

NUMERICAL FLOW FIELD ANALYSIS OF AN AIR AUGMENTED ROCKET USING THE  
AXISYMMETRIC METHOD OF CHARACTERISTICS

A Thesis  
presented to  
the Faculty of California Polytechnic State University,  
San Luis Obispo

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science in Aerospace Engineering

by  
Jeffrey Alan Massman  
November, 2013

© 2013

Jeffrey Alan Massman  
ALL RIGHTS RESERVED

## COMMITTEE MEMBERSHIP

TITLE: Numerical Flow Field Analysis of an Air Augmented Rocket Using the Axisymmetric Method of Characteristics

AUTHOR: Jeffrey Alan Massman

DATE SUBMITTED: November 2013

COMMITTEE CHAIR: Dr. Dianne DeTurris, Professor  
Aerospace Engineering Department  
California Polytechnic State University, San Luis Obispo

COMMITTEE MEMBER: Dr. David Marshall, Professor  
Aerospace Engineering Department  
California Polytechnic State University, San Luis Obispo

COMMITTEE MEMBER: Dr. Kim Shollenberger, Professor  
Mechanical Engineering Department  
California Polytechnic State University, San Luis Obispo

COMMITTEE MEMBER: Kyle Johnson, Industry Advisor  
Boeing  
Everett, Washington

## ABSTRACT

### Numerical Flow Field Analysis of an Air Augmented Rocket Using the Axisymmetric Method of Characteristics

Jeffrey Alan Massman

An Axisymmetric Rocket Ejector Simulation (ARES) was developed to numerically analyze various configurations of an air augmented rocket. Primary and secondary flow field visualizations are presented and performance predictions are tabulated. A parametric study on ejector geometry is obtained following a validation of the flow fields and performance values.

The primary flow is calculated using a quasi-2D, irrotational Method of Characteristics and the secondary flow is found using isentropic relations. Primary calculations begin at the throat and extend through the nozzle to the location of the first Mach Disk. Combustion properties are tabulated before analysis to allow for propellant property selection. Secondary flow calculations employ the previously calculated plume boundary and ejector geometry to form an isentropic solution. Primary and secondary flow computations are iterated along the new pressure distributions established by the 1D analysis until a convergence tolerance is met. Thrust augmentation and Specific Impulse values are predicted using a control volume approach.

For the validation test cases, the nozzle characteristic net is very similar to that of previous research. Plume characteristics are in good agreement but fluctuate in accuracy due to flow structure formulation. The individual unit processes utilized by the Method of Characteristics are found to vary their outputs by up to 0.025% when compared to existing sources. Rocket thrust and specific impulse are increased by up to 22% for a static system and 15% for an ejector flow at Mach 0.5. Evidence of Fabri conditions were observed in the flow visualization and graphically through the performance predictions. It was determined that the optimum ejector divergence angle for an air augmented rocket greatly depends on the stagnation pressure ratio between the primary and secondary flows.

## ACKNOWLEDGMENTS

I want to start off by thanking my family and friends for the continuous support they provided throughout this research. As a key foundation throughout the trials of success, my parents offered words of encouragement when challenges arose and expressions of praise as each was surmounted. My sister, brother, and new nephew each strengthened my resilience in their own way, providing optimism when it was most needed. I would also like to thank my grandfather for the introduction to aviation and further developing my passion for the science of aeronautics. His stories of flight provided entertainment and insight throughout my academic career.

My gratitude is also extended to Dr. DeTurris for expanding my knowledge of hypersonics through past courses and this project. Lessons learned from her guidance and enthusiasm will provide further motivation as my career continues to develop. I must also thank Kyle Johnson for his work on the SCAARD as it helped direct the initial aim of the current research. Furthermore, I would like to thank Brett Morham and Paul Riley for their simulations which provided a foundation for ARES. I must also acknowledge Jeffrey Muss for granting access to the CEA program. His support of my academic pursuit became invaluable as ARES developed.

I want to thank my past three roommates, Peter Sudak, Christian De Leon, and Michael Giglio, for their eagerness to provide distractions at unannounced times. The spirit they displayed helped uplift me throughout the entire year. Finally I would like to say thank you to my good friends Joshua and Nicole Hoover. They both supplied social and intellectual diversion for as long as I can remember and are the definition of true friends.

## TABLE OF CONTENTS

	Page
List of Tables .....	vii
List of Figures .....	ix
Nomenclature .....	xi
1. INTRODUCTION .....	1
1.1 ROCKET BASED COMBINED CYCLE.....	2
1.2 ARES OVERVIEW .....	5
2. LITERATURE REVIEW .....	7
2.1 SEMINAL WORKS.....	7
2.1.1 Fabri .....	7
2.1.2 Emanuel .....	11
2.1.3 Love .....	13
2.2 FLOW STRUCTURES.....	16
2.2.1 Expansion Fan .....	17
2.2.2 Intercepting Shock .....	17
2.2.3 Mach Disk.....	18
2.2.4 Shear Layers .....	19
2.2.5 Double Shock Diamond.....	20
2.3 MOTIVATIONAL WORKS .....	21
3. METHOD OF CHARACTERISTICS.....	27
3.1 BACKGROUND .....	27
3.2 INITIAL DATA LINE .....	28

3.3	CHARACTERISTIC AND COMPATIBILITY EQUATIONS.....	29
3.4	UNIT PROCESSES .....	32
3.5	CONCEPTUALIZING THE MOC .....	38
4.	METHOD OF CHARACTERISTICS IMPLEMENTATION .....	40
5.	COMBUSTION .....	42
6.	PERFORMANCE ANALYSIS .....	44
6.1	THRUST AUGMENTATION .....	44
6.2	SPECIFIC IMPULSE .....	47
6.3	INITIAL EFFECTS.....	47
7.	ARES' PROCESS .....	49
8.	TESTING CONFIGURATIONS.....	57
8.1	VALIDATION.....	57
8.2	ADDITIONAL TESTING .....	60
9.	RESULTS.....	62
9.1	UNIT PROCESSES VALIDATION .....	62
9.2	FLOW VISUALIZATION .....	64
9.3	PERFORMANCE COMPARISONS.....	77
9.4	EJECTOR DIVERGENCE ANGLE STUDY .....	88
10.	CONCLUSION .....	95
11.	FUTURE WORK .....	97
	REFERENCES .....	99
	APPENDIX A .....	103

## LIST OF TABLES

Table	Page
Table 1: MOC Equations of Interest .....	31
Table 2: Unit Process Comparison with Zucrow & Hoffman (Z&H).....	63
Table 3: ARES' Performance Predictions .....	78
Table 4: Ejector Divergence Angle Study Thrust Augmentation Predictions for an Ethanol/Oxygen(G) Rocket-Ejector.....	88
Table 5: Ejector Divergence Angle Study $\Delta I_{sp}$ (sec) Predictions for an Ethanol/Oxygen(G) Rocket-Ejector .....	89



## LIST OF FIGURES

Figure	Page
Figure 1: Air Augmented Rocket .....	2
Figure 2: Hypersonic Vehicle <sup>8</sup> .....	3
Figure 3: Comparison of propulsive efficiencies at various Mach numbers <sup>8</sup> .....	4
Figure 4: ARES High-Level Flow Chart .....	6
Figure 5: Subsonic Mixing Case .....	9
Figure 6: Fabri's Separated Flows Condition <sup>12</sup> .....	9
Figure 7: Fabri Choke Condition <sup>12</sup> .....	10
Figure 8: Blocked Flow Case .....	11
Figure 9: Love's MOC solution with overlapping characteristics due to the Barrel Shock .....	15
Figure 10: Aerodynamic Structure of Jet from Highly Underexpanded Nozzle <sup>16</sup> .....	17
Figure 11: Double Shock Diamond Flow Structure <sup>22</sup> .....	21
Figure 12: Timeline of Cal Poly Air Augmented Rocket Research .....	22
Figure 13: SCAARD Final Design .....	25
Figure 14: Sauer's Method Compared <sup>10</sup> .....	29
Figure 15: Interior Point Unit Process <sup>10</sup> .....	33
Figure 16: Axis of Symmetry Unit Process.....	34
Figure 17: Direct Wall Point Unit Process.....	35
Figure 18: Inverse Wall Point Unit Process <sup>10</sup> .....	36
Figure 19: Free Pressure Boundary Point Unit Process .....	37
Figure 20: Example of a spreading characteristic net due to expanding flow .....	39
Figure 21: Control Volume for Kinzie's Thrust Equation <sup>28</sup> .....	46
Figure 22: Example of a folded characteristic net due to a shockwave <sup>20</sup> .....	52
Figure 23: Johnson's Combustion Chamber and Rocket Nozzle <sup>2</sup> .....	59

Figure 24: Johnson's Straight Ejector (Left) and Diverging Ejector (Right) .....	59
Figure 25: Initial-Value Problem, ARES (Left), Zucrow & Hoffman <sup>10</sup> (Right) .....	65
Figure 26: Nozzle Flow field Entering Throat from Left to Right; ARES (Top), Zucrow & Hoffman (Bottom).....	66
Figure 27: MOC Solution to Underexpanded Plumes; ARES $P_1/P_a = 10$ (Top), Love $P_1/P_a = 10$ (Middle), Love $P_1/P_a = 20$ (Bottom).....	68
Figure 28: MOC Solution to Underexpanded Plume; $P_1/P_a = 20$ .....	69
Figure 29: Calculated Plume Boundary from Nozzle Lip to Maximum Height; ARES (Top), Love <sup>13</sup> (Bottom).....	72
Figure 30: Flow Visualization of Diverging Duct Configuration with $MR = 1.23$ , $P_{0p}/P_a = 22.1$ (Above), Theoretical Structure of Highly Underexpanded Nozzle <sup>20</sup> (Below).....	74
Figure 31: ARES' 1 Iteration Solution (Left) and Final Iteration Solution (Right).....	75
Figure 32: Ejector Thrust Produced: A) Straight Ejector, B) 5.88° Diverging Ejector.....	79
Figure 33: Straight Duct Configuration, $MR = 1.82$ , $P_{0p} = 360\text{psi}$ .....	81
Figure 34: $\Delta I_{sp}$ due to Ejector: A) Straight Ejector, B) 5.88° Diverging Ejector .....	83
Figure 35: Pressure Regions in Straight (Top) and Diverging (Bottom) Ejectors <sup>2</sup> .....	84
Figure 36: Total $I_{sp}$ for Rocket-Ejector: A) Straight Ejector, B) 5.88° Diverging Ejector .....	85
Figure 37: Johnson's Rich Mixture Ratio Rocket Ejector Showing Combustion outside the Ejector.....	86
Figure 38: Secondary Flow Aerodynamic Choke Point Location Vs Stagnation Pressure Ratio .....	87
Figure 39: Thrust Augmentation at 10,000ft: A) Mach = 0, B) Mach = 0.5.....	90
Figure 40: Example of Increasing Ejector Flow Area as Divergence Angle Increases .....	92
Figure 41: $\Delta I_{sp}$ due to Ejector at 10,000ft: A) Mach = 0, B) Mach = 0.5.....	93

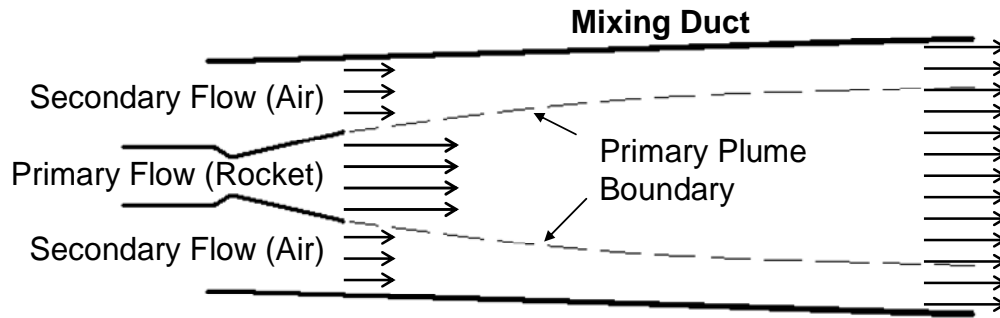
## NOMENCLATURE

$A$	Area	(ft <sup>2</sup> , in <sup>2</sup> )
$a$	Speed of Sound	(ft/s)
$D$	Diameter of Nozzle Exit	(in)
$g$	Gravity	(ft/sec <sup>2</sup> )
$h$	Step Size	(in, ft/sec)
$\dot{m}$	Mass Flow Rate	(lb <sub>m</sub> /s)
$M$	Mach	-
$psi$	Absolute Pressure	(lb <sub>f</sub> /in <sup>2</sup> )
$P$	Pressure (Static unless subscripted with 0)	(lb <sub>f</sub> /in <sup>2</sup> )
$R$	Specific Gas Constant	(lb <sub>f</sub> ft/lb <sub>m</sub> °R)
$T$	Thrust	(lb <sub>f</sub> )
$T$	Temperature	(°F, °R)
$u$	Axial Component of Velocity	(ft/s)
$V$	Velocity	(ft/s)
$v$	Normal Component of Velocity	(ft/s)
$x, z$	Axial Coordinate	(in)
$y, r$	Normal Coordinate	(in)
<i>Greek</i>		
$\gamma$	Specific Heat Ratio	-
$\delta$	Condition for Axisymmetric or Planar Calculations	-
$\theta$	Flow Angle	(degrees)
$\lambda$	Slope of Characteristic Line	-
$\mu$	Mach Angle	(degrees)
$\nu$	Prandtl-Meyer Function	-
$\rho$	Radius of Curvature	(in)
$\Phi$	Velocity Potential	-
$\emptyset$	Thrust Augmentation	-
<i>Subscripts</i>		
0	Stagnation Condition	
1, j	Nozzle Exit	
2	Secondary Flow	
4	MOC Output	
a, $\infty$	Ambient Air	
char	Characteristic	
e	Ejector Exit	
mix	Mixed Primary and Secondary Flow	
n	Incremented Value	
p	Primary Flow	
s	Secondary Flow Input	
t	Rocket Throat	
tot	Combined Primary and Secondary Flow	
$\pm$	Plus or Minus Characteristic	
<i>Superscripts</i>		
*	Aerodynamic Throat	
°	Degree	

## **1. Introduction**

On May 1<sup>st</sup>, 2013 the Boeing X-51A hypersonic vehicle was launched on its fourth and final attempt to achieve scramjet powered flight for 300 seconds. Undeterred by three previously failed attempts, the team of engineers was rewarded with a successful firing which culminated in velocities exceeding that of Mach 5 for the intended flight time<sup>1</sup>. In only a few decades, flight sciences have nearly standardized the process of attaining supersonic speeds and move forward in their pursuit of sustaining ever higher velocities. Though rocketry has long been able to achieve hypersonic speeds, it is greatly limited by a requirement to carry its own oxidizer. While the added weight reduces performance, the explosiveness and simplistic design continues to encourage their use. One method of increasing rocket efficiency while maintaining a large thrust to weight ratio, is to introduce an ejector into the system.

An ejector is essentially a hollow duct which is positioned around any moving fluid in order to direct the outflow. When the moving fluid is in a liquid form, the system is usually referred to as an induction pump. If an ejector is attached to the end of a rocket nozzle, the system is described as an air augmented rocket, rocket-ejector, or ducted rocket. In this instance the rocket exhaust is designated as the primary flow. The duct diameter is large enough so that a secondary flow, commonly air, can travel between its wall and the rocket plume. A cut away drawing of an air augmented rocket is presented in Figure 1 and shows how the duct extends downstream of the nozzle exit plane (2). A



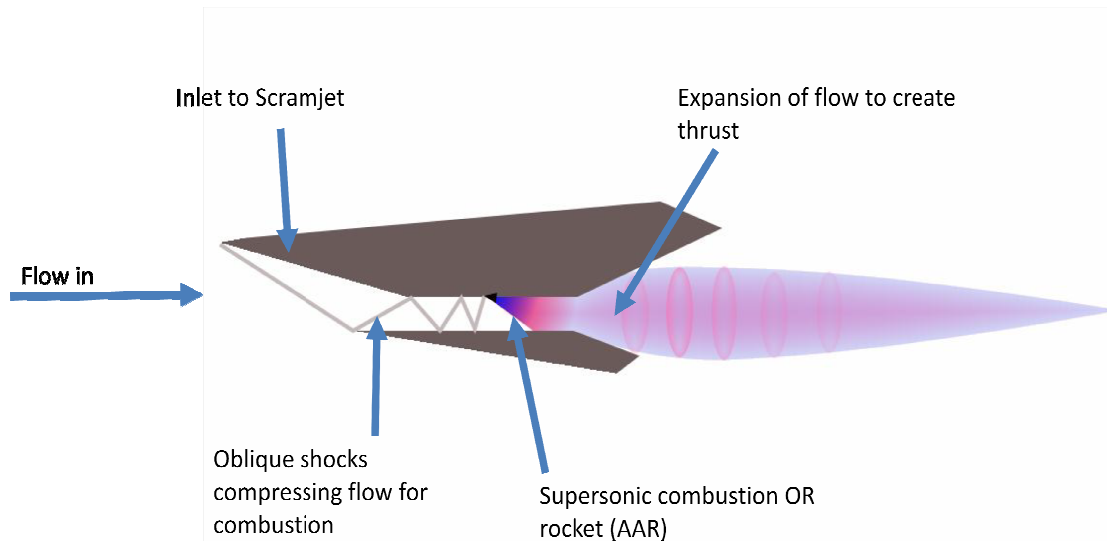
**Figure 1: Air Augmented Rocket**

rocket-ejector works by using a high velocity primary fluid to entrain a lower velocity secondary flow. The secondary flow usually surrounds the primary and adds ejected mass to increase thrust. Rockets operating in the atmosphere use air as the secondary fluid due to its availability, but any fluid can technically be employed. Ejectors have been utilized often in industrial applications but rarely with the intention of added thrust<sup>3</sup>.

### **1.1 Rocket Based Combined Cycle**

The number of applications for a rocket-ejector is great due to the simplicity of construction<sup>4,5,6</sup>. One of the main applications is their use in a Rocket Based Combined Cycle (RBCC) vehicle shown in Figure 2. Though still under development, RBCCs can be used as single stage to orbit vehicles with the potential of being reusable. A RBCC flight plan may look something like this: Takeoff from ground using rocket-ejector, transition to ramjet mode at about Mach 2-3 for more efficient flight, shift to scramjet mode at Mach 4-6, transition to pure rocket mode when the atmosphere becomes too thin for the scramjet to operate efficiently<sup>7</sup>. A reciprocating or turbine based engine may also

be considered for the initial acceleration to ramjet mode; this may not be ideal since, although efficiency is increased, complexity and weight is also greatly increased. A ramjet, scramjet, and rocket-ejector all have simple geometries which lead to lightweight designs and can be combined compactly as in Figure 2. In this configuration



**Figure 2: Hypersonic Vehicle<sup>8</sup>**

a rocket can be placed at the throat of the vehicle which acts as the duct to create an air augmented rocket. Further thrust could theoretically be produced by adding fuel to the secondary flow and create an afterburning rocket-ejector<sup>9</sup>.

Figure 3 shows a good representation of how RBCCs compare to other propulsion systems. While it can be seen that Turbine Based Combined Cycles (TBCCs) have much greater efficiency than RBCCs for velocities up to ramjet mode, the system ends once scramjet operation ceases. It is possible to add a rocket for the remaining burn required for orbit at the end of a TBCC engine, but then the entire turbomachinery section would have to be retained. The excess weight and complexity not only affects the initial

acceleration of the craft, albeit at a better efficiency, but also increases the fuel necessary for the remaining systems as well as the final rocket burn. The extra fuel required to account for the additional engine weight may cause the overall performance of such a system to be less than that of a RBCC. Further costs can be predicted due to the increased system complexity when compared to an RBCC engine.

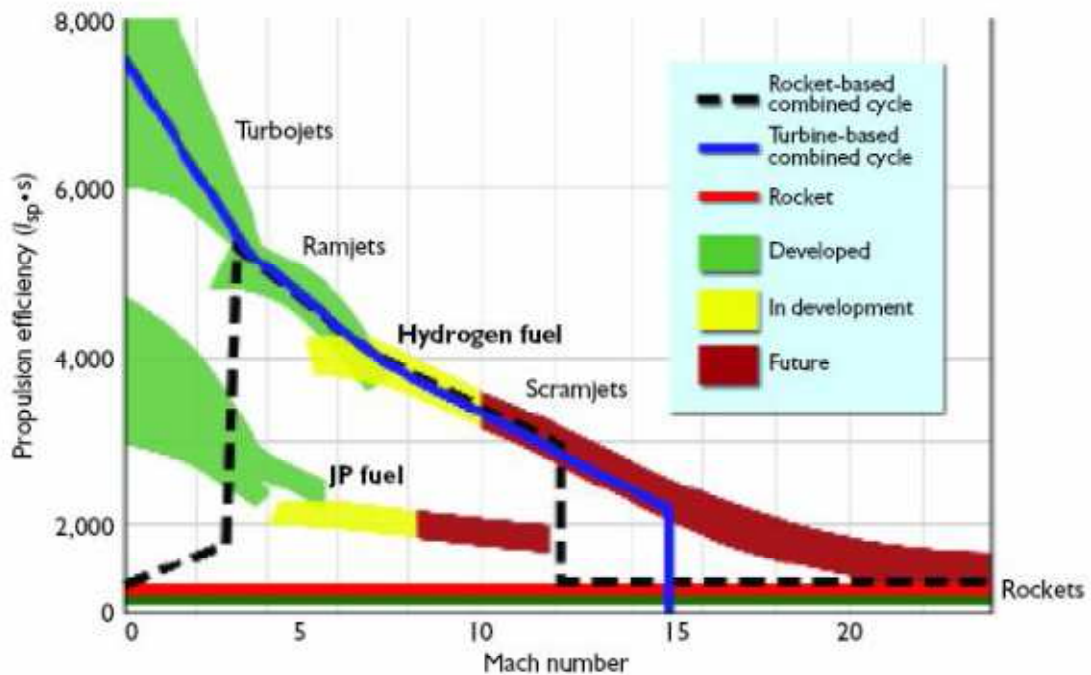


Figure 3: Comparison of propulsive efficiencies at various Mach numbers<sup>8</sup>

Unfortunately many of the technologies required to create an RBCC engine are still underdeveloped, the rocket-ejector portion being one. Complex interactions occur between the primary and secondary flows and require further investigation to be fully understood. The current research compiles the current theoretical knowledge into a

computer simulation in order to predict performance values of a wide variety of rocket-ejector configurations. The Axisymmetric Rocket Ejector Simulation (ARES) is derived from a previous program created by Brett Morham<sup>8</sup> called the Cal Poly Supersonic Ejector (CPSE) simulation. Like the CPSE simulation, ARES is written in the MATLAB programming language and is largely developed from a Method of Characteristics (MOC) FORTRAN code created by Zucrow & Hoffman<sup>10</sup>.

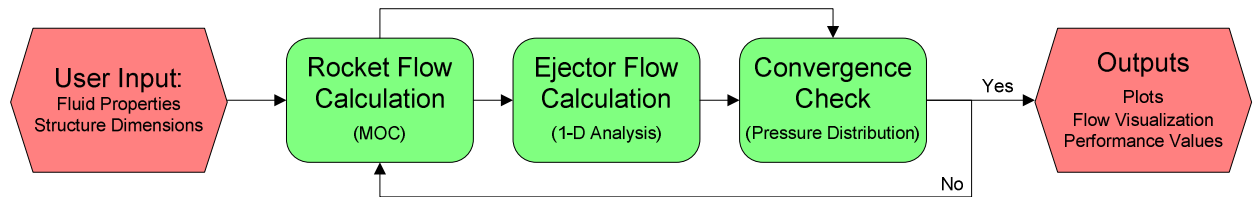
## **1.2 ARES Overview**

ARES' functionality can be broken down into five high-level steps beginning with a user input of fluid properties and structure dimensions. While a number of these inputs go through some pre-processing, most are fed directly into the primary flow MOC algorithm. ARES then calculates the rocket flow field from the throat of the nozzle to the first Mach Disk in the plume. The plume calculation is based off an initial guess for the pressure distribution along the boundary which separates the exhaust from the ambient ejector flow. Flow properties along that boundary are then taken from the MOC solution and fed into the secondary flow calculation. The secondary flow function finds new values for the pressure distribution based on the plume boundary emanating from the nozzle exit. The new and old pressure distributions are checked for convergence and then fed back into the method of characteristics function to recalculate the plume. The cycle repeats itself until the difference between the new and old pressure distribution separating the two flows is negligible. ARES then post-



processes the final values and plots the converged primary and secondary flow fields.

The procedure is demonstrated in Figure 4.



**Figure 4: ARES High-Level Flow Chart**

## **2. Literature Review**

Though a great deal of research has already been completed on the subject of air augmented rockets, a large amount is still required to fully model the flow physics. Much of the work that has been done gives a good understanding of the various interactions between the primary and secondary flows. Other important works that also assisted this research included numerical models utilizing the MOC.

### **2.1 Seminal Works**

A majority of the understanding pertinent to this research was compiled by three authors of interest. Fabri begins background with work on the complicated interactions between the primary and secondary flows. Emmanuel provides further understanding with an application of Fabri's work and details the usefulness of a 1D analysis. Love then presents a great deal of important information on how to utilize the MOC for complex rocket flows. Together these sources form a foundation which resulted in the conception of ARES.

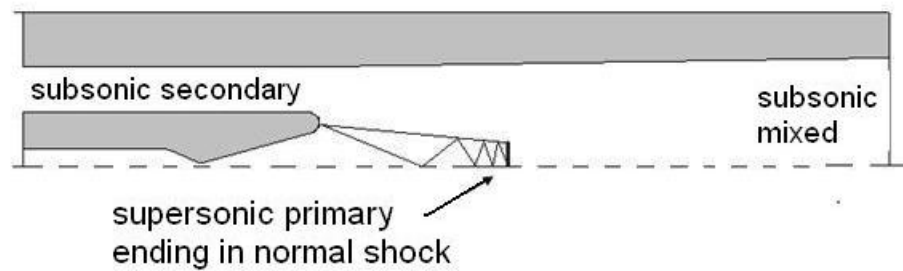
#### **2.1.1 Fabri<sup>11</sup>**

Fabri observed the flow field of a jet ejector and defined the operating conditions it goes through as the primary to secondary stagnation pressure ratios vary. Fabri's research focused on air to air ejectors with supersonic primary flows and expanded upon how it interacted with the secondary flow aerodynamically. His analysis is based off an experiment which used an axisymmetric primary nozzle exhausting into a secondary cylindrical duct. He assumed the two streams were inviscid but added a correction

factor for the friction created between the secondary flow and ejector wall. Fabri solves the primary flow with one dimensional analysis but recommends how the method of characteristics should be employed instead for more accurate results. The plume he approximately solved for incorporated the one dimensional analysis as well as correction factors from experimental data.

The experiment conducted by Fabri led to general trends which described a flow field largely influenced by the ratio of stagnation pressures. These modes of operation are only used when describing the initial plume where the primary and secondary flows interact. As the flow progresses further downstream of the nozzle it becomes increasingly dominated by viscous mixing forces. The four possible modes described are subsonic mixing, separated flows, Fabri choke, and Fabri's blocked case. These conditions are useful references when analyzing rocket-ejector flow and will be employed when describing various testing configurations.

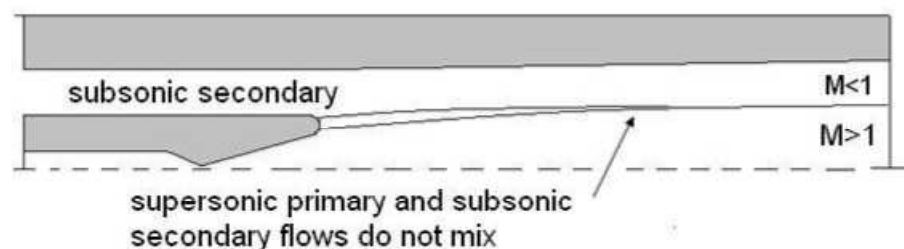
When the primary stagnation pressure is low, or conversely the secondary pressure high, the exhaust plume will be considered overexpanded. Overexpanded flow creates a converging plume due to the secondary flow pressure being greater than the primary after the flow exits the nozzle. As shown in Figure 5, a series of oblique shocks form within the plume which reduces the Mach and raises the pressure (12). The oblique shock train ends in a Mach Disk, or normal shock, which equalizes the primary pressure with the secondary. Here the two flows are considered fully mixed but with great pressure losses due to the shocks. If the exhaust pressure is low enough then the flow



**Figure 5: Subsonic Mixing Case**

will be grossly overexpanded and may even result in a normal shock occurring before the primary flow exits the nozzle. While the sudden discontinuity causes all flow in the ejector to become mixed, it also produces subsonic speeds.

Fabri described the separated flow case as having two streams which act independently of each other. This case occurs when the primary jet plume expands just to the edge of the secondary boundary as shown in Figure 6. Though underexpanded, this primary flow does not interact with the secondary flow except for some viscous mixing which is not accounted for by Fabri. The secondary flow remains subsonic while the primary continues at supersonic speeds; this allows them to be analyzed individually. The separated flow case occurs when the stagnation pressure ratio is not high enough to choke the secondary flow.



**Figure 6: Fabri's Separated Flows Condition<sup>12</sup>**

The Fabri choke case occurs when the stagnation pressure ratios are higher than that of the separated flow case. The primary jet is more underexpanded which causes it to reach farther into the secondary stream area. The developing flow interactions cause a slip line to form. As shown in Figure 7, the slip line travels into the ejector duct area and creates a converging like shape for the secondary flow. The air in the ejector is bound by the wall and the higher pressure flow at the slip line and so must converge in order to continue downstream. Since the secondary flow velocity is initially very low, the convergence causes it to speed up. If the ejector static to stagnation pressure ratio is low enough at the plume's max height, and conversely the secondary's minimum area, the flow will reach supersonic speeds and continue to expand downstream. The flow structure formed can be treated similar to a virtual throat as it creates a choked secondary flow based on the underexpanded primary plume. If no shock occurs, this

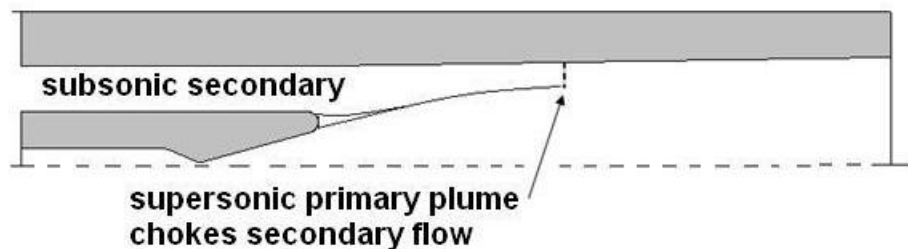


Figure 7: Fabri Choke Condition<sup>12</sup>

secondary flow with increasing Mach can cause the rocket thrust to be noticeably increased. Therefore the Fabri choke case is the most sought after mode for an air augmented rocket.

Fabri block is created when the stagnation pressure ratios are higher than the Fabri choke case. As the underexpanded plume continues to grow in width within the duct the virtual throat created becomes smaller, eventually leading to a blocked flow case. Blocked secondary flow occurs when the jet plume reaches the duct wall and effectively halts all secondary flow from moving through the ejector as shown in Figure 8 (12). The Fabri block condition can be highly detrimental to rocket-ejector performance since the purpose of the duct is to entrain extra flow. With the blocked case present, the ejector only adds drag and weight to the system. The large repercussions of this mode encourage further research so that condition can be avoided.

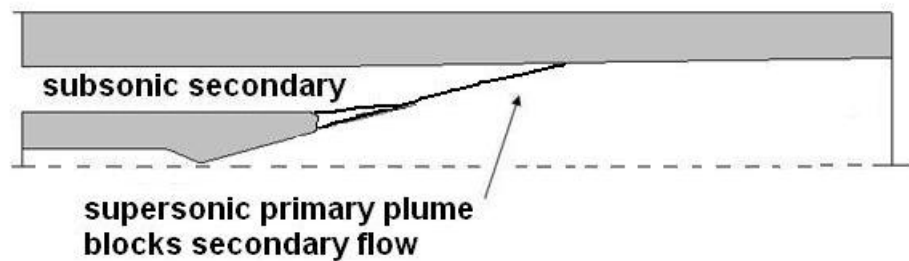


Figure 8: Blocked Flow Case

### 2.1.2 Emanuel<sup>3</sup>

The research presented by Emanuel focuses on a one-dimensional model which is configured as either a constant area or constant pressure rocket-ejector. While both conditions can be applied simultaneously, the current study focused on isolating the individual boundary conditions. Fabri's inviscid theory was applied in the calculations and conclusions were made on the validity of the solutions.

The one-dimensional analysis begins at the nozzle exit axial location and relies on stagnation fluid properties and the rocket-ejector geometry. Initial computation is then applied to find other required fluid properties such as mass flow rate. Emmanuel utilizes common one-dimensional equations for the mixing of two parallel streams to find the uniformly mixed solution. These one-dimensional mixing equations are applied for both the constant area and constant pressure cases for separate analysis.

This method of solving flow fields is optimal for analyzing a wide variety of test cases in a short amount of time. The one-dimensional equations greatly simplified the resulting flow fields by eliminating most flow phenomenon that should be present. While this decreases time spent on computation, it also reduces accuracy of the solution. A useful aspect of this method is that it allows for parametric studies to be easily generated.

The constant area condition applied in the first set of solutions was focused on the streams' areas at one axial location. The condition was incorporated into the calculations by constraining the secondary area at the nozzle exit to be equal to that of the nozzle exit area. The constant pressure stipulation constrained the primary and secondary flows to have equal static pressures at the nozzle lip. From that point, the downstream ejector shape is similar to that of a converging-diverging nozzle. A normal shock is said to be present at the throat of the ejector and is where Emmanuel assumes fully mixed flow.

The constant area assumption was found to provide best performance when the secondary flow at the inlet neared sonic conditions. The constant pressure condition created optimum performance at low secondary inlet Mach numbers. These optimum Mach numbers were found to be dependent on the mass flow ratio of the two streams. Emmanuel found that Fabri's inviscid theory was greatly restricted when applied to one-dimensional models and determined that computations involving the Method of Characteristics could largely enhance the flow field solutions.

### **2.1.3 Love<sup>13</sup>**

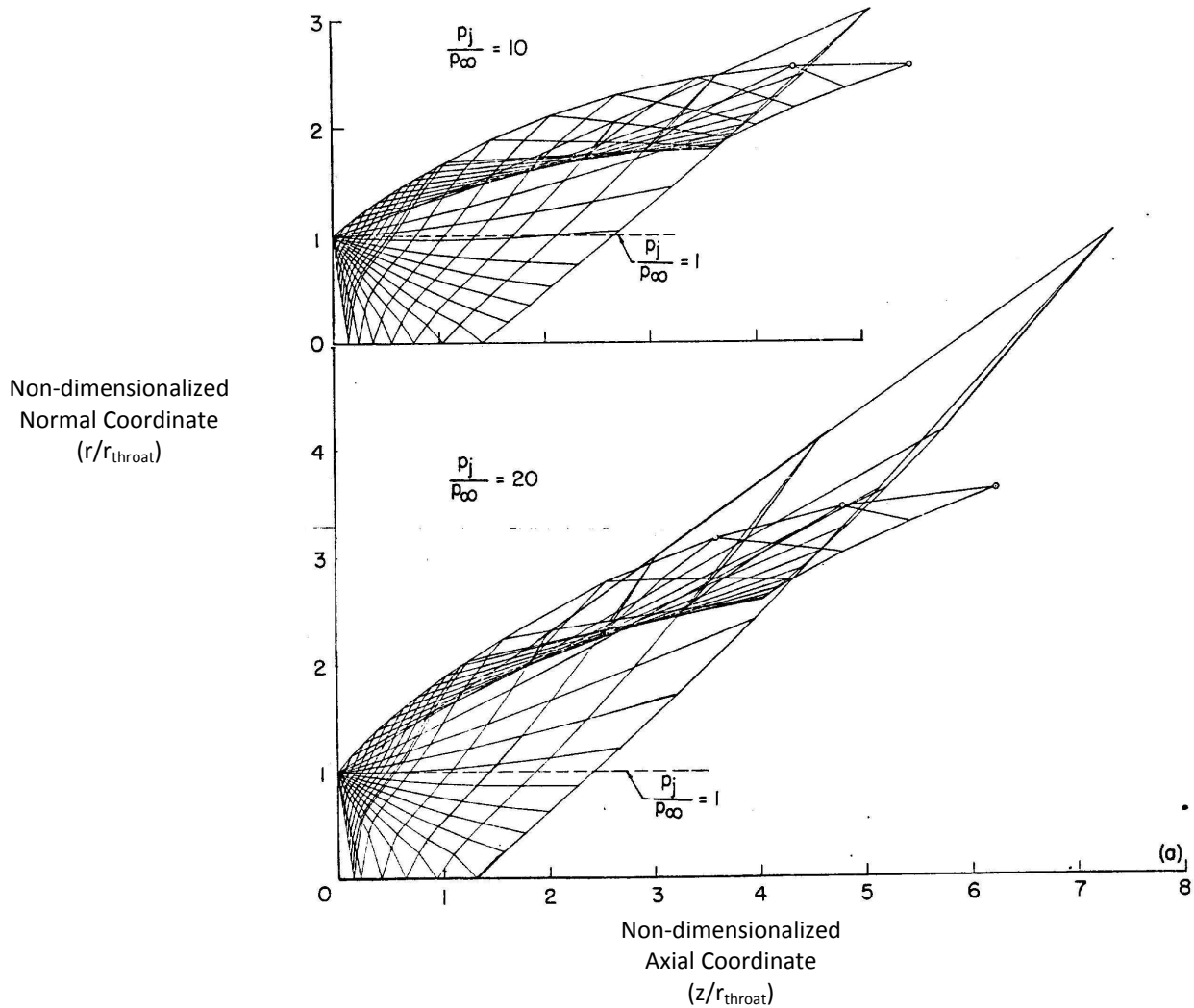
A great deal of work was done by Eugene S. Love on experimental and theoretical axisymmetric free jets. Love performed analysis on a large variety of configurations for sonic and supersonic jets exhausting into still air and supersonic streams. The rotational Method of Characteristics was applied to numerically model these configurations and to provide insight as to the flow structures observed in the experimental images.

Love was able to plot a large number of trends which helped to analyze underexpanded jets. The numerous configurations altered the static pressure to ambient pressure ratio, specific heat ratio, nozzle exit flow angle, nozzle exit Mach number, and secondary stream Mach number. The experimental studies incorporating supersonic secondary flow were conducted in Langley's 9-inch supersonic wind tunnel which allowed for schlieren photography. These images were compared with the predicted flow fields created by the rotational Method of Characteristics.



In Love's application of the MOC, the characteristic net was started at the exit plane of the nozzle and halted once the last right running characteristic created the maximum height of the plume boundary. According to Love, the flow boundary has little practical value after the maximum height has been reached. Halting the MOC as early as possible was also desirable because it reduced the number of calculations required, which was an important aspect due to the computing power in 1959.

Love also noted that applying underexpanded configurations would create shocks within the jet and thus affect the MOC calculation. The shock would be generated at the ending characteristic of the expansion fan and would grow in strength as the flow moved downstream. This intercepting shock, or barrel shock, is due to the secondary flow creating a constant pressure boundary on the jet and will be described more in the following sections. The MOC cannot handle a shockwave because it is a discontinuity in the flow which violates one of its assumptions. Love discussed a number of options to allow calculations to continue, the simplest of which is referred to as the foldback method. The procedure starts by allowing the MOC to continue its calculations of the flow field even when a shock point is discovered. Though this initially causes an overlap in the characteristic net, the MOC unfolds itself as the calculations develop downstream. The folded characteristic net can be seen in Figure 9(13).



**Figure 9: Love's MOC solution with overlapping characteristics due to the Barrel Shock**

Here the horizontal zero location represents the exit plane of the nozzle. The top figure shows the characteristic net of a plume with a primary to ambient static pressure ratio of 10. The bottom shows the same net but for a pressure ratio of 20. Both figures use a specific heat ratio of 1.4 and a nozzle exit Mach of 1.5. Love nondimensionalized the x and y axes by the radius of the nozzle exit so that multiple configurations can be compared. It can be seen that as the pressure ratio increases, the overlapping of characteristics becomes more severe. The second and final step of the foldback

procedure is to edit the completed characteristic net by removing the overlapping characteristics. The foldback procedure was found to be the simplest to apply and at least as accurate, if not the most, of all the discussed possibilities for integrating shock calculations with the MOC. The detailed research Love presented on axisymmetric free jets paved the way for other, more complex, MOC simulations<sup>4,10,14,15</sup>.

## **2.2 Flow Structures**

The flow phenomena found in rocket plumes was a topic of great importance while developing ARES. The structures not only affect the final fluid properties of the plume, but also determines how far the Method of Characteristics can be carried downstream of the nozzle. Common flow structures present in rocket-ejectors include Expansion Fans, Barrel Shocks, Mach Disks, Shear Layers, and occasionally the Double Shock Diamond. Many of these structures are shown in Figure 10 as an aid to the following sections. Though Figure 10 shows a constant pressure condition for the plume boundary, ARES will calculate a variable pressure distribution due to the jet accelerating the secondary flow.

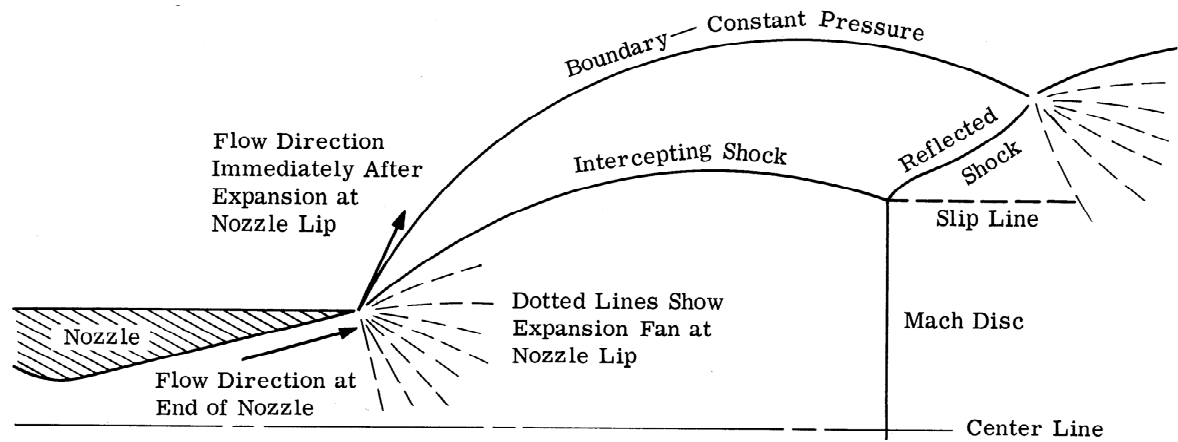


Figure 10: Aerodynamic Structure of Jet from Highly Underexpanded Nozzle<sup>16</sup>

### 2.2.1 Expansion Fan

Ejectors are most effective when used with an underexpanded rocket due to enhanced mixing conditions<sup>15,16</sup>. The expansion fan generated at the nozzle lip encourages the plume to move into the ejector area, constricting the secondary flow. The Mach and direction of the rocket flow is determined by the Prandtl-Meyer function which uses the secondary static pressure as the equalizing value. The Prandtl-Meyer function merely calculates the change in fluid properties required to meet an equilibrium condition. This means that the final flow angle after the expansion fan from a conical C-D nozzle will be greater than that calculated by the Prandtl-Meyer function alone<sup>17</sup>.

### 2.2.2 Intercepting Shock

A great deal of research has been done on the existence and modeling of the Intercepting (Barrel) Shock<sup>7,9,13,14</sup>. As soon as the flow exits the expansion fan, an oblique shock forms due to the pressure imposed on the plume boundary from the

secondary flow. The requirement for static pressure equilibrium between the two flows causes the shape of the jet boundary to be non-isentropic. The compression waves generated at the boundary from the resulting shape intercept the expansion waves from the expansion fan. The point where each compression wave first intersects an expansion wave marks another location on the shock. In a real sense, when these waves meet each other they align rather than overlap. The continuous coalescence of compression waves adds strength to the discontinuity and creates a shockwave<sup>18</sup>. The shockwave curves downstream as more characteristics of the same type intercept each other. The characteristic curve can be looked upon as creating an outline similar to that of a barrel. One important result of having a curved shock as opposed to straight is that the irrotational flow assumption breaks down<sup>10</sup>. While the irrotational MOC can continue to converge on point solutions after the intercepting shock, its error will increase.

### **2.2.3 Mach Disk**

Depending on the static pressure ratio between the primary and secondary flows ( $P_1/P_2$ ), a Mach Disk may form. Love<sup>13</sup> determined that  $P_1/P_2 \geq 4$  for a Mach Disk to be present. An air augmented rocket requires much higher pressure ratios than this to be efficient and so a Mach Disk is almost always formed. Pressure ratios below four would result in an oblique shock reflection at the centerline.

Abbett<sup>19</sup> described the formation of a Mach Disk as a result of an unfavorable pressure gradient. The gradient is created by the expanding jet and higher pressured ambient

flow downstream. The decreasing static pressure causes the exhaust to collapse in on itself in order to equalize with the ambient pressure. Physically this collapse takes the form of a normal shock which is referred to as a Mach Disk in this circumstance. As the flow after the normal shock is subsonic, MOC calculations cannot be continued.

While a concrete process to calculate the location of a Mach Disk is still unformed, many approximations with varying degrees of accuracies have been created. Adamson's<sup>16,20</sup> method placed the Mach Disk at the point where the flow after a normal shock would have same pressure as that of the ambient flow. The assumption that the pressure should be equal to ambient proved to be relatively accurate when compared with the other approximations and the easiest to implement.

A triple point is created where the Mach Disk intercepts the Barrel Shock as shown in Figure 10. The Barrel Shock reflects off this point to produce an impinging oblique shock which will reflect off the jet boundary as another expansion fan. Since the flow downstream of the oblique shock is supersonic and the flow downstream of the Mach Disk is subsonic, a Slip Line is formed. The progression of flow structures beginning at the nozzle lip and ending at the start of the following expansion fan will be repeated until viscous interactions dominate the flow field and distort the cycle<sup>18</sup>.

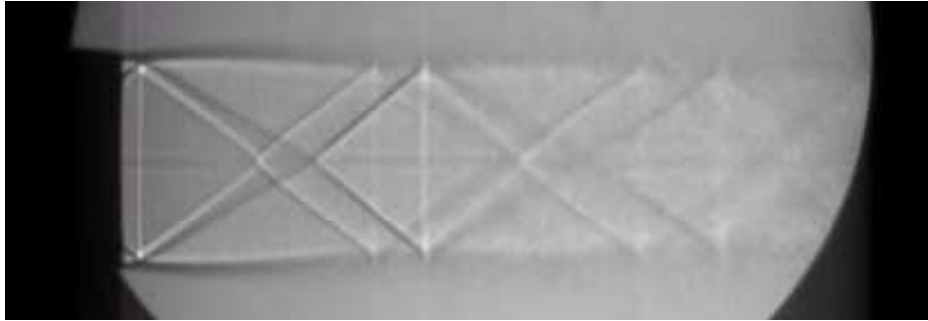
#### **2.2.4 Shear Layers**

The Shear Layer of focus for rocket-ejector development is created where the two streams meet. Turbulence models for the layer have been produced with reasonable

success<sup>7,9,21</sup>. Addy<sup>15</sup> included viscous correction factors in his inviscid MOC calculation and compares resulting entrainment ratios. Though friction is inherent in every flow, research has determined that the initial fluid properties of free jets are largely dominated by non-viscous forces<sup>16</sup>.

### **2.2.5 Double Shock Diamond<sup>22</sup>**

A second shock diamond may occur in the primary flow as a result of using a non-isentropic nozzle. An oblique shock will be created in a supersonic flow if the wall contour and slope are continuous, but the derivative of the slope is not. For conical nozzles, this point is located at the junction between the circular arc throat and the constant sloped wall. This causes compression waves to develop which can coalesce and form an oblique shockwave<sup>10</sup>. The oblique shock reflects off the centerline and wall as it propagates downstream until it exits the nozzle and reaches the plume boundary. The boundary conditions on the plume cause the shock to reflect off as an expansion fan and form a shock diamond<sup>18</sup>. The resulting flow structure, along with the shock diamond usually present in exhaust flow, form what appears to be a Double Shock Diamond as shown in Figure 11.

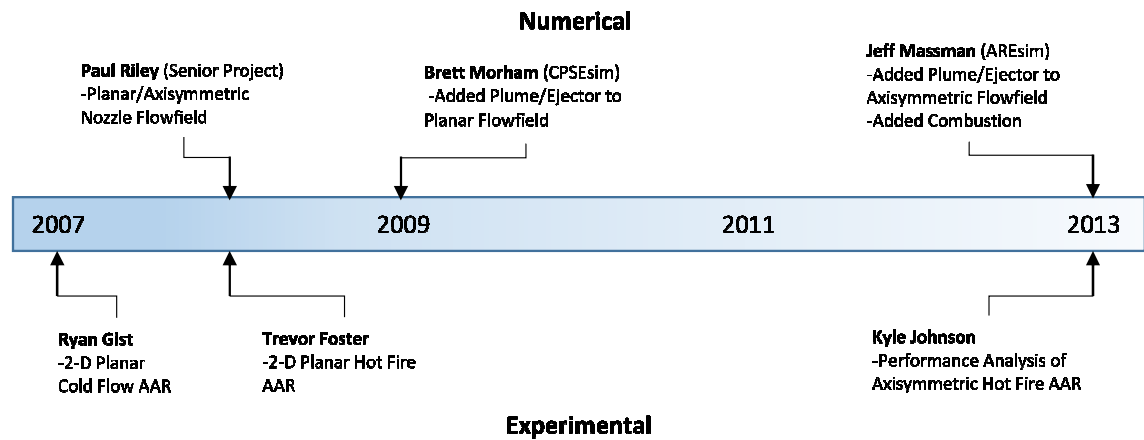


**Figure 11: Double Shock Diamond Flow Structure<sup>22</sup>**

### **2.3 Motivational Works**

A number of Cal Poly theses provided insight and motivation for ARES. ARES was developed from an earlier program named the Cal Poly Supersonic Ejector simulation (CPSE simulation) created by Brett Morham<sup>8</sup> and Paul Riley<sup>23</sup>. While both codes primarily use the MOC, the CPSE simulation was created for a 2-D air augmented rocket. ARES builds upon this to create a program that can handle the axisymmetric case as well as the 2-D case. The two theses of primary importance were completed by Brett Morham and Kyle Johnson<sup>2</sup> but several other projects are significant to note as well. A timeline of the relevant ventures is shown in Figure 12.





**Figure 12: Timeline of Cal Poly Air Augmented Rocket Research**

Ryan Gist<sup>24</sup> begins the list of pertinent theses for ARES with a cold flow experiment of a 2-D planar rocket-ejector in 2007. The experiment took place in Cal Poly's supersonic wind tunnel and analyzed how high pressure ratios affected the primary and secondary flow interactions. The test rig, named the Cal Poly Supersonic Ejector, supplied gaseous nitrogen as the primary flow and ambient air as the secondary flow. Gist focused on how the primary to secondary stagnation pressure ratio affects entrainment ratio. It was found for his apparatus that pressure ratios below 73 created the separated flow case and below 230 created Fabri choke conditions. Pressure ratios above this led to Fabri Block conditions. Gist's thesis was of primary importance in the original CPSE simulation as it was used to validate the code.

The next project chronologically was completed by Trevor Foster<sup>25</sup> in 2008. Foster experimented with a hot fire rocket-ejector to find, similar to Gist, how the two streams interact to produce entrainment. The apparatus used a Methane/Gaseous Oxygen rocket exhausting into ambient air. Various pressure ratios were tested with primary

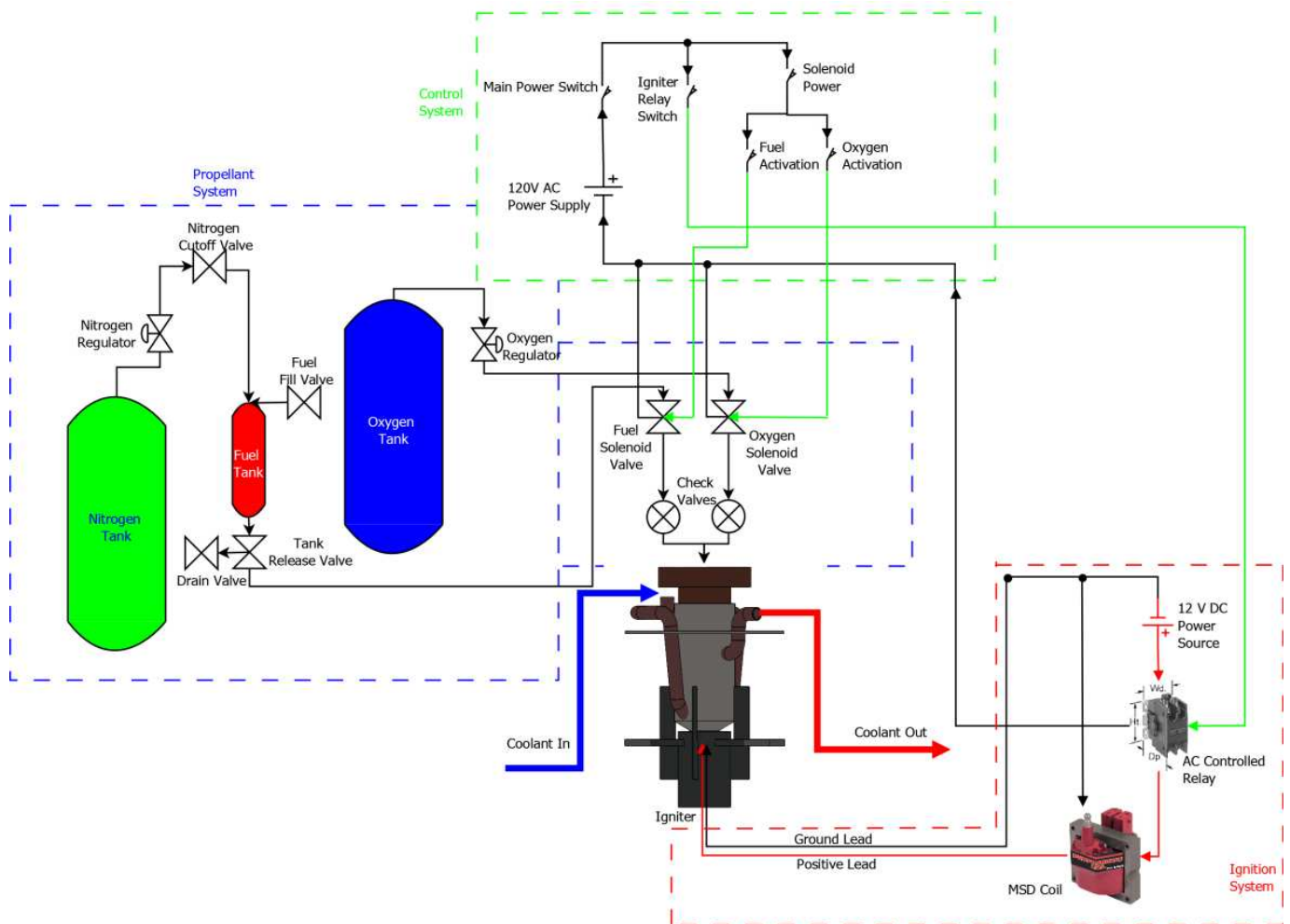
stagnation pressures ranging from 325 to 1032 psi. The highest primary to secondary stagnation pressure ratio reached was 74 which is significantly lower than that of Gist's experiment. Foster concluded that higher entrainment could be reached with a nitrogen-air rocket-ejector.

Around the same time that Foster was completing his experiment, Paul Riley was finishing a senior project. Paul developed a simulation in MATLAB from an irrotational MOC solution presented by Zucrow and Hoffman<sup>10</sup>. Riley used the MOC to analyze a 2-D planar and axisymmetric nozzle flow field. The numerical solution began at the throat and halted once it reached the exit plane of the nozzle. The program computed the flow field for any wall within reason which had a circular arc throat and transitioned seamlessly into a diverging section which would be specified by a 2<sup>nd</sup> order polynomial. Riley's senior project is utilized in both CPSE simulation and ARES.

In 2009 Brett Morham developed the CPSE simulation which is the precursor to ARES. Brett's addition extended the planar case's numerical flow field outside of the nozzle and into an ejector with a varying secondary pressure distribution. The pressure distribution was then iterated via the MOC calculation and another function that analyzed the secondary flow, until it converged. The CPSE simulation allowed for any flows to be used as the primary or secondary streams so long as all necessary fluid properties could be input.

One limitation of the CPSE simulation was that flow structures within the plume were not taken into account. The flow structures associated with underexpanded air augmented rockets include expansion fans, curved shocks, oblique shock reflections, and Mach Disks. Morham calculated entrainment ratios from this inviscid solution by dividing the mass flow rate of the secondary stream with that of the primary. The CPSE simulation also found the stagnation pressure ratios which demonstrated the separated flows, Fabri choke, and Fabri block cases.

The most recent of the relevant theses comes from Kyle Johnson which was completed in March of 2013. Johnson expanded upon Foster's and Gist's experiments by testing an axisymmetric, Methanol/Gaseous Oxygen rocket-ejector in Cal Poly's propulsion lab. The apparatus was named the Static Condition Air Augmented Rocket Demonstrator (SCAARD) and is shown in Figure 13(2). The configuration used ice water to actively cool the primary combustion chamber and allowed for different mixing ducts to be attached as ejectors. Johnson's experiment measured the thrust produced by the nozzle, mixing duct, and the overall system as a redundancy. The measured parameters differ from Gist and Foster who both studied the flow physics caused by the interactions of the two streams. The research tested a straight and diverging mixing duct with identical outer and inlet geometries. The primary combustion pressures used in this experiment had a range of 300 to 400 psi. Since the secondary flow was air, the primary to secondary stagnation pressure ratios were between 20 and 30.



**Figure 13: SCAARD Final Design**

Johnson found that straight mixing duct thrust averaged 0.97 pounds and the diverging mixing duct thrust averaged 0.18 pounds. Though the ejector did generate some positive force, it was found that the overall thrust of the system decreased an average of 0.62 pounds. The thrust reduction was believed to be caused by an unanticipated structural interaction occurring upstream of the ejector inlet. Johnson predicted that a significant low pressure region developed between the cooling jacket and ejector inlet. Since the cooling jacket had a larger structural area in the axial direction, it created a

negative thrust which counteracted the performance gain of the ejector. It was assumed that the negative thrust could be negated if designed accordingly.

ARES is developed as a derivative of the existing CPSE simulation produced by Brett Morham and uses Kyle Johnson's axisymmetric experimental results for validation. Though the general process displayed in Figure 4 is the same, the CPSE simulation and ARES utilize different versions of the MOC in their solution of the primary flow field.

### 3. Method of Characteristics

ARES uses the Method of Characteristics to analyze a steady, adiabatic, inviscid, irrotational supersonic flow field. This form of the MOC can also handle slightly non-isentropic flow with reasonable results<sup>10</sup>. Non-isentropic flow occurs with commonly used nozzles such as those which are parabolic or conically shape.

#### 3.1 Background

The MOC is a mathematical method to solve hyperbolic partial differential equations (PDEs). The governing nonlinear equation for isentropic, irrotational flows is the velocity potential equation shown below.

$$\left(1 - \frac{\Phi_x^2}{a^2}\right)\Phi_{xx} + \left(1 - \frac{\Phi_y^2}{a^2}\right)\Phi_{yy} + \left(1 - \frac{\Phi_z^2}{a^2}\right)\Phi_{zz} - \frac{2\Phi_x\Phi_y}{a^2}\Phi_{xy} - \frac{2\Phi_x\Phi_z}{a^2}\Phi_{xz} - \frac{2\Phi_y\Phi_z}{a^2}\Phi_{yz} = 0 \quad Eqn 1$$

This equation can be manipulated into three different classes of PDEs: for subsonic flow equation 1 becomes an elliptic PDE, sonic flow it turns into a parabolic PDE, and for supersonic flow the velocity potential equation becomes a hyperbolic PDE. The complex methods required to solve the different PDE classifications represents why the MOC can only be applied to supersonic flows<sup>26</sup>. It should suffice to say that, of the three classes, only the hyperbolic PDE has a generalized process that can be applied to solve supersonic flows.

The irrotational MOC solves the potential flow equation and allows all fluid properties to be calculated at discrete locations. The method finds characteristic lines in the flow field and calculates flow variables at the intersections of these lines. The crossing characteristic lines which define the flow field are known as a characteristic net.

### 3.2 Initial Data Line

To start the MOC an initial data line must be created in which all properties are known. For rocket flow analysis, this line is usually located at the throat of the converging-diverging nozzle. A different numerical method is employed to calculate the exact flow variables and locations of the points on the initial data line. ARES uses Sauer's method which is a simple, closed form solution for the flow field in the nozzle throat region. Figure 14 shows a comparison of Sauer's method with that of Hall and Kleigel using the Geometrical Contraction Factor,  $C_e$ . The value of  $C_e$  measures the reduction in mass flow rate due to 2-D flow effects near the throat<sup>10</sup>. It is evident that this method diverges from the experimental data as  $\frac{\rho_t}{y_t}$  decreases. The trend creates a requirement that  $\frac{\rho_t}{y_t} \geq 2$  which is fairly common for rockets because it reduces the required nozzle length. Sauer's method is used to solve for the line where the vertical component of velocity is equal to zero. The calculated points are slightly downstream of the sonic line and curve upstream as it approaches the wall. The parabolic shape of the initial data line occurs due to the multidimensionality of the problem and encourages more accurate flow field analysis from the MOC.

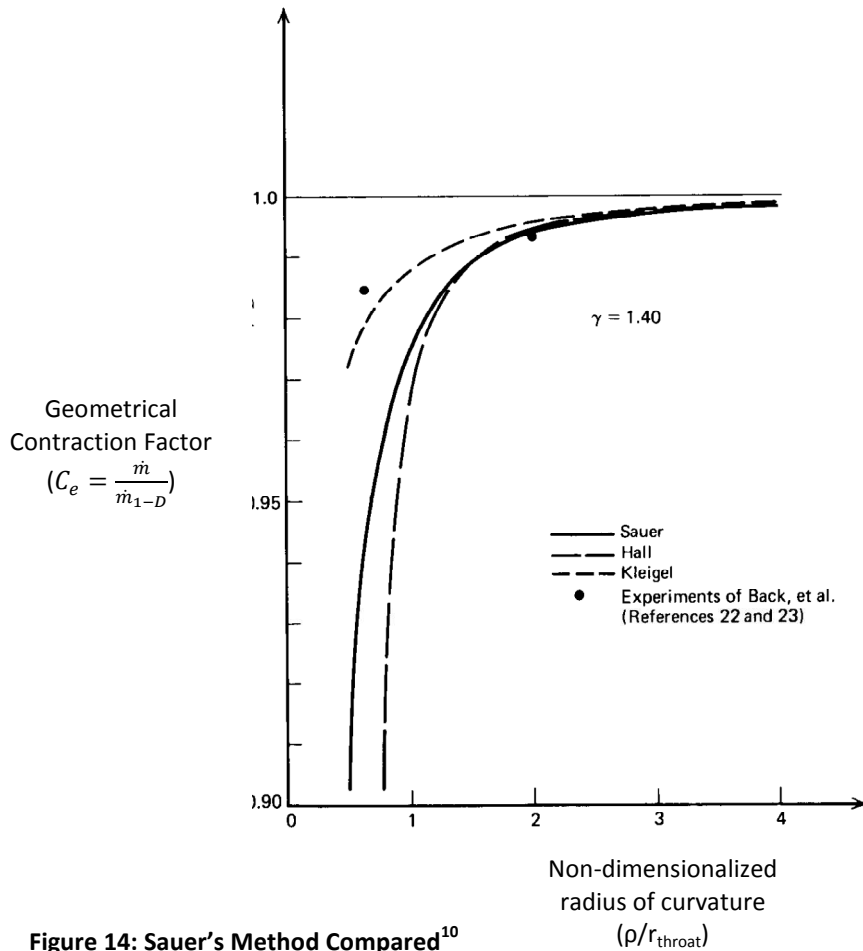


Figure 14: Sauer's Method Compared<sup>10</sup>

### 3.3 Characteristic and Compatibility Equations

Once the initial data line has been approximated the MOC can begin. The characteristic equations which will be shown later are first employed to find the slopes of the characteristic lines, referred to as C- and C+, emanating from each point on the initial data line. These characteristic equations solve for the slope by adding together the flow and Mach angle of the supersonic flow. It can be proven through the derivation of the MOC that along these lines the flow variables are continuous, but their derivatives are



indeterminate and sometimes even discontinuous. The compatibility equations, which solve for the flow variables, can then be created such that they only hold true along these characteristic lines.

The characteristic and compatibility equations are formed by various numerical manipulations of the velocity potential equation. Through the Method of Characteristics, the complex velocity potential PDE is simplified to ODEs along the characteristic lines. The resulting simplified equations allow a solution to the flow field to be formed by calculating fluid properties at the discrete points found from the crisscrossing characteristic lines.

The difference between planar and axisymmetric MOC solutions begins with the derivation of the characteristic and compatibility equations. The planar MOC deals exclusively in the Cartesian coordinate system while the axisymmetric flow field uses the cylindrical coordinate system to formulate the equations of interest. The characteristic and compatibility equations for the 2-D Planar and Axisymmetric MOC are shown in Table 1. It is interesting to note that besides the coordinate variables changing, the characteristic equations are identical.

Equations	2-D Planar	Axisymmetric
Characteristic	$\left(\frac{dy}{dx}\right)_{char} = \tan(\theta \mp \mu)$	$\left(\frac{dr}{dx}\right)_{char} = \tan(\theta \mp \mu)$
Compatibility	$d\theta = \mp \sqrt{M^2 - 1} \frac{dV}{V}$	$d(\theta \pm \nu) = \pm \frac{1}{\sqrt{M^2 - 1} - \cot \theta} \frac{dr}{r}$

**Table 1: MOC Equations of Interest**

However, the compatibility equations have a fair number of dissimilarities which affect the flow field calculation procedure. The compatibility equations for the planar case show that the flow variables are entirely independent of their location in the flow. The equations can then be reduced to simple algebraic formulas through the use of the Prandtl-Meyer function as shown in equations 2,3.

$$\theta \pm \nu(M) = K_{\mp} \quad \text{Eqns 2,3}$$

Here  $\nu$  is the Prandtl-Meyer function and  $K_{\mp}$  is a constant which holds true along a particular characteristic line. On the other hand, the axisymmetric case contains a  $\frac{dr}{r}$  term which signifies the compatibility equations' dependence on the flow field location. The computations require finite differences to be combined with the MOC procedure to solve the compatibility equations; this increases the complexity of the calculation.

Using the two sets of equations presented above, a rocket flow field can be divided up into different sections for the MOC calculation. The use of various boundary conditions allows the program to alter the calculation structure when necessary. The procedures are classified under the title of Unit Processes and will be demonstrated for the planar

case. Though ARES handles both 2-D planar and axisymmetric flows, computations for the planar case are less intensive and generally related to the axisymmetric case.

### **3.4 Unit Processes**

The first and most employed unit process is the interior point calculation. The algorithm is employed when the point of interest lies in free space with no physical boundary condition and is shown in Figure 15. The location and fluid properties at points 1 and 2 are known and will be used to find all values at point 4. The first step is to use the characteristic equations presented in Table 1 to solve for the location of point 4. As shown in the diagram, a C- line is utilized from point 1 and a C+ line from point 2. Their intersection marks the coordinates of point 4 and where the two corresponding compatibility equations are both valid at. Since Mach and theta are known at points 1 and 2 the compatibility equations can be applied to find the constants  $K^-$  and  $K^+$  that hold true along each respective characteristic line. Now there are two equations and two unknowns at point 4 and can be solved by simple substitution. All other fluid property values can be calculated with Mach and theta at point 4 with the isentropic relations.

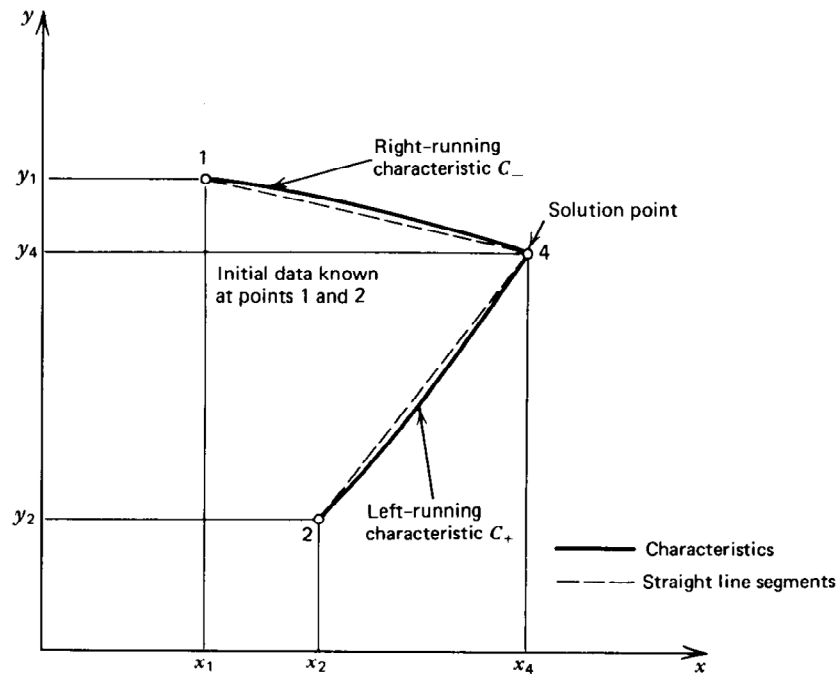
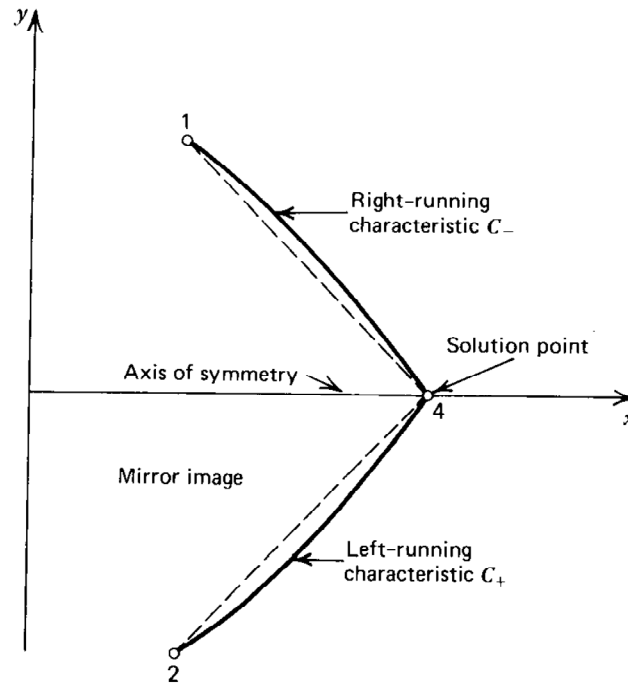


Figure 15: Interior Point Unit Process<sup>10</sup>

The next unit process is for an axis of symmetry point and is shown in Figure 16(10). Point 4 is now located at the rocket axis which means there is no physical point 2 if symmetry is applied. In the figure below all properties for point 1 are mirrored in point 2. Therefore there is only one unique characteristic line and corresponding compatibility equation which both issue from point 1. The MOC now gives only two equations for the four unknowns. In order to solve, boundary conditions must be applied. Since ARES will be employing symmetry to reduce the number of computations, the vertical location of point 4 must be at zero. The characteristic equation from point 1 need only be solved now for one coordinate. Furthermore, the streamline angle must also be zero because the nozzle is assumed to be symmetric and generates purely axial flow at the centerline.

The flow angle boundary condition reduces the amount of unknown variables to two which allows a solution to be formed by substitution.



**Figure 16: Axis of Symmetry Unit Process**

The wall point unit process is similar to the centerline calculations in that it employs two boundary conditions as well. While the axis of symmetry case did not use a  $C+$  line because there was no unique point 2, a wall point does not have a  $C-$  line because there is no point 1 as shown in Figure 17(10). Once again, boundary conditions must be applied to solve for the wall point. In this case the wall contour is known which can be used with the characteristic equation of point 2 to solve for the location of point 4. Since the wall is being described with a 2<sup>nd</sup> order polynomial, it can be differentiated to give the wall slope. It is assumed that the flow angle at the wall is equal to the wall slope which solves a second unknown. Similar to the centerline case, there are now two

unknowns left, the axial coordinate and Mach number, which can be solved with the characteristic and compatibility equation originating from point 2.

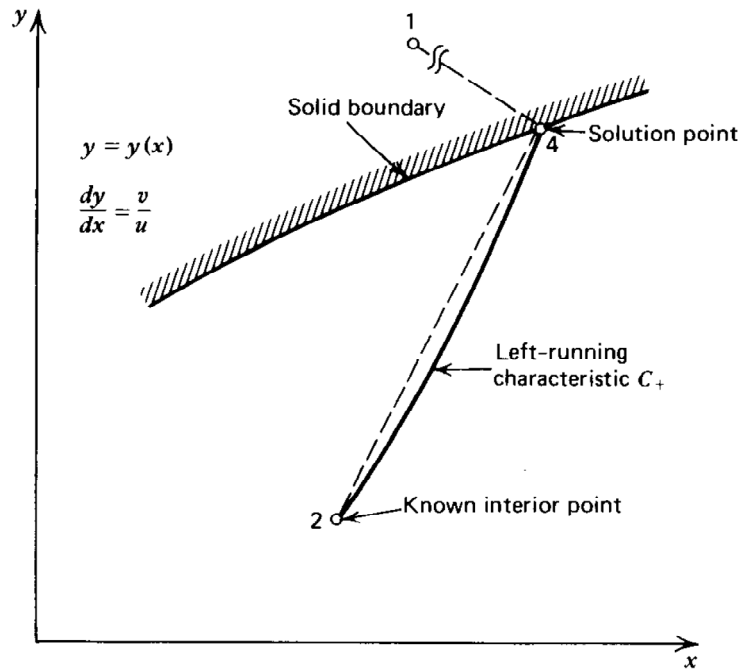


Figure 17: Direct Wall Point Unit Process

An offshoot of the wall point unit process that is also sometimes used is the inverse wall point method. This method allows for the user to choose the location of the next wall point at the cost of employing another iterative algorithm. As demonstrated in Figure 18, a new point, designated 2, is assumed between points 1 and 3 which should create a C+ line that intersects the wall at the desired location. The initial guess for the new point does not create point 4 at the correct location and is iterated between points 1 and 3 until it is found. The inverse wall point method adds quite a bit of complexity but also allows the user to select areas of interest on the wall. One such area would be the circular-arc throat of the rocket nozzle. This area often needs a much denser mesh than

the MOC provides with the wall point calculation because it greatly affects the downstream flow. Without applying the inverse wall point method a gap in the characteristic net would occur after the circular-arc section ends. The inverse wall point method is also used to create the final wall point at the lip of the nozzle. Assigning a point to this location is useful as it marks the last C- line before a Prandtl-Meyer expansion fan or plume boundary unit process occurs.

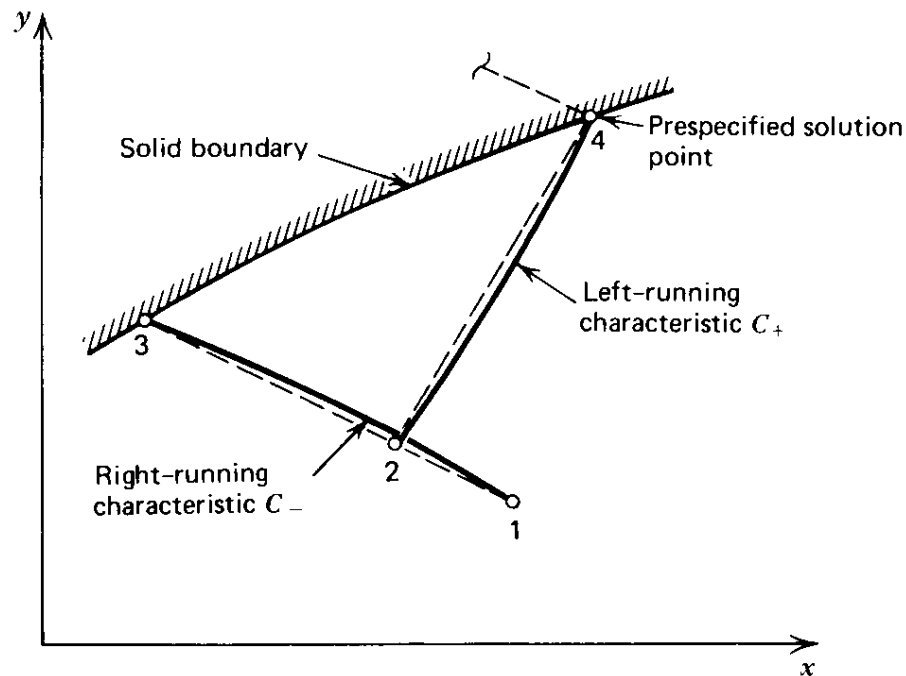


Figure 18: Inverse Wall Point Unit Process<sup>10</sup>

The plume boundary calculation is the last of the unit processes employed by ARES. The process is used to find the flow variables at each point along a jet boundary with a known static pressure. In order to satisfy steady-state equilibrium conditions, the static pressure of the plume boundary must be equal to the ambient flow's static pressure.

Enacting this condition ensures that the jet boundary is not encountering transient static pressure values. The static pressure of the plume boundary must then become an input to the MOC. The isentropic relations along with the stagnation pressure can then be used to determine the required Mach number. Similarly to the wall point calculation, there is only one  $C_+$  characteristic line which gives two equations to solve for the 3 remaining unknown variables:  $x$ ,  $y$ , and  $\theta$ . One more boundary condition is required and comes from the fact that the flow is a streamline at the boundary. As shown in Figure 19, the flow angle from point 3 is used with the characteristic line from point 2 to solve for the location of point 4(10). As with all the other unit processes, maintaining a small step size between points which lie on the same characteristic line is key to higher accuracy. The compatibility equation is then used to solve for the flow angle at point 4.

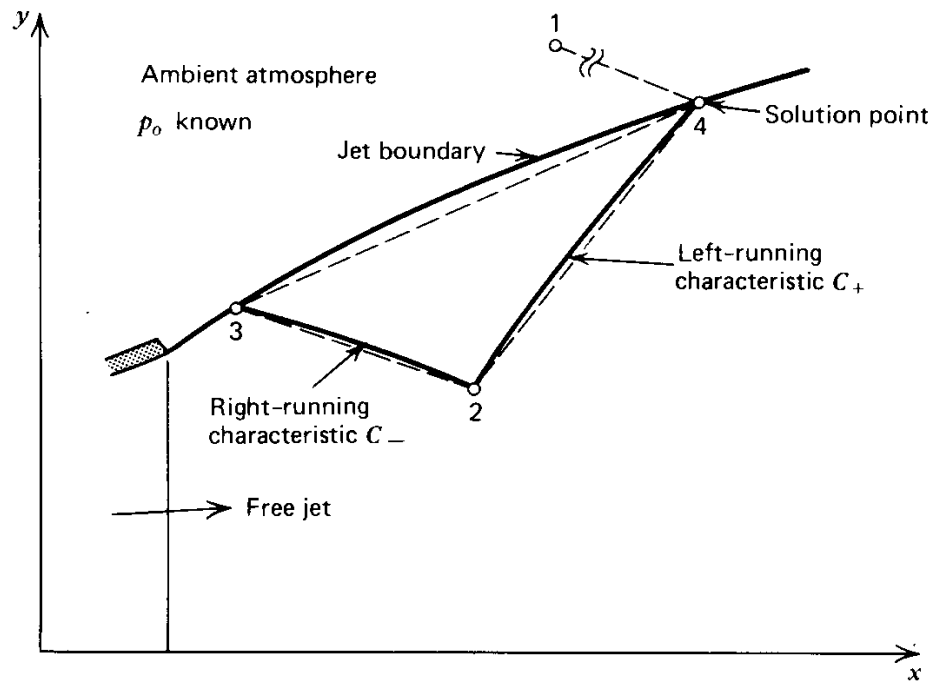


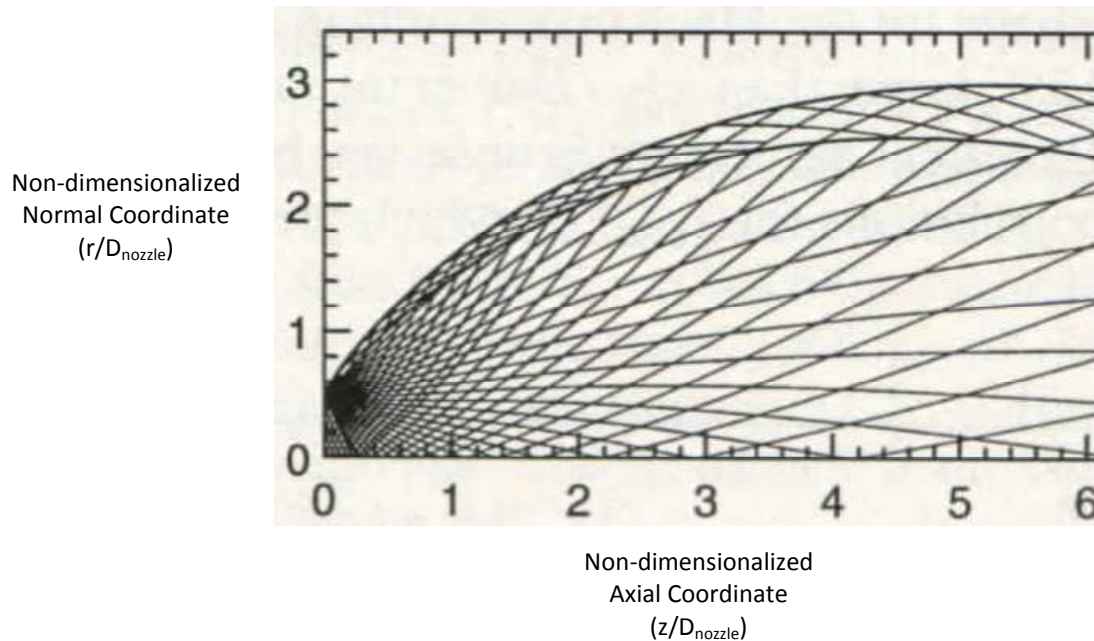
Figure 19: Free Pressure Boundary Point Unit Process



### 3.5 Conceptualizing the MOC

At every wall, inverse wall, and plume boundary point a new C- line is started by the MOC and runs diagonally downstream into the flow. At every centerline point that employs symmetry a new C+ line is started which also runs diagonally downstream into the flow. While the MOC may seem very similar to a purely finite difference approach because discrete points are analyzed, it is fundamentally different in that the characteristic lines which form the mesh do exist in the real flow field. These lines where the complicated non-linear PDE in question simplifies to ODEs are Mach lines when the MOC is applied to solve a supersonic flow field. The resulting conceptual understanding leads to intuitive flow field analysis simply by observing the mesh patterns created by the characteristic net. Two such instances are the various expansion and compression waves located within the various flow phenomena. It can be seen in the characteristic net of a rocket nozzle that the mesh becomes less dense as one moves downstream of the throat as shown in Figure 20(14). Here a flow is exhausting from a sonic nozzle and moves from left to right. As it expands and speeds up, the particles within the plume spread away from each other.

Conversely a coalescence of the characteristic lines indicates a section where the flow is compressing and slowing down. In extreme cases of compression, the characteristic lines may fold over onto themselves as discussed by Love<sup>13</sup>. Such a folding could never occur in a real flow field and happens in the simulation because the MOC does not



**Figure 20: Example of a spreading characteristic net due to expanding flow**

recognize that two physical points occupying the same location is impossible. The folding of characteristics signifies a location of rapid compression and represents a shockwave. The nonphysical solution is usually allowable as it still produces reasonable results and can be later by manually moving or deleting the characteristic lines. This post-processing technique generates a much more realistic solution because the coalescence of characteristics can still be observed but logical violations no longer occur.

#### 4. Method of Characteristics Implementation

It is important to note that ARES uses a slightly altered form of the compatibility equations and are shown in equations 4, 5.

$$(u^2 - a^2)du_{\pm} + [2uv - (u^2 - v^2)\lambda_{\pm}]dv_{\pm} - \left(\frac{\delta a^2 v}{r}\right)dx_{\pm} = 0 \quad Eqns\ 4,5$$

The only difference between these equations and the ones presented in the axisymmetric column of Table 1 is that they are a function of  $u$  and  $v$ , the axial and vertical components of velocity, instead of Mach and flow angle. Mach and flow angle were the variables employed to help describe the MOC because they allowed for a clearer conceptual understanding. ARES utilizes equations 4, 5 because they account for whether the flow is axisymmetric or planar simply from the value of  $\delta$ . Also the source<sup>10</sup> employed to form ARES used  $u$  and  $v$  as the unknown flow variables and so it made sense to do likewise for code validation.

The MOC was described in the previous section with different calculation procedures called unit processes for the simple 2-D planar flow case. Though the axisymmetric case follows the same general procedure, special care must be taken when solving the compatibility equations. Unlike the planar case, these equations contain a term which involves not only the points' spatial coordinates but also some derivatives of these coordinates. The compatibility equations can no longer be simplified to algebraic equations and must now use finite differences to be solved. Unfortunately finite differences can greatly increase error if left unrefined and allowed to propagate. To

counter this, ARES uses a modified Euler predictor-corrector algorithm with iteration to converge each finite difference solution and create more accurate results.

A predictor-corrector algorithm is a two part method which first predicts the answer and then corrects it. The predictor step uses an algorithm of a specific order to find the temporary solution. The corrector then uses a higher order of that algorithm to enhance the accuracy of the predictor's solution. A modified Euler predictor-corrector algorithm uses Euler's method as the specific algorithm and is shown in equation 6.

$$y_{n+1} = y_n + hf(x_n, y_n) \quad \text{Eqn 6}$$

Equation 6 comes from a Taylor series expansion of order 1 and is the general method used for the predictor step. The corrector algorithm is found by taking the derivative and expanding it in another Taylor series of the first order which gives equation 7.

$$y(x_n + h) = y(x_n) + \frac{h}{2}\{f(x_n, y_n) + f[x_n + h, y(x_n + h)]\} \quad \text{Eqn 7}$$

These two equations can be used systematically to increase the accuracy of a numerical solution. Placing these equations in an iterative loop based on convergence further increases the accuracy and is why the Euler predictor-corrector algorithm with iteration is utilized in ARES.

## 5. Combustion

ARES supports two separate methods for inputting the primary fluid property initial conditions. The first method allows the user to select a combustion pressure, combustion temperature, specific heat ratio, and specific gas constant. These, along with the other standard inputs, provide the MOC function with enough known variables to begin calculation of the primary flow. While this method allows for a wide range of values, it requires the user to have prior knowledge of the required inputs. Calculations of combustion temperature, specific heat ratio, and specific gas constant can be tedious and must be repeated for every configuration. In order to reduce time spent on needless calculations, another method has been formed and should be utilized whenever conditions allow.

The simplest method for inputting the required combustion properties is only applicable if a supporting propellant is chosen. ARES currently allows the user to select one of five options: methanol-air, methanol-oxygen, ethanol-air, ethanol-oxygen, and hydrazine-dinitrogen tetroxide. The propellant input pressure and mixture ratio must also be selected for the combustion subroutine to begin. The input temperature of the fuel and oxidizer is assumed to be at 293.15 Kelvin or 68° Fahrenheit. The input combustion pressures available are between 100 and 1200 psi and the oxidizer to fuel ratio has a range which is centered on the stoichiometric mixture ratio for each propellant. The user selects values for both of these parameters and ARES interpolates between data tables created by the Chemical Equilibrium and Applications (CEA) program.

The CEA program was developed by NASA<sup>27</sup> and is capable of calculating a wide range of theoretical thermodynamic properties to solve a variety of chemical engineering problems. Most useful to ARES was its ability to solve finite area combustion chambers. The solution is based on a minimization-of-free-energy formulation and utilizes Gibbs energy due to the required pressure and temperature input variables.

The CEA program contributed to ARES by solving for the combustion temperature, specific heat ratio, and specific gas constant at each combination of pressures and mixture ratios. The data generated formed a comprehensive index which was entered into numerous data tables divided up by propellant combination. These tables were placed within ARES as a combustion data function for quick look-up. Once the user specifies the required inputs, ARES interpolates between the various matrices to determine the thermodynamic conditions exiting the combustion chamber.

## 6. Performance Analysis

Though there are a number of metrics used for quantifying performance of rocket-ejectors, this research utilizes only two: Thrust Augmentation and Specific Impulse. Both of these metrics are easily discernible from the data presented in other works and offer a high-level view of the ejector at hand.

### 6.1 Thrust Augmentation

Though the rocket thrust calculation is generally straightforward, the inclusion of an ejector creates a more complex situation. Even in the ideal case many assumptions must be made so that an equation for thrust can be found. The equation used in ARES was derived by Kinzie<sup>28</sup> and used conservation principles with a control volume analysis to find the closed form thrust prediction shown in equation 8.

$$T = u_e \dot{m}_{mix} - u_s \dot{m}_s + (P_e - P_a)A_e \quad Eqn\ 8$$

The first term in equation 8 is the propulsive force due to the mixed rocket/ejector flows. The middle expression is recognized as ram drag and increases with entrainment. The last term in equation 8 offers adjustment to the net thrust if the mixed flow has a static pressure other than that of ambient conditions. This equation primarily uses fluid properties as opposed to the apparatus geometry. If the flow field calculations account for the various flow phenomena, fluid properties can act as a more accurate method for determining performance.

In most experiments, the secondary mass flow rate and mixed flow exit velocity are difficult to measure<sup>28</sup>. For this reason Johnson employed two equations derived by Presz<sup>29</sup> to find a theoretical thrust augmentation that could be compared to his experimental results. These algorithms, shown in equations 9 and 10, mainly use the geometry of the system to predict the thrust augmentation.

$$\phi = \frac{\left(\frac{\dot{m}_s}{\dot{m}_p} + 1\right)^2}{\left(\frac{A_s}{A_p} + 1\right) \left[1 - \left(\frac{\dot{m}_s^2}{\dot{m}_p^2}\right) \left(\frac{A_p^2}{A_s^2}\right)\right]^{\frac{1}{2}}} \quad \text{Eqn 9}$$

$$\frac{\dot{m}_s^2}{\dot{m}_p^2} \left(1 + \left(\frac{A_p}{A_s}\right)^2\right) + 4 \frac{\dot{m}_s}{\dot{m}_p} - 2 \left(\frac{A_s}{A_p}\right) = 0 \quad \text{Eqn 10}$$

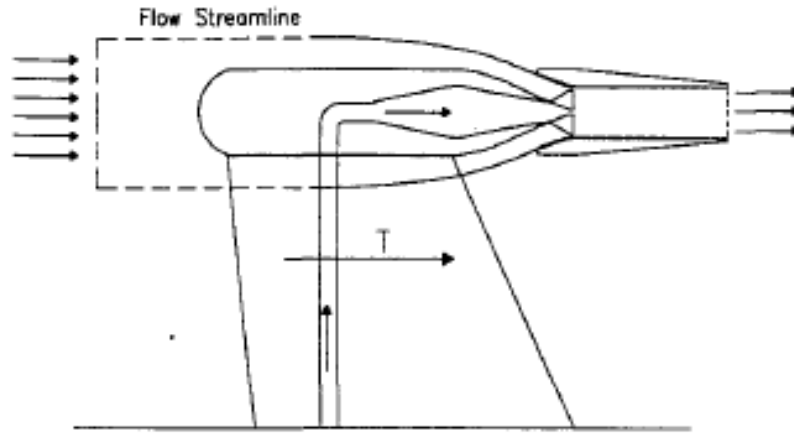
Since these equations require ideal mixing along with ideal primary and secondary flow assumptions, they cannot accurately account for a number of the flow phenomena present. Furthermore, the control volume utilized allows for only a straight ejector configuration which limits its applicability.

Fortunately, a simulation is not bound by the same constraints that required Johnson to compromise with the theoretical thrust calculation. ARES uses reasonable assumptions along with the calculated flow field properties to predict the secondary mass flow rate and mixed flow velocity. In many configurations, the secondary flow calculated by ARES becomes supersonic due to the constricting plume exhausting from the primary nozzle. The virtual throat creates a choke point which allows for the calculation of the secondary mass flow rate. To find the mixed flow velocity, all losses must first be



neglected in the secondary flow. The assumption eliminates effects due to weak shocks and friction and is expected to cause a higher velocity calculation. The primary and secondary flows are then combined through the use of the conservation of momentum to determine the mixed velocity.

Kinzie's derivation uses the control volume shown in Figure 21 as well as the conservation of mass and momentum. The main assumption that must be made for this equation is that the exit flow is fully mixed and at thermal equilibrium.



**Figure 21: Control Volume for Kinzie's Thrust Equation<sup>28</sup>**

While a thrust equation is useful for performance measurement, it cannot be easily compared to other configurations. Kinzie's equation is then non-dimensionalized by dividing the total thrust by the ideal primary fluid thrust. The combined terms are shown in equation 11 which is referred to as thrust augmentation.

$$\phi = \frac{T_{tot}}{T_p} = \frac{u_e \dot{m}_{mix} - u_s \dot{m}_s + (P_e - P_a)A_e}{u_p \dot{m}_p} \quad Eqn 11$$

## 6.2 Specific Impulse

A rocket's performance is most commonly measured with  $I_{sp}$  and is shown in equation 12. Conceptually speaking, specific impulse displays the amount of force a system receives per unit of propellant expelled and is universal among all rocket configurations.

$$I_{sp} = \frac{T_{tot}}{\dot{m}_p g} \quad Eqn 12$$

While this variable will be utilized for comparisons, the more important value will be the change in  $I_{sp}$  due to the ejector. Essentially, this value represents the performance that the ejector alone imparts on the system. The ejector  $I_{sp}$  will be calculated by replacing the  $T_{tot}$  term with  $T_{ejector}$  in the previous equation. The  $T_{ejector}$  term can be found in ARES easily because the two streams are separate due to the inviscid calculation.

## 6.3 Initial Effects

There are two main fluid properties which can affect thrust and  $I_{sp}$  greatly. The first is combustion pressure which has an impact on both performance metrics. Differences in pressure ratio can create the three modes described in section 2.1.1, Saturated Flow, Fabri Choke, and Fabri Block. Depending on which case is present, the secondary flow is accelerated or decelerated which alters the mixed velocity component in equation 8, affecting thrust. The velocity change would then get filtered down through equation 12 and affect  $I_{sp}$ . The other independent variable, mixture ratio, affects performance by altering the primary mass flow rate. Mixture ratio increases the stagnation temperature inside the combustion chamber as it approaches the stoichiometric ratio. In turn, the

temperature difference causes a decrease in the mass flow rate as demonstrated in equation 13.

$$\dot{m} = \frac{A^* P_0}{\sqrt{T_0}} \sqrt{\frac{\gamma}{R}} \left( \frac{\gamma + 1}{2} \right)^{-\frac{\gamma+1}{2(\gamma-1)}} \quad Eqn 13$$

Interestingly enough, the speed of sound is identically altered except that it is directly proportional as shown in equation 14. The result is that calculations which multiply mass flow rate with velocity cancel out most affects due to mixture ratio. The ensuing tolerance to mixture ratio also means that the thrust values will only be slightly affected due to the change in pressure caused by a different velocity. On the other hand, the  $I_{sp}$  values will be significantly altered because there is a lone  $\dot{m}_{prop}$  value in the denominator of equation 12.

$$V^* = \sqrt{\frac{\gamma R T_0}{1 + \frac{\gamma - 1}{2}}} \quad Eqn 14$$

## **7. ARES' Process**

ARES begins with user inputs for the fluid properties and geometric dimensions of the desired rocket-ejector. The simulation allows the user to choose the fuel and oxidizer of the propellant. The possible combinations include methanol-air, methanol-oxygen, ethanol-air, ethanol-oxygen, and hydrazine-dinitrogen tetroxide. The user selects which combination they want and then chooses the combustion pressure as well as the oxidizer to fuel ratio, as described in section 5.

After the values for fluid properties and geometric dimensions have been selected or interpolated, the program proceeds to calculate the primary nozzle flow. The thermodynamic model employed by Zucrow & Hoffman and this simulation includes a calorically perfect gas assumption<sup>4</sup>. A non-reacting flow in thermodynamic equilibrium with constant specific heat has its greatest accuracy when calculating a fully combusted fluid. ARES utilizes the irrotational MOC to compute the flow field from the throat of the nozzle to the first Mach Disk and corresponding final C+ line emanating from the triple point. To save on computation time, ARES employs symmetry so that only half of the flow field needs to be solved. The method is begun by calculating an initial data line at the nozzle throat in which all the points are slightly above Mach 1. As described in section 2, this is because the MOC can only be applied to supersonic flow. Sauer's method is used to find the initial data line points along the throat in which the vertical component of velocity is equal to zero. The result is a curved initial data line with the

point furthest downstream on the axis of symmetry. The primary nozzle function can then begin the MOC computation of the flow field inside the nozzle.

The nozzle wall is comprised of two sections: the circular arc throat and the diverging section. Both sections of the nozzle wall are predefined by user inputs. The MOC utilizes the inverse wall point unit process for the circular arc wall section to ensure a dense mesh at the beginning of the nozzle. Mesh density is important because the resulting characteristic net will become sparser as the flow expands. The circular arc throat ends once its slope and the starting slope of the diverging section become tangent. Aligning the sections in this way is done so that the oblique shock emanating from the wall transition point is minimized. Only a slight coalescence of the resulting compression waves will then be observed. The diverging section of the nozzle is calculated from three geometric values which the user has specified. The three parameters are the starting and ending angles of the diverging section and the overall nozzle length. The employed algorithm allows ARES to calculate flow fields for a range of nozzles from parabolic to conical contours. Though they are ideal, isentropic nozzle are not included in the spectrum because they are complicated and expensive to make for experiments. The flow field continues its calculations until the next wall point found lies beyond the length of the nozzle. At this point the MOC erases the outlying wall point and instead uses the inverse wall point method to create a final characteristic line issuing from the nozzle lip.

Once the flow field has been calculated through the exit of the nozzle, an indicator is set to mark the location in the data matrices. The indicator is used in sequential iterations to mark where the MOC should start when the guess for the pressure distribution on the jet boundary changes. As this is a supersonic flow, changes downstream cannot affect flow values upstream. The only input value changing in the next iteration of the MOC will be the pressure distribution along the plume boundary. Having identical inputs to the nozzle flow field means that the properties within the nozzle are set and can be retained for future iterations; this saves a relatively large amount of computation time.

Next ARES proceeds to calculate the plume flow field based on the pressure distribution it received. The initial guess for the static pressure distribution is set to range from the stagnation pressure of the secondary flow to the stagnation pressure divided by 1.01. The guess predicates from the secondary flow beginning at zero velocity and allows the program to develop an initial analysis of the flow field to be iterated. It is important to note that the logic ARES operates under does not allow for a constant initial pressure distribution guess; hence it is divided by 1.01. The primary plume calculation begins with a determination of whether the primary flow is underexpanded or overexpanded. If the flow is underexpanded then the calculations are diverted to a function which calculates an expansion fan at the lip using the Prandtl-Meyer function in equation 15; this equalizes the primary static pressure with the secondary static pressure which is a plume boundary condition.

$$v(M) = \sqrt{\left(\frac{\gamma + 1}{\gamma - 1}\right)} \tan^{-1} \sqrt{\frac{\gamma - 1}{\gamma + 1} (M^2 - 1)} - \tan^{-1} \sqrt{M^2 - 1} \quad \text{Eqn 15}$$

If the flow is slightly overexpanded, then the MOC continues its calculations without an expansion fan. Though the MOC cannot handle completely non-isentropic flow, it can handle some slight crossing of characteristics as long as the flow remains supersonic. The described conditions occur with the analysis of a slightly overexpanded nozzle due to the resultant weak oblique shockwave. The characteristics eventually uncross themselves as the computations continue downstream resulting in a fold in the flow field fabric as shown in Figure 22. Allowing the characteristics to cross is nonphysical but also permits the MOC to continue calculations that have relatively good results for the proceeding flow field<sup>10</sup>. Post-processing the computed data can remove the crossed characteristics and create a more realistic looking flow field.

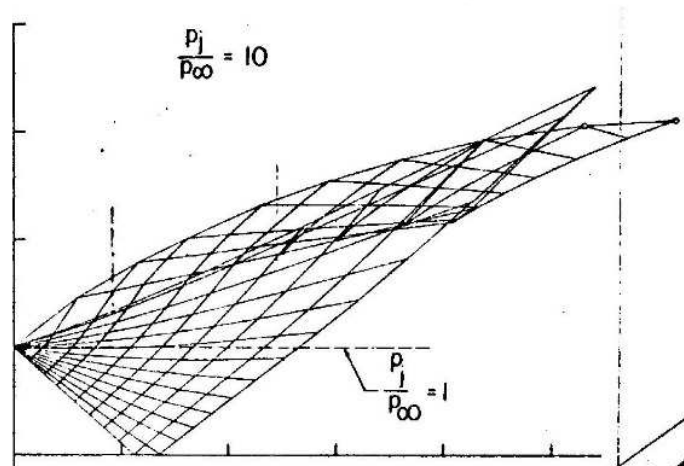


Figure 22: Example of a folded characteristic net due to a shockwave<sup>20</sup>

Since the point of attaching an ejector to a rocket is to entrain the flow and produce more thrust, it is undesirable to have an overexpanded flow enter the ejector. As such, the remainder of ARES' program description will be based on an underexpanded flow assumption.

Once the Prandtl-Meyer expansion fan has been calculated the MOC continues to calculate the flow within the primary plume. The static pressure distribution is used to find the Mach and angle of the flow along the plume boundary. These calculations continue until the first Mach Disk is found.

The Mach Disk location is approximated from the assumption that the flow after it must have a static pressure that is equal to the static pressure of the secondary flow at the same axial position<sup>20</sup>. ARES finds this location by putting every point along the last characteristic of the expansion fan through a normal shockwave calculation. It then checks which has a static pressure after the normal shock equal to the static pressure of the secondary flow. Since it is rare for a point's static pressure to exactly match that of the secondary flow, the location of the Mach Disk is usually interpolated between discrete points in the characteristic net. Since the Mach Disk only encompasses a portion of the center of the plume, the MOC can continue to calculate the flow field above the Mach Disk until the final C+ characteristic emanating from the triple point described in section 2.2.3 intersects the jet boundary. The solution of the final plume point concludes the method of characteristics portion of the first iteration solution. The



calculated values undergo some post-processing and are then routed to the function which recalculates the secondary flow's static pressure distribution.

The secondary flow is based off a 1-D analysis in which the area between the plume and the duct wall dictates the new pressure. Using an isentropic analysis means that only the location of the primary plume boundary points must be analyzed to calculate a new pressure distribution. Every point along the boundary of the primary plume divides the secondary flow up into discrete segments; this will be shown in the results section. Since a 1-D analysis is used, the static pressure along the boundary of the primary plume must match that of the secondary flow at the same axial location. Initially, the minimum area in the downstream direction is found and has its corresponding secondary flow tested for  $M = 1$ . If the static pressure is not low enough to induce  $M = 1$  at that location, then the entire secondary flow will remain subsonic and the new pressure distribution will reflect those values. However, if  $M > 1$  then the minimum area flow will have its pressure altered so that  $M = 1$  instead; this is the Fabri choke location discussed in section 2.1.1. A sonic value is set in this instance because the decreasing area cannot produce supersonic secondary flow. Instead, increasing the pressure at that location will encourage the secondary throat location to move upstream until the static pressures match, thus finding the correct location of the throat. Supersonic flow would then be calculated for the remaining sections in the secondary duct. If the primary to secondary stagnation pressure ratio is too high then the primary plume may create the Fabri block condition. As the name implies, this

condition is highly inefficient as it inhibits secondary flow movement through the duct and creates extra drag with no added thrust.

In the case of a low primary to secondary stagnation pressure ratio, the plume of the primary exhaust may not reach the secondary flow. This circumstance is described by Fabri as the separated flow case and may show the entire secondary flow at Mach 1. A sonic secondary flow occurs because the minimum area, and therefore the throat, is located at the start of secondary duct. If the initial static pressure from the plume is low enough, then the secondary flow will meet the criteria necessary for sonic speeds. A constant velocity secondary flow may sometimes be less efficient than the Fabri choke condition because, apart from viscous interactions, there is no acceleration and thus no thrust produced by the ejector. Such a situation may also arise due to a number of other inputs including but not limited to wall exit angle, lip thickness of the nozzle, and fluid properties.

ARES will also find that the saturated case has a larger stagnation zone beginning from the lip of the nozzle than it would in any other case. In reality, this stagnation zone would be much shorter than what will be shown in the flow visualization since the viscous mixing layer would form and produce an increased velocity in the secondary flow. However, ARES assumes inviscid flow and so substitutes a larger stagnation zone for the mixing layer.

ARES checks for convergence after the new pressure distribution has been created for the boundary of the primary plume by comparing it to the previous pressure distribution. Convergence is checked by taking the norm of the two vectors and finding the difference between them. A relaxation factor is then added to the new pressure distribution to slow down the convergence of ARES. This is necessary as the code is unstable and will error if left unrelaxed<sup>8</sup>.

Once the convergence tolerance has been met, ARES proceeds to plot the final flow fields and displays the thrust augmentation and specific impulse values produced by both the primary and secondary flows.

## **8. Testing Configurations**

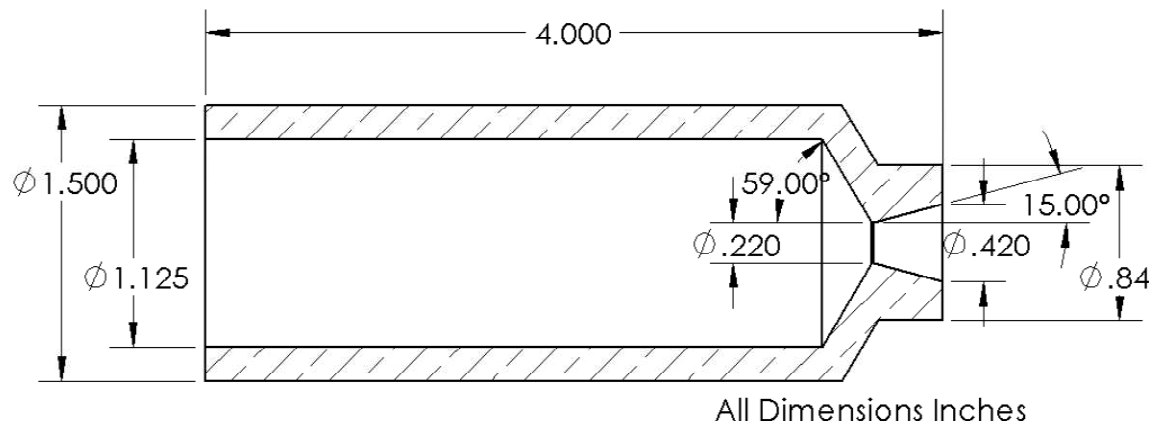
ARES has the ability to simulate variations to the primary flow properties, secondary flow properties, rocket nozzle geometry, and ejector geometry. The program then measures the corresponding performance while also providing a numerical visualization of the calculated flow field. Due to the sheer number of input combinations possible, the current research is first focused on the validation of ARES, followed by a study on how duct divergence angle affects rocket-ejector performance.

### **8.1 Validation**

The first form of validation will compare the outputs created by the various unit processes utilized in ARES. The subsequent section will provide flow visualizations which act as high-level validation tools and aid in conceptual understanding of the resulting flow. Zucrow & Hoffman and Eugene Love have done extensive work on modeling rocket flows with the axisymmetric MOC. Zucrow & Hoffman primarily displays characteristic nets of the initial-value problem and the nozzle flow field. Eugene Love focuses specifically on how the plume net forms for underexpanded nozzles exhausting into still air. It then becomes useful to divide the characteristic net validation section into three parts: initial-value problem, nozzle flow field, and plume flow field. The configurations employed in the previous works will be entered into ARES and have their nets compared. A final validation of the encompassing flow field will then take place to verify that all flow structures are in the same vicinity as theory predicts.

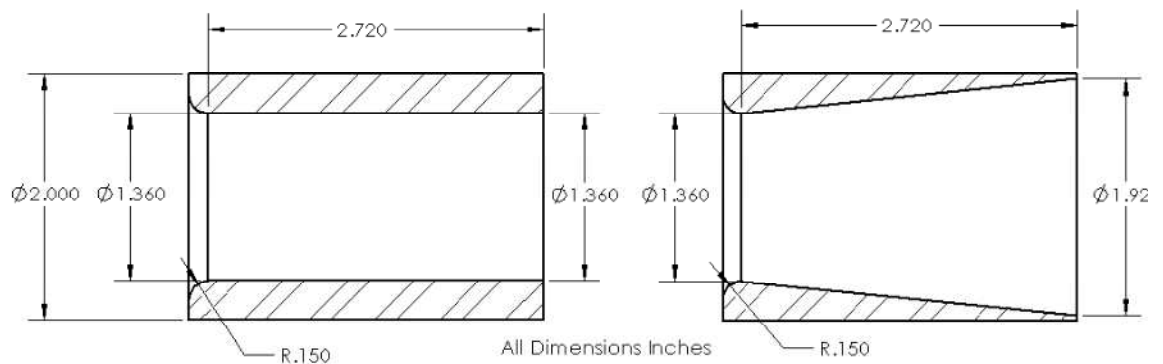
ARES will then compare a configuration employed by Kyle Johnson where an axisymmetric air augmented rocket was statically fired and examined. Johnson's experiment measured thrust and mass flow rates as described in Section 2.3. These values were then used to calculate performance characteristics such as thrust augmentation and specific impulse. The predictions prepared by ARES will be compared to Johnson's thrust, and specific impulse.

The primary rocket consisted of methanol and compressed gaseous oxygen as the fuel and oxidizer, respectively. The oxidizer to fuel mixture ratio was varied between 0.7 and 2.3 for all cases and had combustion pressures ranging between 315 and 360 psi. The corresponding stagnation pressure ratios are 21.4 to 24.5 at sea level conditions. The rocket nozzle configuration, shown in Figure 13, used a cooling jacket and an ice water heat exchanger to cool the high temperatures created by the combustion chamber. The area ratio created by the nozzle exit area and the throat is about 3.64; this corresponds to an exit Mach number of 2.55 with a specific heat ratio of 1.2. Also the shape of the nozzle is non-isentropic as it was constructed with a straight converging section and straight diverging section as shown in Figure 23. A significant aspect of the geometry is that the exit radius of the nozzle flow area is equal to the lip thickness. Though this ensured safety when firing the rocket, this also means that the configuration has a relatively large lip thickness. Lip thickness can have a substantial impact on flow entrainment because it directly affects how far the plume expands into the secondary stream and as such should affect ARES' calculations.



**Figure 23: Johnson's Combustion Chamber and Rocket Nozzle<sup>2</sup>**

The two ducts tested had straight and diverging expansion sections and are shown in Figure 24(2). The diverging ejector tapered away from the centerline at an angle of 5.88°. Both structures had identical, straight inlets so that incoming flow properties would not alter results between configurations.



**Figure 24: Johnson's Straight Ejector (Left) and Diverging Ejector (Right)**

A final comparison will then be made to experiments conducted by Trevor Foster and Ryan Gist<sup>30</sup>. The experiments focused on predicting the Fabri-choke location in a 2-D planar rocket-ejector. Foster's work utilized a hot fire Methane/GOX propellant while

Gist used a cold flow nitrogen primary exhaust. The two theses employed the same apparatus to conduct testing. An axisymmetric rocket-ejector with similar primary throat, primary exit, and ejector geometry will be loaded into ARES with the same fluid properties. The predicted choke point locations downstream of the nozzle exit will be compared with that of the experimental data.

## **8.2 Additional Testing**

Following the validation testing, ARES will be employed to create a more comprehensive study on how ejector divergence angles affect performance. All geometry settings from the validation testing will be maintained except for the ejector divergence angle and lip thickness. The chosen ejector half angles will be  $0^\circ$ ,  $5^\circ$ , and  $10^\circ$ . As the two duct configurations used in Johnson's experiment were  $0^\circ$  and  $5.88^\circ$ , the new selection of divergence angles will provide a wider range of results to draw conclusions from. The lip thickness will be decreased from 0.21 to 0.1 inches. The size reduction would encourage better mixing in a real flow and so should lead ARES to predicting more accurate results due to its mixed flow assumption. The input fluid properties will remain unchanged except for the secondary fluid properties and primary stagnation pressure. The secondary flow will utilize properties of air at 10,000ft. Half of the configurations analyzed will employ a static secondary flow and the rest will assume a Mach of 0.5. The variation of Mach will allow ARES to calculate performance differences due to ram drag. The configurations employed in the validation tests used a stagnation pressure ratio range of 21.4 to 24.5. ARES allows for a much larger range and so it will use

primary stagnation pressures between 200 and 825 psi for both studies. The corresponding stagnation pressure ratio range becomes 20.4 to 81.6 at sea level.

In order to better isolate the results, ARES will use only the optimum mixture ratio for whichever reactants it is employing. The ejector divergence angle study will use an Ethanol/GOX fueled rocket to make performance correlations. The various configurations employed in the study can be fully calculated with the use of one encompassing algorithm which changes the input values in ARES. The algorithm loops through ARES a total of 60 times so that all combinations of primary stagnation pressures and ejector divergence angles are input. Each desired performance value will then be tabulated for analysis.



## **9. Results**

The purpose of ARES is to accurately model an axisymmetric air augmented rocket for a wide variety of configurations. Validation of this simulation includes comparing individual unit processes, flow visualizations, and performance predictions. The numerical flow visualizations created by ARES will be compared to past MOC solutions and theoretical descriptions of flow structures. The performance variables inspected are mainly thrust and specific impulse; a non dimensionalized thrust calculation is not necessary because the configurations in the validation section are identical. These metrics will be isolated in order to display trends specific to the duct, nozzle, and a combination of the two. ARES is then employed to estimate performance results for 30 different configurations so that a parametric study on ejector divergence angle is produced. It was expected that the simulation would over predict the augmented thrust and  $I_{sp}$  for most configurations due to the ideal secondary flow assumption and complete mixing assumption.

### **9.1 Unit Processes Validation**

The various unit processes employed are the key components of the MOC algorithm. The main MOC function operates by calling a specific unit process subroutine at each point in the flow field. As each is employed a large number of times, it is important that the subroutines are calculating accurately.

Zucrow and Hoffman give an example point calculation for each of the 5 unit processes employed in ARES. The outputs of each of the simulations are compared in Table 2. It

can be seen that the values calculated by ARES are in good agreement with that of Zucrow. Each unit process function calculates four unknown variables which are

	Interior Point		Axis Point		Wall Intersection		Plume Boundary		Inverse Wall Intersection	
	Z&H	ARES	Z&H	ARES	Z&H	ARES	Z&H	ARES	Z&H ( $X_2/Y_2$ )	ARES
$X_4$ (m)	0.14118	0.14118	0.083308	0.083309	0.063485	0.063486	0.35283	0.35282	0.005211	0.00521
$Y_4$ (m)	0.04056	0.04055	0	0	0.063273	0.063273	0.12916	0.12915	0.026063	0.026062
$u_4$ (m/s)	2510.1	2510.1	2332.4	2332.3	1977.4	1977.4	2454.7	2454.7	1583.4	1583.4
$v_4$ (m/s)	780.2	780.2	0	0	1144.4	1144.4	532	532	738.3	738.4
<b>Maximum Error:</b>	<b>2.47E-02%</b>		<b>4.29E-03%</b>		<b>-1.58E-03%</b>		<b>7.74E-03%</b>		<b>3.84E-03%</b>	

**Table 2: Unit Process Comparison with Zucrow & Hoffman (Z&H)**

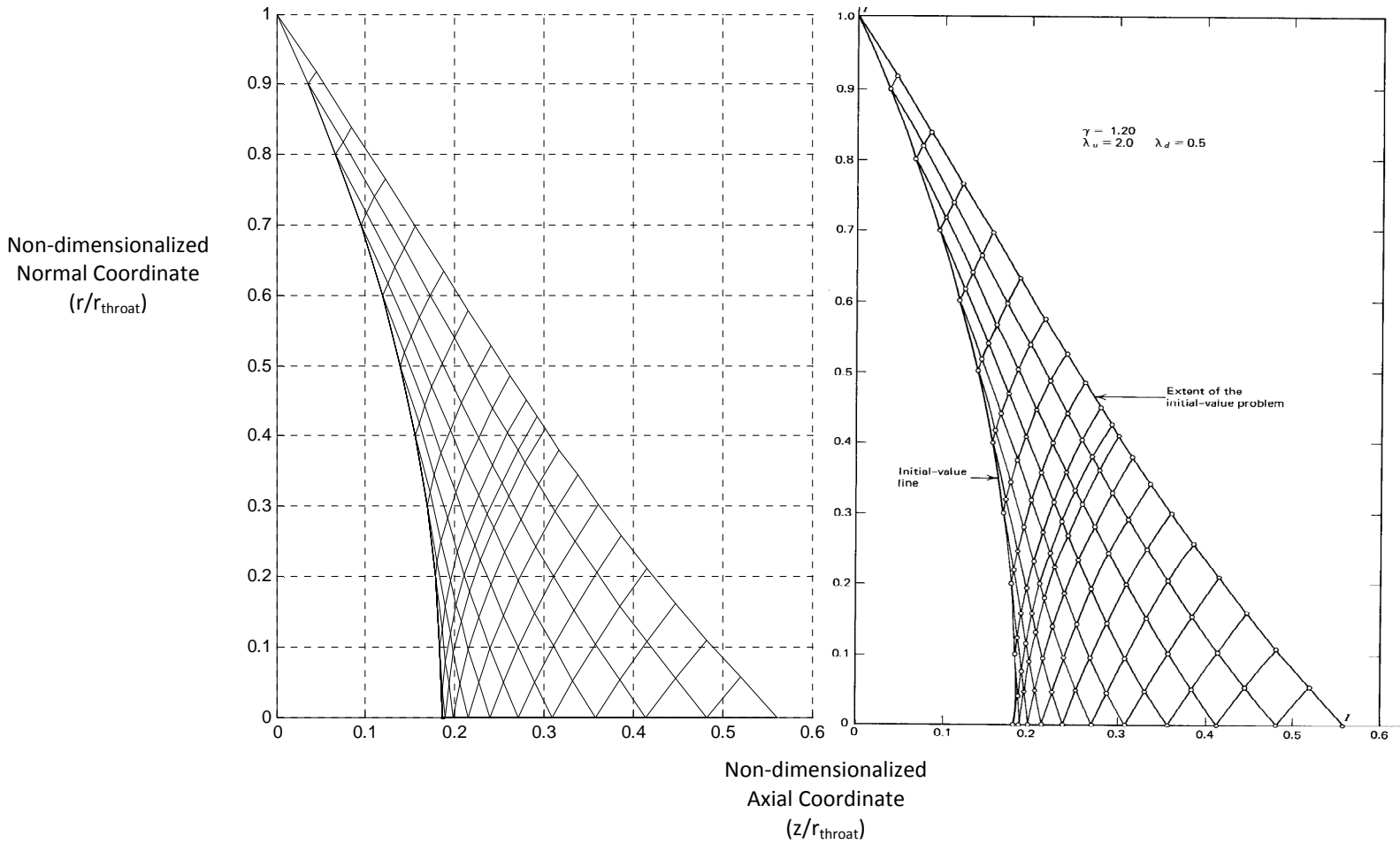
displayed in the left-most column of Table 2. The variable prediction with the largest difference from that of Zucrow had its associated error tabulated in the maximum error row. The maximum error found among all outputs is less than 0.025% and is calculated by the Interior Point unit process. Though it is small for a single point calculation, every error has the potential for propagating downstream as successive subroutines are employed. However, it is likely that the small difference in outputs is caused by the corrector algorithm being applied a potentially larger number of times in ARES. Zucrow applies the corrector algorithm two times in each of his examples while ARES iterates until the position coordinates have changed by less than  $3 \times 10^{-5}$  m and the velocity components have changed by less than  $3 \times 10^{-2}$  m/s. The low tolerance employed was recommended by Zucrow as being a potentially more accurate method of calculating flow field points, though it would also come at a cost of computation time.

## 9.2 Flow Visualization

The next set of comparisons focus on the flow visualization so that it can be used to describe the performance values of the subsequent sections. The flow visualization is comprised of the rocket-ejector geometry, characteristic net of the primary flow, and the segmented 1-D isentropic secondary flow. The section requiring the most rigorous calculation is the primary flow field which utilizes the MOC. To validate this flow field, various configurations were applied and compared to the results obtained by Love<sup>13</sup> and Zucrow<sup>10</sup>.

Figure 25 shows ARES' and Zucrow's MOC solution to the extent of the initial-value problem. The characteristic net begins on the left with an initial data line formed by Sauer's Method and calculates all points which can be described by the first characteristics. With the exception of the initial value line, the negative and positive sloped black lines each represent a C- or C+ characteristic, respectively. The initial-value problem is defined by Zucrow and Hoffman as the flow field solution produced only from the characteristics emanating from the initial data line. It can be seen that the initial line is curved as anticipated due to the multidimensionality of the flow. The two nets are very similar and should provide the nozzle flow field calculation with nearly matching inputs.

A correct initial-value problem is vital to an accurate MOC solution because every error created will grow as the calculations move upstream. An important note to remember when judging the characteristic nets is how the axisymmetric MOC is calculated. The



**Figure 25: Initial-Value Problem, ARES (Left), Zucrow & Hoffman<sup>10</sup> (Right)**

characteristic equations and compatibility equations are mutually dependent on each other. While the characteristic equations are directly responsible for the creation of the characteristic net, the compatibility equations also play a significant role in their development. Since the compatibility equations calculate the fluid properties of the flow field, the matching of the characteristic nets indirectly states that all properties at their respective locations are very similar between the two solutions. An application of this concept is that the full comparison of two MOC solutions can be obtained by only viewing the characteristic nets.

As the focus is shifted upstream of the throat, the nozzle flow field can be compared. Zucrow's full configuration is entered into ARES and the nozzle results are displayed in Figure 26(10). The nozzle wall begins with a circular arc throat and transitions into a constant  $15^\circ$  diverging angle. The flow field begins with the ending of the initial-value problem and concludes at the nozzle exit plane. Again the two characteristic nets are alike. The main difference between the two plots is that Zucrow presents selected characteristics so that a clearer picture of the weak shock generated can be viewed. As

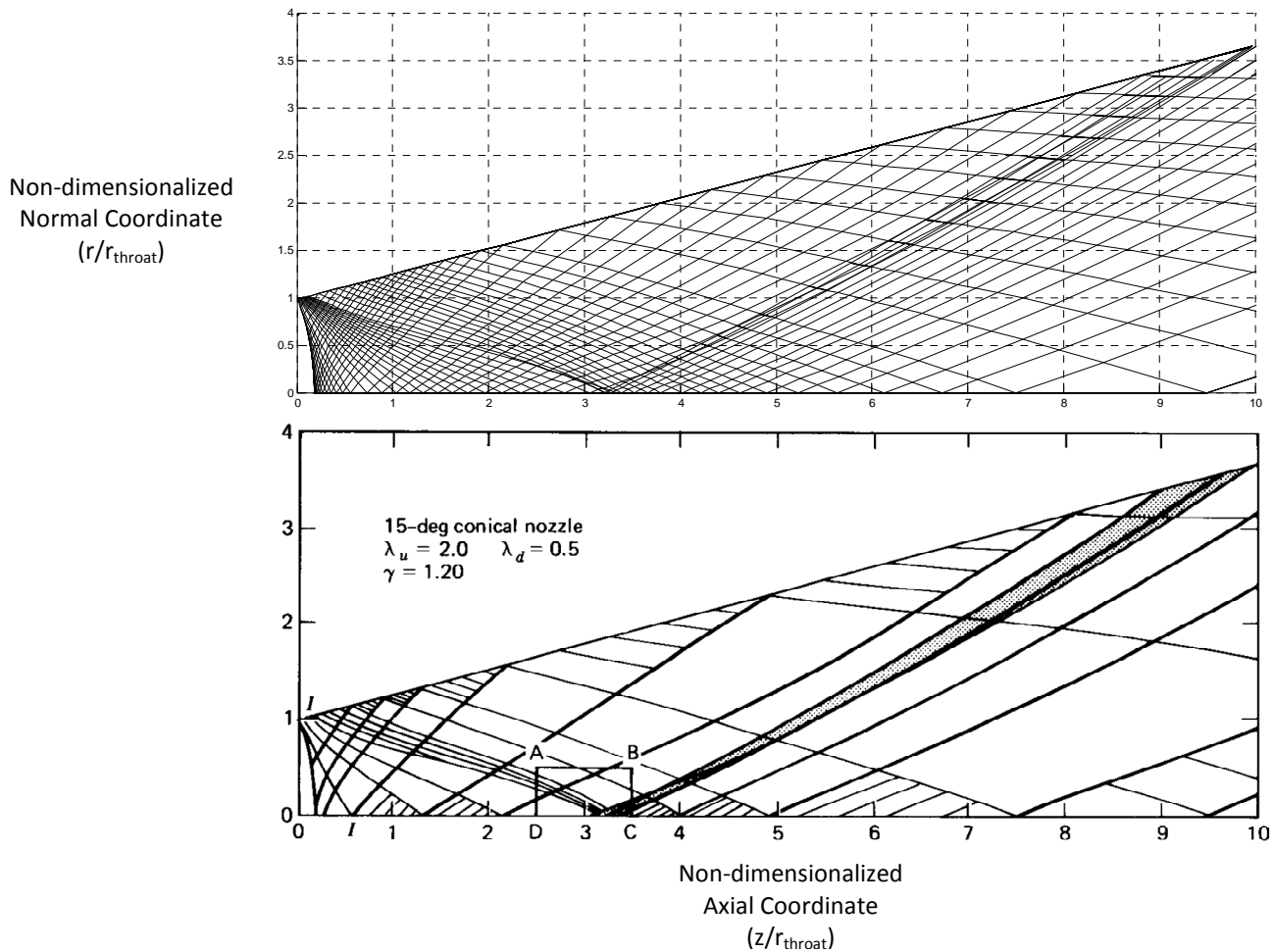


Figure 26: Nozzle Flow field Entering Throat from Left to Right; ARES (Top), Zucrow & Hoffman (Bottom)

described in section 2.2.5, a shock is generated in a non-isentropic nozzle due to a number of reasons. In this case, it occurs as a result of the wall slope being discontinuous at the transition point between the two wall segments. In the case of an overexpanded or slightly underexpanded flow, this would be the first shock which contributes to the double shock diamond configuration.

In both plots the creation of the shock can be traced back to the same location. Also the points where the shock reflects off the centerline and wall have indistinguishable axial positions from that of Zucrow's. The current weak shock validation inside the nozzle is important because all the diverging walls used in the following studies will be non-isentropic.

Now that the solution inside the nozzle has been validated, the plume characteristic net can be evaluated. Unfortunately, most MOC solutions used for research present the characteristic net of either the inside or outside of the nozzle. This means that to validate the plume a new configuration from Love must be applied. Figures 27 and 28 compare two underexpanded plumes from ARES and Love. The top and middle plots in Figure 27 are for a  $\frac{P_1}{P_a} = 10$ . The bottom plot in Figure 27 and Figure 28 are the raw characteristic nets for  $\frac{P_1}{P_a} = 20$ (13).

An underexpanded plume has many complex calculations due to the number of flow structures which must be accounted for. The two that can be seen most evidently in the following figures are the Prandtl-Meyer expansion fan and the intercepting shock. As all

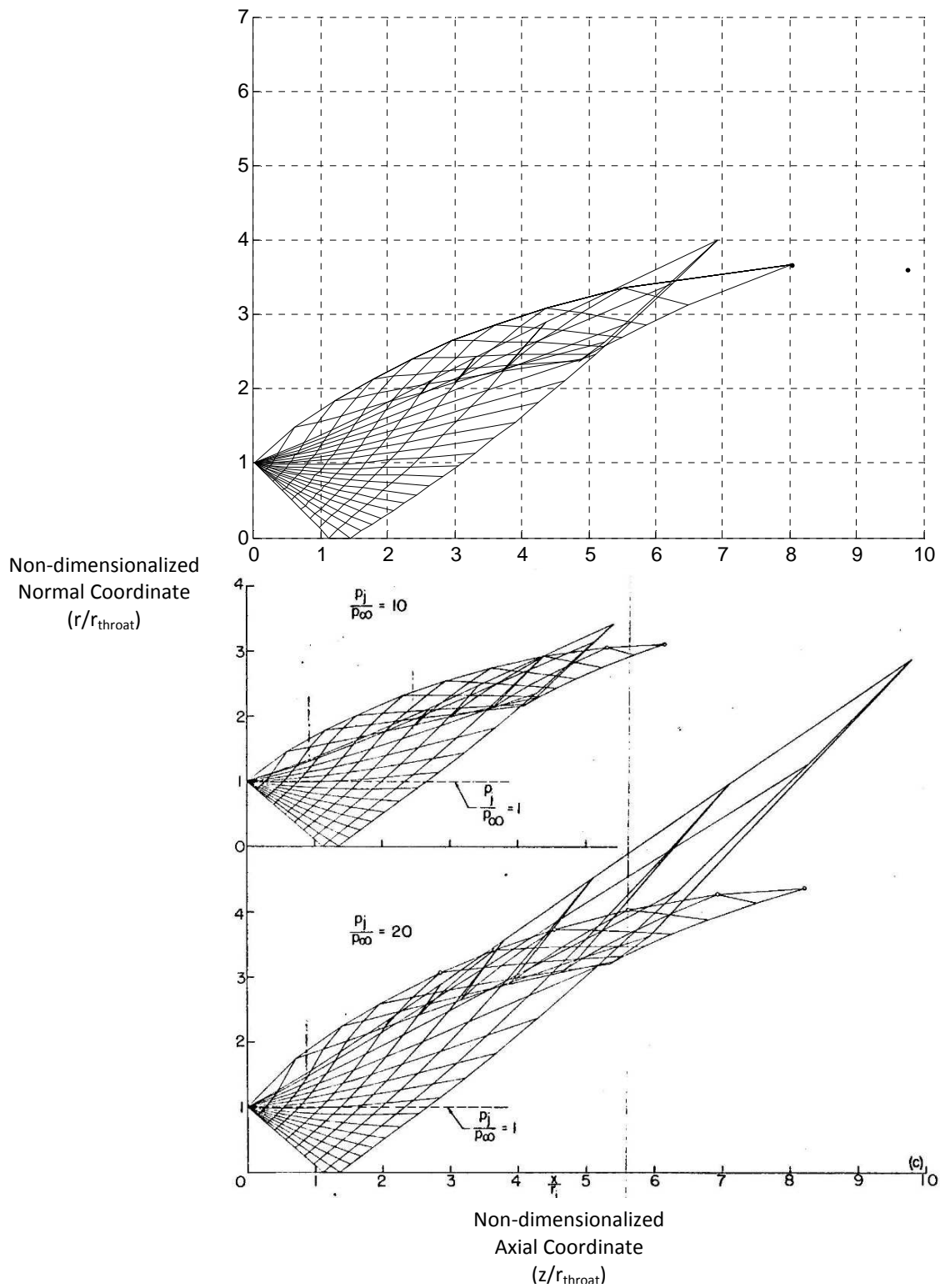
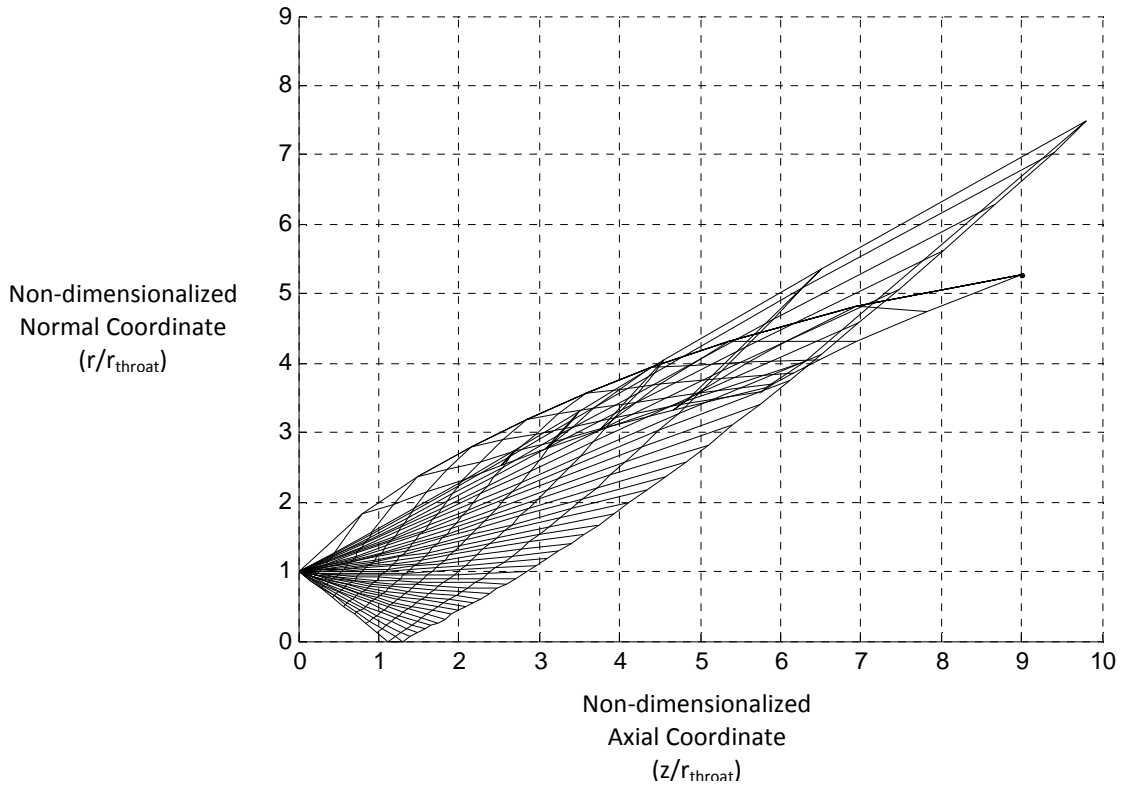


Figure 27: MOC Solution to Underexpanded Plumes; ARES  $P_1/P_a = 10$  (Top), Love  $P_1/P_a = 10$  (Middle). Love  $P_1/P_a = 20$  (Bottom)



**Figure 28: MOC Solution to Underexpanded Plume;  $P_1/P_a = 20$**

the plots are of unprocessed characteristic nets, the intercepting shock can be clearly seen by the overlapping of characteristics. Post-processing of the data partly consists of the Foldback Method described by Love in section 2.1.3 and would result in the absence of overlapping points, thus creating a solution which does not have multiple fluid property values at the same location. Since the Foldback Method is loosely defined in Love, the raw characteristic nets were chosen to be compared because they represent the true values found by the MOC.

Though both configurations produced by ARES are similar to that of Love's, some key differences can be found. The main discrepancy in the  $\frac{P_1}{P_a} = 10$  comparison is that the

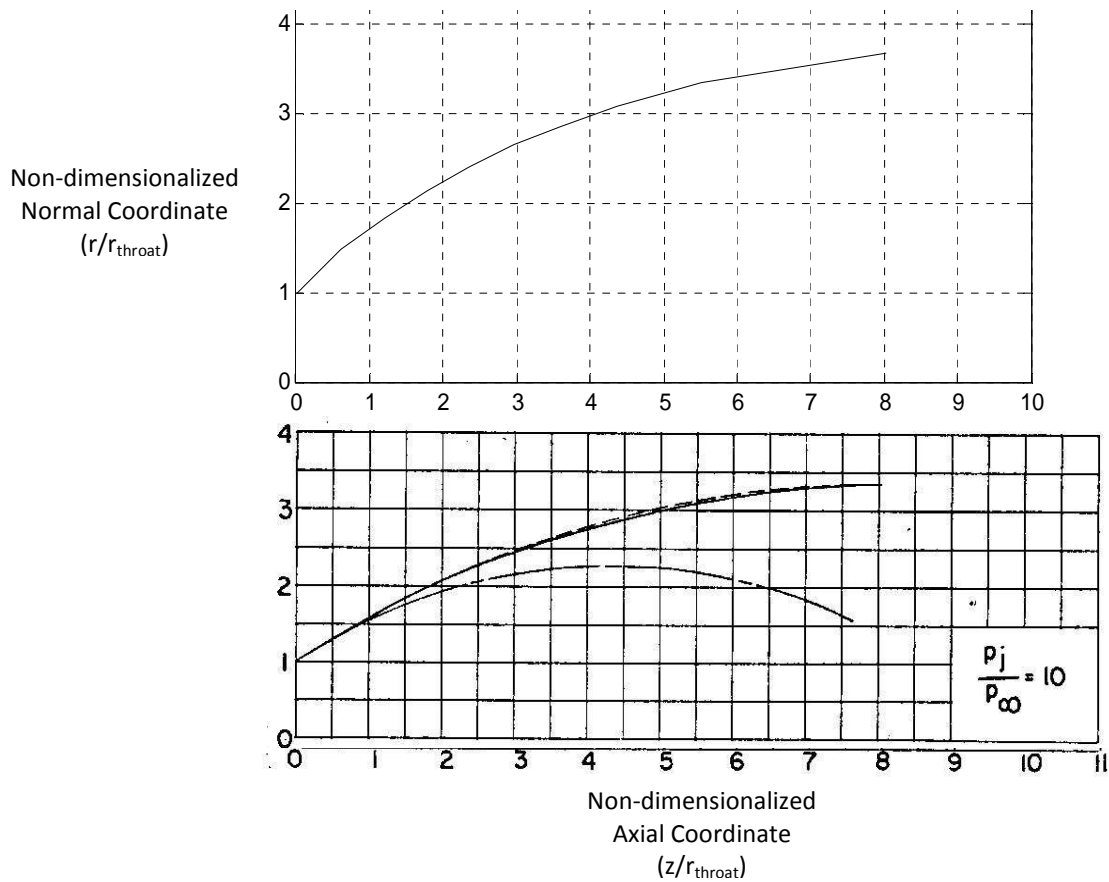


plume boundary in ARES travels further downstream before reaching a maximum. ARES also predicts a slightly higher, or wider, plume than Love. The same increase seems to be present in the fold which indicates the intercepting shock's location. A similar disagreement can be seen in the  $\frac{P_1}{P_a} = 20$  comparison. Interestingly enough, ARES shows here that the folds are nearly matching while the plume boundary continues to grow. This indicates that the plume's characteristic net accuracy is dependent on a number of variables besides the static pressure ratio of the two flows.

The discrepancies described above were somewhat expected due to the procedure Love used. Unlike Zucrow's solution, Love uses the rotational MOC instead of the irrotational MOC like ARES does. While these two forms of the MOC produce very similar results for isentropic flows, they diverge more when the irrotational flow assumption is no longer valid. As described in section 2.2.2, the intercepting shock is curved and so creates a rotational flow field between itself and the plume boundary. The new fluid properties in this region contribute to the differences seen between the two plume boundaries.

Another disparity which may cause inherent variations is that Love used a non-iterative calculation for the MOC in order to save computation time. Love stated that this algorithm required a denser mesh to produce comparable accuracy to that of an iterative solution. A number of the characteristics in each of Love's figures were removed after calculation so that structure visibility was improved. In Figure 27, ARES attempts to match Love's prediction by lower its own resolution to that of the selected

characteristics. While this tactic formed similar results towards the nozzle exit plane, it appears to have reduced accuracy as the calculations proceeded downstream. A requirement of the finite differences employed in ARES is that the step size be small and so may have contributed to the characteristic net differences previously found. The theory seems more likely when it is linked to the variation in fold accuracy described earlier. ARES' plume characteristic nets were calculated with an expansion fan spacing of  $2^\circ$  for  $\frac{P_1}{P_a} = 10$  and  $1^\circ$  for  $\frac{P_1}{P_a} = 20$ . The decreased angle spacing added resolution as well as accuracy to the flow before the irrotationality assumption was violated by the intercepting shock. From this it can be said that ARES should employ a maximum Prandtl-Meyer spacing of  $1^\circ$  for future calculations. While the matching of characteristic nets is important for flow visualization, the secondary flow solution and final performance values only depend on the plume boundary. Figure 29 shows two boundaries produced for an exit Mach of 2 and  $\frac{P_1}{P_a} = 10$ . The top plot is created by ARES and uses an expansion fan spacing of  $1^\circ$  similar to that of the previous figure. Love's plot shows two predictions, the bottom curve is a circular boundary approximation and the top is the MOC solution. For proper comparison, the circular boundary approximation can be ignored. Here it can be seen that the boundaries match quite well. ARES once again predicts a plume which is slightly higher than that of Love's but now the axial locations of the maximum height are nearly the same. Once again, this confirms that decreasing the Prandtl-Meyer expansion spacing will increase the



**Figure 29: Calculated Plume Boundary from Nozzle Lip to Maximum Height; ARES (Top), Love<sup>13</sup> (Bottom)**

accuracy of the prediction. Now that ARES has had its MOC solution verified, the entire rocket-ejector flow field can be viewed.

In Figure 30, the full diverging duct configuration calculated by ARES is shown with a mixture ratio of 1.23 and stagnation pressure ratio of 22.1. The input fluid properties and geometrical dimensions of the shown rocket-ejector are taken from Kyle Johnson's experimental set up. Figure 10 from section 2.2 has been provided below in order to draw more accurate comparisons. Again, the negative and positive sloped black lines in

the primary flow field each represent a C- or C+ characteristic, respectively. The primary flow expands through the nozzle and into the secondary flow stream with increasing Mach. The vertically segmented blue region is the secondary flow. The white area bounded by the nozzle lip, plume boundary, and secondary flow is a stagnation zone created by the high velocity flows. The weak oblique shock generated by the non-isentropic nozzle can be seen propagating downstream as the thin band of black lines. When the weak shock reflects off the plume boundary the characteristics begin to diverge instead of continuing to converge. Another description of the event is that the slowly coalescing compression waves of the weak shock have reflected off as a set of expansion waves into the plume. A shockwave reflecting off in an unlike sense is another indication that the simulation is calculating characteristics correctly because it is an indirect product of the MOC as opposed to a hardcoded constraint.

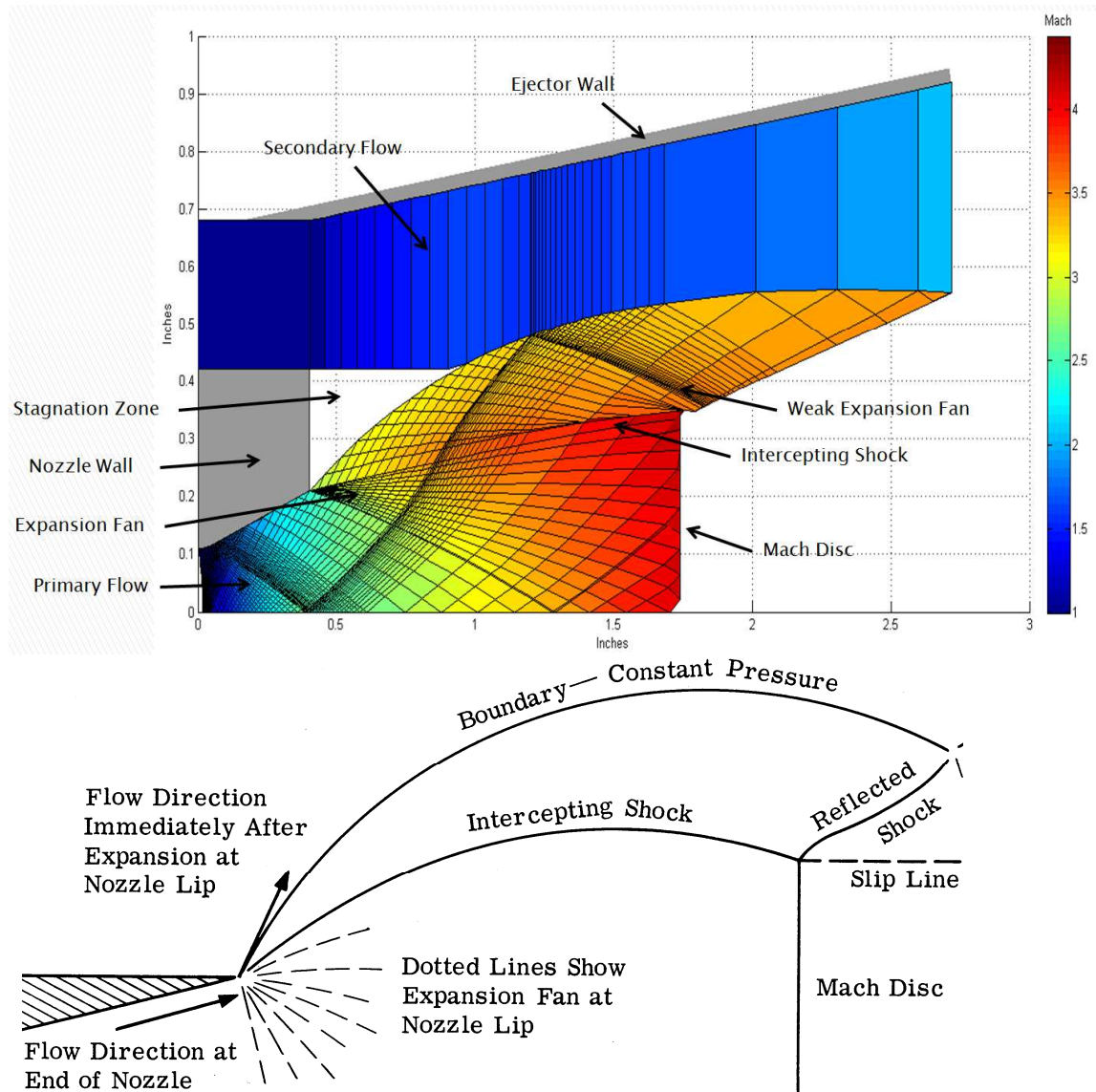
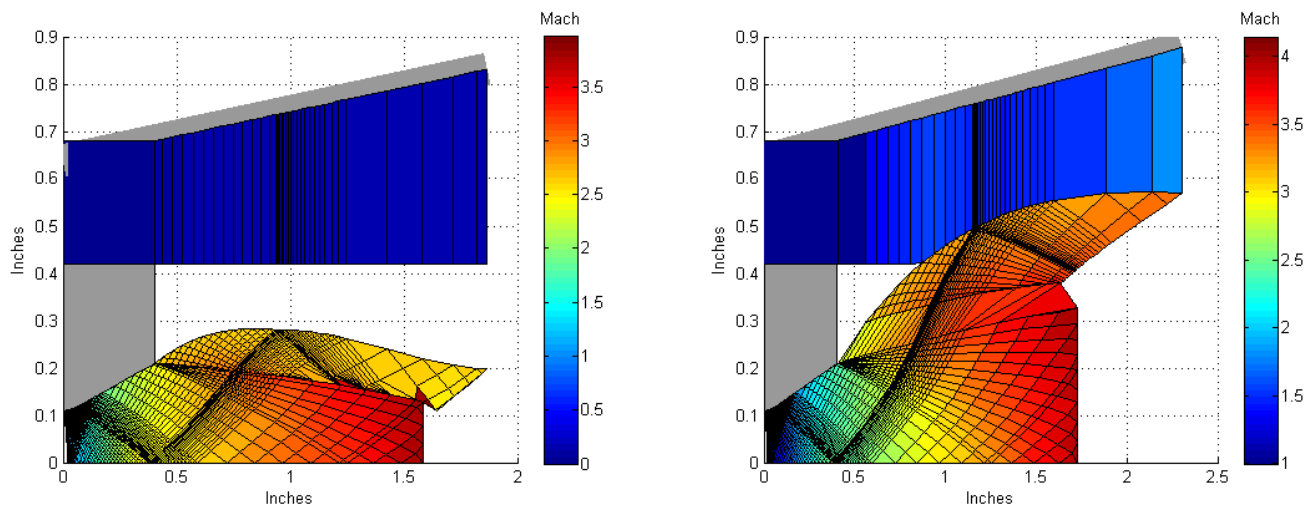


Figure 30: Flow Visualization of Diverging Duct Configuration with  $MR = 1.23$ ,  $P_{op}/P_a = 22.1$  (Above), Theoretical Structure of Highly Underexpanded Nozzle<sup>20</sup> (Below)

It can be seen that a number of phenomena are present in this flow field and align with the theoretical flow structure. The most obvious is the Prandtl-Meyer expansion fan emanating at the exit corner of the nozzle. The expansion fan occurs because the primary flow is highly underexpanded due to the low static pressure of the secondary flow. It is also important to remember that though the bottom picture of Figure 30 shows a constant pressure boundary condition being used, ARES employs a variable pressure distribution. As expected, the variable pressure distribution causes the plume to be more parabolic in shape due to the lower downstream static pressures. An interesting discovery on how ARES forms a Prandtl-Meyer expansion fan can be seen in Figure 31. Here a first iteration solution of ARES is seen next to the converged, final iteration. The first iteration shows the rocket slightly underexpanded while the last exhibits a highly underexpanded flow. The initial guess for the plume boundary is near



**Figure 31: ARES' 1 Iteration Solution (Left) and Final Iteration Solution (Right)**

the ambient pressure and causes the primary flow to be slightly underexpanded. The amount of underexpansion increases throughout the iterations because the guess for the pressure distribution along the boundary is nearing the true steady state value. A real flow would undergo the same transition because the secondary flow initially has no velocity and thus, has an ambient static pressure. Had the primary to secondary static pressure ratio been lower, ARES would have calculated and plotted an overexpanded flow for the first iteration and an underexpanded flow for the final iteration.

A less noticeable flow structure in Figure 30 is the intercepting shock which runs right along and above the expansion fan. To refresh, an intercepting shock grows in strength as it travels through the ending waves of the expansion fan. The flow field below the intercepting shock then ends abruptly near the 1.75 inch axial position because ARES has predicted a Mach Disk. A triple point is then formed at this location as shown in the theoretical flow structures picture. The plume flow field solution then ends downstream of the Mach Disk because it is subsonic and cannot be estimated by ARES' primary flow calculation. The flow above the intercepting shock is continued until the last positive sloped characteristic reaches the plume boundary.

There are two contributing factors which cause the secondary flow to enter the ejector at Mach 1. The first reason is due to the equilibrium requirement. The static pressure of the primary flow along the plume boundary must be equal to that of the secondary at the same axial position. The low, initial stagnation pressure ratio of 22.1 causes the plume to have a low static pressure when it exits the nozzle. The low static pressure, in

turn, encourages the higher pressured secondary flow to accelerate to Mach 1. The other reason for Mach 1 flow is that the area between the ejector and outer nozzle wall is at a minimum. Though it may not appear to be a minimum due to the shape of the expanding plume, the diverging wall increases the secondary flow area which also increases the Mach through the isentropic relationship. An increasing Mach is shown by the gradual lightening of blue as the secondary flow moves through the ejector.

### **9.3 Performance Comparisons**

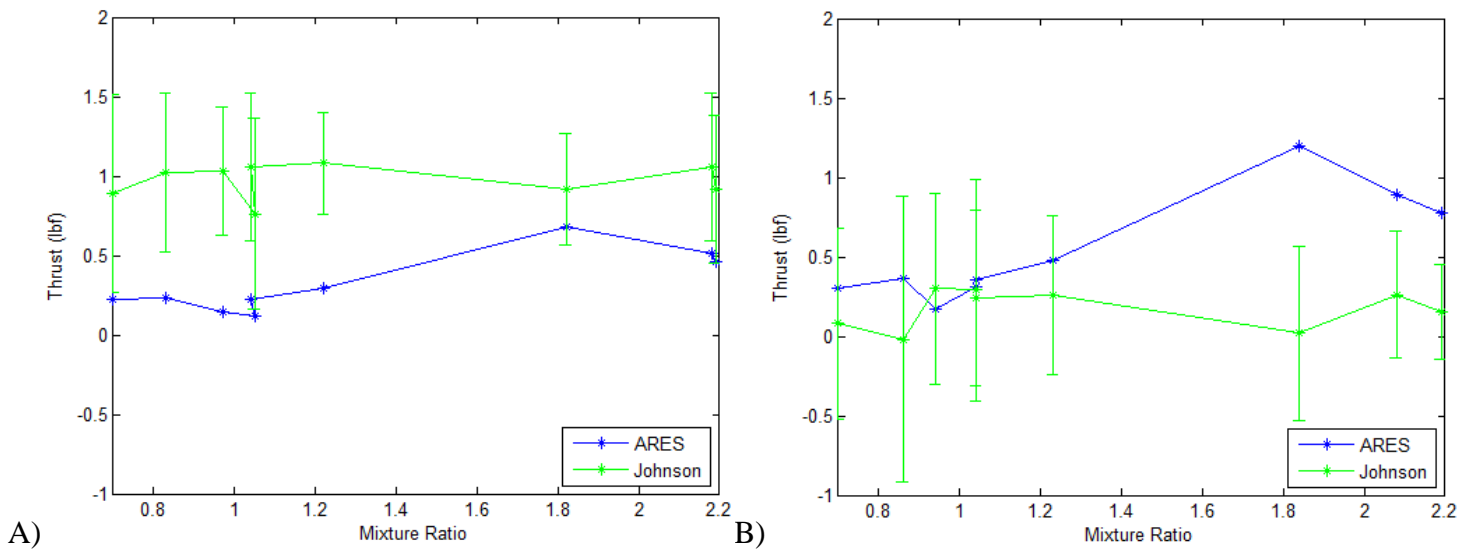
To further validate ARES, the geometry and initial fluid properties described in the Testing Configurations section were entered into the master function and computed. The ejector thrust, ejector  $I_{sp}$ , and total combined  $I_{sp}$  was predicted and tabulated in Table 3. The values were then plotted against Johnson's experimental results to show comparisons in Figures 32, 34, and 36. A final comparison is then made to another study conducted by Trevor Foster and Ryan Gist. Foster and Gist performed experimental testing of a 2-D planar rocket-ejector to discover the aerodynamic choke location of the secondary flow for various stagnation pressure ratios.



	Straight Ejector				Diverging Ejector			
$P_{01}/P_{02}$	Mixture Ratio	Ejector Thrust (lbf)	$\Delta I_{sp}$ due to Ejector (sec)	Total $I_{sp}$ (sec)	Mixture Ratio	Ejector Thrust (lbf)	$\Delta I_{sp}$ due to Ejector (sec)	Total $I_{sp}$ (sec)
23.13	0.7	0.22	1.63	135.16	0.7	0.26	1.94	135.47
22.79	0.83	0.23	2.12	165.08	0.86	0.23	2.15	170.9
21.77	0.97	0.14	1.63	198.28	0.94	0.57	6.32	194.45
21.43	1.05	0.12	1.55	218.78	1.04	0.53	6.81	221.65
22.11	1.04	0.22	2.73	217.39	1.04	0.86	10.58	225.24
22.11	1.22	0.29	4.14	252.3	1.23	0.9	12.86	261.67
24.49	1.82	0.68	8.93	260.36	1.84	1.35	17.6	267.96
23.47	2.18	0.51	6.44	238.2	2.08	0.97	12.51	249.94
23.13	2.19	0.46	5.83	237.08	2.19	0.87	11.14	242.39

**Table 3: ARES' Performance Predictions**

Plots A and B of Figure 32 show thrust comparisons for the straight and diverging walled cases, respectively. The error bars represent the uncertainty of the experimental data. Though the straight ejector values are all under predicted by ARES, the thrust values are within 1 pound of the experimental results with the majority being less than 0.5 pounds away. The diverging duct model also finds that most values are within 1 pound of the experimental results with the exception of one point, though most are over predicted.

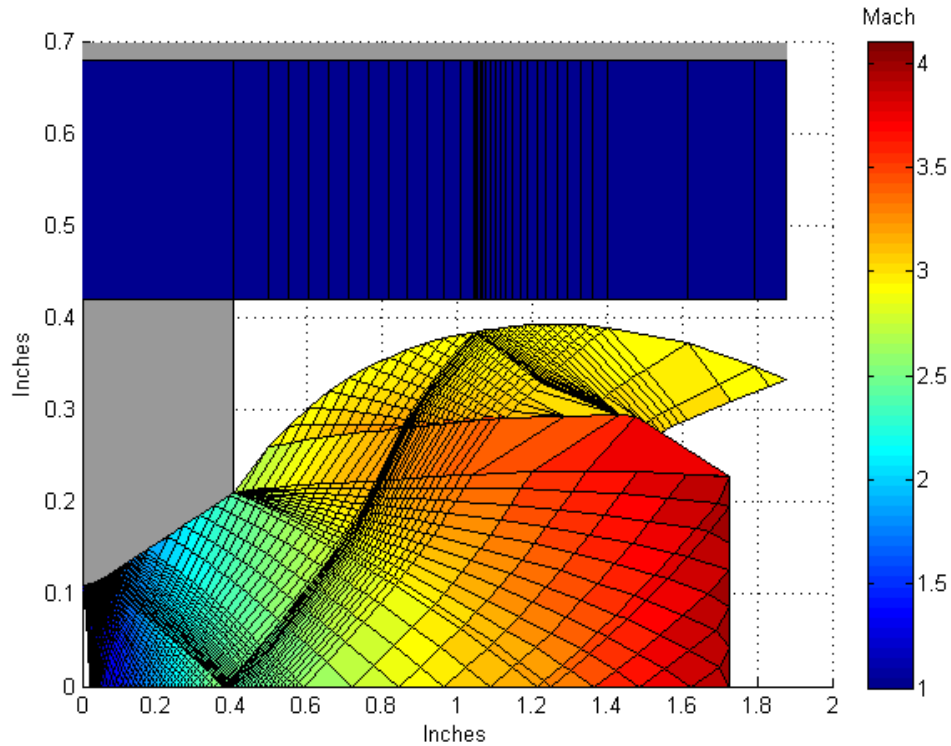


**Figure 32: Ejector Thrust Produced: A) Straight Ejector, B) 5.88° Diverging Ejector**

A number of reasons contribute to the difference in thrust trends between the straight and diverging ejector cases. It was expected that in all cases the thrust would be over predicted due to ARES assuming isentropic conditions for the secondary stream and a fully mixed exit flow from the ejector. The first assumption has an effect when ARES utilizes a 1-D isentropic analysis and eliminates the possibility of shocks. If ARES predicts sonic flow at the ejector inlet, a shock should occur in the secondary flow where the two streams first feel the presence of each other. The shock develops for the same reason that the intercepting shock does in the plume; the flow is not channeled isentropically. When the exhaust exits the nozzle the two flows must reach an equal static pressure. The boundary created does not necessarily form an isentropic shape for either flow but rather only satisfies equilibrium. The requirement results in an intercepting shock in the primary flow and, if supersonic, the secondary flow. As shown in Figure 30, the

secondary flow for the diverging case is supersonic and should have a shock mirroring that of the primary flow's intercepting shock. Since ARES cannot model this flow phenomenon due to the 1-D analysis, some losses are overlooked. Though the exact amount of performance degradation is unknown, previous research suggest wave losses in the secondary flow do not affect the final results greatly<sup>6</sup>. Another reason the diverging case thrust is larger than that of the experimental is because the thrust calculation assumes fully mixed flow. The mixed flow velocity in equation 6 was calculated in ARES through the use of the conservation of momentum. As discussed by Johnson, it is evident that the tested configurations inhibit the primary and secondary flows from mixing. It is hypothesized that this is mainly caused by two reasons: the low stagnation pressure ratio and the large lip thickness of the nozzle when compared to its exit radius.

The case presented in Figure 33 is the straight walled configuration with a chamber pressure of 360 psi and Mixture Ratio of 1.82. It is important to discuss the area right above the expansion fan where the intercepting shock would be located. The lack of C-lines is due to ARES' processing resistance to non-isentropic flow. As the intercepting shock increases in strength, the mesh requires more characteristics at the plume boundary to be removed in order show the discontinuity properly. It is expected that this region of sparse mesh will grow in size as the magnitude of the shock increases.

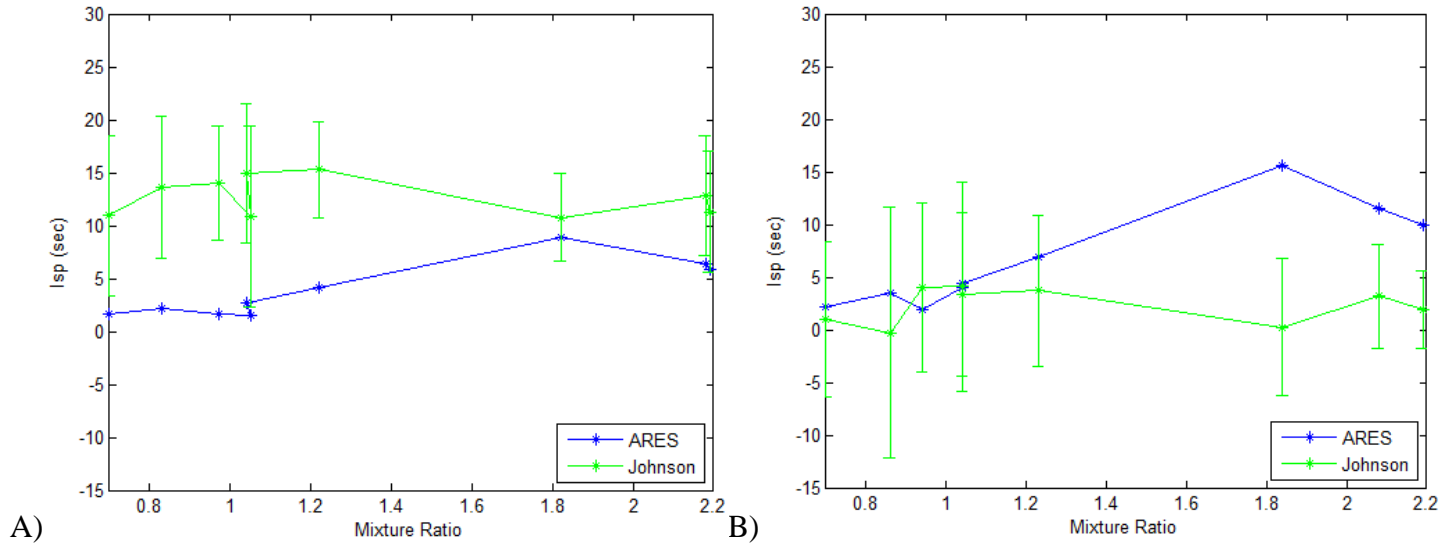


**Figure 33: Straight Duct Configuration, MR = 1.82, P0p = 360psi**

This configuration has a stagnation pressure ratio of 24.5 which is the highest of the test cases and a mixture ratio nearest the optimum value of 1.5. The high pressure ratio results in the most expanded plume boundary of the straight ejector validation cases. Even though this is the widest plume, it can be seen that the boundary does not actually reach into the secondary stream. In terms of plume input parameters, ARES is highly dependent on static pressures and lip thickness to determine the boundary between the primary and secondary flows. From intuition alone, a relationship can be formed stating that as the static pressure ratio decreases or lip thickness increases, the possibility of the plume intersecting the secondary stream decreases. A fully mixed flow is then much more difficult to achieve as the streams are moving nearly independent of each other.

This also explains the seemingly unchanged experimental thrust values for the various chamber pressures. The thrusts predicted by ARES appear to increase with the pressures in the diverging ejector case; this is expected for a fully mixed ejector. Since the primary flow does not create a pressure boundary in the secondary stream there is no source of acceleration for the entrained air. The unchanged velocity can be seen by the consistent blue coloring of the secondary flow. Furthermore, the fact that this is the largest plume explains why the thrust measurements for all the straight ejector test cases are so similar; every other configuration has a narrower plume and cannot entrain airflow either. Ejector thrust, in this situation, is now driven almost entirely by a flow phenomenon occurring downstream of the ejector described by Johnson. The experimental ejectors had a thickness greater than what is usually employed by most air augmented rockets. Johnson hypothesized that the exiting flow created a high pressure region directly downstream of the ejector. The region produced a positive axial force on the rocket-ejector which was not accounted for in ARES due to the given control volume.

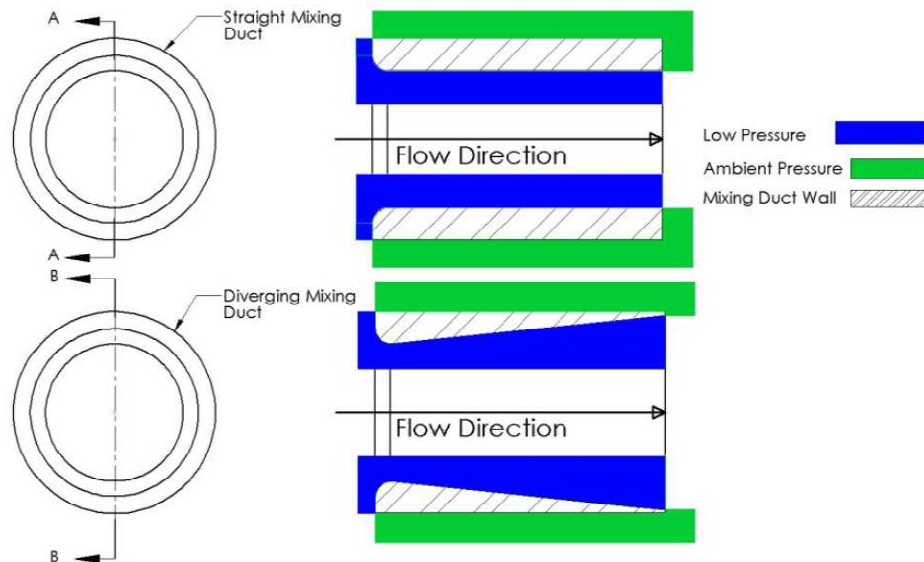
Figure 34 compares the specific impulse created solely by the straight and diverging ejectors. It can be seen that the  $\Delta I_{sp}$  predictions for the ejector-only cases follow very



**Figure 34:  $\Delta I_{sp}$  due to Ejector: A) Straight Ejector, B) 5.88° Diverging Ejector**

similar patterns to that of the corresponding thrust values. As with all the performance comparisons, the irregularities of the lines are due to having two independent variables. Both stagnation pressure and mixture ratio are being varied as shown in Table 3. Predictions which vary only on the vertical axis mean that the input configurations had the same mixture ratio tested with different stagnation pressure ratios employed. Similar to the thrust plots, it is interesting to note that the experimental results show a decrease in average values between the straight and diverging ejector cases while ARES predicts an increase. Johnson explained that this trend was caused by the creation of surface area perpendicular to the axis as the divergence angle of the ejector increased. A visual aid is shown in Figure 35. The fast moving secondary flow has a lower static

pressure than the ambient air and causes a negative pressure force to act on the duct wall.

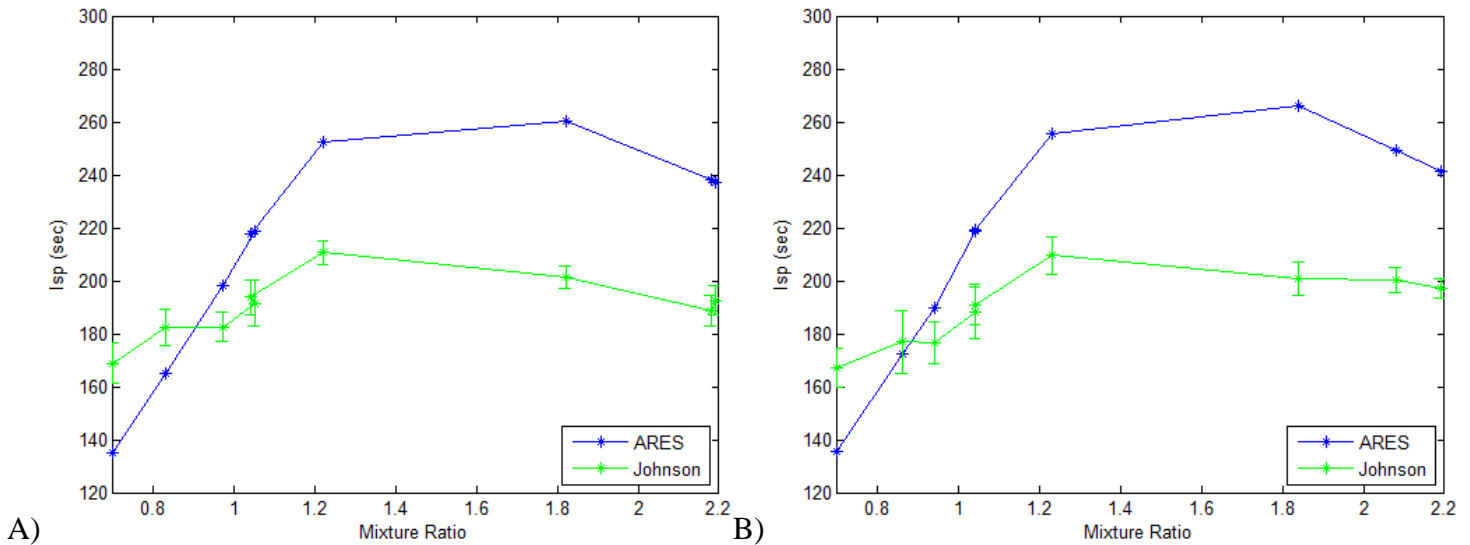


**Figure 35: Pressure Regions in Straight (Top) and Diverging (Bottom) Ejectors<sup>2</sup>**

In a straight ejector this pressure force cannot reduce thrust because there is no area perpendicular to the centerline for it to act against. Diverging ejectors have increasing area in this direction which creates a negative force. The reason this pattern is not seen in ARES' predictions is because of how the large lip thickness affects the accuracy of the fully mixed flow assumption. Johnson's relatively unmixed primary and secondary flow did not add much thrust due to lack of entrainment. A negative force from the duct wall was still generated because the static pressure of the secondary flow must equal that of the low pressure plume boundary. Since ARES assumed fully mixed flow regardless of the lip thickness, the secondary flow for the diverging case was considered fully entrained and created a net positive thrust. As previously expected, this means that

ARES' accuracy will decrease if the tested configuration causes the two flows to remain unmixed.

The total specific impulses predicted by ARES are plotted and compared in Figure 36. As expected, both graphs show a gently curved set of values with a virtual peak nearing the

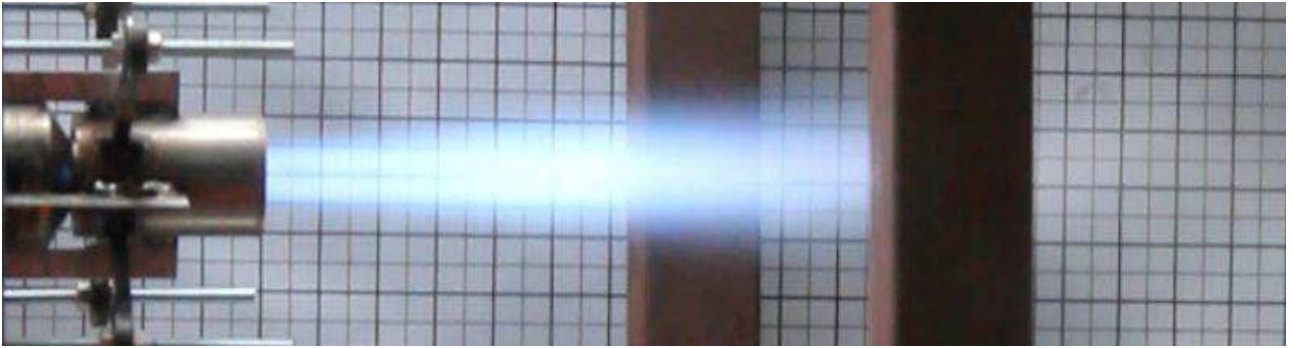


**Figure 36: Total  $I_{sp}$  for Rocket-Ejector: A) Straight Ejector, B) 5.88° Diverging Ejector**

optimum mixture ratio of 1.5. The under predicted values at the lower mixture ratios most likely occurs due to incomplete burning. ARES interpolates values for temperature, specific heat ratio, and specific heat at constant pressure based partly on the mixture ratio. Reactants left over from a rich mixture ratio will not affect downstream flow in ARES because a calorically perfect gas assumption is employed. In Johnson's experimental flow comparison shown in Figure 37, there is still some combustion taking place as the excess fuel travels through the nozzle and ejector(2). The additional combustion continues to change the fluid properties which can affect



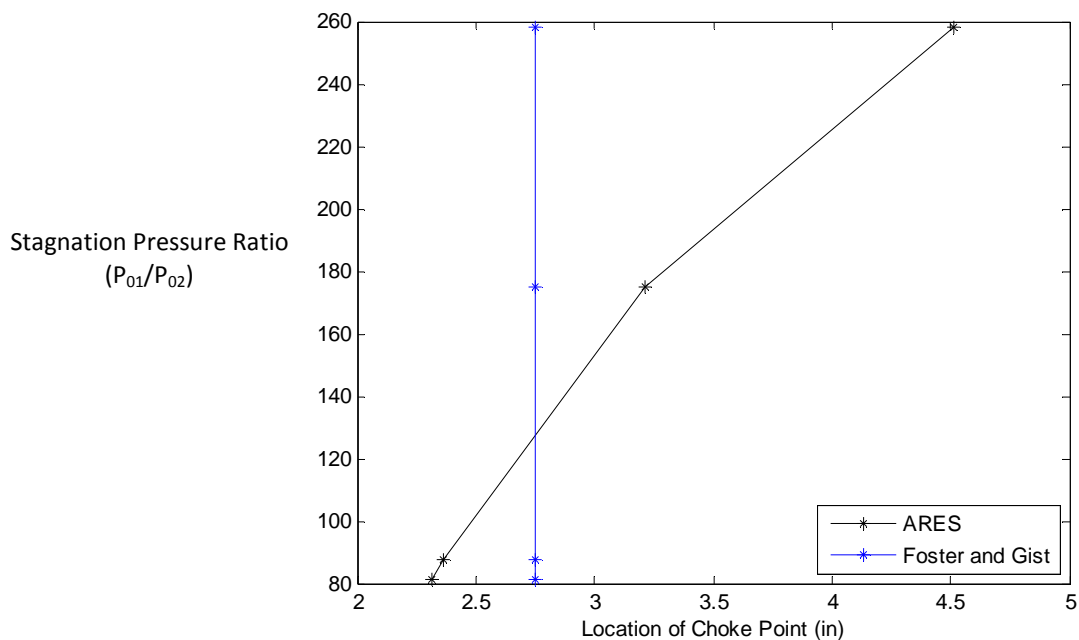
thrust and mass flow rate values. The resulting properties produce increased performance compared to the theoretical case where combustion only takes place in the chamber.



**Figure 37: Johnson's Rich Mixture Ratio Rocket Ejector Showing Combustion outside the Ejector**

Though it was expected that the theoretical total  $I_{sp}$  would be greater than that of the experimental due to the assumptions, a difference of nearly 60 seconds was larger than anticipated. The discrepancy is most likely linked to another unexpected result described by Johnson. It was found that while the nozzle and ejector had positive thrust individually, the total thrust produced was less than that created by a similar rocket without an ejector. The phenomenon was attributed to a large low pressure region being formed at the mouth of the ejector due to the relatively high inflow velocities. The associated low pressure region would react more on the cooling jacket of the rocket than the duct and create a net negative thrust. As ARES does not take forces outside of the ejector into account, the pressure region could not be included. Johnson predicted that this negative thrust could be greatly mitigated by altering the structure for static firings.

The final comparison draws its experimental data from research performed by Trevor Foster and Ryan Gist<sup>30</sup>. Though Foster and Gist found the secondary flow aerodynamic choke point for 2-D planar flow, ARES predicted the location of a similar flow for an axisymmetric case. As described in the Testing Configurations section, four stagnation pressure ratios were tested: 81.5, 87.5, 175, and 258. The results are shown in Figure 38. It can be seen that the choke point locations predicted by ARES for an axisymmetric case varies significantly from those observed in a planar experiment. It is then hypothesized that the choke point location for an axisymmetric case is affected by different flow variables than that of the 2-D planar case.



**Figure 38: Secondary Flow Aerodynamic Choke Point Location Vs Stagnation Pressure Ratio**

## 9.4 Ejector Divergence Angle Study

Now that comparisons have been made with Johnson's results, new configurations are applied to expand the current research on air augmented rockets and demonstrate the robustness of the simulation. A total of 60 configurations with varying ejector divergence angles and input combustion pressures were analyzed and tabulated in Tables 4 and 5. Stagnant secondary fluid properties for all configurations equaled those found at a 10,000 ft altitude. The first set of configurations predicted performance values for a static fire rocket-ejector and the second set assumed the system was traveling at Mach 0.5. In order to evaluate the differences objectively, the thrust augmentation metric will be used instead of thrust.

		Ejector Divergence Angle (Degrees)					
		Mach = 0			Mach = 0.5		
		0°	5°	10°	0°	5°	10°
$\frac{P_{01}}{P_{02}}$ (Primary Stagnation Pressure Secondary Stagnation Pressure)	20.41	1.22	1.13	1.19	1.12	1.04	1.1
	27.21	1.2	1.16	1.2	1.13	1.09	1.13
	34.01	1.18	1.18	1.19	1.12	1.12	1.14
	40.82	1.16	1.19	1.19	1.12	1.14	1.14
	47.62	1.15	1.19	1.18	1.11	1.15	1.14
	54.42	1.14	1.18	1.16	1.1	1.15	1.13
	61.22	1.12	1.18	1.15	1.09	1.15	1.12
	68.03	1.12	1.17	1.15	1.09	1.14	1.12
	74.83	1.12	1.16	1.14	1.09	1.13	1.12
	81.63	1.11	1.15	1.13	1.09	1.13	1.11

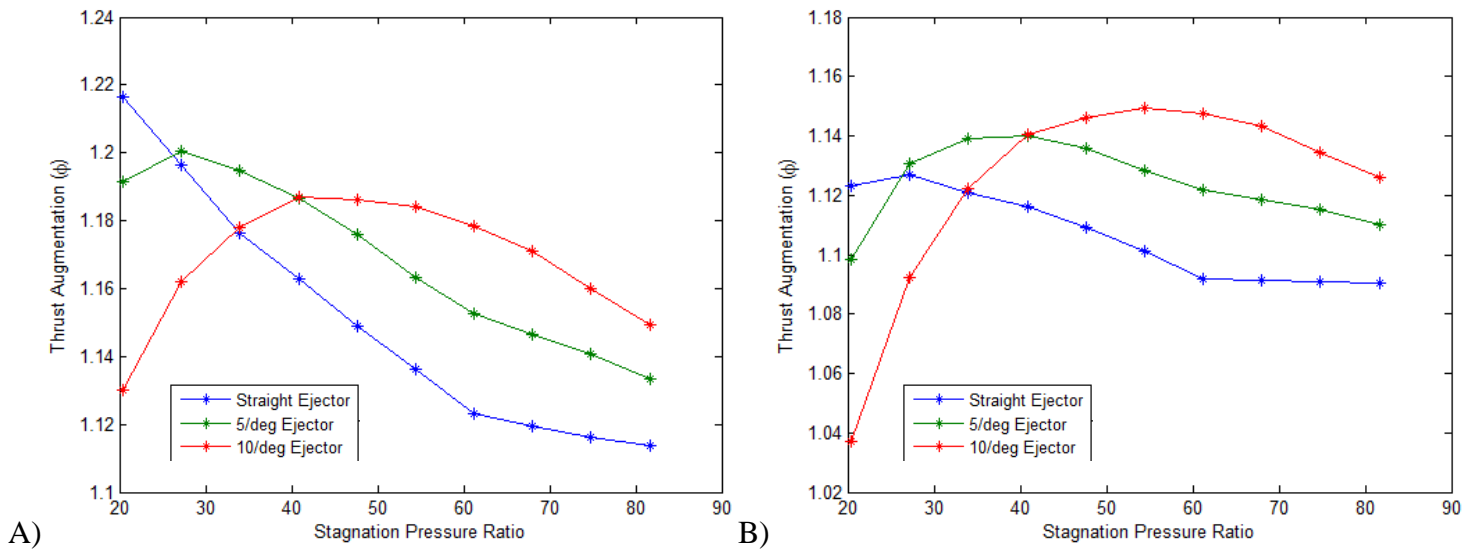
Table 4: Ejector Divergence Angle Study Thrust Augmentation Predictions for an Ethanol/Oxygen(G) Rocket-Ejector

		Ejector Divergence Angle (Degrees)					
		Mach = 0			Mach = 0.5		
		0°	5°	10°	0°	5°	10°
$\frac{P_{01}}{P_{02}}$ (Primary Stagnation Pressure) (Secondary Stagnation Pressure)	20.41	62.78	37.8	55.54	35.79	10.81	28.55
	27.21	56.81	46.85	57.97	36.65	26.69	37.81
	34.01	50.84	51.35	56.17	34.77	35.28	40.1
	40.82	46.8	53.77	53.61	33.44	40.4	40.24
	47.62	42.68	53.38	50.42	31.25	41.95	38.99
	54.42	38.96	52.68	46.76	28.98	42.7	36.78
	61.22	35.15	51.02	43.62	26.3	42.16	34.77
	68.03	34.03	48.83	41.81	26.07	40.88	33.85
	74.83	33.05	45.54	40.07	25.83	38.32	32.84
	81.63	32.35	42.48	37.97	25.74	35.87	31.36

**Table 5: Ejector Divergence Angle Study  $\Delta I_{sp}(\text{sec})$  Predictions for an Ethanol/Oxygen(G) Rocket-Ejector**

This study utilizes an Ethanol/Gaseous Oxygen propellant mixture with ejector divergence angles of 0°, 5°, and 10° applied. The optimum mixture ratio of 2.21 is used for all configurations and the range of stagnation pressure ratios employed is 20.4-81.6. The lip thickness of all tested configurations has been reduced from 0.21 to 0.1 inches to allow more entrainment due to the expansion of the plume boundary. All other fluid properties and structure dimensions remain the same as in Johnson's experiment.

The performance values listed in Table 4 show that the straight, 5°, and 10° diverging ejectors experience a maximum thrust increase of nearly 22% at static-fire conditions and 15% at Mach 0.5. As expected, the added efficiency due to an ejector decreases as the speed of the system increases due to an increase in ram drag. The thrust augmentation produced by each ejector is plotted against stagnation pressure ratio and is shown in Figure 39.



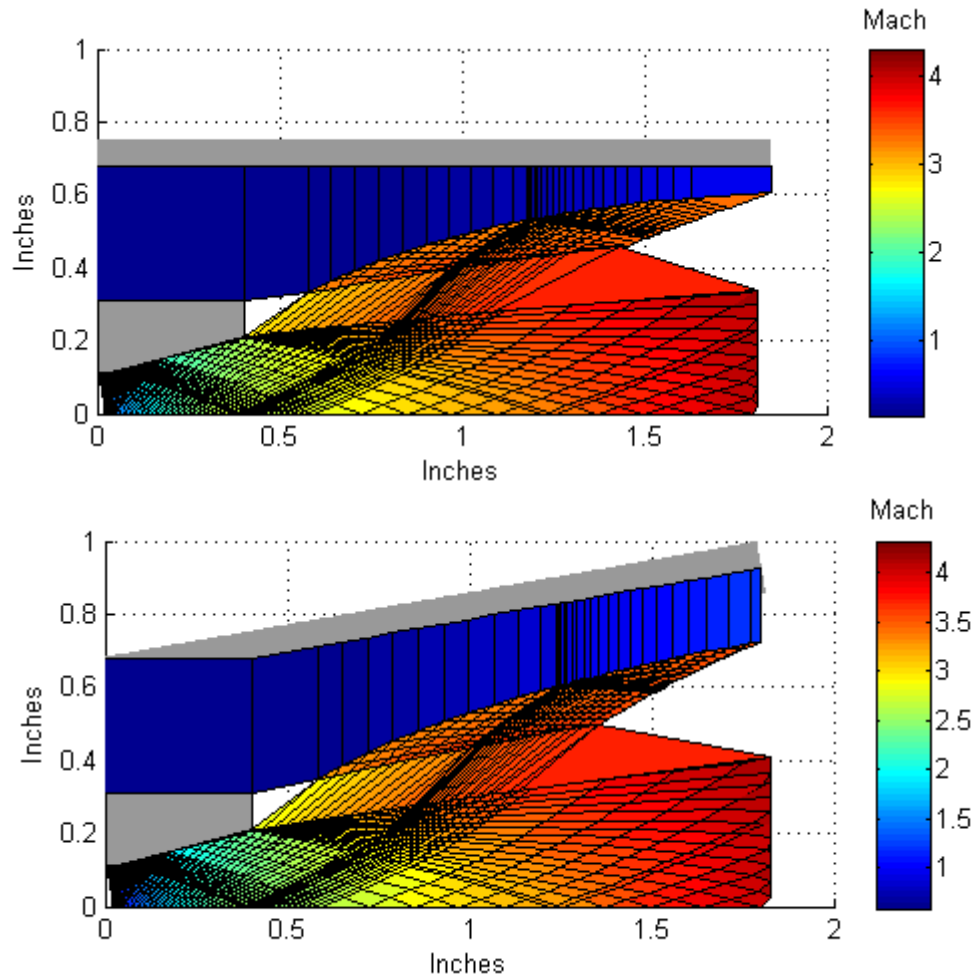
**Figure 39: Thrust Augmentation at 10,000ft: A) Mach = 0, B) Mach = 0.5**

The ejector that produces the greatest thrust augmentation at the lower end of the stagnation pressure ratios is the straight ejector. For the reasons provided at the end of section 9.2, the thrust augmentation decreases as the divergence angle increases at lower stagnation pressure ratios. To reiterate, this is because the diverging ejector angle makes it more difficult to entrain secondary flow because there is an increasingly larger area for the flow to travel through. As described before, this pattern was observed in Johnson's experiment but not in ARES because of the increased lip thickness. Now that the thickness has been reduced, the mixed flow assumption is more accurate and leads to similar results.

The straight ejector provides the highest efficiency until a stagnation pressure ratio of approximately 27 is reached. At this location it can be seen that the 5° angled ejector begins to create a higher thrust augmentation than the straight ejector. A possible

cause is that the amount of secondary flow entrained in the straight ejector begins to decrease as the two flows approach the Fabri Block condition. The ejector with the  $5^\circ$  divergence angle can create a greater thrust augmentation than the straight ejector because it has more area between the plume boundary and duct wall. As the stagnation pressure ratio continues to increase, the ejector with the  $10^\circ$  divergence angle overtakes both the straight and  $5^\circ$  ducts for the same reason. The corresponding trend which is brought to light is that the optimum ejector divergence angle for an air augmented rocket increases with stagnation pressure ratio.

The same trend can also be inferred with ARES' flow visualization in Figure 40. Here the Ethanol/Oxygen rocket-ejector has a stagnation pressure ratio of 81.6 for both the straight and  $10^\circ$  diverging ducts. The high stagnation pressure ratio creates a stronger intercepting shock which translates to a large region of low density mesh above the expansion fan. It is important to remember that though this region appears sparse in terms of characteristics, it is actually quite dense if post-processing techniques are removed. The performance outputs utilize the unprocessed characteristic net which allows for increased accuracy. The post-processing techniques were developed solely for the purpose of locating flow structures.



**Figure 40: Example of Increasing Ejector Flow Area as Divergence Angle Increases**

It can be seen that the secondary flow in the top visualization approaches the Fabri block condition by the expanding plume. Comparatively, the bottom figure still has a relatively large area to accelerate the entrained flow through. As the only variation between the two configurations is the ejector, it can be inferred once again that as the stagnation pressure ratio increases so must the divergence angle in order to maintain optimum performance.

Figure 41 displays the change in specific impulses calculated for the Ethanol/Oxygen rocket-ejector configurations described above. The predicted trend that optimum ejector divergence angle should increase with stagnation pressure ratio appears to hold true for specific impulse as well. Here the straight ejector produces the largest increase in  $I_{sp}$  for stagnation pressure ratios under 27. As the chamber pressure is increased, the  $5^\circ$  ejector overtakes the straight ejector at the same pressure ratio that it does in the thrust augmentation plot. The  $10^\circ$  then overtakes the  $5^\circ$  ejector once a pressure ratio of 40 is reached.

Once again it can be seen that each of the tested ejectors has a performance crest. This can most easily be seen in the  $5^\circ$  and  $10^\circ$  ejectors. Fabri's three conditions can most aptly describe the shape of the plotted values. In the stagnation pressure ratio regime far lower than that of the peak value the associated rocket-ejector is in the separated flow case. This creates very little efficiency gain if at all. As the stagnation pressure

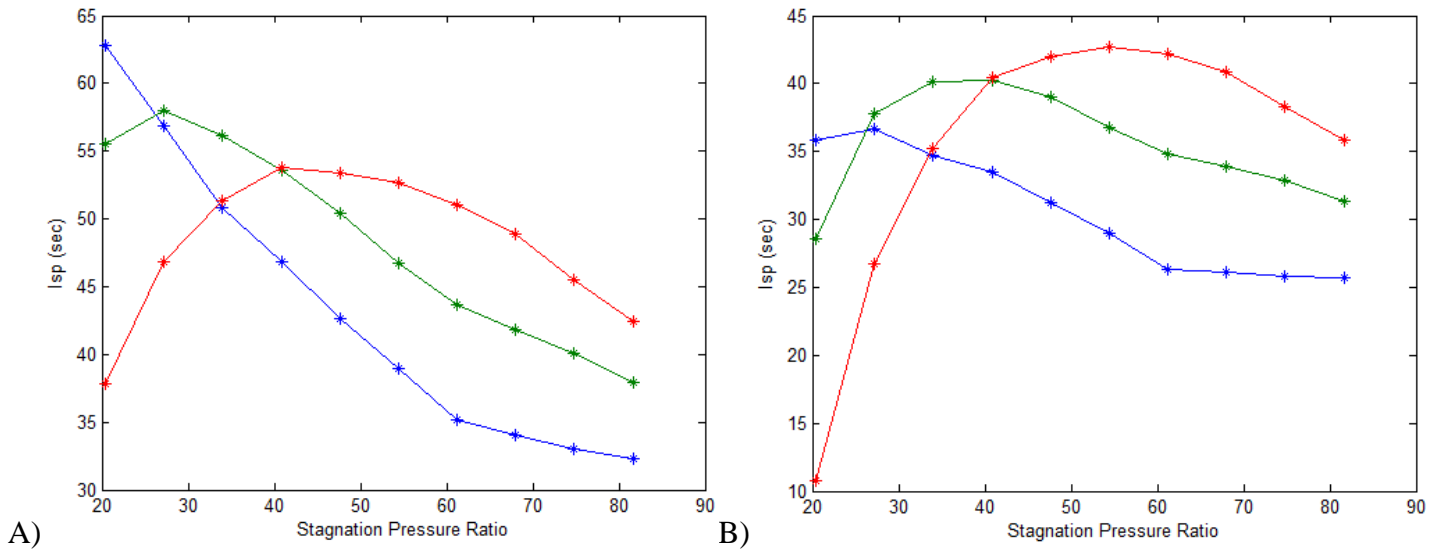


Figure 41:  $\Delta I_{sp}$  due to Ejector at 10,000ft: A) Mach = 0, B) Mach = 0.5



ratio increases the system enters the Fabri-choke condition. Through this condition the max efficiency gain value is predicted due to the Fabri-choke case accelerating the most secondary flow. As the stagnation pressure ratios continue to increase past the crest the rocket-ejector approaches the Fabri-block condition described earlier.

## **10. Conclusion**

The goal of this research was to develop a robust simulation capable of modeling a wide variety of axisymmetric rocket-ejector configurations. The objective culminated in the creation of ARES, an inviscid numerical model which utilizes the irrotational, axisymmetric Method of Characteristics along with a 1-D isentropic analysis to form flow visualizations and performance predictions. The two methods work together to calculate an assumed pressure distribution for the plume boundary and are iterated until they converge on a solution.

ARES was first validated with the use of Zucrow & Hoffman by comparing the outputs of the 5 unit processes employed in ARES. The Interior Point unit process was found to have the largest error at 0.025%. The difference in outputs was attributed to ARES utilizing a more accurate method of point calculation. Subsequent validation engaged Zucrow & Hoffman and Eugene Love to determine the accuracy of the characteristic net produced. Flow inside the nozzle was found to be very similar to that of previous research and the plume comparisons were in good agreement as well. Though calculating with an irrotational assumption, the plume boundary was found to have increased accuracy with a more refined mesh. Flow structures within the plume were also predicted by the Method of Characteristics and provided a high-level flow visualization of the resulting solution.

Performance comparisons were formed with experimental data derived from previous Cal Poly air augmented rocket research. Though predictions were significantly different

from their experimental counterparts, it was largely attributed to unexpected flow phenomena occurring outside the theoretical control volume. It is believed that consideration during structural design such as reducing the size of the experimental cooling jacket could mitigate this inaccuracy.

A parametric study was conducted on how ejector geometry affects performance with varying primary to secondary stagnation pressure ratios. The study used air at 10,000ft for the secondary fluid properties and a Mach of 0 and 0.5. Rocket-ejector thrusts and specific impulses were found to be increased by up to 22% for static fire and 15% for a system at  $M = 0.5$  with the use of an ejector at proper design. It was discovered that the optimum divergence angle for a conical ejector increased nearly linearly with stagnation pressure ratio. Evidence of Fabri conditions were observed in the flow visualization and graphically through the performance predictions.

## **11. Future work**

While ARES has been successful in its goal of becoming a robust program which has modeled a number of axisymmetric rocket-ejector configurations, a great deal of its functionality can still be utilized and improved upon. A large number of tests are yet to be conducted with ARES to develop more trends pertinent to air augmented rockets. Some useful studies that could be conducted are as follows: Varying the nozzle geometry from conical to parabolic, continuing the divergence angle study with converged ejectors, contrasting different lip thickness values in the same configuration, or even observing how well the location of the iterated pressure boundary aligns with the true plume. A more theoretical study could also be done by comparing how ARES' axisymmetric predictions match its 2D rocket-ejector calculations.

Though ARES can be still be employed in a variety of ways, it can also be evolved into a much more accurate simulation. For one, ARES currently uses the irrotational MOC for the entire primary flow solution. Due to the curved intercepting shock in section 2.2, this should be switched to a rotational MOC to improve accuracy. Similarly, the use of a 1D analysis for the secondary flow solution is highly ideal. A large increase in accuracy could be found by using another MOC solution for when ejector flow becomes supersonic. Furthermore, a flow field with many shocks present can only be calculated to a certain degree of accuracy with any form of the MOC. Adding a separate method purely for shock calculations would greatly enhance the credibility of ARES as well as provide additional insight relating to the flow phenomena present. Another assumption

made which could be disposed of is the supposition of inviscid flow. While it has been shown that viscosity plays a very small part in the region ARES is calculating, a separate term could be added to the thrust equation to account for a specific ejector's ability to mix.

## REFERENCES

1. Pollicino, Joe. "Final X-51A WaveRider hypersonic mission achieves Mach 5.1, record flight length." *Engadget*. AOL Inc., 6 May 2013. Web. 21 October 2013.  
<<http://www.engadget.com/2013/05/06/final-x-51a-waverider-hypersonic-mission/>>
2. Johnson, Kyle J. "Axisymmetric Air Augmented Methanol/GOX Rocket Mixing Duct Experimental Thrust Study", Master's Thesis, Department of Aerospace, Engineering, Cal Poly, San Luis Obispo, CA, 2013.
3. Emanuel, N.G., "Comparison of One-Dimensional Solutions with Fabri Theory for Ejectors", *Acta Mechanica*. Vol. 44, pgs. 187-200, 1982.
4. Gaffney, Richard L Jr. "Design of a Pulse-Facility Nozzle using the Rotational Method of Characteristics", *Journal of Spacecraft and Rockets*. Vol. 43, No. 6. November – December 2006.
5. Velasquez, A. and Matesanz, A., "Numerical Simulation of Ejector Rocket Flow", AIAA-98-1615. Madrid, Spain. 1998.
6. Alperin, M. and Wu, J. "Thrust Augmenting Ejectors", Parts I and II, *AIAA Journal*. 21:10 pp. 1428/1698. 1983.
7. Balasubramanyam, M. S., Landrum, D. B., Chen, C. P., and Lineberry, D. "Numerical Investigation of Cold Flow Non-Axisymmetric Ejectors", 43rd AIAA Aerospace Sciences Meeting and Exhibit. Reno, Nevada. January 2005.

8. Morham, Brett G., "Numerical Examination of Flow Field Characteristics and Fabri Choking of 2-D Supersonic Ejectors", Master's Thesis, Department of Aerospace Engineering, Cal Poly, San Luis Obispo, CA, 2010.
9. Etele, J., Parent, B., and Sislian, J. P. "Analysis of Increased Compression Through Area Constriction on Ejector-Rocket Performance", *Journal of Spacecraft and Rockets*. Vol. 44, No. 2. March 2007.
10. Zucrow, Maurice J. and Joe D. Hoffman. *Gas Dynamics Multidimensional Flow*. Vol II. New York: John Wiley And Sons, 1977. Print.
11. Fabri, J., and Siestrunk, "Supersonic Air Ejectors", *Advances in Applied Mechanics*, Vol. 5, Von Karman and Dryden, editors, Academic Press, 1958, pg 1-35.
12. DeTurris, D. J. "Fabri Choking in a Two-Dimensional Reacting Flow Mixer-Ejector", *48<sup>th</sup> AIAA Aerospace Sciences Meeting*. AIAA 2010-384. 4 January 2010. Orlando, Florida.
13. Love, E. S., Grigsby, Lee, and Woodling. *Experimental and Theoretical Studies of Axisymmetric Free Jets*. NASA TR R-6. National Air and Space Administration. Washington, D. C., May 1959 (UNCLASSIFIED).
14. Palmer, Jennifer L. and Hanson. "Application of Method of Characteristics to Underexpanded, Freejet Flows with Vibrational Nonequilibrium", *AIAA Journal*. Vol. 36, No. 2. February 1998.

15. Addy, L. A., "On the Steady State and Transient Operating Characteristics of Long Cylindrical Shroud Supersonic Ejectors", University of Illinois, Ph. D. dissertation. University Microfilms, Inc. Ann Arbor, MI., 1963.
16. Adamson, Thomas C., and J. A. Nicholls, "On the Structure of Jets from Highly Underexpanded Nozzles into Still Air," *Journal of Aeronautical Sciences*, January 1959, Vol. 26, No. 1, pp. 16-24.
17. Liepmann, H. W. and A. Roshko. *Elements of Gas Dynamics*. Canada: General Publishing Company, 1993. Print.
18. Shapiro, Ascher H. *The Dynamics and Thermodynamics of Compressible Fluid Flow*. Vol. 1 & 2. New York: The Ronald Press Company, 1954. Print.
19. Abbett, M., "The Mach Disk in Underexpanded Exhaust Plumes," *AIAA Journal*, Vol. 9, No. 3. 1970. Pp. 512-514.
20. Adamson, Thomas C. "Approximate Methods For Calculating the Structure of Jets From Highly Underexpanded Nozzles", Report of Bamirac. Ann Arbor, Michigan, June, 1961.
21. Lear, W. E., Sherif, S. A., and Parker, G. M. "Effect of Fabri Choking on the Performance of Two-Phase Ejector Pumps", *AIAA 2000-3012*. Florida. 2000.
22. Munday, David, Gutmark, E. "Flow Structure of Supersonic Jets from Conical C-D Nozzles", 39<sup>th</sup> AIAA Fluid Dynamics Conference. San Antonio, Texas. June 2009.
23. Riley, Paul. "Axisymmetric Method of Characteristics: An Algorithm and Its Validation", California Polytechnic State University, June 13, 2008.



24. Gist, D. Ryan, "Examination of Flow Field Characteristics and Fabri-Choking of a 2-D Augmented Rocket", Master's Thesis, Department of Aerospace Engineering, Cal Poly, San Luis Obispo, CA, 2007.
25. Foster, Trevor J., "Rectangular Ducted Methane/GOX Thruster", Master's Thesis, Department of Aerospace Engineering, Cal Poly, San Luis Obispo, CA, 2008.
26. Anderson, John D. *Modern Compressible Flow*. McGraw-Hill, Inc, 1982. Print.
27. Gordon, Sanford and McBride, B. J. "Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications", NASA Reference Publication 1311. October 1994.
28. Kinzie, Kevin W., Yates, David E., and Balan, Chellappa. "A Method for Estimating Thrust for Mixer/Ejector Nozzles", American Institute of Aeronautics and Astronautics. Cincinnati, Ohio, 1998.
29. Presz, W. Jr., Reynolds, G., and Hunter, C. "Thrust Augmentation With Mixer/Ejector Systems", *40<sup>th</sup> AIAA Aerospace Sciences Meeting & Exhibit*. American Institute of Aeronautics and Astronautics. Reno, NV. 2002.
30. Foster, T. J., Gist, D. R., and DeTurris, D. J., "Methane/GOX Air Augmented Rocket Plume Testing", *Cal Poly State University*, San Luis Obispo, CA

## APPENDIX A

### ARES Source Code

```
%%%%% Axisymmetric Rocket-Ejector Simulation (ARES) %%%%%
%Authors/Editors on bottom
clc
clear all
close all
format compact
tic

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Notes on Global
Variables:
global Free %Free must be 0 to indicate
that we have not calculated the nozzle yet. When this turns to 1 in the
global I % Inverse Wall calcs it
means the nozzle is set and does not need to be calculated again
during
global mark % the FPB/Secodary flow
pressure iteration. This saves a lot of computation time.
% 'I' works in conjunction
with 'Free'
Propellants = 5; % 'mark' will keep track of
the indices of the last calculated value of each C- line. This is
MR = [1.5 2.1 1.34]; %used in the next iteration
to keep track of the nozzle values so it isn't recalculated.
DivergAng = [0 10 5];
Pchamb = [300 400 500 600 700 800 900 1000 1100 1200]*10.108/14.7;

a = zeros(length(Pchamb),length(DivergAng),length(Propellants));
Fduct = a;
Fprim = a;
ThrustAugmentation = a;
Isptot = a;
Ispduct = a;
unconverged = [];

for m = 1:length(Propellants)
    for j = 1:length(DivergAng)
        for i = 1:length(Pchamb)
            % Primary Gas Initial Conditions and Identity
            Fuel_Selection = Propellants(m)'; %Choose 1 for Methanol-
Air, 2 for Methanol-GOx, 3 for Hydrazine-N2O4, %4 for Ethanol-Air, 5
for Ethanol-GOx, or [] to choose specific T0p/gamma/R values.
            Oxi_Fuel_Ratio = MR(m); %Choose between
[5.5,7.5] for MethanolAir, [.9,2.1] for MethanolOx [.75,1.75] for
Hydrazine, %[7.5,10.5] for
EthanolAir, or [1.5,2.7] for EthanolOx.
            P0p = Pchamb(i); %chamber pressure
(psia) choose between 100 and 1200 to use CombustionData
```

```

        T0p          = 521.7;                %chamber stagnation
temperature (R)
        gamma        = 1.4;                %ratio of specific
heats for the working fluid
        R            = 1774.864;            %gas constant for the
primary flow (ft-lbf/slug R)
        if isempty(Fuel_Selection) == 0
            [T0p,gamma,R] =
CombustionData(Fuel_Selection,P0p,Oxi_Fuel_Ratio);
        end

        % Secondary Gas Initial Conditions and Identity
        Pa           = 10.108;            %stagnation pressure of
secondary flow [psi]
        Ta           = 483.03;            %stagnation temperature of
secondary flow [R]
        rhoa         = .001756;           %stagnation density of
secondary flow [slug/ft^3]
        gammaS       = 1.4;                %ratio of specific heats of
secondary flow.
        Rs           = 1716.49;           %Gas constant for secondary
flow [ft2/s2*R]
        M            = 0;

        % Geometry
        %----%Nozzle Geometry
        delta         = 1;                %0 for planar, 1 for
axisymmetric (keep this at 1 b/c Planar is not validated)
        theta_attach  = 15;                %attachment turning angle
(degrees)from chap 16 in zucrow&hoffman.
        theta_exit    = 15;                %wall angle at the nozzle exit
(degrees)
        r_th          = 0.11;              %throat radius (in) (distance
from centerline to throat)
        zlen          = 0.402168630226115; %nozzle length (in)
        curvature_up   = 3*r_th;           %(in) radius of curvature
upstream
        curvature_down = 0.22;             %(in) radius of curvature
downstream
        tb            = 0.1;               %thickness of nozzle lip [in]

        Free = 0;
        I = 1;
        mark = 0;

        %----%Duct Geometry
        zduct         = 2.72;              %2.72 inches from nozzle lip to
end of duct [in]
        rduct         = .68;              %half duct width at end of
nozzle [in]
        theta_attach_duct = DivergAng(j); %angle of wall
divergence at end of nozzle [degrees]
        theta_exit_duct  = DivergAng(j); %angle of wall
divergence at end of duct [degrees]

```

```

        %Grid Size
        JStart = 20; %number of points on the initial value line
        IMAX = 1000; %Only here to initialize values and save
computing time

        % Initialize matrices needed for MOC.
        rraw = zeros(JStart*10,IMAX);
        zraw = zeros(JStart*10,IMAX);
        rplot = zeros(JStart*10,IMAX);
        zplot = zeros(JStart*10,IMAX);
        uraw = zeros(JStart*10,IMAX);
        vraw = zeros(JStart*10,IMAX);
        Jbots_for_Cplus = [];

        %%Set initial guess values to start code
        Ps = [Pa,Pa/5]';
        Zs = [zlen,zduct+zlen]';

        if delta == 1
            converged = 2;
        else
            converged = .5;
        end
        update = 10;
        k = 0.8; % relaxation factor for iteration update. 0 for no
relaxation. 1 for no
        % update. crank up for steadier but slower convergence.
        count = 0;

        while update >= converged
            Ps_old = Ps;
            Zs_old = Zs;

            %--Calculate Nozzle
            z=zraw;r=rraw;u=uraw;v=vraw;

            [ZFBD,RFBD,MFBD,Rl,Mp,Pp,i_end,z_tr,r_tr,coeff,coeffDuct,mdotp,u,v,z,r,
            uraw,vraw,zraw,rraw,Fnozz,VE,~,Jbots_for_Cplus] =
            AXImocFn_New_test(u,v,z,r,...
                theta_attach,theta_exit, r_th, zlen, P0p, T0p,
            curvature_up, curvature_down, gamma,...
                R, JStart,
            IMAX,delta,zduct,rduct,theta_attach_duct,theta_exit_duct,Ps,Zs,Pa,Jbots
            _for_Cplus);

            %--Calculate Secondary Flow Based on Nozzle
            [Ms Ps Zs mdots Zd Rd Pd Md Ats,F] =
            SSFLOW2(ZFBD,RFBD,MFBD,zlen,Rl,rduct,...

            coeffDuct,gammaS,Pa,P0p,Ta,Rs,rhoa,tb,delta,zlen,mdotp,R,T0p,gamma,M);

            %--Convergence Update
            [Ps_old,Zs_old] = MultipleValues(Ps_old,Zs_old);
            [Ps,Zs] = MultipleValues(Ps,Zs);
            update = max(abs(norm(Ps)-norm(Ps_old)));

```

```

        %--Add in Convergence Relaxation
        Ps_old = interp1(Zs_old,Ps_old,Zs,'spline');
        Ps = k*Ps_old+(1-k)*Ps;

        %--Shifts Elements to Display Characteristic Net when
using surf()
        [z r Mp] = CPlus_Shifting(Jbots_for_Cplus,z,r,Mp);
        update
        if count > 25
            update = 1;
        end
        count = count+1;
    end
    Fduct(i,j,m) = F;
    Fprim(i,j,m) = Fnozz;
    ThrustAugmentation(i,j,m) = (Fnozz+F)/Fnozz;
    Isptot(i,j,m) = (Fnozz+F)/(mdotp*32.2);
    Ispduct(i,j,m) = (F)/(mdotp*32.2);

    end
end
end

figure
plot(Pchamb/Pa,ThrustAugmentation(:,1,2),'-
*',Pchamb/Pa,ThrustAugmentation(:,3,2),'-
*',Pchamb/Pa,ThrustAugmentation(:,2,2),'-*')
title('Thrust from Ethanol/Oxygen Rocket with Different Diverging
Angled Ejectors')
legend('Straight Ejector','5/deg Ejector','10/deg Ejector',4)
xlabel('Stagnation Pressure Ratio')
ylabel('Thrust Augmentation (\phi)')
figure
plot(Pchamb/Pa,Ispduct(:,1,2),'-*',Pchamb/Pa,Ispduct(:,3,2),'-
*',Pchamb/Pa,Ispduct(:,2,2),'-*')
title('Isp from Ethanol/Oxygen Rocket with Different Diverging Angled
Ejectors')
legend('Straight Ejector','5/deg Ejector','10/deg Ejector',4)
xlabel('Stagnation Pressure Ratio')
ylabel('Isp (sec)')
figure
plot(Pchamb/Pa,Isptot(:,1,2),'-*',Pchamb/Pa,Isptot(:,3,2),'-
*',Pchamb/Pa,Isptot(:,2,2),'-*')
title('Total Isp from Ethanol/Oxygen Rocket with Different Diverging
Angled Ejectors')
legend('Straight Ejector','5/deg Ejector','10/deg Ejector',4)
xlabel('Stagnation Pressure Ratio')
ylabel('Isp (sec)')

toc

%Cal Poly Air Augmented Rocket Code
%
```

```

%Axisymmetric MOC and numerous additional contributions by Jeff Massman
%2012-2013
%
%Modified by Jeff Freeman
%Version 1-1
%April 30, 2011
%
%Plume and Ejector Contributions Formulated and Coded by Brett Morham
%January 14,2009
%
%Nozzle Method Of Characteristics originally coded by Paul Riley

function
[ZFBD,RFBD,MFBD,Rl,M,P,i_end,z_tr,r_tr,coeffs,coeffsDuct,mdot,u,v,z,r,u
raw,vraw,zraw,rraw,F,VE,FPexit,Jbots_for_Cplus] =
AXImocFn_New_test(u,v,z,r,...

theta_attach,theta_exit,r_throat,zlen,P0,T0,curvature_up,curvature_down
,gamma,R,JStart,IMAX,...

delta,zduct,rductstart,theta_attach_duct,theta_exit_duct,Ps,Zs,Pa,Jbots
_for_Cplus)

global I
global Free
global mark
%% Set Constants, make necessary unit conversions, and declare initial
'switches'
CP = gamma*R/(gamma-1);
GL= 144; %unit conversion factor from psi to psf
Ps=Ps*GL;
P0=P0*GL;
Pa=Pa*GL;

initial_val = 75;
MD = IMAX;
b = 0;
IWjbot = 0;
zDisc = 0;
done=false;

%Switches
finalwave = 0;
% 'finalwave' values:
%     finalwave == 0 means flow before the expansion fan
%     finalwave == .5 means flow in the expansion fan
%     finalwave == 1 means flow on the last expansion wave
%     finalwave == 2 means flow after the expansion fan
%     finalwave == 3 means overexpanded flow and there is no
expansion fan, so the MOC continues with a weak oblique shock

%% Find coordinates of attachment points and Wall Coutours (more
description below)

```

```

z_tr = curvature_down*sind(theta_attach); %z-coordinate of the wall
point that connects the circular throat arc downstream of the throat
and the expanding wall section
r_tr = r_throat+(1-cosd(theta_attach))*curvature_down; %r-coordinate of
the wall point that connects the circular throat arc downstream of the
throat and the expanding wall section

%Calculate the coefficients of the wall contour for the bell (or
%"diminishing wall angle") region. Assuming a contour described by a
%second order polynomial  $r = a + b*z + c*z^2$ . This is the same process
as in
%Zucrow & Hoffman. The coefficients are:
coeff_c = (tand(theta_exit)-tand(theta_attach))/(2*(zlen-z_tr));
coeff_b = tand(theta_attach)-2*coeff_c*z_tr;
coeff_a = r_tr-(coeff_b*z_tr)-(coeff_c*(z_tr^2));
R1 = coeff_a+coeff_b*zlen+coeff_c*zlen^2;

%Calculate the coefficients of the wall contour for the secondary duct
(if
%there is an angle or shape). Similarly described as the nozzle contour
with
%second order polynomial. These are used for the WallIntersection
%calculations if the plume contacts the duct wall and the secondary
flow iteration:
coeff_c_duct = (tand(theta_exit_duct)-
tand(theta_attach_duct))/(2*(zduct));
coeff_b_duct = tand(theta_attach_duct)-2*coeff_c_duct*zlen;
coeff_a_duct = rductstart-(coeff_b_duct*zlen)-(coeff_c_duct*(zlen^2));

%%
% Sets starting C- characteristic at which we will begin calculation
i = I+1;
Jbots_for_Cplus(i:end) = [];
% If I == 1 then this is the first iteration of the MOC and initial
steps need to be taken.
% This iteration calculates the whole flowfield. Subsequent iterations
start calculations from
% the end of the nozzle to save on computation time. The plume
calculations cannot affect value
% inside the nozzle.
if I == 1
    % Set initial line values
    [x y u_tilda v_tilda] =
SauersInitialLine_New(gamma,R,curvature_up,r_throat,T0,JStart,delta);
    z(1,1:length(x)) = x;
    r(1,1:length(x)) = y;
    u(1,1:length(x)) = u_tilda;
    v(1,1:length(x)) = v_tilda;
    JStart = JStart-1; %Comment this out for Zucrow IV Line Validation

    % This sets the z-coordinates for the circular arc throat section.
We do this to ensure more C- lines emanating from the wall downstream
of the
    % attachment point. Without it the mesh of characteristics would
be sparse.

```

```

        numofinversepts = round(1*JStart); % 15 was number of inverse wall
        points chosen by book when they had 11 JStart points. Using the same
        ratio of points
        totaltheta = asin(z_tr/curvature_down);
        dthetapts = linspace(0,totaltheta,numofinversepts+1);
        t = 2;
        for B = JStart+1:(numofinversepts+JStart)
            z(1,B) = curvature_down*sin(dthetapts(t));
            t=t+1;
        end
    else
        JStart = JStart-1; %Comment this out for Zucrow IV Line Validation
        numofinversepts = 0;
        %This creates new z/r/u/v matrices and inputs the nozzle values
        into it
        Rtemp = zeros(JStart*10,IMAX);
        Ztemp = zeros(JStart*10,IMAX);
        Utemp = zeros(JStart*10,IMAX);
        Vtemp = zeros(JStart*10,IMAX);
        for ii = 1:length(mark)
            if ii == I+1
                break
            end
            Rtemp(1:mark(ii),ii) = r(1:mark(ii),ii);
            Ztemp(1:mark(ii),ii) = z(1:mark(ii),ii);
            Utemp(1:mark(ii),ii) = u(1:mark(ii),ii);
            Vtemp(1:mark(ii),ii) = v(1:mark(ii),ii);
        end
        r = Rtemp;
        z = Ztemp;
        u = Utemp;
        v = Vtemp;
    end

while ~done
    for j = 1:JStart*10
        % Indexing logic (changes for different parts of the code)
        if i > JStart
            jbot = j+1;
            if IWjbot ~= 0
                jbot = floor(IWjbot)+j+1;
            end
        else
            jbot = j-1;
        end
        if finalwave > 0 && finalwave < 2
            jbot = j;
        end

        % Calculate points along C- lines from the initial data line
        first
        if i > 1 && i <= JStart && j > 1
            if u(jbot,i-1) ~= 0

```



```

        [u,v,z,r] =
InteriorPoint_New_test(u,v,z,r,initial_val,gamma,R,T0,delta,i,j,jbot,CP
);
        else
        [u,v,z,r] =
Centerline_New_test(u,v,z,r,initial_val,gamma,R,T0,delta,i,j,CP);
        b = 1; % no more in this C- line
        end
    end

    % Calculate points along C- characteristics emanating from
    Nozzle Wall
    if i > JStart
        if j == 1
            if i <= JStart+numofinversepts
                % If point lies on the circular arc part of the
                wall
                % (Read bottom for a description on the logic for
                this section)
                if IWjbot ~= 0
                    jbot = jbot+floor(IWjbot)+3;
                end
                [u,v,z,r] =
InverseWallIntersection_New(u,v,z,r,initial_val,gamma,R,T0,delta,i,j,co
eff_a,coeff_b,coeff_c,z_tr,curvature_down,r_throat,jbot,CP);
                for k = 2:length(z(:,i-1))
                    [~,~,zcheck,~] =
WallIntersection_New(u,v,z,r,initial_val,gamma,R,T0,delta,k,i,j,coeff_a
,...
coeff_b,coeff_c,theta_attach,theta_exit,z_tr,curvature_down,r_throat,zl
en,CP);
                    if zcheck(j,i)>z(j,i)
                        IWjbot = k-3;
                        if IWjbot == 0
                            IWjbot = .5;
                        end
                        Jbots_for_Cplus(i) = jbot-j;
                        break
                    end
                end
            elseif i > (JStart+numofinversepts)
                if Free == 0
                    % If point lies on some other part of the wall
                    jbot = j+1;
                    [u,v,z,r] =
WallIntersection_New(u,v,z,r,initial_val,gamma,R,T0,delta,jbot,i,j,coef
f_a,...
coeff_b,coeff_c,theta_attach,theta_exit,z_tr,curvature_down,r_throat,zl
en,CP);
                    IWjbot = 0;
                end
                if Free == 1
                    if i == I

```

```

                                % This ensures that a characteristic
emanates from the nozzle lip
                                z(j,i) = zlen;
                                [u,v,z,r] =
InverseWallIntersection_New(u,v,z,r,initial_val,gamma,R,T0,delta,i,j,co
eff_a,coeff_b,coeff_c,z_tr,curvature_down,r_throat,jbot,CP);
                                finalwave = .5;
%                               done = true; %Set this for Zucrow
Validation
                                elseif finalwave < 1
                                    [u,v,z,r,finalwave] =
LipPoint(u,v,z,r,gamma,R,T0,i,j,Zs,Ps,P0,CP);
                                    if finalwave == 3
                                        %This case means that the flow is only
slightly overexpanded and the MOC can continue on its own.
                                        [u,v,z,r] =
FreePressureBndry_New(u,v,z,r,initial_val,gamma,R,T0,delta,jbot,i,j,Zs,
Ps,P0,CP);
                                    end
                                else
                                    % If point lies on the Free Pressure
Boundary
                                    [u,v,z,r] =
FreePressureBndry_New(u,v,z,r,initial_val,gamma,R,T0,delta,jbot,i,j,Zs,
Ps,P0,CP);
                                    if i > MD && u(jbot+1,i-1) == 0
                                        done = true;
                                        Jbots_for_Cplus(i) = jbot-j;
                                        break
                                    end
                                end
                                end
                                end
                                elseif u(jbot,i-1) ~= 0
                                    [u,v,z,r] =
InteriorPoint_New_test(u,v,z,r,initial_val,gamma,R,T0,delta,i,j,jbot,CP
);
                                    if r(j,i) == 0 %This means the C- line crossed the
previous one. We 'terminate' the current C- line at the point of the
cross and use the previous C- values for future calculations.
                                        b = 1;
                                    end
                                    if i > MD && u(jbot+1,i-1) == 0
                                        b = 1;
                                    end
                                else
                                    [u,v,z,r] =
Centerline_New_test(u,v,z,r,initial_val,gamma,R,T0,delta,i,j,CP);
                                    if finalwave == 1
                                        [zDisc ~] =
Mach_Disc(z(:,i),r(:,i),u(:,i),v(:,i),Zs,Ps,P0,gamma,R,T0,CP);
                                        MD = i;
                                        finalwave = 2;
                                    end
                                    b = 1; % no more in this C- line

```

```

        end
    end

    % Check to see if done and find points at the end of the
    desired flowfield
    if z(j,i)>=zduct+zlen
        if j == 1
            done = true;
        end
        b = 1;
        if finalwave == 1
            [zDisc ~] =
Mach_Disc(z(:,i),r(:,i),u(:,i),v(:,i),Zs,Ps,P0,gamma,R,T0,CP);
            MD = i;
            finalwave = 2;
        end
    end
    Jbots_for_Cplus(i) = jbot-j;
    clear jbot
    if b == 1
        break
    end
end %This "end" ends the for loop for j
if zDisc ~= 0
    for itemp = 1:i
        for jtemp = 1:JStart*10
            if z(jtemp,itemp) > zDisc
                z(jtemp,itemp) = zDisc;
                r(jtemp,itemp) = interp1(z(jtemp-
1:jtemp,itemp),r(jtemp-1:jtemp,itemp),zDisc);
                u(jtemp,itemp) = interp1(z(jtemp-
1:jtemp,itemp),u(jtemp-1:jtemp,itemp),zDisc);
                v(jtemp,itemp) = interp1(z(jtemp-
1:jtemp,itemp),v(jtemp-1:jtemp,itemp),zDisc);
                z(jtemp+1:end,itemp) = 0;
                r(jtemp+1:end,itemp) = 0;
                u(jtemp+1:end,itemp) = 0;
                v(jtemp+1:end,itemp) = 0;
                break
            elseif z(jtemp,itemp) == 0
                break
            end
        end
    end
    zDisc = 0;
end
if ~done
    i=i+1;
end
b = 0;
end %This "end" ends the while loop for i

%% PostProcessing the data

zraw = z;

```

```

rrow = r;
uraw = u;
vraw = v;
New = 0;

for k = (JStart+numofinversepts+1):i
    K = k-New;
    shockdone = 0;
    check = 1;
    count = 1;
    for j = 1:JStart*10-1
        if u(j,K) == 0
            break
        end
        if z(1,K)<z(1,K-1)
            z(:,K) = [];
            r(:,K) = [];
            u(:,K) = [];
            v(:,K) = [];
            if K+1 <= length(Jbots_for_Cplus)
                Jbots_for_Cplus(K+1) = sum(Jbots_for_Cplus(K:K+1));
            end
            Jbots_for_Cplus(K) = [];
            New = New + 1;
            break
        end
        if z(j,K) > zlen+zduct+1
            z(:,K) = [];
            r(:,K) = [];
            u(:,K) = [];
            v(:,K) = [];
            Jbots_for_Cplus(K+1) = sum(Jbots_for_Cplus(K:K+1));
            Jbots_for_Cplus(K) = [];
            New = New + 1;
            break
        end
        if j > 1 && (z(j,K) < z(check,K))
            z(j,K) = z(check,K);
            r(j,K) = r(check,K);
            u(j,K) = u(check,K);
            v(j,K) = v(check,K);
            count = count-1;
        else
            check = j;
            count = 1;
        end
        if z(1,K) ~= zlen && shockdone == 0 && j>1
            L1 = [z(j-1:j,K)';r(j-1:j,K)'];
            L2 = [z(1:find(r(:,K-1)==0,1),K-1)';r(1:find(r(:,K-1)==0,1),K-1)'];
            P = InterX(L1,L2);
            if isempty(P) == 0
                ZZ = z(:,K-1)-P(1);
                count = find(ZZ>0,1);
                X = find(r(:,K-1)==0,1);
            end
        end
    end
end

```

```

        z(j:j+length(count:X)-1,K) = z(count:X,K-1);
        r(j:j+length(count:X)-1,K) = r(count:X,K-1);
        u(j:j+length(count:X)-1,K) = u(count:X,K-1);
        v(j:j+length(count:X)-1,K) = v(count:X,K-1);
        z(j+length(count:X):end,K) = 0;
        r(j+length(count:X):end,K) = 0;
        u(j+length(count:X):end,K) = 0;
        v(j+length(count:X):end,K) = 0;
        shockdone = 1;
    end
end
end
end

% Index for Characteristics in use
i_end = K;

%% Erasing of unused slots in variable matrices
% Index for Points on Characteristics in use
for J = 1:JStart*10
    if isempty(find(u(J,:),1)) == 1
        j_end = J-1;
        break
    end
end
end
% Erase unused Characteristics and Points
z = z(1:j_end,1:i_end);
r = r(1:j_end,1:i_end);
u = u(1:j_end,1:i_end);
v = v(1:j_end,1:i_end);

%% Smoothing of the boundaries:
% Some characteristics have less points than others. This code
requires
% the matrices to be square so the unused "slots" in each
characteristic
% must be set equal to the last valued point.
mark = ones(i_end,1);
z(:,1) = z(1,1);
r(:,1) = r(1,1);
u(:,1) = u(1,1);
v(:,1) = v(1,1);

for i = 2:i_end
    J = find(u(:,i)==0,1);
    if isempty(J) == 0
        z(J:end,i) = z(mark(i-1),i-1);
        r(J:end,i) = r(mark(i-1),i-1);
        u(J:end,i) = u(mark(i-1),i-1);
        v(J:end,i) = v(mark(i-1),i-1);
        mark(i) = J-1;
        if mark(i) == 0
            mark(i) = 1;
        end
    else

```

```

        mark(i) = j_end;
    end
end

%Calculate the state properties.
ZFBD = z(1,I+1:i_end); %Zp and Rp are the plume boundary coordinates
RFBD = r(1,I+1:i_end);
V = zeros(j_end,i_end);
a = zeros(j_end,i_end);
M = zeros(j_end,i_end);
P = zeros(j_end,i_end);
T = zeros(j_end,i_end);

for j=1:j_end
    for i=1:i_end
        V(j,i)=sqrt(u(j,i)^2+v(j,i)^2);
        [M(j,i), ~, a(j,i)] = THERMO_New(V(j,i),gamma,R,T0,CP);
        P(j,i)= P0/(1+((gamma-1)/2)*M(j,i)^2)^(gamma/(gamma-1));
        T(j,i)=T0/(1+((gamma-1)/2)*M(j,i)^2);
    end
end
MFBD = M(1,I+1:i_end);

VE = M(1,I)*sqrt(gamma*R*T(1,I));

if delta == 0
    mdot = (2*r_throat)/12*P0/sqrt(T0)*sqrt(gamma/R)*((gamma+1)/2)^(-(gamma+1)/(2*(gamma-1)));%[slug/s]
    AE = 2*Rl;
else
    mdot =
    (pi*r_throat^2)/144*P0/sqrt(T0)*sqrt(gamma/R)*((gamma+1)/2)^(-(gamma+1)/(2*(gamma-1))); %[slug/s]
    AE = Rl^2*pi;
end
F = mdot*VE;
FPexit = (P(1,I)-Pa)*AE/144;

coeffs = [coeff_a; coeff_b; coeff_c];
coeffsDuct = [coeff_a_duct; coeff_b_duct; coeff_c_duct];

end

% InverseWall logic description:
% The point of the InverseWall.m file is to manually specify more
points on the circular section of the
% throat from which C- lines emanate from. Left on its own, the MOC
calcs would create too few for a
% fine mesh. As a result of specifying the points on the wall some of
the C+ lines are ended before they
% reach the wall in the circular arc section. Ending C+ lines before
they reach the wall is okay in the
% circular section, but unfortunately this also means a large gap in
wall points occurs right as the

```

```
% InverseWall.m hands off the wall calcs to WallIntersection.m. To fix
this, a special indexing scheme
% was developed.
% Right after InverseWall.m calculates the values at the wall, the code
checks which C+ lines should be
% continued on from the last C- line. This causes the interiorpoint.m
calculations to begin with what
% would have been forgotten C+ points. Then the code continues with the
rest of the C+ points in the
% particular C- column. The amount of 'forgotten' C+ points is carried
over into the indexing for the
% next InverseWall.m calcs so that these C+ points will not be used to
find the wall point.
```

```
% Paul Riley
% April 16 2008 SENIOR PROJECT
% ADVISOR: Dr. David D. Marshall
% Aerospace Engineering Department
% California Polytechnic State University
% San Luis Obispo, CA 93407
%
% Modified by Brett Morham January 14, 2009
% April 16, 2009
% MASTER'S THESIS
% ADVISOR: Dr. Dianne D. Deturris
% Aerospace Engineering Department
% California Polytechnic State University
```

```
function [r dr_dz] =
AXI_Wall_New(z_tr,radius_of_curvature,r_th,coeff_a,coeff_b,coeff_c,z)
```

```
%Usage: This function calculates the radial coordinate on the wall and
the
%slope of the wall at that point.
```

```
%z_tr                The axial coordinate of the point that connects
the 2 wall curves
%radius_of_curvature The radius of curvature downstream of the
nozzle throat
%r_th                The radius of the nozzle throat
%coeff_a             The "a" coefficient in the polynomial  $r = a + bz + cz^2$ 
a+bz+cz^2
%coeff_b             The "b" coefficient in the polynomial  $r = a + bz + cz^2$ 
a+bz+cz^2
%coeff_c             The "c" coefficient in the polynomial  $r = a + bz + cz^2$ 
a+bz+cz^2
%z                  The axial coordinate whose corresponding wall
point you need
```

```
%The first (expanding) wall contour is assumed to be a circular contour
%that runs from the throat to the transition point. The second,
%flow straightening contour is assumed to be contour expressed by a
second
```

```

%order polynomial that runs from the transition point to the nozzle
exit lip.

if z<z_tr %these lines will run if you're in the circular throat arc
section of the diverging nozzle
    r = -sqrt(radius_of_curvature^2-z^2)+r_th+radius_of_curvature;
%wall r-coordinate
    dr_dz = -0.5*(radius_of_curvature^2-z^2)^(-0.5)*(-2*z); %wall slope
at the r-coordinate
else %these lines will run if you're in the straighting section of the
diverging nozzle
    r = coeff_a + coeff_b*z + coeff_c*z^2;
    dr_dz = coeff_b + 2*coeff_c*z;
end

end

function [u,v,z,r] =
Centerline_New_test(u,v,z,r,initial_val,gamma,R,T0,delta,i,j,CP)
count = 0;
uc=1;
%Initialize 'new' variables
u_new = initial_val; z_new = initial_val;

while abs(u(j,i)-u_new)>0.03*uc && abs(z(j,i)-z_new)>0.00003 || count
<= 1 %tolerances suggested on p.603 of Z&H Vol I (converted to english
units)
    if count==0;
        u_minus = u(j-1,i);
        v_minus = v(j-1,i);
        r_minus = r(j-1,i);
    else
        %Redefine the "corrected" values so that they're now the old
values.
        u(j,i) = u_new;
        z(j,i) = z_new;

        u_minus = (u(j-1,i)+u(j,i))/2;
        v_minus = (v(j-1,i)+0)/2;
        r_minus = (r(j-1,i)+0)/2;

        uc=abs(u(j,i));
    end
    [theta_minus a_minus mu_minus] =
Minus_Var1_New(u_minus,v_minus,gamma,R,T0,CP);
    [lambda_minus Q_minus R_minus S_minus] =
MinusCoeff_New(theta_minus,mu_minus,u_minus,v_minus,a_minus,r_minus,delta);

    z_new = z(j-1,i)-r(j-1,i)/lambda_minus;

    T_minus = S_minus*(z_new-z(j-1,i))+Q_minus*u(j-1,i)+R_minus*v(j-
1,i);
    u_new = T_minus/Q_minus;

```



```

        count = count+1;
    end

    u(j,i) = u_new;
    v(j,i) = 0;
    z(j,i) = z_new;
    r(j,i) = 0;

    if delta == 0
        if z(j,i) < z(find(r(:,i-1)==0,1),i-1)
            z(j,i) = z(find(r(:,i-1)==0,1),i-1);
        end
    end
end
end

function [T0,Gamma,R] = CombustionData(selection,P0,O_F_Ratio)
% This data comes from the Chemical Equilibrium Analysis (CEA) off the
site https://www.johnsonrockets.com.
% The columns of each T/Cp/Gamma matrix represent a different pressure
and the rows represent a different
% Oxidizer/Fuel ratio. The pressures range from 100-1200 psia for all
propellants and the Oxidizer/Fuel
% ratios vary a small amount from the stoichiometric Oxidizer/Fuel
ratio. This optimum ratio is the center
% value of the Oxidizer/Fuel array.

% All Fuel/Oxidizers have data for these pressures
Pressures = [100 120 140 160 180 200 220 240 260 280 300 320 340 360
380 400 420 440 460 480 500 520 540 560 580 600 620 640 660 680 700 720
740 760 780 800 820 840 860 880 900 920 940 960 980 1000 1020 1040 1060
1080 1100 1200]; %(psia)

switch (selection)
    case 1
        %% Methanol (Oxidizer is Air)
        Oxi_Fuel = [5.5 6 6.5 7 7.5];

        % Temperature (K)
        T = [2067.15 2067.35 2067.51 2067.64 2067.75 2067.84 2067.92
2067.99 2068.05 2068.10 2068.15 2068.19 2068.23 2068.27 2068.30
2068.33 2068.36 2068.39 2068.41 2068.44 2068.46 2068.48 2068.50
2068.52 2068.53 2068.55 2068.57 2068.58 2068.59 2068.61 2068.62
2068.63 2068.65 2068.66 2068.67 2068.68 2068.69 2068.70 2068.71
2068.72 2068.73 2068.74 2068.74 2068.75 2068.76 2068.77 2068.78
2068.78 2068.79 2068.80 2068.80 2068.83;
            2146.04 2146.68 2147.17 2147.57 2147.89 2148.17 2148.40
2148.60 2148.78 2148.94 2149.08 2149.21 2149.33 2149.44 2149.54
2149.63 2149.71 2149.79 2149.86 2149.93 2149.99 2150.05 2150.11
2150.16 2150.21 2150.26 2150.30 2150.35 2150.39 2150.43 2150.47
2150.50 2150.54 2150.57 2150.60 2150.63 2150.66 2150.69 2150.72
2150.74 2150.77 2150.79 2150.82 2150.84 2150.86 2150.89 2150.91
2150.93 2150.95 2150.97 2150.98 2151.07;
            2173.90 2176.11 2177.91 2179.42 2180.71 2181.84 2182.83
2183.72 2184.52 2185.25 2185.92 2186.53 2187.10 2187.62 2188.12
2188.58 2189.01 2189.42 2189.80 2190.17 2190.51 2190.84 2191.15

```

```

2191.45 2191.74 2192.01 2192.28 2192.53 2192.77 2193.00 2193.23
2193.44 2193.65 2193.86 2194.05 2194.24 2194.43 2194.60 2194.78
2194.94 2195.11 2195.27 2195.42 2195.57 2195.72 2195.86 2196.00
2196.13 2196.26 2196.39 2196.52 2197.11;
    2106.39 2107.46 2108.31 2109.01 2109.60 2110.10 2110.54
2110.93 2111.28 2111.59 2111.87 2112.13 2112.37 2112.59 2112.79
2112.97 2113.15 2113.31 2113.47 2113.61 2113.75 2113.88 2114.00
2114.12 2114.23 2114.33 2114.43 2114.53 2114.62 2114.71 2114.80
2114.88 2114.96 2115.03 2115.10 2115.18 2115.24 2115.31 2115.37
2115.43 2115.49 2115.55 2115.61 2115.66 2115.72 2115.77 2115.82
2115.87 2115.91 2115.96 2116.00 2116.21;
    2030.29 2030.84 2031.28 2031.65 2031.95 2032.21 2032.44
2032.64 2032.82 2032.98 2033.12 2033.26 2033.38 2033.49 2033.60
2033.69 2033.78 2033.87 2033.95 2034.02 2034.09 2034.16 2034.23
2034.29 2034.34 2034.40 2034.45 2034.50 2034.55 2034.60 2034.64
2034.68 2034.72 2034.76 2034.80 2034.84 2034.87 2034.91 2034.94
2034.97 2035.00 2035.04 2035.06 2035.09 2035.12 2035.15 2035.17
2035.20 2035.22 2035.25 2035.27 2035.38];

```

% Cp (KJ/(KG\*K))

```

Cp = [1.6120 1.6093 1.6072 1.6056 1.6042 1.6030 1.6020
1.6011 1.6004 1.5997 1.5991 1.5985 1.5980 1.5976 1.5971
1.5968 1.5964 1.5961 1.5957 1.5955 1.5952 1.5949 1.5947
1.5944 1.5942 1.5940 1.5938 1.5936 1.5934 1.5933 1.5931
1.5930 1.5928 1.5927 1.5925 1.5924 1.5923 1.5921 1.5920
1.5919 1.5918 1.5917 1.5916 1.5915 1.5914 1.5913 1.5912
1.5911 1.5910 1.5909 1.5908 1.5905;
    1.6590 1.6500 1.6430 1.6375 1.6330 1.6292 1.6259
1.6231 1.6207 1.6185 1.6165 1.6148 1.6132 1.6117 1.6104
1.6092 1.6080 1.6070 1.6060 1.6051 1.6042 1.6034 1.6027
1.6020 1.6013 1.6007 1.6001 1.5995 1.5989 1.5984 1.5979
1.5974 1.5970 1.5965 1.5961 1.5957 1.5953 1.5949 1.5946
1.5942 1.5939 1.5936 1.5932 1.5929 1.5926 1.5924 1.5921
1.5918 1.5916 1.5913 1.5911 1.5899;
    1.8912 1.8746 1.8612 1.8499 1.8403 1.8320 1.8246
1.8180 1.8120 1.8067 1.8017 1.7972 1.7930 1.7891 1.7855
1.7821 1.7789 1.7759 1.7731 1.7704 1.7678 1.7654 1.7631
1.7609 1.7588 1.7568 1.7548 1.7530 1.7512 1.7495 1.7478
1.7462 1.7447 1.7432 1.7417 1.7403 1.7390 1.7377 1.7364
1.7352 1.7340 1.7328 1.7317 1.7306 1.7295 1.7284 1.7274
1.7264 1.7254 1.7245 1.7236 1.7192;
    1.6984 1.6876 1.6790 1.6719 1.6660 1.6609 1.6565
1.6526 1.6491 1.6460 1.6432 1.6406 1.6382 1.6361 1.6340
1.6322 1.6304 1.6288 1.6273 1.6258 1.6245 1.6232 1.6220
1.6208 1.6198 1.6187 1.6177 1.6168 1.6158 1.6150 1.6141
1.6133 1.6126 1.6118 1.6111 1.6104 1.6097 1.6091 1.6085
1.6079 1.6073 1.6067 1.6062 1.6056 1.6051 1.6046 1.6041
1.6036 1.6032 1.6027 1.6023 1.6002;
    1.6066 1.6009 1.5964 1.5927 1.5896 1.5870 1.5847
1.5826 1.5808 1.5792 1.5778 1.5764 1.5752 1.5741 1.5731
1.5721 1.5712 1.5704 1.5696 1.5688 1.5682 1.5675 1.5669
1.5663 1.5657 1.5652 1.5647 1.5642 1.5637 1.5633 1.5628
1.5624 1.5620 1.5616 1.5613 1.5609 1.5606 1.5602 1.5599
1.5596 1.5593 1.5590 1.5587 1.5584 1.5582 1.5579 1.5577
1.5574 1.5572 1.5569 1.5567 1.5557];

```

```

% Gamma
G = [1.2415 1.2418 1.2421 1.2423 1.2425 1.2426 1.2428
1.2429 1.2430 1.2431 1.2432 1.2432 1.2433 1.2434 1.2434
1.2435 1.2435 1.2436 1.2436 1.2436 1.2437 1.2437 1.2437
1.2438 1.2438 1.2438 1.2439 1.2439 1.2439 1.2439 1.2440
1.2440 1.2440 1.2440 1.2440 1.2441 1.2441 1.2441 1.2441
1.2441 1.2441 1.2441 1.2442 1.2442 1.2442 1.2442 1.2442
1.2442 1.2442 1.2442 1.2443 1.2443;

1.2308 1.2319 1.2327 1.2334 1.2339 1.2344 1.2348
1.2351 1.2354 1.2357 1.2360 1.2362 1.2364 1.2366 1.2367
1.2369 1.2370 1.2372 1.2373 1.2374 1.2375 1.2376 1.2377
1.2378 1.2379 1.2380 1.2381 1.2381 1.2382 1.2383 1.2383
1.2384 1.2385 1.2385 1.2386 1.2386 1.2387 1.2387 1.2388
1.2388 1.2389 1.2389 1.2389 1.2390 1.2390 1.2391 1.2391
1.2391 1.2392 1.2392 1.2392 1.2394;

1.2048 1.2061 1.2073 1.2083 1.2091 1.2098 1.2105
1.2111 1.2116 1.2121 1.2126 1.2130 1.2134 1.2138 1.2141
1.2144 1.2147 1.2150 1.2153 1.2155 1.2158 1.2160 1.2162
1.2165 1.2167 1.2169 1.2170 1.2172 1.2174 1.2176 1.2177
1.2179 1.2181 1.2182 1.2183 1.2185 1.2186 1.2188 1.2189
1.2190 1.2191 1.2192 1.2194 1.2195 1.2196 1.2197 1.2198
1.2199 1.2200 1.2201 1.2202 1.2206;

1.2226 1.2237 1.2247 1.2254 1.2261 1.2267 1.2272
1.2276 1.2280 1.2284 1.2287 1.2290 1.2293 1.2295 1.2298
1.2300 1.2302 1.2304 1.2306 1.2307 1.2309 1.2310 1.2312
1.2313 1.2314 1.2316 1.2317 1.2318 1.2319 1.2320 1.2321
1.2322 1.2323 1.2324 1.2325 1.2326 1.2327 1.2327 1.2328
1.2329 1.2330 1.2330 1.2331 1.2332 1.2332 1.2333 1.2333
1.2334 1.2334 1.2335 1.2336 1.2338;

1.2333 1.2340 1.2345 1.2350 1.2354 1.2357 1.2360
1.2363 1.2365 1.2367 1.2369 1.2371 1.2372 1.2374 1.2375
1.2376 1.2378 1.2379 1.2380 1.2381 1.2382 1.2382 1.2383
1.2384 1.2385 1.2386 1.2386 1.2387 1.2387 1.2388 1.2389
1.2389 1.2390 1.2390 1.2391 1.2391 1.2392 1.2392 1.2393
1.2393 1.2393 1.2394 1.2394 1.2394 1.2395 1.2395 1.2395
1.2396 1.2396 1.2396 1.2397 1.2398];

case 2
%% Methanol (Oxidizer is Gaseous O2)
Oxi_Fuel = [.7 .9 1.2 1.5 1.8 2.3];

% Temperature (K)
T = [2183.59 2184.13 2184.55 2184.90 2185.18 2185.42 2185.63
2185.81 2185.97 2186.12 2186.25 2186.36 2186.47 2186.57 2186.66
2186.74 2186.82 2186.89 2186.96 2187.02 2187.08 2187.14 2187.19
2187.24 2187.29 2187.33 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0;

2717.91 2724.11 2729.11 2733.27 2736.79 2739.84 2742.52
2744.89 2747.01 2748.92 2750.66 2752.26 2753.72 2755.07 2756.33
2757.50 2758.59 2759.61 2760.58 2761.48 2762.34 2763.15 2763.92
2764.65 2765.35 2766.01 2766.65 2767.26 2767.84 2768.40 2768.93
2769.45 2769.94 2770.42 2770.88 2771.32 2771.75 2772.17 2772.57

```

```

2772.96 2773.33 2773.70 2774.05 2774.40 2774.73 2775.05 2775.37
2775.68 2775.98 2776.27 2776.55 2777.87;
    3037.75 3055.61 3070.66 3083.67 3095.12 3105.33 3114.54
3122.92 3130.62 3137.72 3144.32 3150.47 3156.24 3161.66 3166.78
3171.63 3176.23 3180.60 3184.77 3188.75 3192.56 3196.22 3199.73
3203.10 3206.35 3209.48 3212.51 3215.43 3218.25 3220.99 3223.64
3226.21 3228.71 3231.14 3233.49 3235.79 3238.02 3240.20 3242.32
3244.39 3246.41 3248.38 3250.30 3252.19 3254.03 3255.83 3257.59
3259.31 3261.00 3262.65 3264.27 3271.93;
    3081.12 3100.88 3117.63 3132.18 3145.04 3156.55 3166.98
3176.52 3185.30 3193.43 3201.01 3208.10 3214.77 3221.05 3227.00
3232.65 3238.02 3243.14 3248.04 3252.73 3257.23 3261.55 3265.71
3269.72 3273.58 3277.32 3280.93 3284.43 3287.83 3291.12 3294.31
3297.42 3300.44 3303.38 3306.24 3309.03 3311.75 3314.41 3317.00
3319.53 3322.01 3324.43 3326.80 3329.11 3331.38 3333.61 3335.79
3337.93 3340.02 3342.08 3344.10 3353.67;
    3063.19 3082.33 3098.55 3112.61 3125.03 3136.15 3146.21
3155.40 3163.86 3171.69 3178.98 3185.80 3192.21 3198.25 3203.97
3209.39 3214.54 3219.46 3224.15 3228.65 3232.96 3237.10 3241.08
3244.91 3248.61 3252.19 3255.64 3258.99 3262.23 3265.37 3268.42
3271.39 3274.27 3277.08 3279.81 3282.47 3285.06 3287.59 3290.06
3292.47 3294.83 3297.14 3299.39 3301.60 3303.76 3305.87 3307.94
3309.98 3311.97 3313.92 3315.84 3324.93;
    2996.76 3013.77 3028.13 3040.55 3051.48 3061.23 3070.04
3078.07 3085.44 3092.25 3098.58 3104.49 3110.03 3115.25 3120.17
3124.84 3129.27 3133.48 3137.50 3141.35 3145.03 3148.56 3151.96
3155.22 3158.37 3161.41 3164.34 3167.17 3169.92 3172.58 3175.15
3177.66 3180.09 3182.45 3184.75 3186.98 3189.16 3191.29 3193.36
3195.38 3197.35 3199.28 3201.17 3203.01 3204.81 3206.57 3208.30
3209.99 3211.65 3213.28 3214.87 3222.41];

% Cp (KJ/(KG*K))
Cp = [2.5761 2.5680 2.5616 2.5565 2.5523 2.5487 2.5455
2.5428 2.5404 2.5383 2.5364 2.5346 2.5330 2.5316 2.5302
2.5290 2.5278 2.5268 2.5258 2.5248 2.5239 2.5231 2.5223
2.5216 2.5209 2.5202 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0;
    3.4340 3.3604 3.3018 3.2535 3.2129 3.1781 3.1477
3.1209 3.0971 3.0757 3.0563 3.0386 3.0224 3.0075 2.9937
2.9809 2.9689 2.9577 2.9473 2.9374 2.9281 2.9194 2.9110
2.9032 2.8957 2.8885 2.8817 2.8752 2.8690 2.8630 2.8573
2.8518 2.8466 2.8415 2.8366 2.8319 2.8273 2.8229 2.8187
2.8146 2.8106 2.8068 2.8031 2.7994 2.7959 2.7925 2.7892
2.7860 2.7829 2.7798 2.7768 2.7631;
    7.1970 7.0097 6.8545 6.7223 6.6074 6.5060 6.4154
6.3336 6.2591 6.1907 6.1277 6.0692 6.0146 5.9636 5.9157
5.8705 5.8278 5.7873 5.7488 5.7122 5.6773 5.6439 5.6119
5.5813 5.5518 5.5235 5.4963 5.4700 5.4446 5.4201 5.3964
5.3734 5.3512 5.3296 5.3087 5.2883 5.2686 5.2493 5.2306
5.2124 5.1946 5.1773 5.1604 5.1439 5.1278 5.1121 5.0967
5.0817 5.0670 5.0526 5.0385 4.9722;
    8.1625 7.9851 7.8383 7.7135 7.6052 7.5098 7.4246
7.3478 7.2779 7.2138 7.1548 7.1001 7.0491 7.0014 6.9566

```

```

6.9144 6.8746 6.8369 6.8010 6.7669 6.7344 6.7033 6.6735
6.6450 6.6176 6.5913 6.5660 6.5415 6.5179 6.4952 6.4731
6.4518 6.4311 6.4111 6.3917 6.3728 6.3544 6.3366 6.3192
6.3023 6.2858 6.2697 6.2541 6.2388 6.2238 6.2092 6.1950
6.1810 6.1674 6.1540 6.1409 6.0794;
7.5149 7.3472 7.2084 7.0904 6.9879 6.8976 6.8168
6.7440 6.6777 6.6170 6.5609 6.5090 6.4605 6.4152 6.3727
6.3326 6.2947 6.2588 6.2247 6.1923 6.1613 6.1317 6.1034
6.0762 6.0501 6.0250 6.0008 5.9776 5.9551 5.9333 5.9123
5.8920 5.8723 5.8532 5.8346 5.8166 5.7991 5.7820 5.7654
5.7493 5.7335 5.7182 5.7032 5.6886 5.6743 5.6603 5.6467
5.6334 5.6203 5.6075 5.5950 5.5361;
6.1026 5.9593 5.8407 5.7400 5.6525 5.5755 5.5067
5.4447 5.3883 5.3367 5.2890 5.2449 5.2038 5.1654 5.1293
5.0953 5.0632 5.0328 5.0040 4.9766 4.9504 4.9254 4.9015
4.8785 4.8565 4.8354 4.8150 4.7954 4.7765 4.7582 4.7405
4.7234 4.7069 4.6908 4.6752 4.6601 4.6454 4.6311 4.6172
4.6037 4.5905 4.5776 4.5651 4.5529 4.5409 4.5293 4.5179
4.5067 4.4958 4.4852 4.4747 4.4256];

% Gamma
G = [1.2201 1.2206 1.2210 1.2214 1.2216 1.2219 1.2221
1.2222 1.2224 1.2225 1.2227 1.2228 1.2229 1.2230 1.2231
1.2231 1.2232 1.2233 1.2234 1.2234 1.2235 1.2235 1.2236
1.2236 1.2237 1.2237 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0;
1.1665 1.1684 1.1700 1.1713 1.1725 1.1735 1.1744
1.1752 1.1760 1.1766 1.1773 1.1778 1.1784 1.1789 1.1793
1.1797 1.1802 1.1805 1.1809 1.1812 1.1816 1.1819 1.1822
1.1824 1.1827 1.1830 1.1832 1.1834 1.1837 1.1839 1.1841
1.1843 1.1845 1.1847 1.1849 1.1850 1.1852 1.1854 1.1855
1.1857 1.1858 1.1860 1.1861 1.1863 1.1864 1.1865 1.1867
1.1868 1.1869 1.1870 1.1871 1.1877;
1.1197 1.1207 1.1215 1.1223 1.1229 1.1235 1.1240
1.1245 1.1250 1.1254 1.1258 1.1262 1.1265 1.1269 1.1272
1.1275 1.1278 1.1281 1.1283 1.1286 1.1288 1.1291 1.1293
1.1295 1.1297 1.1299 1.1301 1.1303 1.1305 1.1307 1.1309
1.1311 1.1312 1.1314 1.1316 1.1317 1.1319 1.1320 1.1322
1.1323 1.1325 1.1326 1.1328 1.1329 1.1330 1.1331 1.1333
1.1334 1.1335 1.1336 1.1338 1.1343;
1.1145 1.1152 1.1159 1.1164 1.1169 1.1174 1.1178
1.1181 1.1185 1.1188 1.1191 1.1193 1.1196 1.1198 1.1200
1.1202 1.1204 1.1206 1.1208 1.1210 1.1212 1.1213 1.1215
1.1216 1.1218 1.1219 1.1221 1.1222 1.1223 1.1224 1.1226
1.1227 1.1228 1.1229 1.1230 1.1231 1.1232 1.1233 1.1234
1.1235 1.1236 1.1237 1.1238 1.1238 1.1239 1.1240 1.1241
1.1242 1.1242 1.1243 1.1244 1.1247;
1.1149 1.1157 1.1164 1.1170 1.1175 1.1179 1.1184
1.1187 1.1191 1.1194 1.1197 1.1200 1.1203 1.1205 1.1207
1.1210 1.1212 1.1214 1.1216 1.1218 1.1219 1.1221 1.1223
1.1224 1.1226 1.1227 1.1229 1.1230 1.1232 1.1233 1.1234
1.1235 1.1237 1.1238 1.1239 1.1240 1.1241 1.1242 1.1243

```

```

1.1244  1.1245  1.1246  1.1247  1.1248  1.1249  1.1250  1.1251
1.1252  1.1252  1.1253  1.1254  1.1258;
      1.1184  1.1193  1.1200  1.1207  1.1213  1.1218  1.1223
1.1228  1.1232  1.1236  1.1239  1.1242  1.1246  1.1249  1.1251
1.1254  1.1257  1.1259  1.1261  1.1264  1.1266  1.1268  1.1270
1.1272  1.1273  1.1275  1.1277  1.1279  1.1280  1.1282  1.1283
1.1285  1.1286  1.1288  1.1289  1.1290  1.1292  1.1293  1.1294
1.1295  1.1297  1.1298  1.1299  1.1300  1.1301  1.1302  1.1303
1.1304  1.1305  1.1306  1.1307  1.1312];

```

case 3

```

%% Hydrazine (Oxidizer is N2O4) Note: N2O4 initial Temperature is
298.15 K

```

```

Oxi_Fuel = [.75 1 1.34 1.5 1.75];

```

```

% Temperature (K)

```

```

T = [2660.82  2665.95  2670.07  2673.49  2676.39  2678.88  2681.07
2683.01  2684.74  2686.30  2687.72  2689.02  2690.21  2691.31  2692.33
2693.28  2694.16  2694.99  2695.77  2696.51  2697.20  2697.86  2698.48
2699.08  2699.64  2700.18  2700.69  2701.18  2701.65  2702.11  2702.54
2702.96  2703.36  2703.74  2704.11  2704.47  2704.82  2705.16  2705.48
2705.79  2706.10  2706.39  2706.68  2706.96  2707.23  2707.49  2707.74
2707.99  2708.23  2708.47  2708.70  2709.76;
      2927.69  2939.37  2949.00  2957.15  2964.20  2970.38  2975.88
2980.81  2985.28  2989.36  2993.10  2996.55  2999.75  3002.74  3005.52
3008.14  3010.60  3012.92  3015.12  3017.20  3019.18  3021.06  3022.85
3024.57  3026.21  3027.78  3029.29  3030.73  3032.13  3033.47  3034.76
3036.01  3037.21  3038.37  3039.50  3040.59  3041.65  3042.67  3043.67
3044.63  3045.57  3046.48  3047.37  3048.23  3049.07  3049.89  3050.69
3051.47  3052.23  3052.97  3053.69  3057.08;
      3052.67  3069.83  3084.27  3096.71  3107.63  3117.36  3126.12
3134.09  3141.39  3148.12  3154.37  3160.19  3165.64  3170.76  3175.59
3180.16  3184.49  3188.61  3192.53  3196.28  3199.86  3203.30  3206.59
3209.76  3212.81  3215.75  3218.58  3221.32  3223.97  3226.54  3229.02
3231.43  3233.77  3236.04  3238.24  3240.39  3242.48  3244.52  3246.50
3248.43  3250.32  3252.16  3253.96  3255.72  3257.44  3259.12  3260.77
3262.38  3263.96  3265.50  3267.02  3274.16;
      3047.10  3064.16  3078.52  3090.89  3101.76  3111.44  3120.16
3128.09  3135.36  3142.07  3148.29  3154.09  3159.52  3164.63  3169.44
3173.99  3178.31  3182.42  3186.34  3190.07  3193.65  3197.08  3200.37
3203.53  3206.57  3209.51  3212.34  3215.08  3217.72  3220.29  3222.77
3225.18  3227.51  3229.78  3231.99  3234.14  3236.23  3238.27  3240.25
3242.19  3244.07  3245.92  3247.72  3249.48  3251.20  3252.89  3254.54
3256.15  3257.73  3259.28  3260.80  3267.96;
      3001.61  3016.88  3029.65  3040.62  3050.21  3058.73  3066.37
3073.30  3079.63  3085.45  3090.84  3095.85  3100.53  3104.92  3109.05
3112.94  3116.63  3120.13  3123.46  3126.64  3129.67  3132.57  3135.35
3138.01  3140.58  3143.04  3145.42  3147.71  3149.93  3152.07  3154.14
3156.15  3158.09  3159.98  3161.81  3163.59  3165.32  3167.01  3168.65
3170.24  3171.80  3173.32  3174.80  3176.24  3177.66  3179.04  3180.38
3181.70  3182.99  3184.26  3185.50  3191.32];

```

```

% Cp (KJ/(KG*K))

```

```

Cp = [3.2725  3.2160  3.1710  3.1339  3.1026  3.0757  3.0523
3.0316  3.0132  2.9966  2.9816  2.9679  2.9553  2.9437  2.9330

```

2.9231	2.9138	2.9051	2.8970	2.8893	2.8821	2.8753	2.8688
2.8626	2.8568	2.8512	2.8459	2.8408	2.8360	2.8313	2.8269
2.8226	2.8185	2.8145	2.8107	2.8070	2.8034	2.8000	2.7967
2.7935	2.7904	2.7874	2.7844	2.7816	2.7789	2.7762	2.7736
2.7711	2.7686	2.7662	2.7639	2.7531;			
	4.4800	4.3554	4.2542	4.1695	4.0971	4.0343	3.9788
3.9294	3.8850	3.8447	3.8080	3.7742	3.7431	3.7142	3.6873
3.6622	3.6387	3.6166	3.5957	3.5760	3.5573	3.5396	3.5227
3.5067	3.4914	3.4767	3.4627	3.4493	3.4364	3.4240	3.4120
3.4006	3.3895	3.3788	3.3685	3.3586	3.3489	3.3396	3.3305
3.3218	3.3133	3.3050	3.2970	3.2892	3.2816	3.2742	3.2670
3.2600	3.2532	3.2466	3.2401	3.2099;			
	6.2068	6.0527	5.9258	5.8184	5.7255	5.6439	5.5713
5.5059	5.4467	5.3925	5.3427	5.2966	5.2538	5.2138	5.1763
5.1411	5.1079	5.0765	5.0467	5.0184	4.9915	4.9658	4.9412
4.9177	4.8951	4.8735	4.8527	4.8326	4.8133	4.7947	4.7767
4.7593	4.7424	4.7261	4.7103	4.6950	4.6801	4.6656	4.6515
4.6378	4.6245	4.6115	4.5989	4.5865	4.5745	4.5628	4.5513
4.5401	4.5291	4.5184	4.5079	4.4587;			
	6.0431	5.8957	5.7744	5.6718	5.5831	5.5052	5.4359
5.3736	5.3172	5.2655	5.2181	5.1742	5.1334	5.0953	5.0597
5.0262	4.9946	4.9647	4.9364	4.9095	4.8839	4.8595	4.8362
4.8138	4.7924	4.7719	4.7521	4.7331	4.7147	4.6971	4.6800
4.6635	4.6475	4.6320	4.6171	4.6025	4.5884	4.5747	4.5614
4.5484	4.5358	4.5235	4.5115	4.4998	4.4884	4.4773	4.4665
4.4558	4.4455	4.4353	4.4254	4.3789;			
	5.2252	5.0915	4.9817	4.8889	4.8088	4.7386	4.6763
4.6203	4.5696	4.5233	4.4808	4.4416	4.4051	4.3712	4.3394
4.3096	4.2815	4.2549	4.2298	4.2059	4.1833	4.1616	4.1410
4.1213	4.1024	4.0842	4.0668	4.0501	4.0339	4.0184	4.0034
3.9889	3.9749	3.9613	3.9482	3.9354	3.9231	3.9111	3.8995
3.8881	3.8771	3.8664	3.8560	3.8458	3.8359	3.8262	3.8168
3.8076	3.7986	3.7898	3.7812	3.7409];			

% Gamma

G = [1.2018	1.2040	1.2057	1.2073	1.2086	1.2097	1.2107	
1.2116	1.2124	1.2132	1.2139	1.2145	1.2151	1.2156	1.2161
1.2166	1.2170	1.2174	1.2178	1.2182	1.2185	1.2189	1.2192
1.2195	1.2198	1.2201	1.2203	1.2206	1.2208	1.2211	1.2213
1.2215	1.2217	1.2219	1.2221	1.2223	1.2225	1.2227	1.2229
1.2230	1.2232	1.2233	1.2235	1.2236	1.2238	1.2239	1.2241
1.2242	1.2243	1.2244	1.2246	1.2251;			
	1.1595	1.1616	1.1634	1.1649	1.1663	1.1676	1.1687
1.1697	1.1706	1.1715	1.1723	1.1731	1.1738	1.1745	1.1751
1.1757	1.1763	1.1768	1.1773	1.1778	1.1783	1.1787	1.1792
1.1796	1.1800	1.1804	1.1807	1.1811	1.1814	1.1818	1.1821
1.1824	1.1827	1.1830	1.1833	1.1836	1.1839	1.1841	1.1844
1.1847	1.1849	1.1851	1.1854	1.1856	1.1858	1.1861	1.1863
1.1865	1.1867	1.1869	1.1871	1.1880;			
	1.1345	1.1357	1.1367	1.1376	1.1384	1.1391	1.1398
1.1403	1.1409	1.1414	1.1419	1.1423	1.1427	1.1431	1.1435
1.1439	1.1442	1.1445	1.1448	1.1451	1.1454	1.1457	1.1460
1.1462	1.1465	1.1467	1.1469	1.1472	1.1474	1.1476	1.1478
1.1480	1.1482	1.1484	1.1486	1.1487	1.1489	1.1491	1.1492

```

1.1494    1.1496    1.1497    1.1499    1.1500    1.1502    1.1503    1.1505
1.1506    1.1507    1.1509    1.1510    1.1516;
    1.1337    1.1349    1.1358    1.1367    1.1375    1.1382    1.1388
1.1394    1.1399    1.1404    1.1408    1.1413    1.1417    1.1420    1.1424
1.1427    1.1431    1.1434    1.1437    1.1440    1.1442    1.1445    1.1447
1.1450    1.1452    1.1454    1.1457    1.1459    1.1461    1.1463    1.1465
1.1467    1.1469    1.1470    1.1472    1.1474    1.1475    1.1477    1.1479
1.1480    1.1482    1.1483    1.1485    1.1486    1.1487    1.1489    1.1490
1.1491    1.1493    1.1494    1.1495    1.1501;
    1.1370    1.1383    1.1394    1.1404    1.1412    1.1420    1.1427
1.1434    1.1440    1.1445    1.1450    1.1455    1.1460    1.1464    1.1468
1.1472    1.1476    1.1479    1.1483    1.1486    1.1489    1.1492    1.1495
1.1498    1.1500    1.1503    1.1505    1.1508    1.1510    1.1512    1.1515
1.1517    1.1519    1.1521    1.1523    1.1525    1.1527    1.1528    1.1530
1.1532    1.1534    1.1535    1.1537    1.1539    1.1540    1.1542    1.1543
1.1545    1.1546    1.1547    1.1549    1.1555];

    case 4
%% Ethanol (Oxidizer is Air)
Oxi_Fuel = [7.5 8.5 9 9.5 10.5];

% Temperature (K)
T = [2102.52 2102.75 2102.94 2103.08 2103.20 2103.31 2103.40
2103.47 2103.54 2103.60 2103.66 2103.71 2103.75 2103.79 2103.83
2103.87 2103.90 2103.93 2103.96 2103.98 2104.01 2104.03 2104.06
2104.08 2104.10 2104.12 2104.13 2104.15 2104.17 2104.18 2104.20
2104.21 2104.23 2104.24 2104.25 2104.26 2104.27 2104.29 2104.30
2104.31 2104.32 2104.33 2104.34 2104.35 2104.36 2104.36 2104.37
2104.38 2104.39 2104.40 2104.40 2104.44;
    2216.85 2218.16 2219.18 2220.00 2220.67 2221.25 2221.74
2222.16 2222.54 2222.88 2223.18 2223.45 2223.69 2223.92 2224.12
2224.31 2224.49 2224.65 2224.80 2224.95 2225.08 2225.21 2225.32
2225.44 2225.54 2225.64 2225.74 2225.83 2225.92 2226.00 2226.08
2226.15 2226.22 2226.29 2226.36 2226.42 2226.49 2226.55 2226.60
2226.66 2226.71 2226.76 2226.81 2226.86 2226.91 2226.95 2227.00
2227.04 2227.08 2227.12 2227.16 2227.34;
    2226.91 2229.61 2231.82 2233.67 2235.26 2236.65 2237.89
2238.99 2239.98 2240.89 2241.72 2242.48 2243.19 2243.85 2244.46
2245.04 2245.58 2246.09 2246.58 2247.03 2247.47 2247.88 2248.28
2248.66 2249.02 2249.36 2249.69 2250.01 2250.32 2250.62 2250.90
2251.18 2251.44 2251.70 2251.95 2252.19 2252.42 2252.65 2252.87
2253.08 2253.29 2253.49 2253.69 2253.88 2254.07 2254.25 2254.42
2254.60 2254.77 2254.93 2255.09 2255.85;
    2180.98 2182.73 2184.14 2185.30 2186.29 2187.13 2187.88
2188.53 2189.12 2189.65 2190.13 2190.57 2190.97 2191.35 2191.69
2192.01 2192.31 2192.60 2192.86 2193.11 2193.35 2193.57 2193.78
2193.98 2194.17 2194.36 2194.53 2194.70 2194.86 2195.01 2195.16
2195.30 2195.44 2195.57 2195.70 2195.82 2195.94 2196.05 2196.16
2196.27 2196.37 2196.48 2196.57 2196.67 2196.76 2196.85 2196.94
2197.02 2197.11 2197.19 2197.26 2197.63;
    2064.07 2064.72 2065.23 2065.66 2066.01 2066.32 2066.58
2066.81 2067.02 2067.21 2067.38 2067.54 2067.68 2067.81 2067.93
2068.04 2068.15 2068.25 2068.34 2068.43 2068.51 2068.59 2068.66
2068.73 2068.80 2068.86 2068.92 2068.98 2069.04 2069.09 2069.14
2069.19 2069.24 2069.28 2069.33 2069.37 2069.41 2069.45 2069.49

```



2069.53 2069.56 2069.60 2069.63 2069.67 2069.70 2069.73 2069.76  
 2069.79 2069.82 2069.85 2069.87 2070.00];

% Cp (KJ/(KG\*K))

Cp = [1.5601 1.5573 1.5551 1.5533 1.5519 1.5507 1.5496  
 1.5487 1.5479 1.5472 1.5465 1.5459 1.5454 1.5449 1.5445  
 1.5441 1.5437 1.5433 1.5430 1.5427 1.5424 1.5421 1.5418  
 1.5416 1.5414 1.5412 1.5409 1.5407 1.5406 1.5404 1.5402  
 1.5400 1.5399 1.5397 1.5396 1.5394 1.5393 1.5392 1.5390  
 1.5389 1.5388 1.5387 1.5386 1.5385 1.5384 1.5383 1.5382  
 1.5381 1.5380 1.5379 1.5378 1.5374;  
 1.6930 1.6759 1.6626 1.6519 1.6431 1.6356 1.6291  
 1.6236 1.6186 1.6143 1.6104 1.6068 1.6036 1.6007 1.5980  
 1.5956 1.5933 1.5912 1.5892 1.5874 1.5856 1.5840 1.5825  
 1.5810 1.5797 1.5784 1.5772 1.5760 1.5749 1.5738 1.5728  
 1.5718 1.5709 1.5700 1.5692 1.5684 1.5676 1.5668 1.5661  
 1.5654 1.5647 1.5641 1.5634 1.5628 1.5622 1.5617 1.5611  
 1.5606 1.5600 1.5595 1.5590 1.5568;  
 1.9012 1.8826 1.8675 1.8548 1.8440 1.8345 1.8262  
 1.8187 1.8120 1.8059 1.8003 1.7951 1.7904 1.7860 1.7818  
 1.7780 1.7743 1.7709 1.7677 1.7646 1.7617 1.7590 1.7563  
 1.7538 1.7514 1.7491 1.7469 1.7448 1.7427 1.7408 1.7389  
 1.7371 1.7353 1.7336 1.7320 1.7304 1.7288 1.7273 1.7259  
 1.7244 1.7231 1.7217 1.7204 1.7192 1.7179 1.7167 1.7156  
 1.7144 1.7133 1.7122 1.7112 1.7062;  
 1.7465 1.7307 1.7180 1.7074 1.6985 1.6908 1.6841  
 1.6781 1.6727 1.6679 1.6635 1.6595 1.6558 1.6524 1.6493  
 1.6464 1.6436 1.6410 1.6386 1.6363 1.6342 1.6321 1.6302  
 1.6284 1.6266 1.6249 1.6233 1.6218 1.6203 1.6189 1.6176  
 1.6163 1.6150 1.6138 1.6127 1.6116 1.6105 1.6094 1.6084  
 1.6074 1.6065 1.6056 1.6047 1.6038 1.6030 1.6021 1.6013  
 1.6006 1.5998 1.5991 1.5984 1.5950;  
 1.5723 1.5659 1.5608 1.5567 1.5533 1.5503 1.5477  
 1.5455 1.5435 1.5417 1.5400 1.5386 1.5372 1.5360 1.5348  
 1.5337 1.5327 1.5318 1.5309 1.5301 1.5293 1.5286 1.5279  
 1.5272 1.5266 1.5260 1.5254 1.5249 1.5244 1.5239 1.5234  
 1.5229 1.5225 1.5221 1.5216 1.5212 1.5209 1.5205 1.5201  
 1.5198 1.5195 1.5191 1.5188 1.5185 1.5182 1.5179 1.5176  
 1.5174 1.5171 1.5168 1.5166 1.5154];

% Gamma

G = [1.2463 1.2467 1.2470 1.2472 1.2474 1.2476 1.2477  
 1.2479 1.2480 1.2481 1.2482 1.2482 1.2483 1.2484 1.2484  
 1.2485 1.2486 1.2486 1.2487 1.2487 1.2487 1.2488 1.2488  
 1.2488 1.2489 1.2489 1.2489 1.2490 1.2490 1.2490 1.2490  
 1.2491 1.2491 1.2491 1.2491 1.2492 1.2492 1.2492 1.2492  
 1.2492 1.2492 1.2493 1.2493 1.2493 1.2493 1.2493 1.2493  
 1.2493 1.2494 1.2494 1.2494 1.2494;  
 1.2232 1.2250 1.2265 1.2277 1.2287 1.2296 1.2303  
 1.2310 1.2316 1.2321 1.2326 1.2330 1.2334 1.2338 1.2341  
 1.2344 1.2347 1.2350 1.2352 1.2354 1.2357 1.2359 1.2361  
 1.2362 1.2364 1.2366 1.2367 1.2369 1.2370 1.2372 1.2373  
 1.2374 1.2375 1.2377 1.2378 1.2379 1.2380 1.2381 1.2382  
 1.2383 1.2383 1.2384 1.2385 1.2386 1.2387 1.2388 1.2388  
 1.2389 1.2390 1.2390 1.2391 1.2394;

```

1.2019 1.2034 1.2046 1.2057 1.2066 1.2074 1.2081
1.2088 1.2094 1.2099 1.2104 1.2109 1.2113 1.2117 1.2121
1.2125 1.2128 1.2131 1.2134 1.2137 1.2140 1.2143 1.2145
1.2147 1.2150 1.2152 1.2154 1.2156 1.2158 1.2160 1.2162
1.2164 1.2165 1.2167 1.2169 1.2170 1.2172 1.2173 1.2175
1.2176 1.2177 1.2179 1.2180 1.2181 1.2182 1.2184 1.2185
1.2186 1.2187 1.2188 1.2189 1.2194;
1.2152 1.2167 1.2180 1.2191 1.2200 1.2208 1.2215
1.2221 1.2227 1.2232 1.2237 1.2241 1.2245 1.2249 1.2252
1.2256 1.2259 1.2262 1.2264 1.2267 1.2269 1.2272 1.2274
1.2276 1.2278 1.2280 1.2282 1.2283 1.2285 1.2287 1.2288
1.2290 1.2291 1.2293 1.2294 1.2295 1.2297 1.2298 1.2299
1.2300 1.2301 1.2302 1.2303 1.2304 1.2305 1.2306 1.2307
1.2308 1.2309 1.2310 1.2311 1.2315;
1.2347 1.2355 1.2362 1.2367 1.2372 1.2375 1.2379
1.2382 1.2384 1.2387 1.2389 1.2391 1.2393 1.2394 1.2396
1.2397 1.2399 1.2400 1.2401 1.2402 1.2403 1.2404 1.2405
1.2406 1.2407 1.2408 1.2409 1.2409 1.2410 1.2411 1.2411
1.2412 1.2413 1.2413 1.2414 1.2414 1.2415 1.2415 1.2416
1.2416 1.2417 1.2417 1.2418 1.2418 1.2419 1.2419 1.2419
1.2420 1.2420 1.2420 1.2421 1.2422];

case 5
%% Ethanol (Oxidizer is Gaseous O2)
Oxi_Fuel = [1.5 1.8 2.1 2.4 2.7];

% Temperature (K)
T = [3126.49 3145.53 3161.56 3175.40 3187.56 3198.39 3208.15
3217.03 3225.17 3232.67 3239.64 3246.13 3252.20 3257.91 3263.29
3268.38 3273.21 3277.80 3282.17 3286.34 3290.33 3294.15 3297.81
3301.33 3304.72 3307.99 3311.14 3314.18 3317.12 3319.97 3322.72
3325.39 3327.98 3330.50 3332.94 3335.32 3337.63 3339.88 3342.08
3344.21 3346.30 3348.33 3350.32 3352.26 3354.16 3356.01 3357.82
3359.60 3361.33 3363.03 3364.70 3372.55;
3196.37 3218.57 3237.43 3253.82 3268.31 3281.31 3293.10
3303.88 3313.81 3323.01 3331.59 3339.63 3347.18 3354.31 3361.06
3367.47 3373.57 3379.38 3384.95 3390.27 3395.39 3400.30 3405.03
3409.59 3413.99 3418.24 3422.35 3426.34 3430.20 3433.95 3437.59
3441.12 3444.57 3447.91 3451.18 3454.36 3457.46 3460.49 3463.44
3466.33 3469.16 3471.92 3474.62 3477.26 3479.86 3482.39 3484.88
3487.32 3489.72 3492.06 3494.37 3505.30;
3201.42 3223.96 3243.13 3259.80 3274.57 3287.82 3299.84
3310.84 3320.99 3330.40 3339.17 3347.40 3355.13 3362.44 3369.36
3375.93 3382.19 3388.16 3393.87 3399.35 3404.60 3409.66 3414.53
3419.22 3423.75 3428.13 3432.37 3436.48 3440.46 3444.33 3448.09
3451.74 3455.30 3458.76 3462.13 3465.42 3468.63 3471.76 3474.82
3477.81 3480.74 3483.60 3486.40 3489.14 3491.83 3494.46 3497.04
3499.57 3502.06 3504.50 3506.89 3518.25;
3183.78 3205.82 3224.54 3240.83 3255.24 3268.17 3279.90
3290.63 3300.52 3309.69 3318.24 3326.26 3333.79 3340.91 3347.64
3354.04 3360.13 3365.94 3371.50 3376.83 3381.94 3386.85 3391.58
3396.15 3400.55 3404.81 3408.93 3412.92 3416.79 3420.55 3424.19
3427.74 3431.19 3434.55 3437.83 3441.02 3444.13 3447.17 3450.14
3453.04 3455.88 3458.65 3461.37 3464.03 3466.63 3469.18 3471.69
3474.14 3476.55 3478.91 3481.23 3492.24;

```

3156.23	3177.44	3195.44	3211.08	3224.92	3237.32	3248.56
3258.84	3268.30	3277.08	3285.26	3292.91	3300.11	3306.91
3313.34	3319.44	3325.25	3330.79	3336.08	3341.16	3346.02
3350.70	3355.20	3359.54	3363.73	3367.78	3371.70	3375.49
3379.16	3382.73	3386.19	3389.56	3392.83	3396.02	3399.12
3402.15	3405.10	3407.98	3410.79	3413.54	3416.22	3418.85
3421.42	3423.94	3426.40	3428.82	3431.18	3433.50	3435.78
3438.01	3440.20	3450.60];				

% Cp (KJ/(KG\*K))

Cp = [6.9429	6.7400	6.5721	6.4295	6.3058	6.1968	6.0996
6.0120	5.9324	5.8594	5.7923	5.7301	5.6722	5.6181
5.5674	5.5196	5.4746	5.4319	5.3915	5.3530	5.3164
5.2814	5.2479	5.2159	5.1852	5.1557	5.1273	5.0999
5.0736	5.0482	5.0236	4.9998	4.9768	4.9546	4.9330
4.9120	4.8917	4.8719	4.8527	4.8340	4.8158	4.7981
4.7808	4.7640	4.7475	4.7315	4.7158	4.7005	4.6856
4.6710	4.6567	4.5895;				
8.8049	8.5985	8.4277	8.2824	8.1562	8.0449	7.9456
7.8559	7.7743	7.6995	7.6304	7.5664	7.5068	7.4510
7.3986	7.3492	7.3025	7.2583	7.2163	7.1763	7.1381
7.1016	7.0667	7.0332	7.0011	6.9702	6.9404	6.9117
6.8839	6.8571	6.8312	6.8061	6.7818	6.7583	6.7354
6.7131	6.6915	6.6705	6.6500	6.6301	6.6107	6.5917
6.5732	6.5552	6.5376	6.5203	6.5035	6.4870	6.4709
6.4552	6.4397	6.3670;				
8.9140	8.7162	8.5524	8.4132	8.2924	8.1860	8.0909
8.0052	7.9271	7.8557	7.7897	7.7286	7.6717	7.6184
7.5684	7.5214	7.4769	7.4347	7.3947	7.3566	7.3202
7.2855	7.2523	7.2204	7.1898	7.1604	7.1321	7.1048
7.0784	7.0529	7.0283	7.0045	6.9814	6.9590	6.9373
6.9162	6.8956	6.8757	6.8563	6.8374	6.8189	6.8010
6.7834	6.7663	6.7496	6.7333	6.7174	6.7018	6.6865
6.6716	6.6569	6.5881;				
8.4190	8.2311	8.0755	7.9432	7.8283	7.7270	7.6365
7.5549	7.4806	7.4125	7.3497	7.2914	7.2371	7.1864
7.1387	7.0937	7.0513	7.0110	6.9728	6.9364	6.9017
6.8685	6.8368	6.8063	6.7771	6.7489	6.7219	6.6957
6.6705	6.6462	6.6226	6.5998	6.5777	6.5563	6.5355
6.5153	6.4956	6.4765	6.4579	6.4398	6.4222	6.4049
6.3882	6.3718	6.3557	6.3401	6.3248	6.3098	6.2952
6.2809	6.2669	6.2008;				
7.7697	7.5931	7.4470	7.3226	7.2146	7.1193	7.0342
6.9574	6.8874	6.8233	6.7642	6.7093	6.6582	6.6104
6.5654	6.5231	6.4831	6.4452	6.4091	6.3748	6.3421
6.3108	6.2809	6.2522	6.2246	6.1981	6.1725	6.1479
6.1241	6.1011	6.0789	6.0574	6.0365	6.0163	5.9967
5.9776	5.9591	5.9411	5.9235	5.9064	5.8897	5.8735
5.8576	5.8422	5.8270	5.8123	5.7978	5.7837	5.7699
5.7564	5.7431	5.6808];				

% Gamma

G = [1.1259	1.1271	1.1281	1.1290	1.1298	1.1305	1.1312
1.1318	1.1323	1.1328	1.1333	1.1338	1.1342	1.1346
1.1350	1.1354	1.1358	1.1361	1.1364	1.1367	1.1370
1.1373	1.1376	1.1379	1.1381	1.1384	1.1387	1.1389
1.1391	1.1394	1.1396	1.1398	1.1400	1.1402	1.1404
1.1406	1.1408	1.1410	1.1412	1.1413	1.1415	1.1417
1.1419	1.1420	1.1422	1.1423	1.1425	1.1427	1.1428
1.1430	1.1431	1.1438;				
1.1175	1.1183	1.1190	1.1196	1.1202	1.1206	1.1211
1.1215	1.1218	1.1222	1.1225	1.1228	1.1231	1.1233
1.1236						

```

1.1238 1.1240 1.1242 1.1244 1.1246 1.1248 1.1250 1.1252
1.1253 1.1255 1.1257 1.1258 1.1260 1.1261 1.1262 1.1264
1.1265 1.1266 1.1267 1.1269 1.1270 1.1271 1.1272 1.1273
1.1274 1.1275 1.1276 1.1277 1.1278 1.1279 1.1280 1.1281
1.1282 1.1283 1.1283 1.1284 1.1288;
    1.1158 1.1165 1.1172 1.1177 1.1182 1.1187 1.1191
1.1195 1.1198 1.1201 1.1204 1.1207 1.1209 1.1212 1.1214
1.1216 1.1218 1.1220 1.1222 1.1224 1.1225 1.1227 1.1229
1.1230 1.1232 1.1233 1.1234 1.1236 1.1237 1.1238 1.1239
1.1241 1.1242 1.1243 1.1244 1.1245 1.1246 1.1247 1.1248
1.1249 1.1250 1.1251 1.1252 1.1252 1.1253 1.1254 1.1255
1.1256 1.1256 1.1257 1.1258 1.1261;
    1.1156 1.1164 1.1170 1.1176 1.1181 1.1186 1.1190
1.1194 1.1197 1.1200 1.1203 1.1206 1.1208 1.1211 1.1213
1.1215 1.1217 1.1219 1.1221 1.1223 1.1225 1.1226 1.1228
1.1230 1.1231 1.1233 1.1234 1.1235 1.1237 1.1238 1.1239
1.1240 1.1241 1.1242 1.1244 1.1245 1.1246 1.1247 1.1248
1.1249 1.1250 1.1250 1.1251 1.1252 1.1253 1.1254 1.1255
1.1256 1.1256 1.1257 1.1258 1.1262;
    1.1161 1.1169 1.1176 1.1182 1.1187 1.1192 1.1196
1.1200 1.1204 1.1207 1.1210 1.1213 1.1216 1.1218 1.1220
1.1223 1.1225 1.1227 1.1229 1.1231 1.1233 1.1234 1.1236
1.1238 1.1239 1.1241 1.1242 1.1244 1.1245 1.1246 1.1248
1.1249 1.1250 1.1251 1.1252 1.1254 1.1255 1.1256 1.1257
1.1258 1.1259 1.1260 1.1261 1.1262 1.1262 1.1263 1.1264
1.1265 1.1266 1.1267 1.1268 1.1271];

```

end

```

T0 = interp2(Pressures,Oxi_Fuel,T,P0,O_F_Ratio)*9/5; %Tables in Kelvin,
Temperature needed in Rankine
Gamma = interp2(Pressures,Oxi_Fuel,G,P0,O_F_Ratio);
R = (interp2(Pressures,Oxi_Fuel,Cp,P0,O_F_Ratio)*(Gamma-
1)/Gamma)*(5*737.5621493*14.5939029/9); %Tables in (KJ/(KG*K)), R
needed in (ft-lbf/(slug*R))

```

end

```

function [z r M] = CPlus_Shifting(Jbots_for_Cplus,z,r,M)
%This function shifts the elements in each matrix so that the surface
plot
%will create a mesh grid which matches the characteristic net.
i_end = length(Jbots_for_Cplus);
for i = 1:i_end
    k = abs(Jbots_for_Cplus(i));
    if Jbots_for_Cplus(i) < 0
        for j = 1:k
            z(end+1,1:i-1) = z(1,1:i-1);
            z(end,i:end) = z(end-1,i:end);
            z(:,1:i-1) = z([end 1:end-1],1:i-1);

            r(end+1,1:i-1) = r(1,1:i-1);
            r(end,i:end) = r(end-1,i:end);
            r(:,1:i-1) = r([end 1:end-1],1:i-1);

```

```

        M(end+1,1:i-1) = M(1,1:i-1);
        M(end,i:end) = M(end-1,i:end);
        M(:,1:i-1) = M([end 1:end-1],1:i-1);
    end
elseif Jbots_for_Cplus(i) > 0
    for j = 1:k
        z(end+1,1:i-1) = z(end,1:i-1);
        z(end,i:end) = z(1,i:end);
        z(:,i:end) = z([end 1:end-1],i:end);

        r(end+1,1:i-1) = r(end,1:i-1);
        r(end,i:end) = r(1,i:end);
        r(:,i:end) = r([end 1:end-1],i:end);

        M(end+1,1:i-1) = M(end,1:i-1);
        M(end,i:end) = M(1,i:end);
        M(:,i:end) = M([end 1:end-1],i:end);
    end
end
end

end

function [dP] = FinalWave(theta,gamma,Nu1,P0,Pa)
% This code is utilized by LipPoint.m to find the last expansion wave
angle
% which equalizes the ambient and plume boundary static pressures.
%Note: The way the expansion wave was done is inefficient. It would be
%quicker and more accurate to divide up the expansion fan into equal
parts
%based on the initial pressure difference. However, this would not
make an
%equal mesh of C- lines in the fan.
[M2,~,~]=flowprandtlmeyer(gamma,Nu1+theta,'nu');
[~,~,PR2,~,~] = flowisentropic(gamma, M2, 'mach');
P2 = PR2*P0;
dP = abs(Pa-P2);

end

function [u,v,z,r] =
FreePressureBndry_New(u,v,z,r,initial_val,gamma,R,T0,delta,jbot,i,j,Zs,
Ps,P0,CP)
count=0;
uc=1;
vc=1;

%Initialize 'new' variables
u_new = initial_val; v_new = initial_val; z_new = initial_val; r_new =
initial_val;
% Necessary values from Gas Dynamics Vol. 2 pg 139 code
z3=z(j,i-1);
r3=r(j,i-1);
u3=u(j,i-1);
v3=v(j,i-1);

```

```

while (abs(u(j,i)-u_new)>0.03*uc && abs(v(j,i)-v_new)>0.03*vc) &&
abs(r(j,i)-r_new)>0.00003 || abs(z(j,i)-z_new)>0.00003 || count <= 1
%tolerances suggested on p.603 of Z&H Vol I
    if count==0
        u_plus = u(jbot,i-1);
        v_plus = v(jbot,i-1);
        r_plus = r(jbot,i-1);
    else
        %Redefine the "corrected" values so that they're now the old
values.
        u(j,i) = u_new;
        v(j,i) = v_new;
        z(j,i) = z_new;
        r(j,i) = r_new;

        u_plus = (u(jbot,i-1)+u(j,i))/2;
        v_plus = (v(jbot,i-1)+v(j,i))/2;
        r_plus = (r(jbot,i-1)+r(j,i))/2;

        uc=abs(u(j,i));
        vc=abs(v(j,i));
    end

    [theta_plus a_plus mu_plus] =
Plus_Var1_New(u_plus,v_plus,gamma,R,T0,CP);
    [lambda_plus Q_plus R_plus S_plus] =
PlusCoeff_New(theta_plus,mu_plus,u_plus,v_plus,a_plus,r_plus,delta);
    if count == 0
        lambda_0=v3/u3;
    else
        lambda_0=((v3+v_new)/2)/((u3+u_new)/2);
    end

    z_new = (r3-r(jbot,i-1)-lambda_0*z3+lambda_plus*z(jbot,i-
1))/(lambda_plus-lambda_0);
    r_new = r3+lambda_0*(z_new-z3);

    Pa = interp1(Zs,Ps,z_new,'linear','extrap');
    if isnan(Pa) == 1
        Pa = interp1(Zs,Ps,z(j,i-1));
    end
    Q4 = sqrt(2*CP*T0*(1-(Pa/P0)^((gamma-1)/gamma)));

    T_plus = S_plus*(z_new-z(jbot,i-1))+Q_plus*u(jbot,i-
1)+R_plus*v(jbot,i-1);

    u_new = (Q_plus*T_plus-R_plus*sqrt(Q4^2*(Q_plus^2+R_plus^2)-
T_plus^2))/(Q_plus^2+R_plus^2);
    if imag(u_new) ~= 0
        u_new = sqrt((real(u_new))^2+(imag(u_new))^2);
        disp('Imaginary')
    end
    v_new = (T_plus-Q_plus*u_new)/R_plus;

```

```

        count = count+1;
    end
    u(j,i) = u_new;
    v(j,i) = v_new;
    z(j,i) = z_new;
    r(j,i) = r_new;
end

function [u,v,z,r] =
InteriorPoint_New_test(u,v,z,r,initial_val,gamma,R,T0,delta,i,j,jbot,CP
)
count=0;
uc=1;
vc=1;

%Initialize 'new' variables
u_new = initial_val; v_new = initial_val; z_new = initial_val; r_new =
initial_val;

while (abs(u(j,i)-u_new)>0.03*uc && abs(v(j,i)-v_new)>0.03*vc) &&
abs(r(j,i)-r_new)>0.00003 || abs(z(j,i)-z_new)>0.00003 || count <= 1
%tolerances suggested on p.603 of Z&H Vol I
    if count==0
        u_minus = u(j-1,i);
        v_minus = v(j-1,i);
        r_minus = r(j-1,i);

        u_plus = u(jbot,i-1);
        v_plus = v(jbot,i-1);
        r_plus = r(jbot,i-1);
    else
        %Redefine the "corrected" values so that they're now the old
values.
        u(j,i) = u_new;
        v(j,i) = v_new;
        z(j,i) = z_new;
        r(j,i) = r_new;

        u_minus = (u(j-1,i)+u(j,i))/2;
        v_minus = (v(j-1,i)+v(j,i))/2;
        r_minus = (r(j-1,i)+r(j,i))/2;

        u_plus = (u(jbot,i-1)+u(j,i))/2;
        v_plus = (v(jbot,i-1)+v(j,i))/2;
        r_plus = (r(jbot,i-1)+r(j,i))/2;

        uc=abs(u(j,i));
        vc=abs(v(j,i));
    end
    [theta_minus a_minus mu_minus] =
Minus_Var1_New(u_minus,v_minus,gamma,R,T0,CP);
    [theta_plus a_plus mu_plus] =
Plus_Var1_New(u_plus,v_plus,gamma,R,T0,CP);

```

```

    [lambda_minus Q_minus R_minus S_minus] =
MinusCoeff_New(theta_minus,mu_minus,u_minus,v_minus,a_minus,r_minus,delta);
    [lambda_plus Q_plus R_plus S_plus] =
PlusCoeff_New(theta_plus,mu_plus,u_plus,v_plus,a_plus,r_plus,delta);
    if r_plus == 0
        S_plus = delta*a_plus^2*v_minus/r_minus;
    end

    z_new = (r(j-1,i)-r(jbot,i-1)-lambda_minus*z(j-1,i)+lambda_plus*z(jbot,i-1))/(lambda_plus-lambda_minus);
    r_new = r(j-1,i)+lambda_minus*(z_new-z(j-1,i));

    T_minus = S_minus*(z_new-z(j-1,i))+Q_minus*u(j-1,i)+R_minus*v(j-1,i);
    T_plus = S_plus*(z_new-z(jbot,i-1))+Q_plus*u(jbot,i-1)+R_plus*v(jbot,i-1);

    u_new = (T_minus*R_plus-T_plus*R_minus)/(Q_minus*R_plus-R_minus*Q_plus);
    v_new = (Q_minus*T_plus-Q_plus*T_minus)/(Q_minus*R_plus-R_minus*Q_plus);

    count = count+1;

    if r_new < 0 || count > 50
        [u,v,z,r] =
Centerline_New_test(u,v,z,r,initial_val,gamma,R,T0,delta,i,j,CP);
        return
    end
end

u(j,i) = u_new;
v(j,i) = v_new;
z(j,i) = z_new;
r(j,i) = r_new;

%If the C- characteristic crosses the previous one, then we use the
previous C-
%values as the new values for the rest of the current char line.
if delta == 0
    if z(j,i) < z(jbot,i-1)
        z(j:end-1,i) = z(jbot:end,i-1);
        r(j:end-1,i) = r(jbot:end,i-1);
    elseif z(j,i) < z(j-1,i)
        z(j,i) = z(j-1,i);
        r(j,i) = r(j-1,i);
    end
end
end
end

function P = InterX(L1,varargin) %%Note: Taken from Matlab File
Exchange
%INTERX Intersection of curves
% P = INTERX(L1,L2) returns the intersection points of two curves L1

```



```

% and L2. The curves L1,L2 can be either closed or open and are
described
% by two-row-matrices, where each row contains its x- and y-
coordinates.
% The intersection of groups of curves (e.g. contour lines, multiply
% connected regions etc) can also be computed by separating them with
a
% column of NaNs as for example
%
%       L = [x11 x12 x13 ... NaN x21 x22 x23 ...;
%            y11 y12 y13 ... NaN y21 y22 y23 ...]
%
% P has the same structure as L1 and L2, and its rows correspond to
the
% x- and y- coordinates of the intersection points of L1 and L2. If
no
% intersections are found, the returned P is empty.
%
% P = INTERX(L1) returns the self-intersection points of L1. To keep
% the code simple, the points at which the curve is tangent to itself
are
% not included. P = INTERX(L1,L1) returns all the points of the curve
% together with any self-intersection points.
%
% Example:
%       t = linspace(0,2*pi);
%       r1 = sin(4*t)+2;  x1 = r1.*cos(t); y1 = r1.*sin(t);
%       r2 = sin(8*t)+2;  x2 = r2.*cos(t); y2 = r2.*sin(t);
%       P = InterX([x1;y1],[x2;y2]);
%       plot(x1,y1,x2,y2,P(1,:),P(2:),'ro')

% Author : NS
% Version: 3.0, 21 Sept. 2010

% Two words about the algorithm: Most of the code is self-
explanatory.
% The only trick lies in the calculation of C1 and C2. To be brief,
this
% is essentially the two-dimensional analog of the condition that
needs
% to be satisfied by a function F(x) that has a zero in the interval
% [a,b], namely
%       F(a)*F(b) <= 0
% C1 and C2 exactly do this for each segment of curves 1 and 2
% respectively. If this condition is satisfied simultaneously for two
% segments then we know that they will cross at some point.
% Each factor of the 'C' arrays is essentially a matrix containing
% the numerators of the signed distances between points of one curve
% and line segments of the other.

%...Argument checks and assignment of L2
error(nargchk(1,2,nargin));
if nargin == 1,
    L2 = L1;    hF = @lt;    %...Avoid the inclusion of common
points

```

```

else
    L2 = varargin{1}; hF = @le;
end

%...Preliminary stuff
x1 = L1(1,:)'; x2 = L2(1,:);
y1 = L1(2,:)'; y2 = L2(2,:);
dx1 = diff(x1); dy1 = diff(y1);
dx2 = diff(x2); dy2 = diff(y2);

%...Determine 'signed distances'
S1 = dx1.*y1(1:end-1) - dy1.*x1(1:end-1);
S2 = dx2.*y2(1:end-1) - dy2.*x2(1:end-1);

C1 = feval(hF,D(bsxfun(@times,dx1,y2)-bsxfun(@times,dy1,x2),S1),0);
C2 = feval(hF,D((bsxfun(@times,y1,dx2)-
bsxfun(@times,x1,dy2))',S2'),0)';

%...Obtain the segments where an intersection is expected
[i,j] = find(C1 & C2);
if isempty(i),P = zeros(2,0);return; end;

%...Transpose and prepare for output
i=i'; dx2=dx2'; dy2=dy2'; S2 = S2';
L = dy2(j).*dx1(i) - dy1(i).*dx2(j);
i = i(L~=0); j=j(L~=0); L=L(L~=0); %...Avoid divisions by 0

%...Solve system of eqs to get the common points
P = unique([dx2(j).*S1(i) - dx1(i).*S2(j), ...
            dy2(j).*S1(i) - dy1(i).*S2(j)]./[L L],'rows'))';

function u = D(x,y)
    u = bsxfun(@minus,x(:,1:end-1),y).*bsxfun(@minus,x(:,2:end),y);
end

end

function [u,v,z,r] =
InverseWallIntersection_New(u,v,z,r,initial_val,gamma,R,T0,delta,i,j,co
eff_a,coeff_b,coeff_c,z_tr,curvature_down,r_throat,jbot,CP)
count = 0;
count2 = 0;
uc=1;
vc=1;

z_4 = z(j,i);
[r_4 dr_dz] =
AXI_Wall_New(z_tr,curvature_down,r_throat,coeff_a,coeff_b,coeff_c,z_4);
z_plus_new = initial_val;
lambda_minus = (r(j,i-1)-r(jbot,i-1))/(z(j,i-1)-z(jbot,i-1));

u_new = u(j,i-1);
v_new = v(j,i-1);
u_plus_new = u_new;
v_plus_new = v_new;
z_plus = z(j,i-1);

```

```

%For interpolations below
D_z = [z(j,i-1) z(jbot,i-1)];
D_u = [u(j,i-1) u(jbot,i-1)];
D_v = [v(j,i-1) v(jbot,i-1)];

while (abs(u(j,i)-u_new)>0.03*uc && abs(v(j,i)-v_new)>0.03*vc) || count
<= 1 %tolerances suggested on p.603 of Z&H Vol I
    while abs(z_plus_new-z_plus)>.0000003
        if count2 ~= 0
            z_plus = z_plus_new;
        end
        U = (u_plus_new+u_new)/2;
        V = (v_plus_new+v_new)/2;

        [theta_plus , ~, mu_plus] = Plus_Var1_New(U,V,gamma,R,T0,CP);
        lambda_plus = tan(theta_plus+mu_plus);

        z_plus_new = (r_4-r(jbot,i-1)+lambda_minus*z(jbot,i-1)-
lambda_plus*z_4)/(lambda_minus-lambda_plus);
        r_plus_new = r_4+lambda_plus*(z_plus_new-z_4);
        u_plus_new = interp1(D_z,D_u,z_plus_new);
        v_plus_new = interp1(D_z,D_v,z_plus_new);

        if count == 0
            u_new = u_plus_new;
            v_new = v_plus_new;
        end
        count2 = count2+1;
    end
    u(j,i) = u_new;
    v(j,i) = v_new;

    u_plus = (u_plus_new+u_new)/2;
    v_plus = (v_plus_new+v_new)/2;
    r_plus = (r_plus_new+r_4)/2;

    [theta_plus a_plus mu_plus] =
Plus_Var1_New(u_plus,v_plus,gamma,R,T0,CP);
    [~, Q_plus R_plus S_plus] =
PlusCoeff_New(theta_plus,mu_plus,u_plus,v_plus,a_plus,r_plus,delta);

    T_plus = S_plus*(z_4-
z_plus_new)+Q_plus*u_plus_new+R_plus*v_plus_new;

    u_new = T_plus/(Q_plus+dr_dz*R_plus);
    v_new = u_new*dr_dz;

    count = count+1;
    count2 = 0;
    uc = u_new;
    vc = v_new;
end
u(j,i) = u_new;
v(j,i) = v_new;

```

```

r(j,i) = r_4;
end

function [u,v,z,r,finalwave] =
LipPoint(u,v,z,r,gamma,R,T0,i,j,Zs,Ps,P0,CP)
% This function calculates what the flow does at the lip point. If the
% flow is underexpanded then a series of expansion waves occur to
equalize
% the static pressure of the plume with the secondary static pressure.
If
% the flow is only slightly overexpanded then a weak oblique shock
forms.
% The MOC can handle very weak oblique shocks so this means the
% calculations can continue without extra input.
finalwave = .5;

Pa = interp1(Zs,Ps,z(j,i-1),'linear','extrap');
V1 = sqrt(u(j,i-1)^2+v(j,i-1)^2);
[M1,~,~] = THERMO_New(V1,gamma,R,T0,CP);
[~,~,PR1,~,~] = flowisentropic(gamma, M1);
P1 = PR1*P0;

if P1 > Pa
    %Underexpanded Flow
    if finalwave == .5
        theta12 = 1;
    end
    [~,Nu1,~]=flowprandtlmeyer(gamma,M1);
    [M2,~,~]=flowprandtlmeyer(gamma,Nu1+theta12,'nu');
    [~,TR2,PR2,~,~] = flowisentropic(gamma, M2);
    P2 = PR2*P0;

    if P2 > Pa
        else
            % Need to find final wave which equalizes secondary flow with
plume by applying fminbnd
            finalwave = 1;
            theta12 = fminbnd(@(theta)
FinalWave(theta,gamma,Nu1,P0,Pa),0,theta12,optimset('TolX',1e-12));

            %Recompute values with new theta12
            [~,Nu1,~]=flowprandtlmeyer(gamma,M1);
            [M2,~,~]=flowprandtlmeyer(gamma,Nu1+theta12,'nu');
            [~,TR2,~,~,~] = flowisentropic(gamma, M2);
        end
    elseif (Pa-P1)/Pa < 1.5
        finalwave = 3;
        return
    else
        (Pa-P1)/Pa
        keyboard
        error('This code is designed for underexpanded or only slightly
overexpanded nozzles. Increase primary stagnation pressure.')
    end
end

```

```

T2 = TR2*T0;
a2 = sqrt(gamma*R*T2);
V2 = M2*a2;
theta1 = atand(v(j,i-1)/u(j,i-1));

u(j,i) = V2*cosd(theta1+theta12);
v(j,i) = V2*sind(theta1+theta12);
z(j,i) = z(j,i-1);
r(j,i) = r(j,i-1);

end

function [u,v,z,r] =
WallIntersection_New(u,v,z,r,initial_val,gamma,R,T0,delta,jbot,i,j,...

coeff_a,coeff_b,coeff_c,theta_attach,theta_exit,z_tr,curvature_down,r_t
hroat,zlen,CP)
global Free
global I

count=0;
uc=1;
vc=1;
%Initialize 'new' variables
u_new = initial_val; v_new = initial_val; z_new = initial_val; r_new =
initial_val;

%Direct Wall Point Method
while (abs(u(j,i)-u_new)>0.03*uc && abs(v(j,i)-v_new)>0.03*vc) &&
abs(r(j,i)-r_new)>0.00003 || abs(z(j,i)-z_new)>0.00003 || count <= 1
%tolerances suggested on p.603 of Z&H Vol I
    if count==0
        u_plus = u(jbot,i-1);
        v_plus = v(jbot,i-1);
        r_plus = r(jbot,i-1);
    else
        %Redefine the "corrected" values so that they're now the old
values.
        u(j,i) = u_new;
        v(j,i) = v_new;
        z(j,i) = z_new;
        r(j,i) = r_new;

        u_plus = (u(jbot,i-1)+u(j,i))/2;
        v_plus = (v(jbot,i-1)+v(j,i))/2;
        r_plus = (r(jbot,i-1)+r(j,i))/2;

        uc=abs(u(j,i));
        vc=abs(v(j,i));
    end
    [theta_plus a_plus mu_plus] =
Plus_Var1_New(u_plus,v_plus,gamma,R,T0,CP);
    [lambda_plus Q_plus R_plus S_plus] =
PlusCoeff_New(theta_plus,mu_plus,u_plus,v_plus,a_plus,r_plus,delta);

```

```

        if z(jbot,i-1) < z_tr
            % Finds intersection between C+ line and circle
            rintercept = r(jbot,i-1)-lambda_plus*z(jbot,i-1);
            [zintersect, ~] =
linecirc(lambda_plus,rintercept,0,(r_throat+curvature_down),curvature_d
own);
            z_new = min(zintersect);
        elseif theta_attach == theta_exit
            %-linearly diverging nozzle
            z_new = (coeff_a-r(jbot,i-1)+lambda_plus*z(jbot,i-
1))/(lambda_plus-coeff_b);
        else
            %-nonlinear diverging nozzle (quadratic formula) pg 127 Gas
dynamics Vol. 2
            z_new = ((lambda_plus-coeff_b)-sqrt((lambda_plus-coeff_b)^2-
4*coeff_c*(coeff_a-r(jbot,i-1)+lambda_plus*z(jbot,i-1))))/(2*coeff_c);
            end

            [r_new dr_dz] =
AXI_Wall_New(z_tr,curvature_down,r_throat,coeff_a,coeff_b,coeff_c,z_new
);

            T_plus = S_plus*(z_new-z(jbot,i-1))+Q_plus*u(jbot,i-
1)+R_plus*v(jbot,i-1);

            u_new = T_plus/(Q_plus+dr_dz*R_plus);
            v_new = u_new*dr_dz;

            count = count+1;
        end
        if z_new > zlen && Free == 0
            Free = 1;
            I = i;
        end

        u(j,i) = u_new;
        v(j,i) = v_new;
        z(j,i) = z_new;
        r(j,i) = r_new;
    end

function [theta_minus a_minus mu_minus] =
Minus_Var1_New(u_minus,v_minus,gamma,RG,T0,CP)
Q = sqrt(u_minus^2 + v_minus^2);
theta_minus = atan(v_minus/u_minus);%(radians)
[M_minus, ~, a_minus] = THERMO_New(Q,gamma,RG,T0,CP);
mu_minus = asin(1/M_minus);%(radians)%Note: Book uses alpha instead of
mu
end

function [u,v,z,r] =
WallIntersection_New(u,v,z,r,initial_val,gamma,R,T0,delta,jbot,i,j,...

coeff_a,coeff_b,coeff_c,theta_attach,theta_exit,z_tr,curvature_down,r_t
hroat,zlen,CP)

```

```

global Free
global I

count=0;
uc=1;
vc=1;
%Initialize 'new' variables
u_new = initial_val; v_new = initial_val; z_new = initial_val; r_new =
initial_val;

%Direct Wall Point Method
while (abs(u(j,i)-u_new)>0.03*uc && abs(v(j,i)-v_new)>0.03*vc) &&
abs(r(j,i)-r_new)>0.00003 || abs(z(j,i)-z_new)>0.00003 || count <= 1
%tolerances suggested on p.603 of Z&H Vol I
    if count==0
        u_plus = u(jbot,i-1);
        v_plus = v(jbot,i-1);
        r_plus = r(jbot,i-1);
    else
        %Redefine the "corrected" values so that they're now the old
values.
        u(j,i) = u_new;
        v(j,i) = v_new;
        z(j,i) = z_new;
        r(j,i) = r_new;

        u_plus = (u(jbot,i-1)+u(j,i))/2;
        v_plus = (v(jbot,i-1)+v(j,i))/2;
        r_plus = (r(jbot,i-1)+r(j,i))/2;

        uc=abs(u(j,i));
        vc=abs(v(j,i));
    end
    [theta_plus a_plus mu_plus] =
Plus_Var1_New(u_plus,v_plus,gamma,R,T0,CP);
    [lambda_plus Q_plus R_plus S_plus] =
PlusCoeff_New(theta_plus,mu_plus,u_plus,v_plus,a_plus,r_plus,delta);

    if z(jbot,i-1) < z_tr
        % Finds intersection between C+ line and circle
        rintercept = r(jbot,i-1)-lambda_plus*z(jbot,i-1);
        [zintersect, ~] =
linecirc(lambda_plus,rintercept,0,(r_throat+curvature_down),curvature_d
own);
        z_new = min(zintersect);
    elseif theta_attach == theta_exit
        %-linearly diverging nozzle
        z_new = (coeff_a-r(jbot,i-1)+lambda_plus*z(jbot,i-
1))/(lambda_plus-coeff_b);
    else
        %-nonlinear diverging nozzle (quadratic formula) pg 127 Gas
dynamics Vol. 2
        z_new = ((lambda_plus-coeff_b)-sqrt((lambda_plus-coeff_b)^2-
4*coeff_c*(coeff_a-r(jbot,i-1)+lambda_plus*z(jbot,i-1))))/(2*coeff_c);
    end
end

```

```

    [r_new dr_dz] =
AXI_Wall_New(z_tr,curvature_down,r_throat,coeff_a,coeff_b,coeff_c,z_new
);

    T_plus = S_plus*(z_new-z(jbot,i-1))+Q_plus*u(jbot,i-
1)+R_plus*v(jbot,i-1);

    u_new = T_plus/(Q_plus+dr_dz*R_plus);
    v_new = u_new*dr_dz;

    count = count+1;
end
if z_new > zlen && Free == 0
    Free = 1;
    I = i;
end

u(j,i) = u_new;
v(j,i) = v_new;
z(j,i) = z_new;
r(j,i) = r_new;
end

function [P,Z] = MultipleValues(P,Z)
% This function gets rid of multiple values in Ps/Zs due to the
expansion fan (necessary for interp1)

[Zsort, Zsortind] = sort(Z);
UniqueZ(Zsortind) = ([1; diff(Zsort)] ~= 0);

Z = Z(UniqueZ);
P = P(UniqueZ);

end

function [theta_plus a_plus mu_plus] =
Plus_Var1_New(u_plus,v_plus,gamma,RG,T0,CP)
Q = sqrt(u_plus^2 + v_plus^2);
theta_plus = atan(v_plus/u_plus);%(radians)
[M_plus, ~, a_plus] = THERMO_New(Q,gamma,RG,T0,CP);
mu_plus = asin(1/M_plus);%(radians)%Note: Book uses alpha instead of mu
end

function [lambda_plus Q_plus R_plus S_plus] =
PlusCoeff_New(theta_plus,mu_plus,u_plus,v_plus,a_plus,r_plus,delta)
lambda_plus = tan(theta_plus+mu_plus);%theta & mu in radians
Q_plus = u_plus^2-a_plus^2;
R_plus = 2*u_plus*v_plus-Q_plus*lambda_plus;
S_plus = delta*a_plus^2*v_plus/r_plus;
end

%Initial Data Line

%This function generates an initial data line using Sauer's method. The

```



```

%method of characteristics can begin once an initial data line is
%determined. Sauer's method is capable of handling throats with a ratio
of
%upstream throat radius of curvature to throat radius that is >= 2.
%This is from section 15-5 in Gas Dynamics Vol 2, Zucrow and Hoffman.

function [x y u_tilda v_tilda] =
SauersInitialLine_New(gamma,R,curvature_up,r_throat,T0,JMAX,delta)

% gamma = ratio of specific heats for the working fluid (combustion
%           products of LOX and LH2 in the case of SSME)
% R = gas constant for the working fluid (ft-lbf/slug R)
% curvature_up = radius of curvature immediately upstream from the
%           throat (ft)
% r_throat = throat radius (ft)
% T0 = chamber stagnation temperature (deg R)
% JMAX = number of points in the initial column
% Note: u_tilda and v_tilda are x and y components of velocity. The
% "tilda"
% is from the book notation to designate that they are NOT perturbation
% velocities.

epsilon=-
r_throat/(2*(3+delta))*sqrt((gamma+1)*(1+delta)/(curvature_up/r_throat)
);

a_star=sqrt(2*gamma*R*T0/(gamma+1));

alpha=sqrt((1+delta)/((gamma+1)*curvature_up*r_throat));

y=linspace(0,r_throat,JMAX);

x_untransformed=-(gamma+1)*alpha*y.^2/(2*(3+delta)); %This x needs a
transformation of x-E
x_untransformed(1) = x_untransformed(1)+.00001; %This is here so that
the centerline point's M ~ 1
x=x_untransformed-epsilon;

uprime=alpha*x_untransformed+(gamma+1)*alpha^2*y.^2/(2*(1+delta));
u_tilda=a_star*(1+uprime);

v_tilda=zeros(size(u_tilda)); % By definition, the initial data line we
are using is where v_tilda=0.

%Comment this section out for Zucrow IV Line Validation (4 lines below)
x(2) = [];
y(2) = [];
u_tilda(2) = [];
v_tilda(2) = [];

% plot(x/r_throat,y/r_throat)
% axis([-0.5 0.5 0 1])
end

```

```

function [Ms Ps Zs mdotstar Zd Rd Pd Md Ath,F] =
SSFLOW2(Zp,Rp,Mp,Zl,Rl,...

rduct,coeffDuct,gamma,pa,p0p,Ta,R,rhoa,tb,delta,zlen,mdotp,Rrp,T0p,gamm
ap,Min)

% Note: This code REQUIRES the input conditions to be underexpanded or
else
% it will not calculate properly (and it won't error out). It also sets
% Mach=1 at the minimum area point in the secondary flow. It should
check
% the pressure ratio to make sure its .5238 or whatever is actually
% required for M=1 flow. Should also determine if flow goes subsonic or
% supersonic after secondary throat.

%%%Inputs
%Zp                plume boundary location in axial direction [in]
%Rp                plume boundary location in transverse direction
from centerline [in]
%Zl                nozzle lip location in axial direction [in]
%Rl                nozzle lip location in transverse direction [in]
%rduct             duct half width from centerline [in]
%theta_attach_duct Angle of wall divergence at end of nozzle [degrees]
%theta_exit_duct   Angle of wall divergence at end of duct [degrees]
%gamma             Ratio of specific heats
%pa                Ambient inlet pressure [psi]
%Ta                Ambient inlet Temperature [R]
%R                 Gas constant
%rhoa              Ambient inlet density [slug/ft3]
%tb                Thickness of nozzle lip [in]
%delta             0 for 2-D, 1 for Axisymmetric. This affects Area
calcs.

%%%Outputs
%Ms                Secondary Mach Number Distribution
%Ps                Secondary Pressure Distribution [psi]
%Zs                Streamwise location corresponding to pressure and
Mach numbers [in]
%mdotstar          Mass flowrate of secondary stream [lbm/s-ft]

%%% Plotting outputs
%Zd
%Rd                Coordinates of duct walls [in]
%Pd                Vector of pressures of secondary flow [psi]
%Md                Vector of Mach numbers of secondary flow
%At                Secondary Aerodynamic throat width

% Unit Conversions
Zp = Zp/12;
Rp = Rp/12;
Zl = Zl/12;
Rl = Rl/12;
rduct = rduct/12;
pa = pa*144;
tb = tb/12;

```

```

Rl = Rl+tb; %correct for nozzle base thickness

M = zeros(length(Zp)+2,1);
A = M;
Pdratios = A;

RD(1:2) = rduct;
RD(3:length(M)) =
(coeffDuct(1)+coeffDuct(2)*Zp*12+coeffDuct(3)*(Zp*12).^2)/12;

if delta == 0
    A(1:2) = RD(1:2)-Rl;
    for i = 3:length(Rp)+2
        if Rp(i-2) < Rl
            A(i) = RD(i)-Rl;
        else
            A(i) = RD(i)-Rp(i-2);
        end
    end
else
    A(1:2) = pi*(RD(1:2).^2-Rl^2);
    for i = 3:length(Rp)+2
        if Rp(i-2) < Rl
            A(i) = pi*(RD(i)^2-Rl^2);
        else
            A(i) = pi*(RD(i)^2-Rp(i-2)^2);
        end
    end
end

for i=1:length(A)
    if A(i) <= 0
        A(i) = eps;
        display('primary plume intersects duct wall')
    end
end

[Ath t] = min(A); %determine the throat dimensions
if t<3
    t=3;
end
[~,~,Pcheck,~,~] = flowisentropic(gamma,Mp(t-2));
[M(t),~,~,~,~] = flowisentropic(gamma,p0p*Pcheck/pa,'pres');
%p0p*Pcheck is the static pressure of the boundary of the plume and
secondary flow
if M(t) >= 1
    x2 = 'sup';
else
    x2 = 'sub';
    [~,~,~,~,Aratio] = flowisentropic(gamma,M(t));
    Ath = Aratio*Ath;
end
X = 'sub';

```

```

if Ath <= eps
    M(t) = 0;
end
for i = 1:length(A)
    if i == t
        X = x2;
    end
    [M(i),~,Pdratios(i),~,~] = flowisentropic(gamma,A(i)/Ath,X);
end
Pd = pa*Pdratios;

[~,ind] = max(Zp(Zp-2.25/12<0));
if Zp(ind) ~= Zp(end) && Zp(ind+1) > 2.25/12
    re = interp1(Zp(ind:ind+1),RD(2+ind:3+ind),2.25/12);
    if RD(2+ind) == RD(3+ind)
        ADE = A(2+ind);
    else
        ADE = interp1(RD(2+ind:3+ind),A(2+ind:3+ind),re);
    end
    [~,~,Pdratiose,~,~] = flowisentropic(gamma,ADE/Ath,'sup');
    Pe = Pdratiose*pa;
else
    ADE = A(end);
    Pe = Pd(end);
end

% Mass Flow Rate Calculation
[~,Tratio,~, rhoratio,~] = flowisentropic(gamma,M(t));
rhostar = rhoa*rhoratio; %[slg/ft^3]
Tstar = Ta*Tratio;
Vstar = sqrt(gamma*R*Tstar);
mdotstar = rhostar*Ath*Vstar; %[slg/s]

% The final step is to plot the outputs. In order to do this, we need a
mesh of points. To do this we augment
% the Rp,Zp,Pd, and M vectors to make them matrices and simply add the
points along the ejector shroud in the
% same z location. In short, duplicate Zp,Pd, and Md column in the next
row. New Rp column is half width of duct.

Zd = zeros(length(Zp)+2,2);
Rd = zeros(size(Zd));
Md = zeros(size(Zd));

Zd(1,:) = 0;
Zd(2,:) = Z1;
Zd(3:end,1) = Zp;
Zd(3:end,2) = Zp;
Rd(1:2,1) = R1;
Rd(3:end,1) = Rp;
for i=1:length(Rd(:,1))
    if Rd(i,1) < R1
        Rd(i,1) = R1;
    end
end

```

```

end
Rd(:,2) = RD;
Md(:,1) = M;
Md(:,2) = M;

[~,Tratioend,~,~,~] = flowisentropic(gamma,M(end));
[~,Tratioin,~,rhoratioIN,~] = flowisentropic(gamma,Min);
[~,Tratioendp,~,~,~] = flowisentropic(gammap,Mp(end));
VE = M(end)*sqrt(gamma*R*Ta*Tratioend);
VIN = Min*sqrt(gamma*R*Ta*Tratioin);
VEp = M(end)*sqrt(gammap*Rp*T0p*Tratioendp);
mdotin = A(1)*VIN*rhoa*rhoratioIN;

F = (mdotstar*VE+mdotp*VEp-mdotin*VIN+(Pe-pa)*ADE)*.298; %unit
conversion

Zd = Zd*12;
Rd = Rd*12;
Ms = Md(:,1);
Ps = Pd(3:end,1)/144;
Zs = Zd(3:end,1);
if abs(Zs(1)-zlen) < 1e-9
    Zs(1) = zlen;
end
end

function [M T C] = THERMO_New(Q,gamma,RG,T0,CP)
T = T0-Q.^2/(2*CP);
C = sqrt(gamma*RG*T);
M = Q./C;
end

function [u,v,z,r] =
WallIntersection_New(u,v,z,r,initial_val,gamma,R,T0,delta,jbot,i,j,...

coeff_a,coeff_b,coeff_c,theta_attach,theta_exit,z_tr,curvature_down,r_t
hroat,zlen,CP)
global Free
global I

count=0;
uc=1;
vc=1;
%Initialize 'new' variables
u_new = initial_val; v_new = initial_val; z_new = initial_val; r_new =
initial_val;

%Direct Wall Point Method
while (abs(u(j,i)-u_new)>0.03*uc && abs(v(j,i)-v_new)>0.03*vc) &&
abs(r(j,i)-r_new)>0.00003 || abs(z(j,i)-z_new)>0.00003 || count <= 1
%tolerances suggested on p.603 of Z&H Vol I
    if count==0
        u_plus = u(jbot,i-1);
        v_plus = v(jbot,i-1);

```

```

        r_plus = r(jbot,i-1);
    else
        %Redefine the "corrected" values so that they're now the old
values.
        u(j,i) = u_new;
        v(j,i) = v_new;
        z(j,i) = z_new;
        r(j,i) = r_new;

        u_plus = (u(jbot,i-1)+u(j,i))/2;
        v_plus = (v(jbot,i-1)+v(j,i))/2;
        r_plus = (r(jbot,i-1)+r(j,i))/2;

        uc=abs(u(j,i));
        vc=abs(v(j,i));
    end
    [theta_plus a_plus mu_plus] =
Plus_Var1_New(u_plus,v_plus,gamma,R,T0,CP);
    [lambda_plus Q_plus R_plus S_plus] =
PlusCoeff_New(theta_plus,mu_plus,u_plus,v_plus,a_plus,r_plus,delta);

    if z(jbot,i-1) < z_tr
        % Finds intersection between C+ line and circle
        rintercept = r(jbot,i-1)-lambda_plus*z(jbot,i-1);
        [zintersect, ~] =
linecirc(lambda_plus,rintercept,0,(r_throat+curvature_down),curvature_d
own);
        z_new = min(zintersect);
    elseif theta_attach == theta_exit
        %-linearly diverging nozzle
        z_new = (coeff_a-r(jbot,i-1)+lambda_plus*z(jbot,i-
1))/(lambda_plus-coeff_b);
    else
        %-nonlinear diverging nozzle (quadratic formula) pg 127 Gas
dynamics Vol. 2
        z_new = ((lambda_plus-coeff_b)-sqrt((lambda_plus-coeff_b)^2-
4*coeff_c*(coeff_a-r(jbot,i-1)+lambda_plus*z(jbot,i-1))))/(2*coeff_c);
    end

    [r_new dr_dz] =
AXI_Wall_New(z_tr,curvature_down,r_throat,coeff_a,coeff_b,coeff_c,z_new
);

    T_plus = S_plus*(z_new-z(jbot,i-1))+Q_plus*u(jbot,i-
1)+R_plus*v(jbot,i-1);

    u_new = T_plus/(Q_plus+dr_dz*R_plus);
    v_new = u_new*dr_dz;

    count = count+1;
end
if z_new > zlen && Free == 0
    Free = 1;
    I = i;
end

```

```
u(j,i) = u_new;  
v(j,i) = v_new;  
z(j,i) = z_new;  
r(j,i) = r_new;  
end
```