JOHNS HOPKINS
UNIVERSITY

Applied Physics Laboratory
Laurel MD 20723-6099

RTDC-TPS-481

2D and 3D Method of Characteristic Tools for Complex Nozzle Development Final Report

Tharen Rice

June 30, 2003

Prepared under Grant No. NAG3-2460, "Design of Exhaust Nozzle for the RBCC- GTX Concept," with the NASA Glenn Research Center

THE JOHNS HOPKINS UNIVERSITY • APPLIED PHYSICS LABORATORY
Johns Hopkins Road, Laurel, Maryland 20723-6099

Table of Contents

List of Figures

List of Tables

JOHNS HOPKINS
UNIVERSITY
Applied Physics Laboratory
Laurel MD 20723-6099

RTDC-TPS-481

2D and 3D Method of Characteristic Tools for Complex Nozzle Development
Final Report

Tharen Rice
The Johns Hopkins University
Applied Physics Laboratory
June 2003

1.0 Purpose

This report details the development of a 2D and 3D Method of Characteristic (MOC) tool for the design of complex nozzle geometries. These tools are GUI driven and can be run on most Windows-based platforms. The report provides a user's manual for these tools as well as explains the mathematical algorithms used in the MOC solutions.

2.0 Introduction

Under a successful proposal submission to NRA-99-LeRC-2, APL was awarded a small grant (NAS3-99146) to investigate candidate nozzle designs for the scramjet flowpath of the NASA Glenn Research Center (GRC) developed GTX concept. In this effort, APL developed a nozzle design methodology using streamline tracing techniques. After successful completion of that task (see Reference 1), a follow-on effort (NAG3-2460) was funded to investigate performance and operating characteristics of the GTX rocket nozzle exhaust system. This effort was reported in Reference 2 in May 2001. An extension of that follow-on task was awarded in September 2001 and is reported herein.

To better understand the work done in this effort as well as the previous studies, basic definitions and an explanation of the generalized streamline tracing technique need to be presented. By definition, for a steady flow, a streamline defines the path of particle whose velocity is tangent to the path at all points. Figure 1 shows a graphical explanation of a streamline [Ref. 3], which can be considered to be a generalized function of the spatial dimensions of the problem, f(x,y,z) = 0. If the velocity vector field is denoted by *V* and the local vector tangent to the streamline as *ds*, then the equation defining the streamline path can be written as:

$$\vec{V} \, x \, \vec{ds} = 0 \qquad (1)$$

Furthermore, a boundary defined by multiple streamlines defines a streamtube. Note that conservation of mass requires that the mass flow be constant within the streamtube.

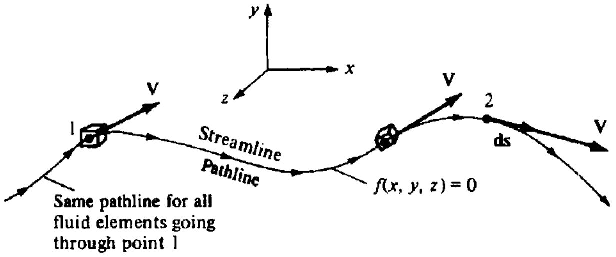


Figure 1. Streamline Definition [Ref. 3]

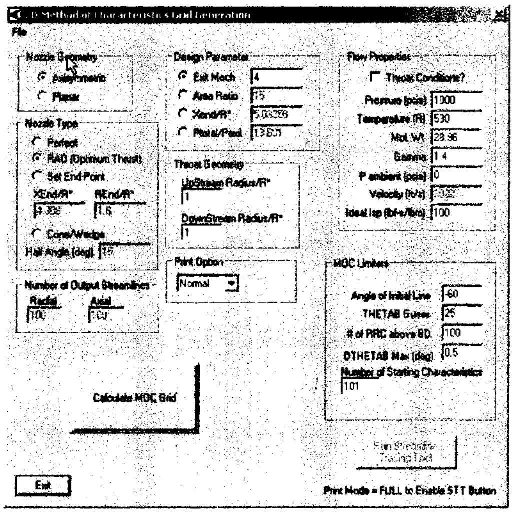
Neglecting viscous effects, a wall can be inserted along any portion of a streamtube without modifying the shape or characteristics of the streamtube itself. This feature serves as the basis for using streamline tracing as a technique for designing three-dimensional objects. By using streamlines from a known flowfield, such as a two-dimensional or conical flow, complex three-dimensional shapes can be traced. Since the streamlines are derived from a known flowfield, all of the flow properties (i.e. pressure, temperature, velocity, etc.) along those streamlines are known for the design operating condition.

It should be noted that the use of a streamline tracing technique assumes that the boundary layer characteristics of the derived three-dimensional design will not significantly alter the inviscid flowfield. Also note that the streamline tracing technique is only used to determine performance at the design condition. Use of high-fidelity modeling techniques is required to determine the performance at off-design operating conditions.

As a general design technique, streamline tracing has been used previously in the development of various vehicle and inlet contours. Reference 4-10 describes streamline tracing applications for waverider vehicles and supersonic inlet design. Before this nozzle work for NASA GRC began, limited work had been done on streamline tracing of nozzle flowfields.

The effort reported herein focuses on the development of a 2D and a 3D MOC tool that can quickly design and analyze nozzle geometries. This effort was an off-shoot of the previous streamline tracing efforts. Through the two previous grants, APL developed several versions a nozzle streamline tracing tool called STT2000. In order for STT2000 to work correctly, a known nozzle flowfield had to be created. The commercial code, TDK, developed by SEA, was used to accomplish this. Reliance on TDK was not desired so the development of an independent nozzle design tool was initiated. This resulted in the 2D MOC tool being developed. Even with the new 2D tool, the streamline tracing techniques applied in STT2000 were still limited by the 2D flowfield assumption. An effort to provide a truly 3D flowfield in which to start the streamline tracing was desired; and therefore, the 3D MOC tool was developed.

3.0 2D Method of Characteristics Tool


2D Method of Characteristics Grid Generation
File
Calculate MOC Grid
Exit
Run Streamline Tracing Tool

Nozzle Geometry
● Axisymmetric
○ Planar

Nozzle Type
○ Perfect
● RAO (Optimum Thrust)
○ Set End Point
XEnd/R* 4.338 REnd/R* 1.6
○ Cone/Wedge
Half Angle (deg) 15

Number of Output Streamlines
Radial 100 Axial 100

Design Parameter
● Exit Mach 4
○ Area Ratio 15
○ Xend/R* 5.03258
○ Ptotal/Pexit 13.637

Throat Geometry
UpStream Radius/R* 1
DownStream Radius/R* 1

Print Option
Normal

Flow Properties
□ Throat Conditions?
Pressure (psia) 1000
Temperature (R) 530
Mol. Wt. 28.96
Gamma 1.4
P ambient (psia) 0
Velocity (ft/s) 1032
Ideal Isp (lbf-s/lbm) 100

MOC Limiters
Angle of Initial Line -60
THETAB Guess 25
# of RRC above BD 100
DTHETAB Max (deg) 0.5
Number of Starting Characteristics 101

Print Mode = FULL to Enable STT Button

Figure 2. 2D MOC Tool GUI

The 2D MOC tool developed under this effort is a GUI driven, Windows based code capable of designing planar and axisymmetric nozzle designs based on a wide range of user inputs. The nozzle designs created are based on a perfect gas, inviscid nozzle solution algorithm. Figure 2 shows the main GUI window. In this main window, there are 8 types of inputs. The following sections describe these inputs and how they are implemented.

3.1 2D MOC Nozzle Algorithm Overview

Before the various inputs are addressed, a general overview of a nominal 2D MOC solution is discussed. A schematic defining several of the nozzle design terms is shown in Figure 3. The 2D MOC algorithm implemented in this tool is based on iterating the initial nozzle expansion angle, θ_B, until a given exit parameter (Mach number, pressure ratio, area ratio, etc.) at point E is reached. The arc (TB) calculated by $R_{DOWN}*\theta_B$ defines the initial expansion region of the nozzle. Downstream of point B, the nozzle contour begins to turn the flow back towards the nozzle centerline.

The MOC solution begins with the creation of an initial data line (TT' in Figure 3). Several methods are available to determine the shape and flow properties along this along and are discussed in Section 3.3. For now, the important aspects to know about this line are: (1) the flow properties are known at every point along the line; and (2) the flow is supersonic at all points. Figure 4 shows an example of the initial data line and the initial expansion region of the nozzle.

Starting with point (1,0), the properties along the left-running characteristic (LRC) are found using a finite differencing method to solve the governing flow equations along this characteristic line. The equation used is called the compatibility equation and is defined as:

$$d(\theta-\mu)=-\frac{1}{\sqrt{M^2-1}+\cot(\theta)}\frac{dr}{r} \tag{2}$$

$$\text{Where: } \mu=\sin^{-1}\left(\frac{1}{M}\right) \tag{3}$$

A derivation of the finite differencing method used can be found in Reference 11.

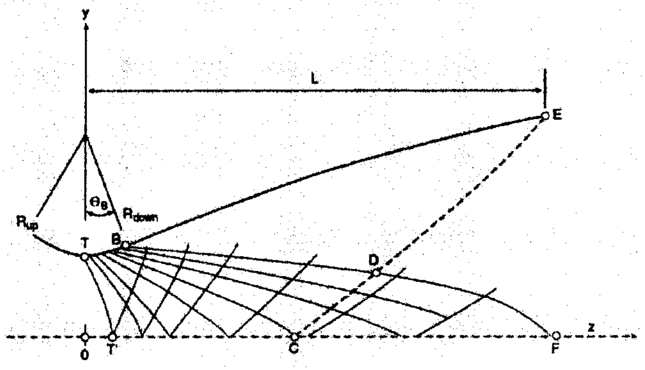


Figure 3. 2D MOC Parameters

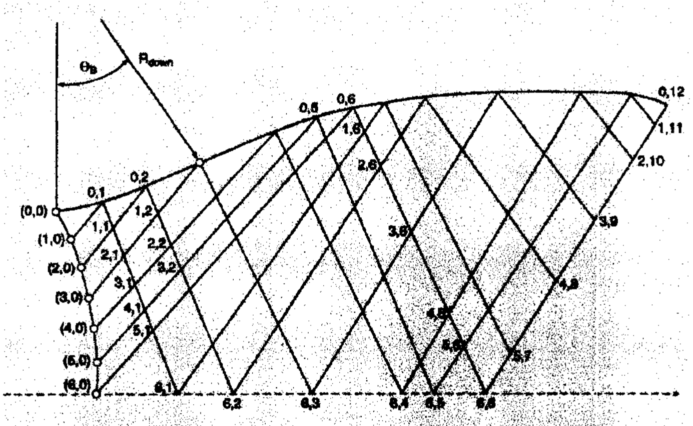


Figure 4. Initial Data Line and Nozzle Expansion Region

Point (0,1) is then found at the intersection of the calculated LRC and the nozzle wall. The nozzle wall is defined as an arc of given radius (R_{DOWN}) with a center located at the z = 0 station. At the wall, the flow angle, θ, is equal to the wall angle.

From point (0,1) a right-running characteristic (RRC) is created based on the following equation.

$$d(\theta+\mu)=\frac{1}{\sqrt{M^2-1}-\cot(\theta)}\frac{dr}{r} \tag{4}$$

Point (1,1) is found at the intersection of the RRC and a newly calculated LRC from point (2,0). The above equations are based on an axisymmetric nozzle type. Solving for a planar nozzle simplifies the equations; however the process remains the same. This process is continued for all characteristic intersections up to Point B. The location of Point B is defined as the last point on the initial expansion region defined by θ_B. Since this is a 2D solution, the nozzle centerline acts as an axis of symmetry. The resultant characteristic mesh (TBFT'), or kernel, is shown in Figure 5. Point F is the centerline point along the RRC starting from B.

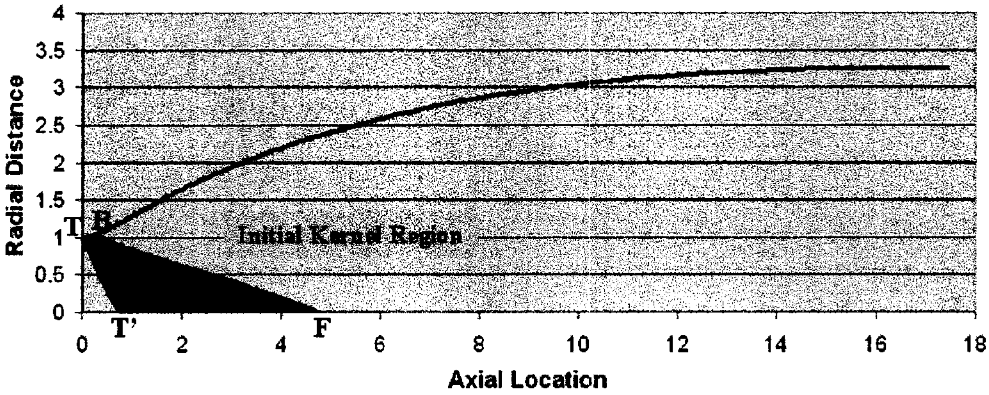


Figure 5. Initial Kernel Region

Once this kernel has been found, a second iteration begins to determine Point D, which lies on the last RRC of the kernel BF (see Figure 6). For a given Point D, the mass flow crossing the BD is found. A LRC is then constructed from D to an unknown Point E where the known mass flow crossing BD is equal to calculated mass flow crossing DE. For a perfect nozzle, the properties along DE are uniform and point E is found rather simply. For other nozzle types, the location of E and properties along DE are found using a Runge Kutta Fehlberg method to integrate dM/dr, and dθ/dr as defined by the compatibility equation (eq. 2) and dx/dr as defined as follows.

$$\frac{dx}{dr}=\frac{1}{\tan(\theta+\mu)} \tag{5}$$

Once Point E has been found, it is set as the nozzle wall exit point. The properties at point E are compared with the given design constraints (nozzle type and design parameter). If the values match, then a streamline from B to E is calculated and the nozzle contour has been found. This streamline is determined by back calculating RRCs from DE to a region upstream of DE. Figure 6 shows this graphically.

If the properties at E do not match, then D and ultimately θ_B is iterated on until they do match. There exists only one combination of θ_B and D that satisfies all of the equations and constraints.

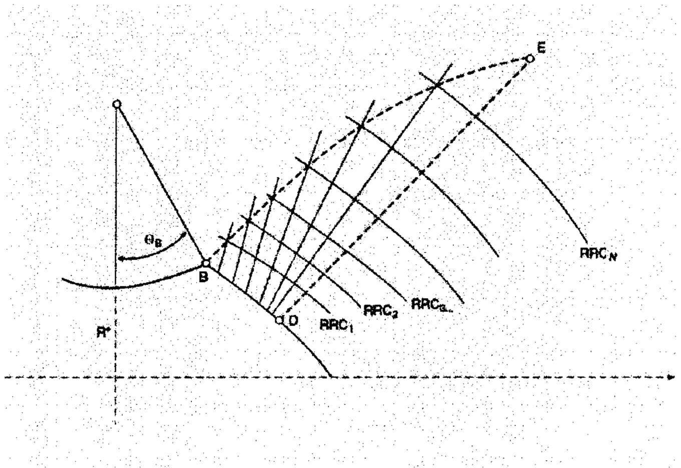


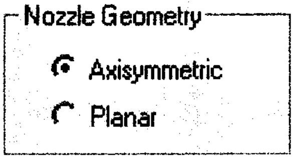
Figure 6. Point D Solution

3.2 2D MOC Tool Inputs

Before describing the inputs in detail, a short discussion on dimensions should be presented. As soon will be shown, all of the nozzle geometry inputs (length, area, etc.) have been non-dimensionalized. All lengths are non-dimensionalized by the throat half-height (distance from centerline to wall) for planar nozzles and the throat radius for axisymmetric nozzles. This is continued in the tool's output files. The code calculates mass flow, thrust and other dimensional parameters by assuming the following. For an axisymmetric nozzle, a throat radius of 1" is used. For a planar nozzle solution, a throat half-height of 1" is used as well as a reference width of 12". The performance numbers obtained for a planar nozzle are only for one-half of the total nozzle. This is applicable to SERN nozzle performance calculations.

3.2.1 Nozzle Geometry Input

There are two types of nozzle geometries to choose: axisymmetric and planar (see Figure 7 for input graphic). Figure 8 shows a typical axisymmetric nozzle and Figure 9 shows a typical planar nozzle.



Nozzle Geometry

● Axisymmetric

○ Planar

Figure 7. Nozzle Geometry Input

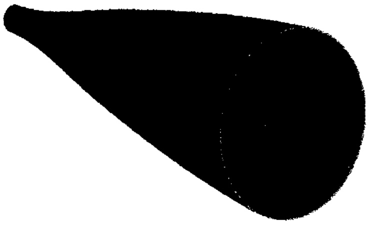


Figure 8. Axisymmetric Nozzle

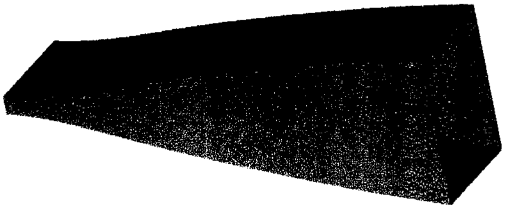
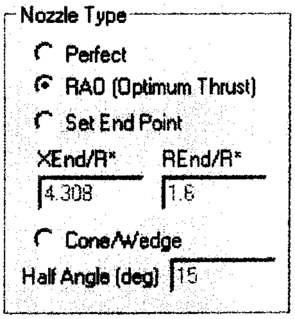


Figure 9. Planar Nozzle

3.2.2 Nozzle Type Input

There are four types of nozzles that this 2D MOC tool can solve. The input for selecting the type of nozzle is shown in Figure 10.



Nozzle Type

○ Perfect

● RAO (Optimum Thrust)

○ Set End Point

XEnd/R* 4.308

REnd/R* 1.6

○ Cone/Wedge

Half Angle (deg) 15

Figure 10. Nozzle Type Input

3.2.2.1 Perfect Nozzle

The first type is a *perfect* nozzle. For a perfect nozzle, the exit plane is required to be uniform, meaning that all exit values (Mach, pressure, temperature, etc.) are constant at every radial station. Also, the flow angle at the nozzle exit plane is 0.0 (fully axial flow). This type of nozzle is generally used in wind tunnel facilities where uniform flow is desired for testing and length is not constrained.

3.2.2.2 Rao Nozzle

The second nozzle type is a Rao nozzle. This nozzle is derived from work done by G.V.R. Rao in the late 1950's. In this work, a mathematical analysis of optimum nozzle contours for a given exit condition (Mach, area ratio, etc.) was developed[12]. There are several subtleties as to the optimum contours with respect to the different exit conditions; however, in general, a Rao nozzle provides a minimum length nozzle for a given exit condition such that any change in the exit condition at the given length will produce lower performance. For example, assume that a Rao solution has resulted in nozzle area ratio of 4 and the optimum nozzle length is 10". This means that for a fixed nozzle of length of 10", a nozzle designed with an area ratio of 3 or 5, will produce less thrust then the original area ratio = 4 nozzle. Again, there are subtleties to the Rao method; however, in general the above statements hold true.

The basis of the Rao method revolves around finding the nozzle contour where the following nozzle wall exit condition (point E) holds true (for a zero-backpressure solution).

$$\sin(2\theta_E) = \frac{2\cot(\mu_E)}{\gamma M_E^2} \tag{6}$$

3.2.2.3 Set End Point Nozzles

The third nozzle type forces the contour to go through an explicitly set nozzle exit wall point (*Set End Point* option). This type of nozzle is bounded in length by the Rao and perfect nozzle solutions. For a given area ratio nozzle, the Rao nozzle will solve for the minimum length nozzle and the perfect nozzle solution will solve for the maximum length nozzle. The Set End point option can be used to find all of the contours in between these two lengths. No solution exists if the chosen length is less than the Rao nozzle length or greater than the perfect nozzle length. The *Set End Point* solution is not required to have a uniform exit flow field as in a perfect nozzle, nor does it have to meet the Rao nozzle exit criteria.

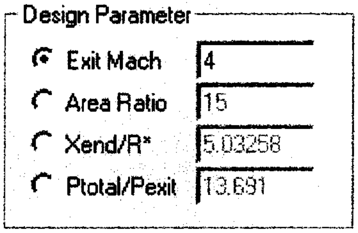
3.2.2.4 Cone/Wedge Nozzles

The fourth type of nozzle is a cone/wedge contour of a given half-angle. The cone is solved if the axisymmetric option is chosen. The wedge is solved if the planar option is chosen.

3.2.3 Nozzle Design Parameters

The design parameter section sets the required condition at the nozzle wall exit. A screen capture of this input is shown in Figure 11. This parameter can be set to the following types.

- □ Mach number
- □ Area ratio: Ratio of the exit-to-throat areas
- □ Pressure ratio: Ratio of nozzle total pressure-to-exit static pressure
- □ Length ratio: Ratio of nozzle length-to-throat radius



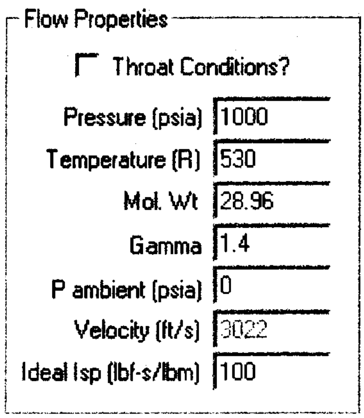
Design Parameter

● Exit Mach 4
○ Area Ratio 15
○ Xend/R* 5.03258
○ Ptotal/Pexit 13.691

Figure 11. Design Parameter Input

3.2.4 Flow Properties

These inputs define the initial flow properties for the nozzle solution (see Figure 12). For a given set of flow properties, the user has two choices as to what these properties represent. The default is that the flow properties are the total conditions. Given these total conditions, the tool solves for an initial data line near the throat in which to begin the solution. If the *Throat Conditions* box is checked, then these properties will be used as the initial data line flow properties. Also, choosing the throat conditions enables the *Velocity* input where the user is required to input a flow velocity. The flow has to be greater than Mach = 1 for a nozzle solution to be calculated.



Flow Properties

□ Throat Conditions?

Pressure (psia) 1000

Temperature (R) 530

Mol. Wt 28.96

Gamma 1.4

P ambient (psia) 0

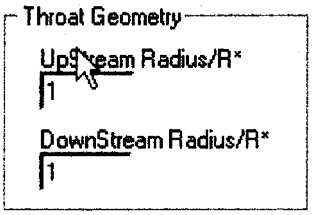
Velocity (ft/s) 3022

Ideal Isp (lbf-s/lbm) 100

Figure 12. Flow Properties Input

3.2.5 Throat Geometry Input

This input (shown in Figure 13) defines the throat geometry as two arcs of given radius as depicted in Figure 3. Note that these input are non-dimensional and are normalized by the throat radius (R*). The *UpStream Radius* input helps to define the initial data line given the total flow conditions. Section 3.3 describes the creation of the initial data line in detail. The *DownStream Radius* is used to define the initial nozzle expansion region.



Throat Geometry

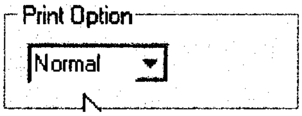
UpStream Radius/R* 1

DownStream Radius/R* 1

Figure 13. Throat Geometry Input

3.2.6 Print Option Input

The code is capable of outputting various parameters in various ways. The user is allowed to choose the level of output desired for each run. The choices are *Normal* and *Full* (see Figure 14). The *Full* option outputs all of the data available. The *Normal* option outputs a subset of the *Full* option. Table 1 shows a listing of the various outputs for each option.



Print Option

Normal

Figure 14. Print Option Input

Table 1. Print Option Output Files

Data File Name	Description	Normal	Full
Summary.plt	TECPLOT formatted file showing the nozzle contour, the last LRC, and the RRC (BF)	●	●
Summary.out	Summary file describing all of the nozzle details	●	●
rao.dat	File containing the nozzle contour that can be used with the RAO option in TDK.	●	●
TT'.out	Initial data plane properties	●	●
ThetaB.out	θ_B iteration parameters	●	●
MOC_Grid.plt	TECPLOT formatted file showing all of the RRCs for the nozzle. This file can be used to look at contour plots through the nozzle.		●
MOC_SL.plt	TECPLOT formatted file showing streamlines for the nozzle. This file can be used to look at contour plots through the nozzle.		●
center.out	Contains centerline flow data.		●
wall.out	Contains nozzle wall flow data and contour.		●
TT'BF_Kernel.out	Contains a matrix of data for all of the RRCs in region TT'BF, See Figure XX		●
BFE_Kernel.out	Contains a matrix of data for all of the RRCs in the region BFE, See Figure XX		●
wall_i.out	Data for the initial wall expansion		●
axis_i.out	Data for the centerline up to point F		●
LastKernel.out	Data for the last RRC (BF) defined by θ_B		●
Uncropped Kernel.out	Data for the entire MOC grid generated extending beyond the nozzle exit plane		●

3.2.6.1 Summary.out File

The primary output file for the code is the 'summary.out' file. This file contains MOC grid data, performance values and other nozzle parameters. This section gives a brief description of the items in this file.

The first part of the file contains a general definition overview followed by the input parameters that were used in the nozzle design. The next section contains the flow properties along the initial data line. The massflow data is an integration of the massflow from the nozzle centerline to the nozzle wall.

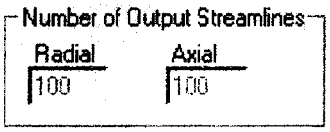
This list is followed by a summary of the performance parameters. These parameters are separated into two parts. The first part reports the 2D performance (massflow, thrust, etc.); the second part reports the 1D performance. For 2D calculations, performance is taken from the integration of individual flow points, where as the 1D calculations are based on a 1D isentropic process from the nozzle total conditions to a uniform Mach 1 throat.

The next section contains information on the initial expansion region, including the converged initial expansion angle and the difference in the integrated massflow along the last RRC at the initial expansion region (BDF in Fig. 3) and the massflow at the initial data line. This is a good check to make sure that the initial grid and resultant solution is progressing smoothly. The code checks to make sure the percent difference in these values is less than 2%. If it is not, the code will notify the user and the solution will terminate. Future versions of this tool may make this tolerance a user input.

The next two sections contain the flow properties along the LRC (DE) and the entire nozzle wall respectively. These sections are followed by a list of the flow properties at the nozzle exit plane, as well as nozzle geometry parameters (surface area, area ratio, etc.) and exit plane performance.

3.2.7 Number of Output Streamlines Input

This input sets the number of streamlines and points that will be printed in the MOC_SL.plt file (see Figure 15). The *Radial* input defines the total number of streamlines. The *Axial* input defines the number of points along each streamline.

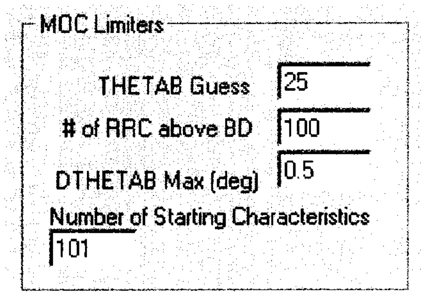


Number of Output Streamlines
Radial: 100
Axial: 100

Figure 15. Number of Output Streamline Input

3.2.8 MOC Limiters Input

These inputs affect the initial MOC grid as well as the efficiency of the code (see Figure 16). The *THETAB Guess* (θ_B) input gives the tool a starting point for the θ_B iteration. In general, longer nozzles (i.e. perfect) have lower θ_B values than short nozzles (Rao). A nominal value of 25° is supplied and works for most cases.



MOC Limiters

THETAB Guess 25

# of RRC above BD 100

DTHETAB Max (deg) 0.5

Number of Starting Characteristics 101

Figure 16. MOC Limiters Input

The *# of RRC above BD* inputs refers to the number of Right Running Characteristics the tool will solve for after the nozzle end point is found. As described in Section 3.1, once the nozzle end point (E) is found, the nozzle contour from B to E needs to be determined. This contour is defined by a streamline that is solved for by completing the MOC grid in the BDE region (See Figure 17). The density of the grid in region BDE is defined by the number of RRCs.

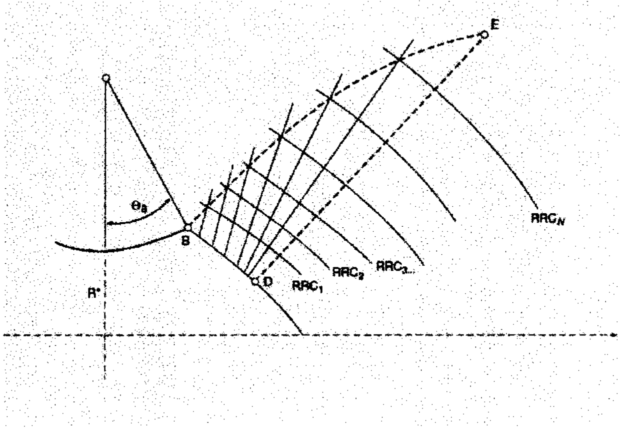


Figure 17. # of RRC Above BD Schematic

The *Number of Starting Characteristics* input determines the number points to be found on the initial data line.

The *DTHETAB* ($\Delta\theta_B$) *Max* input is a parameter that helps establish the RRC density along the initial nozzle expansion arc (TB). For a given number of starting characteristics, the MOC solution begins. As a LRC reaches the nozzle wall it is reflected toward the centerline. DTHETAB refers to the maximum difference in θ_B between any two nozzle wall points. If the difference is greater than the input DTHETAB, a new RRC is created at that point. An example is shown in Figure 18. The angle between the initial wall point and Point (0,1) is greater than the defined $(\Delta\theta_B)_{MAX}$. A new RRC (defined by the red dashed line) is created at an angle $(\Delta\theta_B)_{MAX}$ from the initial wall point. This process allows for a better definition of the flow throughout the entire nozzle. The smaller the value of DTHETAB, the more refined the grid may be, at the cost of run time. Nominal values are between 0.25° and 0.5°.

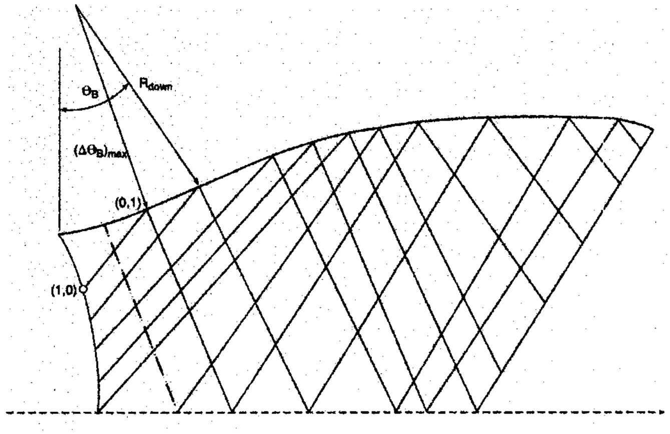


Figure 18. DTHETAB Schematic

3.2.9 Run Streamline Tracing Tool Button

When this button (Figure 19) is pressed the streamline tracing tool STT2000 can be executed. The files needed to run STT2000 are only created when the *Full* print option is chosen; therefore this button is only enabled at that time.

Figure 19. Run Streamling Tracing Tool Button

3.2.10 Calculate MOC Grid Button

When this button is pressed (Figure 20), the solution cycle is started using the inputs displayed in their respective boxes. When the cycle is complete a graph of the nozzle contour appears in a new window. This window has to be closed in order for the tool to output the required files. Figure 21 shows a sample of the chart geometry window.

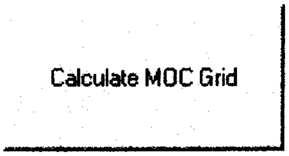


Figure 20. Calculate MOC Button

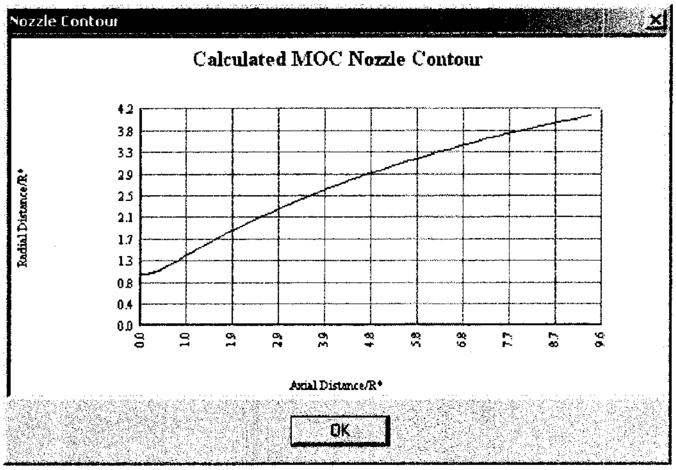


Figure 21. Nozzle Contour Window

3.3 Initial Data Line Definition

The initial data line (TT') is calculated using a modified Hall method developed by Kliegel and Levine for transonic flow around nozzle throat regions[13]. The method uses a toroid coordinate system to develop an analytical solution for flow velocity (axial and transverse) at all points in the throat region of a nozzle.

The true art in the initial data line definition is determining the shape of the line where the transonic regions. A line that is too shallow or too steep can ultimately result in a failure in the solution convergence. For this tool, the calculation starts at the nozzle wall. The position of all subsequent points is determined by constructing a RRC from the proceeding point. If the Mach number at a given point is greater than 1.5, the line shape is changed so that Mach = 1.5 is never exceeded.

The *Upstream Radius* parameter has a pronounced effect on the initial data line. As the upstream radius increases, the calculated flow velocity increases by nearly $1/(R_{up}+1)$. Figure 22 shows initial data lines for two upstream radii without the Mach 1.5 constraint. As the upstream radius decreases, the calculated Mach number increases. This causes the constructed RRC to move downstream because the calculated flow angles (μ) are shallower. In some cases, this causes difficulties in the nozzle solutions. The third line in Figure 22 shows the solution with the Mach 1.5 constraint, resulting in a *well behaved* data line. The Mach constraint as well as the line shape is arbitrary. This tool uses this particular method because is has shown to work for many nozzle designs.

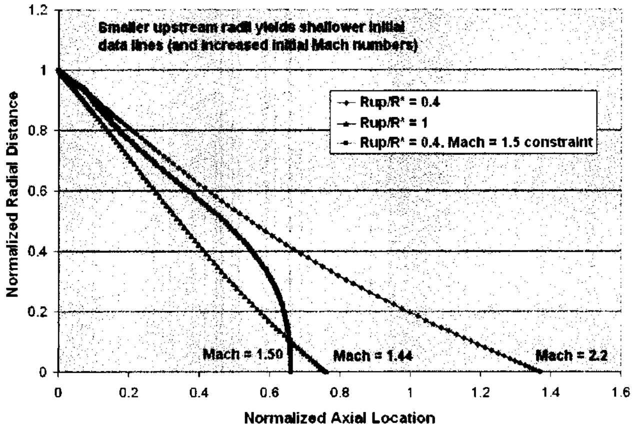


Figure 22. Initial Data Lines

3.4 Tricks of the Trade

This section will try to explain techniques that should be used to arrive at a converged nozzle solution. The primary cause of a poor nozzle solution starts with the development of a poor initial MOC grid and the parameters that create it. These parameters are:

- ❑ Downstream Radius
- ❑ Number of Starting Characteristics
- ❑ Upstream Radius
- ❑ $(\Delta\theta_B)_{MAX}$

Understanding the interaction between these parameters should provide a good guide to good nozzle development. The following discussion provides a summary explanation of these parameters. It is recommended that each user conduct a short study of these parameters by using the nozzle tools and investigate their effects.

A good nozzle solution starts with a well defined grid in the initial nozzle expansion region (from initial data line to θ_B). The term 'well defined' is subjective; however, the default values for the four aforementioned parameters seem to work for most cases. Figure 23 shows the initial expansion region around the nozzle wall using these defaults parameters for a perfect, Mach = 4 nozzle.

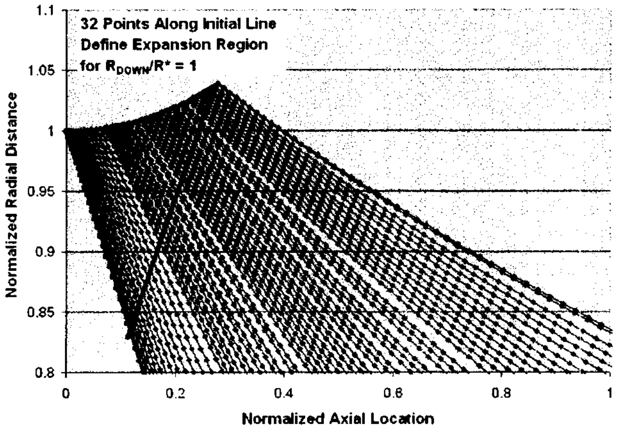


Figure 23. Initial Kernel Region with Nominal Design Parameters

As is shown, 32 points (out of the initial 101 points) along the initial data line are used to determine the initial nozzle expansion region. This seems to give the MOC grid a solid start to the calculation. In general, more points are better, however if there are too many points, the RRC's can run too close together (and sometimes even cross) causing solution difficulties. Again, there is no set number that is better or worse for a given design.

As the four parameters deviate from these defaults, the initial kernel region changes. For example, Figure 24 shows this region when the downstream radius is decreased to 0.2. As is shown, the number of points that now define the initial expansion region is reduced from 32 to 14. In CFD related terms, this new mesh is courser (by a factor of two). As the grid gets courser, the code has a more difficult time converging on a solution. The MOC algorithm makes calculations at the intersection of the LRCs and RRCs and at the nozzle wall. As these intersections get further apart (courser mesh), the solution produces bigger errors. Errors in this region of the nozzle get amplified as the solution progresses downstream. The downstream radius input seems to have the greatest effect on whether or not the nozzle solution will converge. Keeping this parameter around the nominal 1.0 value is recommended.

Decreasing the downstream nozzle radius is not the only way for the mesh to become courser. Changing the number of starting characteristics is a straightforward way to affect the mesh. Changing the upstream radius as well as changing the $(\Delta\theta_B)_{MAX}$ parameter also affects the mesh, and is explained in Section 3.2.8.

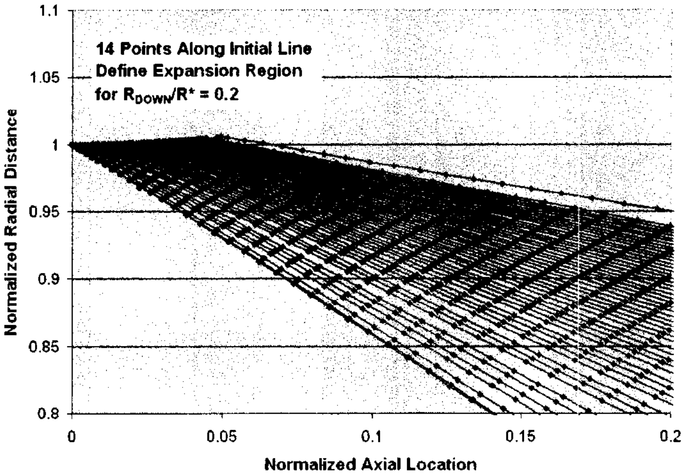
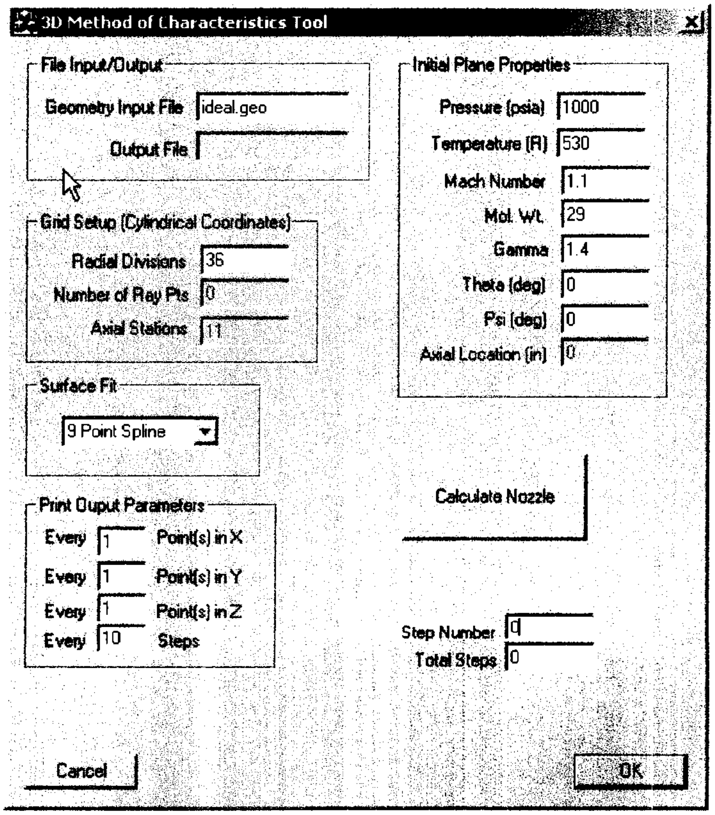


Figure 24. Initial Expansion Region for Downstream Radius of 0.2

4.0 3D Method of Characteristic Tool

The 3D MOC tool developed under this effort allows a user to determine three-dimensional nozzle flowfield properties given initial throat conditions and a nozzle contour. The solution algorithm solves the 3D MOC equations using a reference plane method where the nozzle is parsed into numerous axial stations. For a given axial station, flow properties are determined using the flow information of the preceding 'reference' axial plane. The first axial station is determined by user defined throat properties.

The tool was developed to be GUI driven, just as the 2D MOC tool. Figure 25 show the main GUI Window. With in this main window, there are 5 types of inputs. The following sections describe these inputs and how they are implemented.


3D Method of Characteristics Tool

File Input/Output
Geometry Input File: ideal.geo
Output File:

Grid Setup (Cylindrical Coordinates)
Radial Divisions: 36
Number of Ray Pts: 0
Axial Stations: 11

Surface Fit: 9 Point Spline

Print Ouput Parameters
Every 1 Point(s) in X
Every 1 Point(s) in Y
Every 1 Point(s) in Z
Every 10 Steps

Initial Plane Properties
Pressure (psia): 1000
Temperature (R): 530
Mach Number: 1.1
Mol. Wt.: 29
Gamma: 1.4
Theta (deg): 0
Psi (deg): 0
Axial Location (in): 0

Step Number: 0
Total Steps: 0

Figure 25. 3D MOC Tool Main GUI

4.1 3D MOC Nozzle Algorithm Overview

4.1.1 Mathematical Equations

As stated above, the 3D MOC tool uses a reference plane method to solve for the 3D MOC flowfield equations (compatibility equations). A complete explanation of this method can be found in Reference 14. A majority of the explanation is echoed below.

The derivation of the compatibility equations begins with the equations of motion for steady, inviscid ideal flow. For these equations, two sets of characteristic surfaces can be derived. These surfaces are defined as follows.

$$(uf_x + vf_y + wf_z)^2 = 0 \tag{7}$$

$$(ug_x + vg_y + wg_z)^2 - a^2(g_x^2 + g_y^2 + g_z^2) = 0 \tag{8}$$

Where f(x,y,z) = 0 defines a surface composed of streamlines and g(x,y,z) = 0 defines a Mach conoid. Along the conoid surface , a ray, also referred to as a bicharacteristic, is defined by the following equations.

$$dx = (\cos\beta\sin\theta + \sin\beta\cos\theta\cos\delta)dL \tag{9}$$

$$dy = (\cos\beta\cos\theta\sin\psi - \sin\beta(\sin\theta\sin\psi\cos\delta - \cos\psi\sin\delta))dL \tag{10}$$

$$dz = (\cos\beta\cos\theta\cos\psi - \sin\beta(\sin\theta\cos\psi\cos\delta + \sin\psi\sin\delta))dL \tag{11}$$

Where θ and ψ are related to the velocity vector (q) by:

$$u = q\sin\theta \tag{12}$$

$$v = q\cos\theta\sin\psi \tag{13}$$

$$w = q\cos\theta\cos\psi \tag{14}$$

The parametric angle, δ, lies in the plane normal to q and is measured from the plane containing q and x. Figure 26 shows the nozzle coordinate system used. Figure 27 shows a graphical representation of these parameters.

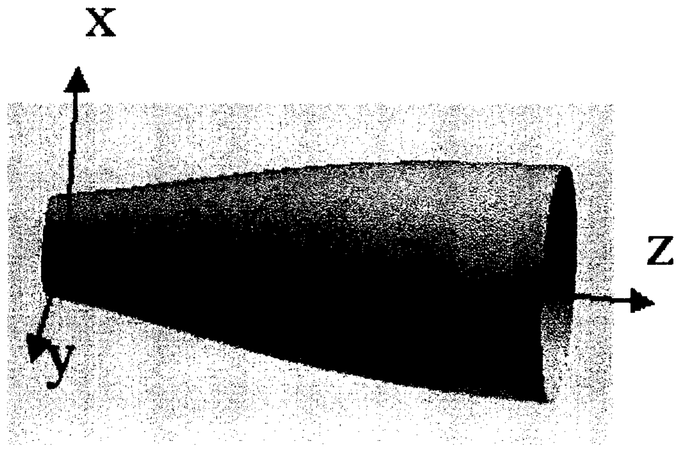


Figure 26. Nozzle Coordinate System

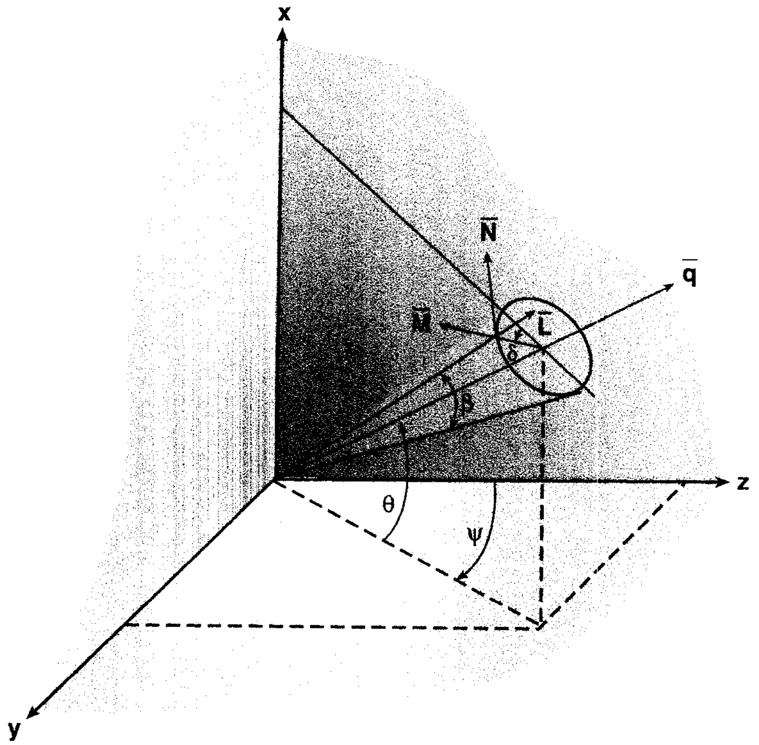


Figure 27. Mach Conoid Coordinate System

For this 3D solution, the compatibility equations are determined from the flow equations (eqs. 8-10) with a constraint that their derivatives in the direction normal to the characteristic surface are zero.

For the surface defined by the Mach conoid (eq. 8), the compatibility equation in difference form is:

$$\frac{\cot\beta_i}{\rho_i q_i^2}(P_2 - P_i) + \cos\delta_i(\theta_2 - \theta_i) + \cos\theta_i \sin\theta_i(\psi_2 - \psi_i) + \sin\beta_i\left(\cos\theta_i \cos\delta_i\left(\frac{\partial\psi}{\partial N}\right)_i - \sin\delta_i\left(\frac{\partial\theta}{\partial N}\right)_i\right)dL_i = 0 \quad (15)$$

Where $\frac{\partial}{\partial L}$ and $\frac{\partial}{\partial N}$ are derivatives along and normal to the bicharacteristic.

The compatibility equations along a streamline are defined as:

$$\frac{\gamma}{\gamma - 1}RdT = \frac{1}{\rho}dP = -qdq \quad (16)$$

4.1.2 Solution Methodology

The solution begins with the definition of an initial reference plane. This and all subsequent planes are normal to the z-axis as defined in Figure 26. The flow properties (P, T, V, etc.) at the initial reference plane are set by the user. The number of points within this plane is also user defined. Given the number of points, the tool creates a grid of nearly equally spaced points. Figure 28 shows an example of this grid spacing.

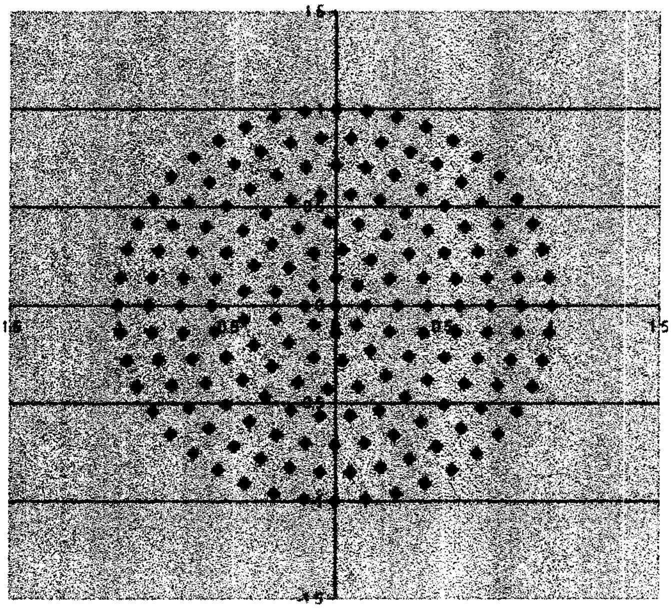


Figure 28. Initial Reference Plane Points Calculation

The points on the outer most radius define the initial wall contour, and will be referred to as body points. The remaining points define the internal flowfield and will be referred to a field points.

In addition to the initial reference plane, the tool also needs the nozzle wall contour in order to reach a solution. In this tool, the wall contour is defined by the following equation.

$$r_i^2 = (x - x_i)^2 + (y - y_i)^2 \tag{17}$$

The values of r_i, x_i and y_i are read from a datafile. Figure 29 shows a sample of this file. The value in the first row defines the number of axial (z) stations in the nozzle geometry. The next row includes a data header ('Z r0 x0 y0'). The parameter z defines the axial location. The proceeding rows contain the data values.

```
158
Z           r0          x0    Y0
0           1           0     0
0.003408    1.00001     0     0
0.006792    1.00002     0     0
0.008956    1.00004     0     0
0.011449    1.00007     0     0
0.014282    1.0001      0     0
```

Figure 29. Nozzle Geometry Input File

4.1.2.1 New Reference Plane Determination

After the initial plane and nozzle geometry has been defined, a new reference plane is created at some distance, z, downstream of the initial reference plane. The value of the new 'z' is taken from the nozzle geometry file. The distance from the initial reference plane to this new plane is defined as 'dz'. For every point (P_1) defined on the initial plane, a corresponding point (P_2) on the new reference plane is found. This process is repeated until the last reference plane has been computed.

4.1.2.2 Field Point Calculation

The field point calculation is used to determine all internal (not wall) points on the newly created reference plane. Figure 30 shows a graphical representation of the parameters involved. For a given point P1 on the initial reference plane, a streamline is constructed from P_1 to P_2 which lies on the new reference plane. The position of P_2 is defined by:

$$x_2 = x_1 + \frac{\tan\theta}{\cos\psi}dz \tag{18}$$

$$y_2 = y_1 + (\tan\psi)dz \tag{19}$$

$$z_2 = z_1 + dz \tag{20}$$

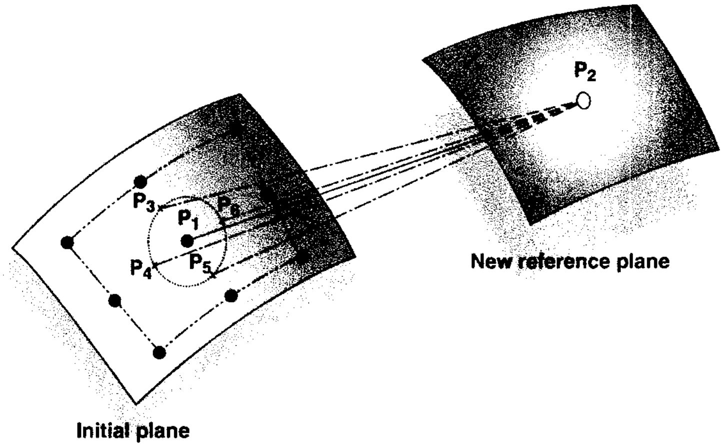


Figure 30. Field Point Calculation Parameters

To determine the flow properties at P$_2$, four bicharacteristic lines (i) are constructed from P$_2$, back to the initial reference plane. These lines (which are rays along a Mach conoid) are defined by equations 9-11. The intersection of these lines and the initial plane are shown as points P$_3$-P$_6$ in Figure 30. By manipulating equations 9-11, the position of these points is defined as follows:

$$x_i = x_2 - (\cos\beta\sin\theta + \sin\beta\cos\theta\cos\delta_i)dL_i \tag{21}$$

$$y_i = y_2 - [\cos\beta\cos\theta\sin\psi - \sin\beta(\sin\theta\sin\psi\cos\delta_i - \cos\psi\sin\delta_i)]dL_i \tag{22}$$

$$dL_i = \frac{dz}{[\cos\beta\cos\theta\cos\psi - \sin\beta(\sin\theta\cos\psi\cos\delta_i + \sin\psi\sin\delta_i)]} \tag{23}$$

$$z_i = z_1 \tag{24}$$

The parametric angles, δ_i, for points 3-6 are 0, $\pi/2$, π and $3\pi/2$ respectively.

4.1.2.2.1 Surface Fit Algorithm

The flow properties at these points are then determined based on a surface fit of the initial reference plane. The surface fit algorithm used is based on the small deflection equation of an infinite plate and results in the following surface fit equation.

$$W(x,y) = a_0 + a_1x + a_2y + \sum_{i=1}^{N} b_i r_i^2 \ln r_i^2 \tag{25}$$

Where W(x, y) represents the flow property (P, T, V, etc.) to be found; N defines the number of the points to use in the fit; and X, y and r are defined in equation 17. See reference 14 for a detailed derivation of this method. Using this method, there are N+3 unknowns (a$_0$, a$_1$, a$_2$,

$b_1 \ldots b_N$), however equation 25 yields only N equations to be solved, so three other equations are needed. They are:

$$\sum_{i=1}^{N} b_i = \sum_{i=1}^{N} x_i b_i = \sum_{i=1}^{N} y_i b_i = 0 \qquad (26\text{-}28)$$

Initial implementation of this surface fit, had N=9, where for a given point P_1, P_1 and its eight nearest neighbors were used to determine the fit. This yielded slight errors in the fit. In particular, for several test cases where a uniform flowfield was expected, this fit yielded slight difference in the flow properties for points at constant radial locations. Ultimately N was set to include all of the points in the reference plane, and this error was eliminated.

Further research was conducted to determine if there was a better algorithm to use. Three algorithms were found (Algorithm 792[15], CSHEP2D[16] and Algorithm 761[17]), each with its own benefits and detriments, however none have been implemented to date.

4.1.2.2.2 Field Point Compatibility Equation Solution

Once the four bicharacteristic points (P_3-P_6) have been found and their flow properties determined, the compatibility equation is then solved from each point (P_3-P_6) to P_2. As previously highlighted in eq. 15, the compatibility equation in difference form is:

$$\frac{\cot\beta_i}{\rho_i q_i^2}(P_2 - P_i) + \cos\delta_i(\theta_2 - \theta_i) + \cos\theta_i \sin\theta_i(\psi_2 - \psi_i) + \sin\beta_i\left(\cos\theta_i\cos\delta_i\left(\frac{\partial\psi}{\partial N}\right)_i - \sin\delta_i\left(\frac{\partial\theta}{\partial N}\right)_i\right)dL_i = 0 \qquad (29)$$

Where:

$$\left(\frac{\partial\theta}{\partial N}\right)_i = \left(\frac{\partial\theta}{\partial x}\right)_i\left(\frac{\partial x}{\partial N}\right)_i + \left(\frac{\partial\theta}{\partial y}\right)_i\left(\frac{\partial y}{\partial N}\right)_i - \left[\theta_i + \left(\frac{\partial\theta}{\partial x}\right)_i(x_2 - x_1) + \left(\frac{\partial\theta}{\partial y}\right)_i(y_2 - y_1)\right]\frac{\left(\frac{\partial z}{\partial N}\right)_i}{dz} + \theta_2\frac{\left(\frac{\partial z}{\partial N}\right)_i}{dz} \qquad (30)$$

$$\left(\frac{\partial x}{\partial N}\right)_i = -\cos\theta\sin\delta_i \qquad (31)$$

$$\left(\frac{\partial y}{\partial N}\right)_i = \sin\theta\sin\psi\sin\delta_i + \cos\psi\cos\delta_i \qquad (32)$$

$$\left(\frac{\partial z}{\partial N}\right)_i = \sin\theta\cos\psi\sin\delta_i - \sin\psi\cos\delta_i \qquad (33)$$

Note that the same form of equation 30 holds true for ψ.

The unknowns in eq. 29 are p_2, θ_2, and ψ_2, so only three bicharacteristics are required for a solution. Reference 14 recommended averaging four separate solutions (each using 3 points) to improve accuracy. This tool uses this recommendation. The compatibility equation is solved for using four sets of points [(P_3, P_4, P_5), (P_3, P_5, P_6), (P_3, P_4, P_6) and (P_6, P_4, P_5)] and the resultant values for p_2, θ_2, and ψ_2 are averaged.

Using the compatibility equations along an isentropic streamline (eq. 16) the values of ρ_2, T_2 and q_2 can be found as follows:

$$T_2 = T_T\left(\frac{P_2}{P_T}\right)^{\gamma-1/\gamma} \tag{34}$$

$$\rho_2 = \frac{RT_2}{P_2} \tag{35}$$

$$q_2 = \sqrt{2\frac{\gamma}{\gamma-1}R(T_T - T_2)} \tag{36}$$

Where P_T and T_T are the total conditions along the streamline.

Given the calculated flow quantities at P_2, a new streamline from P_1 (based on the average properties of P_1 and P_2) is constructed to a new P_2 location and new flow properties are found using the same process. The iteration continues until the location and flow properties at P_2 remain constant (within some tolerance). This process is repeated for all of the field points.

4.1.2.2 Body Point Calculation

The Body point calculation is performed for every point along the nozzle wall. Figure 31 shows a graphical representation of the parameters involved.

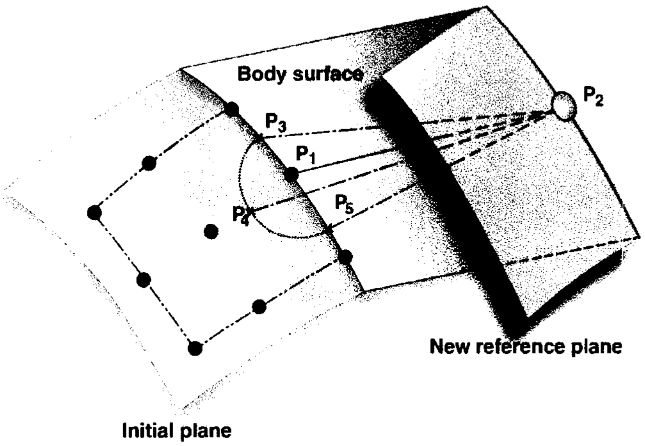


Figure 31. Body Point Calculation Parameters

For a given body point P_1, a new body point P_2 is found at the intersection of a plane defined by the body surface unit normal and the unit velocity vector tangent to body surface at P_1 and the body surface at P_2. This leads to solving the following equations simultaneously.

$$B(x,y,z)=0 \quad \text{[Body surface]} \qquad (37)$$

$$(n_3\cos\theta\cos\psi - n_1\cos\theta\sin\psi)(x-x_1)+(n_1\sin\theta - n_2\cos\theta\cos\psi)(y-y_1)=(n_3\sin\theta - n_1\cos\theta\sin\psi)dz \quad (38)$$

Where n_1, n_2, and n_3 are unit normals to the body surface.

4.1.2.2.1 Body Surface Calculation

Based on the nozzle wall geometry around P_1 and P_2, the body surface is approximated as one of the following four shapes.

□	Vertical line:	x = c
□	Horizontal line:	y = c
□	Sloped line:	ax + by = c
□	Circle:	$(x\text{-}a)^2 + (y\text{-}b)^2\text{=}c$

At each point, the values of a, b, and c are solved and then used to determine the body normal coefficients.

4.1.2.3 Compatibility Equation Solution

After the location of P_2 has been found, three bicharacteristics are constructed from P_2 back to the initial reference plane. The intersection of these lines with the initial plane is defined as P_3-P_5. P_4 is found so that it lies on the line normal to the body surface and P_1. Mathematically, the parametric angle for P_4 is defined as

$$\delta_4 = \cos^{-1}\left(-n_1 \sin\theta \cos\psi + n_2 \cos\theta - n_3 \sin\theta \sin\psi\right) \tag{39}$$

The parametric angles for P_3 and P_5 are iterated so that the points are within the nozzle flow region.

For this calculation, there are three unknowns, p_2, θ_2, and ψ_2 (from eq. 29). To solve for the parameters, the compatibility equation along two bicharacteristics (eq. 29) and a flow tangency constraint are solved. The tangency constraint sets the condition that the flow is tangent to the surface at P_2 and is defined as:

$$n_1 \cos\psi_2 + n_2 \tan\theta_2 + n_3 \sin\psi_2 = 0 \tag{40}$$

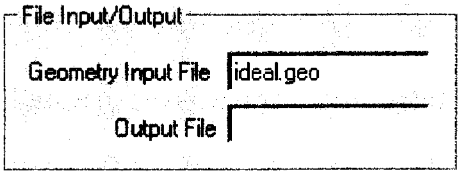
To increase accuracy, three sets of equations are solved and their results averaged. Each set has a different set of bicharacteristics, (P_3, P_4), (P_3, P_5) and (P_4, P_5). The values for ρ_2, T_2 and q_2 are solved used eq. 34-36. The iteration continues until the location and flow properties at P_2 remain constant (within some tolerance). This process is repeated for all of the body points.

4.2 3D MOC Tool Inputs

In this version of the tool, the inputs shown in white are functional. The grayed inputs have been included as place holders for future implementation.

4.2.1 File Input/Output

Within this input, the user is asked to input the name of nozzle geometry file. Figure 32 shows a screen capture of the input. The format of the nozzle geometry must follow the format as discussed in Section 4.1.2 and Figure 29.



File Input/Output

Geometry Input File: ideal.geo

Output File:

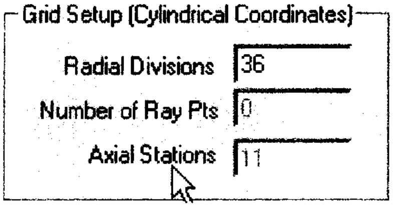
Figure 32. File Input/Ouput Input

4.2.2 Grid Setup

The available input in the *Grid Setup* input is the number of radial division in the initial reference plane (see Figure 33). This value defines the number of equally spaced points to be calculated along the nozzle wall. The distance between the points is defined the following equation, where R is the known initial plane radius:

$$d = \frac{2*\pi*R}{n_{Divisions}} \tag{41}$$

The calculated distance, d, is used to create a nearly equally space grid within the nozzle geometry as shown in Figure 28. A nominal value of 36 yields a good balance between solution accuracy and run time.

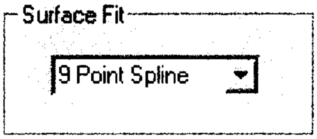

Grid Setup (Cylindrical Coordinates)

Radial Divisions: 36
Number of Ray Pts: 0
Axial Stations: 11

Figure 33. Grid Setup

4.2.3 Surface Fit

The *Surface Fit* input allows the user to define the type of surface fit used in the calculation. Figure 34 shows a screen capture of this input. The choices as discussed in Section 4.1.2.2.1 are a 9-point fit and a fit that uses the entire reference plane.

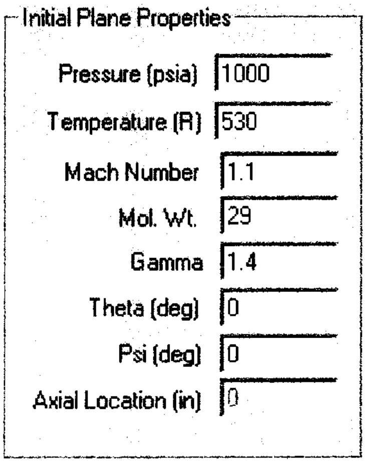

Surface Fit

9 Point Spline

Figure 34. Surface Fit Input

4.2.4 Initial Plane Properties

The *Initial Plane Properties* input defines the properties of the initial reference plane (see Figure 35). Within the current implementation, the flow properties over the entire initial plane are constant. This limits the tool's capability to calculate non-uniform initial flowfields. The solution algorithm presented herein is capable of obtaining a solution starting with a non-uniform flowfield, the hindrance really occurs in the user input. To truly implement a non-uniform flowfield, the input would have to be changed, where the user specifies all of the flow properties at every point on initial plane.



Initial Plane Properties

Pressure (psia) 1000
Temperature (R) 530
Mach Number 1.1
Mol. Wt. 29
Gamma 1.4
Theta (deg) 0
Psi (deg) 0
Axial Location (in) 0

Figure 35. Initial Plane Properties Input

4.2.5 Calculate Nozzle Button and Progress Indicators

The calculation is started by pressing the *Calculate Nozzle* button as shown in Figure 36. The code determines the number of axially steps needed to reach the end of the nozzle and inputs that number into the *Total Steps* box (see Figure 37). As the code calculates each new axial station, the *Step Number* box is updated.

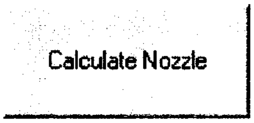
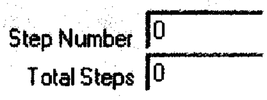


Figure 36. Calculate Nozzle Button



Step Number 0
Total Steps 0

Figure 37. Progress Indicators

4.2.6 Print Output Parameters

These parameters (see Figure 38) define the grid resolution on the output files. The *Every N Point in X* parameter helps define the number of points in a given plane to output as follows:

$$n_{\text{points}} = \frac{n_{\text{Total Points}}}{\text{Every N Point in X}} \tag{42}$$

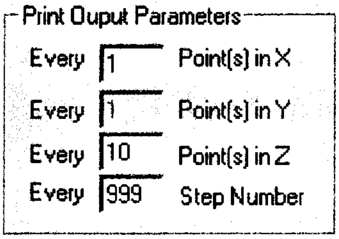
This is most useful when outputting streamlines. The number of points, n_{points}, will equal the number of streamlines output.

The *Every N Points in Z* controls the number of axial stations (z-direction) that the code outputs based on the following equation.

$$n_{\text{planes}} = \frac{n_{\text{Total Steps}}}{\text{Every N Point in Z}} \tag{43}$$

The first and last planes will always be output regardless of this value.

The *Every N Step Number* input determines when the output files are generated based on *Step Number* discussed in Section 4.2.5. If the *Every N Step Number* value is greater than the *Total Steps*, then the files are created at the end of the calculation. This option may come in handy if the tool bombs before completion. By setting this number to the *Step Number* where the tool bombed, the code will output the flow at that station where it can then be reviewed. Table 2 summarizes the various output files produced by the tool.



Print Ouput Parameters

Every 1 Point(s) in X

Every 1 Point(s) in Y

Every 10 Point(s) in Z

Every 999 Step Number

Figure 38. Print Output Parameters

Table 2. 3D MOC Tool Output Files

Data File Name	Description
Z=0.out	Contains the location of the initial throat plane points and their respective flow properties
Wall.plt	TECPLOT formatted file showing all of the nozzle wall flow properties
Full_mesh.plt	TECPLOT formatted file showing all of the nozzle flowfield data. The data is organized as 1 surface from the centerline to the wall
AxialStations.plt	TECPLOT formatted file showing all of the nozzle flowfield data. The data is organized in separate axial stations.
Initial Wall.plt	TECPLOT formatted file showing the original nozzle geometry as read from the input nozzle geometry file. The number of points to print out does not affect this file.
Streamlines.plt	TECPLOT formatted file showing all of the nozzle flowfield data. The data is organized in separate streamlines starting from each of the initial reference plane starting points.

5.0 2D and 3D MOC Tool Verification

5.1 2D MOC Tool Verification

In order to ensure a valid solution, numerous example cases were run on these tools. The most extensive verification process was conducted on the 2D MOC tool. For given nozzle design parameters, the tool was run to verify that the calculated exit plane met these parameters. For example, for a perfect nozzle solution with a prescribed exit Mach number of 4.0, the nozzle solution was checked to make sure that the resultant exit Mach number was 4.0, and that the exit plane was uniform (as defined by a perfect nozzle). Figure 39 shows several nozzle contours created by the 2D MOC tool. This figure is also useful in understanding the different nozzle types (Perfect, Rao, and Set Endpoint) for a given exit condition (Mach = 4.0)

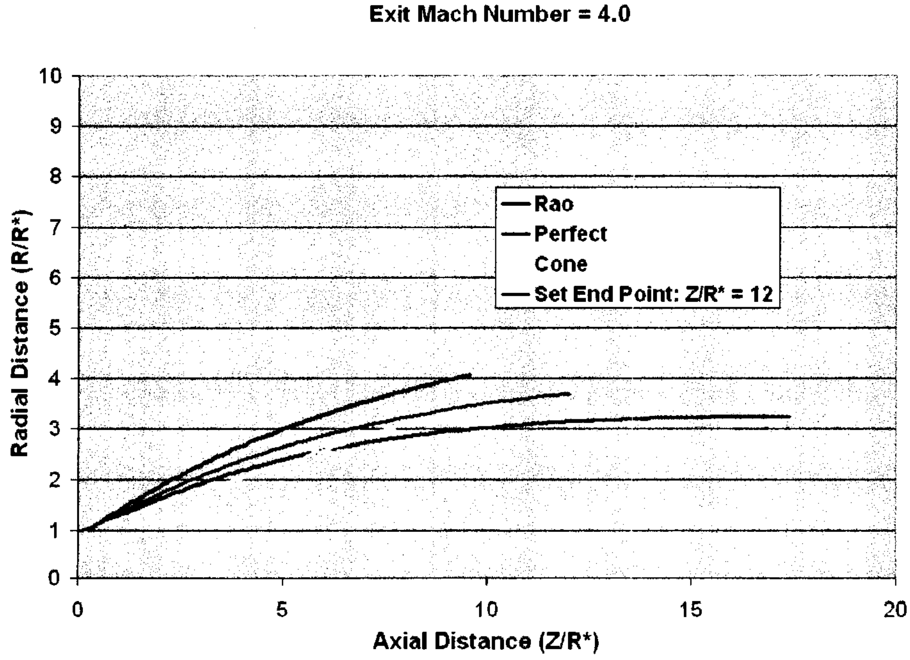


Figure 39. 2D MOC Tool Nozzle Contours

5.2 3D MOC Tool Verification

The 3D MOC Tool was verified by running the two samples cases. In both cases, the nozzle geometries generated from the 2D MOC tool were used as inputs to the 3D tool. The resultant 3D flowfield was then compared to the known 2D flowfield. The first case run was a perfect nozzle with a prescribed exit Mach number = 4.0. Table 3 shows the 2D MOC tool parameters in the definition of the nozzle. The resultant initial Mach number is 1.15181.

Table 3. 2D MOC Tool Parameters for 3D Perfect Nozzle Verification

Parameter	Value
Nozzle Geometry	Axisymmetric
Nozzle Type	Perfect
Design Parameter	Mach = 4.0
UpStream Radius/R*	1.0
DownStream Radius/R*	1.0
Throat Conditions Box	CHECK
Pressure, psia	1000
Temperature, R	530
Mol. Wt.	28.96
Gamma	1.4
P ambient, psia	0.0
Velocity, ft/s	3022
ThetaB Guess	25
# of RRC above BD	100
DTHETAB Max, deg	0.5
Number of Starting Characteristics	100

Figure 40 shows a Mach contoured plot of the resultant 3D flowfield. As is seen, the Mach contours along the wall are uniform at each axial station, which is what should be expected for a perfect nozzle.

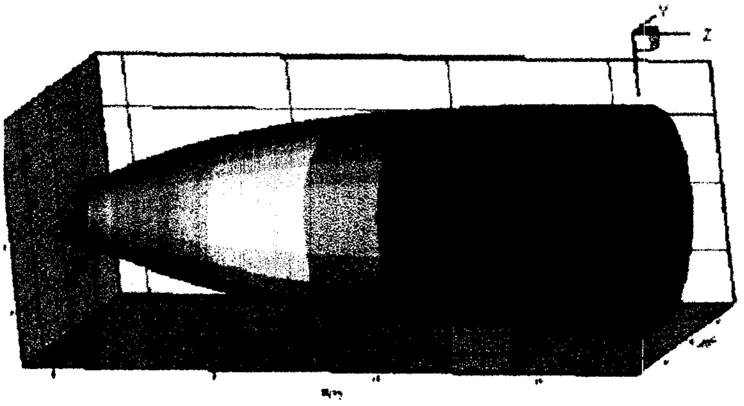


Figure 40. 3D MOC Tool Nozzle Mach Contours

The nozzle exit plane was also interrogated to analyze the flowfield more closely. Figure 41 shows a Mach profile plot for all of the points at the nozzle exit plane. The nominal exit Mach number is 4.0. Three sets of points are presented, each based on a different *Radial Division* input. Table 4 details the statistical results. All of the cases over predict the exit Mach number; however, the deviation in the resultant Mach number is small. Moreover, as the number of radial divisions increases (increased number of points) the error increases slightly, but the deviation in all the values decreases. The over-prediction is most likely caused by definition of the initial nozzle plane. The 2D MOC solution calculates the initial data line as a right-running characteristic (Z varies), where as the 3D MOC initial plane is calculated as a vertical plane (Z is constant). Also, small errors due to calculation approximations and tolerances can get amplified as the flowfield is propagated downstream.

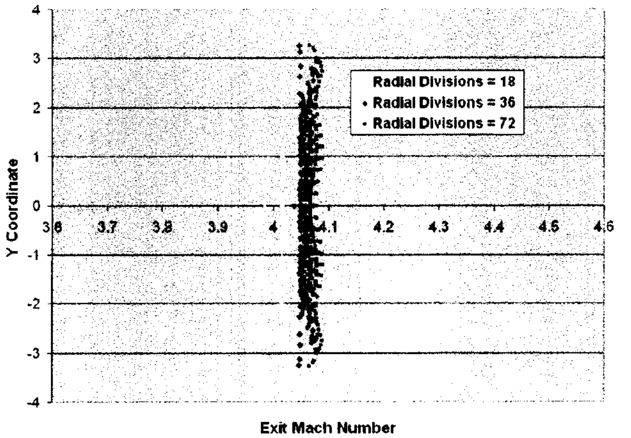


Figure 41. Perfect Nozzle Exit Plane Mach Number for Various Radial Division Inputs

Table 4. Mach 4 Perfect Nozzle Exit Plane Analysis

	Radial Divs. = 18	Radial Divs. = 36	Radial Divs. = 72
Number of points	67	181	580
Average Exit Mach Number	4.027	4.051	4.067
3-σ Deviation	0.067	0.032	0.019
Average % Difference	0.664	1.29	1.67

The 3D MOC Tool was also checked for a Mach 4 Rao nozzle design. The same inputs (except for the Nozzle Type) as shown in Table 3 were used in the generation of the 2D MOC nozzle geometry. Figures 42 and 43 show the resultant pressure traces along the wall and centerline for the 2D tool and the 3D tool, as well as the results from the SEA RAO and TDK codes respectively. As is seen all the results compare favorably.

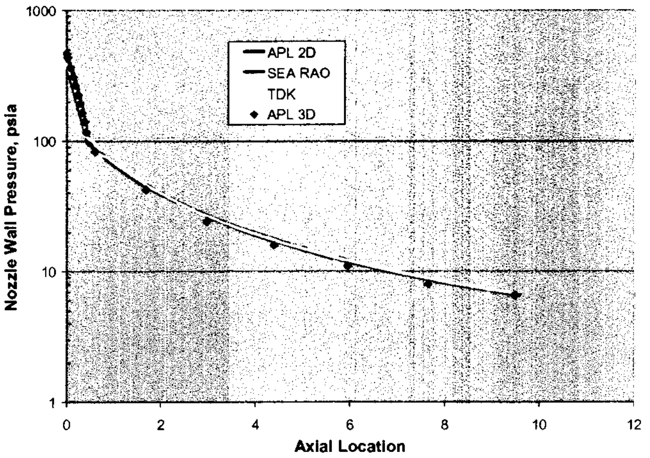


Figure 42. Nozzle Wall Pressure Comparison

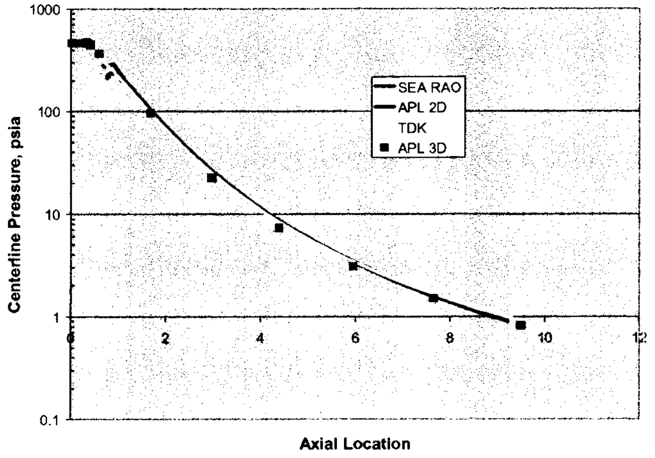


Figure 43. Nozzle Centerline Pressure Comparison

Figure 44 shows a comparison of the nozzle exit plane pressure. The SEA RAO code does not include the exit plane profile in its output and therefore is not plotted in the figure. As is shown, the results are similar, however the 2D tool returns a discontinuous profile which is different from the other two codes. It is unclear as to why this occurs and more investigation may be necessary. It should be noted that the 2D methodology calculates the nozzle exit profile after the nozzle contour is determined; therefore, this discontinuity has no effect of the nozzle contour. It should also be noted that this profile is only seen in the RAO nozzle case. As was shown in Figure 41, this is not seen in a 'perfect' nozzle solution. Mach contours for the RAO nozzle solution are shown in Figure 45.

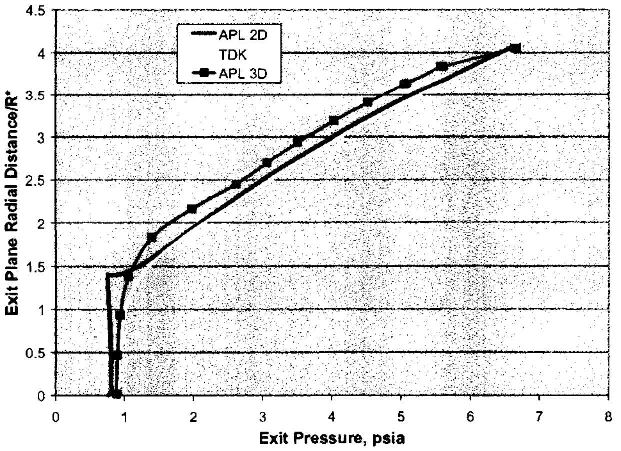


Figure 44. Exit Plane Pressure Comparison

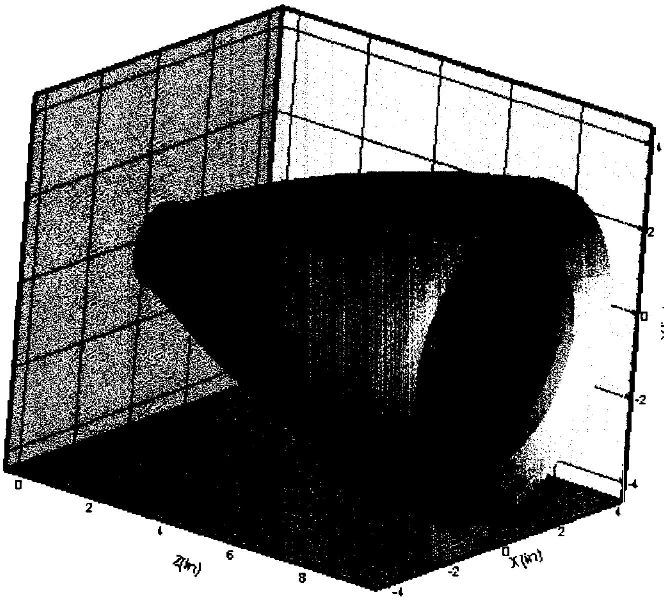


Figure 45. Mach Contours for 3D RAO Nozzle Solution

6.0 Summary

Under this task APL has developed a 2D nozzle design tool and 3D nozzle analysis tool based on a Method of Characteristics (MOC) solution algorithm. The 2D nozzle design tool can quickly solve for an inviscid nozzle contour for various types of nozzles and flow inputs. The 3D tool can be used to determine nozzle performance and flowfield properties for truly 3-Dimensional shapes and inflows. Several additional efforts have been identified throughout this report, which would improve the accuracy and robustness of the tools.

7.0 References

1. Rice, T., and VanWie, D. M., "Design of Exhaust Nozzle for the RBCC-Trailblazer Concept Final Report," AATDL-00-033, JHU/APL, February 2000.

2. Rice, T. "Design of Exhaust Nozzles for the RBCC-GTX Concept Final Report," RTDC-TPS-335, NASA Grant NAG3-2460, May 2001.

3. Anderson, J. D., Fundamentals of Aerodynamics, McGraw Hill Publishing Company, New York, New York, 1984.

4. VanWie, D., and Molder, S., "Applications of Busemann Inlet Designs for Flight at Hypersonic Speeds," AIAA-92-1210, February 1992.

5. Nonweiler, T.R.F, "Delta Wings of Shape Amenable to Exact Shock-Wave Theory," Journal of the Royal Aeronautical Society, Vol. 67, 1963.

6. Maikapar, G.I., “Bodies Formed by the Stream Surfaces of Conical Flows,” Fluid Dynamics, Vol. 1, No. 1, 1966.

7. Rassmussen, M.L., “Waverider Configurations Derived from Inclined Circular and Elliptic Cones,” Journal of Spacecraft and Rockets, Vol. 17, Nov.-Dec., 1980.

8. Taylor, G.I., and Maccoll, J.W., “The Air Pressure on a Cone Moving at High Speeds,” Proc. Roy. Soc. (London) A319, 1933.

9. Bowcutt, K. G., Anderson, J. D., and Capriotti, D, “Viscous Optimized Hypersonic Waveriders,” AIAA-87-0272, Jan. 1987.

10. Corda, S., and Anderson, J. D., “Viscous Optimized Hypersonic Waveriders Designed from Axisymmetric Flow Fields,” AIAA-88-0369, Jan. 1988.

11. Moe, M.M. and Troesch, B.A., “The Computation of Jet Flows with Shocks,” STL Report TR-59-0000-00661, May 1959.

12. Rao, G.V.R, “Exhaust Nozzle Contour for Optimum Thrust,” Marquadt Aircraft Co, Presented at the ARC Semi Annual Meeting, San Fransisco, CA, June 10-13, 1957.

13. Kliegel, J.R and Levine, J.N.,”Transonic Flow in Small Throat Radius of Curvature Nozzles,” AIAA Journal, Vol. 7, No. 7, July 1969, pp. 1375-1378.

14. Armstrong, W.C., “A Method of Characteristic Computer Program For Three-Dimensional Supersonic Internal Flows,” AEDC-TR-78-68, January 1979.

15. Renka, R.J., “Algorithm 792: Accuracy Tests of ACM Algorithms for Interpolation of Scattered Data in the Plane,” *ACM Transactions on Mathematical Software*, Vol. 25, No. 1 (March 1999), Pages 78-94.

16. Renka, R.J., “Algorithm 790: CSHEP2D: Cubic Shepard Method for Bivariate Interpolation of Scattered Data,” *ACM Transactions on Mathematical Software*, Vol. 25, No. 1 (March 1999), Pages 70-73.

17. Akima, H., “Algorithm 761: Scattered-Data Surface Fitting that Has the Accuracy of a Cubic Polynomial,” *ACM Transactions on Mathematical Software*, Vol. 22, No. 3 (September 1996), Pages 362-371.

8.0 Acknowledgements

The author would like to thank Jim DeBonis (GRC), Nick Georgiadis (GRC), Maria Rigling (APL), Tim Smith (GRC) and Chuck Trefny (GRC), for their assistance in the development of these two MOC nozzle tools. A special thanks goes out to W. C. Armstrong, the author of record for Reference 14, which was referenced extensively in the development of the 3D MOC tool.

9.0 Nomenclature

BD	Last RRC of the initial nozzle expansion region
δ	Parametric angle
γ	Ratio of specific heats
L	Length
LRC	Left Running Characteristic
M	Mach
MOC	Method of Characteristics
μ, β	Mach angle
n	Normal surface coefficient
P	Pressure
Point E	Nozzle exit wall point
Point B	End of the initial nozzle expansion region
ψ	Flow angle w/rt the y-z plane
R	Gas constant
RRC	Right Running Characteristic
R_{DOWN}	Radius of the arc defining the initial nozzle expansion region
R_{UP}	Radius of arc defining the converging throat section
R^*	Nozzle throat radius
r	Radial distance
ρ	Density
θ	Flow angle (also nozzle wall angle) w/rt the x-z plane
T	Temperature
TT’	Initial data line of nozzle solution
TB	Arc defining the initial nozzle expansion region
V,q	Velocity vector
u	Velocity in the x-direction
v	Velocity in the y-direction
w	Velocity in the z-direction
x	Transverse direction used in nozzle solution
y	Transverse direction used in nozzle solution
z	Axial direction used in nozzle solution
dz	Distance between axial planes