [30]. The stream thrust analysis can be elaborated with a one-dimensional, steady-state Newton's second law of motion for a control volume as follows [23, 30]

$$F_{uninstalled} = (PA)_{out} + (\dot{\mathrm{m}}_f + \dot{m}_{in})V_{out} - (PA)_{in} - \dot{m}_{in}V_{in} - F_{add} \qquad (3.1)$$

where is the sum of spillage drag and the plume drag, combined into a single parameter called the additive drag [23]. The stream thrust term (Sa) is define as [30]

$$Sa = PA + \dot{m}V \qquad (3.2)$$

Hence with the knowledge of the stream thrust values, the uninstalled engine thrust can be written as

$$F_{uninstalled} = (\mathrm{Sa})_{out} - (\mathrm{Sa})_{in} - F_{add} \qquad (3.3)$$

Once the value of stream thrust is known, the specific thrust and the specific impulse can be calculated easily [30]. An important note to be made here is that the stream thrust change may be estimated theoretically or measured experimentally, and it is an easy matter to introduce specific component performance into this analysis or alternatively to use the analysis as a means to express the impact of component performance on flight performance [30].

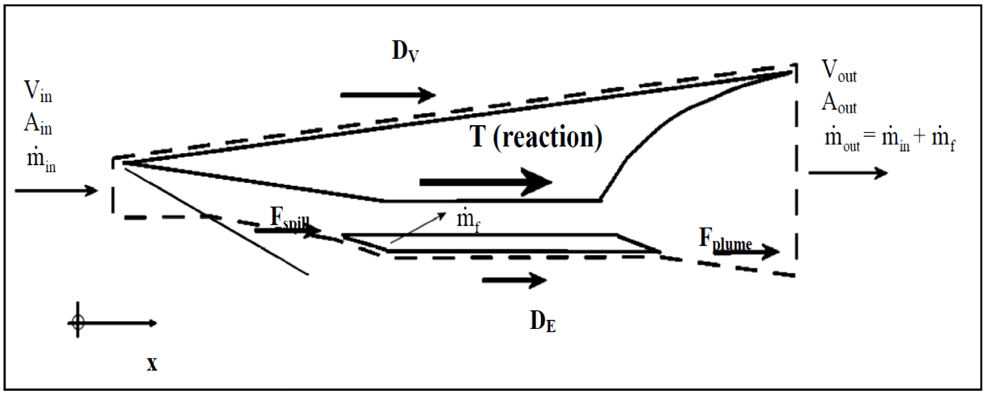


Figure 3-5 Control Volume Representation of a Scramjet Flow Path [23]

Stream thrust analysis is preferred over the conservation of energy methods such as thermodynamic closed cycle analysis and first law analysis for this research because it has the greatest depth to account for several phenomena that considerably influence performance, namely: “the mass, momentum and kinetic energy fluxes contributed by the fuel, the geometry of the burner, and exhaust flows that are not matched to the ambient pressure” [2, 25].This method is employed to handle the basic level of fidelity (zero dimensional) in the components to be built in NPSS since it only calculates the values of the flow parameters in designated stations. This zero-dimensional approach can be explained precisely by a good comparison to the use of multi-grid algorithms used in fluid flow analysis to accelerate convergence of the non-linear flow equations [8]. A zero-dimensional cycle analysis serves as a type of “course mesh” to improve communication between the “fine mesh “detailed component calculations [8]. This system should not have any problem in convergence to represent the performance of the complete engine as far as the components and sub-components are designed accurately and interconnected carefully in NPSS [8]. This analysis can also be used for interpretation of results from component tests in terms of overall engine performance, when experimental data or prior knowledge about the component through simulation is available, by fitting hypothetical components [30].

3.3.3 Generalized One-Dimensional Flow Analysis

One-dimensional methods in propulsion simulation are known for their ability to help the designers and researchers [31]. This serves as an excellent method for analysis if the flow within a scramjet engine is confined within the definite boundaries of the engine flow path, making a two- or three-dimensional analysis unnecessary for achieving an understanding of the flow [2, 25]. Due to the one-dimensional nature of these equations, they can be solved quickly and are ideal as a reduced order model in the current research [33]. Although the one-dimensional approach can never be perfectly correct, the alternatives are both hopelessly complex and completely unwieldy for reaching a basic understanding built upon fundamental principles [2, 25]. The one-dimensional analysis may have a drawback of being unable to account for the growth of boundary layers and other multidimensional flow phenomena such as shock/boundary layer interactions, but it can still provide useful results if the physics of the flow is broken down to average quantities such as skin friction coefficient [31]. The time-dependent, quasi-one-dimensional form of the equations can be used to capture the geometry of a compression-expansion section and replicate many of the physical features of actual flows in scramjets [33]. A good knowledge about previous work done on developing one-dimensional codes to estimate the performance of a scramjet can be gained from the work of Tran [31]. The one-dimensional analysis in this research follows the single stream tube theory applied to generalized one-dimensional flow concepts using the influence coefficients discussed by Shapiro [32]. It is derived from the logarithmic form of the conservation of mass, momentum and energy equations along with the equation of the state, the definition of the Mach number and the second law of thermodynamics [32]. The logarithmic forms of the conservation equations are used to account for the change in the values of influence coefficients along the axial direction. The assumptions applied to this method are that the flow is steady and one dimensional, changes in the stream properties are continuous and the working gas is semi perfect where it obeys Charles' law and Boyle's law and has a specific heat which varies only with the temperature [32]. The differential conservation equations to be solved for using the influence coefficient method in this research are as follows [32, 34]

$$\frac{d\dot{m}}{m}=\frac{d\rho}{\rho}+\frac{dV}{V}+\frac{dA}{A} \qquad (3.4)$$

$$\frac{dp}{p}+\frac{\gamma M^2}{2}\frac{dV^2}{V^2}+\frac{\gamma M^2}{2}\frac{4C_f dx}{D}=0 \qquad (3.5)$$

$$\frac{dT}{T}+\frac{\gamma-1}{2}M^2\frac{dV^2}{V^2}=\left(1+\frac{\gamma-1}{2}M^2\right)\frac{dT_t}{T_t} \qquad (3.6)$$

The above equations are solved along with the equation of state and definition of Mach number [32, 34]

$$\frac{dp}{p}-\frac{d\rho}{\rho}-\frac{dT}{T}=0 \qquad (3.7)$$

$$\frac{dM^2}{M^2}-\frac{dV^2}{V^2}+\frac{dT}{T}=0 \qquad (3.8)$$

The differential equation, derived from the above equations, for the Mach number distribution of the flow in axial direction is given by

$$\frac{1}{M^2}\frac{dM^2}{dx}=\frac{\psi(M)}{\left(1-M^2\right)}\left(-\frac{1}{A(x)}\frac{dA(x)}{dx}+\frac{\gamma M^2}{2}\frac{4f}{D(x)}+\frac{1+\gamma M^2}{2}\frac{1}{T_o(x)}\frac{dT_o(x)}{dx}+\frac{(1+\gamma M^2)}{\dot{m}}\frac{d\dot{m}}{dx}\right) \qquad (3.9)$$

where

$$\psi(M)=1+\frac{\gamma-1}{2}M^2 \qquad (3.10)$$

Equation (3.90) takes the change in area, friction, total temperature and mass flow rate into account to calculate the axial distribution of the Mach number. But the equation is tailored to each individual component to match its need and physics. The set of equations applicable to each component is explained in their respective sections. Once the distribution of Mach number is obtained, the other flow properties are calculated using the "useful integral relations" available from Shapiro [32].

Let us assume that the primary properties at station i are known. The flow variables are calculated from station 3 in the axial direction after determining the distribution of Mach number $M(x)$ using the set of equations as follows [32].

$$T(x) = T_i \cdot \frac{T_t(x)}{T_{ti}} \cdot \left[\frac{1 + \frac{(\gamma - 1)}{2} M_i^2}{1 + \frac{(\gamma - 1)}{2} M(x)^2} \right] \tag{3.11}$$

$$p(x) = p_i \cdot \frac{A_i}{A(x)} \cdot \frac{M_i}{M(x)} \sqrt{\frac{T(x)}{T_i}} \tag{3.12}$$

$$P_t(x) = P_{ti} \cdot \frac{p(x)}{p_i} \cdot \left[\frac{T_i}{T(x)} \cdot \frac{T_t(x)}{T_{ti}} \right]^{\frac{\gamma}{(\gamma-1)}} \tag{3.13}$$

$$u(x) = u_i \cdot \frac{M(x)}{M_i} \sqrt{\frac{T(x)}{T_i}} \tag{3.14}$$

3.3.4 Computational Fluid Dynamics Model

The CFD simulation marks the highest level of fidelity in this research. The flow in a hypersonic compression system has many complications which should be considered important for the proper operation of a complete scramjet. The interaction among oblique shock waves and their interaction as a system with the boundary layer adjacent to the wall will create complex flow fields [2]. Hypersonic boundary layers are relatively thicker leading to high shear stress and create very high temperatures and low densities which cause hypersonic viscous interactions [2]. One of the important reasons to use CFD in this research is to analyze the complications created by combustioninduced disturbances which may cause flow reversal, inlet unstart which eventually leads to loss of successful propulsion system [2]. The physical and chemical status of the air entering the combustion chamber is also to be determined [2]. The only tool which can assess all these phenomena up to some level at the same time reducing the cost of expensive ground testing would be CFD [2]. Hence it is sought to perform a 2D steady-state CFD simulation using the Reynolds-averaged Navier-Stokes (RANS) equations to understand the physics involved. Apart from the reason of time conservation, the RANS approach was chosen because of its accuracy over the LES filter models in comparison with experiments involving multiple shock interactions [35].

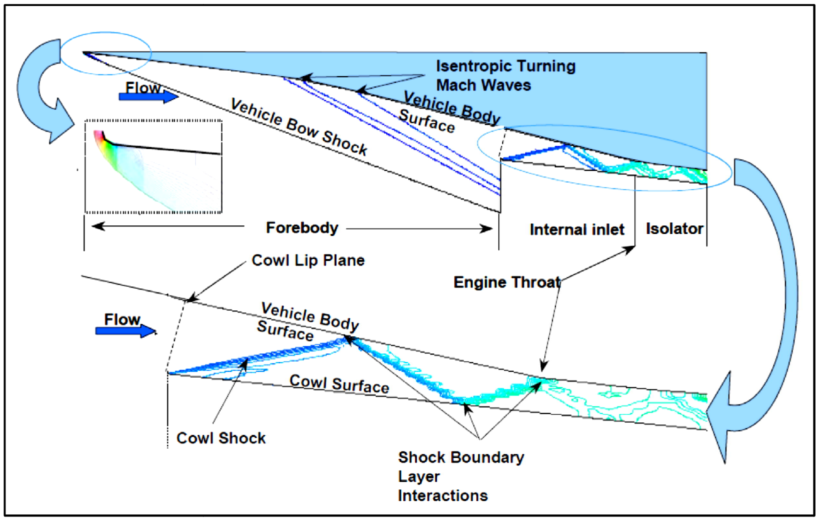


Figure 3-6 Complexities in Flow Analysis of Hypersonic Inlet [23]

The commercial CFD code, ANSYS® FLUENT 14.5 solver was used throughout this research. It provides comprehensive modeling capabilities for a wide range of incompressible and compressible, laminar and turbulent fluid flow problems and includes a broad range of mathematical models for transport phenomena (like heat transfer and chemical reactions) [36]. The conservation equations for mass and momentum and for flows involving heat transfer or compressibility an additional equation for energy conservation are solved for [36]. The equations solved by FLUENT for a transient case is mentioned as follows [36]

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\bar{v})=S_m \tag{3.15}$$

$$\frac{\partial(\rho\bar{v})}{\partial t}+\nabla\cdot(\rho\overline{vv})=\nabla p+\rho\bar{g}+\bar{F} \tag{3.16}$$

$$\frac{\partial(\rho E)}{\partial t}+\nabla\cdot(\bar{v}(\rho E+p))=-\nabla\cdot\left(\sum_{j}h_j J_j\right)+S_h \tag{3.17}$$

where is the source term, is the gravitational body force and is external body force [37]. They seek to apply Reynolds averaging to the unsteady Navier-Stokes equation to obtain the RANS equations [37]. All scales of the turbulence field are being modeled to close the set of RANS equations and the resulting models are known as statistical turbulence models due to the statistical averaging procedure employed to obtain the equations [37]. An implicit, density-based solver, which solves the governing equations of continuity, momentum, energy and species transport simultaneously as a set, or vector of equations, are chosen since it is suitable for solving the domain which has the nature of the compressible flow [59].

The result of the simulation is very sensitive to the turbulence model employed and a careful review of the turbulence model used was to be performed. The turbulence model which suits the requirements of this research has to be selected and employed to run a successful viscous simulation with turbulent environment. Especially, the use of a RANS CFD code can yield a significant error when applied to practical flows involving shock waves [38]. From previous works on shock/turbulent boundary-layer interaction, it was understood that the models involving an equation for turbulent kinetic energy overWpredicted the value of *k* after the interaction region and the problem was corrected by imparting a shock unsteadiness modification to the existing CFD codes [38]. Another research which has a similar problem overcame it by setting an upper limit of 10 times the amplification of the turbulent kinetic energy parameter to discourage spurious production of turbulence across shock waves [39]. The present work does not focus on developing or modifying any turbulence model and had to choose the best possible turbulence model available in ANSYS FLUENT database. This problem was overcome by selecting the Spalart-Allmaras turbulence model which solves only for a scalar parameter called the modified eddy viscosity and does not solve for any turbulent kinetic energy directly. The Spalart-Allmaras model is a one-equation model that solves a modeled transport equation for the kinematic eddy (turbulent) viscosity [36]. It is well known for aerospace applications involving wall-bounded flows and has been shown to give good results for boundary layers subjected to adverse pressure gradients [36]. The modified eddy viscosity was solved by the Quadratic Upstream Interpolation for Convective Kinetics scheme (QUICK). The wall y^+ is maintained within the full turbulent portion or the log-law region (30 < y^+ < 300) for the viscous simulations throughout this research. The solution-based adaptive approach is followed to reduce the efforts towards grid convergence. Solution-adaptive mesh refinement can be used to increase and/or decrease mesh density based on the evolving flow field, and thus provides the potential for more economical use of grid points [59]. Dynamic solution-adaptive mesh refinement is performed to efficiently reduce the numerical error in the digital solution, with minimal numerical cost [59]. ANSYS FLUENT multiplies the Euclidean norm of the gradient of the selected solution variable by a characteristic length scale [59];

$$|e_{i1}| = \left(A_{cell}\right)^{\frac{r}{2}} |\nabla f| \qquad (3.18)$$

where e_{i1} is the error indicator, A_{cell} is the area of the cell, r is the gradient volume weight and ∇f the Euclidean norm of the gradient of the desired field variable. The normalization method is chosen to scale the values of the error indicator by their maximum value in the domain and returns a problem-independent range of [0, 1] for any variable used for adaption [59]

$$\frac{|e_{i1}|}{\max|e_{i1}|} \qquad (3.19)$$

The designing and meshing was performed in ANSYS Designmodeler® and ANSYS Meshmodeler® since they had a good communication with each other through the ANSYS Workbench and a scope for automation in the future was realized. Having discussed the setup used for the simulation environment in this research, the following topics give a prelude on the designing process of the hypersonic compression system.

3.4 Design Requirements

Some basic knowledge of the design criteria and difficulties are noted in this section before the design process, without which the design requirements may be very vague.

3.4.1 Selection of Fuel

The design of the scramjet compression system was started by choosing the cycle static temperature ratio ($\psi = T_3/T_0$), which will be the natural choice of independent variable since it is the principal determinant of the thermodynamic cycle efficiency. As mentioned earlier, the static temperature of the compressed fluid must be able to auto ignite the fuel injected right from the minimum flight Mach number and mixed efficiently within the length of the combustion chamber. The auto ignition temperature (IT) of a fuel is the temperature at which the fuel will self-ignite in air without a flame source or spark

[2]. Fuel selection plays a very important role not only in engine design, but also on the impact that the propulsion system has on the vehicle and mission design [25]. This research focuses on using a hydrocarbon fuel instead of hydrogen fuel because of its mission requirement which has been explained earlier.

Hydrocarbon fuels are more readily associated with current aerospace and military flight applications. They are grouped into two categories, methane and its derivatives or the kerosene (JP) range. Hydrocarbon fuels are up to eleven times denser than liquid hydrogen and although they possess far less energy capacity per unit mass, their overall energy per unit volume is still up to 3 ½ times greater than that of hydrogen [27].This feature lets the vehicle size to be much smaller than a hydrogen-fueled scramjet giving more room to the payload to be carried. The fuels to be considered for this research are based on the fuels on which active interest was shown and were used successfully in the past. Since the purpose of fuel in a propulsion system is, in the most basic sense, to convert the chemical energy stored in the fuel to thermal energy which can produce thrust [2], it makes sense to distinguish between the various fuels on a chemical level [25]. The fuels are categorized based on their heat of reaction values (h_{PR}) which is defined as the "heat that must be removed from the final combustion products in order to return them to the same temperature as the initial reactants (at the same pressure)" [2, 25]. The list of hydrocarbon fuels, along with their IT and h_{PR}, that were considered for this research is mentioned below.

Table 3-1 List of Potential Supersonic Combustion Fuels [2, 23]

$Fuels$	IT (K)	f_{ST}	$h_{PR}(KJ/kg)$
Hydrogen	845.15	0.0294	119954.00
Methane	810.15	0.0583	50010.00
Ethane	745.15	0.0624	47484.00
JP-7	518.15	0.0674	43903.25
Hexane	498.15	0.0659	45100.00
Octane	479.15	0.0664	44786.00

Methane is mentioned often as a fuel for a hypersonic cruiser with Mach 6+ cruising speed [25]. Since in this research, the operational limit starts from Mach 3.5, methane is ruled out. Out of the remaining hydrocarbon fuels, JP 7 is selected for this research due to the following reasons. The endothermic property of the JP 7 fuel acts as a heat sink and could extend the upper operational limit of the hydrocarbon fueled scramjet engines [23]. In the SR-71 engine, JP-7 was used in the engine hydraulic system in addition to being the propellant [23]. It also reduces the radar signature with its history of success in the field of high speed engines when considering X51 and SR 71 [23]. The minimum temperature required for JP 7 fuel to ignite is 518.15 K at 1 atm pressure.

3.4.2 Determination of Inlet Unstart

Usually, hypersonic inlets are designed for a particular cruise Mach number which will be its upper limit of operation [23]. It is done to capture the maximum mass flow rate available to the inlet by designing for the shock-on-lip condition and avoiding any spillage. It also prevents the bow shock of the vehicle from entering the inlet and creating unwanted internal shock/boundary layer interactions, since it is the strongest of the shocks formed by the fore body. Deciding the cycle static temperature ratio is one of the highly tricky aspects of scramjet inlet design. The inlet can be designed for the maximum cycle static temperature possible at its design Mach number to achieve maximum thermodynamic efficiency and performance. But it is likely to unstart at lower Mach numbers, limiting the range of operation of the scramjet. For proper operation, all scramjet inlets must operate in a started mode, which can be defined as the operation of the inlet under conditions where flow phenomenon in the internal portion of the inlet do not alter the capture characteristics of the inlet [29]. The event of unstart is amplified in a hypersonic inlet compared to a turbojet or a ramjet inlet since, when the inlet unstarts in a turbojet or a ramjet, a normal shock is expelled and the spillage is subsonic, whereas when a scramjet engine unstarts, the expelled shock system is strong enough to separate the boundary layer and the spillage is supersonic, reducing the efficiency and increasing the thermal loads [29]. Inlet starting is influenced by various parameters such as the local Mach number, internal contraction ratio and time dependent details of the starting process which represents that for a specific inlet geometry with a set of free stream conditions an inlet will start at a certain internal contraction ratio [29].

The preliminary internal contraction ratio is obtained by assuming a normal shock, which is a typical result of an unstarted inlet and the flow becomes subsonic. The internal contraction ratio is then calculated with a one-dimensional, isentropic and perfect gas assumption that will produce sonic flow at the throat of the compression system. This concept is also known as the “Kantrowitz limit” and is used in many gas dynamics applications to determine the theoretical fluid choke points at supersonic velocity [29]. The Kantrowitz limit is given by

$$\left(\frac{A_2}{A_4}\right)_{Kantrowitz} = \frac{1}{M_2}\left[\frac{(\gamma+1)\mathrm{M}_2^2}{(\gamma-1)\mathrm{M}_2^2+2}\right]^{\frac{\gamma}{\gamma-1}}\left[\frac{\gamma+1}{2\gamma\,\mathrm{M}_2^2-(\gamma-1)}\right]^{\frac{1}{\gamma-1}}\left[\frac{1+\frac{(\gamma-1)}{2}\mathrm{M}_2^2}{\frac{(\gamma+1)}{2}}\right]^{\frac{\gamma+1}{2(\gamma-1)}} \qquad (3.20)$$

Though the initial estimate of the internal contraction ratio is provided by the equation above, the Kantrowitz limit becomes conservative at higher Mach number since it assumes only one normal shock at the lip of the inlet, which usually is the case for a unstarted supersonic inlet In the case of a hypersonic inlet, the pressure recovery from the shock system is higher than a single shock wave [29]. Because of the inaccuracy of the Kantrowitz limit in hypersonic speeds, an empirical data correlation known as the maximum steady flow contraction limit is proposed for scramjets with flight Mach number 2.5 < M_0 < 10, which was obtained from experimental data of a wide range of cases [29]:

$$\left(\frac{A_4}{A_2}\right) = 0.05 - \frac{0.52}{M_0} + \frac{3.65}{{M_0}^2} \qquad (3.21)$$

Usually, a successful area contraction falls within the above mentioned two limits, but in this case, the contraction ratio is finally checked with CFD simulations to assess the validity of the limits.

3.4.3 Determination of Boundary Layer Separation

The impact of the boundary layer on compression of the incoming hypersonic flow through the inlet is unavoidable. At hypersonic speeds, boundary layer development within an inlet has a major influence on the performance of the inlet since its growth adds to the effective compression of the captured flow [29]. The losses due to the presence of skin friction are the largest of all the losses of a hypersonic inlet [29]. The desire to reduce the length of the inlet to save weight creates a scenario where the captured airstream is aimed to be compressed in a short length, exposing the boundary layer to high levels of adverse pressure gradients [29]. This separates the boundary layer from the wall resulting in separation bubbles which are undesirable and to be avoided at all cases since it creates additional shock waves which were not a part of the compression process [29]. It also results in high levels of heat transfer at the zone of reattachment, generation of acoustic loads due to flow unsteadiness and weakening of boundary layer such that the downstream effects can easily propagate upstream which eventually leads to inlet unstart [29]. Chances for shock/boundary layer interaction are maximum at the inlet section of the engine. The term incipient separation pressure rise is defined as the highest pressure rise that can be sustained by the boundary layer without the event of separation [29]. For a two-dimensional, shock-induced boundary layer separation, the separation shock is followed by expansion waves and a reattachment shock which occurs to turn the flow parallel to the wall [29].

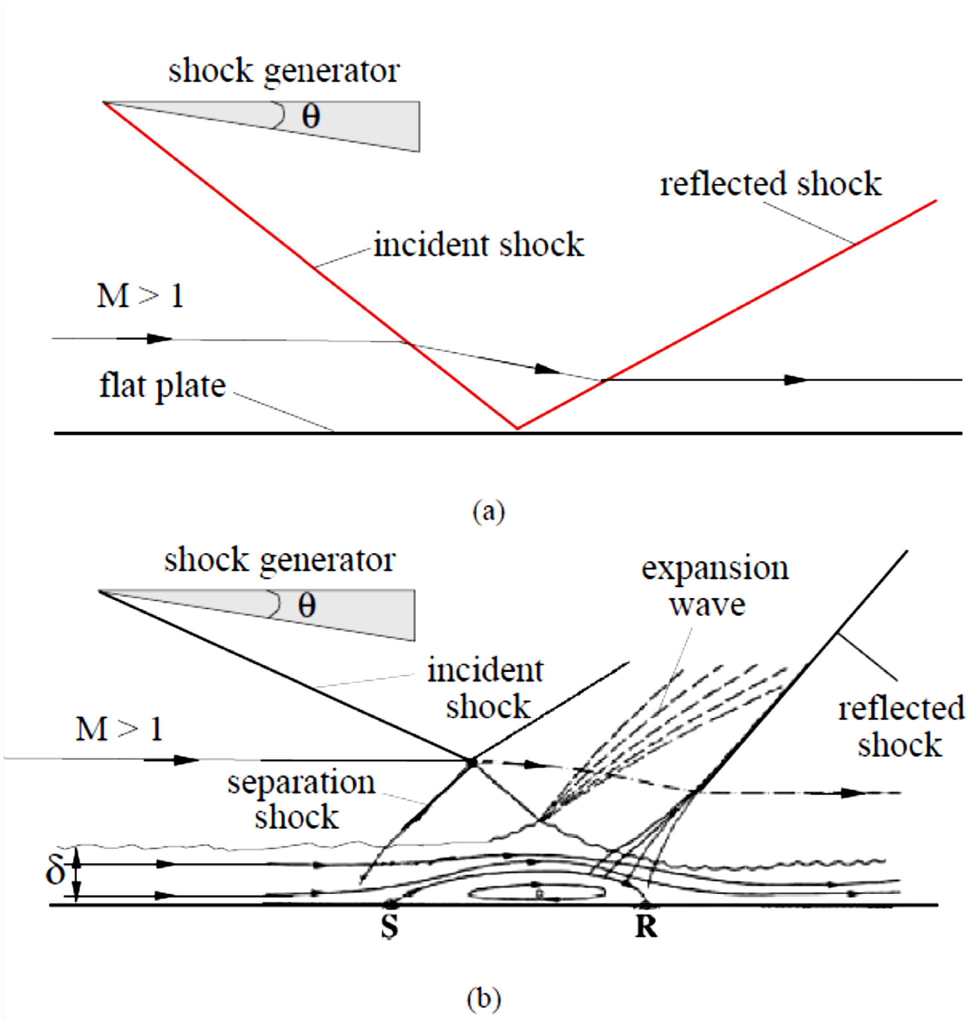


Figure 3-7 Shock Wall Interactions (a) For Inviscid Flow [40], (b) With Boundary Layer For Viscous Flow [40].

For a turbulent boundary layer, the pressure rise required to separate the flow is almost five times of that required to separate a laminar boundary layer [29]. A simple correlation to obtain the maximum incipient pressure rise as a function of Mach number was obtained by Korkegi [41] which is applied to determine boundary layer separation throughout this research. It is given by

$$\frac{P_2}{P_1} = 1 + 0.3M^2 \quad \text{for } M < 4.5 \qquad (3.22)$$

$$\frac{P_2}{P_1} = 0.17M^{2.5} \text{ for } M > 4.5 \qquad (3.23)$$

where 1 and 2 are the station number before and after oblique shock wave. Thus the important factors leading to failure of a hypersonic inlet were explained in this section. Now, more details on the code developed and their integrations will be explained in further sections

Chapter 4

Development of NPSS and CFD Models

This chapter explains the components developed using NPSS and their integration with CFD. NPSS has an object oriented program structure, which lets the components of an engine to be developed as elements and sub-components as sub-elements. Elements are the main building blocks of an engine model in NPSS and generally represent the major components of the engine. Subelements are element-like objects designed to support the object to which they are attached [42]. NPSS comes with a standard set of elements and sub-elements, most of them being turbine engine components and some rocket engine components, that fulfill many modeling needs [42]. In this research, the DMScramjet engine components are built by maintaining the same architecture of NPSS for clarity purposes and are combined to form the complete scramjet. The DMScramjet components such as the inlet, burner and nozzle are modelled as elements and the isolator is modelled as subelement since it is used interchangeably. Additional functions are created to aid the calculations of the components.

4.1 Thermodynamic Package

One of the most outstanding features of NPSS is the availability of a number of thermodynamic packages and relevant data. Their modular design allows a user to select the desired package at run time [42]. The thermodynamic package is a program with database which stores the properties of single as well as mixtures of gases which is collected from a list of leading aircraft industries to support the thermodynamic analysis of aerospace as well as industrial turbine engines. The thermodynamic package is capable of performing the thermodynamic calculation of the flow under extreme conditions such as gas vibration and dissociation up to molecular level resolution. Out of the five thermodynamic packages available, two packages named “JANAF” and “ALLFUEL” are used for the current research. Janaf is a program which calculates chemical equilibrium product concentrations made up of seventeen reactants namely H2O, CO2, O2, N2, AR, CH4O, NH3, H2, O, H, CO, OH, NO, N, CH4, C2H4, C10H8 [43]. It offers flexibility and matches the NIST standard (NIST−JANAF, Revision 3) at the expense of some computational speed [42]. It is similar to CEA in that it solves for composition based on minimization of Gibb’s free energy but unlike CEA it does not have an unlimited number of products [43]. The product list has been cut down to those typically seen in hydrocarbon combustion. This thermo package is used for the hypersonic compression system to take molecular vibration as well as dissociation effects into account if they occur.

The ALLFUEL thermo package is from General Electric and contains both gas and fuel properties [42]. It is generally consistent with NASA TP-1908 but specific agreement varies with the fuel and working fluid options selected [42]. It defines the thermodynamic state of a gaseous mixture composed of air, water vapor, and a variety of fuels but only JP class fuel and methane are recommended [43]. It is a thermo package which contains only the burned properties of fuel gas mixture [43]. Only the JP class of fuel is used in this research. The 2:1 H-C ratio value is close for all of these JP fuels [43]. This model applies a 1962 atmosphere (with 0.03% CO2) and is the GE current standard [43]. This thermo package is used for the combustion system as well as the expansion system of the DMScramjet engine. Further details about how this package is used are provided when each component is explained. A flowstation is an inbuilt program which works with the thermodynamic package to set and return fluid properties [45]. It can be set anywhere in a model or element to calculate the local flow properties. The flowstation calculates a partial list of total properties if the flow total temperature and total pressure are known [45]. Once the flow mass flow rate along with one of its properties such as static temperature, static pressure, static enthalpy, velocity, Mach number or area is fed in, the flowstation calculates the complete list of flow properties which are provided in reference [45]. Local flowstations have been initialized throughout the scramjet elements where ever it is necessary, with the help of the local flow stations the specific heat ratio γ and specific heat capacity at constant pressure C_p are modelled with respect to local temperature making real gas model achievable throughout the DMScramjet elements. Another feature to be noted when building elements is the ports which connect them. One of them is a fluid port which functions just like flow station, but it also links elements together and is typically declared inside of all the elements in NPSS [45]. The other types of ports are explained at the relevant section where they are applied.

4.2 On Design Code for DMScramjet Engine

All the elements of the DMScramjet are subdivided into two subelements to handle the calculations for on- and off-design cases separately. The design case calculations are acceptable only at Mach 6 flight and are based on the expectation of optimum performance for the case it is built for. For all other flight Mach numbers, the off-design code must be used. This section explains the on-design code of the hypersonic compression system which includes the hypersonic inlet and isolator and the approach used for each model respectively.

The hypersonic inlet is operated with the JANAF thermodynamic program with air assumed to be composed of 21% oxygen and 79% atmospheric nitrogen which is from the 1970’s standard tables [43]. The property of air can be changed to any fraction of seventeen reactants available in the package where the addition of all the fractions should result in 1.0 [43]. The scramjet inlet is attached to two standard elements of NPSS which are the ambient element and the inlet start element. The ambient element provides a list of air properties at the given altitude and Mach number. The properties include ambient pressure, ambient temperature, and delta temperature from selected atmosphere, calibrated airspeed, equivalent airspeed, true airspeed, total pressure, total temperature, relative humidity, specific humidity and dynamic pressure [44]. The ambient element is followed by the inlet start element. The inlet start element sets the inlet start conditions based on flight parameters are obtained from the ambient element and the mass flow rate provided [44]. The inlet start element initiates the flow to the inlet and also calculates the inlet corrected or actual flow which should be passed to the inlet if a turbine engine is used [44]. The hypersonic inlet element is connected to the inlet start by a fluid port. The design and off-design subelements are contained by this element. Details about the subelements are discussed in the following sections.

4.2.1 On-Design Inlet NPSS Code (Station 0 – 2)

The on-design inlet subelement makes the initial calculations using a realistic stream thrust analysis approach. This approach can be applied to any combination of external and internal shock wave compression [2]. The realistic stream thrust approach varies from the usual stream thrust analysis by dividing the inlet into two control volumes. For the first control volume, analysis is performed only for the oblique shock wave system for which skin friction effects are neglected [2]. The second control volume analysis is performed by assuming the inlet element to be a constant pressure duct to simplify the ensuing mathematics [2]. These assumptions may look absurd but it is built upon the previous findings that the precise details of the flow process have only a slight impact on the resulting performance as well as the fact that this model is primarily designed to probe the trends and limits of the influence of viscosity rather than to precisely investigate any particular case [2]. It is also to be noted that the amount of wall friction assigned can and will be arranged to account for that generated on the actual external surfaces of the compression component. The compression component assessment performed by this analysis is portrayed in a T-s diagram in Fig. 4.1 below [2]

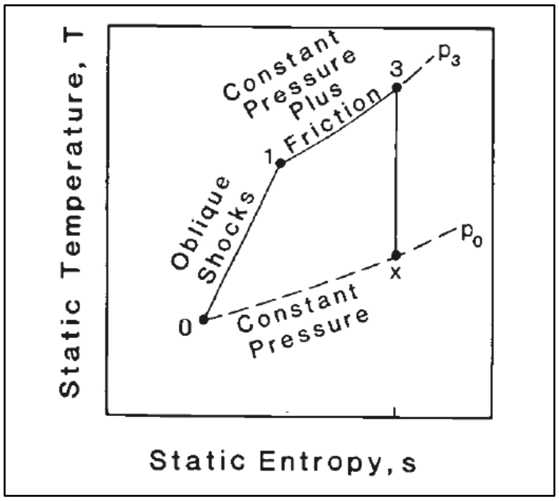


Figure 4-1 T-S Diagram Portrayal of Realistic Stream Thrust Analysis of Compression Component [2]

The inviscid analysis takes only the oblique shockwaves into account. The number of turns and the turning angle of the flow with respect to the flow path created by the inlet should be provided to this analysis. This analysis uses the work of Emanuel [46] to represent the $\theta - \beta - M$ implicit relation into an explicit quadratic form which can determine weak and strong solutions. It can be expressed as

$$\tan\beta = \frac{M^2 - 1 + 2\gamma\lambda\cos\left[\dfrac{4\pi\delta + \cos^{-1}\chi}{3}\right]}{3\left(1 + \dfrac{\gamma-1}{2}M^2\right)\tan\theta} \qquad (4.1)$$

where δ = 0 yields strong shock solution and 1 yields a week shock solution,

$$\lambda = \left[\left(M^2 - 1\right)^2 - 3\left(1 + \frac{\gamma - 1}{2}M^2\right)\left(1 + \frac{\gamma + 1}{2}M^2\right)\tan^2\theta \right]^{\frac{1}{2}} \quad (4.2)$$

and

$$\chi = \frac{\left(M^2 - 1\right)^3 - 9\left(1 + \frac{\gamma - 1}{2}M^2\right)\left(1 + \frac{\gamma - 1}{2}M^2 + \frac{\gamma + 1}{2}M^4\right)\tan^2\theta}{\lambda^3} \quad (4.3)$$

Once the oblique shock angle β is determined, the change in the flow after each shockwave is determined using the oblique shock relations derived from the one-dimensional conservation of mass, momentum and energy equations [46]. They are presented with the station numbering 1 and 2 representing flow conditions before shock and after the shock respectively as follows [46]

$$M_{n1} = M_1 \sin\beta \quad (4.4)$$

$$\frac{\rho_2}{\rho_1} = \frac{(\gamma + 1)\,\mathrm{M}_{n1}^2}{(\gamma - 1)\,\mathrm{M}_{n1}^2 + 2} \quad (4.5)$$

$$\frac{p_2}{p_1} = 1 + \frac{2\gamma}{\gamma + 1}\left(\mathrm{M}_{n1}^2 - 1\right) \quad (4.6)$$

$$M_{n2}^2 = \frac{M_{n1}^2 + 2/\gamma - 1}{\left[2/\gamma - 1\right] M_{n1}^2 - 1} \quad (4.7)$$

$$\frac{T_2}{T_1} = \frac{p_2}{p_1}\frac{\rho_1}{\rho_2} \quad (4.8)$$

$$M_2 = \frac{M_{n2}}{\sin\left(\beta - \theta\right)} \quad (4.9)$$

The above relations provide the pressure ratio, temperature ratio and Mach number after the oblique shock wave. Between each successful shockwave formation, Korkegi’s incipient pressure rise limit (eq 3.22) governs the maximum pressure rise that can be achieved without boundary-layer separation. The code notifies for turbulent boundary layer separation once this limit is crossed during the design condition assessment. The temperature ratio across each shock is multiplied to obtain the overall temperature ratio $\left(\psi_1 = T_1/T_0\right)$ created by the system of oblique shock waves through compression of the incoming gas. The adiabatic shock efficiency after every shock wave is determined using [2]

$$\eta_{S.W} = \frac{{T_2}\big/{T_1} - \left\{\dfrac{(\gamma-1)M_{n1}{}^2 + 2}{(\gamma+1)M_{n1}{}^2}\right\}\left\{\dfrac{2\gamma M_{n1}{}^2 - (\gamma-1)}{\gamma+1}\right\}^{1/\gamma}}{\left({T_2}\big/{T_1} - 1\right)} \qquad (4.10)$$

By multiplying the adiabatic shock efficiency after each shock wave, the overall adiabatic shock efficiency η_1 is determined. Then the analysis is carried on to the second control volume which assumes a constant pressure inlet. The temperature increase produced only by the hypersonic viscous interactions with the inviscid flow can be obtained by modifying the one-dimensional momentum equation as [2]

$$\psi_{viscous} = 1 + \frac{\gamma-1}{2} M_0^2 \left\{1 - \left[1 + \frac{C_f}{2}\cdot\frac{A_w}{A_3}\right]^2 \left[1 + \frac{2(1-\psi_1)}{(\gamma-1)M_0^2}\right]\right\} \qquad (4.11)$$

The value of $\dfrac{C_f}{2}\cdot\dfrac{A_w}{A_3}$ is the dimensionless boundary layer friction quantity and, from previous studies performed, it varied from $0.01 - 0.05$ [2]. The overall temperature rise produced by the hypersonic inlet is determined by multiplying the temperature ratio obtained from the oblique shock system and the viscous interaction $\left(\psi = \psi_1 \times \psi_{viscous}\right)$.

Since the total static pressure rise produced by the inlet is only by the shockwave system, the total adiabatic compression efficiency of the inlet can be determined from [2]

$$\eta_c = \eta_1 \left(\frac{1 - \frac{1}{\psi_1}}{1 - \frac{1}{\psi}} \right) \qquad (4.12)$$

Once the overall pressure and temperature ratios of the inlet are determined, the stream thrust analysis results in other preliminary flow variables and initial geometry of the inlet. The velocity of the fluid stream at the exit of the inlet is given by [2]

$$V_2 = \sqrt{V_0 - 2C_{pc}T_0(\psi - 1)} \qquad (4.13)$$

The stream thrust function at the inlet exit is given by [2]

$$Sa_2 = V_2\left(1 + \frac{RT_2}{V_2^2}\right) \qquad (4.14)$$

and the overall Mach number at the exit of the inlet is calculated using the relation for local total temperature [2]

$$M_2 = \sqrt{\frac{\left[\left(Tt_2 / T_2\right) - 1\right]}{\left(\gamma - 1/2\right)}} \qquad (4.15)$$

The total pressure at the exit of the inlet is obtained from the local isentropic relation since the local static pressure and Mach number are available from oblique shock calculations. Having determined the total pressure, total temperature, mass flow rate and Mach number at the exit of the inlet, these values are provided to the exit fluid port which calculates the rest of the fluid properties and passes over to the next element connected to it.

4.2.2 On-Design Isolator NPSS code (station 2 – 3)

As mentioned earlier, the isolator is modelled as a subelement. Even though the isolator plays a role of secondary compression unit, the properties of the flow in the isolator switch between the compression unit and the combustion unit during the time of precombustion shock formation. At the on-design condition, the isolator does not contain any shock train since no adverse pressure gradient or thermal choking is expected in the combustion chamber during on design conditions. In order to increase the fidelity from zero- to one-dimensional, the Fanno flow approach is opted inside a constant area duct for on-design isolator calculations. The axial distribution of Mach number is obtained from [47]

$$\frac{dM^2}{M^2} = \frac{\gamma M^2 \psi(M)}{1 - M^2} \frac{4C_f}{D_h} dx \qquad (4.16)$$

where

$$\psi(M) = 1 + \frac{\gamma - 1}{2} M^2 \quad (4.17)$$

The above ODE equation was solved in the x direction using a Runge–Kutta (RK) second-order method [48]

$$\frac{dy}{dx} = f(x, y), \mathrm{y}(0) = y_0 \qquad (4.18)$$

Heun’s method was preferred out of the different RK second-order methods because of algorithmic simplicity. Additional functions were built to solve for the Mach number using Heun’s method. These functions are used throughout this research whenever a first-order ODE is to be solved. At each axial step, a local flowstation calculates the local specific heat value γ which is used for the next step when advancing spatially. The JANAF thermodynamic package is used for the on-design isolator in order to account for gas vibration or dissociation. After the axial distribution of the Mach number is solved for, the rest of the fluid properties are determined using the “useful integral relations” explained earlier [32]. The fluid properties are then fed into the exit fluid port of the isolator which is connected to the burner element.

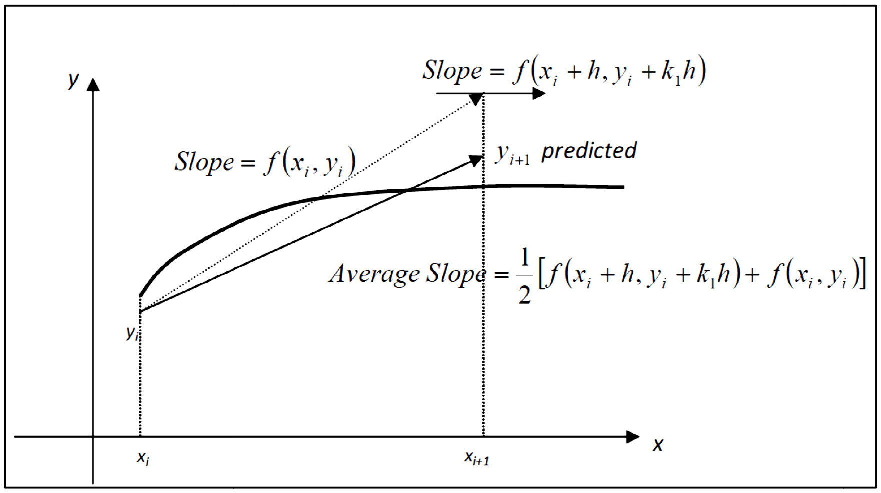

y
$Slope = f(x_i + h, y_i + k_1 h)$
$Slope = f(x_i, y_i)$
y_{i+1} predicted
$Average\ Slope = \frac{1}{2}[f(x_i + h, y_i + k_1 h) + f(x_i, y_i)]$
y_i
x
x_i
x_{i+1}

Figure 4-2 Runge Kutta Second Order Method Graphical Representation (Heun’s Method) [48]

4.2.3 On-Design Combustion Chamber NPSS code (Station 3 – 4)

At the design point, for a constant pressure combustor, the combustion is expected to be smooth without any boundary-layer separation and the pressure change due to fuel injection and burning should be within separation limits [34]. The design-point code is based on this theory and it also combines the stream thrust analysis and one dimensional analysis for higher fidelity numerical analysis. The stream thrust analysis equations are solved initially to obtain the area distribution of the combustion chamber.

The equations that are solved for are also determined from modifying the one-dimensional conservation of mass, momentum and energy equations and are presented as follows [2]

$$V_4 = V_3 \left\{ \frac{1 + f \cdot \dfrac{V_{fx}}{V_3}}{1+f} - \frac{C_f \cdot \dfrac{A_w}{A_3}}{2\left(1+f\right)} \right\} \tag{4.19}$$

$$T_4 = \frac{T_3}{1+f} \left\{ 1 + \frac{1}{C_{pb} T_3} \left[\eta_b f h_{PR} + f h_f + f C_{pb} T^o + \left(1 + f \cdot \frac{V_f^2}{V_3^2} \right) \frac{V_3^2}{2} \right] \right\} - \frac{V_4^2}{2C_{pb}} \tag{4.20}$$

$$\frac{A_4}{A_3} = \left(1+f\right) \cdot \frac{T_4}{T_3} \cdot \frac{V_3}{V_4} \tag{4.21}$$

Thus from the above set of equations, the temperature ratio and the area ratio of the constant pressure combustion chamber is obtained. The average specific heat at constant pressure C_{pb} is used only in the sections to determine the area ratio of the component. The rest of the flow variables that are provided to the exit fluid port of the element are based upon one-dimensional analysis using the influence coefficient method [32]. In order to carry out the generalized one-dimensional flow analysis by Shapiro [32], prior knowledge of $A(x)$ and $T_t(x)$ in the combustion chamber is necessary. The distribution of $A(x)$ is already determined using the stream thrust analysis in the previous set of calculations. But determining the distribution of $T_t(x)$ is complex because it is dependent on the rate of combustion heat release, finite rate of mixing and chemical reactions [2]. Prior experimental experience and knowledge plays a great role in this part of modelling. From the data available from previous experiments, it is well known that both mixing and chemical heat release are greatest at onset and relax asymptotically towards their respective fully mixed and chemical equilibrium zero rate values with infinite convective time or distance [2]. This results in the value of axial total temperature gradient dT_t/dx to be greatest after ignition usually near burner entry and decreases monotonically to its least value at burner exit [2]. This behavior can be validated by seeing the trend of supersonic combustion from the plot below from reference [49]

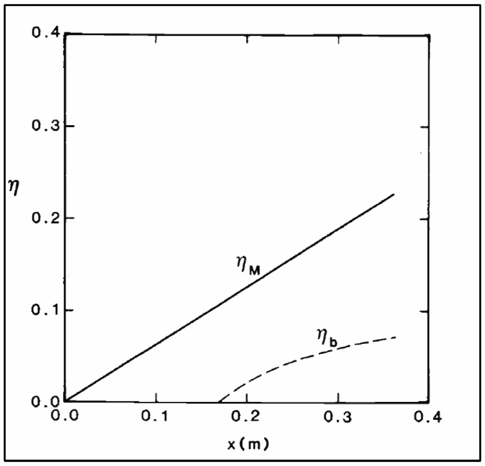


Figure 4-3 Axial Distributions of Mixing Efficiency and Burning Efficiency in a Supersonic Combustion Chamber [49]

This pattern is not only seen in experimental work but also seen in the numerical analysis such as in the work of Tran [31] on the supersonic combustion chamber. With this approach the amount of fuel that is allowed to react with air at a particular station is dictated by a mixing efficiency curve which is given by [34]

$$\eta_c(x) = \eta_{c.tot}\left[\frac{\vartheta X(x)}{1+(\vartheta-1)X(x)}\right] \qquad (4.22)$$

where $X(x) = (x - x_3)/(x_4 - x_3)$ and $\eta_{c.tot}$ is the combustion efficiency at the end of the combustor and ϑ is an empirical constant of order 1 to 10 and which depends on the rate of mixing [34]. The distribution of total temperature in the combustion chamber due to fuel mixing and chemical energy heat released can be modelled using the following relation [34]

$$T_t(x) = T_{t3} + \left(h_{PR} f_{st} \phi \eta_c(x) - dQ\right) \big/ C_p \quad (4.23)$$

The internal drag due to fuel jets and mass addition due to fuel injection can be neglected since they are secondary in importance [2]. When modelling in NPSS, the ALLFUEL thermo package is used in the combustion chamber element for this research. The major advantage of using NPSS for one-dimensional simulation can be pointed towards the level of combustion characteristics and calculations it performs. An in-built function called “burn ()” provides a one-dimensional temperature profile of mixed and burnt fuel when provided with an incoming mass flow rate from the exit of the compression system and the total amount of fuel injected to the combustion chamber with the efficiency of combustion. Since the combustion efficiency curve was calculated earlier using equation (4.21), all the parameters are available to use the “burn ()” function to calculate the chemical heat energy released. The thermodynamic package also outputs the fraction of fuel burnt from the total amount of fuel injected for a particular FAR. From the fraction of burnt fuel, the fraction of unburnt fuel is calculated and that quantity becomes the input for the next step in the axial direction along with the respective combustion efficiency. This method takes the gas mixture dissociation as well as the mass flow rate added to the stream of air into account and makes the variable C_p model possible for one-dimensional analysis. Hence a separate term to take the change of mass flow rate into account can be neglected from the generalized one-dimensional equation [32].

With the distribution of $A(x)$ and $T_t(x)$, we can calculate the Mach number distribution $M(x)$ of the combustion chamber from [32, 47]

$$\frac{dM}{dx}=\frac{M\left(1+\frac{\gamma-1}{2}M^2\right)}{1-M^2}\left(-\frac{1}{A(x)}\frac{dA}{dx}+\frac{\gamma M^2}{2}\frac{4C_f}{D_h}+\frac{1+\gamma M^2}{2}\frac{1}{T_t(x)}\frac{dT_t}{dx}\right) \tag{4.24}$$

The Runge-Kutta solver, which was built and explained for the on-design code of the isolator, was used to solve the above equation for the Mach number distribution. A completely supersonic combustion is expected throughout the combustion chamber without any loss in static pressure. The rest of the flow variables are calculated using the useful integral equations from Shapiro [32]. The variables are passed on to the hypersonic expansion element using exit fluid port of the combustion chamber.

4.2.4 On-Design Expansion Element NPSS Code (Station 4 – 10)

The expansion nozzle (station 9) is combined with the underside of the aft of the vehicle which are often known as “expansion ramps” or “single expansion ramp nozzles (SERNs)” to form the expansion component of the hypersonic vehicle [2]. The expansion ramps are used to minimize the weight of the expansion system but they also perform the task of extracting thrust from the high-enthalpy, high-speed gas expelled from the combustion chamber. The design of the nozzle is based on the two-dimensional method of characteristics and assumes that the exit profile of gases from station 4 is uniform [2]. From the stream thrust analysis, the following set of equations is used to determine the preliminary flow variables and the geometry [2]

$$T_{10}=T_4\left\{1-\eta_e\left[1-\left(\frac{p_{10}}{p_0}\frac{p_0}{p_4}\right)^{\left(R/C_{pe}\right)}\right]\right\} \tag{4.25}$$

$$V_{10}=\sqrt{V_4^2+2C_{pe}\left(T_4-T_{10}\right)} \tag{4.26}$$

$$Sa_{10}=V_{10}\left(1+\frac{RT_{10}}{V_{10}^2}\right) \tag{4.27}$$

$$\frac{A_{10}}{A_0} = \left(1 + f\right) \cdot \frac{p_0}{p_{10}} \cdot \frac{T_{10}}{T_0} \cdot \frac{V_0}{V_{10}} \tag{4.28}$$

The above set of equation yields the area ratio of the expansion ramp of the hypersonic vehicle. By using the nomenclature for designing an ideal, minimum length, two dimensional expansion nozzle, the angle of deflection and length of the expansion system of DMScramjet can be determined [2]. A representation of the different zones formed by reflection of characteristics is presented below for an example case of a uniform entry Mach number of 2 [2].

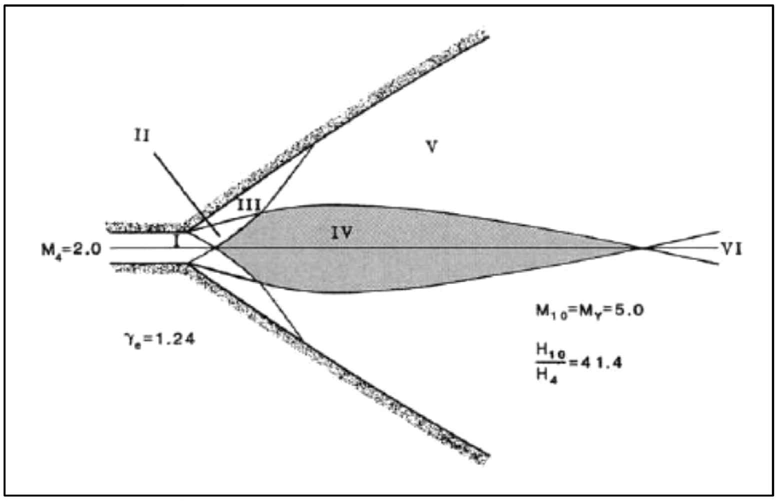


Figure 4-4 Diagram Showing Different Zones formed by Reflecting Characteristic for an Expansion Nozzle Designed by Method of Characteristics [2]

1) Zone I Mach wave angle [2]

$$\mu_4 = \arcsin\left(\frac{1}{M_4}\right) \tag{4.29}$$

2) Zone VI Mach wave angle [2]

$$\mu_{10} = \arcsin\left(\frac{1}{M_Y}\right) \tag{4.30}$$

3) Ratio of expansion component height to entry component height from mass flow rate parameter (MFP) [2]

$$\frac{H_{10}/2}{H_4/2} = \frac{M_4}{M_Y}\left\{\frac{1+\frac{\gamma_e-1}{2}M_Y{}^2}{1+\frac{\gamma_e-1}{2}M_4{}^2}\right\}^{(\gamma_e+1)\big/2(\gamma_e-1)} \tag{4.31}$$

4) Ratio of Zone VI axial length to entry height [2].

$$\frac{L_{VI}}{H_4\big/2} = \frac{1}{\tan\mu_{10}}\cdot\frac{H_{10}}{H_4} \tag{4.32}$$

From the above set of equations, the area ratio and the length of the expansion unit can be obtained. After determining the dimensions of the expansion chamber, one-dimensional analysis for the flow with influence of friction and change in area is performed to determine the axial Mach number distribution of the flow through the expansion component to the atmosphere [2]. The ALLFUEL thermodynamic package is used to calculate the intermediate thermodynamic properties and the flow is frozen at the end of the combustion chamber. The following equation is solved by using the Runge-Kutta solver [32]

$$\frac{dM^2}{M^2} = -\frac{2\psi(M)}{1-M^2}\frac{dA}{A(x)} + \frac{\gamma M^2\psi(M)}{1-M^2}\frac{4C_f}{D_h}dx \tag{4.33}$$

where $\psi(M) = 1+\frac{\gamma-1}{2}M^2$

After determining the flow variables along with the Mach number [32], they are fed into the fluid exit port of the expansion element to be connected to the flowend element. The flowend element is added for the purpose of architecture of NPSS assembly solver to denote the termination of the cycle. It does not perform any additional calculations. This ends the on-design code calculation of DMScramjet engine for flight Mach number $M_0 = 6$.

4.3 Coupling of NPSS and FLUENT

Before explaining the off-design NPSS component codes, the methodology adopted for communication between ANSYS FLUENT and NPSS is explained in this section. Three general approaches to multidisciplinary and multifidelity simulations; namely loosely coupled, coupled process, and multiphysics, provide a comprehensive collection of capabilities for multidisciplinary aeropropulsion analysis [53]. In the loosely coupled approach existing disciplinary simulations are run, and data is generated and made available and used for subsequent analysis [53]. The data must be in the correct format for implementation by the subsequent analysis but the subsequent code need not directly communicate with the previous code [53]. In the coupled process approach, individual disciplinary codes are used, similar to the loosely coupled approach, but the disciplinary codes need to run concurrently with each other [53]. The multiphysics approach addresses those applications whose characteristics require that the disciplines be coupled at the fundamental equation level to accurately, or more efficiently, capture the multidisciplinary physics of the problem [53].

The coupled process approach is selected for this research since it was the best suited approach for the manner the off-design codes were built. A function named “fluentinput ()” which uses the “outstreamfunction ()” was built to output the necessary flow variables and initialize the flow from the start of the inlet. The “outstreamfunction ()” is an inbuilt function in NPSS which notifies the assembly solver to output necessary data available in the element to an external source file [42]. The output file which is produced by using the fluentinput () function is written in a particular UDF template that is readable by the FLUENT.

A user-defined function, or UDF, is a C function that can be dynamically loaded with the ANSYS FLUENT solver to customize boundary conditions, material property definitions, surface and volume reaction rates, source terms in ANSYS FLUENT transport equations, source terms in user-defined scalar (UDS) transport equations, diffusivity functions etc. [52]. In this case, the output file writes the flow variables in uniform profile UDF (DEFINE_PROFILE) format that is used to define the boundary profile that varies as a function of spatial coordinates or time [52]. F_PROFILE is typically used along with DEFINE_PROFILE and is a predefined macro supplied by ANSYS FLUENT which stores a boundary condition in memory for a given face and thread and is nested within the face loop [52]. Three arguments namely t, f and i are required by the profile UDF [52]. F_PROFILE uses the thread pointer t, face identifier f, and index i to set the appropriate boundary face value in memory.

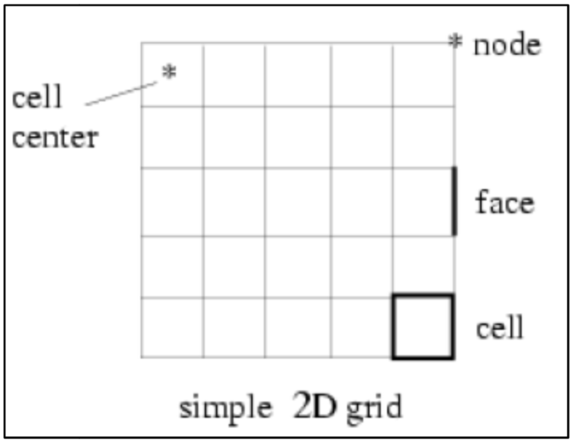


Figure 4-5 Diagram Showing Different Zones where the Values Can Be Stored in a Grid using UDF in FLUENT [52]

Once the fluid simulation is complete and the residual norm is within required limits, the required data at the outlet of the respective element are averaged. Some of the information contained in the multi-dimensional flow is inevitably lost when any one-dimensionalization technique is applied [54]. A variety of methods exist for extracting performance measures from multi-dimensional data sets [54]. Weighted methods, particularly the mass flux-weighted approach, consistently predicts the qualitative trends present in the multi-dimensional data [54]. The mass flux weighted approach is defined as [54]

$$\phi_{averaged} = \frac{\int \phi\left(\rho\left(\vec{v}\cdot\vec{n}\right)dA\right)}{\int \rho\left(\vec{v}\cdot\vec{n}\right)dA} \qquad (4.34)$$

where ϕ is the quantity to be averaged. The flow variables that are averaged are total temperature, total pressure, Mach number and mass flow rate since with these values, a local flowstation in the NPSS element can calculate the rest of the flow variables. These averaged flow variables from the ANSYS FLUENT solver are output to a file. A function named fluentoutput () which uses the instreamfunction () of NPSS is built to extract the required data from the output file of the FLUENT solver. The instreamfunction () is a built in function which is used to open an external file and extract the necessary information from it [42]. This approach is followed throughout this research, whenever NPSS and FLUENT are communicating with each other.

4.4 Off-Design Code for DMScramjet Engine

In developing a high-speed inlet system, the main objectives should be to define a geometry that provides efficient shock compression, produces near-uniform flow to the combustor, generates low drag and provides these characteristics over a wide range of flight operating conditions, i.e. good `off-design' performance [50]. Since the on-design case has many approximations and estimations such as constant thermodynamic properties, constant pressure combustion chamber, shock free isolator etc, during design phase calculations using stream thrust analysis, the off-design code plays an important role in realistic performance assessment of the hypersonic vehicle. In the off-design operation mode, the area of the components which were obtained from the on-design code are fixed throughout the analysis. This replicates the practical scenario where components are designed for a particular case and its performance without much alteration is checked for the off-design cases. The NPSS code developed for this purpose is explained in the following sections.

4.4.1 Off-Design Compression System Code (Station 0 – 3)

The airflow captured by the inlet largely depends on the vehicle velocity and the engine’s ability to pass the mass flow to the nozzle [1]. With supersonic acceleration, the capture is larger than the inlet airstream and the exceeding mass flow is diverted through spillage around the inlet’s capture [1]. The amount of mass flow rate entering the inlet after spillage cannot be judged accurately with any analytical method. When the scramjet engine is operated in off-design condition, the shock on lip condition for which the inlet is designed for is not achieved since it is a fixed geometry inlet, and determining the amount of mass flow spilled is of high importance in order to achieve a right proportion of fuel-air ratio in combustion chamber. The shock cancellation shoulder also fails during off-design conditions, causing the inlet oblique shocks to enter the flow path of the engine. These events make CFD a vital tool to be used for off-design simulations. The off-design simulation of the hypersonic compression system is completely taken over by the ANSYS FLUENT tool. For the off-design analysis case, the inlet and isolator are merged together to form a single component for the initial run, before the fuel is injected to the stream of air. The off-design code starts with the ambient element and the inletstart element to initiate the flow to the inlet. The scramjet inlet switches to “off-design” mode using the switch variable available in NPSS. After entering the off-design mode, the “fluentinput ()” function is executed which passes over the values of the ambient flow conditions to the FLUENT solver. After the flow solution is converged, the fluentoutput () function is executed to read the passed on averaged variables back into scramjet inlet element. These variables are then passed into the next element.

4.4.2 Off-Design Isolator (Station 2 – 3) and Combustion Chamber (station 3 – 4) Code

For the off-design analysis code in NPSS, the isolator subelement is embedded inside the combustion chamber element. The incoming variables from the compression system along with the fuel mass flow rate from the fuelstart element, are taken to be the input for the combustion chamber element. The ODE (eq 4.23) which is solved to obtain the axial Mach number distribution along the length of the combustion chamber has a singularity when the flow Mach number approaches unity. This singularity can be visualized as the thermal choking point in the combustion chamber environment, after which no more mass can be added without inlet unstart. A subelement named “ThermalChoking” was built to determine if the pressure rise created by the fuel combustion is strong enough to choke the flow. The ThermalChoking subelement is coded with a root finder which works using the secant method. It works just like the Newton Raphson method, but uses the succession of roots from the secant line to get better approximation of the roots for the function. The secant method is given by the recurrence relation below

$$x_n = x_{n-1} - f\left(x_{n-1}\right)\frac{x_{n-1} - x_{n-2}}{f\left(x_{n-1}\right) - f\left(x_{n-2}\right)} \qquad (4.35)$$

If the root determined is within the length of the combustion chamber, then the engine is operating in a ramjet mode and the combustion chamber contains the thermal choking point. A subelement called dmdx was built to solve for the value of the function at the neighborhood of the sonic point using secant method. If the engine operates in the ramjet mode, then the limiting value of the function at the neighborhood of the sonic point is determined using the Bean's procedure [47]. The formula to obtain the value of the slope of the Mach number near sonic point is given by [47]

$$\left(\frac{dM}{dx}\right)^2 = -\frac{\gamma+1}{8}\left[-2\frac{d}{dx}\left(\frac{1}{A}\frac{dA}{dx}\right)_{M=1} + \gamma\frac{d}{dx}\left(\frac{4C_f}{D_h}\right)_{M=1} + (1+\gamma)\frac{d}{dx}\left(\frac{1}{T_t}\frac{dT_t}{dx}\right)_{M=1}\right] - \frac{\gamma+1}{8}\left(\frac{dM}{dx}\right)_{M=1}\left[2\gamma\frac{4C_f}{D_h} + 2\gamma\left(\frac{1}{T_t}\frac{dT_t}{dx}\right)\right]_{M=1} \tag{4.36}$$

A subelement called dmdx was built to solve for the value of the function at the neighborhood of the sonic point using the secant method [47]. Then, starting from the sonic point location, the Runge Kutta solvers built earlier are used to march in the negative x direction from the sonic point to the entrance of the combustion chamber. With this back integration, the subsonic Mach number at the entrance of the combustion chamber is determined. This coincides with the existence of the normal shock train in isolator. To compute the engine thrust correctly, it is important to determine the stagnation pressure loss across the pre-combustion shock train (PCST) that will exist in the isolator under certain conditions [55]. Only three methods can be used to evaluate the strength of a precombustion shock which are empirical relations from successful experimental results [2,34], constant impulse function method of Heiser and Pratt [2], and the full CFD solution [55]. All three methods have been implemented in the off-design burner and isolator subelement code. The A variable switch asks the user whether the calculation should be only through NPSS ("EMPIRICAL") or if a higher fidelity simulation should be involved to validate the results from NPSS ("FLUENT").When the EMPRICAL switch is selected, the isolator calculates the separation length of the boundary layer to compensate the pressure rise using Ortwerth's separation correlation [56, 34]. Based on a large amount of experimental data at different Mach and Reynolds numbers, and in different duct geometries, Ortwerth [56] determined that the rate of pressure rise (diffusion) in a duct is directly proportional to the dynamic pressure of the incoming flow and the skin friction coefficient at the initial point of separation in the duct, and inversely proportional to the duct hydraulic diameter [34]. The correlation is given as follows [56, 34]

$$\frac{dP}{dx} \approx \frac{89}{D_h} C_{fo} \left(\frac{\rho v^2}{2} \right) \qquad (4.37)$$

where C_{fo} is the coefficient at the initial separation point [34]. This correlation is used to predict the length of the separation of the boundary layer. The pressure rise by the shock train is determined by the constant impulse function method which is a result of control volume analysis on the separated flow in the isolator [2].

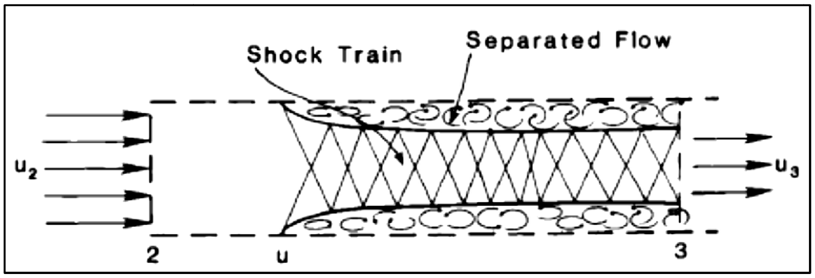


Figure 4.6 Control Volume Analysis of the Confined Flow at the Exit of an Isolator [2]

The upper limit of the pressure rise capacity of the isolator is set by an equivalent single normal shock which would have been the case if there was no boundary layer. As far as the pressure rise by the isolator is less than the pressure rise by a single normal shock the precombustion shock would remain in the isolator. Since the flow gets separated when a shock train resides in an isolator, the wall friction can be neglected in the region of the flow reversal inside the separated flow [2]. With the neglected skin friction, the axial stream impulse function, $I = pA + \dot{m}u$ is constant at the entry and the exit of the isolator [2]. Since the separation velocity is either zero, or negligible compared to the supersonic inviscid flow, the impulse function at the exit of the isolator is given by

$$I_3 = \left[p_3 \left(A_2 - A_{3c} \right) + 0 \right] + \left(p_3 A_{3c} + \dot{m}u_3 \right) = p_3 A_2 + \dot{m}u_3 \tag{4.38}$$

where the subscript c denotes the inviscid core flow [2]. From the one dimensional conservation of momentum equation, the pressure rise due to the shock train can be given as,

$$\frac{p_3}{p_2} = 1 + \gamma M_2^2 - \gamma M_2 M_3 \sqrt{\frac{1 + \frac{\gamma - 1}{2} M_2^2}{1 + \frac{\gamma - 1}{2} M_3^2}} \tag{4.39}$$

A line monitor is created at the end of the inlet in the FLUENT compression system analysis and the flow variables at the end of the inlet are taken into the burner combustion system using the “fluentoutput ()” function and data ports in NPSS. Now with the knowledge about both entry and exit of the isolator, the pressure rise by the normal shock train and the separation length of the boundary layer to support the compression is calculated. The length of the isolator required is calculated using the normal shock theory since it gives the required margin for the shock train to settle within the isolator. The static pressure rise that can be obtained with a normal shock stationed in the isolator for a particular incoming Mach number is given as [2]

$$\frac{p_3}{p_2} = \frac{2\gamma}{\gamma + 1} {M_2}^2 - \frac{\gamma - 1}{\gamma + 1} \tag{4.40}$$

Using this pressure rise, the length of the isolator required is calculated using the experimental correlation obtained by Waltrup and Billig [57, 34] given by

$$\frac{L}{H}=\frac{\sqrt{\frac{\theta}{H}}}{\sqrt[4]{\mathrm{Re}_\theta}}\cdot\frac{\left\{50\left(\frac{p_3}{p_2}-1\right)+170\left(\frac{p_3}{p_2}-1\right)^2\right\}}{\left(M_2^2-1\right)} \qquad (4.41)$$

The above correlation reflects the influence of the most pertinent physical parameters, namely the incoming Mach number, back pressure ratio, inlet boundary momentum thickness θ and Reynolds number based on momentum thickness Re_θ [2]. When the isolator has reached the maximum possible pressure rise limit, the separation of the boundary layer extends till the entrance of the isolator after which the shock system is expelled through the inlet leading to an unstart. Hence the momentum thickness is obtained for the separated flow. With the entrance Mach number known, the ODE in equation (4.23) is integrated forward using Runge Kutta solvers, with substituting the limiting value of the function near the region of thermal choking to obtain the flow variables distribution throughout the length of the combustion chamber [47]. The FLUENT switch is selected when higher fidelity results of the shock train are required. The compression by a shock train tries to match the highest static pressure level set by combustion which we can see from the following diagram [2].

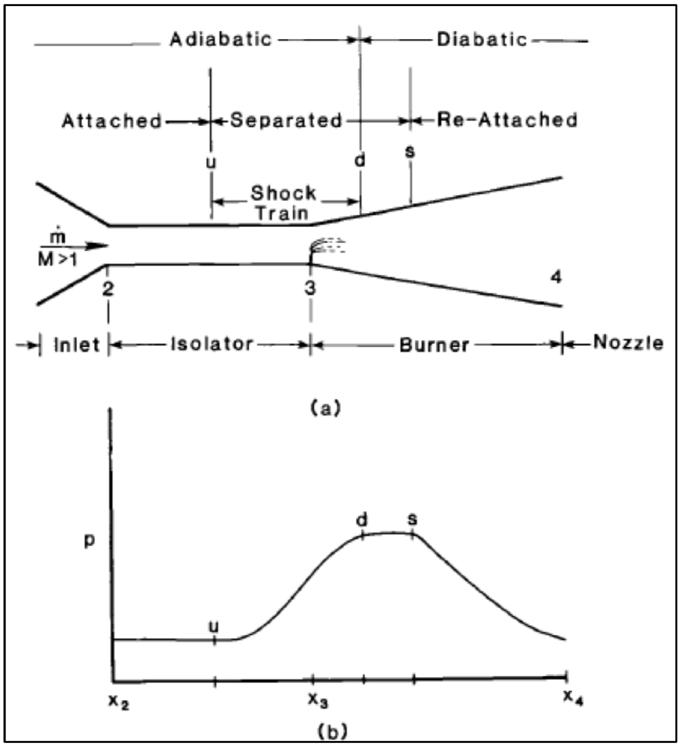


Figure 4-7 Isolator Description (a) Designation of Axial Locations (b) Axial Distribution of Wall Static Pressure throughout the system [2]

As we can see from the above figure, the maximum pressure rise produced by the shock train matches with the maximum pressure rise produced due to combustion [2]. So when the FLUENT switch is selected, the flow conditions at $station\ s$ is fed to the FLUENT solver which is loaded with the geometry and mesh of isolator through the isolator subelement. The flow profile from a monitor at the end of the inlet is used as the entrance flow conditions for the isolator module. The data fetching and initializing are performed by the fluentinput () and fluentoutput () functions in the isolator subelement.

The flow simulation is then executed and the results are fetched back after the solution convergence to be used further inside the combustion chamber NPSS element.

In case of absence of a thermal choking point inside the combustion chamber, the combustion chamber code operates in a scramjet mode. When the fuel is added at the exit of the isolator, the sudden pressure rise due to combustion causes an adverse effect in the boundary layer region of the supersonic flow, since the dynamic pressure to withstand the large pressure gradient is lower when compared to the inviscid flow. When the pressure rise is greater than the incipient pressure rise limit by Korkegi for turbulent flow [41], the boundary layer separates due to adverse pressure gradient and a reversed flow prevails inside the separated region. This separation extends upstream through the flow path of the isolator creating a flow blockage. Since the inviscid region of the flow in the isolator is supersonic, the turning of the flow due to blockage results in oblique shock formation followed by reflection of the oblique shocks inside the core flow leading to an oblique shock train. The calculations performed by the off-design combustion chamber code for “EMPRICAL” switch is the same as for the normal shock train with neglecting the calculation for the length of the isolator, since the isolator is designed for the maximum pressure rise and that is achieved only in the ramjet mode due to low flight Mach number with relatively high backpressure. The “FLUENT” switch also follows the same procedure as explained earlier for ramjet mode. The only difference with the oblique shock train procedure is that, only the back pressure and the total temperature can be obtained through NPSS combustion chamber element. The flow Mach number after the oblique shock train can be obtained only after the FLUENT simulation since the backward integration cannot be performed for this mode of operation. After obtaining the averaged flow properties from the end of the isolator, the combustion chamber one dimensional calculations are completed and the flow properties are passed on to the off-design nozzle element.

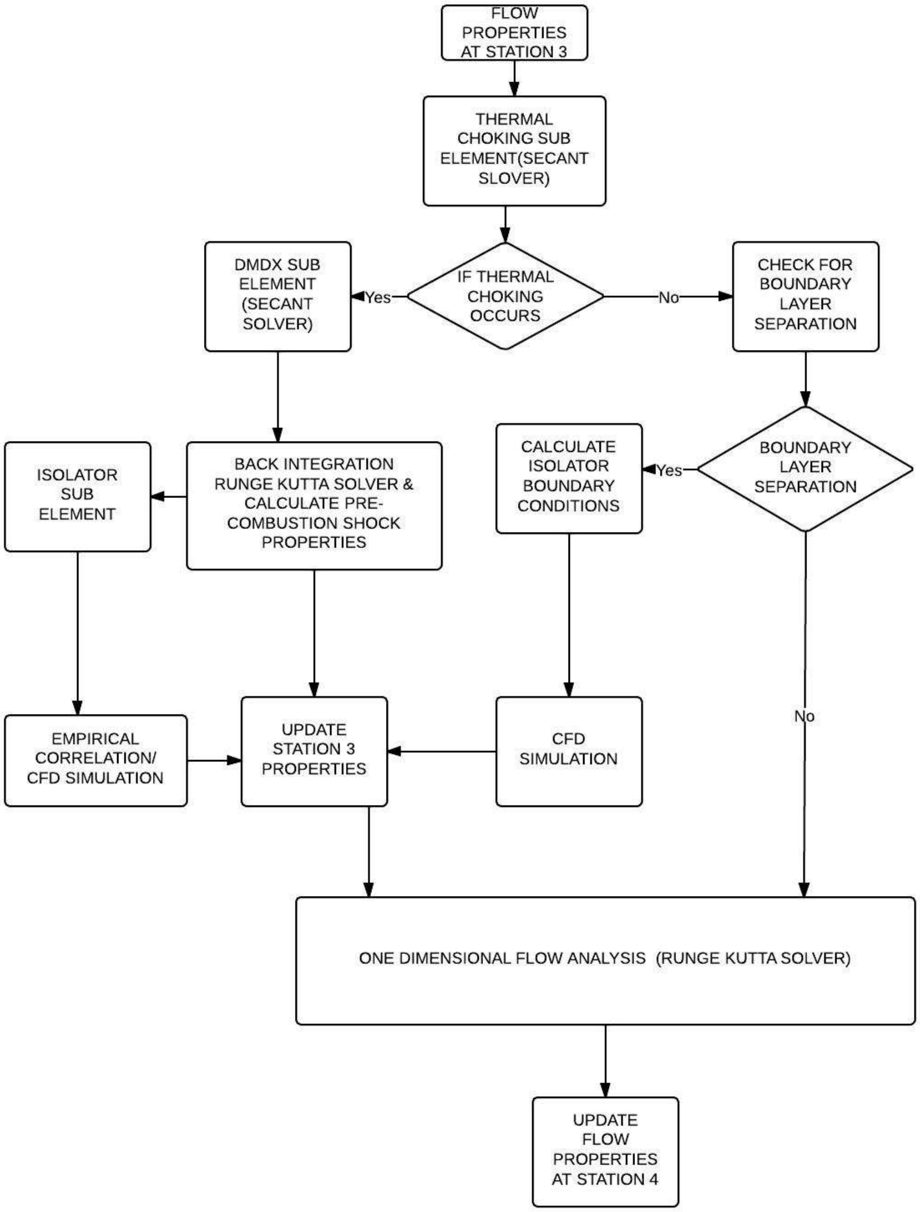

FLOW PROPERTIES AT STATION 3
THERMAL CHOKING SUB ELEMENT(SECANT SLOVER)
IF THERMAL CHOKING OCCURS
Yes
No
DMDX SUB ELEMENT (SECANT SOLVER)
CHECK FOR BOUNDARY LAYER SEPARATION
BOUNDARY LAYER SEPARATION
Yes
CALCULATE ISOLATOR BOUNDARY CONDITIONS
ISOLATOR SUB ELEMENT
BACK INTEGRATION RUNGE KUTTA SOLVER & CALCULATE PRE-COMBUSTION SHOCK PROPERTIES
No
EMPIRICAL CORRELATION/ CFD SIMULATION
UPDATE STATION 3 PROPERTIES
CFD SIMULATION
ONE DIMENSIONAL FLOW ANALYSIS (RUNGE KUTTA SOLVER)
UPDATE FLOW PROPERTIES AT STATION 4

Figure 4-8 Flowchart of Methodology Followed by Offdesign NPSS Burner

4.4.3 Off-Design Expansion Element (station 4 – 10) Code

The off-design expansion component code fixes the area variation of the nozzle and external ramp from the on-design code. Once the linear area distribution is fixed, the generalized one dimensional flow equation with the influence of friction and change in area [32] is solved using a variable C_p model with the help of ALLFUEL thermodynamic package. The off-design code assumes no discontinuities in the flow path and freezes the flow at the exit of combustion chamber. After the calculation is complete the fluid properties are passed on to the flowend element to terminate the cycle. Thus, the off-design NPSS elements built for DMScramjet can be summarized in Fig. 4.9. The flowchart provides an overall picture of NPSS components built and assembled for this research. The following section explains about the approach and results obtained from using the created engine code.

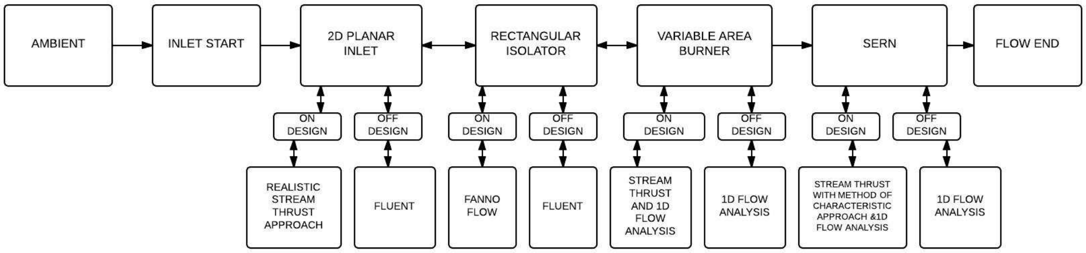


Figure 4-9 Flowchart of DMScramjet Engine Assembly built for this Research

Chapter 5

Results

In the previous chapter, the calculation methodologies and theory followed by the NPSS elements, which was built for this research were discussed along with the role played by CFD simulation. The majority of the role played by the NPSS, in this research, was to provide a fast and realistic flow simulation. In order to predict the geometry required for the one dimensional flow analysis and perform numerical calculation, parameters such as the number of shock waves, inlet turning angle etc should be supplied to the NPSS code based on few other initial calculations. Additionally, iterations for the final dimensions of the inlet and isolator were involved to get an ideal operating case for the on-design condition as well as the maximum possible performance in the off-design conditions. The methodology followed for those initial calculations, iterations and the results obtained after the full cycle simulations are presented in this chapter of the paper.

5.1 Results from On-Design DMScramjet code

As explained earlier, the system of shockwaves selected for the compression system of a scramjet plays an important role in the compression efficiency and level of compression achieved. The 2 D geometry of a planar inlet can be achieved only after fixing the shockwave system. Of the universe of possible compression component configurations, the special families chosen for compression have two distinct features that they are characterized by [2]. The first is that a cardinal number of oblique shockwaves are available to produce a specified cycle static temperature ratio ψ at a specified free stream Mach number, and when more than one oblique shock is involved in producing the required temperature ratio, they must provide equal amounts of geometric turning of the flow [2]. They will be evaluated at the shock on lip design point condition [2]. The first step towards the design process was to decide on the area ratio of the compression system based on which the oblique shock family was selected. Higher compression can be achieved for inlets operating at high Mach numbers as can be seen in the graphical representation of contraction ratio versus flight Mach number in reference [29]. The goal of the design process is to achieve the best 2D planar hypersonic inlet performance at Mach 6 which starts at Mach 3.5 for the turbojet/ramjet transition. This provoked the initial step to determine the maximum possible contraction limit for the lowest operating Mach number of the DMScramjet engine. The maximum contraction ratio limit is presented again as follows.

$$\left(\frac{A_4}{A_0}\right) = 0.05 - \frac{0.52}{M_0} + \frac{3.65}{M_0^{\ 2}} \tag{5.1}$$

For M_0 = 3.5, $\left(\frac{A_4}{A_0}\right)_{\max} = 5.015$

Having discovered the $A_4/A_{0\,\max}$ at Mach 3.5 which sets the upper limit beyond which compression is not possible, the focus of the design shifts to the on conditional operation of the inlet. The on-design engine cycle code built for this purpose was used for this analysis. Even though the design approach by Heiser [2], which starts by determining the set of oblique shockwaves for necessary cycle temperature ratio is not completely followed, the system of shockwave is still planned to have equal number of flow turnings so that the flow into the combustion chamber is parallel to the axial direction to reduce the internal drag of the engine. Having equal number of flow turning also gives the opportunity to design a mixed compression inlet with minimum inlet ramp angle. Six oblique shocks were selected since, the more the number of oblique shocks is created to achieve a particular temperature ratio, the weaker they are which leads to less total pressure loss. After few iterations using the on design NPSS engine code, for an angle of 3.65^o and six turnings, the compression area ratio $\left(A_4/A_0\right)$ of 4.78 was obtained. Any increase in the magnitude of the flow turning angle resulted in the increase in contraction ratio beyond the maximum contraction ratio limit. The flow properties undergoing compression through the shockwave system (station 0 – 1) are presented below

Table 5-1 Oblique Shock System Properties using On-Design NPSS Inlet Code

Shockwave Cardinal number	M_j	$\frac{T_j}{T_i}$	$\frac{P_j}{P_i}$	$\frac{P_{tj}}{P_{ti}}$	$(S_j - S_i)/C_p$
1	5.49	1.16	1.68	0.98	4.04×10^{-3}
2	5.06	1.33	`2.71	0.97	7.21×10^{-3}
3	4.68	1.52	4.22	0.96	9.73×10^{-3}
4	4.34	1.71	6.37	0.95	1.17×10^{-2}
5	4.04	1.92	9.35	0.95	1.34×10^{-2}
6	3.77	2.13	13.41	0.94	1.47×10^{-2}

The station i represents the property of the flow before a shockwave and station j represents the property of the flow after passing through the shockwave in the table. The parameters are multiplied with the previously obtained results to get the overall performance of the shocksystem. The above table results using inviscid oblique shock analysis with variable C_p through local flow station. It can be concluded that the shock system has a total pressure ratio π_c of 0.94 and adiabatic compression efficiency of $\eta_{c_{shocksystem}} = 0.89$. The dimension in the z axis of the inlet is fixed to be constant [25] and the aspect ratio (AR) of the inlet is taken to be 4 [23]. With the width of the inlet fixed, the height ratio of the inlet can be directly obtained from the area ratio. The height of the isolator is taken to be 0.152 m from reference [25] since it was also a successful hydrocarbon fueled scramjet engine design. The initial length of the isolator was chosen to be 0.912 m from reference [23] and the final length was determined to be 2.02 m after a number of iterations. The reason for iteration will be explained in the off-design section of isolator. The NPSS uses English units to perform the calculations and output them on screen or a file and the values are converted to SI units to be presented in this research. The kilo units are presented with up to three decimals and the other units are presented with up to two decimals. The second control volume analysis for inlet with assumption of constant pressure (station 1 – 2) was performed and the results are presented as follows [2].

Table 5-2 Station 2 Flow Properties After Realistic Stream Thrust Analysis Approach

Station number i	M_i	$p_i\ (kPa)$	$T_i\ (^oK)$	$P_{t_i}\ (kPa)$	$T_{t_i}\ (^oK)$
2	3.19	25.027	545.67	2881.06	1662.31

After the second control volume analysis, the static temperature ratio T_2/T_0 achieved by the inlet was 2.45 and the adiabatic compression efficiency was equal to 0.805. The value of the dimensionless boundary layer skin friction $\left(\left(C_f/2\right)\cdot\left(A_w/A_2\right)\right)$, was taken to be 0.01 which is usually within a range of 0.01 – 0.05 [2]. The approximation was based on the fact that the shocks formed by the inlet are weak oblique shocks which can be seen from the dimensionless entropy change after each shockwave, and their influence on the boundary layer is expected to be in the lower scale of the range of dimensionless boundary layer skin friction. The total static temperature rise of the compression system includes the temperature rise of the core flow by influence of friction. Hence above properties are taken in by the isolator subelement to perform the one dimensional Fanno flow analysis. The results of the 1D analysis are provided below.

Table 5-3 Station 3 Flow Properties After One Dimensional Fanno Flow Approach

Station number i	M_i	$p_i\ (kPa)$	$T_i\ (^oK)$	$P_{t_i}\ (kPa)$	$T_{t_i}\ (^oK)$
3	2.81	34.779	711.95	1622.47	1662.31

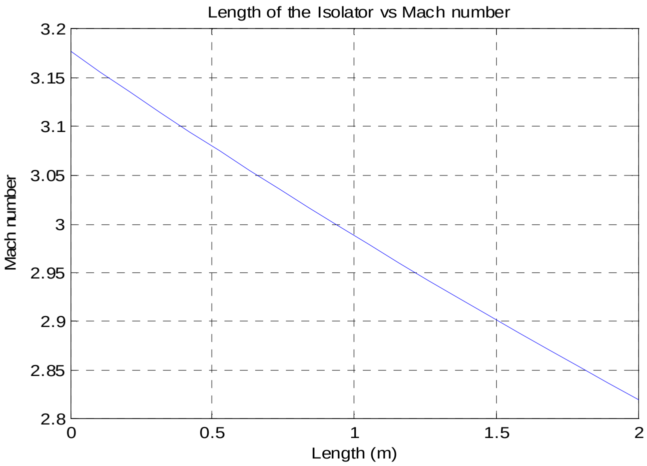


Figure 5-1 Mach number distribution in Isolator through Fanno flow approach

The flow properties are then passed over to the combustion chamber element. The maximum fuel air ratio at which the engine is operated at design point is 0.0674 (which is the stoichiometric fuel to air ratio of JP-7 fuel). Hence the design point equivalence ratio (ϕ) is taken to be 1.0. Even the off-design case analysis is performed using the same fuel equivalence ratio. The combustion chamber element first uses the stream thrust analysis concept of a constant pressure combustion chamber and the following results were obtained.

Table 5-4 Station 4 Flow Properties After Stream Thrust Analysis

Station number i	M_i	$p_i\ (kPa)$	$T_i\ (^oK)$	$P_{t_i}\ (kPa)$	$T_{t_i}\ (^oK)$
4	1.46	34.779	2420.24	193.743	3040.32

The area ratio of the combustion chamber $\left(A_4/A_3\right)$ obtained through stream thrust analysis was equal to 3.88. By implementing the burner effective drag coefficient value in the stream thrust analysis, the code takes account of various parts the cause form drag such as fuel injectors, mechanical mixing devices, structural members and steps [2]. But the code does not take the reactant gas dissociation due to large chemical energy being imparted into thermal energy into account. This drawback is eliminated by solving the one dimensional generalized flow equation by Shapiro [32] using Runge Kutta solvers.

Since the mixing and combustion of the fuel is handled by the combustion efficiency curve, whatever length of the combustion chamber is chosen, the combustion at the end of the chamber is expected to be completed by the efficiency of combustion provided to the code. The work of Tourani [58] was referred to obtain the length of the combustion chamber where two successful scramjet combustion chambers are analyzed using 2D numerical techniques. The dimensions of the combustion chamber were taken from the physical dimensions of DLR and NASA□s SCHOLAR scramjets which operated on hydrogen fuel and the length of the combustion chamber was 1.5 meters [58]. Since the hydrocarbon fuel has a longer ignition delay and slower burning [25] than the hydrogen fuel, an additional margin of 0.5 meters was added with the length of the combustion chamber for realistic approximation and the length of the DMScramjet combustion chamber for this research was taken to be 2 meters. The length of the combustion chamber can be varied according to the required mission since the total temperature profile changes automatically along with it. With the length and area distribution of the combustion chamber available, the on-design combustion chamber code solves the Mach number distribution along the axis using the burn() function of NPSS to obtain the temperature profile from the thermodynamic package for the given amount of fuel and combustion efficiency along with the incoming Mach number. The results from the one dimensional analysis are presented below

Table 5-5 Station 4 Flow Properties after One Dimensional Combustor Analysis

Station number i	M_i	$p_i\ (kPa)$	$T_i\ (^oK)$	$P_{t_i}\ (kPa)$	$T_{t_i}\ (^oK)$
4	1.57	45.028	2275.73	175.758	2871.69

It can be seen that, when the gas dissociation is taken into account, the flow variables have a noticeable change with reduction in the temperature from the profile resulted using stream thrust analysis. Because the dissociation results in a specific heat increase and the enthalpy is essentially unchanged for dissociated or nondissociated gas, it means that the temperature decreases when dissociation is considered, which is expected because energy is absorbed in the dissociation process from the translational energy [1]. This loss also makes a constant static pressure combustion chamber not possible for the chosen area distribution with 24.64 % loss in the static pressure estimated by stream thrust analysis.

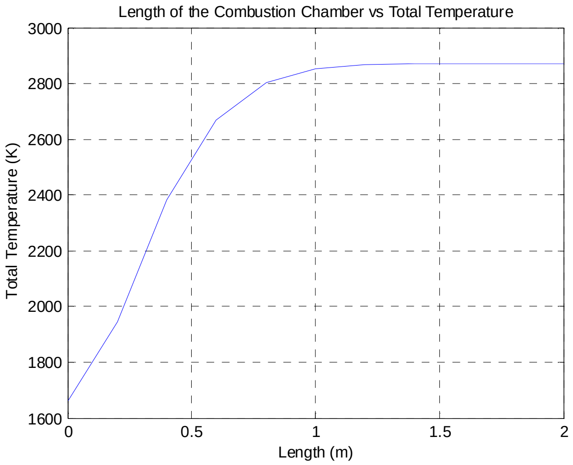


Figure 5-2 Total Temperature Distribution in Combustion Chamber using On-Design NPSS code

As expected in supersonic combustion, there is a sudden change in the total temperature of the flow after combustion initiation and the gradient gradually relaxes as the distance from injection increases. The total temperature is evaluated by creating a separate flow and fuel station in the burner on-design subelement and the entire mass flow rate of the fuel is fed into the incoming air stream. Instead of using the efficiency curve, the total combustion efficiency, $\eta_c = 0.90$ is provided directly as an input to the burn () function. The final total temperature is checked with the total temperature obtained through the combustion efficiency curve to check if the curve induces any error and corrected accordingly. Similar total temperature distribution pattern is also observed in an one dimensional scramjet analysis program called VTMODEL in the work of Tran [31].

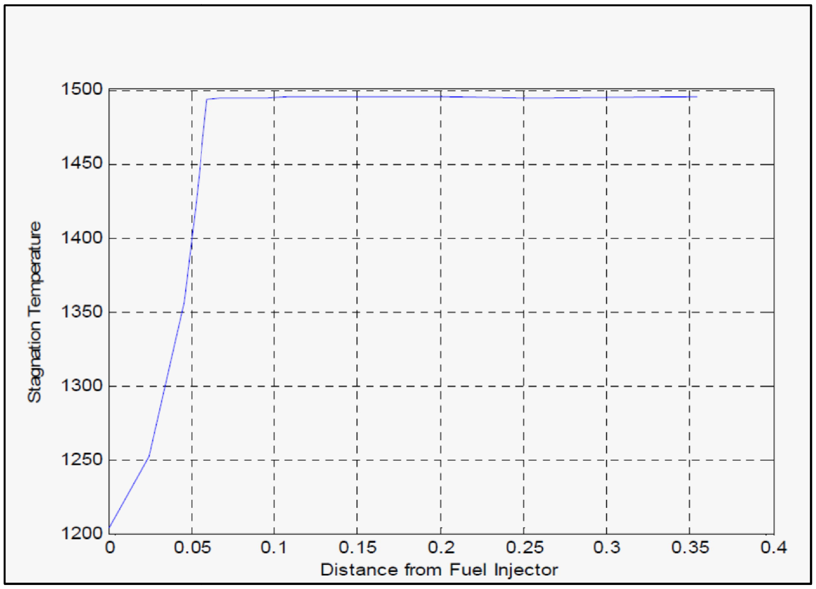


Figure 5-3 Total Temperature Distribution in Combustion Chamber using VTMODEL code [31]

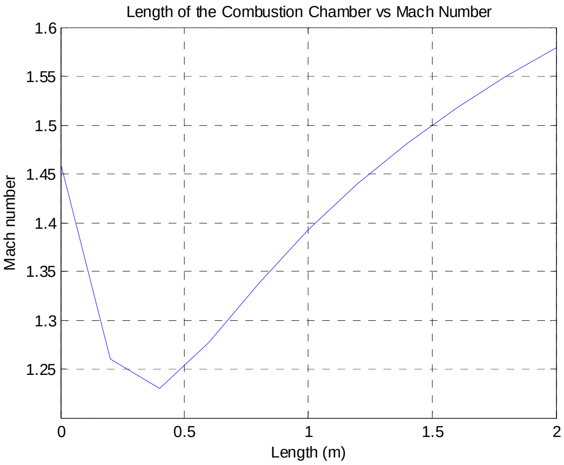


Figure 5-4 Mach Number Distribution in Combustion Chamber using On-Design NPSS code

After the flow variables are calculated using one dimensional analysis, they are passed on to the expansion component.

The expansion component also operates similar to the combustion chamber where it first uses the stream thrust analysis and method of characteristics approach to get the dimension of an ideal nozzle, and then performs one dimensional flow analysis with influence of area change and friction on the obtained geometry. The code freezes the flow at the exit of the combustion chamber. Often the nozzle must be shorter than the ideal expansion would require so that the flow then remains under expanded and some thrust is consequently lost [1]. The gain in nozzle efficiency in these cases is considered unjustified when the additional weight required is considered [10]. Hence in this case an under expanded nozzle $\left(P_{10}/P_0 = 0.8\right)$ which is 20% under expanded is aimed to be designed using the stream thrust analysis. The results from stream thrust analysis are as follows;

Table 5-6 Station 10 Flow Properties after Stream Thrust Analysis

Station number i	M_i	$p_i\ (kPa)$	$T_i\ \left({}^oK\right)$	$P_{t_i}\ (kPa)$	$T_{t_i}\ \left({}^oK\right)$
10	3.05	1.493	1532.14	108.141	3040.32

The method of characteristics analysis results in area ratio $\left(A_{10}/A_4\right)$ of 6.15 and length $\left(L_{10}\right)$ of 5.43 meters. With these geometric values fixed and skin friction drag coefficient $\left(C_f\right)$ chosen to be 0.002 [2], the one dimensional analysis results are presented as follows

Table 5-7 Station 10 Flow Properties after One Dimensional Analysis

Station number i	M_i	$p_i\ (kPa)$	$T_i\ (^oK)$	$P_{t_i}\ (kPa)$	$T_{t_i}\ (^oK)$
10	3.76	0.864	986.47	134.307	2871.69

The flow properties vary between stream thrust and one dimensional analysis since, the one dimensional analysis uses the properties for which gas dissociation was taken into account with the ALLFUEL thermodynamic package. A variable γ model is used in the nozzle too with local flow stations. The distribution of Mach number along the length of the expansion unit is given below.

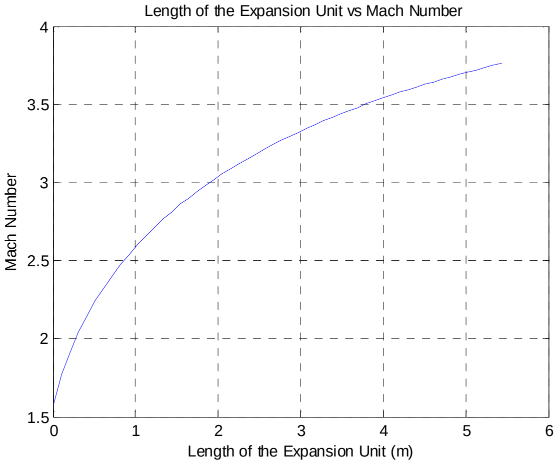


Figure 5-5 Mach number Distribution in Expansion System using On Design NPSS code

In case the thrust produced by this engine is not sufficient for a particular mission, the results can be improved by developing a low weight ideal SERN. Having presented the output from the on-design DMScramjet code in NPSS, the next stage of developing the inlet model is presented in the following sections.

5.2 Design of 2D Planar Inlet

The initial dimensions of the scramjet inlet are available from the previous analysis. The dimensions of the inlet are as follows

Table 5-8 Initial Inlet Dimensions

$Length_{inlet}$	9.002 m
$Length_{cowl}$	5.605 m
$Width_{inlet}$	0.608 m
$Height_{entry}$	0.726 m
$Height_{throat}$	0.152 m
Number of Flow Turns	6
Angle of Turn	3.65^{o}

A CFD simulation for the inlet with the above geometry was executed for the accuracy. Two different mesh resolutions were used for the CFD analysis of inlets throughout this research. A coarse mesh of 136500 elements was used to obtain initial solutions and a fine mesh of 243180 elements was used with solution-based (normalized pressure gradient) dynamic mesh adaptation to obtain the final results for all the inlet FLUENT cases. Only results from the fine mesh are presented in this paper. An example of the mesh adaptation at the region of high pressure gradient is shown below.

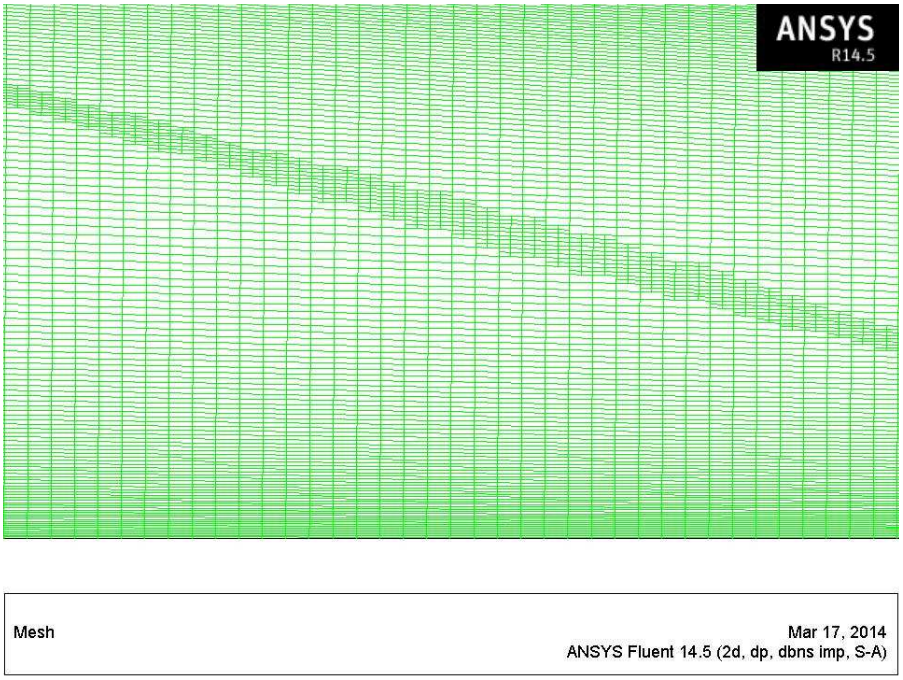
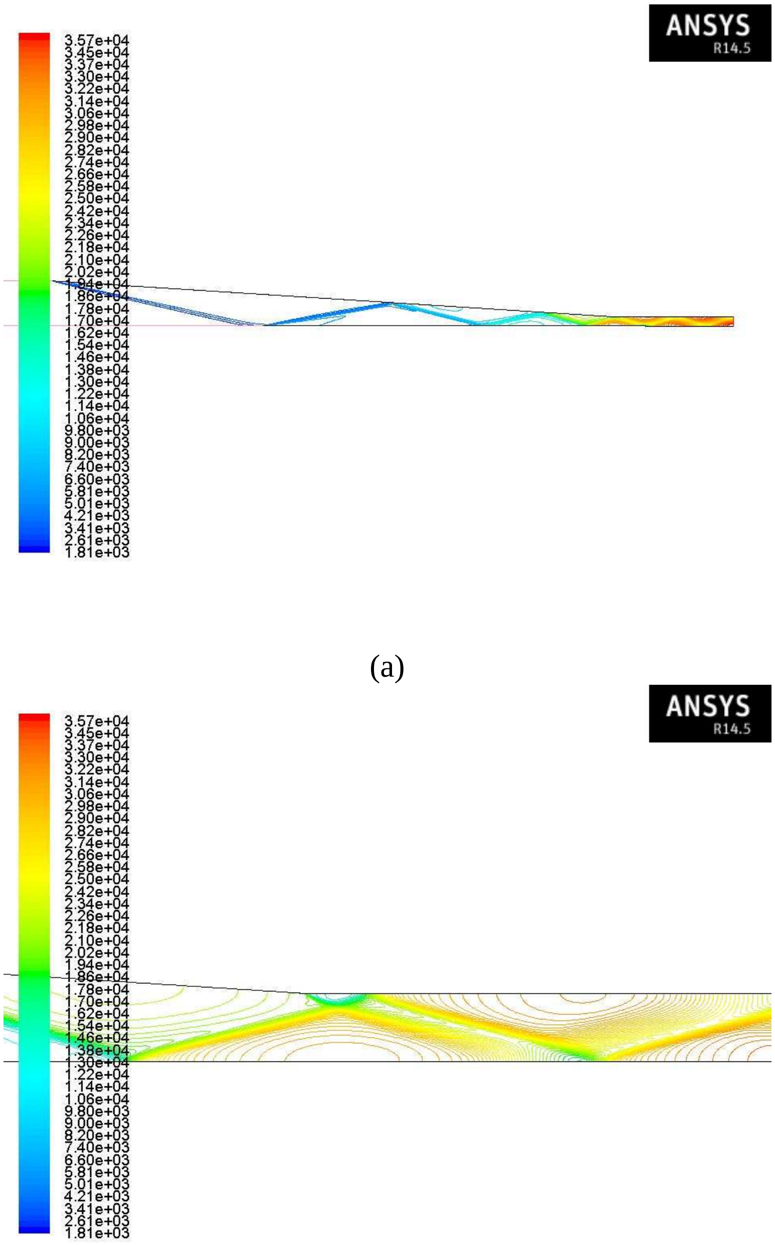


Figure 5-6 Solution Based Gradient Adaption Performed on Inlet Grid

A thermally perfect gas with variable C_p model was modeled in FLUENT solver by using the seventh order polynomial equation which is a function of temperature from reference [60]. Though the isolator component is included for the simulation, its performance is not evaluated at this point. It was added for the reason of checking shock reflections and penetrations beyond inlet. The static pressure contour is chosen so that the type of wave (either a compression wave or an expansion wave) propagating through inlet and isolator can be determined.



(a)

(b)

Figure 5-7 Flow Contour of Compression System At $M_o = 6.0$, (a) Static Pressure, (b) Static Pressure with Shock Reflections inside Isolator at $M_o = 6.0$

From the above contours, it can be seen that, even though the area contraction ratio is sufficient to produce the six oblique shockwaves, it does not cancel the final shockwave at the shoulder of the throat, leading to unwanted shock reflections entering the isolator and combustion chamber. This creates additional distortions in the exit profile of the compression system. Hence additional case runs were performed to cancel the final shockwave by increasing the contraction ratio to the maximum possible. This resulted in an inlet with a height $\left(H_{inlet}\right)$ of 0.77 meters and contraction ratio of 0.506.

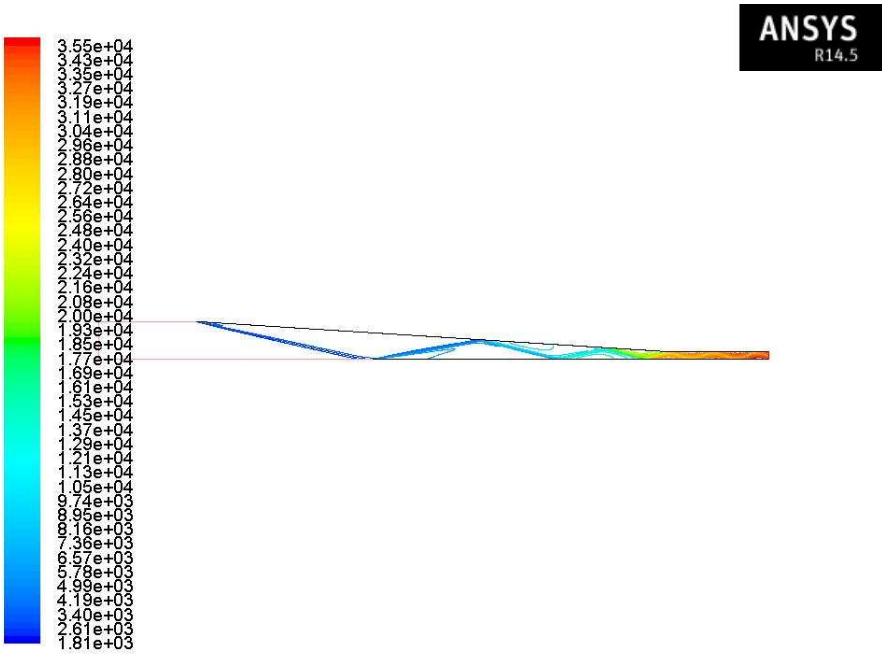
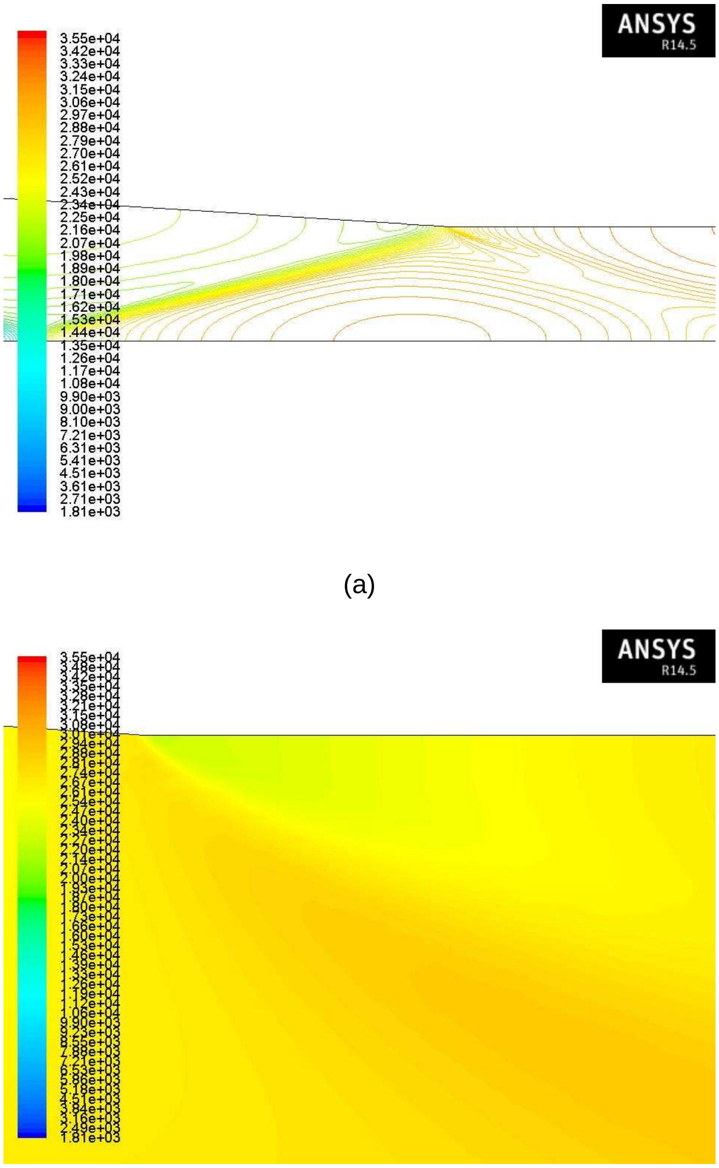


Figure 5-8 Static Pressure Contour of Compression System at $M_o = 6.0$ after Optimization



(a)

(b)

Figure 5-9 Flow Contour of Compression System at $M_o = 6.0$ (a) Static Pressure Contour Showing Shock Cancelled at Throat Shoulder at $M_o = 6.0$, (b) Filled Static Pressure Contour of Inlet Throat after Optimization at $M_o = 6.0$

A small disturbance at the end of the shoulder was due to an emanating expansion wave which arises to change the flow parallel to the axial direction at the sharp corner. Though the shock reflection was controlled to the maximum, the contraction area ratio has exceeded the maximum contraction ration by a factor of 0.99%. Hence the inlet start was checked for the inlet flight Mach number of 3.5.

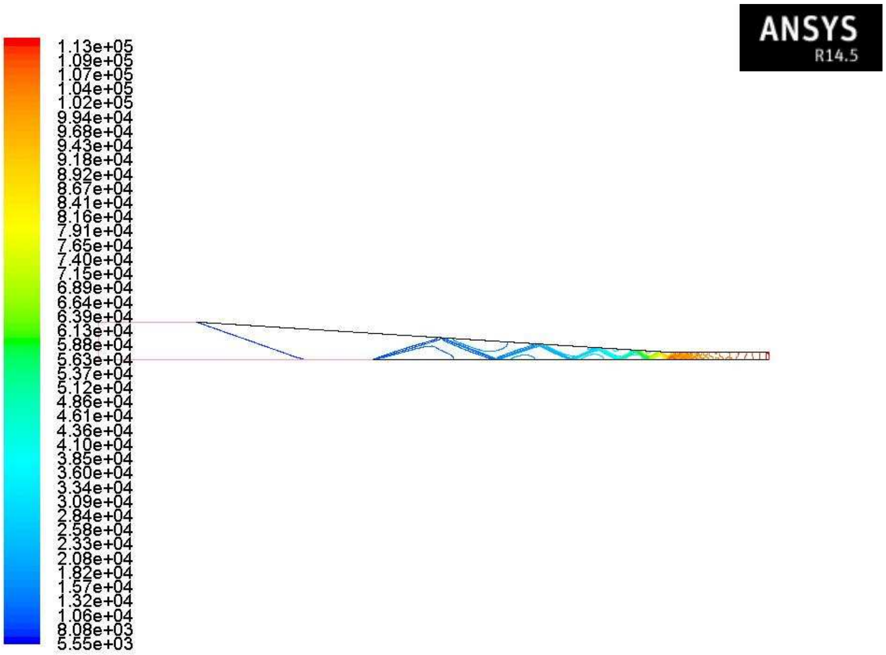


Figure 5-10 Static Pressure Contour of Compression System at $M_o = 3.5$ after Optimization

The above contour confirms successful operation of the DMScramjet inlet at flight Mach number of 3.5.

Table 5.9 Final Inlet Dimensions

$Length_{inlet}$	9.687 m
$Length_{cowl}$	6.083 m
$Width_{inlet}$	0.608 m
$Height_{entry}$	0.770 m
$Height_{throat}$	0.152 m
Number of Flow Turns	6
Angle of Turn	3.65^{o}

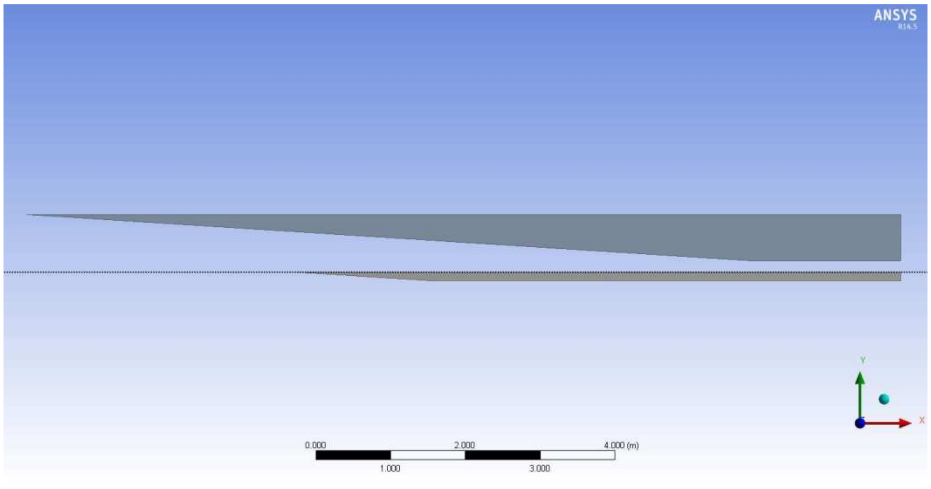


Figure 5-11 XY plane view of Generic Scramjet Inlet and Isolator Designed for Current Research

5.3 Inlet Unstart

Having discussed the empirical relations used to predict unstart of the hypersonic inlet, the maximum contraction ratio is predicted using pseudo transient CFD simulation in this section. Determination of the exact contraction ratio at which the inlet unstarts is necessary for a scramjet engine which is planned to operate over a wide margin as in this research. The incoming flow is simulated for a flight Mach number of 3.5 with thermally perfect C_p gas model. The turbulence production parameters are provided in the form of inlet hydraulic diameter and turbulence intensity. For internal flows, the turbulence intensity at the inlets is totally dependent on the upstream history of the flow [59]. A function called TubulenceIntensity () was created to perform the required turbulence calculations. The following formula [59] is used to provide turbulence production data to the FLUENT solver.

$$I = \frac{u'}{u_{average}} = 0.16\left(\mathrm{Re}_{DH}\right)^{-1/8} \qquad (5.2)$$

The inlet unstarted at contraction ratio $\left(A_3/A_0\right)_{unstart}$ of 5.26. The results from the simulation are found below

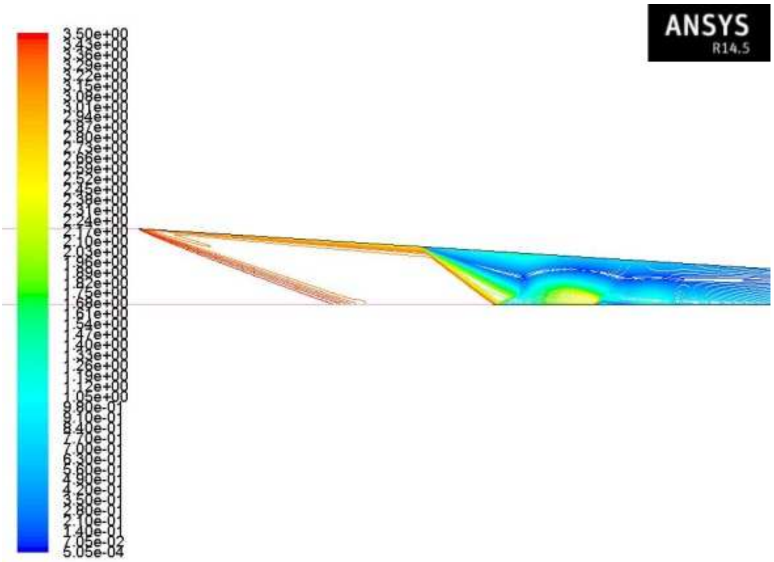
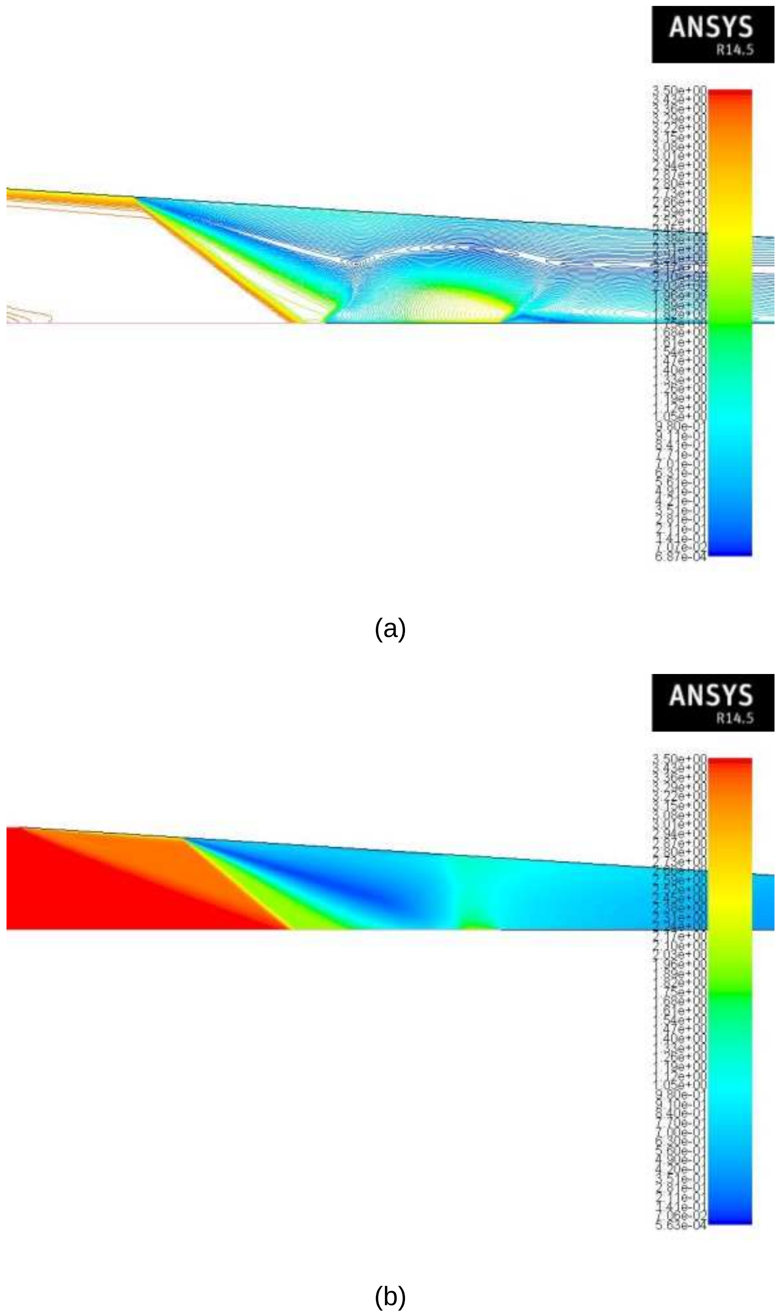


Figure 5-12 Initial Stage of Inlet Unstart at $M_o = 3.5$



(a)

(b)

Figure 5-13 Mach Number Contour of Compression System at $M_o = 3.5$ (a) Intermediate Stage of Inlet Unstart, (b) Filled Contour at Final Stage of Inlet Unstart

The above Mach number contour shows the system of shockwaves propagating towards the entrance of the inlet leading to unstart. The exit Mach number from the isolator becomes subsonic with a value of 0.85. The reason for the failure of the Kantrowitz limit to predict accurate maximum contraction ratio due to the presence of a complex shock system instead of a single normal shock at the cowl lip can be understood from the above Mach contours [29]. The area ratio predicted by CFD has a 4.75% additional margin over the experimental maximum contraction ratio correlation. Though the contraction ratio is well within the limit of the value predicted by flow analysis, further contraction to achieve maximum pressure ratio is avoided due to unwanted reflecting expansion waves in the isolator. As the throat flow distortion increases, the Mach number to choke the flow increases which again indirectly reduces the contraction ratio of the inlet [29]. Hence the contraction ratio of the inlet is fixed as explained in the previous section for minimum flow distortion at the exit of the compression system.

5.4 Off-Design Analysis and Transition

Once the inlet design is completed, the isolator component design is initiated. Since the isolator is a constant area duct, height and width of the isolator is same as that of inlet throat. The length of the isolator is the only and most import parameter in the design of isolator. In principle, the isolator adds weight, internal drag, and heat loads on the engine structure, and therefore its length must be limited to the minimum required by operability constraints [1]. But minimum lengths without some margin can lead to inlet unstart when there is a perturbation downstream in the combustion chamber. To determine the length of the isolator, the maximum pressure difference that is needed to be achieved by the isolator should be determined. This leads to an iterative approach on correcting the isolator length according to each flight Mach number and studying the backpressure variation accordingly. This also creates a necessity to determine the ram/scram transition Mach number. The off-design cases are split with Mach number interval of 0.5 from 3.5 to 6 for analysis purpose. The off-design compression chamber code relies upon the FLUENT result and does not perform any calculations other than inlet boundary condition to be fed. The methodology in which the length of the isolator is provided as follows

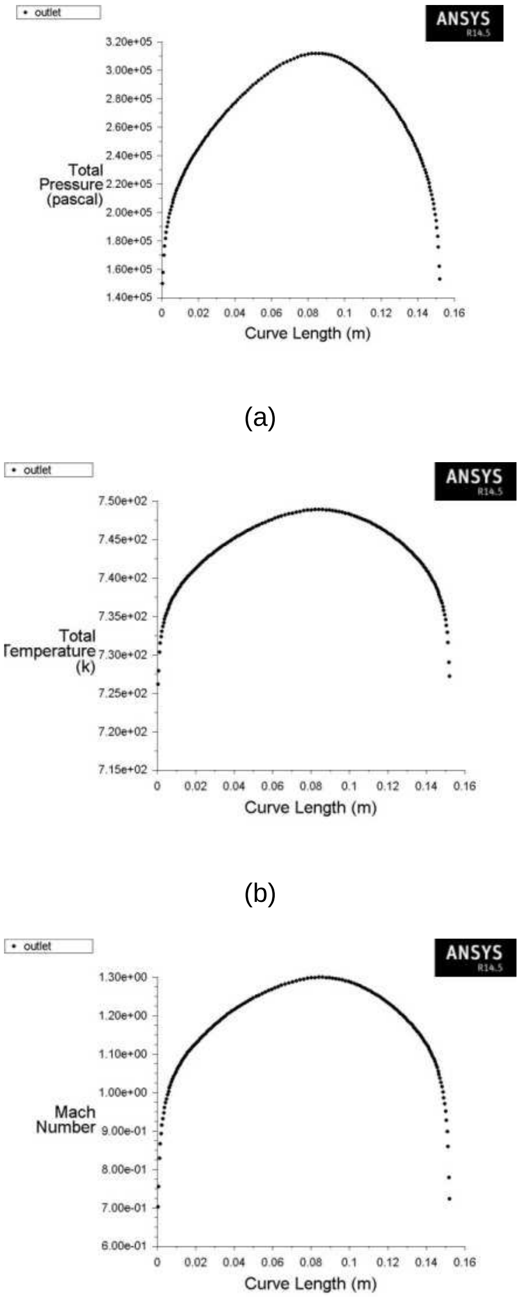
- The isolator with its initial length (0.914 m) [25] was attached to the end of the inlet, and 2D CFD analysis were performed to get the flow variables at the end of the compression unit.
- With the flow variables at the combustion chamber entry, the possibility of thermal choking and ramjet mode isolator is checked using the combustion chamber NPSS code.
- If there is a presence of thermal choking, the backward integration solvers using RK method are used to determine the entry flow variables by solving the one dimensional flow equations by Shapiro [32].
- With the knowledge of the flow properties at the entry and exit of the isolator, the length of the isolator and length of the separated boundary layer are determined using the correlations explained earlier.
- The length of the isolator is designed for maximum pressure rise at respective Mach numbers to allow margin for oscillation of the shock train.
- If the length of the isolator used for the CFD analysis is smaller than whatever is predicted by the NPSS code, then the length is increased by required proportions.

- The above steps are performed to all the off-design Mach number until the length predicted by the combustion chamber element in NPSS is smaller with respect to the length used for CFD analysis.
- The predicted length is validated using the CFD analysis by using the isolator subelement in the combustion chamber element.

After using the above cycle and performing a set of repetition, the length of isolator was obtained to be 2.02 meters at a ram/scram transition Mach number of 5.0. A fine mesh of 510×510 with solution based dynamic mesh adaptation was used the isolator CFD model throughout this research. The results for three particular cases are discussed elaborately while for other cases the results are published station wise in the following sections.

5.5 Case Study – Mach 3.5

The Mach 3.5 case was chosen for the detailed study since the starting characteristics of the engine was of high importance and the ramjet mode during this stage of the engine was also planned to be evaluated. Without a steady starting point operation, the engine may fail to operate successfully. The data from the ambient element is passed on to ANSYS FLUENT and the 2D steady state CFD simulation is performed. Just the mass flux weighted average of the three primitive variables; namely total pressure, total temperature, and Mach number along with mass flow rate from the exit of the isolator are passed on to NPSS burner element.



(a)

(b)

(c)

Figure 5-14 Flow Profile from Compression System at $M_o = 3.5$ (a) Total Pressure, (b) Total Temperature, (c) Mach Number

After the flow variables from the CFD analysis is passed on to the NPSS combustion chamber, combustion analysis are performed with a fuel equivalence ratio of ϕ-0.2 where the affinity of thermal choking to occur within the combustion chamber is checked. Any increase in the equivalence ratio can result in inlet unstart since the pressure rise that should be generated by the isolator exceeds the maximum value, $\left(P_3/P_2\right)_{\max} = 1.92$. A scenario which resulted in the formation of a normal shock for a back pressure of 187.279 kPa which was equivalent to $\left(P_3/P_2\right) = 1.87$ inside the isolator is presented below

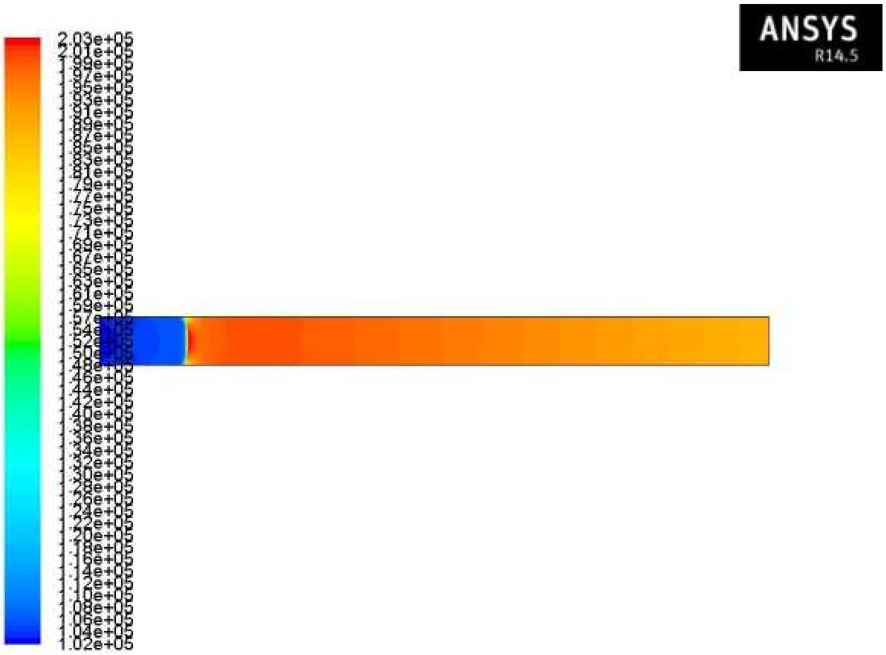


Figure 5-15 Filled Static Pressure Contour at $M_o = 3.5$ showing Normal Shock inside Isolator

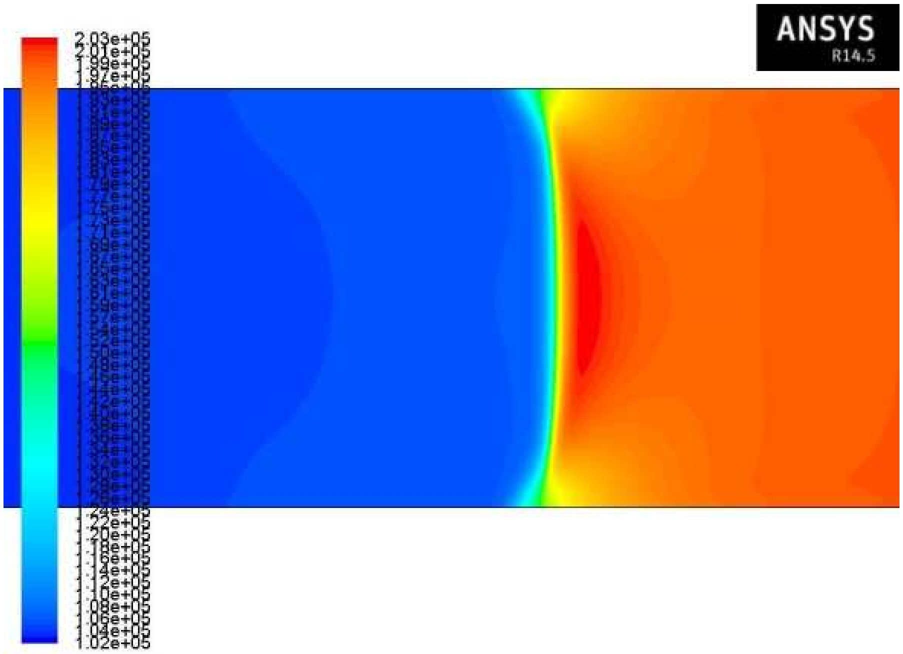


Figure 5-16 Filled Static Pressure Contour at $M_o = 3.5$ Zoomed for Normal Shock inside Isolator at $M_o = 3.5$

This normal shock eventually unstarts the inlet by escaping from the isolator entrance. The CFD simulation plays a good role in providing better upper limitations to ensure good performance of the engine. The presence of thermal choking inside combustion chamber was observed, with separation occurring at 0.28 m inside combustion chamber. The back integration of the Mach distribution leads to a combustion chamber entry Mach number of 0.861. The length of the separation predicted by Billig's correlation was equal to 0.39 meters and Ortwerth's correlation was equal to 0.40 meters. The FLUENT switch in NPSS is turned on during the simulation so that the length of the separation as well as the flow variables at the end of the shock train is obtained and compared with the values calculated using NPSS. Two types of simulation were run for the isolator element. First simulation was performed by splitting the domain symmetrically and performing the simulation for one half of the domain using uniform boundary conditions to obtain theoretical results. The symmetrical approximation is used since the numerical calculations do not take asymmetry of the shock train into account.

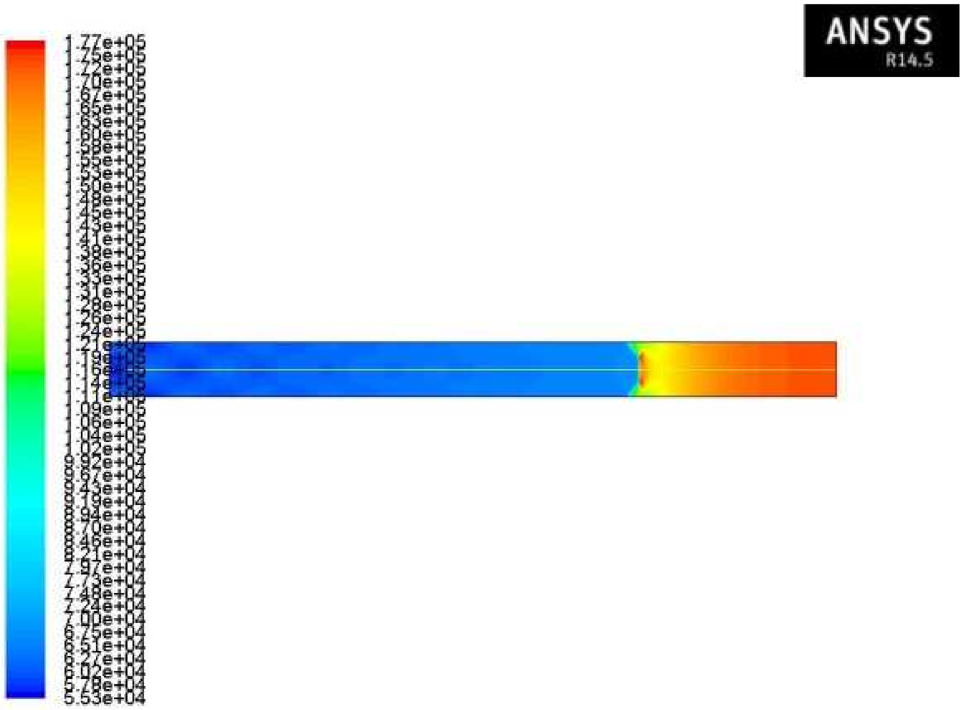


Figure 5-17 Filled Static Pressure Contour showing Normal Shock Train inside Isolator $M_o = 3.5$ with Symmetry

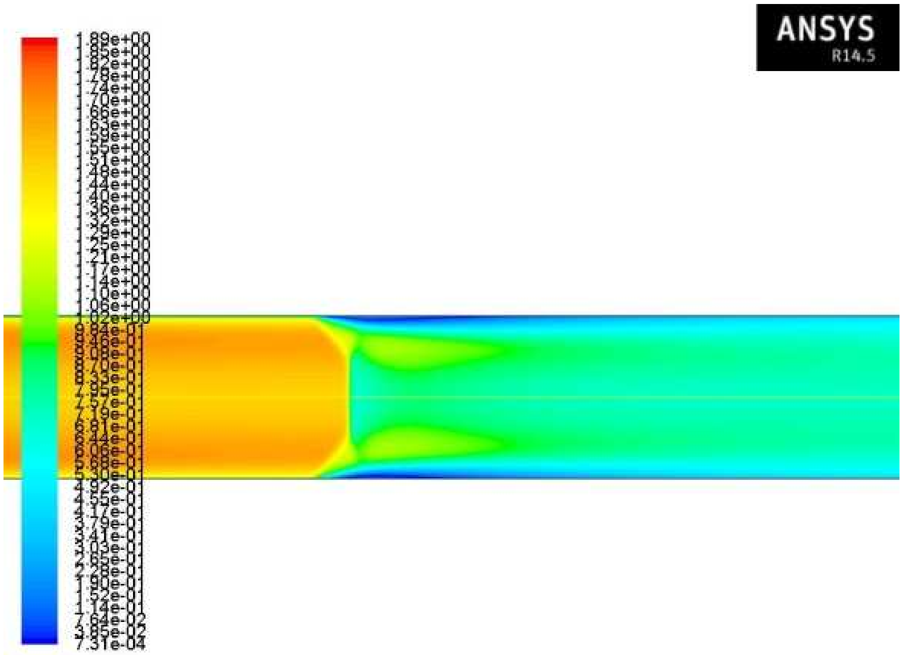
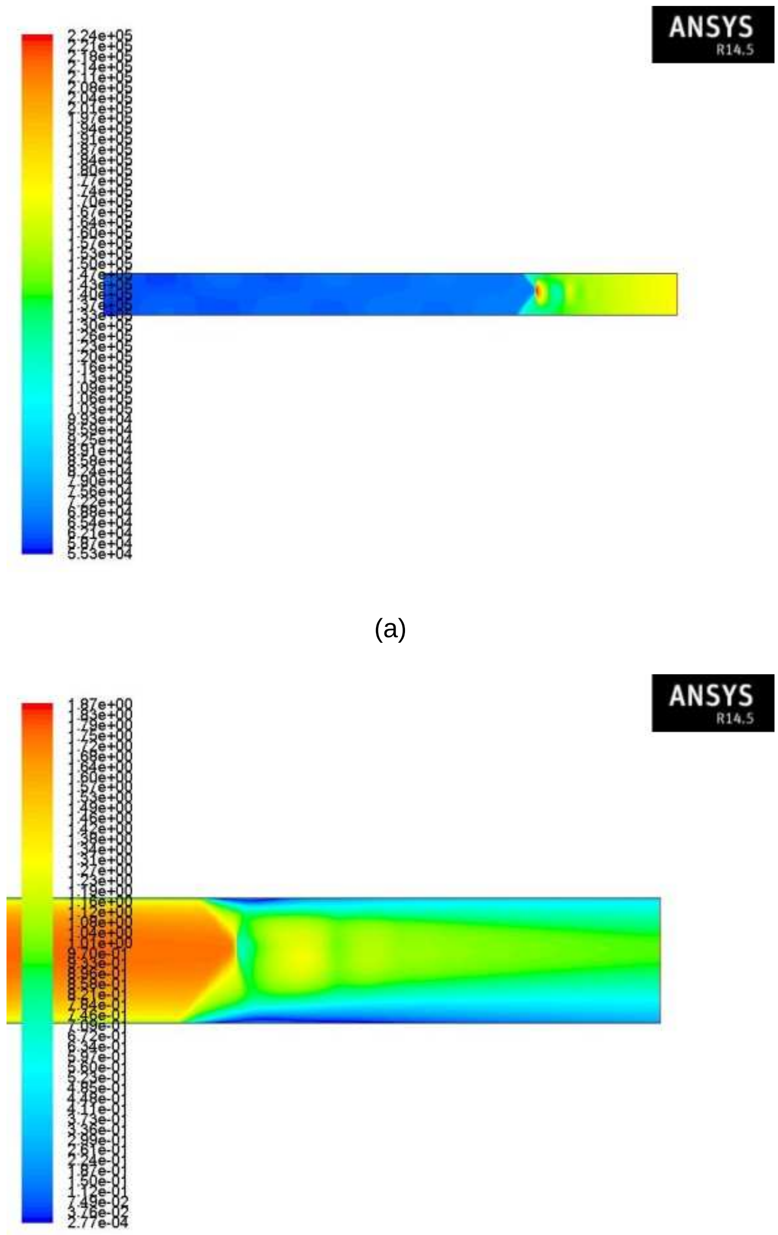


Figure 5-18 Filled Mach number Contour showing Normal Shock Train inside Isolator $M_o = 3.5$ with Symmetry

The results obtained had an averaged exit Mach number of 0.70 and the length of the shock train was equal to 0.59 meters. This result reveals that the leading shockwave is strong enough to produce the desired pressure ratio, with the secondary shockwaves less effective.

The second simulation was performed by using the flow profile from the end of the inlet without any symmetrical assumption to get realistic results. The results from the FLUENT simulation are provided below.

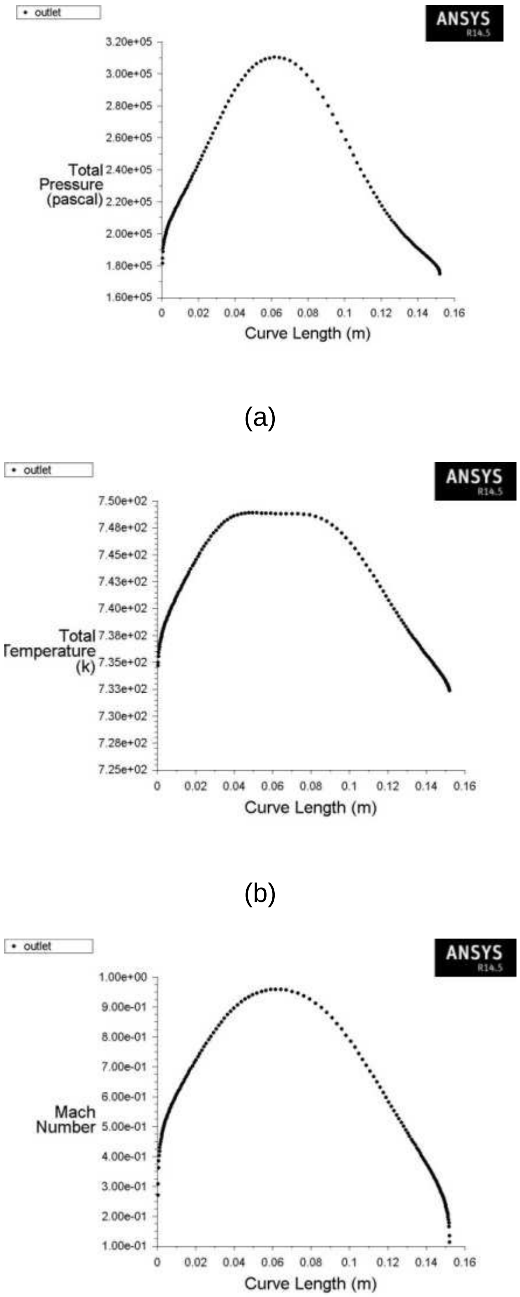


(a)

(b)

Figure 5-19 Filled Contour at $M_o = 3.5$ Showing Normal Shock Train inside Isolator without Symmetry (a) Static Pressure, (b) Mach Number

The simulation predicted a length of 0.57 meters from the exit of the combustion chamber which is longer than the length predicted by the analytical methods. A normal shock train which has asymmetrical development of boundary layer along the walls of the isolator in nature was obtained in the result. It can be seen that the initial normal shock bifurcates into two oblique shocks as it interacts with the separating boundary layer as explained in reference [2]. The shock train has several advantages over a single normal shock such as negligible total wall friction when compared with the stream thrust which makes analytical calculations easier and the flow properties are almost uniform along the exit plane of the isolator [2]. This simulation is more realistic since the boundary layer along the top of the isolator wall is different from the bottom wall which was also observed in the experimental work of Dutton and Carroll [61] as well as in the RANS CFD simulation performed by Baurle and Gaffney [62] on scramjet isolators. The maximum pressure ratio that the shock train was able to create without resulting in a normal shock was equal to 1.72 which is about 92 % of the strength of the normal shock. Same conclusion was drawn from the experimental work of Sullins and McLafferty (1992) on rectangular isolator which states the shock train can reach up to 95% of the normal shock strength, but at a cost of a substantially long duct [64]. For the same back pressure the value of entry Mach number of the combustion chamber from one dimensional analysis differ by 8.86 % from CFD results. The isolator exit flow profile obtained from CFD simulation are follows



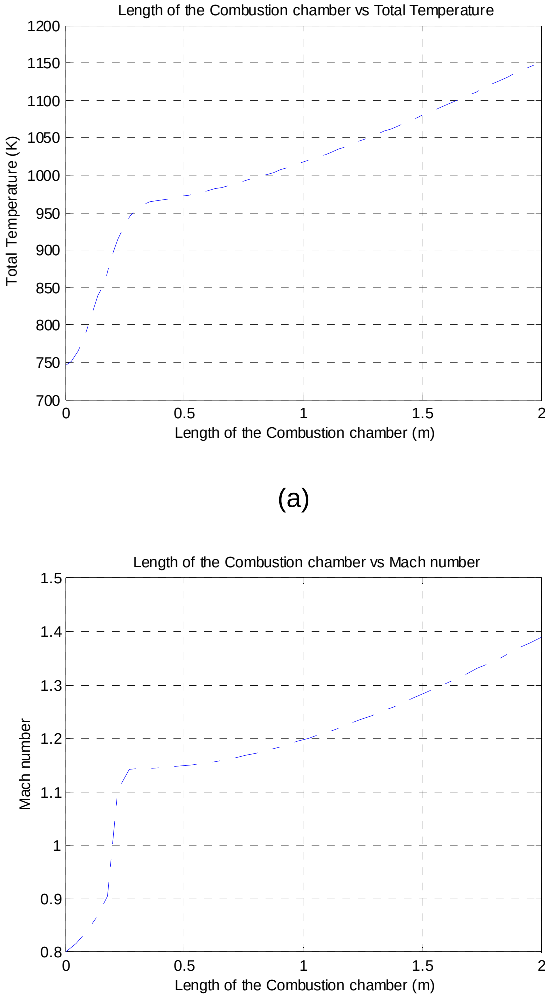
(a)

(b)

(c)

Figure 5-20 Flow Profile from Isolator with NST at $M_o = 3.5$ (a) Total Pressure, (b) Total Temperature, (c) Mach Number

The profile from the isolator exit is passed on to the combustion chamber element for one dimensional analysis. Only the Mach number distribution and the total temperature values are presented in graphical form since the other flow variables are dependent on them.



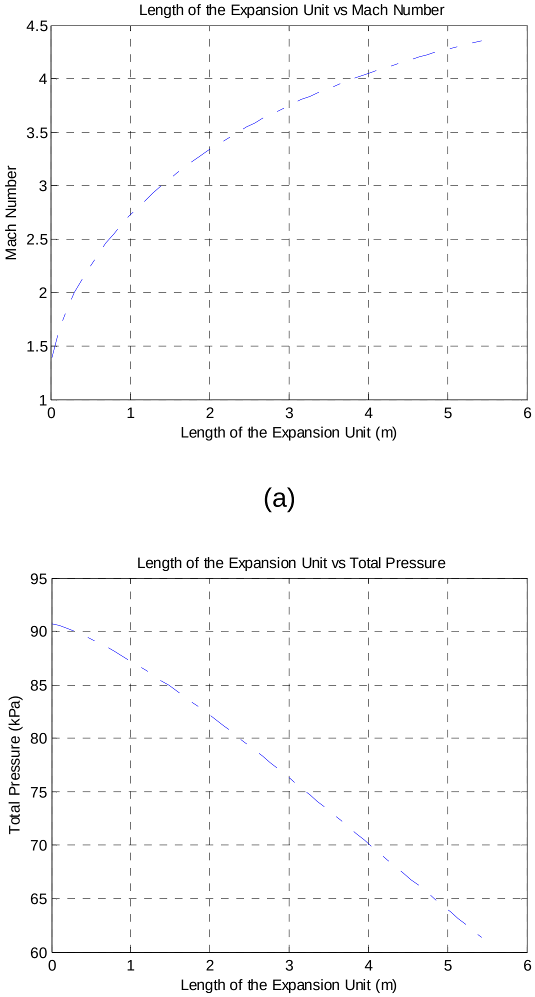
(a)

(b)

Figure 5-21 Axial variation of Flow Variable in Combustion Chamber at $M_o = 3.5$

(a) Total Temperature, (b) Mach Number

The slope of the Mach number is steeper since it was determined by the secant solver within the neighborhood when Mach number approaches unity to avoid singularity. It can be seen that the gradient of total temperature is higher near the injection region and gradually relaxes as the distance increases. The flow at the exit of the combustion chamber is expanded through the expansion ramp of the scramjet engine.



(a)

(b)

Figure 5-22 Axial variation of Flow Variable in Expansion Unit at $M_o = 3.5$ (a) Mach number, (b) Total pressure

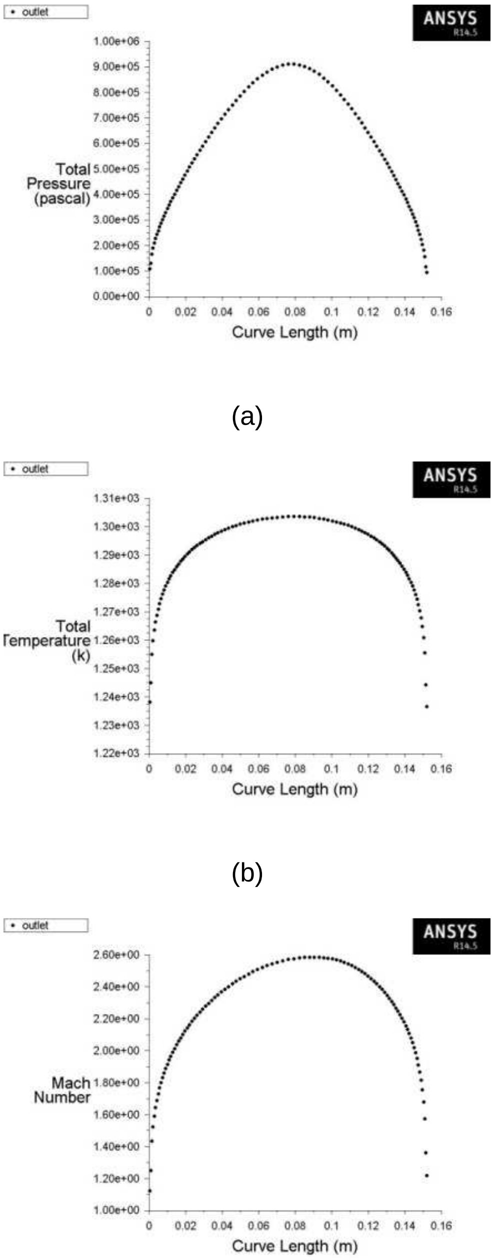
The flow properties at each station of the DMScramjet engine are presented below,

Table 5.10 Station Flow Properties for $M_0 = 3.5$

Station Number i	M_i	$p_i\ (kPa)$	$T_i\ (^oK)$	$P_{t_i}\ (kPa)$	$T_{t_i}\ (^oK)$	$\dot{m}\ (kg/s)$
0	3.5	5.518	217.00	428.241	744.00	42.85
2	1.35	100.201	545.10	301.377	744.96	35.93
3	1.22	110.633	582.43	275.759	744.91	35.93
3_{AC}	0.79	172.229	661.16	266.811	745.64	35.93
4	1.38	29.491	868.706	90.744	1152.19	36.41
10	4.35	0.234	253.66	61.374	1152.19	36.41

5.6 Case Study – Mach 5.0

The Mach 5.0 case was selected for detailed analysis since it marks the transition Mach number after which the DMScramjet engine operates as a pure scramjet engine. The isolator is designed for this Mach number since the pressure ratio produced by the isolator is directly proportional to the incoming dynamic pressure from the experimental correlations. The results from the experimental work Dutton and Carroll also confirms that as Mach number increases, the length of the interacting boundary-layer and shock increases correspondingly [61]. Hence the highest Mach number at which the DMScramjet operates in ramjet mode is chosen. The architecture of the off-design code for Mach 5 is similar to Mach 3.5. Hence only the results are briefed in this section. The flow profile obtained from CFD simulation of the compression component is presented below



(a)

(b)

(c)

Figure 5-23 Flow Profile from Compression System at $M_o = 5.0$ (a) Total Pressure, (b) Total Temperature, (c) Mach Number

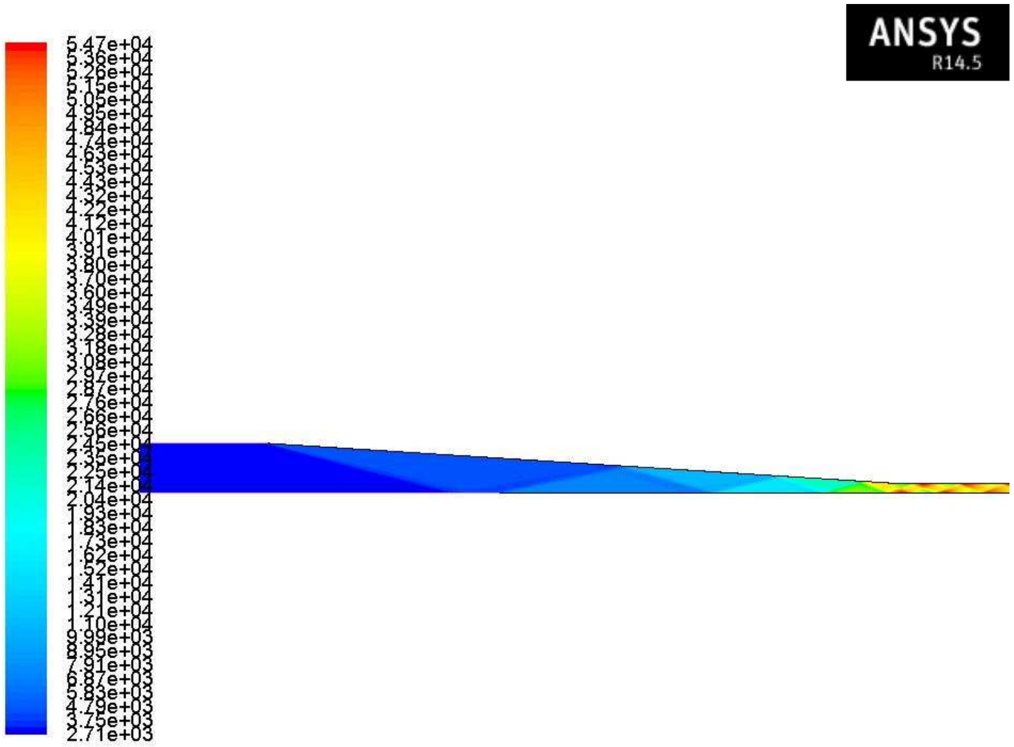
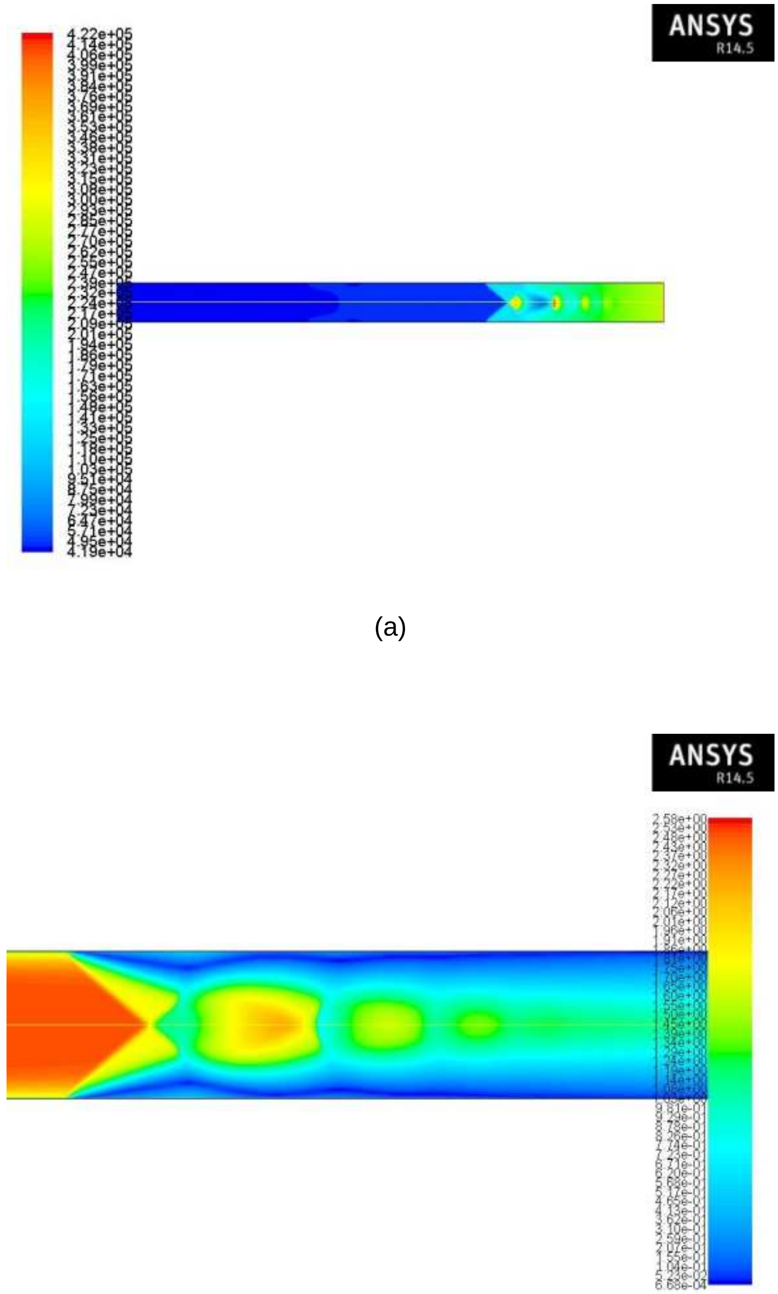


Figure 5-24 Filled Static Pressure Contour of the Compression System at $M_o = 5.0$

The combustion chamber component predicts thermal choking at 0.39 meters. The flow Mach number obtained through back integration was equal to 0.69. Since the static pressure is interpreted from the exit Mach number of the isolator, the length predicted by the two correlations are applied only for the shock train in the isolator and does not take the boundary layer separation zone in the combustion chamber into account. The length of the isolator is predicted by Billig's correlation by using the pressure ratio of a normal shock, $\left(P_3/P_2\right)_{\max}$ inside the isolator instead of a shock train and was equal to 2.02 meters. Ortwerth's correlation predicted the separation length of the boundary layer for pressure ratio $\left(P_3/P_2\right)$ of 6.41 to be equal to 0.80 meters. The CFD results from the theoretical approximation are as follows,



(a)

(b)

Figure 5-25 Filled Contour at $M_o = 5.0$ Showing NST inside Isolator with Symmetry (a) Static Pressure, (b) Mach Number

The exit Mach number obtained for the given back pressure ratio was equal to 0.73 and the length of the isolator was equal to 0.67 meters. The non-symmetrical CFD simulation results using the profile from the exit of the planar inlet for the same back pressure ratio are provided below

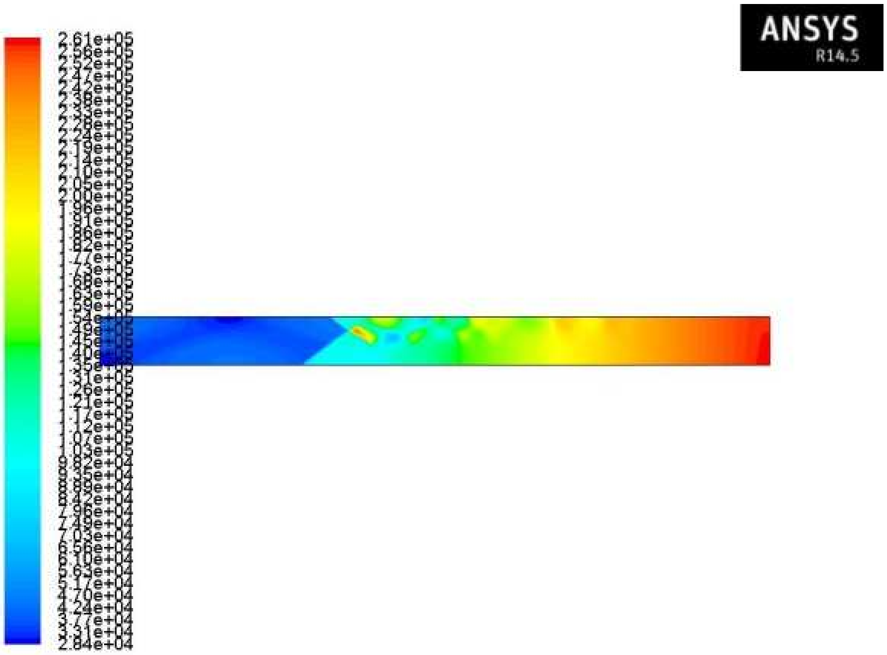
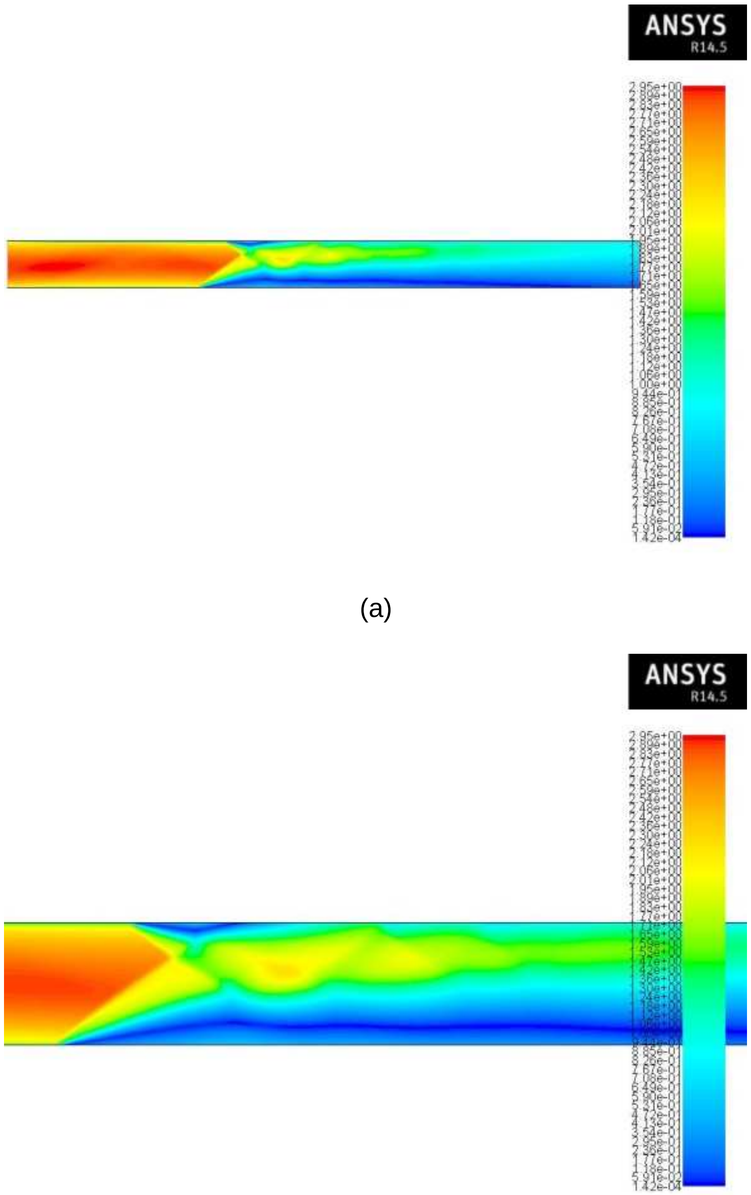


Figure 5-26 Filled Static Pressure Contour at $M_o = 5.0$ Showing NST inside Isolator without Symmetry

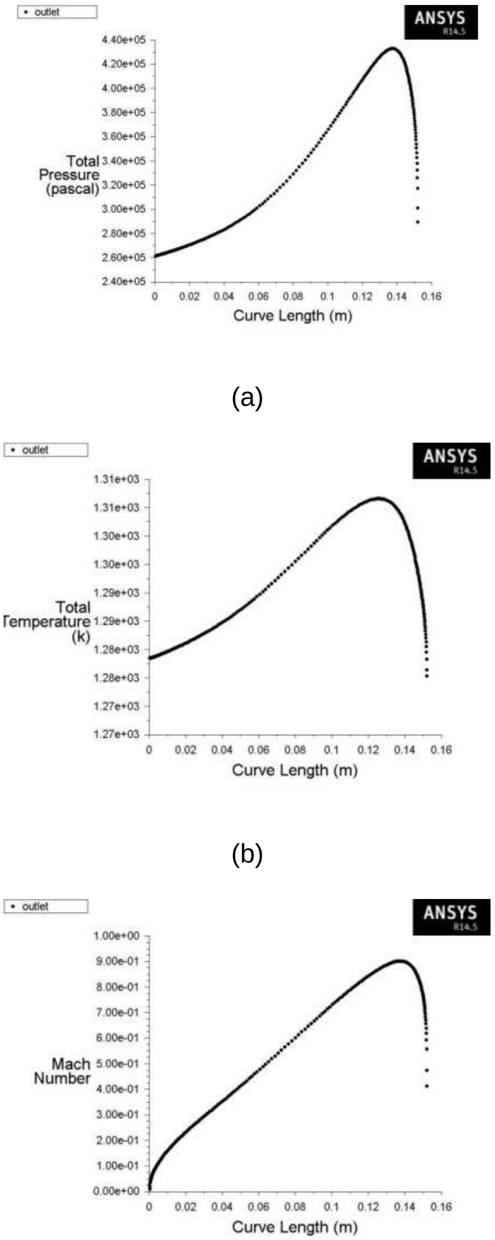


(a)

(b)

Figure 5-27 Filled Contour at $M_o = 5.0$ Showing NST inside Isolator without Symmetry

(a) Mach Number, (b) Zoomed Mach Number Contour



(a)

(b)

(c)

Figure 5-28 Flow Profile from Isolator with NST at $M_o = 5.0$ (a) Total Pressure, (b) Total Temperature, (c) Mach Number

The Mach number value from the CFD analysis has only 2.8 % variation from the one dimensional analysis result. The length of the shock train was equal to 1.43 meters. The length of the shock train covers up to 70.79 % of the isolator length. Though the obtained length is 29.21% lesser than the length predicted by Billig's correlation, the length of the isolator is not curtailed in order to give some margin as it was observed that the stability of the shock train decreased as the Mach number increased [61]. It also coincides with the experimental results of Billig's which states that with modest heat release rates, the pressure rise at the first shock wave of a shock train is sufficient to separate the boundary layer [63]. From the pressure contour we can see that the nonsymmetrical flow path is more prominent in the Mach 5 case because of the multiple oblique shock reflection which is not cancelled at the shoulder of the inlet throat. We can also see the expansion wave and shockwave interaction which leads to highly distorted flow and uneven development of boundary layer.

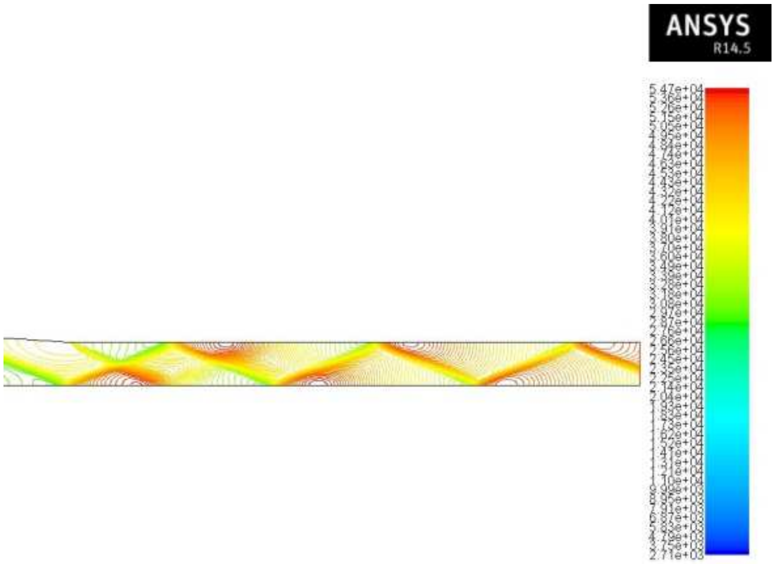
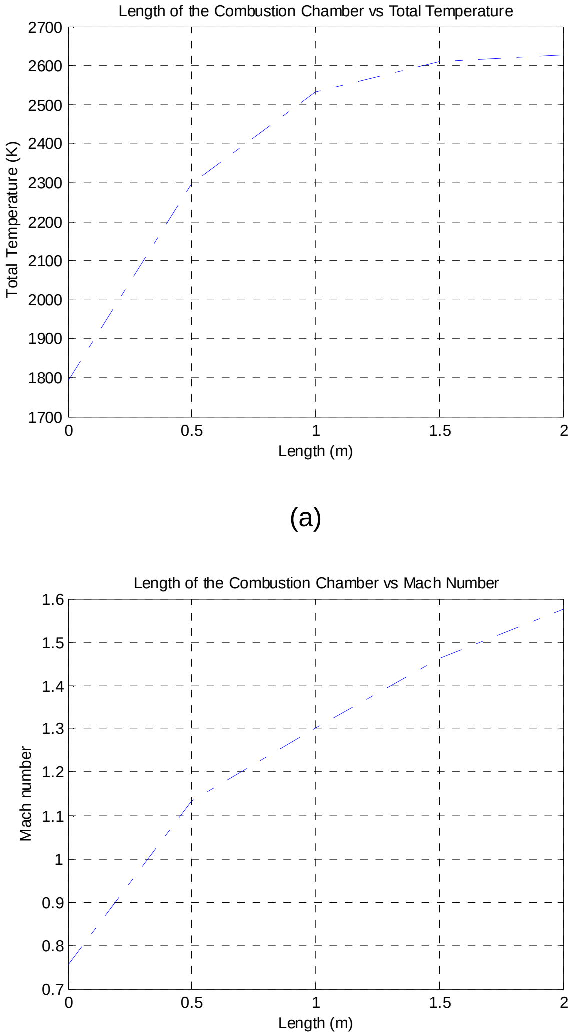


Figure 5-29 Static Pressure Contour showing Multiple Shock Reflection inside Isolator

$$M_o = 5.0$$

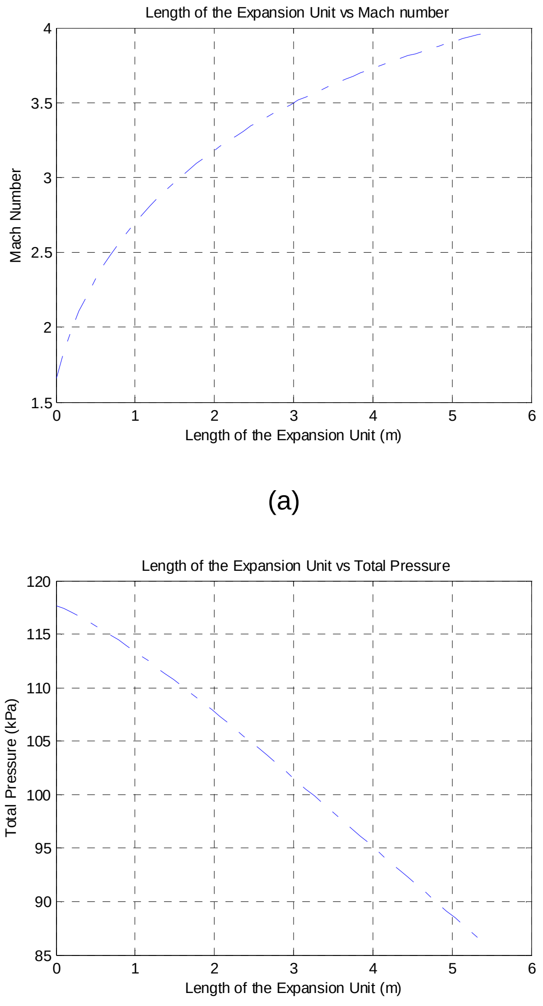
Since the total temperature and the static pressure profile are fed in from the NPSS isolator subelement to ANSYS FLUENT, only the change in combustion chamber entry Mach number for particular back pressure is taken in again for one dimensional analysis. The one dimensional analysis is performed using the averaged values from the isolator exit flow profiles.



(a)

Figure 5-30 Axial variation of Flow Variable in Combustion Chamber at $M_o = 5.0$ (a) Total Temperature, (b) Mach Number

The expansion unit NPSS code approximates no discontinuity through the flow path. This is a valid approximation since the uncancelled shock from the inlet does not penetrate through the shock train which we can see from the shock train pressure contour presented earlier. The results from expansion unit one dimensional analysis are presented below, following which the station property table is provided.



(a)

(b)

Figure 5-31 Axial variation of Flow Variable in Expansion Unit at $M_o = 5.0$ (a) Mach Number, (b) Total Pressure

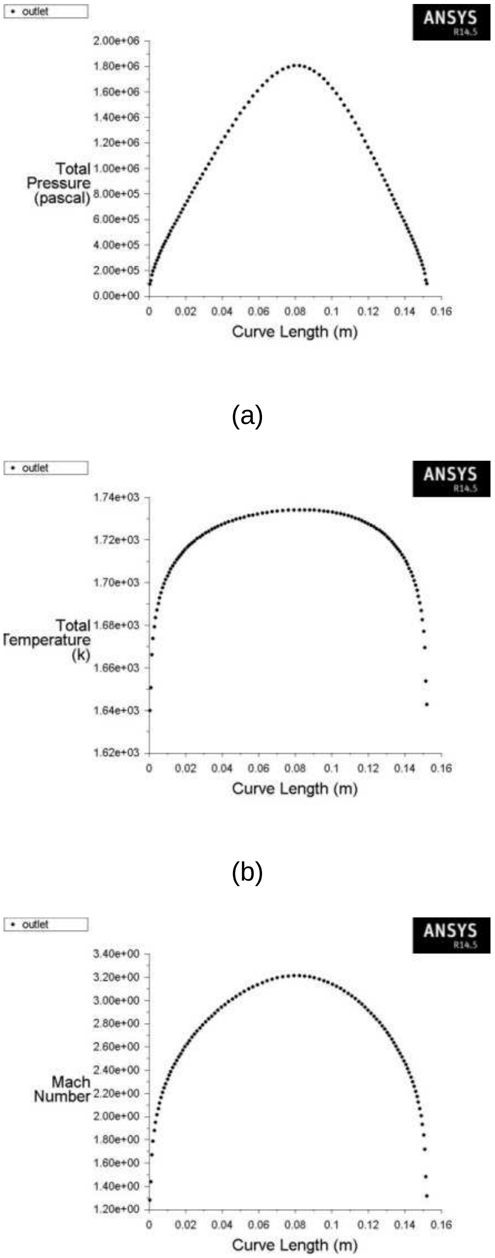
The flow properties at each station of the engine are presented below

Table 5-11 Station Flow Properties for $M_0 = 5.0$

Station Number i	M_i	$p_i\ (kPa)$	$T_i\ (^oK)$	$P_{t_i}\ (kPa)$	$T_{t_i}\ (^oK)$	$\dot{m}\ (kg/s)$
0	5.0	2.310	222.00	1349.806	1251.18	29.59
2	2.58	41.887	596.52	860.889	719.84	27.39
3	2.37	47.112	654.57	685.743	719.99	27.39
3_{AC}	0.67	259.025	1192.91	361.185	1297.16	27.39
4	1.65	26.758	1966.39	119.965	2627.23	29.23
10	3.96	0.434	792.65	87.608	2627.23	29.23

5.7 Case Study – Mach 6.0

The Mach 6.0 case was already analyzed using the on-design NPSS code. But the on-design NPSS code does not simulate the compression unit using CFD and performs numerical calculations to obtain the flow out of each station. It also does not predict any boundary layer separation as well as shock trains if present, since those events are not expected during the on conditional operation of the DMScramjet. The off-design code is used to analyze and simulate the flight Mach number 6.0 condition to check for any flow behavior which is not expected. The architecture of the off-design code remains the same as in all the cases before. Since the pressure contour of the compression unit was presented earlier during the design phase, only the profile at the exit of the isolator is presented below



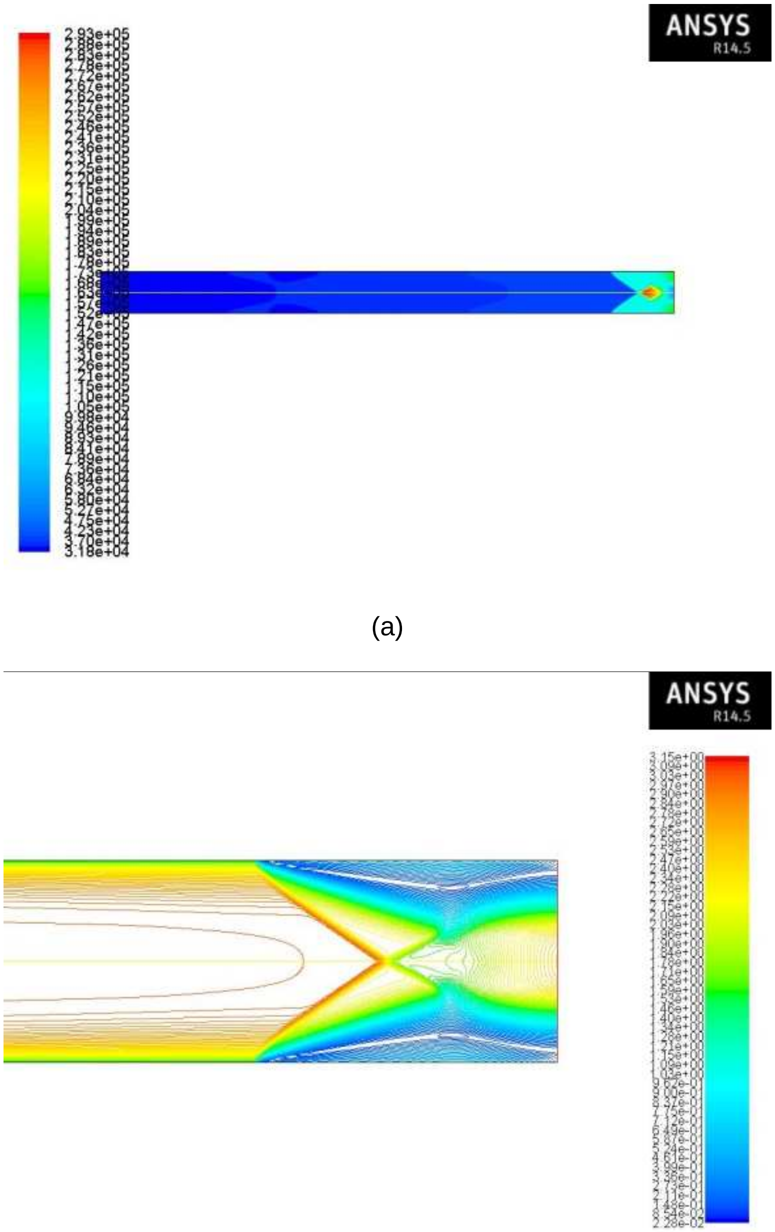
(a)

(b)

(c)

Figure 5-32 Flow Profile from Compression System at $M_o = 6.0$ (a) Total Pressure, (b) Total Temperature, (c) Mach Number

These flow profiles were averaged and passed on to the NPSS combustion chamber. The combustion chamber code predicts a boundary layer separation according to the Korkegi's boundary layer separation criteria at 0.18 meters from the entry of the combustion chamber. This separation is due to pressure gradient beyond the maximum incipient pressure rise and not due to thermal choking. The core flow never turns subsonic in this case. The seperation should give rise to an oblique shock train in the isolator which provides the mechanism for a supersonic flow to adjust to a back pressure higher that its inlet static pressure [2]. The presence of the subsonic portion of the boundary layer allows information to be transmitted upstream at least for a short distance [2]. The length of the isolator calculation is not required any more since it has been fixed for Mach 5.0. So only the Ortwerth's boundary layer separation calculation is performed to determine the length of the oblique shock train which was 0.42 meters. The supersonic flow is compressed adiabatically through the oblique shock train produced by the separated boundary layer until the peak pressure attained in the combustion chamber is achieved [2]. So the NPSS combustion chamber code takes the value of peak pressure p_s and applies it as the exit boundary condition for the CFD model of the isolator. The theoretical CFD model was simulated initially and the results are provided below,



(a)

(b)

Figure 5-33 Filled Contour at $M_o = 6.0$ Showing OST inside Isolator with Symmetry (a) Static Pressure, (b) Mach Number.

The length of the separated region predicted was equal to 0.23 meters. The Mach number contour clearly shows the layer of separated boundary region which forms a confined core area. The reduction in the core area acts as a nozzle throat and reaccelerates the flow by expansion waves. This process repeats again till the desired pressure ratio is achieved by the shock train. The CFD simulation of the oblique shock train with flow profile from the inlet exit is performed next and the results are as follows,

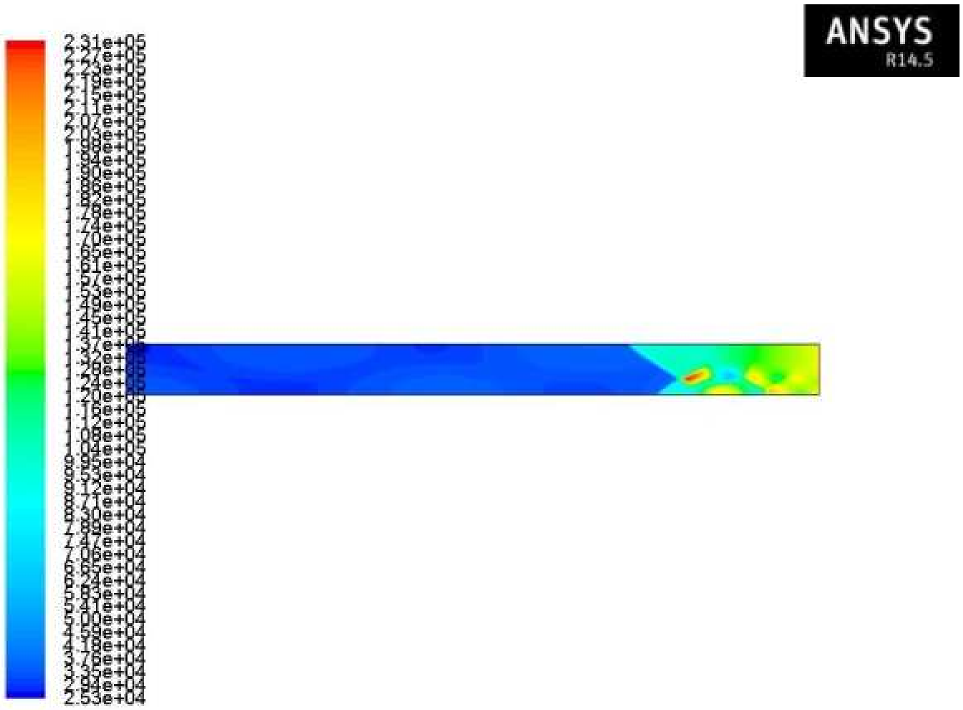


Figure 5-34 Filled Contour at $M_o = 6.0$ Showing OST inside Isolator without Symmetry

Static Pressure

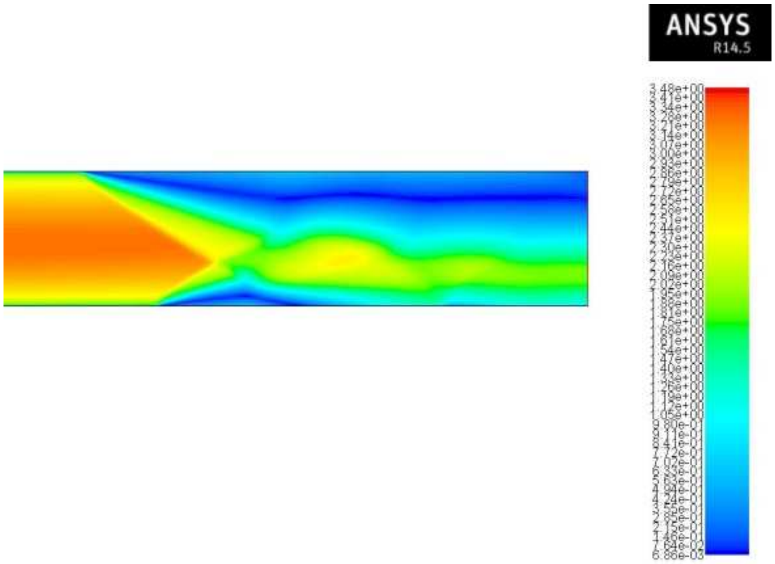


Figure 5-35 Filled Contour at $M_o = 6.0$ Showing OST inside Isolator without Symmetry

Mach Number

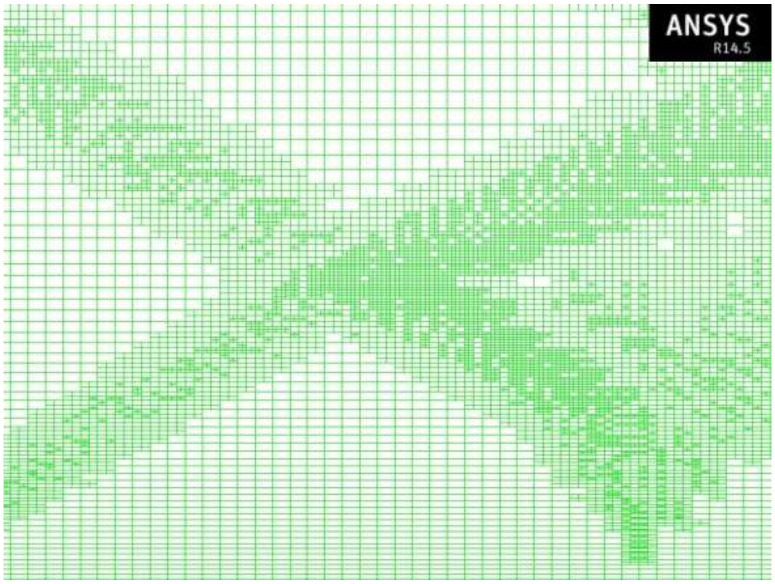
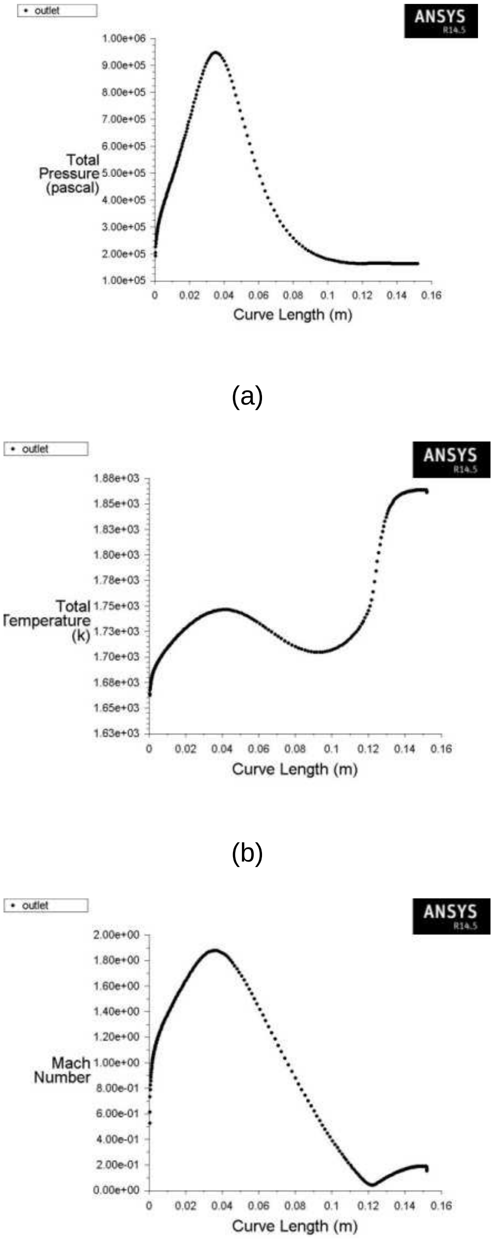


Figure 5-36 Solution Based Dynamic Mesh Adaptation Performed on the Isolator Grid at

$M_o = 6.0$



(a)

(b)

(c)

Figure 5-37 Flow Profile from Isolator with OST at $M_o = 6.0$ (a) Total Pressure, (b) Total Temperature, (c) Mach Number

The oblique shock train which was obtained was highly asymmetric. Two modes off unsteadiness are observed for oblique shocks from experimental results; namely high frequency stream wise oscillation, and a low frequency transverse oscillation [61]. The low frequency oscillation is believed to be related to the neutrally stable shock pattern which has an asymmetric profile [61]. The profile obtained through this CFD simulation can also be related to low frequency oscillation which causes the asymmetry of boundary layer formation along the walls. From the Mach number contour we can see that the core of the flow remains supersonic throughout the length of the shock train. The length of the shock train was predicted to be equal to 0.56 meters. The profile from the isolator exit is passed on to the combustion chamber element and the results from the one dimensional analysis is presented below,

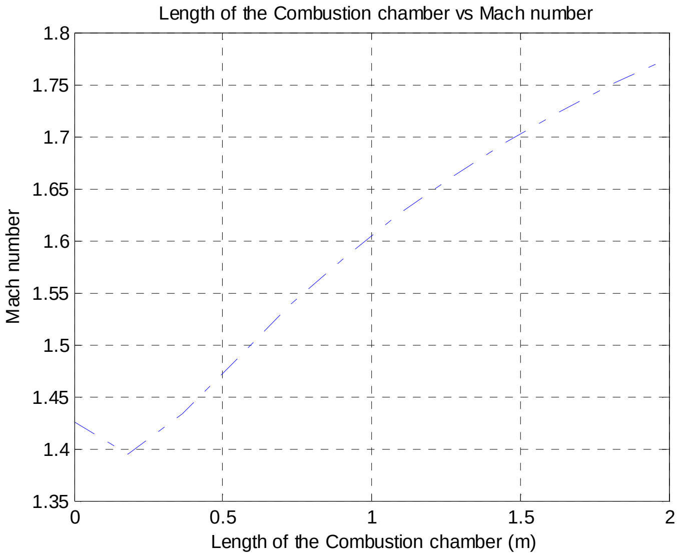


Figure 5-38 Axial variation of Total Temperature in Combustion Chamber at $M_o = 6.0$

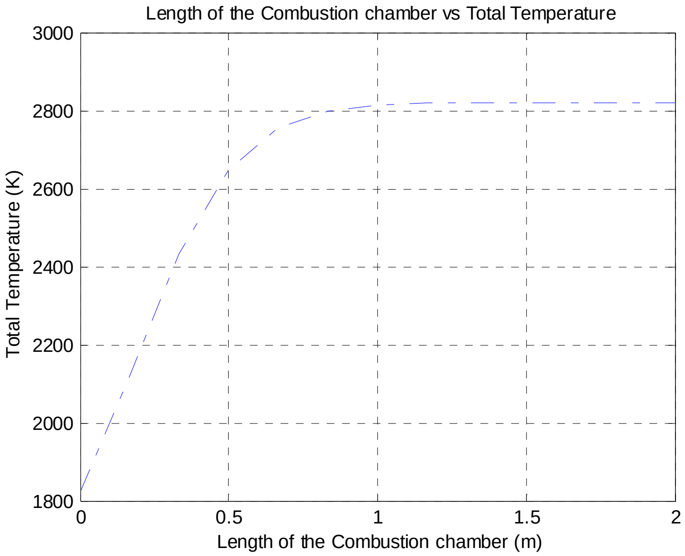


Figure 5-39 Axial variation of Mach Number in Combustion Chamber at $M_o = 6.0$

The resulting flow variable from the combustion chamber element is passed on to the expansion unit element and the results after one dimensional analysis are as follows,

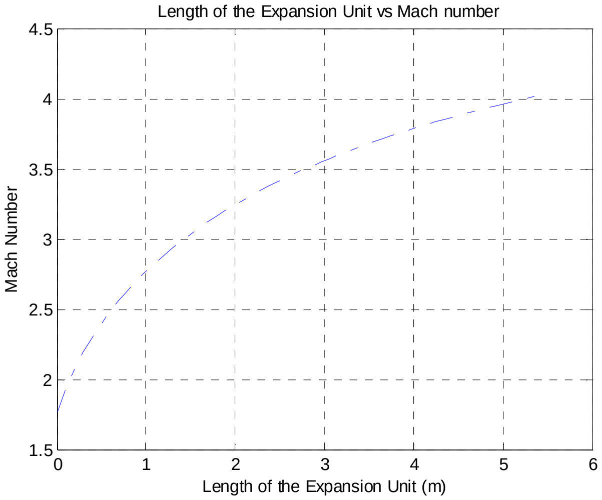


Figure 5-40 Axial variation of Mach Number in Expansion Unit at $M_o = 6.0$

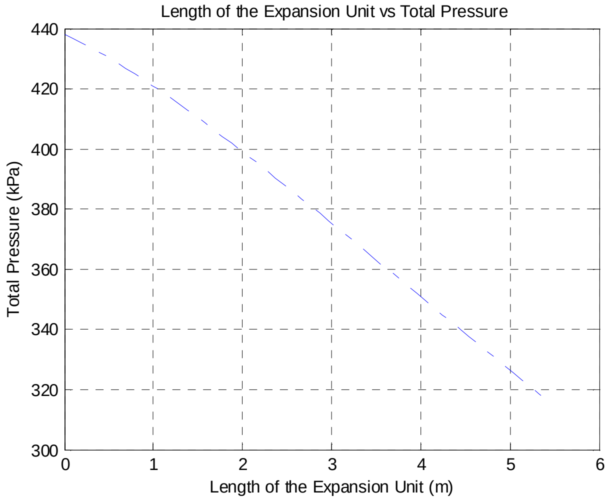


Figure 5-41 Axial variation of Total Pressure in Expansion Unit at $M_o = 6.0$

The flow properties at each station of the engine are presented below,

Table 5-12 Station Flow Properties for $M_0 = 6.0$

Station Number i	M_i	$p_i\ (kPa)$	$T_i\ (^oK)$	$P_{t_i}\ (kPa)$	$T_{t_i}\ (^oK)$	$\dot{m}\ (kg/s)$
0	6.0	1.867	223.57	3576.509	1662.31	24.38
2	3.15	31.788	645.98	1659.950	1724.29	23.55
3	2.90	36.495	716.92	1262.300	1724.87	23.55
3_{BS}	1.42	152.794	1285.89	595.465	1730.97	23.55
4	1.77	81.813	2046.85	438.063	2820.13	25.13
10	4.02	1.411	847.98	316.209	2820.13	25.13

The off-design NPSS code analysis reveals some important insights of on-conditional operation environment of the DMScramjet engine. The engine was designed for a constant static pressure combustion. But it was an theoretical approximation and was not achievable according to the on-design code analysis once the gas dissociation was taken into account. From the above analysis we can see that the separated boundary layer along the oblique shock train creates a new effective wall shape [2]. As the separated region approaches the entrance of the combustion chamber, the core area expands leading to decrease in static pressure to a certain extent. This static pressure is further reduced after the burnt gas expands through the variable area section of the combustion chamber compensating the static pressure gained due to combustion. Thus the constant static pressure is not maintained throughout the axial direction of combustion as approximated for the stream thrust analysis. Similar trend of static pressure though out the scramjet engine is also observed in the work of Torrez [55]. This analysis also revealed the existence of the oblique shock train and the amount of additional flow distortion at the exit plane of the compression system. Since the burner element in this research has hypothetical mixing efficiency, it does not affect the performance to a great extent. But the distortion should be considered during the design of the combustion chamber without any approximations.

The on-design code revealed that the expansion unit underperformed at $M_0 = 6.0$. But after taking intercomponent interactions into account in the above analysis, we can see that the expansion unit reaches the performance for which it was designed for. Incase the intercomponent interactions were ignored by a designer or an engineer, he or she would have taken efforts in designing another expansion unit for better SERN performance. But only after a complete engine simulation the drawbacks would have been determined leading to a loss of time as wells as money. This analysis also stresses the importance of the intercomponent interactions when designing individual components by presenting and comparing the performance of the expansion unit.

Since the same methodology is followed for the performance analysis of the remaining off-design cases, the results from that analysis are presented in Appendix A section of this paper. The formation of oblique shock train in the isolator for a flight Mach number of 5.5 when the respective back pressure is applied at the exit of the isolator component after boundary layer separation at a fuel equivalence ratio of 1.0 proves that the transition of ram to scram mode has successfully occurred. Those results are also provided in Appendix A section of this paper.

5.8 Performance and Evaluation

As a completeness for the performance assessment through this thermodynamic cycle analysis, general performance evaluation terms of an air breathing engine are used and the performance of DMScramjet engine designed for this research at various altitudes and flight Mach number is presented below

Table 5-13 DMScramjet Engine Performance Parameters

M_o	$F/\dot{m}_o\left(\text{N-s}/kg\text{A}\right)$	f	$S\left(kgF/\text{s-N}\right)$	$\text{I}_{sp}\left(s\right)$	η_{th}	η_p	η_o
3.5	346.64	1.34×10^{-02}	1.94×10^{-04}	524.27	0.68	0.17	0.12
4.0	986.20	6.74×10^{-02}	6.83×10^{-05}	1491.55	0.50	0.77	0.39
4.5	790.12	6.74×10^{-02}	8.53×10^{-05}	1195.00	0.41	0.86	0.35
5.0	714.44	6.74×10^{-02}	9.43×10^{-05}	1080.54	0.39	0.95	0.36
5.5	657.10	6.74×10^{-02}	1.03×10^{-04}	993.82	0.38	0.91	0.36
6.0	598.80	6.74×10^{-02}	1.13×10^{-04}	905.63	0.36	1.00	0.36

The overall performance of the engine at Mach 3.5 is low because the engine is capable of operating only within the fuel equivalence ratio of 0.2 above which the back pressure becomes strong enough to unstart the engine. The flight Mach number of 3.5 is avoided for the plots since the fuel equivalence ratio differs from the other cases.

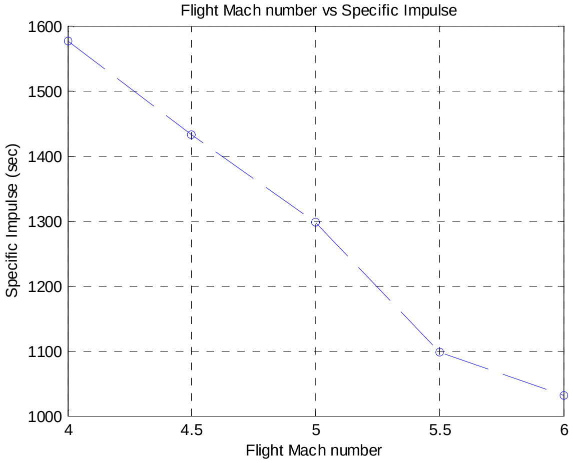


Figure 5-42 Specific Impulse Plot for Various Flight Mach numbers

The specific impulse which was obtained from this research matches well with the theoretical performance of a hydrocarbon scramjet engine in the work by Moses [65].

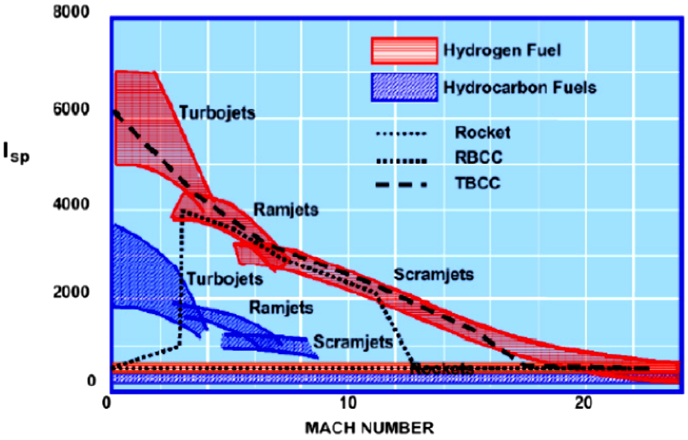


Figure 5-43 Specific Impulse Plot for Various Propulsion Systems by Moses [65].

Then the performance parameter for the hypersonic compression system including inlet and the isolator are calculated and tabulated below. The total pressure recovery varies according to the intensity of the reflected oblique shockwave or expansion wave in the isolator section. The table is split into two subdivisions, one part represents the performance of the compression system without thermal choking and boundary layer separation and the other part takes shock train into account.

Table 5-14 DMScramjet Compression Component Performance Parameters

M_0	$\pi_{c_{BS}}$	$\psi_{c_{BS}}$	$\eta_{c_{BS}}$	$\eta_{KE_{BS}}$	$\left(\left(S_3-S_0\right)/C_{pc}\right)_{BS}$	$\pi_{c_{AS}}$	$\psi_{c_{AS}}$	$\eta_{c_{AS}}$	$\eta_{KE_{AS}}$	$\left(\left(S_3-S_0\right)/C_{pc}\right)_{AS}$
3.5	0.64	2.68	0.92	0.94	0.11	0.62	3.04	0.93	0.93	0.12
4.0	0.56	2.68	0.90	0.94	0.15	0.47	7.19	0.96	0.92	0.19
4.5	0.39	2.81	0.84	0.92	0.24	0.26	7.36	0.93	0.88	0.35
5.0	0.5	2.94	0.89	0.95	0.18	0.26	5.36	0.90	0.90	0.35
5.5	0.39	2.95	0.85	0.94	0.24	0.22	4.96	0.87	0.90	0.40
6.0	0.35	3.2	0.85	0.95	0.27	0.16	5.75	0.86	0.90	0.48

The mass flow spillage in both the on-condition and off-condition operation from the CFD simulation are tabulated and shown below

Table 5-15 DMScramjet Inlet Spillage Parameters

M_0	$\dot{m}_{o_{freestream}}$	$\dot{m}_{o_{captured}}$	$Spillage\%$
3.5	42.85	35.93	19.25
4.0	37.10	31.85	16.48
4.5	32.06	29.16	9.95
5.0	29.59	27.39	8.03
5.5	26.78	25.55	4.81
6.0	24.38	23.55	3.51

The comparison between the results from stream thrust analysis and one dimensional analysis was already performed during the on-design calculation of the DMScramjet engine. Now the comparison between results from analytical model which contains both the realistic stream thrust analysis as well as one dimensional flow analysis and CFD model are made for both the inlet and the isolator as shown below.

Table 5-16 Station 2 Flow Properties Comparison

Station 2	Analytical	CFD	Error %
Static Pressure (kPa)	25.055	31.788	21.18
Static Temperature (K)	545.68	645.98	15.52
Total Pressure (kPa)	1278.28	1659.95	22.99
Total Temperature (K)	1662.31	1724.29	3.59
Mach Number	3.19	3.15	1.26

Table 5-17 Station 3 Flow Properties Comparison

Station 3	Analytical	CFD	Error %
Static Pressure (kPa)	34.782	36.495	4.69
Static Temperature (K)	720.92	716.92	0.55
Total Pressure (kPa)	1278.28	1262.30	1.26
Total Temperature (K)	1662.47	1724.87	3.70
Mach Number	2.79	2.90	3.94

As we can see from the above analysis, the static pressure term has maximum error percentage when compared with other parameters from both the tables. The beauty of the scramjet combustion element is that it requires only the area distribution, total temperature distribution and entry Mach number for performance calculation. Both the total temperature as well as the Mach number error is well within 4% when compared to higher fidelity simulation. Hence the analytical method developed can be sufficient enough for the initial design and analysis of components with the on-design NPSS code which is within 4% of error. The graphical representation of performance of the expansion system and the engine is provided in the APPENDIX B section of this paper.

Chapter 6

Conclusions and Recommendations

This chapter presents the conclusions of the research and recommendations for future research.

6.1 Summary and Conclusions

Several goals were planned to be achieved at the end of this research and most of them were accomplished. The first and foremost goal of this research was to develop a hypersonic compression system which operates successfully over a regime of Mach numbers with a fixed geometry suitable for TBCC and cruise at Mach 6. The type of the inlet from various basis and categories were decided from previous research work done for same type of mission. The inlet was then tailored to the current mission of starting at Mach 3.5 and reaching Mach 6.0 with reduced distortion levels at the exit of the isolator at design condition operation. The methodology to design the isolator component and predict the length of the shock train if present was also discussed in detail. The different modes of the isolator and the transition of ramjet mode to scramjet mode of operation were also discussed. The compression system as a whole has good performance with average adiabatic efficiency of 90.8% and kinetic efficiency of 90.5%.

The next goal was to develop a reduced order scramjet inlet that could aid the process of designing before stepping into higher order simulations during the Mach 6 condition. NPSS was used to develop the model because of the vast amount of technical data available in its inbuilt database as well as the object oriented programming language can suit the propulsion system analysis to the right extend needed by dividing the components into elements and sub elements and connecting each. A scramjet inlet element was built as the first step of the complete reduced order model. The inlet element used the realistic stream thrust analysis approach which composed of two control volume analysis. The oblique shock relations were used to aid the first control volume analysis. The element took both the discontinuities and viscous effects in terms of friction into account. The isolator subelement used the one dimensional Fanno flow approach for the on-design calculations.

Another goal of this research was to take intercomponent interactions into account for a realistic simulation. A DMScramjet combustion chamber and expansion SERN element was created by using NPSS for this purpose. The methodology followed for both of the components and the procedure for connecting all the elements together to form a complete scramjet were discussed in detail. Both the elements had few approximations which were considered reasonable [2]. By forming a complete assembly, the performance assessment of the DMScramjet was possible with variations in the mode of operation. The importance of intercomponent interactions was also discussed. Additional ODE solvers were developed for solving the generalized one dimensional flow equations. All the components used the technical data available from NPSS regarding the flow properties to the maximum extend. This made a realistic reduced order simulation possible with considerable reduction is time. A maximum of 5 s was consumed by the complete NPSS DMScramjet assembly for each run.

An important perspective of this research was to apply higher order simulations to the hypersonic compression system to capture the complex physics involved in the flow. ANSYS FLUENT CFD simulations were performed for the compression system for both the on-design and off-design cases. CFD simulation for isolator component was performed separately for the three modes of operation. Thermally perfect gas model was used for all the CFD simulations in this research to account for the vibrational excitation of the molecules. Dynamic adaptive mesh technique was used for both inlet and isolator to reduce the numerical error due to mesh to a given threshold limit. The NPSS was used to script input and output functions using which can be used with any element to communicate with the CFD solver. The flow variables obtained from different fidelity analysis were compared for accuracy and reliability. The performance of the compression system and complete engine for all the scenarios to be faced by the engine were discussed. This concludes the summary of the tasks completed through this research .The next section discusses about the recommendation for possible research which can be carried out as a continuation from the current work.

6.2 Recommendation

A few recommendations are given to improve the scope of the current research in this section. The third dimension of the inlet increases the complexity of the flow and must be considered before prototype processing. The influence of the side ramp along the inlet should be tested since additional shock boundary layer interaction takes place. The lip of the external compression ramp as well as the cowl must be carefully designed with appropriate radius to distribute the heat formed due to friction as well as shock-shock interactions. The empirical correlations to predict the length of the shock train is not accurate enough to rely upon for all the cases since they are applicable for specific set of isolator shape and inlet Mach number for which the experiments were performed. Care should be taken to predict the length of the shock train during design phase of the isolator. The influence of the wall on the shock train along with the boundary layer separation at the corners of the duct is also an important factor to be considered during design. Hence a suitable 3D CFD analysis should be performed before any further stages of design or manufacture. Using a laminar turbulence transition model with less error due to multiple shock boundary layer interaction can also improve the accuracy of the simulation.

The mixing correlation in the combustion chamber requires the total combustion chamber efficiency to distribute the total temperature. Though the efficiency can be approximated to 90% during on-design case, it can vary for the off-design cases due to the additional distortions in the exit plane of the compression system. Hence using total efficiency of combustion chamber from experimental results can provide more insight during numerical analysis. The accuracy of the expansion unit analysis can improve by taking two additional factors namely chemically reacting flow through the nozzle and discontinuities from the combustion chamber. The performance assessment can be improved by taking the heat loss to the external environment into account. There has been a national level interest to advance the technology readiness level of TBCC systems by developing the simulation and controls software to model all modes of operation over its mission, including mode transition from gas turbine to dual-mode scramjet propulsion, a requirement before controlled wind tunnel testing or flight testing can be accomplished through this region of operation [66]. The addition of a turbojet assembly which is prebuilt in NPSS attached with the DMScramjet assembly to simulate the complete TBCC makes it feasible and apt to be used for industrial TBCC and subsystems applications such as for the simulation of engine proposed by Allison Advanced Development Company (AADC) and SPIRITECH Advanced Products Inc. (SPIRITECH) [66].

Appendix A

Off-Design Case Analysis Results

Mach 4

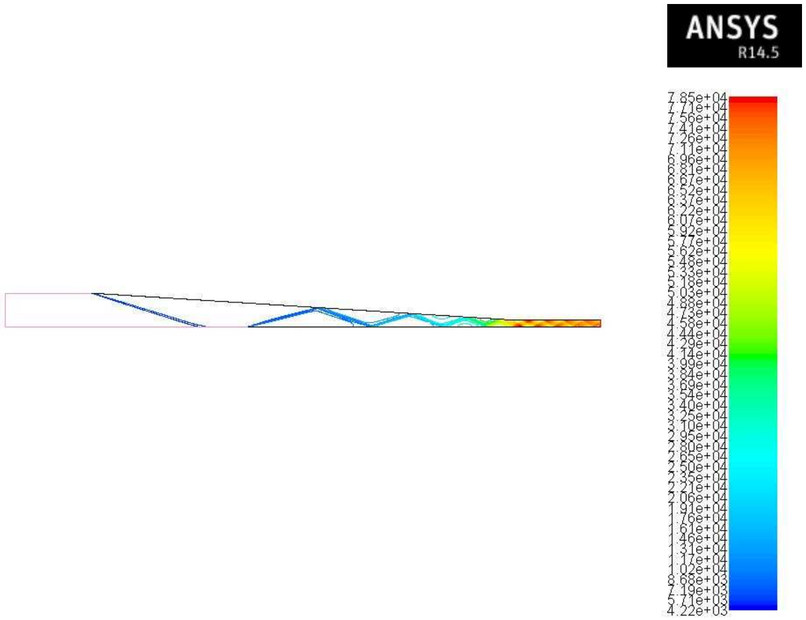
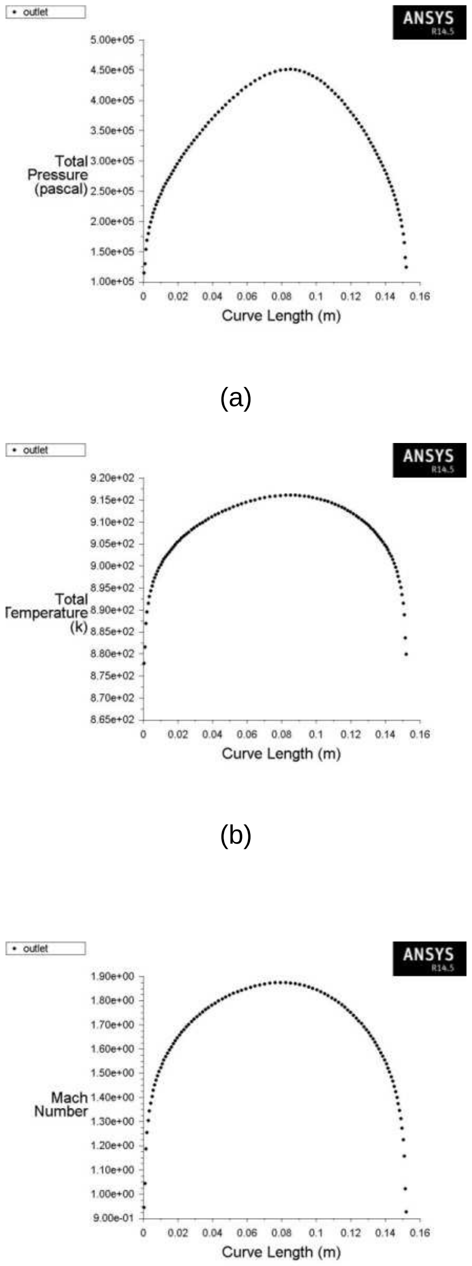


Figure A-1. Static pressure contour of the compression system at $M_o = 4.0$



(a)

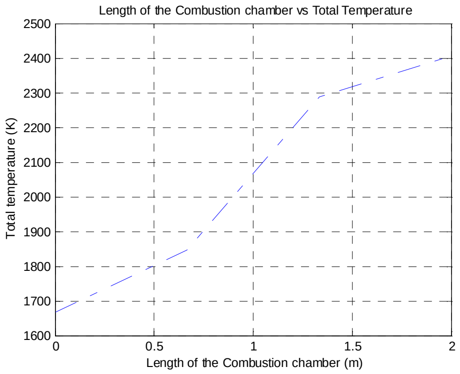
(b)

(c)

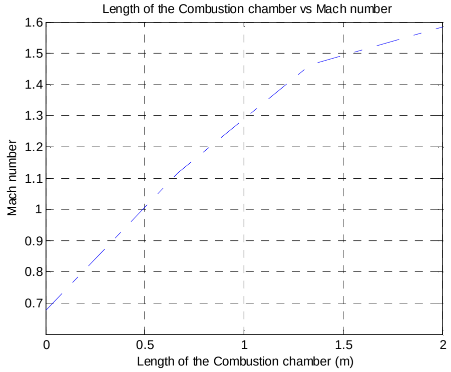
Figure A-2. Flow Profile from Compression System at $M_o = 4.0$ (a) Total Pressure, (b) Total Temperature, (c) Mach Number

$\left(P_3/P_2\right)_{\max}$	3.93
Length of Isolator	1.04 m
$\left(P_3/P_2\right)_{ST}$	3.56
Length of Separation	0.71 m

Figure A-3. Normal shock train parameters from isolator subelement at $M_o = 4.0$



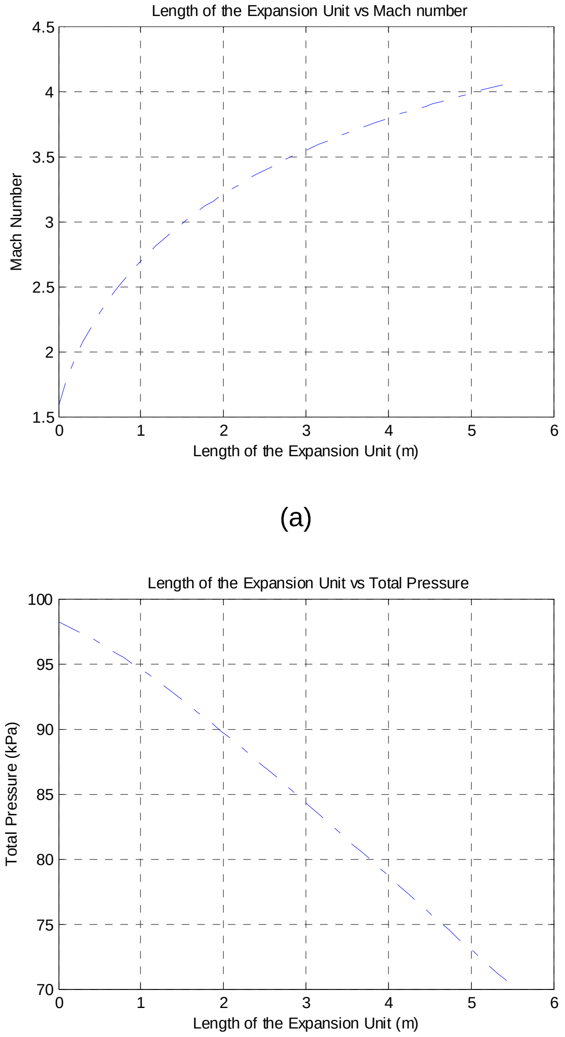
(a)



(b)

Figure A-4. Axial variation of Flow Variable in Combustion Chamber at $M_o = 4.0$ (a)

Total Temperature, (b) Mach Number



(a)

(b)

Figure A-5. Axial variation of Flow Variable in Expansion Unit at $M_o = 4.0$ (a) Mach Number, (b) Total Pressure

Station Number i	M_i	$p_i \ \left(kPa\right)$	$T_i \ \left({}^oK\right)$	$P_{t_i} \ \left(kPa\right)$	$T_{t_i} \ \left({}^oK\right)$	$\dot{m}(kg/s)$
0	4.0	4.204	218.33	661.559	888.96	37.10
2	1.88	65.098	551.26	428.919	910.49	31.85
3	1.74	70.894	586.98	371.921	910.60	31.85
3_{AC}	0.67	234.225	1570.72	312.283	1665.72	31.85
4	1.58	24.563	1804.40	98.182	2404.79	32.64
10	4.05	0.322	675.38	70.612	2404.79	32.64

Figure A-6. Station flow properties for $M_o = 4.0$

Mach 4.5

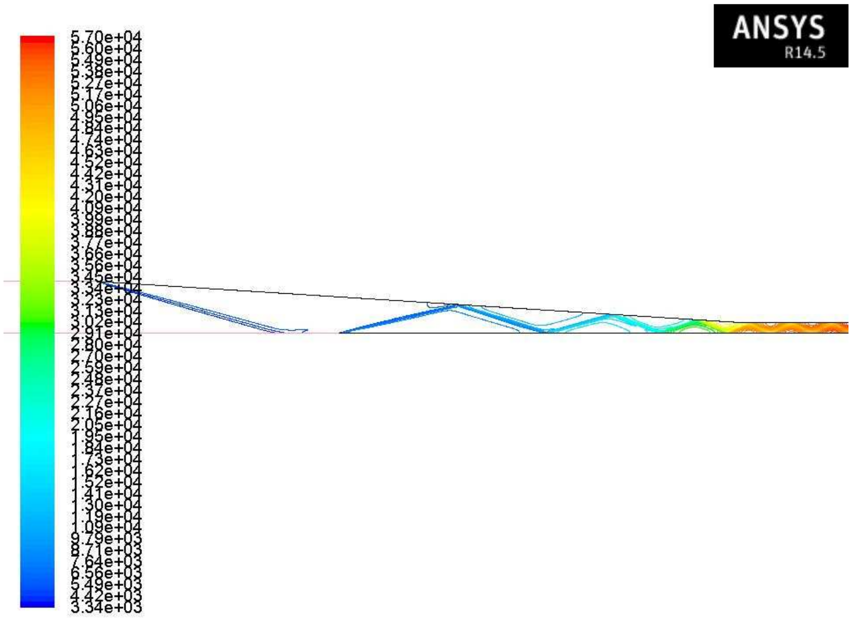
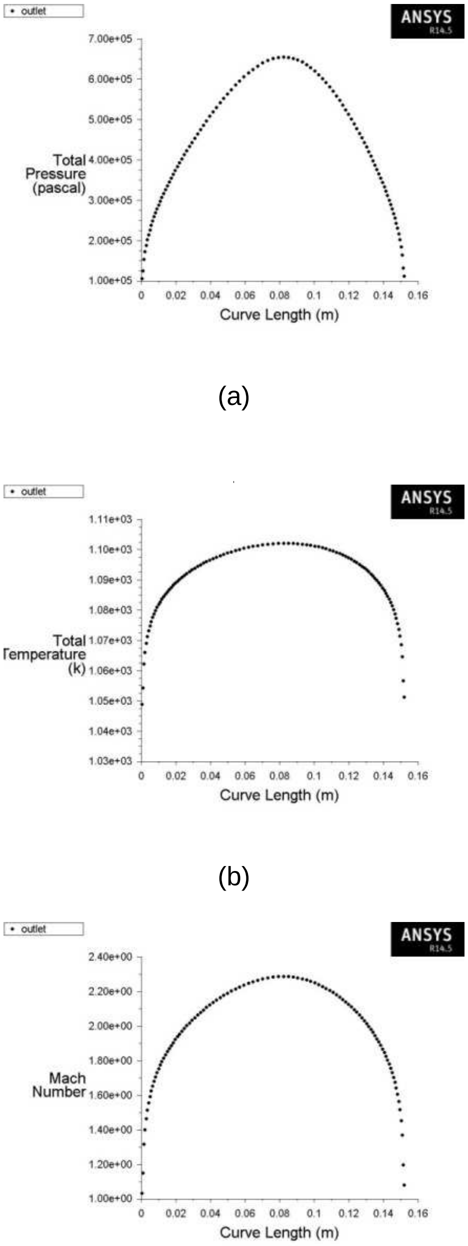


Figure A-7. Static pressure contour of the compression system at $M_o = 4.5$



(a)

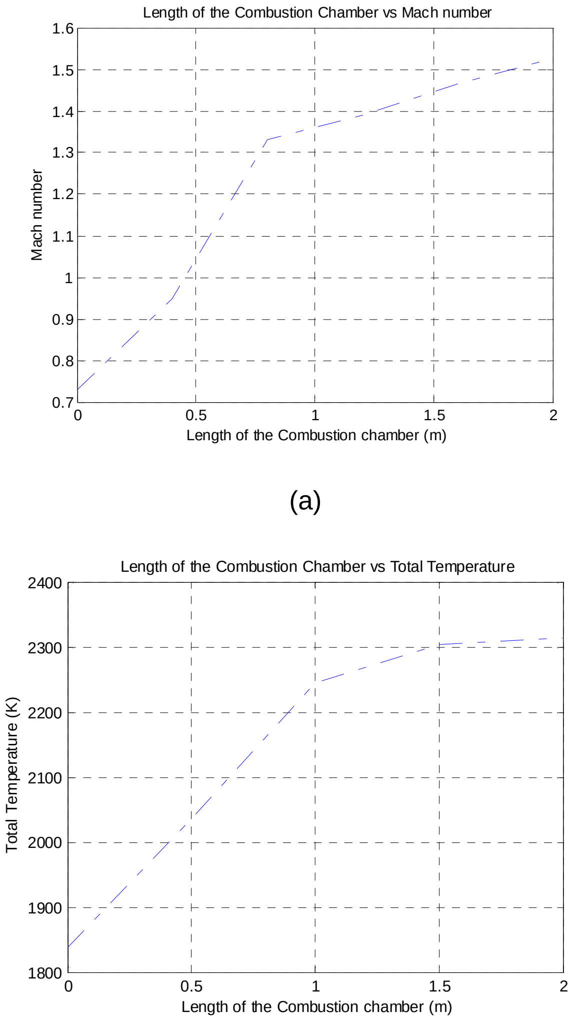
(b)

(c)

Figure A-8. Flow Profile from Compression System at $M_o = 4.5$ (a) Total Pressure, (b) Total Temperature,

Position of Thermal Choking	0.48 m
$\left(P_3/P_2\right)_{\max}$	5.68
Length of Isolator	1.48 m
$\left(P_3/P_2\right)_{ST}$	4.84
Length of Separation	0.74 m

Figure A-9. Normal shock train parameters from isolator subelement at $M_o = 4.5$

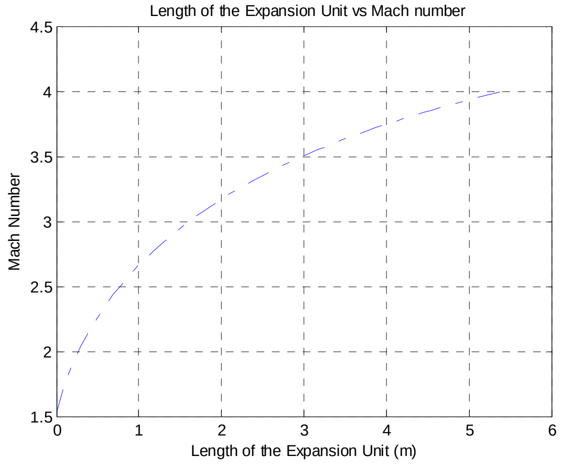


(a)

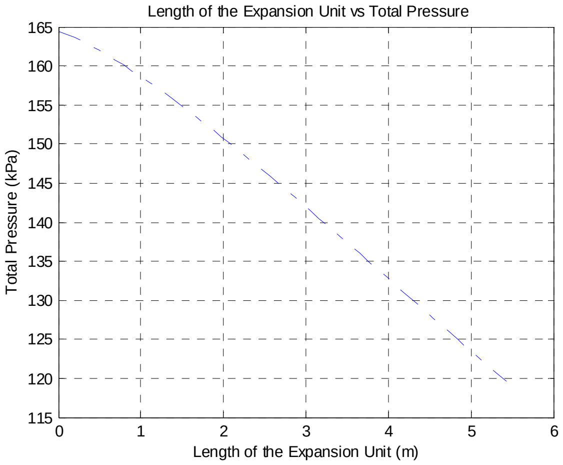
(b)

Figure A-10 Axial variation of Flow Variable in Combustion Chamber at $M_o = 4.5$

(a) Total Temperature, (b) Mach Number



(a)



(b)

Figure A-11. Axial variation of Flow Variable in Expansion Unit at $M_o = 4.5$ (a) Mach Number, (b) Total Pressure

Station Number i	M_i	$p_i \ (kPa)$	$T_i \ \left({}^oK\right)$	$P_{t_i} \ (kPa)$	$T_{t_i} \ \left({}^oK\right)$	$\dot{m} \ (kg/s)$
0	4.5	4.204	218.33	1293.753	1093.21	32.06
2	2.2	50.906	572.19	610.657	1095.13	29.16
3	2.09	55.509	615.03	511.229	1095.32	29.16
3_{AC}	0.73	246.429	1608.54	346.134	1721.41	29.16
4	1.53	44.258	1927.77	164.489	2512.62	31.12
10	4.00	0.579	729.81	119.560	2512.62	31.12

Figure A-12. Station flow properties for $M_o = 4.5$

Mach 5.5

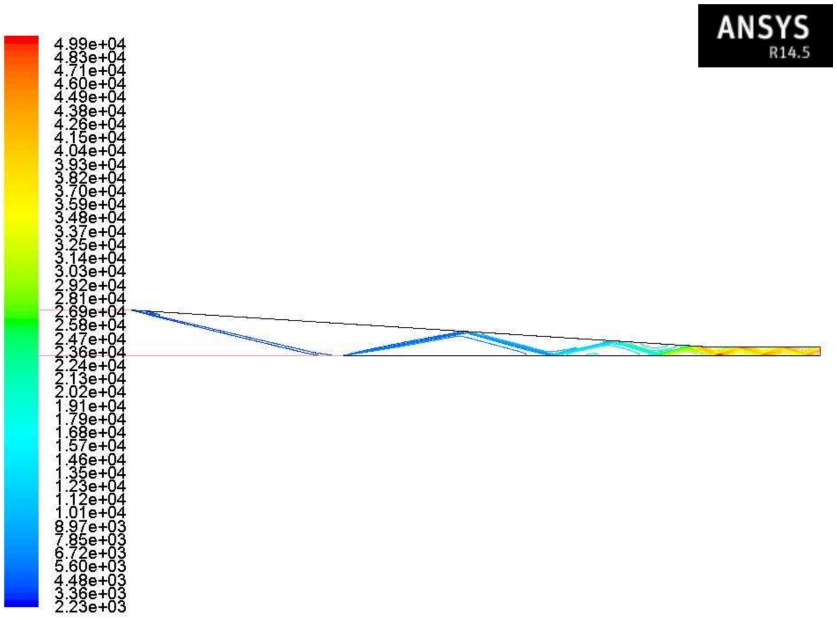
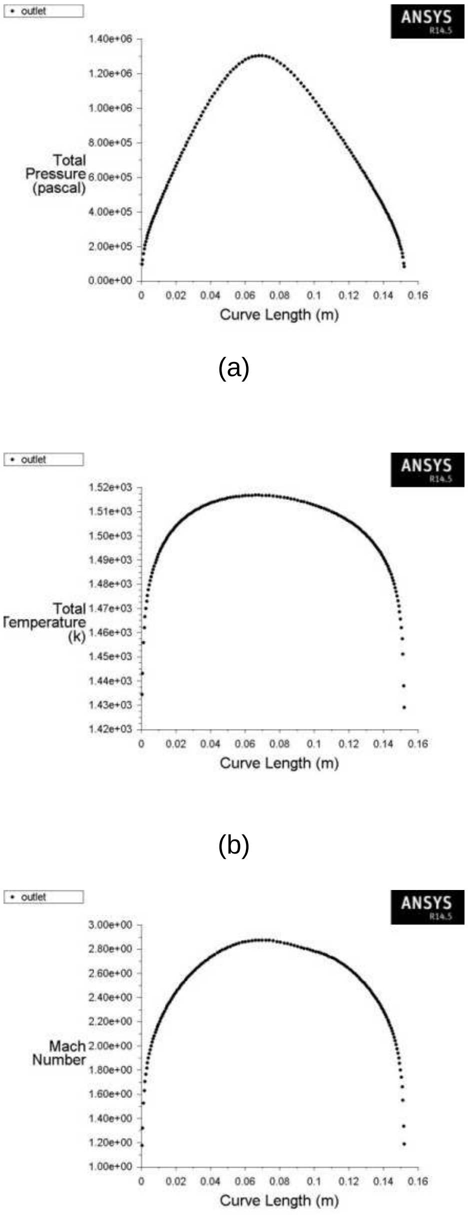


Figure A-13. Static pressure contour of the compression system at $M_o = 5.5$



(a)

(b)

(c)

Figure A-14. Flow Profile from Compression System at $M_o = 4.5$ (a) Total Pressure, (b) Total Temperature, (c) Mach Number

Position of Seperation	0.21
$\left(P_3/P_2\right)_{\max}$	9.24
$\left(P_3/P_2\right)_{ST}$	4.75
Length of Separation from NPSS	0.47 m
Length of Separation from CFD	0.48 m

Figure A-15. Oblique shock train parameters from isolator subelement and CFD at

$M_o = 5.5$

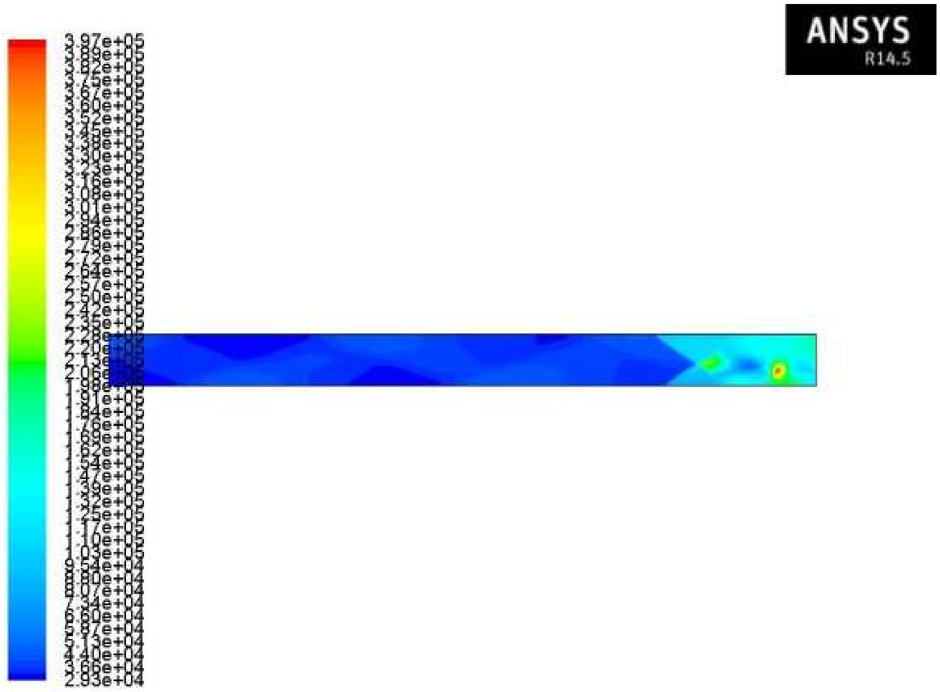


Figure A-16 Filled Contour of Static Pressure at $M_o = 5.5$ Showing OST inside Isolator without Symmetry

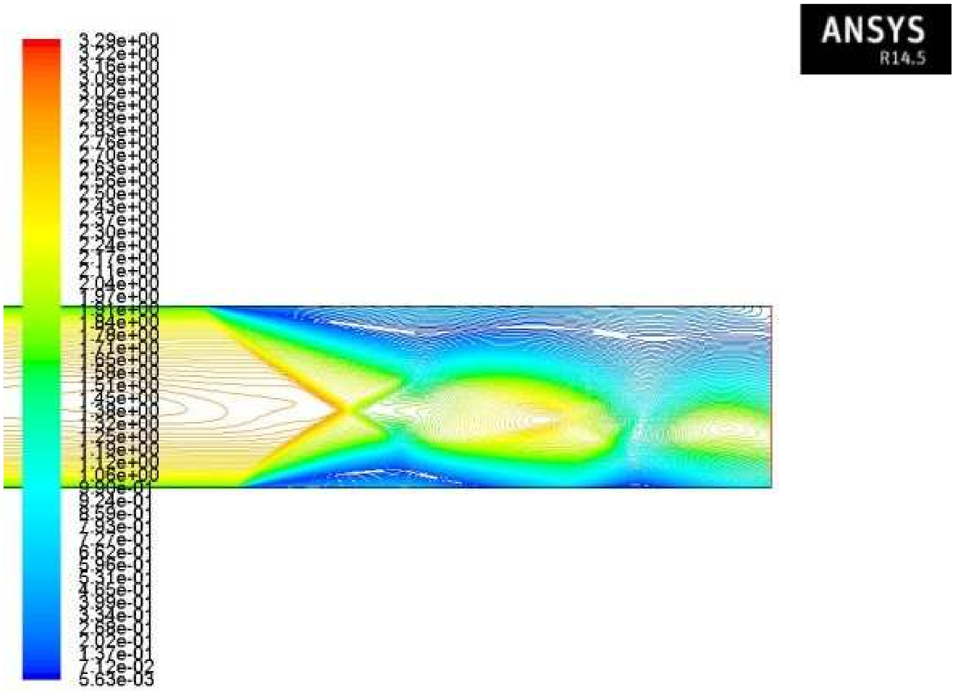
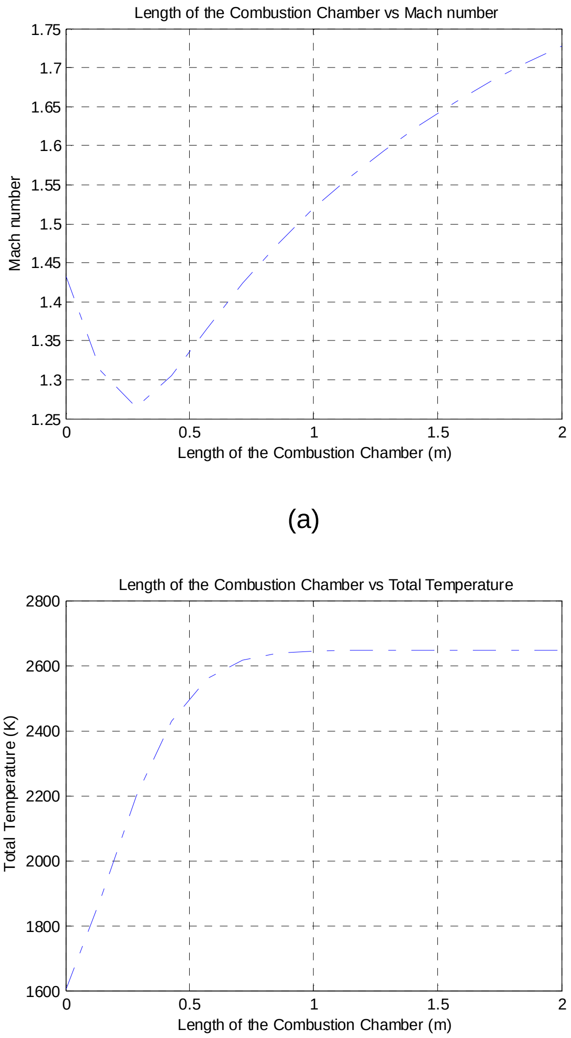


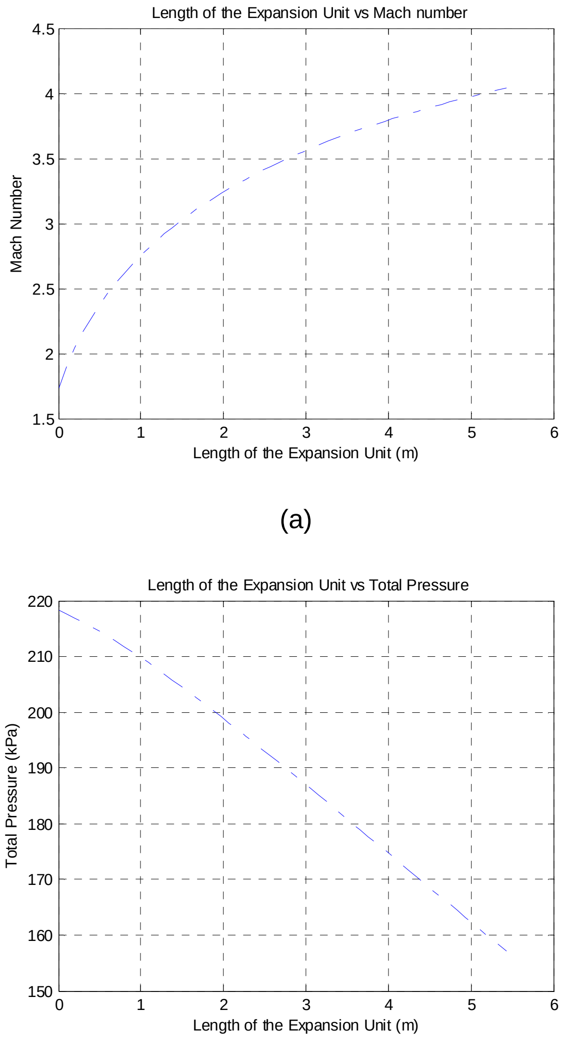
Figure A-17 Filled Contour of Mach Number at $M_o = 5.5$ Showing OST inside Isolator without Symmetry



(a)

(b)

Figure A-18 Axial variation of Flow Variable in Combustion Chamber at $M_o = 5.5$ (a) Total Temperature, (b) Mach Number



(a)

(b)

Figure A-19. Axial variation of Flow Variable in Expansion Unit at $M_o = 5.5$ (a) Mach Number, (b) Total Pressure

Station Number i	M_i	$p_i \ \left(kPa\right)$	$T_i \ \left({}^oK\right)$	$P_{t_i} \ \left(kPa\right)$	$T_{t_i} \ \left({}^oK\right)$	$\dot{m} \ \left(kg/s\right)$
0	5.5	2.223	222.44	2383.894	1447.2	26.78
2	2.88	34.879	596.23	1211.291	1508.22	25.55
3	2.64	39.535	656.44	941.370	1508.01	25.55
3_{AC}	1.43	139.061	1103.67	536.631	1508.41	25.55
4	1.72	43.665	1925.26	218.242	2646.25	27.28
10	4.07	0.704	773.24	157.149	2646.25	27.28

Figure A-20 Station flow properties for $M_o = 5.5$

Appendix B

Performance Results of DMScramjet Engine

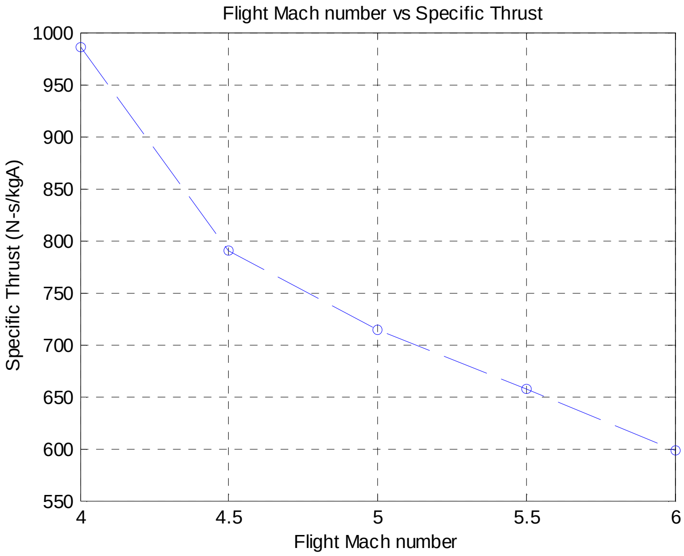


Figure B-1. Specific Thrust Plot for Various flight Mach numbers

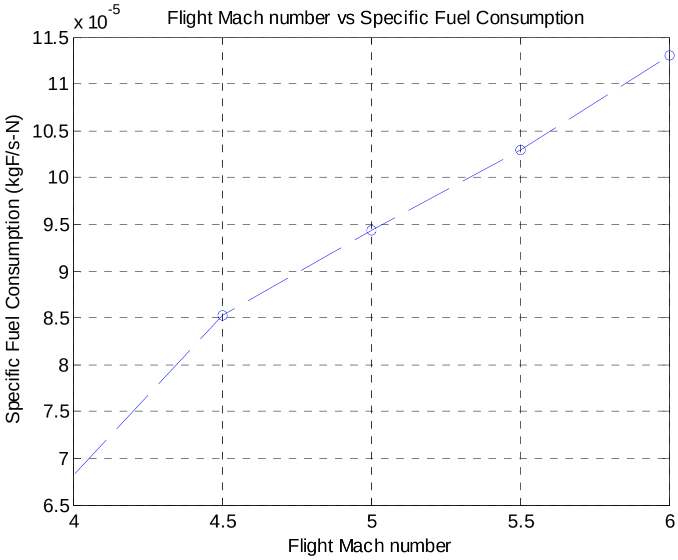


Figure B-2. Specific fuel consumption for various flight Mach numbers

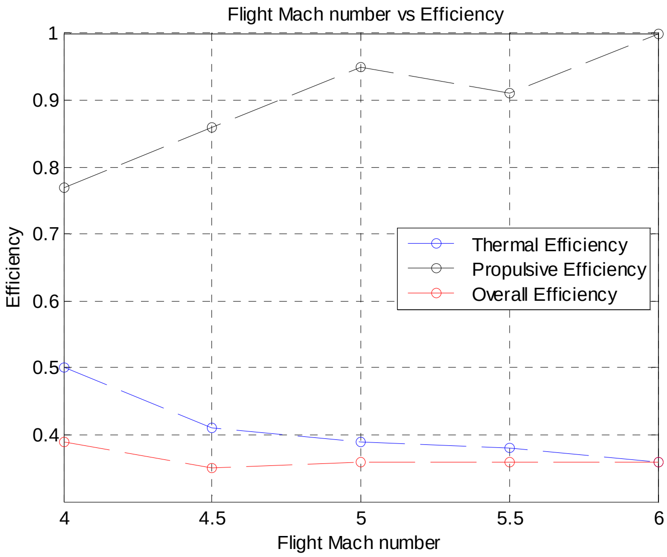


Figure B-3. Engine efficiency plot for various flight Mach numbers

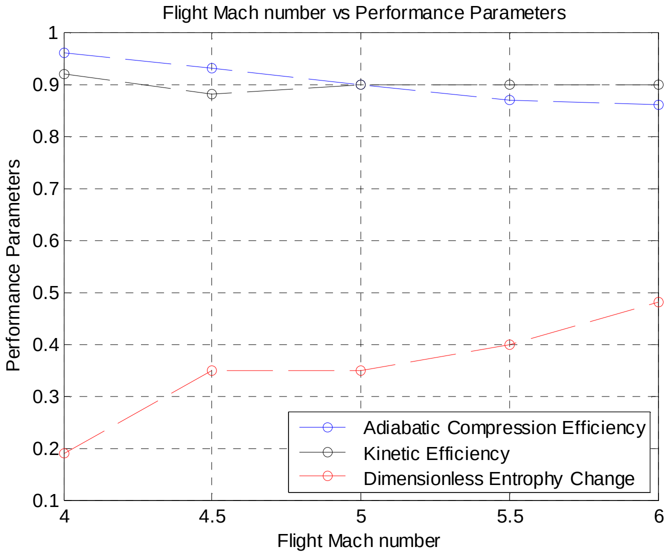


Figure B-4. Compression system performance plot for various flight Mach numbers

Figure B.5. Expansion System Performance Table for Various Flight Mach Numbers

M_o	π_e	η_e
3.5	0.676	0.938
4.0	0.716	0.944
4.5	0.727	0.946
5.0	0.730	0.943
5.5	0.727	0.940
6.0	0.735	0.940

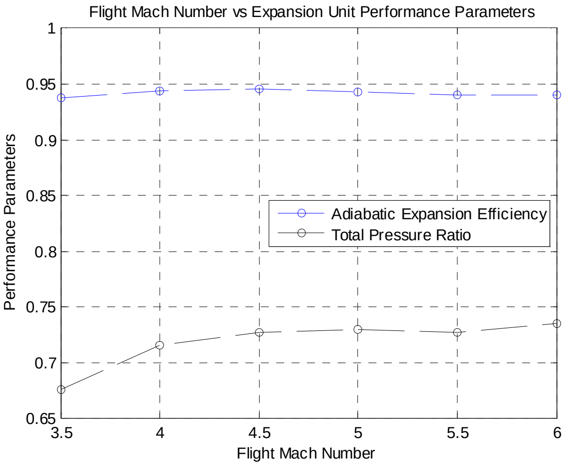


Figure B.6. Expansion System Performance Plot for Various Flight Mach Numbers

APPENDIX C

NPSS Code Built For DMScramjet

INLET ELEMENT

```
setThermoPackage("Janaf");
class SjIn extends Element {
//------------------------------------------------------------
  // ****** SETUP PORTS, FLOW STATIONS, SOCKETS, TABLES ********
  //------------------------------------------------------------

  Option switchdes {
    allowedValues = { "DESIGN", "OFFDESIGN" };
    description = "Design switch";
    trigger=FALSE;
    rewritableValues = TRUE;
  }
 //DATA PORT
DataOutputPort MachoutInlet;
DataOutputPort PoutInlet;
DataOutputPort ToutInlet;
DataOutputPort RhooutInlet;
DataOutputPort VoutInlet;
DataOutputPort MuoutInlet;
DataOutputPort PinInlet;
DataOutputPort VinInlet;
DataOutputPort TinInlet;
  // FLUID PORTS
  FluidInputPort Fl_I {
    description = "Incoming flow";

  }

  FluidOutputPort Fl_O {
    description = "Exiting flow";
             }

//Function
void fluentoutput(){
real Mach,TP,TT,MFR;
string border1,report,border2,Machoutlet,TPoutlet,TToutlet,MFRoutlet;
string TPunit,TTunit,MFRunit;
InFileStream takein;
takein.open("fluoutput3");
cout<<"     OUTPUT FROM ANSYS FLUENT"<<endl;
//takein >> border1;
border1=takein.getline();
report=takein.getline();
border2=takein.getline();
takein >> Machoutlet >> Mach;
takein >> TPoutlet >> TP >> TPunit;
takein >> TToutlet >> TT >> TTunit;
takein >> MFRoutlet >> MFR >> MFRunit;
```

```
cout<<  "     "<< border1 << endl;
cout<<  "     "<<report << endl;
cout<<  "     "<< border2 << endl;
cout<<  "Mach Number          "        << Mach <<"         "<< endl;
cout<<  "Total Pressure       "     << TP <<"         "<< TPunit << endl;
cout<<  "Total Temperature   "  << TT <<"         "<< TTunit<< endl;
//cout<<  "Mass Flow Rate       "  << MFR <<"         "<< MFRunit << endl;
// INITIALIZING THE VALUES OF THE NPSS EXIT PARAMETERS WITH FLUENT
OUTPUT
Mout=Mach;
//setting the units to be converted while feeding in
TtOut.units="R";
PtOut.units="psia";
Win.units="lbm/sec";
Mout.units="none";
TtOut=TT"K";
PtOut=TP"Pa";
Win=MFR "kg/sec";
}

// FLOW STATIONS

  FlowStation Fl_IInlet {

 setJanafComposition(
 0.0, // H2O
 0.000, // CO2
 0.21, // O2
 0.79, // N2
 0.0, // AR
 0.0, // CH4O
 0.0, // NH3
 .0, // H2
 0.0, // O
 0.0, // H
 0.0, // CO
 0.0, // OH
 0.0, // NO
 0.0, // N
 0.0, // CH4
 0.0, // CH4O
 0.0 // C10H8
 );
                description = "Inlet station to combustion section of burner
 (after the initial pressure loss is applied)";
  }

  FlowStation Fl_OIsolator {
    description = "Exit station to combustion section of burner
 (before thermal storage heat transfer is calculated)";
  }

   FlowStation Fl_IIsolator {

                description = "Inlet station to combustion section of burner
 (after the initial pressure loss is applied)";
  }
             // declaring variables
// Initializing the variables
    real gama=1.360;
    int shock;

 Fl_O.copyFlow("Fl_I");
           real  Win;
    real  TtIn;
    real  PtIn;
           real  TsIn;
    real  PsIn;
           real  htIn;
           real  Sin;
    real  RtIn;
    real  VIn;
    real  MIn;
    real  Tstream;
    real  Pstream;
    real  Vstream;

           // Setting Units


           //creating null parameters
           real Mout=0.0;
           real TtOut=0.0;
           real PtOut=0.0;
           real PsOut=0.0;
           real TsOut=0.0;
           real PR=0.0;
           real TR=0.0;
           real theta[10];
           real Inletcompeff;



 Socket OnDesign{
 allowedValues= {
"Fl_IInlet.Tt","Fl_IInlet.Pt","Fl_IInlet.MN","Fl_IInlet.W","TtOut","PtOut","PsOut","TsOut"}
 required = TRUE;
 description = "Socket for inviscid Pre calculation entering fluent";
}
```

```
 Socket Ondesignisolator{
 allowedValues= {"Fl_IIsolator.Tt","Fl_IIsolator.Pt","Fl_IIsolator.MN","Fl_IIsolator.W"}
 required = TRUE;
 description = "Socket for inviscid Pre calculation entering fluent";
}
            // Calculate of the element
void calculate(){
Fl_IInlet.W=Fl_I.W;
Fl_IInlet.MN=Fl_I.MN;
Fl_IInlet.setTotalTP(Fl_I.Tt,Fl_I.Pt);
PinInlet.val=Fl_IInlet.Ps;
VinInlet.val=Fl_IInlet.V;
TinInlet.val=Fl_IInlet.Ts;
cout<<"value in PinInlet = "<<PinInlet.val<<endl;
//cout << "fs static composition = " << Fl_IInlet.staticCompAll() << endl;
//cout<<"Fl_IInlet.Tt = "<<Fl_IInlet.Tt<<Fl_IInlet.Tt.units<<endl;
//cout<<"Fl_IInlet.Ts = "<<Fl_IInlet.Ts<<Fl_IInlet.Ts.units<<endl;
//cout<<"Fl_IInlet.Pt = "<<Fl_IInlet.Pt<<Fl_IInlet.Pt.units<<endl;
//cout<<"Fl_IInlet.Ps = "<<Fl_IInlet.Ps<<Fl_IInlet.Ps.units<<endl;
//cout<<"Fl_IInlet.MN = "<<Fl_IInlet.MN<<Fl_IInlet.MN.units<<endl;
    Win = Fl_I.W;
    TtIn= Fl_I.Tt;
    MIn = Fl_I.MN;
    PtIn= Fl_I.Pt;
    TsIn= Fl_I.Ts;
            PsIn= Fl_I.Ps;
            htIn = Fl_I.ht;
    Sin  = Fl_I.s;
    RtIn = Fl_I.Rt;
            VIn = Fl_I.V;

            // stream functions
    Vstream = Fl_I.V;
            Tstream = Fl_I.Ts;
            Pstream = Fl_I.Ps;
 if ( switchdes == "DESIGN" ) {
 //fluentinput(Fl_IInlet.Ps,Fl_IInlet.Ts,Fl_IInlet.MN);
OnDesign.execute();
//ASSIGING VALUES TO DATA PORTS

MachoutInlet.val=Fl_OStreamThrust.MN;
PoutInlet.val=Fl_OStreamThrust.Ps;
ToutInlet.val=Fl_OStreamThrust.Ts;
RhooutInlet.val=Fl_OStreamThrust.rhos;
VoutInlet.val=Fl_OStreamThrust.V;
MuoutInlet.val=Fl_OStreamThrust.mus;

//return 0;
Ondesignisolator.execute();
}

if ( switchdes == "OFFDESIGN" ) {

//TubulenceIntensity(Fl_IInlet.rhos,Fl_IInlet.V,Fl_IInlet.mus);
//cout<<"Turbulence Intensity = "<<I<<endl;
real PSSI,TSSI;
PSSI.units="Pa";
TSSI.units="K";
PSSI=Fl_IInlet.Ps"psia";
TSSI=Fl_IInlet.Ts"R";
fluentinput(PSSI,TSSI,Fl_IInlet.MN);
int fileExists(string fluoutput3);
fluentoutput();
cout<<"TtOut = "<< TtOut << TtOut.units<<endl;
cout<<"PtOut = "<< PtOut << PtOut.units <<endl;
cout<<"MOut = "<< Mout<< Mout.units <<endl;
Fl_OIsolator.W=Fl_I.W;
Fl_OIsolator.MN=Mout;
Fl_OIsolator.setTotalTP(TtOut,PtOut);
cout<<"Fl_OIsolator.Ts = "<<Fl_OIsolator.Ts<<Fl_OIsolator.Ts.units<<endl;
cout<<"Fl_OIsolator.Tt = "<<Fl_OIsolator.Tt<<Fl_OIsolator.Tt.units<<endl;
cout<<"Fl_OIsolator.Ps = "<<Fl_OIsolator.Ps<<Fl_OIsolator.Ps.units<<endl;
cout<<"Fl_OIsolator.MN = "<<Fl_OIsolator.MN<<Fl_OIsolator.MN.units<<endl;
Fl_O.W=Fl_OIsolator.W;
Fl_O.MN=Fl_OIsolator.MN;
Fl_O.setTotalTP(Fl_OIsolator.Tt,Fl_OIsolator.Pt);
//cout<<"result of fileExists "<<fileExists<<endl;

}

 }

 }
```

REALISTIC STREAM THRUST ANALYSIS SUBELEMENT

```
extern real Fl_IInlet.Tt;
extern real Fl_IInlet.Pt;
extern real Fl_IInlet.MN;
extern real Fl_IInlet.W;
extern real PsOut;
extern real TsOut;
extern real TtOut;
extern real PtOut;
```

```
// Flow stations
FlowStation Fl_IStreamThrust {

              description = "Inlet station to combustion section of burner
 (after the initial pressure loss is applied)";
  }

  FlowStation Fl_IInviscid {

              description = "Inlet station to combustion section of burner
 (after the initial pressure loss is applied)";
  }

  FlowStation Fl_OStreamThrust {

              description = "Inlet station to combustion section of burner
 (after the initial pressure loss is applied)";
  }
class StreamthrustCalc extends Subelement
{
int shock;
real R_Comp,To,Vo,Cpc,TR_Comp,Po,M2;
// Constants

//TR_Comp=3.2;
// Compression Component Station 0 to Station 3

real PR_Comp,gama_Comp,AR_Comp;
real Tout_Comp,Vout_Comp,aout_Comp,Mout_Comp;
real SaComp_one,SaComp_three;
real nc;


// CALCULATION SECTION
void calculate(){
Fl_IStreamThrust.W=Fl_IInlet.W;
Fl_IStreamThrust.MN=Fl_IInlet.MN;
Fl_IStreamThrust.setTotalTP(Fl_IInlet.Tt,Fl_IInlet.Pt);
cout<<"Incoming total pressure = "<<Fl_IInlet.Pt<<endl;
// ACCOUNTING FOR THE SHOCK LOSSES
//SHOCK TABLES CALCULATION
shock=6;
real lam,X,M,thetarad,nfric;
real beta,Mn1,Mn2;
real a,b,o,ncad,n1,temp1,dimlessentro;
int i;
real delta = 1.;//weak shock solution should be 0 for strong shock
//First phase of final equation
real BLCheck;
real Parray[100];
real Tarray[100];
M=Fl_IStreamThrust.MN;
gama=Fl_IStreamThrust.gams;
temp1=1;
for(i=0 ; i<shock ;i++)
{
thetarad=theta[i]*(3.14159/180);

lam = (((M*M-1)*(M*M-1))-(3*(1+(((gama-
1)/2)*M*M))*(1+(((gama+1)/2)*M*M))*(tan(thetarad)*tan(thetarad))))**0.5;
X = (((M**2-1)**3)-(9*(1+(((gama-1)/2)*M**2))*(1+(((gama-
1)/2)*M**2)+(((gama+1)/4)*M**4))*(tan(thetarad)**2)))/(lam**3);
//Numerator cos(((12.56637*delta)+(acos(X)))/3)
a=((M*M))-1+(2*lam*cos(((4*3.14159*delta)+acos(X))/3));
//denominator
b=(3*(1+(((gama-1)/2)*M**2))*tan(thetarad));
o=a/b;
//Shock Angle
beta=atan(o);
beta=beta*180/3.14159;
//cout<<"beta Angle = "<<beta<<endl;
BLCheck=KorkegiBLSP(M);
//theta[i]=theta[i]*180/3.14159;
// Normal components of the shock

Mn1=M*sin(beta*3.14159/180);
Mn2=(((Mn1**2)+(2/(gama-1)))/((((2*gama)/(gama-1))*Mn1**2)-1))**0.5;
// Final M out of every shock
Mout=Mn2/(sin((beta-theta[i])*3.14159/180));
//cout<<"Mout after each shock = "<<Mout<<endl;
// Pressure ratio of every shock
PR=1+((2*gama)/(gama+1)*((Mn1*Mn1)-1));

// Boundary Layer separation Check
if (PR>BLCheck)
{
cout << endl;
cout << endl;
cout<< "WARNING : Boundary Layer Separation Expected at the end of  "<< i+1 <<"th
shock wave";
cout << endl;
}

// To store the Pressure ratio of all the shock waves
Parray[i+1]=PR;

// Temperature ratio of every shock
TR=(((gama-1)*(Mn1*Mn1)+2)/((gama+1)*(Mn1*Mn1)))*PR;

//Adiabatic efficiency after every shock
ncad=(TR-((((gama-1)*Mn1**2+2)/((gama+1)*Mn1**2))*((2*gama*Mn1**2-(gama-
1))/(gama+1))**(1/gama)))/(TR-1);

//cout<<"Efficiency after each shock = "<<ncad<<endl;
dimlessentro=log(TR*(1-ncad)+ncad);
//cout<<"dimless entrophy = "<<dimlessentro<<endl;
//cout<<"TR = "<<TR<<endl;
//cout<<"PR = "<<PR<<endl;

temp1=ncad*temp1;
cout<<"Overall Efficiency = "<<temp1<<endl;
//
Tarray[i+1]=TR;
// Commands for the next loop
M=Mout;
// Static Temperature and Pressure
TsOut=TsIn*TR;
PsOut=PR*PsIn;
TsIn=TsOut;
PsIn=PsOut;
// Total Temperature and Pressure
TtOut=TsOut*(1+(((gama-1)/2)*(Mout**2)));
PtOut=PsOut*(1+(((gama-1)/2)*(Mout**2)))**(gama/(gama-1));
cout<<"Ptout = "<<PtOut<<endl;
Fl_IInviscid.MN=Mout;
Fl_IInviscid.setTotalTP(TtOut,PtOut);
gama=Fl_IInviscid.gams;
cout<<"Value of gama is = "<<gama<<endl;
if (i==0)
{
KANTROWITZLIMIT(Mout,gama);// Kantrowitz Limit Calculation
MAXIMUMCR(MIn);// Maximum Contraction Limit Calculation
}
}
n1=temp1;

real y=1.,s=1.;

for(i=0 ; i<shock ;i++)
{
real k,l;
            k=Parray[i+1];
            l=k*y;
    y=l;
real e,r;
            e=Tarray[i+1];
            r=e*s;
    s=r;
}
```

```
// Pressure Ratio and Temperature Ratios
PR=y;
TR=s;
cout<<"Pressure Ratio due to oblique shocks = "<<PR<<endl;
cout<<"Temperature Ratio due to oblique shocks = "<<TR<<endl;

//Accounting for the Frictions Losses
real TR1,PR1,Tt2,T2;
M=Fl_IStreamThrust.MN;
//gama=1.36;
TR1=(1+(((gama-1)/2)*(M**2)))*(1-(1+0.01)**2*(1+(2*(1-TR)/((gama-1)*M**2))));

cout<<"Temperature Ratio due to friction = "<<TR1<<endl;

TR=TR*TR1;
T2=Fl_IStreamThrust.Ts*TR;
Tt2=Fl_IStreamThrust.Tt;
//Tt2=TsOut*(1+(((gama-1)/2)*(Mout**2)));
M2=(((Tt2/T2)-1)/((gama-1)/2))**0.5;
cout<<"Final Mach number at station 2= "<<M2<<endl;
cout<<"Final Total Temperature at station 2= "<<Tt2<<endl;
cout<<"Final Total Pressure at station 2= "<<PsOut<<endl;
cout<<"Final Static Temperature at station 2= "<<T2<<endl;
//PR1=(Fl_IStreamThrust.Pt/Fl_IStreamThrust.Ps)*(((1+(((gama-
1)/2)*(M2**2)))/(1+(((gama-1)/2)*(M**2)))))**(gama/(gama-1));
//<<"pressure Ratio due to friction = "<<PR1<<endl;
//PR=PR*PR1;
cout<<"Final Temperature Ratio at station 2= "<<TR<<endl;
cout<<"Final pressure Ratio  at station 2= "<<PR<<endl;
nc=n1*((1-(1/s))/(1-(1/TR)));//(1-(1/PR)**((gama-1)/gama))/(1-(1/TR));
cout<<"TOTAL ADIABATIC EFFICIENCY  EFFICIENCY = "<<nc<<endl;



//FINAL Stream Thrust Calculation
To= Fl_IStreamThrust.Ts;
cout<<"Fl_IStreamThrust.Ts =
"<<Fl_IStreamThrust.Ts<<Fl_IStreamThrust.Ts.units<<endl;
Vo= Fl_IStreamThrust.V;
Po= Fl_IStreamThrust.Ps;
cout<<"Incoming Vo = "<<Vo<<Vo.units<<endl;
real mueo,rhoo;
rhoo.units="kg/m3";
rhoo=Fl_IStreamThrust.rhos"lbm/ft3";
cout<<"Incoming rhoo = "<<rhoo<<rhoo.units<<endl;
Cpc.units="Btu/(lbm*R)";
Cpc=1510."J/(kg*K)";
cout<<"Cpc = "<<Cpc<<Cpc<<endl;
gama_Comp=1.36;
cout<<"gama_Comp = "<<gama_Comp<<endl;
```

```
 R_Comp.units ="Btu/(lbm*R)";
  R_Comp =289.3"J/(kg*K)";
  cout<<"R_Comp = "<<R_Comp<<endl;
 cout<<"Fl_IStreamThrust.Rt =
"<<Fl_IStreamThrust.Rt<<Fl_IStreamThrust.Rt.units<<endl;
// Compression Component Station 0 to Station 3

// Computations Necessary for Compression Component
SaComp_one = Vo*(1+((R_Comp*To)/(Vo**2/25037)));
cout<<"SaComp_one = "<<SaComp_one<<endl;
//Assigning values to the local variables
TR_Comp=TR;
cout<<" TR_Comp= "<<TR_Comp<<endl;
Tout_Comp=TR_Comp*To;
cout<<"Tout_Comp = "<<Tout_Comp<<endl;
aout_Comp = (gama*R_Comp*Tout_Comp)**0.5;
cout<<"aout_Comp = "<<aout_Comp<<endl;
real ao;
ao=(gama*R_Comp*To)**0.5;
cout<<"Cpc = "<<Cpc<<endl;
cout<<"Vo = "<<Vo<<endl;
cout<<"To = "<<To<<endl;
cout<<"TR_Comp = "<<TR_Comp<<endl;
Vout_Comp = (((Vo**2))-((2*Cpc*To)*(TR_Comp-1)))**0.5;
cout<<"Vout_Comp = "<<Vout_Comp<<endl;
SaComp_three=Vout_Comp*(1+((R_Comp*Tout_Comp)/(Vout_Comp**2/25037)));
PR_Comp= PR;//((TR)/((TR*(1-nc))+nc))**(Cpc/R_Comp);
// Initialising local Station
real Pout_Comp;
Pout_Comp=PR*Fl_IStreamThrust.Ps;
TtOut=T2*(1+(((gama-1)/2)*(M2**2)));
//PtOut=Pout_Comp*(1+(((gama-1)/2)*(M2**2)))**(gama/(gama-1));
cout<<"Ttout = "<<TtOut<<endl;
cout<<"Ptout = "<<PtOut<<endl;
Fl_OStreamThrust.W=Fl_IStreamThrust.W;
Fl_OStreamThrust.MN=M2;
Fl_OStreamThrust.setTotalTP(TtOut,PtOut);
cout<<"PR_Comp = "<<PR_Comp<<endl;
AR_Comp=TR*(1/PR)*(Vo/Fl_OStreamThrust.V);
cout<<"AR_Comp = "<<AR_Comp<<endl;


// GIVING THE FLOW VALUES TO THE ISOLATOR SUBELEMENT
Fl_IIsolator.copyFlow("Fl_OStreamThrust");
Fl_IIsolator.MN=Fl_OStreamThrust.MN;
 cout<<"Fl_IIsolator.Rt = "<<Fl_IIsolator.Rt<<Fl_IIsolator.Rt.units<<endl;

}

}
```

ISOLATOR FANNO FLOW SUBELEMENT

```
extern real Fl_IIsolator.Tt;
extern real Fl_IIsolator.Pt;
extern real Fl_IIsolator.MN;
extern real Fl_IIsolator.W;
class SJIsolator extends Subelement {
//-----------------------------------------------------------
  // ****** SETUP PORTS, FLOW STATIONS, SOCKETS, TABLES ********
  //-----------------------------------------------------------


// FLOW STATIONS


  FlowStation Fl_OIsolator {
    description = "Exit station to combustion section of burner
 (before thermal storage heat transfer is calculated)";
  }

            // declaring variables
// Initializing the variables
    real gama;


 Socket Fanno{
 allowedValues= {
"Fl_IFannoFlow.Tt","Fl_IFannoFlow.Pt","Fl_IFannoFlow.MN","Fl_IFannoFlow.W"}
 required = TRUE;
 description = "Socket for inviscid Pre calculation entering fluent";
}


// FLOW STATIONS for property calculation according to the fuel burn
 FlowStation Fl_IFannoFlow {
                description = "Local flow station for Stream thrust calculation at the
Combustion chamber entry";
}

  FlowStation Fl_OFannoFlow {
    description = "Local flow station for Stream thrust calculation at the Combustion
chamber exit";
  }

  // CALCULATION SECTION


void calculate(){
```

```
// Stream Thrust Analysis Calculation
Fl_IFannoFlow.W=Fl_IIsolator.W;
Fl_IFannoFlow.MN=Fl_IIsolator.MN;
Fl_IFannoFlow.setTotalTP(Fl_IIsolator.Tt,Fl_IIsolator.Pt);
// One dimensional Calculation by Influence Co efficient Method
Fanno.execute();
}//Calculate ends

 }
```

ISOLATOR ONDESIGN SUBELEMENT

```
extern real Fl_IFannoFlow.Tt;
extern real Fl_IFannoFlow.Pt;
extern real Fl_IFannoFlow.MN;
extern real Fl_IFannoFlow.W;

class IsoOnDesign extends Subelement
{
// FLOW STATIONS for property calculation according to the fuel burn
 FlowStation Fl_IFannoFlow {
                description = "Local flow station for Stream thrust calculation at the
Combustion chamber entry";
}

  FlowStation Fl_OFannoFlow {
    description = "Local flow station for Stream thrust calculation at the Combustion
chamber exit";
  }

// CALCULATION SECTION


void calculate(){

// Stream Thrust Analysis Calculation
Fl_IFannoFlow.W=Fl_IIsolator.W;
Fl_IFannoFlow.MN=Fl_IIsolator.MN;
Fl_IFannoFlow.setTotalTP(Fl_IIsolator.Tt,Fl_IIsolator.Pt);
// One dimensional Calculation by Influence Co efficient Method

// BURNING THE FUEL FOR REQUIRED EFFICIENCY

cout<<endl;
int i;
int length,no;
real Mnew;
real x,X,nc,Tt,A,H,Dh,DTt,fric,xinitial,temp8;
real M2,Mr,T2,M,M3,T3;
real theta=5.;
real Wiso,Dh2,Dh3,Tt2,Tt3,A2,A3,P2,P4,P,Pt,f,towb;
real h,Pprev,Pratio,Pt2;
real k1,k2,sigma,xsp;
real Tterm,Aterm,fterm,BSP;
real temp1,temp2,temp3,temp4,temp5,temp9,temp10;
real dMdx;
real T;
real nctot,Ml;
real Cp;
real AR=4.;
real x2=0.,x3=2.;
length=x3-x2;
real Hiso=0.152;
real gama=1.238,Hcomb;
real fst=0.0674;
real phi=1.0;
real j=0.,dA;
real slope;

real temp6,Ttinitial,Ptinitial,Pt3;
xinitial=0.;
h=0.1;
Wiso=Hiso*AR;
Dh2=(2*Hiso*Wiso)/(Hiso+Wiso);
A2=Hiso*Wiso;
A3=A3;
slope=A3-A2;
cout<<"A3 = "<<A2<<endl;

//Assigning Values to the Variables
M2=Fl_IFannoFlow.MN;
T2=Fl_IFannoFlow.Ts;
Tt2=Fl_IFannoFlow.Tt;
T2=Fl_IFannoFlow.Ts;
//cout<<"Tt3 ="<<Tt3<<endl;
Pt2=Fl_IFannoFlow.Pt;
P2=Fl_IFannoFlow.Ps;
//cout<<"P3 ="<<P3<<endl;
//Tt4=Fl_Ostreamthrust.Tt;
Cp=Fl_IFannoFlow.Cps;
//Cp.units="Btu/(lbm*R)";
//P3=Fl_Ionedimensional.Ps;
f=0.002;//pg 189 heiser book

//Functions to solve the RK

real FannovalueK1(real xi,real Mi){
cout<<endl;
x=xi;
M=Mi;
X= (x-xinitial)/length;
gama=Fl_IFannoFlow.gams;
//cout<<"gama = "<<gama<<endl;
H=Hiso;
Dh=(H*2*Wiso)/(H+Wiso);
fric=(4*f)/Dh;
A=A2;
sigma=1+((gama-1)/2*M**2);
k1=(M*sigma/(1-M**2))*(((gama*M**2/2)*fric));
return k1;
}

real FannovalueK2(real xi,real Mi){
x=xi+h;
Mr=Mi+(k1*h);
gama=Fl_IFannoFlow.gams;
X= (x-xinitial)/length;
H=Hiso;
Dh=(H*2*Wiso)/(H+Wiso);
fric=(4*f)/Dh;
A=A2;
sigma=1+((gama-1)/2*Mr**2);
k2=(Mr*sigma/(1-Mr**2))*(((gama*Mr**2/2)*fric));
return k2;
}

for(i=0;x2<=length;i++){
FannovalueK1(x2,M2);
FannovalueK2(x2,M2);
Mnew=M2+(h*(k1/2+k2/2));
Tt=Tt2;
T=T2*(Tt/Tt2)*((1+((gama-1)/2*M2**2))/(1+((gama-1)/2*Mnew**2)));
//T.units="R";
P=P2*(A2/A)*(Fl_IFannoFlow.MN/Mnew)*(T/T2)**0.5;
Pt=P*(1+((gama-1)/2)*Mnew**2)**(gama/(gama-1));
Fl_IFannoFlow.setTotalTP(Fl_IIsolator.Tt,Pt);
cout<<"P = "<<P<<endl;
cout<<"Pt = "<<Pt<<endl;
x2=x2+h;
M2=Mnew;
//cout<<M2<<endl;
}
M3=M2;

Pt3=Pt;
Tt3=Tt;
Fl_OFannoFlow.W=Fl_IFannoFlow.W;
Fl_OFannoFlow.MN=M3;
Fl_OFannoFlow.setTotalTP(Tt3,Pt3);
```

```
cout<<"Fl_OFannoFlow.MN =
"<<Fl_OFannoFlow.MN<<Fl_OFannoFlow.MN.units<<endl;
cout<<"Fl_OFannoFlow.Ts = "<<Fl_OFannoFlow.Ts<<Fl_OFannoFlow.Ts.units<<endl;
cout<<"Fl_OFannoFlow.Ps = "<<Fl_OFannoFlow.Ps<<Fl_OFannoFlow.Ps.units<<endl;
cout<<"Fl_OFannoFlow.Tt = "<<Fl_OFannoFlow.Tt<<Fl_OFannoFlow.Tt.units<<endl;
cout<<"Fl_OFannoFlow.Pt = "<<Fl_OFannoFlow.Pt<<Fl_OFannoFlow.Pt.units<<endl;

Fl_O.W=Fl_OFannoFlow.W;
Fl_O.MN=Fl_OFannoFlow.MN;
Fl_O.setTotalTP(Fl_OFannoFlow.Tt,Fl_OFannoFlow.Pt);

}//Calculate ends
}
```

SCRAMJET BURNER ELEMENT

```
class SJBurner extends Element {
  // Setting Thermopackage for Burner element
setThermoPackage("allFuel");

  //-----------------------------------------------------------
  //     ******* SETUP VARIABLES ********
  //-----------------------------------------------------------
   int FLUENT;
   real Tt4 = 0.0;
  real Pt4=0.0;
  real M4=0.0;
  real M2,T2,P2,rho2,V2,mu2,AR_Comb,TtOut,PtOut,Mout,Win;
  AR_Comb=0.0;
  //-----------------------------------------------------------
  //   ******* OPTION VARIABLE SETUP *******
  //-----------------------------------------------------------


  Option switchdes {
    allowedValues = { "DESIGN", "OFFDESIGN" };
    description = "Design switch";
    trigger=FALSE;
    rewritableValues = TRUE;
  }
 Option switchmethod {
    allowedValues = { "FLUENT","EMPRICAL" };
    description = "Design switch";
    trigger=FALSE;
    rewritableValues = TRUE;
  }
  // input kept in for backward compatible (remove later)

  //-----------------------------------------------------------
  // ****** SETUP PORTS, FLOW STATIONS, SOCKETS, TABLES ********
  //-----------------------------------------------------------

  // FLUID PORTS
  FluidInputPort Fl_I {
    description = "Incoming flow";
  }

  FluidOutputPort Fl_O {
    description = "Exiting flow";
  }


// FUEL PORTS

  FuelInputPort Fu_I {
    description = "Incoming fuel flow";
  }

  //FuelInputPort Fu_I {
  // description = "Incoming fuel flow";
  //}
  //DATAPORT

DataInputPort MachoutInlet;
DataInputPort PoutInlet;
DataInputPort ToutInlet;
DataInputPort RhooutInlet;
DataInputPort VoutInlet;
DataInputPort MuoutInlet;
DataOutputPort COMBAR;
  // BLEED PORTS

  // THERMAL PORTS

  // MECHANICAL PORTS

  // FLOW STATIONS

  FlowStation Fl_Icomb {
              description = "Inlet station to combustion section of burner
 (after the initial pressure loss is applied)";
  }

    FlowStation Fl_IIsolator {
              description = "Inlet station to combustion section of burner
 (after the initial pressure loss is applied)";
  }

  FlowStation Fl_Ocomb {
    description = "Exit station to combustion section of burner
 (before thermal storage heat transfer is calculated)";
  }

  // SOCKETS

  Socket OnDesign {
    allowedValues = {
"Fl_Icomb.Tt","AR_Comb","Fl_Icomb.Pt","Fl_Icomb.MN","Pt4","Tt4","M4","Fu_I.LHV","Fu
_I.Wfuel","Fl_Icomb.W","FLUENT" };
    description = "Dry duct pressure loss";
            required = TRUE;

  }
 Socket OffDesign {
    allowedValues = {
"Fl_Icomb.Tt","Fl_Icomb.Pt","Fl_Icomb.MN","Pt4","Tt4","M4","Fu_I.LHV","Fu_I.Wfuel","FL
UENT","Fl_Icomb.W","mu2","rho2","V2","P2","T2","M2" };
    description = "Dry duct pressure loss";
            required = TRUE;

  }
    void fluentoutput(){
real Mach,TP,TT,MFR;
string border1,report,border2,Machoutlet,TPoutlet,TToutlet,MFRoutlet;
string TPunit,TTunit,MFRunit;
InFileStream takein;
takein.open("fluentoutput");
cout<<"     OUTPUT FROM ANSYS FLUENT"<<endl;
cout<<" TAKEN FOR BURNER "<<endl;
//takein >> border1;
border1=takein.getline();
report=takein.getline();
border2=takein.getline();
takein >> Machoutlet >> Mach;
takein >> TPoutlet >> TP >> TPunit;
takein >> TToutlet >> TT >> TTunit;
takein >> MFRoutlet >> MFR >> MFRunit;

cout<<  "     "<< border1 << endl;
cout<<  "     "<<report << endl;
cout<<  "     "<< border2 << endl;
cout<<  "Mach Number         "        << Mach <<"         "<< endl;
cout<<  "Total Pressure      "     << TP <<"         "<< TPunit << endl;
cout<<  "Total Temperature   "  << TT <<"         "<< TTunit<< endl;
//cout<<  "Mass Flow Rate      "  << MFR <<"         "<< MFRunit << endl;
```

```
// INITIALIZING THE VALUES OF THE NPSS EXIT PARAMETERS WITH FLUENT
OUTPUT
Mout=Mach;
//setting the units to be converted while feeding in
TtOut.units="R";
PtOut.units="psia";
Win.units="lbm/sec";
Mout.units="none";
TtOut=TT"K";
PtOut=TP"Pa";
Win=MFR "kg/sec";
}

  // TABLES

  //----------------------------------------------------------
  // ******* INTERNAL SOLVER SETUP *******
  //----------------------------------------------------------

  //----------------------------------------------------------
  //  ******  ADD SOLVER INDEPENDENTS & DEPENDENTS  ******
  //----------------------------------------------------------

  //----------------------------------------------------------
  // ******* VARIABLE CHANGED METHODOLOGY *******
  //----------------------------------------------------------
  void calculate() {
  real gama=1.238;
  real Tt,Pt,Ts,M,Ps;
  M2=MachoutInlet.val;
  cout<<"M2 in element = "<<M2<<endl;
  P2=PoutInlet.val;
  T2=ToutInlet.val;
  rho2=RhooutInlet.val;
  V2=VoutInlet.val;
  mu2=MuoutInlet.val;
//cout<<"MachoutInlet = "<<MachoutInlet.val<<endl;
//cout<<"PoutInlet = "<<PoutInlet.val<<endl;
//cout<<"ToutInlet = "<<ToutInlet.val<<endl;
//cout<<"Wfuel = "<<Fu_I.Wfuel<<Fu_I.Wfuel.units<<endl;
 //Fl_O.copyFlow( "Fl_I" );
  //M=1.22;
  //Ps.units="psia";
  //Ts.units="R";
  //Ps=108502.1"Pa";
  //Ts=572.68"K";
  Fl_Icomb.W=Fl_I.W;
//Tt=Ts*(1+((gama-1)/2)*M**2);
//cout<<"Tt = "<<Tt<<Tt.units<<endl;
//Pt=Ps*(1+((gama-1)/2)*M**2)**(gama/(gama-1));
```

```
  // Copy the total and static conditions from the inlet fluid port to the combustion flow
station
            //Fl_Icomb.copyFlow( "Fl_I" );
            Fl_Icomb.MN=Fl_I.MN;
            Fl_Icomb.W=Fl_I.W;
            Fl_Icomb.setTotalTP( Fl_I.Tt, Fl_I.Pt );

            Fl_O.copyFlow( "Fl_Icomb" );
            //cout<<"Fl_Icomb.Tt = "<<Fl_Icomb.Tt<<endl;
            //cout<<"Fl_Icomb.MN = "<<Fl_Icomb.MN<<endl;
      if ( switchdes == "DESIGN" ) {
              if(switchmethod == "FLUENT"){
              FLUENT=1;
              }
               if(switchmethod == "EMPRICAL"){
              FLUENT=0;
              }
             OnDesign.execute();
             COMBAR.val=AR_Comb;
             }



              if ( switchdes == "OFFDESIGN" ) {
              if(switchmethod == "FLUENT"){
              FLUENT=1;
              }
              if(switchmethod == "EMPRICAL"){
              FLUENT=0;
              }
              fluentoutput();
Fl_IIsolator.MN=Mout;

Fl_IIsolator.W=Fl_Icomb.W;
Fl_IIsolator.setTotalTP( TtOut, PtOut );
V2=Fl_IIsolator.V;
P2=Fl_IIsolator.Ps;
T2=Fl_IIsolator.Ts;
rho2=Fl_IIsolator.rhos;
mu2=Fl_IIsolator.mus;
M2=Fl_IIsolator.MN;
cout<<"FLOW PROPERTIES BEFORE ISOLATOR "<<endl;
cout<<"Fl_IIsolator.Ts = "<< Fl_IIsolator.Ts << Fl_IIsolator.Ts.units<<endl;
cout<<"Fl_IIsolator.Ps = "<< Fl_IIsolator.Ps << Fl_IIsolator.Ps.units <<endl;
cout<<"Fl_IIsolator.MN = "<< Fl_IIsolator.MN<< Fl_IIsolator.MN.units <<endl;
             OffDesign.execute();
            }


//cout<<"Fl_O.W = "<<Fl_O.W<<Fl_O.W.units<<endl;
```

```
       // Set the combustion flow station inlet conditions
  //Fl_Ocomb.setTotal_hP( hin, Pin );
       //Fl_Ocomb.MN=Min;

 }

 // Fl_O.setTotalTP(Tt4,Pt4);
  //Fl_O.MN=M4;
  }
```

SCRAMJET BURNER ON-DESIGN CALCULATION SUBELEMENT

```
extern real Fl_Icomb.Tt;
extern real Fl_Icomb.Pt;
extern real Fl_Icomb.MN;
extern real Fl_Icomb.W;
extern real Pt4,Tt4,M4;
extern real Fu_I.Wfuel;
extern real Fu_I.LHV;
extern real AR_Comb;

class BrnOnDesign extends Subelement
{
// FLOW STATIONS for property calculation according to the fuel burn
 FlowStation Fl_Istreamthrust {
        description = "Local flow station for Stream thrust calculation at the
Combustion chamber entry";
             // = "JP";
             //Dissociated = "ON";

 }

 FlowStation Fl_IJustfuel {
        description = "Local flow station for Stream thrust calculation at the
Combustion chamber entry";
             switchFuelType = "JP";
             Dissociated = "ON";
             FAR=0.0674;
             WAR=0.0;
 }

  FlowStation Fl_IJustfuel2 {
        description = "Local flow station for Stream thrust calculation at the
Combustion chamber entry";
             switchFuelType = "JP";
             Dissociated = "ON";
             FAR=0.0674;
             WAR=0.0;
 }
```

```
  FlowStation Fl_Ostreamthrust {
    description = "Local flow station for Stream thrust calculation at the Combustion
chamber exit";
  }

  FlowStation Fl_Ionedimensional {
		description = "Local flow station for One Dimensional calculation at the
Combustion chamber entry";
  }
  FlowStation Fl_Oonedimensional {
		description = "Local flow station for One Dimensional calculation at the
Combustion chamber entry";
  }

  // Local Fuel Station
   FuelStation fus;
   FuelStation fus2;
   // Set fuel station flow


// Initial Stream Thrust Calculation
real R_Comb,To,Vo,Cpc,Cpb,Cpe,hPR,TR_Comp,Po;

// Constants

real nb=0.9;

// Combustion  Component Station 3 to Station 4
// Assumption
real avelratio=0.5,totvelratio=0.5,gama_Comb=1.238;
real hf=0.0;
real burndrag=0.10, f=0.0674,fst=0.0674,Tabs;

// Units Specification
R_Comb.units="Btu/(lbm*R)";
R_Comb=290.29"J/(kg*K)";
Tabs.units="R";
Tabs=222.0"K";
cout<<"Tabs = "<<Tabs<<Tabs.units<<endl;
Cpb.units="Btu/(lbm*R)";
Cpb=1510.0"J/(kg*K)";
cout<<"Cpb = "<<Cpb<<Cpb.units<<endl;
hPR.units="Btu/lbm";
hPR=43903.250"kJ/kg";
cout<<"hPR = "<<hPR<<hPR.units<<endl;
f.units="none";
nb.units="none";
hf.units="none";
totvelratio.units="none";
avelratio.units="none";
```

```
burndrag.units="none";

// CALCULATION SECTION


void calculate(){

// Stream Thrust Analysis Calculation
Fl_Istreamthrust.W=Fl_Icomb.W;
Fl_Istreamthrust.MN=Fl_Icomb.MN;
Fl_Istreamthrust.setTotalTP(Fl_Icomb.Tt,Fl_Icomb.Pt);
cout<<"Fl_Istreamthrust.Tt = "<<Fl_Istreamthrust.Tt<<Fl_Istreamthrust.Tt.units<<endl;
cout<<"Fl_Istreamthrust.Pt = "<<Fl_Istreamthrust.Pt<<Fl_Istreamthrust.Pt.units<<endl;
cout<<"Fl_Istreamthrust.MN =
"<<Fl_Istreamthrust.MN<<Fl_Istreamthrust.MN.units<<endl;
cout<<"Fl_Istreamthrust.V = "<<Fl_Istreamthrust.V<<Fl_Istreamthrust.V.units<<endl;
cout<<"Fl_Istreamthrust.Ts = "<<Fl_Istreamthrust.Ts<<Fl_Istreamthrust.Ts.units<<endl;
cout<<"Fl_Istreamthrust.W = "<<Fl_Istreamthrust.W<<Fl_Istreamthrust.W.units<<endl;
// Connecting with the mass flow rate entry after the isolator
cout<<endl;
cout<<endl;
cout<<endl;
Fl_IJustfuel.MN=Fl_Istreamthrust.MN;
Fl_IJustfuel.copyFlow( "Fl_Istreamthrust" );

Fl_IJustfuel2.MN=Fl_Istreamthrust.MN;
Fl_IJustfuel2.copyFlow( "Fl_Istreamthrust" );
cout<<"Fl_IJustfuel.Tt = "<<Fl_IJustfuel.Tt<<Fl_Istreamthrust.Tt.units<<endl;
cout<<"Fl_IJustfuel.Pt = "<<Fl_IJustfuel.Pt<<Fl_Istreamthrust.Pt.units<<endl;
cout<<"Fl_IJustfuel.MN = "<<Fl_IJustfuel.MN<<Fl_Istreamthrust.MN.units<<endl;
cout<<"Fl_IJustfuel.V = "<<Fl_IJustfuel.V<<Fl_Istreamthrust.V.units<<endl;
cout<<"Fl_IJustfuel.Ts = "<<Fl_IJustfuel.Ts<<Fl_Istreamthrust.Ts.units<<endl;
cout<<"Fl_IJustfuel.W = "<<Fl_IJustfuel.W<<Fl_Istreamthrust.W.units<<endl;
cout<<"Fl_IJustfuel.Ps = "<<Fl_IJustfuel.Ps<<Fl_Istreamthrust.Ps.units<<endl;
//Fl_Istreamthrust.




// Set Local Variables
real
Vout_Comb,Tout_Comb,aout_Comb,Mout_Comb,SaComb_four,Pt_Comb,Ps_Comb,Tt_
Comb;
real terma,termb,termc;

//Assigning the terms from flowstation
real Vout_Comp=Fl_Istreamthrust.V;
real Tout_Comp=Fl_Istreamthrust.Ts;
Ps_Comb=Fl_Istreamthrust.Ps;
// Computations Necessary for Combustion Component
```

```
Vout_Comp.units="ft/sec";
Vout_Comb.units="ft/sec";
Tout_Comp.units="R";
// Constant Pressure Combustion
Vout_Comb = Vout_Comp*(((1+(f*avelratio))/(1+f))-(burndrag/(2*(1+f))));
cout<<"Vout_Comp = "<<Vout_Comp<<Vout_Comp.units<<endl;
cout<<"Vout_Comb = "<<Vout_Comb<<endl;
cout<<"Tout_Comp = "<<Tout_Comp<<endl;
terma=Tout_Comp/(1+f);//
termb=
1+((1/(Cpb*Tout_Comp))*((nb*f*hPR)+(f*hf)+(f*Cpb*Tabs)+((1+(f*(totvelratio**2)))*((Vout
_Comp**2)/(2*25037)))));
termc= (Vout_Comb**2)/(2*25037*Cpb);

//cout<<"Terma = "<<terma<<endl;
//cout<<"Termb = "<<termb<<endl;
//cout<<"Termc = "<<termc<<endl;
Tout_Comb = (terma*termb)-termc;
aout_Comb = (gama_Comb*R_Comb*Tout_Comb)**0.5;
Mout_Comb = Vout_Comb/(158.23*aout_Comb);
cout<<"Mout_Comb = "<<Mout_Comb<<Mout_Comb.units<<endl;
cout<<"Tout_Comb looking= "<<Tout_Comb<<endl;
Tt_Comb=Tout_Comb*(1+((gama_Comb-1)/2)*Mout_Comb**2);
cout<<"Ttout_Comb looking= "<<Tt_Comb<<endl;
AR_Comb=(1+f)*(Tout_Comb/Tout_Comp)*(Vout_Comp/Vout_Comb);
cout<<"ARout_Comb looking= "<<AR_Comb<<endl;
SaComb_four=Vout_Comb*(1+((R_Comb*Tout_Comb)/Vout_Comb**2));
Pt_Comb=Ps_Comb*(1+((gama_Comb-
1)/2)*Mout_Comb**2)**(gama_Comb/(gama_Comb-1));

// Setting the Flowstation on the Basis of Streamthrust Calculation
Fl_Ostreamthrust.V=Vout_Comb;
Fl_Ostreamthrust.setTotalTP(Tt_Comb,Pt_Comb);
cout<<"Fl_Ostreamthrust.MN =
"<<Fl_Ostreamthrust.MN<<Fl_Ostreamthrust.MN.units<<endl;
cout<<"Fl_Ostreamthrust.Ts =
"<<Fl_Ostreamthrust.Ts<<Fl_Ostreamthrust.Ts.units<<endl;
cout<<"Fl_Ostreamthrust.Tt =
"<<Fl_Ostreamthrust.Tt<<Fl_Ostreamthrust.Ts.units<<endl;
cout<<"Fl_Ostreamthrust.Pt =
"<<Fl_Ostreamthrust.Pt<<Fl_Ostreamthrust.Pt.units<<endl;

// Area Calculation
real H3,H4;


// One dimensional Calculation by Influence Co efficient Method

// BURNING THE FUEL FOR REQUIRED EFFICIENCY
```

```
cout<<"Fl_IJustfuel.Pt = "<<Fl_IJustfuel.Pt<<Fl_Istreamthrust.Pt.units<<endl;
cout<<"Fl_IJustfuel.MN = "<<Fl_IJustfuel.MN<<Fl_Istreamthrust.MN.units<<endl;
cout<<"Fl_IJustfuel.V = "<<Fl_IJustfuel.V<<Fl_Istreamthrust.V.units<<endl;
cout<<"Fl_IJustfuel.Ts = "<<Fl_IJustfuel.Ts<<Fl_Istreamthrust.Ts.units<<endl;
cout<<"Fl_IJustfuel.W = "<<Fl_IJustfuel.W<<Fl_Istreamthrust.W.units<<endl;
cout<<"Fl_IJustfuel.Ps = "<<Fl_IJustfuel.Ps<<Fl_Istreamthrust.Ps.units<<endl;
cout<<"Fl_IJustfuel.gams = "<<Fl_IJustfuel.gams<<Fl_Istreamthrust.gams.units<<endl;
cout<<"Fl_IJustfuel.Cps = "<<Fl_IJustfuel.Cps<<Fl_Istreamthrust.Cps.units<<endl;
cout<<"Fl_IJustfuel.FARst = "<<Fl_IJustfuel.FARst<<Fl_Istreamthrust.FARst.units<<endl;

// Copying Flow Station for Local Use
Fl_Ionedimensional.MN=Fl_Istreamthrust.MN;
Fl_Ionedimensional.copyFlow( "Fl_Istreamthrust" );
// Connecting Local Mass flow rate

fus.LHV = Fu_I.LHV;
fus2.LHV = Fu_I.LHV;
fus2.Wfuel = Fu_I.Wfuel;
// Initialization of the Local Variables
int i;
int length,no;
real Mnew;
real x,X,nc,Tt,A,H,Dh,DTt,fric,xinitial,temp8;
real M3,Mr,T3,M,M4,T4,Ttr;
real theta=5.;
real Wiso,Dh3,Dh4,Tt3,Tt4,A3,A4,P3,P4,P,Pt,towb;
real h,Pprev,Pratio,Pt4;
real k1,k2,sigma,xsp;
real Tterm,Aterm,fterm,BSP;
real temp1,temp2,temp3,temp4,temp5,temp9,temp10;
real dMdx;
real T;
real nctot,Ml;
real Cp;
real AR=4.;
real x3=0.,x4=2.;
length=x4-x3;
real Hiso=0.152;
real gama=1.238,Hcomb;
real fst=0.0674;
real phi=1.0;
real j=0.,dA,combAR=AR_Comb;
real slope;
real temp6,Ttinitial,Ptinitial,Pt3;
xinitial=0.;
h=0.19;//Got by Iteration
real hpr=hPR;
Wiso=Hiso*AR;
Dh3=(2*Hiso*Wiso)/(Hiso+Wiso);
A3=Hiso*Wiso;
```

```
cout<<"A3 = "<<A3<<endl;
Hcomb=Hiso*combAR;
temp9.units="lbm/sec";
A4=Wiso*Hcomb;
cout<<"A4 = "<<A4<<endl;
slope=(A4-A3)/x4;
cout<<"slope"<<slope<<endl;
dA=slope;
//Assigning Values to the Variables
M3=Fl_Ionedimensional.MN;
T3=Fl_Ionedimensional.Ts;
Tt3=Fl_Ionedimensional.Tt;
T3=Fl_Ionedimensional.Ts;
cout<<"Tt3 ="<<Tt3<<endl;
Pt3=Fl_Ionedimensional.Pt;
P3=Fl_Ionedimensional.Ps;
cout<<"P3 ="<<P3<<endl;
Tt4=Fl_Ostreamthrust.Tt;
nctot=nb;
Cp=Cpb;
Cp.units="Btu/(lbm*R)";
P3=Fl_Ionedimensional.Ps;
fric=0.010;//pg 189 heiser book
f=fric;
towb=Tt4/Tt3;
temp9=Fl_IJustfuel.W;
//Functions to solve the RK

real valueK1(real xi,real Mi){
cout<<endl;
x=xi;
M=Mi;
X= (x-xinitial)/length;
//cout<<"X = "<<X<<endl;
nc=(nctot*theta*X)/(1+(theta-1)*X);
fus.Wfuel = Fu_I.Wfuel*nc;
Fl_IJustfuel.burn("fus",nc);
temp9=Fl_IJustfuel.W;
Fu_I.Wfuel=Fu_I.Wfuel-fus.Wfuel;
gama=Fl_IJustfuel.gamt;
Cp=Fl_IJustfuel.Cpt;
Tt=Fl_IJustfuel.Tt;//Tt3*(1+(towb-1)*((theta*X)/(1+(theta-1)*X)));
//cout<<Tt<<endl;
H=Hiso+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
temp8=(length*((X*(theta-1))+1));
DTt=((nctot*hpr*fst*phi)/Cp)*(((4.5/length)/((4*x/length)+1))-
((18*x/length**2)/((4*x/length)+1)**2));//Tt3*(((theta*(towb-1))/temp8)-((theta*X*(theta-
1)*(towb-1))/(length*((X*(theta-1))+1)**2)));
```

```
dA=slope*Hiso*AR;
fric=(4*0.01)/Dh;
sigma=1+((gama-1)/2*M**2);
k1=(M*sigma/(1-M**2))*((-dA/A)+((gama*M**2/2)*fric)+(((1+gama*M**2)/2)*DTt/Tt));
return k1;
}

real valueK2(real xi,real Mi){
x=xi+h;
Mr=Mi+(k1*h);
X= (x-xinitial)/length;
nc=(nctot*theta*X)/(1+(theta-1)*X);
fus.Wfuel = Fu_I.Wfuel*nc;
Fl_IJustfuel.burn("fus",nc);
temp9=Fl_IJustfuel.W;
Fl_IJustfuel.W=temp9-fus.Wfuel;
//cout<<"Remaining Fuel  = "<<Fu_I.Wfuel<<Fu_I.Wfuel.units<<endl;
//cout<<"Fl_IJustfuel.Tt = "<<Fl_IJustfuel.Tt<<Fl_Istreamthrust.Tt.units<<endl;
gama=Fl_IJustfuel.gamt;
Cp=Fl_IJustfuel.Cpt;
Tt=Fl_IJustfuel.Tt;
H=Hiso+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
temp8=(length*((X*(theta-1))+1));
DTt=((nctot*hpr*fst*phi)/Cp)*(((4.5/length)/((4*x/length)+1))-
((18*x/length**2)/((4*x/length)+1)**2));//Tt3*(((theta*(towb-1))/temp8)-((theta*X*(theta-
1)*(towb-1))/(length*((X*(theta-1))+1)**2)));
dA=slope*Hiso*AR;
fric=(4*0.01)/Dh;
sigma=1+((gama-1)/2*Mr**2);
k2=(Mr*sigma/(1-Mr**2))*((-dA/A)+((gama*Mr**2/2)*fric)+(((1+gama*Mr**2)/2)*DTt/Tt));
return k2;
}

for(i=0;x3<=length;i++){
valueK1(x3,M3);
valueK2(x3,M3);
//cout<<"k1 = "<<k1<<endl;
//cout<<"k2 = "<<k2<<endl;
//cout<<"x3 = "<<x3<<endl;
//cout<<"M3 = "<<M3<<endl;
Mnew=M3+(h*(k1/2+k2/2));
T=T3*(Tt/Tt3)*((1+((gama-1)/2*M3**2))/(1+((gama-1)/2*Mnew**2)));
T.units="R";
P=P3*(A3/A)*(Fl_Ionedimensional.MN/Mnew)*(T/T3)**0.5;
Pt=P*(1+((gama-1)/2)*Mnew**2)**(gama/(gama-1));
//cout<<"P = "<<P<<endl;
//cout<<"Pt = "<<Pt<<endl;
x3=x3+h;
```

```
M3=Mnew;
cout<<M3<<endl;
//Fl_IJustfuel.W=Fl_Icomb.W;
//fus.Wfuel=Fu_I.Wfuel;
//Fl_IJustfuel.MN=M3;
//Fl_IJustfuel.W=Fl_IJustfuel.W*temp9;
//Fu_I.Wfuel=Fu_I.Wfuel*temp10;
//Tt=0;
//Fl_IJustfuel.Tt=0;
}
//Fl_IJustfuel.W=Fl_Icomb.W;
//fus.Wfuel=Fu_I.Wfuel;
//Fl_IJustfuel.burn("fus",90);
cout<<"Fl_IJustfuel final.Tt = "<<Fl_IJustfuel.Tt<<Fl_IJustfuel.Tt.units<<endl;
// if loop ends
// Passing on to the Parent Element
M4=M3;
Pt4=Pt;
Tt4=Tt;
Fl_Oonedimensional.MN=M4;
Fl_Oonedimensional.setTotalTP(Tt4,Pt4);
  cout<<"Fl_Oonedimensional.MN =
"<<Fl_Oonedimensional.MN<<Fl_Oonedimensional.MN.units<<endl;
cout<<"Fl_Oonedimensional.Ts =
"<<Fl_Oonedimensional.Ts<<Fl_Oonedimensional.Ts.units<<endl;
//cout<<"Fl_Oonedimensional.Tt =
"<<Fl_Oonedimensional.Tt<<Fl_Oonedimensional.Tt.units<<endl;
cout<<"Fl_Oonedimensional.Pt =
"<<Fl_Oonedimensional.Pt<<Fl_Oonedimensional.Pt.units<<endl;
cout<<"Fl_Oonedimensional.Ps =
"<<Fl_Oonedimensional.Ps<<Fl_Oonedimensional.Ps.units<<endl;
Fl_O.setTotalTP(Tt4,Pt4);
Fl_O.MN=M4;
Fl_O.W=Fl_IJustfuel.W;
//Fl_O.copyFlow("Fl_Oonedimensional");
Fl_IJustfuel2.burn("fus2",0.90);
cout<<"Fl_IJustfuel.Tt looking= "<<Fl_IJustfuel2.Tt<<Fl_IJustfuel2.Tt.units<<endl;

}//Calculate ends

}
```

SCRAMJET BURNER OFF-DESIGN CALCULATION SUBELEMENT

```
extern real Fl_Icomb.Tt;
extern real Fl_Icomb.Pt;
extern real Fl_Icomb.MN;
extern real Fl_Icomb.W;
extern real Pt4,Tt4,M4;
extern real P2,T2,M2,rho2,V2,mu2;
extern real Fu_I.Wfuel;
extern real Fu_I.LHV;
extern int FLUENT;

class BrnOffDesign extends Subelement {

real leftSide, rightSide;
            real X,xsp;
            xsp=0.0;
            real dP;
int i,ramjetmode,scramjetmode;
int length,no;
real Mnew,POS;
real x,nc,Tt,A,H,Dh,DTt,fric,xinitial,temp8;
real M3,Mr,T3,M,M4,T4;
real theta=5.;
real Wiso,Dh3,Dh4,Tt3,Tt4,A3,A4,P3,P4,P,Pt,towb;
real h,Pprev,Pratio,Pt4;
real k1,k2,sigma;
real Tterm,Aterm,fterm,BSP;
real temp1,temp2,temp3,temp4,temp5,temp9,temp10;
real dMdx=0.;
real T,sepdx;
real nctot,Ml;
real Cpb,Cp;
real AR=4.;
real x3=0.,x4=2.;
length=x4-x3;
real Hiso=0.152;
real gama=1.238,Hcomb;
real fst=0.0674;
real phi=1.0;
real j=0.,dA,combAR=3.88;
real slope;
real temp6,Ttinitial,Ptinitial,Pt3;
real Mout;
xinitial=0.;
h=0.1;
real f;
f=fst*phi;
real hpr;
Wiso=Hiso*AR;
Cpb.units="Btu/(lbm*R)";
Cpb=1510.0"J/(kg*K)";
Dh3=(2*Hiso*Wiso)/(Hiso+Wiso);
A3=Hiso*Wiso;
cout<<"A3 = "<<A3<<endl;
Hcomb=Hiso*combAR;
temp9.units="lbm/sec";
```

```
A4=Wiso*Hcomb;
cout<<"A4 = "<<A4<<endl;
slope=(Hcomb-Hiso)/x4;
cout<<"slope"<<slope<<endl;
dA=slope*Wiso;
ramjetmode=0;
scramjetmode=0;
// Initializing Local Flowstation
nctot=0.9;
hpr.units="Btu/lbm";
hpr=43903.250"kJ/kg";
real Pressratio;

// FLOW STATIONS for property calculation according to the fuel burn

  FlowStation Fl_IJustfuel {
               description = "Local flow station for Stream thrust calculation at the
Combustion chamber entry";
                       switchFuelType = "JP";
                       Dissociated = "ON";
                       FAR=0.0674;
                       WAR=0.0;
  }
      FlowStation Fl_IJustfuel2 {
               description = "Local flow station for Stream thrust calculation at the
Combustion chamber entry";
                       switchFuelType = "JP";
                       Dissociated = "ON";
                       FAR=0.0674;
                       WAR=0.0;
  }
  FlowStation Fl_Ionedimensional {
               description = "Local flow station for One Dimensional calculation at the
Combustion chamber entry";
  }
  FlowStation Fl_Oonedimensional {
               description = "Local flow station for One Dimensional calculation at the
Combustion chamber exit";
  }

  // Local Fuel Station
   FuelStation fus;
   FuelStation fus2;
   // Sockets
     Socket TC {
    allowedValues = { "nctot","xsp","Fl_Ionedimensional.Tt"};
    description = "Dry duct pressure loss";
            required = TRUE;

  }
```

```
   Socket DMDX {
  allowedValues = { "temp5","temp4","dMdx"};
  description = "Dry duct pressure loss";
       required = TRUE;

 }

  Socket sjmode {
  allowedValues = { "scramjetmode"};
  description = "Dry duct pressure loss";
       required = FALSE;

//Switch Variable for using Fluent


 }
 // !!!!! *************************************************************************************** !!!!!
// FUNCTIONS SECTION
/*
void preexecute(){

cout<<"I am now here "<<endl;


}
*/
real valuedmdxxsp(real xi){
x=xi;
X= (x-0)/length;
//cout<<"X = "<<X<<endl;
nc=(nctot*theta*X)/(1+(theta-1)*X);
cout<<"nc = "<<nc<<endl;
fus.Wfuel = Fu_I.Wfuel*nc;
//cout<<"fus.Wfuel = "<<fus.Wfuel<<endl;
//cout<<"Fl_IJustfuel.Wfuel = "<<Fl_IJustfuel.W<<endl;
Fl_IJustfuel.burn("fus",nc);
temp9=Fl_IJustfuel.W;
cout<<"temp9 = "<<temp9<<endl;
Fu_I.Wfuel=Fu_I.Wfuel-fus.Wfuel;
//cout<<"Fl_IJustfuel.Tt = "<<Fl_IJustfuel.Tt<<Fl_IJustfuel.Tt.units<<endl;
gama=Fl_IJustfuel.gamt;
Cp=Fl_IJustfuel.Cpt;
Tt=Fl_IJustfuel.Tt;
H=Hiso+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
DTt=((nctot*hpr*fst*phi)/Cpb)*(((4.5/length)/((4*x/length)+1))-
((18*x/length**2)/((4*x/length)+1)**2));
dA=slope*Hiso*AR;
fric=(4*0.01)/Dh;
```

```
//k2=M/(1-M**2)*(1+(gama-1)/2*M**2)*((-
dA/A)+((gama*M**2/2)*fric)+(((1+gama*M**2)/2)*DTt/Tt));
//xsp terms
temp1=((144*x/length**3)/((4*x/length)+1)**3)-((36/length**2)/((4*x/length)+1)**2);
//cout<<"temp 1 = "<<temp1<<endl;
temp2=(((4.5*x/length)/((4*x/length)+1))+Tt3);
//cout<<"temp 2 = "<<temp2<<endl;
//temp3=(((4.5/length)/((4*x/length)+1))-((18*x/length**2)/((4*x/length)+1)**2))**2;
//cout<<"temp 3 = "<<temp3<<endl;
Tterm=((nctot*hpr*fst*phi)/Cp)*((temp1/temp2)-(temp3/temp2**2));
Aterm=-1*((slope*Wiso)**2)/((Hiso+slope*x)*Wiso)**2;
fterm=((4*0.01*slope)/(Wiso*(2*A3+2*slope*x)))-
((8*0.01*slope*(Wiso+((A3+slope*x)/Wiso)))/(2*A3+2*slope*x)**2);
cout<<"Aterm"<<Aterm<<endl;
cout<<"Tterm"<<Tterm<<endl;
cout<<"fterm"<<fterm<<endl;
//Tterm=((nctot*hpr*fst*phi)/Cp)*(((((144*x/length**3)/((4*x/length)+1)**3)-
((36/length**2)/((4*x/length)+1)**2))/(((4.5*x/length)/((4*x/length)+1))+1800))-
((((4.5/length)/((4*x/length)+1))-
((18*x/length**2)/((4*x/length)+1)**2))**2/(((4.5*x/length)/((4*x/length)+1))+1800)**2));
temp4=(gama+1)/(-8)*(-2*Aterm+(gama*fterm)+((1+gama)*Tterm));
//cout<<"temp 4 = "<<temp4<<endl;
return temp4;
}

real valuedmdx(real xi){
x=xi;
X= (x-0)/length;
nc=(nctot*theta*X)/(1+(theta-1)*X);
fus.Wfuel = Fu_I.Wfuel*nc;
//cout<<"fus.Wfuel = "<<fus.Wfuel<<endl;
//cout<<"Fl_IJustfuel.Wfuel = "<<Fl_IJustfuel.W<<endl;
Fl_IJustfuel.burn("fus",nc);
temp9=Fl_IJustfuel.W;
//cout<<"temp9 = "<<temp9<<endl;
Fu_I.Wfuel=Fu_I.Wfuel-fus.Wfuel;
//cout<<"Fl_IJustfuel.Tt = "<<Fl_IJustfuel.Tt<<Fl_IJustfuel.Tt.units<<endl;
gama=Fl_IJustfuel.gamt;
Cp=Fl_IJustfuel.Cpt;
Tt=Fl_IJustfuel.Tt;
H=Hiso+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
DTt=((nctot*hpr*fst*phi)/Cp)*(((4.5/length)/((4*x/length)+1))-
((18*x/length**2)/((4*x/length)+1)**2));
dA=slope*Hiso*AR;
fric=(4*0.01)/Dh;
temp5=((gama+1)*2*gama/(-8))*((4*0.01/Dh)+(DTt/Tt));
//cout<<"temp 5 = "<<temp5<<endl;
return temp5;
```

```
}

real backfirstvalueK2(real xi,real Mi,real ki){
x=xi+h;
k1=ki;
Ml=Mi+(k1*h);
X= (x-xinitial)/length;
nc=(nctot*theta*X)/(1+(theta-1)*X);
fus.Wfuel = Fu_I.Wfuel*nc;
//cout<<"fus.Wfuel = "<<fus.Wfuel<<endl;
//cout<<"Fl_IJustfuel.Wfuel = "<<Fl_IJustfuel.W<<endl;
Fl_IJustfuel.burn("fus",nc);
temp9=Fl_IJustfuel.W;
//cout<<"temp9 = "<<temp9<<endl;
Fu_I.Wfuel=Fu_I.Wfuel-fus.Wfuel;
//cout<<"Fl_IJustfuel.Tt = "<<Fl_IJustfuel.Tt<<Fl_IJustfuel.Tt.units<<endl;
gama=Fl_IJustfuel.gamt;
Cp=Fl_IJustfuel.Cpt;
//cout<<"gama ="<<gama<<endl;
Tt=Fl_IJustfuel.Tt;//Tt3*(1+(towb-1)*((theta*X)/(1+(theta-1)*X)));
H=Hiso+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
DTt=((nctot*hpr*fst*phi)/Cp)*(((4.5/length)/((4*x/length)+1))-
((18*x/length**2)/((4*x/length)+1)**2));
dA=slope*Wiso;
fric=(4*0.01)/Dh;
k2=Ml/(1-Ml**2)*(1+(gama-1)/2*Ml**2)*((-
dA/A)+((gama*Ml**2/2)*fric)+(((1+gama*Ml**2)/2)*DTt/Tt));

return k2;
}
real backvalueK1(real xi,real Mi){
cout<<endl;
x=xi;
M=Mi;
X= (x-xinitial)/length;
//cout<<"X = "<<X<<endl;
nc=(nctot*theta*X)/(1+(theta-1)*X);
fus.Wfuel = Fu_I.Wfuel*nc;
Fl_IJustfuel.burn("fus",nc);
temp9=Fl_IJustfuel.W;
Fu_I.Wfuel=Fu_I.Wfuel-fus.Wfuel;
gama=Fl_IJustfuel.gamt;
Cp=Fl_IJustfuel.Cpt;
Tt=Fl_IJustfuel.Tt;//Tt3*(1+(towb-1)*((theta*X)/(1+(theta-1)*X)));
//cout<<"Tt = "<<Tt<<endl;
H=Hiso+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
```

```
temp8=(length*((X*(theta-1))+1));
DTt=((nctot*hpr*fst*phi)/Cp)*(((4.5/length)/((4*x/length)+1))-
((18*x/length**2)/((4*x/length)+1)**2));//Tt3*(((theta*(towb-1))/temp8)-((theta*X*(theta-
1)*(towb-1))/(length*((X*(theta-1))+1)**2)));
dA=slope*Hiso*AR;
fric=(4*0.01)/Dh;
sigma=1+((gama-1)/2*M**2);
k1=(M*sigma/(1-M**2))*((-dA/A)+((gama*M**2/2)*fric)+(((1+gama*M**2)/2)*DTt/Tt));
return k1;
}

real backvalueK2(real xi,real Mi){
x=xi+h;
Mr=Mi+(k1*h);
X= (x-xinitial)/length;
nc=(nctot*theta*X)/(1+(theta-1)*X);
fus.Wfuel = Fu_I.Wfuel*nc;
Fl_IJustfuel.burn("fus",nc);
temp9=Fl_IJustfuel.W;
Fl_IJustfuel.W=temp9-fus.Wfuel;
//cout<<"Remaining Fuel  = "<<Fu_I.Wfuel<<Fu_I.Wfuel.units<<endl;
//cout<<"Fl_IJustfuel.Tt = "<<Fl_IJustfuel.Tt<<Fl_Istreamthrust.Tt.units<<endl;
gama=Fl_IJustfuel.gamt;
Cp=Fl_IJustfuel.Cpt;
Tt=Fl_IJustfuel.Tt;
H=Hiso+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
temp8=(length*((X*(theta-1))+1));
DTt=((nctot*hpr*fst*phi)/Cp)*(((4.5/length)/((4*x/length)+1))-
((18*x/length**2)/((4*x/length)+1)**2));//Tt3*(((theta*(towb-1))/temp8)-((theta*X*(theta-
1)*(towb-1))/(length*((X*(theta-1))+1)**2)));
dA=slope*Hiso*AR;
fric=(4*0.01)/Dh;
sigma=1+((gama-1)/2*Mr**2);
k2=(Mr*sigma/(1-Mr**2))*((-dA/A)+((gama*Mr**2/2)*fric)+(((1+gama*Mr**2)/2)*DTt/Tt));
return k2;
}

real valueK1(real xi,real Mi){
cout<<endl;
x=xi;
M=Mi;
X= (x-xinitial)/length;
//cout<<"X = "<<X<<endl;
nc=(nctot*theta*X)/(1+(theta-1)*X);
fus.Wfuel = Fu_I.Wfuel*nc;
Fl_IJustfuel.burn("fus",nc);
temp9=Fl_IJustfuel.W;
Fu_I.Wfuel=Fu_I.Wfuel-fus.Wfuel;
```

```
gama=Fl_IJustfuel.gamt;
Cp=Fl_IJustfuel.Cpt;
Tt=Fl_IJustfuel.Tt;//Tt3*(1+(towb-1)*((theta*X)/(1+(theta-1)*X)));
//cout<<"Tt = "<<Tt<<endl;
H=Hiso+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
temp8=(length*((X*(theta-1))+1));
DTt=((nctot*hpr*fst*phi)/Cp)*(((4.5/length)/((4*x/length)+1))-
((18*x/length**2)/((4*x/length)+1)**2));//Tt3*(((theta*(towb-1))/temp8)-((theta*X*(theta-
1)*(towb-1))/(length*((X*(theta-1))+1)**2)));
dA=slope*Hiso*AR;
fric=(4*0.01)/Dh;
sigma=1+((gama-1)/2*M**2);
k1=(M*sigma/(1-M**2))*((-dA/A)+((gama*M**2/2)*fric)+(((1+gama*M**2)/2)*DTt/Tt));
return k1;
}

real valueK2(real xi,real Mi){
x=xi+h;
Mr=Mi+(k1*h);
X= (x-xinitial)/length;
nc=(nctot*theta*X)/(1+(theta-1)*X);
fus.Wfuel = Fu_I.Wfuel*nc;
Fl_IJustfuel.burn("fus",nc);
temp9=Fl_IJustfuel.W;
Fl_IJustfuel.W=temp9-fus.Wfuel;
//cout<<"Remaining Fuel  = "<<Fu_I.Wfuel<<Fu_I.Wfuel.units<<endl;
//cout<<"Fl_IJustfuel.Tt = "<<Fl_IJustfuel.Tt<<Fl_Istreamthrust.Tt.units<<endl;
gama=Fl_IJustfuel.gamt;
Cp=Fl_IJustfuel.Cpt;
Tt=Fl_IJustfuel.Tt;
H=Hiso+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
temp8=(length*((X*(theta-1))+1));
DTt=((nctot*hpr*fst*phi)/Cp)*(((4.5/length)/((4*x/length)+1))-
((18*x/length**2)/((4*x/length)+1)**2));//Tt3*(((theta*(towb-1))/temp8)-((theta*X*(theta-
1)*(towb-1))/(length*((X*(theta-1))+1)**2)));
dA=slope*Hiso*AR;
fric=(4*0.01)/Dh;
sigma=1+((gama-1)/2*Mr**2);
k2=(Mr*sigma/(1-Mr**2))*((-dA/A)+((gama*Mr**2/2)*fric)+(((1+gama*Mr**2)/2)*DTt/Tt));
return k2;
}
real valueKK2(real xi,real Mi,real ki){
x=xi+h;
k1=ki;
X= (x-xinitial)/length;
nc=(nctot*theta*X)/(1+(theta-1)*X);
```

```
fus.Wfuel = Fu_I.Wfuel*nc;
Fl_IJustfuel.burn("fus",nc);
temp9=Fl_IJustfuel.W;
Fl_IJustfuel.W=temp9-fus.Wfuel;
//cout<<"Remaining Fuel  = "<<Fu_I.Wfuel<<Fu_I.Wfuel.units<<endl;
//cout<<"Fl_IJustfuel.Tt = "<<Fl_IJustfuel.Tt<<Fl_Istreamthrust.Tt.units<<endl;
gama=Fl_IJustfuel.gamt;
Cp=Fl_IJustfuel.Cpt;
Tt=Fl_IJustfuel.Tt;
H=Hiso+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
temp8=(length*((X*(theta-1))+1));
DTt=((nctot*hpr*fst*phi)/Cp)*(((4.5/length)/((4*x/length)+1))-
((18*x/length**2)/((4*x/length)+1)**2));//Tt3*(((theta*(towb-1))/temp8)-((theta*X*(theta-
1)*(towb-1))/(length*((X*(theta-1))+1)**2)));
dA=slope*Hiso*AR;
fric=(4*0.01)/Dh;
sigma=1+((gama-1)/2*Mr**2);
k2=(Mr*sigma/(1-Mr**2))*((-dA/A)+((gama*Mr**2/2)*fric)+(((1+gama*Mr**2)/2)*DTt/Tt));
return k2;
}

void fluentoutput(){
real Mach,TP,TT,MFR;
string border1,report,border2,Machoutlet,TPoutlet,TToutlet,MFRoutlet;
string TPunit,TTunit,MFRunit;
InFileStream takein;
takein.open("fluentisoshockout");
cout<<"     OUTPUT FROM ANSYS FLUENT"<<endl;
//takein >> border1;
border1=takein.getline();
report=takein.getline();
border2=takein.getline();
takein >> Machoutlet >> Mach;
takein >> TPoutlet >> TP >> TPunit;
takein >> TToutlet >> TT >> TTunit;
takein >> MFRoutlet >> MFR >> MFRunit;

cout<<  "     "<< border1 << endl;
cout<<  "     "<<report << endl;
cout<<  "     "<< border2 << endl;
cout<<  "Mach Number          "        << Mach <<"         "<< endl;
cout<<  "Total Pressure      "     << TP <<"         "<< TPunit << endl;
cout<<  "Total Temperature   "  << TT <<"         "<< TTunit<< endl;
//cout<<  "Mass Flow Rate      "  << MFR <<"         "<< MFRunit << endl;
// INITIALIZING THE VALUES OF THE NPSS EXIT PARAMETERS WITH FLUENT
OUTPUT

Mout=Mach;
```

```
}

// !!!!! ****************************************************************************************** !!!!!
//CALCULATION BEGINS


void calculate(){
// Initialization of the Local Variables
//cout<<"Fl_Icomb.Tt = "<<Fl_Icomb.Tt<<endl;
//cout<<"Fl_IJustfuel.gamt = "<<Fl_Icomb.Tt<<endl;
Fl_Ionedimensional.W=Fl_Icomb.W;
Fl_Ionedimensional.MN=Fl_Icomb.MN;
Fl_Ionedimensional.setTotalTP(Fl_Icomb.Tt,Fl_Icomb.Pt);
Tt3=Fl_Ionedimensional.Tt;
T3=Fl_Ionedimensional.Ts;
P3=Fl_Ionedimensional.Ps;
M3=Fl_Ionedimensional.MN;
//cout<<"Tt3 = "<<Tt3<<endl;
Fl_IJustfuel.copyFlow( "Fl_Ionedimensional" );
fus.LHV = Fu_I.LHV;
fus.Wfuel = Fu_I.Wfuel;
fus2.LHV = Fu_I.LHV;
fus2.Wfuel = Fu_I.Wfuel;
Fl_IJustfuel2.MN=Fl_Ionedimensional.MN;
Fl_IJustfuel2.copyFlow( "Fl_Ionedimensional");
//Fl_IJustfuel.burn("fus",nctot);
//gama=Fl_IJustfuel.gamt;
//cout<<"Fl_IJustfuel.gamt = "<<gama<<endl;
//Cp=Fl_IJustfuel.Cpt;
//cout<<"Fl_IJustfuel.Cpt = "<<Cp<<endl;
// Local Solver for finding the point of Thermal choking

TC.execute();
cout<<"xsp due to thermal choking = "<<xsp<<endl;

if(xsp>0){
if(xsp<=x4){
ramjetmode=1;
//cout<<"I am here ? "<<endl;
 valuedmdxxsp(xsp);
  valuedmdx(xsp);
  DMDX.execute();
 // cout<<"Value of dmdx = "<<dMdx<<endl;

x=xsp;
M=1.0;
h=-1*h;
Pprev=P3;
for(i=0;x+h>=0;i++){
if(x==xsp){
```

```
ramjetmode=1;
cout<<"Flow has choked thermally in the burner at "<<x<<endl;
cout<<"Performing back integration from the separation point"<<endl;
backfirstvalueK2(x,M,dMdx);
}
cout<<endl;
 if(x<xsp){

cout<<"value of x "<<x<<endl;

if(x>=-h){
backvalueK1(x,M);
backvalueK2(x,M);

}

//cout << "K1 = " << k1 << endl;
//cout << "K2 = " << k2<< endl;
}

//cout<<"M ="<<M<<endl;
Mnew=M+(h*(k1/2+k2/2));
cout<<"x = "<<x<<endl;
cout<<"Mnew"<<Mnew<<endl;
x=x+h;
//cout<<"x = "<<x<<endl;
M=Mnew;
}
}
M3=Mnew;
cout<<"M3 out = "<<Mnew<<endl;
temp6=M3;
cout<<"gama!!!!! ="<<gama<<endl;
cout<<"M3 obtained from back integration = "<<M3<<endl;
h=-1*h;

if(ramjetmode == 1){
cout<<" I am inside "<<endl;
Mnew=M3;
real A3c,H3c,momentumthickness,Retheta,Pressrationormal,Liso,Pisoend;
real V2length,rho2length,dPlength,T2length;
gama=Fl_IJustfuel.gamt;
//Length of the Isolator
cout<<"M2 = "<<M2<<endl;
cout<<"P2 = "<<P2<<endl;
cout<<"M3 = "<<Mnew<<endl;
Pressratio=1+(gama*M2**2)-((gama*M2*Mnew)*((1+((gama-1)/2)*M2**2)/(1+((gama-
1)/2)*Mnew**2))**0.5);
cout<<"Pressure ratio = "<<Pressratio<<endl;
A3c=A3*(1/(gama*Mnew**2))*(((1/Pressratio)*(1+(gama*M2**2)))-1);
```

```
Pisoend=P2*Pressratio;
//V2=Fl_Ionedimensional.V;
//rho2=Fl_Ionedimensional.rhos;
dP=Pisoend-P2;
//Pressratio = Pisoend/P2;
T=Tt/(1+((gama-1)/2)*Mnew**2);
V2length.units="m";
V2length=V2"ft";
rho2length.units="kg/m3";
rho2length=rho2"lbm/ft3";
T2length.units="K";
T2length=T2"R";
dPlength.units="Pa";
dPlength=dP"psia";
cout<<"rho2length = "<<rho2length<<endl;
cout<<"V2length = "<<V2length<<endl;
sepdx=(2*dPlength*Dh3)/(rho2length*0.0040*(V2length**2)*89);
//sepdx=(2*dP*(144.000773/32.17)*(2*Hiso*Wiso*3.28084*3.28084)/((Hiso*3.28084)+(Wi
so*3.28084)))/(rho2*0.010*(V2**2)*89);



cout<<"T 2 = "<<T2<<endl;
Pressrationormal=(((2*gama)/(gama+1))*M2**2)-((gama-1)/(gama+1));
A3c=A3*(1/(gama*Mnew**2))*(((1/Pressrationormal)*(1+(gama*M2**2)))-1);
cout<<"Pressrationormal = "<<Pressrationormal<<endl;
cout<<"Pressratiol = "<<Pressratio<<endl;
H3c=A3c/Wiso;
H3c=H3c;
//Hiso=Hiso*3.28084;
momentumthickness=H3c/1.4;
mu2=(1.458*T2length**(3/2))/((T2+110.4)*10**6);
Retheta=(rho2length*V2length*(momentumthickness))/mu2;
Liso=(Hiso)*(((H3c/(Hiso))**0.5)/(Retheta**0.25))*((50*(Pressrationormal-
1)+170*(Pressrationormal-1)**2)/(M2**2-1));
cout<<"Length of the Isolator required = "<<Liso<<"m"<<endl;

cout<<"A3c = "<<A3c<<endl;
cout<<"dP = "<<dP<<endl;
cout<<"X = "<<X<<endl;
cout<<"x = "<<x<<endl;
cout<<"Pisoend = "<<Pisoend<<endl;
//cout<<"P = "<<P<<endl;
cout<<"Tt = "<<Tt<<endl;
cout<<"Length of separation = "<<sepdx<<" m "<<endl;

if (FLUENT==1){
cout<<"FLUENT FLU"<<endl;
real TinSI,PinSI,ToutSI,PoutSI;
TinSI.units="K";
TinSI=T2"R";
PinSI.units="Pa";
PinSI=P2"psia";
PoutSI.units="Pa";
PoutSI=Pisoend"psia";
ToutSI.units="K";
ToutSI=T"R";
fluentisolatorinput(PinSI,TinSI,M2,PoutSI,ToutSI,Mnew);

}

/*
real A3c,H3c,momentumthickness,Retheta,Pressrationormal,Liso;
dP=P-P2;
sepdx=(2*dP*(144.000773/32.17)*(2*Hiso*Wiso*3.28084*3.28084)/((Hiso*3.28084)+(Wis
o*3.28084)))/(rho2*0.010*(V2**2/25037)*89);
cout<<"Length of separation = "<<sepdx<<sepdx.units<<endl;
//Length of the Isolator

Pressratio=1+(gama*M2**2)-((gama*M2*M3)*((1+((gama-1)/2)*M2**2)/(1+((gama-
1)/2)*M3**2))**0.5);
A3c=A3*(1/(gama*M3**2))*(((1/Pressratio)*(1+(gama*M2**2)))-1);

*/

}

}

h=0.156750;//0.06837380  0.148000;0.154650;0.157250;0.156750;
fluentoutput();
M3=Mout;
for(i=0;x3<length;i++){
if(M3>0) {
//cout<<"Hi = "<<endl;
valueK1(x3,M3);
//cout<<"Value k!!!!1 = "<<k1<<endl;
valueK2(x3,M3);
}
Mnew=M3+(h*(k1/2+k2/2));
//Boundary layer separation criteria Krokegi's condition
BSP=1+0.3*Mnew**2;//for M lesser than 4.5
real BSP2;
//Boundary layer separation criteria Krokegi's condition for turbulent boundary layer

BSP2=0.762;
real Md,Mu;
Mu=M3;
Md=Mu*BSP2;
//cout<<"Value of x = "<<x<<endl;
```

```
//cout<<"Md limit = "<<Md<<endl;
//cout<<" M new = "<<Mnew<<endl;
//cout<<"Boundary Layer seperation = "<<BSP2<<endl;
//cout<<"BSP = "<<BSP<<endl;
T.units="R";
gama=Fl_IJustfuel.gamt;
//cout<<"GAMA USED = "<<gama<<endl;
T=T3*(Tt/Tt3)*((1+((gama-1)/2*Fl_Ionedimensional.MN**2))/(1+((gama-1)/2*Mnew**2)));
//cout<<"T calculated = "<<T << T.units<<endl;
P=P3*(A3/A)*(Fl_Ionedimensional.MN/Mnew)*(T/T3)**0.5;
Pt=P*(1+((gama-1)/2)*Mnew**2)**(gama/(gama-1));
P.units="psia";
Pratio=P/Pprev;

if(Md>BSP2){
//cout<<"Combustion chamber pressure = "<<P<<endl;
//cout<<"Combustion chamber Temperature = "<<T<<endl;
}
if(Pratio>BSP){
cout<<" Separation at xkiya = "<<x<<endl;
POS=x;
//cout<<"Pressure at the exit of the inlet = "<<P2<<endl;
//cout<<"Pressure at Combustion chamber = "<<P<<endl;
//For Scramjet mode
if(ramjetmode==0){
scramjetmode=1;
cout<<"OBLIQUE SHOCK TRAIN "<<endl;
real A3c,H3c,momentumthickness,Retheta,Pressrationormal,Liso,Pisoend;
real V2length,rho2length,dPlength;
gama=1.238;
//gama=Fl_Ionedimensional.gams;
cout<<"value of gama ="<<gama<<endl;
//Length of the Isolator
Pressratio=Pressratio=1+(gama*Fl_Ionedimensional.MN**2)-
((gama*Fl_Ionedimensional.MN*Mnew)*((1+((gama-
1)/2)*Fl_Ionedimensional.MN**2)/(1+((gama-1)/2)*Mnew**2))**0.5);
A3c=A3*(1/(gama*Mnew**2))*(((1/Pressratio)*(1+(gama*M2**2)))-1);
Pisoend=P3*(A3/A3c)*(Fl_Ionedimensional.MN/Mnew)*(T/T3)**0.5;
real Mfinal,Tfinal;
Tfinal=T;
Mfinal=Mnew;
dP=Pisoend-P2;
Pressratio = Pisoend/P2;
T=Tt/(1+((gama-1)/2)*Mnew**2);
V2length.units="m";
V2length=V2"ft";
rho2length.units="kg/m3";
rho2length=rho2"lbm/ft3";
dPlength.units="Pa";
dPlength=dP"psia";
sepdx=(2*dPlength*Dh3)/(rho2length*0.0040*(V2length**2)*89);
//sepdx=(2*dP*(144.000773/32.17)*(2*Hiso*Wiso*3.28084*3.28084)/((Hiso*3.28084)+(Wiso*3.28084)))/(rho2*0.010*(V2**2)*89);
cout<<"T = "<<T<<endl;
cout<<"Pressure ratio = "<<Pressratio<<endl;
cout<<"A3c = "<<A3c<<endl;
cout<<"dP = "<<dP<<endl;
cout<<"X = "<<X<<endl;
cout<<"x = "<<x<<endl;
cout<<"Pisoend = "<<Pisoend<<endl;
cout<<"P = "<<P<<endl;
cout<<"Tt = "<<Tt<<endl;
cout<<"Length of separation = "<<sepdx<<" m "<<endl;


if (FLUENT==1){
real TinSI,PinSI,ToutSI,PoutSI;
TinSI.units="K";
TinSI=Fl_Ionedimensional.Ts"R";
PinSI.units="Pa";
PinSI=Fl_Ionedimensional.Ps"psia";
PoutSI.units="Pa";
PoutSI=P"psia";
ToutSI.units="K";
ToutSI=T"R";
fluentisolatorinput(PinSI,TinSI,Fl_Ionedimensional.MN,PoutSI,ToutSI,Mnew);

}
}
cout<<"RAMJET MODE BEFORE = "<<ramjetmode<<endl;

}

x3=x3+h;
Pprev=P;

M3=Mnew;
cout<<Tt;


}

M4=Mnew;
//
Fl_IJustfuel2.burn("fus2",0.90);
gama=Fl_IJustfuel2.gamt;
P=Pisoend*(A3/A)*(Mfinal/M4)*(T/Tfinal)**0.5;
Pt=P*(1+((gama-1)/2)*M4**2)**(gama/(gama-1));
Tt4=Tt;
Pt4=Pt;
```

```
cout<<"Ramjet mode = "<<ramjetmode<<endl;
cout<<"Scramjet mode = "<<scramjetmode<<endl;
cout<<"M4 = "<<M4<<endl;
//cout<<" Separation at x = "<<POS<<endl;
Fl_O.setTotalTP(Tt4,Pt4);
Fl_O.MN=M4;
Fl_O.W=temp9;

cout<<"Looking for Fl_IJustfuel2.Tt = "<<Fl_IJustfuel2.Tt<<Fl_IJustfuel2.Tt.units<<endl;
}
}

}
```

THERMAL CHOKING POINT SECANT SOLVER SUBELEMENT

```
extern real nctot;
extern real xsp;
extern real Fl_Ionedimensional.Tt;

class ThermalChoking extends Subelement {


real leftSide, rightSide;
real X;
int i;
int length,no;
real Mnew;
real x,nc,Tt,A,H,Dh,DTt,fric,xinitial,temp8;
real M3,Mr,T3,M,M4,T4;
real theta=5.;
real Wiso,Dh3,Dh4,H4,Tt3,Tt4,A3,A4,P3,P4,P,Pt,towb;
real h,Pprev,Pratio,Pt4;
real T;
real nctot;
real Cp;
real AR=4.;
real x3=0.,x4=2.;
length=x4-x3;
real Hiso=0.152;
real gama=1.238,Hcomb;
real fst=0.0674;
real phi=1.0;
real j=0.,dA,combAR=4.34;
real slope;
real temp6,Ttinitial,Ptinitial,Pt3;
xinitial=0.;
h=0.08;
real f;
f=fst*phi;
```

```
real hpr;
H4=combAR*Hiso;
Wiso=Hiso*AR;
Cp=1510.0;
Dh3=(2*Hiso*Wiso)/(Hiso+Wiso);
A3=Hiso*Wiso;
cout<<"A3 = "<<A3<<endl;
Hcomb=Hiso*combAR;
A4=Wiso*Hcomb;
cout<<"A4 = "<<A4<<endl;
slope=(H4-Hiso)/x4;
cout<<"slope"<<slope<<endl;
dA=slope*Wiso;

// Initializing Local Flowstation
nctot=0.9;
hpr=43903250.;
real error;
real fx,fxi,fxj,y,xout;
// Step 1: Create the PsSecant object
X=0.1;
// Step 1: Create the PsSecant object
real valuechokingpoint(real x,real y){
X=x;
fx=(gama*4*0.01/((((Hiso+slope*X))*2*Wiso)/(((Hiso+slope*X))+Wiso)))+(-
dA*2/(Wiso*(Hiso+slope*X)))+((1+gama)*(((hpr*fst*phi*nctot*theta)/(Cp*length*((((X-
x3)/length*(theta-1)))+1)))-((hpr*fst*phi*nctot*theta*(X-x3)*(theta-1))/(Cp*length**2*((((X-
x3)/length*(theta-1)))+1)**2)))/(Tt3+(hpr*fst*phi*((nctot*theta*(X-x3)/length)/(1+(theta-
1)*((X-x3)/length))))/Cp));
X=y;
fxi=(gama*4*0.01/((((Hiso+slope*X))*2*Wiso)/(((Hiso+slope*X))+Wiso)))+(-
dA*2/(Wiso*(Hiso+slope*X)))+((1+gama)*(((hpr*fst*phi*nctot*theta)/(Cp*length*((((X-
x3)/length*(theta-1)))+1)))-((hpr*fst*phi*nctot*theta*(X-x3)*(theta-1))/(Cp*length**2*((((X-
x3)/length*(theta-1)))+1)**2)))/(Tt3+(hpr*fst*phi*((nctot*theta*(X-x3)/length)/(1+(theta-
1)*((X-x3)/length))))/Cp));
xout=y-((fxi*(y-x))/(fxi-fx));
return xout;
}


//initial guess
x=0;
y=0.50;
void calculate()
{
Tt3=Fl_Ionedimensional.Tt*0.555555556;

valuechokingpoint(x,y);
x=xout;
fxj=(gama*4*0.01/((((Hiso+slope*X))*2*Wiso)/(((Hiso+slope*X))+Wiso)))+(-
dA*2/(Wiso*(Hiso+slope*X)))+((1+gama)*(((hpr*fst*phi*nctot*theta)/(Cp*length*((((X-
x3)/length*(theta-1)))+1)))-((hpr*fst*phi*nctot*theta*(X-x3)*(theta-1))/(Cp*length**2*((((X-
x3)/length*(theta-1)))+1)**2)))/(Tt3+(hpr*fst*phi*((nctot*theta*(X-x3)/length)/(1+(theta-
1)*((X-x3)/length))))/Cp));



for(i=0;i<12;i++){
if(fxj>-1){
x=y;
y=xout;
valuechokingpoint(x,y);
X=xout;
fxj=(gama*4*0.01/((((Hiso+slope*X))*2*Wiso)/(((Hiso+slope*X))+Wiso)))+(-
dA*2/(Wiso*(Hiso+slope*X)))+((1+gama)*(((hpr*fst*phi*nctot*theta)/(Cp*length*((((X-
x3)/length*(theta-1)))+1)))-((hpr*fst*phi*nctot*theta*(X-x3)*(theta-1))/(Cp*length**2*((((X-
x3)/length*(theta-1)))+1)**2)))/(Tt3+(hpr*fst*phi*((nctot*theta*(X-x3)/length)/(1+(theta-
1)*((X-x3)/length))))/Cp));
}
}
xsp=xout;
cout<<"xsp out = "<<xsp<<endl;
} // end of self-solving duct element.

}
```

DMDX SECANT SOLVER SUBELEMENT

```
extern real temp4;
extern real temp5;
extern real dMdx;

class dmdx extends Subelement {


real leftSide, rightSide;
real X;
int i;
int length,no;
real Mnew;
real x,nc,Tt,A,H,Dh,DTt,fric,xinitial,temp8;
real M3,Mr,T3,M,M4,T4;
real theta=5.;
real Wiso,Dh3,Dh4,Tt3,Tt4,A3,A4,P3,P4,P,Pt,towb;
real h,Pprev,Pratio,Pt4;
real T;
real nctot;
real Cp;
real AR=4.;
real x3=0.,x4=2.;
length=x4-x3;
real Hiso=0.152;
real gama=1.238,Hcomb;
real fst=0.0674;
real phi=1.0;
real j=0.,dA,combAR=4.34;
real slope;
real temp6,Ttinitial,Ptinitial,Pt3;
xinitial=0.;
h=0.1;
real f;
f=fst*phi;
real hpr;
Wiso=Hiso*AR;
Cp=1510.0;
Dh3=(2*Hiso*Wiso)/(Hiso+Wiso);
A3=Hiso*Wiso;
cout<<"A3 = "<<A3<<endl;
Hcomb=Hiso*combAR;
A4=Wiso*Hcomb;
cout<<"A4 = "<<A4<<endl;
slope=(A4-A3)/x4;
cout<<"slope"<<slope<<endl;
dA=slope;

// Initializing Local Flowstation
nctot=0.9;

real error;
real fx,fxi,fxj,y,xout;
//eq_lhs = "X**2-(temp4+(temp5*X))"; // left hand side of equation
//    eq_rhs = "0";

// Step 1: Create the PsSecant object
real valuedmdxpoint(real x,real y){
X=x;
fx=X**2-(temp4+(temp5*X));
X=y;
fxi=X**2-(temp4+(temp5*X));
xout=y-((fxi*(y-x))/(fxi-fx));
return xout;
}


//initial guess
x=0;
y=0.56;
void calculate()
{
```

```
//cout<<"Value ssdsd"<<endl;
valuedmdxpoint(x,y);
x=xout;
fxj=X**2-(temp4+(temp5*X));



for(i=0;i<35;i++){
if(fxj>-1){
x=y;
y=xout;
valuedmdxpoint(x,y);
X=xout;
fxj=X**2-(temp4+(temp5*X));
}
}
real k;
dMdx=xout;
cout<<"Value of DMDX = "<<dMdx<<endl;
} // end of self-solving duct element.

}
```

SCRAMJET SERN ELEMENT

```
class SJNozzle extends Element {
//-----------------------------------------------------------
  // ****** SETUP PORTS, FLOW STATIONS, SOCKETS, TABLES ********
  //-----------------------------------------------------------
setThermoPackage("allFuel");
   Option switchdes {
    allowedValues = { "DESIGN", "OFFDESIGN" };
    description = "Design switch";
    trigger=FALSE;
    rewritableValues = TRUE;
  }
 //DATA PORT
real COMB_AR=0.0;
  // FLUID PORTS
  FluidInputPort Fl_I {
    description = "Incoming flow";

  }

  FluidOutputPort Fl_O {
    description = "Exiting flow";
```

```
             }
               //Data Port

  DataInputPort PinInlet;
  DataInputPort VinInlet;
  DataInputPort TinInlet;
  DataInputPort COMBAR;
   FlowStation Fl_INozzle {
                 description = "Inlet station to combustion section of burner
 (after the initial pressure loss is applied)";
  }

  FlowStation Fl_ONozzle {
    description = "Exit station to combustion section of burner
 (before thermal storage heat transfer is calculated)";
  }
 Socket OnDesign {
    allowedValues = {
"Fl_INozzle.Tt","Fl_INozzle.Pt","Fl_INozzle.MN","Fl_INozzle.W","PInlet","TInlet","VInlet","
COMB_AR"};
    description = "Dry duct pressure loss";
             required = FALSE;

  }
   Socket OffDesign {
    allowedValues = {
"Fl_INozzle.Tt","Fl_INozzle.Pt","Fl_INozzle.MN","Fl_INozzle.W","PInlet","TInlet","VInlet","
COMB_AR"};
    description = "Dry duct pressure loss";
             required = FALSE;

  }
// TABLES
real PInlet,VInlet,TInlet;
  //-----------------------------------------------------------
  // ******* INTERNAL SOLVER SETUP *******
  //-----------------------------------------------------------

  //-----------------------------------------------------------
  //  ******  ADD SOLVER INDEPENDENTS & DEPENDENTS  ******
  //-----------------------------------------------------------

  //-----------------------------------------------------------
  // ******* VARIABLE CHANGED METHODOLOGY *******
  //-----------------------------------------------------------
  void calculate() {
  real gama;
  PInlet=PinInlet.val;
  TInlet=TinInlet.val;
  VInlet=VinInlet.val;
  Fl_INozzle.MN=Fl_I.MN;
  Fl_INozzle.W=Fl_I.W;
  Fl_INozzle.setTotalTP( Fl_I.Tt, Fl_I.Pt );
  Fl_O.copyFlow( "Fl_I" );
  COMB_AR=COMBAR.val;
  if(switchdes=="DESIGN"){
  OnDesign.execute();
  }
    if(switchdes=="OFFDESIGN"){
  OffDesign.execute();
  }

}//Calculate ends
}
```

SCRAMJET SERN ONDESIGN CALCULATION SUBELEMENT

```
extern real Fl_INozzle.Tt;
extern real Fl_INozzle.Pt;
extern real Fl_INozzle.MN;
extern real Fl_INozzle.W;
extern real PInlet,VInlet,TInlet;
extern real COMB_AR;



class nozzlecalc extends Subelement
{
// FLOW STATIONS for property calculation according to the fuel burn
 FlowStation Fl_Istreamthrust {
                description = "Local flow station for Stream thrust calculation at the
Combustion chamber entry";
                         // = "JP";
                         //Dissociated = "ON";

  }


  FlowStation Fl_Ostreamthrust {
    description = "Local flow station for Stream thrust calculation at the Combustion
chamber exit";
  }

  FlowStation Fl_Ionedimensional {
                description = "Local flow station for One Dimensional calculation at the
Combustion chamber entry";
  }
  FlowStation Fl_Oonedimensional {
```

```
            description = "Local flow station for One Dimensional calculation at the
Combustion chamber entry";

  }

 void calculate(){

real
Tout_Expan,Tout_Comb,AR_Expan,Vout_Expan,Vout_Comb,aout_Expan,Mout_Expan,g
ama_Expan,Cpe,R_Expan,Mcompare;
real ne=0.90;
real f=0.0674;
// Stream Thrust Analysis Calculation
Fl_Istreamthrust.W=Fl_INozzle.W;
Fl_Istreamthrust.MN=Fl_INozzle.MN;
Fl_Istreamthrust.setTotalTP(Fl_INozzle.Tt,Fl_INozzle.Pt);
cout<<"Fl_Istreamthrust.Tt = "<<Fl_Istreamthrust.Tt<<Fl_Istreamthrust.Tt.units<<endl;
cout<<"Fl_Istreamthrust.Pt = "<<Fl_Istreamthrust.Pt<<Fl_Istreamthrust.Pt.units<<endl;
cout<<"Fl_Istreamthrust.MN =
"<<Fl_Istreamthrust.MN<<Fl_Istreamthrust.MN.units<<endl;
cout<<"Fl_Istreamthrust.V = "<<Fl_Istreamthrust.V<<Fl_Istreamthrust.V.units<<endl;
cout<<"Fl_Istreamthrust.Ts = "<<Fl_Istreamthrust.Ts<<Fl_Istreamthrust.Ts.units<<endl;
cout<<"Fl_Istreamthrust.W = "<<Fl_Istreamthrust.W<<Fl_Istreamthrust.W.units<<endl;
 gama_Expan=Fl_Istreamthrust.gamt;
Cpe=Fl_Istreamthrust.Cpt;
R_Expan=Fl_Istreamthrust.Rt;
// Computations Necessary for Expansion Component
// Combustion  Component Station 4 to Station 10
// Assumption
real Pratexittoentry = 1.2;
//Local Initialization
Tout_Comb=Fl_Istreamthrust.Ts;
Vout_Comb=Fl_Istreamthrust.V;
//PR_Comp=10;
Tout_Expan = Tout_Comb*(1-(ne*(1-
(Pratexittoentry*(PInlet/Fl_Istreamthrust.Ps))**(R_Expan/Cpe))));

Vout_Expan = ((Vout_Comb**2/25037)+(2*Cpe*(Tout_Comb-Tout_Expan)))**0.5;
aout_Expan=(gama_Expan*R_Expan*Tout_Expan)**0.5;
Mout_Expan= Vout_Expan/aout_Expan;
//AR_Expan = (1+f)*0.5*(Tout_Expan/TInlet)*(VInlet/Vout_Expan);
//cout<<"AR_Expan = "<<AR_Expan<<endl;
cout<<"M10 = "<<Mout_Expan<<endl;
cout<<"T10 = "<<Tout_Expan<<endl;
//cout<<"V10 = "<<Vout_Expan<<endl;
real R;
R.units="Btu/(lbm*R)";
R=287."J/(kg*K)";
Mcompare=Vout_Expan/(1.4*397.8*R)**0.5;
cout<<"Mcompare = "<<Mcompare<<endl;
```

```
//BY METHOD OF CHARACTERISTICS
real mue4,mue10,HR_Expan,L_Expan,H4,H10;
mue4=asin(1/Fl_Istreamthrust.MN);
mue10=asin(1/Mout_Expan);
HR_Expan=(Fl_Istreamthrust.MN/Mout_Expan)*((1+((gama_Expan-
1)/2*Mout_Expan**2))/(1+((gama_Expan-
1)/2*Fl_Istreamthrust.MN**2)))**((gama_Expan+1)/(2*(gama_Expan-1)));
//cout<<"COMB_AR = "<<COMB_AR<<endl;
H4=COMB_AR*0.152;
H10=HR_Expan*H4;
cout<<"H10 = "<<H10 <<" m"<<endl;
cout<<"H4 = "<<H4 <<" m"<<endl;
L_Expan=0.5*(H10/tan(mue10));
cout<<"L_Expan = "<<L_Expan <<" m"<<endl;

//cout<<"mue4 = "<<mue4<<endl;

//ONE DIMENSIONAL ANALYSIS
Fl_Ionedimensional.W=Fl_Istreamthrust.W;
Fl_Ionedimensional.MN=Fl_Istreamthrust.MN;
Fl_Ionedimensional.setTotalTP(Fl_Istreamthrust.Tt,Fl_Istreamthrust.Pt);

int i,no;
real Mnew,length,gama;
real x,X,nc,Tt,A,H,Dh,fric,xinitial,temp8;
real Mr,T3,M4,T4;
real Wiso,Dh4,M,Tt4,A4,P4,P,Pt;
real h,Pratio,Pt4;
real k1,k2,sigma;
real Tterm,Aterm,A10,fterm,BSP;
real temp1,temp2,temp3,Mi,temp4,temp5,temp9,temp10;
real T;
real nctot,Ml;
real Cp;
real AR=4.;
real x4=0.,x10;
real Hcomb,Hiso;
real dA;
real slope;
real temp6,Ttinitial,Ptinitial,Pt3;
xinitial=0.;
h=0.1;
Hiso=0.152;
Wiso=Hiso*AR;
Cp.units="Btu/(lbm*R)";
Dh4=(2*H4*Wiso)/(H4+Wiso);
A4=H4*Wiso;
A10=H10*Wiso;

x10=L_Expan;
```

```
length=x10-x4;
slope=(H10-H4)/length;

M4=Fl_Ionedimensional.MN;
T4=Fl_Ionedimensional.Ts;
Tt4=Fl_Ionedimensional.Tt;
Pt4=Fl_Ionedimensional.Pt;
P4=Fl_Ionedimensional.Ps;
real valueK1(real xi,real Mi){
cout<<endl;
x=xi;
M=Mi;
X= (x-0)/length;
//cout<<"X = "<<X<<endl;
gama=Fl_Ionedimensional.gamt;
Cp=Fl_Ionedimensional.Cpt;
 //cout<<"value of slope = "<<slope<<endl;
H=H4+slope*x;
//cout<<"H = "<<H<<endl;
A=H*Wiso;
//cout<<"A = "<<A<<endl;
Dh=(H*2*Wiso)/(H+Wiso);
//cout<<"Dh = "<<Dh<<endl;
dA=Wiso;
fric=(4*0.002)/Dh;
sigma=1+((gama-1)/2*M**2);
k1=(M*sigma/(1-M**2))*((-dA/A)+((gama*M**2/2)*fric));
return k1;
}

real valuePK1(real xi,real Mi,real Pi){
cout<<endl;
x=xi;
M=Mi;
P=Pi;
X= (x-0)/length;
//cout<<"X = "<<X<<endl;
gama=Fl_Ionedimensional.gamt;
Cp=Fl_Ionedimensional.Cpt;
 //cout<<"value of slope = "<<slope<<endl;
H=H4+slope*x;
//cout<<"H = "<<H<<endl;
A=H*Wiso;
//cout<<"A = "<<A<<endl;
Dh=(H*2*Wiso)/(H+Wiso);
//cout<<"Dh = "<<Dh<<endl;
dA=Wiso;
fric=(4*0.002)/Dh;
Pk1=-((0.5*gama*M**2)*fric)*P;
return Pk1;
```

```
}

real valueK2(real xi,real Mi){
x=xi+h;
Mr=Mi+(k1*h);
gama=Fl_Ionedimensional.gamt;
Cp=Fl_Ionedimensional.Cpt;
H=H4+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
dA=Wiso;
fric=(4*0.002)/Dh;
sigma=1+((gama-1)/2*Mr**2);
k2=(Mr*sigma/(1-Mr**2))*((-dA/A)+((gama*Mr**2/2)*fric));
return k2;
}

real valuePK2(real xi,real Mi,real Pi){
x=xi+h;
Pr=Pi+(Pk1*h);
gama=Fl_Ionedimensional.gamt;
Cp=Fl_Ionedimensional.Cpt;
H=H4+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
dA=Wiso;
fric=(4*0.002)/Dh;
Pk2=-((0.5*gama*M**2)*fric)*Pr;
return k2;
}
P=Pt4;

for(i=0;x4<=length;i++){
valueK1(x4,M4);
valueK2(x4,M4);
//cout<<"k1 = "<<k1<<endl;
//cout<<"k2 = "<<k2<<endl;
//cout<<"x3 = "<<x<<endl;
//cout<<"M4 = "<<M<<endl;
Mnew=M4+(h*(k1/2+k2/2));
valuePK1(x4,Mnew,P);
valuePK2(x4,Mnew,P);
Pnew=P+(h*(Pk1/2+Pk2/2));
Fl_Ionedimensional.setTotalTP(Fl_Istreamthrust.Tt,Pnew);

T=T4*((1+((gama-1)/2*M4**2))/(1+((gama-1)/2*Mnew**2)));
//<<"T = "<<T<<endl;
//T.units="R";
P=P4*(A4/A)*(Fl_Ionedimensional.MN/Mnew)*(T/T4)**0.5;
```

```
Pt=Fl_Ionedimensional.Pt*(P/Fl_Ionedimensional.Ps)*(Fl_Ionedimensional.Ts/T)**(gama/
(gama-1));
//cout<<"P = "<<P<<endl;
//cout<<"Pt = "<<Pt<<endl;
x4=x4+h;
M4=Mnew;
P=Pnew;
cout<<M4;
//Fl_IJustfuel.W=Fl_Icomb.W;
//fus.Wfuel=Fu_I.Wfuel;
//Fl_IJustfuel.MN=M3;
//Fl_IJustfuel.W=Fl_IJustfuel.W*temp9;
//Fu_I.Wfuel=Fu_I.Wfuel*temp10;
//Tt=0;
//Fl_IJustfuel.Tt=0;
}
//cout<<"Final Mach number = "<<M4<<endl;
Fl_Oonedimensional.MN=M4;
Fl_Oonedimensional.W=Fl_Ionedimensional.W;
Fl_Oonedimensional.setTotalTP(Tt4,Pt);

/*
cout<<"Fl_Oonedimensional.Tt =
"<<Fl_Oonedimensional.Tt<<Fl_Ostreamthrust.Tt.units<<endl;
cout<<"Fl_Oonedimensional.Pt =
"<<Fl_Oonedimensional.Pt<<Fl_Ostreamthrust.Pt.units<<endl;
cout<<"Fl_Oonedimensional.MN =
"<<Fl_Oonedimensional.MN<<Fl_Ostreamthrust.MN.units<<endl;
cout<<"Fl_Oonedimensional.V =
"<<Fl_Oonedimensional.V<<Fl_Ostreamthrust.V.units<<endl;
cout<<"Fl_Oonedimensional.Ts =
"<<Fl_Oonedimensional.Ts<<Fl_Ostreamthrust.Ts.units<<endl;
cout<<"Fl_Oonedimensional.Ps =
"<<Fl_Oonedimensional.Ps<<Fl_Ostreamthrust.Ps.units<<endl;
cout<<"Fl_Oonedimensional.W =
"<<Fl_Oonedimensional.W<<Fl_Ostreamthrust.W.units<<endl;
*/
Fl_O.MN=Fl_Oonedimensional.MN;
Fl_O.W=Fl_Oonedimensional.W;
Fl_O.setTotalTP(Fl_Oonedimensional.Tt,Fl_Oonedimensional.Pt);

  }//calculate ends

 }
```

SCRAMJET SERN OFFDESIGN CALCULATION SUBELEMENT

```
extern real Fl_INozzle.Tt;
extern real Fl_INozzle.Pt;
extern real Fl_INozzle.MN;
extern real Fl_INozzle.W;
extern real PInlet,VInlet,TInlet;
extern real COMB_AR;



class nozzlecalcoffdesign extends Subelement
{
// FLOW STATIONS for property calculation according to the fuel burn
 FlowStation Fl_Istreamthrust {
                description = "Local flow station for Stream thrust calculation at the
Combustion chamber entry";
                         // = "JP";
                         Dissociated = "ON";

  }


  FlowStation Fl_Ostreamthrust {
    description = "Local flow station for Stream thrust calculation at the Combustion
chamber exit";
  }

  FlowStation Fl_Ionedimensional {
                description = "Local flow station for One Dimensional calculation at the
Combustion chamber entry";
  }
  FlowStation Fl_Oonedimensional {
                description = "Local flow station for One Dimensional calculation at the
Combustion chamber entry";

  }
   void calculate(){

// Stream Thrust Analysis Calculation
Fl_Istreamthrust.W=Fl_INozzle.W;
Fl_Istreamthrust.MN=Fl_INozzle.MN;
Fl_Istreamthrust.setTotalTP(Fl_INozzle.Tt,Fl_INozzle.Pt);
cout<<"Fl_Istreamthrust.Tt = "<<Fl_Istreamthrust.Tt<<Fl_Istreamthrust.Tt.units<<endl;
cout<<"Fl_Istreamthrust.Pt = "<<Fl_Istreamthrust.Pt<<Fl_Istreamthrust.Pt.units<<endl;
cout<<"Fl_Istreamthrust.MN =
"<<Fl_Istreamthrust.MN<<Fl_Istreamthrust.MN.units<<endl;
cout<<"Fl_Istreamthrust.V = "<<Fl_Istreamthrust.V<<Fl_Istreamthrust.V.units<<endl;
cout<<"Fl_Istreamthrust.Ts = "<<Fl_Istreamthrust.Ts<<Fl_Istreamthrust.Ts.units<<endl;
cout<<"Fl_Istreamthrust.W = "<<Fl_Istreamthrust.W<<Fl_Istreamthrust.W.units<<endl;
```

```
//ONE DIMENSIONAL ANALYSIS
Fl_Ionedimensional.W=Fl_Istreamthrust.W;
Fl_Ionedimensional.MN=Fl_Istreamthrust.MN;
Fl_Ionedimensional.setTotalTP(Fl_Istreamthrust.Tt,Fl_Istreamthrust.Pt);
 real H4,H10;
int i,no;
real Mnew,Pnew,length,gama;
real x,X,nc,Tt,A,H,Dh,fric,xinitial,temp8;
real Mr,Pr,T3,M4,T4;
real Wiso,Dh4,M,Tt4,A4,P4,P,Pt;
real h,Pratio,Pt4;
real k1,k2,sigma,Pk1,Pk2;
real Tterm,Aterm,A10,fterm,BSP;
real temp1,temp2,temp3,Mi,temp4,temp5,temp9,temp10;
real T;
real nctot,Ml;
real Cp;
real AR=4.;
real x4=0.,x10;
real Hcomb,Hiso;
real dA;
real slope;
real temp6,Ttinitial,Ptinitial,Pt3;
xinitial=0.;
h=0.1;
Hiso=0.152;
Wiso=Hiso*AR;
H4=0.584627;
H10=3.5945;
Cp.units="Btu/(lbm*R)";
Dh4=(2*H4*Wiso)/(H4+Wiso);
A4=H4*Wiso;
A10=H10*Wiso;
real L_Expan=5.43;
x10=L_Expan;
length=x10-x4;
slope=(H10-H4)/length;

M4=Fl_Ionedimensional.MN;
T4=Fl_Ionedimensional.Ts;
Tt4=Fl_Ionedimensional.Tt;
Pt4=Fl_Ionedimensional.Pt;
P4=Fl_Ionedimensional.Ps;

real valueK1(real xi,real Mi){
cout<<endl;
x=xi;
M=Mi;
X= (x-0)/length;
//cout<<"X = "<<X<<endl;
```

```
gama=Fl_Ionedimensional.gamt;
Cp=Fl_Ionedimensional.Cpt;
 //cout<<"value of slope = "<<slope<<endl;
H=H4+slope*x;
//cout<<"H = "<<H<<endl;
A=H*Wiso;
//cout<<"A = "<<A<<endl;
Dh=(H*2*Wiso)/(H+Wiso);
//cout<<"Dh = "<<Dh<<endl;
dA=Wiso;
fric=(4*0.002)/Dh;
sigma=1+((gama-1)/2*M**2);
k1=(M*sigma/(1-M**2))*((-dA/A)+((gama*M**2/2)*fric));
return k1;
}

real valuePK1(real xi,real Mi,real Pi){
cout<<endl;
x=xi;
M=Mi;
P=Pi;
X= (x-0)/length;
//cout<<"X = "<<X<<endl;
gama=Fl_Ionedimensional.gamt;
Cp=Fl_Ionedimensional.Cpt;
 //cout<<"value of slope = "<<slope<<endl;
H=H4+slope*x;
//cout<<"H = "<<H<<endl;
A=H*Wiso;
//cout<<"A = "<<A<<endl;
Dh=(H*2*Wiso)/(H+Wiso);
//cout<<"Dh = "<<Dh<<endl;
dA=Wiso;
fric=(4*0.002)/Dh;
Pk1=-((0.5*gama*M**2)*fric)*P;
return Pk1;
}

real valueK2(real xi,real Mi){
x=xi+h;
Mr=Mi+(k1*h);
gama=Fl_Ionedimensional.gamt;
Cp=Fl_Ionedimensional.Cpt;
H=H4+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
dA=Wiso;
fric=(4*0.002)/Dh;
sigma=1+((gama-1)/2*Mr**2);
k2=(Mr*sigma/(1-Mr**2))*((-dA/A)+((gama*Mr**2/2)*fric));
```

```
return k2;
}

real valuePK2(real xi,real Mi,real Pi){
x=xi+h;
Pr=Pi+(Pk1*h);
gama=Fl_Ionedimensional.gamt;
Cp=Fl_Ionedimensional.Cpt;
H=H4+slope*x;
A=H*Wiso;
Dh=(H*2*Wiso)/(H+Wiso);
dA=Wiso;
fric=(4*0.002)/Dh;
Pk2=-((0.5*gama*M**2)*fric)*Pr;
return k2;
}
P=Pt4;

for(i=0;x4<=length;i++){
valueK1(x4,M4);
valueK2(x4,M4);
//cout<<"k1 = "<<k1<<endl;
//cout<<"k2 = "<<k2<<endl;
//cout<<"x3 = "<<x<<endl;
//cout<<"M4 = "<<M<<endl;
Mnew=M4+(h*(k1/2+k2/2));
valuePK1(x4,Mnew,P);
valuePK2(x4,Mnew,P);
Pnew=P+(h*(Pk1/2+Pk2/2));
Fl_Ionedimensional.setTotalTP(Fl_Istreamthrust.Tt,Pnew);

T=T4*((1+((gama-1)/2*M4**2))/(1+((gama-1)/2*Mnew**2)));
//<<"T = "<<T<<endl;
//T.units="R";
P=P4*(A4/A)*(Fl_Ionedimensional.MN/Mnew)*(T/T4)**0.5;
Pt=Fl_Ionedimensional.Pt*(P/Fl_Ionedimensional.Ps)*(Fl_Ionedimensional.Ts/T)**(gama/
(gama-1));
//cout<<"P = "<<P<<endl;
//cout<<"Pt = "<<Pt<<endl;
x4=x4+h;
M4=Mnew;
P=Pnew;
cout<<M4;
//Fl_IJustfuel.W=Fl_Icomb.W;
//fus.Wfuel=Fu_I.Wfuel;
//Fl_IJustfuel.MN=M3;
//Fl_IJustfuel.W=Fl_IJustfuel.W*temp9;
//Fu_I.Wfuel=Fu_I.Wfuel*temp10;
//Tt=0;
//Fl_IJustfuel.Tt=0;
```

```
}
//cout<<"Final Mach number = "<<M4<<endl;
Fl_Oonedimensional.MN=M4;
Fl_Oonedimensional.W=Fl_Ionedimensional.W;
Fl_Oonedimensional.setTotalTP(Tt4,P);

/*
cout<<"Fl_Oonedimensional.Tt =
"<<Fl_Oonedimensional.Tt<<Fl_Ostreamthrust.Tt.units<<endl;
cout<<"Fl_Oonedimensional.Pt =
"<<Fl_Oonedimensional.Pt<<Fl_Ostreamthrust.Pt.units<<endl;
cout<<"Fl_Oonedimensional.MN =
"<<Fl_Oonedimensional.MN<<Fl_Ostreamthrust.MN.units<<endl;
cout<<"Fl_Oonedimensional.V =
"<<Fl_Oonedimensional.V<<Fl_Ostreamthrust.V.units<<endl;
cout<<"Fl_Oonedimensional.Ts =
"<<Fl_Oonedimensional.Ts<<Fl_Ostreamthrust.Ts.units<<endl;
cout<<"Fl_Oonedimensional.Ps =
"<<Fl_Oonedimensional.Ps<<Fl_Ostreamthrust.Ps.units<<endl;
cout<<"Fl_Oonedimensional.W =
"<<Fl_Oonedimensional.W<<Fl_Ostreamthrust.W.units<<endl;
*/
Fl_O.MN=Fl_Oonedimensional.MN;
Fl_O.W=Fl_Oonedimensional.W;
Fl_O.setTotalTP(Fl_Oonedimensional.Tt,Fl_Oonedimensional.Pt);
real Mcompare,Vout_Expan;
  real R;
R.units="Btu/(lbm*R)";
R=287."J/(kg*K)";

Vout_Expan=Fl_Oonedimensional.MN*(Fl_Oonedimensional.gams*Fl_Oonedimensional.
Rs*Fl_Oonedimensional.Ts)**0.5;
  Mcompare=Vout_Expan/(1.4*397.8*R)**0.5;
  cout<<"Mcompare = "<<Mcompare<<endl;
real Sa10;
Sa10=Fl_Oonedimensional.V*(1+(25.037*Fl_Oonedimensional.Rs*Fl_Oonedimensional.
Ts/(Fl_Oonedimensional.V**2)));
cout<<"Sa 10 = "<<Sa10<<endl;
   }//calculate ends
 }
```

ADDITIONAL FUNCTIONS SECTION

```
//-----------------------------------------------------------------------
//   Input MACROS of Fluent
//-----------------------------------------------------------------------
void fluentinput(real a,real b,real c) {
OutFileStream diag{
            // Name of the output file that is created by this out stream
            filename = "fluent_input.c";
}
// To include the header file

diag<<"#include "<<"<udf.h>"<<endl;

// STATIC PRESSURE PROFILE OF INLET AND FREESTREAM

diag<<"DEFINE_PROFILE(staticpressure_freestream,t,i)"<<endl;
diag<<"{"<<endl;
diag<<"real x[ND_ND];"<<endl;
diag<<"real y;"<<endl;
diag<<"face_t f;"<<endl;
diag<<"begin_f_loop(f,t)"<<endl;
diag<<"{"<<endl;
diag<<"F_CENTROID(x,f,t);"<<endl;
diag<<"y=x[1];"<<endl;
diag<<"F_PROFILE(f,t,i)="<<a<<"+(y-y);"<<endl;
diag<<"}"<<endl;
diag<<"end_f_loop(f,t)"<<endl;
diag<<"}"<<endl;
cout << endl;

// STATIC TEMPERATURE PROFILE OF INLET AND FREESTREAM
diag<<"DEFINE_PROFILE(statictemperature_freestream,t,i)"<<endl;
diag<<"{"<<endl;
diag<<"real x[ND_ND];"<<endl;
diag<<"real y;"<<endl;
diag<<"face_t f;"<<endl;
diag<<"begin_f_loop(f,t)"<<endl;
diag<<"{"<<endl;
diag<<"F_CENTROID(x,f,t);"<<endl;
diag<<"y=x[1];"<<endl;
diag<<"F_PROFILE(f,t,i)="<<b<<"+(y-y);"<<endl;
diag<<"}"<<endl;
diag<<"end_f_loop(f,t)"<<endl;
diag<<"}"<<endl;
cout << endl;

// MACH NUMBER PROFILE OF INLET AND FREESTREAM

diag<<"DEFINE_PROFILE(Machnumber_freestream,t,i)"<<endl;
diag<<"{"<<endl;
diag<<"real x[ND_ND];"<<endl;
diag<<"real y;"<<endl;
diag<<"face_t f;"<<endl;
diag<<"begin_f_loop(f,t)"<<endl;
diag<<"{"<<endl;
diag<<"F_CENTROID(x,f,t);"<<endl;
diag<<"y=x[1];"<<endl;
diag<<"F_PROFILE(f,t,i)="<<c<<"+(y-y);"<<endl;
diag<<"}"<<endl;
diag<<"end_f_loop(f,t)"<<endl;
diag<<"}"<<endl;
cout << endl;



diag.close();
}

//
void fluentisolatorinput(real a,real b,real c,real d,real e,real f) {
OutFileStream diag{
        // Name of the output file that is created by this out stream
        filename = "fluent_isolatorinput.c";
}
// To include the header file

diag<<"#include "<<"<udf.h>"<<endl;

// STATIC PRESSURE PROFILE OF INLET AND FREESTREAM

diag<<"DEFINE_PROFILE(staticpressure_IsoIn,t,i)"<<endl;
diag<<"{"<<endl;
diag<<"real x[ND_ND];"<<endl;
diag<<"real y;"<<endl;
diag<<"face_t f;"<<endl;
diag<<"begin_f_loop(f,t)"<<endl;
diag<<"{"<<endl;
diag<<"F_CENTROID(x,f,t);"<<endl;
diag<<"y=x[1];"<<endl;
diag<<"F_PROFILE(f,t,i)="<<a<<"+(y-y);"<<endl;
diag<<"}"<<endl;
diag<<"end_f_loop(f,t)"<<endl;
diag<<"}"<<endl;
cout << endl;

// STATIC TEMPERATURE PROFILE OF INLET AND FREESTREAM
diag<<"DEFINE_PROFILE(statictemperature_IsoIn,t,i)"<<endl;
diag<<"{"<<endl;
diag<<"real x[ND_ND];"<<endl;
diag<<"real y;"<<endl;
diag<<"face_t f;"<<endl;
diag<<"begin_f_loop(f,t)"<<endl;
diag<<"{"<<endl;
diag<<"F_CENTROID(x,f,t);"<<endl;
diag<<"y=x[1];"<<endl;
diag<<"F_PROFILE(f,t,i)="<<b<<"+(y-y);"<<endl;
diag<<"}"<<endl;
```

```
diag<<"end_f_loop(f,t)"<<endl;
diag<<"}"<<endl;
cout << endl;

// MACH NUMBER PROFILE OF INLET AND FREESTREAM

diag<<"DEFINE_PROFILE(Machnumber_IsoIn,t,i)"<<endl;
diag<<"{"<<endl;
diag<<"real x[ND_ND];"<<endl;
diag<<"real y;"<<endl;
diag<<"face_t f;"<<endl;
diag<<"begin_f_loop(f,t)"<<endl;
diag<<"{"<<endl;
diag<<"F_CENTROID(x,f,t);"<<endl;
diag<<"y=x[1];"<<endl;
diag<<"F_PROFILE(f,t,i)="<<c<<"+(y-y);"<<endl;
diag<<"}"<<endl;
diag<<"end_f_loop(f,t)"<<endl;
diag<<"}"<<endl;
cout << endl;
//Conditions for isolator out
diag<<"DEFINE_PROFILE(staticpressure_IsoOut,t,i)"<<endl;
diag<<"{"<<endl;
diag<<"real x[ND_ND];"<<endl;
diag<<"real y;"<<endl;
diag<<"face_t f;"<<endl;
diag<<"begin_f_loop(f,t)"<<endl;
diag<<"{"<<endl;
diag<<"F_CENTROID(x,f,t);"<<endl;
diag<<"y=x[1];"<<endl;
diag<<"F_PROFILE(f,t,i)="<<d<<"+(y-y);"<<endl;
diag<<"}"<<endl;
diag<<"end_f_loop(f,t)"<<endl;
diag<<"}"<<endl;
cout << endl;

// STATIC TEMPERATURE PROFILE OF INLET AND FREESTREAM
diag<<"DEFINE_PROFILE(statictemperature_IsoOut,t,i)"<<endl;
diag<<"{"<<endl;
diag<<"real x[ND_ND];"<<endl;
diag<<"real y;"<<endl;
diag<<"face_t f;"<<endl;
diag<<"begin_f_loop(f,t)"<<endl;
diag<<"{"<<endl;
diag<<"F_CENTROID(x,f,t);"<<endl;
diag<<"y=x[1];"<<endl;
diag<<"F_PROFILE(f,t,i)="<<e<<"+(y-y);"<<endl;
diag<<"}"<<endl;
diag<<"end_f_loop(f,t)"<<endl;
diag<<"}"<<endl;
```

```
cout << endl;

// MACH NUMBER PROFILE OF INLET AND FREESTREAM

diag<<"DEFINE_PROFILE(Machnumber_IsoOut,t,i)"<<endl;
diag<<"{"<<endl;
diag<<"real x[ND_ND];"<<endl;
diag<<"real y;"<<endl;
diag<<"face_t f;"<<endl;
diag<<"begin_f_loop(f,t)"<<endl;
diag<<"{"<<endl;
diag<<"F_CENTROID(x,f,t);"<<endl;
diag<<"y=x[1];"<<endl;
diag<<"F_PROFILE(f,t,i)="<<f<<"+(y-y);"<<endl;
diag<<"}"<<endl;
diag<<"end_f_loop(f,t)"<<endl;
diag<<"}"<<endl;
cout << endl;


diag.close();
}

// TO GET VALUES FROM FLUENT OUTPUT FILE
// ALL THE PROFILE VALUES ARE AVERAGED

/*
void fluentoutput(){
real Mach,TP,TT,MFR;
string border1,report,border2,Machoutlet,TPoutlet,TToutlet,MFRoutlet;
string TPunit,TTunit,MFRunit;
InFileStream takein;
takein.open("fluoutput3");
cout<<"     OUTPUT FROM ANSYS FLUENT"<<endl;
//takein >> border1;
border1=takein.getline();
report=takein.getline();
border2=takein.getline();
takein >> Machoutlet >> Mach;
takein >> TPoutlet >> TP >> TPunit;
takein >> TToutlet >> TT >> TTunit;
takein >> MFRoutlet >> MFR >> MFRunit;

cout<<  "     "<< border1 << endl;
cout<<  "     "<<report << endl;
cout<<  "     "<< border2 << endl;
cout<<  "Mach Number         "        << Mach <<"         "<< endl;
cout<<  "Total Pressure      "     << TP <<"         "<< TPunit << endl;
cout<<  "Total Temperature   "  << TT <<"         "<< TTunit<< endl;
cout<<  "Mass Flow Rate      "  << MFR <<"         "<< MFRunit << endl;
```

```
// INITIALIZING THE VALUES OF THE NPSS EXIT PARAMETERS WITH FLUENT
OUTPUT
myin.Mout=Mach;
//setting the units to be converted while feeding in
myin.TtOut.units="R";
myin.PtOut.units="psia";
myin.Win.units="lbm/sec";

myin.TtOut=TT"K";
myin.PtOut=TP"Pa";
myin.Win=MFR "kg/sec";
}

*/
// TO Calculate the COntraction Limits
void KANTROWITZLIMIT(real Mout,real gama){
cout << endl;
cout << "=========================================== \n";
cout << "==  CONTRACTION RATIO LIMITS == \n";
cout << "=========================================== \n";
cout << endl;
real KL;
real EmpLmt;
real a,b,c;
a=(1/Mout)*((((gama+1)*Mout**2)/((gama-1)*Mout**2+2))**(gama/(gama-1)));
b=(((gama+1)/((2*gama*Mout**2)-(gama-1)))**(1/(gama-1)));
c=((1+(((gama-1)/2)*Mout**2))/((gama+1)/2))**((gama+1)/(2*(gama-1)));
KL=a*b*c;
cout << endl;
cout << "KANTROWITZ LIMIT {INTERIOR CONTRACTION RATIO} ,[A2/A4]= "<< KL;
cout << endl;
cout << "Area of throat/Area at Cowl Lip = "<< 1/KL;
cout << endl;
}

void MAXIMUMCR(real M){
real MCR;
MCR=(0.05-(0.52/M)+(3.65/(M**2)));
cout << "Maximum contraction Ratio { For Free Stream Mach no of  "<< M<<"} =  "<<
MCR;
}
// To calculate Boundary Layer Seperation

real KorkegiBLSP(real M){
real BLSP;
if (M >= 4.5)
{
BLSP=0.17*(M**2.5);
}
if (M <= 4.5)
```

```
{
BLSP=1+(0.3*(M**2));
}
return BLSP;
}

real TubulenceIntensity(real rho,real V,real mu){
real Re,I;
Re=(rho*V*Dh)/mu;
I=0.16*Re**(-1/8);
return I;
}
```

References

[1] Seigal, C., "The Scramjet Engine: Processes and Characteristics," Cambridge University Press, 2009.

[2] Heiser, W. H., Pratt, D. H., “Hypersonic Air Breathing Propulsion,” Washington, D.C.: American Institute of Aeronautics and Astronautics, 1994.

[3] Anderson, D. J., “Hypersonic and High Temperature Gas Dynamics,” American Institute of Aeronautics and Astronautics Education Series, 1994.

[4] Milligan, T. R., “Dual Mode: A Computational Investigation of Combustor Design and Operation,” M S Thesis, Wright State University, 2009.

[5] Fry, R. S., "A Century of Ramjet Propulsion Technology Evolution,” Journal of Propulsion and Power, Volume 20, No. 1, January-February 2004.

[6] Hall, Edward,. "Energy Efficient Engine Low Pressure Subsystem Aerodynamic Analysis," 34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, 1998.

[7] Evans, A.L., Follen, G., Nainman,C. ,Lopez,J. , “Numerical Propulsion System Simulation’s National Cycle Program,” NASA Lewis Research Center , Cleveland ,Ohio, AIAA Paper 1998-3113,1998.

[8] Russell, W. C., Thomas, L., Scott, T., and Mark, T., “Coupled Higher Fidelity Engine System,” 26th International Congress of the Aeronautical Sciences. 2008.

[9] Waltrup, P. J., White, M. E., Zarlingo, F., and Edward, S. G., "History of Ramjet and Scramjet Propulsion Development for U.S. Navy Missiles," Johns Hopkins APL Technical Digest, Volume 18, 1997.

[10] NASA Dryden Flight Research Center, PAO, “NASA’s X-43A Proves Hypersonic Scramjet Flight,” (http://www.nasa.gov/missions/research/x43_suc.html) March27, 2004.

[11] Dean, A. ,”Scramjet Engines Enabling the Seamless Integration of Air & Space Operations,” Pratt & Whitney Space Propulsion, Hypersonics, West Palm Beach, FL, 33410-9600.

[12] Kandebo, S.W., “USAF/Pratt Scramjet Breakthrough Landmark Tests Boost Scramjet’s Future”, Aviation Week & Space Technology, March 2001, page 58. New York.

[13] Kandebo, S.W., “Exploring the Envelop”, Aviation Week and Space Technology, June2, 2003, page22, Wright-Patterson AFB, Ohio.

[14] Kandebo,S. W., “New Power plant Key to Missile Demonstrator”, Aviation Week and Space Technology, September 2, 2002, page 56, New York and Washington.

[15] Guy, N., Los Angles, “Next generation Aircraft: Beyond Blackbird”, Aviation Week and Space Technology, November 4, 2013, page 18.

[16] Scott, M. J., “Steady State Modeling of Gas Turbine Engines Using the Numerical Propulsion System Simulation Code,” Proceedings of ASME Turbo Expo 2010: Power for Land, Sea and Air GT2010, June 14-18, 2010, Glasgow, UK.

[17] Evans, A.L., Follen, G., Nainman,C. ,Lopez,J. , “Numerical Propulsion System Simulation’s National Cycle program,” NASA Lewis Research Center ,Cleveland ,Ohio, AIAA Paper 1998-3113,1998.

[18] Lytle, J. K., “Multi-fidelity Simulations of Air Breathing Propulsion Systems” 42nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit 9 - 12 July 2006, Sacramento, California.

[19] Lytle, J. K., “The Numerical Propulsion System Simulation: An Overview”, Glenn Research Center, Cleveland, Ohio, NASA/TM—2000-209915.

[20] Mark, G. T., “High Fidelity 3D Simulation of the GE90”, 33rd AIAA Fluid Dynamics Conference and Exhibit, 23-26 June 2003, Orlando, Florida

[21] G. Follen Glenn Research Center, Cleveland, Ohio “Numerical Zooming Between a NPSS Engine System Simulation and a One-Dimensional High Compressor Analysis Code”

[22] Anderson, J. D., “Computational Fluid Dynamics: The Basics with Applications”, McGraw-Hill Series in Aeronautical and Aerospace Engineering,1995.

[23] Smart, M. K., “Scramjet Inlets”, NATO Science and Technology Organization .RTO-EN-AVT-185, University of Queensland Brisbane.

[24] Bussey, G. H., “Application of Compound Compressible Flow to Hypersonic Three Dimensional Inlets,” Master of Science Thesis, University of Maryland ,College Park, 2009.

[25] Roberts,K.,N.,“ Analysis and Design of a Hypersonic Scramjet Engine with a Starting Mach Number of 4.00 ” , Master of Science Thesis ,University of Texas at Arlington, August 2008.

[26] Molina, F. J. M., “Numerical Study of Innovative Scramjet Inlets Coupled to a Combustion Chamber Using Hydrocarbon Mixture”, Doctor of Philosophy Thesis, School of Aerospace Engineering, Georgia Institute of Technology, May 2010.

[27] Lewis, M. J. , “Significance of Fuel Selection for Hypersonic Vehicle Range”, Journal of Propulsion and Power, vol. 17, no. 6, 2001, pp. 1214-1221.

[28] National Research Council, “Evaluation of the National Aerospace Initiative”, Committee on the National Aerospace Initiative, Air Force Science and Technology Board, Division on Engineering and Physical Sciences,. National Academies Press, Jun 17, 2004.

[29] Curran, E.T., and S.N.B. Murthy. Scramjet Propulsion. Vol. 189: Progress in Astronautics and Aeronautics. Reston, Virginia: American Institute of Aeronautics and Astronautics, Inc., 2000.

[30] Curran, E.T., Craig, R.R. , " The Use of Stream Thrust Concepts for the Approximate Evaluation of Hypersonic Ramjet Engine Performance ", Air force Aero Propulsion Laboratory , Wright Patterson Air force Base , Ohio, AFAPL – TR – 73 - 38, July 1973.

[31] Tran, K., "One Dimensional Analysis Program for Scramjet and Ramjet Flow paths ", Master of Science Thesis, Virginia Polytechnic Institute and State University,2010.

[32] Shapiro, A. H., "The Dynamics and Thermodynamics of Compressible Flow, "Ronald Press, New York 1953.

[33] West, N., Papanicolaou, G., Glynn, P., Laccarino, G., "A Numerical Study of Filtering and Control for Scramjet Engine Flow", 20th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, 27 - 30 June 2011.

[34] Smart, M. K., "Scramjet Isolators", NATO Science and Technology Organization .RTO-EN-AVT-185, University of Queensland Brisbane.

[35] Allen, J. B., Hauser, T., Tam, C. J., "Numerical Simulation of a Scramjet Isolator using RANS and LES Approaches" 45th AIAA Aerospace Sciences Meeting and Exhibit,Reno,Nevada, AIAA Paper 2007-115,2007.

[36] "ANSYS FLUENT Theory Guide 14.5" , Ansys,Inc.Canonsburg,PA 15317.October 2012.

[37] "ANSYS CFX Theory Guide 14.5", Ansys, Inc.Canonsburg,PA 15317. October 2012.

[38] Sinha, K., Mahesh, K., Candler, G.,V., "Modelling Shock – Unsteadiness in Shock/Turbulence Interaction " Physics of Fluids, Vol. 15,No.8, Pg. 2290-2297, 2003.

[39] Baurle, R. A., Middleton, T. F., Wilson ,L. G., " Reynolds Averaged Turbulence Model Assessment of a Highly Back Pressured Isolator Flow Field ",Spacecraft Propulsion And Power,JANNAF 45th Combustion Meeting Joint Subcommittee Meeting ,Monterey,CA,2012, Doc ID 20130000726.

[40] Pasha, A. A., “Application of Shock-Unsteadiness Model to Hypersonic Shock/turbulent boundary-layer Interaction,” Doctoral Dissertation, IIT Bombay 2011.

[41] Korkegi, R.H., “A Simple correlation for incipient turbulent boundary layer separation due to a skewed shock wave ,” AIAA Journal,Vol 11,No 11 , pp 1578 – 1579, 1973.

[42] “NPSS User Guide”, Numerical Propulsion System Simulation Consortium, Wolverine Ventures Headquarters, FL 33478 June 21, 2012.

[43] “NPSS Thermodynamic Package Guide”, Numerical Propulsion System Simulation Consortium, Wolverine Ventures Headquarters, FL 33478 June 21, 2012.

[44] “NPSS Component Comments”, Numerical Propulsion System Simulation Consortium, Wolverine Ventures Headquarters, FL 33478 June 21, 2012.

[45] “NPSS Slides”, Numerical Propulsion System Simulation Consortium, Wolverine Ventures Headquarters, FL 33478 June 21, 2012.

[46] Anderson, J. D.,”Modern Compressible Flow: With Historical Perspective”, McGraw Hill. 2002.

[47] Hodge, B. K., and Koenig, K., “Compressible Fluid Dynamics with Personal Computed Applications”, Prentice Hall, 1995.

[48] Autar Kaw, “Runge-Kutta 2nd Order Method for Ordinary Differential Equations”, University of South Florida: Holistic Numerical Methods Institute (http://www.saylor.org/site/wp-content/uploads/2011/11/ME205-8.3-TEXT.pdf)

[49] Pulsonetti, M. V., Erdos, J., and Early, K., “An Engineering Model for Analysis of Scramjet Combustor Performance with Finite Rate Chemistry”, AIAA 28th Joint Propulsion Conference, AIAA Paper 88-3258, Boston, 1988.

[50] Bachchan, N. , and Hilliery, R., “Hypersonic Inlet Flow Analysis at Off-Design Conditions”, 22nd Applied Aerodynamics Conference and Exhibit 16 - 19 August 2004, Providence, Rhode Island, AIAA 2004-5380.

[51] Deree, K. A., Asbury, S. C., "Experimental and Computational investigation of a translating throat Single Expansion Nozzle", NASA/TP-1999-209138, Hampton, Virginia, 1999.

[52] "ANSYS UDF Guide 14.5" , Ansys,Inc.Canonsburg,PA 15317.October 2012.

[53] Lawrence, C., "An Overview of Three Approaches to Multidisciplinary Aeropropulsion Simulations", NASA Technical Memorandum 107443, NASA Lewis Research Center, Cleveland, Ohio 44135.

[54] Baurle,R.A.,Gaffney,R.L.,"The Art of Extracting One Dimensional Flow Properties from Multi-Dimensional Data Sets" ,45th Aerospace Science Meeting and Exhibit,Reno,Nevada ,AIAA Paper 2007 – 639.2007

[55] Torrez, S.M., Scholten, N. A., Micka, D. J., Driscoll, J.F., Bolender, M. A., Doman. D. B., Oppenheimer, M. W., "A Scramjet Engine Model Including Effects of Precombustion Shocks and Dissociation", 44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit 21 - 23 July 2008, Hartford, CT.

[56] Ortwerth, P.J., "Scramjet Vehicle Integration", Scramjet Propulsion, Progress in Astronautics and Aeronautics, AIAA Washington DC, Chapter 17, 2001.

[57] Waltrup, P.J. and Billig, F.S., "Structure of shock waves in cylindrical ducts", AIAA journal, Vol .11, No. 10, 1973, pp 1404-1408.

[58]Tourani, C., "Computational Simulation of Scramjet Combustors – A Comparison between Quasi-One Dimensional and 2-D Numerical Simulations", Master of Science Thesis, University Of Kansas,2011.

[59] "ANSYS FLUENT user Guide 14.5", Ansys, Inc.Canonsburg,PA 15317.October 2012.

[60] Mattingly, J. M., "Elements of Gas Turbine Propulsion", AIAA education series,2005.

[61] Carroll, P.J., Dutton, J.C., “Characteristics of Multiple Shockwaves/Turbulent Boundary Layer Interactions in a Rectangular Ducts”, Journal of Propulsion and Power, Vol.6, No.2.pp 186-193.1990

[62] Baurle,R.A.,Gaffney,R.L.,“The Art of Extracting One Dimensional Flow Properties from Multi-Dimensional Data Sets” ,45th Aerospace Science Meeting and Exhibit,Reno,Nevada ,AIAA Paper 2007 – 639.2007

[63] Billig, F. S., Corda, S., Stockbridge, R. D., “Combustor-Inlet Interaction in Scramjet Engines,” APL Technical Review, Vol. 2, No. 1, Nov. 1990, pp. 118-126.

[64] Sullins, G., McLafferty, G., “Experimental Results of Shock Trains in Rectangular Ducts,” AIAA Paper 92-5103, Orlando, FL, Dec. 1-4, 1992.

[65] Moses, P. L., “X-43C Plans and Status, 12th AIAA International Space Planes and

[66] Gamble, H. J., Haid, D., Alessandro, D., S., Defrancesco, R., “Dual-Mode Scramjet Performance Model for TBCC Simulation,” 45th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, Colorado 2009, AIAA 2009 – 5298.

[67]Hays,P.R.,“History of the Talos Ramjet Engine,” http://www.okieboat.com/Ramjet%20history.html, LT USNR-R.

[68] Maruta and Nakamura Lab, “Researches - Supersonic combustion,” Institute of Fluid Science,TohokuUniversity,http://www.ifs.tohoku.ac.jp/enerdyn/en/en_researches_supersonic.htm.

Biographical Information

Nandakumar Vijayakumar earned his Bachelor of Engineering degree in Aeronautical Engineering from Hindustan University, India in 2012. During his senior year, he was a research intern at aerodynamics and propulsion unit of Hindustan Aeronautics Limited, India, where his research study focused on computational fluid dynamics and heat transfer simulation of single engine helicopter propulsion system. He was influenced and inspired by his rocket propulsion professor; Dr.K.Ramamurthi which led him to pursue his Master of Science at University of Texas, Arlington. Due to his interest in high speed gas dynamics and detonation propulsion he joined the Aerodynamic Research Center and began his research on hypersonic vehicles.

He is an active member of AIAA High Speed Air Breathing Propulsion Technical Committee, Royal Aeronautical Society, American Physical Society and National Space Society. After graduating his Master of Science degree, he plans to continue with his research on high speed propulsion for obtaining his Doctoral degree at UT Arlington.