Design and Analysis of Rocket Nozzle Contours for Launching Pico-Satellites

Signature Date

Title

Design and Analysis of Rocket Nozzle Contours for Launching Pico-Satellites

By
Brandon Lee Denton

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

Approved by:

Jeffery Kozak, Ph.D. Rochester Institute of Technology	____________________ (Thesis Advisor)	 (Date)
Steven Day, Ph.D. Rochester Institute of Technology	____________________	 (Date)
Amitabha Ghosh, Ph.D. Rochester Institute of Technology	____________________	 (Date)
Edward Hensel, Ph.D. Rochester Institute of Technology	____________________ (Department Head, Mechanical Engineering)	 (Date)

Department of Mechanical Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology
Rochester, NY 14623

November 2007

Acknowledgments

This thesis has been two years in the making. Along the way there have been several people who have helped in its successful completion.

Dr. Jeffrey Kozak has been wonderful mentor, advisor and friend. Thank you for your endless patience, guidance and knowledge. I appreciate the countless hours you have graciously afforded me in discussing, reviewing and critiquing my work.

I would like to thank Dr. Amitabha Ghosh and Dr. Steven Day for their time in reviewing and critiquing my work. I have enjoyed the time spent over the past two years discussing this work. Your sharing of knowledge has been instrumental in its success.

I would also like to thank Dr. Edward Hensel for seeing the potential in me and giving me the opportunity to continue my education. I appreciate the time you have taken out of your busy schedule to discuss and critique my work.

Thanks to the METEOR Project team at RIT. Your collaboration has allowed for a real-world application of the principles discussed and developed in this paper. Good luck. I know you'll reach your ultimate goal.

Mom and Dad, thanks for the endless love and support you have given me all these years. Your general interest in my work has kept it fresh and interesting. Thank you for instilling in me a never-ending curiosity, desire to learn and confidence to continue expanding my horizons. Love you.

Lastly, I would like to thank my girlfriend Mariko, without your love and support none of this would be possible. Your smile has given me the strength to get through the tough times when I thought I'd never find the solutions. I love you.

Table of Contents

List of Figures

List of Tables

Nomenclature

Symbols

r	- denotes radial coordinate	e_x	- unit axisymmetric directional vector
ϕ	- denotes angular coordinate	N	- denotes the numerator
x	- denotes axial coordinate	D	- denotes the denominator
v	- r direction component velocity	θ	- flow direction
u	- x direction component velocity	α	- Mach number
w	- ϕ direction component velocity	υ	- Prandtl-Meyer Expansion angle
M	- Mach number	γ	- ratio of specific heats
ρ	- density	β	- radius defining the arc of the expansion region
$\vec{V}$	- velocity	b	- side length of a right triangle
p	- pressure	ψ	- Stream Function
a	- speed of sound	m	- slope of a straight line
e_r	- unit radial directional vector	z	- y-intercept of a straight line
e_ϕ	- unit angular directional vector	σ	-angle the characteristic makes with the axisymmetric line for the Aerospike nozzle

Subscripts

char	- for characteristics	*guess*	- variable is a guess variable
C_-	- along the C_- characteristic (right-running characteristic)	*ExpansionPoint*	- variable is associated with the expansion point of the Aerospike nozzle
C_+	- along the C_+ characteristic (left-running characteristic)	*MachLine*	- variable is associated with the Mach line
$i+1$	- variable is unknown	*sonicline*	- variable is associated with the sonic line of the nozzle
i	- variable is known	*ei*	- with respect to the internal expansion section of the IE Aerospike Nozzle
exit	- variable is an exit condition	*ext*	- with respect to the external expansion section of the IE Aerospike Nozzle
o	- variable is at stagnation condition	max	- maximum value
1	- variable is at point 1	*throat*	- variable is at the throat of the nozzle
2	- variable is at point 2	*CurrentPoint*	- the current point in the calculation
θ	- variable is w/respect to the flow direction	*chord*	- with respect to the chord of the expansion arc
Calc	- variable is associated with the calculated variable based on a guess variable	*arc*	- with respect to the entrance region arc
temp	- temporary solution	*ContourPlot*	- largest value on FLUENT contour plot

upper	- upper limit	*FluentCalculated*	- value calculated by FLUENT
lower	- lower limit	*ComputerCalculated*	- calculated by the code developed in Section 3.0
desired	- user wanted value		

Prefixes

Δ	- incremental increase in the variable
$f(\ )$	- function of the internal variables

Abstract

Rochester Institute of Technology's Microsystems Engineering & Technology for the Exploration of Outer Regions (METEOR) project has been investigating and pursuing a low cost alternative launch system for launching pico-satellites to Low Earth Orbit (LEO). A major component of this system is a three-stage rocket for orbital insertion. Of all the parts that make up a rocket engine, the nozzle and its ability to convert thermal energy into kinetic energy is the most important in creating an efficient rocket. This paper develops a computer code which uses the Method of Characteristics and the Stream Function to define high-efficiency nozzle contours for isentropic, inviscid, irrotational supersonic flows of any working fluid for any user-defined exit Mach number. The contours were compared to theoretical isentropic area ratios for the selected fluid and desired exit Mach number. The accuracy of the nozzle to produce the desired exit Mach number was also checked. The flowfield of the nozzles created by the code were independently checked with the commercial Computational Fluid Dynamics (CFD) code FLUENT. FLUENT predictions were used to verify the isentropic flow assumption and that the working fluid reached the user-defined desired exit Mach number. Good agreement in area ratio and exit Mach number were achieved, verifying that the code is accurate.

1.0 Introduction

The first artificial satellite to orbit the Earth, Sputnik I, was launched on October 5th, 1957 by the Soviet Union. The United States' first artificial satellite, Explorer I, was launched on January 31st, 1958. The first Orbiting Satellite Carrying Amateur Radio, OSCAR I, was successfully launched four years later and orbited the Earth for 21 days. What's most unique about OSCAR I is that it was launched as a secondary payload on an Air Force rocket and only weighted 4.5 kg (The Radio Amateur's Satellite Handbook 2001; Anthology of Radio Amateur Satellites).

Even though there have been tremendous advances in technology over the past 50 years, space exploration has only become less expensive due to budget restrictions imposed on the manufacturing aerospace companies. Many view reaching Low Earth Orbit as the main barrier to the utilization and exploration of space (Dale, 2006). Current launch technologies and procedures can eclipse the cost of the satellite being launched by one or two orders of magnitude. It is necessary for the design and analysis of an alternative launch vehicle for launching Pico-Satellites to break the Low Earth Orbit barrier that many view as the final piece to the puzzle blocking the utilization of space.

Currently there are only a few alternatives to the standard satellite launch with the space shuttle and Pegasus, offered by Orbital Sciences Inc. Pegasus is designed for launching payloads of a minimum 285 kg in polar (97°), low earth orbit (400 km) (www.orbital.com). Sea-Launch can launch up to a 6000 kg in Geosynchronous Transfer Orbit (www.sea-launch.com).

With current technologies in Micro-Electro-Mechanical Systems (MEMS), satellites have been built with weights of 10 to 100 kg (Pico-Satellite). This size of satellite has been launched as a secondary payload due to the cost of launching objects into space (Dale, 2006). Typical costs associated with launches usually fall between $100,000 to a few million dollars with an expected wait time of anywhere from 3-5 years on average (Dale, 2006).

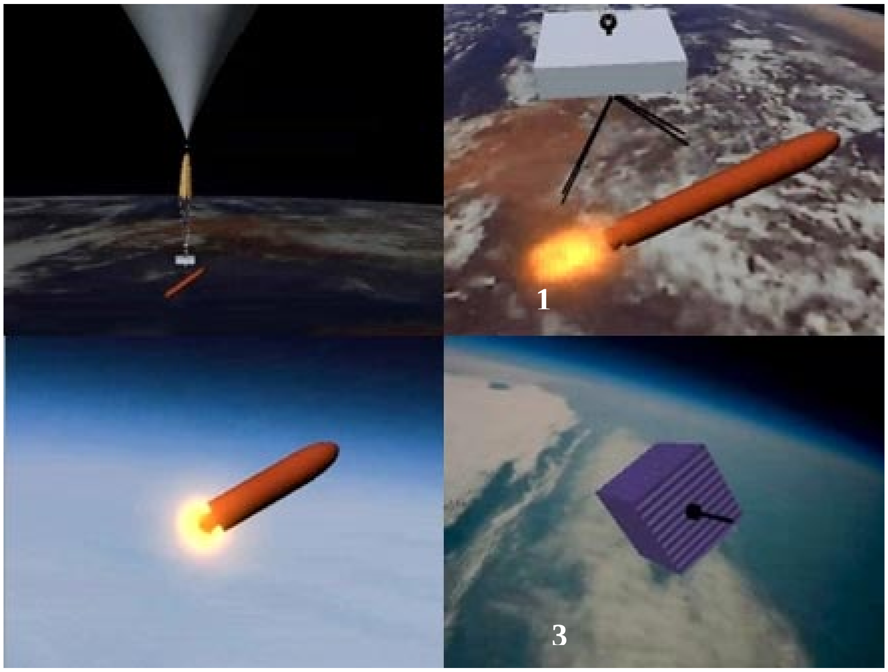
The advantages of an alternative launching method, once an alternative system has been developed, are the system would be mobile, cost effective and reduce the wait time for a successful launch. A secondary advantage would be the opportunity, in the case of a failed launch, to quickly plan another launch. The system would possibly be able to relaunch 3-4 months after the initial launch; clearly a huge advantage over the current methods of launching pico-satellites.

1.1 Alternate Design Concept

The proposed design that has been started at the Rochester Institute of technology's (RIT) Microsystems Engineering & Technology for the Exploration of Outer Regions (METEOR) project is launching a rocket at 30,000m above sea-level. The rocket would reach this elevation via a platform attached to helium balloons. Current balloon configuration can lift to at least 30,000m and float for an extended period of time with payloads as heavy as 1000 kg (www.ravenind.com). At this altitude, gravitational forces are smaller and atmospheric pressure and density are less than 1% that of sea-level. Therefore it is reasonable to assume that external forces on the balloon and rocket are essentially zero. This is a substantial advantage compared to group launched rockets.

Once the balloon is in flight, the platform will passively stabilize translation in the z-axis and tilt in the x-y plane, parallel to the Earth's surface. Assuming that the angle of the rocket with the z-axis is preset on the ground before launch, all remaining orientation in the x-y plane can be controlled before the rocket's launch, determining the orbit inclination of the satellite.

The rocket is attached to the platform and can be rotated about the z-axis relative to the platform. The current thought is the rocket will be a two- or three-stage vehicle, depending on the characteristics of the target orbit. In the current concept, all three stages are envisioned as a hybrid propellant system, which would enable throttling and repeated turn on and shutdown. Ultimately allowing a more accurate orbit insertion with the possibility to elevate the orbit or reach escape velocity. Micro thrusters using a pressurized inert gas would provide altitude control during the powered flight. Figure 1.1 shows an illustration of the launch concept.



Legend

1: Balloon Lift
2: Rocket Launch
3: Orbital Insertion
4: Satellite in Orbit

Figure 1.1: METEOR Project Launch Concept

1.2 Motivation

Although the burn characteristics of the fuel are an important part of the analysis of a rocket, the rocket's efficiency is primarily dependant upon the nozzle's ability to convert the thermal energy of the fluid to kinetic energy. The main nozzle wall contour plays a critical role in this conversion. It is also important to ensure shocks do not occur within the nozzle. Shocks in the nozzle will disrupt the supersonic flow and will create large losses during the conversion of thermal energy to kinetic energy. The wall contour of the nozzle is the defining factor in whether shocks will or will not form within the nozzle. The pressure ratio between the chamber and the exit plane of the nozzle dictate the maximum potential Mach Number reached by the working fluid.

There are many configurations of supersonic nozzles that will achieve the necessary conversion of thermal energy to kinetic energy to create a rocket's thrust. Some of the most common are conical, Bell and plug nozzles. An example of all three types can be seen in Figure 1.2 below.

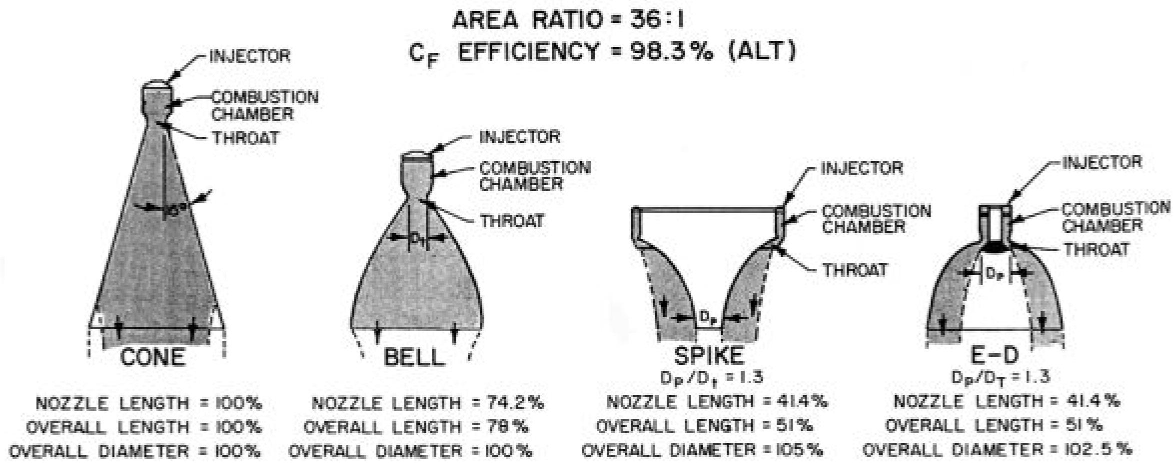


Figure 1.2: Rocket Nozzle Profiles

Conical nozzles have a constant expansion rate and look like a cone, hence their name. These nozzles tend to be the longest and heaviest type of nozzle. They do have the advantage however of being the easiest to manufacture. The Bell or annular nozzle, have a curved expansion contour which allows for a higher efficiency for the conversion of thermal energy to kinetic energy. These nozzles tend to be shorter and lighter than the conical nozzle but have the distinct disadvantage of being much more difficult to manufacture and therefore more costly. It is important to note that there are many different variation of the Bell nozzle each having a distinct advantage and disadvantages over another depending on their particular application. The last type of nozzle is the plug nozzles. These nozzles are characterized by having an expansion point about which the fluid is accelerated as well as a wall contour. Plug nozzles have the distinct advantage of altitude adjustment during flight allowing maximum thrust throughout the entire trajectory of the rocket where the conical and Bell nozzles only have a maximum thrust at a designed altitude which is at some optimized altitude during its flight. Plug nozzle’s major disadvantage to conical and Bell nozzles is they are typically less efficient in converting thermal energy into kinetic energy because the flow is bounded on one side by a constant pressure boundary. This in turn means that for a given set of conditions the Bell and conical nozzles produce more thrust than a plug nozzle. An aerospike nozzle is a particular type of plug nozzle in which the flow is only one wall contour and the other side is defined by a constant pressure boundary. This constant pressure boundary creates an imaginary contour which allows the nozzle to generate the maximum thrust possible by the nozzle as the imaginary contour adjusts to the changes in pressure as the rocket gains altitude. An advantage that the aerospike has over the Bell and conical nozzles is that it can be truncated to reduce the weight and length of the nozzle while maintaining a thrust similar an aerospike nozzle that was not truncated. Another advantage that is unique to the aerospike nozzle is that it can be designed to incorporate attitude controls without the need for fins or additional thrusters. This lends itself to additional weight advantages of the overall rocket design.

Since the balloon platform has a maximum carrying weight constraint, it is important to maximize the thrust the rocket can produce to achieve orbital insertion while minimizing the total weight of the rocket. The majority of the weight of the rocket is associated with the rocket’s nozzle since it is customary to neglect the weight of propellants and their storage containers in the rocket’s calculated weight. By minimizing the weight and energy losses associated with the nozzle, the efficiency of the overall launching system will increase as well as the maximum weight of the satellite that could be launched with this system. High-efficiency nozzles are therefore a crucial part in the successful realization of the METEOR project.

1.3 Thesis Overview

This thesis designs and analyzes three high-efficiency rocket nozzle contours to help optimize the rocket to launch a pico-satellite into Low-Earth Orbit. The nozzles discussed are annular and two types of aerospike nozzles. Testing of the annular nozzle design will be conducted in the Spring Quarter at RIT and its results are not included in this thesis. The nozzles tested will be designed using the axisymmetric Method of Characteristic in conjunction with the Stream Function as outlined in this paper. The Stream Function is used to define the wall contour of the nozzles.

Considering that the rocket will be launched from a high-altitude and the effects of density changes in the air will be negligible, the annular nozzle will probably be more appropriate for this application since the automatic altitude adjustment characteristics of the aerospike nozzle will not be necessary. However, in the interest of knowledge and the application of the data to future proposals for research grants, the aerospike design will be valuable and advantageous. Note that there is not much data publicly available for aerospike nozzles.

In Section 2.0, Previous Research, the methods and results of others' research will be discussed. This section will also allow for a clear differentiation between this thesis' work and the work previously available to the public.

Section 3.0, Theory, details the discretation of the characteristic equations, development of the boundary conditions and Stream Function as well as how the curvature of the characteristic equations is captured in the calculations. This process will be written as a generic computer code in MatLab so multiple nozzle contours can be calculated for different user inputs such as desired exit Mach number, working fluid's ratio of specific heats and predetermined throat radius. These codes can be found in Appendix B. The development of the characteristic and compatibility equations for axisymmetric flow is available in Appendix A.

Section 4.0, CFD Setup & Solutions, will outline the procedure and techniques used in running the previously mentioned nozzle designs through the well-known CFD program FLUENT. This was done to verify that the nozzles reach their desired exit Mach number and the conservation of mass is satisfied.

Section 5.0, Results and Discussion, examines the error analysis of the computer code outlined in Section 3.0 for multiple ratios of specific heats, exit Mach numbers and incremental changes in Prandtl-Meyer Expansion angle. A complete set of results and comparisons of the error in the code can be found in Appendix C. This section also discusses the results given by the FLUENT simulations. The exit Mach number given by FLUENT is used as a second method of calculating the accuracy of the computer code's calculated wall contour. The contours will be considered efficient if FLUENT solutions show no shocks waves developing in the flowfield. Complete simulation setups, results, mesh geometries, mesh qualities, Mach contours and entropy contours for air and the combustion products of Nitrous Oxide and HTPB can be found in Appendix D for the nozzles discussed in this paper. Exit Mach numbers of 3.0 and 4.0 were simulated for each of the nozzles.

Section 6.0, Conclusions, will reiterate the major findings of this paper and highlight its applicability in advancing the METEOR project.

Lastly, Section 7.0, Future Work and Recommendations, outlines future work that will be investigated in the METEOR project. It will also discuss suggested improvements needed in the computer code outlined in Section 3.0 to enhance the code's ability to produce nozzle wall contours for non-isentropic, viscid supersonic flow.

2.0 Previous Work

Presented in this section are the previous works done in supersonic nozzle design pertinent to the current investigation. Since the nozzles designed in this paper are for irrotational, inviscid, isentropic flows, only previous works dealing with these types of nozzles will be discussed. The first part of this section will deal with annular nozzles and the second part will deal with aerospike nozzles.

2.1 Annular Nozzles

The method typically used for defining the contour of an annular supersonic nozzle is the Method of Characteristics. In many cases the Method of Characteristics is solved using a Finite-Difference solution. Although there are many variations of this method, the distinct principles of this method remain the same.

The method of characteristics is fully defined by Shapiro, 1953-54, and Anderson, 1982. Both books describe the derivation of the characteristic and compatibility equations as well as explain how to approximate the contour of the nozzle which turns the flow parallel to the nozzle's axisymmetric line with the least amount of losses incurred. They differ by the technique used in defining the contour which turns the flow parallel to the nozzle's axisymmetric line. Anderson defines the nozzle's contour by calculating line segments between the C_+ Characteristics which are unresolved by a wall point. A C_+ Characteristic is a characteristic that runs to the left as you are looking downstream of the flow. The slope of the line segments defining the wall contour are defined by averaging the flow direction predicted by the two characteristics which the line segment operates between. The intersection of the line segment, i.e. wall contour, and the C_+ characteristic defines the nozzles contour. The line segments always emanate from the end of the previous line segment. For the first line segment, its origination point is the last point on the known expansion arc. An illustration of this method is given in Figure 2.1.1 below. This method becomes more accurate as the number of characteristics used in the calculation increases. The increased number of characteristics also results in a smoother contour. The accuracy of the solution is calculated by comparing the exit area ratio of the calculated nozzle to its ideal isentropic exit area ratio for the desired exit Mach number.

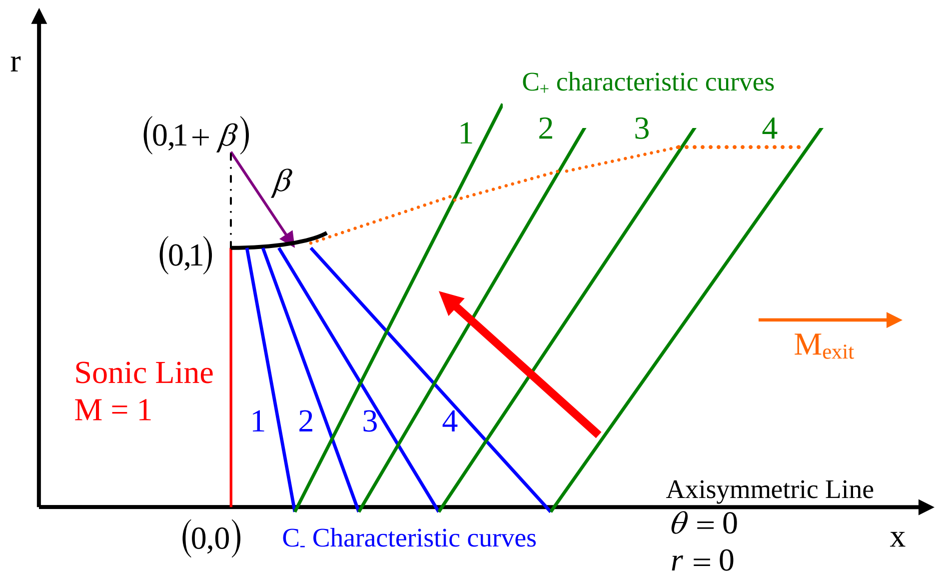


Figure 2.1.1: *Geometry of the Nozzle's Contour Defined by Anderson, 1982*

Shapiro takes a slightly different approach in defining the nozzle's contour. He employs a combination of "backward" C_- Characteristics and the Stream Function. A "backward" C_- Characteristic is a C_- Characteristic that is calculated from the last C_+ Characteristic to the first. The direction is indicated by the red arrow in Figure 2.1.1. The "backward" C_- Characteristics are calculated by assuming that the flow properties along the last C_+ Characteristic remain constant along that characteristic and are the same as the last point on the axisymmetric line. By choosing a point on this C_+ Characteristic as the starting point of the "backward" C_- Characteristic and assuming that a C_+ Characteristic is emanating from the last point on the known expansion arc, a complete calculation of the "backward" C_- Characteristic can be accomplished. Assuming the last point on the known expansion arc is the first point that satisfies the Stream Function of the flow, the calculation along the "backward" C_- Characteristic is continued until the point on the "backward" C_- Characteristic has a radial-component (r-component) greater than the r-component that will satisfy the Stream Function. Once this condition is met, a line segment between the last two points on the "backward" C_- Characteristic can approximate the change in position and flow properties along the "backward" C_- Characteristic. After this line segment is defined, the intersection of the Stream Function emanating from the last calculated streamline point and the "backward" C_- Characteristic defines the flow properties and position of the next point that satisfies the Stream Function and in turn defines the nozzles contour. The number of "backward" C_- Characteristics is increased until the starting point of the "backward" C_- Characteristic on the last C_+ Characteristic exhibits an r-component greater than the r-component that would satisfy the Stream Function. In this case, the intersection of the last C_+ Characteristic and the Stream Function emanating from the last streamline point defines the position of the last point on the nozzle's contour. Connecting all the points that satisfy the Stream Function with line segments yields the nozzle's contour. Unlike Anderson, Shapiro chose not to use non-dimensional characteristic and compatibility equations. Figure 2.1.2 illustrates the method Shapiro used to define the nozzle's contour. As with the method outlined by Anderson, Shapiro's method also becomes more accurate by increasing the number of characteristics and "backward" characteristics used. Once again the accuracy is checked by comparing the exit area ratio of the calculated contour with the idealized exit area ratio for an isentropic flow of the desired exit Mach number.

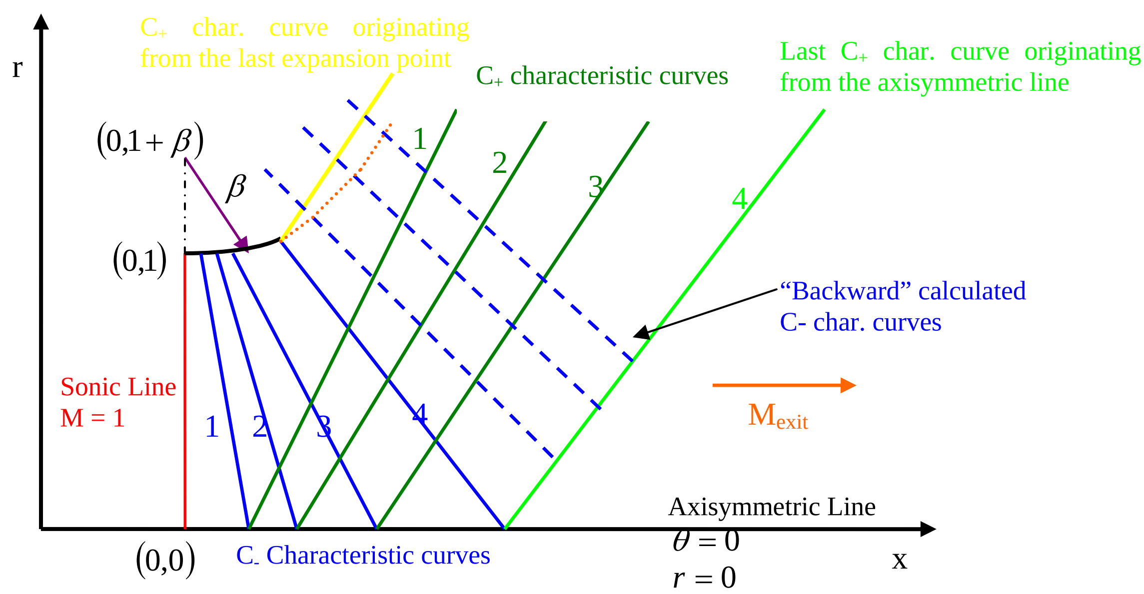


Figure 2.1.2: *Characteristic Geometry Used to Calculate the Wall Contour by Shapiro, 1953-54*

Another example of how the nozzle's contour can be calculated is described in Foelsch, 1949. He describes how the solution to the characteristic equations can be approximated by comparing the conditions of a nozzle to that of a cone. He first deals with equations for a transition curve which converts a conical source flow into a uniform parallel stream of uniform velocity. The equations are obtained by integrating along a Mach line in the region of the conversion of the flow from conical to uniform and parallel. By stepping through Mach numbers by a user-defined increment from 1 to a user-defined exit Mach number, the contour can be established. The last part of the calculation is to resolve the location of the nozzles throat as well as the spherical nature of the sonic line located there. He did this by shifting the location of the conical source flow by a geometric x-component and superimposing a spherical face into a flat face at the throat. The accuracy of the method is again checked by comparing the exit area ratio to the idealized exit area ratio for an isentropic supersonic flow of the desired exit Mach number.

Each of the methods described above, have varying degrees of accuracy depending on user-defined variables. Keep in mind that these are only a sample of the many variations that are available for defining the contour of a supersonic nozzle using the Method of Characteristics.

This thesis will utilize a modified version of the method outlined by Shapiro. The technique is the same but the equations evaluated are the ones given by Anderson. This method was chosen because of its ability to be modified into a universal code which can calculate multiple types of nozzle contours with only a few user inputs. This paper will focus on this code's ability to calculate annular and aerospike nozzle contours.

2.2 Aerospike Nozzles

Aerospike nozzles are investigated in this paper for two reasons. The first is to demonstrate that the code developed in Section 3.0 is robust and the techniques used are applicable for designing different types of supersonic nozzles. The second reason for investigating aerospike nozzles is to expand upon the limited public information available on them.

There are two types of aerospike nozzles, the minimum-length, or traditional aerospike nozzle and the Internal-External (IE) aerospike nozzle. The traditional aerospike nozzle allows the expansion of the flow to happen completely externally where the IE aerospike nozzle allows a portion of the flow to expansion within a confined section and the remaining expansion occurs externally. A schematic of these two different types can be found in Figures 2.2.1 and 2.2.2, respectively. Aerospike nozzles have predominately two methods for defining their contours. One is the Method of Characteristics and the other employs the use of the isentropic area ratio equation.

Traditional or Minimum-Length Aerospike Nozzle

Greer, 1960, describes a method which uses geometry and the isentropic area ratio equation to define the contour of the aerospike nozzles. First, before we discuss the method, it is important to note that the angle the direction of the flow at the throat makes with the nozzle's axisymmetric line at the beginning of the external expansion is equal to the Prandtl-Meyer expansion angle for the user-defined desired exit Mach number. Using this angle as the sonic flow direction, the Prandtl-Meyer expansion fan centered at the tip of the cowl located at the end of the sonic line furthest from the nozzle's axisymmetric line can be stepped through by a user-defined Prandtl-Meyer expansion angle increment. For each Prandtl-Meyer expansion angle stepped through, its associated Mach number can be calculated. Using the Mach number and geometry, the length of the line from the tip of the cowl is known from the isentropic area ratio equation. From geometric manipulation and the flow properties of an expansion fan, the slope of the line emanating from the tip of the cowl can be calculated. Since the tip of the cowl can be geometrically set by the designer, the points located on the nozzle's contour can be calculated using trigonometry. Greer non-dimensionalized the calculation by dividing the length of the lines emanating from the tip by the length associated with the desired exit Mach number. The calculations are stepped through until the desired exit Mach number is obtained. It is also important to note that the flow properties along the lines emanating from the tip of the cowl are assumed to be constant. This is important because the curved nature of the characteristics is not taken into account for the calculation of axisymmetric nozzles introducing errors. The points on the contour are then connected by line segments to make the aerospike's contour. This method is accurate when comparing the exit to throat area ratio since the isentropic area ratio is used in defining the contour. The contour becomes smoother as the number of points defining the contour increase, aka the Prandtl-Meyer expansion angle increment decreases.

Another approximate method for defining a traditional aerospike nozzle is outlined by Angelino, 1964. He used a method similar to Greer only instead of using geometry to define the angle the characteristic made with the axisymmetric line, he used the characteristic equations. For an expansion fan, the angle the Mach line (characteristic) makes with the geometric x-axis is known through the equation $\phi = \alpha(M) + \theta$. Like the method described by Greer, the direction the flow at the throat makes with the x-axis is equal to the Prandtl-Meyer expansion angle, and the calculation sweeps through the expansion fan by a user-defined Prandtl-Meyer expansion angle increment. For each step, the Mach number associated with each Prandtl-Meyer expansion angle is calculated. After the Mach number is known, the length of the characteristic line can be calculated using the isentropic area ratio equation and geometry for the given Mach number. Since the flow direction at the throat is known, the flow direction at each calculation step is known because for every incremental increase in the Prandtl-Meyer expansion angle in an expansion fan, the angle the direction of the flow makes with the x-axis decreases the same incremental amount. Since the location of the tip of the cowl is a user-defined quantity, the location of each point on the aerospike's contour can be calculated using trigonometry for each characteristic. This is continued until the user-defined exit Mach number is reached. To non-dimensionalize the equations, Angelino also divided the equation for the length of the characteristic by the isentropic area ratio associated with the desired exit Mach number. As with the method described by Greer, Angelino's method is also accurate with respect to the isentropic area ratio for the desired exit Mach number since this relation is used in the derivation of the lengths of the characteristics.

It is important to note that although the exit area of the aerospike nozzles described by Greer and Angelino are the same and equal to the idealized isentropic area ratio, the contours calculated by the methods will be slightly different from one another.

Another method used for calculating the contour of a Traditional Aerospike contour is described by Lee and Thompson, 1964. Their method uses the Method of Characteristics in conjunction with the Stream Function to define the contour. This method is similar to the technique employed by Shapiro. As with the methods outlined by Greer and Angelino, the flow direction at the throat is set at an angle equal to the Prandtl-Meyer expansion angle associated with the user-defined exit Mach number. The Prandtl-Meyer expansion fan is centered at the tip of the cowl and its location is user-defined. Unlike the other two methods described above, this method calculates the end point of the aerospike's contour first and sweeps through the Prandtl-Meyer expansion fan backwards, starting with a flow direction equal to zero and the Prandtl-Meyer expansion angle equal to the Prandtl-Meyer expansion angle associated with the desired exit Mach number. The flow direction is increase while the Prandtl-Meyer expansion angle is decreased by a user-defined Prandtl-Meyer expansion angle increment. The last contour point is calculated using the isentropic area ratio for the desired exit Mach number in conjunction with the angle the characteristic makes with the x-axis given by $\phi = \alpha(M) + \theta$ where $\theta = 0$, the location of the tip of the cowl and trigonometry. From this point, the Prandtl-Meyer expansion fan is swept backwards as described above, each time calculating the angle the characteristic makes with the x-axis. Once the slope is known for each characteristic, the Stream Function approximated by a line is employed originating from the end point of the aerospike's contour. Their intersection yields the location of the next point on the contour. These steps are continued until the slope of the characteristic is perpendicular to the flow direction, i.e. the throat. Connecting these calculated points define the aerospike's contour. The accuracy of this technique can once again be evaluated by comparing the isentropic area ratio with the area ratio calculated by the method. As the incremental Prandtl-Meyer expansion angle decreases, the accuracy and smoothness of the contour increases.

All methods used for defining a traditional aerospike nozzle have a common geometry. This geometry is given below in Figure 2.2.1.

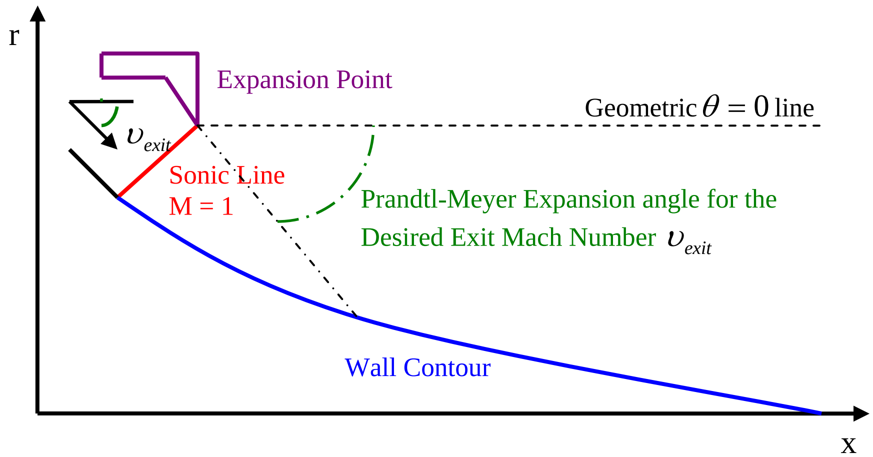


Figure 2.2.1: *Generic Geometry of a Traditional Aerospike Nozzle*

Internal-External Aerospike Nozzle

In essence, the Internal-External Aerospike nozzle is a combination of two Prandtl-Meyer expansion fans; one consisting of C_- Characteristics and the other of C_+ Characteristics. Lee and Thompson, 1964, described how to determine the contour of an IE Aerospike nozzle. The first step in the method is to determine the Prandtl-Meyer expansion angle associated by the user-defined exit Mach number of the internal section of the nozzle. Next, the Prandtl-Meyer expansion angle associated with the user-defined desired exit Mach number is determined. Since the desired flow direction at the exit of the nozzle is zero, from expansion fan theory, the flow direction entering the second expansion fan must be equal to the difference between the Prandtl-Meyer expansion angle of the desired exit Mach number and the Prandtl-Meyer expansion angle associated with the exit Mach number of the internally expanded section. From geometry, the direction of the flow at the throat can be calculated by the equation $\theta_{throat} = \upsilon_{NozzleExitMachNumber} - (2 \cdot \upsilon_{IntenalExitMachNumber})$. Since the throat is perpendicular to the flow's direction, the first point on the cowl's contour can be calculated using trigonometry with the location of the other end of the throat being user-defined. Lee and Thompson suggest defining the spike's contour in the internal section as an arc with a user-defined radius. This is similar to the expansion are commonly used in annular nozzles. The internal expansion wave is assumed to have a center located at the center of the arc defining the spike's contour in the internal section. The method calls for the first expansion fan, internal section, to be stepped through by a user-defined Prandtl-Meyer expansion angle increment. The slope of the characteristics can be calculated using $\phi = \alpha(M) - \theta$ for C_- Characteristic. From trigonometry, the points located at the intersection of the spike's arc and the characteristic can be found. Once this is completed, the Stream Function can be used emanating from the user-defined location of the end of the sonic line located on the cowl. Its intersection with the characteristic emanating from the center of the arc defines the next point on the aerospike cowl's contour. These steps are completed until the user-defined internal exit Mach number is reached resulting in the defining of the aerospike's cowl's contour.

The external section of the spike's contour is calculated the same way Lee and Thompson calculated their traditional Aerospike's contour. The only difference is that the location of the centered external expansion wave is located at the last point on the aerospike cowl's contour described above.

Lee and Thompson's method is always accurate when comparing the isentropic area ratio with the calculated area ratio since the isentropic area ratio is used in defining the nozzle's external contour.

Figure 2.2.2 illustrates a generic geometry of an Internal-External Aerospike nozzle.

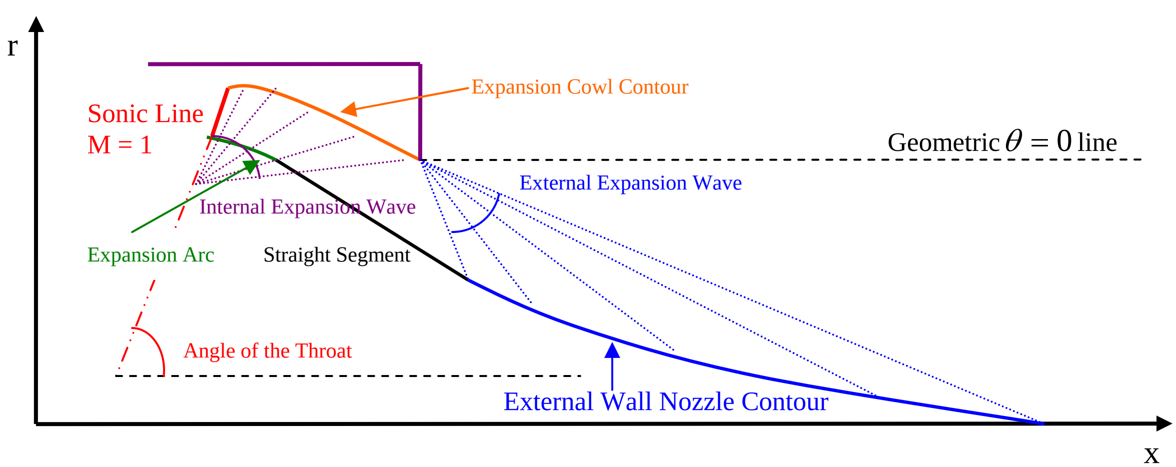


Figure 2.2.2: *Generic Geometry of an Internal-External Aerospike Nozzle*

2.3 Previous Work vs. Current Work Discussed Within this Paper

This section will compare the supersonic nozzle design techniques discussed above with the techniques used within this paper.

Annular Nozzles

The technique used in this paper to calculate the nozzle's contour is a combination of the techniques outlined by Anderson and Shapiro. The technique outlined in Section 3.0 uses the axisymmetric characteristic and compatibility equations derived by Anderson and the method of using "backward" characteristics and Stream Function described by Shapiro. The technique used in this paper is non-dimensionalized as long as the radius of the throat is set to unity. As with all the techniques described above, as the number of characteristics increase, aka the user-defined Prandtl-Meyer expansion angle increment decreases, the contours calculated becomes smoother and more accurate with respect to the isentropic area ratio for the desired exit Mach number. The technique is outlined in detail in Section 3.0 and an extensive accuracy check between the isentropic area ratio and the area ratio calculated by the technique used in this paper is tabulated in Appendix C. The derivation of the characteristic and compatibility equations is available in Appendix A.

Aerospike Nozzles

The technique used in this paper to calculate the contours of the aerospike nozzles are similar to the technique outlined by Lee and Thompson, 1969. For the traditional aerospike nozzle, the technique used in this paper defines the location of the ends of the throat and sets the flow direction at the throat equal to the Prandtl-Meyer expansion angle associated with the desired exit Mach number. Unlike the technique outlined by Lee and Thompson, the calculations used in this paper step forward through the expansion fan by a user-defined Prandtl-Meyer expansion angle increment. The intersection of the characteristics emanating from the expansion point on one end of the throat and the Stream Function originating from the last point calculated on the nozzle's contour define the nozzle's contour. For the first characteristic, the Stream Function originates from the opposite end of the throat than the expansion point. The Prandtl-Meyer expansion fan is stepped through until the Mach number along the characteristic being analyzed is greater than or equal to the desired user-defined exit Mach number, in which case, the intersection of the Stream Function and characteristic signify the location of the last point on the nozzle’s contour.

The technique used in this paper for defining the cowl and nozzle’s contour for the Internal-External aerospike nozzle is almost identical to the method described by Lee and Thompson. In this paper, the internal section of the IE aerospike nozzle, the spike and cowl contour, is calculated in a similar way as the internal section of the nozzle described by Lee and Thompson. Instead of approximating the characteristics as lines, this paper uses the characteristic equations emanating from the expansion arc. The technique used in this paper also differs from Lee and Thompson in the method used to calculate the external portion of the contour. Instead of calculating the end of the spike’s contour first and stepping backward through the second expansion fan, the technique in this paper steps forward through the second expansion fan continuing to use the intersection of the Stream Function and characteristics to define the nozzle’s contour using the same technique described above for defining the traditional aerospike contour.

As with the annular nozzles, the accuracy and smoothness of the contours obtained by the techniques used in this paper for the aerospike nozzles increase with the decrease in Prandtl-Meyer expansion angle increment size. The techniques used are described in detail in Section 3.0 with extensive accuracy checks between the isentropic area ratios and the area ratios obtained by the technique used in this paper are tabulated in Appendix C.

3.0 Theory

This section outlines the methods used to calculate higher efficiency annular and aerospike rocket nozzles using the Method of Characteristics and Stream Function. Taking the development of the characteristic equations assuming axisymmetric, irrotational, inviscid flow from Anderson (1982) and combining them with the outlined method for employing the Stream Function to define the nozzle's wall contour given in Shapiro (1954), allows for the calculation of wall contours that are smoother than the traditional Method of Characteristics. The assumption of irrotational, inviscid flow is an appropriate simplification since a favorable pressure gradient is typical within nozzles. Complete derivations of the characteristic and compatibility equations are available in Appendix A.

The axisymmetric solution differs from the classical 2-Dimensional solution because it takes into account for the squared radial dimension in the area calculation. The classical 2-Dimensional solution increases the cross-sectional area of the nozzle on a 2-Dimensional plane. The 2-Dimensional method translates into an area ratio that it larger than the one intended for the desired exit Mach number when the classic 2-Dimensional solution is rotated about its axis. The axisymmetric solution resolves this issue.

3.1 Discretation of Equations, Boundary Condition and Stream Function Analysis

This section outlines the original work done in this thesis.

Discretizing the Characteristic and Compatibility Equations

To implement the characteristic and compatibility equations into a computer code for designing supersonic nozzle contours, the equations for axisymmetric, irrotational, inviscid flow developed in Appendix A must be discretized with boundary conditions defined and applied.

The first step in designing a computer code is to discretize the characteristic and compatibility equations. They are rewritten below.

$$\left(\frac{dr}{dx}\right)_{char} = \tan(\theta \mp \alpha) \qquad \textit{Eq. 3.1}$$

$$d(\theta + \upsilon) = \frac{1}{\sqrt{M^2 - 1} - \cot\theta}\frac{dr}{r} \qquad \textit{Eq. 3.2a}$$

(along a C_- characteristic)

$$d(\theta - \upsilon) = -\frac{1}{\sqrt{M^2 - 1} + \cot\theta}\frac{dr}{r} \qquad \textit{Eq. 3.2b}$$

(along a C_+ characteristic)

Equation *3.1,* can be split to illustrate the two separate C_- and C_+ characteristic equations. They are written below.

$$\left(\frac{dr}{dx}\right)_{C_-} = \tan(\theta - \alpha) \qquad \textit{Eq. 3.3a}$$

$$\left(\frac{dr}{dx}\right)_{C_+} = \tan(\theta + \alpha) \qquad \textit{Eq. 3.3b}$$

Using the Forward Difference Technique and rearranging equations *3.3a* and *b* yields

$$r_{i+1} - \tan(\theta_i - \alpha_i)\cdot x_{i+1} = r_i - \tan(\theta_i - \alpha_i)\cdot x_i \qquad \textit{Eq. 3.4a}$$

(along a C_- characteristic)

$$r_{i+1} - \tan(\theta_i + \alpha_i)\cdot x_{i+1} = r_i - \tan(\theta_i + \alpha_i)\cdot x_i \qquad \textit{Eq. 3.4b}$$

(along a C_+ characteristic)

Note that all variables with subscript *i* are known quantities and variables with subscript *i+1* are unknown quantities. Equations *3.4a* and *3.4b* are the discretized characteristic equations that will define the location in the x-r space where the C_- and C_+ characteristics curves intersect. This collection of points is called the Characteristic Net.

Equation *3.2a* and *3.2b*, the compatibility equations, can also be discretized. Using the Forward Difference Technique and rearranging gives

$$\theta_{i+1} + \upsilon_{i+1} = (\theta_i + \upsilon_i) + \frac{1}{\sqrt{M_i^2 - 1} - \cot\theta_i} \frac{r_{i+1} - r_i}{r_i}$$

Eq. 3.5a
(along a C_- characteristic)

$$\theta_{i+1} - \upsilon_{i+1} = (\theta_i - \upsilon_i) - \frac{1}{\sqrt{M_i^2 - 1} + \cot\theta_i} \frac{r_{i+1} - r_i}{r_i}$$

Eq. 3.5b
(along a C_+ characteristic)

Note that r_{i+1} is on the right side of the compatibility equations. In the calculation, this is a known quantity from the solution of the Characteristic Net, equations *3.4a* and *b*. Therefore, the compatibility equations can be solved simultaneously to find the direction of the flow and the Prandtl-Meyer Expansion angle at the point where the characteristics intersect. The speed, Mach number, and Mach angle of the flow at that point of intersection can be extrapolated from the Prandtl-Meyer expansion angle.

In order to develop the computer code, boundary conditions must be established. Figure 3.1.1 defines the system in which the boundary conditions will be defined.

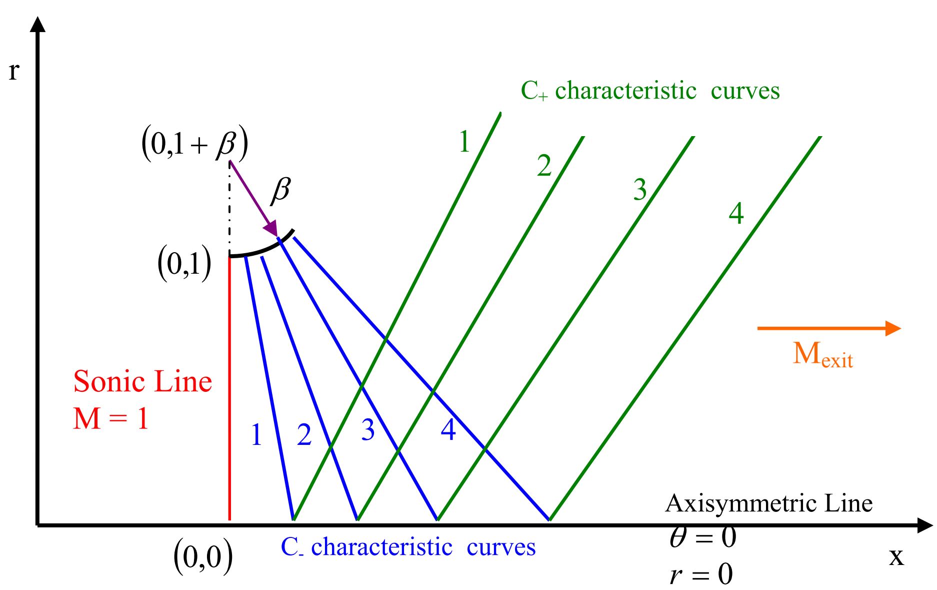


Figure 3.1.1: *Geometry of the Characteristics' Interaction in the Annular Nozzle*

Expansion Arc Contour Boundary Condition:

The first boundary condition is the expansion arc of the nozzle. In Figure 3.1.1, this is the black curved line that starts at the throat and ends at the last C_- characteristic which achieves the desired exit Mach number on the axisymmetric line. To achieve a smooth transition in flow direction, a circular arc with a center located at $\left(0, r_{throat} + \beta\right)$ is assumed to define the wall contour of the nozzle's expansion arc.

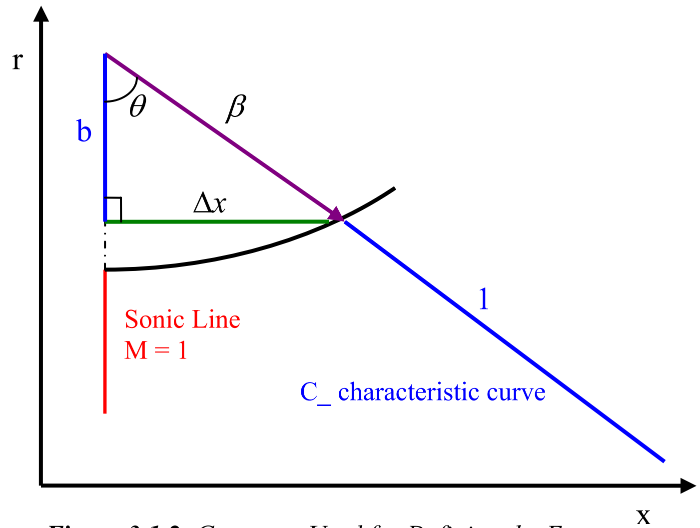


Figure 3.1.2: *Geometry Used for Defining the Entrance Region of the Annular Nozzle*

Since the flow at the throat is assumed to be sonic in the axial direction and the direction of the flow along a curve is tangent to the curve, the location of the center of the arc can be defined from geometry with the user-defined radius at the throat, r_{throat}, and the radius of the arc, β. The location of points on the expansion arc can be calculated by using trigonometry and the user-specified incremental change in the Prandtl-Meyer Expansion angle, $\Delta\upsilon$, and radius of the arc, β. Figure 3.1.2 illustrates the geometry utilized to solve for the location of the points that define the expansion arc.

From geometry, the direction of the flow is always tangent to the arc, making the arc a streamline. The direction of the flow is also equal to the total angle the calculation has been swept through, making the flow direction along the wall of the expansion region equal to the Prandtl-Meyer expansion angle of that point.

$$\theta_i = \upsilon_i \qquad \textit{Eq. 3.6}$$

(Along the arc of the expansion region)

Using trigonometry of a right triangle and user-defined $\Delta\upsilon$ and β, the x-component and r-component of a point's location along the arc that defines the expansion region can be calculated by equations *3.7* and *3.8*, respectively.

$$x_i = \beta \cdot \sin(\theta_i) \qquad \textit{Eq. 3.7}$$

$$r_i = 1 + (\beta - \beta \cdot \cos(\theta_i)) \qquad \textit{Eq. 3.8}$$

Since the Prandtl-Meyer expansion angle is known for any given point on the expansion arc, a root finding routine can be employed to solve for the Mach number associated with the Prandtl-Meyer expansion angle and user-defined ratio of specific heats for the working fluid. This root finding routine will use a guess Mach number to calculate its associated Prandtl-Meyer expansion angle using equation *3.9* and compare this calculated Prandtl-Meyer expansion angle with the known Prandtl-Meyer expansion angle at the given point.

$$\upsilon = \sqrt{\frac{\gamma+1}{\gamma-1}} \tan^{-1}\left(\sqrt{\frac{\gamma-1}{\gamma+1}\left(M_i^{\,2}-1\right)}\right) - \tan^{-1}\left(\sqrt{M_i^{\,2}-1}\right) \qquad \textit{Eq. 3.9}$$

If the calculated Prandtl-Meyer expansion angle is not within the user-defined accepted error parameter, the program will calculate a new guess Mach number and recalculate its associated Prandtl-Meyer expansion angle. This cycle will continue until the difference in calculated and known Prandtl-Meyer expansion angle is less than the accepted error parameter. The guess Mach number that satisfies the accepted error parameter will then be saved as the Mach number of the given point on the expansion arc. Flowchart B.1 in Appendix B details this calculation process. This procedure is also employed to find the Mach number of all the points in the flowfield. Details on the specific methods used in calculating the Mach number will be discussed in Section 3.4, subroutine PMtoMA.

From this Mach number, the Mach angle of the C_- characteristic emanating from the point on the expansion arc can be calculated by using equation *3.10*.

$$\alpha_i = \sin^{-1}\left(\frac{1}{M_i}\right) \qquad \textit{Eq. 3.10}$$

Axisymmetric Boundary Condition:

The second boundary condition associated with the calculation of the nozzle's contour is the axisymmetric line. The conditions at the axisymmetric line for the flow direction and r-coordinate are known, $\theta = 0$ and $r = 0$, respectively. These conditions do not pose a problem for the calculation of points in the Characteristic Net on the axisymmetric line.

The conditions of the axisymmetric line become a problem when equation *3.5b* is used to solve for the flow direction and Prandtl-Meyer expansion angle of a point where the C_+ characteristic curve originates from a point on the axisymmetric line. In this case, the term $\frac{1}{r}\left(\frac{1}{\sqrt{M^2-1}+\cot\theta}\right)$ in equation *3.5b* becomes indeterminate. To find the limiting case, a double limit of this term must be solved. Since the equation is dimensionless, the solution is independent of fluid type.

$$\lim_{r\to 0,\theta\to 0}\frac{1}{r}\frac{1}{\sqrt{M^2-1}+\cot\theta} \qquad \textit{Eq. 3.11}$$

Using the program *Derive*® by Texas Instruments, equation *3.11* simplifies to

$$\lim_{r\to 0,\theta\to 0}\frac{1}{r}\frac{1}{\sqrt{M^2-1}+\cot\theta} = \frac{d\theta}{dr} \qquad \textit{Eq. 3.12}$$

Discretizing equation *3.12* gives

$$\frac{d\theta}{dr} = \frac{\theta_{i+1}-\theta_i}{r_{i+1}-r_i} \qquad \textit{Eq. 3.13}$$

Substituting equation *3.13* into equation *3.5b*, simplifying and rearranging yields eq. 3.14, the limiting case equation for a C_+ characteristic curve emanating from a point on the axisymmetric line.

$$2\theta_{i+1} - \upsilon_{i+1} = \theta_i - \upsilon_i \qquad \textit{Eq. 3.14}$$

Equations *3.14* and *3.5a* form a system of equations that can be solved simultaneously to calculate the flow direction and Prandtl-Meyer expansion angle of a point where its C_+ characteristic curve originates from a point on the axisymmetric line.

Nozzle Contour Boundary Condition:

The last set of equations that need to be discussed is used to calculate the nozzle's wall contour after the last point on the expansion arc. Figure 3.1.3 illustrates the system under which the following calculations are based.

To begin the process of finding the unknown nozzle contour that turns the flow back to the

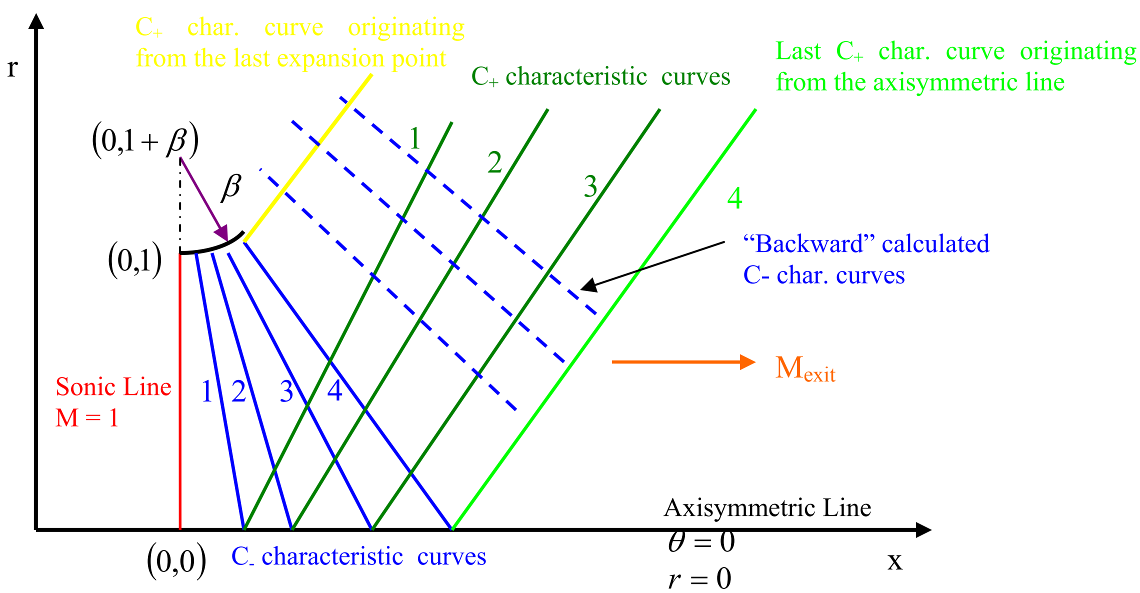


***Figure 3.1.3:** Characteristic Geometry Used to Calculate the Wall Contour of the Annular Nozzle*

axial direction, the last C_+ characteristic curve originating from the axisymmetric line is assumed to be straight. In this case, the Mach number, flow direction, θ, Prandtl-Meyer expansion angle, υ, and the Mach angle, α, are all constant and equal to the conditions at the point where the C_- characteristic originating from the last expansion region intersects the axisymmetric line. Since all the flow conditions are known along this C_+ characteristic, the position of the starting point for the "backward" C_- characteristic can be found. The first step in calculating the starting point of the "backward" C_- characteristic is to calculate the point's x-component. This is done by using

$$x_{i+1} = x_i + \left(StepSize \cdot \Delta x\right) \qquad \textit{Eq. 3.15}$$

The user-defined factor StepSize in equation *3.15* is used for the calculation of the first "backward" characteristic starting point x-component to compensate for the curvature of the "backward" calculated C_- characteristic and avoid a solution of the coupled equations *3.4a* and *3.4b* to have an r-component less than the r-component of the last expansion point. In this case, the solution is useless in calculating the nozzle's contour. In all other calculations of the x-component of the starting point of the "backward" C_- Characteristic, the StepSize multiplier is divided by two. Details of the calculation of StepSize are discussed in section 3.4, subroutine AxisymmetricSAD.

Once the x-component of the starting point of the "backward" C_- characteristic is known, equation *3.4b* can be solved for the r-component of the starting point. Upon rearrangement, equation *3.4b* becomes

$$r_{i+1} = r_i + \tan\left(\theta_i + \alpha_i\right) \cdot \left(x_{i+1} - x_i\right) \qquad \textit{Eq. 3.16}$$

Now that the conditions and location of the "backward" C_- characteristic's starting point are known, equations *3.4a*, *3.4b*, *3.5a* and *3.5b* can be employed to find the conditions and location of points where C_+ and the "backward" C_- characteristics intersect. The calculations are repeated until the location of the intersection's r-coordinate is larger than the r-coordinate of the last known wall contour point. To satisfy this condition, a C_+ characteristic is assumed to emanate from the last expansion point as illustrated in Figure 3.1.3. If the r-coordinate on the first "backward" C_- characteristic is less than the r-coordinate of the last point on the expansion arc, the StepSize is increased and the entire calculation of the first "backward" C-characteristic is repeated until the r-coordinate condition is satisfied.

Once the location of the intersection's r-coordinate is larger than the r-coordinate of the last known wall contour point, the wall contour point that satisfies the streamline condition can be found. Since this calculation is based upon the condition of irrotational flow, a streamline exists in the flow and, therefore, is valid. The first step in solving for the location of the point that will satisfy the streamline is to assume that the change in flow properties can be approximated by a straight line between the point where the r-coordinate is greater that the r-coordinate of the last known wall contour point and the previous point along the "backward" calculated C_- characteristic. Since the objective of the nozzle's contour after the expansion region is to turn the flow back axially, it is conceivable that the r-coordinate of the point that will satisfy the stream function will have a r-coordinate that will be less than the point located at the intersection of the "backward" C_- characteristic and C_+ characteristic whose r-coordinate is larger than the previously known wall contour (streamline) point. The details of how to define the line that will approximate the properties between the two points on the "backward" C_- characteristic will be discussed after the analysis of the streamline. It will become evident that all the flow properties will not be necessary in the streamline calculations.

Stream Function Equation and Discretation:

Since in steady, axisymmetric flow there are only two space coordinates, the statement of the continuity equation is the necessary and sufficient condition for the existence of a Stream Function.

The most generic form of the Stream Function is

$$\psi = f(x,r) \qquad \textit{Eq. 3.17}$$

and $\psi =$ a constant. To satisfy the streamline condition, there is no change in the constant. Therefore, $d\psi = 0$.

Differentiating equation *3.17* and substituting $d\psi = 0$, gives

$$\frac{\partial \psi}{\partial x} dx + \frac{\partial \psi}{\partial r} dr = 0 \qquad \textit{Eq. 3.18}$$

Equation *3.18* is valid for small changes in dx and dr. To satisfy the continuity equation at a point, ψ is defined by

$$v = -\frac{\rho_o}{\rho} \frac{1}{r} \frac{\partial \psi}{\partial x} \qquad \textit{Eq. 3.19a}$$

$$u = \frac{\rho_o}{\rho} \frac{1}{r} \frac{\partial \psi}{\partial r} \qquad \textit{Eq. 3.19b}$$

Solving equations *3.19a* and *3.19b* for $\partial \psi / \partial x$ and $\partial \psi / \partial r$, respectively, and substituting into equation *3.18* yields

$$-rv \frac{\rho}{\rho_o} dx + ru \frac{\rho}{\rho_0} dr = 0 \qquad \textit{Eq. 3.20}$$

Since u and v are not explicitly known at the points in a supersonic flow field using the non-dimensional Method of Characteristics, they must be converted to their associated Mach number at the point through the following transformation. From geometry

$$u = \vec{V} \cos\theta \qquad \textit{Eq. 3.21a}$$

$$v = \vec{V} \sin\theta \qquad \textit{Eq. 3.21b}$$

The velocity and Mach number at the point is known through the following relation.

$$\vec{V} = a \cdot M \qquad \textit{Eq. 3.22}$$

Substituting equation *3.22* into equation *3.21a* and *3.21b* gives

$$u = a \cdot M \cos\theta \qquad \textit{Eq. 3.23a}$$

$$v = a \cdot M \sin\theta \qquad \textit{Eq. 3.23b}$$

Substituting these results into equation *3.20* yields

$$-r \cdot a \cdot M \sin\theta \frac{\rho}{\rho_o} dx + r \cdot a \cdot M \cos\theta \frac{\rho}{\rho_o} dr = 0 \qquad \textit{Eq. 3.24}$$

Since the equation is satisfied at a point, r, a, M and ρ/ρ_o can be divided out, simplifying equation *3.24* to

$$-\sin\theta \cdot dx + \cos\theta \cdot dr = 0 \qquad \textit{Eq. 3.25}$$

Discretizing and rearranging equation *3.25* for the unknown quantities *i+1* on the left side of the equations yield

$$r_{i+1} - \tan\theta_i \cdot x_{i+1} = r_i - \tan\theta_i \cdot x_i \qquad \textit{Eq. 3.26}$$

Equation *3.26* illustrates that a steady, irrotational, supersonic flow's streamline is dependant only on the direction of the flow and location of its previous point. In conjunction with an approximation by a straight line between the point where the r-coordinate is greater that the r-coordinate of the last known wall contour point and the previous point along the "backward" calculated C_- characteristic, equation *3.26* can be employed to find the location of the next point on the streamline by the solution of two intersecting lines.

Calculating the Point Satisfying the Stream Function:

The last step in satisfying the Stream Function equation is to develop straight line approximations for the flow property changes between the last two points along the "backward" C_- characteristic. The general equation for a straight line

$$r = mx + z \qquad \textit{Eq. 3.27}$$

where m is the slope, z is the r-intercept, r is the r-coordinate and x is the x-coordinate of a point on the line. Since the positions, (x,r), of two points are known along the approximation line, the slope and r-intercept of the line can be solved using the system of equations.

$$r_1 = mx_1 + z \qquad \textit{Eq. 3.28a}$$

$$r_2 = mx_2 + z \qquad \textit{Eq. 3.28b}$$

Since the Stream Function is also dependent on the flow direction at the previous point, the change in θ along the straight line approximation must also be calculated. Modifying equation *3.27* for θ on the $\theta - r$ plane yields

$$\theta = m_\theta x + z_\theta \qquad \textit{Eq. 3.29}$$

where m_θ is the slope, z_θ is the θ-intercept, θ is the θ-coordinate and x is the x-coordinate of a point on the line. Since the positions, (x,θ), of two points are known along the approximation line, the slope and θ-intercept of the line can be solved using the system of equations

$$\theta_1 = m_\theta x_1 + z_\theta \qquad \textit{Eq. 3.30a}$$

$$\theta_2 = m_\theta x_2 + z_\theta \qquad \textit{Eq. 3.30b}$$

Now that the equations for the approximation lines are known, the intersection of the Stream Function equation with the (x, r) space approximation line of the "backward" C_- characteristic yields the solution of the position for the next point that satisfies the Stream Function. The x-component of the solution point that satisfies the Stream Function can be used in the (x,θ) space line approximation equation to find the flow direction at the solution streamline point. This is illustrated in equation *3.31*.

$$\theta_{i+1} = m_\theta x_{i+1} + z_\theta \qquad \textit{Eq. 3.31}$$

This solution wall contour point is used in the calculation of the next streamline point for the next "backward" calculated C_- characteristic. A complete program description is located in Section 3.4 with the program's flowchart and MatLab source code available in Appendix B.

3.2 Aerospike Method of Characteristics Background

Axisymmetric, Irrotational External Aerospike Nozzle:

Aerospike nozzles and annular nozzles are similar. For aerospike nozzles, the flow is bounded on one side by a wall where as in an annular nozzle the flow is bounded by a wall contour on all sides. Figure 3.2.1 illustrates a generic Aerospike nozzle configuration.

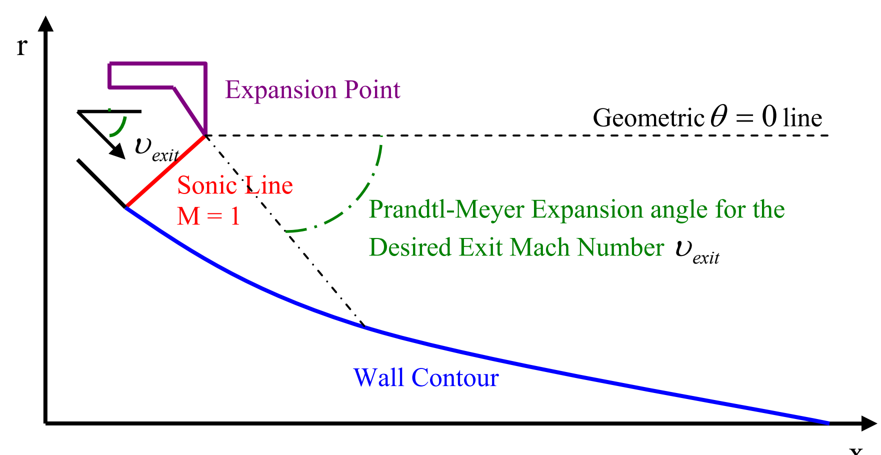


Figure 3.2.1: *Geometry of a Generic Aerospike Nozzle*

It has been found that for maximum thrust, the flow direction of the fluid under sonic conditions should be offset from the axisymmetric line by an angle equal to υ_{exit}, the Prandtl-Meyer expansion angle associated with the desired exit Mach number of the nozzle.

To achieve the desired exit Mach number, the flow must be turned through a Prandtl-Meyer expansion fan. The total angle of the expansion fan can be calculated using equation *3.9* for the desired exit Mach number. Since the flow is turned only by an expansion fan, the characteristics emanating from the expansion point can be approximated by straight lines. As shown for an annular nozzle, the characteristics for axisymmetric, irrotational flows are Mach lines. In the case of an aerospike nozzle, these characteristics originate from the expansion point. The expansion fan can be thought of as containing an infinite number of these characteristics (Mach waves), each making the Mach angle α with the local flow direction. These Mach waves turn and accelerate the flow. For smooth, shock-free expansion, each Mach wave must be non-reflectively canceled by the nozzles contour. From Expansion Fan Theory, the characteristic equations for an axisymmetric aerospike nozzle can be derived. Figure 3.2.2 helps illustrate this derivation.

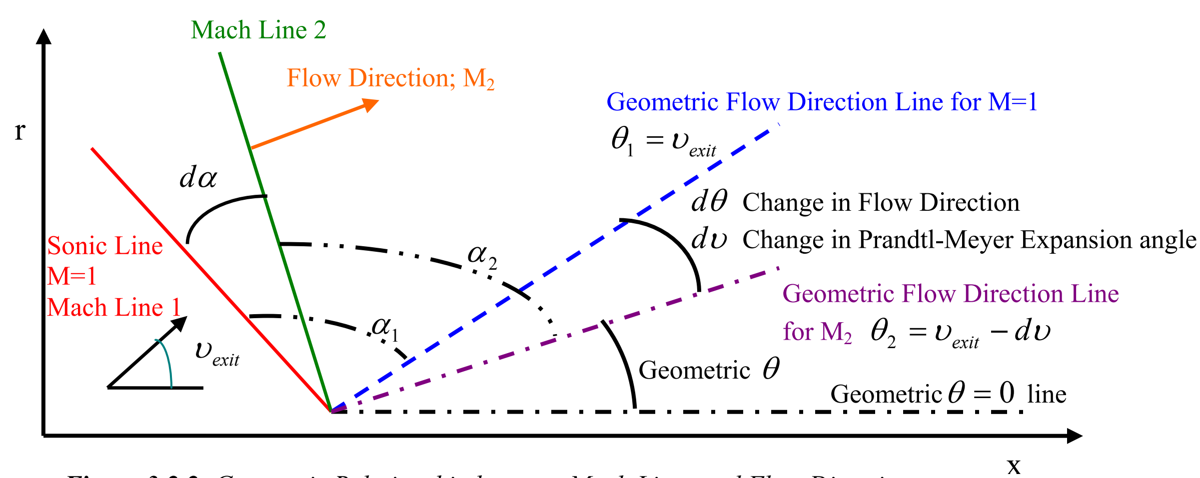


Figure 3.2.2: *Geometric Relationship between Mach Lines and Flow Direction*

Looking at Figure 3.2.2, it can be see that the blue-dashed line indicates the direction of the flow at the sonic line given by the relationship

$$\theta_{SonicLine} = \upsilon_{exit} \qquad \textit{Eq. 3.32}$$

The purple-dot-dashed line indicates the direction of the flow after the flow has past through a characteristic line (Mach wave) with a change in Prandtl-Meyer expansion angle of $d\upsilon$ and in turn, a change in flow direction of $d\theta = d\upsilon$. Since the flow at the throat is sonic, i.e. a Mach number equal to one, equation 3.9 shows that the Prandtl-Meyer expansion angle at the throat is equal to zero. If the incremental change in Prandtl-Meyer expansion angle, $d\upsilon$, is known for each Mach wave, the Mach number along each Mach wave could be calculated by using a root finding routine as previously described previously for equation *3.9*. Once the Mach number is calculated, the Mach angle α for each Mach wave can be calculated using equation *3.10*. Now that Mach angle is now known, the geometric angle the characteristic makes with the x-axis can be calculated from Expansion Fan Theory using

$$\sigma_{MachLine} = \alpha_{MachLine} + \left(\upsilon_{exit} - d\upsilon\right) \qquad \textit{Eq. 3.33}$$

If we assume that the location of the expansion point is known combined with the knowledge of the calculated geometric angle the characteristic makes with the x-axis, an equation approximating the characteristic emanating from the expansion point can be obtained. Assuming that the expansion point is located at $(0,0)$, the characteristic equation for an aerospike nozzle becomes

$$r_{MachLine} = \tan\left(\sigma_{MachLine}\right) \cdot x \qquad \textit{Eq. 3.34}$$

Notice that except for the variable r, the analysis does not stipulate that this is an axisymmetric solution. As with the previously discussed annular nozzle, the Stream Function can be utilized to calculate the wall contour of the aerospike nozzle. The axisymmetric Stream Function solved for and discretized in equations *3.17*-*3.26* ensures that the solution is axisymmetric.

The next step is to define an initial streamline condition. This is done by assuming a throat length of one. This also allows for a non-dimensional calculation scheme. Assuming the length of the sonic line is 1, solving equation *3.10* for the sonic condition of M = 1, solving equation *3.34* for the slope of the characteristic defining the sonic line and assuming that the location of the expansion point is $(0,0)$, the location of the initial point on the wall contour can be solved for from geometry by

$$x_1 = 1.0\cos\left(\frac{\pi}{2} + \upsilon_{exit}\right) + x_{ExpansionPo\,\text{int}} \qquad \textit{Eq. 3.35a}$$

$$r_1 = 1.0\sin\left(\frac{\pi}{2} - \upsilon_{exit}\right) + r_{ExpansionPo\,\text{int}} \qquad \textit{Eq. 3.35b}$$

The final condition needed to solve for in order to calculate the rest of the wall contour points satisfying the Stream Function is the geometric flow direction along each characteristic. According to Angelino, the flow direction at the sonic line should be equal to the Prandtl-Meyer expansion angle with respect to the desired exit Mach number to obtain maximum thrust, see equation *3.32*.

Since the characteristics are straight lines and no other characteristics intersect them, the flow along the characteristics exhibit a constant Prandtl-Meyer expansion angle and flow direction equal to their values at the expansion point associated with their respective characteristics. According to Figure 3.2.1, this means the flow direction angle is decreasing at the same rate as the Prandtl-Meyer expansion angle is increasing.

$$\theta = \upsilon_{exit} - \upsilon \qquad \textit{Eq. 3.36}$$

$$\upsilon = d\upsilon \cdot Step\# \qquad \textit{Eq. 3.37}$$

Sweeping through the expansion fan by an incremental change in Prandtl-Meyer expansion angle, $d\upsilon$, all variables for the characteristics are known using equations *3.33* through *3.37*.

Since each characteristic's equation is defined throughout the expansion fan, a similar method described above for calculating the points satisfying the Stream Function for the annular nozzle can be utilized to find the points defining the wall contour of the aerospike nozzle. The calculation is stepped through the expansion fan by an user-defined incremental change in Prandtl-Meyer expansion angle until the direction of the flow, θ, is equal to zero.

In order to achieve a better thrust to weight ratio, the aerospike nozzle can be truncated. The truncation is based on a user-defined percentage of the total length of as if the flow was allowed to reach its final flow direction of 0 radians. The truncated nozzle's contour points are the same as the ideal length nozzle's. The truncated nozzle's contour ends when its x-component equals the user-defined percentage of the x-component of the last contour point of the ideal length.

Axisymmetric, Irrotational Internal-External Aerospike Nozzle:

An Internal-External aerospike nozzle is very similar to an external aerospike nozzle except that a portion of the flow is accelerated in a confined internal nozzle. Essentially the flow is accelerated by two separate Prandtl-Meyer expansion fans. Figure 3.2.3 gives a typical IE aerospike configuration.

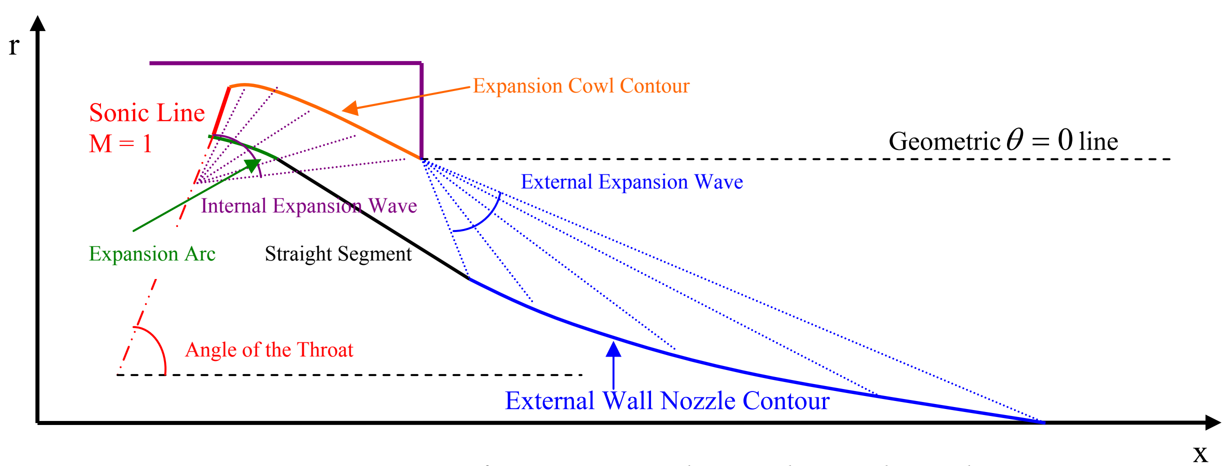


Figure 3.2.3: *Geometry of a Generic Internal-External Aerospike Nozzle*

The two expansion fans are divided up per the designer's desire. For the cases studied in this paper, the exit Mach number of the internal expansion section is set to 2.0. Using equation

3.9, the Prandtl-Meyer expansion angle of the internal expansion fan, υ_{ei}, is known. This leaves the external Prandtl-Meyer expansion angle, υ_{ext}, equal to $\upsilon_{\max} - \upsilon_{ei}$, equation *3.38*.

$$\upsilon_{ext} = \upsilon_{\max} - \upsilon_{ei} \qquad \textit{Eq. 3.38}$$

It is important to note that the two expansion fans exhibit different types of characteristics. As drawn in Figure 3.2.3, the Mach lines of the internal expansion wave are C_+ characteristics and the Mach lines of the external expansion fan are C_- characteristics. The geometry the calculations are evaluated upon dictates which characteristics are associated with each expansion fan.

Regardless of the orientation of the geometry on which the calculations are carried out, it is important to find the global flow direction at the throat of the IE aerospike. From geometry, it can be shown that the flow direction at the throat is

$$\theta_{throat} = \upsilon_{\max} - \left(2 \cdot \upsilon_{ei}\right) \qquad \textit{Eq. 3.39}$$

in the case where the nozzle's exit flow direction is 0 radians. Knowing the flow direction at the throat and fixing one end of the throat at a point of the designer's choosing, the throat and the center of the expansion arc from which Mach lines originate can be defined. Assuming that the end of the throat that is on the expansion cowl has the coordinates (0,0), the length of the throat is one, and the internal expansion region of the IE aerospike is an arc, the center of the expansion arc can be calculated. It is found by calculating along the characteristic of the sonic line from the end of the throat on the expansion cowl to a distance of $\left(1+\beta\right)$, where β is the user-defined radius of the expansion arc. Geometry yields equations *3.40* and *3.41* which calculate the x- and r-component of the center's coordinates, respectively.

$$xCenter = \left(1+\beta\right)\sin\left(\theta_{throat}\right) \qquad \textit{Eq. 3.40}$$

$$rCenter = \left(1+\beta\right)\cos\left(\theta_{throat}\right) \qquad \textit{Eq. 3.41}$$

The next step is to define the geometry from which the expansion arc can be defined. This is done utilizing isosceles triangles and geometry. Figure 3.3.4 illustrates the geometry that will be used in the following discussion. The technique used in calculating the expansion arc contour was developed for this thesis.

Since the calculation of the nozzle's contour is driven by an incremental change in the Prandtl-Meyer expansion angle, the flow direction tangent to the arc is equal to the Prandtl-Meyer expansion angle the calculation has been swept through. Using this knowledge and geometry in Figure 3.2.4, the coordinates of the points on the expansion arc can be calculated. From the Law of Cosines for an isosceles triangle, the chord length between the end point of the throat on the expansion arc and the point on the expansion arc for the current Mach line can be calculated by equation *3.42*.

$$ChordLength = \sqrt{2\beta^2\left(1-\cos(\upsilon_{CurrentPoint}\right)} \qquad \textit{Eq. 3.42}$$

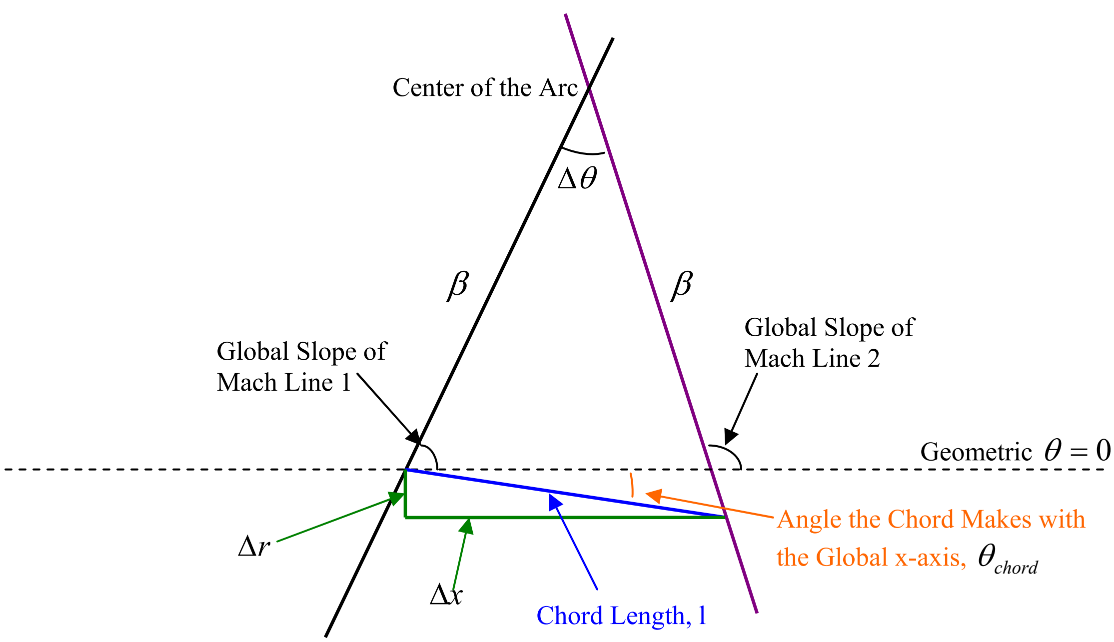


Figure 3.2.4: *Geometry Used to Calculate the Expansion Arc of the Internal-External Aerospike*

From geometry and the isosceles triangle consisting of two radii of the arc and the chord length between them, the angle the chord makes with the x-axis can be calculated using equation *3.43*.

$$\theta_{chord} = \frac{\pi}{2} - \theta_{throat} - \left(\frac{\pi - \upsilon_{CurrentPo\,\mathrm{int}}}{2}\right) \qquad \textit{Eq. 3.43}$$

Using the geometry of a right triangle, the change in the x- and r-component from the end point of the throat on the expansion arc to the next point on the expansion arc can be calculated by equations *3.44* and *3.45*.

$$\Delta r = ChordLength \cdot \sin(\theta_{chord}) \qquad \textit{Eq. 3.44}$$

$$\Delta x = ChordLength \cdot \cos(\theta_{chord}) \qquad \textit{Eq. 3.45}$$

Which lead to equations *3.46* and *3.47* for calculating the coordinates of the next point on the expansion arc.

$$x_i = x_{throat,arc} + \Delta x \qquad \textit{Eq. 3.46}$$

$$r_i = r_{throat,arc} + \Delta r \qquad \textit{Eq. 3.47}$$

These are the coordinates from which the C_- characteristics used in calculating the points that satisfies the Stream Function and define the expansion cowl contour. These calculations are continued until the Prandtl-Meyer expansion angle of the last point on the expansion arc and therefore the expansion cowl contour are greater than or equal to the Prandtl-Meyer expansion angle associated with the user-defined internal exit Mach number. The expansion cowl contour is calculated the same way the contour of the external aerospike is calculated.

Once the last point on the expansion cowl is calculated, the calculations switch from one type of characteristic to the other. The second expansion fan originates from the last point on the expansion cowl. The flow direction starts to turn parallel to the axisymmetric line of the nozzle and the Prandtl-Meyer expansion angle continues to increase. The Mach lines emanating from the last point on the expansion cowl are calculated using the appropriate characteristic equations as described for the external aerospike. The external contour of the IE aerospike is calculated by satisfying the Stream Function the same way the contours of the external aerospike was.

For a truncated IE aerospike nozzle, the contour is calculated with the same method as a truncated external aerospike nozzle.

3.3 Conical Nozzle Design

For completeness, a discussion of conical nozzle design is included. The conical nozzle is the easiest of the nozzles to design because it does not take into account the properties changes occurring in the flow field. It is calculated solely from geometry and the exit to throat area ratio for a desired exit Mach number. Since we are assuming inviscid, irrotational, isentropic flow, the exit to throat area ratio is given by equation 3.48, which is based on the desired exit Mach number and ratio of specific heats of the working fluid.

$$\frac{A_{exit}}{A_{throat}} = \sqrt{\frac{1}{Ma_{exit}^2} \cdot \left[\frac{2}{\gamma+1} \cdot \left(1 + \frac{\gamma-1}{2} \cdot Ma_{exit}^2\right)\right]^{\frac{\gamma+1}{\gamma-1}}} = \frac{\pi \cdot r_{exit}^2}{\pi \cdot r_{throat}^2} \qquad \textit{Eq. 3.48}$$

Using equation 3.48, assuming r_{throat} is equal to 1 and solving for r_{exit}, the r-component of the point on the exit of the conical nozzle's contour is known. Realizing that the contour of a conical nozzle is a straight line from the throat to the point on the exit, the only thing left to calculate is the x-component of the exit point. From Sutton, a conical nozzle works best if the half-angle of the cone is between 8° and 15°. Given this knowledge, the x-component of the point on the exit of the nozzle can be found using basic trigonometry. Figure 3.3.1 illustrates the basic geometry of a generic conical nozzle.

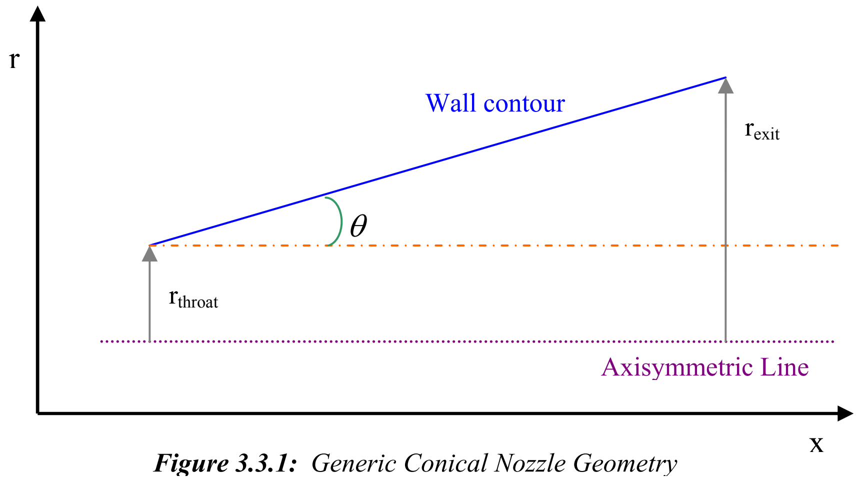


Figure 3.3.1: *Generic Conical Nozzle Geometry*

Using trigonometry of a right triangle, the x-component on the point on the exit of the conical nozzle can be found using equation 3.49.

$$x_{exit} = \frac{r_{exit} - r_{throat}}{\tan(\theta)} + x_{throat} \qquad \textit{Eq. 3.49}$$

Since in most cases the x-component of the throat is set to zero, x_{throat} can be neglected in the calculation. To complete the nozzle, the designer only needs to designate the half-angle they would like the cone to be defined by.

Conical nozzles are used in many cases because of their simplicity to design and manufacture but they also have the disadvantage of incurring larger losses than annular or aerospike nozzles.

3.4 Computer Program Calculation Details

The three supersonic nozzles discussed above are all combined into one program that will calculate the nozzles' contour using the Method of Characteristics and the Stream Function. A brief description of the subroutines developed for this paper and their associated flow charts are included below. The complete set MatLab source code and program flowcharts are available in Appendix B. Since all three nozzle types are based on isentropic relations, the codes' error can be directly quantified using isentropic area ratios for given desired exit Mach numbers and ratios of specific heats. The codes' error comparisons are available in Appendix C for different incremental changes in Prandtl-Meyer expansion angle.

Program SupersonicNozzle

The program begins by asking the user for all necessary design variables the program will need to calculate the nozzle contours. The list of variables required is described in Table 3.4.1 with description below.

Table 3.4.1	
Program Variable	**Description**
Beta	The throat multiplier that will be used to calculate the radius of the arc used in the expansion region for annular and internal-external aerospike nozzles
DeltaVAeroD	The desired incremental step size of the Prandtl-Meyer expansion angle used in the calculation. It is also used as the x-space step direction for determining the x-component of the starting point for the "backward" C_- characteristic
Gamma	Ratio of Specific Heats of the working fluid
Mexit	Desired Exit Mach Number
Percent	The % of the ideal length the user would like in the event they choose to calculate truncated versions of the aerospike nozzles
Truncate	Control variable that lets the program know if the user would like to calculate truncated versions of the aerospike nozzles

The program then passes the necessary input variables to the subroutines that need them. All input variables are passed to subroutines DentonAerospike and IEaerospike. These subroutines calculate the contour of an external and internal-external aerospike nozzle, respectively, as well as there truncated versions if applicable. Axisymmetric, the subroutine that calculates the annular nozzle contour, only requires input variables Beta, DeltaVAeroD, Gamma and Mexit. A fourth subroutine, PMtoMA, is used in calculating the Mach numbers of the points in the flowfield and will be discussed last. Once all subroutines return their solutions, subroutine SupersonicNozzle plots their nozzle contours. Figure 3.4.1 is an example of a nozzle solution plot.

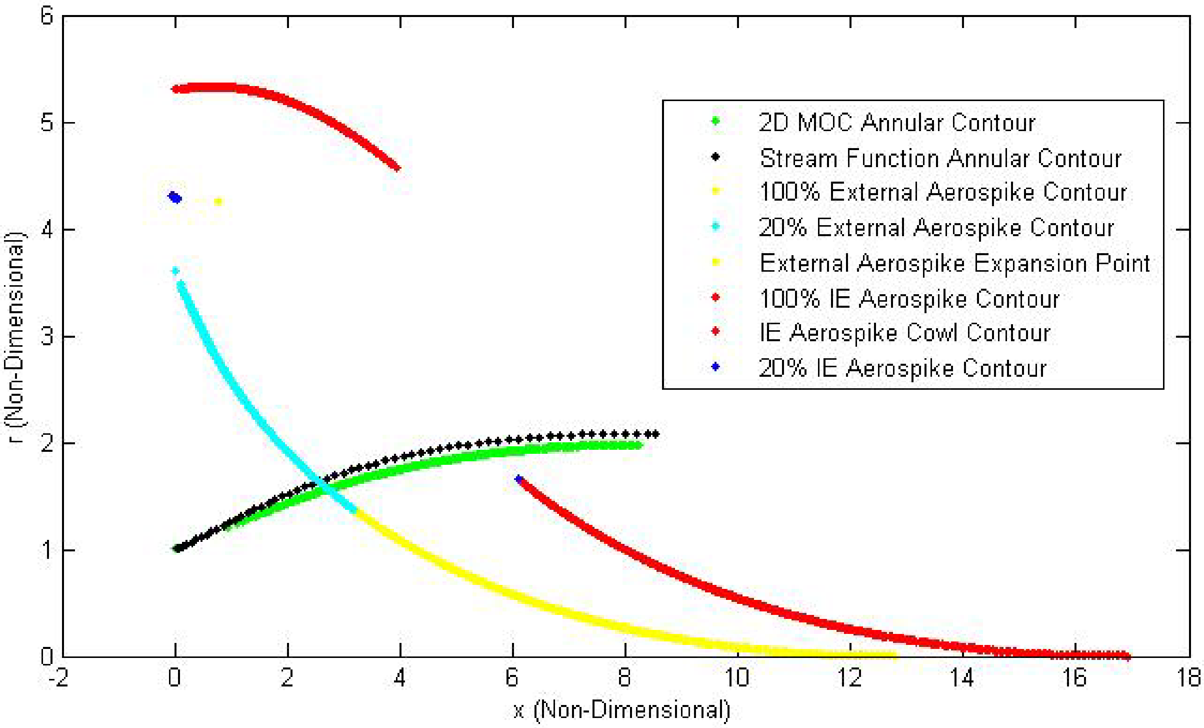


Figure 3.4.1: *Example Plot of the Program's Output Contours*

Figure 3.4.2 is the flow chart for subroutine SupersonicNozzle. The source code and a copy of the flow chart are available in Appendix B.

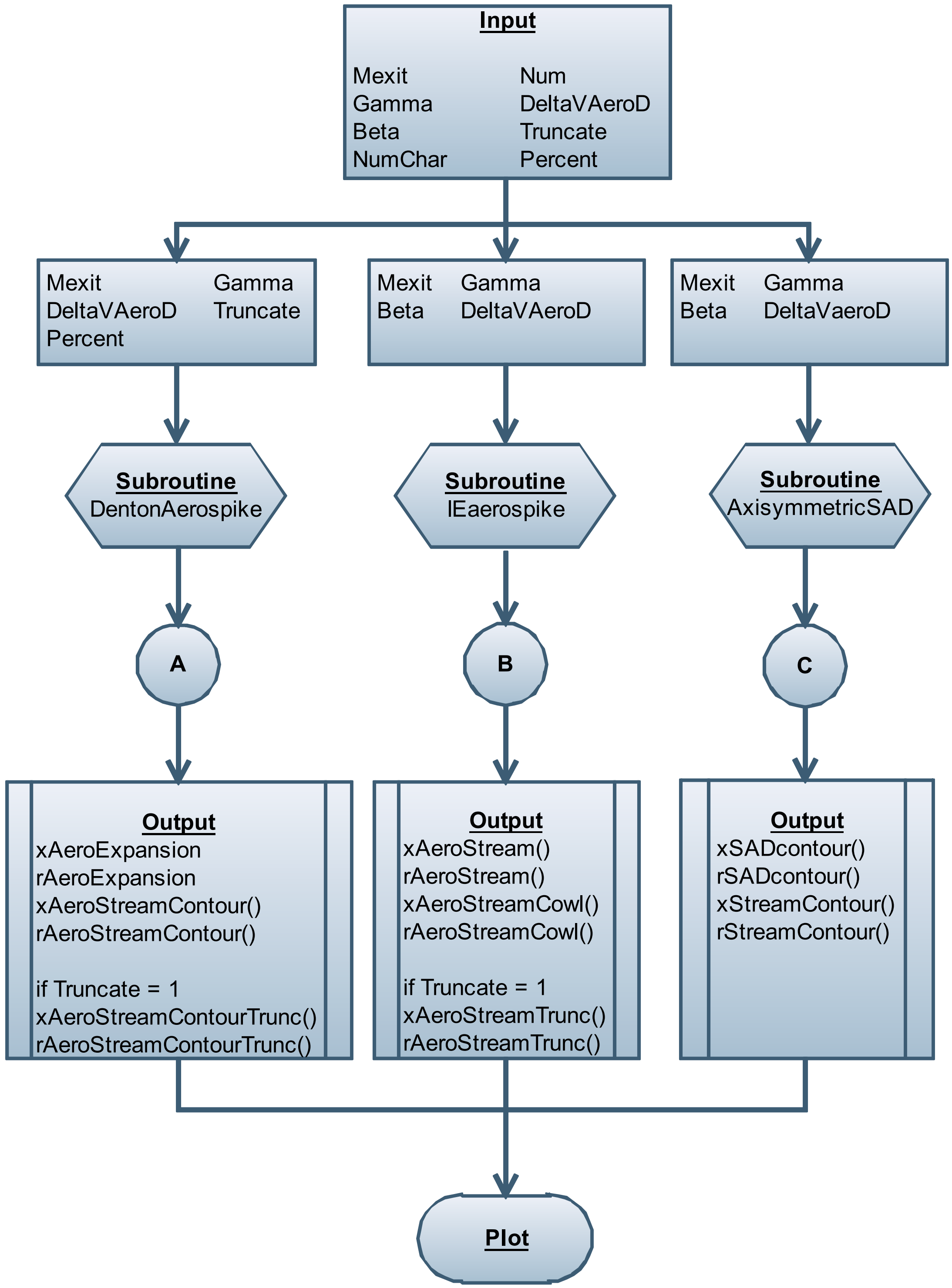


Figure 3.4.2: *Subroutine SupersonicNozzle.m Flow Chart*

Subroutine DentonAerospike

The external aerospike nozzle is calculated by subroutine DentonAerospike. DentonAerospike begins by calculating the Prandtl-Meyer expansion angle associated with the desired exit Mach number of the nozzle using equation 3.9.

The subroutine then assigns the flow conditions at the throat and calculates the coordinates of the ends of the throat. The expansion point of the throat is set to (0,0). As stated in Section 3.2, the flow direction at the throat is set equal to the Prandtl-Meyer expansion angle of the desired exit Mach number to constitute maximum thrust.

Once the throat's flow properties and position are set, the ideal length and truncated, if desired, contours of the nozzles are calculated as described in Section 3.2. The contour(s) are then plotted with the contours of the annular an IE aerospike nozzles. Figure 3.4.3 shows the flow chart of this subroutine. The source code and a copy of the flow chart for this subroutine are available in Appendix B.

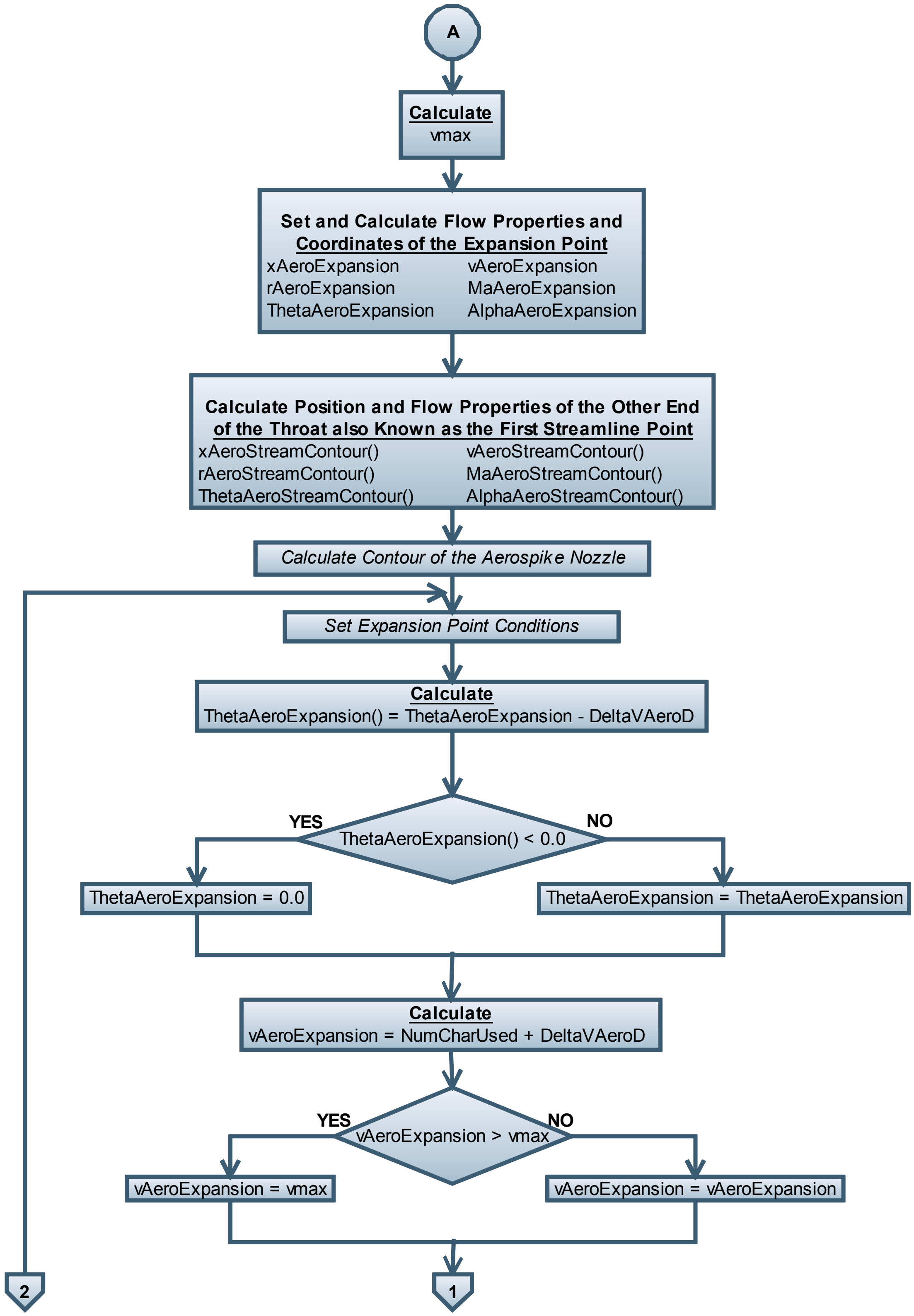


Figure 3.4.3a: *Subroutine DentonAerospike.m Flow Chart*

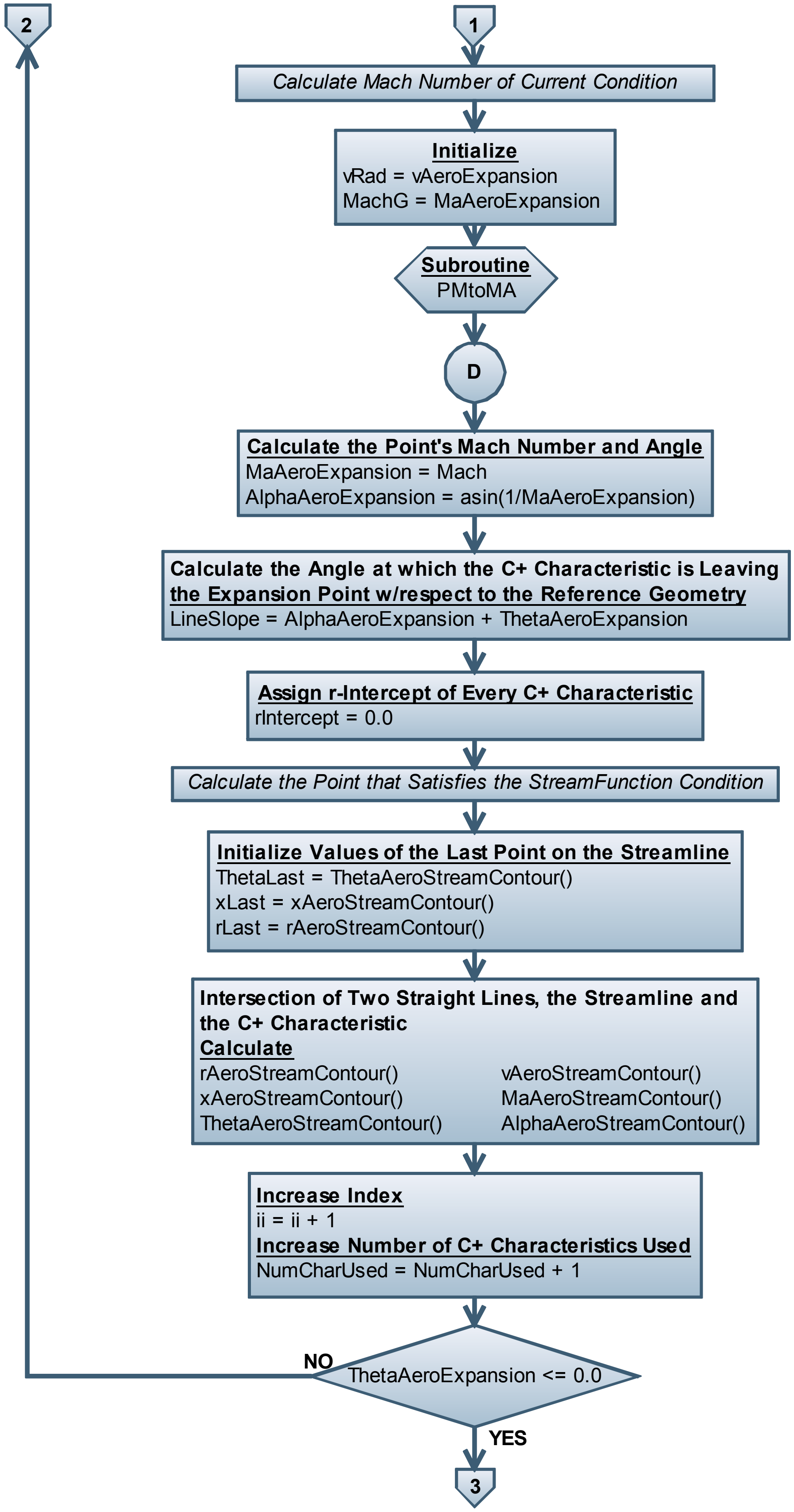


Figure 3.4.3b: *Subroutine DentonAerospike.m Flow Chart*

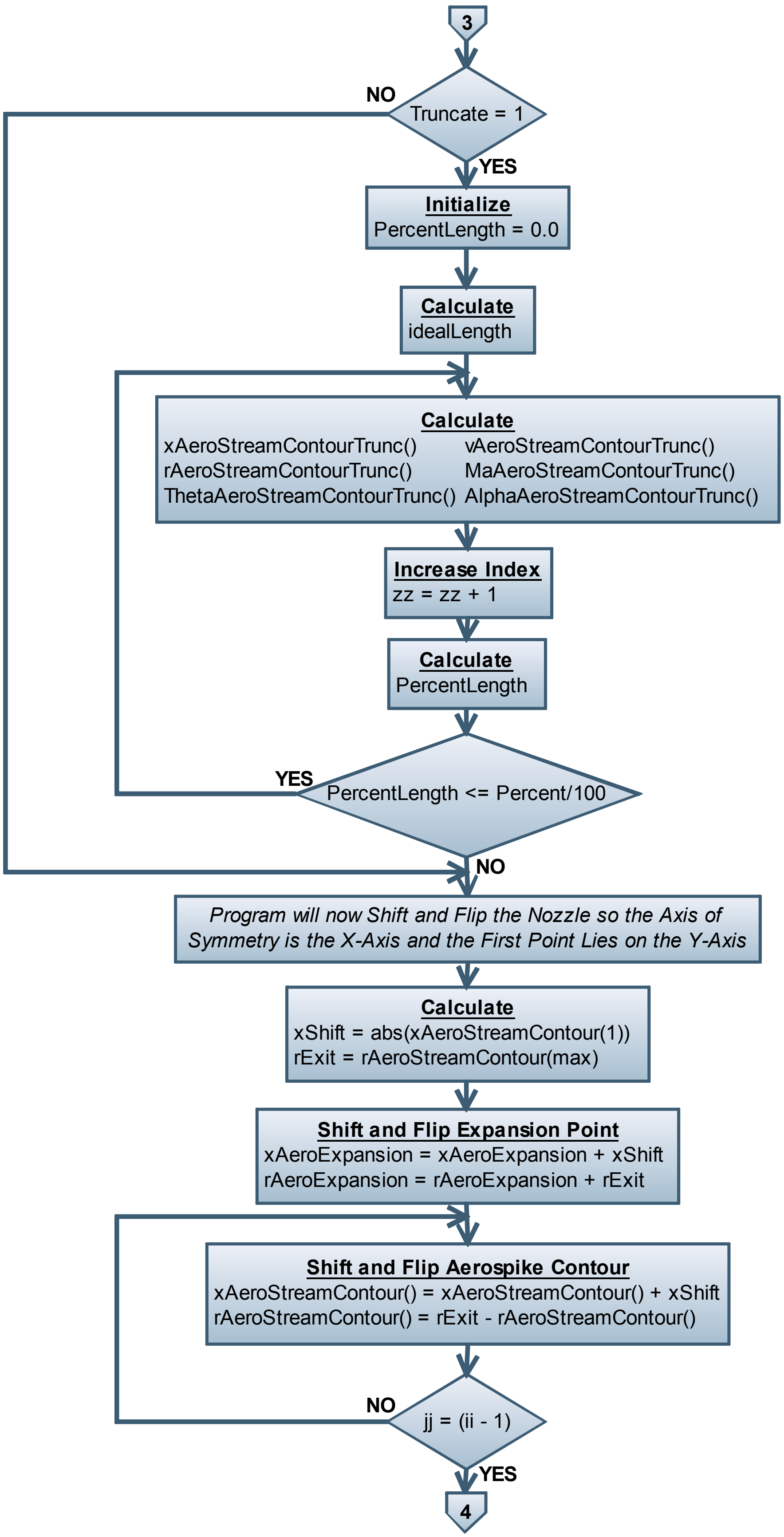


Figure 3.4.3c: *Subroutine DentonAerospike.m Flow Chart*

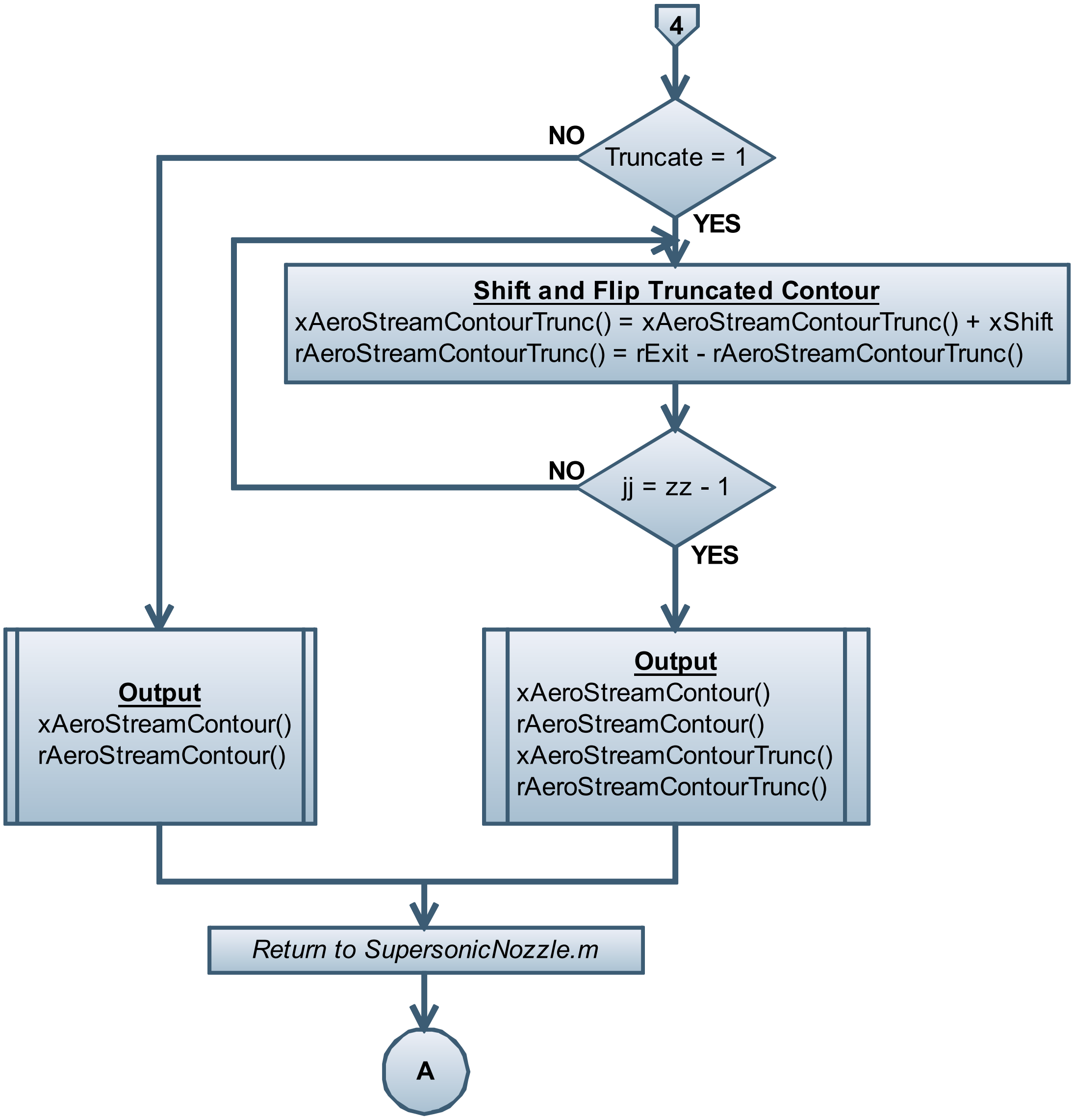


Figure 3.4.3d: *Subroutine DentonAerospike.m Flow Chart*

Subroutine IEaerospike

Subroutine IEaerospike calculates the nozzle contour for internal-external aerospike nozzles. The subroutine begins by calculating the Prandtl-Meyer expansion angles for the two expansion fans which accelerate the flow. The subroutine currently is hard coded for an exit Mach number of 2.0 for the internal section of the nozzle. The program then sets the throat conditions with the cowl end set to the coordinates (0, 0). The program then calculates the expansion arc, cowl and external contour of the IE aerospike as described in Section 3.2.

Figure 3.4.4 shows the flowchart for subroutine IEaerospike. The source code and a copy of the flow chart are available in Appendix B.

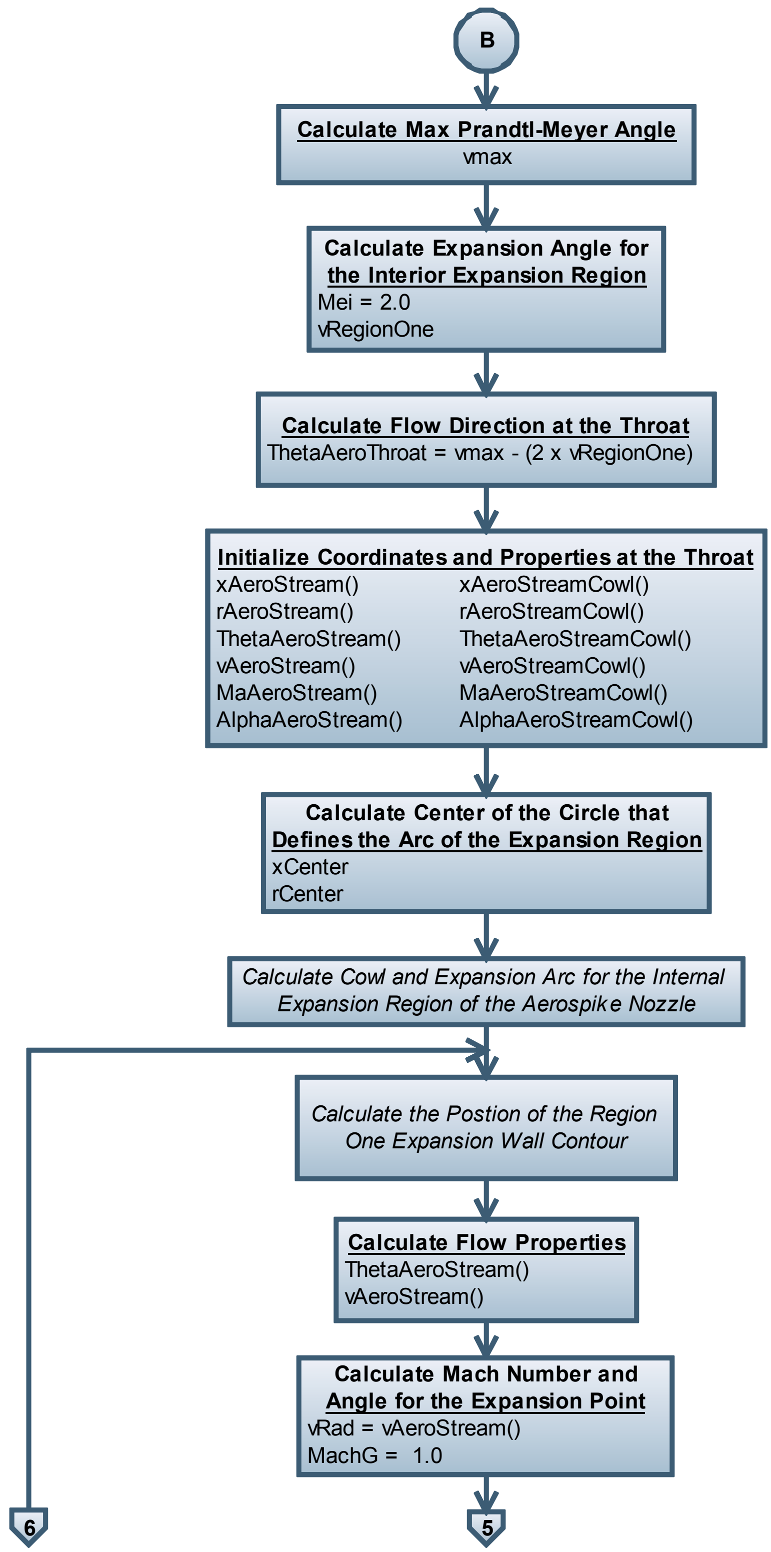


Figure 3.4.4a: *Subroutine IEaerospike.m Flow Chart*

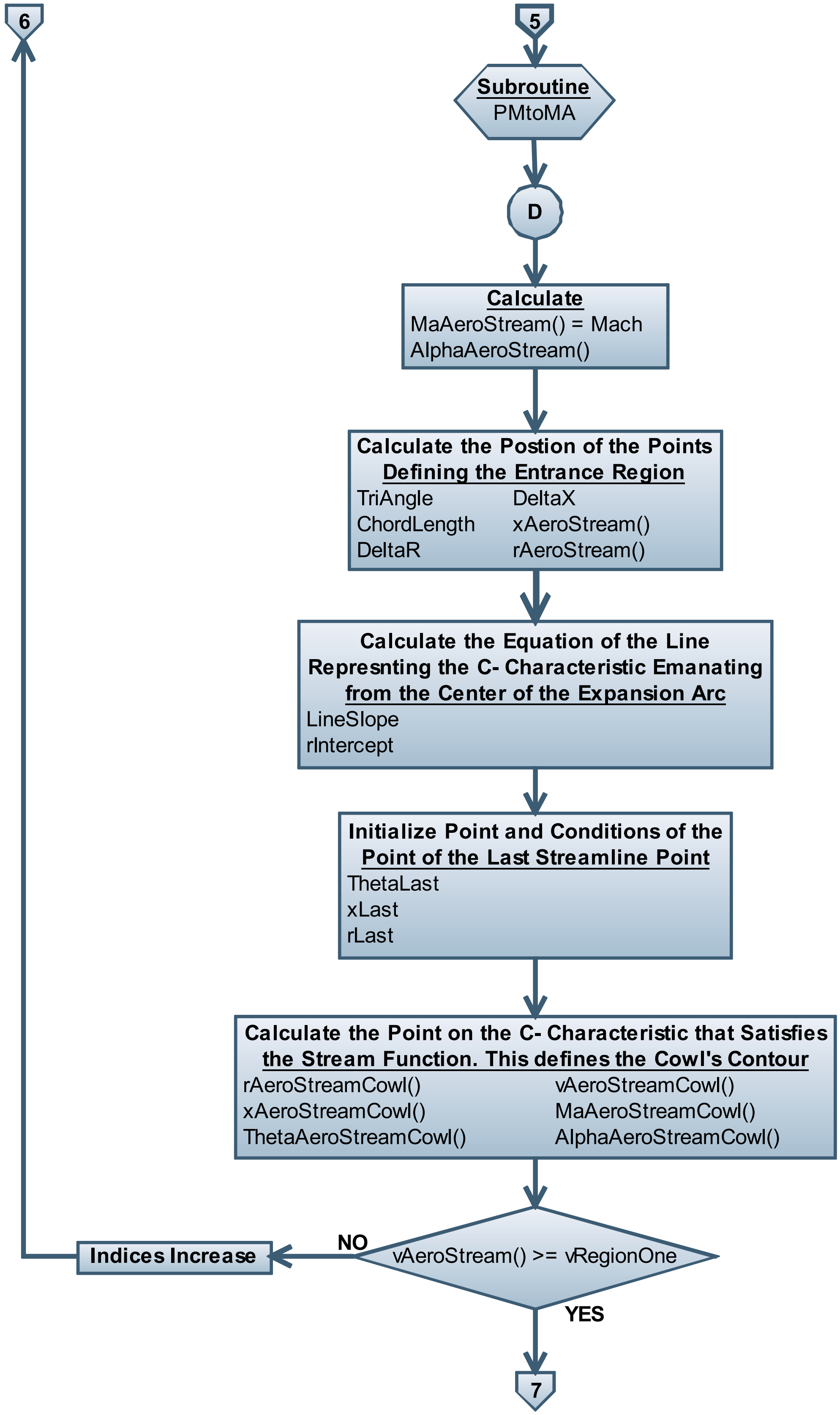


Figure 3.4.4b: *Subroutine IEaerospike.m Flow Chart*

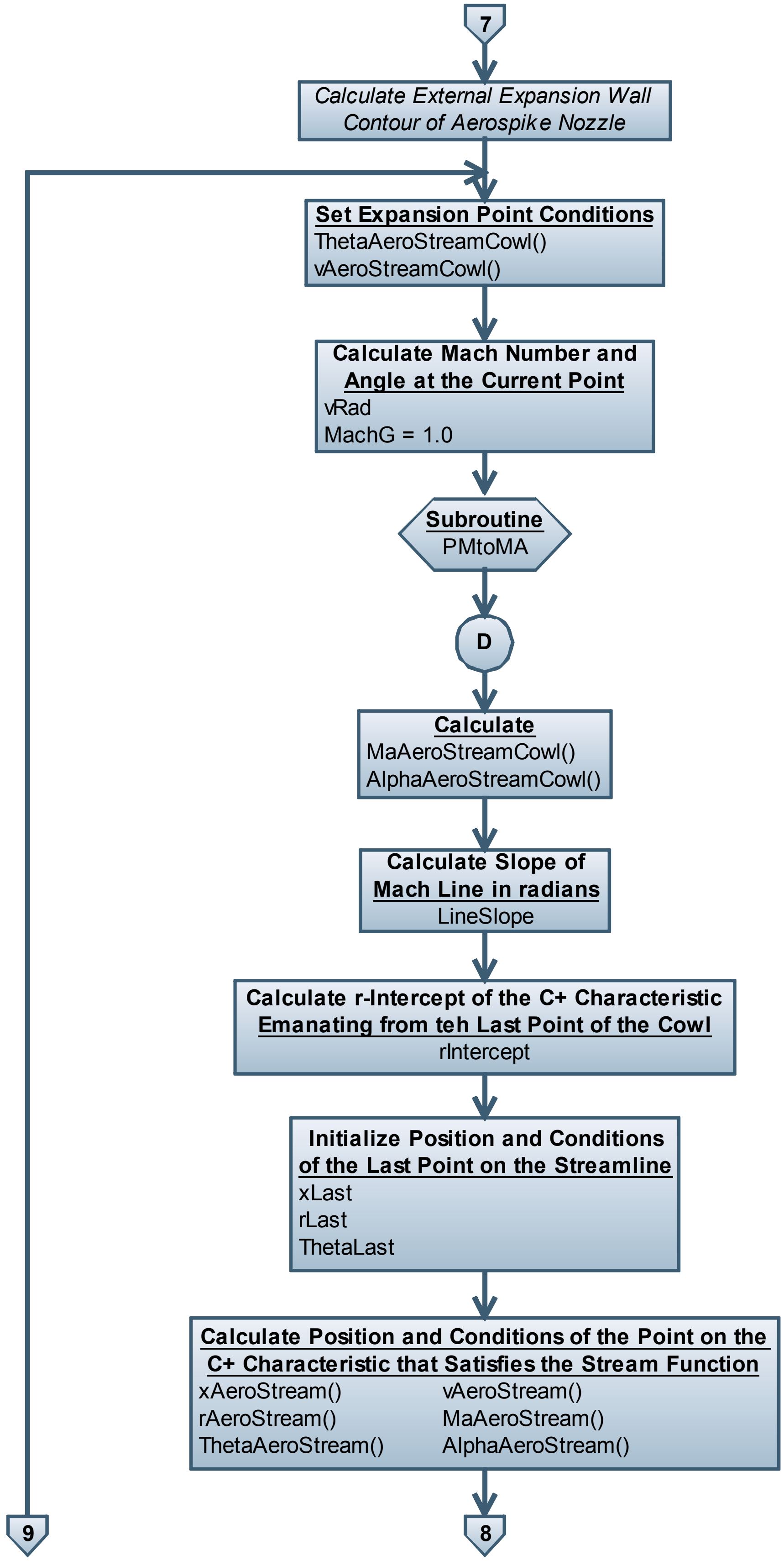


Figure 3.4.4c: *Subroutine IEaerospike.m Flow Chart*

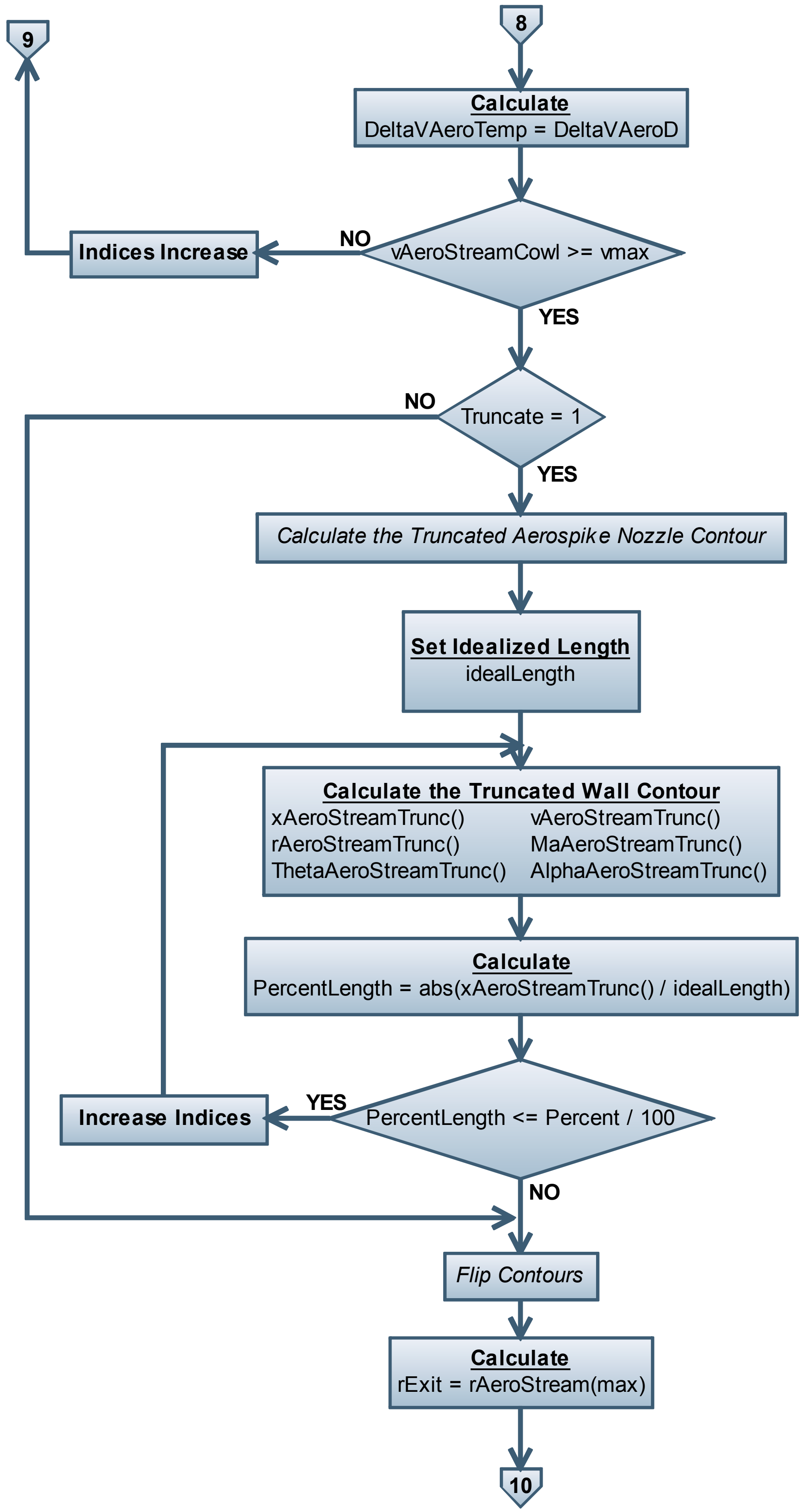


Figure 3.4.4d: *Subroutine IEaerospike.m Flow Chart*

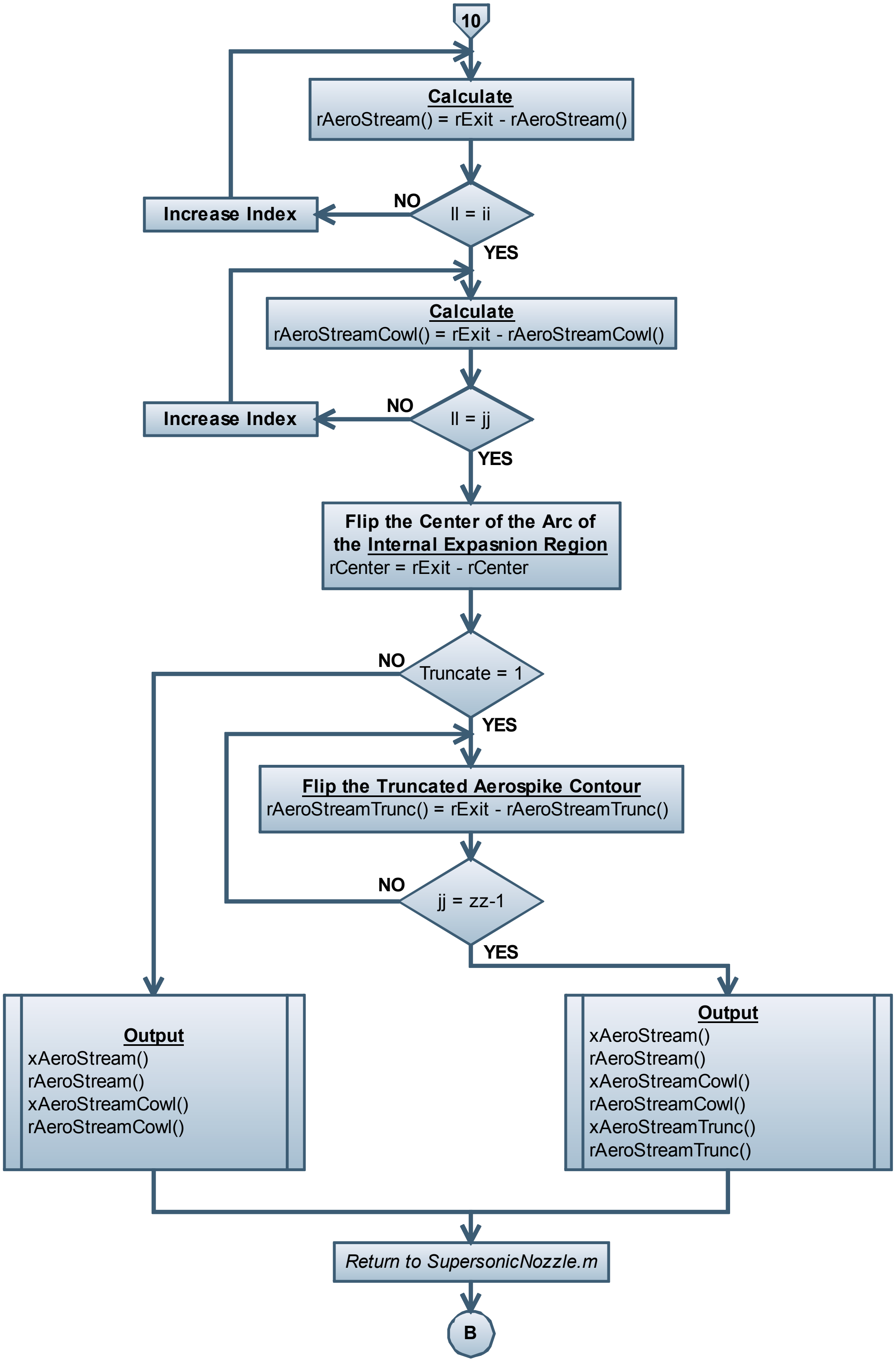


Figure 3.4.4e: *Subroutine IEaerospike.m Flow Chart*

Subroutine Axisymmetric

Subroutine AxisymmetricSAD calculates the contours of annular nozzles. The subroutine begins by setting the conditions of the throat and expansion arc. The program then calculates the expansion arc wall contour and flow field properties until the flow reaches the user-defined desired exit Mach number at the nozzle's axisymmetric line. The program then calculates the necessary "backward" C_- characteristics and wall contour using the techniques outlined in Section 3.1.

The curved nature of the characteristics is simulated by recalculating the solution to the characteristic equations until the difference in flow direction between iterations is less than abs(1e^{-10}). The flow direction was found to be the last solution property that satisfied this condition, therefore, it was used as the convergence criteria for the solution. The iterate method employed uses the average of the solution's position and properties and the origination position and properties of the C_- and C_+ characteristics and recast them as the starting conditions of their respective characteristics. This procedure is repeated until the flow direction criterion is met. Equations 3.50 through 3.61 illustrate this averaging technique.

For the C_- Characteristic:

$$\theta_{temp,C_-} = \frac{\theta_{temp,C_-} + \theta_{i+1}}{2} \qquad \textit{Eq. 3.50}$$

$$\alpha_{temp,C_-} = \frac{\alpha_{temp,C_-} + \alpha_{i+1}}{2} \qquad \textit{Eq. 3.51}$$

$$M_{temp,C_-} = \frac{M_{temp,C_-} + M_{i+1}}{2} \qquad \textit{Eq. 3.52}$$

$$\upsilon_{temp,C_-} = \frac{\upsilon_{temp,C_-} + \upsilon_{i+1}}{2} \qquad \textit{Eq. 3.53}$$

$$x_{temp,C_-} = \frac{x_{temp,C_-} + x_{i+1}}{2} \qquad \textit{Eq. 3.54}$$

$$r_{temp,C_-} = \frac{r_{temp,C_-} + r_{i+1}}{2} \qquad \textit{Eq. 3.55}$$

For the C_+ Characteristic:

$$\theta_{temp,C_+} = \frac{\theta_{temp,C_+} + \theta_{i+1}}{2} \qquad \text{Eq. 3.56}$$

$$\alpha_{temp,C_+} = \frac{\alpha_{temp,C_+} + \alpha_{i+1}}{2} \qquad \text{Eq. 3.57}$$

$$M_{temp,C_+} = \frac{M_{temp,C_+} + M_{i+1}}{2} \qquad \text{Eq. 3.58}$$

$$\upsilon_{temp,C_+} = \frac{\upsilon_{temp,C_+} + \upsilon_{i+1}}{2} \qquad \text{Eq. 3.59}$$

$$x_{temp,C_+} = \frac{x_{temp,C_+} + x_{i+1}}{2} \qquad \text{Eq. 3.60}$$

$$r_{temp,C_+} = \frac{r_{temp,C_+} + r_{i+1}}{2} \qquad \text{Eq. 3.61}$$

Figure 3.4.5 and 3.4.6 illustrate the calculation process of the flow field and axisymmetric line solution points, respectively.

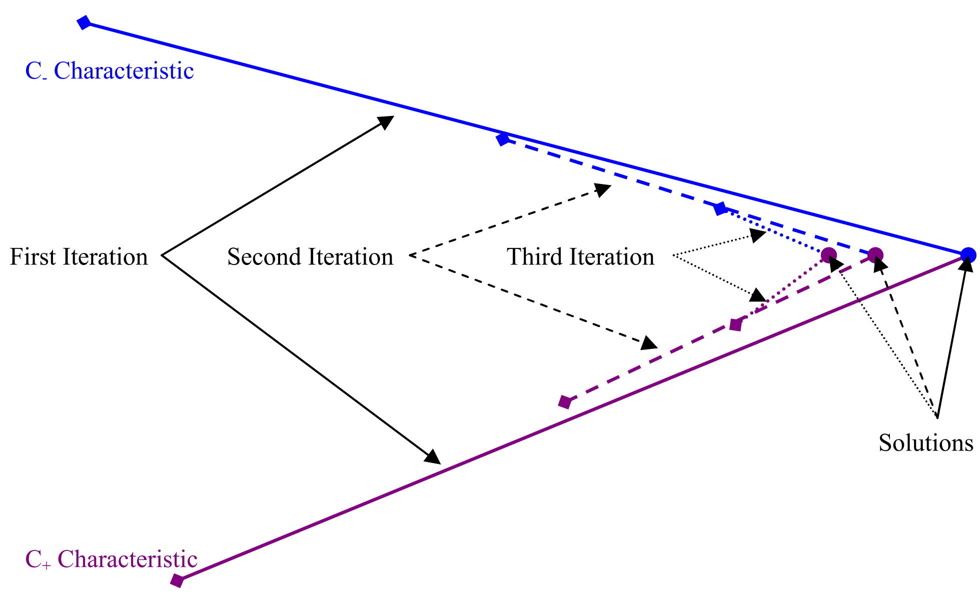


Figure 3.4.5: *Schematic of the Iteration Loop Capturing the Curved Nature of the Characteristics in the Flowfield*

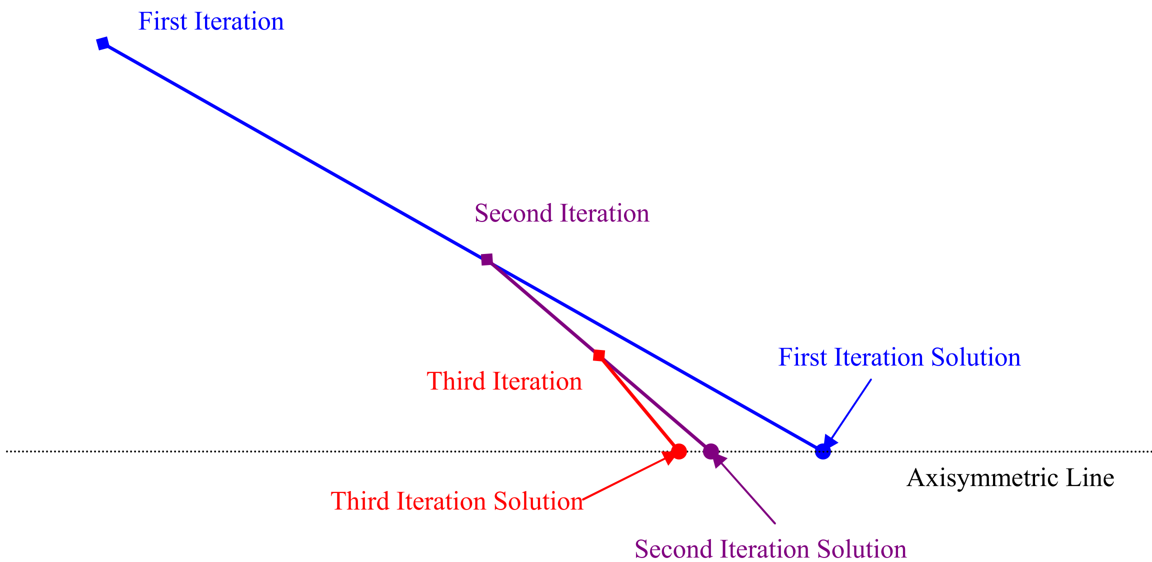


Figure 3.4.6: *Averaging Schematic Used to Capture the Curvature of the Characteristics on the Axisymmetric Line*

The flow chart of the subroutine's calculation procedure is given in Figure 3.4.7 below. The source code and a copy of the flowchart are available in Appendix B.

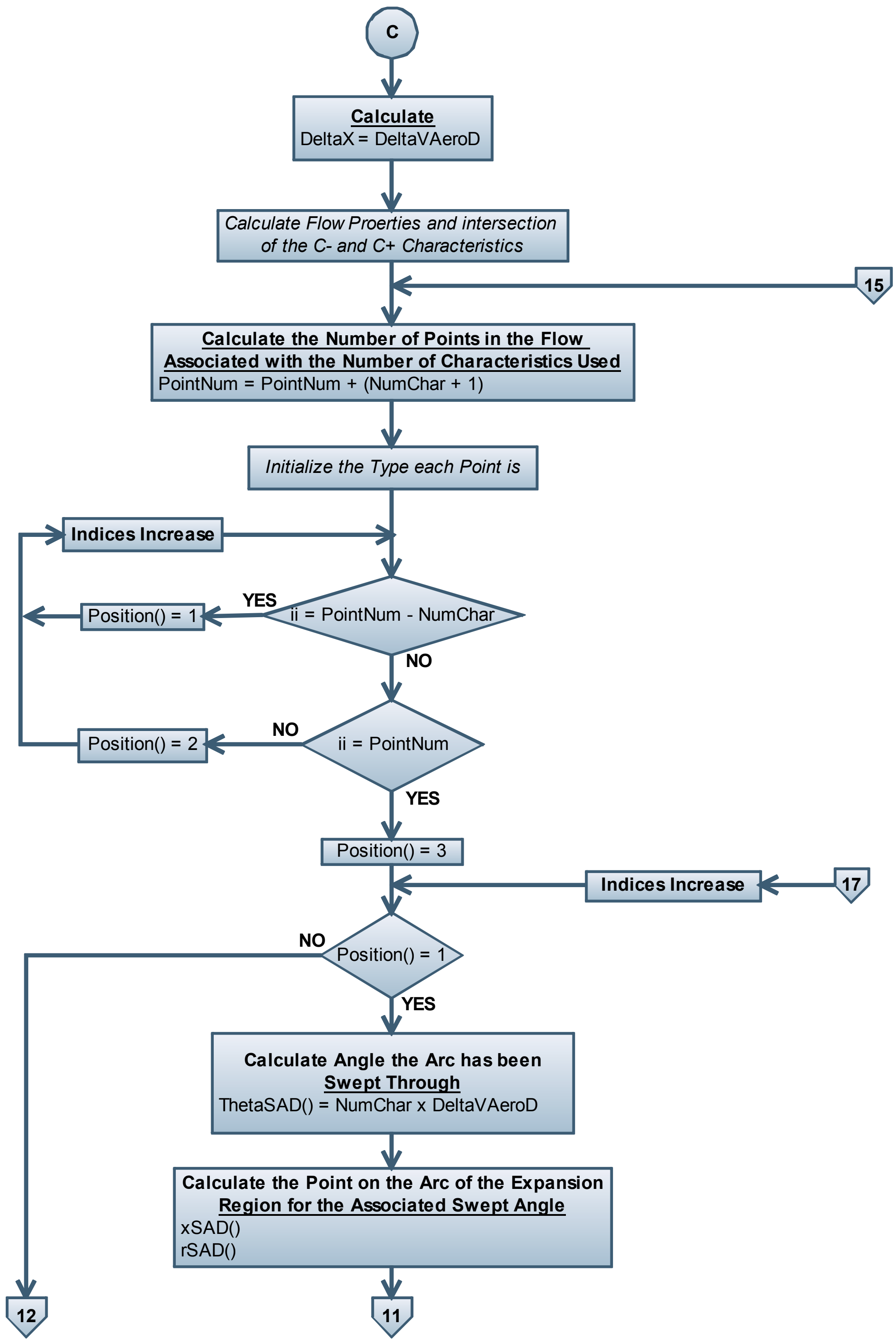


Figure 3.4.7a: *Subroutine Axisymmetric.m Flow Chart*

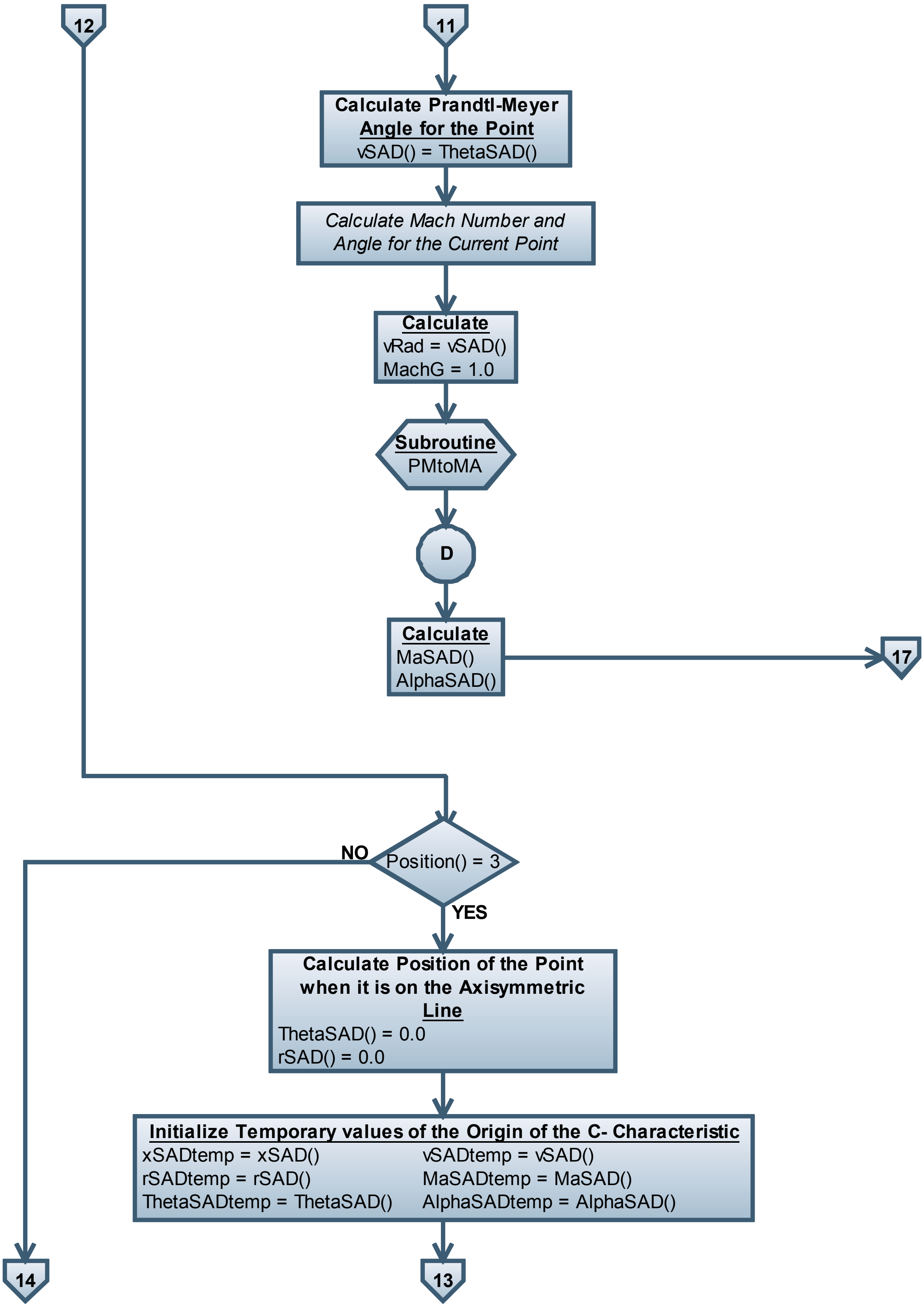


Figure 3.4.7b: *Subroutine Axisymmetric.m Flow Chart*

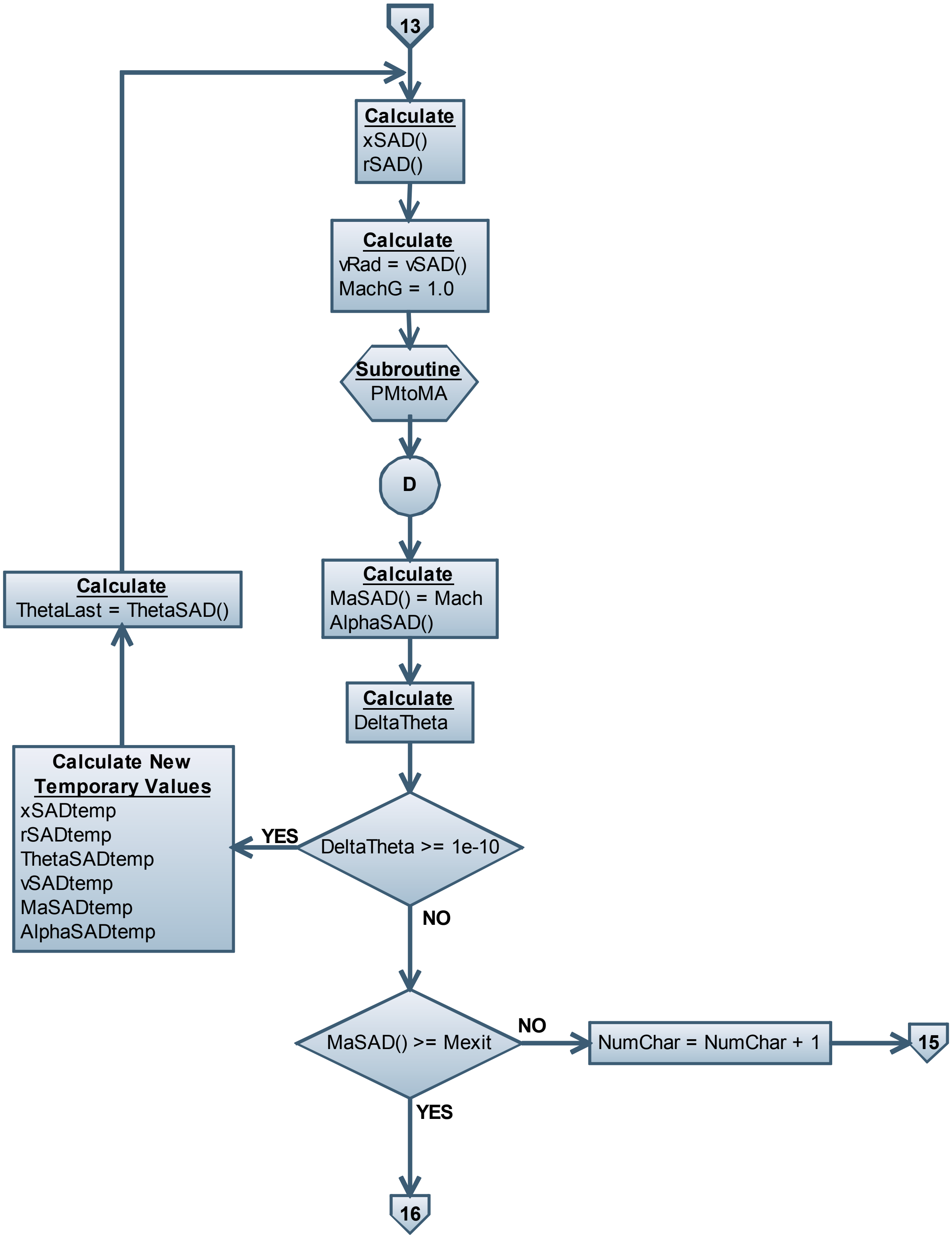


Figure 3.4.7c: *Subroutine Axisymmetric.m Flow Chart*

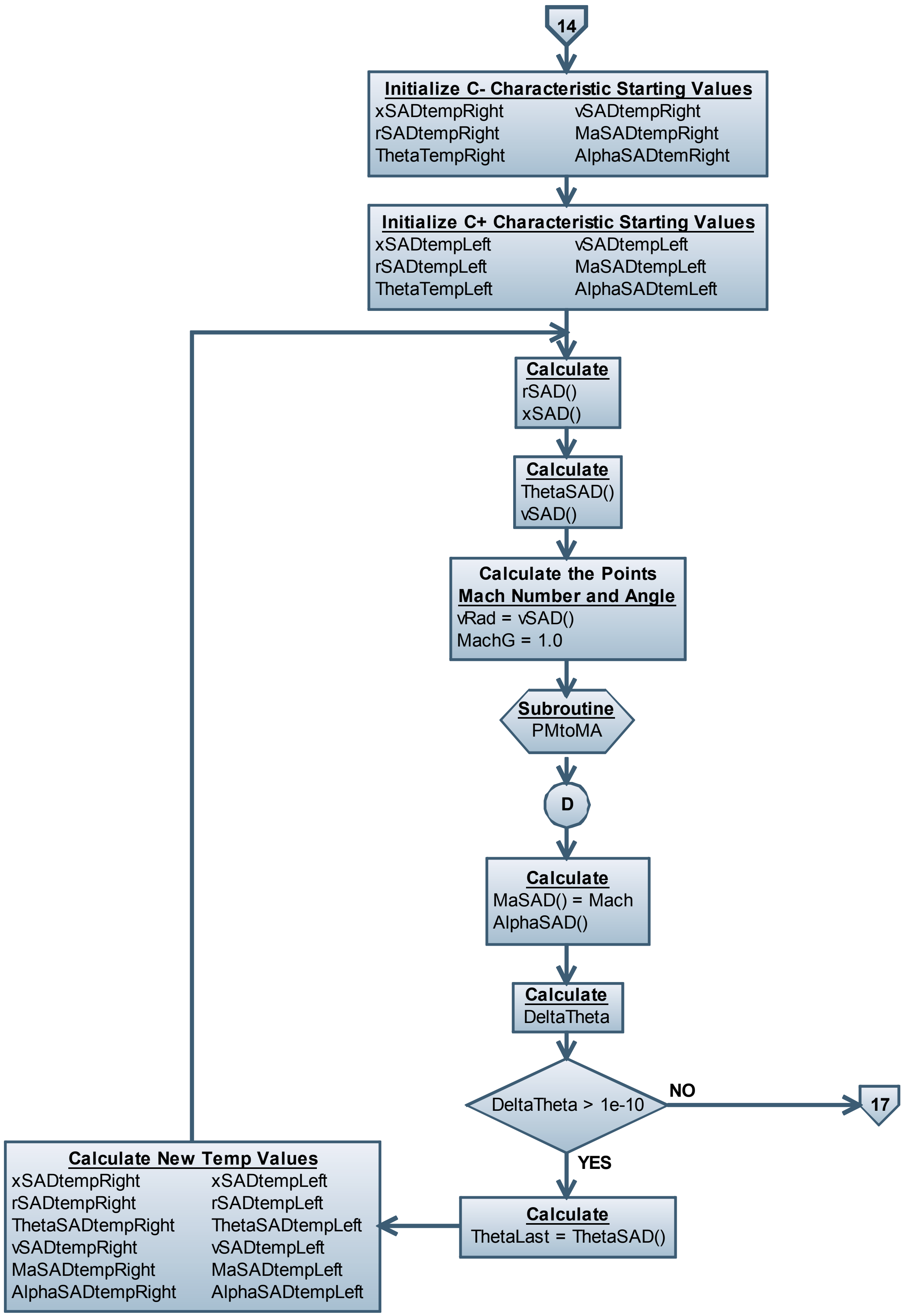


Figure 3.4.7d: *Subroutine Axisymmetric.m Flow Chart*

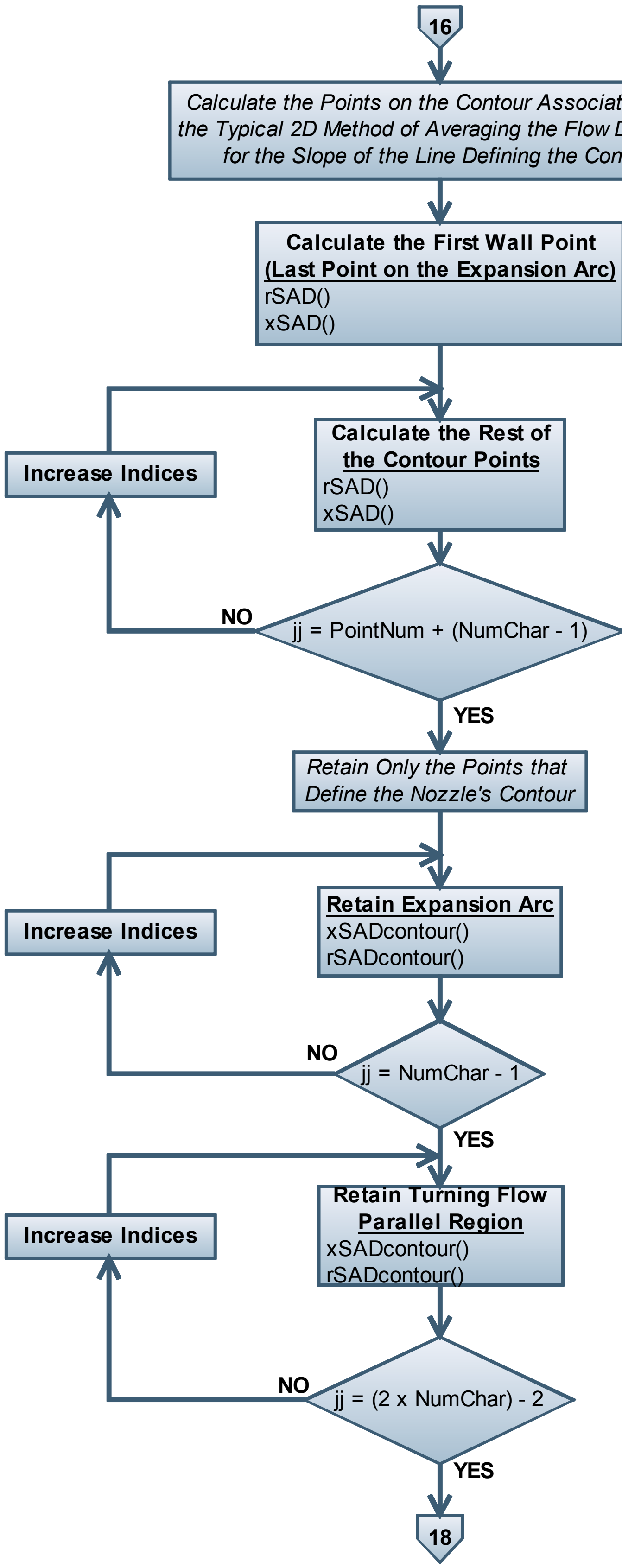


Figure 3.4.7e: *Subroutine Axisymmetric.m Flow Chart*

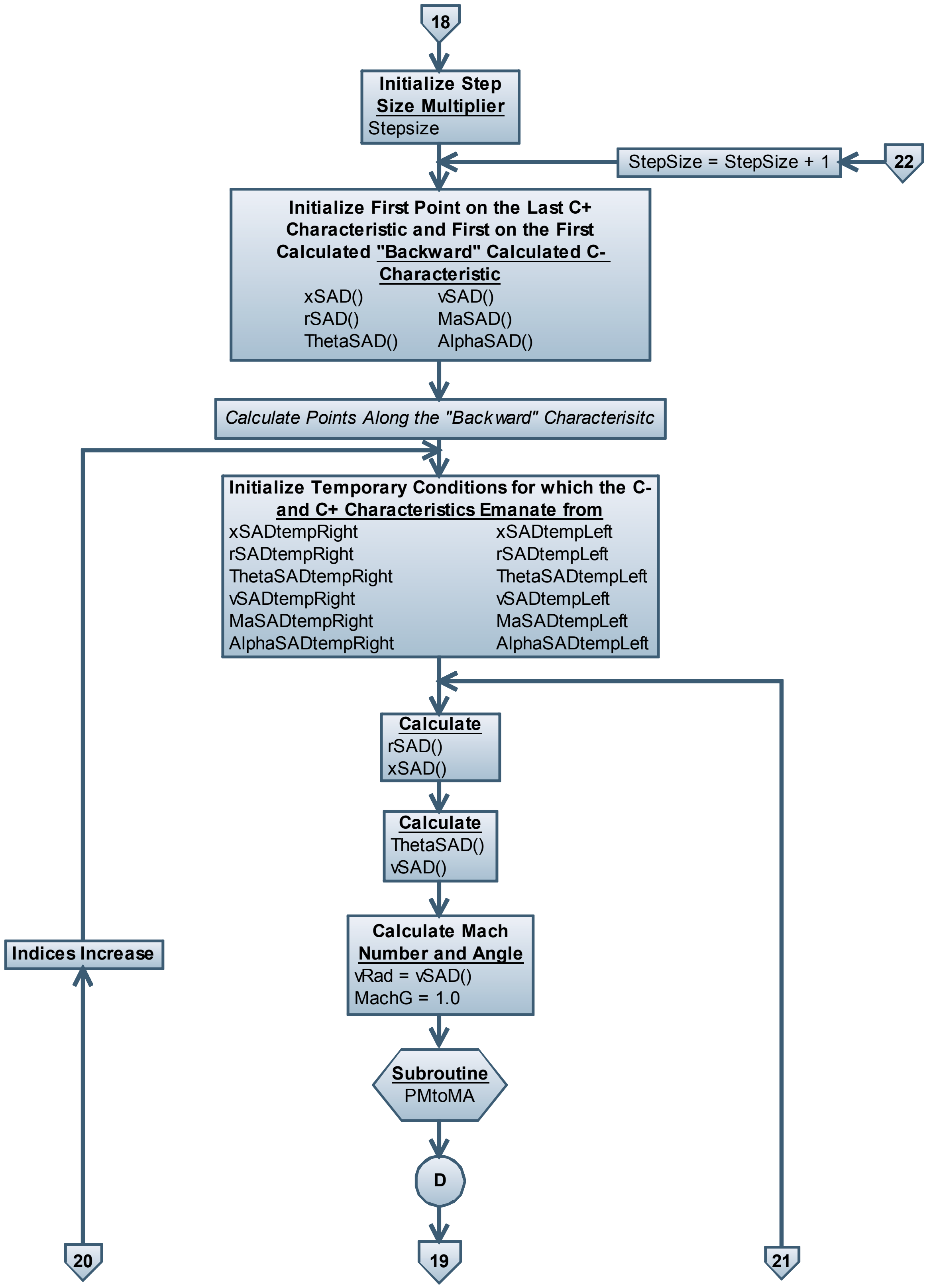


Figure 3.4.7f: *Subroutine Axisymmetric.m Flow Chart*

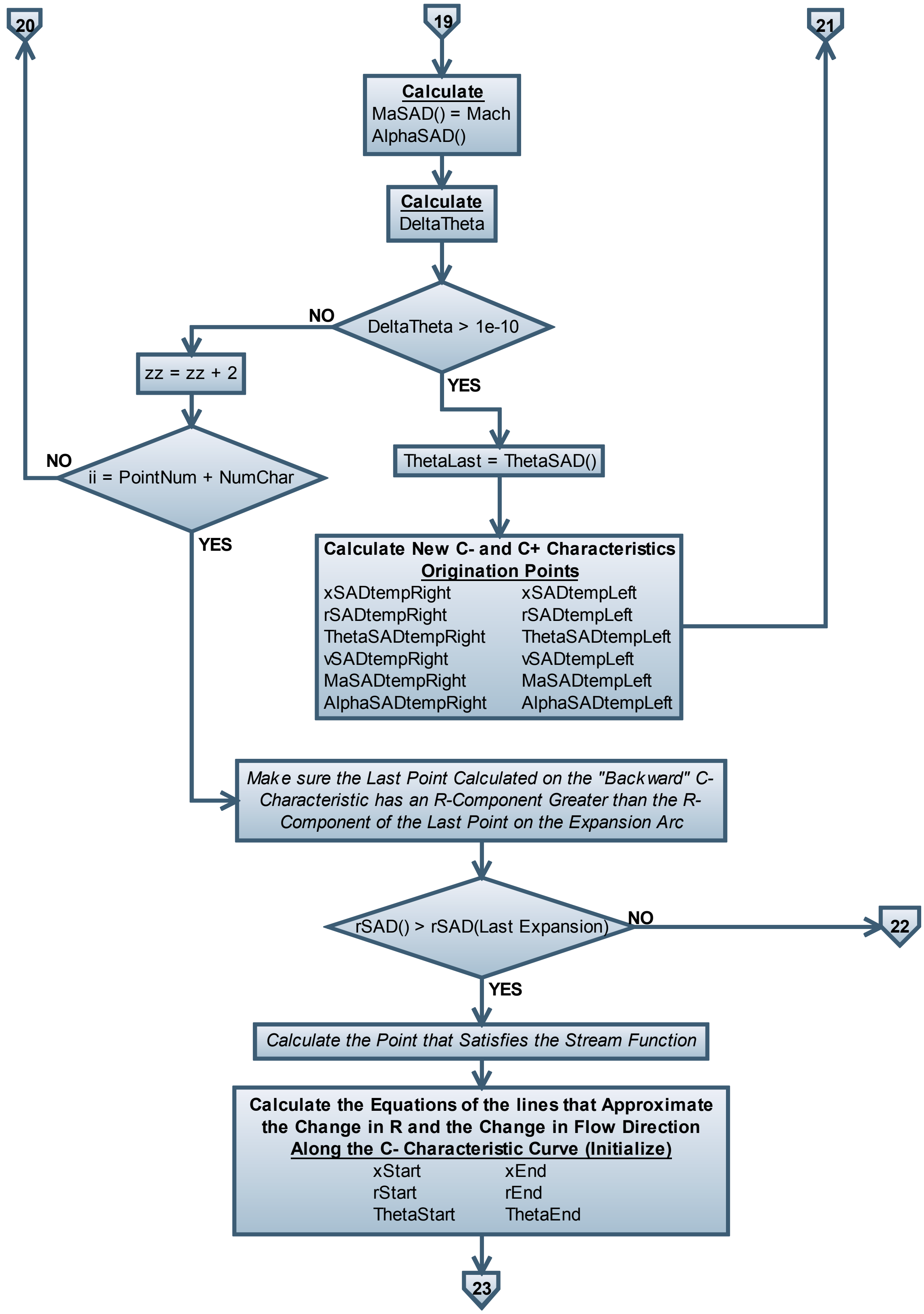


Figure 3.4.7g: *Subroutine Axisymmetric.m Flow Chart*

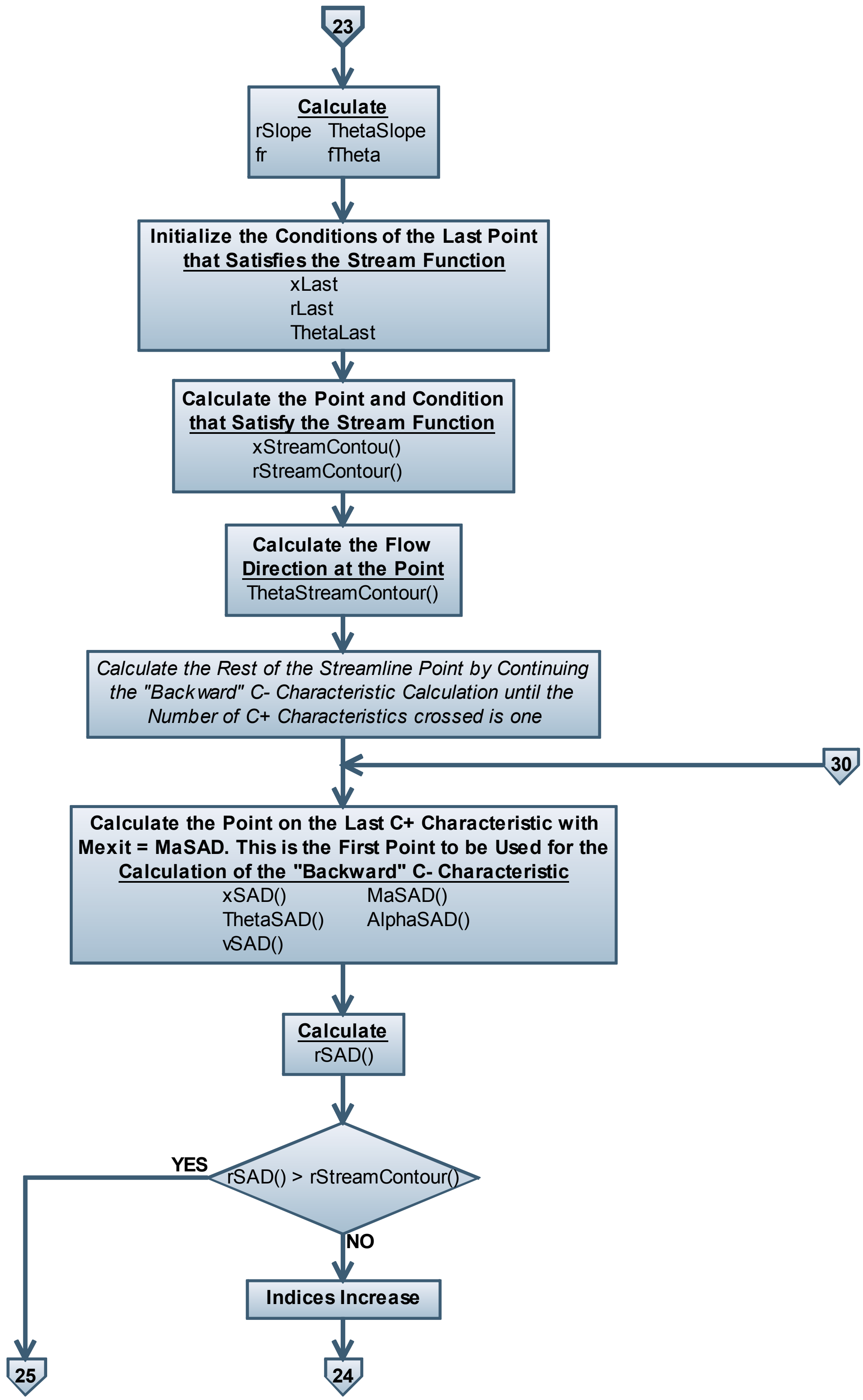


Figure 3.4.7h: *Subroutine Axisymmetric.m Flow Chart*

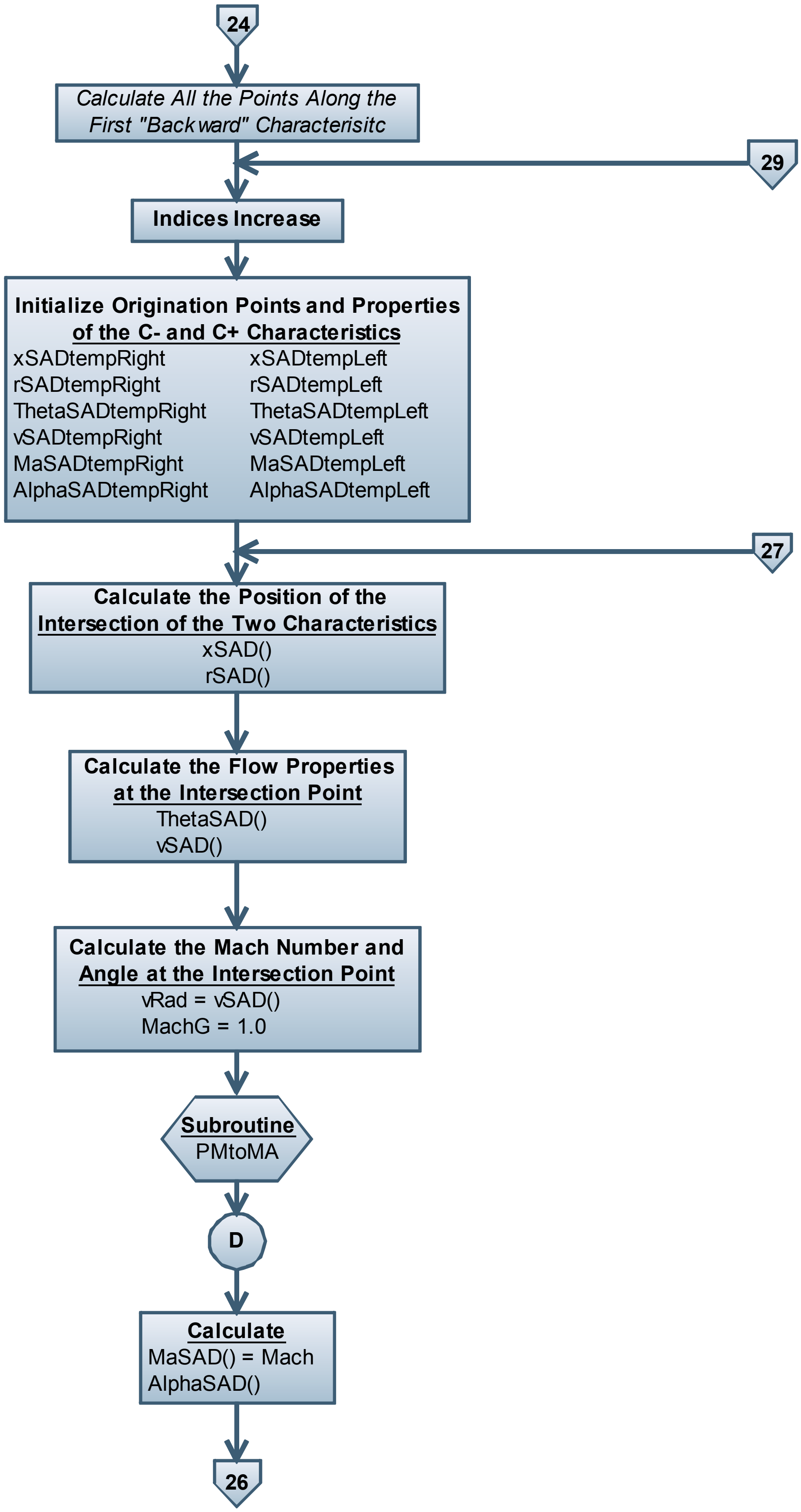


Figure 3.4.7i: *Subroutine Axisymmetric.m Flow Chart*

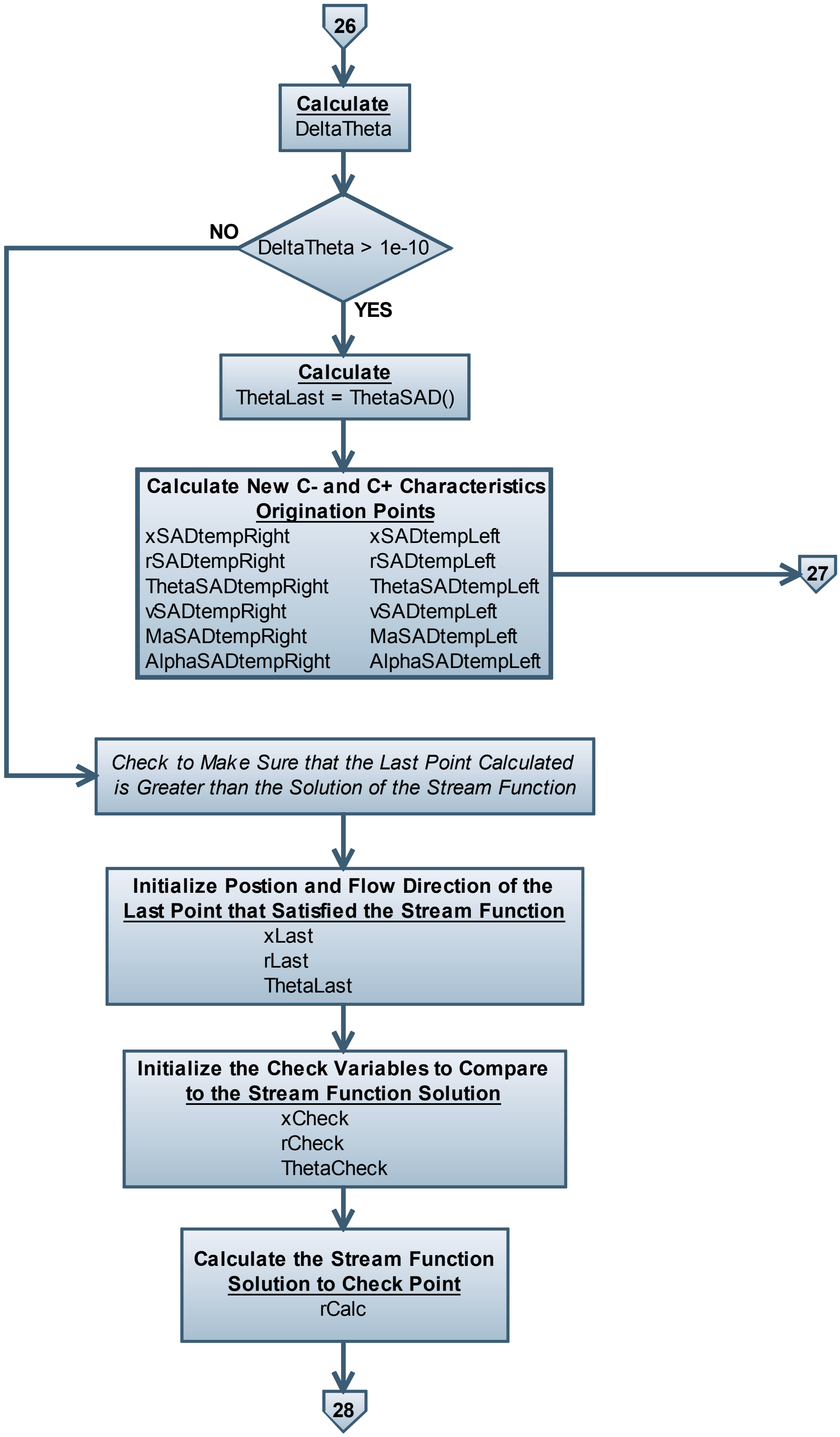


Figure 3.4.7j: *Subroutine Axisymmetric.m Flow Chart*

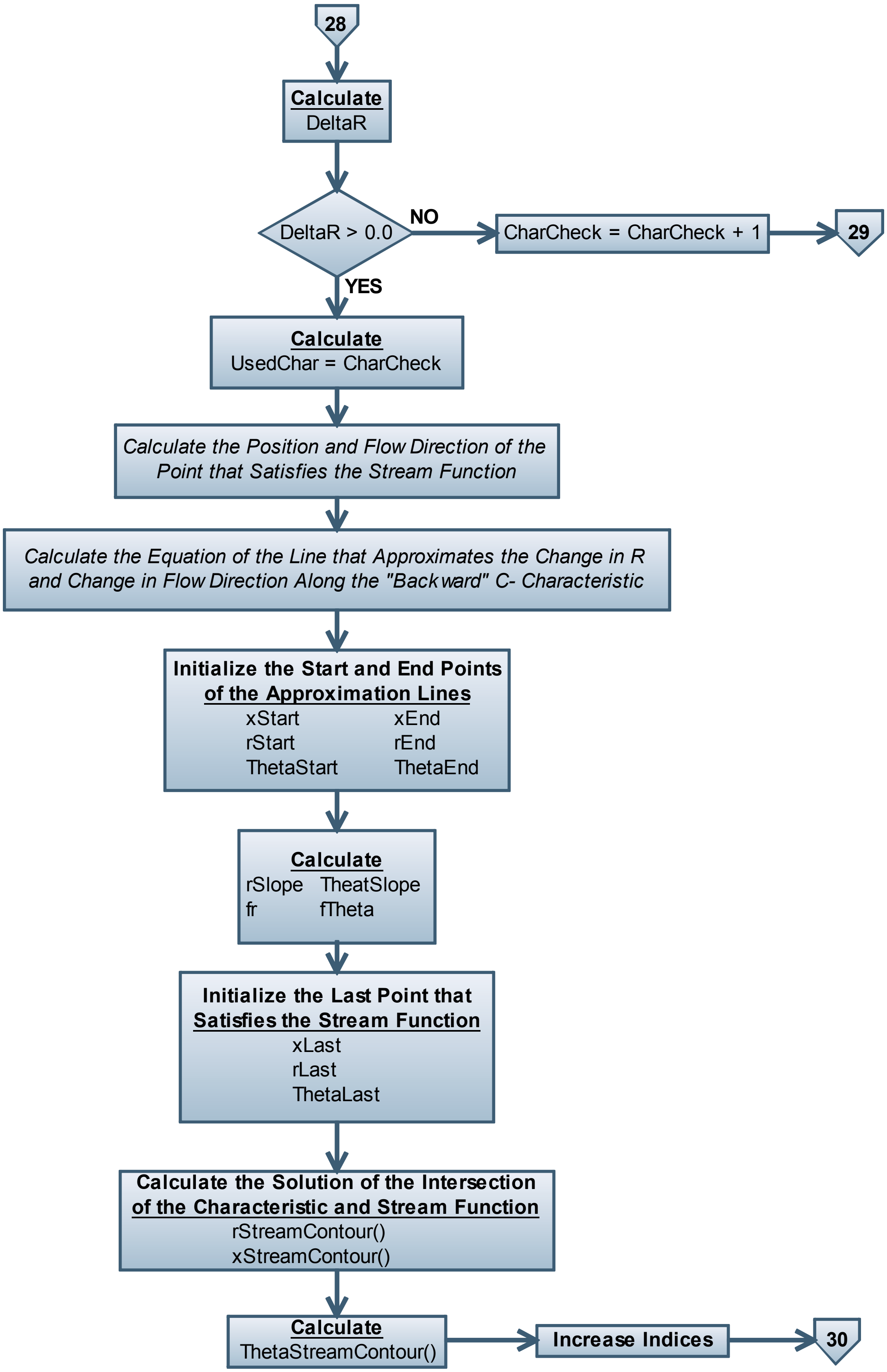


Figure 3.4.7k: *Subroutine Axisymmetric.m Flow Chart*

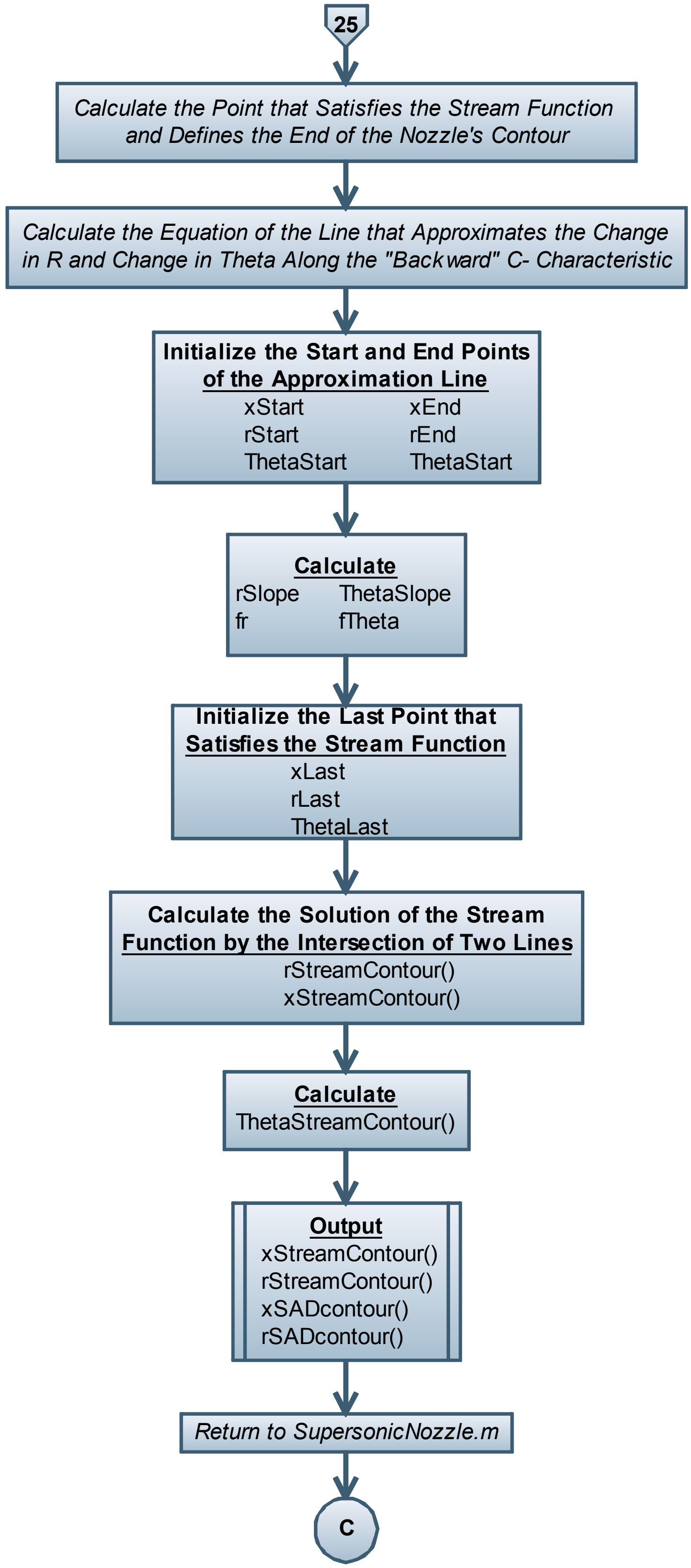


Figure 3.4.7l: *Subroutine Axisymmetric.m Flow Chart*

Subroutine PMtoMA

Subroutine PMtoMA is a root finding subroutine that uses the Bisection Method to calculate the Mach number associated with each flow field point's Prandtl-Meyer expansion angle. The program sets the lower limit of the range to a Mach number of 1 and the upper limit to a 100 times the desired exit Mach number. The program then calculates an initial guess Mach number by averaging the range's limits.

$$Ma_{guess} = \frac{Ma_{upper} + Ma_{lower}}{2} \qquad \textit{Eq. 3.62}$$

Next the subroutine calculates the Prandtl-Meyer expansion angle of the guess Mach number using equation *3.9* and compares it to the Prandtl-Meyer expansion angle of the point. If the Prandtl-Meyer expansion angle of the guess Mach number is greater than the point's Prandtl-Meyer expansion angle, the subroutine sets the upper limit of the range equal to the guess Mach number. If the Prandtl-Meyer expansion angle of the guess Mach number is less than the Prandtl-Meyer expansion angle of the point, the subroutine sets the lower limit of the range equal to the guess Mach number. If the difference between the Prandtl-Meyer expansion angle of the guess Mach number and the point's Prandtl-Meyer expansion angle is greater than abs($1e^{-10}$), the subroutine recalculates a new guess Mach number using equation *3.62* with the new range limits. If the difference in Prandtl-Meyer expansion angles is less than abs($1e^{-10}$), the subroutine returns the guess Mach number as the point's Mach number.

Figure 3.4.8 shows the calculation procedural flow chart for this subroutine. The source code and a copy of the flow chart are available to Appendix B.

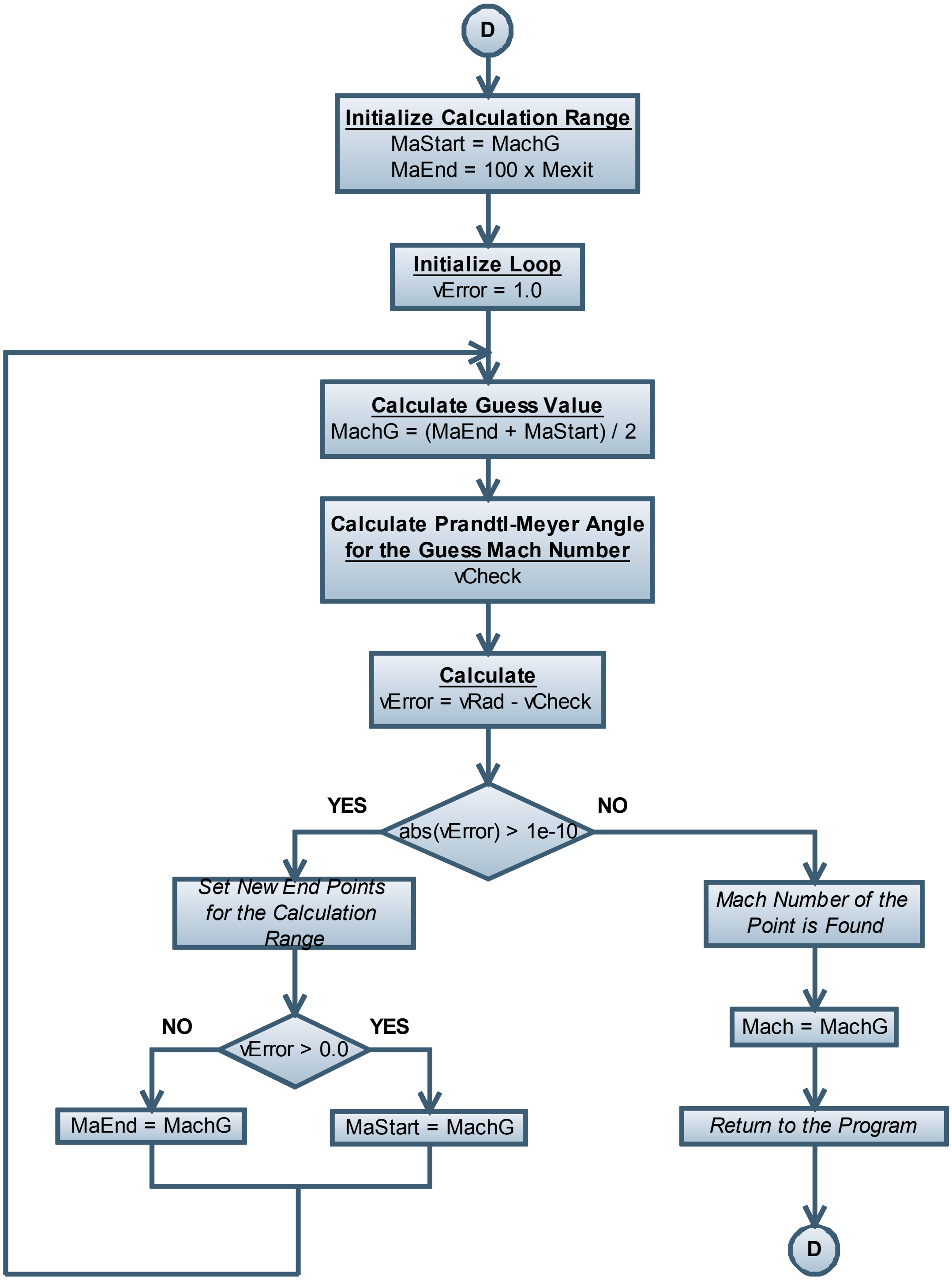


Figure 3.4.8: *Subroutine PMtoMA.m Flow Chart*

4.0 CFD Setup

For cost effective design, it is advantageous to validate the nozzles' contour in a Computational Fluid Dynamics (CFD) code such as FLUENT. This step in the design process can save money. It allows the designer to see if the flow reaches the desired exit velocity and if shocks develop in the flow without the need for materials or time for testing. Below is an outline of the steps taken to validate the nozzle contours developed from the theory above. For simulation convergence, continuity, x-velocity, y-velocity and energy changes were all required to be less than 0.001 which is FLUENT's default parameter.

To predict the fluid properties of the exhaust gas, the Complex Chemical Equilibrium Composition and Application program developed by NASA was used. This program has been developed and used by NASA for the past 40 years and is still used today to predict the exhaust properties of combustion processes.

4.1 Complex Chemical Equilibrium Composition and Application Program

The Complex Chemical Equilibrium Composition and Application (CEA) Program developed by NASA uses the minimization of Gibb's Free Energy to predict the composition of the exhaust products of a combustion system. In doing so, the properties of the exhaust fluid are predicted using mass averaging of the species produced by the combustion system.

The CEA program has multiple subroutines to choose from for different combustion systems. Since we are analyzing rocket nozzles, the *rocket* subroutine was chosen to predict the exhaust properties. Within the *rocket* subroutine, the *finite area* combustion chamber was utilized because the test chamber of the test apparatus is small with an interior radius of 1.25 inches. To complete the simulation, the pressure at the injector, chamber to throat area ratio, oxidizer and fuel chemical formulas and amounts with respect to the desired oxidizer to fuel ratio must all be entered. Using conditions from a previous single firing of the test apparatus, the CEA program was used to predict the ratio of specific heats, chamber pressure and temperature for the exhaust fluid. The results from the CEA program give information for three planes in the apparatus, at the injector, at the end of the combustion chamber and at the throat of the nozzle. The ratio of specific heats predicted at the throat is used as the input for the supersonic nozzle program discussed in Section 3.0. The chamber pressure and temperature are used as boundary conditions in the CFD simulations discussed in the next few subsections. Table 4.1.1 gives the inputted data used for the CEA simulation. All nozzles designed were assumed to have the same combustion system and working fluid.

Table 4.1.1: **CEA Program Inputs**			
Subroutine:	Rocket		
Combustion Chamber:	Finite Area		
Chamber to Throat Area Ratio:	44.44		
Initial Pressure:	360 psia		
Combustion Temperature:	3800 K (estimate)		
Reactants Found in the Thermodynamic Library:	N_2O (Nitrous Oxide)	*Amount:*	320 kg
Reactants with User-Provided Names and Properties:	$C_{224}H_{155}O_{27}N$ (Papi 94)	*Amount:*	12 kg
	$C_{667}H_{999}O_5$ (HTPB)	*Amount:*	88 kg

Figure 4.1.1 shows the results produced by the CEA program for the inputs given in Table 4.1.1.

```
Pin =   360.0 PSIA
Ac/At = 44.4400      Pinj/Pinf =  1.000056
CASE =

            REACTANT                    WT FRACTION      ENERGY      TEMP
                                         (SEE NOTE)    KJ/KG-MOL      K
OXIDANT     N2O                          1.0000000         0.000     0.000
FUEL        Papi                         0.1200000         0.000     0.000
FUEL        R45M                         0.8800000         0.000     0.000

O/F=    3.20000  %FUEL= 23.809524  R,EQ.RATIO= 2.649119  PHI,EQ.RATIO= 2.682408

                INJECTOR  COMB END  THROAT
Pinj/P            1.0000   1.0001   1.8306
P, BAR            24.821   24.818   13.559
T, K             1639.22  1639.20  1427.31
RHO, KG/CU M    3.8978 0 3.8974 0 2.4518 0
H, KJ/KG         0.00000 -0.03553  -359.00
U, KJ/KG         -636.80  -636.82  -912.00
G, KJ/KG        -16418.1 -16417.9 -14654.7
S, KJ/(KG)(K)    10.0158  10.0158  10.0158

M, (1/n)          21.403   21.403   21.460
(dLV/dLP)t      -1.00251 -1.00251 -1.00648
(dLV/dLT)p        1.0195   1.0195   1.0638
Cp, KJ/(KG)(K)    1.7015   1.7015   1.8557
GAMMAs            1.3069   1.3069   1.2983
SON VEL,M/SEC      912.3    912.2    847.3
MACH NUMBER        0.000    0.009    1.000

PERFORMANCE PARAMETERS

Ae/At                      63.236   1.0000
CSTAR, M/SEC               1194.7   1194.7
CF                         0.0071   0.7093
Ivac, M/SEC               75549.5   1500.0
Isp, M/SEC                    8.4    847.3
```

Figure 4.1.1: *CEA Program Results for Inputs in Table 4.1.1*

4.2 Annular Nozzle CFD Simulation

In order to run a simulation of the flow in supersonic annular nozzles, the nozzles must be built virtually so that a mesh can be generated in the fluid region. Using the ratio of specific heats of the exhaust gas predicted by the CEA program, the supersonic nozzle program described in Section 3.0 produces a set of points which define the nozzle's contour. These points are imported into Gambit. Gambit is a mesh generating program used to mesh the fluid domain of the simulation. It is important to note that the points generated by the supersonic nozzle program only yields points of the wall contour after the throat. Since the fluid experiences few losses in the convergent section of a supersonic nozzle, the user can design the convergent section of the nozzle given the known geometry of the combustion chamber. All points are connected to produce a 2D axisymmetric virtual geometry. Figure 4.2.1 below shows the typical geometry and boundary conditions used to simulate an annular nozzle.

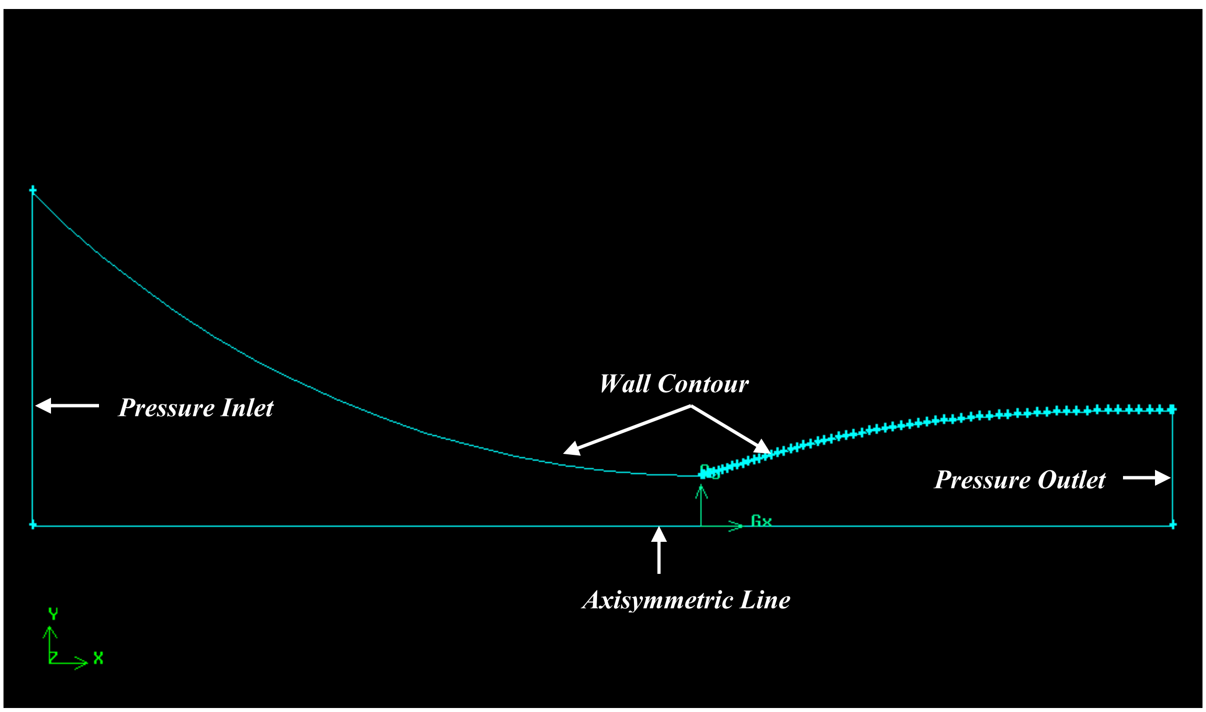


Figure 4.2.1: *Typical Annular Nozzle CFD Boundary Conditions*

Once the geometry of the nozzle has been virtually created, the fluid region can be meshed. Figure 4.2.2 a typical the meshed geometry of an annular nozzle. produced Table 4.2.1 gives the meshing inputs used for this particular mesh. There were no attempts to optimize the mesh because FLUENT converged in a reasonable amount of time. Table 4.2.1 also gives results for the quality of the mesh produced with the given conditions. These are typical mesh quality results for annular nozzles produced by the program discussed in Section 3.0.

Table 4.2.1	**Typical Gambit Meshing Inputs and Results for an Annular Nozzle**		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	36800	

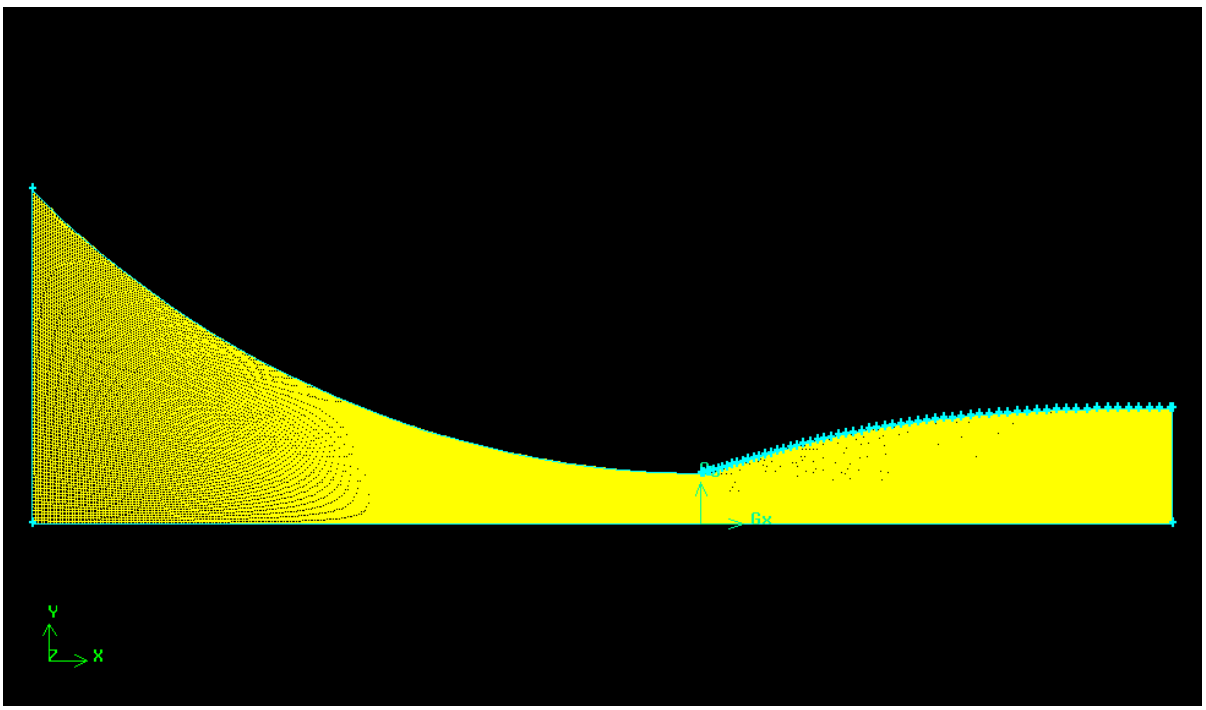


Figure 4.2.2: *Typical Annular Nozzle Mesh*

As a general rule of thumb, a mesh with an Equiangle Skew of less than 0.8 is considered a good mesh; however, there are times when this rule can be bent.

Now that the geometry has been meshed, it can be imported into FLUENT, the fluid flow simulation program. Once imported, the solver type, material and properties, operating conditions and boundary conditions must all be defined. In many cases, the time step, controlled by the Courant number, must be reduced or modulated throughout the simulation to facilitate convergence. Table 4.2.2 defines the conditions used in the simulations for annular nozzles.

Table 4.2.2	**FLUENT Input Conditions Used for Annular Nozzle Simulations**		
Solver:	Solver:	Coupled	
	Space:	Axisymmetric	
	Velocity Formation:	Absolute	
	Gradient Option:	Cell-based	
	Formulation:	Implicit	
	Time:	Steady	
	Porous Formulation:	Superficial Velocity	
	Energy Equation:	Checked	
	Viscous Model:	Inviscid Checked	
Material:	Name:	HTPB	
	Chemical Formulation:	N/A	
	Material Type:	Fluid	
	FLUENT Fluid Material:	HTPB	
	Properties:	Density:	Ideal Gas
		Cp:	1885.7 J/kg*K
		Molecular Weight:	21.403 kg/kmol
Operating Conditions:	Pressure:	Operating Pressure:	0 Pa
	Gravity:	Not Checked	
	Reference Pressure Location:	X(m):	0
		Y(m):	0
Pressure Inlet:	Gauge Total Pressure:	2481800 Pa	Constant
	Supersonic/Initial Gauge Pressure:	2481800 Pa	Constant
	Total Temperature:	1639.2 K	Constant
	Direction Specification Method:	Normal to Boundary	
Pressure Outlet:	Gauge Pressure:	101325 Pa	Constant
	Backflow Total Temperature:	300 K	Constant
	Backflow Direction Specification Method:	Normal to Boundary	
	Non-reflecting Boundary:	Not Checked	
	Target Mass-flow Rate:	Not Checked	
Solution Controls:	Discretization:	Second Order Upwind	
	Solver Parameter:	Courant Number:	1
Solution Initialization:	Compute From:	Pressure Inlet	
	Reference Frame:	Relative to Cell Zone	
	Initial Values:	Automatically Set by *Compute From*	

Since the nozzle contours were built on the assumption of inviscid, irrotational, isentropic flow, the CFD simulations need to reflect this. The inviscid assumption was satisfied by selecting the inviscid model for the simulations. The isentropic assumption, which implies irrotationality, was achieved by assigning the specific heat at constant pressure as a constant property of the working fluid.

To validate the code in Section 3.0, two quantities are checked once the simulations converge, entropy changes in the flowfield and area-weighted Mach number at the exit of the nozzle. The entropy changes were checked by looking at the entropy change contours of the nozzle. These contours should be constant or minimal to validate the isentropic flow assumption. A shock is non-isentropic; therefore, entropy plots will show shocks if they exist in the flowfield. The simulation results are discussed in Section 5.0, Results and Discussion. The complete collection of simulation results is available in Appendix D.

4.3 Aerospike Nozzle CFD Simulations

Aerospike nozzle simulations are fundamentally different than the simulations for annular and conical nozzles. They require the use of a non-reacting mixing model for two fluids in the domain, atmospheric air and the exhaust produced by the combustion process. As a control group, simulations using air as the working fluid were also simulated. Before we devolve into that, we will first discuss the boundary conditions and mesh characteristics for the aerospike geometries. This section will cover the simulation conditions of the external and Internal-External aerospike nozzles as well as their truncated versions.

The exhaust fluid properties predicted by the CEA program were used with the program discussed in Section 3.0 produced the contour of the nozzles after the throat as well as the expansion point (external aerospike) or expansion contour (Internal-External aerospike) of the cowl. Nozzle designs prior to the throat were user-defined and as stated before exhibit minimal losses in the flow as long as the convergent section of the nozzle is sufficiently smooth.

External Aerospike

The contour points solved for by the program discussed in Section 3.0 and user-defined convergent section of the nozzle were imported into Gambit to virtually create the geometry for simulation. Figure 4.3.1 and 4.3.2 show the typical virtual geometry and boundary conditions applied for 100% and 20% length external aerospike nozzles. It is important to note that the line segments defining the external portion of the nozzle of both geometries were merged into a virtual curve. This was necessary to facilitate meshing of the fluid domain.

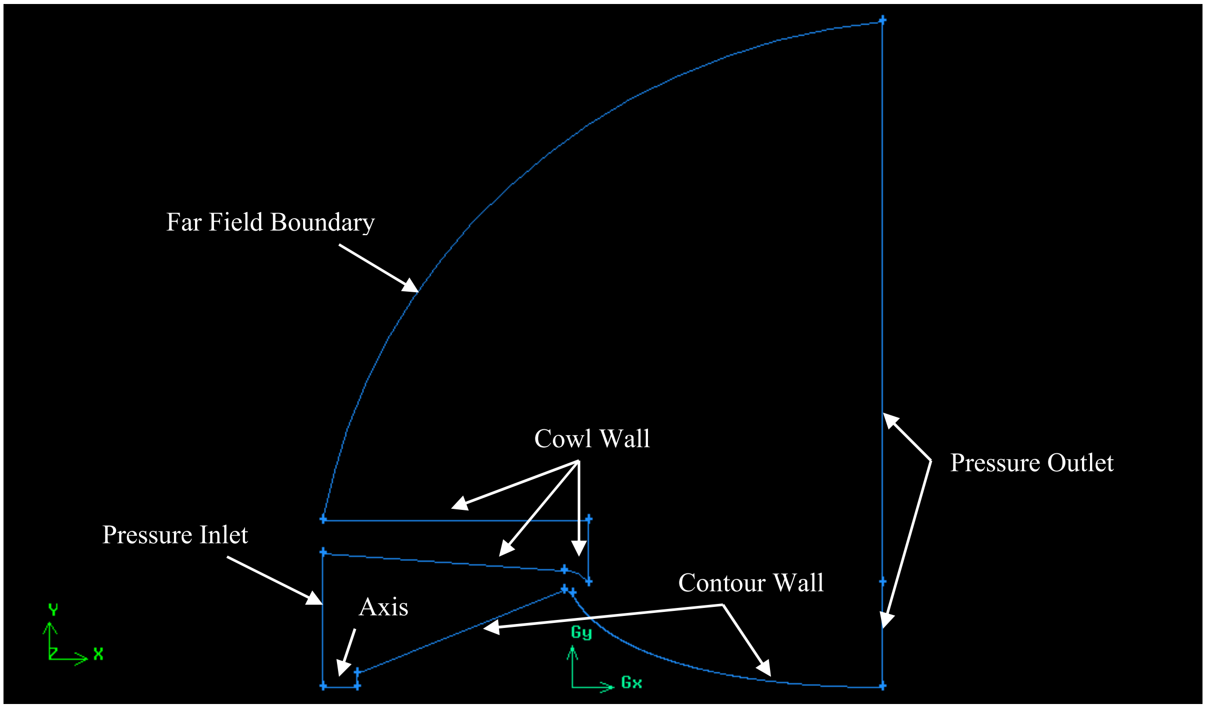


Figure 4.3.1: *Generic Geometry of a 100% Length External Aerospike Nozzle*

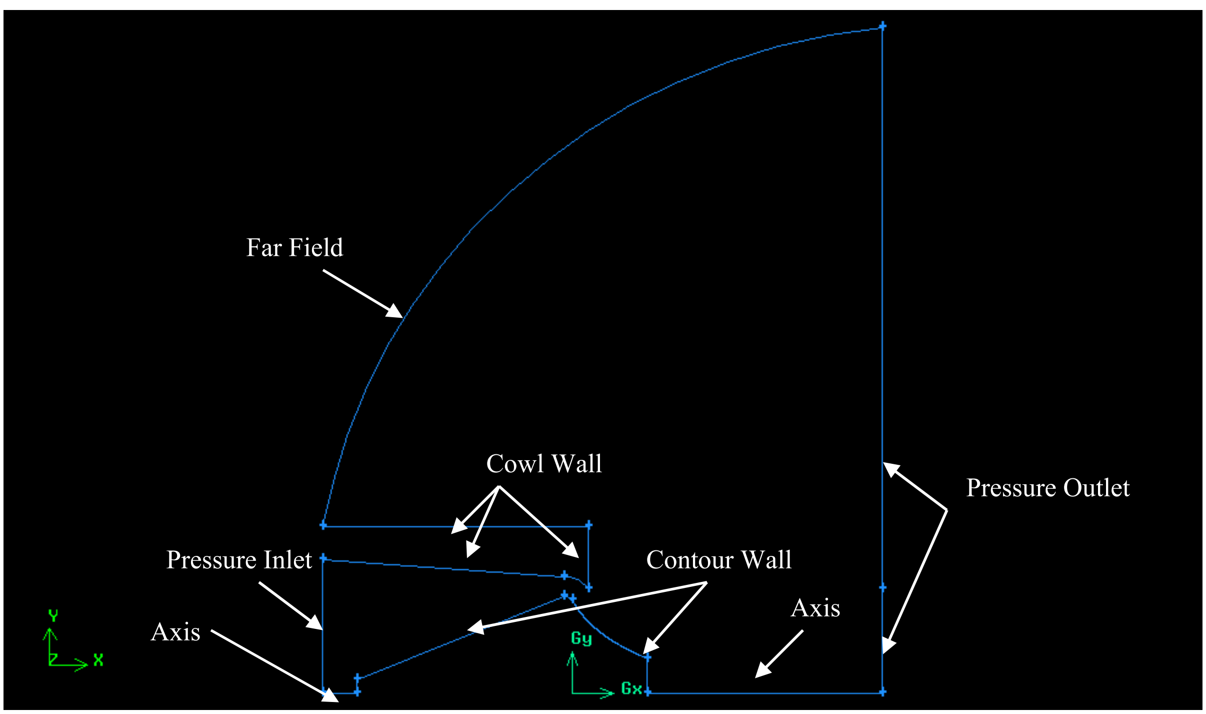


Figure 4.3.2: *Generic Geometry of a 20% Length External Aerospike Nozzle*

With the geometries of the nozzles virtually created, they can now be meshed in preparation for simulating using FLUENT. Tables 4.3.1 and 4.3.2 give the mesh input parameters used and quality results for the 100% and 20% length geometries, respectively. Figures 4.3.3 and 4.3.4 show the meshed geometries of these nozzles, respectively.

Table 4.3.1	**Gambit Meshing Inputs and Results for 100% Length Geometry in Figure 4.3.3**		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.40
		Number of Cells > 0.97	0
	Total Number of Cells	5958	

Table 4.3.2	**Gambit Meshing Inputs and Results for 20% Length Geometry in Figure 4.3.4**		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.45
		Number of Cells > 0.97	0
	Total Number of Cells	6042	

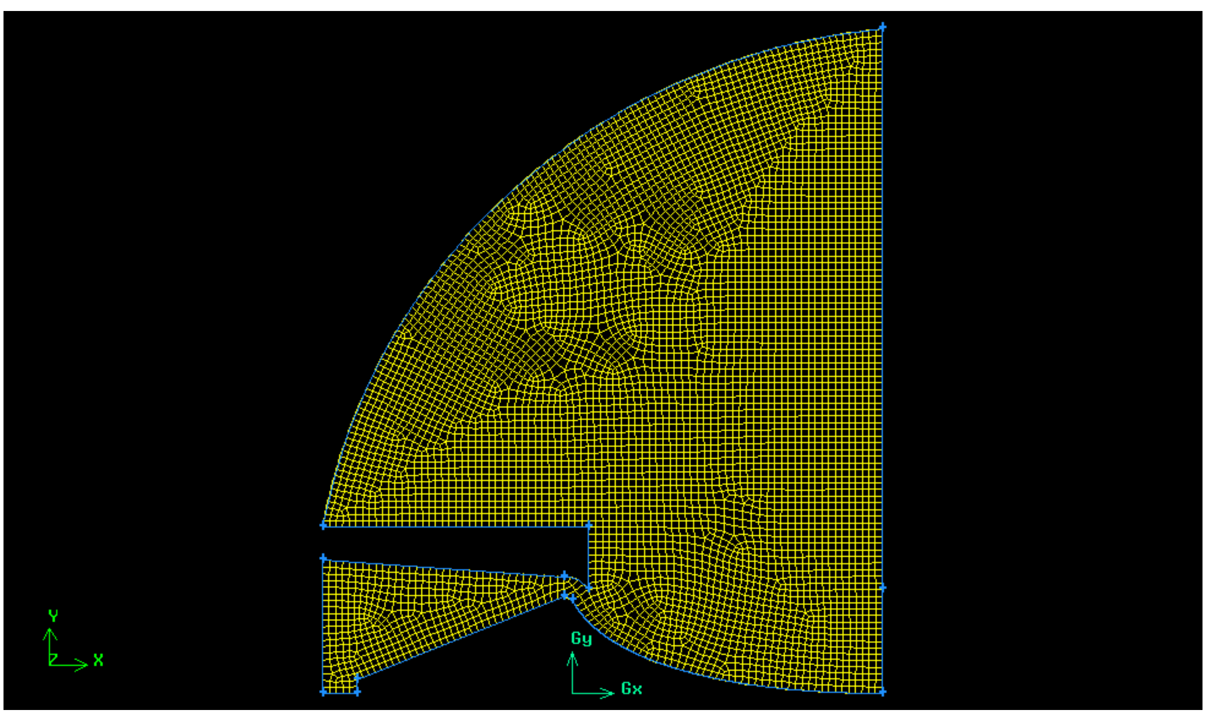


Figure 4.3.3: *Generic Meshed Geometry of a 100% Length External Aerospike Nozzle*

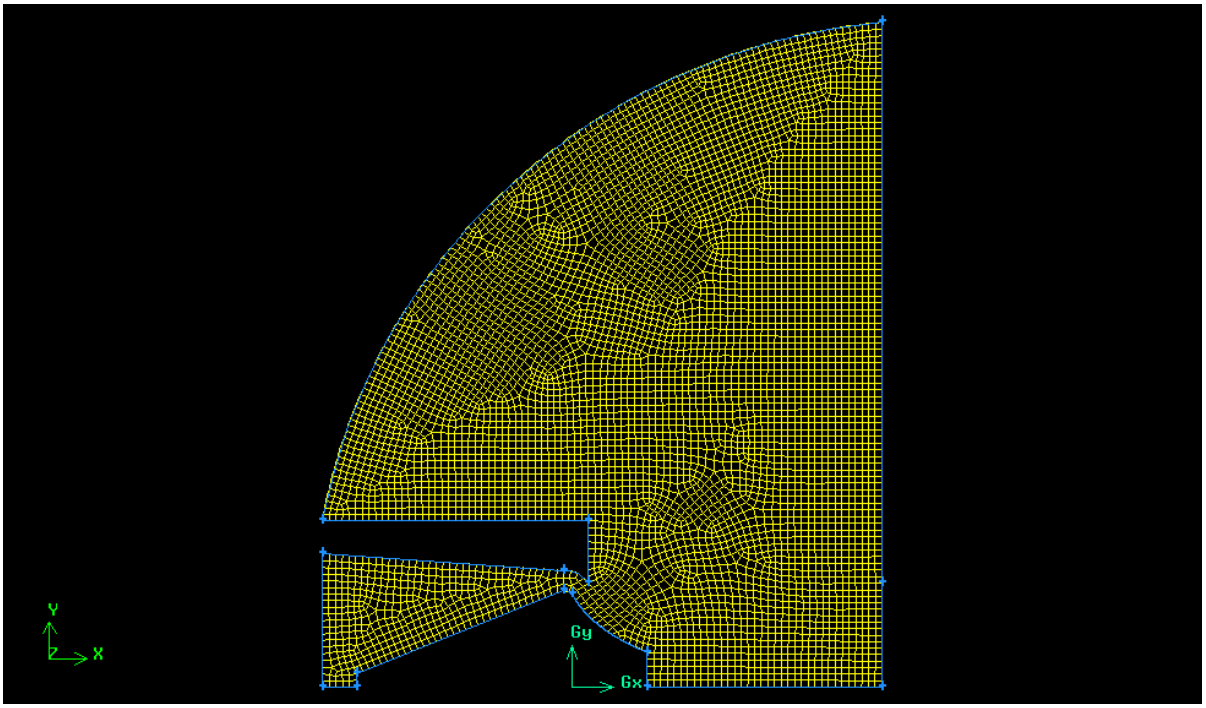


Figure 4.3.4: *Generic Meshed Geometry of a 20% Length External Aerospike Nozzle*

With the geometries meshed, they can be imported into FLUENT. Once imported, the solver, material and properties, operating condition, mixing model and boundary conditions were defined. As with the annular nozzle simulations, in many cases the time step, controlled by the Courant number, had to be reduced or modulated throughout the simulation to facilitate convergence. Table 4.3.3 defines the input conditions used for the simulations of all external aerospike nozzles.

Table 4.3.3	FLUENT Input Conditions Used for Aerospike Nozzle Simulations		
Solver:	Solver:	Coupled	
	Space:	Axisymmetric	
	Velocity Formation:	Absolute	
	Gradient Option:	Cell-based	
	Formulation:	Implicit	
	Time:	Steady	
	Porous Formulation:	Superficial Velocity	
	Energy Equation:	Checked	
	Viscous Model:	Inviscid Checked	
	Species:	Transport & Reaction:	Species Transport
			Volumetric
			Mixture-Template
			Laminar Finite-Rate
Material:	Name:	Mixture-Template	
	Chemical Formulation:	N/A	
	Material Type:	Mixture	
	FLUENT Fluid Material:	Mixture-Template	
	Properties:	Density:	Ideal Gas
		Cp:	Mixing Law
		Mechanism:	Reaction-mech
		Reaction:	Finite Rate
		Mixture Species:	names
Operating Conditions:	Pressure:	Operating Pressure:	0 Pa
	Gravity:	Not Checked	
	Reference Pressure Location:	X(m):	0
		Y(m):	0
Pressure Inlet:	Gauge Total Pressure:	2481800 Pa	Constant
	Supersonic/Initial Gauge Pressure:	2481800 Pa	Constant
	Total Temperature:	1639.2 K	Constant
	Direction Specification Method:	Normal to Boundary	
	Species Mass Fraction:	H_2O:	0.0048988
		O_2:	0
		NH_3:	6.365719e^{-5}
		HCN:	0.00032836
		H_2:	0.0238565
		H^+:	4.709419e^{-7}
		CO_2:	0.005469665
		CO:	0.4800784
		CH_4:	0.0004572298
		N_2:	Auto. Calculated
Pressure Outlet:	Gauge Pressure:	101325 Pa	Constant
	Backflow Total Temperature:	300 K	Constant
	Backflow Direction Specification Method:	Normal to Boundary	
	Non-reflecting Boundary:	Not Checked	
	Target Mass-flow Rate:	Not Checked	
	Species Mass Fraction:	All set to 0	
Far Field:	Gauge Pressure:	101325 Pa	Constant
	Mach Number:	0.6	Constant
	Temperature:	300 K	Constant
	Axial-Component of Flow Direction:	1	Constant
	Radial-Component of Flow Direction:	0	
	Species Mass Fraction:	H_2O:	0.016
		O_2:	0.224
		N_2:	Auto. Calculated
		All others:	0
Solution Controls:	Discretization:	First Order Upwind	
	Solver Parameter:	Courant Number:	1
Solution Initialization:	Compute From:	Pressure Inlet	
	Reference Frame:	Relative to Cell Zone	
	Initial Values:	Automatically Set by *Compute From*	

Since the nozzle contours were built on the assumption of inviscid, irrotational, isentropic flow, the CFD simulations need to reflect this. The inviscid assumption was satisfied by selecting the inviscid model for the simulations. The isentropic assumption, which implies irrotationality, was achieved by defining the density of the fluid according to the ideal gas law and allowing FLUENT to calculate the specific heat at constant pressure according to the mixing law for the working fluid.

To validate the program detailed in Section 3.0, two quantities were checked once the simulations converged, entropy changes in the flowfield and area-weighted Mach number at the exit of the nozzle. For the 100% length external aerospike nozzles, an outlet boundary is created that had one endpoint at the end of the contour and the other with the x-coordinate of the end of the contour and an r-coordinate equal to the r-coordinate of the nozzle's expansion point. For the 20% length external aerospike nozzles, the outlet boundary of the nozzle was created with one endpoint having the coordinates of the last point on the 100% length contour and the other endpoint having the x-coordinate of the last point on the 100% length contour and an r-coordinate equal to the r-coordinate of the expansion point. These outlet boundaries were created to allow FLUENT to calculate an area-weighted exit Mach number that would not be influenced by the excess outlet boundary area needed to simulate external flow in FLUENT. These outlet boundaries can be seen in Figures 4.3.1 and 4.3.2. The entropy was checked by looking at the entropy change contours of the nozzle's flowfield. These contours should be constant or exhibit minimal changes to validate the isentropic flow assumption. The nozzle's exit Mach number is checked by the Mach number contours of the simulation as well as having FLUENT calculate the area-weighted Mach number at the exit plane of the nozzle. A discussion of the simulation results is available in Section 5.0, Results and Discussion. All simulation results are available in Appendix D.

Internal-External Aerospike Nozzle

IE aerospike simulations share many of the same setup requirements as the external aerospike. Figures 4.3.5 and 4.3.6 show the virtual geometries of typical 100% and 20% length IE aerospike nozzles created in Gambit. As with the external aerospike nozzles, the external portion of the nozzle and expansion cowl contour utilizes a virtual curve to facilitate a mesh of the fluid domain. The supersonic regions of these nozzles were created using the program detailed in Section 3.0 with the converging section design by the user.

Tables 4.3.4 and 4.3.5 give the inputs used and quality results of the mesh created by Gambit. Figures 4.3.7 and 4.3.8 show the typical mesh for 100% and 20% length IE aerospike nozzles. Once meshed and boundary conditions applied, the geometries are imported into FLUENT.

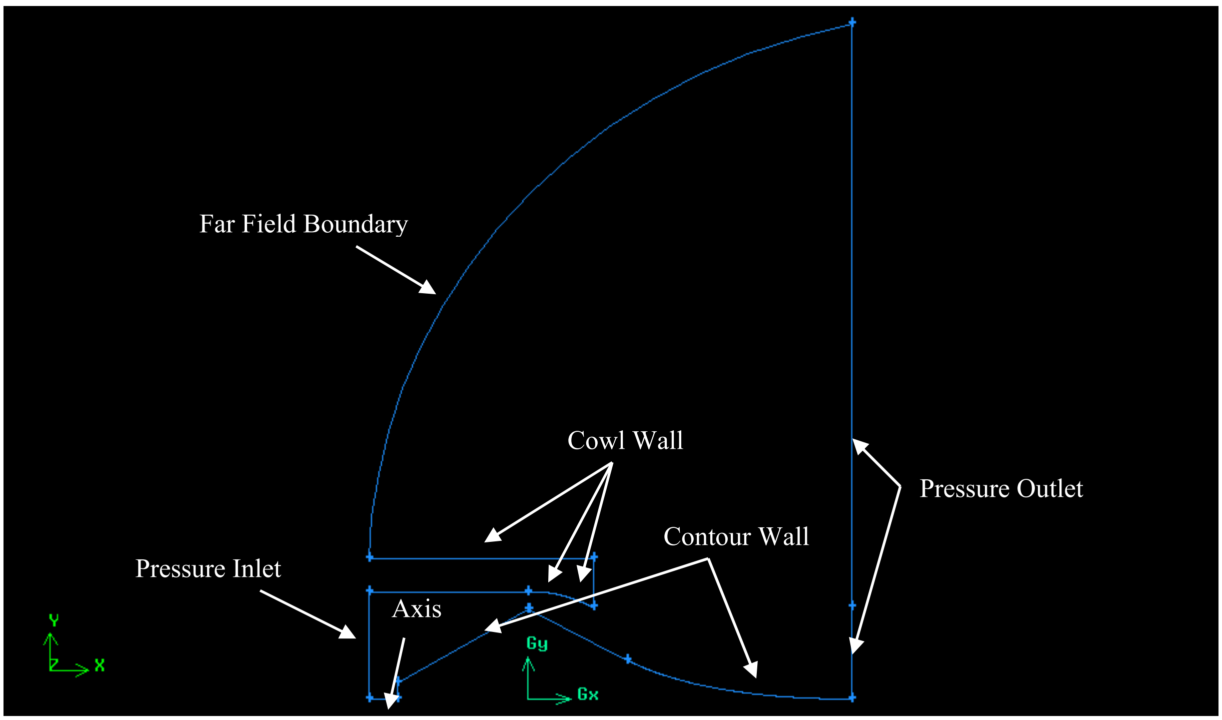


Figure 4.3.5: *Generic Geometry of a 100% Length IE Aerospike Nozzle*

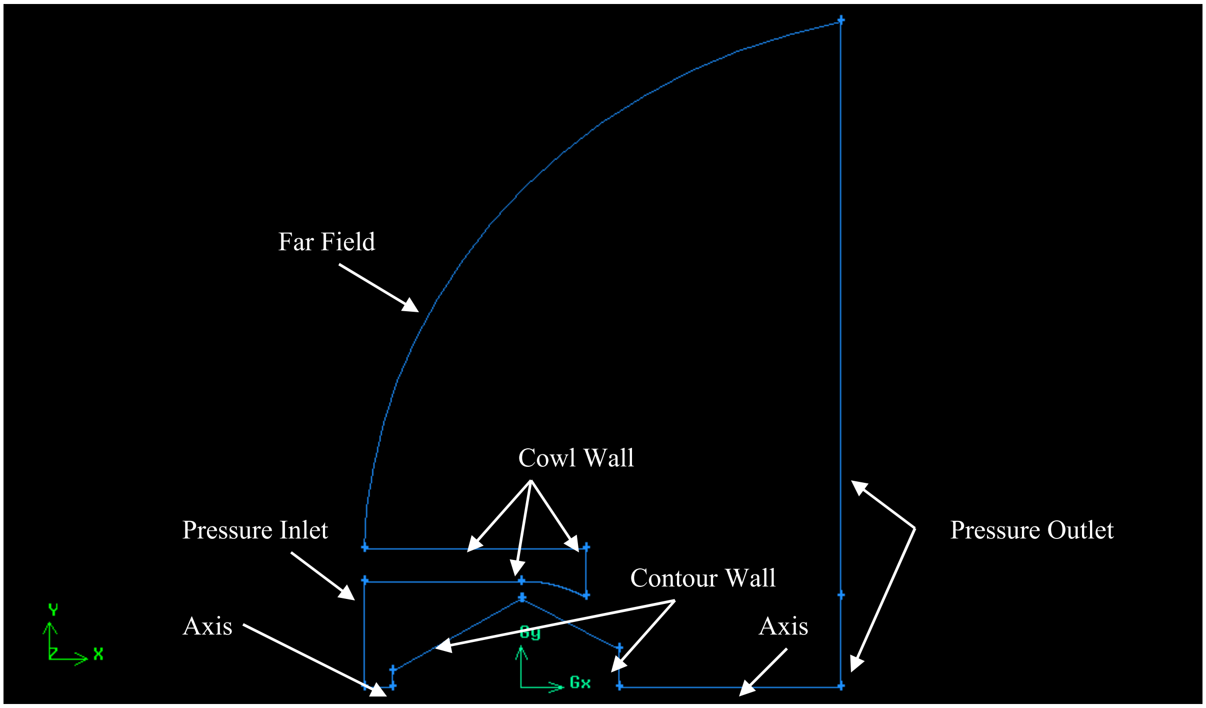


Figure 4.3.6: *Generic Geometry of a 20% Length IE Aerospike Nozzle*

Table 4.3.4	**Gambit Meshing Inputs and Results for 100% Length Geometry in Figure 4.3.8**		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.46
		Number of Cells > 0.97	0
	Total Number of Cells	7668	

Table 4.3.5	**Gambit Meshing Inputs and Results for 20% Length Geometry in Figure 4.3.9**		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.46
		Number of Cells > 0.97	0
	Total Number of Cells	7664	

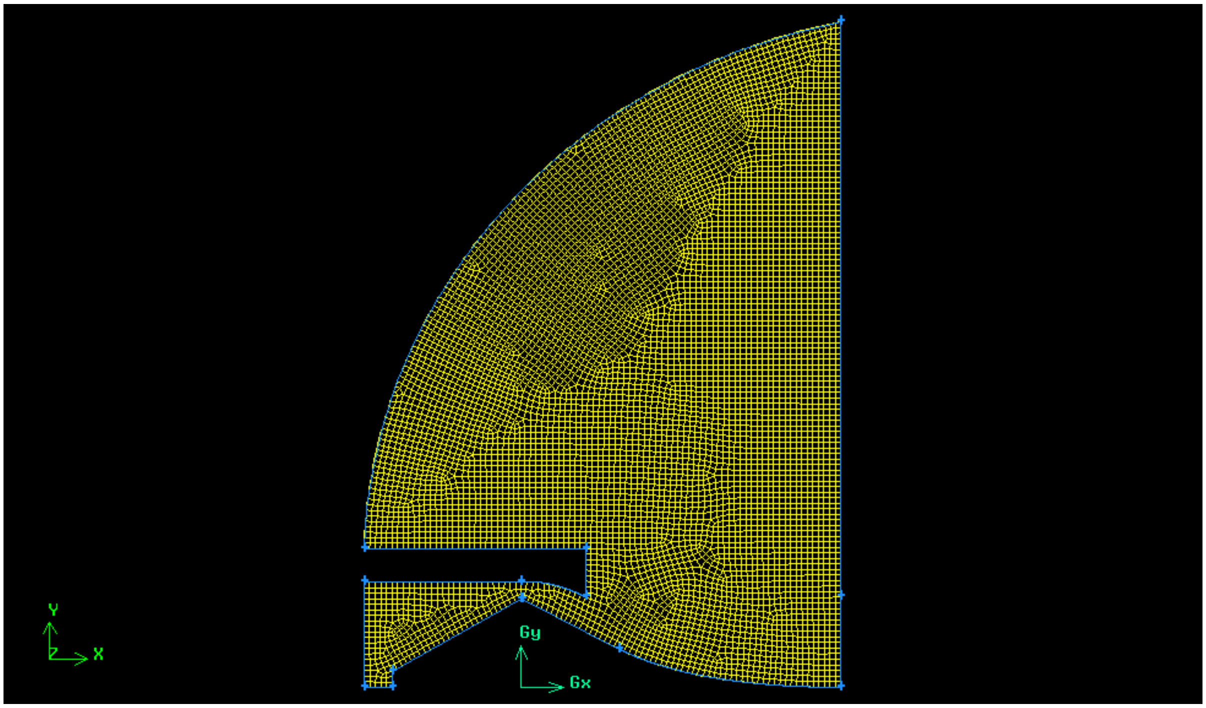


Figure 4.3.7: *Generic Meshed Geometry of a 100% Length IE Aerospike Nozzle*

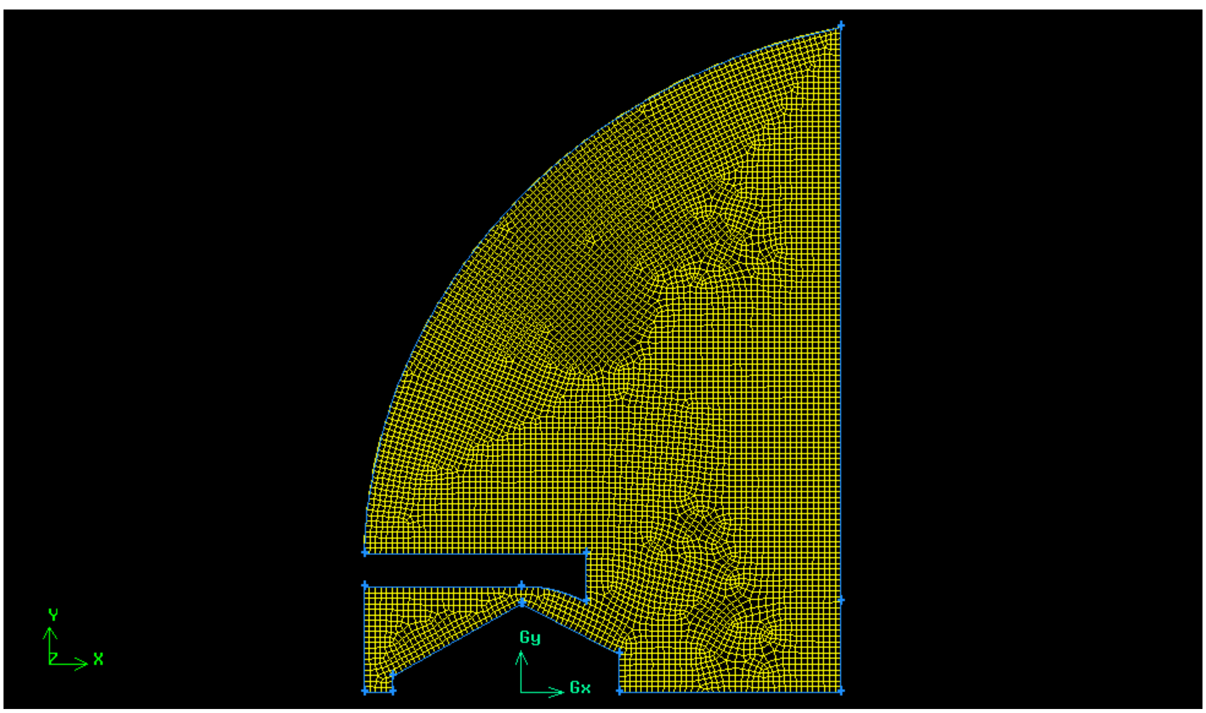


Figure 4.3.8: *Generic Meshed Geometry of a 20% Length IE Aerospike Nozzle*

Since the only difference between external aerospike nozzles and IE aerospike nozzles is the geometry, FLUENT is setup identically for both. Table 4.3.3 above gives all the inputs used for setting up all simulations for IE aerospike nozzles.

After the FLUENT simulations converge, two quantities are checked, entropy changes in the flowfield and area-weighted exit Mach number. As with the external aerospike nozzles, the entropy contour plots are used to validate the isentropic and irrotationality assumptions. The desired exit Mach number is checked with the Mach contour plots and the FLUENT calculated area-weighted Mach number on the nozzle's exit plane. The simulation results are discussed in Section 5.0, Results and Discussions with all simulation results available in Appendix D.

4.4 Conical Nozzle CFD Simulation

For completeness and basis for comparison, conical nozzles have been simulated. The conical nozzles investigated have the same entrance region as the annular and aerospike nozzles to eliminate the potential effects the convergent section may have on its flowfield and performance. Isentropic exit to throat area ratios and trigonometry were used to set the end wall contour points of the conical nozzle. Conical nozzles with 8, 10 and 12 degree half angles were simulated. The conical nozzles were assumed to have the same combustion system as the annular and aerospike nozzles. Therefore, they have the same working fluid. Figure 4.4.1 shows the virtual geometry and boundary conditions of a typical conical nozzle built in Gambit.

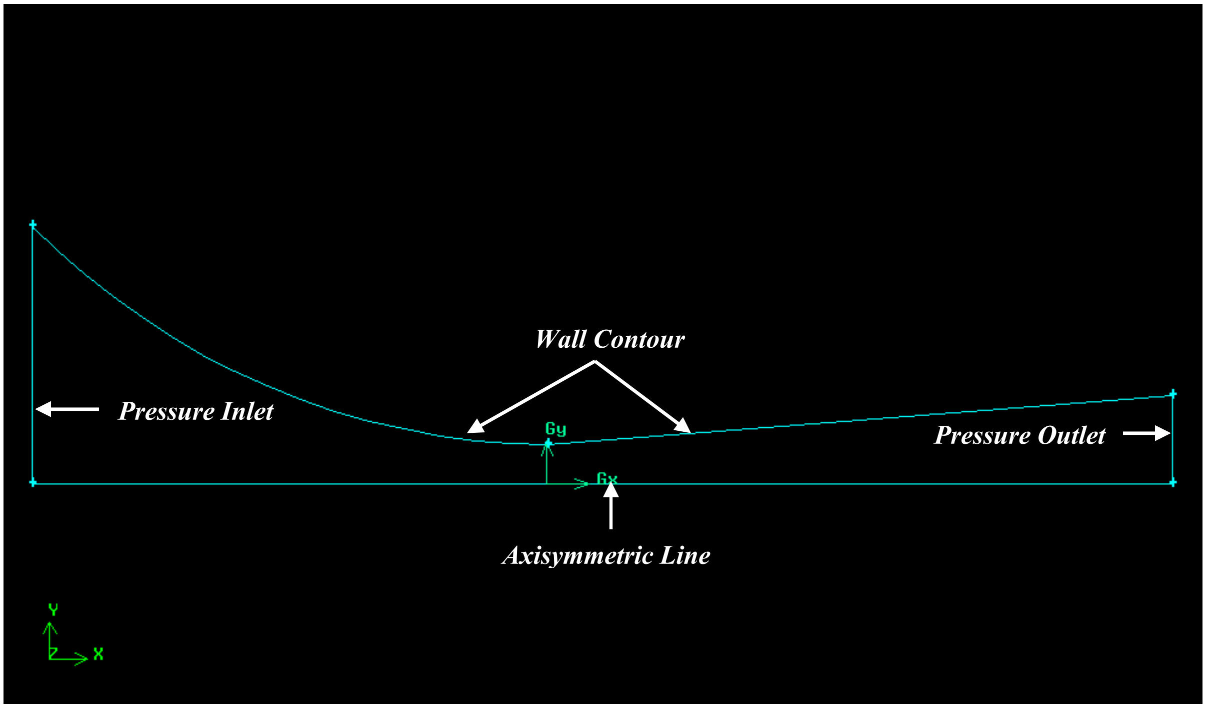


Figure 4.4.1: *Typical Conical Nozzle CFD Boundary Conditions*

With the geometry created virtually, it was meshed. Table 4.4.1 shows the input conditions and quality results of a typical mesh for a conical nozzle. Figure 4.4.2 shows a typical mesh.

Table 4.4.1	**Gambit Meshing Inputs and Results for Geometry a Typical Conical Nozzle**		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	46700	

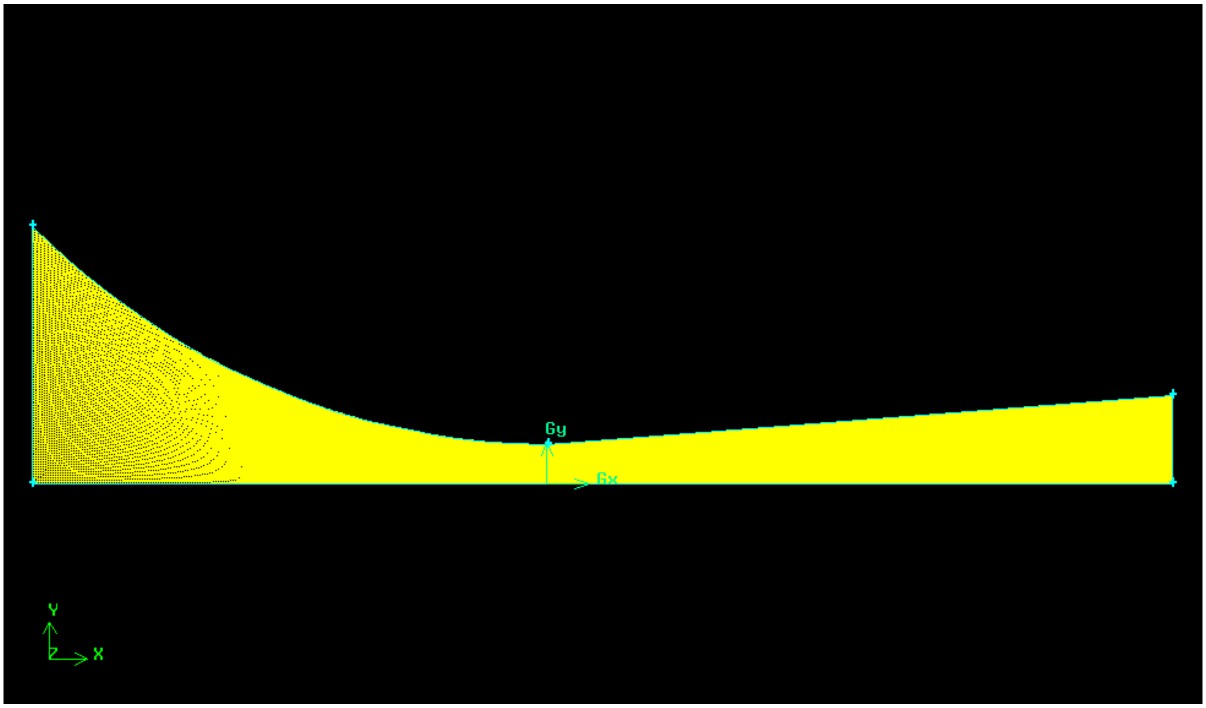


Figure 4.4.2: *Typical Conical Nozzle Mesh*

With the mesh complete, it was imported into FLUENT to simulate the flowfield. Since we are working with the same combusting system as annular nozzles, Table 4.2.2 also shows the solver setup for conical nozzles. Once convergence was reached, the entropy change and Mach contour plots were created to verify their respective design points. A discussion of the simulation results is available in Section 5.0, Results and Discussions, with all simulation results available in Appendix D.

5.0 Results and Discussion

This section discusses the checks performed to verify the accuracy of the code developed in Section 3.0. As mentioned in Sections 3.0 and 4.0, a combination of theoretical and CFD simulations were employed to verify the accuracy of the code for all the rocket nozzle configurations. This section highlights the general trends in each of the nozzles for the various checks performed. The complete set of accuracy check results can be seen in Appendices C and D.

5.1 Theoretical Accuracy of Computer Code

The first check of accuracy for the program was comparing the desired exit Mach number with the exit Mach number calculated by the program. Table 5.1.1 below shows the percent difference between the desired and computer calculated exit Mach numbers. Table 5.1.1 also shows how the code becomes more accurate as a smaller change in Prandtl-Meyer expansion angle is used during calculations.

Since the equations were based on isentropic flow theory, the accuracy of the code was also checked by calculating the exit to throat area ratio using equation *5.1* substituting in the user-defined ratio of specific heats and computer calculated exit Mach number. This yields the theoretical area ratio for the Mach number actually calculated by the program.

$$\frac{A_{exit}}{A_{throat}} = \sqrt{\frac{1}{Ma_{exit}^2} \cdot \left[\frac{2}{\gamma+1} \cdot \left(1 + \frac{\gamma-1}{2} \cdot Ma_{exit}^2\right)\right]^{\frac{\gamma+1}{\gamma-1}}} \qquad \textit{Eq. 5.1}$$

The theoretical and computer calculated isentropic area ratios for the desired exit Mach number were also compared for a user-defined ratio of specific heats in Table 5.1.1.

Table 5.1.1	$\gamma = 1.4$ Ma = 3.0 $\beta = 1.0 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	5.6588	4.2346	33.63%	3.2489	8.30%	5.3635	5.51%
IE Aerospike	5.2253	4.2346	23.40%	3.1278	4.26%	4.7820	50.55%
Aerospike	4.9828	4.2346	17.67%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	5.5827	4.2346	31.84%	3.2455	8.18%	5.3463	4.42%
IE Aerospike	4.7699	4.2346	12.64%	3.0509	1.70%	4.4448	46.61%
Aerospike	4.5875	4.2346	8.33%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	4.4882	4.2346	5.99%	3.0289	0.96%	4.3527	3.11%
IE Aerospike	4.5288	4.2346	6.95%	3.0059	0.20%	4.2584	44.52%
Aerospike	4.3716	4.2346	3.24%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	4.4938	4.2346	6.12%	3.0323	1.08%	4.3668	2.91%
IE Aerospike	4.4528	4.2346	5.15%	3.0059	0.20%	4.2584	41.83%
Aerospike	4.3025	4.2346	1.60%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	4.3940	4.2346	3.76%	3.0093	0.31%	4.2722	2.85%
IE Aerospike	4.4154	4.2346	4.27%	3.0059	0.20%	4.2584	40.50%
Aerospike	4.2684	4.2346	0.80%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	4.3815	4.2346	3.47%	3.0067	0.22%	4.2617	2.81%
IE Aerospike	4.3932	4.2346	3.75%	3.0029	0.10%	4.2463	40.11%
Aerospike	4.2481	4.2346	0.32%	3.0000	0.0%	-	-

Appendix C contains extensive tables that illustrate the accuracy of the code with different Mach numbers, ratios of specific heats, γ, and radii of the entrance region, β, while varying the step interval in Prandtl-Meyer expansion angle. A general trend of increasing accuracy in Mach number and area ratio was observed with decreasing step size of Prandtl-Meyer expansion angle for all desired exit Mach numbers, β's and ratios of specific heats for all nozzles.

Figures 5.1.1 and 5.1.2 show the effects the incremental Prandtl-Meyer expansion angle has on the % error in the area ratio and calculated exit Mach number for the annular nozzles designed by the program detailed in Section 3.0, respectively.

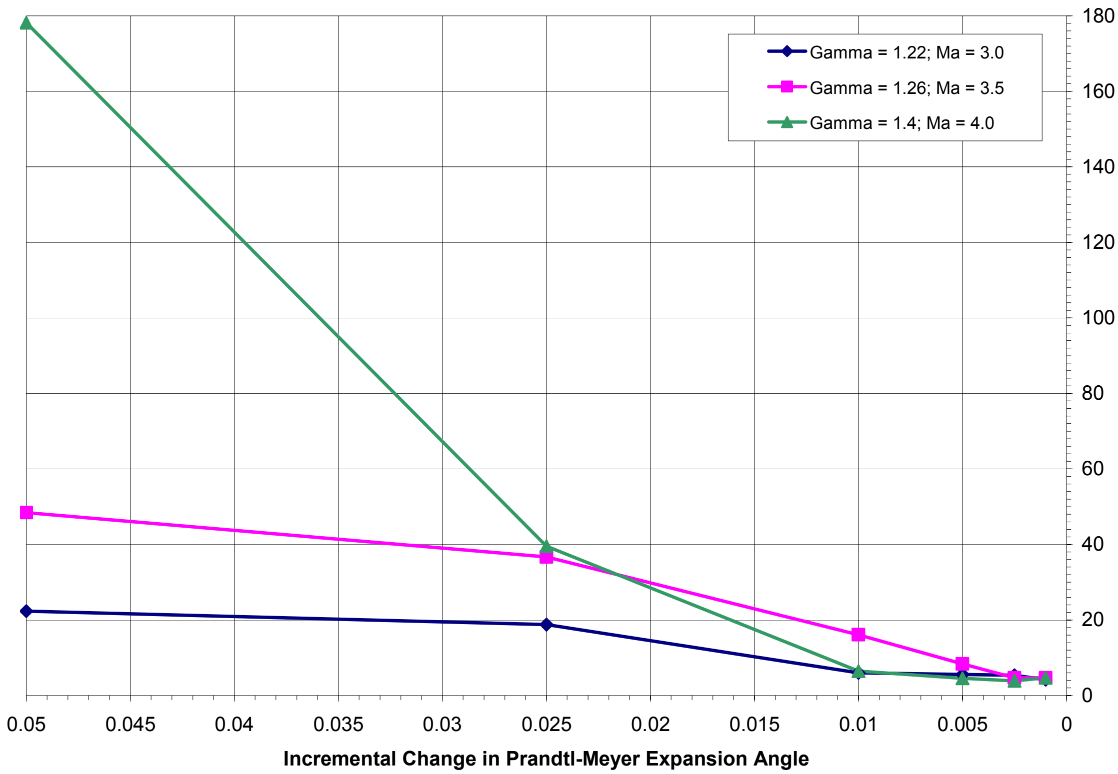


Figure 5.1.1: *% Error in Area Ratio vs. Incremental Prandtl-Meyer Expansion Angle for Annular Nozzle Calculations*

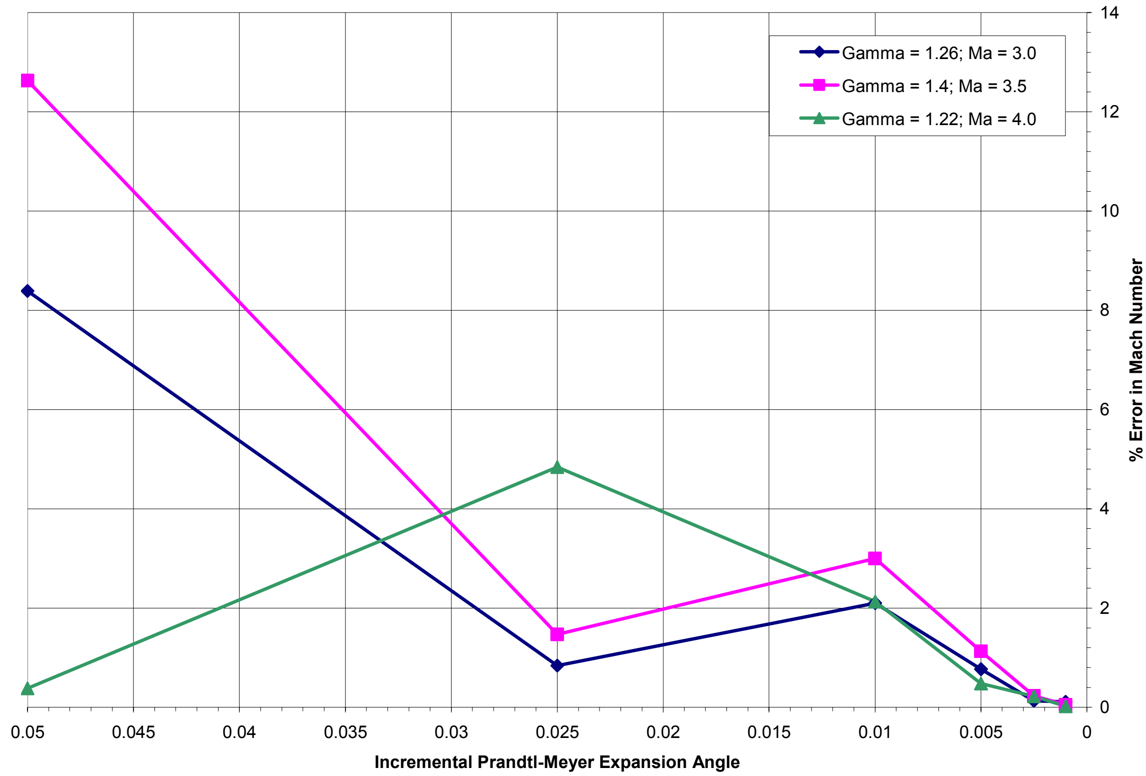


Figure 5.1.2: *% Error in Mach Number vs. Incremental Prandtl-Meyer Expansion Angle for Annular Nozzle Calculations*

The peaks and valleys in Figures 5.1.1 and 5.1.2 are due to the C_- and C_+ characteristics' interactions in the flowfield. As the number of characteristics increase in the calculation, the computational flow accelerates differently resulting in the peaks and valleys seen in the figures.

The same trends in % error in area ratio and Mach number with respect to the Prandtl-Meyer expansion angle increment can be seen in Figures 5.1.3 through 5.1.5 for the IE aerospike and external aerospike nozzles. It is important to note that since the last characteristic in the external aerospike nozzle calculations are set to the correct Prandtl-Meyer expansion angle for the user-defined desired exit Mach number, the Mach number calculated by the program will always be equal to the desired exit Mach number and exhibit zero error.

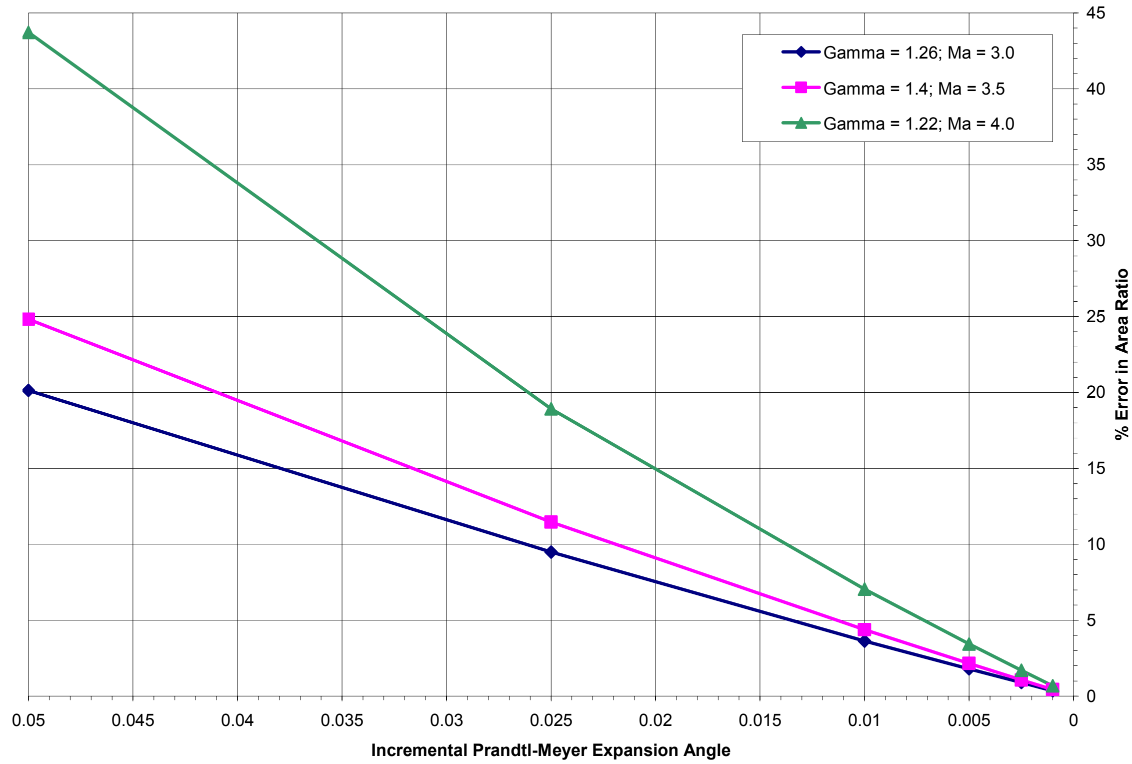


Figure 5.1.3: *% Error in Area Ratio vs. Incremental Prandtl-Meyer Expansion Angle for External Aerospike Nozzle Calculations*

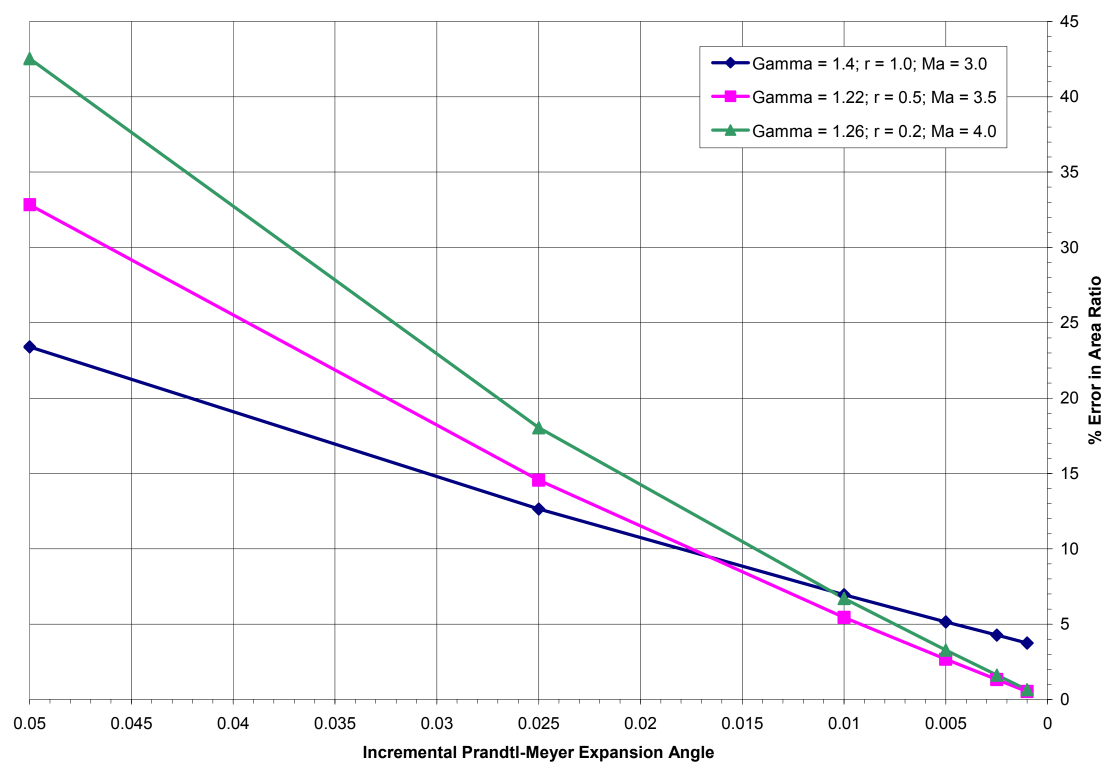


Figure 5.1.4: *% Error in Area Ratio vs. Incremental Prandtl-Meyer Expansion Angle for Internal-External Aerospike Nozzle Calculations*

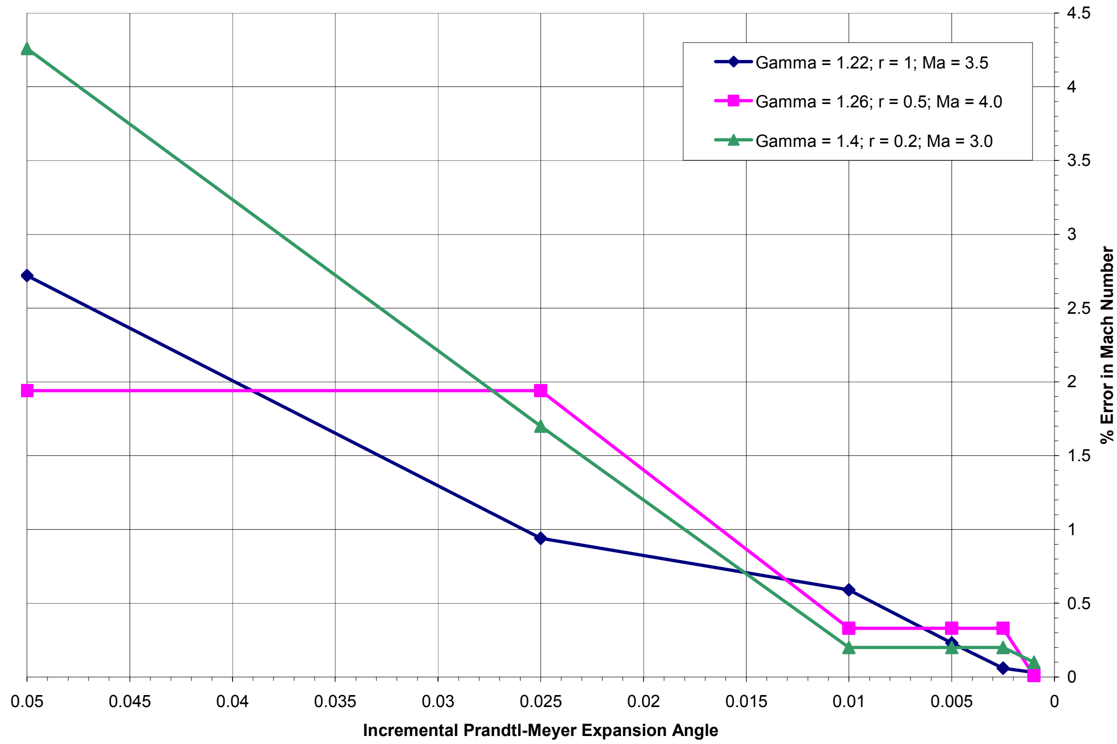


Figure 5.1.5: *% Error in Mach Number vs. Incremental Prandtl-Meyer Expansion Angle for Internal-External Aerospike Nozzle Calculations*

The incremental step size in Prandtl-Meyer expansion angle is the largest contributor to error propagation in the program. Generally speaking, the smaller the step size in Prandtl-Meyer expansion angle, i.e. more characteristic curves, the more accurate the program becomes. Decreasing the incremental Prandtl-Meyer expansion angle's size also has the added advantage of creating a smoother contour since there are more points defining its shape.

5.2 Isentropic Assumption Check

Since the equations developed for the program were derived assuming isentropic flow, it must be verified that the program executes isentropic calculations. To show that the program's solutions have minimal entropy change, i.e. isentropic, each subroutine calculates the entropy change of each point in the flowfield from the stagnation conditions in the combustion chamber. The conditions in the combustion chamber predicted by the CEA program developed by NASA discussed in Section 4.1 were used. Assuming an ideal gas and the predicted fluid properties, the fluid's gas constant was calculated using equation 5.2.

$$c_p(T) = \frac{\gamma \cdot R}{\gamma - 1} \qquad \textit{Eq. 5.2}$$

Since each point's Mach number is calculated during the program, its temperature and pressure can also be solved for from isentropic supersonic nozzle flow theory using

$$T_{point} = \frac{1}{\left(1 + \frac{\gamma-1}{2} \cdot M_{point}^2\right)} \cdot T_{stagnation} \qquad \textit{Eq. 5.3}$$

$$P_{point} = \left(1 + \frac{\gamma-1}{2} \cdot M_{point}^2\right)^{-\gamma/\gamma-1} \cdot P_{stagnation} \qquad \textit{Eq. 5.4}$$

where T is the temperature, P is the pressure, γ is the ratio of specific heats and M is the Mach number.

With each point's temperature and pressure known, the change in entropy from the reference state for each point can be calculated using equation 5.5.

$$\Delta s = c_p \cdot \ln \frac{T_{point}}{T_{reference}} - R \ln \frac{P_{point}}{P_{reference}} \qquad \textit{Eq. 5.5}$$

Figures 5.2.1 through 5.2.3 show a sample of the calculated entropy changes in the flowfield for the annular, IE aerospike and aerospike nozzles, respectively.

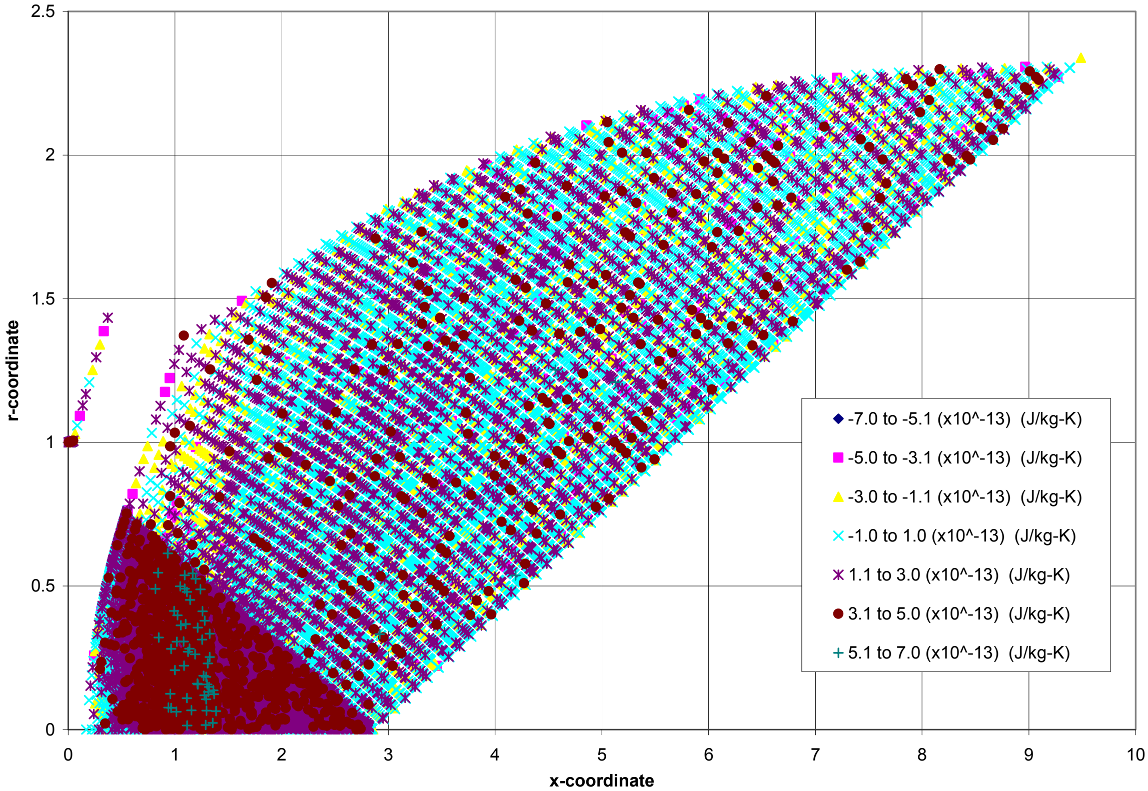


Figure 5.2.1: *Calculated Change in Entropy for an Annular Nozzle*

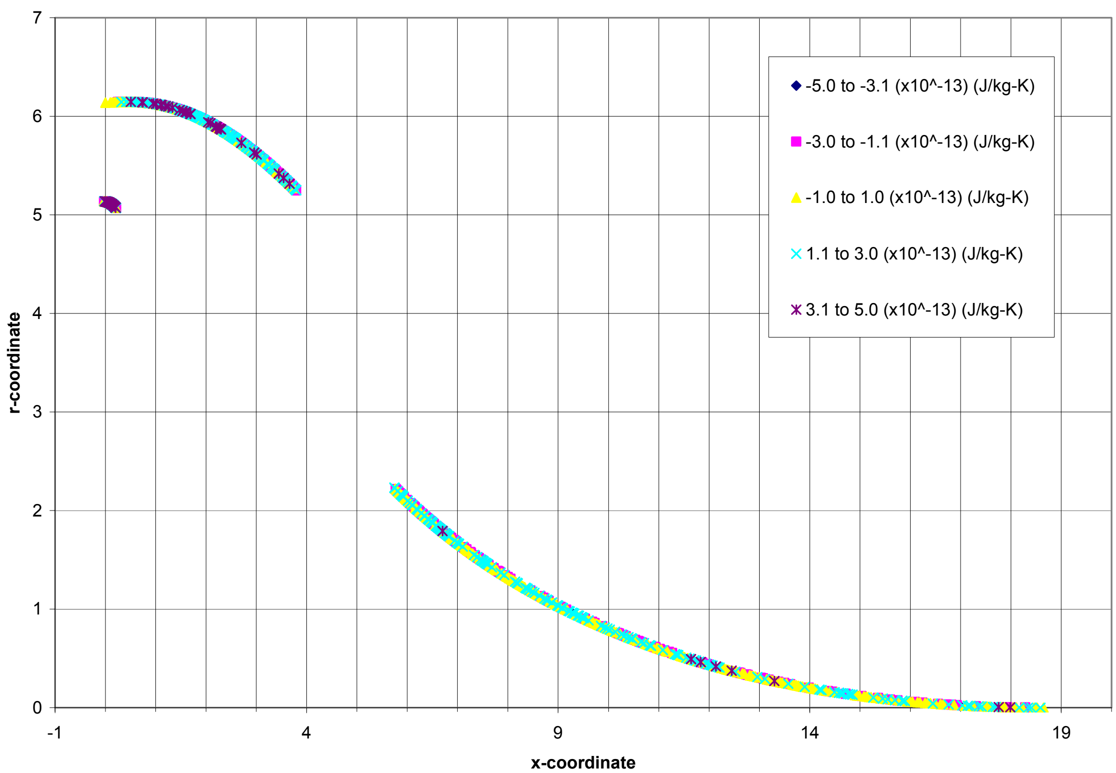


Figure 5.2.2: *Calculated Change in Entropy for an Internal-External Aerospike Nozzle*

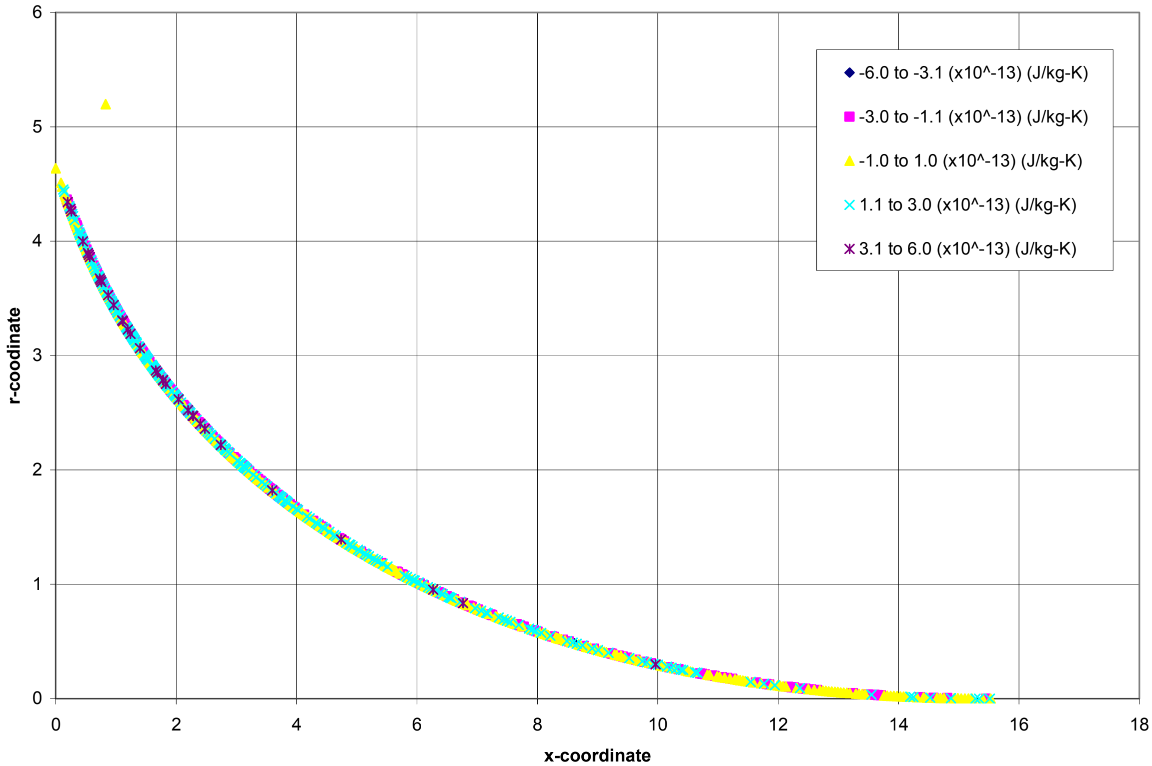


Figure 5.2.3: *Calculated Change in Entropy for an External Aerospike Nozzle*

The program exhibits a calculated entropy change from the reference state on the order of 10^{-13}. This is smaller than the 10^{-10} error tolerance allowed in calculating the properties in the flowfield. Therefore, the change in entropy is negligible and the assumption of isentropic flow is validated in the program's calculation of all nozzle types.

5.3 CFD Accuracy of Computer Code

As mentioned in Section 4.0, FLUENT was used to independently check the accuracy of the program detailed in Section 3.0. The CFD program was used to simulate the flow and produce dimensional entropy change and Mach contour plots to evaluate if the flow was shock free and the desired exit Mach was reached.

Annular Nozzles

Using the setup configurations outlined in Section 4.2., Figure 5.3.1 shows a typical dimensional entropy contour for an annular nozzle.

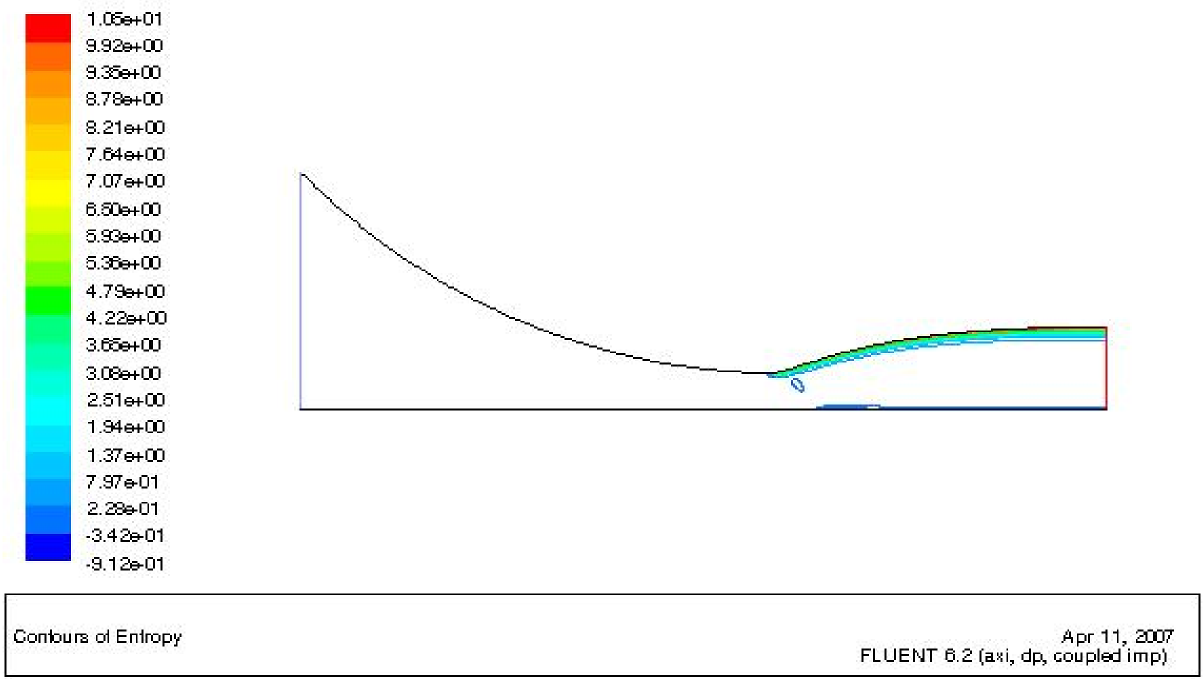


Figure 5.3.1: *Typical Entropy Contour for an Annular Nozzle*

From Figure 5.3.1, it can be seen that the large majority of the fluid domain demonstrates constant entropy signifying that the isentropic flow assumption is valid. The region near the wall contour where the entropy is changing is a result of the discontinuities in the wall contour. Since the wall contour was defined by a set of points that were connected by straight line segments, it is discontinuous at the points that connect them. The change in entropy in the flowfield is a propagation of these discontinuities.

The exit Mach number is checked by the Mach contours of the simulation as well as having FLUENT calculate the area-weighted Mach number at the exit plane of the nozzle. The area-weighted Mach number calculated by FLUENT is compared to the Mach number calculated by the program in Section 3.4 and the desired exit Mach number. Figure 5.3.2 shows the typical Mach contours of an annular nozzle designed for a Mach number of 3.0.

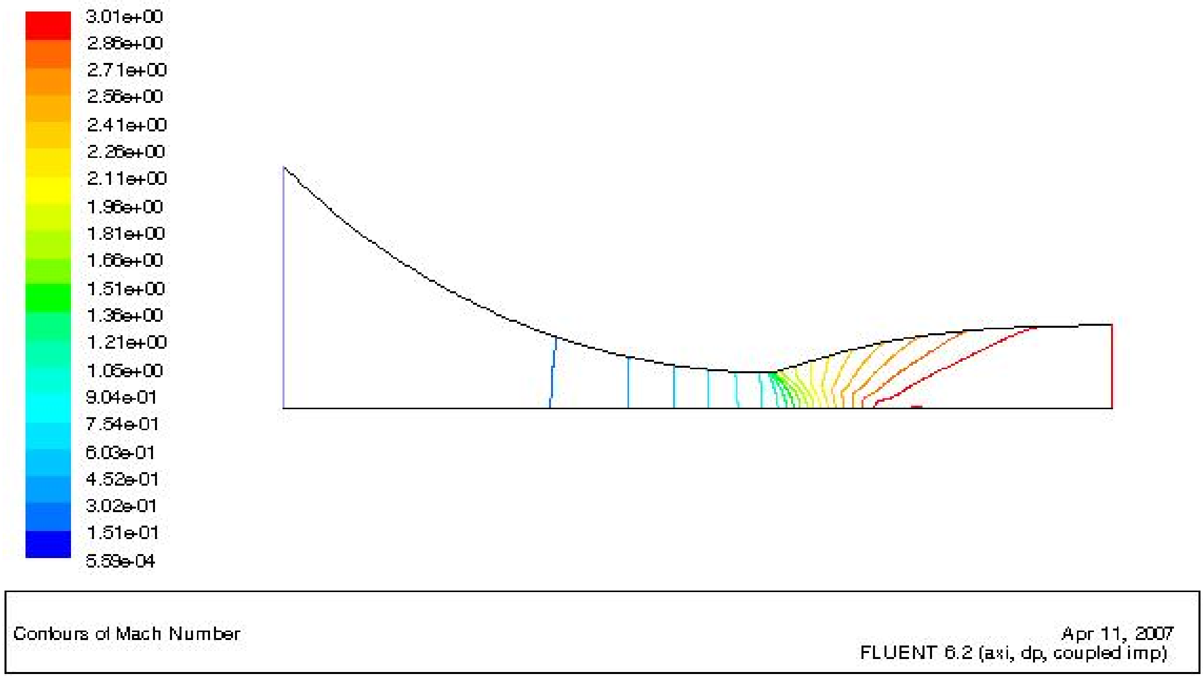


Figure 5.3.2: *Typical Mach Contours for an Annular Nozzle*

From Figure 5.3.1, it is clear that the Mach number at the exit has a maximum of 3.01. Table 5.3.1 shows typical % error in exit Mach number for annular nozzles designed by the program detailed in Section 3.0 when compared to FLUENT results.

Table 5.3.1		**Mach Number Comparisons for Annular Nozzle**		
	$Ma_{Desired}$	$Ma_{ComputerCalculated}$	$Ma_{FluentCalculated}$	$Ma_{ContourPlot}$
	3.0	3.0062	2.9528	3.01
Percent Error From $Ma_{Desired}$		0.21%	-1.57%	0.33%

After comparing the Mach numbers, it is clear that the code described in Section 3.0 is valid and accurate. An array of CFD simulations for annular nozzles designed for air and the exhaust products of N_2O/HTPB as the working fluids can be found in Appendix D complete with Gambit models with meshing inputs and results, FLUENT inputs, entropy change contours, Mach contours and Mach comparison tables. The additional annular nozzle simulation results presented in Appendix D have similar results as presented in this section.

External Aerospike Nozzles

Using the simulation setup outlined in Section 4.3, Figures 5.3.3 and 5.3.4 are typical entropy change contours seen in 100% and 20% length external aerospike nozzles. For perspective, the figures used were taken from simulations of an external aerospike nozzle designed for an exit Mach number of 3.0.

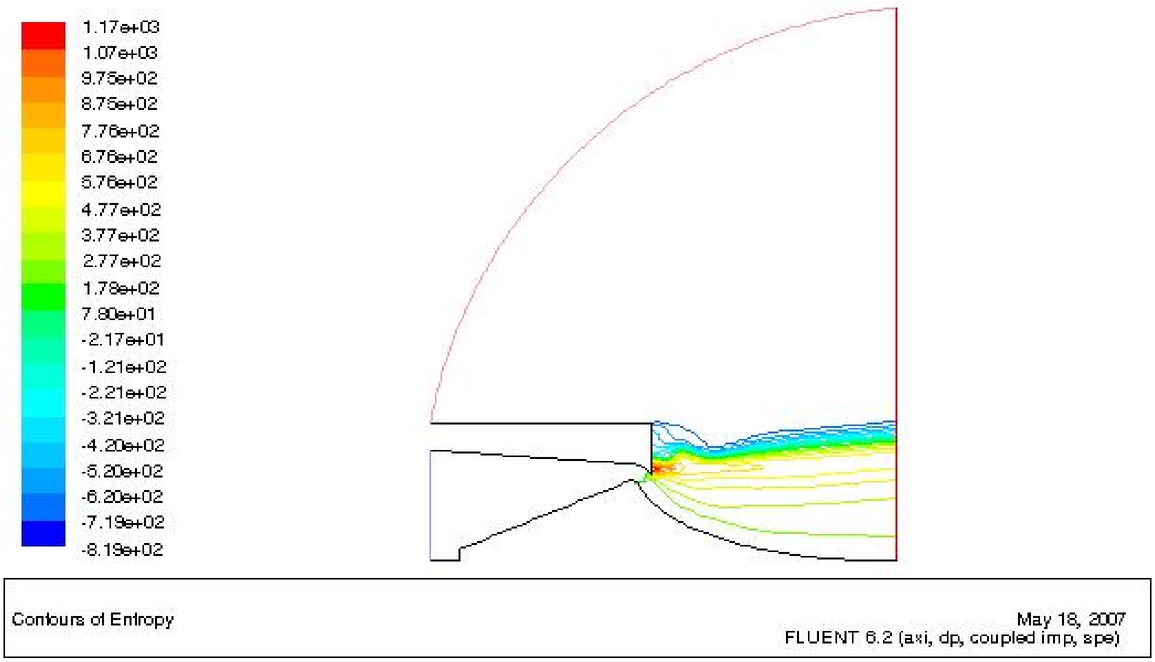


Figure 5.3.3: *Typical Entropy Change Contours of a 100% Length External Aerospike Nozzle*

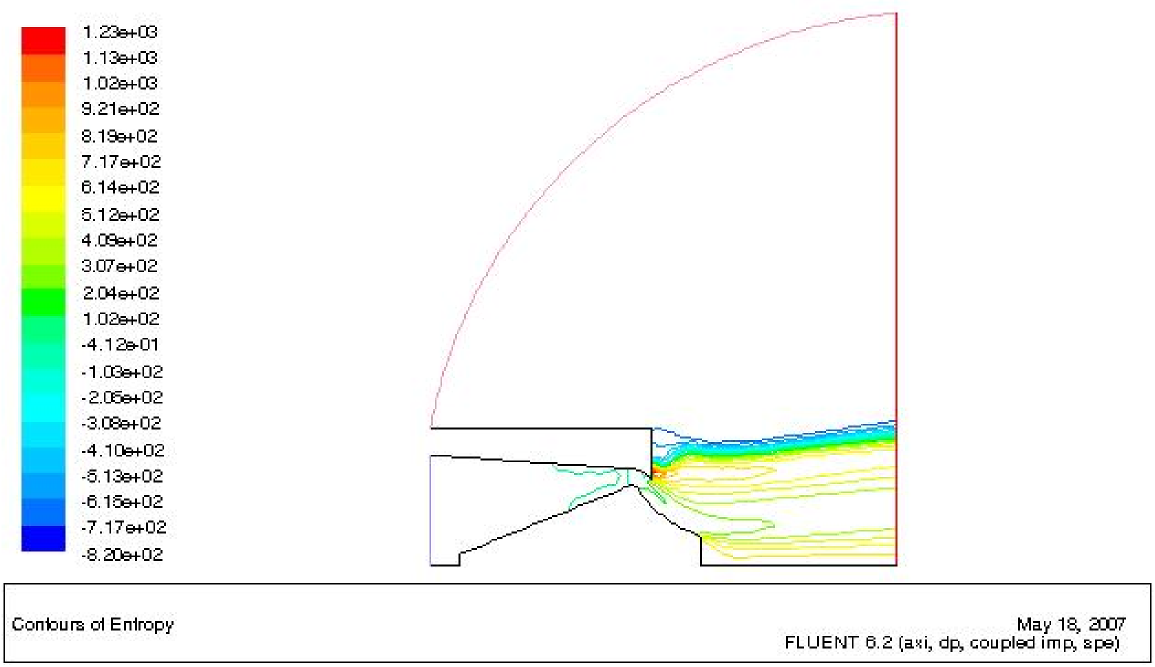


Figure 5.3.4: *Typical Entropy Change Contours of a 20% Length External Aerospike Nozzle*

As expected, there are significant changes in entropy where the “imaginary” wall(s) of the external aerospike nozzle develop along the constant pressure boundaries. Taking a closer look at the plots, it is evident that the main body of the exhaust plume remains at constant entropy validating the isentropic assumption used to develop the computer program.

Figure 5.3.5 and 5.3.6 are the Mach contours of typical 100% and 20% length external aerospike nozzles, respectively.

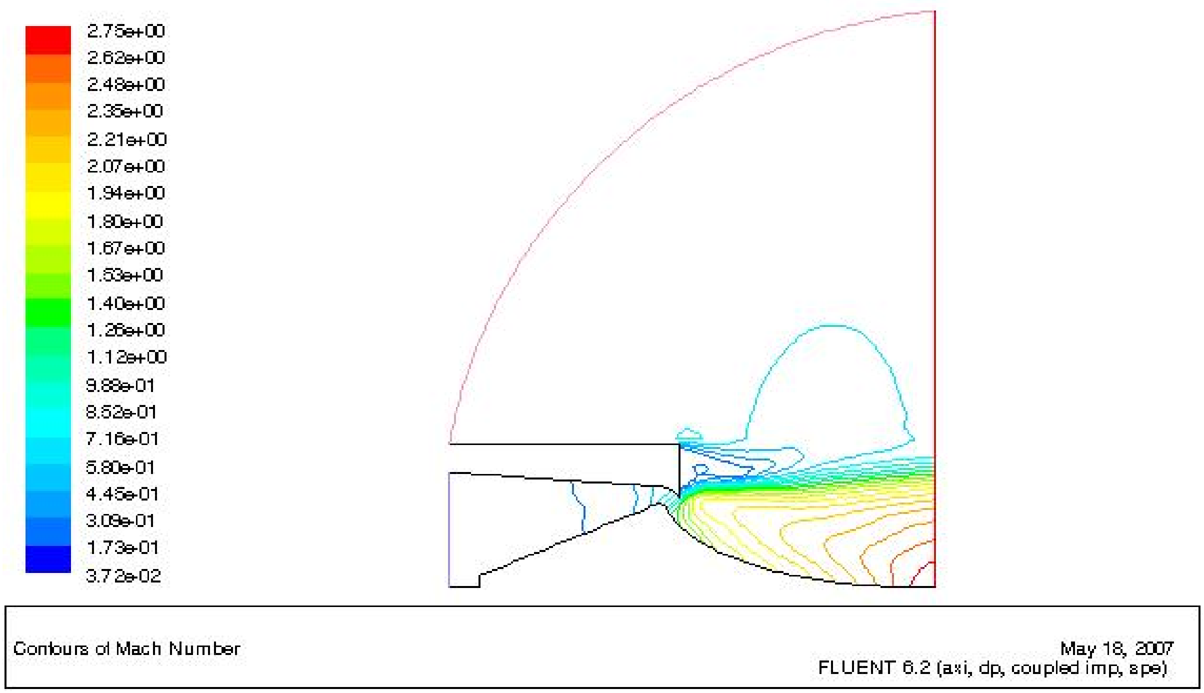


Figure 5.3.5: *Typical Mach Contours of a 100% Length External Aerospike Nozzle*

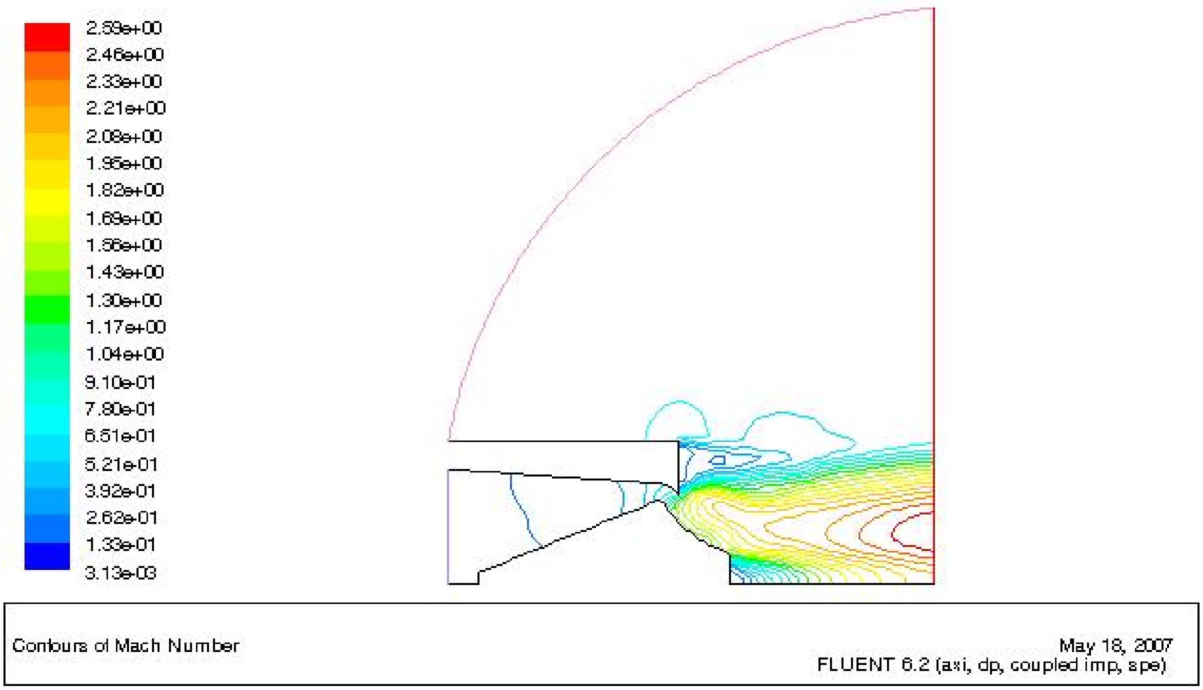


Figure 5.3.6: *Typical Mach Contours of a 20% Length External Aerospike Nozzle*

Using FLUENT to calculate the area-weighted exit Mach number, the 100% and 20% length external aerospike nozzle simulations yield an exit Mach number of 2.3682 and 2.4298, respectively. Although the results of the simulations for the external aerospike nozzles are less encouraging when compared to the desired exit Mach number, this is not all together unexpected. Reflecting back on the calculations used to define the nozzle's contour, only one characteristic equation was used with no consideration taken for the characteristic's curvature or effects the constant pressure boundary may cause. Further investigation is needed to devise a method to incorporate the characteristic's curvature and constant pressure boundary which in turn should improve the accuracy of the code described in Section 3.0. Tables 5.3.2 and 5.3.3 show Mach number comparisons between the desired, computer calculated and largest contour value for 100% and 20% length aerospike nozzles, respectively.

Table 5.3.2		**Mach Number Comparisons for 100% Length Aerospike Nozzle**		
	$Ma_{Desired}$	$Ma_{ComputerCalculated}$	$Ma_{FluentCalculated}$	$Ma_{ContourPlot}$
	3.0	3.0000	2.3682	2.75
Percent Error From $Ma_{Desired}$		0.0%	-21.06%	-8.33%

Table 5.3.3		**Mach Number Comparisons for 20% Length Aerospike Nozzle**		
	$Ma_{Desired}$	$Ma_{ComputerCalculated}$	$Ma_{FluentCalculated}$	$Ma_{ContourPlot}$
	3.0	3.0000	2.4298	2.59
Percent Error From $Ma_{Desired}$		0.0%	-19.01%	-13.67

After comparing the Mach numbers, it is clear that the code described in Section 3.4 is valid, but much less accurate for aerospike nozzles than annular nozzles. An array of CFD simulations for external aerospike nozzles and designed for air and the combustion products of N_2O/HTPB as the working fluids are available in Appendix D complete with Gambit models with meshing inputs and results, FLUENT inputs, entropy change contours, Mach contours and Mach comparison tables.

Internal-External Aerospike Nozzles

Using the setup described in Section 4.3, Figures 5.3.7 and 5.3.8 show the entropy change contours for typical 100% and 20% length IE aerospike nozzles, respectively. As expected, the fluid regions where “imaginary” contours are located show high variations in entropy. The entropy change contours also show little change in the fluid flowfield between the wall and “imaginary” contours validating the assumption of isentropic flow.

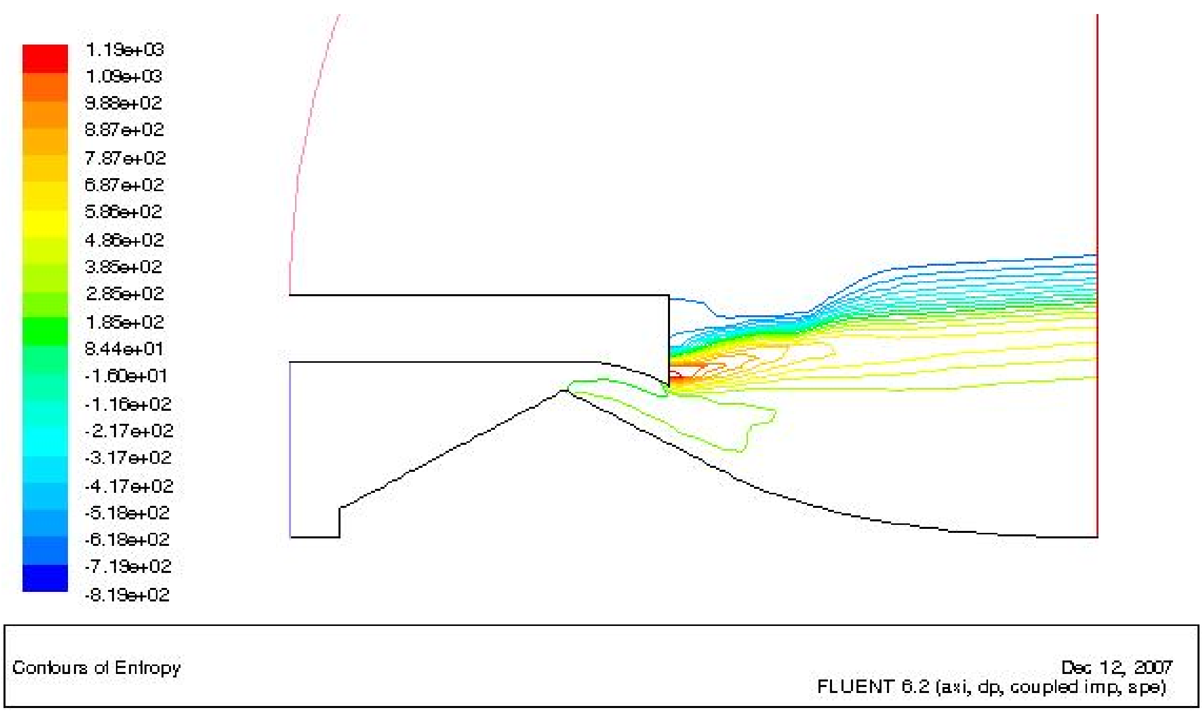


Figure 5.3.7: *Typical Entropy Contours of a 100% Length IE Aerospike Nozzle*

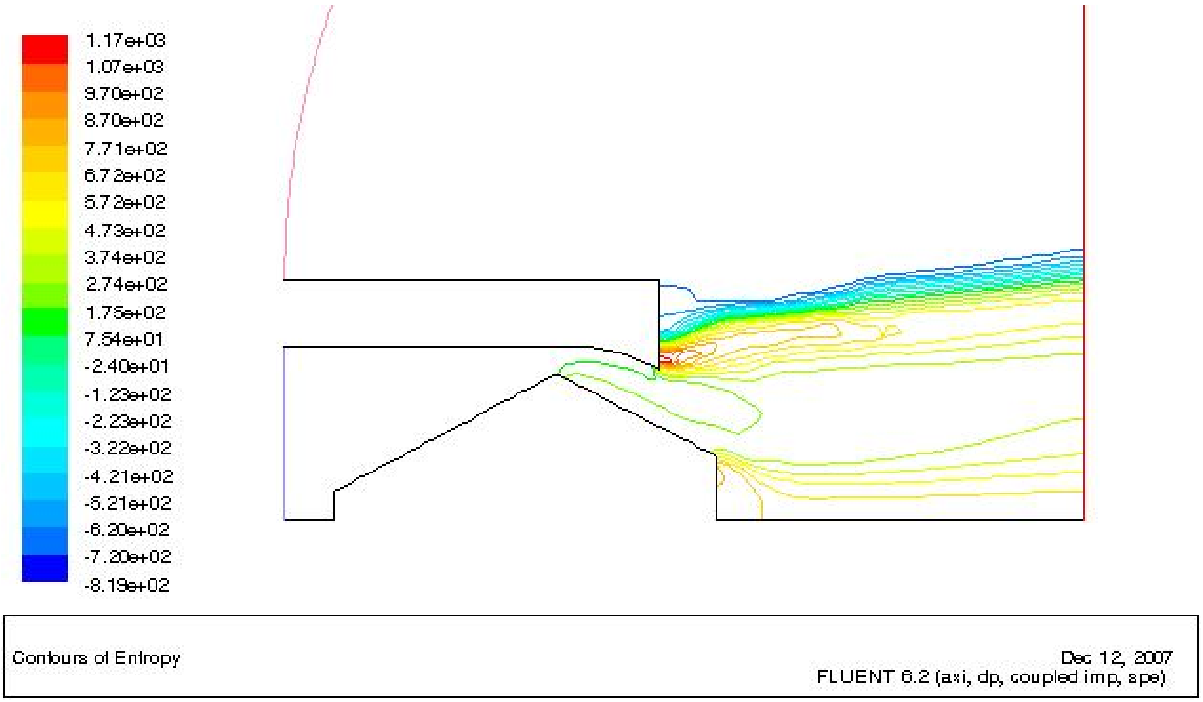


Figure 5.3.8: *Typical Entropy Contours of a 20% Length IE Aerospike Nozzle*

Figures 5.3.9 and 5.3.10 show typical Mach contours produced by FLUENT. Tables 5.3.4 and 5.3.5 compare the FLUENT results to the user-defined desired exit Mach number and the program calculated Mach number for 100% and 20% length IE aerospike nozzles designed for an exit Mach number of 3.0, respectively.

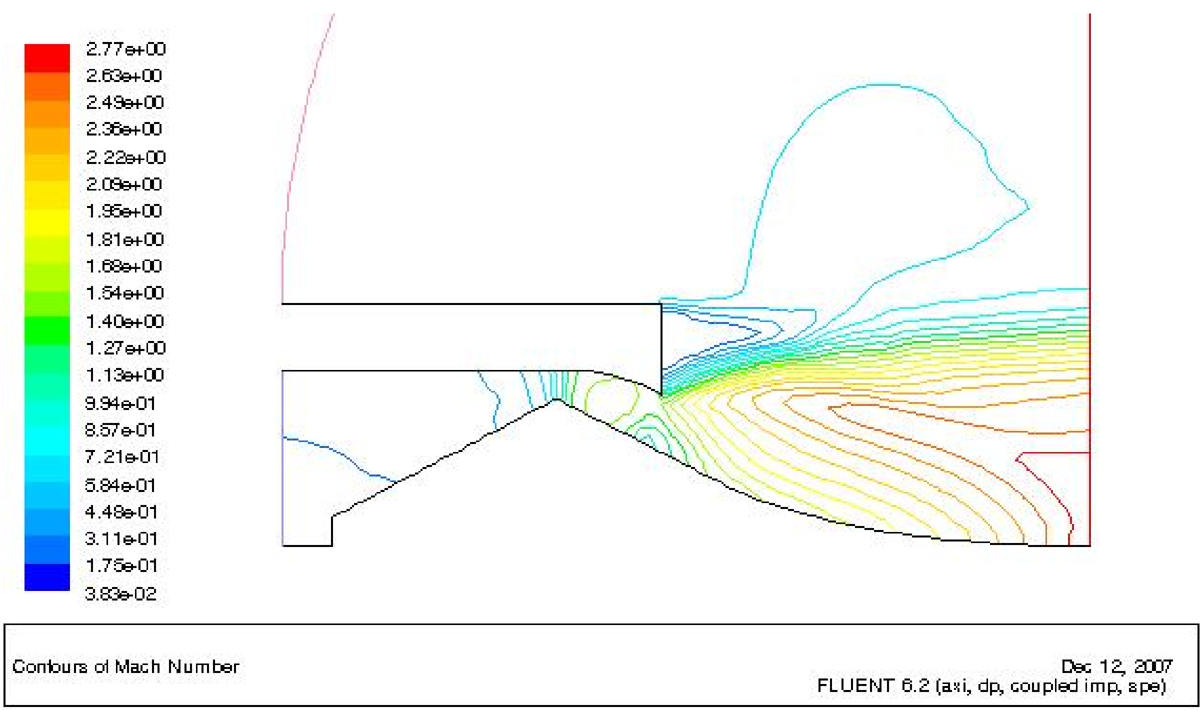


Figure 5.3.9: *Typical Mach Contours of a 100% Length IE Aerospike Nozzle*

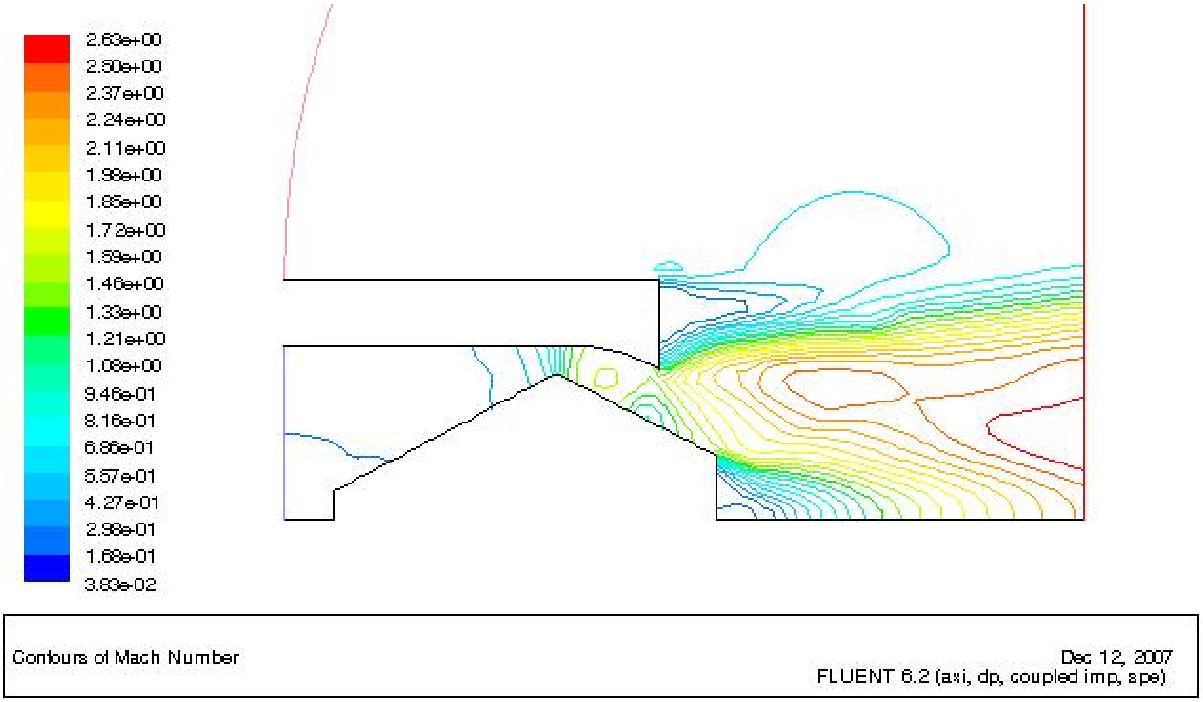


Figure 5.3.10: *Typical Mach Contours of a 20% Length IE Aerospike Nozzle*

Table 5.3.4		**Mach Number Comparisons for 100% Length IE Aerospike Nozzle**		
	$Ma_{Desired}$	$Ma_{ComputerCalculated}$	$Ma_{FluentCalculated}$	$Ma_{ContourPlot}$
	3.0	3.0003	2.57	2.77
Percent Error From $Ma_{Desired}$		0.01%	-14.33%	-7.67%

Table 5.3.5		**Mach Number Comparisons for 20% Length IE Aerospike Nozzle**		
	$Ma_{Desired}$	$Ma_{ComputerCalculated}$	$Ma_{FluentCalculated}$	$Ma_{ContourPlot}$
	3.0	3.0003	2.52	2.63
Percent Error From $Ma_{Desired}$		0.01%	-16.00%	-12.33%

Although agreement between desired exit Mach number and simulated exit Mach number are below expectations, it is evident that the code is still relatively valid and that adjustment and refinement is needed to more accurately design an IE aerospike contour for a desired exit Mach number. Better understanding of the constant pressure boundary as well as a method to account for the curvature of the characteristics seen in axisymmetric flow, which is currently not associated with the design of an IE Aerospike nozzle in this paper, is needed to improve the accuracy of the code described in Section 3.0.

An array of CFD simulations for IE aerospike nozzles designed for air and the combustion products of N_2O/HTPB as the working fluids are available in Appendix D complete with Gambit models with meshing inputs and results, FLUENT inputs, entropy change contours, Mach contours and Mach comparison tables.

In Appendix D for aerospike nozzles designed for the combustion products of N_2O/HTPB with a design exit Mach number of 4.0, you will notice that the supersonic region in the flow does not fully expand. This is because the ambient pressure is too large for the exhaust gases to fully develop its supersonic flow field. It can be seen from isentropic pressure relations that the exhaust gases do not have the necessary potential energy to accelerate to the desired exit Mach number of 4.0.

5.4 Conical Nozzle CFD Results

As discusses in Section 4.4, conical nozzle CFD simulations were included as a basis of comparison. Using the setup given in Section 4.4, Figure 5.4.1 shows typical entropy change contours of conical nozzles.

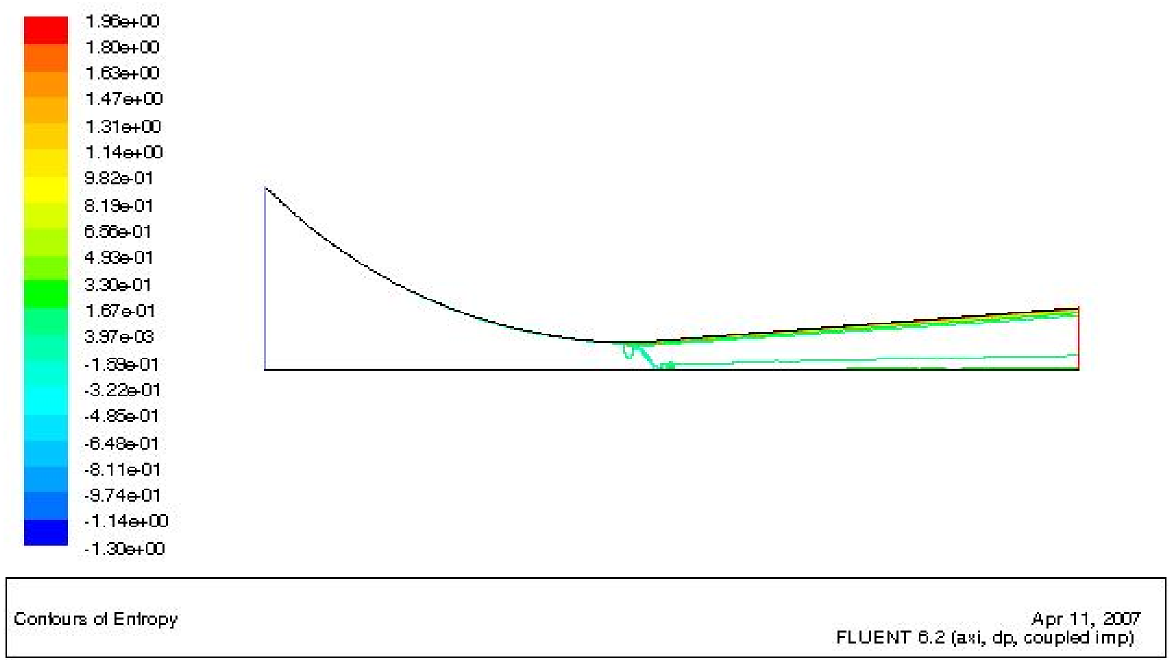


Figure 5.4.1: *Typical Entropy Contours for Conical Nozzles*

As expected, since the conical nozzle contour is not based on isentropic flow properties, the fluid exhibits a greater change in entropy through the nozzle, ultimately leading to a lower exit Mach number. Figure 5.4.2, the Mach contours of the conical nozzle, supports this conclusion when compared to Figure 5.3.1 for a similar annular nozzle.

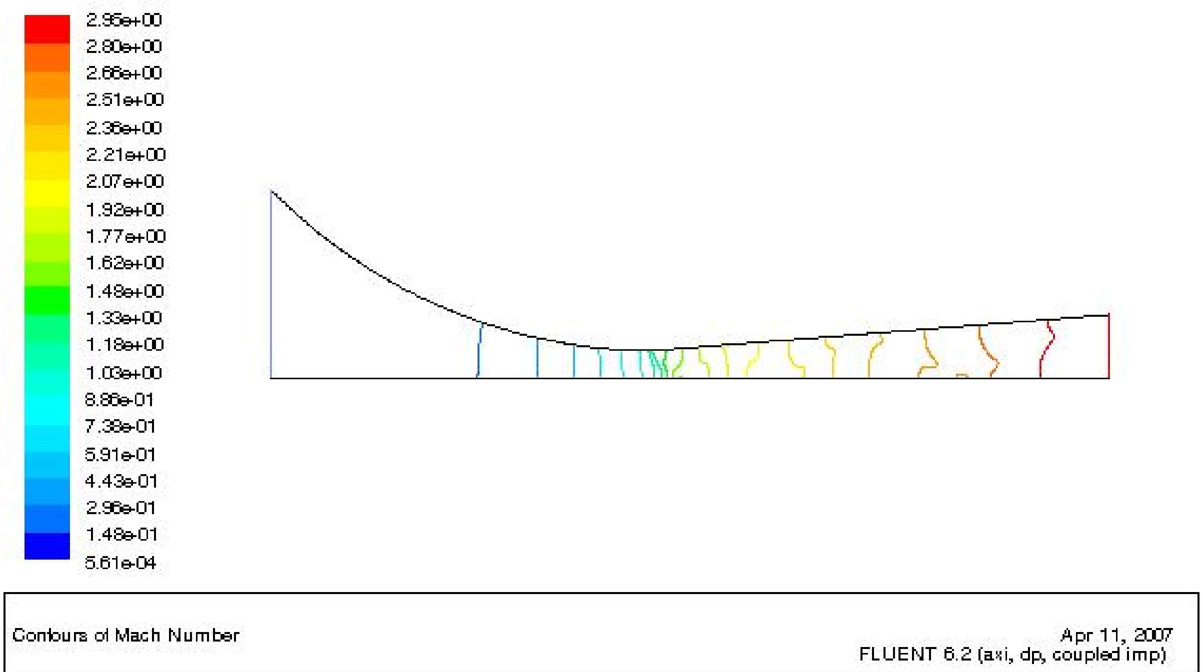


Figure 5.4.2: *Typical Mach Contours for a Conical Nozzle*

As seen from Figure 5.4.2, the Mach number at the exit of the nozzle does not reach the desired exit Mach number. Table 5.4.1 compares the contour plot and FLUENT calculated Mach numbers with the desired exit Mach number below.

Table 5.4.1		**Mach Number Comparisons for a Conical Nozzle**	
	$Ma_{Desired}$	$Ma_{FluentCalculated}$	$Ma_{ContourPlot}$
	3.0	2.9317	2.95
Percent Error From $Ma_{Desired}$		-2.28%	-1.67%

After comparing the Mach numbers, it is clear that the trigonometry used to create the conical nozzle is valid and accurate. An array of conical nozzle CFD simulations for air and the combustion products of N_2O/HTPB as the working fluids are available in Appendix D complete with Gambit models with meshing inputs and results, FLUENT inputs, entropy contours, Mach contours and Mach comparison tables.

6.0 Conclusions

The code developed in this thesis proves to be a useful tool in creating annular and aerospike supersonic nozzle contours for isentropic, irrotational, inviscid flow. The program exhibits increasing accuracy in the exit Mach number and exit area ratio as the incremental Prandtl-Meyer expansion angle decreases. This accuracy increase is independent of fluid or desired exit Mach number. The exit Mach number of the nozzles calculated with the program described in Section 3.0 shows good agreement with the FLUENT simulated ext Mach numbers. This independently confirms the accuracy of the program in calculating supersonic nozzle contours for inviscid, isentropic, irrotational supersonic flows.

The code developed in this thesis will enable the METEOR project team to investigate other types of rocket nozzles besides conical. It will further advance the team in achieving its ultimate goal, designing a low-cost system to launch pico-satellites into Low Earth Orbit.

7.0 Future Work and Recommendations

7.1 Recommendations

Upon error analysis of the code developed in this paper, it is evident that for isentropic, irrotational, inviscid flow the code is valid and accurate. It is recommended that the code is expanded upon to include the effects of viscosity, entropy change and rotation in its calculation of a supersonic nozzle contour. This will increase the code's ability to accurately predict real world flowfields and ultimately produce even higher efficient nozzle contours.

It is also recommended that further research is devoted to better characterizing the flowfield of aerospike nozzles. Although these nozzles are less efficient than annular nozzle at the same design altitude, the aerospike's ability to automatically adjust for altitude changes allows it to be more efficient over the entire mission. This suggests that a single-stage rocket may be designed to further reduce cost. In order for this to be realized, multiple challenges need to be overcome. The constant pressure boundary which forms the "imaginary" wall contour and better characterizing the curved nature of the characteristics in axisymmetric supersonic flowfields are two examples of the challenges which need to be overcome. Due to limited test data available on aerospike nozzles, it is recommended that test firings of aerospike nozzles are conducted.

7.2 Future Work

In pursuit of their ultimate goal of developing a low-cost alternative launch platform for pico-satellites, the METEOR project at RIT continues to develop new technologies. Currently the team is developing a vertical test stand to measure the thrust of various rocket engine designs. In the Spring Semester of the 2007-08 school year, the team will test multiple rocket nozzles that were designed by the program described in Section 3.0. A comparison of the theoretical and measured thrusts produced by each nozzle will be conducted. These tests will validate the accuracy of the program for real-world flows and also give an indication of the energy losses observed in the nozzles. The testing will also enable the team to compare the thrust/weight/cost ratios to develop the most cost effective rocket engine.

The team is also working on optimizing the helium balloon lift platform which will lift the rocket to 30,000m, its initial firing altitude. Additional work is being done to develop and incorporate attitude controls to the rocket.

The METEOR team at the Rochester Institute of Technology continues to spear head the initiative in developing a low-cost alternative launch platform for pico-satellites and the exploration of Low-Earth Orbit and space.

<u>**Bibliography**</u>

Anderson, JD., 2001, *Fundamentals of Aerodynamics*, 3rd Edition, pp. 532-537, pp.555-585.

Anderson, JD., 1982, *Modern Compressible Flow with Historical Perspective*, pp. 268-270, pp. 282-286.

Angelino, G., Oct. 1964, “Approximate Method for Plug Nozzle Design”, *AIAA Journal*, Vol. 2, No. 10, pp. 1834-1835.

Dale, D., Kozak, J., Patru, D., 2006, “Design and Testing of a Small Scale Rocket for Pico-Satellite Launching”, Rochester Institute of Technology METEOR Project, Senior Design Project 06006.

Foelsch, K., Mar. 1949, “The Analytic Design of an Axially Symmetric Laval Nozzle for a Parallel and Uniform Jet”, *Journal of the Aeronautical Sciences*, Vol. 16, No. 3, pp. 161-166, 188.

Greer, H., Nov. 1960, “Rapid Method for Plug Nozzle Design”, *ARS Journal*, pp. 560-561.

Lee, C.C., Thompson, D.D., Mar. 1969, “Fortran Program for Plug Nozzle Design”, *NASA Technical Memorandum, NASA TM-53019*, pp. 1-51.

Shapiro, AH., 1953, *The Dynamics and Thermodynamics of Compressible Fluid Flow*, Vol. I, pp. 294-295.

Shapiro, AH., 1954, *The Dynamics and Thermodynamics of Compressible Fluid Flow*, Vol. II, pp. 694-695.

Sutton, GP, *Rocket Propulsion Elements*, 7th Edition.

“The Radio Amateur’s Satellite Handbook”, Published by The American Radio Relay League, 1st Edition Revised, 2001.

“Anthology of Radio Amateur Satellites”, Published by The American Radio Relay League.

http://www.atlanticresearchcorp.com/docs/space_solidspace.shtml

http://www.orbital.com

http://www.ravenind.com/RavenCorporate/eng_films/high_alt_balloons_index.htm

http://www.sea-launch.com

Appendix A: Derivation of the Characteristic and Compatibility Equations

A.1 Annular Method of Characteristics Background

Axisymmetric Irrotational Flow:

Consider a cylindrical coordinate system as sketched in Figure A.1.1. The cylindrical coordinates are r, Φ and x, with corresponding velocity components v, w and u, respectively.

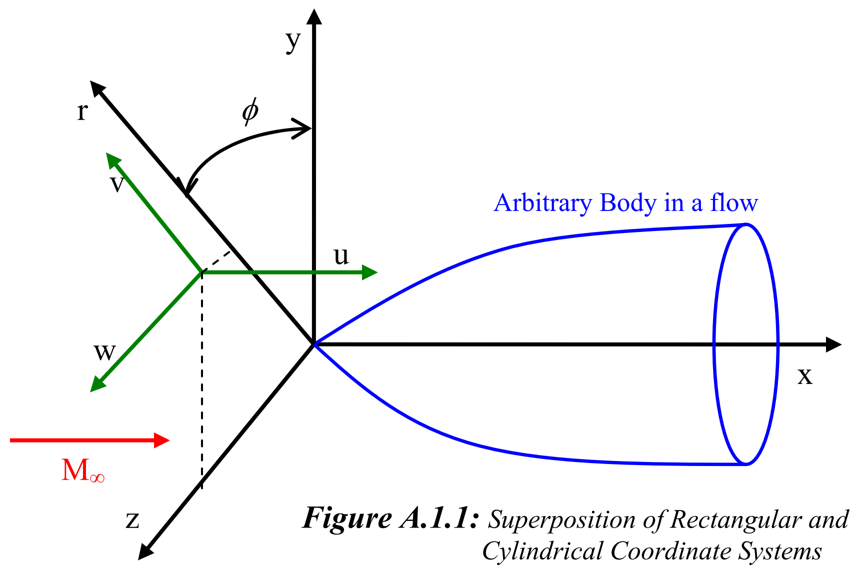


Figure A.1.1: *Superposition of Rectangular and Cylindrical Coordinate Systems for Axisymmetric Flows*

The continuity equation

$$\nabla \cdot \left(\rho \vec{V} \right) = 0 \qquad \textit{Eq. A.1}$$

Then becomes

$$\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial r} + \frac{1}{r}\frac{\partial(\rho w)}{\partial \phi} + \frac{\rho v}{r} = 0 \qquad \textit{Eq. A.2}$$

Since axisymmetric, irrotational flow is assumed, $\partial/\partial\phi = 0$, *Eq. 3.2* simplifies to

$$\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial r} + \frac{\rho v}{r} = 0 \qquad \textit{Eq. A.3}$$

From Euler's equation for irrotational flow,

$$dp = -\rho \vec{V} \cdot d\vec{V} = \frac{\rho}{2} \cdot d\left(\vec{V}^2\right) = -\frac{\rho}{2} \cdot d\left(u^2 + v^2 + w^2\right) \qquad \textit{Eq. A.4}$$

However, the speed of sound $a = \left(\partial p / \partial \rho\right)_s = dp/d\rho$, along with $w = 0$ for axisymmetric flow, *Eq. A.4* becomes

$$dp = -\frac{\rho}{a^2}\left(u \cdot du + v \cdot dv\right) \qquad \textit{Eq. A.5}$$

from which follows

$$\frac{\partial \rho}{\partial x} = -\frac{\rho}{a^2}\left(u \cdot \frac{\partial u}{\partial x} + v \cdot \frac{\partial v}{\partial x}\right) \qquad \textit{Eq. A.6}$$

$$\frac{\partial \rho}{\partial r} = -\frac{\rho}{a^2}\left(u \cdot \frac{\partial u}{\partial r} + v \cdot \frac{\partial v}{\partial r}\right) \qquad \textit{Eq. A.7}$$

Substituting *Eqs A.6* and *A.7* into *Eq A.3*, yields, after factoring,

$$\left(1 - \frac{u^2}{a^2}\right)\frac{\partial u}{\partial x} - \frac{uv}{a^2}\frac{\partial v}{\partial x} - \frac{uv}{a^2}\frac{\partial u}{\partial r} + \left(1 - \frac{v^2}{a^2}\right)\frac{\partial v}{\partial r} = -\frac{v}{r} \qquad \textit{Eq. A.8}$$

To satisfy the condition of irrotationality

$$\nabla \times \vec{V} = 0 \qquad \textit{Eq. A.10}$$

which can be written in matrix form for cylindrical coordinates as

$$\nabla \times \vec{V} = \frac{1}{r}\begin{vmatrix} e_r & re_\phi & e_x \\ \left(\frac{\partial}{\partial r}\right) & \left(\frac{\partial}{\partial \phi}\right) & \left(\frac{\partial}{\partial x}\right) \\ v & rw & u \end{vmatrix} = 0 \qquad \textit{Eq. A.11}$$

For axisymmetric flow, *Eq A.11* yields

$$\frac{\partial u}{\partial r} = \frac{\partial v}{\partial x} \qquad \textit{Eq. A.12}$$

Substituting *Eq A.12* into *Eq A.8*, yields

$$\left(1 - \frac{u^2}{a^2}\right)\frac{\partial u}{\partial x} - 2\frac{uv}{a^2}\frac{\partial v}{\partial x} + \left(1 - \frac{v^2}{a^2}\right)\frac{\partial v}{\partial r} = -\frac{v}{r} \qquad \textit{Eq. A.13}$$

Keeping in mind that $u = u(x,r)$ and $v = v(x,r)$, there can also be written

$$du = \frac{\partial u}{\partial x}dx + \frac{\partial u}{\partial r}dr = \frac{\partial u}{\partial x}dx + \frac{\partial v}{\partial x}dr \qquad \textit{Eq. A.14}$$

and

$$dv = \frac{\partial v}{\partial x}dx + \frac{\partial v}{\partial r}dr \qquad \textit{Eq. A.15}$$

Equations *A.13, A.14* and *A.15* can be solved simultaneously for the three derivatives $\partial u/\partial x$, $\partial v/\partial x$, and $\partial v/\partial r$.

In order to determine the characteristic lines and compatibility equations, $\partial v/\partial x$ must be finite. Equations *A.13* through *A.15* are solved for $\partial v/\partial x$ using Cramer's Rule yielding:

$$\frac{\partial v}{\partial x}=\frac{\begin{vmatrix}\left(1-\frac{u^2}{a^2}\right) & -\frac{v}{r} & \left(1-\frac{v^2}{a^2}\right)\\ dx & du & 0\\ 0 & dv & dr\end{vmatrix}}{\begin{vmatrix}\left(1-\frac{u^2}{a^2}\right) & -2\frac{uv}{a^2} & \left(1-\frac{v^2}{a^2}\right)\\ dx & dr & 0\\ 0 & dx & dr\end{vmatrix}}=\frac{N}{D} \qquad \textit{Eq. A.16}$$

where N is the numerator and D is the denominator of the solution.

To find a solution, $\partial v/\partial x$ must be solved for an arbitrary choice of *dx* and *dr*. However, to obtain a defined solution for $\partial v/\partial x$, *dx* and *dr* must be chosen such that the denominator D is not equal to zero. If *dx* and *dr* are chosen so that D is equal to zero, then $\partial v/\partial x$ is not defined in the direction dictated by *dx* and *dr*. It must also be stipulated that $\partial v/\partial x$ is at least finite, even though it is not uniquely determined. An infinite value of $\partial v/\partial x$ would be physically inconsistent. Making $\partial v/\partial x$ indeterminate, satisfies both conditions.

To find the characteristic directions, set $D = 0$ yielding

$$\left(\frac{dr}{dx}\right)_{char}=\frac{-uv/a^2 \pm\sqrt{\left[\left(u^2+v^2\right)/a^2\right]-1}}{1-\left(u^2/a^2\right)} \qquad \textit{Eq. A.17}$$

Equation *A.17* defines the characteristic curves in the physical xr space.

Upon closer inspection, the term inside the square root becomes

$$\frac{u^2+v^2}{a^2}-1=\frac{V^2}{a^2}-1=M^2-1 \qquad \textit{Eq. A.18}$$

Therefore we can state

1. If M > 1, there are two real characteristics through each point of the flowfield.
2. If M = 1, there is one real characteristic through each point of the flow.
3. If M < 1, the characteristics are imaginary.

Since the object of this paper is to design a supersonic nozzle, only case one is of interest. Although case two is exhibited at the throat, the calculation of supersonic flowfield can only be calculated from the upstream to down stream direction dictating that the position of point defining the throat must be a user defined quantity. This negates the need for a calculation of the characteristic equation at the throat. Consider a streamline as sketched in Figure A.1.2. At point A, $u = V\cos\theta$ and $v = V\sin\theta$. Substituting these equations into *Eq A.17* yields

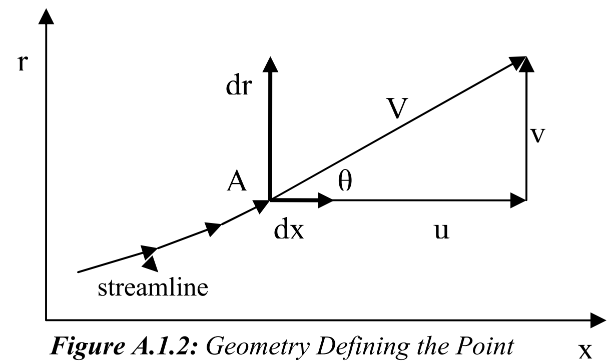


Figure A.1.2: *Geometry Defining the Point Flow Along a Streamline*

$$\left(\frac{dr}{dx}\right)_{char} = \frac{\dfrac{-V^2\cos\theta\sin\theta}{a^2} \pm \sqrt{\dfrac{V^2}{a^2}\left(\cos^2\theta+\sin^2\theta\right)-1}}{1-\dfrac{V^2}{a^2}\cos^2\theta} \qquad \textit{Eq. A.19}$$

Recalling that the Mach angle α is given by $\alpha = \sin^{-1}(1/M)$, or $\sin\alpha = 1/M$ and therefore $V^2/a^2 = M^2 = 1/\sin\alpha$ and equation *A.19* becomes

$$\left(\frac{dr}{dx}\right)_{char} = \frac{\dfrac{-\cos\theta\sin\theta}{\sin^2\alpha} \pm \sqrt{\dfrac{\cos^2\theta+\sin^2\theta}{\sin^2\alpha}-1}}{1-\dfrac{\cos^2\theta}{\sin^2\alpha}} \qquad \textit{Eq. A.20}$$

From trigonometry,

$$\sqrt{\frac{\cos^2\theta+\sin^2\theta}{\sin^2\alpha}-1}=\sqrt{\frac{1}{\sin^2\alpha}-1}=\sqrt{\csc^2\alpha-1}=\sqrt{\cot^2\alpha}=\frac{1}{\tan\alpha} \qquad \textit{Eq. A.21}$$

Substituting *Eq A.21* into *Eq A.20*, equation *A.20* simplifies to

$$\left(\frac{dr}{dx}\right)_{char}=\frac{-\cos\theta\sin\theta/\sin^2\alpha\pm 1/\tan\alpha}{1-\left(\cos^2\theta/\sin^2\theta\right)} \qquad \textit{Eq. A.22}$$

After more algebraic and trigonometric manipulations, equation *A.22* becomes

$$\left(\frac{dr}{dx}\right)_{char}=\tan(\theta\mp\alpha) \qquad \textit{Eq. A.23}$$

Equation *A.23* shows that for axisymmetric irrotational flow, the characteristic lines are Mach lines. A sketch of the C_+ and C_- characteristics are shown in Figure A.1.3 below.

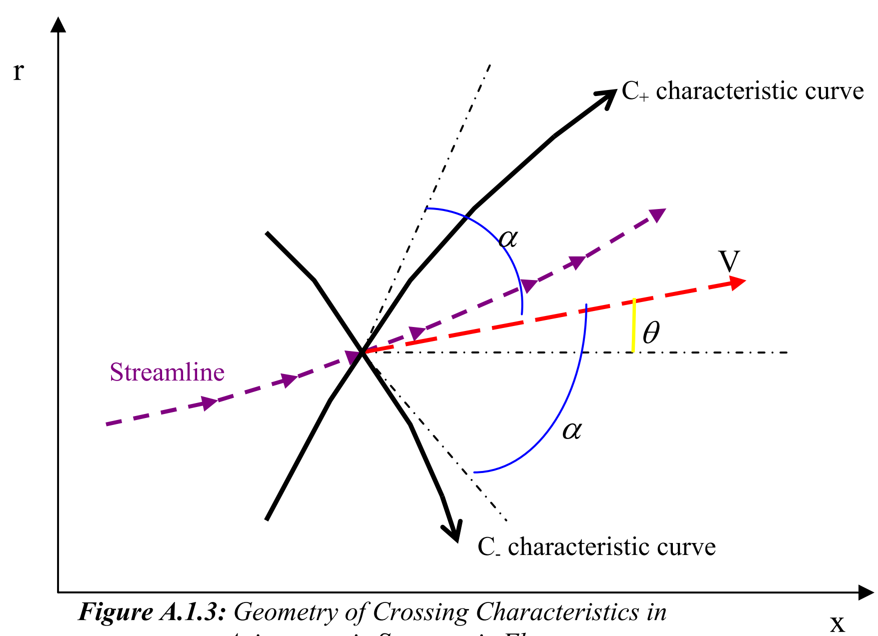


Figure A.1.3: *Geometry of Crossing Characteristics in Axisymmetric Supersonic Flow*

The compatibility equations, which describe the variation of flow properties along the characteristic curves, are found by setting N = 0 in equation *A.16*. The result of this is

$$\frac{dv}{du}=\frac{-\left(1-\frac{u^2}{a^2}\right)-\frac{v}{u}\frac{dx}{du}}{\left(1-\frac{v^2}{a^2}\right)\frac{dx}{dr}} \qquad \textit{Eq. A.24a}$$

or

$$\frac{dv}{du}=-\frac{\left(1-\frac{u^2}{v^2}\right)}{\left(1-\frac{v^2}{a^2}\right)}\frac{dr}{dx}-\frac{\frac{v}{r}\frac{dr}{du}}{\left(1-\frac{v^2}{a^2}\right)} \qquad \textit{Eq. A.24b}$$

The term dr/dx in equation *A.24b* is the characteristic direction given by equation *A.17*. Substituting equation *A.17* into equation *A.24b*, yields

$$\frac{dv}{du}=\frac{\frac{uv}{a^2}\mp\sqrt{\frac{u^2+v^2}{a^2}-1}}{\left(1-\frac{v^2}{a^2}\right)}-\frac{\frac{v}{r}\frac{dr}{du}}{\left(1-\frac{v^2}{a^2}\right)} \qquad \textit{Eq. A.25}$$

Referring again to Figure A.1.2, the substitutions $u=V\cos\theta$ and $v=V\sin\theta$ are made into equation *A.25*. After some algebraic manipulation, equation *A.25* becomes.

$$d\theta=\mp\sqrt{M^2-1}\frac{dV}{V}\pm\frac{1}{\sqrt{M^2-1}\mp\cot\theta}\frac{dr}{r} \qquad \textit{Eq. 3.26}$$

With the differential of the Prandtl-Meyer function given by $d\upsilon=\sqrt{M^2-1}\frac{dV}{V}$, equation *A.26* simplifies to

$$d(\theta \pm \upsilon) = \pm \frac{1}{\sqrt{M^2 - 1} \mp \cot\theta} \frac{dr}{r} \qquad \textit{Eq. A.27}$$

Equation *A.27* can be split into the two characteristic equations C_- and C_+ giving the final form of the compatibility equations.

$$d(\theta + \upsilon) = \frac{1}{\sqrt{M^2 - 1} - \cot\theta} \frac{dr}{r} \qquad \textit{Eq. }\mathrm{A.}\textit{28a}$$

(along a C_- characteristic)

$$d(\theta - \upsilon) = -\frac{1}{\sqrt{M^2 - 1} + \cot\theta} \frac{dr}{r} \qquad \textit{Eq. A.28b}$$

(along a C_+ characteristic)

Appendix B: Computer Codes

B.1 SupersonicNozzle.m Program

Figure B.1.1: SupersonicNozzle.m Program Flowchart

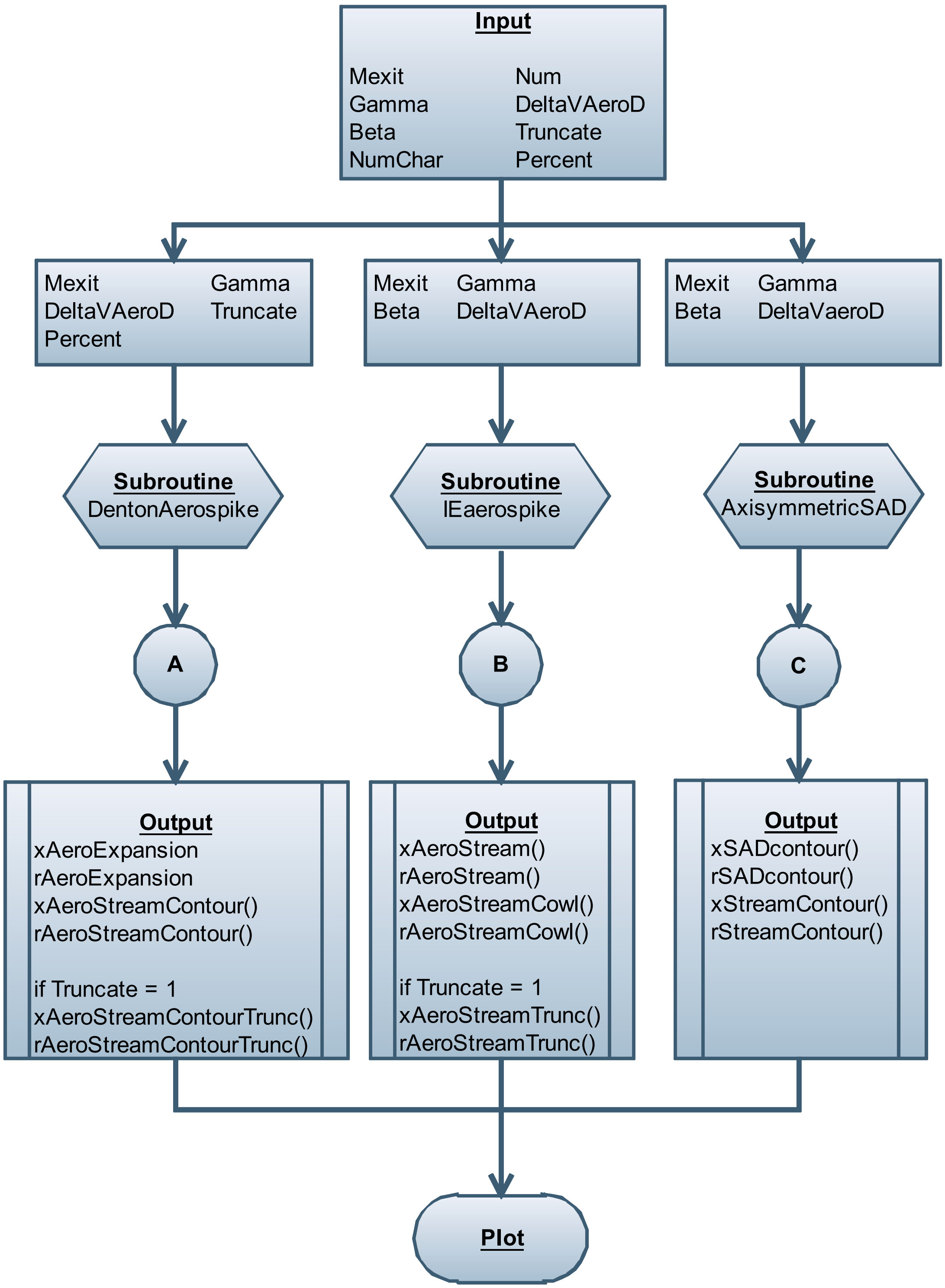


Figure B.1.2: SupersonicNozzle.m MatLab Source Code

```
%*************************************************************************
%*                       Supersonic Nozzle Design
%*                           By: Brandon Denton
%*                               RIT Graduate Student
%*************************************************************************
%*************************************************************************
%*       Dictionary of Variable
%*************************************************************************
%*   1. AxisymmetricSAD ---------- Subroutine that calculates a contour for
%*                                 an annular supersonic nozzle using the
%*                                 stream function and a combination of
%*                                 techniques outlined by Anderson, JD, and
%*                                 Shapiro, complied by Brandon Denton
%*                                 (Thesis Attached)
%*   2. Beta --------------------- Multiplication factor for radius of
%*                                 entrance region
%*   3. Choice ------------------- Holds the value of the Choice variable
%*                                 for the the truncation of the aerospike
%*                                 nozzle's contour
%*   4. DentonAerospike ---------- Subroutine that calculates the contour of
%*                                 a supersonic aerospike nozzle written by
%*                                 Brandon Denton, 2007. (Attached Thesis)
%*                                 Using the stream function
%*   5. Gamma -------------------- Ratio of Specific Heats of working fluid
%*   6. Mexit -------------------- Desired exit Mach Number
%*   7. nb ----------------------- Variable that tells the subroutine
%*                                 AnnularAerospike what percentatge of the
%*                                 length of the idealized Aerospike nozzle
%*                                 should be retained
%*   8. NumChar ------------------ Number of Characteristics used in the
%*                                 Method of Characteristic Calculations
%*                                 aerospikes nozzle's contour
%*   9. Percent ------------------ Variable that holds the percentage the
%*                                 user would like to retain of the
%*                                 idealized Aerospike nozzle
%*  10. Pstag -------------------- Chamber Stagnation Pressure
%*  11. rAeroExpansion ----------- Array that holds the r-component of the
%*                                 expansion point calculated by subroutine
%*                                 DentonAerospike
%*  12. rAeroStream -------------- Array that holds the r-component of the
%*                                 Internal-External Aerospike nozzle
%*                                 contour
%*  13. rAeroStreamContour ------- Array that holds the r-component of the
%*                                 aerospike nozzle calculated by subroutine
%*                                 DentonAerospike
%*  14. rAeroStreamCowl ---------- Array that holds the r-components of the
%*                                 cowl of the Internal-External Aerospike
%*                                 nozzle
%*  15. rAeroStreamContourTrunc -- Array that holds the r-component of the
%*                                 annular aerospike calculated by
%*                                 subroutine DentonAerospike
%*  16. rAeroStreamTrunc --------- Array that holds the r-components of the
%*                                 truncated Internal-External Aerospike
%*                                 nozzle
%*  17. rSADcontour -------------- Array that holds the r-component of the
%*                                 annular nozzle contour calculated by
%*                                 subroutine AxisymmetricSAD but calculated
%*                                 it using the same technique used for the
%*                                 2D nozzles
%*  18. rStreamContour ----------- Array that holds the r-component of the
%*                                 annular nozzle contour calculated by
%*                                 subroutine AxisymmetricSAD using the
%*                                 stream function technique
%*  19. rThroat ------------------ User-defined radius of the throat
%*  20. Truncate ----------------- Variable that tells the program whether
%*                                 or not the user would like to truncate
%*                                 the aerospike nozzle
%*  21. Tstag -------------------- Chamber Stagnation Temperature
```

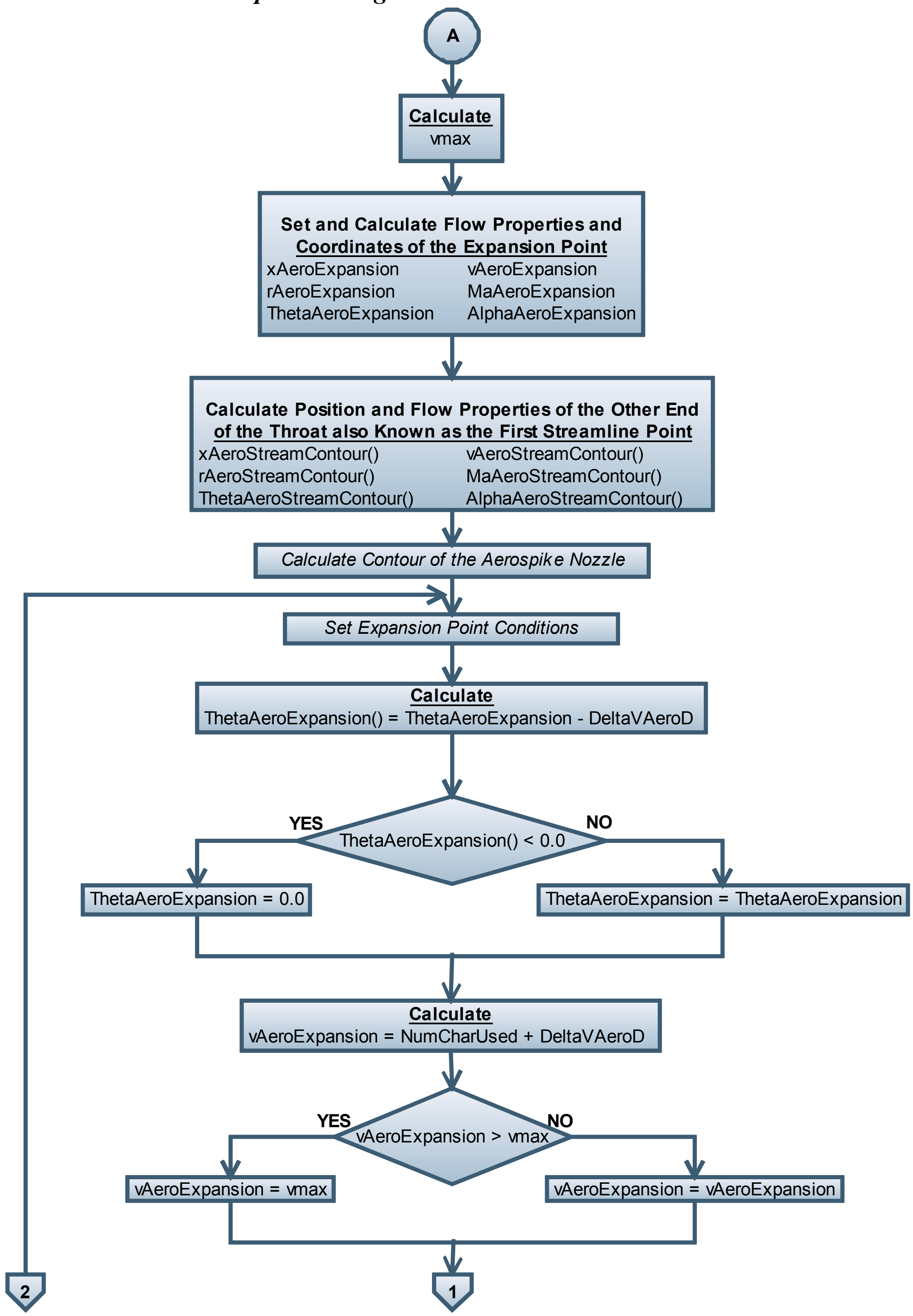
```
%* 22. xAeroExpansion ----------- Array that holds the x-component of the
%*                                expansion point calculated by subroutine
%*                                DentonAerospike
%* 23. xAeroStream -------------- Array that holds the x-component of the
%*                                Internal-External Aerospike nozzle
%* 24. xAeroStreamContour ------- Array that holds the x-component of the
%*                                aerospike nozzle calculated by subroutine
%*                                DentonAerospike
%* 25. xAeroStreamContourTrunc -- Array that holds the x-component of the
%*                                annular aerospike calculated by
%*                                subroutine DentonAerospike
%* 26. xAeroStreamCowl ---------- Array that holds the x-components of the
%*                                cowl of the Internal-External Aerospike
%*                                nozzle
%* 27. xAeroStreamTrunc --------- Array that holds the x-component of the
%*                                truncated Internal-External Aerospike
%*                                nozzle
%* 28. xSADcontour -------------- Array that holds the x-component of the
%*                                annular nozzle contour calculated by
%*                                subroutine AxisymmetricSAD but calculated
%*                                it using the same technique used for the
%*                                2D nozzles
%* 29. xStreamContour ----------- Array that holds the x-component of the
%*                                annular nozzle contour calculated by
%*                                subroutine AxisymmetricSAD using the
%*                                stream function technique
%* 30. zz ----------------------- Indices indicator used in subroutine
%*                                Annular Aerospike
%**************************************************************************
%**************************************************************************
%*                       Start Program
%**************************************************************************
%**************************************************************************

disp(' ');

% Program will ask for the desired Exit Mach Number
Mexit = input('Please enter the desired Exit Mach Number:  ');
disp(' ');

% Program will ask for the ratio of specific heats for the fluid
Gamma = input('Please enter the ratio of specific heats for the fluid:  ');
disp(' ');

% Program will ask if the designer would like to perform a entropy check to
% see if the change in entropy is zero or at least very small to validate
% the isentropic assumption used in the calculation
EntroLoop = 1;

while EntroLoop == 1
    disp('Would you like to perform an entropy check of the flowfield?');
    disp('    1 = Yes');
    disp('    2 = No');
    EntroCheck = input('->  ');
    disp(' ');

    % Program asks for the Rocket Chambers Pressure and Temperature
    %(Stagnation or Total) and specific heat at constant pressure
    if EntroCheck == 1
        Pstag = input...
            ('Please enter the Rocket Chamber Stagnation Pressure:(Pa)  ');
        disp(' ');

        Tstag = input...
           ('Please enter the Rocket Chamber Stagnation Temperature:(K)  ');
        disp(' ');

        cp = input...
  ('Please enter the specific heat at constant pressure of the fluid:(J/kg-K)  ');
        disp(' ');

        Rgas = (cp * (Gamma - 1))/Gamma;
```

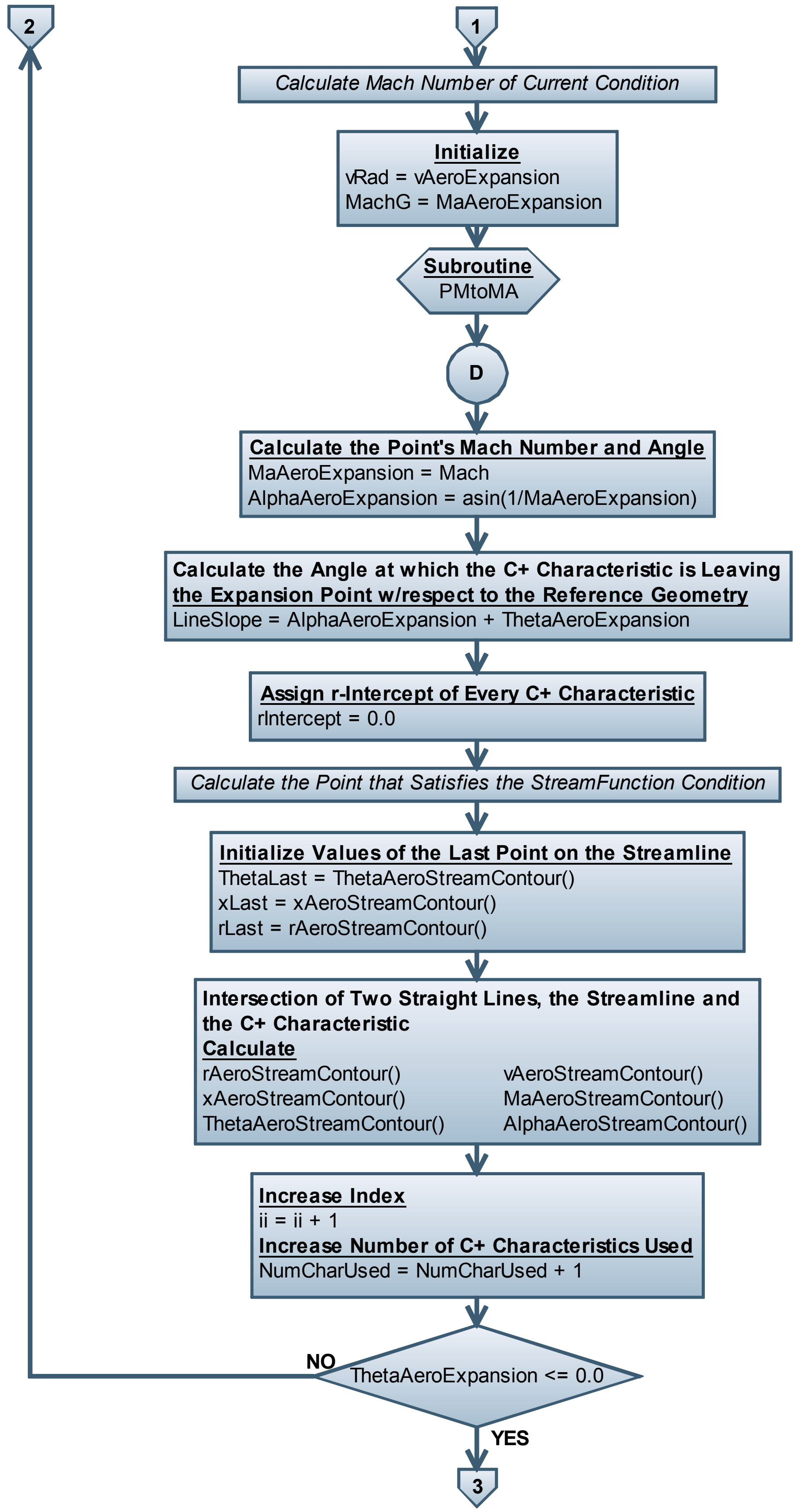
```
        EntroLoop = 0;
    elseif EntroCheck == 2
        EntroLoop = 0;
    else
        disp('You have entered an invalid value');
        disp(' ');
    end
end

% Program will ask for the radius of the throat
disp('Please enter the radius of the throat of the nozzle.');
disp('for a dimensionless nozzle, enter a radius of 1');
rThroat = input('->  ');
disp(' ');

% Program will ask for the Multiplication factor of the throat to determine
% the radius of the circle defining the entrance region
disp('FOR THE AXISYMMETRIC AND INTERNAL-EXTERNAL AEROSPIKE NOZZLE');
disp('Please enter the factor of the throat that will define the');
Beta = input...
      ('radius of the circle for the entrance region of the nozzles:  ');
disp(' ');

% Program will ask if the user would like to calculate the contour of a
% truncated Aerospike nozzle and if so what percent of the idealized length
% would the user like to retain
zz = 0;
kk = 0;
while zz == 0
    disp('Do you want to truncate the Aerospike Nozzle?');
    disp('    1 = Yes');
    disp('    2 = No');
    Truncate = input('->  ');
    if Truncate == 2
        nb = 0.0;
        zz = 1;
    elseif Truncate == 1
        disp(' ');
        Percent = input...
('Please enter the percentage of the idealize length you would like to retain:  ');
        nb = Percent/100;
        zz = 1;
    else
        disp(' ');
        disp('You have entered and invalid response, please retry.');
    end
end

% Program will ask for the desired change in Prandtl-Meyer Expansion Angle
% for the Aerospike design using the streamline technique
disp(' ');
DeltaVAeroD = input...
 ('Please enter the desired change in Prandtl-Meyer Expansion Angle:   ');

% Program will now calculate the Non-Dimensional Axi-symmetric Aerospike
% Nozzle for the desired Exit Mach Number
% NOTE******* This aerospike nozzle is calculated using the streamline
%             technique
DentonAerospike

% Program will now calculate the Non-Dimensional Axi-symmetric
% Internal-External Aerospike nozzle for the desired Exit Mach Number
IEaerospike

% Program will now calculate the Non-Dimensional Aerospike nozzle for the
% desired Exit Mach Number using the Shapiro, Anderson, Denton Method
AxiSymmetricSAD

% Output Plot
if Truncate == 1
    plot(xSADcontour, rSADcontour, 'g.', xStreamContour,...
```

```
        rStreamContour, 'k.', xAeroStreamContour,...
        rAeroStreamContour, 'y.', xAeroStreamContourTrunc,...
        rAeroStreamContourTrunc, 'c.', xAeroExpansion,...
        rAeroExpansion, 'y.', xAeroStream, rAeroStream, 'r.',...
        xAeroStreamCowl, rAeroStreamCowl, 'r.', xAeroStreamTrunc,...
        rAeroStreamTrunc, 'b.');
else
    plot(xSADcontour, rSADcontour, 'g.', xStreamContour,...
        rStreamContour, 'k.', xAeroStreamContour,...
        rAeroStreamContour, 'y.', xAeroExpansion, rAeroExpansion,...
        'y.', xAeroStream, rAeroStream, 'r.', xAeroStreamCowl,...
        rAeroStreamCowl, 'r.');
end

 disp('MaSAD(PointNum)');
 disp(MaSAD(PointNum));
 disp(' ');
 disp('rAeroExpansion');
 disp(rAeroExpansion);
 disp(' ');
 disp('Area Ratio');
 disp((rStreamContour(ll))^2);
 disp(' ');
```

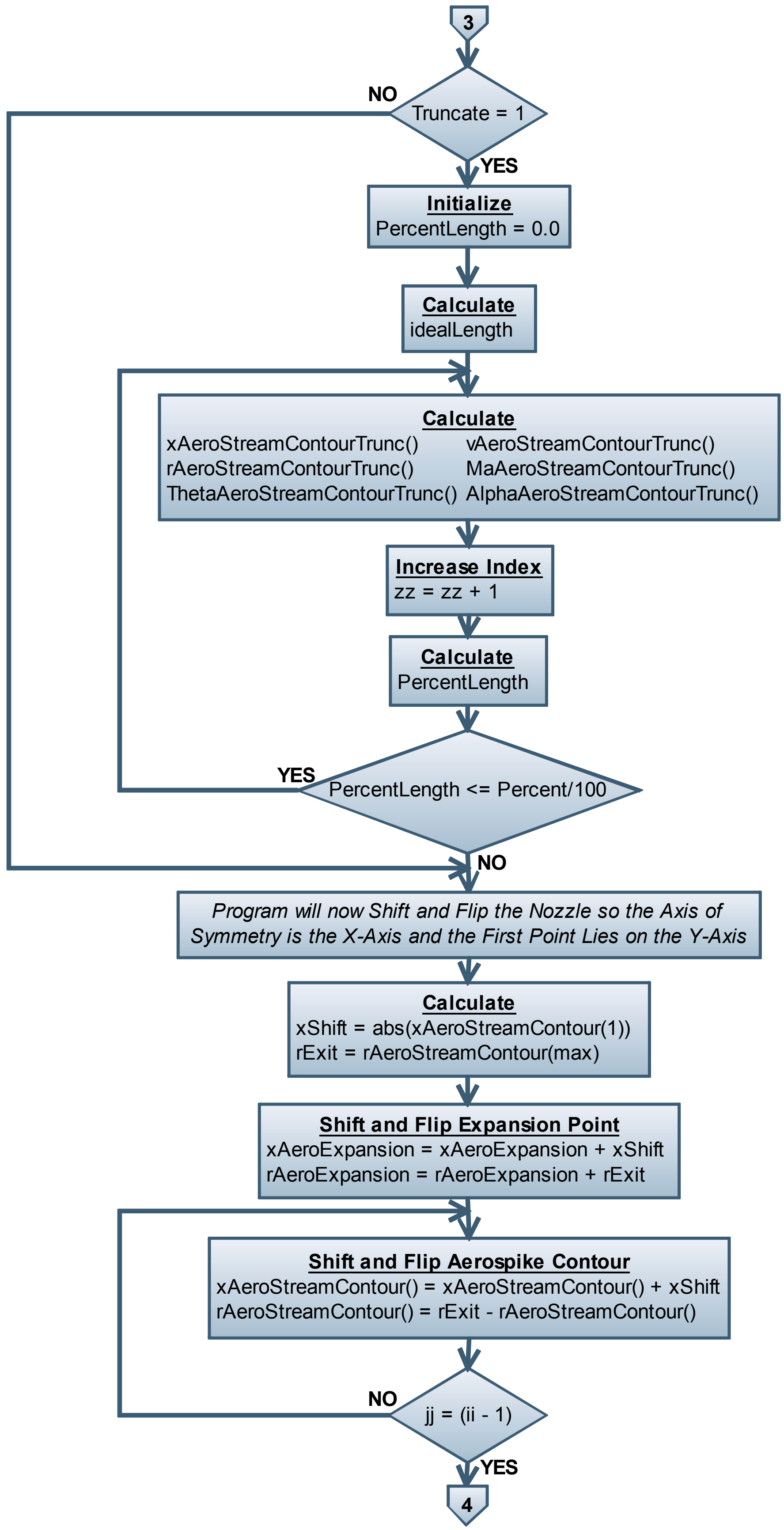
B.2 DentonAerospike.m Program

Figure B.2.1: DentonAerospike.m Program Flowchart





2
1
Calculate Mach Number of Current Condition
Initialize
vRad = vAeroExpansion
MachG = MaAeroExpansion
Subroutine
PMtoMA
D
Calculate the Point's Mach Number and Angle
MaAeroExpansion = Mach
AlphaAeroExpansion = asin(1/MaAeroExpansion)
Calculate the Angle at which the C+ Characteristic is Leaving the Expansion Point w/respect to the Reference Geometry
LineSlope = AlphaAeroExpansion + ThetaAeroExpansion
Assign r-Intercept of Every C+ Characteristic
rIntercept = 0.0
Calculate the Point that Satisfies the StreamFunction Condition
Initialize Values of the Last Point on the Streamline
ThetaLast = ThetaAeroStreamContour()
xLast = xAeroStreamContour()
rLast = rAeroStreamContour()
Intersection of Two Straight Lines, the Streamline and the C+ Characteristic
Calculate
rAeroStreamContour()
xAeroStreamContour()
ThetaAeroStreamContour()
vAeroStreamContour()
MaAeroStreamContour()
AlphaAeroStreamContour()
Increase Index
ii = ii + 1
Increase Number of C+ Characteristics Used
NumCharUsed = NumCharUsed + 1
NO
ThetaAeroExpansion <= 0.0
YES
3


3
Truncate = 1
NO
YES
Initialize
PercentLength = 0.0
Calculate
idealLength
Calculate
xAeroStreamContourTrunc()
rAeroStreamContourTrunc()
ThetaAeroStreamContourTrunc()
vAeroStreamContourTrunc()
MaAeroStreamContourTrunc()
AlphaAeroStreamContourTrunc()
Increase Index
zz = zz + 1
Calculate
PercentLength
YES
PercentLength <= Percent/100
NO
Program will now Shift and Flip the Nozzle so the Axis of Symmetry is the X-Axis and the First Point Lies on the Y-Axis
Calculate
xShift = abs(xAeroStreamContour(1))
rExit = rAeroStreamContour(max)
Shift and Flip Expansion Point
xAeroExpansion = xAeroExpansion + xShift
rAeroExpansion = rAeroExpansion + rExit
Shift and Flip Aerospike Contour
xAeroStreamContour() = xAeroStreamContour() + xShift
rAeroStreamContour() = rExit - rAeroStreamContour()
NO
jj = (ii - 1)
YES
4

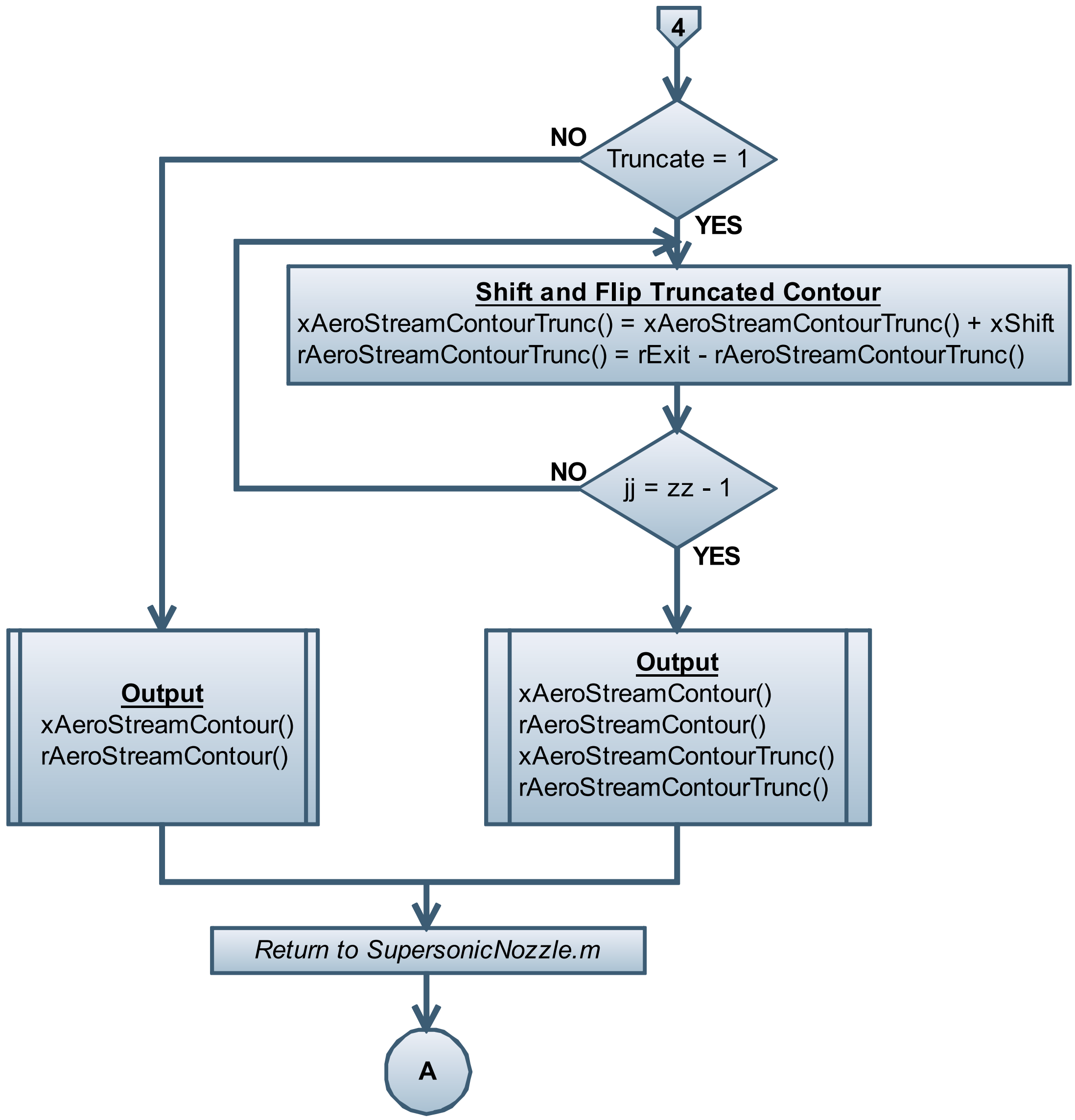

4
Truncate = 1
NO
YES
Shift and Flip Truncated Contour
xAeroStreamContourTrunc() = xAeroStreamContourTrunc() + xShift
rAeroStreamContourTrunc() = rExit - rAeroStreamContourTrunc()
jj = zz - 1
NO
YES
Output
xAeroStreamContour()
rAeroStreamContour()
Output
xAeroStreamContour()
rAeroStreamContour()
xAeroStreamContourTrunc()
rAeroStreamContourTrunc()
Return to SupersonicNozzle.m
A

Figure B.2.2: DentonAerospike.m MatLab Source Code

```
%***************************************************************************
%*          AxiSymmetric Aerospike Nozzle Design MOC
%*              Brandon Denton
%*              RIT Graduate Student
%***************************************************************************
%***************************************************************************
%*       Dictionary of Variables
%***************************************************************************
%*  1. a, b, c, d -------------------- Variables that temporarily hold
%*                                     numbers used to evaluate the point
%*                                     that satisifies the Stream Function
%*  2. A, B -------------------------- Matrices that hold the equations to
%*                                     solve for the position of the point
%*                                     that satisfies the Stream Function
%*  3. AlphaAeroExpansion ------------ The Mach angle of the fluid at the
%*                                     expansion point
%*  4. AlphaAeroStreamContour() ------ Array that holds the Mach angle of
%*                                     the points on the contour of the
%*                                     Aerospike nozzle
%*  5. AlphaAeroStreamContourTrunc() - Array that holds the Mach angle of
%*                                     the points on the contour of the
%*                                     truncated Aerospike nozzle
%*  6. DeltaVAeroD ------------------- The desired change in Prandtl-Meyer
%*                                     Expansion Angle
%*  7. Gamma ------------------------- Ratio of Specific Heats of the
%*                                     working fluid
%*  8. idealLength ------------------- Variable that holds the length of an
%*                                     idealized (100% Length) Aerospike
%*                                     nozzle
%*  9. ii, jj ------------------------ Loop and position indices
%* 10. LineSlope --------------------- The slope of the line in radians
%*                                     that approximates the Mach Line in
%*                                     the flowfield
%* 11. MaAeroExpansion --------------- Mach Number of the fluid at the
%*                                     expansion point
%* 12. MaAeroStreamContour() --------- Array that holds the Mach Number of
%*                                     the points on the contour of the
%*                                     Aerospike nozzle
%* 13. MaAeroStreamContourTrunc() ---- Array that holds the Mach Number of
%*                                     the points on the contour of the
%*                                     Aerospike nozzle
%* 14. MaContinue -------------------- Control variable for the loop that
%*                                     calculates the Aerospike nozzle's
%*                                     contour
%* 15. Mach -------------------------- Solution variable of the Mach Number
%*                                     returned from subroutine PMtoMA
%* 16. MachG ------------------------- Initial guess value of the Mach
%*                                     Number used in the Subroutine PMtoMA
%*                                     to find the point's Mach Number
%* 17. Mexit ------------------------- Desired Exit Mach Number
%* 18. NumCharUsed ------------------- The number of C+ characteristics
%*                                     used in the calculation of the
%*                                     Aerospike's contour
%* 19. PercentLength ----------------- Control variable that is calculated
%*                                     to make sure that the Aerospike
%*                                     nozzle is truncate appropriately
%* 20. PMtoMA ------------------------ Subroutine used to find the Mach
%*                                     Number of the point in the flowfield
%* 21. rAeroExpansion ---------------- The r-component of the position of
%*                                     the expansion point of the Aerospike
%*                                     nozzle
%* 22. rAeroStreamContour() ---------- Array that holds the r-component of
%*                                     the points on the contour of the
%*                                     Aerospike nozzle
%* 23. rAeroStreamContourTrunc() ----- Array that holds the r-component of
%*                                     the points on the contour of the
%*                                     truncated Aerospike nozzle
```

```
%* 24. rExit ------------------------- Variable used to "flip" the
%*                                     Aerospike's contour so that the axis
%*                                     of symmetry of the nozzle is the
%*                                     x-axis
%* 25. rIntercept -------------------- The r-intercept of the line that
%*                                     approximates the Mach Line in the
%*                                     flowfield
%* 26. rLast ------------------------- The variable that holds the
%*                                     r-component of the point that last
%*                                     satisfied the Stream Function
%* 27. rThroat ----------------------- User-defined radius of the throat
%* 28. solution() -------------------- Matrix that holds the solution to
%*                                     the coordinates of the point that
%*                                     satisfies the Stream Function
%* 29. ThetaAeroExpansion ------------ The flow direction of the fluid at
%*                                     the expansion point
%* 30. ThetaAeroStreamContour() ------ Array that holds the flow direction
%*                                     of the point on the contour of the
%*                                     Aerospike nozzle
%* 31. ThetaAeroStreamContourTrunc() - Array that holds the flow direction
%*                                     of the point on the contour of the
%*                                     truncated Aerospike nozzle
%* 32. ThetaLast --------------------- The variable the holds the flow
%*                                     direction of the point that last
%*                                     satisfied the Stream Function
%* 33. Truncate ---------------------- Control variable that controls
%*                                     whether or not the user wanted to
%*                                     truncate the Aerospike nozzle
%* 34. vAeroStreamContour() ---------- Array that holds the Prandtl-Meyer
%*                                     expansion angle of the point on the
%*                                     contour of the Aerospike nozzle
%* 35. vAeroStreamContourTrunc() ----- Array that holds the Prandtl-Meyer
%*                                     expansion angle of the point on the
%*                                     contour of the Aerospike nozzle
%* 36. vmax -------------------------- Prandtl-Meyer Expansion Angle for
%*                                     the desired Mach Number
%* 37. vRad -------------------------- Variable used in subroutine PMtoMA
%*                                     to calculate the Mach number at the
%*                                     point in question of the flow
%* 38. xAeroExpansion ---------------- The x-component of the position of
%*                                     the expansion point of the Aerospike
%*                                     nozzle
%* 39. xAeroStreamContour() ---------- Array that holds the x-component of
%*                                     the points on the contour of the
%*                                     Aerospike nozzle
%* 40. xAeroStreamContourTrunc() ----- Array that holds the x-component of
%*                                     the points on the contour of the
%*                                     truncated Aerospike nozzle
%* 41. xLast ------------------------- The variable that holds the
%*                                     x-component of the point that last
%*                                     satisfied the Stream Function
%* 42. xShift ------------------------ Variable used to shift the Aerospike
%*                                     contour so that the first point is
%*                                     on the r-axis
%* 43. zz ---------------------------- Indice variable
%***************************************************************************
%***************************************************************************
%*      Start Program
%***************************************************************************
%***************************************************************************
format long

disp(' ');

% Calculate the Maximum Prandtl-Meyer Expansion Angle
vmax = (sqrt((Gamma + 1)/(Gamma - 1)))*...
       atan(sqrt(((Gamma - 1)/(Gamma + 1)) *...
       ((Mexit^2) - 1))) - atan((sqrt((Mexit^2) - 1)));

% Calculate the Position of the end points of the sonic line
xAeroStreamContour(1,1) = -rThroat * cos((pi/2) - vmax);
rAeroStreamContour(1,1) = rThroat * sin((pi/2) - vmax);
```

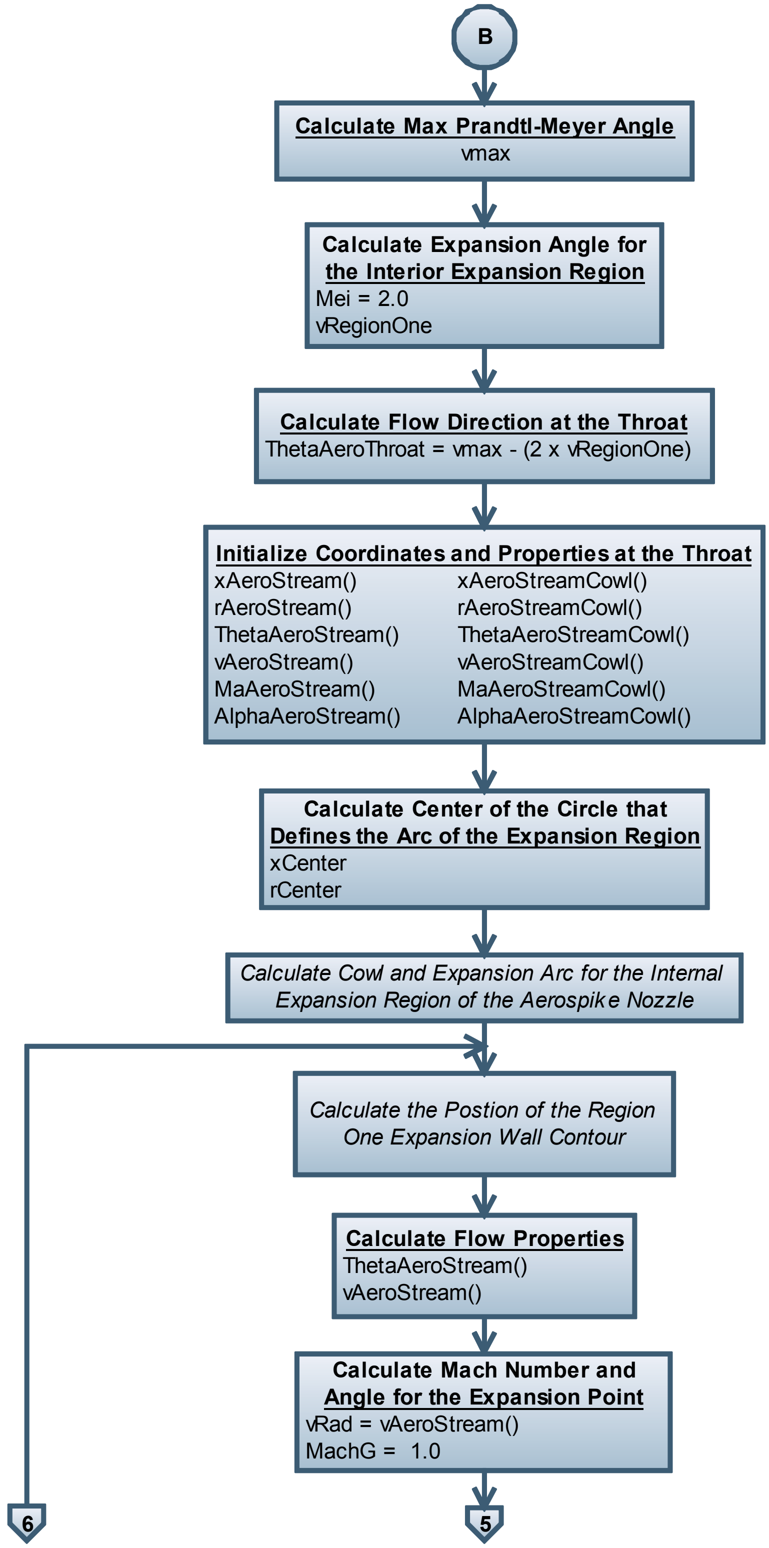
```
ThetaAeroStreamContour(1,1) = vmax;
vAeroStreamContour(1,1) = 0.0;
MaAeroStreamContour(1,1) = 1.0;
AlphaAeroStreamContour(1,1) = asin(1/MaAeroStreamContour(1,1));

if EntroCheck == 1
% Calculate the Pressure, Temperature and change in Entropy
% at the point    log = natural log
    PressExtAero(1,1) = Pstag * ((1 + (((Gamma - 1)/2) *...
                ((MaAeroStreamContour(1,1)) ^ 2))) ^ (-Gamma/(Gamma - 1)));

    TempExtAero(1,1) = Tstag * (1 + (((Gamma - 1)/2) *...
                ((MaAeroStreamContour(1,1)) ^ 2))) ^ (-1);

    DeltaSExtAero(1,1) = cp * log(TempExtAero(1,1)/Tstag) - Rgas *...
                          log(PressExtAero(1,1)/Pstag);
end

% Position of the Expansion point
xAeroExpansion = 0.0;
rAeroExpansion = 0.0;
ThetaAeroExpansion = vmax;
vAeroExpansion = 0.0;
MaAeroExpansion = 1.0;
AlphaAeroExpansion = asin(1/MaAeroExpansion);

% Calculate the Aerospike contour by assuming that the C+ Characteristice
% eminating from the expansion point is a straight line and terminates
% when it satisfies the streamline condition (becomes the point on the
% contour)
MaContinue = 1;
NumCharUsed = 1;
ii = 2;

while MaContinue == 1
    % Set expansion point conditions
    ThetaAeroExpansion = ThetaAeroExpansion - DeltaVAeroD;
    if ThetaAeroExpansion < 0.0
        ThetaAeroExpansion = 0.0;
    else
        ThetaAeroExpansion = ThetaAeroExpansion;
    end

    vAeroExpansion = (NumCharUsed * DeltaVAeroD);
    if vAeroExpansion > vmax
        vAeroExpansion = vmax;
    else
        vAeroExpansion = vAeroExpansion;
    end

    %Calculate the Mach Number at the current point
    vRad = vAeroExpansion;

    MachG = MaAeroExpansion;
    PMtoMA %calls subprogram to find the Mach Number

    MaAeroExpansion = Mach;
    AlphaAeroExpansion = asin(1/MaAeroExpansion);

    LineSlope = AlphaAeroExpansion + ThetaAeroExpansion;

    rIntercept = 0.0;  % Center of the expansion wave is located at (0,0)

    % Calculate the point that satisfies the Stream Function condition at
    % the last expansion point on the contour

    % Initiate values of the last point on the streamline
    ThetaLast = ThetaAeroStreamContour((ii-1),1);
    xLast = xAeroStreamContour((ii-1),1);
    rLast = rAeroStreamContour((ii-1),1);

    % Calculate the position of the point the satisfies the streamline
```

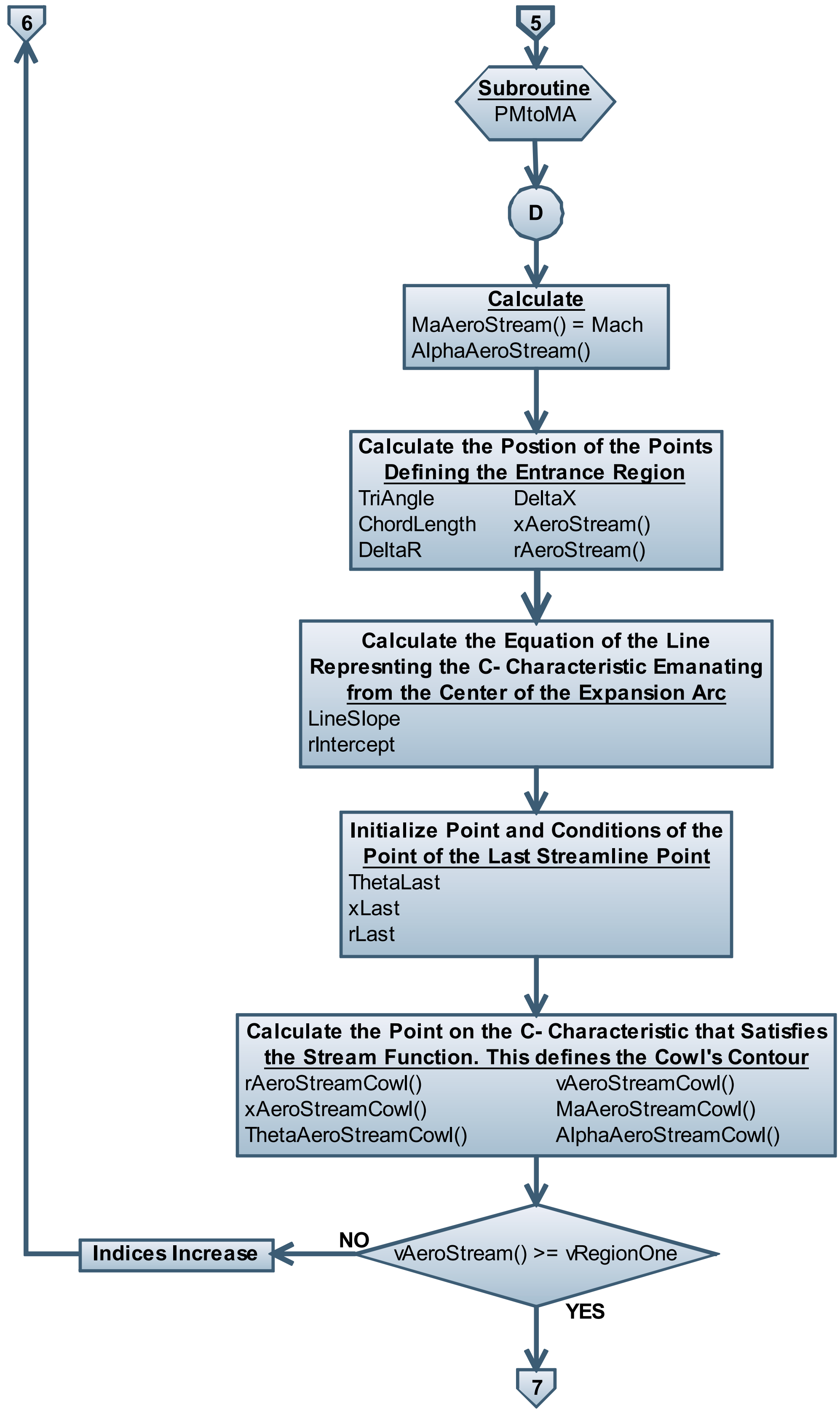
```
    % condition
    a = -tan(ThetaLast);
    b = -tan(LineSlope);
    c = rLast - tan(ThetaLast) * xLast;
    d = rIntercept;

    A = [1 a; 1 b];
    B = [c; d];

    solution = A\B;

    rAeroStreamContour(ii,1) = solution(1,1);
    xAeroStreamContour(ii,1) = solution(2,1);
    ThetaAeroStreamContour(ii,1) = ThetaAeroExpansion;
    vAeroStreamContour(ii,1) = vAeroExpansion;
    MaAeroStreamContour(ii,1) = MaAeroExpansion;
    AlphaAeroStreamContour(ii,1) = asin(1/MaAeroExpansion);

    if EntroCheck == 1
    % Calculate the Pressure, Temperature and change in Entropy
    % at the point    log = natural log
        PressExtAero(ii,1) = Pstag * ((1 + (((Gamma - 1)/2) *...
               ((MaAeroStreamContour(ii,1)) ^ 2))) ^ (-Gamma/(Gamma - 1)));

        TempExtAero(ii,1) = Tstag * (1 + (((Gamma - 1)/2) *...
                    ((MaAeroStreamContour(ii,1)) ^ 2))) ^ (-1);

        DeltaSExtAero(ii,1) = cp * log(TempExtAero(ii,1)/Tstag) - Rgas *...
                   log(PressExtAero(ii,1)/Pstag);
    end

    ii = ii + 1;    %Increase the index
    NumCharUsed = NumCharUsed + 1;

    if ThetaAeroExpansion <= 0.0
        MaContinue = 0;
    else
        MaContinue = 1;
    end
end

if EntroCheck == 1
% Calculate the Pressure, Temperature and change in Entropy
% at the point    log = natural log
    PressExtAero(ii,1) = Pstag * ((1 + (((Gamma - 1)/2) *...
                 ((MaAeroExpansion) ^ 2))) ^ (-Gamma/(Gamma - 1)));

    TempExtAero(ii,1) = Tstag * (1 + (((Gamma - 1)/2) *...
                ((MaAeroExpansion) ^ 2))) ^ (-1);

    DeltaSExtAero(ii,1) = cp * log(TempExtAero(ii,1)/Tstag) - Rgas *...
                         log(PressExtAero(ii,1)/Pstag);
end

if Truncate == 1
    zz = 1;
    PercentLength = 0.0;

    idealLength = xAeroStreamContour((ii-1),1);

    while PercentLength <= (Percent/100)
        xAeroStreamContourTrunc(zz,1) = xAeroStreamContour(zz,1);
        rAeroStreamContourTrunc(zz,1) = rAeroStreamContour(zz,1);
        ThetaAeroStreamContourTrunc(zz,1) = ThetaAeroStreamContour(zz,1);
        vAeroStreamContourTrunc(zz,1) = vAeroStreamContour(zz,1);
        MaAeroStreamContourTrunc(zz,1) = MaAeroStreamContour(zz,1);
        AlphaAeroStreamContourTrunc(zz,1) = AlphaAeroStreamContour(zz,1);

        zz = zz + 1;

        PercentLength = abs(xAeroStreamContourTrunc((zz-1),1))/idealLength;
    end
```

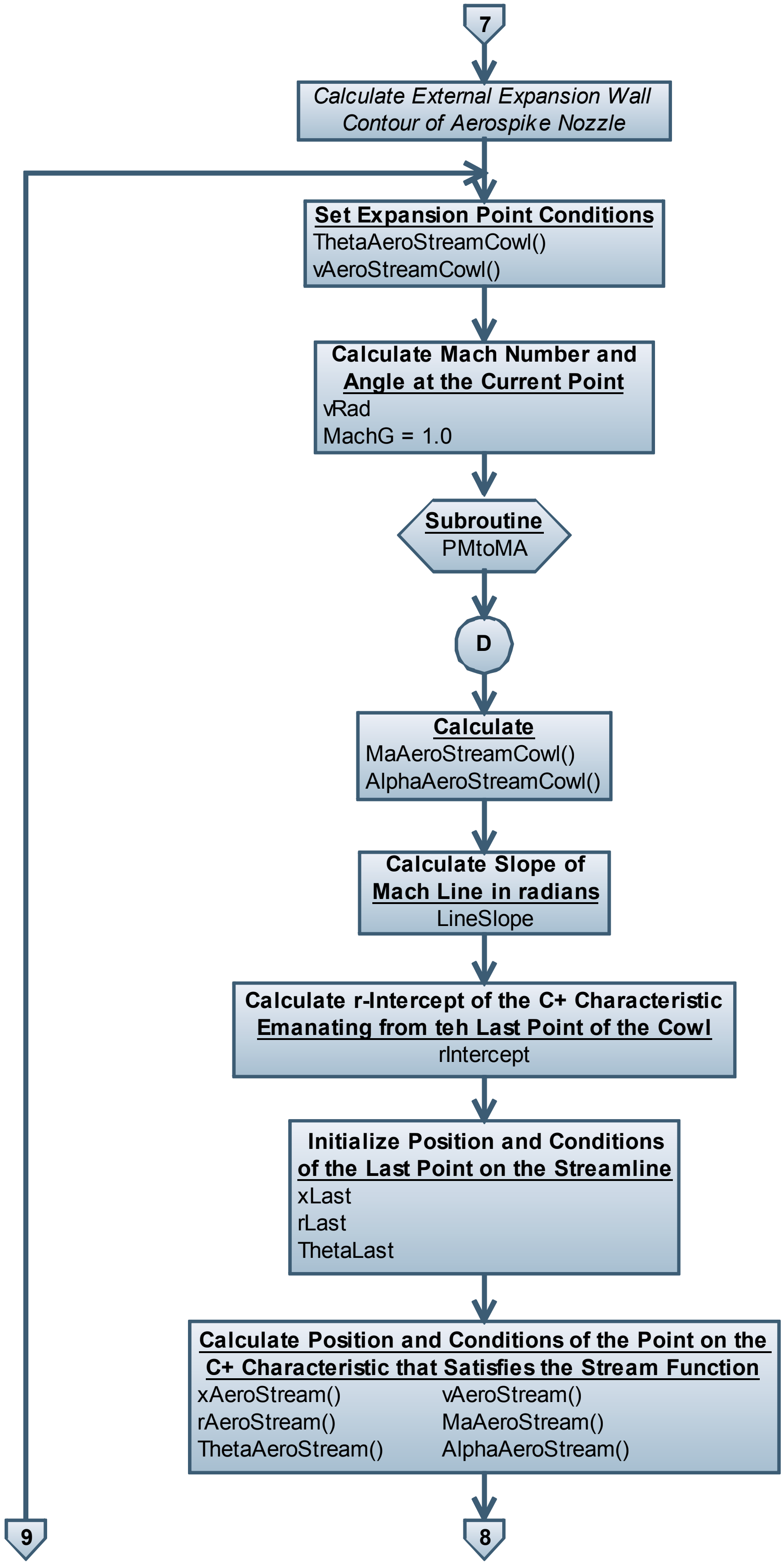
```
end

% The program will now "flip" the contour so that the axisymmetric line
% aligns with r=0 and the expansion point will occur at (0,rexit). it will
% also shift the nozzle to that the beginning of the contour will exit on
% the x=0 axis
xShift = abs(xAeroStreamContour(1,1));
rExit = rAeroStreamContour((ii-1),1);

xAeroExpansion = xAeroExpansion + xShift;
rAeroExpansion = rAeroExpansion + rAeroStreamContour((ii-1),1);

for jj=1:1:(ii-1)
    xAeroStreamContour(jj,1) = xAeroStreamContour(jj,1) + xShift;
    rAeroStreamContour(jj,1) = rExit - rAeroStreamContour(jj,1);
end

% This section only calculates if the Aerospike is truncated
if Truncate == 1
    for jj=1:1:(zz-1)
        xAeroStreamContourTrunc(jj,1) = xAeroStreamContourTrunc(jj,1)...
                                        + xShift;
        rAeroStreamContourTrunc(jj,1) = rExit -...
                                        rAeroStreamContourTrunc(jj,1);
    end
end
```

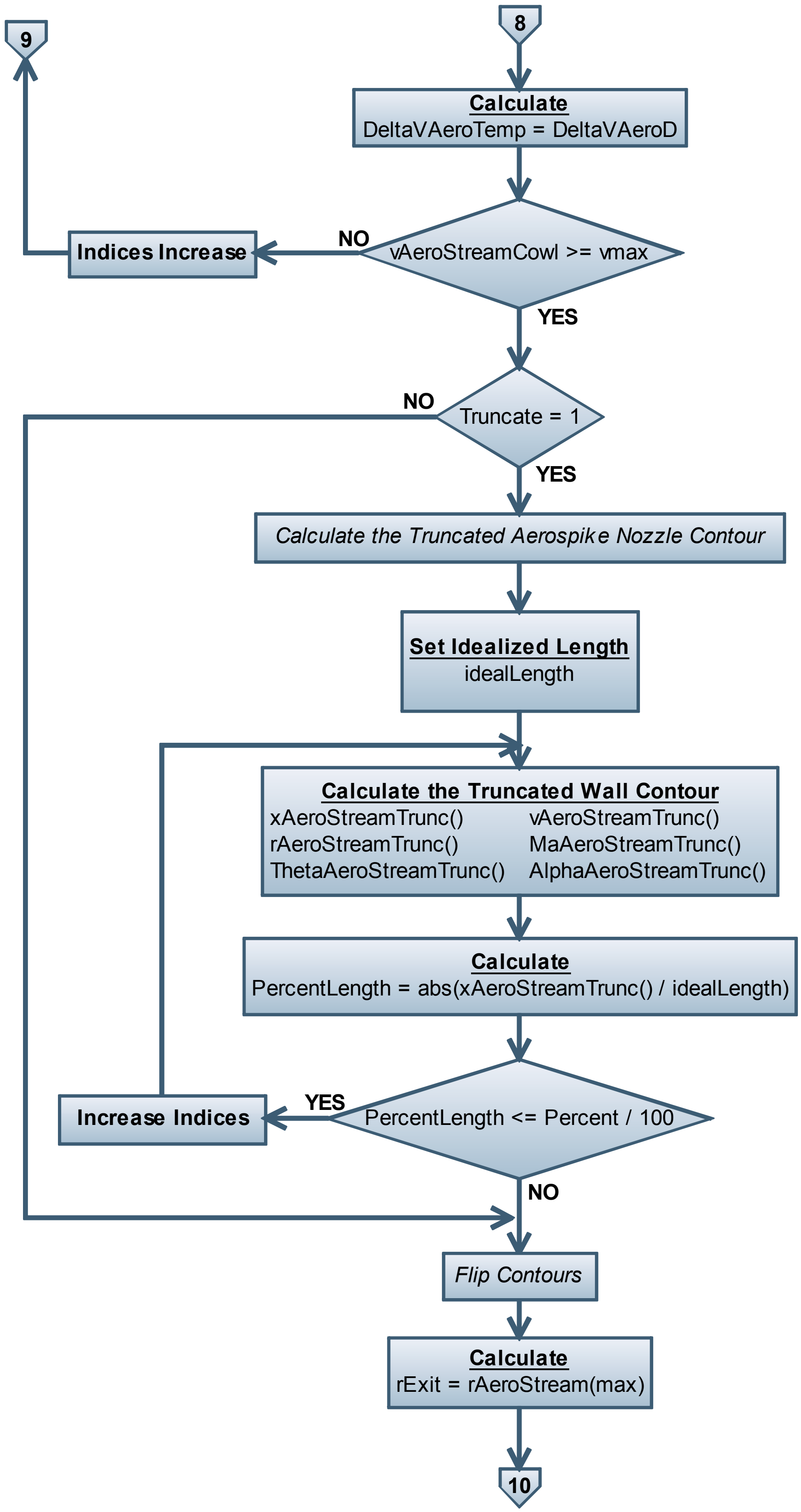
B.3 IEaerospike.m Program

Figure B.3.1: IEaerospike.m Program Flowchart




6
5
Subroutine
PMtoMA
D
Calculate
MaAeroStream() = Mach
AlphaAeroStream()
Calculate the Postion of the Points
Defining the Entrance Region
TriAngle
DeltaX
ChordLength
xAeroStream()
DeltaR
rAeroStream()
Calculate the Equation of the Line
Represnting the C- Characteristic Emanating
from the Center of the Expansion Arc
LineSlope
rIntercept
Initialize Point and Conditions of the
Point of the Last Streamline Point
ThetaLast
xLast
rLast
Calculate the Point on the C- Characteristic that Satisfies
the Stream Function. This defines the Cowl's Contour
rAeroStreamCowl()
vAeroStreamCowl()
xAeroStreamCowl()
MaAeroStreamCowl()
ThetaAeroStreamCowl()
AlphaAeroStreamCowl()
NO
vAeroStream() >= vRegionOne
Indices Increase
YES
7


7
Calculate External Expansion Wall Contour of Aerospike Nozzle
Set Expansion Point Conditions
ThetaAeroStreamCowl()
vAeroStreamCowl()
Calculate Mach Number and Angle at the Current Point
vRad
MachG = 1.0
Subroutine
PMtoMA
D
Calculate
MaAeroStreamCowl()
AlphaAeroStreamCowl()
Calculate Slope of Mach Line in radians
LineSlope
Calculate r-Intercept of the C+ Characteristic Emanating from teh Last Point of the Cowl
rIntercept
Initialize Position and Conditions of the Last Point on the Streamline
xLast
rLast
ThetaLast
Calculate Position and Conditions of the Point on the C+ Characteristic that Satisfies the Stream Function
xAeroStream()
rAeroStream()
ThetaAeroStream()
vAeroStream()
MaAeroStream()
AlphaAeroStream()
9
8


8
9
Calculate
DeltaVAeroTemp = DeltaVAeroD
NO
vAeroStreamCowl >= vmax
Indices Increase
YES
NO
Truncate = 1
YES
Calculate the Truncated Aerospike Nozzle Contour
Set Idealized Length
idealLength
Calculate the Truncated Wall Contour
xAeroStreamTrunc()
vAeroStreamTrunc()
rAeroStreamTrunc()
MaAeroStreamTrunc()
ThetaAeroStreamTrunc()
AlphaAeroStreamTrunc()
Calculate
PercentLength = abs(xAeroStreamTrunc() / idealLength)
YES
PercentLength <= Percent / 100
Increase Indices
NO
Flip Contours
Calculate
rExit = rAeroStream(max)
10

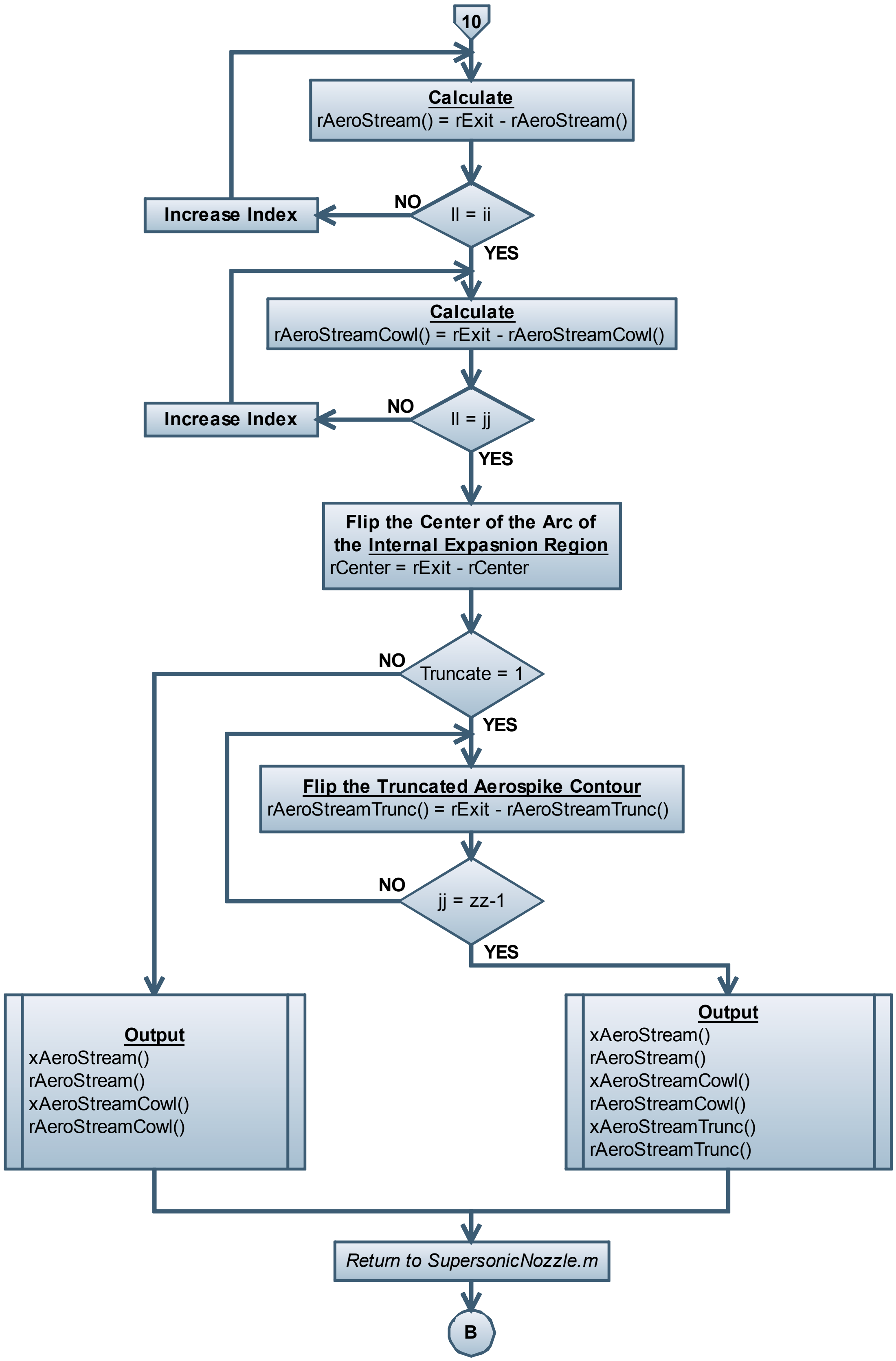

10
Calculate
rAeroStream() = rExit - rAeroStream()
ll = ii
NO
Increase Index
YES
Calculate
rAeroStreamCowl() = rExit - rAeroStreamCowl()
ll = jj
NO
Increase Index
YES
Flip the Center of the Arc of the Internal Expasnion Region
rCenter = rExit - rCenter
Truncate = 1
NO
YES
Flip the Truncated Aerospike Contour
rAeroStreamTrunc() = rExit - rAeroStreamTrunc()
jj = zz-1
NO
YES
Output
xAeroStream()
rAeroStream()
xAeroStreamCowl()
rAeroStreamCowl()
Output
xAeroStream()
rAeroStream()
xAeroStreamCowl()
rAeroStreamCowl()
xAeroStreamTrunc()
rAeroStreamTrunc()
Return to SupersonicNozzle.m
B

Figure B.3.2: IEaerospike.m MatLab Source Code

```
%**************************************************************************
%*               Internal-External Expansion Aerospike
%*                   Brandon Denton
%*                   RIT Graduate Student
%*                   January 31, 2007
%**************************************************************************
%**************************************************************************
%*       Dictionary of Terms
%**************************************************************************
%*  1. a, b, c, d ------------ Variables that hold temporary values in the
%*                             calculation of the point that satisfy the
%*                             Stream Function condition using matrix
%*                             algebra
%*  2. A, B ------------------ Matrices that hold the temporary values in
%*                             the calculation of the points that satisfy
%*                             the Stream Function condition using matrix
%*                             algebra
%*  3. AlphaAeroStream()------ Array that holds the Mach angle of the
%*                             nozzle's wall contour
%*  4. AlphaAeroStreamCowl()-- Array that holds the Mach angle of the
%*                             nozzle's cowl wall contour
%*  5. AlphaAeroStreamTrunc()- Array that holds the Mach angle of the
%*                             nozzle's truncated wall contour
%*  6. Beta ------------------ Factor of the throat that defines that
%*                             radius of the arc that defines the first
%*                             expansion region of the nozzle
%*  7. ChordLength ----------- Length of the chord of the arc between points
%*                             used to calculate the position of points on
%*                             the expansion arc of region one
%*  8. DeltaR ---------------- The change in r-component relative to the
%*                             first expansion arc point of the throat
%*                             defining the expansion of region one
%*  9. DeltaVAeroD ----------- User-defined change in Prandtl-Meyer
%*                             expansion angle
%* 10. DeltaVAeroTemp -------- Variable that temporary changes and stores
%*                             DeltaVAeroD in order to calculate the first
%*                             C+ characteristic and flow condition in
%*                             region two
%* 11. DeltaX ---------------- The change in the x-component relative to
%*                             the first expansion arc point of the throat
%*                             defining the expansion of region one
%* 12. EndCheck -------------- Variable that controls the loop to make sure
%*                             the the end guess of the streamline
%*                             calculation is larger than the actual
%*                             streamline solution
%* 13. Gamma ----------------- Ratio of Specific Heats of the working fluid
%* 14. idealLength ----------- The total length of the 100% aerospike
%*                             nozzle
%* 15. ii, jj, zz ------------ Loop and position indices
%* 16. LineSlope ------------- Variable that holds the radian angle of the
%*                             Mach line accelerating the flow
%* 17. MaAeroStream() -------- Array that holds the Mach number of the
%*                             nozzle's wall contour
%* 18. MaAeroStreamCowl()----- Array that holds the Mach number of the
%*                             nozzle's cowl wall contour
%* 19. MaAeroStreamTrunc()---- Array that holds the Mach number of the
%*                             nozzle's truncated wall contour
%* 20. MaContinue ------------ Variable that controls the loop that
%*                             calculates the streamline in the second
%*                             expansion region
%* 21. Mach ------------------ Mach Number solution for the point returned
%*                             by subroutine PMtoMA
%* 22. MachG ----------------- Initial guess Mach number used by the
%*                             subroutine PMtoMA to find the points
%*                             corresponding Mach Number
%* 23. Mei ------------------- Exit Mach number of the internal expansion
%*                             of the aerospike nozzle
%* 24. Mexit ----------------- Desired exit Mach number of nozzle
```

```
%* 25. NumChar --------------- Number of characteristics used in the first
%*                             expansion region
%* 26. Percent --------------- Variable that holds the User-Defined desired
%*                             retained length of the aerospike nozzle
%* 27. PercentLength --------- Variable that temporarily holds the
%*                             calculated length percentage of the
%*                             truncated nozzle with respect to the ideal
%*                             (100%) nozzle
%* 28. rAeroStream() --------- Array that holds the r-component of the
%*                             nozzle's wall contour
%* 29. rAeroStreamCowl()------ Array that holds the r-component of the
%*                             nozzle's cowl wall contour
%* 30. rAeroStreamTrunc()----- Array that holds the r-component of the
%*                             nozzle's truncated wall contour
%* 31. rCalc ----------------- Calculated r-component guess value that
%*                             satisfies the streamline condition
%* 32. rCenter --------------- r-coordinate of the center of the circle
%*                             defining the entrance region of the internal
%*                             expansion region of the aerospike nozzle
%* 33. rExit ----------------- Exit radius of the nozzle. Also used in
%*                             flipping the nozzle so that the axisymmetric
%*                             line is on the x-axis
%* 34. rLast ----------------- Variable that temporarily holds the
%*                             r-component of the last calculated point
%*                             that satisfies the streamline condition
%* 35. rThroat --------------- User-defined radius of the throat
%* 36. solution()------------- Matrix that holds the solution coordinates
%*                             of the points that satisfy the Stream
%*                             Function condition
%* 37. ThetaAeroStream()------ Array that holds the values for the flow
%*                             direction on the nozzle's wall contour
%* 38. ThetaAeroStreamCowl()-- Array that holds the values for the flow
%*                             direction for the nozzle's cowl contour
%* 39. ThetaAeroStreamTrunc()- Array that holds the flow direction values
%*                             of the nozzle's truncated wall contour
%* 40. ThetaAeroThroat ------- Angle the sonic line makes with the x-axis
%* 41. ThetaLast ------------- Variable that temporarily holds the flow
%*                             direction of the last point that satisfied
%*                             the streamline condition
%* 42. TriAngle -------------- The angle the chord of the expansion arc in
%*                             region one makes with the x-axis
%* 43. Truncate -------------- Control variable that tells the program
%*                             whether or not to calculate a truncated
%*                             version of the aerospike nozzle
%* 44. vmax ------------------ Prandtl-Meyer Expansion angle for the
%*                             desired exit Mach Number
%* 45. vRad ------------------ Variable that hold the Prandtl-Meyer
%*                             expansion angle of the point used by the
%*                             subroutine PMtoMA to find its corresponding
%*                             Mach number. In Radians
%* 46. vRegionOne ------------ Maximum Pradtl-Meyer Expansion angle allowed
%*                             in the internal expansion region of the
%*                             aerospike nozzle
%* 47. vRegionOneCheck ------- Dummy variable that defines the beginning
%*                             and end of the internal expansion region
%*                             calculation loop of the aerospike nozzle
%* 48. vAeroStream() --------- Array that holds the values of the expansion
%*                             angle for the nozzle's wall contour
%* 49. vAeroStreamCowl()----- Array that holds the values of the expansion
%*                             angle for the nozzle's cowl contour
%* 50. vAeroStreamTrunc()----- Array that holds the flow direction values
%*                             for the nozzle's truncated wall contour
%* 51. xAeroStream() --------- Array that holds the x-components of the
%*                             nozzle's wall contour
%* 52. xAeroStreamCowl()------ Array that holds the x-componen fo the
%*                             nozzle's cowl wall contour
%* 53. xAeroStreamTrunc()----- Array that holds the x-component of the
%*                             nozzle's truncated wall contour
%* 54. xCenter --------------- x-coordinate of the center of the circle
%*                             defining the entrance region of the internal
%*                             expansion region of the aerospike nozzle
```

```
%* 55. xLast ----------------- Variable that temporarily holds the
%*                             x-component of the last point that satisfies
%*                             the streamline condition
%***************************************************************************
%***************************************************************************
%*          START PROGRAM
%***************************************************************************
%***************************************************************************
format long

% Calculate the Maximum Prandtl-Meyer expansion angle for the desired exit
% Mach Number
vmax = (sqrt((Gamma + 1)/(Gamma - 1)))*...
       atan(sqrt(((Gamma - 1)/(Gamma + 1)) *...
       ((Mexit^2) - 1))) - atan((sqrt((Mexit^2) - 1)));

% Calculate the expansion angle for the first expansion region. I have
% chosen 25% of the expansion to take place in this region. For Maximum
% Prandtl-Meyer expansion angles greater than 45 degrees (pi/2 radians) the
% remainder of the expansion angle less pi/2 radians will be used in the
% first expansion region.
Mei = 2.0;
vRegionOne = (sqrt((Gamma + 1)/(Gamma - 1)))*...
              atan(sqrt(((Gamma - 1)/(Gamma + 1)) *...
              ((Mei^2) - 1))) - atan((sqrt((Mei^2) - 1)));

% Calculate the flow direction at the throat
ThetaAeroThroat = vmax - (2 * vRegionOne);

% Initialize the condition at both ends of the throat.
% Farther from axisymmetric line end
xAeroStreamCowl(1,1) = 0.0;
rAeroStreamCowl(1,1) = 0.0;
ThetaAeroStreamCowl(1,1) = ThetaAeroThroat;
vAeroStreamCowl(1,1) = 0.0;
MaAeroStreamCowl(1,1) = 1.0;
AlphaAeroStreamCowl(1,1) = asin(1/MaAeroStreamCowl(1,1));

if EntroCheck == 1
% Calculate the Pressure, Temperature and change in Entropy
% at the point    log = natural log
    PressAeroStreamCowl(1,1) = Pstag * ((1 + (((Gamma - 1)/2) *...
                   ((MaAeroStreamCowl(1,1)) ^ 2))) ^ (-Gamma/(Gamma - 1)));

    TempAeroStreamCowl(1,1) = Tstag * (1 + (((Gamma - 1)/2) *...
                    ((MaAeroStreamCowl(1,1)) ^ 2))) ^ (-1);

    DeltaSAeroStreamCowl(1,1) = cp * log(TempAeroStreamCowl(1,1)/Tstag)...
                     - Rgas * log(PressAeroStreamCowl(1,1)/Pstag);
end

% Axisymmetric line closer end
ThetaAeroStream(1,1) = ThetaAeroThroat;
vAeroStream(1,1) = 0.0;
MaAeroStream(1,1) = 1.0;
AlphaAeroStream(1,1) = asin(1/MaAeroStream(1,1));

if EntroCheck == 1
% Calculate the Pressure, Temperature and change in Entropy
% at the point    log = natural log
    PressAeroStream(1,1) = Pstag * ((1 + (((Gamma - 1)/2) *...
                     ((MaAeroStream(1,1)) ^ 2))) ^ (-Gamma/(Gamma - 1)));

    TempAeroStream(1,1) = Tstag * (1 + (((Gamma - 1)/2) *...
                    ((MaAeroStream(1,1)) ^ 2))) ^ (-1);

    DeltaSAeroStream(1,1) = cp * log(TempAeroStream(1,1)/Tstag) - Rgas *...
                   log(PressAeroStream(1,1)/Pstag);
end
```

```
if ThetaAeroThroat < 0.0
    xAeroStream(1,1) = rThroat * sin(ThetaAeroThroat);
    rAeroStream(1,1) = rThroat * cos(ThetaAeroThroat);
elseif ThetaAeroThroat > 0.0
    xAeroStream(1,1) = -rThroat * sin(ThetaAeroThroat);
    rAeroStream(1,1) = rThroat * cos(ThetaAeroThroat);
else
       xAeroStream(1,1) = 0.0;
       rAeroStream(1,1) = rThroat;
end



% Calculate the center of the circle that defines the arc of the expansion
% region
if ThetaAeroThroat < 0.0
    xCenter = (rThroat + (Beta * rThroat)) * sin(ThetaAeroThroat);
    rCenter = (rThroat + (Beta * rThroat)) * cos(ThetaAeroThroat);
elseif ThetaAeroThroat > 0.0
    xCenter = -(rThroat + (Beta * rThroat)) * sin(ThetaAeroThroat);
    rCenter = (rThroat + (Beta * rThroat)) * cos(ThetaAeroThroat);
else
    xCenter = 0.0;
    rCenter = rThroat + (Beta * rThroat);
end

% Initialize loop and indices conditions
ii = 1;
jj = 1;
vRegionOneCheck = 1;

% Loop that calculates point values and streamline points for the cowl wall
% contour
while vRegionOneCheck == 1
    ii = ii + 1;    %Increase indices every loop iteration
    jj = jj + 1;

    % Calculate the position of the Region One expansion wall contour
    ThetaAeroStream(ii,1) = ThetaAeroStream((ii-1),1) + DeltaVAeroD;
    vAeroStream(ii,1) = vAeroStream((ii-1),1) + DeltaVAeroD;

    if vAeroStream(ii,1) > vRegionOne
        vAeroStream(ii,1) = vRegionOne;
        ThetaAeroStream(ii,1) = ThetaAeroThroat + vRegionOne;
    end

    %Calculate the Mach Number of the current point
    vRad = vAeroStream(ii,1);
    MachG = 1.0;    % Initial Guess of the point's Mach Number
    PMtoMA          % Calls subroutine to calculate the point's Mach Number

    MaAeroStream(ii,1) = Mach;
    AlphaAeroStream(ii,1) = asin(1/MaAeroStream(ii,1));

    if EntroCheck == 1
        % Calculate the Pressure, Temperature and change in Entropy
        % at the point    log = natural log
           PressAeroStream(ii,1) = Pstag * ((1 + (((Gamma - 1)/2) *...
                      ((MaAeroStream(ii,1)) ^ 2))) ^ (-Gamma/(Gamma - 1)));

           TempAeroStream(ii,1) = Tstag * (1 + (((Gamma - 1)/2) *...
                      ((MaAeroStream(ii,1)) ^ 2))) ^ (-1);

           DeltaSAeroStream(ii,1) = cp * log(TempAeroStream(ii,1)/Tstag)...
                        - Rgas * log(PressAeroStream(ii,1)/Pstag);
    end

    % Calculate the position of the points defining the entrance region
    if ThetaAeroThroat < 0.0
        TriAngle = ((pi/2) - ThetaAeroThroat) - ((pi -...
                     vAeroStream(ii,1)) / 2);
    elseif ThetaAeroThroat > 0.0
```

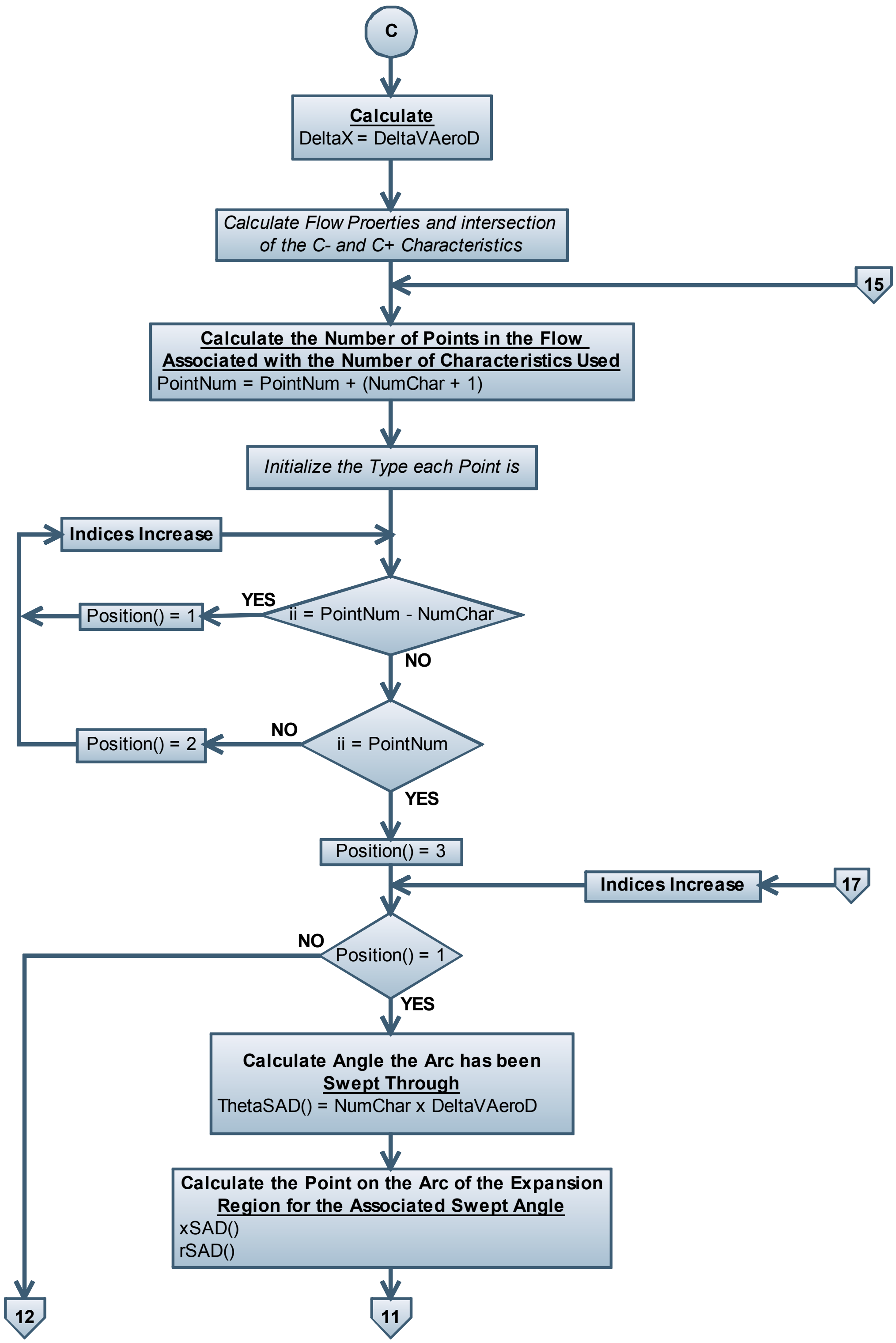
```
            TriAngle = ((pi/2) + ThetaAeroThroat) - ((pi -...
                    vAeroStream(ii,1)) / 2);
    else
        TriAngle = (pi/2) - ((pi - vAeroStream(ii,1)) / 2);
    end

    ChordLength = sqrt(2 * ((Beta * rThroat)^2) *...
                      (1 - cos(vAeroStream(ii,1))));

    DeltaR = ChordLength * sin(TriAngle);
       DeltaX = ChordLength * cos(TriAngle);

       xAeroStream(ii,1) = xAeroStream(1,1) + DeltaX;
    rAeroStream(ii,1) = rAeroStream(1,1) + DeltaR;

    % For the C- Characteristics in Region One
    LineSlope = ThetaAeroStream(ii,1) - AlphaAeroStream(ii,1);

    % Initiate the values of the last point on the cowl streamline
    ThetaLast = ThetaAeroStreamCowl((jj-1),1);
    xLast = xAeroStreamCowl((jj-1),1);
    rLast = rAeroStreamCowl((jj-1),1);

    % Calculate the point that satisfies the Stream Function
    a = -tan(ThetaLast);
    b = -tan(LineSlope);
    c = rLast - tan(ThetaLast) * xLast;
    d = rAeroStream(ii,1) - tan(LineSlope)*xAeroStream(ii,1);

    A = [1 a; 1 b];
    B = [c; d];

    solution = A\B;

    rAeroStreamCowl(jj,1) = solution(1,1);
    xAeroStreamCowl(jj,1) = solution(2,1);
    ThetaAeroStreamCowl(jj,1) = ThetaAeroStream(ii,1);
    vAeroStreamCowl(jj,1) = vAeroStream(ii,1);
    MaAeroStreamCowl(jj,1) = MaAeroStream(ii,1);
    AlphaAeroStreamCowl(jj,1) = AlphaAeroStream(ii,1);

    if EntroCheck == 1
    % Calculate the Pressure, Temperature and change in Entropy
    % at the point    log = natural log
      PressAeroStreamCowl(jj,1) = Pstag * ((1 + (((Gamma - 1)/2) *...
                  ((MaAeroStreamCowl(jj,1)) ^ 2))) ^ (-Gamma/(Gamma - 1)));

      TempAeroStreamCowl(jj,1) = Tstag * (1 + (((Gamma - 1)/2) *...
                  ((MaAeroStreamCowl(jj,1)) ^ 2))) ^ (-1);

      DeltaSAeroStreamCowl(jj,1) = cp * log(TempAeroStreamCowl(jj,1)/...
                  Tstag) - Rgas * log(PressAeroStreamCowl(jj,1)/Pstag);
    end

    % Check to see if the expansion in the first expansion region is
    % complete
    if vAeroStream(ii) >= vRegionOne
        vRegionOneCheck = 0;
    else
        vRegionOneCheck = 1;
    end
end

% Now the program will calculate the Aerospike wall contour for the
% external portion of the nozzle. This will use C+ characteristics
% equations. The expansion point will be the last calculated cowl
% streamline point. The flow must also be turned back to Theta = 0

% Initialize loop conditions
% use ii and jj from previous loop; essentially ii = ii and jj = jj
MaContinue = 1;
DeltaVAeroTemp = 0.0;
```

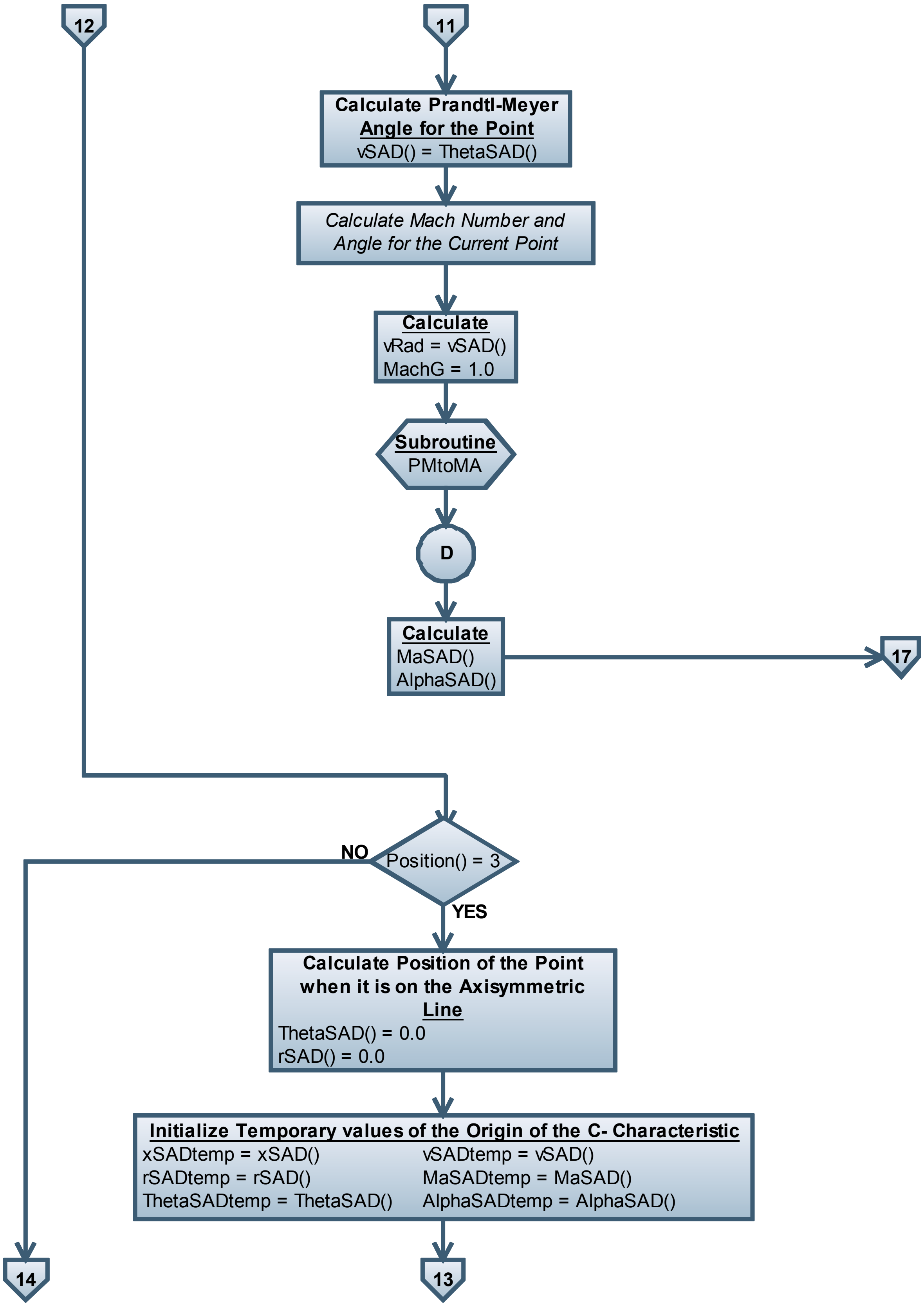
```
% Loop will calculate the external aerospike wall contour
while MaContinue == 1;
    ii = ii + 1;    % Increase indices every iteration

    %Set Expansion point conditions
    ThetaAeroStreamCowl(jj,1) = ThetaAeroStreamCowl(jj,1) - DeltaVAeroTemp;
    vAeroStreamCowl(jj,1) = vAeroStreamCowl(jj,1) + DeltaVAeroTemp;

    % Calculate the Mach number at the current point
    vRad = vAeroStreamCowl(jj,1);
    MachG = MaAeroStreamCowl(jj,1);
    PMtoMA      % Call subroutine to find point's Mach Number
    MaAeroStreamCowl(jj,1) = Mach;
    AlphaAeroStreamCowl(jj,1) = asin(1/MaAeroStreamCowl(jj,1));

    % Slope of the Mach line in radians
    LineSlope = ThetaAeroStreamCowl(jj,1) + AlphaAeroStreamCowl(jj,1);

    % Calculate the point that satisfies the Stream Function condition at
    % the last expansion point on the contour
    rStart = rAeroStream((ii-1),1);

    % Calculate r-intercept of the Mach line
    rIntercept = rAeroStreamCowl(jj,1) -...
                (tan(LineSlope) * xAeroStreamCowl(jj,1));

    % Initialize the values of the last point on the streamline
    ThetaLast = ThetaAeroStream((ii-1),1);
    xLast = xAeroStream((ii-1),1);
    rLast = rAeroStream((ii-1),1);

    % Calculate the point that satisfies the Stream Function

    a = -tan(ThetaLast);
    b = -tan(LineSlope);
    c = rLast - tan(ThetaLast) * xLast;
    d = rIntercept;

    A = [1 a; 1 b];
    B = [c; d];

    solution = A\B;

    rAeroStream(ii,1) = solution(1,1);
    xAeroStream(ii,1) = solution(2,1);
    ThetaAeroStream(ii,1) = ThetaAeroStreamCowl(jj,1);
    vAeroStream(ii,1) = vAeroStreamCowl(jj,1);
    MaAeroStream(ii,1) = MaAeroStreamCowl(jj,1);
    AlphaAeroStream(ii,1) = AlphaAeroStreamCowl(jj,1);

    if EntroCheck == 1
    % Calculate the Pressure, Temperature and change in Entropy
    % at the point    log = natural log
       PressAeroStream(ii,1) = Pstag * ((1 + (((Gamma - 1)/2) *...
                  ((MaAeroStream(ii,1)) ^ 2))) ^ (-Gamma/(Gamma - 1)));

       TempAeroStream(ii,1) = Tstag * (1 + (((Gamma - 1)/2) *...
                  ((MaAeroStream(ii,1)) ^ 2))) ^ (-1);

       DeltaSAeroStream(ii,1) = cp * log(TempAeroStream(ii,1)/Tstag)...
                    - Rgas * log(PressAeroStream(ii,1)/Pstag);
    end

    DeltaVAeroTemp = DeltaVAeroD;

    if vAeroStreamCowl(jj,1) >= vmax
        MaContinue = 0;
    else
        MaContinue = 1;
    end
end
```

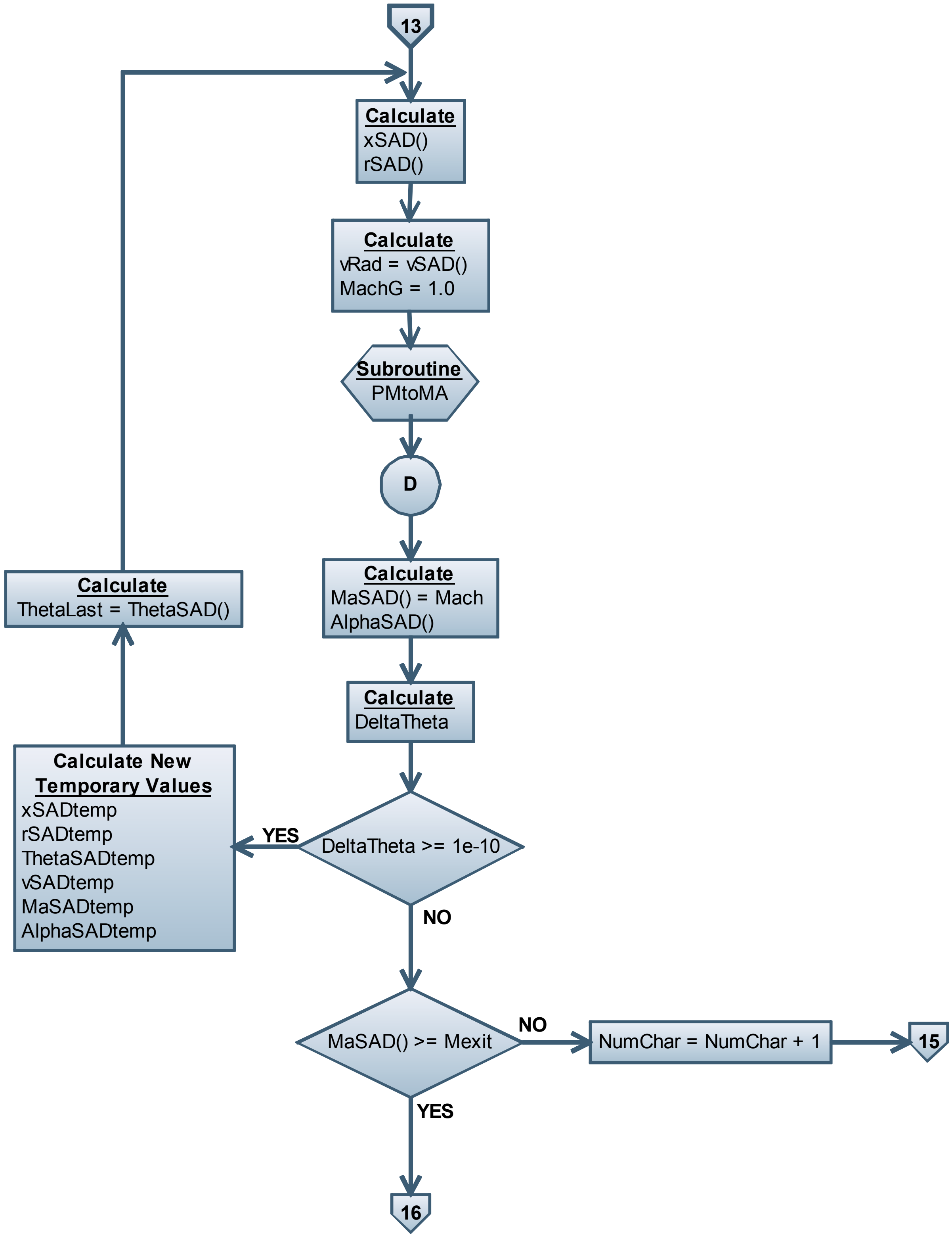
```
% Program will calculate the truncated contour if the user has decided to
if Truncate == 1
    zz = 1;
    PercentLength = 0.0;    % Initializes loop

    idealLength = xAeroStream(ii,1);

    while PercentLength <= (Percent/100)
        xAeroStreamTrunc(zz,1) = xAeroStream(zz,1);
        rAeroStreamTrunc(zz,1) = rAeroStream(zz,1);
        ThetaAeroStreamTrunc(zz,1) = ThetaAeroStream(zz,1);
        vAeroStreamTrunc(zz,1) = vAeroStream(zz,1);
        MaAeroStreamTrunc(zz,1) = MaAeroStream(zz,1);
        AlphaAeroStreamTrunc(zz,1) = AlphaAeroStream(zz,1);

        zz = zz + 1;

        PercentLength = abs(xAeroStreamTrunc((zz-1),1))/idealLength;
    end
end

% The program will now "flip" the contour so that the axisymmetric line
% aligns with r=0.
rExit = rAeroStream(ii,1);

for ll = 1:1:(ii)
    rAeroStream(ll,1) = rExit - rAeroStream(ll,1);
end

for ll = 1:1:(jj)
    rAeroStreamCowl(ll,1) = rExit - rAeroStreamCowl(ll,1);
end

% "Flip" the center of the circle defining the expansion contour of Region
% One
rCenter = rExit - rCenter;

% This section "flips" the truncated Aerospike
if Truncate == 1
    for jj=1:1:(zz-1)
        rAeroStreamTrunc(jj,1) = rExit - rAeroStreamTrunc(jj,1);
    end
end
```

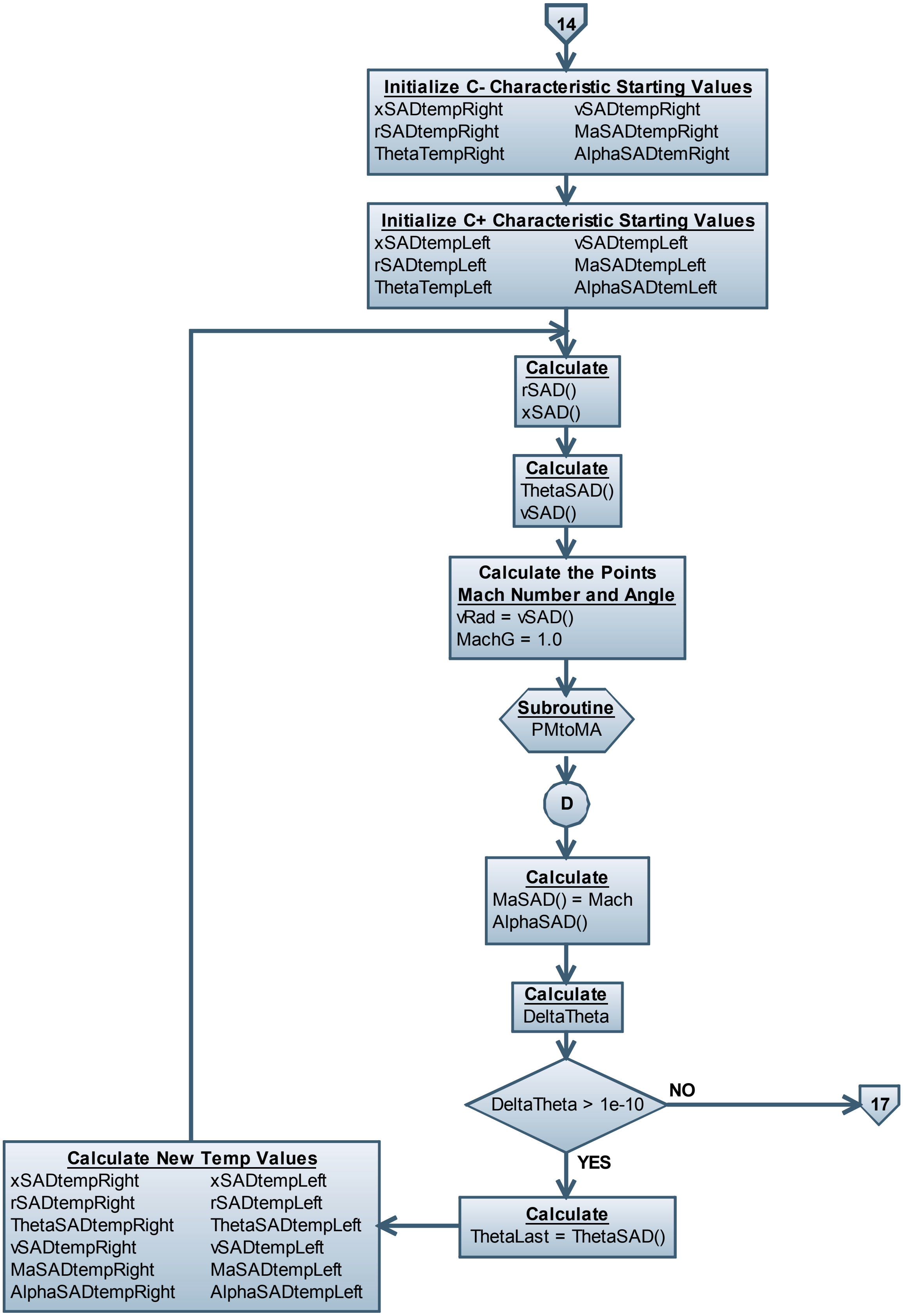
B.4 Axisymmetric.m Program

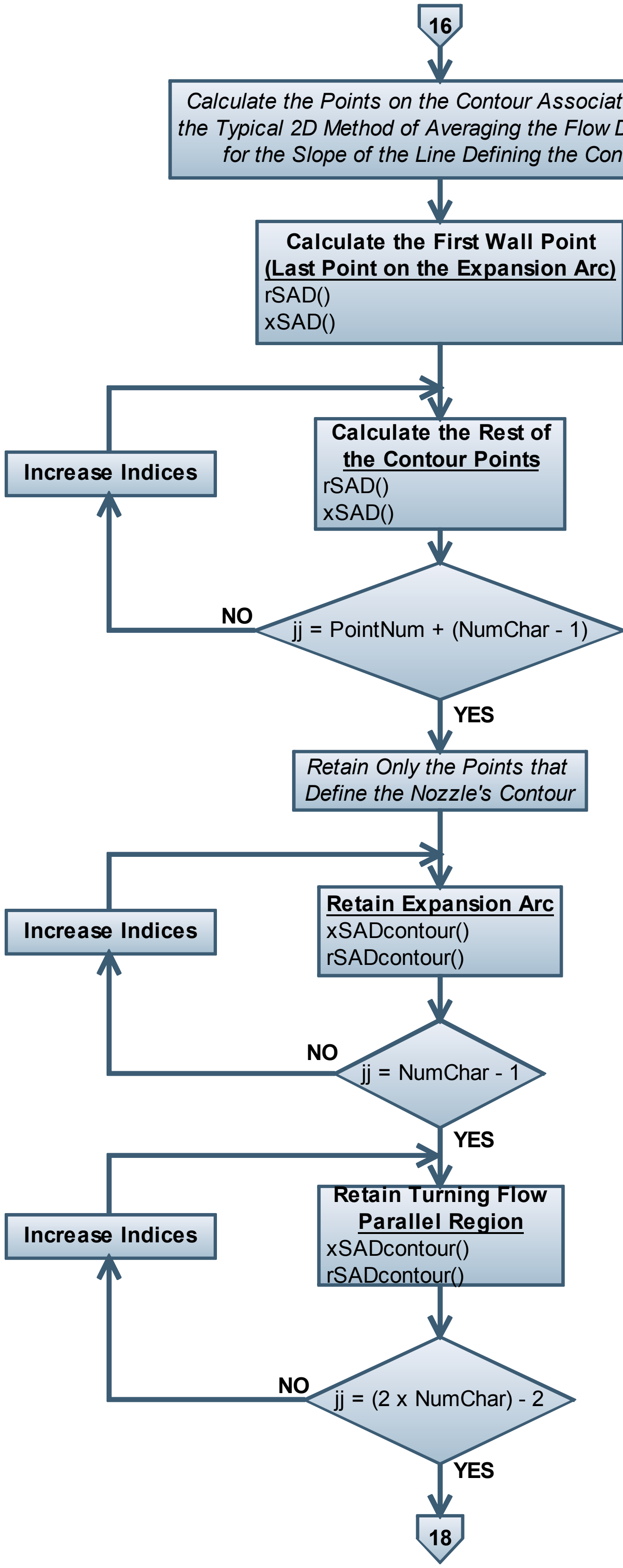
Figure B.4.1 Axisymmetric.m Program Flowchart

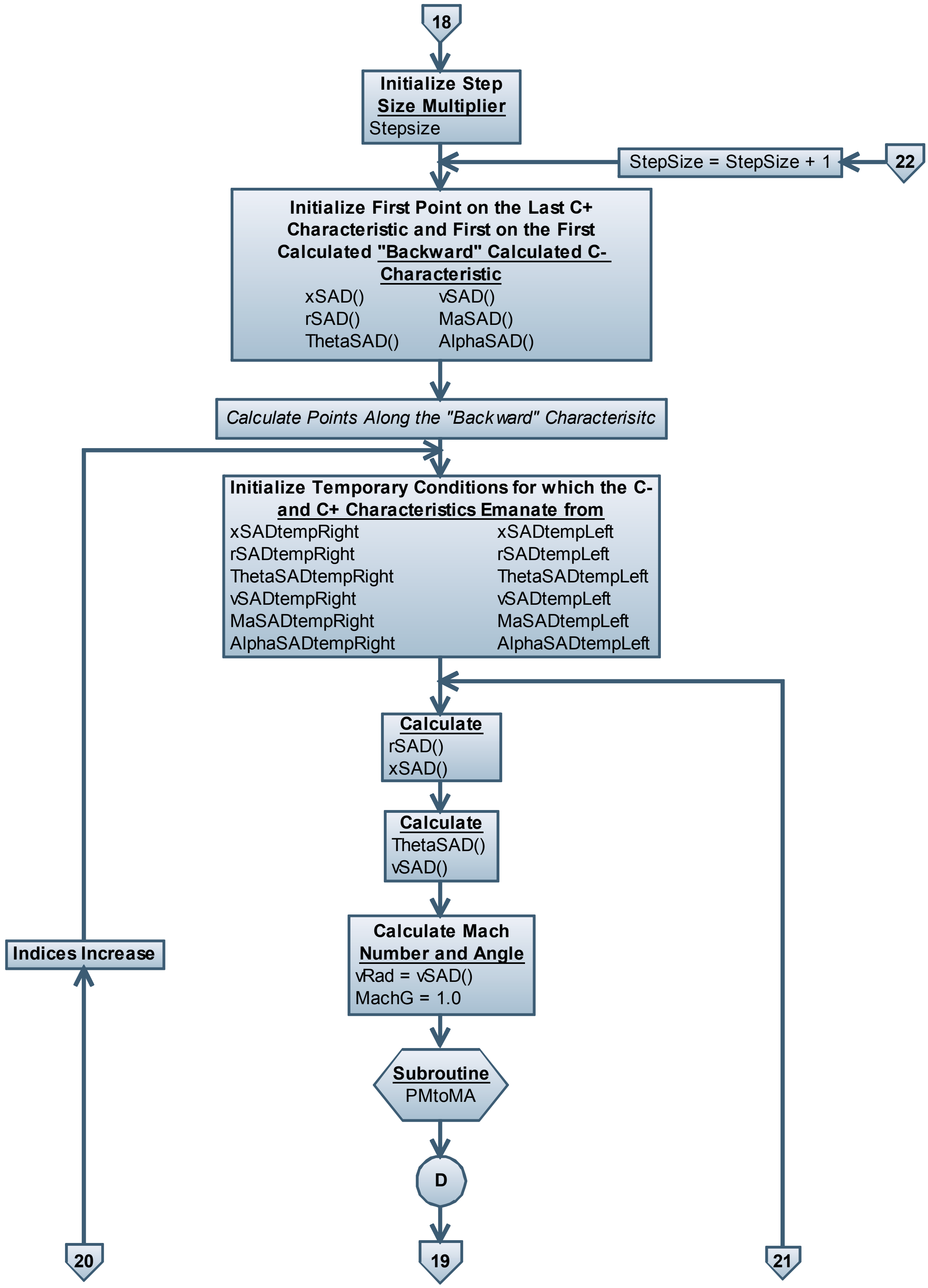


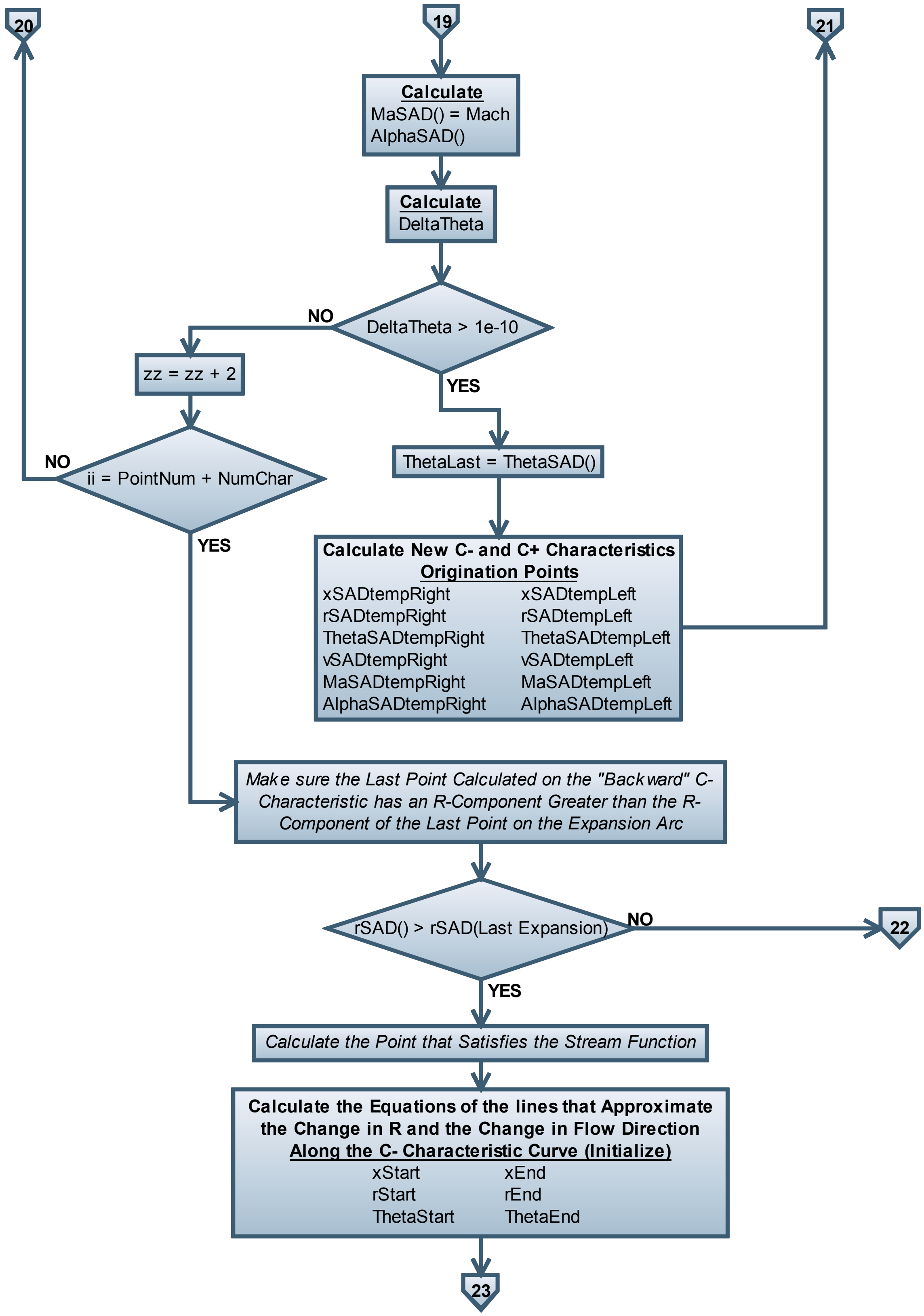

12
11
Calculate Prandtl-Meyer Angle for the Point
vSAD() = ThetaSAD()
Calculate Mach Number and Angle for the Current Point
Calculate
vRad = vSAD()
MachG = 1.0
Subroutine
PMtoMA
D
Calculate
MaSAD()
AlphaSAD()
17
NO
Position() = 3
YES
Calculate Position of the Point when it is on the Axisymmetric Line
ThetaSAD() = 0.0
rSAD() = 0.0
Initialize Temporary values of the Origin of the C- Characteristic
xSADtemp = xSAD()
vSADtemp = vSAD()
rSADtemp = rSAD()
MaSADtemp = MaSAD()
ThetaSADtemp = ThetaSAD()
AlphaSADtemp = AlphaSAD()
14
13

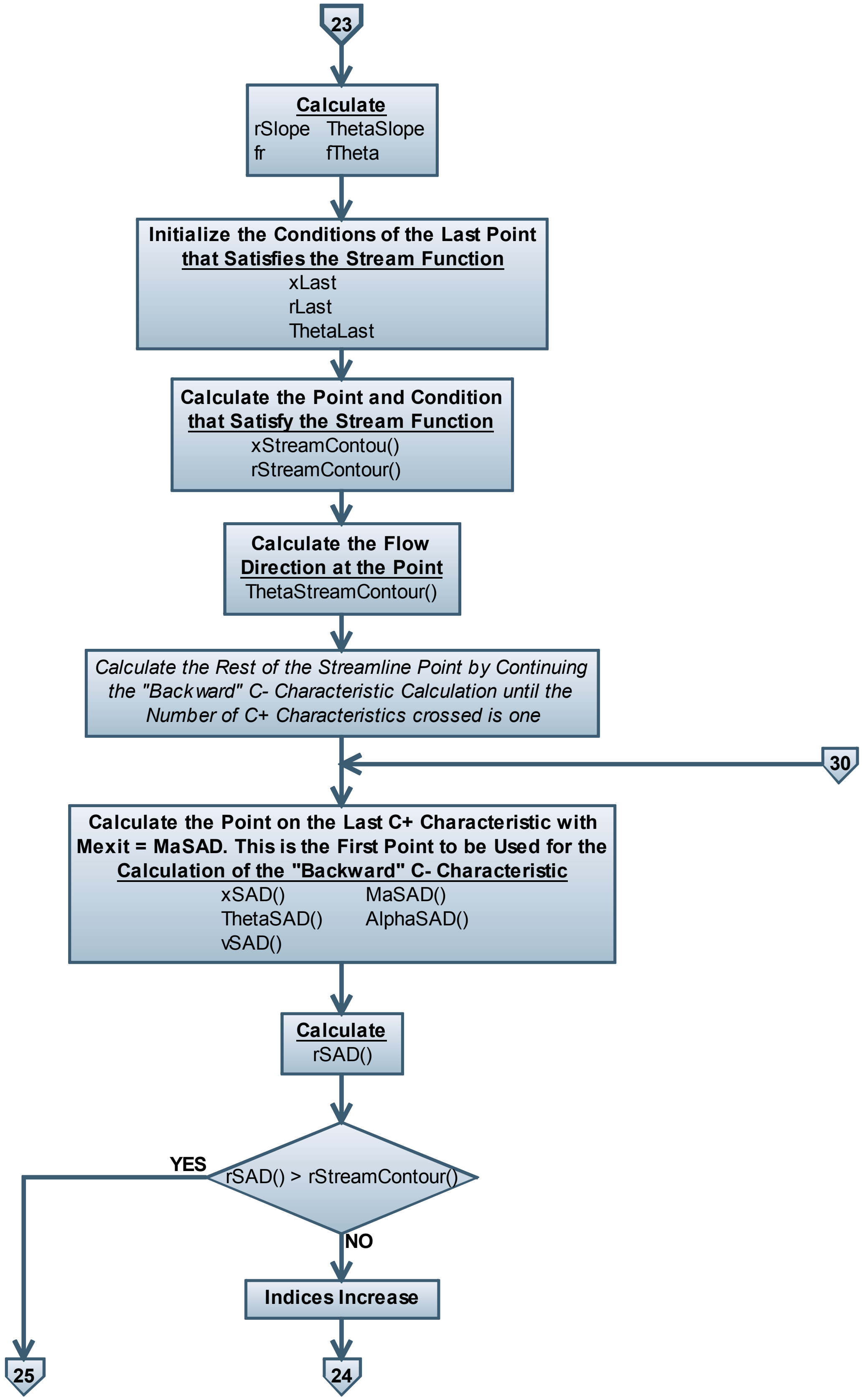

13
Calculate
xSAD()
rSAD()
Calculate
vRad = vSAD()
MachG = 1.0
Subroutine
PMtoMA
D
Calculate
MaSAD() = Mach
AlphaSAD()
Calculate
DeltaTheta
DeltaTheta >= 1e-10
YES
Calculate New
Temporary Values
xSADtemp
rSADtemp
ThetaSADtemp
vSADtemp
MaSADtemp
AlphaSADtemp
Calculate
ThetaLast = ThetaSAD()
NO
MaSAD() >= Mexit
NO
NumChar = NumChar + 1
15
YES
16

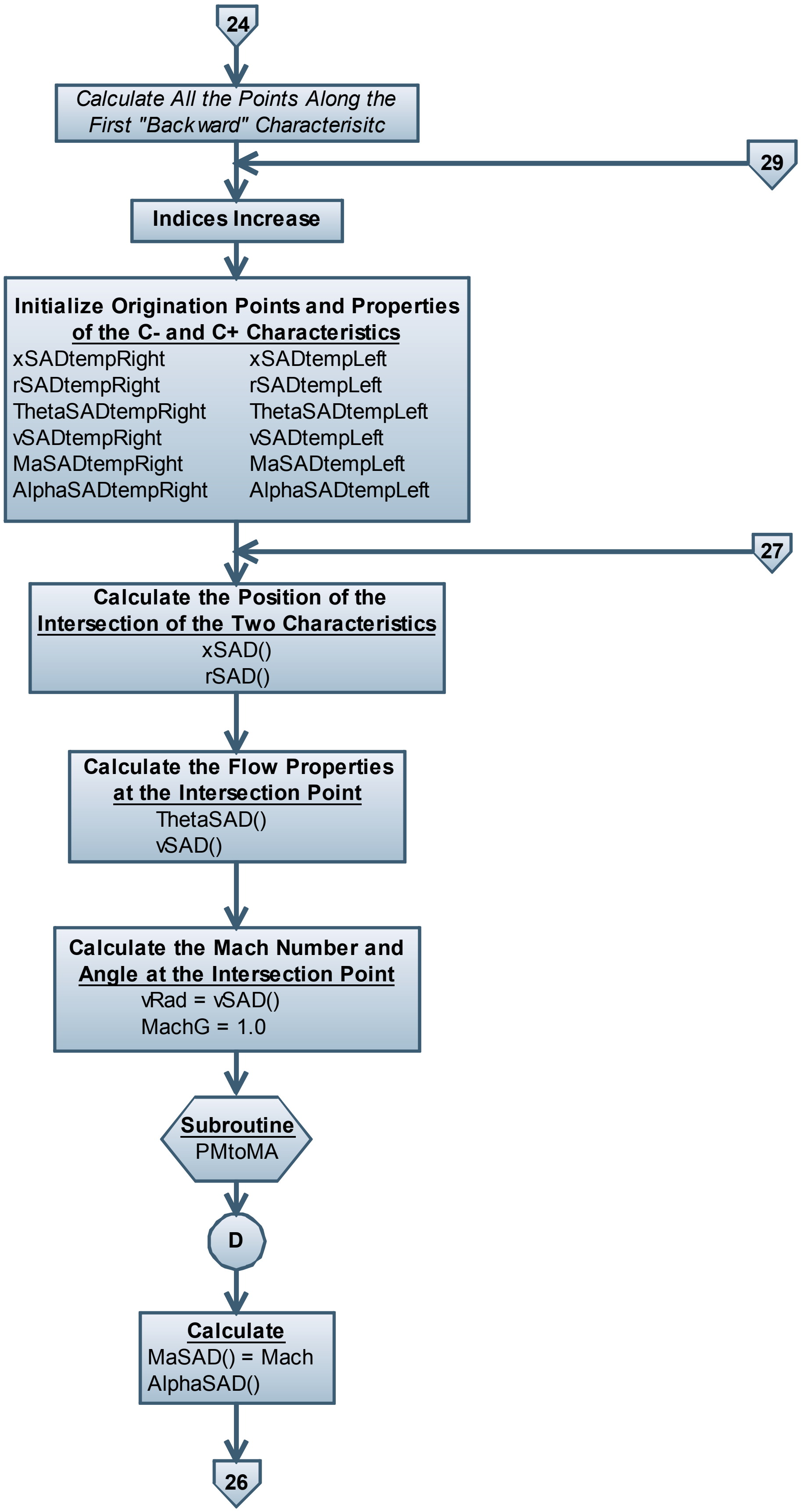

14
Initialize C- Characteristic Starting Values
xSADtempRight
vSADtempRight
rSADtempRight
MaSADtempRight
ThetaTempRight
AlphaSADtemRight
Initialize C+ Characteristic Starting Values
xSADtempLeft
vSADtempLeft
rSADtempLeft
MaSADtempLeft
ThetaTempLeft
AlphaSADtemLeft
Calculate
rSAD()
xSAD()
Calculate
ThetaSAD()
vSAD()
Calculate the Points
Mach Number and Angle
vRad = vSAD()
MachG = 1.0
Subroutine
PMtoMA
D
Calculate
MaSAD() = Mach
AlphaSAD()
Calculate
DeltaTheta
DeltaTheta > 1e-10
NO
17
YES
Calculate
ThetaLast = ThetaSAD()
Calculate New Temp Values
xSADtempRight
xSADtempLeft
rSADtempRight
rSADtempLeft
ThetaSADtempRight
ThetaSADtempLeft
vSADtempRight
vSADtempLeft
MaSADtempRight
MaSADtempLeft
AlphaSADtempRight
AlphaSADtempLeft

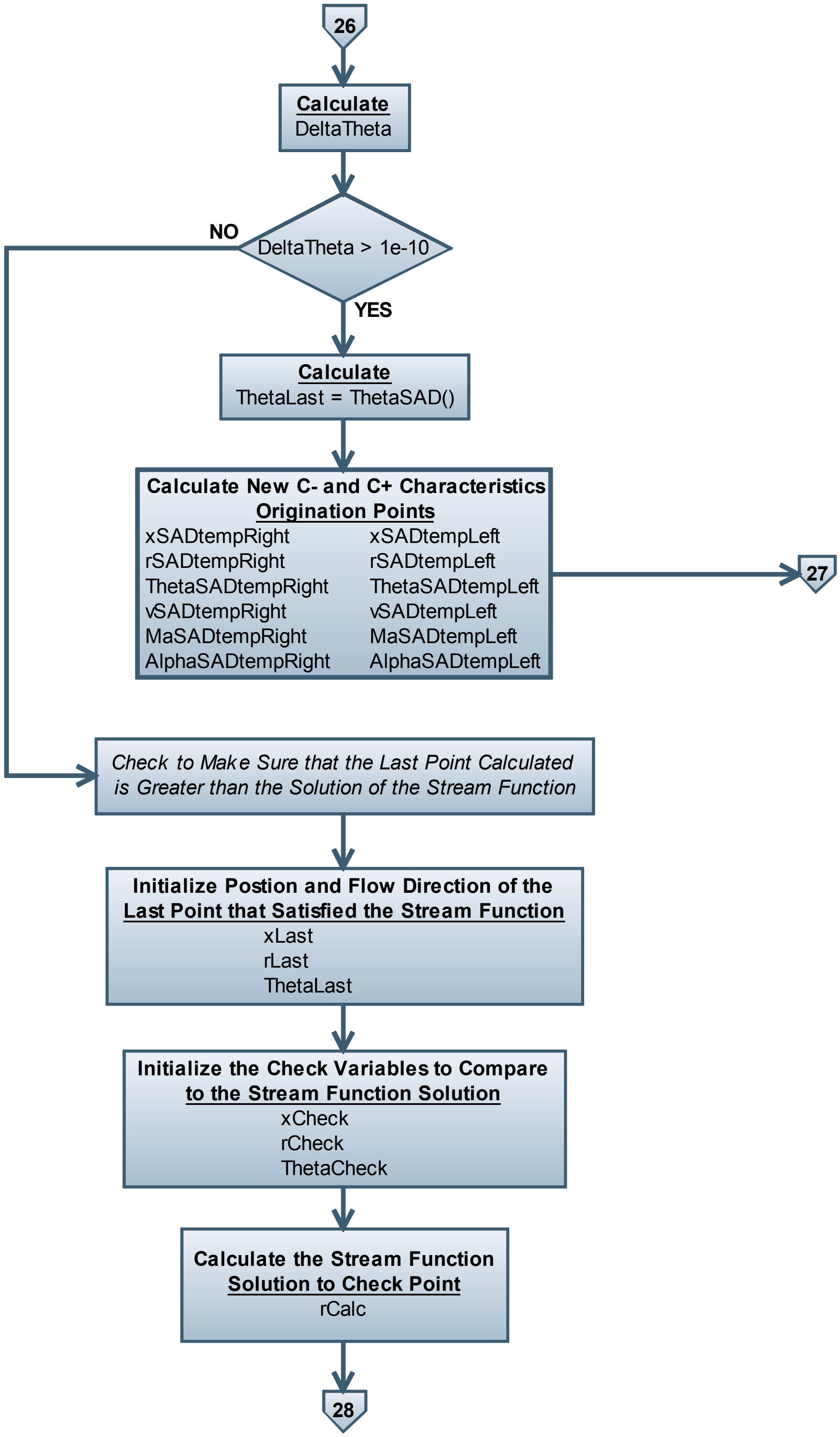

16
Calculate the Points on the Contour Associated with the Typical 2D Method of Averaging the Flow Direction for the Slope of the Line Defining the Contour
Calculate the First Wall Point (Last Point on the Expansion Arc)
rSAD()
xSAD()
Increase Indices
Calculate the Rest of the Contour Points
rSAD()
xSAD()
NO
jj = PointNum + (NumChar - 1)
YES
Retain Only the Points that Define the Nozzle's Contour
Increase Indices
Retain Expansion Arc
xSADcontour()
rSADcontour()
NO
jj = NumChar - 1
YES
Increase Indices
Retain Turning Flow Parallel Region
xSADcontour()
rSADcontour()
NO
jj = (2 x NumChar) - 2
YES
18


18
Initialize Step Size Multiplier
Stepsize
StepSize = StepSize + 1
22
Initialize First Point on the Last C+ Characteristic and First on the First Calculated "Backward" Calculated C- Characteristic
xSAD() vSAD()
rSAD() MaSAD()
ThetaSAD() AlphaSAD()
Calculate Points Along the "Backward" Characterisitc
Initialize Temporary Conditions for which the C- and C+ Characteristics Emanate from
xSADtempRight xSADtempLeft
rSADtempRight rSADtempLeft
ThetaSADtempRight ThetaSADtempLeft
vSADtempRight vSADtempLeft
MaSADtempRight MaSADtempLeft
AlphaSADtempRight AlphaSADtempLeft
Calculate
rSAD()
xSAD()
Calculate
ThetaSAD()
vSAD()
Calculate Mach Number and Angle
vRad = vSAD()
MachG = 1.0
Indices Increase
Subroutine
PMtoMA
D
20
19
21

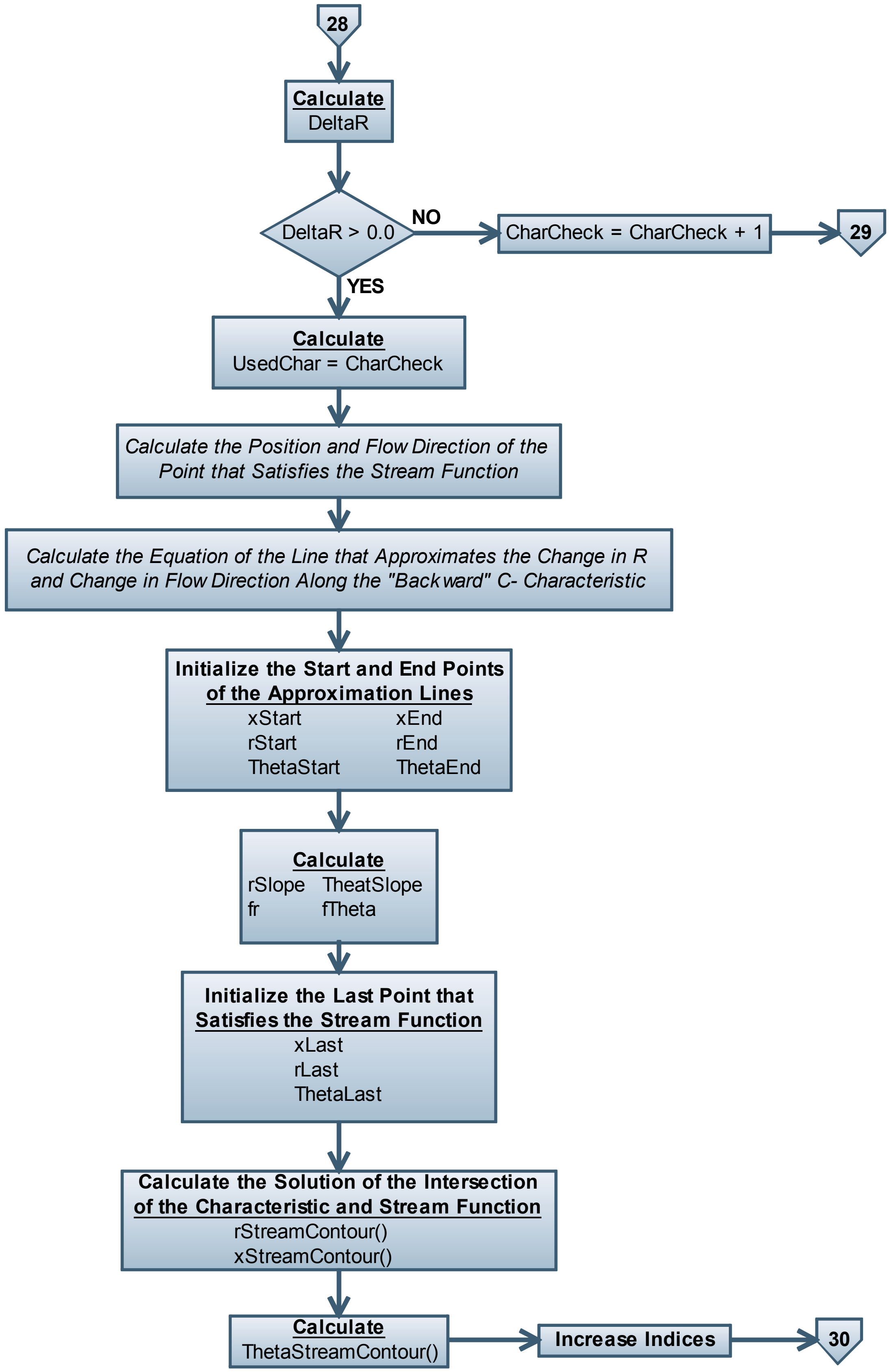

20
19
21
Calculate
MaSAD() = Mach
AlphaSAD()
Calculate
DeltaTheta
NO
DeltaTheta > 1e-10
YES
zz = zz + 2
NO
ii = PointNum + NumChar
YES
ThetaLast = ThetaSAD()
Calculate New C- and C+ Characteristics
Origination Points
xSADtempRight xSADtempLeft
rSADtempRight rSADtempLeft
ThetaSADtempRight ThetaSADtempLeft
vSADtempRight vSADtempLeft
MaSADtempRight MaSADtempLeft
AlphaSADtempRight AlphaSADtempLeft
Make sure the Last Point Calculated on the "Backward" C-Characteristic has an R-Component Greater than the R-Component of the Last Point on the Expansion Arc
rSAD() > rSAD(Last Expansion)
NO
22
YES
Calculate the Point that Satisfies the Stream Function
Calculate the Equations of the lines that Approximate the Change in R and the Change in Flow Direction
Along the C- Characteristic Curve (Initialize)
xStart xEnd
rStart rEnd
ThetaStart ThetaEnd
23


23
Calculate
rSlope ThetaSlope
fr fTheta
Initialize the Conditions of the Last Point that Satisfies the Stream Function
xLast
rLast
ThetaLast
Calculate the Point and Condition that Satisfy the Stream Function
xStreamContou()
rStreamContour()
Calculate the Flow Direction at the Point
ThetaStreamContour()
Calculate the Rest of the Streamline Point by Continuing the "Backward" C- Characteristic Calculation until the Number of C+ Characteristics crossed is one
30
Calculate the Point on the Last C+ Characteristic with Mexit = MaSAD. This is the First Point to be Used for the Calculation of the "Backward" C- Characteristic
xSAD() MaSAD()
ThetaSAD() AlphaSAD()
vSAD()
Calculate
rSAD()
rSAD() > rStreamContour()
YES
NO
Indices Increase
25
24


24
Calculate All the Points Along the First "Backward" Characterisitc
29
Indices Increase
Initialize Origination Points and Properties of the C- and C+ Characteristics
xSADtempRight
rSADtempRight
ThetaSADtempRight
vSADtempRight
MaSADtempRight
AlphaSADtempRight
xSADtempLeft
rSADtempLeft
ThetaSADtempLeft
vSADtempLeft
MaSADtempLeft
AlphaSADtempLeft
27
Calculate the Position of the Intersection of the Two Characteristics
xSAD()
rSAD()
Calculate the Flow Properties at the Intersection Point
ThetaSAD()
vSAD()
Calculate the Mach Number and Angle at the Intersection Point
vRad = vSAD()
MachG = 1.0
Subroutine
PMtoMA
D
Calculate
MaSAD() = Mach
AlphaSAD()
26



26
Calculate
DeltaTheta
DeltaTheta > 1e-10
NO
YES
Calculate
ThetaLast = ThetaSAD()
Calculate New C- and C+ Characteristics
Origination Points
xSADtempRight
rSADtempRight
ThetaSADtempRight
vSADtempRight
MaSADtempRight
AlphaSADtempRight
xSADtempLeft
rSADtempLeft
ThetaSADtempLeft
vSADtempLeft
MaSADtempLeft
AlphaSADtempLeft
27
Check to Make Sure that the Last Point Calculated
is Greater than the Solution of the Stream Function
Initialize Postion and Flow Direction of the
Last Point that Satisfied the Stream Function
xLast
rLast
ThetaLast
Initialize the Check Variables to Compare
to the Stream Function Solution
xCheck
rCheck
ThetaCheck
Calculate the Stream Function
Solution to Check Point
rCalc
28


28
Calculate
DeltaR
DeltaR > 0.0
NO
CharCheck = CharCheck + 1
29
YES
Calculate
UsedChar = CharCheck
Calculate the Position and Flow Direction of the Point that Satisfies the Stream Function
Calculate the Equation of the Line that Approximates the Change in R and Change in Flow Direction Along the "Backward" C- Characteristic
Initialize the Start and End Points of the Approximation Lines
xStart xEnd
rStart rEnd
ThetaStart ThetaEnd
Calculate
rSlope TheatSlope
fr fTheta
Initialize the Last Point that Satisfies the Stream Function
xLast
rLast
ThetaLast
Calculate the Solution of the Intersection of the Characteristic and Stream Function
rStreamContour()
xStreamContour()
Calculate
ThetaStreamContour()
Increase Indices
30

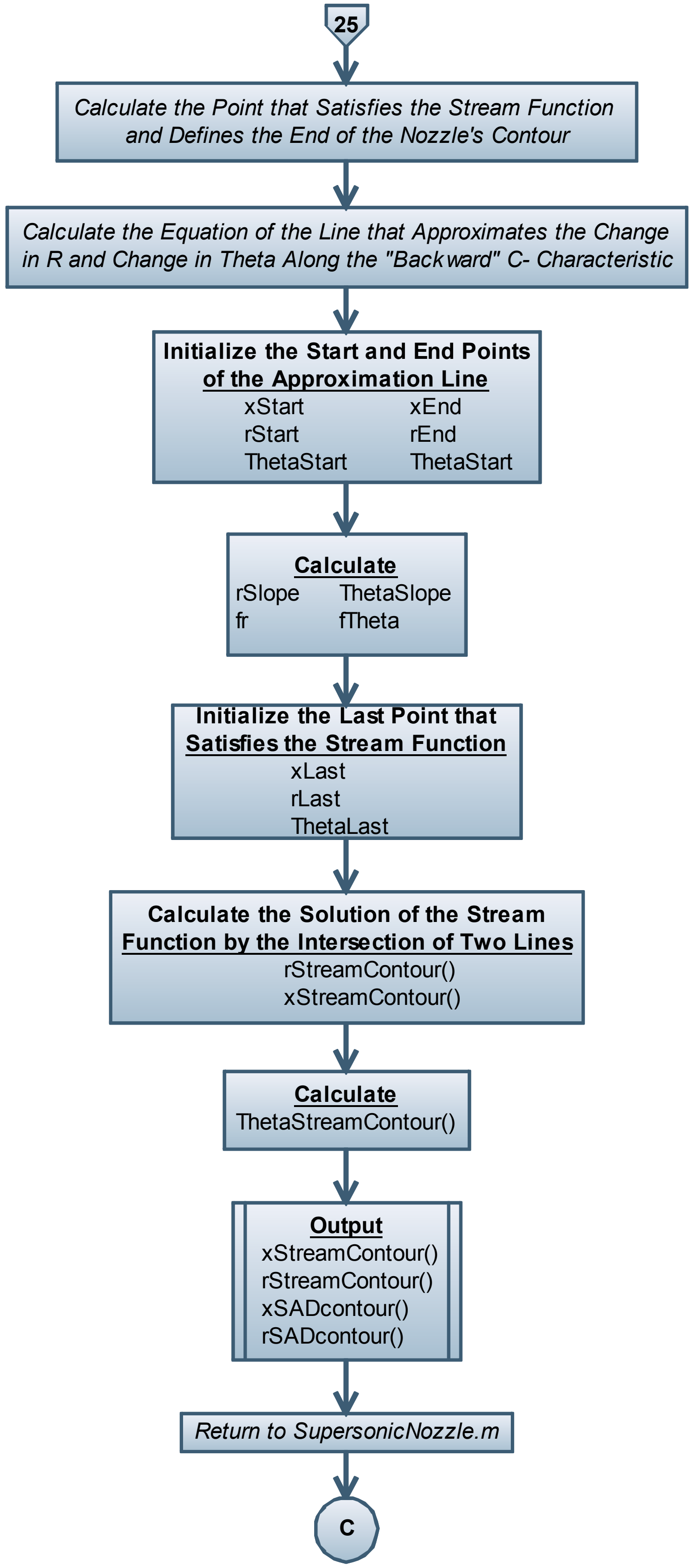

25
Calculate the Point that Satisfies the Stream Function and Defines the End of the Nozzle's Contour
Calculate the Equation of the Line that Approximates the Change in R and Change in Theta Along the "Backward" C- Characteristic
Initialize the Start and End Points of the Approximation Line
xStart xEnd
rStart rEnd
ThetaStart ThetaStart
Calculate
rSlope ThetaSlope
fr fTheta
Initialize the Last Point that Satisfies the Stream Function
xLast
rLast
ThetaLast
Calculate the Solution of the Stream Function by the Intersection of Two Lines
rStreamContour()
xStreamContour()
Calculate
ThetaStreamContour()
Output
xStreamContour()
rStreamContour()
xSADcontour()
rSADcontour()
Return to SupersonicNozzle.m
C

Figure B.4.2: Axisymmetric.m MatLab Source Code

```
%***************************************************************************
%*                  SAD Axially Symmetric Nozzle
%*                      Brandon Denton
%*                      RIT Graduate Student
%*                      Oct. 6, 2006
%***************************************************************************
% Program will use a combination of Shapiro, Anderson and Denton Methods to
% calculate the contour of an Axially Symmetric Supersonic Nozzle.
%***************************************************************************
%***************************************************************************
%* Directory of Variables
%***************************************************************************
%*  1. a, b, c, d ----------- Variables that store values for the matrices
%*                            used to calculate the coordinates and flow
%*                            properties of the point where the C- and C+
%*                            Characteristics intersect
%*  2. A, B ----------------- Matrices used the calculate the coordinates
%*                            and flow properties of the point where the C-
%*                            and C+ Characteristics intersect
%*  3. AlphaSAD() ----------- Array that holds the Mach angle of the points
%*  4. AlphaSADtemp --------- Variable that holds the temporary Mach angle
%*                            of the point from which the C- Characteristic
%*                            emanates from when the point to be calculated
%*                            is on the axisymmetric line
%*  5. AlphaSADtempLeft ----- Variable that temporarily stores teh Mach
%*                            angle of the point from which the C+
%*                            Characteristic emanates from
%*  6. AlphaSADtempRight ---- Variable that temporarily stores the Mach
%*                            angle of the point from which the C-
%*                            Characteristic emanates from
%*  7. AlphaSADtempRight ---- Variable that holds the temporary value of
%*                            the Mach angle from which the C-
%*                            Characteristic emanates from
%*  8. BackContinue --------- Control variable for the loop that makes sure
%*                            that the last point calculated on the
%*                            "backward" characteristics will be greater
%*                            than the point which will satisfy the Stream
%*                            Function
%*  9. Beta ----------------- Variable that holds the multipling factor of
%*                            the throat and dictates the radius of the
%*                            circle which defines the arc of the expansion
%*                            region
%* 10. CalcContinue --------- Control variable for the loop which
%*                            calculates the points that satisfy the Stream
%*                            Function and define the nozzle's contour
%* 11. Char ----------------- Variable which holds the number of
%*                            characteristics crossed when calculating the
%*                            nozzle's contour by the typical 2D method
%* 12. CharCheck ------------ Variable which keep tracks of how many C+
%*                            Characteristics are crossed during the
%*                            calculation of the "backward" characteristics
%* 13. DeltaR --------------- The difference between the solution to the
%*                            Stream Function and the value on the line
%*                            segment where the point lies
%* 14. DeltaTheta ----------- Control variable and flow direction error for
%*                            the convergence of the points position and
%*                            properties
%* 15. DeltaVAeroD ---------- Variable that holds the change in angle of the
%*                            arc that defines the expansion region
%* 16. DeltaX --------------- Variable that holds the change in x value that
%*                            will be used during the "backward" C-
%*                            Characteristic calculations
%* 17. fTheta --------------- The theta-intercept of the line segment on
%*                            which a point that satisfies the Stream
%*                            Function can be found
%* 18. fr ------------------- The r-intercept of line segment on which a
%*                            point which satisfies the Stream Function can
%*                            be found
```

```
%* 19. Gamma ---------------- Ratio of specific heats of the working fluid
%* 20. ii ------------------- Indices index
%* 21. jj ------------------- Indices index
%* 22. kk ------------------- Indices index
%* 23. ll ------------------- Indices index
%* 24. Mach ----------------- Solution value returned by the subroutine
%*                           PMtoMA for the Mach Number at a point
%* 25. MachG ---------------- Initial guess value used by the subroutine
%*                           PMtoMA to find the point's Mach Number
%* 26. MaContinue ----------- Control variable for the loop that calculates
%*                           the flowfield until the desired exit Mach
%*                           Number is reached on the axisymmetric line
%* 27. MaSAD() -------------- Array that holds the Mach Numbers of the
%*                           points
%* 28. MaSADtemp ------------ Variable that holds the temporary Mach Number
%*                           of the point from which the C- Characterisitc
%*                           emanates from when the point to be calculated
%*                           is located on the axisymmetric line
%* 29. MaSADtempLeft -------- Variable that temporarily stores the Mach
%*                           Number of the point from which the C+
%*                           Characteristic emanates from
%* 30. MaSADtempRight ------- Variable that temporarily stores the Mach
%*                           Number of the point from which the C-
%*                           Characteristic emanates from
%* 31. Mexit ---------------- User-defined desired exit Mach Number of the
%*                           nozzle
%* 32. NumChar -------------- Number of characteristics used in the
%*                           calculation
%* 33. PMtoMA --------------- Subroutine used to find the Mach Number of a
%*                           point in the flowfield
%* 34. PointNum ------------- Number of points in the flowfield until the
%*                           desired exit Mach number is reached on the
%*                           axisymmetric line
%* 35. Position() ----------- Array that holds the value which dictates the
%*                           type of flowfield point (1 = wall, 2 = flow,
%*                           3 = axisymmetric)
%* 36. rCalc ---------------- The Stream Function solution of the last
%*                           point on the "backward" characteristic to
%*                           make sure it is greater than the point which
%*                           satisfies the Stream Function
%* 37. rCheck --------------- R-component of the last point on the
%*                           "backward" characteristic to check if the
%*                           point is greater that the point which
%*                           satisfies the Stream Function
%* 38. rEnd ----------------- R-component of the point which defines the
%*                           end of the line segment on which a point
%*                           which satisfies the Stream Function can be
%*                           found
%* 39. rLast ---------------- R-component of the point which last satisfed
%*                           the Stream Function
%* 40. rSAD() --------------- Array that holds the r-component of the point
%* 41. rSADcontour ---------- Variable that holds the r-component of the
%*                           points which define the nozzles contour
%*                           calculated by the 2D method
%* 42. rSADtemp ------------- Variable that holds the temporary r-component
%*                           of the point from which the C- Characteristic
%*                           emanates from when the point to be calculated
%*                           is located on the axisymmetric line
%* 43. rSADtempLeft --------- Variable that temporarily stores the
%*                           r-component of the point from which the C+
%*                           Characteristic is emanating from
%* 44. rSADtempRight -------- Variable that temporarily stores the
%*                           r-component from the point where the C-
%*                           Characteristic is emanating from
%* 45. rSlope --------------- The change in r for a change in x on the line
%*                           segment for which a point which satisfies the
%*                           Stream Function can be found
%* 46. rStart --------------- R-component of the point which defines the
%*                           start of the line segment on which the point
%*                           which satisfies the Stream Function can be
%*                           found
```

```
%* 47. rStreamContour() ----- Array which holds the r-component of the
%*                            points which satisfy the Stream Function and
%*                            define the nozzle's contour
%* 48. rThroat -------------- User-defined throat radius
%* 49. solution() ----------- Solution to the Matrices used to calculate
%*                            the coordinates and flow properties of the
%*                            intersection of the C- and C+ Characteristics
%* 50. StepSize ------------- Step multiplier used when incrementally
%*                            increaseing the number of "backward"
%*                            characteristics used in the calculation
%* 51. StepSizePart --------- The decimal part of variable StepSize aslo
%*                            used to add in the difference between 1 and
%*                            the decimal part of StepSize back in to
%*                            StepSize to make it an integer
%* 52. StepSizeWhole -------- The whole number part of variable StepSize
%* 53. StreamContinue ------- Control variable for the loop that calculates
%*                            the nozzle's contour using the streamline
%*                            method
%* 54. ThetaCheck ----------- Flow direction of the last point on the
%*                            "backward" Characteristics to check if the
%*                            point is greater than the point which
%*                            satisfies the Stream Function
%* 55. ThetaEnd ------------- Flow direction of the point which defines the
%*                            end of the line segment which a point that
%*                            satisfies the Stream Function can be found
%* 56. ThetaLast ------------ Temporary value of the flow direction of the
%*                            last point that satisfies the Stream Function
%*                            also dubbes as a temporary value in the
%*                            convergence of DeltaTheta
%* 57. ThetaSAD() ----------- Array that holds the flow direction at the
%*                            point
%* 58. ThetaSADtemp --------- Variable that holds the temporary flow
%*                            direction of the point from which the C-
%*                            Characteristic emanates from when the point
%*                            to be calculated is located on the
%*                            axisymmetric line
%* 59. ThetaSADtempLeft ----- Variable that temporarily stores the flow
%*                            direction of the point from which the C+
%*                            Characteristic is emananting from
%* 60. ThetaSADtempRight ---- Variable that temporarily stores the flow
%*                            direction of the point where the C-
%*                            Characteristic is emanating from
%* 61. ThetaSlope ----------- The change in flow direction per change in x
%*                            of the line segment on which a point that
%*                            satisfies the Stream Function can be found
%* 62. ThetaStart ----------- Flow direction of the point which defines the
%*                            start of the line segment on which a point
%*                            which satisfies the Stream Function can be
%*                            found
%* 63. ThetaStreamContour() - Array which holds the flow direction of the
%*                            points which satisfy the Stream Function and
%*                            define the nozzle's contour
%* 64. UsedChar ------------- Temporarily holds the number of C+
%*                            Characteristics crossed when calculating the
%*                            point which satisfies the Stream Function
%* 65. vRad ----------------- Variable used by subroutine PMtoMA to find
%*                            the Mach Number of the point
%* 66. vSAD() --------------- Array that holds the Prandtl-Meyer expansion
%*                            angle of the point
%* 67. vSADtemp ------------- Variable that holds the temporary
%*                            Prandtl-Meyer expansion angle of the point
%*                            from which the C- Characteristic emanates
%*                            from when the point ot be calculated is
%*                            located on the axisymmetric line
%* 68. vSADtempLeft --------- Variable that temporarily stores the
%*                            Prandtl-Meyer expansion angle of the point
%*                            from which the C+ Characteristic emanates
%* 69. vSADtempRight -------- Variable that temporarily stores the
%*                            Prandtl-Meyer expansion angle of the point
%*                            from which the C- Characteristic emanates
```

```
%* 70. xCheck --------------- X-component of the last point on the
%*                            "backward" characteristics to check if it is
%*                            greater than the point which satisfies the
%*                            Stream Function
%* 71. xEnd ----------------- X-component of the point which defines the end
%*                            of the line segment on which a point which
%*                            satisfies the Stream Function can be found
%* 72. xLast ---------------- X-component of the point which last satisfied
%*                            the Stream Function
%* 73. xSAD() --------------- Array that holds the x-component of the point
%* 74. xSADcontour ---------- Array that holds the x-component of the
%*                            points which define the nozzle's contour
%*                            calculate by the method for 2D nozzles
%* 75. xSADtemp ------------- Variable that holds the temporary x-component
%*                            of the point from which the C- Characteristic
%*                            emanates from when the point to be calculated
%*                            is located on the axisymmetric line
%* 76. xSADtempLeft --------- Variable that temporarily stores the
%*                            x-component of the point where the C+
%*                            Characteristic is emanatine from
%* 77. xSADtempRight -------- Variable that temporarily stores the
%*                            x-component of the point where the C-
%*                            Characteristic is emanating from
%* 78. xStart --------------- X-component of the starting point of the line
%*                            segment on which the point which satisfies
%*                            the Stream Function can be found
%* 79. xStreamContour() ----- Array which holds the x-components of the
%*                            points which satisfy the Stream Function and
%*                            define the nozzle's contour
%* 80. zz ------------------- Indices index
%***************************************************************************
%***************************************************************************
%*          Start Program
%***************************************************************************
%***************************************************************************
format long

DeltaX = DeltaVAeroD;

%The calculation must be started with the first characteristic
NumChar = 1;
PointNum = 0;       %Initializes the variable PointNum
ii = 1;             %Initializes the looping variable ii
jj = 1;             %Initializes the looping variable jj
MaContinue = 0;

while MaContinue == 0
    PointNum = PointNum + (NumChar + 1);

    %Initialize the position type of the points
    for ii = (PointNum - NumChar):1:PointNum
        if ii == (PointNum - NumChar)
            Position(ii,1) = 1;
        elseif ii == PointNum
            Position(ii,1) = 3;
        else
            Position(ii,1) = 2;
        end
    end

    for ii = (PointNum - NumChar):1:PointNum
        if Position(ii,1) == 1
            %Set the angle at which the calculation has been swept through
            ThetaSAD(ii,1) = NumChar * DeltaVAeroD;

            %Calculate the x-coordinate of the point that corresponds to the
            %current characteristic (Wall point)
            xSAD(ii,1) = (Beta * rThroat) * sin(ThetaSAD(ii,1));
```

```
    %Calculate the radial position of the point on the wall of the
    %current characteristic (Wall point)
    rSAD(ii,1) = rThroat + ((Beta * rThroat) *...
                         (1 - cos(ThetaSAD(ii,1))));

    %Calculate the Prandtl-Meyer Expansion Angle vSAD() for the
    %current point (Wall point)
    vSAD(ii,1) = ThetaSAD(ii,1);

    %Now calculate the Mach number and Alpha associated with the
    %point. Must convert for subprogram to find the associated Mach
    %Number
    vRad = vSAD(ii,1);

    MachG = 1.0;
    PMtoMA      %calls subprogram to find Mach Number

    MaSAD(ii,1) = Mach;
    AlphaSAD(ii,1) = asin(1/MaSAD(ii,1));

    if EntroCheck == 1
    % Calculate the Pressure, Temperature and change in Entropy
    % at the point    log = natural log
        PressSAD(ii,1) = Pstag * ((1 + (((Gamma - 1)/2) *...
                     ((MaSAD(ii,1)) ^ 2))) ^ (-Gamma/(Gamma - 1)));

        TempSAD(ii,1) = Tstag * (1 + (((Gamma - 1)/2) *...
                      ((MaSAD(ii,1)) ^ 2))) ^ (-1);

        DeltaS(ii,1) = cp * log(TempSAD(ii,1)/Tstag) - Rgas *...
                     log(PressSAD(ii,1)/Pstag);
    end

elseif Position(ii,1) == 3
    %Calculate the position of the point when it is on the
    %axisymmetric line
    ThetaSAD(ii,1) = 0;
    rSAD(ii,1) = 0;

    %Initialize temperary values for the loop calculation of the
    %curved characteristics
    ThetaSADtemp = ThetaSAD((ii-1),1);
    rSADtemp = rSAD((ii-1),1);
    xSADtemp = xSAD((ii-1),1);
    AlphaSADtemp = AlphaSAD((ii-1),1);
    MaSADtemp = MaSAD((ii-1),1);
    vSADtemp = vSAD((ii-1),1);

    DeltaTheta = 1.0; %Initialize the loop below
    ThetaLast = 100;

    while DeltaTheta >= 1e-10
        %Calculate the position of the current point
        xSAD(ii,1) = ((rSADtemp - (tan(ThetaSADtemp -...
                   AlphaSADtemp) * xSADtemp)) /...
                   (-tan(ThetaSADtemp - AlphaSADtemp))) -...
                   rSAD(ii,1);

        %Calculate the flow properties of the current point
        vSAD(ii,1) = ThetaSADtemp + vSADtemp - ThetaSAD(ii,1) + ...
                   ((1/(sqrt((MaSADtemp^2) - 1) -...
                   (1/tan(ThetaSADtemp)))) * ((rSAD(ii,1) -...
                   rSADtemp) / rSADtemp));

       %Calculate the Mach Number at the current point
       vRad = vSAD(ii,1);

       MachG = 1.0;
       PMtoMA    %calls subprogram to find Mach Number

       MaSAD(ii,1) = Mach;
       AlphaSAD(ii,1) = asin(1/MaSAD(ii,1));
```

```
            if EntroCheck == 1
            % Calculate the Pressure, Temperature and change in Entropy
            % at the point
                 PressSAD(ii,1) = Pstag * ((1 + (((Gamma - 1)/2) *...
                          ((MaSAD(ii,1)) ^ 2))) ^ (-Gamma/(Gamma - 1)));

                 TempSAD(ii,1) = Tstag * (1 + (((Gamma - 1)/2) *...
                          ((MaSAD(ii,1)) ^ 2))) ^ (-1);

                 DeltaS(ii,1) = cp * log(TempSAD(ii,1)/Tstag) - Rgas*...
                         log(PressSAD(ii,1)/Pstag);
            end

            %Calculate the change in Theta for the loop
            DeltaTheta = abs(ThetaLast - ThetaSAD(ii,1));

            if DeltaTheta > 1e-10
                ThetaLast = ThetaSAD(ii,1);

                 % Calculate averages of all values and replace the
                 % "temp" values with these. This is an approximation
                 % that the charactertistics are curved
                 ThetaSADtemp = (ThetaSADtemp + ThetaSAD(ii,1)) / 2;
                 AlphaSADtemp = (AlphaSADtemp + AlphaSAD(ii,1)) / 2;
                 MaSADtemp = (MaSADtemp + MaSAD(ii,1)) / 2;
                 rSADtemp = (rSADtemp + rSAD(ii,1)) / 2;
                 xSADtemp = (xSADtemp + xSAD(ii,1)) / 2;
                 vSADtemp = (vSADtemp + vSAD(ii,1)) / 2;
            end
        end

        %Check to see if the point on the axisymmetric point has
        %achieved the desired exit mach number
        if MaSAD(ii,1) >= Mexit
            MaContinue = 1;
        else
            MaContinue = 0;
        end

        %Increase the Number of Characteristics
        NumChar = NumChar + 1;

    else
        %Initialize C- Characteristic values for the calculation of the
        %flowfield point
        ThetaSADtempRight = ThetaSAD((ii-1),1);
        rSADtempRight = rSAD((ii-1),1);
        xSADtempRight = xSAD((ii-1),1);
        AlphaSADtempRight = AlphaSAD((ii-1),1);
        MaSADtempRight = MaSAD((ii-1),1);
        vSADtempRight = vSAD((ii-1),1);

        %Initialize C+ Characteristic values for the calculation of the
        %flowfield point
        ThetaSADtempLeft = ThetaSAD((ii-NumChar),1);
        rSADtempLeft = rSAD((ii-NumChar),1);
        xSADtempLeft = xSAD((ii-NumChar),1);
        AlphaSADtempLeft = AlphaSAD((ii-NumChar),1);
        MaSADtempLeft = MaSAD((ii-NumChar),1);
        vSADtempLeft = vSAD((ii-NumChar),1);

        DeltaTheta = 1.0; %Initialize the following loop
        ThetaLast = 100;

        while DeltaTheta >= 1e-10
            %Calculate the position of the current point
            a = tan(ThetaSADtempRight - AlphaSADtempRight);
            b = tan(ThetaSADtempLeft + AlphaSADtempLeft);
            c = rSADtempRight - (a * xSADtempRight);
            d = rSADtempLeft - (b * xSADtempLeft);
```

```
A = [1 -a; 1 -b];
B = [c; d];

solution = A\B;

rSAD(ii,1) = solution(1,1);
xSAD(ii,1) = solution(2,1);

%Calculate the flow properties of the current point
c = (ThetaSADtempRight + vSADtempRight) + ((1 /...
    ((sqrt(MaSADtempRight^2 - 1)) -...
    (1/tan(ThetaSADtempRight)))) * ((rSAD(ii,1) -...
    rSADtempRight) / rSADtempRight));

if ThetaSADtempLeft == 0.0
    d = (2 * ThetaSADtempLeft) - vSADtempLeft;
    A = [1 1; 2 -1];

else
    d = (ThetaSADtempLeft - vSADtempLeft) - ((1 /...
        ((sqrt(MaSADtempLeft^2 - 1)) +...
        (1/tan(ThetaSADtempLeft)))) * ((rSAD(ii,1) -...
        rSADtempLeft) / rSADtempLeft));
    A = [1 1; 1 -1];
end

B = [c; d];
solution = A\B;

ThetaSAD(ii,1) = solution(1,1);
vSAD(ii,1) = solution(2,1);

%Calculate the Mach Number at the current point
vRad = vSAD(ii,1);

MachG = 1.0;
PMtoMA %calls subprogram to find the Mach Number

MaSAD(ii,1) = Mach;
AlphaSAD(ii,1) = asin(1/MaSAD(ii,1));

if EntroCheck == 1
% Calculate the Pressure, Temperature and change in Entropy
% at the point
    PressSAD(ii,1) = Pstag * ((1 + (((Gamma - 1)/2) *...
               ((MaSAD(ii,1)) ^ 2))) ^ (-Gamma/(Gamma - 1)));

    TempSAD(ii,1) = Tstag * (1 + (((Gamma - 1)/2) *...
              ((MaSAD(ii,1)) ^ 2))) ^ (-1);

    DeltaS(ii,1) = cp * log(TempSAD(ii,1)/Tstag) - Rgas*...
             log(PressSAD(ii,1)/Pstag);
end

%Calculate the change in Theta for the loop
DeltaTheta =  abs(ThetaLast - ThetaSAD(ii,1));

if DeltaTheta > 1e-10
    ThetaLast = ThetaSAD(ii,1);

    %Calculate averages of all values and replace the
    %"temp" values with these. This is an approximation
    %that the charactertistics are curved.

    %C- Characteristic
    ThetaSADtempRight = (ThetaSADtempRight +...
                         ThetaSAD(ii,1)) / 2;
    AlphaSADtempRight = (AlphaSADtempRight +...
                         AlphaSAD(ii,1)) / 2;
    MaSADtempRight = (MaSADtempRight + MaSAD(ii,1)) / 2;
    rSADtempRight = (rSADtempRight + rSAD(ii,1)) / 2;
    xSADtempRight = (xSADtempRight + xSAD(ii,1)) / 2;
```

```
                    vSADtempRight = (vSADtempRight + vSAD(ii,1)) / 2;

                    %C+ Characteristic
                    ThetaSADtempLeft = (ThetaSADtempLeft +...
                                        ThetaSAD(ii,1)) / 2;
                    AlphaSADtempLeft = (AlphaSADtempLeft +...
                                        AlphaSAD(ii,1)) / 2;
                    MaSADtempLeft = (MaSADtempLeft + MaSAD(ii,1)) / 2;
                    rSADtempLeft = (rSADtempLeft + rSAD(ii,1)) / 2;
                    xSADtempLeft = (xSADtempLeft + xSAD(ii,1)) / 2;
                    vSADtempLeft = (vSADtempLeft + vSAD(ii,1)) / 2;
                end
            end
        end
    end
end

%***************************************************************************
%***************************************************************************
%Need to calculate the the points on the contour of the nozzle. We will
%assume that at straight line eminates from the last point on the entrance
%region with a slope of the average of the point and the next point on the
%last calculated characteristic. We will assume that the fluid at the wall
%will have the same properties as the point on the last calculated
%characteristic and the C+ characteristic from this point and the previous
%wall point intersect will be the wall contour point. (This may sound
%confusing)

%Find the first wall point on the wall contour
ii = PointNum + 1; %Initialize the ii loop index and the index of the first
                   %point on the wall contour

%Calculate the first wall contour point
ThetaSAD(ii,1) = ThetaSAD((ii-(NumChar-1)),1);
a = tan(ThetaSAD((ii-(NumChar-1)),1) + AlphaSAD((ii-(NumChar-1)),1));
b = (ThetaSAD((ii-NumChar),1) + ThetaSAD((ii-(NumChar-1)),1)) / 2;
c = rSAD((ii-(NumChar-1)),1) - (a * xSAD((ii-(NumChar-1)),1));
d = rSAD((ii-NumChar),1) - (b * xSAD((ii-NumChar),1));

A = [1 -a; 1 -b];
B = [c; d];

solution = A\B;

rSAD(ii,1) = solution(1,1);
xSAD(ii,1) = solution(2,1);

%Calculate the rest of the wall contour points
for jj = (ii + 1):1:(PointNum+(NumChar-1))
    ThetaSAD(jj,1) = ThetaSAD((jj-(NumChar-1)),1);
    a = tan(ThetaSAD((jj-(NumChar-1)),1) + AlphaSAD((jj-(NumChar-1)),1));
    b = (ThetaSAD((jj-1),1) + ThetaSAD((jj-(NumChar-1)),1)) / 2;
    c = rSAD((jj-(NumChar-1)),1) - (a * xSAD((jj-(NumChar-1)),1));
    d = rSAD((jj-1),1) - (b * xSAD((jj-1),1));

    A = [1 -a; 1 -b];
    B = [c; d];

    solution = A\B;

    rSAD(jj,1) = solution(1,1);
    xSAD(jj,1) = solution(2,1);
end

%***************************************************************************
%***************************************************************************

%Program will now retain only the points that lie on the contour of the
%wall
jj = 1;   %Initialize the index of finding the right point on the wall
Char = 1; %Initialize the variable that holds the temperary characteristic
          %number
```

```
for ii = 1:1:(NumChar-1)
    xSADcontour(ii,1) = xSAD(jj,1);
    rSADcontour(ii,1) = rSAD(jj,1);
    jj = jj + (Char + 1);
    Char = Char + 1;
end

jj = PointNum + 1;
for ii = NumChar:1:((NumChar-1)+(NumChar-1))
    xSADcontour(ii,1) = xSAD(jj,1);
    rSADcontour(ii,1) = rSAD(jj,1);
    jj = jj + 1;
end

%**************************************************************************
%**************************************************************************
%Another approach to calculate for the contour that uses the streamline and
%"backward" calculated characteristics
StreamContinue = 1;

%Initialize Stepsize to reduce the number of wasted iterations of the prog.
%StepSize = PointNum/(NumChar-1);
%StepSizeWhole = fix(StepSize);
%StepSizePart = StepSize - StepSizeWhole;
%StepSizePart = 1 - StepSizePart;
%StepSize = StepSize + StepSizePart + 1;
StepSize = 1.0;

while StreamContinue == 1
    kk = PointNum + 1;
    ThetaSAD(kk,1) = ThetaSAD((kk-1),1);
    vSAD(kk,1) = vSAD((kk-1),1);
    xSAD(kk,1) = xSAD((kk-1),1) + (StepSize * DeltaX);
    MaSAD(kk,1) = MaSAD((kk-1),1);
    AlphaSAD(kk,1) = AlphaSAD((kk-1),1);

    %Calculate the r position of the next point
    rSAD(kk,1) = rSAD((kk-1),1) + (tan(ThetaSAD((kk-1),1) +...
                 AlphaSAD((kk-1),1)) * (xSAD(kk,1) - xSAD((kk-1),1)));

    %Calculate all of the points along the first "backward" characteristic
    zz = 3;
    for ii = (PointNum+2):1:(PointNum+NumChar)
        %Initialize C- Characteristic values for the calculation of the
        %flowfield point
        ThetaSADtempRight = ThetaSAD((ii-1),1);
        rSADtempRight = rSAD((ii-1),1);
        xSADtempRight = xSAD((ii-1),1);
        AlphaSADtempRight = AlphaSAD((ii-1),1);
        MaSADtempRight = MaSAD((ii-1),1);
        vSADtempRight = vSAD((ii-1),1);

        %Initialize C+ Characteristic values for the calculation of the
        %flowfield point
        ThetaSADtempLeft = ThetaSAD((ii-zz),1);
        rSADtempLeft = rSAD((ii-zz),1);
        xSADtempLeft = xSAD((ii-zz),1);
        AlphaSADtempLeft = AlphaSAD((ii-zz),1);
        MaSADtempLeft = MaSAD((ii-zz),1);
        vSADtempLeft = vSAD((ii-zz),1);

        DeltaTheta = 1.0; %Initialize the following loop
        ThetaLast = 100;

        while DeltaTheta >= 1e-10
            %Calculate the position of the current point
            a = tan(ThetaSADtempRight - AlphaSADtempRight);
            b = tan(ThetaSADtempLeft + AlphaSADtempLeft);
            c = rSADtempRight - (a * xSADtempRight);
            d = rSADtempLeft - (b * xSADtempLeft);

            A = [1 -a; 1 -b];
```

```
B = [c; d];

solution = A\B;

rSAD(ii,1) = solution(1,1);
xSAD(ii,1) = solution(2,1);

%Calculate the flow properties of the current point
if ThetaSADtempRight == 0
    c = ThetaSADtempRight + vSADtempRight;
    A = [1 1; 1 -1];
else
    c = (ThetaSADtempRight + vSADtempRight) + ((1 /...
        ((sqrt(MaSADtempRight^2 - 1)) -...
        (1/tan(ThetaSADtempRight)))) * ((rSAD(ii,1) -...
         rSADtempRight) / rSADtempRight));
end

d = (ThetaSADtempLeft - vSADtempLeft) - ((1 /...
    ((sqrt(MaSADtempLeft^2 - 1)) +...
    (1/tan(ThetaSADtempLeft)))) * ((rSAD(ii,1) -...
     rSADtempLeft) / rSADtempLeft));

A = [1 1; 1 -1];
B = [c; d];
solution = A\B;

ThetaSAD(ii,1) = solution(1,1);
vSAD(ii,1) = solution(2,1);

%Calculate the Mach Number at the current point
vRad = vSAD(ii,1);

MachG = 1.0;
PMtoMA %calls subprogram to find the Mach Number

MaSAD(ii,1) = Mach;
AlphaSAD(ii,1) = asin(1/MaSAD(ii,1));

if EntroCheck == 1
% Calculate the Pressure, Temperature and change in Entropy
% at the point
    PressSAD(ii,1) = Pstag * ((1 + (((Gamma - 1)/2) *...
                 ((MaSAD(ii,1)) ^ 2))) ^ (-Gamma/(Gamma - 1)));

    TempSAD(ii,1) = Tstag * (1 + (((Gamma - 1)/2) *...
                  ((MaSAD(ii,1)) ^ 2))) ^ (-1);

    DeltaS(ii,1) = cp * log(TempSAD(ii,1)/Tstag) - Rgas *...
                 log(PressSAD(ii,1)/Pstag);
end

%Calculate the change in Theta for the loop
DeltaTheta =  abs(ThetaLast - ThetaSAD(ii,1));

if DeltaTheta > 1e-10
    ThetaLast = ThetaSAD(ii,1);

    %Calculate averages of all values and replace the "temp"
    %values with these. This is an approximation that the
    %charactertistics are curved.

    %C- Characteristic
    ThetaSADtempRight = (ThetaSADtempRight +...
                        ThetaSAD(ii,1)) / 2;
    AlphaSADtempRight = (AlphaSADtempRight +...
                        AlphaSAD(ii,1)) / 2;
    MaSADtempRight = (MaSADtempRight + MaSAD(ii,1)) / 2;
    rSADtempRight = (rSADtempRight + rSAD(ii,1)) / 2;
    xSADtempRight = (xSADtempRight + xSAD(ii,1)) / 2;
    vSADtempRight = (vSADtempRight + vSAD(ii,1)) / 2;
```

```
                %C+ Characteristic
                ThetaSADtempLeft = (ThetaSADtempLeft + ThetaSAD(ii,1)) / 2;
                AlphaSADtempLeft = (AlphaSADtempLeft + AlphaSAD(ii,1)) / 2;
                MaSADtempLeft = (MaSADtempLeft + MaSAD(ii,1)) / 2;
                rSADtempLeft = (rSADtempLeft + rSAD(ii,1)) / 2;
                xSADtempLeft = (xSADtempLeft + xSAD(ii,1)) / 2;
                vSADtempLeft = (vSADtempLeft + vSAD(ii,1)) / 2;
            end
        end

        zz = zz + 2;
    end

    % This if statement makes sure that the last point calculated in the
    % flow has an r-coordinate greater than the r-coordinate of the last
    % expansion point. If it doesn't the stream function solution will
    % fail.
    if rSAD((PointNum+NumChar),1) > rSAD((PointNum-(NumChar-1)),1)
        StreamContinue = 0;
    else
        StreamContinue = 1;
        StepSize = StepSize + 1;
    end
end

%Calculate the point that satisfies the Stream Function condition at the
%last expansion point on the contour
xStart = xSAD((ii-1),1);
rStart = rSAD((ii-1),1);
ThetaStart = ThetaSAD((ii-1),1);

ThetaEnd = ThetaSAD(ii,1);
rEnd = rSAD(ii,1);
xEnd = xSAD(ii,1);

%Calculate the slope and y-intercept of the straight line that is
%approximating the characteristic between the last two calculated
%characteristic points
A = [xStart 1; xEnd 1];
B = [rStart; rEnd];
solution = A\B;

rSlope = solution(1,1); %Slope of r with respect to x
fr = solution(2,1);     %y-intercept of the r line

B = [ThetaStart; ThetaEnd];
solution = A\B;

ThetaSlope = solution(1,1); %Slope of Theta with respect to x
fTheta = solution(2,1);     %y-intercept of the Theta line

%Initiate values of the last point on the streamline
ThetaLast = ThetaSAD((ii-((2*NumChar)-1)),1);
xLast = xSAD((ii-((2*NumChar)-1)),1);
rLast = rSAD((ii-((2*NumChar)-1)),1);

%Calculate the x- and Theta- coordinate on the line based on the
%r-coordinate of the previous point that satisfies the stream function.
%This is avoid extra computational time due to the fact that the point next
%point that satisfies the stream function must have an r-coordinate greater
%than the previous stream line point's r-coordinate

%Calculate the position of the point the satisfies the streamline condition
a = -tan(ThetaLast);
b = -rSlope;
c = rLast - tan(ThetaLast) * xLast;
d = fr;

A = [1 a; 1 b];
B = [c; d];

solution = A\B;
```

```
rStreamContour(1,1) = solution(1,1);
xStreamContour(1,1) = solution(2,1);

ThetaStreamContour(1,1) = ThetaSlope * xStreamContour(1,1) + fTheta;

%Calculate the rest of the streamline points by continuing the calculation
%until the number of characteristics crossed is equal to 1
ll = 2;     %Index for stream line points
UsedChar = NumChar;
CalcContinue = 1;

if MaSAD((PointNum),1) > 4.5
    StepSize = (2 * StepSize);
else
    Stepsize = StepSize;
end

while CalcContinue ==1
    CharCheck = 1;      %Resets the number of Characteristics calculated
    kk = ii + 1;
    ThetaSAD(kk,1) = ThetaSAD((kk-UsedChar),1);
    vSAD(kk,1) = vSAD((kk-UsedChar),1);
    xSAD(kk,1) = xSAD((kk-UsedChar),1) + ((StepSize) * DeltaX);
    MaSAD(kk,1) = MaSAD((kk-UsedChar),1);
    AlphaSAD(kk,1) = AlphaSAD((kk-UsedChar),1);

    %Calculate the r position of the next point
    rSAD(kk,1) = rSAD((kk-UsedChar),1) + (tan(ThetaSAD((kk-UsedChar),1)+...
        AlphaSAD((kk-UsedChar),1)) * (xSAD(kk,1) - xSAD((kk-UsedChar),1)));

    if rSAD(kk,1) > rStreamContour((ll-1),1)
        CalcContinue = 0;
    else
        ii = ii + 1;    %Increases the ii index to match kk index

        %Calculate all of the points along the first "backward" characteristic
        BackContinue = 1.0;

        while BackContinue == 1.0
            ii = ii + 1;    %Increases the point index

            %Initialize C- Characteristic values for the calculation of the
            %flowfield point
            ThetaSADtempRight = ThetaSAD((ii-1),1);
            rSADtempRight = rSAD((ii-1),1);
            xSADtempRight = xSAD((ii-1),1);
            AlphaSADtempRight = AlphaSAD((ii-1),1);
            MaSADtempRight = MaSAD((ii-1),1);
            vSADtempRight = vSAD((ii-1),1);

            %Initialize C+ Characteristic values for the calculation of the
            %flowfield point
            ThetaSADtempLeft = ThetaSAD((ii-UsedChar),1);
            rSADtempLeft = rSAD((ii-UsedChar),1);
            xSADtempLeft = xSAD((ii-UsedChar),1);
            AlphaSADtempLeft = AlphaSAD((ii-UsedChar),1);
            MaSADtempLeft = MaSAD((ii-UsedChar),1);
            vSADtempLeft = vSAD((ii-UsedChar),1);

            DeltaTheta = 1.0; %Initialize the following loop
            ThetaLast = 100;

            while DeltaTheta >= 1e-10
                %Calculate the position of the current point
                a = tan(ThetaSADtempRight - AlphaSADtempRight);
                b = tan(ThetaSADtempLeft + AlphaSADtempLeft);
                c = rSADtempRight - (a * xSADtempRight);
                d = rSADtempLeft - (b * xSADtempLeft);

                A = [1 -a; 1 -b];
                B = [c; d];
```

```
solution = A\B;

rSAD(ii,1) = solution(1,1);
xSAD(ii,1) = solution(2,1);

%Calculate the flow properties of the current point
if ThetaSADtempRight == 0
    c = ThetaSADtempRight + vSADtempRight;
else
    c = (ThetaSADtempRight + vSADtempRight) + ((1 /...
        ((sqrt(MaSADtempRight^2 - 1)) -...
        (1/tan(ThetaSADtempRight)))) * ((rSAD(ii,1) -...
         rSADtempRight) / rSADtempRight));
end

d = (ThetaSADtempLeft - vSADtempLeft) - ((1 /...
    ((sqrt(MaSADtempLeft^2 - 1)) +...
    (1/tan(ThetaSADtempLeft)))) * ((rSAD(ii,1) -...
     rSADtempLeft) / rSADtempLeft));

A = [1 1; 1 -1];
B = [c; d];
solution = A\B;

ThetaSAD(ii,1) = solution(1,1);
vSAD(ii,1) = solution(2,1);

%Calculate the Mach Number at the current point
vRad = vSAD(ii,1);

MachG = 1.0;
PMtoMA %calls subprogram to find the Mach Number

MaSAD(ii,1) = Mach;
AlphaSAD(ii,1) = asin(1/MaSAD(ii,1));

if EntroCheck == 1
% Calculate the Pressure, Temperature and change in Entropy
% at the point
    PressSAD(ii,1) = Pstag * ((1 + (((Gamma - 1)/2) *...
              ((MaSAD(ii,1)) ^ 2))) ^ (-Gamma/(Gamma - 1)));

    TempSAD(ii,1) = Tstag * (1 + (((Gamma - 1)/2) *...
               ((MaSAD(ii,1)) ^ 2))) ^ (-1);

    DeltaS(ii,1) = cp * log(TempSAD(ii,1)/Tstag) - Rgas*...
              log(PressSAD(ii,1)/Pstag);
end

%Calculate the change in Theta for the loop
DeltaTheta = abs(ThetaLast - ThetaSAD(ii,1));

if DeltaTheta > 1e-10
    ThetaLast = ThetaSAD(ii,1);

    %Calculate averages of all values and replace the
    %"temp" values with these. This is an approximation
    %that the charactertistics are curved.

    %C- Characteristic
    ThetaSADtempRight = (ThetaSADtempRight +...
                         ThetaSAD(ii,1)) / 2;
    AlphaSADtempRight = (AlphaSADtempRight +...
                         AlphaSAD(ii,1)) / 2;
    MaSADtempRight = (MaSADtempRight + MaSAD(ii,1)) / 2;
    rSADtempRight = (rSADtempRight + rSAD(ii,1)) / 2;
    xSADtempRight = (xSADtempRight + xSAD(ii,1)) / 2;
    vSADtempRight = (vSADtempRight + vSAD(ii,1)) / 2;
```

```
            %C+ Characteristic
            ThetaSADtempLeft = (ThetaSADtempLeft +...
                                ThetaSAD(ii,1)) / 2;
            AlphaSADtempLeft = (AlphaSADtempLeft +...
                                AlphaSAD(ii,1)) / 2;
            MaSADtempLeft = (MaSADtempLeft + MaSAD(ii,1)) / 2;
            rSADtempLeft = (rSADtempLeft + rSAD(ii,1)) / 2;
            xSADtempLeft = (xSADtempLeft + xSAD(ii,1)) / 2;
            vSADtempLeft = (vSADtempLeft + vSAD(ii,1)) / 2;
        end
    end

    xLast = xStreamContour((ll-1),1);
    rLast = rStreamContour((ll-1),1);
    ThetaLast = ThetaStreamContour((ll-1),1);

    rCheck = rSAD(ii,1);
    xCheck = xSAD(ii,1);
    ThetaCheck = ThetaSAD(ii,1);

    rCalc = tan(ThetaLast) * (xCheck - xLast) + rLast;

    DeltaR = rCheck - rCalc;

    if DeltaR > 0.0
        BackContinue = 0.0;
    else
        BackContinue = 1.0;
    end

    CharCheck = CharCheck + 1;  %Increases the Number of Used Chars

end

UsedChar = CharCheck;   % Sets UsedChar to the correct value of the
                        % distance of points between any two
                        % "backward" calculated characteristics

%Calculate the point that satisfies the Stream Function condition
%at the last expansion point on the contour
xStart = xSAD((ii-1),1);
rStart = rSAD((ii-1),1);
ThetaStart = ThetaSAD((ii-1),1);

xEnd = xSAD(ii,1);
rEnd = rSAD(ii,1);
ThetaEnd = ThetaSAD(ii,1);

%Calculate the slope and y-intercept of the straight line that is
%approximating the characteristic between the last two calculated
%characteristic points
A = [xStart 1; xEnd 1];
B = [rStart; rEnd];
solution = A\B;

rSlope = solution(1,1); %Slope of r with respect to x
fr = solution(2,1);     %y-intercept of the r line

B = [ThetaStart; ThetaEnd];
solution = A\B;

ThetaSlope = solution(1,1); %Slope of Theta with respect to x
fTheta = solution(2,1);     %y-intercept of the Theta line

%Initiate values of the last point on the streamline
ThetaLast = ThetaStreamContour((ll-1),1);
xLast = xStreamContour((ll-1),1);
rLast = rStreamContour((ll-1),1);

%Calculate the x- and Theta- coordinate on the line based on the
%r-coordinate of the previous point that satisfies the stream
%function. This is avoid extra computational time due to the fact
```

```
        %that the point next point that satisfies the stream function must
        %have an r-coordinate greater than the previous stream line point's
        %r-coordinate

        %Calculate the position of the point the satisfies the streamline
        %condition.

        a = -tan(ThetaLast);
        b = -rSlope;
        c = rLast - tan(ThetaLast) * xLast;
        d = fr;

        A = [1 a; 1 b];
        B = [c; d];

        solution = A\B;

        rStreamContour(ll,1) = solution(1,1);
        xStreamContour(ll,1) = solution(2,1);

        ThetaStreamContour(ll,1) = ThetaSlope*xStreamContour(ll,1)+fTheta;

        ll = ll + 1;    % Increases index variable

        CalcContinue = 1;
    end
end

%Calculate the point that satisfies the Stream Function condition at the
%last expansion point on the contour
xStart = xSAD((kk-UsedChar),1);
rStart = rSAD((kk-UsedChar),1);
ThetaStart = ThetaSAD((kk-UsedChar),1);

xEnd = xSAD(kk,1);
rEnd = rSAD(kk,1);
ThetaEnd = ThetaSAD(kk,1);

%Calculate the slope and y-intercept of the straight line that is
%approximating the characteristic between the last two calculated
%characteristic points
A = [xStart 1; xEnd 1];
B = [rStart; rEnd];
solution = A\B;

rSlope = solution(1,1); %Slope of r with respect to x
fr = solution(2,1);     %y-intercept of the r line

B = [ThetaStart; ThetaEnd];
solution = A\B;

ThetaSlope = solution(1,1); %Slope of Theta with respect to x
fTheta = solution(2,1);     %y-intercept of the Theta line

%Initiate values of the last point on the streamline
ThetaLast = ThetaStreamContour((ll-1),1);
xLast = xStreamContour((ll-1),1);
rLast = rStreamContour((ll-1),1);

%Calculate the position of the point the satisfies the streamline
%condition
a = -tan(ThetaLast);
b = -rSlope;
c = rLast - tan(ThetaLast) * xLast;
d = fr;

A = [1 a; 1 b];
B = [c; d];

solution = A\B;

rStreamContour(ll,1) = solution(1,1);
```

```
xStreamContour(ll,1) = solution(2,1);
ThetaStreamContour(ll,1) = ThetaSlope * xStreamContour(ll,1) + fTheta;
```

B.5 PMtoMA.m Program

Figure B.5.1: PMtoMA.m Program Flowchart

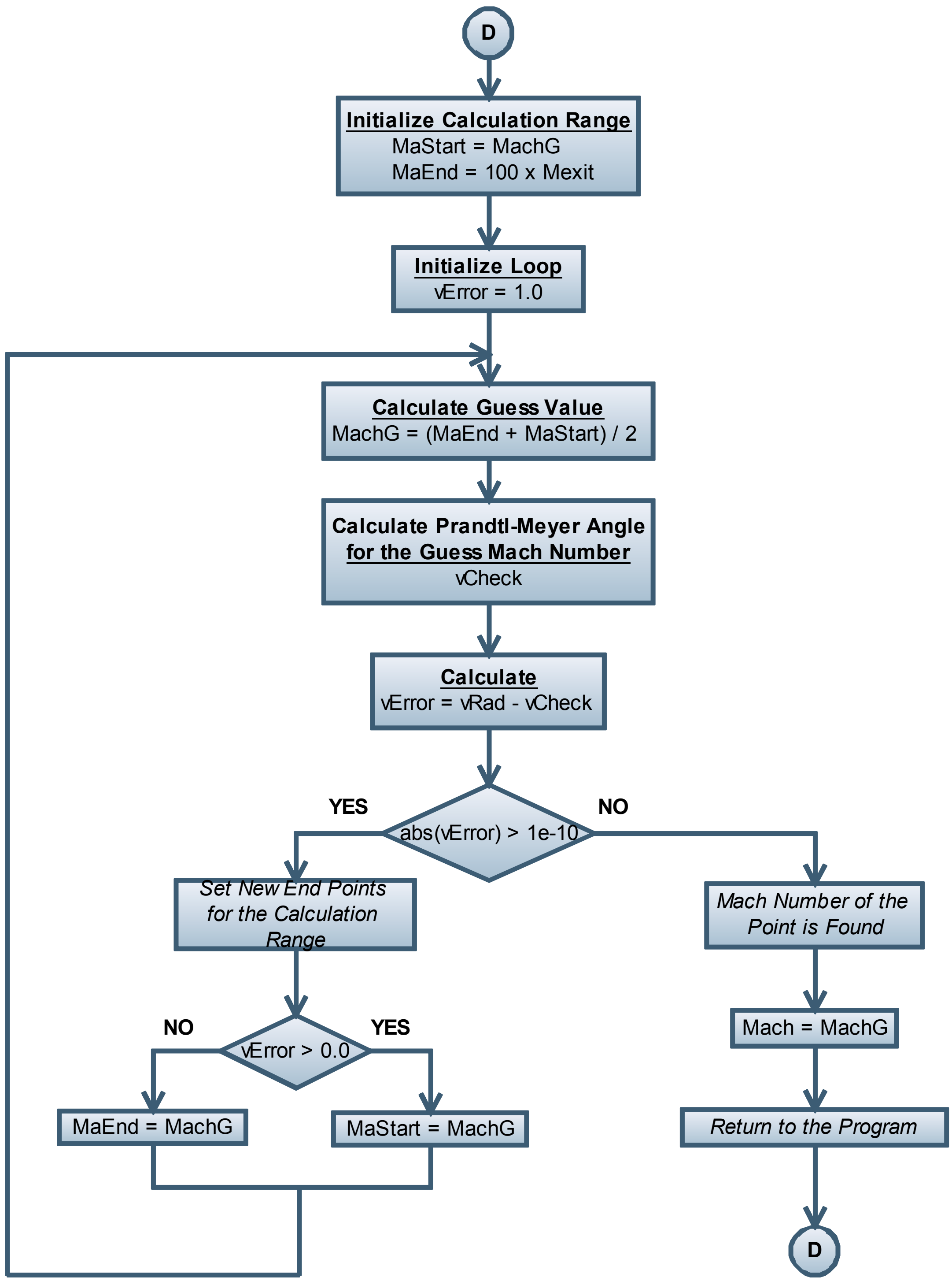


Figure B.5.2: PMtoMA.m MatLab Source Code

```
%**************************************************************************
%* Program will calculate the Mach Number associated with a given
%* Prandtl-Meyer Expansion Angle and Ratio of Specific Heats, Gamma
%*             Brandon Denton
%*             RIT Graduate Student
%*             February 11, 2007
%**************************************************************************
%**************************************************************************
%*     Variable Dictionary
%**************************************************************************
%*  1. Mach -------------------- Points Calculated Mach Number
%*  2. MachG ------------------- Guess Mach Number value
%*  3. MaEnd ------------------- Variable that holds the high end value of
%*                               the calculation range
%*  4. MaStart ----------------- Variable that holds the low end value of
%*                               the calculation range
%*  5. Mexit ------------------- Desired exit Mach Number of the nozzle
%*  6. vCheck ------------------ Variable that holds the Prandtl-Meyer
%*                               angle associated with the guess Mach
%*                               Number
%*  7. vError ------------------ Error and Loop Control Variable
%*  8. vRad -------------------- Acutal Prandtl-Meyer Expansion angle of
%*                               the current point
%**************************************************************************
%**************************************************************************
%*                 Start Program
%**************************************************************************
%**************************************************************************
MaStart = MachG;
MaEnd = 100 * Mexit;

vError = 1.0;

while abs(vError) > 1e-10
    MachG = (MaEnd + MaStart) / 2;

    vCheck = (sqrt((Gamma + 1)/(Gamma - 1))*...
              atan(sqrt(((Gamma - 1)/(Gamma + 1)) *...
              ((MachG^2) - 1))) - atan((sqrt((MachG^2) - 1)));

   vError = vRad - vCheck;

   if abs(vError) > 1e-10
       if vError > 0.0
           MaStart = MachG;
       else
           MaEnd = MachG;
       end
   else
       Mach = MachG;
   end
end
```

Appendix C: Computer Code Accuracy Check Tables

Table C.1	Code Accuracy Check for $\gamma = 1.4$ Ma = 3.0 $\beta = 1.0 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	5.6588	4.2346	33.63%	3.2489	8.30%	5.3635	5.51%
IE Aerospike	5.2253	4.2346	23.40%	3.1278	4.26%	4.7820	9.27%
Aerospike	4.9828	4.2346	17.67%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	5.5827	4.2346	31.84%	3.2455	8.18%	5.3463	4.42%
IE Aerospike	4.7699	4.2346	12.64%	3.0509	1.70%	4.4448	7.31%
Aerospike	4.5875	4.2346	8.33%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	4.4882	4.2346	5.99%	3.0289	0.96%	4.3527	3.11%
IE Aerospike	4.5288	4.2346	6.95%	3.0059	0.20%	4.2584	6.35%
Aerospike	4.3716	4.2346	3.24%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	4.4938	4.2346	6.12%	3.0323	1.08%	4.3668	2.91%
IE Aerospike	4.4528	4.2346	5.15%	3.0059	0.20%	4.2584	4.57%
Aerospike	4.3025	4.2346	1.60%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	4.3940	4.2346	3.76%	3.0093	0.31%	4.2722	2.85%
IE Aerospike	4.4154	4.2346	4.27%	3.0059	0.20%	4.2584	3.69%
Aerospike	4.2684	4.2346	0.80%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	4.3815	4.2346	3.47%	3.0067	0.22%	4.2617	2.81%
IE Aerospike	4.3932	4.2346	3.75%	3.0029	0.10%	4.2463	3.46%
Aerospike	4.2481	4.2346	0.32%	3.0000	0.0%	-	-

Table C.2	Code Accuracy Check for $\gamma = 1.4$ Ma = 3.0 $\beta = 0.5 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	6.1418	4.2346	45.04%	3.3454	11.51%	5.8746	4.55%
IE Aerospike	5.1035	4.2346	20.52%	3.1278	4.26%	4.7820	6.72%
Aerospike	4.9828	4.2346	17.67%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	4.5685	4.2346	7.89%	3.0521	1.74%	4.4499	2.67%
IE Aerospike	4.6689	4.2346	10.26%	3.0509	1.70%	4.4448	5.04%
Aerospike	4.5875	4.2346	8.33%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	4.3815	4.2346	3.47%	3.0127	0.42%	4.2861	2.23%
IE Aerospike	4.4386	4.2346	4.82%	3.0059	0.20%	4.2584	4.23%
Aerospike	4.3716	4.2346	3.24%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	4.4089	4.2346	4.12%	3.0213	0.71%	4.3213	2.03%
IE Aerospike	4.3659	4.2346	3.10%	3.0059	0.20%	4.2584	2.52%
Aerospike	4.3025	4.2346	1.60%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	4.3249	4.2346	2.13%	3.0010	0.03%	4.2386	2.04%
IE Aerospike	4.3302	4.2346	2.26%	3.0059	0.20%	4.2584	1.69%
Aerospike	4.2684	4.2346	0.80%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	4.3341	4.2346	2.35%	3.0021	0.07%	4.2430	2.15%
IE Aerospike	4.3090	4.2346	1.76%	3.0029	0.10%	4.2463	1.48%
Aerospike	4.2481	4.2346	0.32%	3.0000	0.0%	-	-

Table C.3	Code Accuracy Check for $\gamma = 1.4$ Ma = 3.0 $\beta = 0.2 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	6.5010	4.2346	53.52%	3.4096	13.65%	6.2396	4.19%
IE Aerospike	5.0305	4.2346	18.80%	3.1278	4.26%	4.7820	5.20%
Aerospike	4.9828	4.2346	17.67%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	4.7928	4.2346	13.18%	3.1066	3.55%	4.6866	2.27%
IE Aerospike	4.6083	4.2346	8.82%	3.0509	1.70%	4.4448	3.68%
Aerospike	4.5875	4.2346	8.33%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	4.6240	4.2346	9.20%	3.0697	2.32%	4.5250	2.19%
IE Aerospike	4.3844	4.2346	3.54%	3.0059	0.20%	4.2584	2.96%
Aerospike	4.3716	4.2346	3.24%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	4.4195	4.2346	4.37%	3.0251	0.84%	4.3370	1.90%
IE Aerospike	4.3139	4.2346	1.87%	3.0059	0.20%	4.2584	1.30%
Aerospike	4.3025	4.2346	1.60%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	4.3352	4.2346	2.38%	3.0064	0.21%	4.2605	1.75%
IE Aerospike	4.2791	4.2346	1.05%	3.0059	0.20%	4.2584	0.49%
Aerospike	4.2684	4.2346	0.80%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	4.3488	4.2346	2.70%	3.0098	0.33%	4.2743	1.74%
IE Aerospike	4.2585	4.2346	0.56%	3.0029	0.10%	4.2463	0.29%
Aerospike	4.2481	4.2346	0.32%	3.0000	0.0%	-	-

Table C.4	Code Accuracy Check for $\gamma = 1.4$ Ma = 3.5 $\beta = 1.0 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	11.1265	6.7896	63.88%	3.9420	12.63%	10.1763	9.34%
IE Aerospike	8.6315	6.7896	27.13%	3.6438	4.11%	7.7576	11.27%
Aerospike	8.4750	6.7896	24.82%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	7.4947	6.7896	10.39%	3.5516	1.47%	7.1234	5.21%
IE Aerospike	7.6156	6.7896	12.17%	3.5504	1.44%	7.1154	7.03%
Aerospike	7.5685	6.7896	11.47%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	7.7903	6.7896	14.74%	3.6051	3.00%	7.4853	4.07%
IE Aerospike	7.1009	6.7896	4.58%	3.5322	0.92%	6.9962	1.50%
Aerospike	7.0863	6.7896	4.37%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	7.3009	6.7896	7.53%	3.5395	1.13%	7.0438	3.65%
IE Aerospike	6.9429	6.7896	2.26%	3.5140	0.40%	6.8787	0.93%
Aerospike	6.9358	6.7896	2.15%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	7.0795	6.7896	4.27%	3.5080	0.23%	6.8404	3.50%
IE Aerospike	6.8657	6.7896	1.12%	3.5050	0.14%	6.8213	0.65%
Aerospike	6.8623	6.7896	1.07%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	7.0298	6.7896	3.54%	3.5016	0.05%	6.7997	3.38%
IE Aerospike	6.8200	6.7896	0.45%	3.5032	0.09%	6.8099	0.15%
Aerospike	6.8187	6.7896	0.43%	3.5000	0.0%	-	-

Table C.5	Code Accuracy Check for $\gamma = 1.4$ Ma = 3.5 $\beta = 0.5 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	12.8009	6.7896	88.54%	4.1090	17.40%	11.8082	8.41%
IE Aerospike	8.5917	6.7896	26.54%	3.6438	4.11%	7.7576	10.75%
Aerospike	8.4750	6.7896	24.82%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	8.4460	6.7896	24.40%	3.6851	5.29%	8.0581	4.81%
IE Aerospike	7.5979	6.7896	11.90%	3.5504	1.44%	7.1154	6.78%
Aerospike	7.5685	6.7896	11.47%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	7.8328	6.7896	15.36%	3.6153	3.29%	7.5562	3.66%
IE Aerospike	7.0942	6.7896	4.49%	3.5322	0.92%	6.9962	1.40%
Aerospike	7.0863	6.7896	4.37%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	7.3467	6.7896	8.21%	3.5527	1.51%	7.1307	3.03%
IE Aerospike	6.9396	6.7896	2.21%	3.5140	0.40%	6.8787	0.89%
Aerospike	6.9358	6.7896	2.15%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	7.1214	6.7896	4.89%	3.5244	0.70%	6.9456	2.53%
IE Aerospike	6.8641	6.7896	1.10%	3.5050	0.14%	6.8213	0.63%
Aerospike	6.8623	6.7896	1.07%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	7.0274	6.7896	3.50%	3.5098	0.28%	6.8519	2.56%
IE Aerospike	6.8193	6.7896	0.44%	3.5032	0.09%	6.8099	0.14%
Aerospike	6.8187	6.7896	0.43%	3.5000	0.0%	-	-

Table C.6	Code Accuracy Check for $\gamma = 1.4$ Ma = 3.5 $\beta = 0.2 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	14.1239	6.7896	108.02%	4.2233	20.67%	13.0548	8.19%
IE Aerospike	8.5679	6.7896	26.19%	3.6438	4.11%	7.7576	10.45%
Aerospike	8.4750	6.7896	24.82%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	9.1809	6.7896	35.22%	3.7747	7.85%	8.7467	4.96%
IE Aerospike	7.5873	6.7896	11.75%	3.5504	1.44%	7.1154	6.63%
Aerospike	7.5685	6.7896	11.47%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	7.7000	6.7896	13.41%	3.5668	1.91%	7.2245	6.58%
IE Aerospike	7.0902	6.7896	4.43%	3.5322	0.92%	6.9962	1.34%
Aerospike	7.0863	6.7896	4.37%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	7.0353	6.7896	3.62%	3.5080	0.23%	6.8404	2.85%
IE Aerospike	6.9376	6.7896	2.18%	3.5140	0.40%	6.8787	0.86%
Aerospike	6.9358	6.7896	2.15%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	7.0580	6.7896	3.95%	3.5151	0.43%	6.8858	2.50%
IE Aerospike	6.8631	6.7896	1.08%	3.5050	0.14%	6.8213	0.61%
Aerospike	6.8623	6.7896	1.07%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	6.9957	6.7896	3.04%	3.5096	0.27%	6.8506	2.12%
IE Aerospike	6.8189	6.7896	0.43%	3.5032	0.09%	6.8099	0.13%
Aerospike	6.8187	6.7896	0.43%	3.5000	0.0%	-	-

Table C.7	Code Accuracy Check for $\gamma = 1.4$ Ma = 4.0 $\beta = 1.0 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	29.8171	10.7188	178.18%	4.8936	22.34%	22.9479	29.93%
IE Aerospike	14.5843	10.7188	36.06%	4.0537	1.34%	11.2438	29.71%
Aerospike	14.3692	10.7188	34.06%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	14.9533	10.7188	39.51%	4.2890	7.22%	13.8226	8.18%
IE Aerospike	12.4042	10.7188	15.72%	4.0537	1.34%	11.2438	10.32%
Aerospike	12.3244	10.7188	14.98%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	11.4090	10.7188	6.44%	4.0160	0.40%	10.8729	4.93%
IE Aerospike	11.3443	10.7188	5.84%	4.0098	0.25%	10.8129	4.91%
Aerospike	11.3193	10.7188	5.60%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	11.2067	10.7188	4.55%	4.0060	0.15%	10.7763	3.99%
IE Aerospike	11.0249	10.7188	2.86%	4.0098	0.25%	10.8129	1.96%
Aerospike	11.0132	10.7188	2.75%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	11.1343	10.7188	3.88%	4.0003	0.01%	10.7216	3.85%
IE Aerospike	10.8702	10.7188	1.41%	4.0098	0.25%	10.8129	0.53%
Aerospike	10.8649	10.7188	1.36%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	11.2212	10.7188	4.69%	4.0102	0.26%	10.8168	3.74%
IE Aerospike	10.7791	10.7188	0.56%	4.0011	0.03%	10.7293	0.46%
Aerospike	10.7770	10.7188	0.54%	4.0000	0.0%	-	-

Table C.8	Code Accuracy Check for $\gamma = 1.4$ Ma = 4.0 $\beta = 0.5 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	12.8009	10.7188	19.42%	4.1090	2.73%	11.8082	8.41%
IE Aerospike	14.5172	10.7188	35.44%	4.0537	1.34%	11.2438	29.11%
Aerospike	14.3692	10.7188	34.06%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	12.1034	10.7188	12.92%	4.0662	1.66%	11.3692	6.46%
IE Aerospike	12.3754	10.7188	15.46%	4.0537	1.34%	11.2438	10.06%
Aerospike	12.3244	10.7188	14.98%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	11.7283	10.7188	9.42%	4.0490	1.23%	11.1969	4.75%
IE Aerospike	11.3337	10.7188	5.74%	4.0098	0.25%	10.8129	4.82%
Aerospike	11.3193	10.7188	5.60%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	11.5968	10.7188	8.19%	4.0458	1.14%	11.1969	3.57%
IE Aerospike	11.0197	10.7188	2.81%	4.0098	0.25%	10.8129	1.91%
Aerospike	11.0132	10.7188	2.75%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	11.1315	10.7188	3.85%	4.0077	0.19%	10.7927	3.14%
IE Aerospike	10.8676	10.7188	1.39%	4.0098	0.25%	10.8129	0.51%
Aerospike	10.8649	10.7188	1.36%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	11.0310	10.7188	2.91%	4.0017	0.04%	10.7350	2.76%
IE Aerospike	10.7780	10.7188	0.55%	4.0011	0.03%	10.7293	0.45%
Aerospike	10.7770	10.7188	0.54%	4.0000	0.0%	-	-

Table C.9	Code Accuracy Check for $\gamma = 1.4$ Ma $= 4.0$ $\beta = 0.2 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	$\mathrm{Ma}_{\mathrm{Comp}}$	$\mathrm{Ma}_{\%\mathrm{Error}}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	14.1239	10.7188	31.77%	4.2233	5.58%	13.0548	8.19%
IE Aerospike	14.4769	10.7188	35.06%	4.0537	1.34%	11.2438	28.75%
Aerospike	14.3692	10.7188	34.06%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	13.3816	10.7188	24.84%	4.1821	4.55%	12.5927	6.26%
IE Aerospike	12.3582	10.7188	15.29%	4.0537	1.34%	11.2438	9.91%
Aerospike	12.3244	10.7188	14.98%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	11.2442	10.7188	4.90%	4.0061	0.15%	10.7773	4.33%
IE Aerospike	11.3273	10.7188	5.68%	4.0098	0.25%	10.8129	4.76%
Aerospike	11.3193	10.7188	5.60%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	11.2093	10.7188	4.58%	4.0084	0.21%	10.7994	3.80%
IE Aerospike	11.0166	10.7188	2.78%	4.0098	0.25%	10.8129	1.88%
Aerospike	11.0132	10.7188	2.75%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	11.1858	10.7188	4.36%	4.0122	0.30%	10.8361	3.23%
IE Aerospike	10.8661	10.7188	1.37%	4.0098	0.25%	10.8129	0.49%
Aerospike	10.8649	10.7188	1.36%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	11.0236	10.7188	2.84%	4.0029	0.07%	10.7465	2.58%
IE Aerospike	10.7774	10.7188	0.55%	4.0011	0.03%	10.7293	0.45%
Aerospike	10.7770	10.7188	0.54%	4.0000	0.0%	-	-

Table C.10	Code Accuracy Check for $\gamma = 1.4$ Ma = 5.0 $\beta = 1.0 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	104.3507	25.0000	317.40%	6.5061	30.12%	75.4426	38.32%
IE Aerospike	39.7007	25.0000	58.80%	5.1116	2.23%	27.3188	45.32%
Aerospike	38.6480	25.0000	54.59%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	38.7283	25.0000	54.91%	5.2922	5.84%	31.4589	23.11%
IE Aerospike	30.9600	25.0000	23.84%	5.1116	2.23%	27.3188	13.33%
Aerospike	30.6855	25.0000	22.74%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	29.6609	25.0000	18.64%	5.0649	1.30%	26.3271	12.66%
IE Aerospike	27.1445	25.0000	8.58%	5.0488	0.98%	25.9924	4.43%
Aerospike	27.0780	25.0000	8.31%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	27.9865	25.0000	11.95%	5.0287	0.57%	25.5797	9.41%
IE Aerospike	26.0391	25.0000	4.16%	5.0178	0.36%	25.3582	2.69%
Aerospike	26.0096	25.0000	4.04%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	26.9508	25.0000	7.80%	5.0082	0.16%	25.1645	7.10%
IE Aerospike	25.5113	25.0000	2.05%	5.0025	0.05%	25.0500	1.84%
Aerospike	25.4990	25.0000	2.00%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	26.9722	25.0000	7.89%	5.0121	0.24%	25.2430	6.85%
IE Aerospike	25.2029	25.0000	0.81%	5.0055	0.11%	25.1102	0.37%
Aerospike	25.1980	25.0000	0.79%	5.0000	0.0%	-	-

Table C.11	Code Accuracy Check for $\gamma = 1.4$ Ma = 5.0 $\beta = 0.5 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	39.2057	25.0000	56.82%	5.2094	4.19%	29.4991	32.90%
IE Aerospike	39.5180	25.0000	58.07%	5.1116	2.23%	27.3188	44.65%
Aerospike	38.6480	25.0000	54.59%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	30.5321	25.0000	22.13%	5.0345	0.69%	25.6982	18.81%
IE Aerospike	30.8881	25.0000	23.55%	5.1116	2.23%	27.3188	13.07%
Aerospike	30.6855	25.0000	22.74%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	32.3358	25.0000	29.34%	5.1748	3.50%	28.7114	12.62%
IE Aerospike	27.1190	25.0000	8.48%	5.0488	0.98%	25.9924	4.33%
Aerospike	27.0780	25.0000	8.31%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	28.5495	25.0000	14.20%	5.0361	0.72%	25.7310	10.95%
IE Aerospike	26.0268	25.0000	4.11%	5.0178	0.36%	25.3582	2.64%
Aerospike	26.0096	25.0000	4.04%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	27.6746	25.0000	10.70%	5.0226	0.45%	25.4555	8.72%
IE Aerospike	25.5053	25.0000	2.02%	5.0025	0.05%	25.0500	1.82%
Aerospike	25.4990	25.0000	2.00%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	26.9339	25.0000	7.74%	5.0140	0.28%	25.2813	6.54%
IE Aerospike	25.2005	25.0000	0.80%	5.0055	0.11%	25.1102	0.36%
Aerospike	25.1980	25.0000	0.79%	5.0000	0.0%	-	-

Table C.12	Code Accuracy Check for $\gamma = 1.4$ Ma = 5.0 $\beta = 0.2 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	46.4452	25.0000	85.78%	5.4414	8.83%	35.2693	31.69%
IE Aerospike	39.4083	25.0000	57.63%	5.1116	2.23%	27.3188	44.25%
Aerospike	38.6480	25.0000	54.59%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	36.3386	25.0000	45.35%	5.2401	4.80%	30.2133	20.27%
IE Aerospike	30.8451	25.0000	23.38%	5.1116	2.23%	27.3188	12.91%
Aerospike	30.6855	25.0000	22.74%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	32.5420	25.0000	30.17%	5.1537	3.07%	28.2399	15.23%
IE Aerospike	27.1037	25.0000	8.41%	5.0488	0.98%	25.9924	4.28%
Aerospike	27.0780	25.0000	8.31%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	28.3875	25.0000	13.55%	5.0192	0.38%	25.3865	11.82%
IE Aerospike	26.0195	25.0000	4.08%	5.0178	0.36%	25.3582	2.61%
Aerospike	26.0096	25.0000	4.04%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	27.6946	25.0000	10.78%	5.0107	0.21%	25.2148	9.83%
IE Aerospike	25.5017	25.0000	2.01%	5.0025	0.05%	25.0500	1.80%
Aerospike	25.4990	25.0000	2.00%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	27.0873	25.0000	8.35%	5.0071	0.14%	25.1423	7.74%
IE Aerospike	25.1991	25.0000	0.80%	5.0055	0.11%	25.1102	0.35%
Aerospike	25.1980	25.0000	0.79%	5.0000	0.0%	-	-

Table C.13	Code Accuracy Check for $\gamma = 1.22$ Ma = 3.0 $\beta = 1.0 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta \upsilon = 0.05$							
Annular	7.7531	6.3395	22.30%	3.1005	3.35%	7.2573	6.83%
IE Aerospike	7.7948	6.3395	22.96%	3.0181	0.60%	6.4952	20.01%
Aerospike	7.6770	6.3395	21.10%	3.0000	0.0%	-	-
$\Delta \upsilon = 0.025$							
Annular	7.5297	6.3395	18.77%	3.0883	2.94%	7.1387	5.48%
IE Aerospike	7.0079	6.3395	10.54%	3.0181	0.60%	6.4952	7.89%
Aerospike	6.9645	6.3395	9.86%	3.0000	0.0%	-	-
$\Delta \upsilon = 0.01$							
Annular	6.7226	6.3395	6.04%	3.0098	0.33%	6.4233	4.66%
IE Aerospike	6.5947	6.3395	4.03%	3.0181	0.60%	6.4952	1.53%
Aerospike	6.5805	6.3395	3.80%	3.0000	0.0%	-	-
$\Delta \upsilon = 0.005$							
Annular	6.6904	6.3395	5.54%	3.0088	0.29%	6.4147	4.30%
IE Aerospike	6.4655	6.3395	1.99%	3.0075	0.25%	6.4035	0.97%
Aerospike	6.4586	6.3395	1.88%	3.0000	0.0%	-	-
$\Delta \upsilon = 0.0025$							
Annular	6.6783	6.3395	5.34%	3.0084	0.28%	6.4113	4.16%
IE Aerospike	6.4022	6.3395	0.99%	3.0023	0.08%	6.3590	0.68%
Aerospike	6.3988	6.3395	0.94%	3.0000	0.0%	-	-
$\Delta \upsilon = 0.001$							
Annular	6.6056	6.3395	4.20%	3.0011	0.04%	6.3488	4.04%
IE Aerospike	6.3645	6.3395	0.39%	3.0012	0.04%	6.3497	0.23%
Aerospike	6.3632	6.3395	0.37%	3.0000	0.0%	-	-

Table C.14	Code Accuracy Check for $\gamma = 1.22$ Ma = 3.0 $\beta = 0.5 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	8.6122	6.3395	35.85%	3.1887	6.29%	8.1777	5.31%
IE Aerospike	7.7543	6.3395	22.32%	3.0181	0.60%	6.4952	19.39%
Aerospike	7.6770	6.3395	21.10%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	8.4234	6.3395	32.87%	3.1817	6.06%	8.1004	3.99%
IE Aerospike	6.9896	6.3395	10.25%	3.0181	0.60%	6.4952	7.61%
Aerospike	6.9645	6.3395	9.86%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	6.7566	6.3395	6.58%	3.0260	0.87%	6.5645	2.93%
IE Aerospike	6.5877	6.3395	3.92%	3.0181	0.60%	6.4952	1.42%
Aerospike	6.5805	6.3395	3.80%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	6.7851	6.3395	7.03%	3.0288	0.96%	6.5892	2.97%
IE Aerospike	6.4621	6.3395	1.93%	3.0075	0.25%	6.4035	0.92%
Aerospike	6.4586	6.3395	1.88%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	6.6447	6.3395	4.81%	3.0128	0.43%	6.4492	3.03%
IE Aerospike	6.4005	6.3395	0.96%	3.0023	0.08%	6.3590	0.65%
Aerospike	6.3988	6.3395	0.94%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	6.5745	6.3395	3.71%	3.0045	0.15%	6.3778	3.08%
IE Aerospike	6.3638	6.3395	0.38%	3.0012	0.04%	6.3497	0.22%
Aerospike	6.3632	6.3395	0.37%	3.0000	0.0%	-	-

Table C.15	Code Accuracy Check for $\gamma = 1.22$ Ma = 3.0 $\beta = 0.2 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	9.2671	6.3395	46.18%	3.2474	8.25%	8.8568	4.63%
IE Aerospike	7.7301	6.3395	21.94%	3.0181	0.60%	6.4952	19.01%
Aerospike	7.6770	6.3395	21.10%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	6.8330	6.3395	7.78%	3.0360	1.20%	6.6533	2.70%
IE Aerospike	6.9786	6.3395	10.08%	3.0181	0.60%	6.4952	7.44%
Aerospike	6.9645	6.3395	9.86%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	6.5386	6.3395	3.14%	3.0049	0.16%	6.3812	2.47%
IE Aerospike	6.5835	6.3395	3.85%	3.0181	0.60%	6.4952	1.36%
Aerospike	6.5805	6.3395	3.80%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	6.5801	6.3395	3.80%	3.0106	0.35%	6.4302	2.33%
IE Aerospike	6.4600	6.3395	1.90%	3.0075	0.25%	6.4035	0.88%
Aerospike	6.4586	6.3395	1.88%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	6.6201	6.3395	4.43%	3.0153	0.51%	6.4709	2.31%
IE Aerospike	6.3995	6.3395	0.95%	3.0023	0.08%	6.3590	0.64%
Aerospike	6.3988	6.3395	0.94%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	6.5409	6.3395	3.18%	3.0049	0.16%	6.3812	2.50%
IE Aerospike	6.3634	6.3395	0.38%	3.0012	0.04%	6.3497	0.22%
Aerospike	6.3632	6.3395	0.37%	3.0000	0.0%	-	-

Table C.16	Code Accuracy Check for $\gamma = 1.22$ Ma = 3.5 $\beta = 1.0 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	13.9993	12.5064	11.94%	3.5222	0.63%	12.8924	8.59%
IE Aerospike	16.7004	12.5064	33.53%	3.5953	2.72%	14.2500	17.20%
Aerospike	16.4788	12.5064	31.76%	3.5000	0.00%	-	-
$\Delta\upsilon = 0.025$							
Annular	17.6998	12.5064	41.53%	3.7041	5.83%	16.5392	7.02%
IE Aerospike	14.3632	12.5064	14.85%	3.5329	0.94%	13.0827	9.79%
Aerospike	14.2761	12.5064	14.15%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	14.1203	12.5064	12.90%	3.5487	1.39%	13.3689	5.62%
IE Aerospike	13.2012	12.5064	5.56%	3.5205	0.59%	12.8624	2.63%
Aerospike	13.1722	12.5064	5.32%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	13.9141	12.5064	11.26%	3.5404	1.15%	13.2178	5.27%
IE Aerospike	12.8473	12.5064	2.73%	3.5082	0.23%	12.6476	1.58%
Aerospike	12.8333	12.5064	2.61%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	13.4168	12.5064	7.28%	3.5155	0.44%	12.7746	5.03%
IE Aerospike	12.6754	12.5064	1.35%	3.5021	0.06%	12.5424	1.06%
Aerospike	12.6685	12.5064	1.30%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	13.2472	12.5064	5.92%	3.5077	0.22%	12.6389	4.81%
IE Aerospike	12.5737	12.5064	0.54%	3.5009	0.03%	12.5218	0.41%
Aerospike	12.5710	12.5064	0.52%	3.5000	0.0%	-	-

Table C.17	Code Accuracy Check for $\gamma = 1.22$ Ma = 3.5 $\beta = 0.5 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	16.5578	12.5064	32.39%	3.6492	4.26%	15.3417	7.93%
IE Aerospike	16.6138	12.5064	32.84%	3.5953	2.72%	14.2500	16.59%
Aerospike	16.4788	12.5064	31.76%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	15.4811	12.5064	23.79%	3.6148	3.28%	14.6357	5.78%
IE Aerospike	14.3256	12.5064	14.55%	3.5329	0.94%	13.0827	9.50%
Aerospike	14.2761	12.5064	14.15%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	13.2157	12.5064	5.67%	3.5111	0.32%	12.6979	4.08%
IE Aerospike	13.1872	12.5064	5.44%	3.5205	0.59%	12.8624	2.53%
Aerospike	13.1722	12.5064	5.32%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	13.0814	12.5064	4.60%	3.5080	0.23%	12.6441	3.46%
IE Aerospike	12.8405	12.5064	2.67%	3.5082	0.23%	12.6476	1.53%
Aerospike	12.8333	12.5064	2.61%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	13.0673	12.5064	4.48%	3.5069	0.20%	12.6251	3.50%
IE Aerospike	12.6720	12.5064	1.32%	3.5021	0.06%	12.5424	1.03%
Aerospike	12.6685	12.5064	1.30%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	13.0726	12.5064	4.53%	3.5067	0.19%	12.6216	3.57%
IE Aerospike	12.5724	12.5064	0.53%	3.5009	0.03%	12.5218	0.40%
Aerospike	12.5710	12.5064	0.52%	3.5000	0.0%	-	-

Table C.18	Code Accuracy Check for $\gamma = 1.22$ Ma = 3.5 $\beta = 0.2 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	18.3641	12.5064	46.84%	3.7357	6.73%	17.2700	6.34%
IE Aerospike	16.5618	12.5064	32.43%	3.5953	2.72%	14.2500	16.22%
Aerospike	16.4788	12.5064	31.76%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	17.3586	12.5064	38.80%	3.7017	5.76%	16.4849	5.30%
IE Aerospike	14.3030	12.5064	14.37%	3.5329	0.94%	13.0827	9.33%
Aerospike	14.2761	12.5064	14.15%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	13.0676	12.5064	4.49%	3.5043	0.12%	12.5802	3.87%
IE Aerospike	13.1789	12.5064	5.38%	3.5205	0.59%	12.8624	2.46%
Aerospike	13.1722	12.5064	5.32%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	13.0037	12.5064	3.98%	3.5042	0.12%	12.5785	3.38%
IE Aerospike	12.8364	12.5064	2.64%	3.5082	0.23%	12.6476	1.49%
Aerospike	12.8333	12.5064	2.61%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	12.9694	12.5064	3.70%	3.5054	0.15%	12.5992	2.94%
IE Aerospike	12.6700	12.5064	1.31%	3.5021	0.06%	12.5424	1.02%
Aerospike	12.6685	12.5064	1.30%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	12.9923	12.5064	3.89%	3.5076	0.22%	12.6372	2.81%
IE Aerospike	12.5716	12.5064	0.52%	3.5009	0.03%	12.5218	0.40%
Aerospike	12.5710	12.5064	0.52%	3.5000	0.0%	-	-

Table C.19	Code Accuracy Check for $\gamma = 1.22$ Ma = 4.0 $\beta = 1.0 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	32.8311	24.7658	32.57%	4.0153	0.38%	25.2858	29.84%
IE Aerospike	36.1181	24.7658	45.84%	4.1426	3.56%	30.0445	20.22%
Aerospike	35.5924	24.7658	43.72%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	34.7134	24.7658	40.17%	4.1936	4.84%	32.1850	7.86%
IE Aerospike	29.6490	24.7658	19.72%	4.0690	1.72%	27.1966	9.02%
Aerospike	29.4527	24.7658	18.92%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	29.4527	24.7658	18.92%	4.0854	2.13%	27.8076	5.92%
IE Aerospike	26.5715	24.7658	7.29%	4.0257	0.64%	25.6454	3.61%
Aerospike	26.5126	24.7658	7.05%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	26.8747	24.7658	8.52%	4.0193	0.48%	25.4235	5.71%
IE Aerospike	25.6466	24.7658	3.56%	4.0113	0.28%	25.1489	1.98%
Aerospike	25.6183	24.7658	3.44%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	26.4152	24.7658	6.66%	4.0089	0.22%	25.0670	5.38%
IE Aerospike	25.2012	24.7658	1.76%	4.0042	0.10%	24.9075	1.18%
Aerospike	25.1873	24.7658	1.70%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	26.0586	24.7658	5.22%	4.0006	0.02%	24.7860	5.13%
IE Aerospike	24.939	24.7658	0.70%	4.0028	0.07%	24.8602	0.32%
Aerospike	24.9334	24.7658	0.68%	4.0000	0.0%	-	-

Table C.20	Code Accuracy Check for $\gamma = 1.22$ Ma = 4.0 $\beta = 0.5 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	39.0806	24.7658	57.80%	4.2074	5.18%	32.7890	19.19%
IE Aerospike	35.9308	24.7658	45.08%	4.1426	3.56%	30.0445	19.59%
Aerospike	35.5924	24.7658	43.72%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	31.1019	24.7658	25.58%	4.1180	2.95%	29.0619	7.02%
IE Aerospike	29.5714	24.7658	19.40%	4.0690	1.72%	27.1966	8.73%
Aerospike	29.4527	24.7658	18.92%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	28.8462	24.7658	16.48%	4.0732	1.83%	27.3518	5.46%
IE Aerospike	26.5433	24.7658	7.18%	4.0257	0.64%	25.6454	3.50%
Aerospike	26.5126	24.7658	7.05%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	26.1801	24.7658	5.71%	4.0094	0.24%	25.0840	4.37%
IE Aerospike	25.6330	24.7658	3.50%	4.0113	0.28%	25.1489	1.92%
Aerospike	25.6183	24.7658	3.44%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	25.7534	24.7658	3.99%	4.0026	0.07%	24.8534	3.62%
IE Aerospike	25.1945	24.7658	1.73%	4.0042	0.10%	24.9075	1.15%
Aerospike	25.1873	24.7658	1.70%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	25.9415	24.7658	4.75%	4.0075	0.19%	25.0194	3.69%
IE Aerospike	24.9361	24.7658	0.69%	4.0028	0.07%	24.8602	0.31%
Aerospike	24.9334	24.7658	0.68%	4.0000	0.0%	-	-

Table C.21	Code Accuracy Check for $\gamma = 1.22$ Ma $= 4.0$ $\beta = 0.2 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	44.2644	24.7658	78.73%	4.3471	8.68%	39.5513	11.92%
IE Aerospike	35.8184	24.7658	44.63%	4.1428	3.57%	30.0526	19.19%
Aerospike	35.5924	24.7658	43.72%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	35.5167	24.7658	43.41%	4.2447	6.12%	34.4769	3.02%
IE Aerospike	29.5248	24.7658	19.22%	4.0690	1.72%	27.1966	8.56%
Aerospike	29.4527	24.7658	18.92%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	29.1450	24.7658	17.68%	4.0867	2.17%	27.8566	4.63%
IE Aerospike	26.5265	24.7658	7.11%	4.0257	0.64%	25.6454	3.44%
Aerospike	26.5126	24.7658	7.05%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	26.7126	24.7658	7.86%	4.0235	0.59%	25.5689	4.47%
IE Aerospike	25.6248	24.7658	3.47%	4.0113	0.28%	25.1489	1.89%
Aerospike	25.6183	24.7658	3.44%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	26.3612	24.7658	6.44%	4.0188	0.47%	25.4063	3.76%
IE Aerospike	25.1904	24.7658	1.71%	4.0042	0.10%	24.9075	1.14%
Aerospike	25.1873	24.7658	1.70%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	25.7548	24.7658	3.99%	4.0065	0.16%	24.9854	3.08%
IE Aerospike	24.9345	24.7658	0.68%	4.0028	0.07%	24.8602	0.30%
Aerospike	24.9334	24.7658	0.68%	4.0000	0.0%	-	-

Table C.22	Code Accuracy Check for $\gamma = 1.22$ Ma = 5.0 $\beta = 1.0 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	**Failed**	93.0258	***	***	***	***	***
IE Aerospike	168.4724	93.0258	81.10%	5.1788	3.58%	116.7639	44.28%
Aerospike	165.2100	93.0258	77.60%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	**Failed**	93.0258	***	***	***	***	***
IE Aerospike	122.8573	93.0258	32.07%	5.0796	1.59%	102.9715	19.31%
Aerospike	121.8200	93.0258	30.95%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	**Failed**	93.0258	***	***	***	***	***
IE Aerospike	103.5807	93.0258	11.35%	5.0213	0.43%	95.5949	8.35%
Aerospike	103.2900	93.0258	11.03%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	162.1161	93.0258	74.27%	5.3060	6.12%	136.9851	18.35%
IE Aerospike	98.0944	93.0258	5.45%	5.0021	0.04%	93.2762	5.17%
Aerospike	97.9795	93.0258	5.33%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	106.0026	93.0258	13.95%	5.0342	0.68%	97.1830	9.08%
IE Aerospike	95.5155	93.0258	2.68%	5.0021	0.04%	93.2762	2.40%
Aerospike	95.4623	93.0258	2.62%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	102.8623	93.0258	10.57%	5.0165	0.33%	95.0102	8.26%
IE Aerospike	94.0116	93.0258	1.06%	5.0021	0.04%	93.2762	0.79%
Aerospike	93.9913	93.0258	1.04%	5.0000	0.0%	-	-

Table C.23	Code Accuracy Check for $\gamma = 1.22$ Ma = 5.0 $\beta = 0.5 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	**Failed**	93.0258	***	***	***	***	***
IE Aerospike	167.5986	93.0258	80.16%	5.1788	3.58%	116.7639	43.54%
Aerospike	165.2121	93.0258	77.60%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	**Failed**	93.0258	***	***	***	***	***
IE Aerospike	122.5356	93.0258	31.72%	5.0796	1.59%	102.9715	19.00%
Aerospike	121.8209	93.0258	30.95%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	101.6893	93.0258	9.31%	5.0081	0.16%	93.9951	8.19%
IE Aerospike	103.4711	93.0258	11.23%	5.0213	0.43%	95.5949	8.24%
Aerospike	103.2935	93.0258	11.04%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	105.7671	93.0258	13.70%	5.0406	0.81%	97.9801	7.95%
IE Aerospike	98.0423	93.0258	5.39%	5.0021	0.04%	93.2762	5.11%
Aerospike	97.9795	93.0258	5.33%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	115.1296	93.0258	23.76%	5.0934	1.87%	104.7942	9.86%
IE Aerospike	95.4901	93.0258	2.65%	5.0021	0.04%	93.2762	2.37%
Aerospike	95.4623	93.0258	2.62%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	100.2745	93.0258	7.79%	5.0097	0.19%	94.1876	6.46%
IE Aerospike	94.0016	93.0258	1.05%	5.0021	0.04%	93.2762	0.78%
Aerospike	93.9913	93.0258	1.04%	5.0000	0.0%	-	-

Table C.24	Code Accuracy Check for $\gamma = 1.22$ Ma = 5.0 $\beta = 0.2 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	**Failed**	93.0258	***	***	***	***	***
IE Aerospike	167.0744	93.0258	79.60%	5.1788	3.58%	116.7639	43.09%
Aerospike	165.2122	93.0258	77.60%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	**Failed**	93.0258	***	***	***	***	***
IE Aerospike	122.3425	93.0258	31.51%	5.0796	1.59%	102.9715	18.81%
Aerospike	121.8209	93.0258	30.95%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	**Failed**	93.0258	***	***	***	***	***
IE Aerospike	103.4053	93.0258	11.16%	5.0213	0.43%	95.5949	8.17%
Aerospike	103.2935	93.0258	11.04%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	**Failed**	93.0258	***	***	***	***	***
IE Aerospike	98.0111	93.0258	5.36%	5.0021	0.04%	93.2762	5.08%
Aerospike	97.9795	93.0258	5.33%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	105.7238	93.0258	13.65%	5.0043	0.09%	93.5392	13.03%
IE Aerospike	95.4749	93.0258	2.63%	5.0021	0.04%	93.2762	2.36%
Aerospike	95.4623	93.0258	2.62%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	101.7451	93.0258	9.37%	5.0018	0.04%	93.2404	9.12%
IE Aerospike	93.9956	93.0258	1.04%	5.0021	0.04%	93.2762	0.77%
Aerospike	93.9913	93.0258	1.04%	5.0000	0.0%	-	-

Table C.25	Code Accuracy Check for $\gamma = 1.26$ Ma = 3.0 $\beta = 1.0 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	8.2763	5.6820	45.66%	3.2516	8.39%	7.7534	***
IE Aerospike	6.9986	5.6820	23.17%	3.1062	3.54%	6.4763	8.06%
Aerospike	6.8260	5.6820	20.13%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	6.1580	5.6820	8.38%	3.0253	0.84%	5.8615	5.06%
IE Aerospike	6.3261	5.6820	11.34%	3.0472	1.57%	6.0217	5.06%
Aerospike	6.2210	5.6820	9.49%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	6.4018	5.6820	12.67%	3.0631	2.10%	6.1409	4.25%
IE Aerospike	5.9688	5.6820	5.05%	3.0124	0.41%	5.7692	3.46%
Aerospike	5.8885	5.6820	3.63%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	6.0797	5.6820	7.00%	3.0231	0.77%	5.8457	4.00%
IE Aerospike	5.8559	5.6820	3.06%	3.0008	0.03%	5.6875	2.96%
Aerospike	5.7843	5.6820	1.80%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	5.9272	5.6820	4.32%	3.0038	0.13%	5.7085	3.83%
IE Aerospike	5.8009	5.6820	2.09%	3.0008	0.03%	5.6875	1.99%
Aerospike	5.7330	5.6820	0.90%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	5.8646	5.6820	3.21%	3.0032	0.11%	5.7043	2.81%
IE Aerospike	5.7681	5.6820	1.52%	3.0008	0.03%	5.6875	1.42%
Aerospike	5.7024	5.6820	0.36%	3.0000	0.0%	-	-

Table C.26	Code Accuracy Check for $\gamma = 1.26$ Ma = 3.0 $\beta = 0.5 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	9.2388	5.6820	62.60%	3.3519	11.73%	8.7807	5.22%
IE Aerospike	6.9238	5.6820	21.85%	3.1062	3.54%	6.4763	6.91%
Aerospike	6.8260	5.6820	20.13%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	6.7490	5.6820	18.78%	3.1137	3.79%	6.5366	3.25%
IE Aerospike	6.2737	5.6820	10.41%	3.0472	1.57%	6.0217	4.18%
Aerospike	6.2210	5.6820	9.49%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	6.4097	5.6820	12.81%	3.0752	2.51%	6.2332	2.83%
IE Aerospike	5.9282	5.6820	4.33%	3.0124	0.41%	5.7692	2.76%
Aerospike	5.8885	5.6820	3.63%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	6.1121	5.6820	7.57%	3.0372	1.24%	5.9480	2.76%
IE Aerospike	5.8190	5.6820	2.41%	3.0008	0.03%	5.6875	2.31%
Aerospike	5.7843	5.6820	1.80%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	5.9863	5.6820	5.36%	3.0201	0.67%	5.8241	2.78%
IE Aerospike	5.7658	5.6820	1.47%	3.0008	0.03%	5.6875	1.38%
Aerospike	5.7330	5.6820	0.90%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	5.8673	5.6820	3.26%	3.0034	0.11%	5.7057	2.83%
IE Aerospike	5.7341	5.6820	0.92%	3.0008	0.03%	5.6875	0.82%
Aerospike	5.7024	5.6820	0.36%	3.0000	0.0%	-	-

Table C.27	Code Accuracy Check for $\gamma = 1.26$ Ma = 3.0 $\beta = 0.2 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	9.9796	5.6820	75.64%	3.4189	13.96%	9.5420	4.59%
IE Aerospike	6.8789	5.6820	21.06%	3.1062	3.54%	6.4763	6.22%
Aerospike	6.8260	5.6820	20.13%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	7.2429	5.6820	27.47%	3.1723	5.74%	7.0279	3.06%
IE Aerospike	6.2423	5.6820	9.86%	3.0472	1.57%	6.0217	3.66%
Aerospike	6.2210	5.6820	9.49%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	6.1715	5.6820	8.61%	3.0480	1.60%	6.0277	2.39%
IE Aerospike	5.9039	5.6820	3.91%	3.0124	0.41%	5.7692	2.33%
Aerospike	5.8885	5.6820	3.63%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	5.8929	5.6820	3.71%	3.0121	0.40%	5.7671	2.18%
IE Aerospike	5.7969	5.6820	2.02%	3.0008	0.03%	5.6875	1.92%
Aerospike	5.7843	5.6820	1.80%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	5.9301	5.6820	4.37%	3.0178	0.59%	5.8077	2.11%
IE Aerospike	5.7447	5.6820	1.10%	3.0008	0.03%	5.6875	1.01%
Aerospike	5.7330	5.6820	0.90%	3.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	5.8620	5.6820	3.17%	3.0068	0.23%	5.7296	2.31%
IE Aerospike	5.7137	5.6820	0.56%	3.0008	0.03%	5.6875	0.46%
Aerospike	5.7024	5.6820	0.36%	3.0000	0.0%	-	-

Table C.28	Code Accuracy Check for $\gamma = 1.26$ Ma = 3.5 $\beta = 1.0 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	15.6635	10.5519	48.44%	3.7476	7.07%	14.3311	9.30%
IE Aerospike	13.8629	10.5519	31.38%	3.6234	3.53%	12.2942	12.76%
Aerospike	13.6402	10.5519	29.27%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	14.4217	10.5519	36.67%	3.6998	5.71%	13.5110	6.74%
IE Aerospike	12.0353	10.5519	14.06%	3.5538	1.54%	11.2794	6.70%
Aerospike	11.9654	10.5519	13.40%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	12.2501	10.5519	16.09%	3.5796	2.27%	11.6456	5.19%
IE Aerospike	11.1105	10.5519	5.29%	3.5128	0.37%	10.7207	3.64%
Aerospike	11.0837	10.5519	5.04%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	11.4356	10.5519	8.37%	3.5267	0.76%	10.9070	4.85%
IE Aerospike	10.8254	10.5519	2.59%	3.5128	0.37%	10.7207	0.98%
Aerospike	10.8140	10.5519	2.48%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	11.0441	10.5519	4.66%	3.5004	0.01%	10.5571	4.61%
IE Aerospike	10.6876	10.5519	1.29%	3.5060	0.17%	10.6307	0.54%
Aerospike	10.6820	10.5519	1.23%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	11.0442	10.5519	4.67%	3.5018	0.05%	10.5755	4.43%
IE Aerospike	10.6060	10.5519	0.51%	3.5019	0.05%	10.5768	0.28%
Aerospike	10.6038	10.5519	0.49%	3.5000	0.0%	-	-

Table C.29	Code Accuracy Check for $\gamma = 1.26$ Ma = 3.5 $\beta = 0.5 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	18.6715	10.5519	76.95%	3.8986	11.39%	17.2508	8.24%
IE Aerospike	13.7932	10.5519	30.72%	3.6234	3.53%	12.2942	12.19%
Aerospike	13.6402	10.5519	29.27%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	12.3389	10.5519	16.94%	3.5850	2.43%	11.7237	5.25%
IE Aerospike	12.0046	10.5519	13.77%	3.5538	1.54%	11.2794	6.43%
Aerospike	11.9654	10.5519	13.40%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	11.3209	10.5519	7.29%	3.5252	0.72%	10.8867	3.99%
IE Aerospike	11.0990	10.5519	5.18%	3.5128	0.37%	10.7207	3.53%
Aerospike	11.0837	10.5519	5.04%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	11.2323	10.5519	6.45%	3.5235	0.67%	10.8638	3.39%
IE Aerospike	10.8198	10.5519	2.54%	3.5128	0.37%	10.7207	0.92%
Aerospike	10.8140	10.5519	2.48%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	11.2138	10.5519	6.27%	3.5234	0.67%	10.8625	3.23%
IE Aerospike	10.6849	10.5519	1.26%	3.5060	0.17%	10.6307	0.51%
Aerospike	10.6820	10.5519	1.23%	3.5000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	10.9750	10.5519	4.01%	3.5055	0.16%	10.6241	3.30%
IE Aerospike	10.6049	10.5519	0.50%	3.5019	0.05%	10.5768	0.27%
Aerospike	10.6038	10.5519	0.49%	3.5000	0.0%	-	-

Table C.30	Code Accuracy Check for $\gamma = 1.26$ Ma = 3.5 $\beta = 0.2 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta \upsilon = 0.05$							
Annular	20.6901	10.5519	96.08%	4.0025	14.36%	19.5835	5.65%
IE Aerospike	13.7514	10.5519	30.32%	3.6234	3.53%	12.2942	11.85%
Aerospike	13.6402	10.5519	29.27%	3.5000	0.0%	-	-
$\Delta \upsilon = 0.025$							
Annular	13.6769	10.5519	29.62%	3.6698	4.85%	13.0198	5.05%
IE Aerospike	11.9862	10.5519	13.59%	3.5538	1.54%	11.2794	6.27%
Aerospike	11.9654	10.5519	13.40%	3.5000	0.0%	-	-
$\Delta \upsilon = 0.01$							
Annular	11.0728	10.5519	4.94%	3.5096	0.27%	10.6782	3.70%
IE Aerospike	11.0922	10.5519	5.12%	3.5128	0.37%	10.7207	3.47%
Aerospike	11.0837	10.5519	5.04%	3.5000	0.0%	-	-
$\Delta \upsilon = 0.005$							
Annular	11.0532	10.5519	4.75%	3.5113	0.32%	10.7008	3.29%
IE Aerospike	10.8164	10.5519	2.51%	3.5128	0.37%	10.7207	0.89%
Aerospike	10.8140	10.5519	2.48%	3.5000	0.0%	-	-
$\Delta \upsilon = 0.0025$							
Annular	11.0397	10.5519	4.62%	3.5138	0.39%	10.7340	2.85%
IE Aerospike	10.6832	10.5519	1.24%	3.5060	0.17%	10.6307	0.49%
Aerospike	10.6820	10.5519	1.23%	3.5000	0.0%	-	-
$\Delta \upsilon = 0.001$							
Annular	10.9311	10.5519	3.59%	3.5076	0.22%	10.6518	2.62%
IE Aerospike	10.6043	10.5519	0.50%	3.5019	0.05%	10.5768	0.26%
Aerospike	10.6038	10.5519	0.49%	3.5000	0.0%	-	-

Table C.31	Code Accuracy Check for $\gamma = 1.26$ Ma $= 4.0$ $\beta = 1.0 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	34.6496	19.5240	77.47%	4.3577	8.94%	30.0380	15.35%
IE Aerospike	28.0541	19.5240	43.69%	4.0777	1.94%	21.4568	30.75%
Aerospike	27.5300	19.5240	41.01%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	**Failed**	19.5240	***	***	***	***	***
IE Aerospike	23.1400	19.5240	18.52%	4.0776	1.94%	21.4542	7.86%
Aerospike	22.9987	19.5240	17.80%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	22.7772	19.5240	16.66%	4.0809	2.02%	21.5402	5.74%
IE Aerospike	20.8685	19.5240	6.89%	4.0131	0.33%	19.8378	5.20%
Aerospike	20.8159	19.5240	6.62%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	20.8408	19.5240	6.74%	4.0112	0.28%	19.7920	5.30%
IE Aerospike	20.1790	19.5240	3.35%	4.0131	0.33%	19.8378	1.72%
Aerospike	20.1574	19.5240	3.24%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	20.5488	19.5240	5.25%	4.0019	0.05%	19.5692	5.01%
IE Aerospike	19.8480	19.5240	1.66%	4.0051	0.13%	19.6456	1.03%
Aerospike	19.8374	19.5240	1.61%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	20.5754	19.5240	5.39%	4.0048	0.12%	19.6385	4.77%
IE Aerospike	19.6529	19.5240	0.66%	4.0003	0.01%	19.5312	0.62%
Aerospike	19.6487	19.5240	0.64%	4.0000	0.0%	-	-

Table C.32	Code Accuracy Check for $\gamma = 1.26$ Ma = 4.0 $\beta = 0.5 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	50.0512	19.5240	156.36%	4.6051	15.13%	40.2136	24.46%
IE Aerospike	27.9131	19.5240	42.97%	4.0776	1.94%	21.4542	30.11%
Aerospike	27.5300	19.5240	41.01%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	25.3402	19.5240	29.79%	4.1422	3.55%	23.1983	9.23%
IE Aerospike	23.0809	19.5240	18.22%	4.0776	1.94%	21.4542	7.58%
Aerospike	22.9987	19.5240	17.80%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	21.7372	19.5240	11.34%	4.0457	1.14%	20.6397	5.32%
IE Aerospike	20.8470	19.5240	6.78%	4.0131	0.33%	19.8378	5.09%
Aerospike	20.8159	19.5240	6.62%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	21.2428	19.5240	8.80%	4.0347	0.87%	20.3657	4.31%
IE Aerospike	20.1686	19.5240	3.30%	4.0131	0.33%	19.8378	1.67%
Aerospike	20.1574	19.5240	3.24%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	20.2580	19.5240	3.76%	4.0020	0.05%	19.5716	3.51%
IE Aerospike	19.8429	19.5240	1.63%	4.0051	0.13%	19.6456	1.00%
Aerospike	19.8374	19.5240	1.61%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	20.3031	19.5240	3.99%	4.0042	0.10%	19.6241	3.46%
IE Aerospike	19.6508	19.5240	0.65%	4.0003	0.01%	19.5312	0.61%
Aerospike	19.6487	19.5240	0.64%	4.0000	0.0%	-	-

Table C.33	Code Accuracy Check for $\gamma = 1.26$ Ma $= 4.0$ $\beta = 0.2 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	20.6901	19.5240	5.97%	4.0025	0.06%	19.5835	5.65%
IE Aerospike	27.8285	19.5240	42.53%	4.0776	1.94%	21.4542	29.71%
Aerospike	27.5300	19.5240	41.01%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	**Failed**	19.5240	***	***	***	***	***
IE Aerospike	23.0455	19.5240	18.04%	4.0776	1.94%	21.4542	7.42%
Aerospike	25.1754	19.5240	28.95%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	21.6728	19.5240	11.01%	4.0462	1.15%	20.6523	4.94%
IE Aerospike	20.8341	19.5240	6.71%	4.0131	0.33%	19.8378	5.02%
Aerospike	20.8159	19.5240	6.62%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	21.3459	19.5240	9.33%	4.0387	0.97%	20.4650	4.30%
IE Aerospike	20.1623	19.5240	3.27%	4.0131	0.33%	19.8378	1.64%
Aerospike	20.1574	19.5240	3.24%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	20.4243	19.5240	4.61%	4.0077	0.19%	19.7079	3.64%
IE Aerospike	19.8398	19.5240	1.62%	4.0051	0.13%	19.6456	0.99%
Aerospike	19.8374	19.5240	1.61%	4.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	20.1462	19.5240	3.19%	4.0017	0.04%	19.5645	2.97%
IE Aerospike	19.6496	19.5240	0.64%	4.0003	0.01%	19.5312	0.61%
Aerospike	19.6487	19.5240	0.64%	4.0000	0.0%	-	-

Table C.34	Code Accuracy Check for $\gamma = 1.26$ Ma = 5.0 $\beta = 1.0 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	132.9868	63.3021	110.08%	5.3962	7.92%	98.2382	35.37%
IE Aerospike	109.8516	63.3021	73.54%	5.0189	0.38%	64.6667	69.87%
Aerospike	107.5203	63.3021	69.85%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	**Failed**	63.3021	***	***	***	***	***
IE Aerospike	82.0034	63.3021	29.54%	5.0189	0.38%	64.6667	26.81%
Aerospike	81.2856	63.3021	28.41%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	81.7780	63.3021	29.19%	5.1333	2.67%	73.5209	11.23%
IE Aerospike	69.9626	63.3021	10.52%	5.0189	0.38%	64.6667	8.19%
Aerospike	69.7675	63.3021	10.21%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	73.9363	63.3021	16.80%	5.0554	1.11%	67.3789	9.73%
IE Aerospike	66.5068	63.3021	5.06%	5.0189	0.38%	64.6667	2.85%
Aerospike	66.4338	63.3021	4.95%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	70.2601	63.3021	10.99%	5.0196	0.39%	64.7177	8.56%
IE Aerospike	64.8777	63.3021	2.49%	5.0080	0.16%	63.8764	1.57%
Aerospike	64.8423	63.3021	2.43%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	69.0814	63.3021	9.13%	5.0092	0.18%	63.9630	8.00%
IE Aerospike	63.9263	63.3021	0.99%	5.0015	0.03%	63.4094	0.82%
Aerospike	63.9127	63.3021	0.96%	5.0000	0.0%	-	-

Table C.35	Code Accuracy Check for $\gamma = 1.26$ Ma = 5.0 $\beta = 0.5 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	**Failed**	63.3021	***	***	***	***	***
IE Aerospike	109.2996	63.3021	72.66%	5.0189	0.38%	64.6667	69.02%
Aerospike	107.5203	63.3021	69.85%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	**Failed**	63.3021	***	***	***	***	***
IE Aerospike	81.7942	63.3021	29.21%	5.0189	0.38%	64.6667	26.49%
Aerospike	81.2856	63.3021	28.41%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	87.7593	63.3021	38.64%	5.1623	3.25%	75.9357	15.57%
IE Aerospike	69.8906	63.3021	10.41%	5.0189	0.38%	64.6667	8.08%
Aerospike	69.7675	63.3021	10.21%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	71.1602	63.3021	12.41%	5.0016	0.03%	63.4166	12.21%
IE Aerospike	66.4725	63.3021	5.01%	5.0189	0.38%	64.6667	2.79%
Aerospike	66.4338	63.3021	4.95%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	69.9702	63.3021	10.53%	5.0086	0.17%	63.9197	9.47%
IE Aerospike	64.8610	63.3021	2.46%	5.0080	0.16%	63.8764	1.54%
Aerospike	64.8423	63.3021	2.43%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	68.3921	63.3021	8.04%	5.0093	0.19%	63.9702	6.91%
IE Aerospike	63.9197	63.3021	0.98%	5.0015	0.03%	63.4094	0.80%
Aerospike	63.9127	63.3021	0.96%	5.0000	0.0%	-	-

Table C.36	Code Accuracy Check for $\gamma = 1.26$ Ma = 5.0 $\beta = 0.2 \cdot r_{throat}$ $r_{throat} = 1.0$ (Dimensionless)						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
$\Delta\upsilon = 0.05$							
Annular	103.6278	63.3021	63.70%	5.2071	4.14%	79.8096	29.84%
IE Aerospike	108.9684	63.3021	72.14%	5.0189	0.38%	64.6667	68.51%
Aerospike	107.5203	63.3021	69.85%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.025$							
Annular	76.1317	63.3021	20.27%	5.1004	2.01%	70.8667	7.43%
IE Aerospike	81.6687	63.3021	29.01%	5.0189	0.38%	64.6667	26.29%
Aerospike	81.2856	63.3021	28.41%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.01$							
Annular	71.9428	63.3021	13.65%	5.0228	0.46%	64.9516	10.76%
IE Aerospike	69.8473	63.3021	10.34%	5.0189	0.38%	64.6667	8.01%
Aerospike	69.7675	63.3021	10.21%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.005$							
Annular	74.7474	63.3021	18.08%	5.0503	1.01%	66.9938	11.57%
IE Aerospike	66.4519	63.3021	4.98%	5.0189	0.38%	64.6667	2.76%
Aerospike	66.4338	63.3021	4.95%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.0025$							
Annular	66.4476	63.3021	4.97%	5.0192	0.38%	64.6885	2.72%
IE Aerospike	64.8509	63.3021	2.45%	5.0080	0.16%	63.8764	1.53%
Aerospike	64.8423	63.3021	2.43%	5.0000	0.0%	-	-
$\Delta\upsilon = 0.001$							
Annular	70.0094	63.3021	10.60%	5.0148	0.30%	64.3684	8.76%
IE Aerospike	63.9157	63.3021	0.97%	5.0015	0.03%	63.4094	0.80%
Aerospike	63.9127	63.3021	0.96%	5.0000	0.0%	-	-

Table C.37	Experimental Nozzles Code Error Analysis $\gamma = 1.2983 \quad \Delta\upsilon = 0.001 \quad r_{throat} = 0.15$						
	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Comp}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{\%Error}$	Ma_{Comp}	$Ma_{\%Error}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},Theory}$	$\left(\frac{A_{exit}}{A_{throat}}\right)_{Ma_{Comp},\%Error}$
Ma = 3.0 $\beta_{Annular} = 0.2 \cdot r_{throat}$ $\beta_{IEAerospike} = 0.1 \cdot r_{throat}$							
Annular	5.3094	5.1797	2.50%	3.0062	0.21%	5.2164	1.78%
IE Aerospike	5.2015	5.1797	0.42%	3.0003	0.01%	5.1815	0.39%
Aerospike	5.1977	5.1797	0.35%	3.0000	0.0%	-	-
Ma = 3.0 $\beta_{Annular} = 0.5 \cdot r_{throat}$ $\beta_{IEAerospike} = 0.1 \cdot r_{throat}$							
Annular	5.3076	5.1797	2.47%	3.0072	0.24%	5.2224	1.63%
IE Aerospike	5.2015	5.1797	0.42%	3.0003	0.01%	5.1815	0.39%
Aerospike	5.1977	5.1797	0.35%	3.0000	0.0%	-	-
Ma = 3.0 $\beta_{Annular} = 1.0 \cdot r_{throat}$ $\beta_{IEAerospike} = 0.1 \cdot r_{throat}$							
Annular	5.2840	5.1797	2.01%	3.0030	0.10%	5.1975	1.66%
IE Aerospike	5.2015	5.1797	0.42%	3.0003	0.01%	5.1815	0.39%
Aerospike	5.1977	5.1797	0.35%	3.0000	0.0%	-	-
Ma = 4.0 $\beta_{Annular} = 0.5 \cdot r_{throat}$ $\beta_{IEAerospike} = 0.1 \cdot r_{throat}$							
Annular	16.5458	16.0722	2.95%	4.0019	0.05%	16.1061	2.73%
IE Aerospike	16.1702	16.0722	0.61%	4.0014	0.04%	16.0971	0.45%
Aerospike	16.1699	16.0722	0.61%	4.0000	0.0%	-	-

Appendix D: CFD Models and Results

D.1 Air Nozzle Simulations

D.1.1 Gambit Inputs and Mesh Results

Annular Nozzles

Table D.1.1	**Ma = 3.0 $\beta = 0.2 \cdot r_{throat}$ Annular Air Nozzle Gambit Meshing Inputs and Results**		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	35700	

Table D.1.2	**Ma = 3.0 $\beta = 0.5 \cdot r_{throat}$ Annular Air Nozzle Gambit Meshing Inputs and Results**		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	35100	

Table D.1.3	**Ma = 3.0 $\beta = 1.0 \cdot r_{throat}$ Annular Air Nozzle Gambit Meshing Inputs and Results**		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	36300	

Table D.1.4	Ma = 4.0 $\beta = 0.2 \cdot r_{throat}$ Annular Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	48400	

Table D.1.5	Ma = 4.0 $\beta = 0.5 \cdot r_{throat}$ Annular Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	48700	

Table D.1.6	Ma = 4.0 $\beta = 1.0 \cdot r_{throat}$ Annular Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	49600	

Aerospike Nozzles

Table D.1.7	Ma = 3.0 100% Length External Aerospike Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.42
		Number of Cells > 0.97	0
	Total Number of Cells	19453	

Table D.1.8	Ma = 3.0 20% Length External Aerospike Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.42
		Number of Cells > 0.97	0
	Total Number of Cells	4715	

Table D.1.9	Ma = 4.0 100% Length External Aerospike Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.42
		Number of Cells > 0.97	0
	Total Number of Cells	22188	

Table D.1.10	Ma = 4.0 20% Length External Aerospike Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.47
		Number of Cells > 0.97	0
	Total Number of Cells	22447	

Table D.1.11	Ma = 3.0 100% Length Internal-External Aerospike Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.41
		Number of Cells > 0.97	0
	Total Number of Cells	6202	

Table D.1.12	Ma = 3.0 20% Length Internal-External Aerospike Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.40
		Number of Cells > 0.97	0
	Total Number of Cells	6142	

Table D.1.13	Ma = 4.0 100% Length Internal-External Aerospike Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.1
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.58
		Number of Cells > 0.97	0
	Total Number of Cells	6493	

Table D.1.14	Ma = 4.0 20% Length Internal-External Aerospike Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.1
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.43
		Number of Cells > 0.97	0
	Total Number of Cells	6615	

Conical Nozzles

Table D.1.15	Ma = 3.0 $\theta = 8°$ Conical Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	33700	

Table D.1.16	Ma = 3.0 $\theta = 10°$ Conical Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	31400	

Table D.1.17	Ma = 3.0 $\theta = 12°$ Conical Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	29900	

Table D.1.18	Ma = 4.0 $\theta = 8°$ Conical Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	46800	

Table D.1.19	Ma = 4.0 $\theta = 10°$ Conical Air Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	41900	

Table D.1.20	**Ma = 4.0 $\theta = 12°$ Conical Air Nozzle Gambit Meshing Inputs and Results**		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	38700	

D.1.2 FLUENT Inputs

Table D.1.21	FLUENT Input Conditions Used for All Air Nozzles Designed for an Exit Mach Number = 3.0 Simulations		
Solver:	Solver:	Coupled	
	Space:	Axisymmetric	
	Velocity Formation:	Absolute	
	Gradient Option:	Cell-based	
	Formulation:	Implicit	
	Time:	Steady	
	Porous Formulation:	Superficial Velocity	
	Energy Equation:	Checked	
	Viscous Model:	Inviscid Checked	
Material:	Name:	Air	
	Chemical Formulation:	N/A	
	Material Type:	Fluid	
	FLUENT Fluid Material:	Air	
	Properties:	Density:	Ideal Gas
		Cp:	1006.43 J/kg*K
		Molecular Weight:	28.966 kg/kmol
Operating Conditions:	Pressure:	Operating Pressure:	0 Pa
	Gravity:	Not Checked	
	Reference Pressure Location:	X(m):	0
		Y(m):	0
Pressure Inlet:	Gauge Total Pressure:	3721943 Pa	Constant
	Supersonic/Initial Gauge Pressure:	3721943 Pa	Constant
	Total Temperature:	840.0 K	Constant
	Direction Specification Method:	Normal to Boundary	
Pressure Outlet:	Gauge Pressure:	101325 Pa	Constant
	Backflow Total Temperature:	300 K	Constant
	Backflow Direction Specification Method:	Normal to Boundary	
	Non-reflecting Boundary:	Not Checked	
	Target Mass-flow Rate:	Not Checked	
Solution Controls:	Discretization:	Second Order Upwind	
	Solver Parameter:	Courant Number:	1
Solution Initialization:	Compute From:	Pressure Inlet	
	Reference Frame:	Relative to Cell Zone	
	Initial Values:	Automatically Set by *Compute From*	

Table D.1.22	FLUENT Input Conditions Used for All Air Nozzles Designed for an Exit Mach Number = 4.0 Simulations		
Solver:	Solver:	Coupled	
	Space:	Axisymmetric	
	Velocity Formation:	Absolute	
	Gradient Option:	Cell-based	
	Formulation:	Implicit	
	Time:	Steady	
	Porous Formulation:	Superficial Velocity	
	Energy Equation:	Checked	
	Viscous Model:	Inviscid Checked	
Material:	Name:	Air	
	Chemical Formulation:	N/A	
	Material Type:	Fluid	
	FLUENT Fluid Material:	Air	
	Properties:	Density:	Ideal Gas
		Cp:	1006.43 J/kg*K
		Molecular Weight:	28.966 kg/kmol
Operating Conditions:	Pressure:	Operating Pressure:	0 Pa
	Gravity:	Not Checked	
	Reference Pressure Location:	X(m):	0
		Y(m):	0
Pressure Inlet:	Gauge Total Pressure:	1.53847×10^7 Pa	Constant
	Supersonic/Initial Gauge Pressure:	1.53847×10^7 Pa	Constant
	Total Temperature:	1260.0 K	Constant
	Direction Specification Method:	Normal to Boundary	
Pressure Outlet:	Gauge Pressure:	101325 Pa	Constant
	Backflow Total Temperature:	300 K	Constant
	Backflow Direction Specification Method:	Normal to Boundary	
	Non-reflecting Boundary:	Not Checked	
	Target Mass-flow Rate:	Not Checked	
Solution Controls:	Discretization:	Second Order Upwind	
	Solver Parameter:	Courant Number:	1
Solution Initialization:	Compute From:	Pressure Inlet	
	Reference Frame:	Relative to Cell Zone	
	Initial Values:	Automatically Set by *Compute From*	

D.1.3 Entropy Plots from FLUENT Simulations

Annular Nozzles

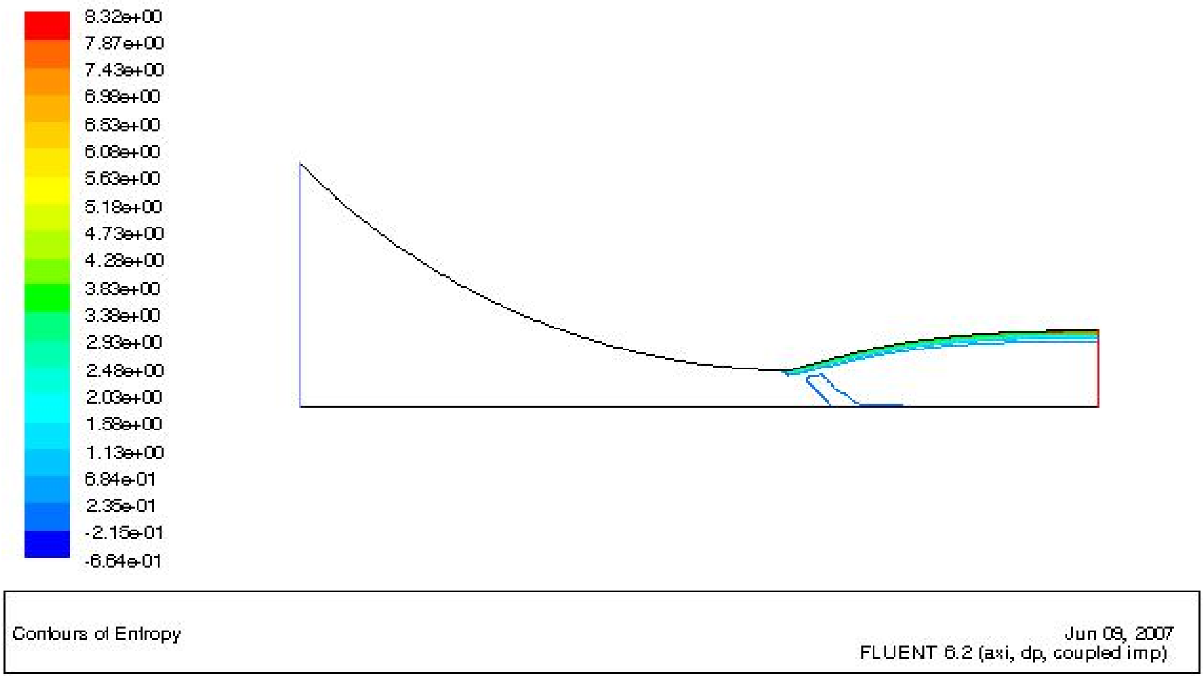


Figure D.1.1: *Entropy Plot for Mach 3.0* $\beta = 0.2 \cdot r_{throat}$ *Annular Air Nozzle*

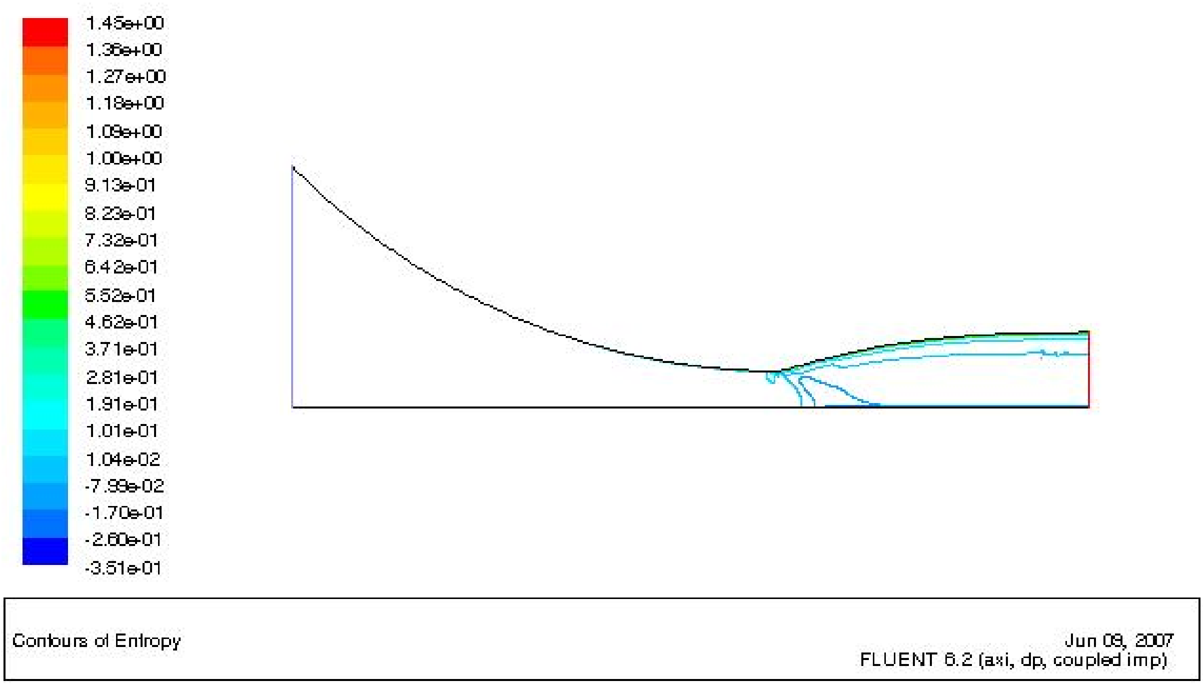


Figure D.1.2: *Entropy Plot for Mach 3.0* $\beta = 0.5 \cdot r_{throat}$ *Annular Air Nozzle*

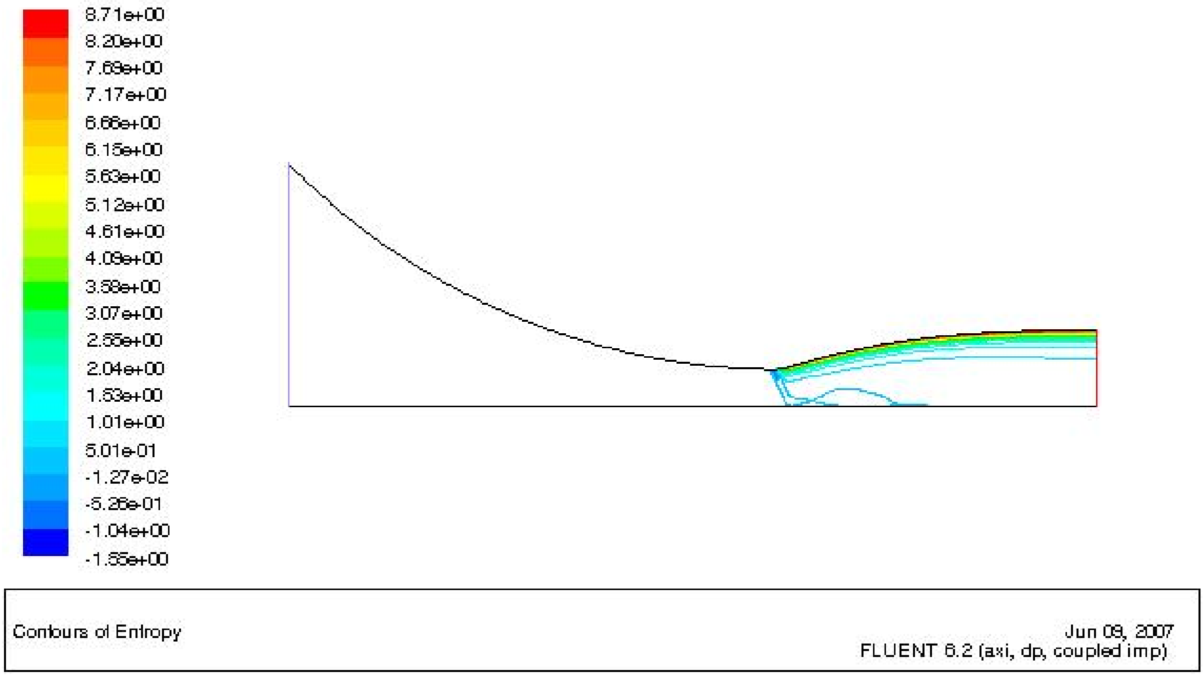


Figure D.1.3: *Entropy Plot for Mach 3.0* $\beta = 1.0 \cdot r_{throat}$ *Annular Air Nozzle*

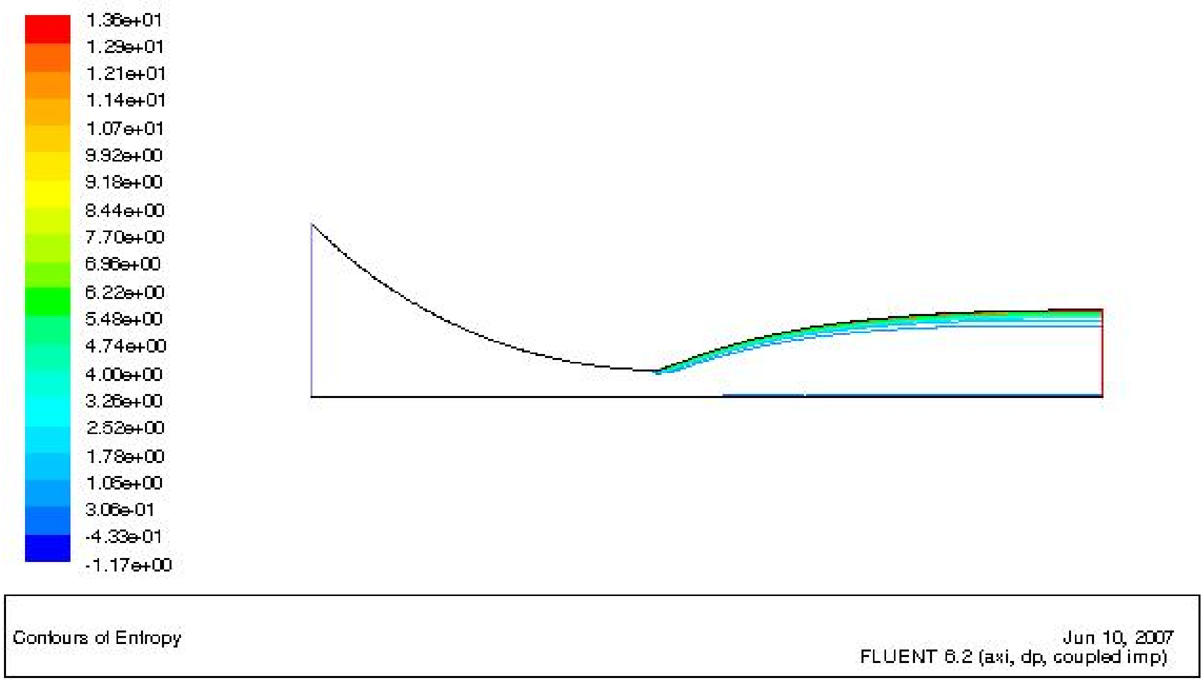


Figure D.1.4: *Entropy Plot for Mach 4.0* $\beta = 0.2 \cdot r_{throat}$ *Annular Air Nozzle*

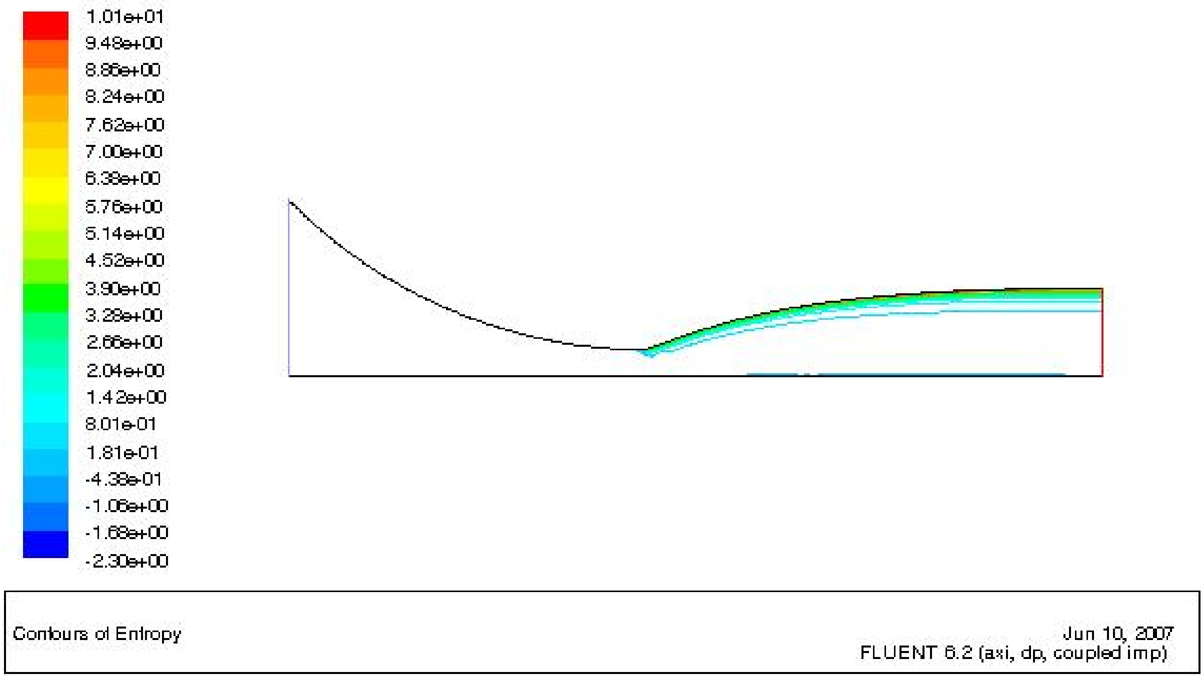


Figure D.1.5: *Entropy Plot for Mach 4.0* $\beta = 0.5 \cdot r_{throat}$ *Annular Air Nozzle*

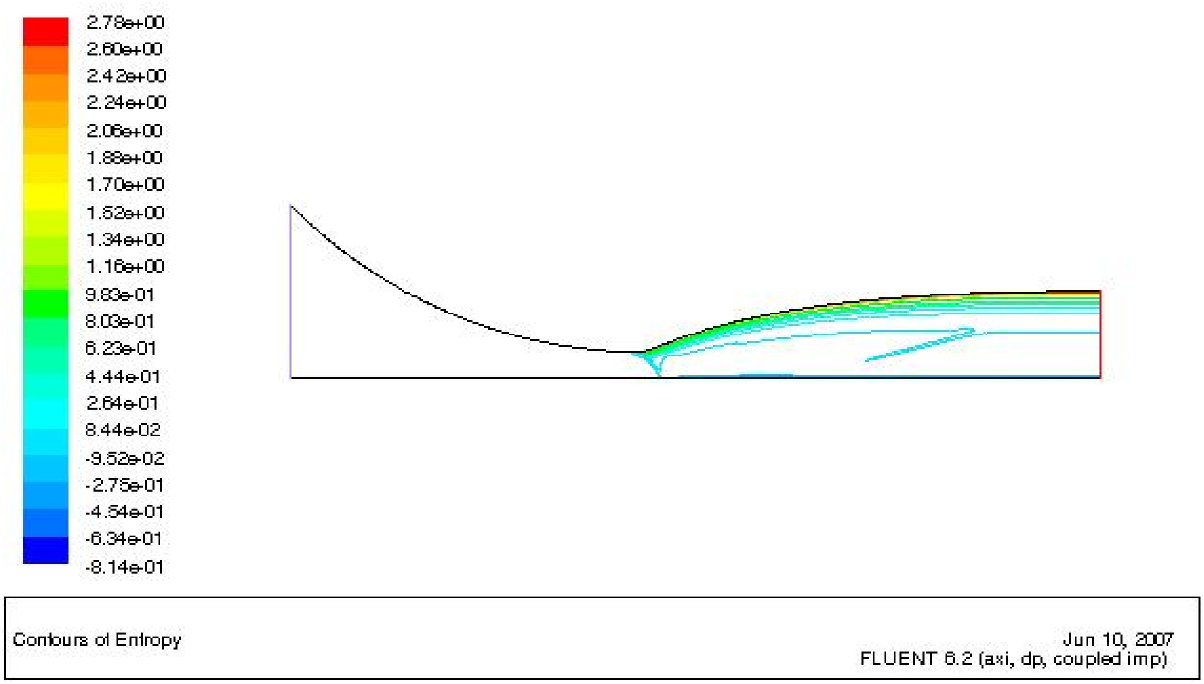


Figure D.1.6: *Entropy Plot for Mach 4.0* $\beta = 1.0 \cdot r_{throat}$ *Annular Air Nozzle*

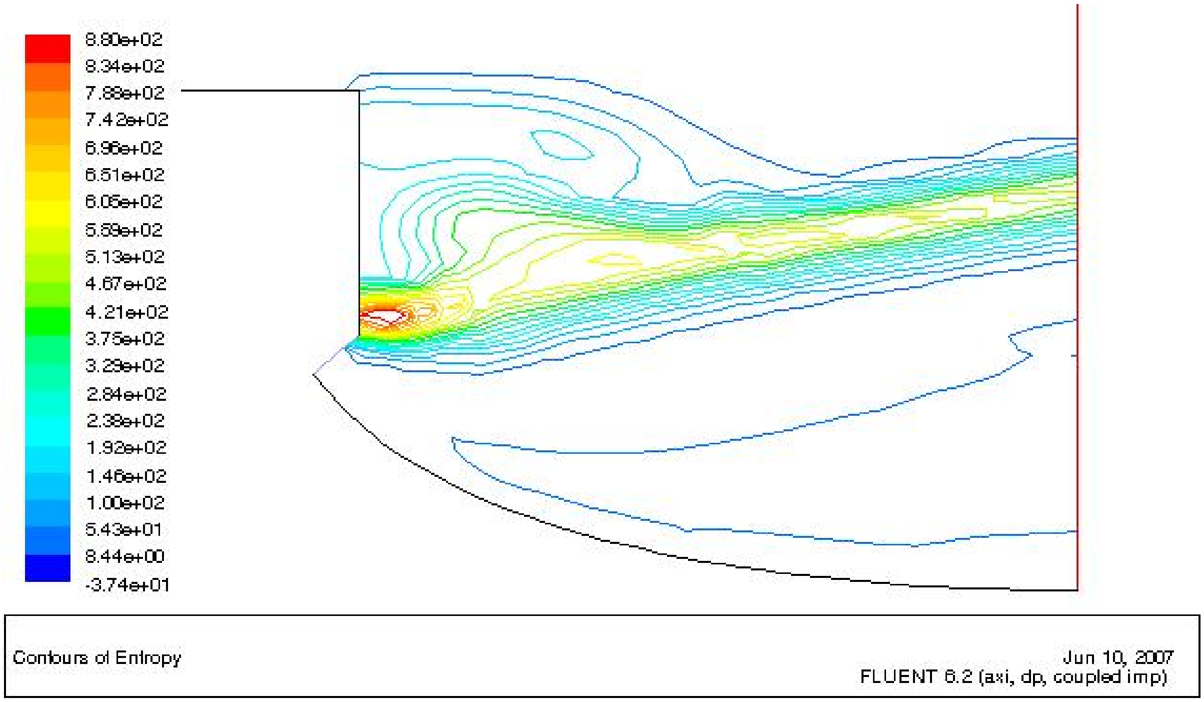


Figure D.1.7: *Entropy Plot for Mach 3.0 100% Length External Aerospike Air Nozzle*

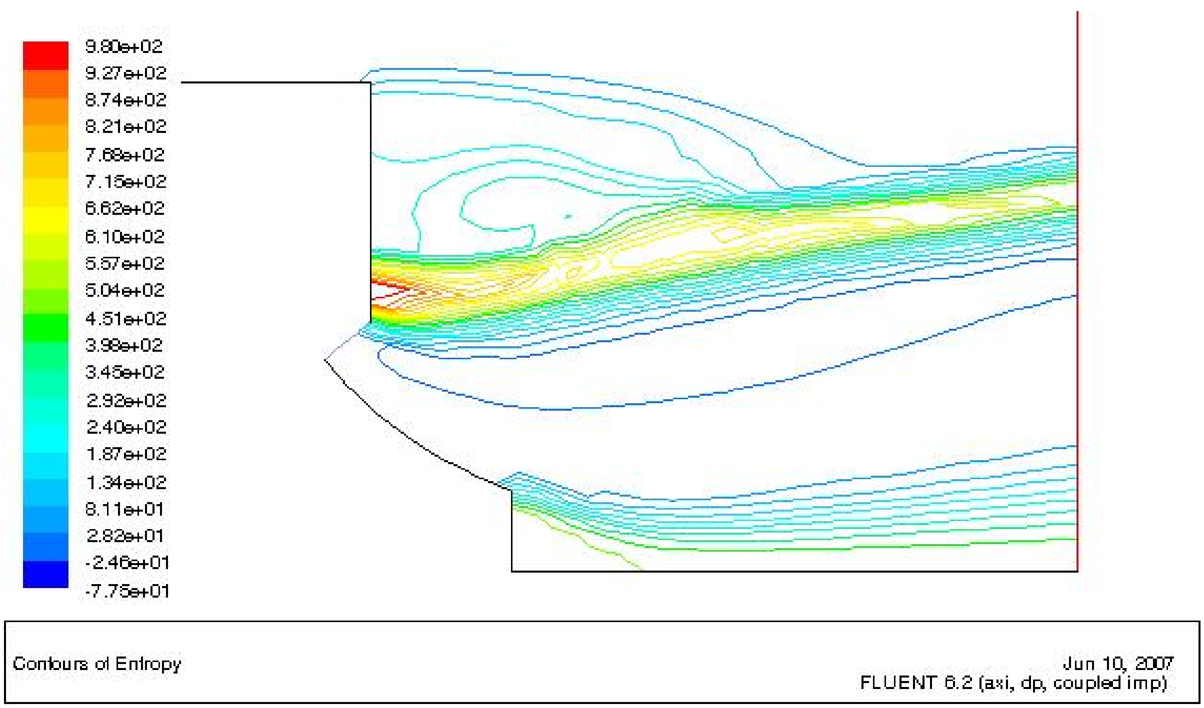


Figure D.1.8: *Entropy Plot for Mach 3.0 20% Length External Aerospike Air Nozzle*

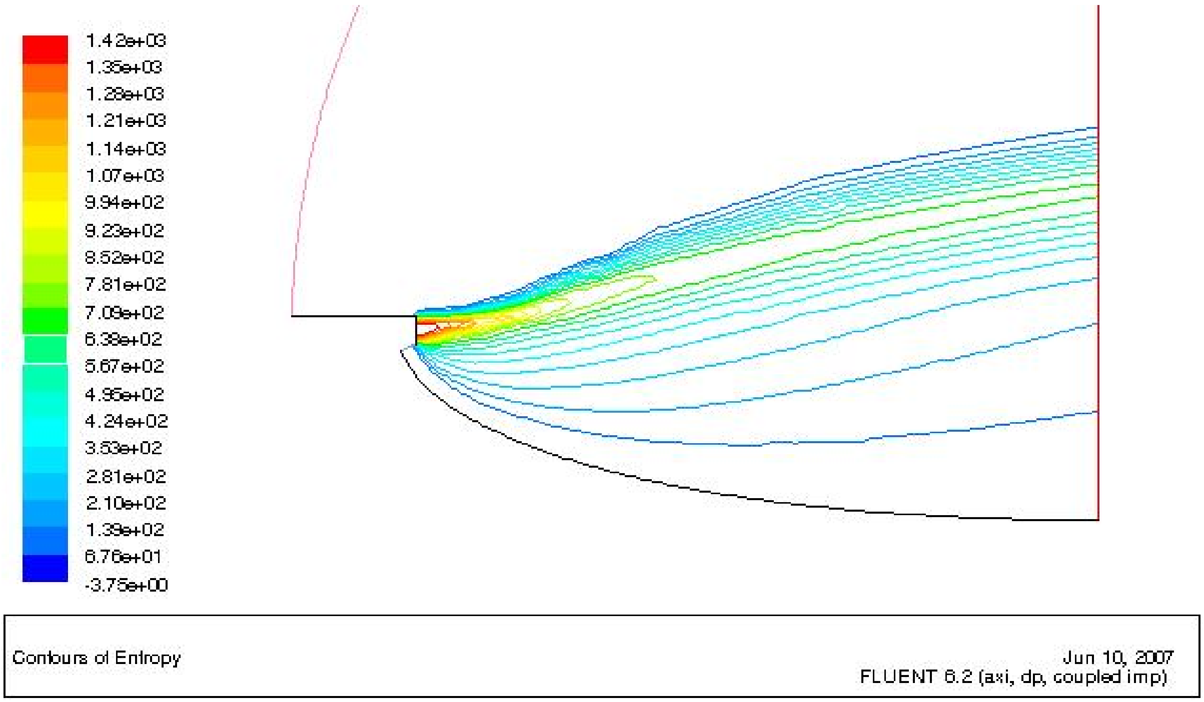


Figure D.1.9: *Entropy Plot for Mach 4.0 100% Length External Aerospike Air Nozzle*

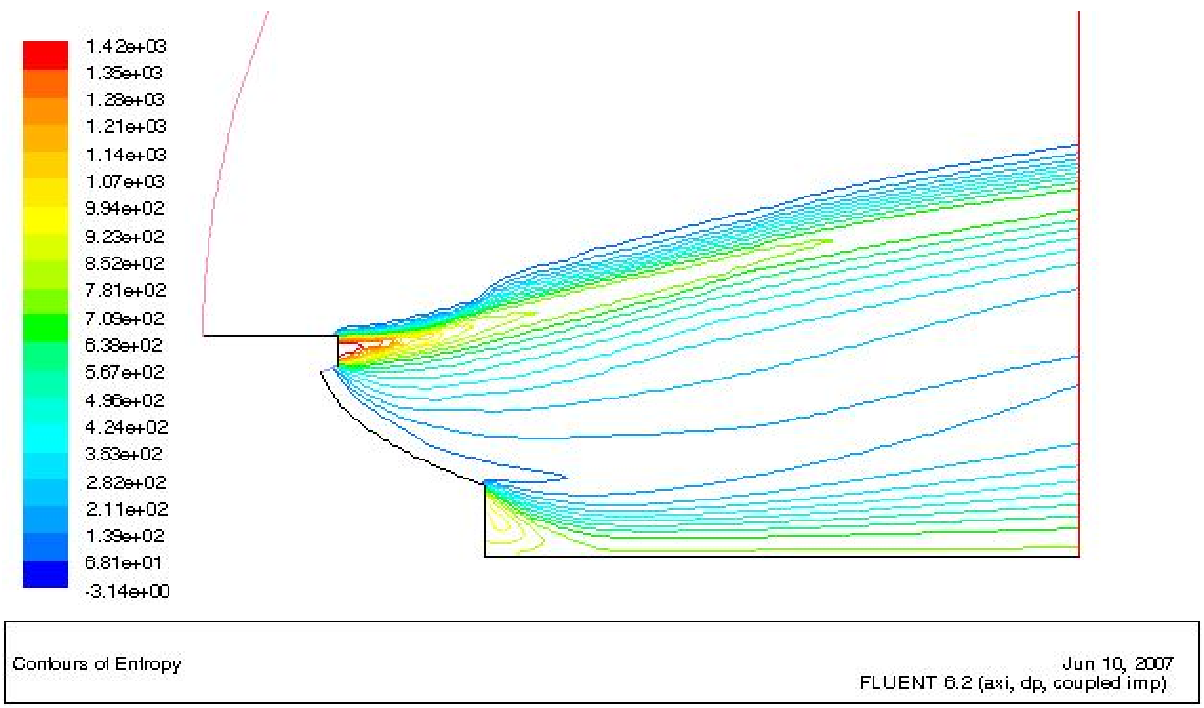


Figure D.1.10: *Entropy Plot for Mach 4.0 20% Length External Aerospike Air Nozzle*

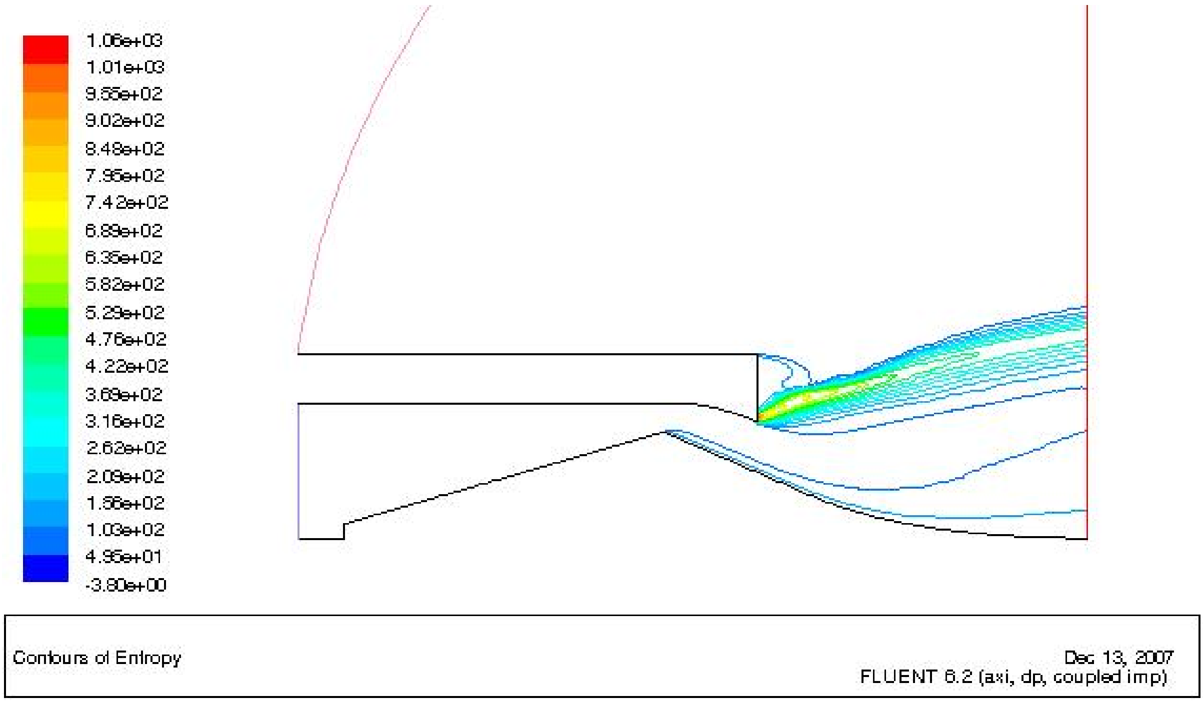


Figure D.1.11: *Entropy Plot for Mach 3.0 100% Length Internal-External Aerospike Air Nozzle*

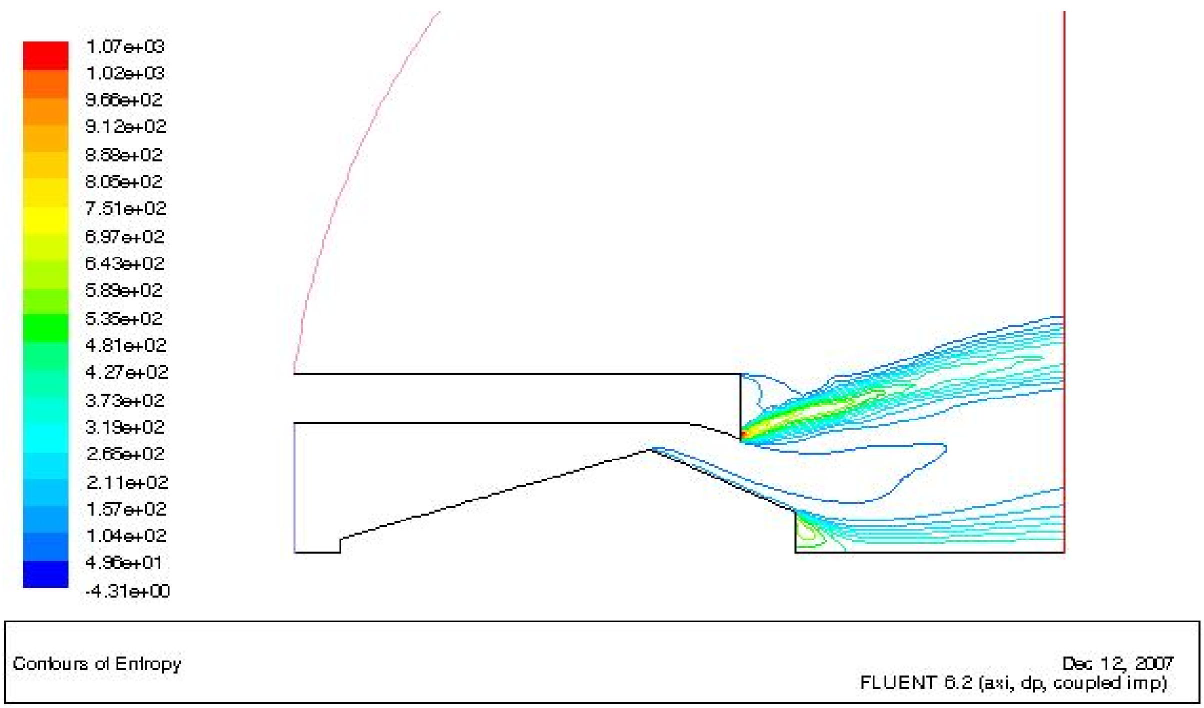


Figure D.1.12: *Entropy Plot for Mach 3.0 20% Length Internal-External Aerospike Air Nozzle*

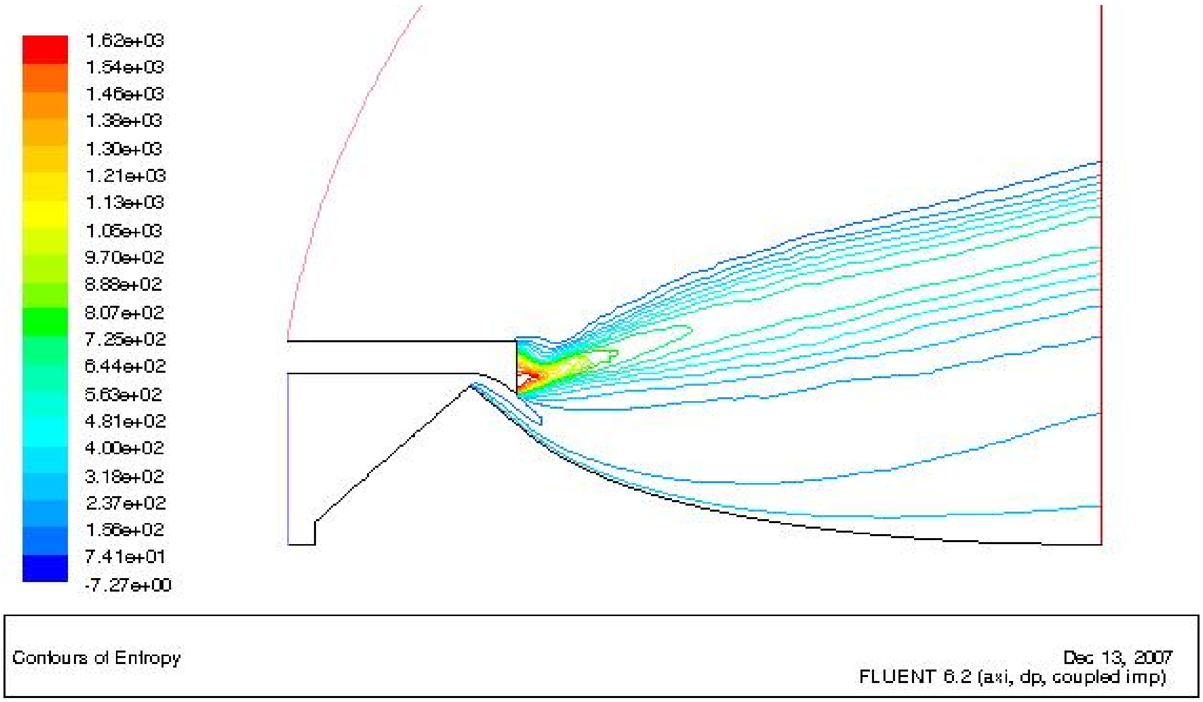


Figure D.1.13: *Entropy Plot for Mach 4.0 100% Length Internal-External Aerospike Air Nozzle*

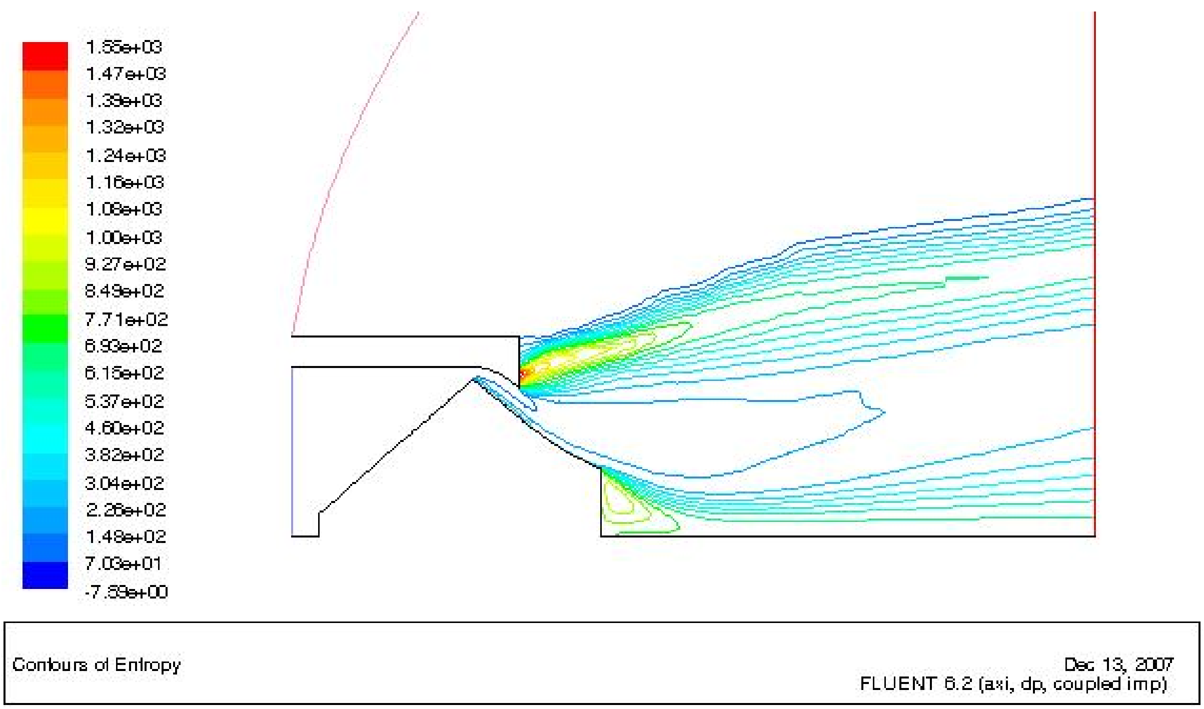


Figure D.1.14: *Entropy Plot for Mach 4.0 20% Length Internal-External Aerospike Air Nozzle*

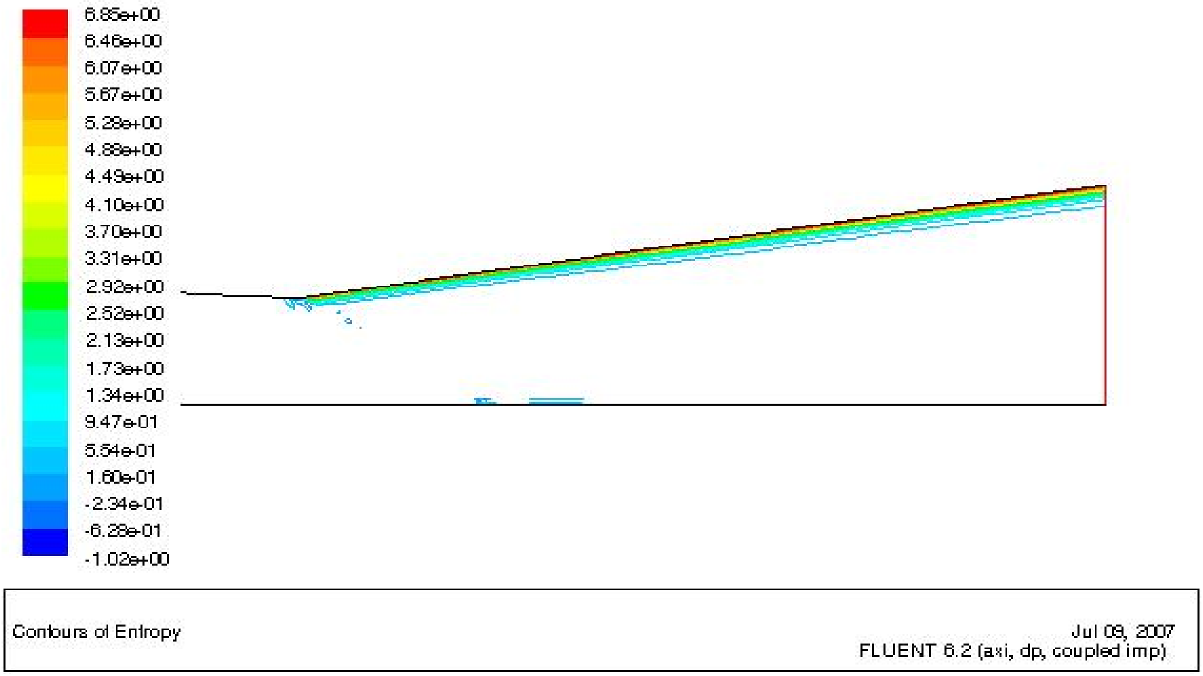


Figure D.1.15: *Entropy Plot for Mach 3.0, 8 Degree Conical Air Nozzle*

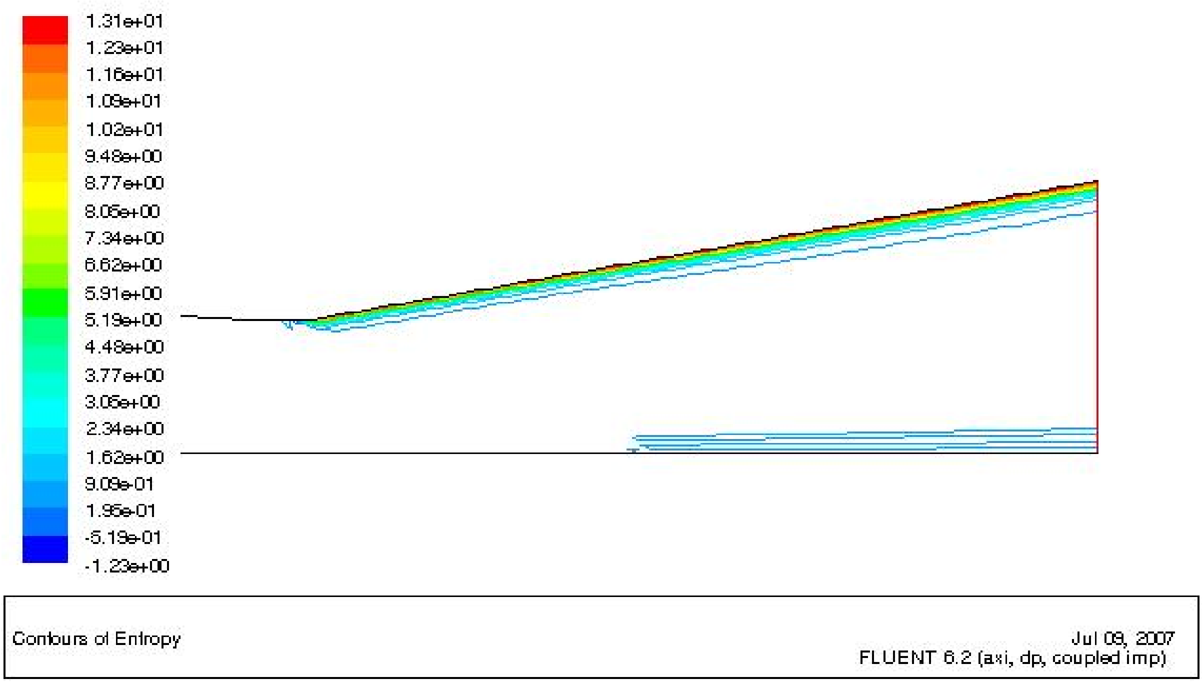


Figure D.1.16: *Entropy Plot for Mach 3.0, 10 Degree Conical Air Nozzle*

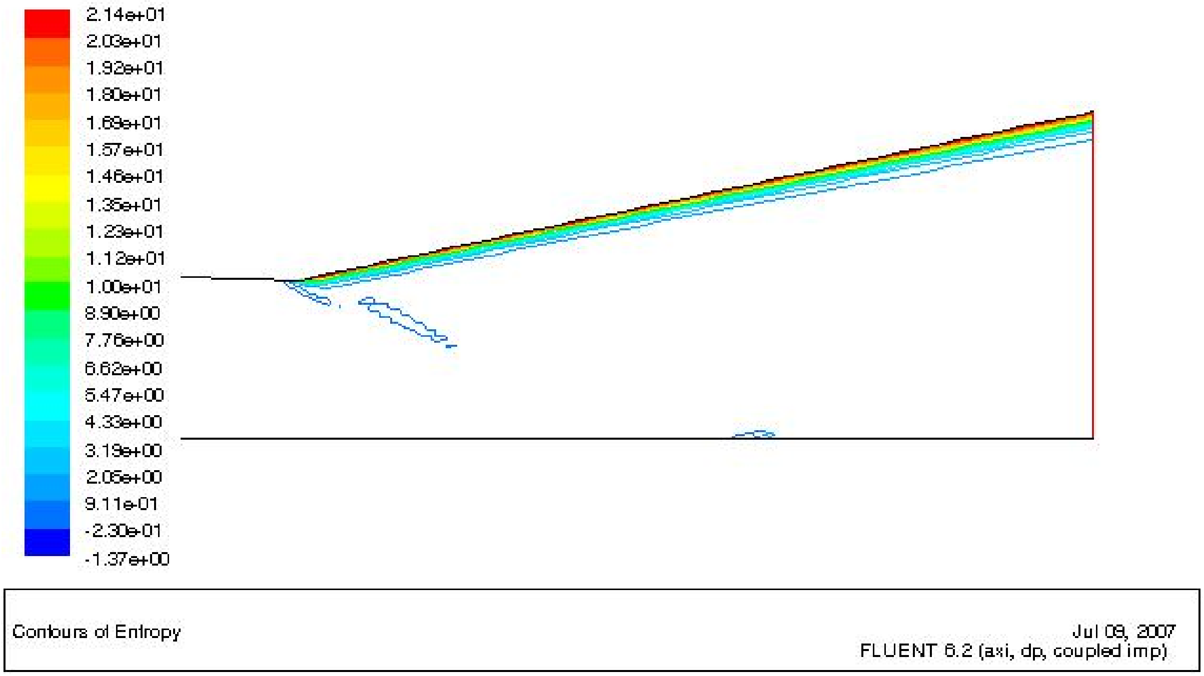


Figure D.1.17: *Entropy Plot for Mach 3.0, 12 Degree Conical Air Nozzle*

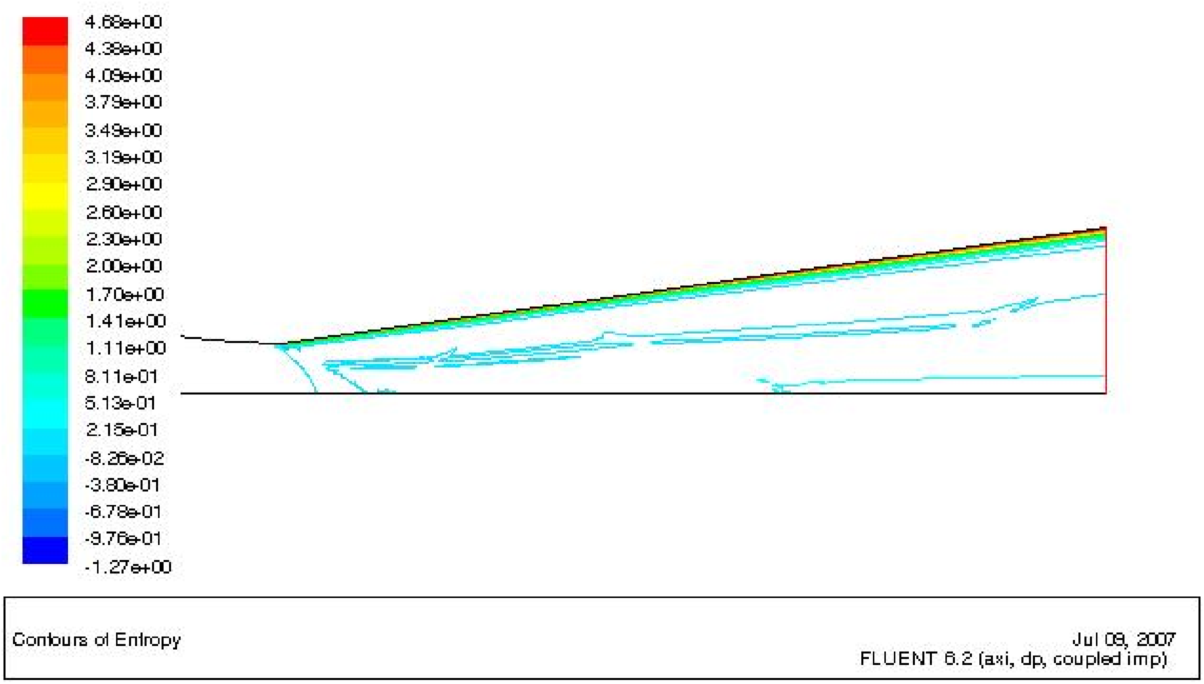


Figure D.1.18: *Entropy Plot for Mach 4.0, 8 Degree Conical Air Nozzle*

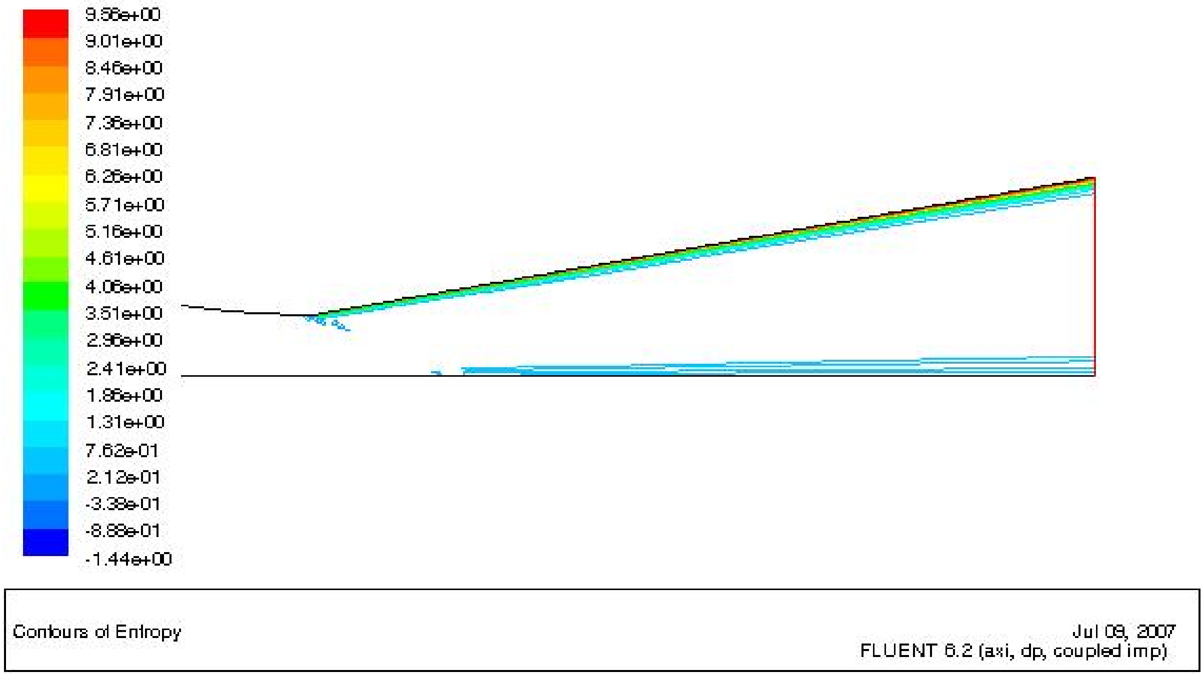


Figure D.1.19: *Entropy Plot for Mach 4.0, 10 Degree Conical Air Nozzle*

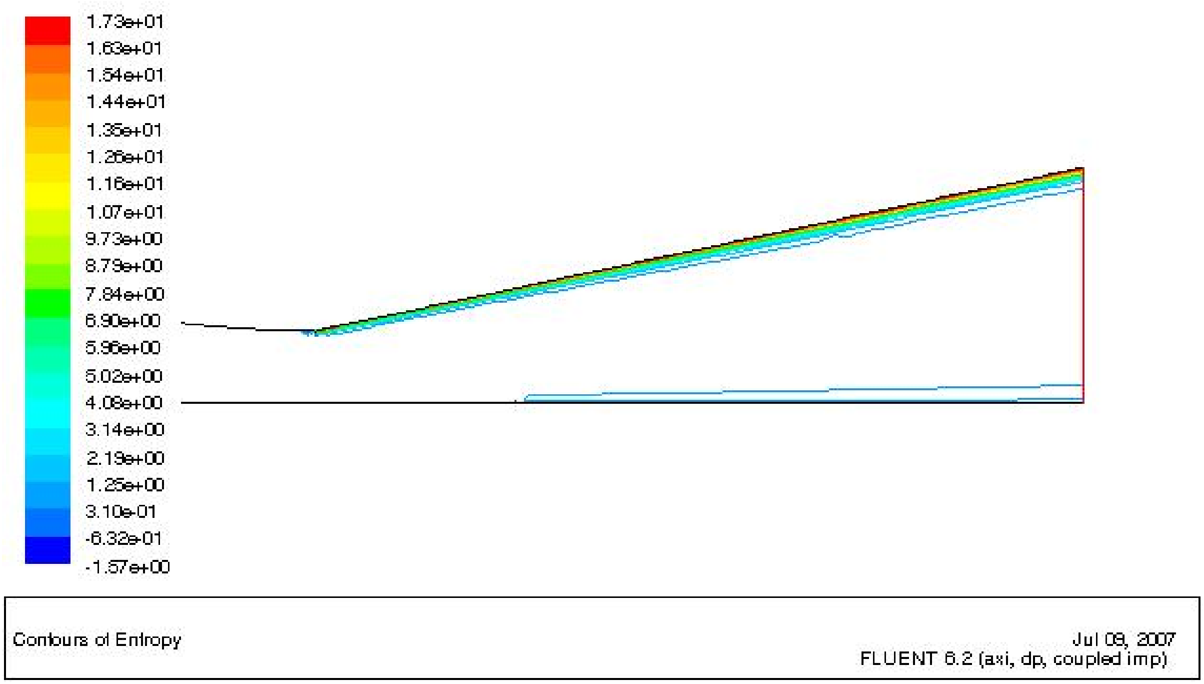


Figure D.1.20: *Entropy Plot for Mach 4.0, 12 Degree Conical Air Nozzle*

D.1.4 Mach Plot from FLUENT Simulations

Annular Nozzles

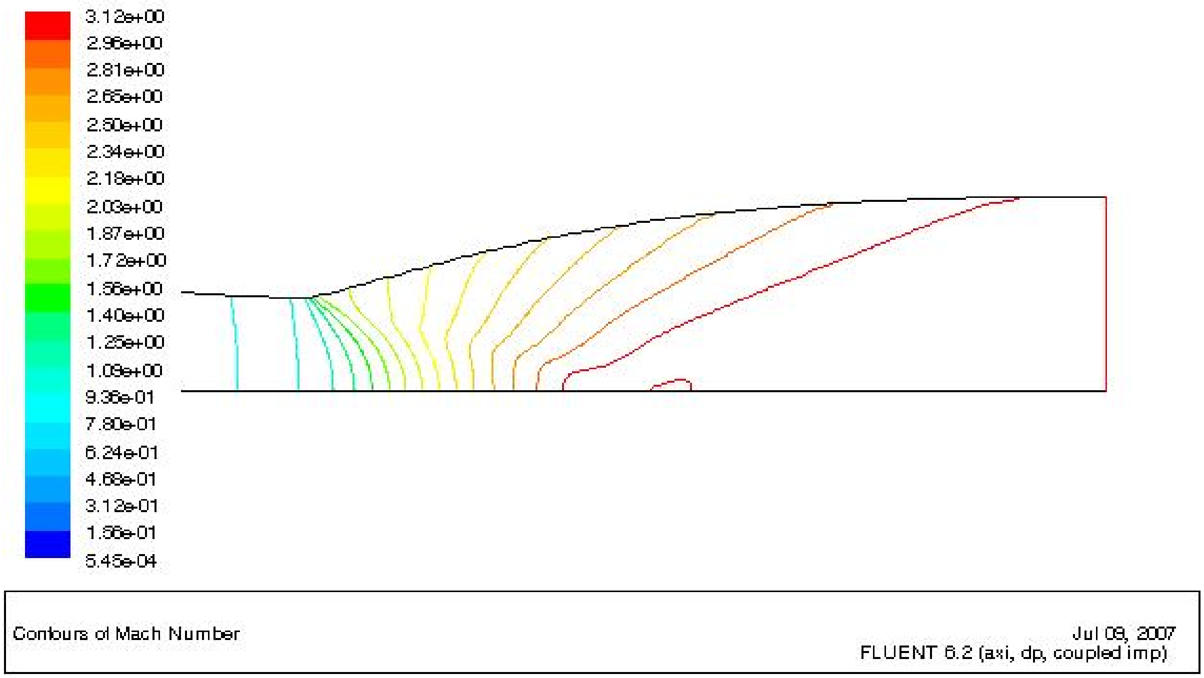


Figure D.1.21: *Mach Plot for Mach 3.0* $\beta = 0.2 \cdot r_{throat}$ *Annular Air Nozzle*

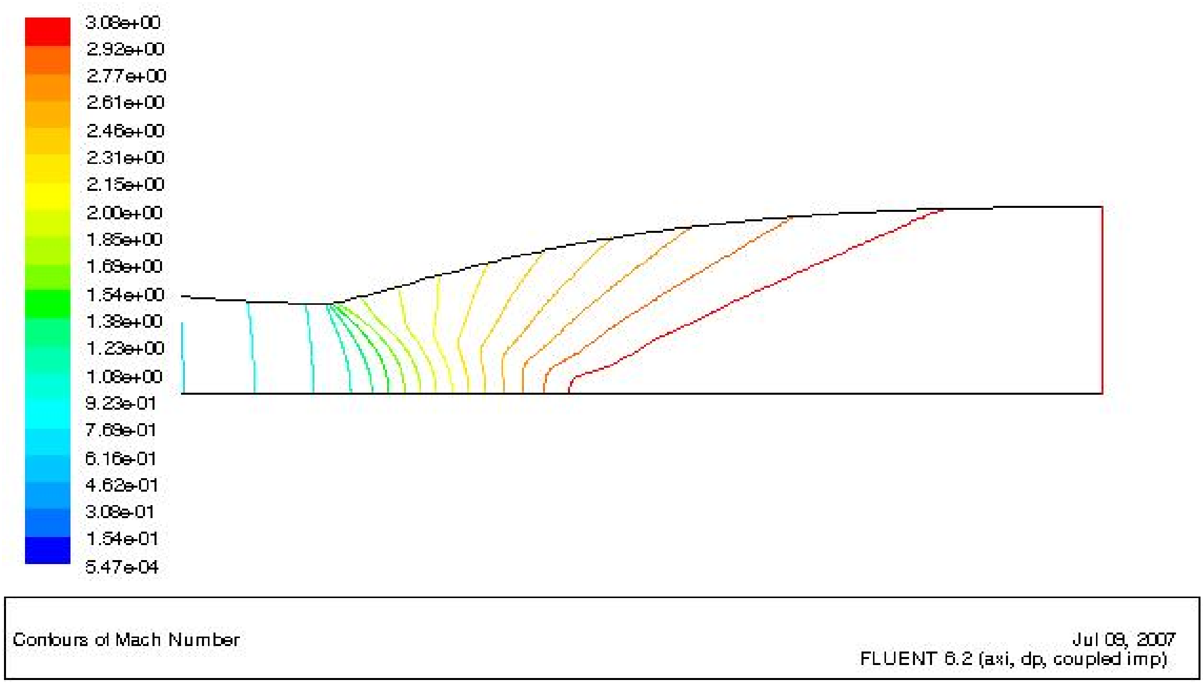


Figure D.1.22: *Mach Plot for Mach 3.0* $\beta = 0.5 \cdot r_{throat}$ *Annular Air Nozzle*

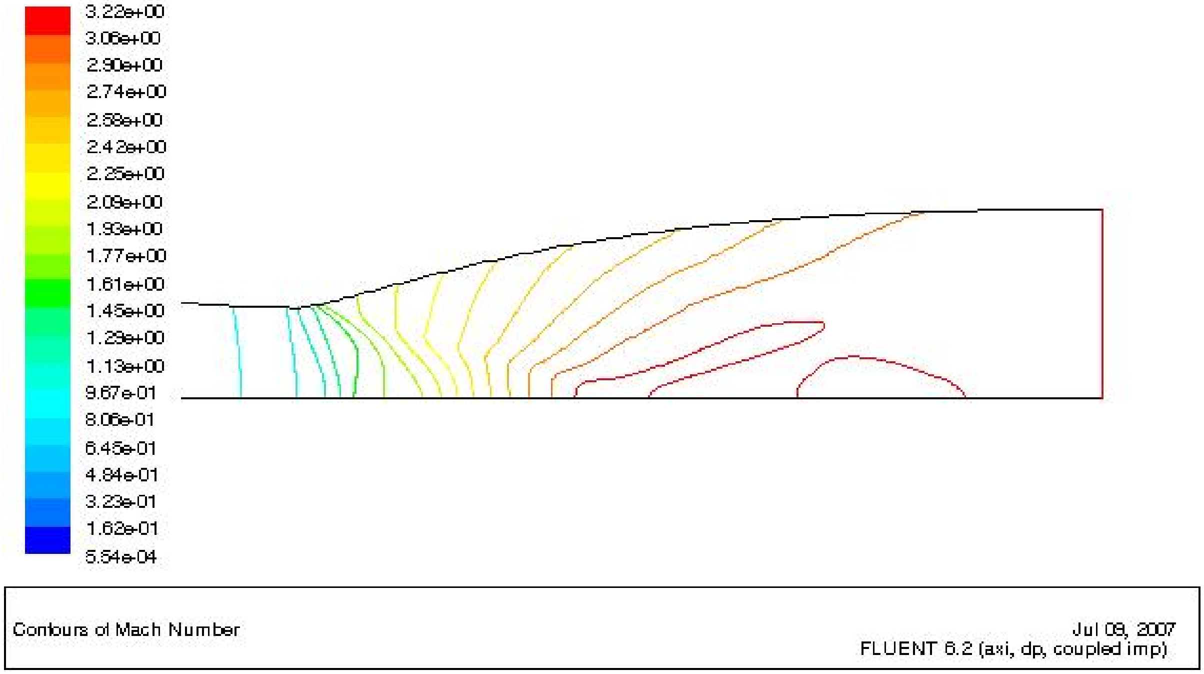


Figure D.1.23: *Mach Plot for Mach 3.0* $\beta = 1.0 \cdot r_{throat}$ *Annular Air Nozzle*

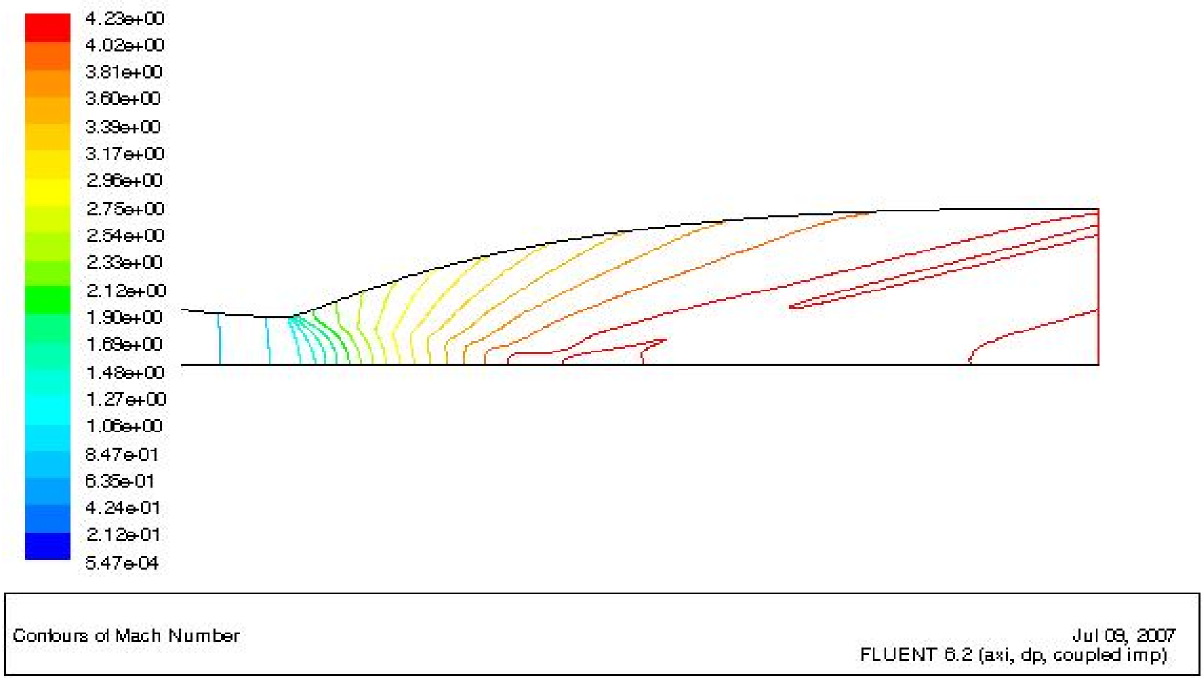


Figure D.1.24: *Mach Plot for Mach 4.0* $\beta = 0.2 \cdot r_{throat}$ *Annular Air Nozzle*

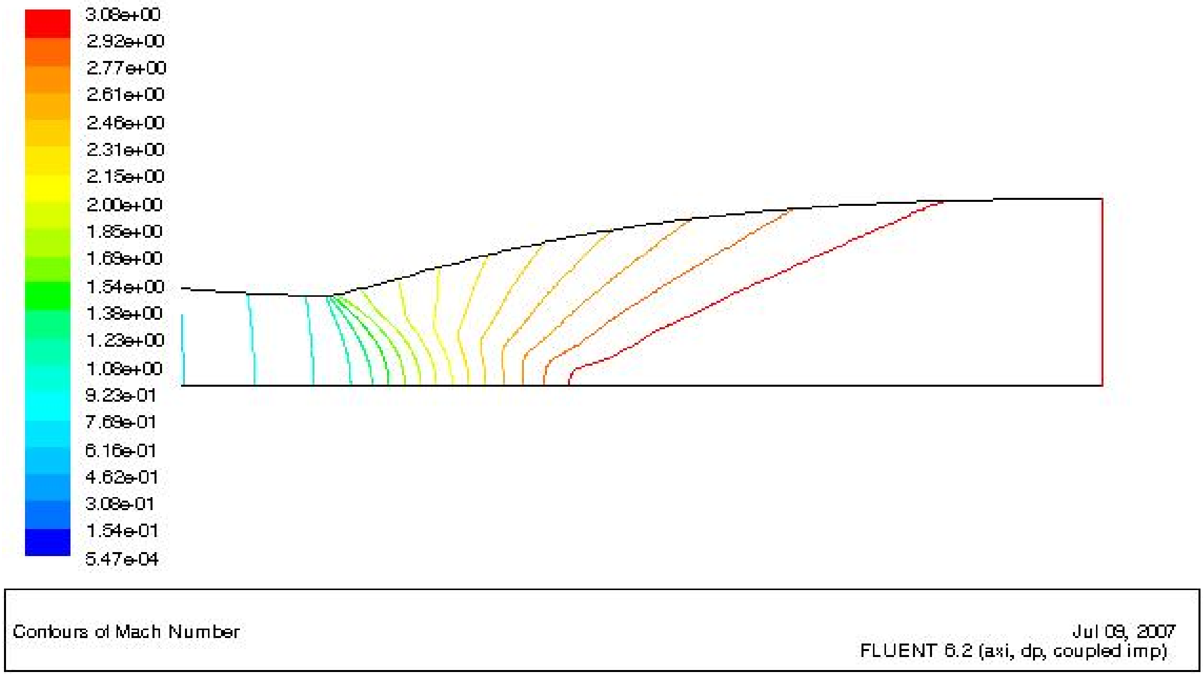


Figure D.1.25: *Mach Plot for Mach 4.0* $\beta = 0.5 \cdot r_{throat}$ *Annular Air Nozzle*

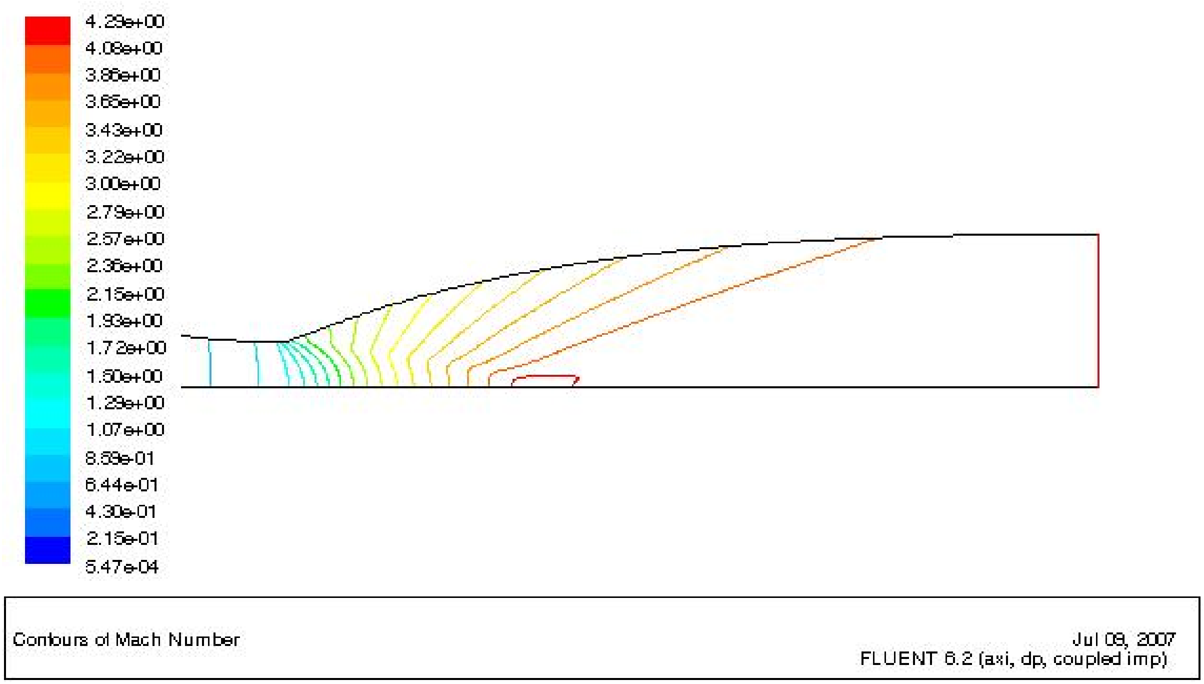


Figure D.1.26: *Mach Plot for Mach 4.0* $\beta = 1.0 \cdot r_{throat}$ *Annular Air Nozzle*

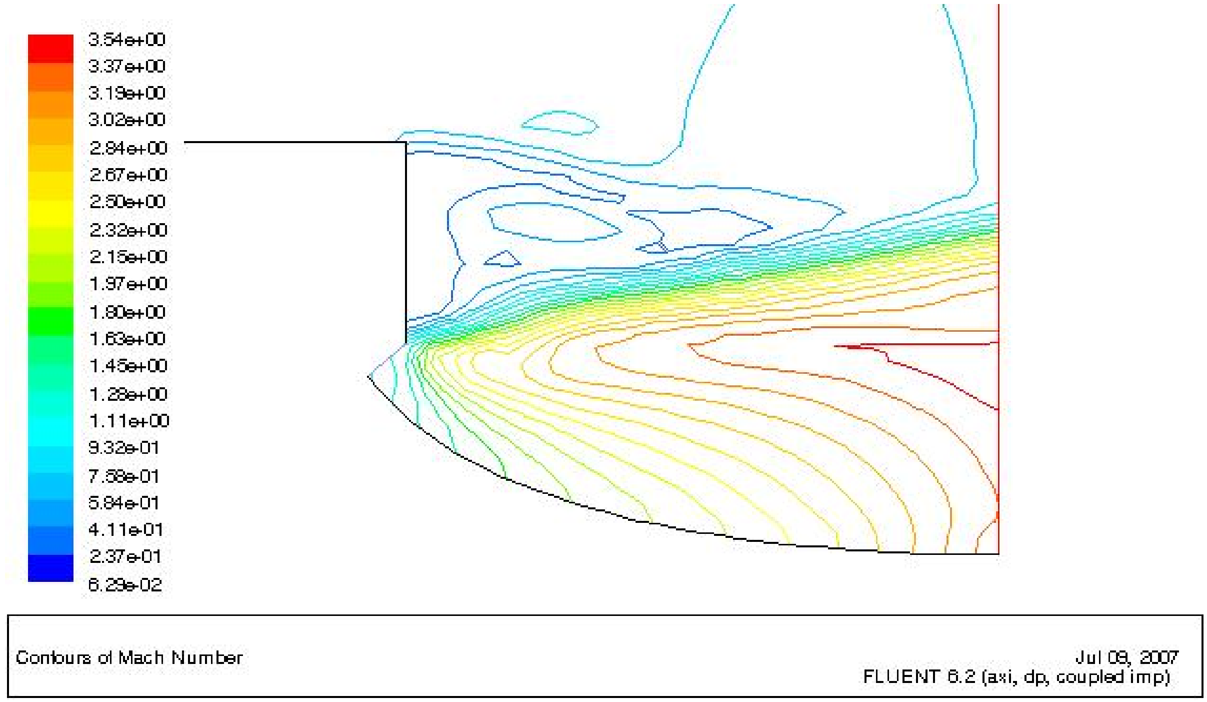


Figure D.1.27: *Mach Plot for Mach 3.0, 100% Length External Aerospike Air Nozzle*

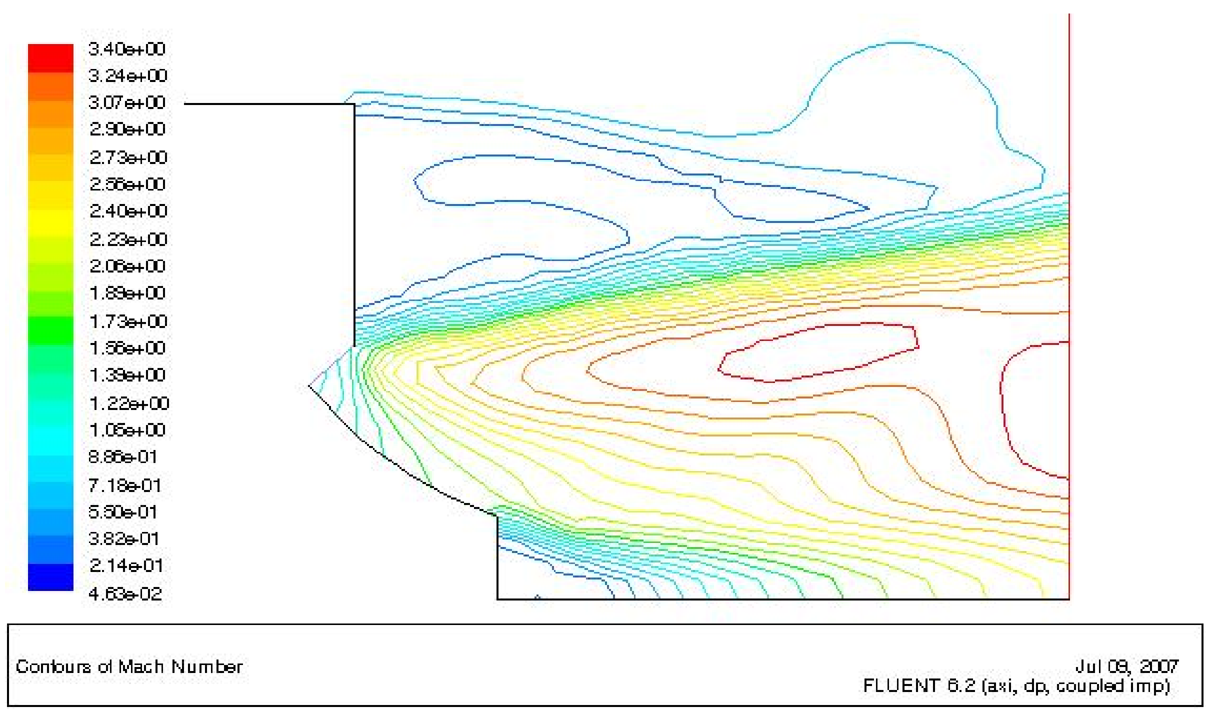


Figure D.1.28: *Mach Plot for Mach 3.0, 20% Length External Aerospike Air Nozzle*

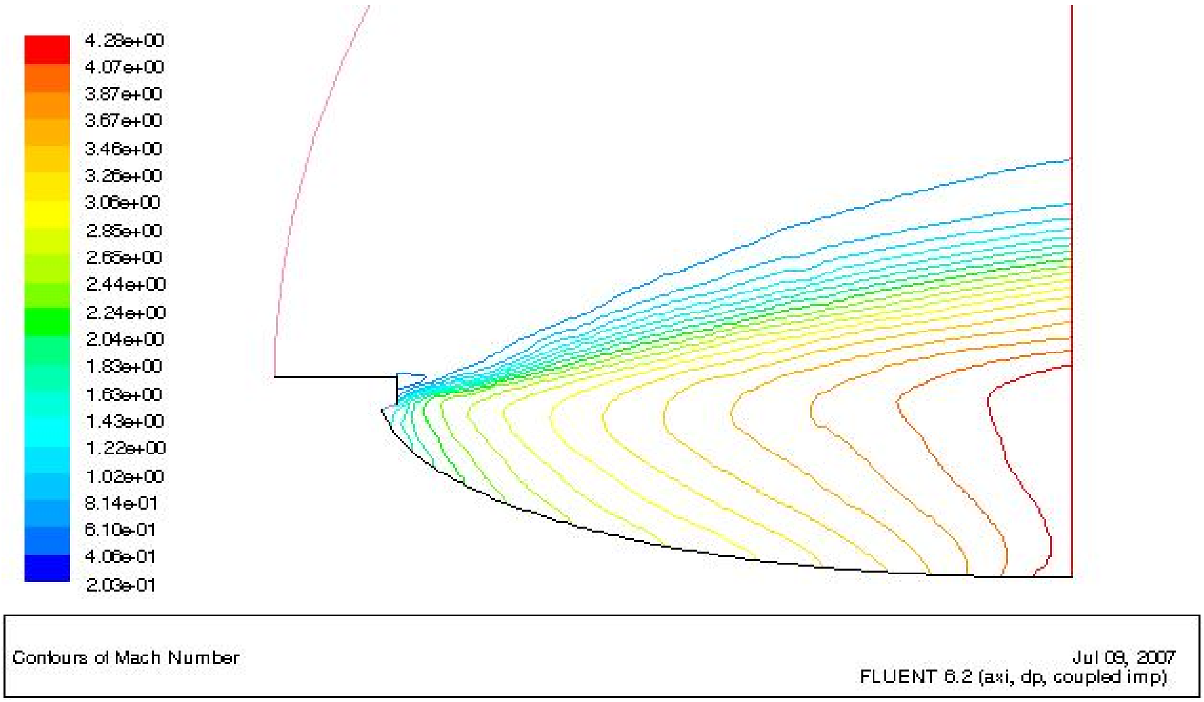


Figure D.1.29: *Mach Plot for Mach 4.0, 100% Length External Aerospike Air Nozzle*

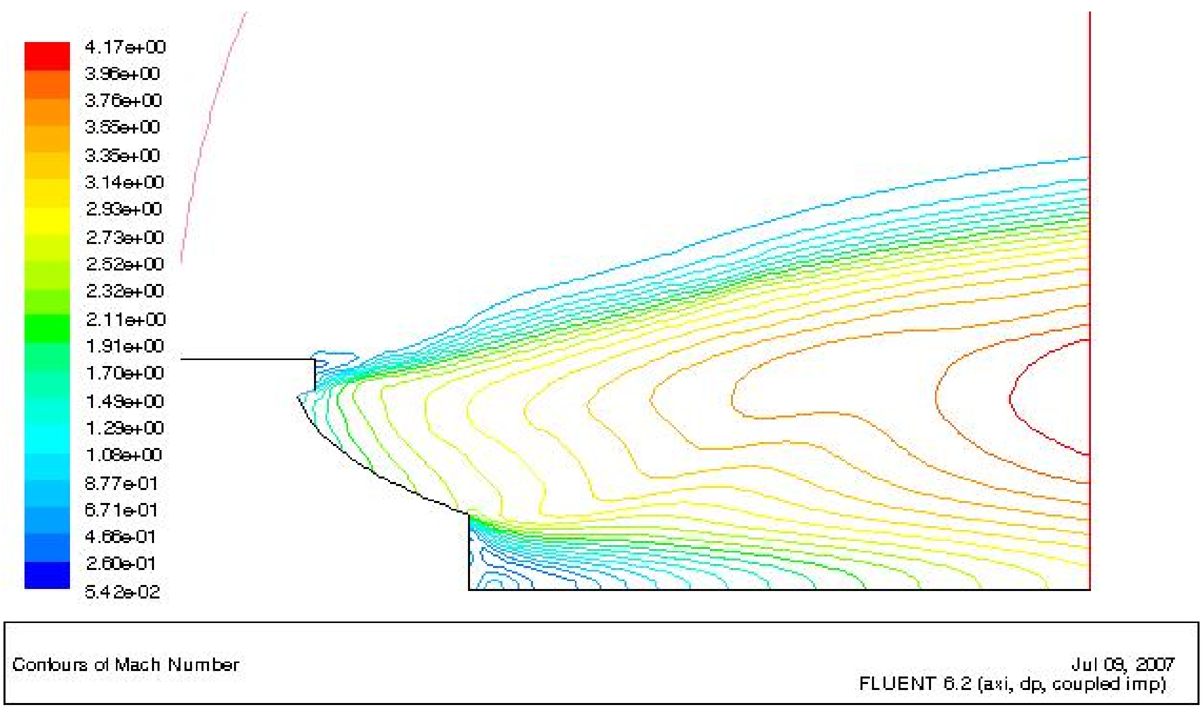


Figure D.1.30: *Mach Plot for Mach 4.0, 20% Length External Aerospike Air Nozzle*

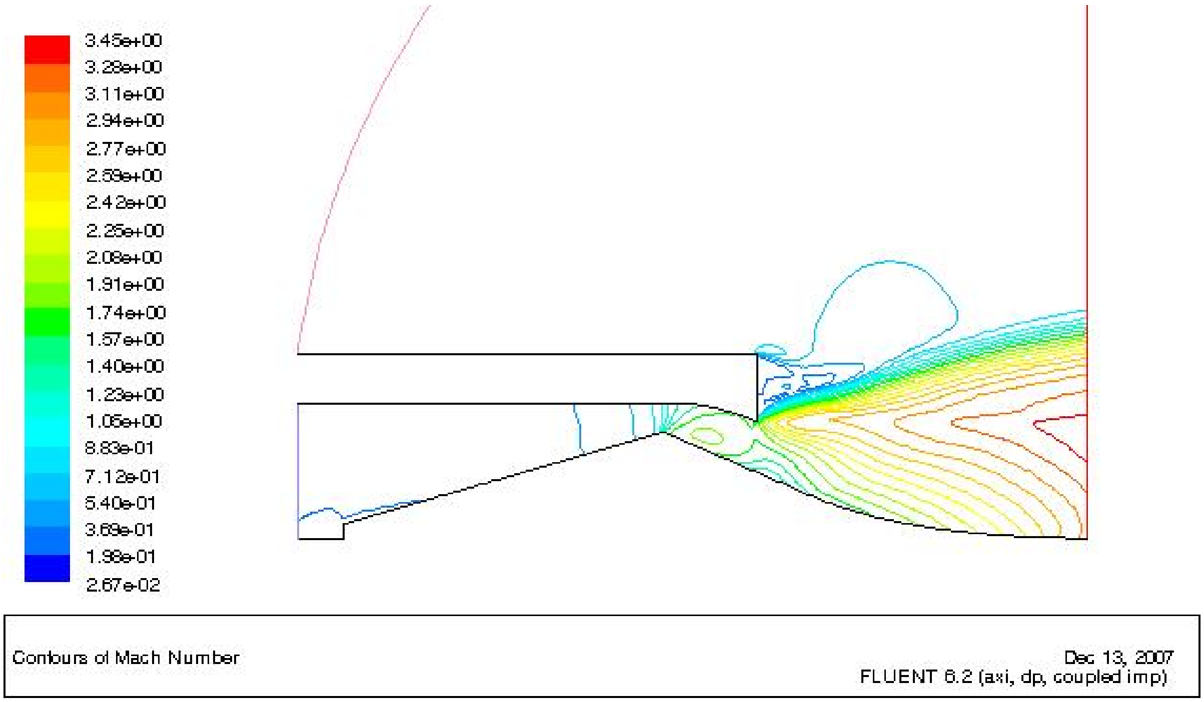


Figure D.1.31: *Mach Plot for Mach 3.0, 100% Length Internal-External Aerospike Air Nozzle*

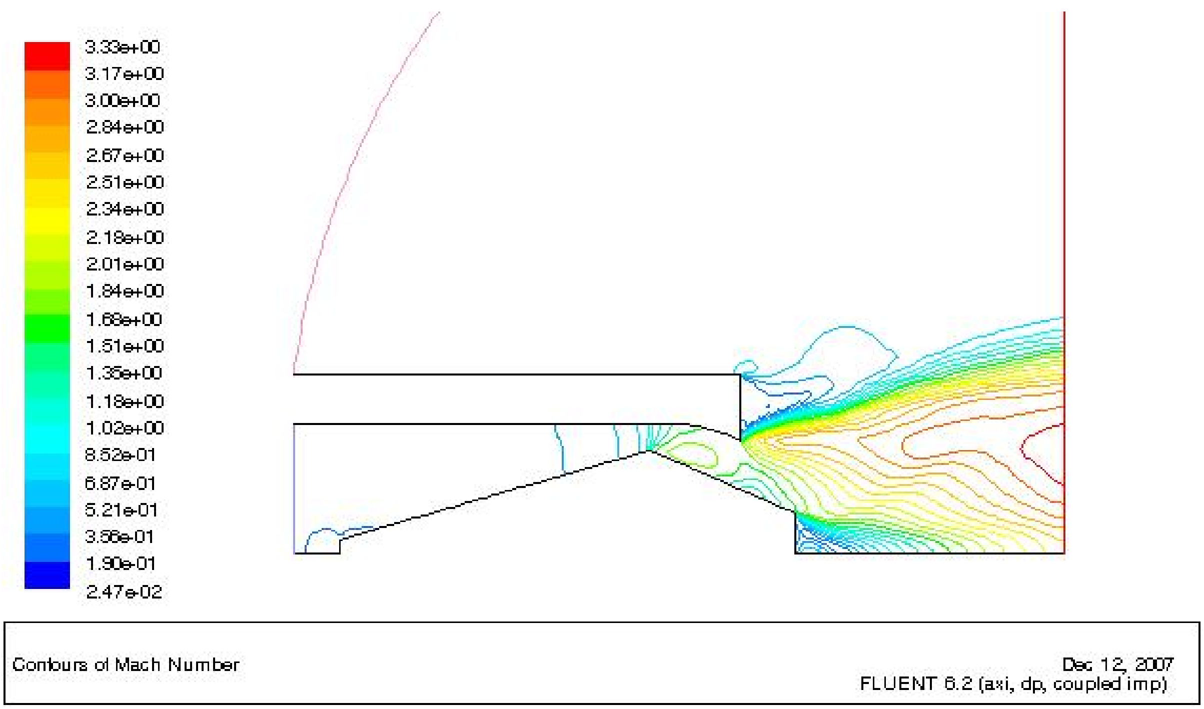


Figure D.1.32: *Mach Plot for Mach 3.0, 20% Length Internal-External Aerospike Air Nozzle*

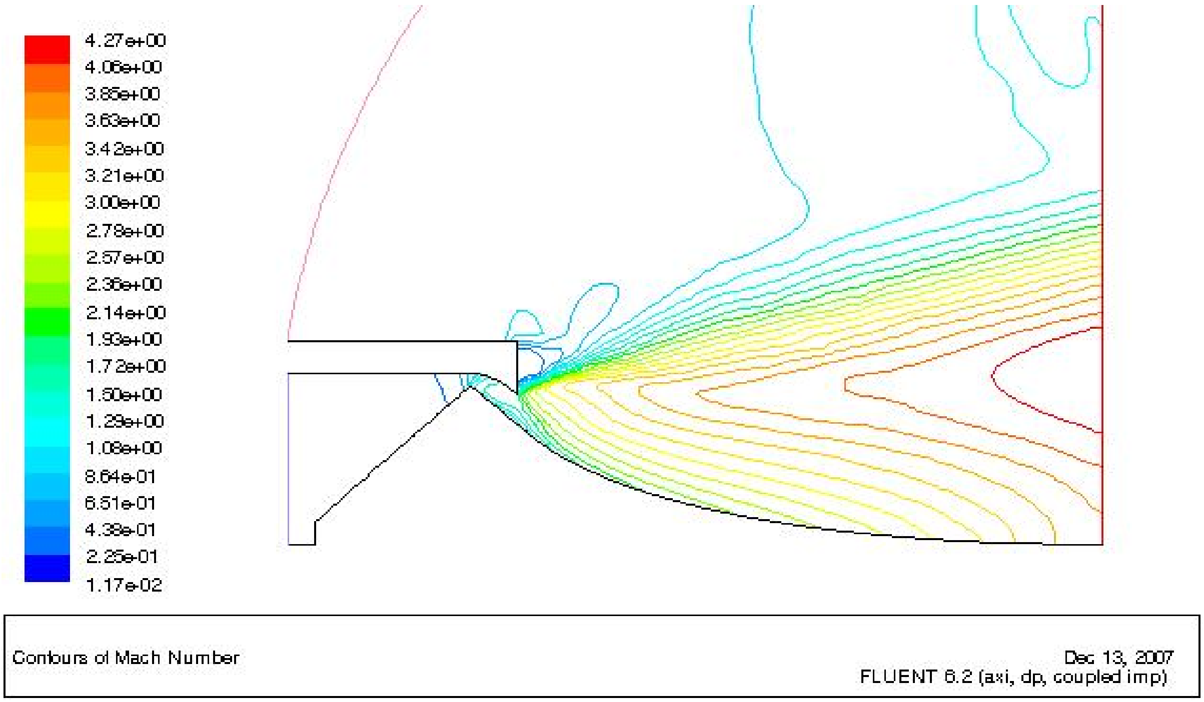


Figure D.1.33: *Mach Plot for Mach 4.0, 100% Length Internal-External Aerospike Air Nozzle*

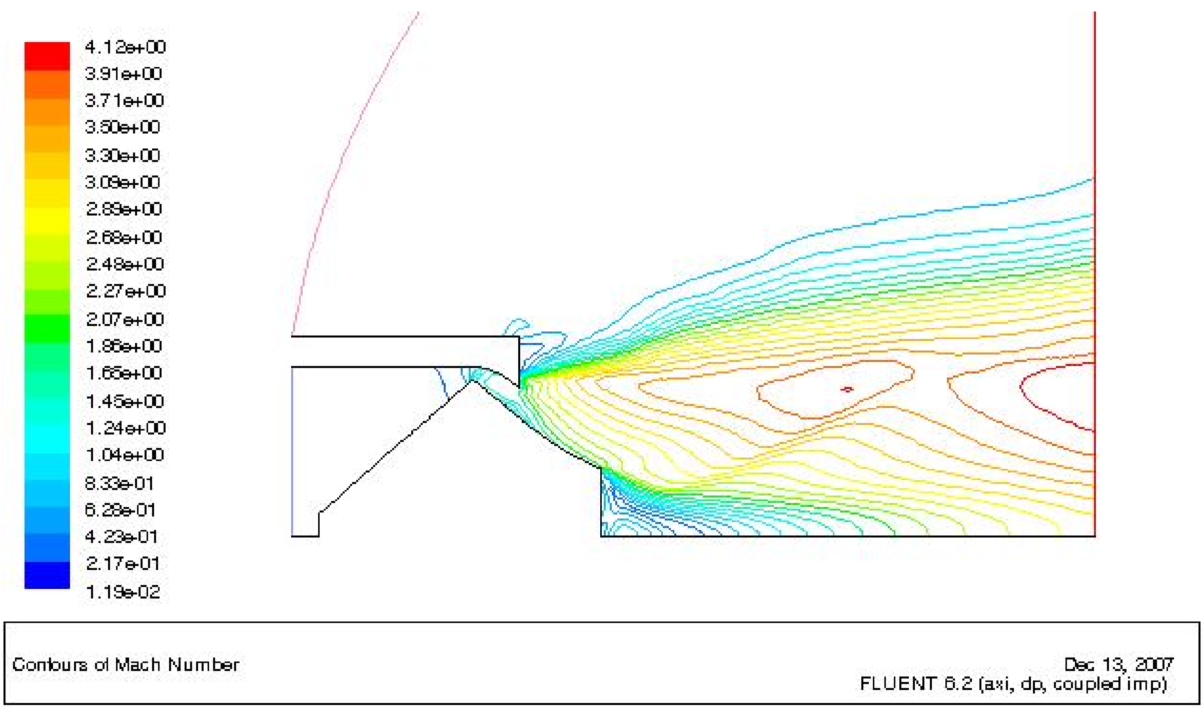


Figure D.1.34: *Mach Plot for Mach 4.0, 20% Length Internal-External Aerospike Air Nozzle*

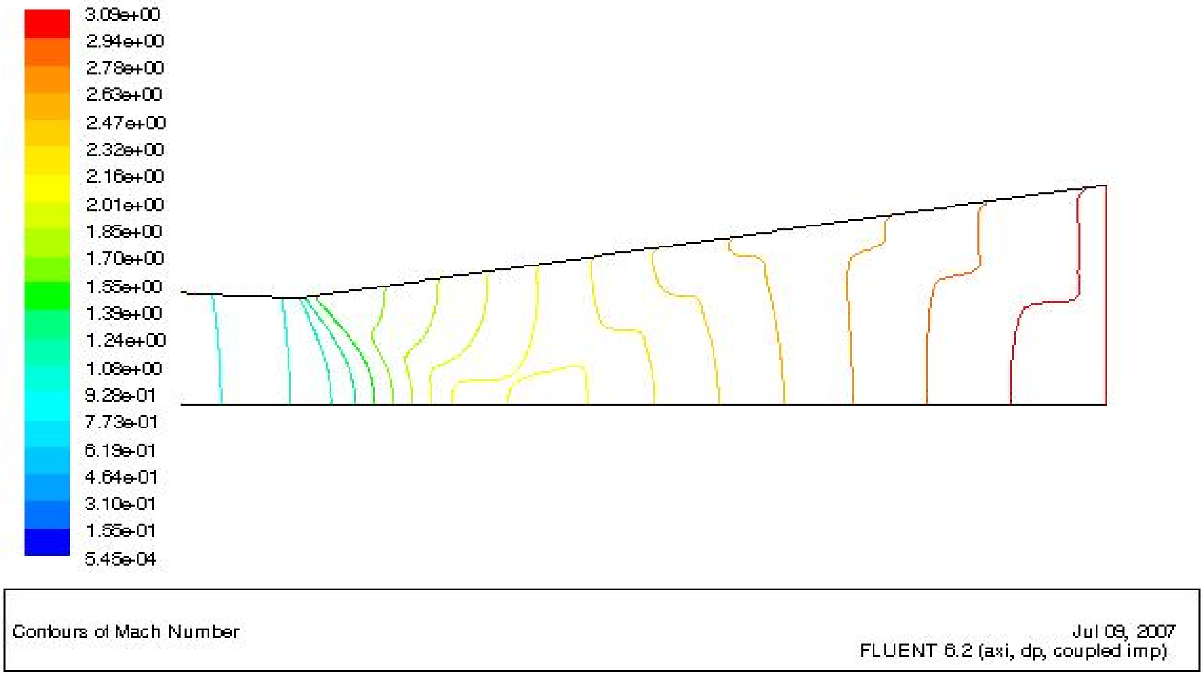


Figure D.1.35: *Mach Plot for Mach 3.0, 8 Degree Conical Air Nozzle*

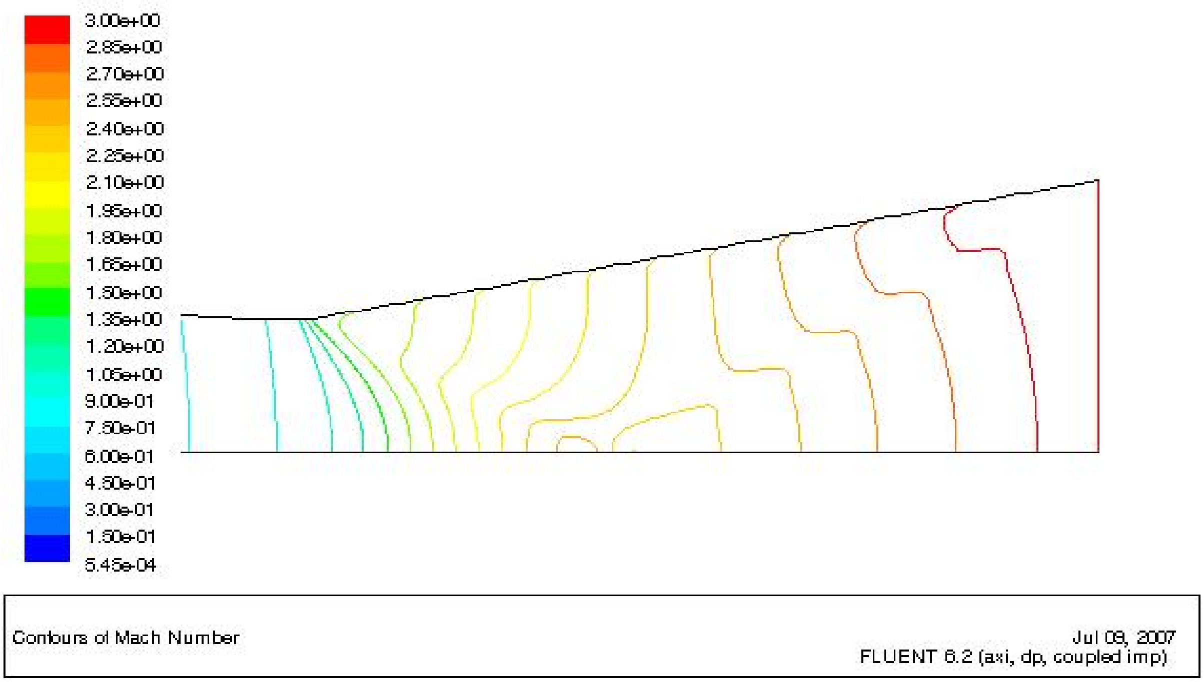


Figure D.1.36: *Mach Plot for Mach 3.0, 10 Degree Conical Air Nozzle*

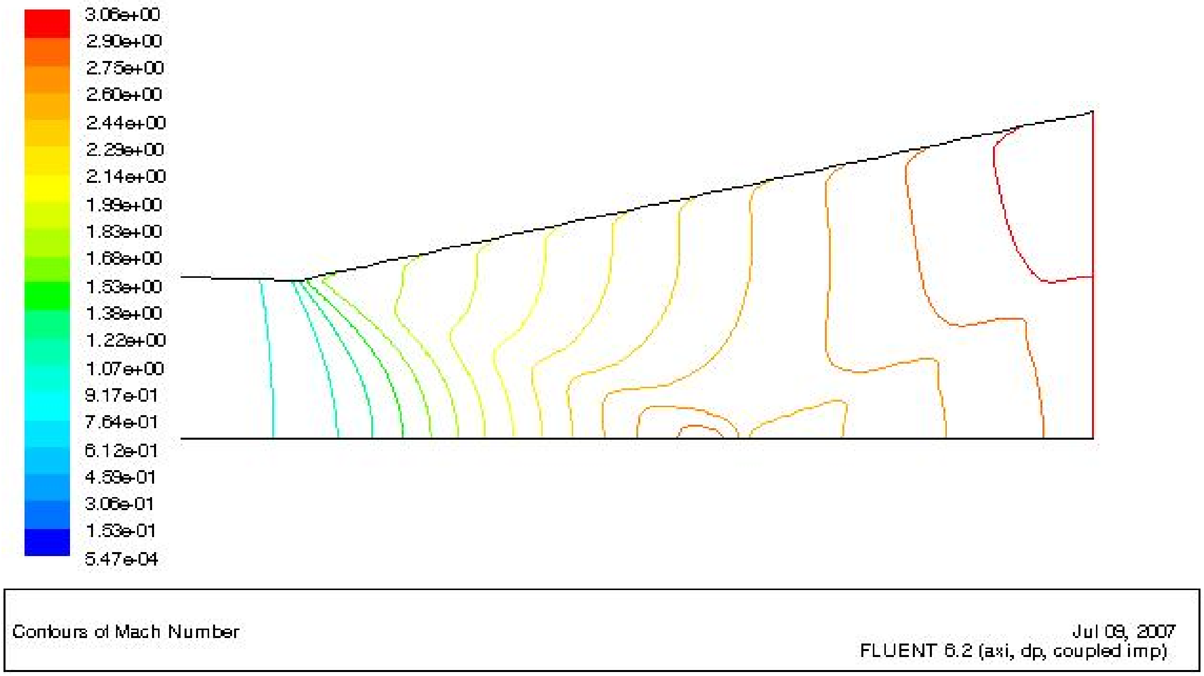


Figure D.1.37: *Mach Plot for Mach 3.0, 12 Degree Conical Air Nozzle*

Figure D.1.38: *Mach Plot for Mach 4.0, 8 Degree Conical Air Nozzle*

Figure D.1.39: *Mach Plot for Mach 4.0, 10 Degree Conical Air Nozzle*

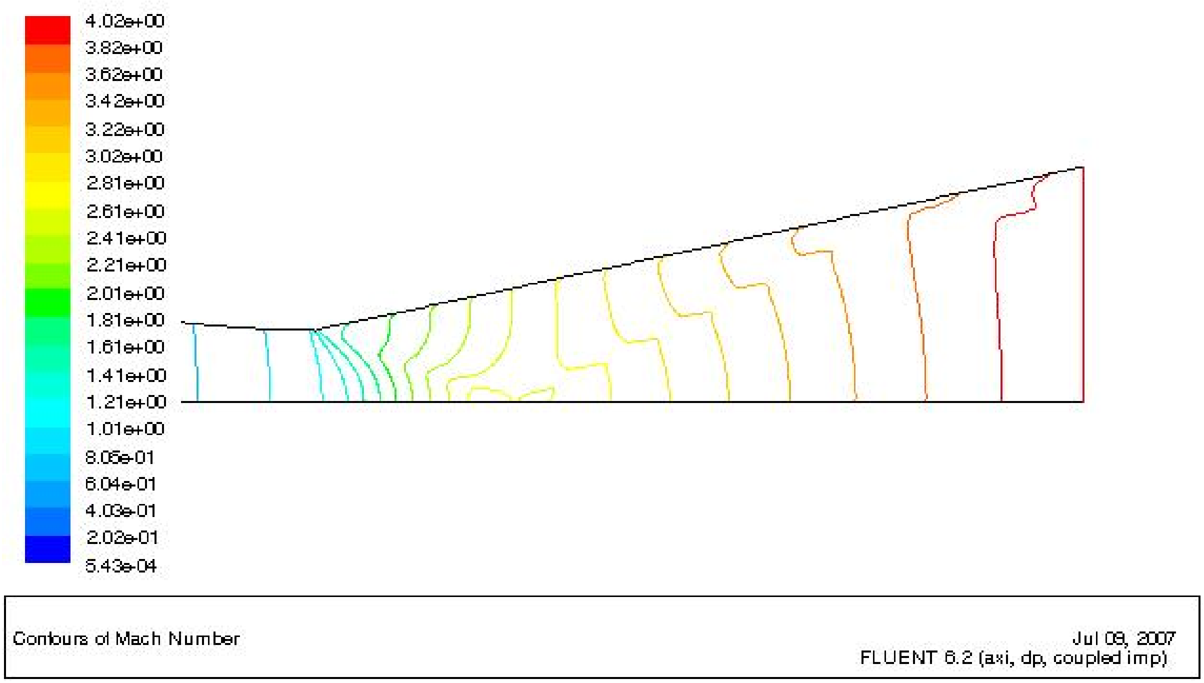


Figure D.1.40: *Mach Plot for Mach 4.0, 12 Degree Conical Air Nozzle*

D.1.5 Exit Mach Number Comparisons

Table D.1.23	**Exit Mach Number Comparisons for Simulated Air Nozzles for $Ma_{Desired}$ = 3.0, r_{throat} = 1.0 (Dimensionless)**			
	$Ma_{ComputerCalculated}$	% Error	$Ma_{FluentCalculated}$	% Error
Annular Nozzle $\beta = 0.2 \cdot r_{throat}$	3.0098	0.33%	3.0227	0.76%
Annular Nozzle $\beta = 0.5 \cdot r_{throat}$	3.0021	0.07%	3.0161	0.54%
Annular Nozzle $\beta = 1.0 \cdot r_{throat}$	3.0067	0.22%	3.0139	0.46%
External Aerospike, 100 % Length	3.0000	0.00%	-	-
External Aerospike, 20% Length	3.0000	0.00%	-	-
Internal-External Aerospike, 100% Length	3.0029	0.10%	3.2791	9.30%
Internal-External Aerospike, 20% Length	3.0029	0.10%	3.1535	5.12%
Conical Nozzle, 8 Degree	-	-	2.9904	-0.32%
Conical Nozzle, 10 Degree	-	-	2.9815	-0.62%
Conical Nozzle, 12 Degree	-	-	2.9830	-0.57%

Table D.1.24	**Exit Mach Number Comparisons for Simulated Air Nozzles for $Ma_{Desired}$ = 4.0, r_{throat} = 1.0 (Dimensionless)**			
	$Ma_{ComputerCalculated}$	% Error	$Ma_{FluentCalculated}$	% Error
Annular Nozzle $\beta = 0.2 \cdot r_{throat}$	4.0020	0.05%	4.0249	0.62%
Annular Nozzle $\beta = 0.5 \cdot r_{throat}$	4.0017	0.04%	4.0233	0.58%
Annular Nozzle $\beta = 1.0 \cdot r_{throat}$	4.0102	0.26%	4.0321	0.80%
External Aerospike, 100 % Length	4.0000	0.00%	-	-
External Aerospike, 20% Length	4.0000	0.00%	-	-
Internal-External Aerospike, 100% Length	4.0011	0.03%	3.9879	-0.30%
Internal-External Aerospike, 20% Length	4.0011	0.03%	3.7966	-5.09%
Conical Nozzle, 8 Degree	-	-	3.9882	-0.30%
Conical Nozzle, 10 Degree	-	-	3.9856	-0.36%
Conical Nozzle, 12 Degree	-	-	3.9781	-0.55%

D.2 N_2O/HTPB Exhaust Nozzle Simulations

D.2.1 Gambit Inputs and Mesh Results

Annular Nozzles

Table D.2.1	**Ma = 3.0 $\beta = 0.2 \cdot r_{throat}$ Annular Exhaust Nozzle Gambit Meshing Inputs and Results**		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	36800	

Table D.2.2	**Ma = 3.0 $\beta = 0.5 \cdot r_{throat}$ Annular Exhaust Nozzle Gambit Meshing Inputs and Results**		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	36500	

Table D.2.3	**Ma = 3.0 $\beta = 1.0 \cdot r_{throat}$ Annular Exhaust Nozzle Gambit Meshing Inputs and Results**		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	37300	

Table D.2.4	Ma = 4.0 $\beta = 0.2 \cdot r_{throat}$ Annular Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	54300	

Aerospike Nozzles

Table D.2.5	Ma = 3.0 100% Length External Aerospike Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.40
		Number of Cells > 0.97	0
	Total Number of Cells	5958	

Table D.2.6	Ma = 3.0 20% Length External Aerospike Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.44
		Number of Cells > 0.97	0
	Total Number of Cells	6042	

Table D.2.7	Ma = 4.0 100% Length External Aerospike Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.42
		Number of Cells > 0.97	0
	Total Number of Cells	41860	

Table D.2.8	Ma = 4.0 20% Length External Aerospike Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.42
		Number of Cells > 0.97	0
	Total Number of Cells	42247	

Table D.2.9	Ma = 3.0 100% Length Internal-External Aerospike Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.49
		Number of Cells > 0.97	0
	Total Number of Cells	7180	

Table D.2.10	Ma = 3.0 20% Length Internal-External Aerospike Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.05
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.40
		Number of Cells > 0.97	0
	Total Number of Cells	7218	

Table D.2.11	Ma = 4.0 100% Length Internal-External Aerospike Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.1
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.7
		Number of Cells > 0.97	0
	Total Number of Cells	12808	

Table D.2.12	Ma = 4.0 20% Length Internal-External Aerospike Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Pave
		Smoother:	None
	Spacing:	Interval Size:	0.1
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.7
		Number of Cells > 0.97	0
	Total Number of Cells	13059	

Conical Nozzles

Table D.2.13	Ma = 3.0 $\theta = 8°$ Conical Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	36100	

Table D.2.14	Ma = 3.0 $\theta = 10°$ Conical Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	33300	

Table D.2.15	Ma = 3.0 $\theta = 12°$ Conical Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	31500	

Table D.2.16	Ma = 4.0 $\theta = 8°$ Conical Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.53
		Number of Cells > 0.97	0
	Total Number of Cells	54700	

Table D.2.17	Ma = 4.0 $\theta = 10°$ Conical Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	48300	

Table D.2.18	Ma = 4.0 $\theta = 12°$ Conical Exhaust Nozzle Gambit Meshing Inputs and Results		
Mesh Conditions:	Scheme:	Elements:	Quad
		Type:	Map
		Smoother:	None
	Spacing:	Interval Size:	0.01
Mesh Quality:	Equiangle Skew:	Lowest:	0.0
		Highest:	0.51
		Number of Cells > 0.97	0
	Total Number of Cells	44000	

D.2.2 FLUENT Inputs

Table D.2.19	**FLUENT Input Conditions Used for Annular and Conical Exhaust Nozzle Simulations**		
Solver:	Solver:	Coupled	
	Space:	Axisymmetric	
	Velocity Formation:	Absolute	
	Gradient Option:	Cell-based	
	Formulation:	Implicit	
	Time:	Steady	
	Porous Formulation:	Superficial Velocity	
	Energy Equation:	Checked	
	Viscous Model:	Inviscid Checked	
Material:	Name:	HTPB	
	Chemical Formulation:	N/A	
	Material Type:	Fluid	
	FLUENT Fluid Material:	HTPB	
	Properties:	Density:	Ideal Gas
		Cp:	1885.7 J/kg*K
		Molecular Weight:	21.403 kg/kmol
Operating Conditions:	Pressure:	Operating Pressure:	0 Pa
	Gravity:	Not Checked	
	Reference Pressure Location:	X(m):	0
		Y(m):	0
Pressure Inlet:	Gauge Total Pressure:	2481800 Pa	Constant
	Supersonic/Initial Gauge Pressure:	2481800 Pa	Constant
	Total Temperature:	1639.2 K	Constant
	Direction Specification Method:	Normal to Boundary	
Pressure Outlet:	Gauge Pressure:	101325 Pa	Constant
	Backflow Total Temperature:	300 K	Constant
	Backflow Direction Specification Method:	Normal to Boundary	
	Non-reflecting Boundary:	Not Checked	
	Target Mass-flow Rate:	Not Checked	
Solution Controls:	Discretization:	Second Order Upwind	
	Solver Parameter:	Courant Number:	1
Solution Initialization:	Compute From:	Pressure Inlet	
	Reference Frame:	Relative to Cell Zone	
	Initial Values:	Automatically Set by *Compute From*	

Table D.2.20	FLUENT Input Conditions Used for N_2O/HTPB Aerospike Exhaust Nozzle Simulations		
Solver:	Solver:	Coupled	
	Space:	Axisymmetric	
	Velocity Formation:	Absolute	
	Gradient Option:	Cell-based	
	Formulation:	Implicit	
	Time:	Steady	
	Porous Formulation:	Superficial Velocity	
	Energy Equation:	Checked	
	Viscous Model:	Inviscid Checked	
	Species:	Transport & Reaction:	Species Transport
			Volumetric
			Mixture-Template
			Laminar Finite-Rate
Material:	Name:	Mixture-Template	
	Chemical Formulation:	N/A	
	Material Type:	Mixture	
	FLUENT Fluid Material:	Mixture-Template	
	Properties:	Density:	Ideal Gas
		Cp:	Mixing Law
		Mechanism:	Reaction-mech
		Reaction:	Finite Rate
		Mixture Species:	names
Operating Conditions:	Pressure:	Operating Pressure:	0 Pa
	Gravity:	Not Checked	
	Reference Pressure Location:	X(m):	0
		Y(m):	0
Pressure Inlet:	Gauge Total Pressure:	2481800 Pa	Constant
	Supersonic/Initial Gauge Pressure:	2481800 Pa	Constant
	Total Temperature:	1639.2 K	Constant
	Direction Specification Method:	Normal to Boundary	
	Species Mass Fraction:	H_2O:	0.0048988
		O_2:	0
		NH_3:	$6.365719e^{-5}$
		HCN:	0.00032836
		H_2:	0.0238565
		H^+:	$4.709419e^{-7}$
		CO_2:	0.005469665
		CO:	0.4800784
		CH_4:	0.0004572298
		N_2:	Auto. Calculated
Pressure Outlet:	Gauge Pressure:	101325 Pa	Constant
	Backflow Total Temperature:	300 K	Constant
	Backflow Direction Specification Method:	Normal to Boundary	
	Non-reflecting Boundary:	Not Checked	
	Target Mass-flow Rate:	Not Checked	
	Species Mass Fraction:	All set to 0	
Far Field:	Gauge Pressure:	101325 Pa	Constant
	Mach Number:	0.6	Constant
	Temperature:	300 K	Constant
	Axial-Component of Flow Direction:	1	Constant
	Radial-Component of Flow Direction:	0	
	Species Mass Fraction:	H_2O:	0.016
		O_2:	0.224
		N_2:	Auto. Calculated
		All others:	0
Solution Controls:	Discretization:	First Order Upwind	
	Solver Parameter:	Courant Number:	1
Solution Initialization:	Compute From:	Pressure Inlet	
	Reference Frame:	Relative to Cell Zone	
	Initial Values:	Automatically Set by *Compute From*	

D.2.3 Entropy Plots from FLUENT Simulations

Annular Nozzle

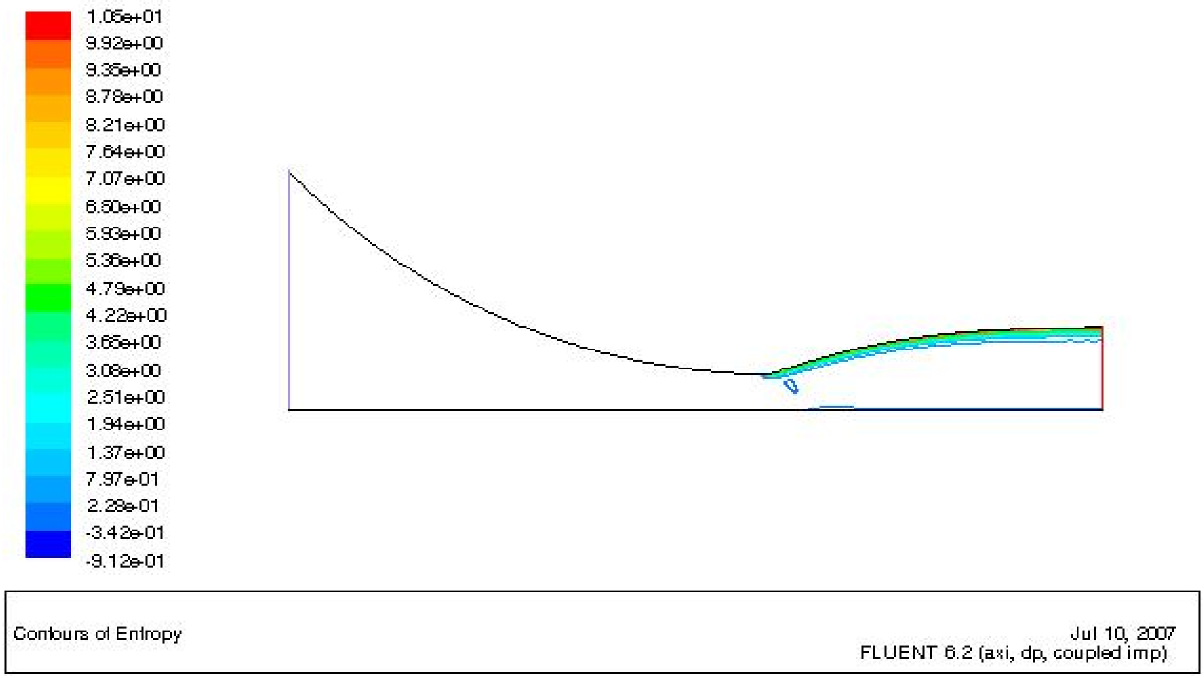


Figure D.2.1: *Entropy Plot for Mach 3.0,* $\beta = 0.2 \cdot r_{throat}$ *Annular Exhaust Nozzle*

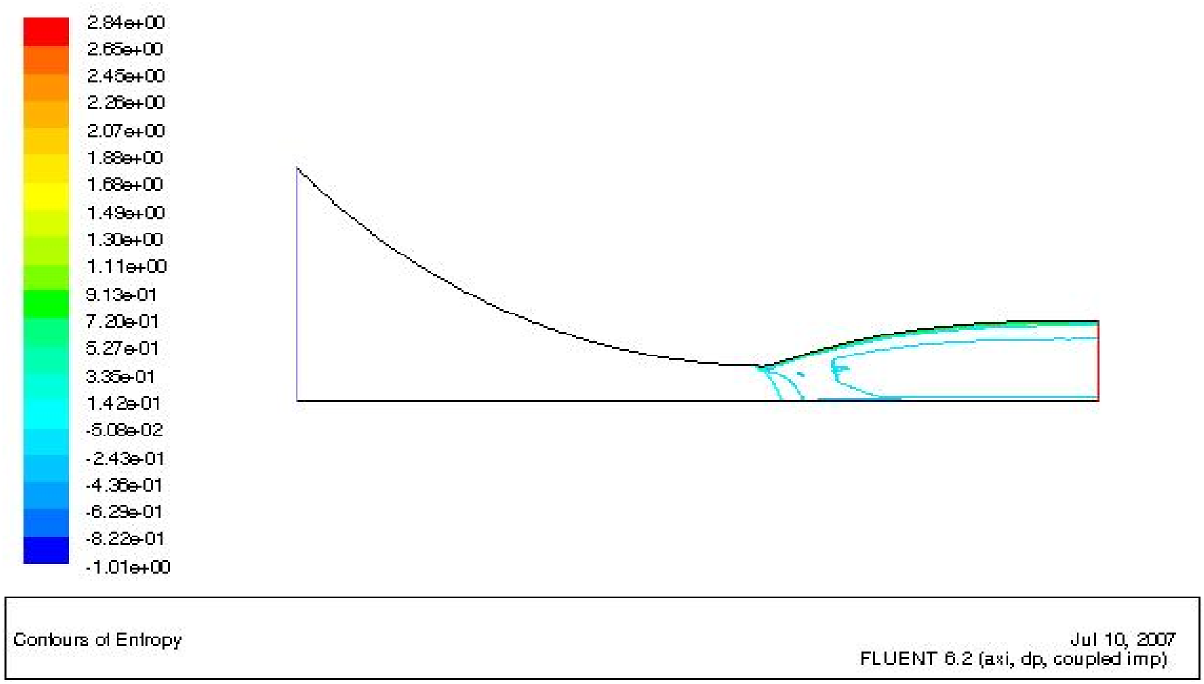


Figure D.2.2: *Entropy Plot for Mach 3.0* $\beta = 0.5 \cdot r_{throat}$ *Annular Exhaust Nozzle*

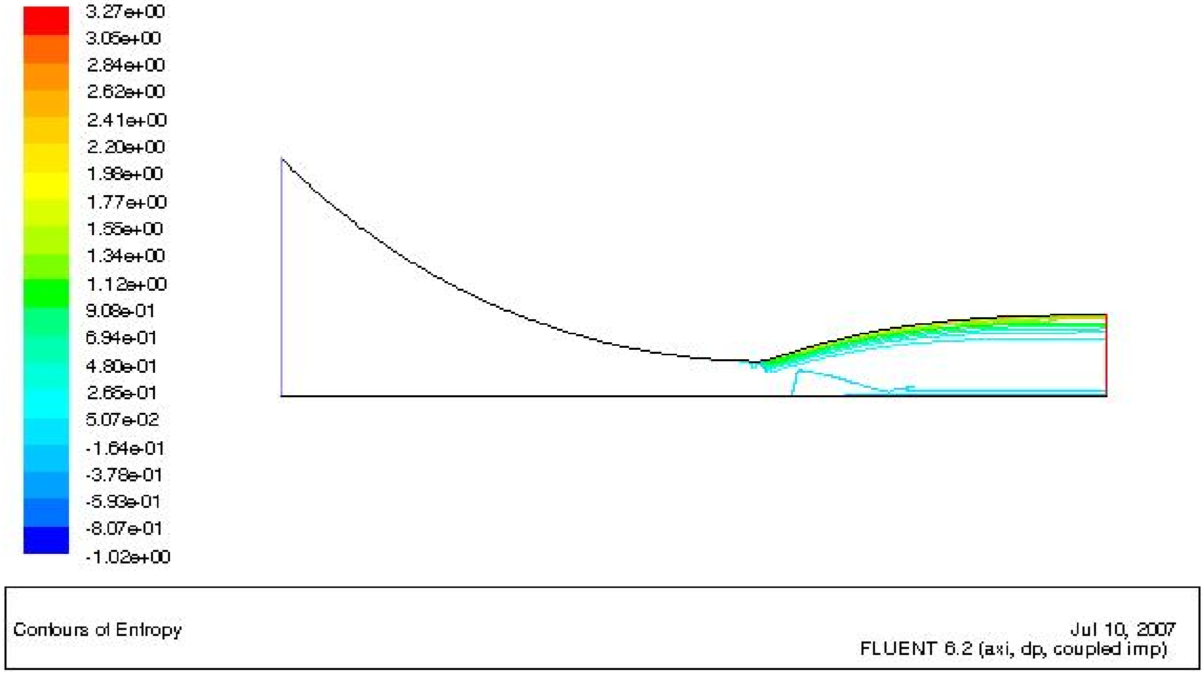


Figure D.2.3: *Entropy Plot for Mach 3.0,* $\beta = 1.0 \cdot r_{throat}$ *Annular Exhaust Nozzle*

Aerospike Nozzles

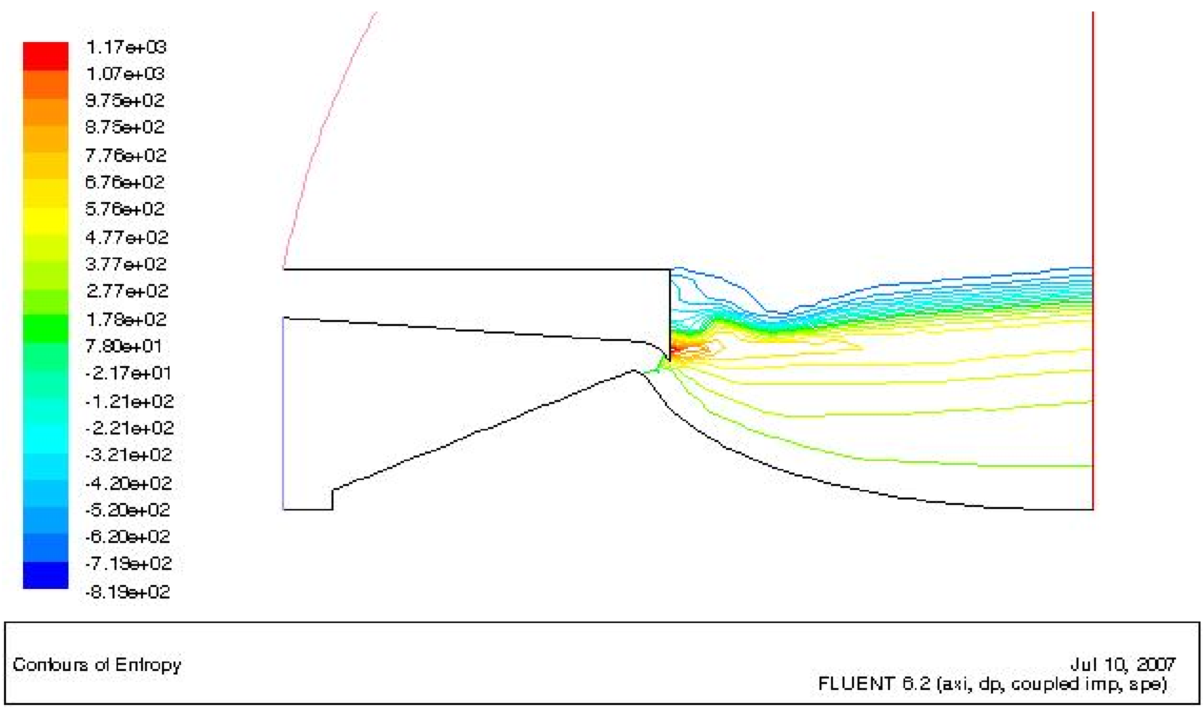


Figure D.2.4: *Entropy Plot for Mach 3.0, 100% Length External Aerospike Exhaust Nozzle*

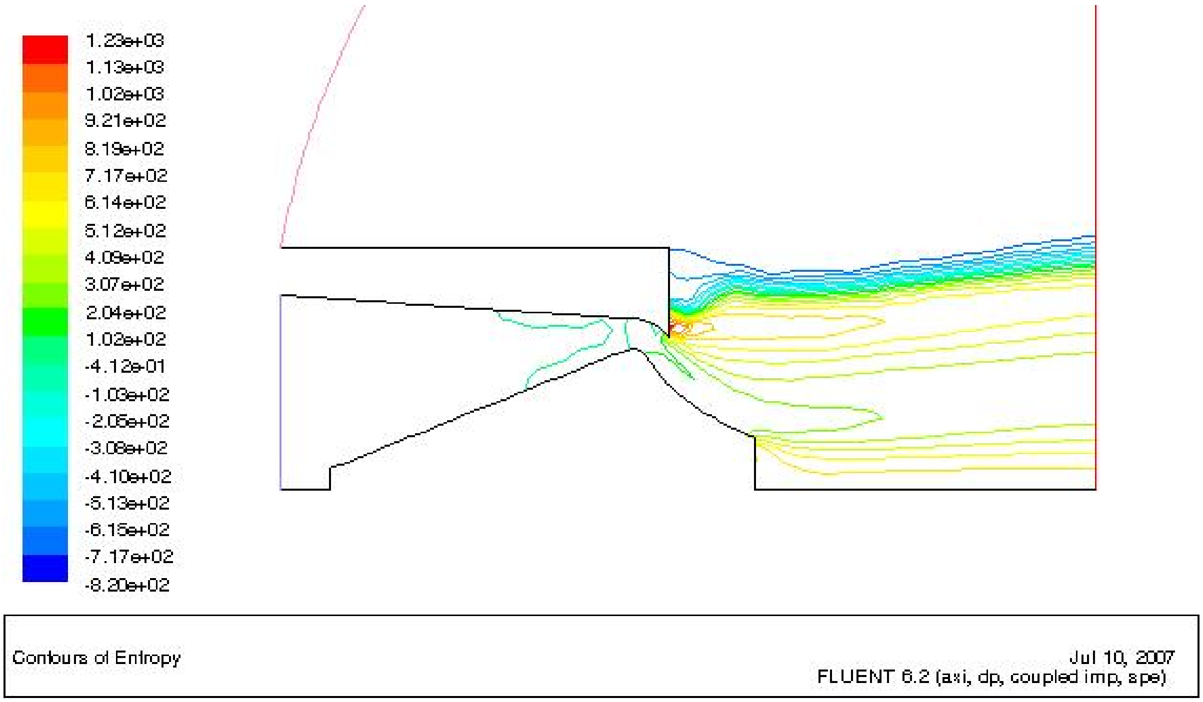


Figure D.2.5: *Entropy Plot for Mach 3.0, 20% Length External Aerospike Exhaust Nozzle*

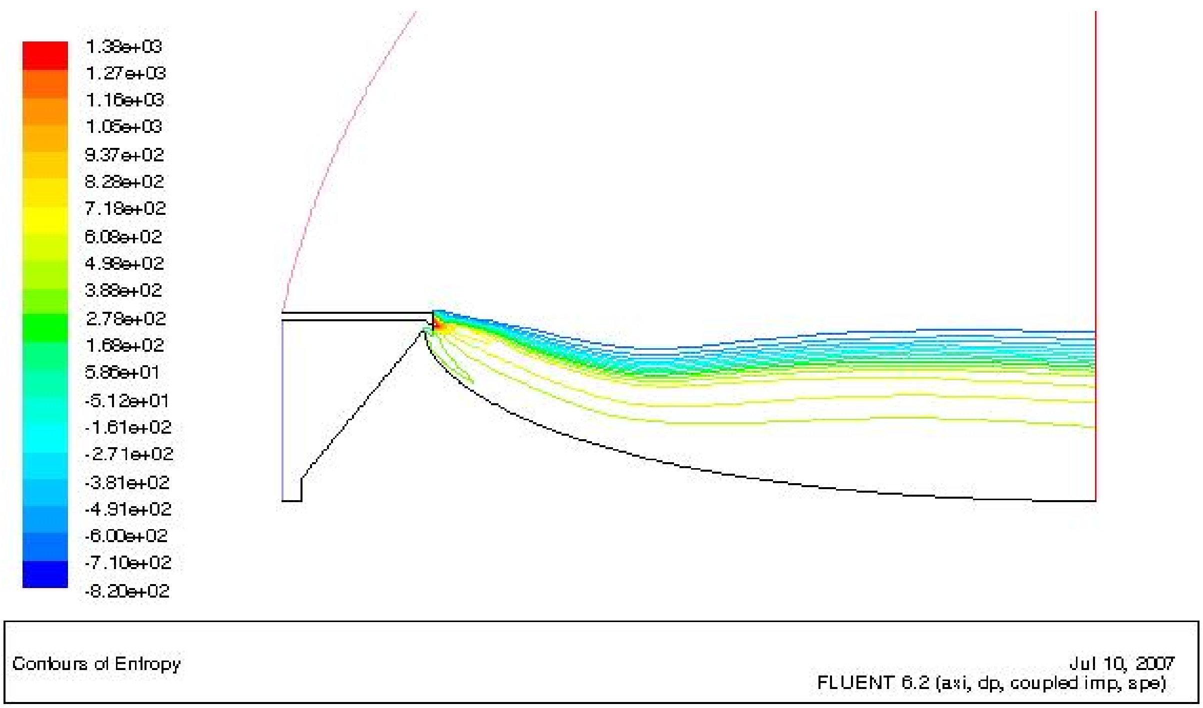


Figure D.2.6: *Entropy Plot for Mach 4.0, 100% Length External Aerospike Exhaust Nozzle*

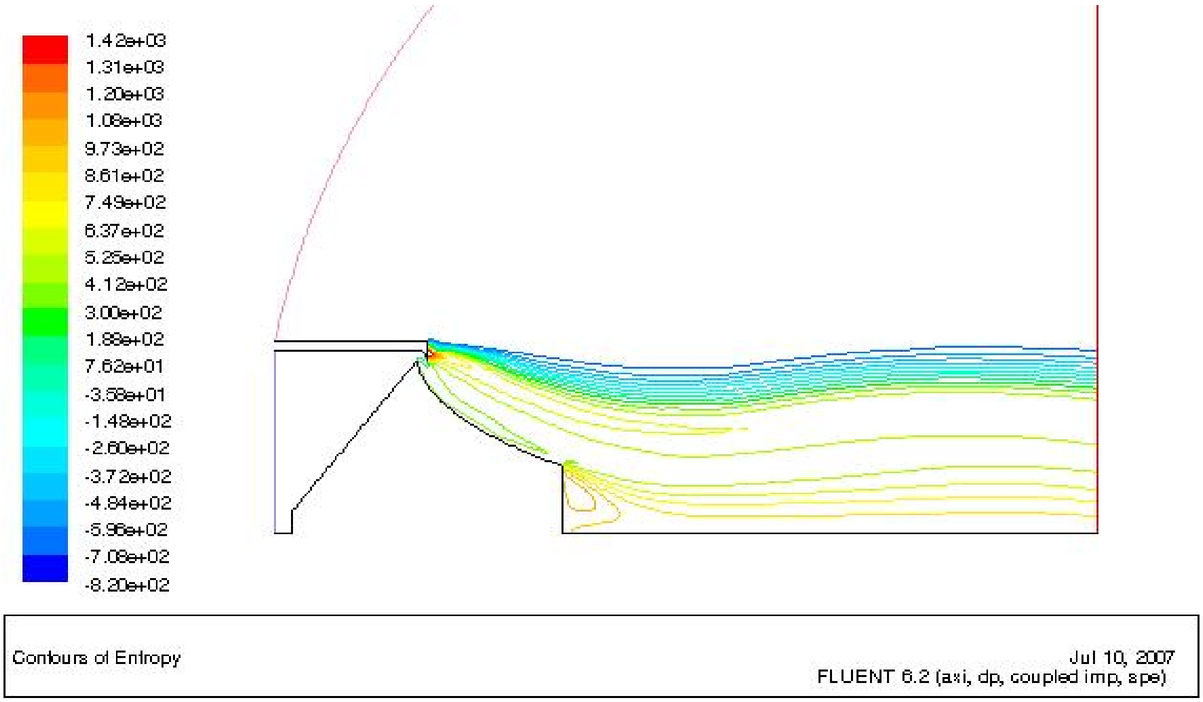


Figure D.2.7: *Entropy Plot for Mach 4.0, 20% Length External Aerospike Exhaust Nozzle*

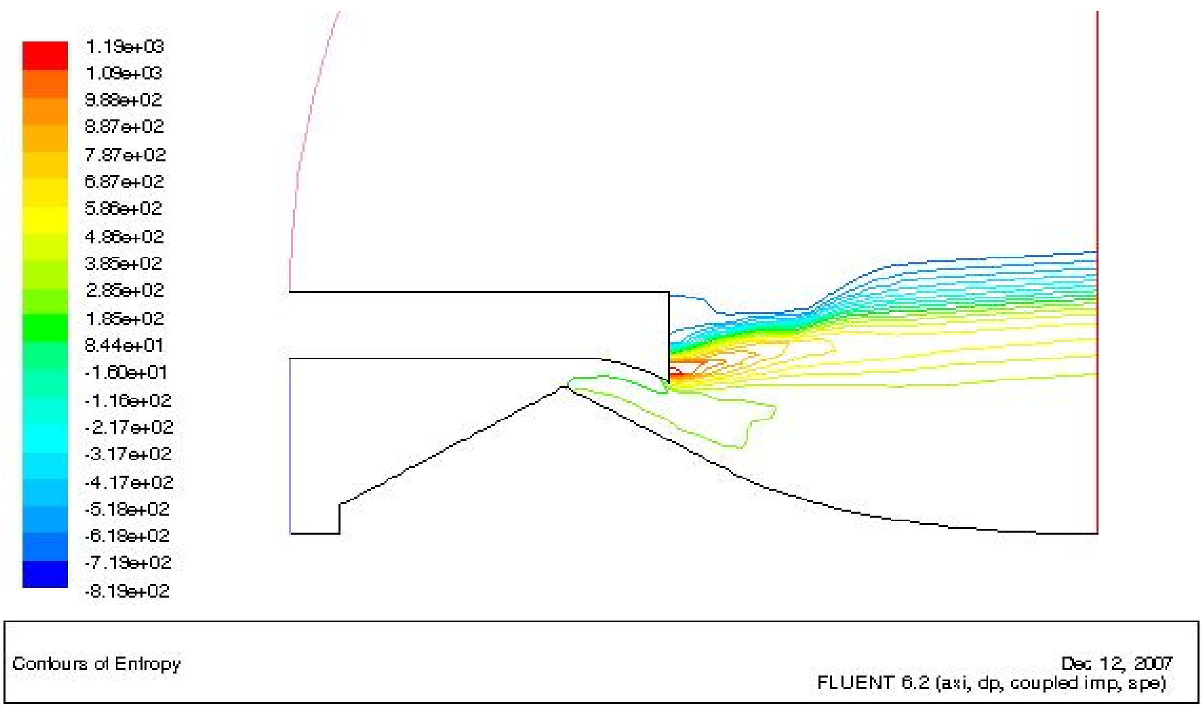


Figure D.2.8: *Entropy Plot for Mach 3.0, 100% Length Internal-External Aerospike Exhaust Nozzle*

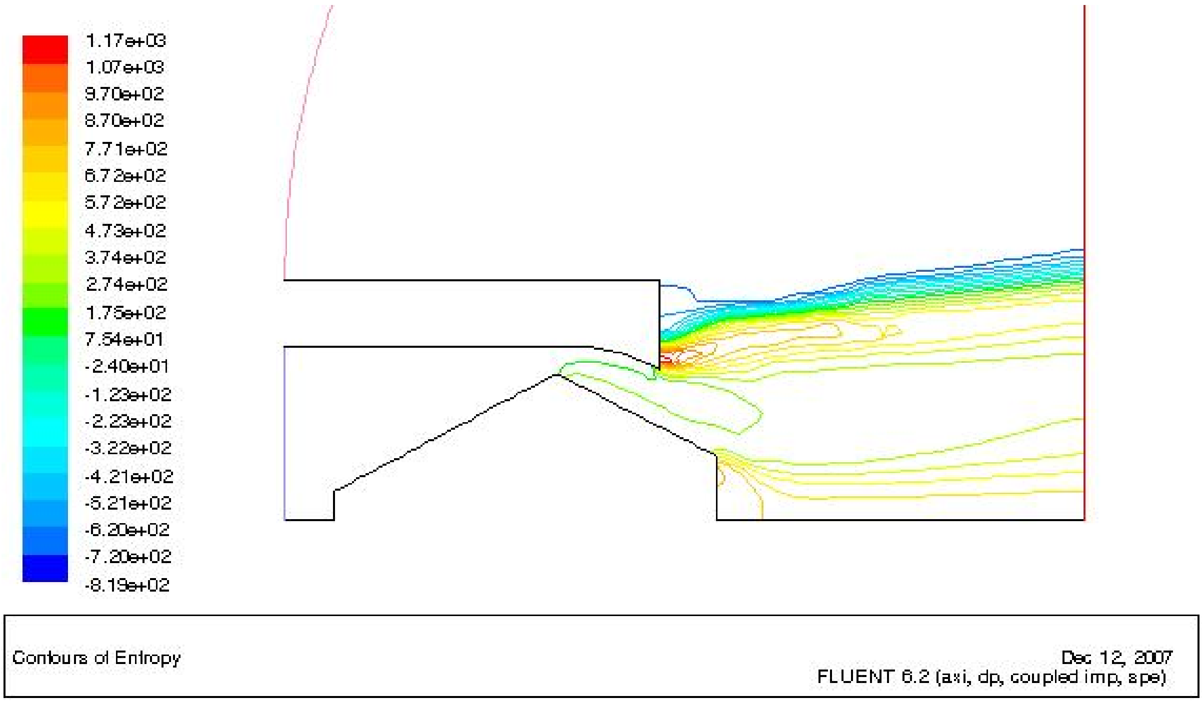


Figure D.2.9: *Entropy Plot for Mach 3.0, 20% Length Internal-External Aerospike Exhaust Nozzle*

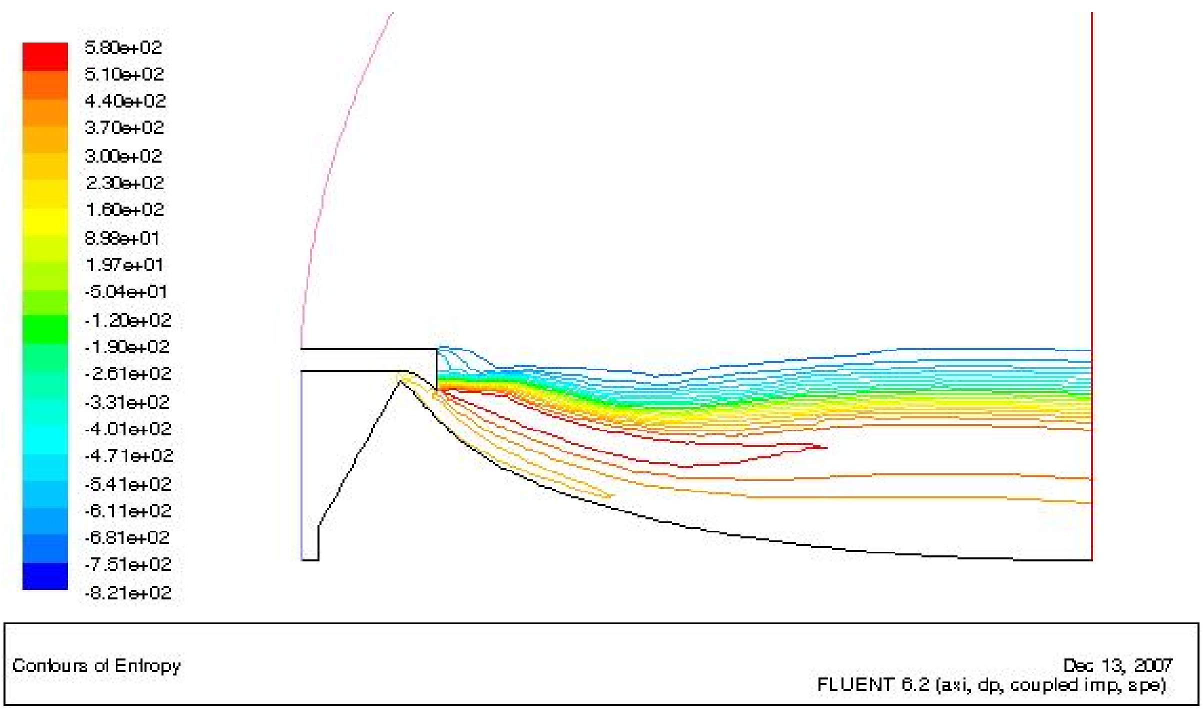


Figure D.2.10: *Entropy Plot for Mach 4.0, 100% Length Internal-External Aerospike Exhaust Nozzle*

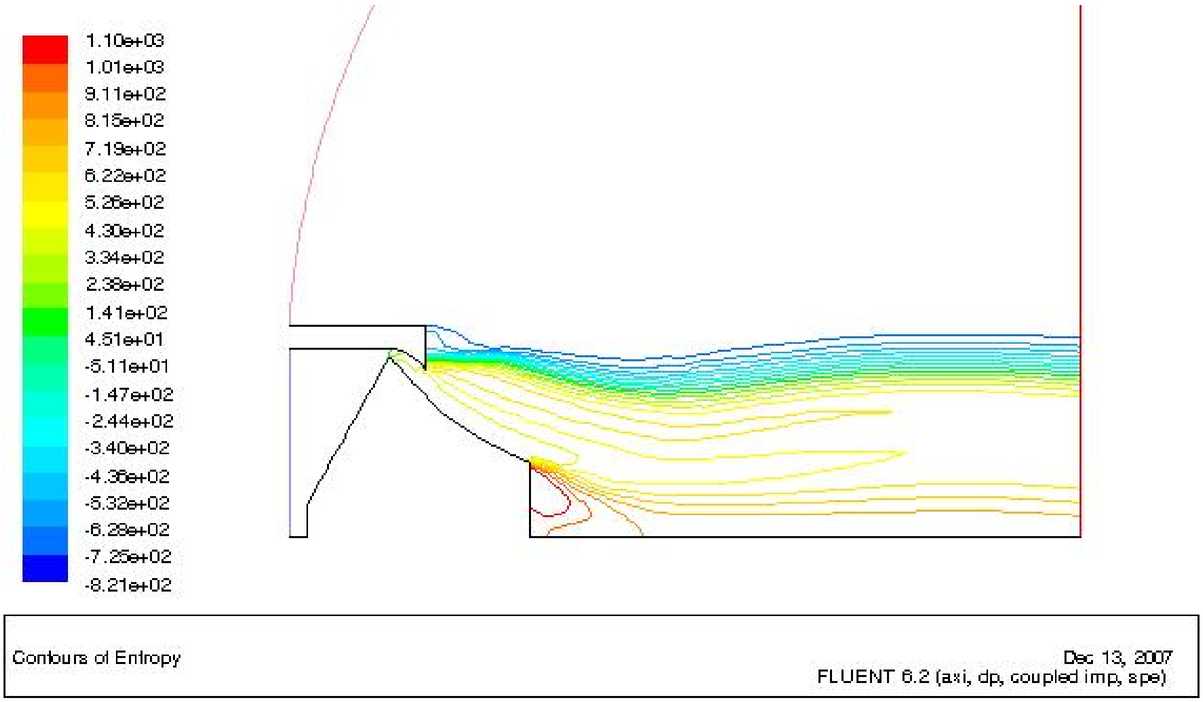


Figure D.2.11: *Entropy Plot for Mach 4.0, 20% Length Internal-External Aerospike Exhaust Nozzle*

Conical Nozzles

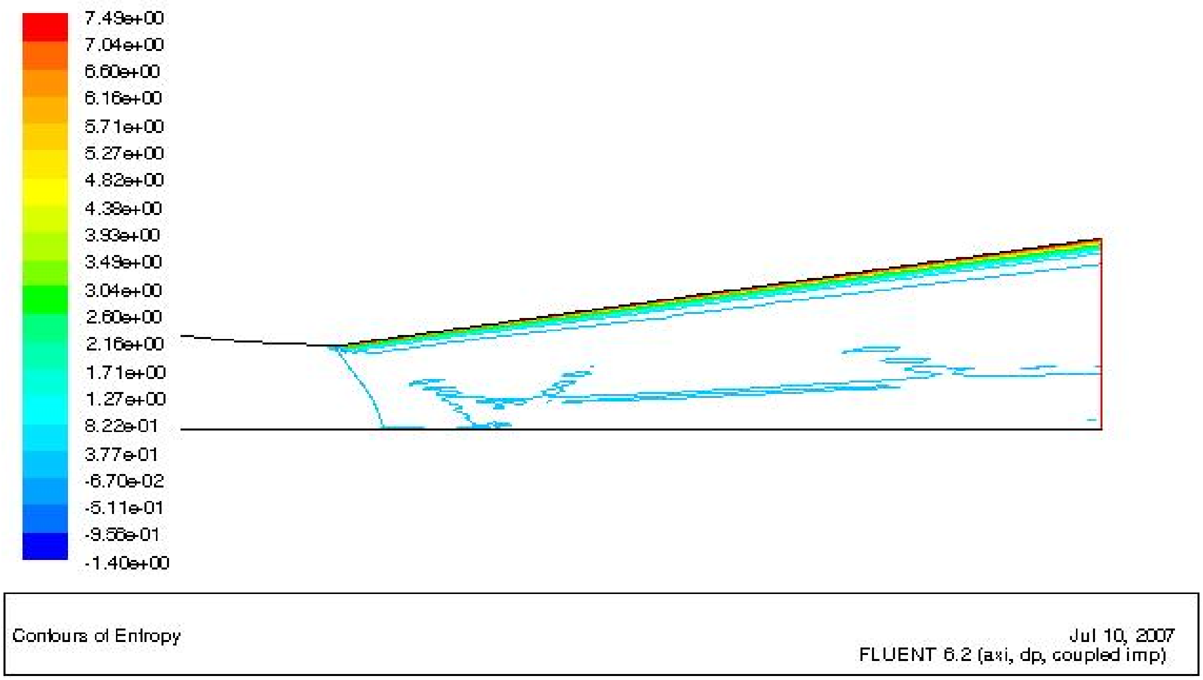


Figure D.2.12: *Entropy Plot for Mach 3.0, 8 Degree Conical Exhaust Nozzle*

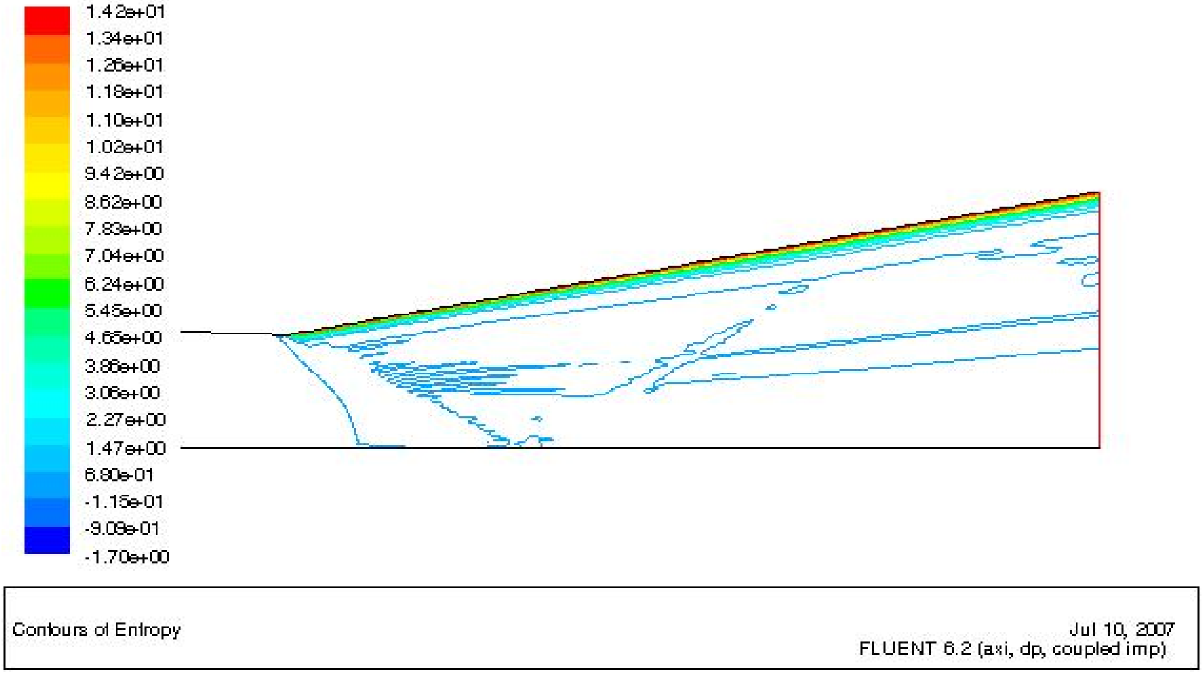


Figure D.2.13: *Entropy Plot for Mach 3.0, 10 Degree Conical Exhaust Nozzle*

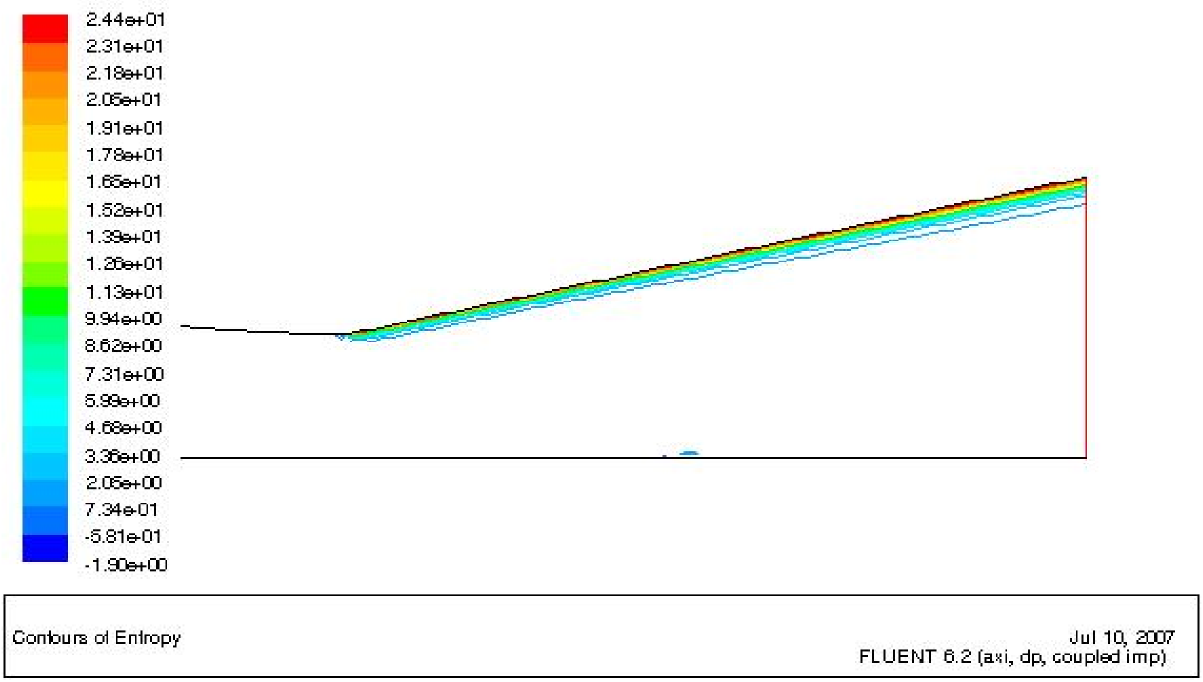


Figure D.2.14: *Entropy Plot for Mach 3.0, 12 Degree Conical Exhaust Nozzle*

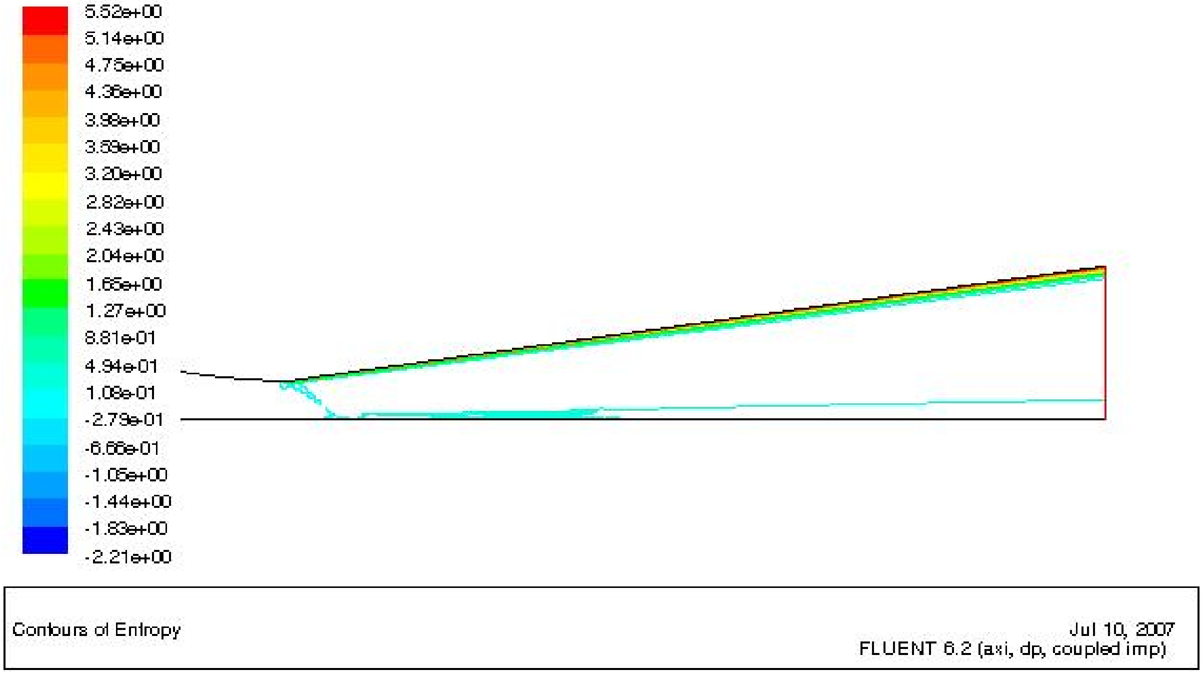


Figure D.2.15: *Entropy Plot for Mach 4.0, 8 Degree Conical Exhaust Nozzle*

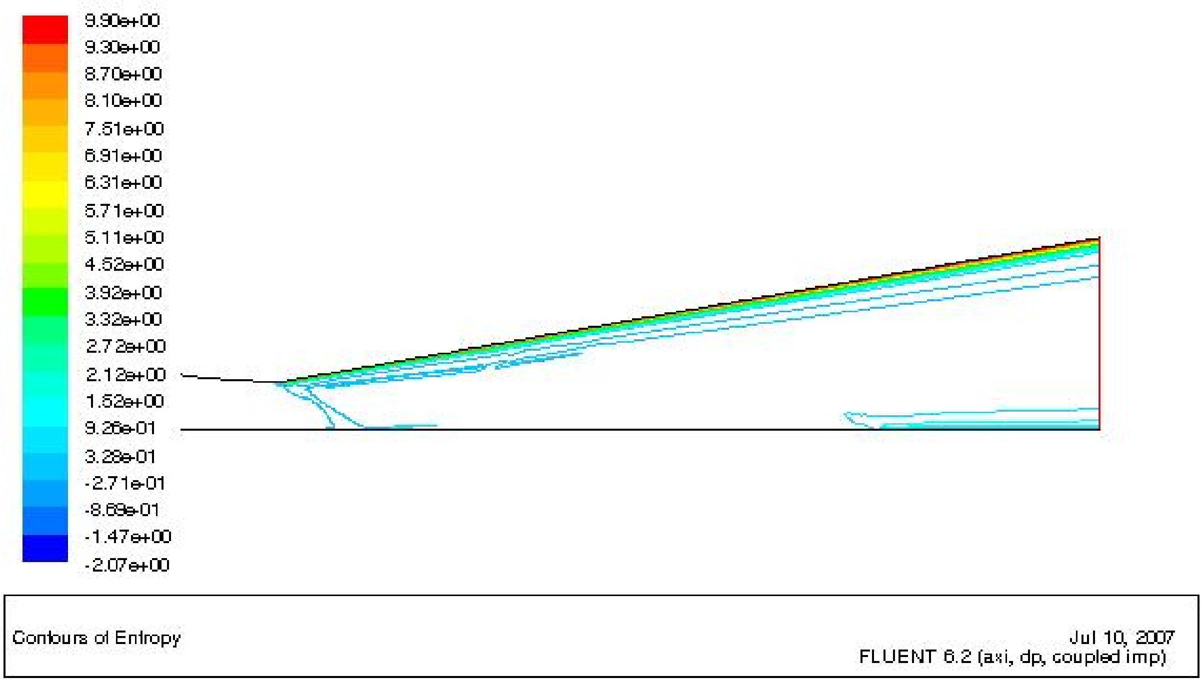


Figure D.2.16: *Entropy Plot for Mach 4.0, 10 Degree Conical Exhaust Nozzle*

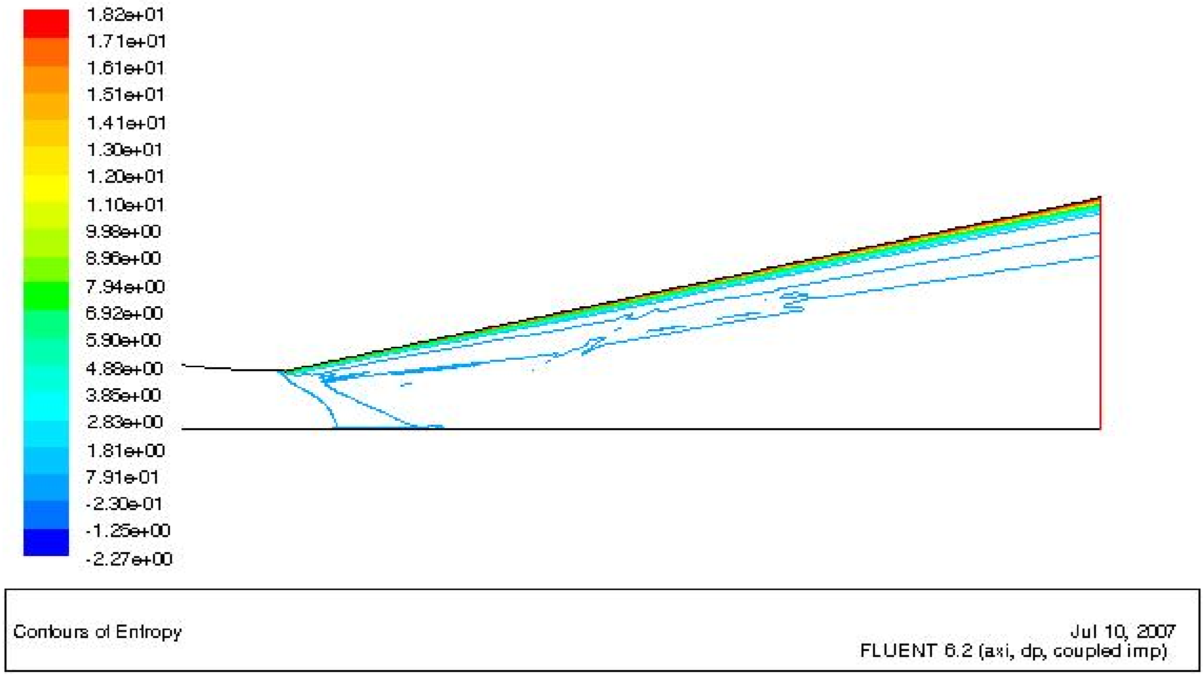


Figure D.2.17: *Entropy Plot for Mach 4.0,12 Degree Conical Exhaust Nozzle*

D.2.4 Mach Plots from FLUENT Simulations

Annular Nozzles

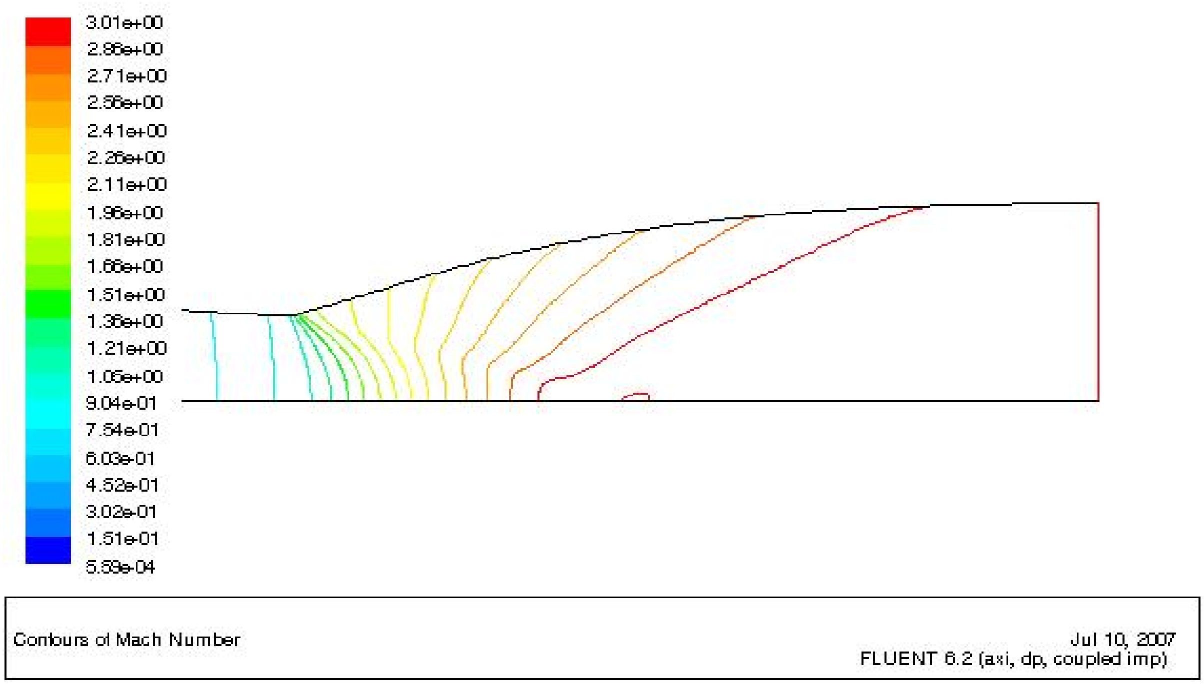


Figure D.2.18: *Mach Plot for Mach 3.0,* $\beta = 0.2 \cdot r_{throat}$ *Annular Exhaust Nozzle*

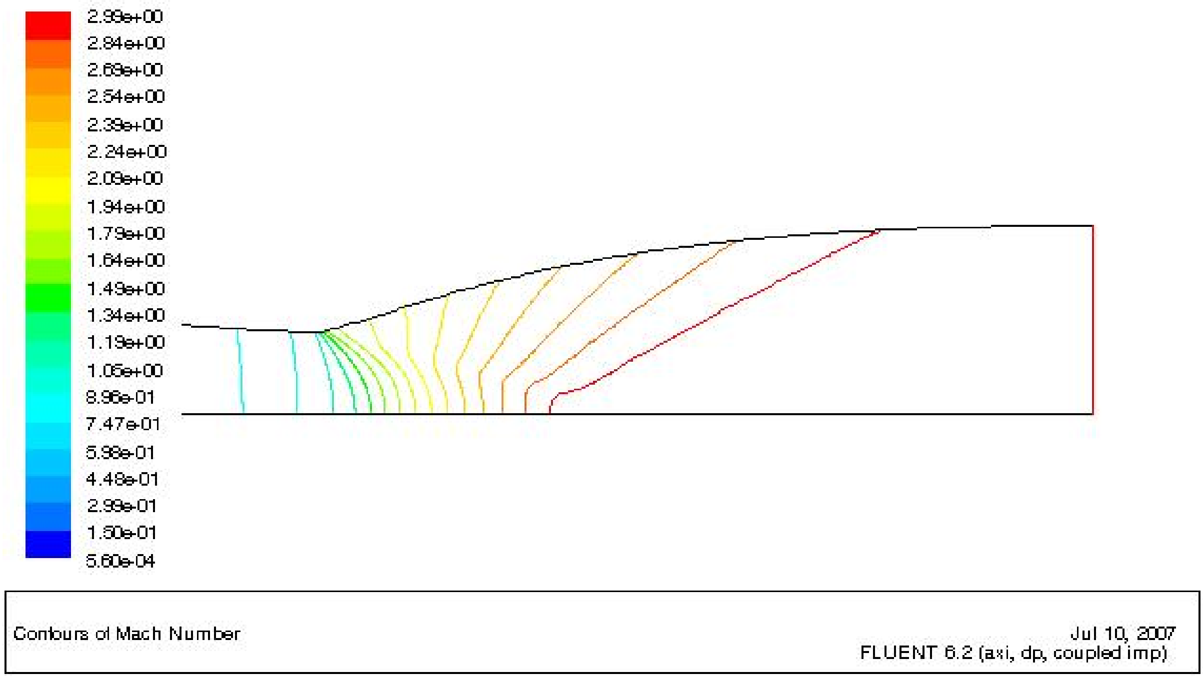


Figure D.2.19: *Mach Plot for Mach 3.0,* $\beta = 0.5 \cdot r_{throat}$ *Annular Exhaust Nozzle*

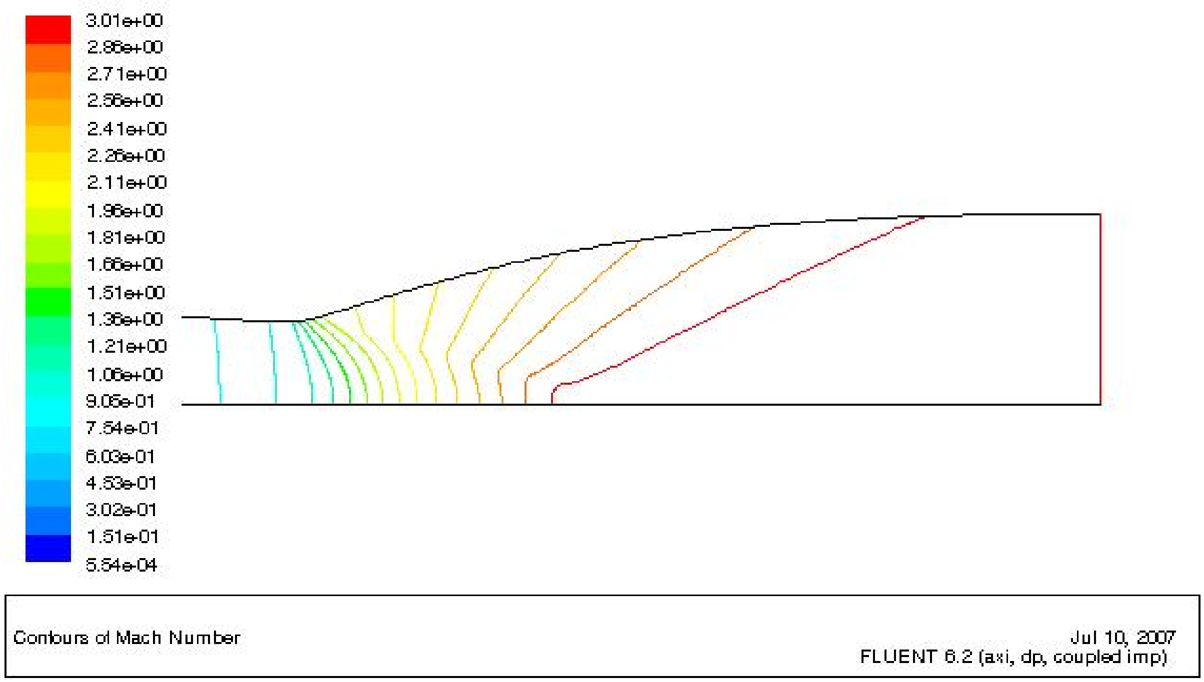


Figure D.2.20: *Mach Plot for Mach 3.0,* $\beta = 1.0 \cdot r_{throat}$ *Annular Exhaust Nozzle*

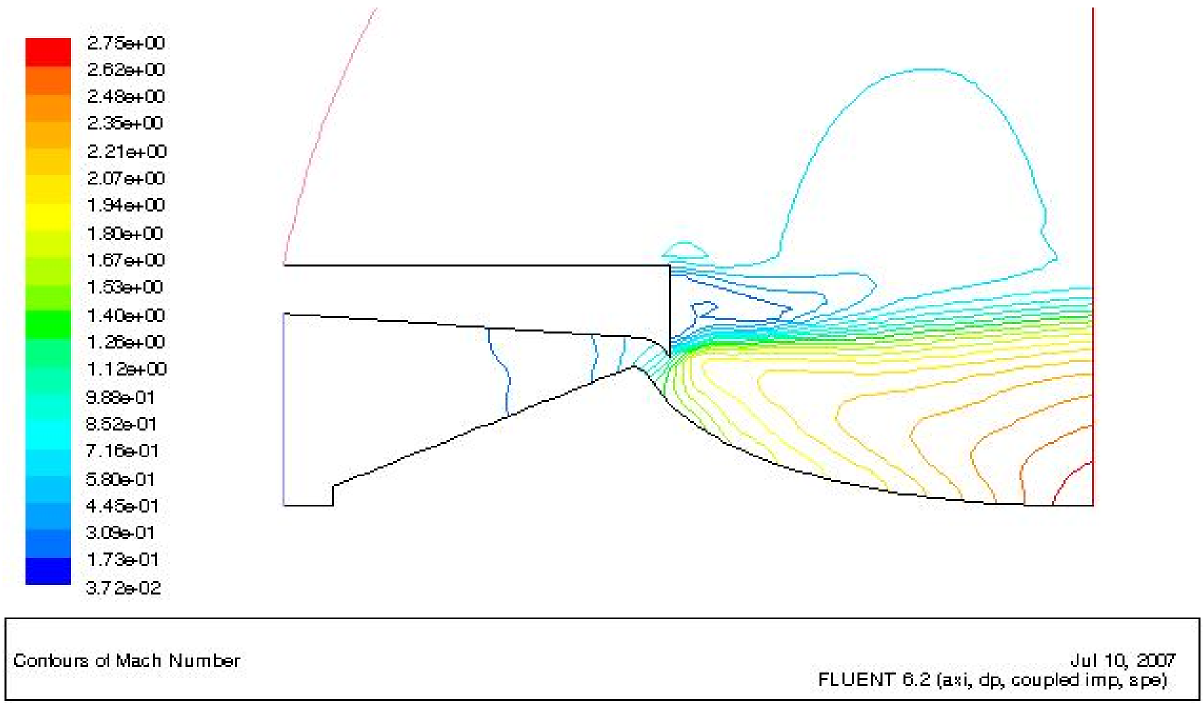


Figure D.2.21: *Mach Plot for Mach 3.0, 100% Length External Aerospike Exhaust Nozzle*

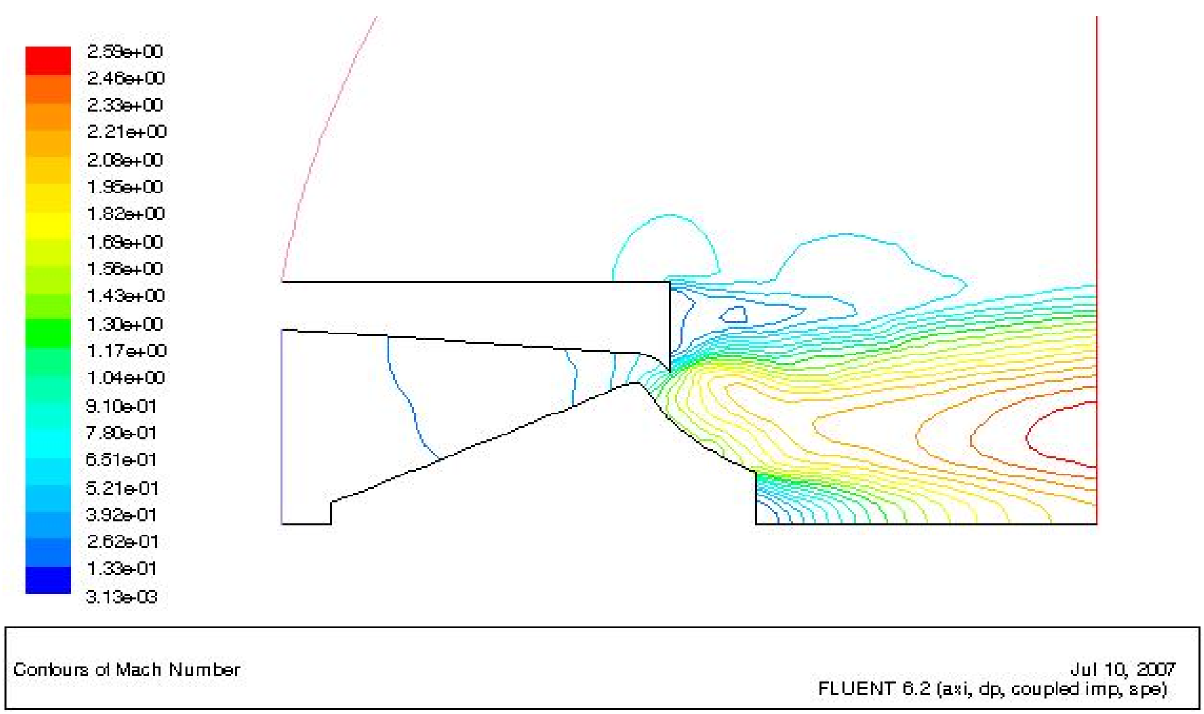


Figure D.2.22: *Mach Plot for Mach 3.0, 20% Length External Aerospike Exhaust Nozzle*

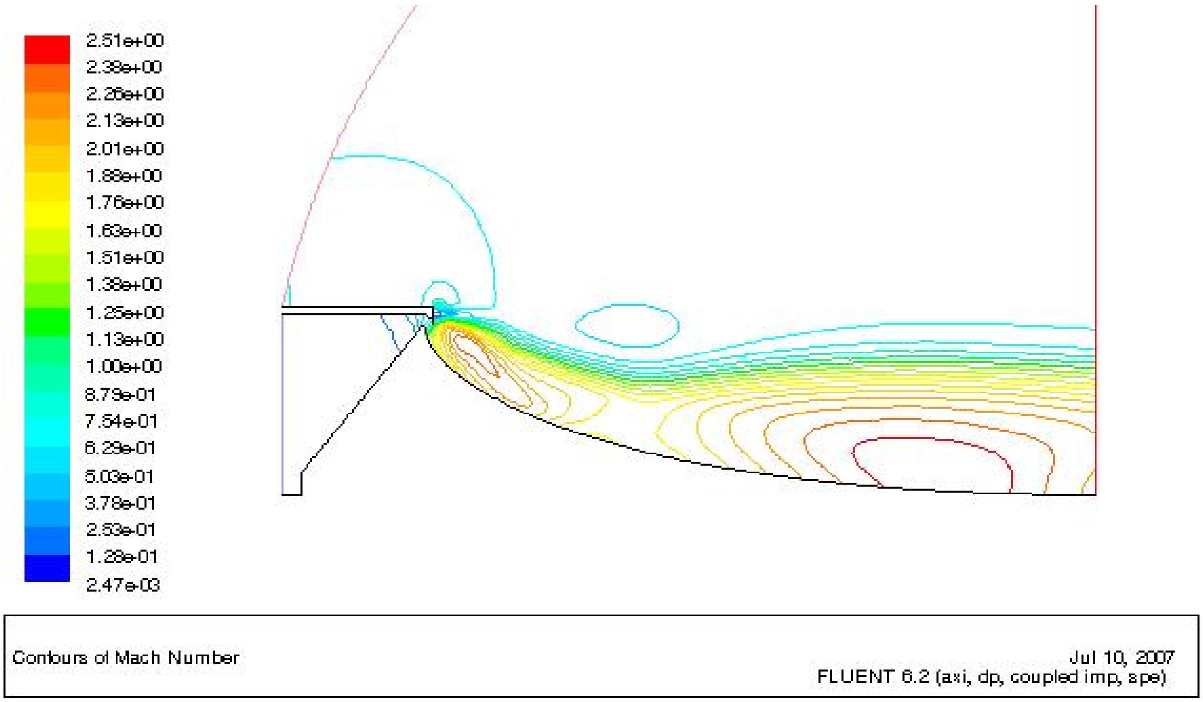


Figure D.2.23: *Mach Plot for Mach 4.0, 100% Length External Aerospike Exhaust Nozzle*

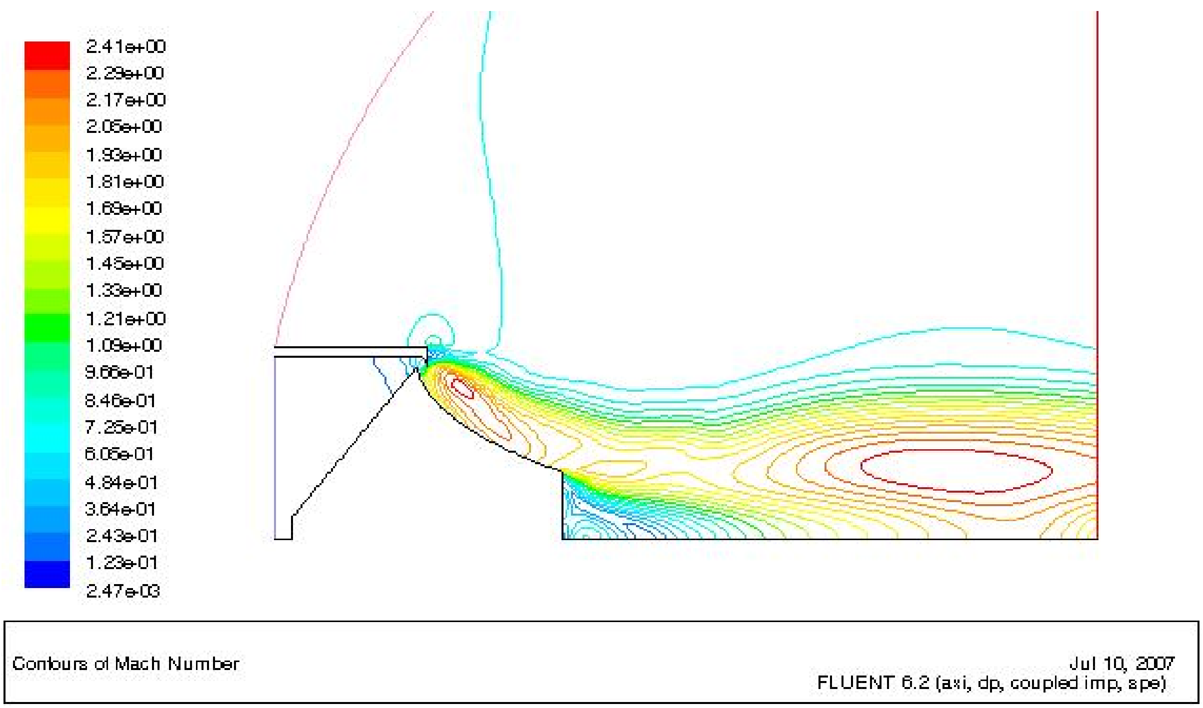


Figure D.2.24: *Mach Plot for Mach 4.0, 20% Length External Aerospike Exhaust Nozzle*

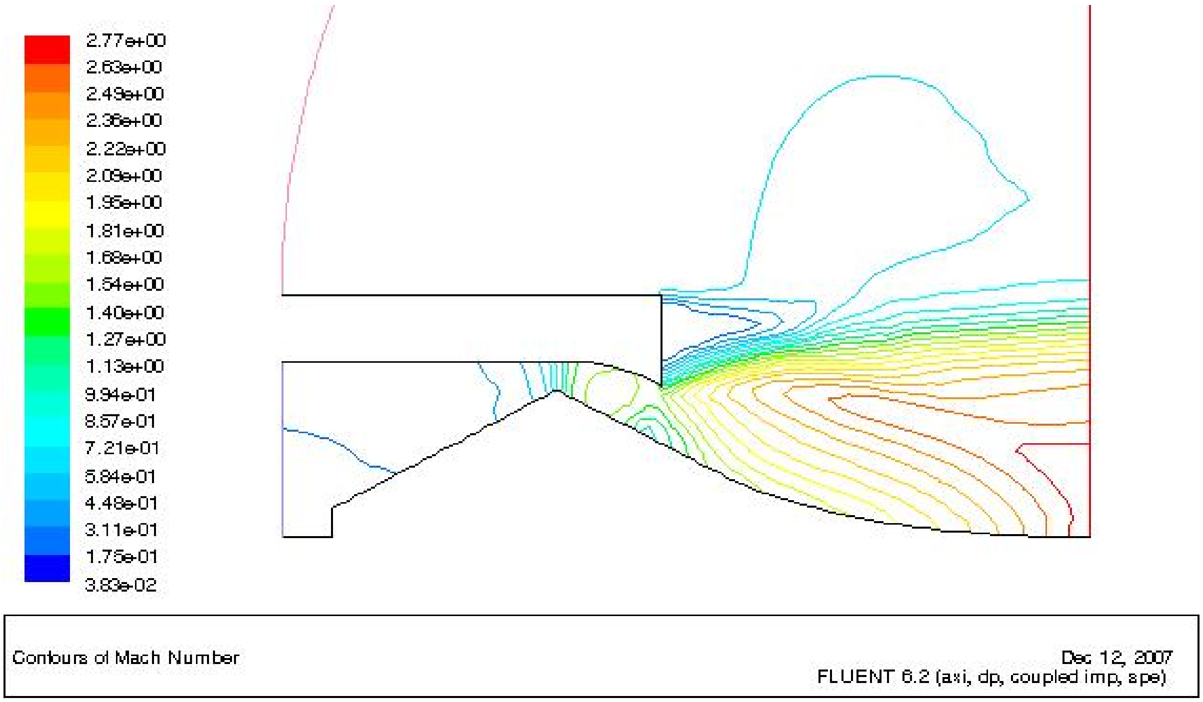


Figure D.2.25: *Mach Plot for Mach 3.0, 100% Length Internal-External Aerospike Exhaust Nozzle*

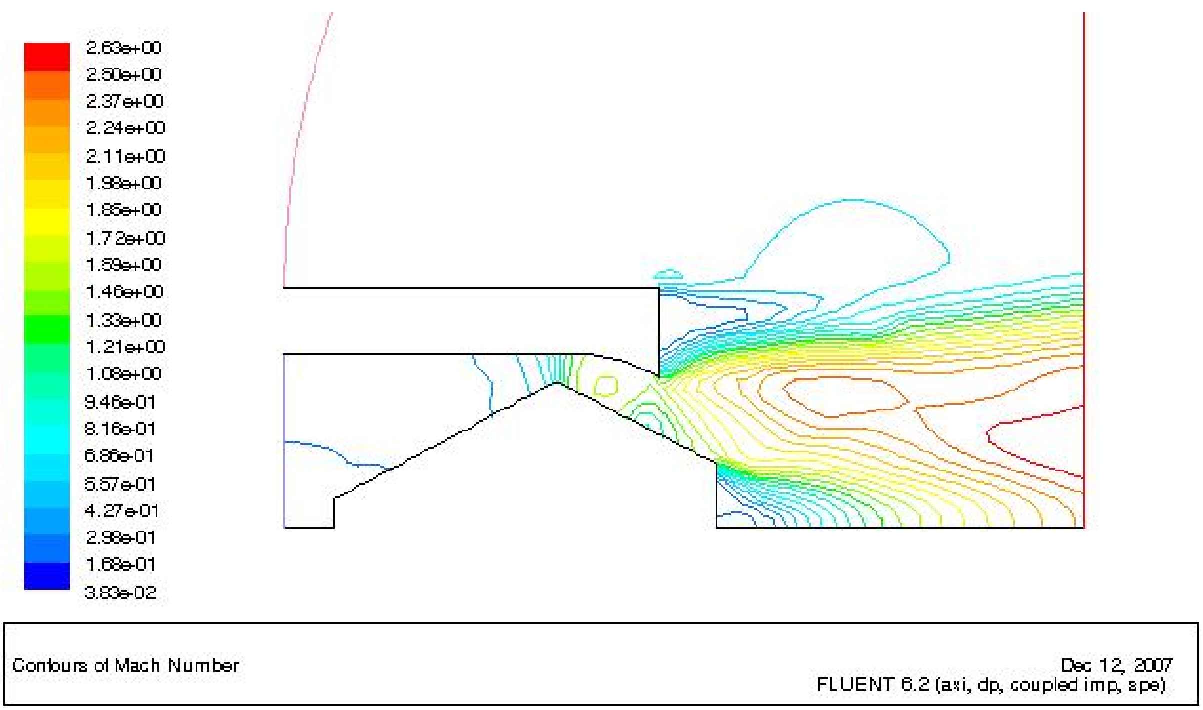


Figure D.2.26: *Mach Plot for Mach 3.0, 20% Length Internal-External Aerospike Exhaust Nozzle*

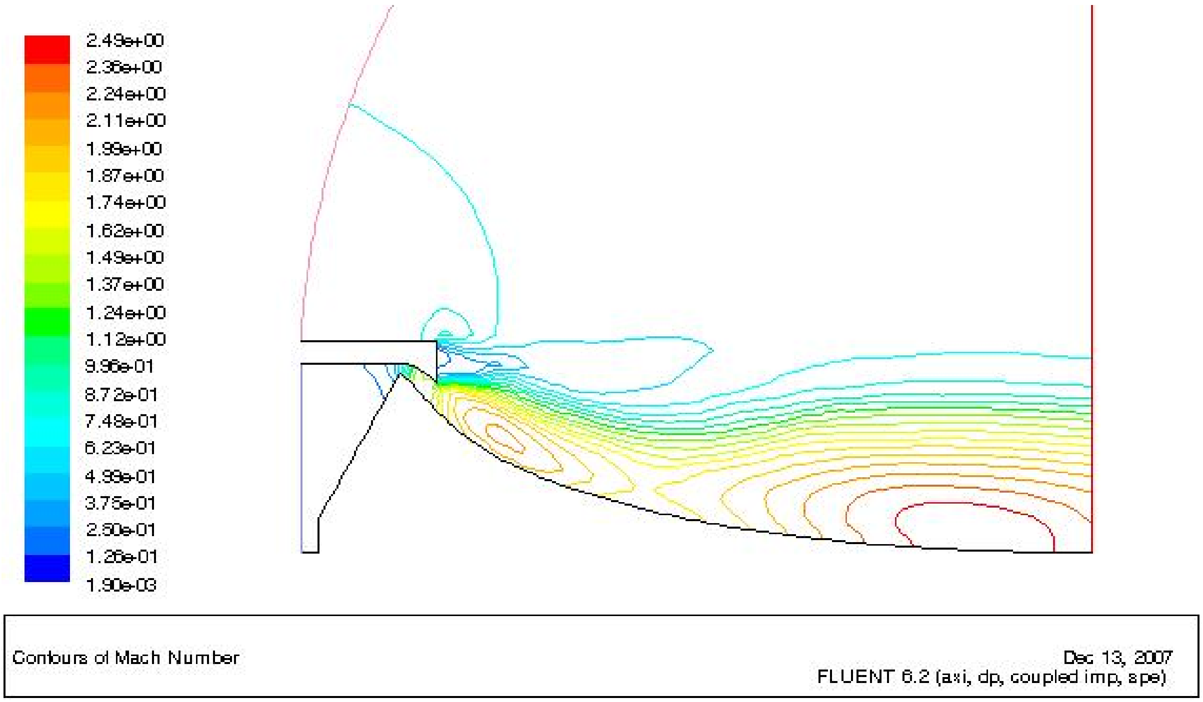


Figure D.2.27: *Mach Plot for Mach 4.0, 100% Length Internal-External Aerospike Exhaust Nozzle*

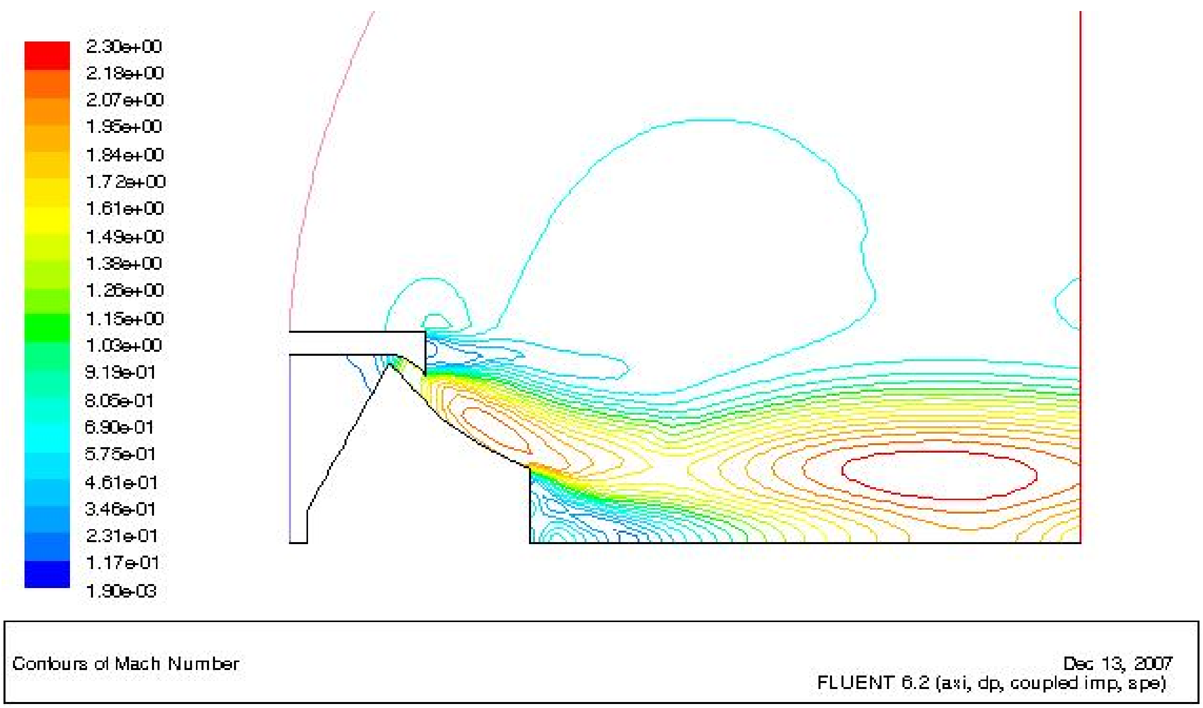


Figure D.2.28: *Mach Plot for Mach 4.0, 20% Length Internal-External Aerospike Exhaust Nozzle*

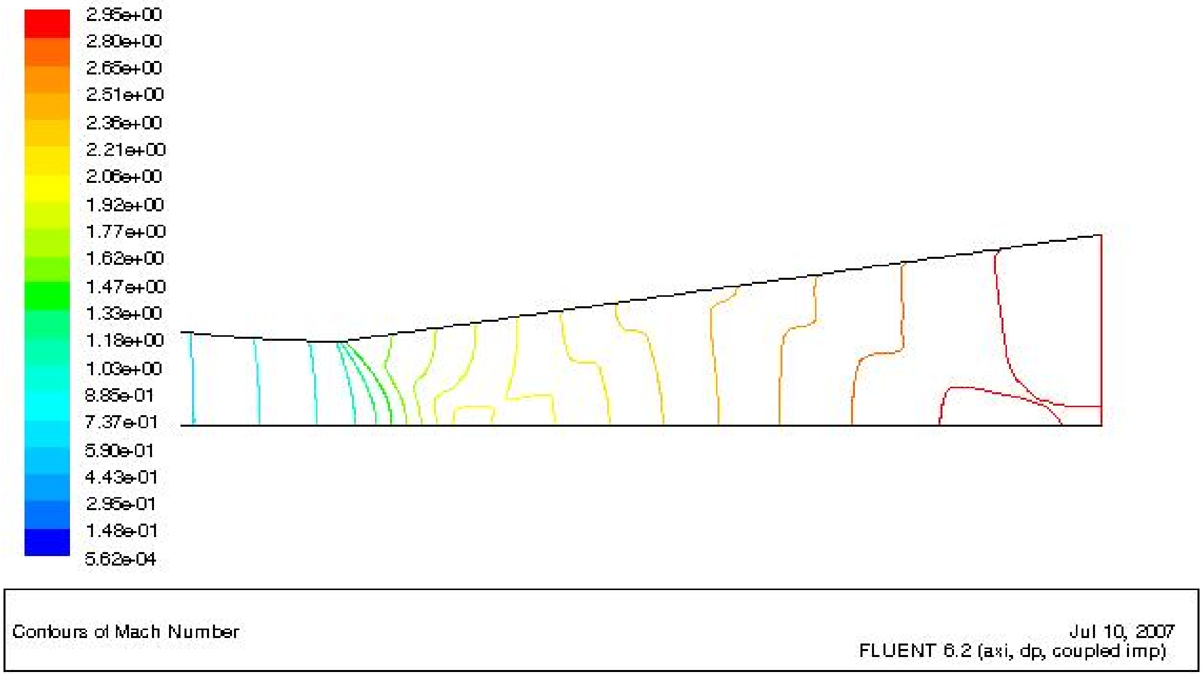


Figure D.2.29: *Mach Plot for Mach 3.0, 8 Degree Conical Exhaust Nozzle*

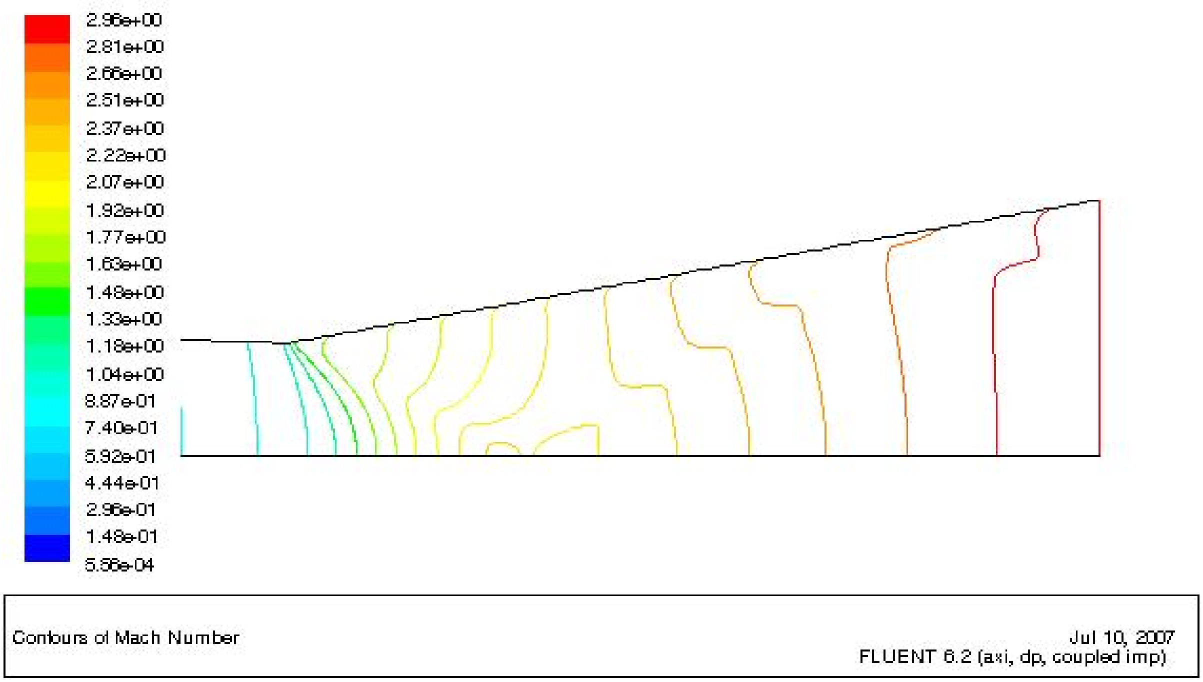


Figure D.2.30: *Mach Plot for Mach 3.0, 10 Degree Conical Exhaust Nozzle*

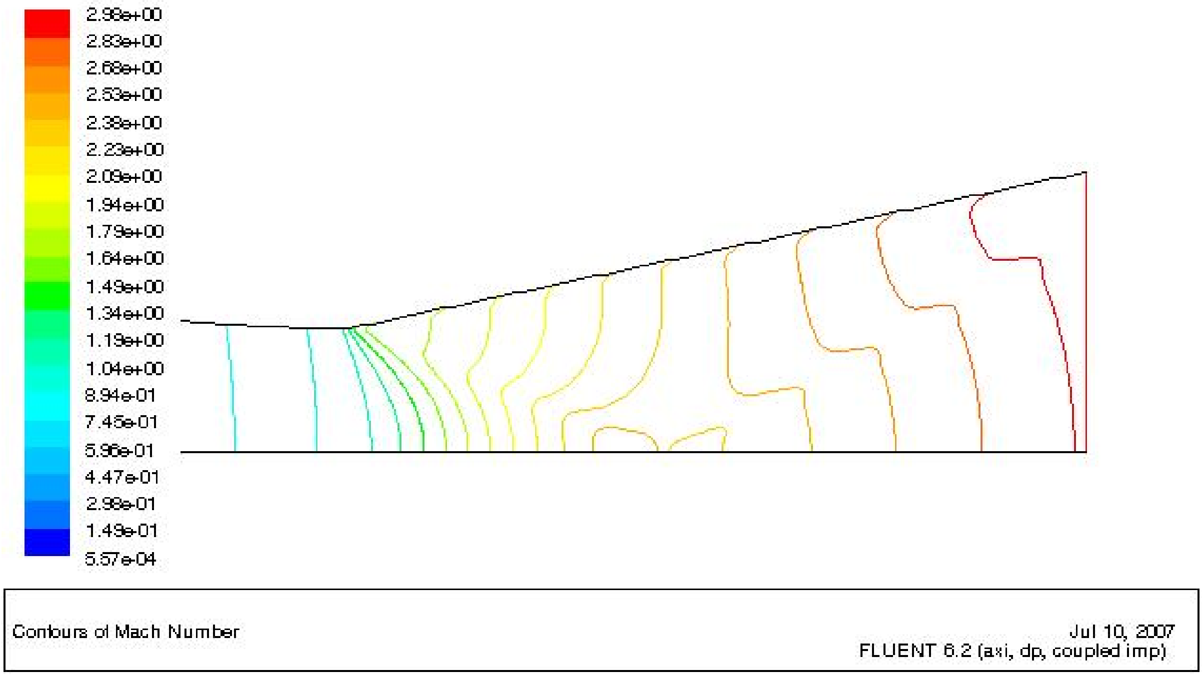


Figure D.2.31: *Mach Plot for Mach 3.0, 12 Degree Conical Exhaust Nozzle*

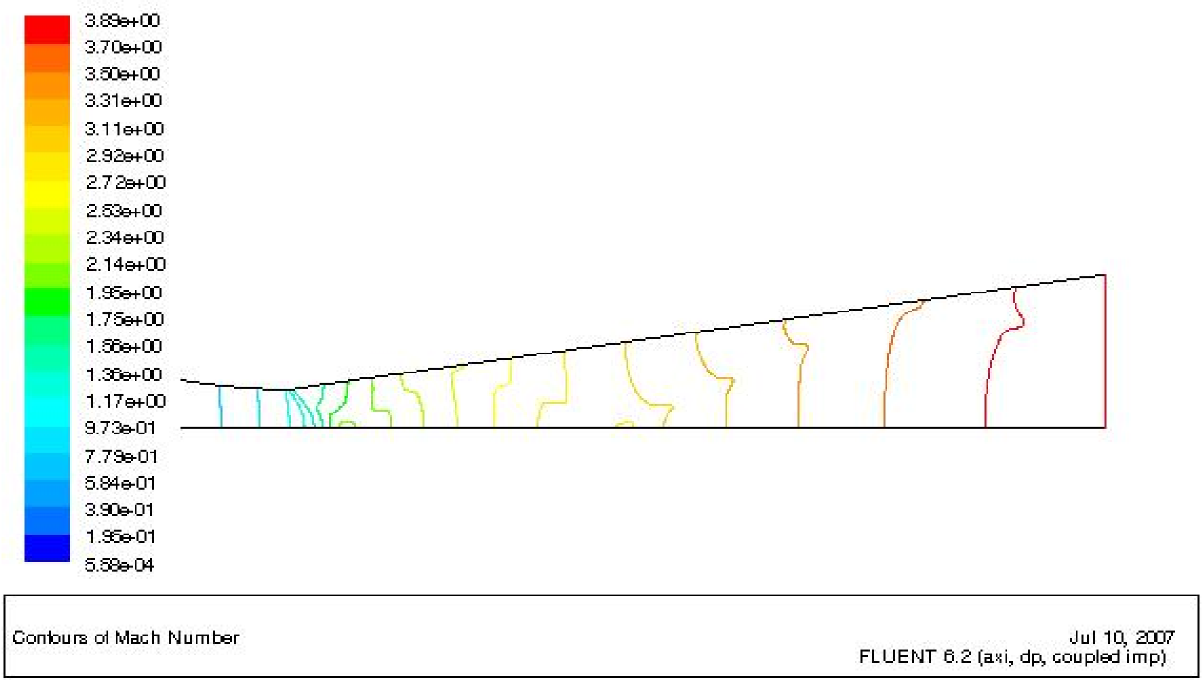


Figure D.2.32: *Mach Plot for Mach 4.0, 8 Degree Conical Exhaust Nozzle*

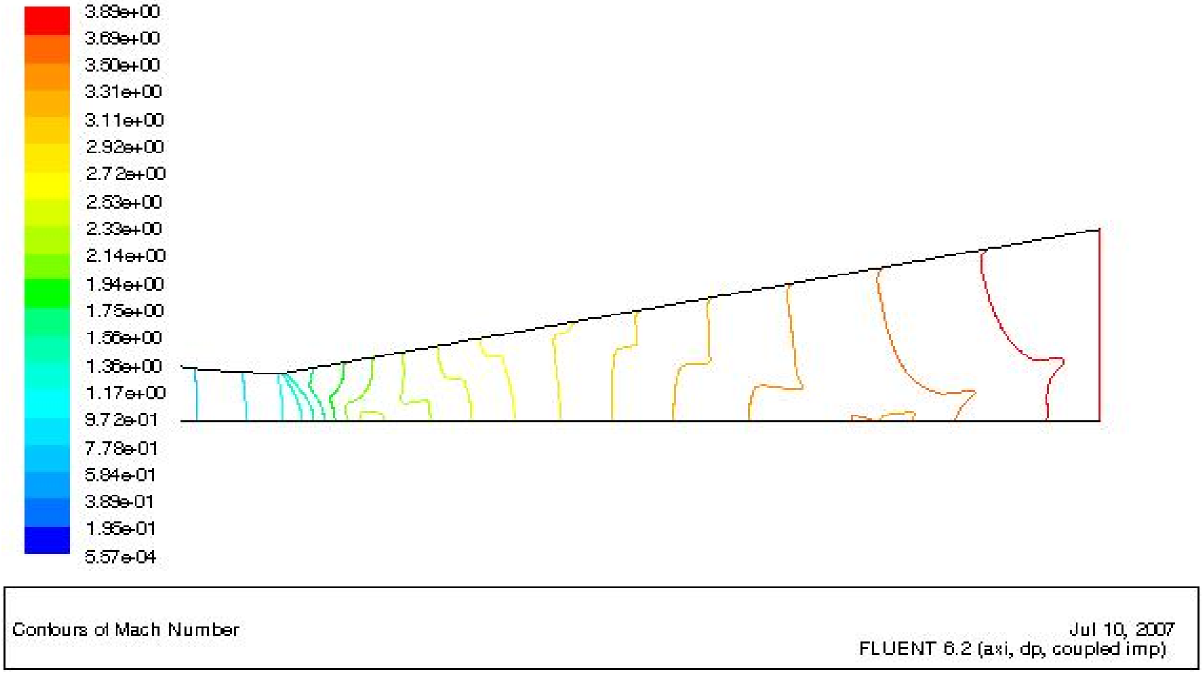


Figure D.2.33: *Mach Plot for Mach 4.0, 10 Degree Conical Exhaust Nozzle*

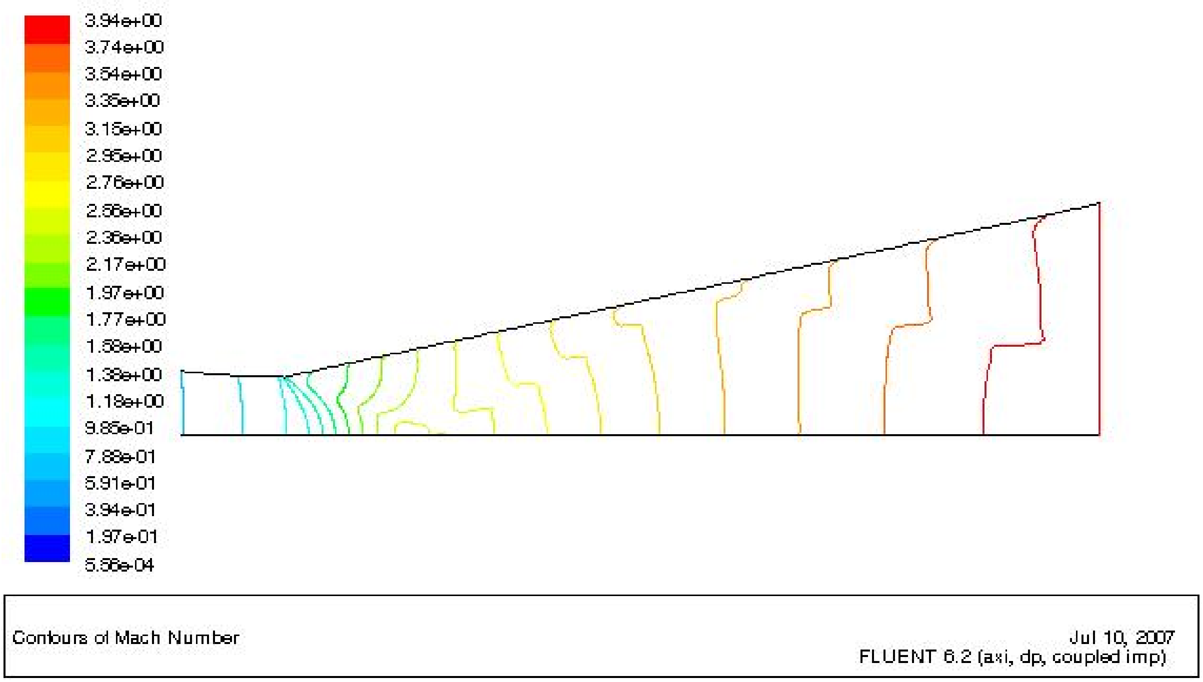


Figure D.2.34: *Mach Plot for Mach 4.0, 12 Degree Conical Exhaust Nozzle*

D.2.5 Exit Mach Number Comparisons

Table D.2.21	**Exit Mach Number Comparisons for Simulated N_2O/HTPB Nozzles for $Ma_{Desired} = 3.0$, $r_{throat} = 0.15$ inches**			
	$Ma_{ComputerCalculated}$	% Error	$Ma_{FluentCalculated}$	% Error
Annular Nozzle $\beta = 0.2 \cdot r_{throat}$	3.0062	0.21%	2.9528	-1.57%
Annular Nozzle $\beta = 0.5 \cdot r_{throat}$	3.0073	0.24%	2.9442	-1.86%
Annular Nozzle $\beta = 1.0 \cdot r_{throat}$	3.0030	0.10%	2.9392	-2.03%
External Aerospike, 100 % Length	3.0000	0.00%	2.3682	-21.06%
External Aerospike, 20% Length	3.0000	0.00%	2.4298	-19.01%
Internal-External Aerospike, 100% Length	3.0003	0.01%	2.5676	-14.41%
Internal-External Aerospike, 20% Length	3.0003	0.01%	2.5152	-16.16%
Conical Nozzle, 8 Degree	-	-	2.9285	-2.38%
Conical Nozzle, 10 Degree	-	-	2.9136	-2.88%
Conical Nozzle, 12 Degree	-	-	2.9082	-3.06%

Table D.2.22	**Exit Mach Number Comparisons for Simulated N_2O/HTPB Nozzles for $Ma_{Desired} = 4.0$, $r_{throat} = 0.15$ inches**			
	$Ma_{ComputerCalculated}$	% Error	$Ma_{FluentCalculated}$	% Error
Annular Nozzle $\beta = 0.2 \cdot r_{throat}$	4.0003	0.01%	-	-
External Aerospike, 100 % Length	4.0000	0.00%	1.3425	-66.44%
External Aerospike, 20% Length	4.0000	0.00%	1.4630	-63.43%
Internal-External Aerospike, 100% Length	4.0014	0.04%	1.4581	-63.55%
Internal-External Aerospike, 20% Length	4.0014	0.04%	1.5063	-62.34%
Conical Nozzle, 8 Degree	-	-	3.8547	-3.63%
Conical Nozzle, 10 Degree	-	-	3.8526	-3.69%
Conical Nozzle, 12 Degree	-	-	3.8493	-3.77%