4

AL-TR-89-011 AD:

Report
Period
988 to
1989

Gas Flows in Rocket Motors
Volume 1

st 1989

Authors:

Science Applications International
Corporation
21151 Western Avenue
Torrance CA 90501

Y. Kronzon
C. Lyan-Chang
C.L. Merkle

Pennsylvania State University
Mechanical Engineering Department
University Park PA 16802

F04611-88-C-0014

Approved for Public Release

Distribution is unlimited. The AL Technical Services Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

Prepared for the **Astronautics Laboratory (AFSC)**

Air Force Space Technology Center
Space Systems Division
Air Force Systems Command
Edwards Air Force Base, California 93523-5000

NOTICE

When U.S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, or in any way licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may be related thereto.

FOREWORD

This is the final report for Task 1, Navier Stokes Analysis of Rocket Nozzles, for SETA contract F04611-88-C-0014 with the Astronautics Laboratory (AFSC), Edwards AFB CA. This work was performed by Pennsylvania State University as a subcontractor to Science Applications International Corporation (the SETA contractor). Dr Philip A. Kessel was the project manager for this analysis task.

This report has been reviewed and is approved for release and distribution in accordance with the distribution statement on the cover and on the DD Form 1473.

Philip A. Kessel

PHILIP A. KESSEL, Ph.D.
Project Manager

Lawrence P Quinn

LAWRENCE P. QUINN, Ph.D.
Chief, Aerothermochemistry
Branch

FOR THE DIRECTOR

Robert C Corley

ROBERT C. CORLEY
Deputy Director, Astronautical
Sciences Division

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for Public Release; Distribution is unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AL-TR-89-011
6a. NAME OF PERFORMING ORGANIZATION Pennsylvania State University 6b. OFFICE SYMBOL *(If applicable)*	7a. NAME OF MONITORING ORGANIZATION Astronautics Laboratory (AFSC)
6c. ADDRESS *(City, State, and ZIP Code)* Mechanical Engineering Department University Park PA 16802	7b. ADDRESS *(City, State, and ZIP Code)* AL/LSCF Edwards AFB CA 93523-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F04611-88-C-0014
8c. ADDRESS *(City, State, and ZIP Code)*	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO
62302F	3058	00	SD

11. TITLE *(Include Security Classification)*
Gas Flows in Rocket Nozzles. Volume 1

12. PERSONAL AUTHOR(S)
Kronzon, Yigal; Lyan-Chang, Chau; and Merkle, Charles L.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Final	FROM 88/07 TO 89/03	89/06	

16. SUPPLEMENTARY NOTATION
This is volume 1 of a 3 volume report

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP
20	04	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
nozzle analysis, Navier-Stokes, turbulent flow, equilibrium chemistry

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Detailed descriptions of the governing equations, the method of solution, and the computer code for calculating perfect gas or real gas flow in axisymmetric nozzles are given. The codes permit calculation of a perfect gas or a real gas for inviscid flow by solving the Euler equations, and for viscous flow by solving the thin layer Navier-Stokes equations. These equations are written in a conservative form and solved implicitly in body-fitted coordinates. The solution obtained by the conservative variables is expressed in terms of the density, ρ, the momentum parallel to the axis of symmetry (u), the momentum perpendicular to the axis (ρv), and the total internal energy, e_o. These variables are then used to calculate the nonconservative primitive variables, the velocity components, u and v, the pressure, p, and the temperature, T. The nozzle performance including the rate of mass flow, $\dot{m}$, the thrust, T, and the specific impulse are also computed.

The codes were written in FORTRAN V and ran on the CYBER 180/840, NOS/BE system which limited the number of grid points to 20 x 44 for solving the Euler equations and 60 x 40 for

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE *(Include Area Code)*	22c. OFFICE SYMBOL
Dr Philip A. Kessel	(805) 275-5591	AL/LSCF

AL-TR-89-011 Vol 1 (continued)

Block 19.

the TLNS equations. The results obtained for the nozzle flowfield showed maximum global mass flux errors of lwss than ±1% for the Euler equations and less tha ±2% for the TLNS equations. Solutions with more dense grids (typically 100 x 50 or higher) consistently showed global mass conservation of better than one percent.

PREFACE

This document is the final report for Task 01, Navier Stokes Analysis of Rocket Nozzles, of the Research Applications Scientific and Engineering Technical Assistance (SETA) contract, F04611-88-C-0014. This contract provides support to the Air Force Astronautics Laboratory (AFAL). Dr. Lawrence Quinn, AFAL/LSC, is the Air Force Project Manager for this contract. Science Applications International Corporation (SAIC) is the prime contractor and Dr. Robert Long, Jr. is the SAIC Program Manager. The duration of this task was 9 June 1988 - 31 March 1989.

The Air Force Task Manager for this task was Dr. Philip Kessel, AFAL/LSCF. The work was performed by Pennsylvania State University under subcontract 11-890022-69 with SAIC. Dr. Charles L. Merkle, Dept. of Mechanical Engineering, was the Principal Investigator at Penn State. The other principal technical contributors were Dr. Y. Kronzon, Dr. V. Yang, and Dr. R. Santoro. We appreciate the assistance of Mrs. Phyllis Mosesman in the preparation of this report.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

LIST OF FIGURES

LIST OF FIGURES (Cont'd)

LIST OF TABLES

INTRODUCTION

The present document is a final report for work on supersonic nozzle analysis for both perfect and real gases and for viscous and inviscid flows. The primary focus of this research was to develop an accurate, efficient procedure for computing nozzle flows with careful attention to global mass conservation. In general, the goal was to demonstrate computations with mass flux errors of less than 1% on moderately resolved grids. Calculation procedures for both viscous and inviscid flows and for parabolized Navier-Stokes (PNS) algorithms have consistently demonstrated this goal. Primary emphasis has been on perfect gas calculations, but a real gas analysis has been developed and demonstrated to a lesser degree. Also included in the effort was a procedure for coupling the nozzle wall cooling flow with the nozzle flowfield so that both nozzle wall temperature and heat flux were simultaneously determined by the calculation. Finally, a procedure for determining the effect of small variations in back pressure on the nozzle flowfield has been developed.

The present final report includes as Appendix C a Ph.D. thesis[1] which describes the detail of the perfect gas work. In particular, this Appendix includes the coupled nozzle wall cooling and nozzle back pressure calculations as an appendix. Results of this work have also appeared as parts of three papers[2,3,4] which are available in the literature. The main body of the report contains nozzle back pressure calculations, including flows with weakly separated boundary layers and associated recirculation regions. This Appendix also describes methods for extending the present results to the fully three-dimensional problem.

The main body of the report is focussed on the real gas formulation and on representative results for real gas calculations. In addition, some recent perfect gas calculations which do not appear in Appendix C are included.

The high temperatures associated with rocket combustion processes lead to vibrational excitation and dissociation of the flowing gases, and as the gas expands through the nozzle, these high temperature effects have a significant influence on the physics of the flow and, therefore, cannot be computed by perfect gas assumptions.

The most complete model for describing these phenomena is to add the appropriate species diffusion equations and take into account the complete chemical kinetic effects. The present analysis, however, addresses the simpler problem of a real gas undergoing shifting equilibrium chemistry. This implies that the real gas chemistry is infinitely rapid. During the expansion, the gases in the nozzle generally reach a point at which their temperature and pressure are low enough that this infinite rate assumption fails and non-equilibrium effects begin to appear. The simplest expedient for handling this phenomena is to switch from equilibrium to frozen flow assumptions at a given point. Such issues are, however, not included in the present analysis which discusses only the equilibrium, real gas problem.

In the next section, the governing systems of equations are described, and the real gas model is defined. The real gas effects assume that the gas properties, including the molecular weight, M_w, the temperature, T, the internal energy, e, the viscosity, μ, and the thermal conductivity, K, are functions of the thermodynamic variables. For completeness, both the Euler equations and the thin-layer Navier-Stokes (TLNS) equations are considered.

In the following section, the implicit numerical schemes for solving the Euler and TLNS equations are described. The Euler solutions are limited to fully supersonic flows, while the TLNS solutions contain a thin subsonic layer along the wall. Corresponding transonic codes (as described in Appendix C) are used to generate the supersonic (or predominantly supersonic) start lines for these supersonic-oriented algorithms.

An important difference between the perfect gas analysis of Appendix C and the real gas equations of the main body of the report are that the flux vectors of the former are homogeneous while those for the real gas equations are non-homogeneous. This requires that a different splitting of the Jacobians associated with convective terms be used for the difference equations.

Differential equations to compute the nozzle rate of mass flow and thrust are also evaluated in this section. The next section includes the user's manual which explains how to run the supersonic codes. The following section presents representative real gas results. For simplicity, these calculations are based on the properties of equilibrium air which were more readily available than combustion gas properties. The extension to combustion gas properties is straightforward once the thermodynamic properties are available for the desired fuel, oxidizer and equivalence ratio.

THE GOVERNING EQUATIONS

The Euler and TLNS Equations

The two-dimensional axisymmetric form of the unsteady Navier-Stokes equations written in generalized body-fitted transformed coordinates are given by:

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial \xi} + \frac{\partial F}{\partial \eta} = H + \frac{\partial R}{\partial \xi} + \frac{\partial S}{\partial \eta} \qquad (1)$$

where,

$$Q = \frac{y}{J}\begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ e_o \end{Bmatrix} \qquad E = \frac{y}{J}\begin{Bmatrix} \rho U \\ \rho u U+\xi_x p \\ \rho v U+\xi_y p \\ (e_o+p)U \end{Bmatrix} \qquad F = \frac{y}{J}\begin{Bmatrix} \rho V \\ \rho u V+\eta_x p \\ \rho v V+\eta_y p \\ (e_o+p)V \end{Bmatrix}$$

$$H = \frac{1}{J}\begin{Bmatrix} 0 \\ 0 \\ P+\mu(\frac{2}{3}\ \nabla\bullet\vec{V}-2\frac{v}{y}) \\ 0 \end{Bmatrix} \qquad R = \frac{y}{J}\begin{Bmatrix} 0 \\ \xi_x\tau_{xx}+\ \xi_y\tau_{yx} \\ \xi_x\tau_{xy}+\ \xi_y\tau_{yy} \\ \xi_x f\ +\ \xi_y g \end{Bmatrix}$$

$$S = \frac{y}{J}\begin{Bmatrix} 0 \\ \eta_x\tau_{xx}\ +\ \eta_y\tau_{yx} \\ \eta_x\tau_{xy}\ +\ \eta_y\tau_{yy} \\ \eta_x f\ +\ \eta_y g \end{Bmatrix}$$

Here, (x,y) are the Cartesian coordinates parallel and perpendicular to the axis of symmetry, respectively; (ξ,η) are the body-fitted transformed coordinates,

$$\begin{aligned} \xi &= \xi(x,y) \\ \eta &= \eta(x,y) \end{aligned} \qquad (2)$$

where ξ is the streamwise coordinate which coalesces with the nozzle axis and the nozzle wall; J is the transformation Jacobian,

$$\frac{1}{J} = x_\xi y_\eta - x_\eta y_\xi \tag{3}$$

and U and V are the contravariant velocities,

$$U = \xi_x u + \xi_y V \text{ and } V = \eta_x u + \eta_y v \tag{4}$$

The viscous and conductive terms are,

$$\tau_{xx} = \mu[2(\xi_x \frac{\partial u}{\partial \xi} + \eta_x \frac{\partial u}{\partial y}) - \frac{2}{3} \nabla \bullet \vec{v}]$$

$$\tau_{yx} = \tau_{xy} = \mu[(\xi_x \frac{\partial v}{\partial \xi} + \eta_x \frac{\partial v}{\partial y}) + (\xi_y \frac{\partial u}{\partial \xi} + \eta_y \frac{\partial u}{\partial \eta})]$$

$$\tau_{yy} = \mu[2(\xi_y \frac{\partial v}{\partial \xi} + \eta_y \frac{\partial v}{\partial \eta}) - \frac{2}{3} \nabla \bullet \vec{v}] \tag{5}$$

$$f = \tau_{xx} u + \tau_{xy} v + k(\xi_x \frac{\partial T}{\partial \xi} + \eta_x \frac{\partial T}{\partial \eta}$$

$$g = \tau_{yx} u + \tau_{yy} v + k(\xi_y \frac{\partial T}{\partial \xi} + \eta_y \frac{\partial T}{\partial \eta})$$

$$\nabla \bullet \vec{v} = (\xi_x \frac{\partial u}{\partial \xi} + \eta_x \frac{\partial u}{\partial \eta}) + \frac{v}{y} + (\xi_y \frac{\partial v}{\partial \xi} + \eta_y \frac{\partial v}{\partial \eta})$$

The vector, Q, contains the dependent variables. Here, the standard notation has been used for density, ρ, velocity components, (u,v), pressure, p, temperature, T, and total internal energy, e_o. The vectors E and F are the convective fluxes, R and S are the viscous fluxes, and H is the source term.

The equation of state is,

$$P = \rho \frac{R}{M_w} T \tag{6}$$

This is the ideal gas equation where R is the universal gas constant.

The internal energy is defined in terms of the total energy, the density, and the velocity components,

$$e = \frac{e_o}{\rho} - \frac{(u^2 + v^2)}{2} \tag{7}$$

Four other equations are needed to close the set of equations. These equations are functions of the gas properties,

$$M_w = M_w\ (\rho,\ T) \quad \text{molecular weight} \tag{8}$$

$$T = T\ (\rho,\ e) \quad \text{temperature} \tag{9}$$

$$\mu = \mu\ (\rho,\ e) \quad \text{viscosity} \tag{10}$$

$$K = K\ (\rho,\ e) \quad \text{conductivity} \tag{11}$$

The gas properties can be represented in functional or in tabulated data form. For a perfect gas, M_W is constant and the temperature can be related to the other variables by,

$$T = \frac{\gamma}{Cp}\ (\frac{e_o}{\rho} - \frac{1}{2}\ (u^2 + v^2) \tag{12}$$

where Cp is the specific heat at constant pressure and γ is the specific heat ratio.

The Euler equations are derived from the full Navier-Stokes equations by eliminating all the viscous and heat transfer terms, i.e., $\mu = k = 0$. The inviscid equations then become identical to Eqn. 1 with R and S set to zero.

The advantage of this reduced set of equations is that it can be solved numerically using much less computer time and memory than is required for the complete Navier-Stokes equations.

The thin-layer Navier-Stokes (TLNS) equations take into account only those viscous and conductive terms containing derivatives in the direction perpendicular to the body surface. Terms containing derivatives parallel to the surface are dropped. Specifically, the thin-layer approach assumes that $\partial/\partial\xi = 0$. The TLNS equations are thus reduced to,

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial \xi} + \frac{\partial F}{\partial \eta} = H' + \frac{\partial S'}{\partial \eta} \tag{13}$$

where the vectors H′ and S′ are given by,

$$H' = \frac{1}{J} \left\{ \begin{matrix} 0 \\ -\frac{2}{3}\ \eta_x \frac{\partial}{\partial \eta}\ (\mu v) \\ \frac{2}{3}\ \eta_x\ \mu \frac{\partial u}{\partial \eta} - \frac{2}{3}\ \eta_y v \frac{\partial \mu}{\partial \eta} + p - \frac{4}{3}\ \mu\ \frac{v}{y} \\ -\frac{2}{3}\ \eta_x \frac{\partial}{\partial \eta}\ (\mu u v) - \frac{2}{3}\ \eta_y \frac{\partial}{\partial \eta}\ (\mu v^2) \end{matrix} \right\} \tag{13a}$$

and

$$S' = \frac{y}{J} \left\{ \begin{matrix} 0 \\ \mu\alpha_1 \frac{\partial u}{\partial \eta} + \mu\alpha_2 \frac{\partial v}{\partial \eta} \\ \mu\alpha_3 \frac{\partial v}{\partial \eta} + \mu\alpha_2 \frac{\partial u}{\partial \eta} \\ \frac{1}{2}\ \mu\alpha_1 \frac{\partial u^2}{\partial \eta} + \mu\alpha_2 \frac{\partial (uv)}{\partial \eta} + \frac{1}{2}\ \mu\alpha_3 \frac{\partial v^2}{\partial \eta} + K\alpha_4 \frac{\partial T}{\partial \eta} \end{matrix} \right\} \tag{14}$$

where,

$$\alpha_1 = \frac{4}{3}\ \eta_x^{\,2} + \eta_y^{\,2}; \quad \alpha_2 = \frac{1}{3}\ \eta_x \eta_y; \quad \alpha_3 = \eta_x^{\,2} + \frac{4}{3}\ \eta_y^{\,2}; \quad \alpha_4 = \eta_x^{\,2} + \eta_y^{\,2}$$

The viscous term, S′, can be further rearranged as,

$$\frac{\partial S'}{\partial \eta} = \frac{\partial}{\partial \eta}\left(R_1 \frac{\partial Q_1}{\partial \eta} + R_2 \frac{\partial Q_2}{\partial \eta}\right) \tag{15}$$

in which R_1 and R_2 are 4 x 4 matrices given by,

$$R_1 = \mu \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \alpha_1 & \alpha_2 & 0 \\ 0 & \alpha_2 & \alpha_3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad R_2 = \mu \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{k}{\mu}\alpha_4 & \frac{\alpha_2}{2} & \frac{\alpha_3}{2} & \alpha_2 \end{pmatrix} \tag{16}$$

The vectors, Q_1 and Q_2, are defined by,

$$Q_1 = (\rho,\ u,\ v,\ e_o)^T, \quad Q_2 = (T,\ u^2,\ v^2,\ uv)^T \tag{17}$$

In this form, the viscous dissipation in the energy equation is separated from the remaining viscous terms and the matrices R_1 and R_2 contain only the viscosity and thermal conductivity and the metrics of the transformation.

Important Relations For The Real Gas Equations

The boundary layers in rocket nozzles can be either laminar or turbulent, and frequently it is difficult to tell what the nature of the flow will be. The favorable pressure gradient, the wall cooling and the supersonic character of the flow all tend to stabilize the boundary layer and favor laminar flow, but for high Reynolds numbers (such as occur in large thrust systems), and high expansion ratio nozzles (which are typically quite long), the boundary layers are generally turbulent. Modeling turbulent flows is never an easy assignment, but for these supersonic, strongly accelerating, highly cooled boundary layers with chemical reaction and heat release, it is safe to speculate that existing turbulence models which have been developed and calibrated for incompressible flows with weak pressure gradients will be less realistic than in their normal, more benign, environments.

Consequently, the use of complex turbulence models which are interested in the present regime is hardly justifiable. Accordingly, we have used the simple Baldwin-Lomax two-layer mixing length model, along with a constant turbulent Prandtl number assumption. This approach gives flowfield profiles which are representative of turbulence conditions and verifies that the codes continue to work in the turbulent flow environment. Incorporation of more advances turbulence models can readily be incorporated if and when they become available.

NUMERICAL SOLUTION PROCEDURES

Although the fluid dynamic equations are formulated in the complete unsteady sense, only steady state solutions are considered. These solutions are obtained by applying time-dependent schemes to the Euler and TLNS equations. Implicit time-marching methods are used throughout.

The spatial discretization that is chosen for the axisymmetric problem uses central differencing in the cross-stream or radial (η) direction, and second order upwind differencing in the streamwise or axial (ξ) direction. For the inviscid supersonic case, this upwind-central differencing combination nicely mimics the physics of the flow. For the parabolized case, it also mimics the type of differencing that is typically chosen for PNS algorithms. The extension of this hybrid differencing scheme to the TLNS equations is also straightforward if we interpret the upwind differencing in a flux split sense.

With this choice of differencing, both the supersonic inviscid case and the PNS formulation can be solved by a single sweep through the flowfield. The TLNS formulation then becomes a fully iterative (forward-backward sweeps) solution procedure for which the PNS solution is just the forward sweep.

Details of this discretization and solution method are given in the present section fdor the real gas formulation. Similar developments for the perfect gas case are given in Appendix D. Because the flux vectors in the perfect gas formulation are homogeneous functions of the dependent variable, Q, flux vector splitting, rather than flux difference splitting, is used in Appendix D. The flux difference method described here is necessary for real gases where the flux vectors are non-homogeneous, but should perform as well as flux vector splitting in the perfect gas case. Consequently, our recommendation is to use flux-difference splitting throughout rather than only in the real gas case.

We begin by considering the thin-layer Navier-Stokes equations and then drop viscous effects to obtain an Euler equation procedure.

Discretization Of The Thin-Layer Navier-Stokes Equations

We begin by discretizing the TLNS equations (Eqn. 12) in time using Euler implicit differencing in time,

$$\frac{Q^{n+1} - Q^{n}}{\Delta t} + \left(\frac{\partial E}{\partial \xi} + \frac{\partial F}{\partial \eta} - \frac{\partial}{\partial \eta} \left(R_1 \frac{\partial Q_1}{\partial \eta} + R_2 \frac{\partial Q_2}{\partial \eta}\right) - H'\right)^{n+1} = 0 \qquad (18)$$

where n and n+1 refer to the previous and current time levels, respectively. We then express Eqn. 18 in delta form,

$$\Delta Q + \Delta t(\frac{\partial}{\partial \xi}\Delta E + \frac{\partial}{\partial \eta}\Delta F - \frac{\partial}{\partial \eta} \Delta S' - \Delta H = -\Delta t\left(\frac{\partial E}{\partial \xi} + \frac{\partial F}{\partial \eta} - \frac{\partial S'}{\partial \eta} - H'\right)^{n} \qquad (19)$$

where $\Delta Q = Q^{n+1} - Q^{n}$, etc.

The spatial discretization of the right-hand side of Eqn. 19 is accomplished as follows,

$$\frac{\partial E}{\partial \xi} = \frac{\tilde{E}_{i+1/2j} - \tilde{E}_{i-1/2j}}{\Delta\xi} \qquad \frac{\partial F}{\partial \eta} = \frac{\tilde{F}_{ij+1/2} - \tilde{F}_{ij-1/2}}{\Delta\eta}$$

and,

$$\frac{\partial}{\partial\eta} R_1 \frac{\partial Q_1}{\partial\eta} = \left[(R_1 \frac{\partial Q_1}{\partial\eta})_{ij+1/2} - (R_1 \frac{\partial Q_1}{\partial\eta})_{ij-1/2} \right] \frac{1}{\Delta\eta} \qquad (20)$$

where the tilde denotes the numerical flux of the indicated vectors. These quantities are identified below.

For the streamwise vector, E, the discretization is defined by,

$$\tilde{E}_{i+1/2j} = \frac{1}{2} [E_{ij} + E_{i+1j} - \Delta E^{+}_{i+1/2j} + \Delta E^{-}_{i+1/2j} + \Delta E^{+}_{i-1/2j} - \Delta E^{-}_{i+3/2j}] \qquad (21)$$

whereby a Taylors series in ξ, we define,

$$\Delta E^{\pm}_{i+1/2j} = A^{\pm}(\bar{Q})_{i+1/2j} (Q_{i+1j} - Q_{ij})$$

where A^+ and A^- are the eigenvalue split forms of A. When Eqn. 21 is substituted into the discretized form of $\partial E/\partial\xi$ defined in Eqn. 20, this gives a flux split second order upwind differenced scheme.

The splitting of matrix A is determined on the basis of the eigenvalues of A. Thus, we define,

$$A = A^+ + A^- \qquad (22a)$$

where the eigenvalues of A^+ are positive and those of A^- are negative. We define this matrix by diagonalizing A by the similarity transformation,

$$\Lambda_A = T^{-1}P^{-1}APT$$

where,

$$\Lambda_A = \text{diag } (U, U, U+C, U-C) \qquad (23)$$

Here, U is the contravariant velocity, $U = \xi_x u + \xi_y v$, and C is the transformed speed of sound, $C = c(\xi_x^2 + \xi_y^2)^{1/2}$, where c is the physical speed of sound.

The split eigenvalues are defined by $\lambda_i^{\pm} = (\lambda_i \pm \lambda_i)/2$ where λ_i are the individual eigenvalues in Eqn. 23. As can be seen by inspection, this choice leads to the definition of two diagonal eigenvalues, Λ_A^+ and Λ_A^- where all eigenvalues of Λ_A^+ are either positive or zero and those of Λ_A^- are negative or zero. From these split diagonal matrices, we can define the splitting of A as,

$$A^+ = TP\Lambda_A^+ P^{-1}T^{-1}$$

$$A^- = TP\Lambda_A^- P^{-1}T^{-1} \qquad (24)$$

and by inspection we see $A^+ + A^- = A$. The matrix PT is given in Appendix A.

Thus far, we have demonstrated the method of splitting A and its upwind differencing. We now return to Eqn. 20 to define the spatial differencing in η. Because we use central differencing in η, we define,

$$\tilde{F}_{ij+1/2} = 1/2(F_{ij+1} + F_{ij}) \qquad (25)$$

which leads to the standard three point differencing. The differencing of the two viscous term is given as,

$$\frac{\partial}{\partial \eta} R_1 \frac{\partial Q_1}{\partial \eta} = R_{1_{ij+1/2}} \left(\frac{Q_{ij+1} - Q_{ij}}{\Delta\eta^2}\right) - R_{1_{ij-1/2}} \left(\frac{Q_{ij} - Q_{ij-1}}{\Delta\eta^2}\right) \qquad (26)$$

The differencing of the first order derivations in H' are done in a manner identical to that for $\partial F/\partial \eta$. The derivatives on the left-hand side of Eqn. 19 are accomplished with the aid of a Taylor's series in time. For the vector, E, we have,

$$E^{n+1} = E^n + \frac{\partial E}{\partial Q} \Delta Q \qquad (27)$$

so that,

$$\Delta E = E^{n+1} - E^n = \frac{\partial E}{\partial Q} \Delta Q \qquad (28)$$

Similar results can be obtained for the vectors F, Q_1, Q_2, and H'. The notation for the Jacobians of E and F has already been introduced (as A and B, respectively). We also define the Jacobians,

$$D = \frac{\partial H'}{\partial Q}\ , \quad B_{v1} = \frac{\partial Q_1}{\partial Q}\ , \quad B_{v2} = \frac{\partial Q_2}{\partial Q} \qquad (29)$$

These matrices are given in Appendix A. We also use the flux split matrices A^+ and A^- for the streamwise derivatives. Incorporating these Taylor's series expansions on the left-hand side and the previously defined differences for the right-hand side, the fully discretized equation becomes,

$$[I = \Delta t(\frac{\partial}{\partial \xi}(A^+ + A^-) + \frac{\partial}{\partial \eta} B - \frac{\partial}{\partial \eta} R_1 \frac{\partial}{\partial \eta} B_{v1} - \frac{\partial}{\partial \eta} R_2 \frac{\partial}{\partial \eta} B_{v2}$$

$$- D] = -\Delta t\left[\frac{\partial E}{\partial \xi} + \frac{\partial F}{\partial \eta} - \frac{\partial}{\partial \eta} R_1 \frac{\partial Q_1}{\partial \eta} - \frac{\partial}{\partial \eta} R_2 \frac{\partial Q_2}{\partial \eta} - H'\right]^n \qquad (30)$$

For simplicity of notation, we express,

$$D = D' + D'' \qquad (31)$$

where D' represents the Jacobian of the algebraic terms in H' (Eqn. 13) and D'' represents the Jacobian of the first derivative terms in H'. These matrices are defined in Appendix A. We then also express the η derivatives on the left-hand side of Eqn. 30 by B_T (for total derivative) so that,

$$\frac{\partial}{\partial\eta} B_T\Delta Q = [\frac{\partial}{\partial\eta}(B - R_1 \frac{\partial}{\partial\eta} B_{v1} - R_2 \frac{\partial}{\partial\eta} B_{v2}) - D'']\Delta Q \tag{32}$$

and again use the S′ notation for the viscous terms on the right-hand side of Eqn. 30 (see Eqn. 15). We can then simplify Eqn. 30 as,

$$[I - \Delta tD' + \Delta t \frac{\partial}{\partial\xi}(A^+ + A^-) + \Delta t \frac{\partial}{\partial\eta} B_T]\Delta Q = -\Delta t R_{es} \tag{33}$$

where R_{es} is the residual evaluated at time level n,

$$R_{es} = [\frac{\partial E}{\partial\xi} + \frac{\partial F}{\partial\eta} - \frac{\partial S'}{\partial\eta} - H']^n \tag{34}$$

Here, it is assumed that the right-hand side of Eqn. 33 (or Eqn. 30) is evaluated according to Eqn. 20 et seq. We also note that for central differencing in η, the term incorporating B_T will contain both first and second derivatives and will include quantities evaluated at j+1, j, and j-1. The numerical solution of Eqn. 33 is described in the following subsection.

Numerical Solution Of The Thin-Layer Navier-Stokes Equations

The direct inversion of the matrix on the left-hand side of Eqn. 33 contains unknowns from the five different grid-lines in the ξ direction and from three in the η direction. In particular, the derivative $\partial(A^+\Delta Q)/\partial\xi$ introduces unknowns at i, i-1 and i-2, while $\partial(A^-\Delta Q/\partial\xi)$ introduces unknowns at i, i+1 and i+2. This results in a matrix which is quite costly to invert by direct means. Consequently, we look to approximately factored procedures to solve Eqn. 33. For this purpose, we express the equations in fully discretized form and factor using the parabolized Navier-Stokes alternating direction implicit (PNS-ADI) procedure.

We can approximately factor the discretized version of the left-hand side and rewrite Eqn 33 as, approximately:

$$[\Gamma - \frac{\Delta t}{2\Delta x}(4A^+_{i-1} - A^+_{i-2}) + \frac{\partial}{\partial\eta} B_T]\Gamma^{-1}[\Gamma + \frac{\Delta t}{2\Delta x} (4A^-_{i+1} - A^-_{i+2}]\Delta Q \simeq -\Delta tR_{es} \tag{35}$$

where the matrix Γ contains all terms on the diagonal,

$$\Gamma = I - \Delta tD' + \frac{3\Delta t}{2\Delta x} (A^+_i - A^-_i) \tag{36}$$

By performing the indicated multiplication on the left-hand side of Eqn. 35, we see that Eqn. 35 is equal to Eqn. 33 except for terms of order Δt^2.

The solution of Eqn. 35 proceeds directly. The first operator is tridiagonal in η with two lower diagonals corresponding to $A^+{}_{i-1}$ and $A^+{}_{i-2}$. This operator can be solved by a sweep in the streamwise direction which requires a block tridiagonal matrix solution. The second operator in brackets is lower (block) diagonal and can be solved by marching from the downstream to the upstream end of the nozzle. Thus, Eqn. 35 represents a forward-backward iteration procedure. This iteration proceeds until ΔQ is driven to zero, or to an acceptable tolerance. As in the perfect gas formulation (see Appendix C), this approximate factorization procedure is very efficient for predominantly supersonic flows.

Solution Of The Euler Equations

The description above is also applicable to the Euler equations, except for several simplifications. First of all, the viscous terms are omitted (μ and k set to zero) in Eqns. 13 and 14 so the vector H' simplifies to only one term and the viscous flux S' vanishes. Similarly, the matrix B_T in Eqn. 33 (see Eqn. 32) simplifies to $B_T = B$, and we have a forward-backward iteration procedure for inviscid flow.

In the special case where the flow is entirely supersonic, the matrix A^- becomes identically zero, and A^+ becomes equal to A. In this case, the backward sweep degenerates to an identity operator, and Eqn. 35 reduces to:

$$[\Gamma - \frac{\Delta t}{2\Delta x}(4A^+_{i-1} - A^+_{i-2}) + \frac{\partial}{\partial \eta} B]\,\Delta Q = -\Delta t\, R_{es} \qquad (37)$$

where Γ and R_{es} are simplified as noted. As in Appendix D, it is recommended that each line be time-marched till ΔQ goes to zero rather than sweeping the entire field. This then results in an iterative space-marching solution for supersonic flows.

Solution Of The Parabolized Navier-Stokes Equations

The parabolized Navier-Stokes (PNS) procedure for supersonic flow is based upon removing the terms in the equations that allow upstream propagation of information. This involves two steps. The first step involves removing viscous diffusion in the streamwise direction which has already been done in the TLNS equations. The second step involves omitting the upstream propagation of information through the subsonic portion of the boundary layer. These approximations make the equations parabolic and allow space marching procedures to be used.

The present parabolized procedure is obtained by setting the matrix A^- to zero in the TLNS formulation of Eqn. 35. By omitting A^-, the back sweep of Eqn. 35 simplifies to the identity matrix and the remaining algorithm corresponds exactly to the forward sweep of the full TLNS procedure. In omitting A^-, we drop all upstream running characteristics and thus obtain a well-posed space marching procedure. The resulting numerical algorithm becomes identical to the inviscid procedure for the supersonic Euler equations except that the viscous terms are retained. (Recall the A^- operator is identically zero in supersonic flow.) Setting A^- to zero and eliminating the upstream operator is completely analogous to multiplying the pressure gradient by a value of ω less than one and ignoring the $(1-\omega)\partial p/\partial \xi$ term as is done in standard PNS procedures[5]. The advantage of the present procedure is that it is more firmly founded from a physical viewpoint.

Mathematically, the PNS algorithm is expressed as,

$$[\Gamma - \frac{\Delta t}{2\Delta x} (4A^+_{i-1} - A^+_{i-2}) + \frac{\partial}{\partial\eta} B_T]\Delta q = \Delta t \; R_{es} \qquad (38)$$

where Γ is as defined in Eqn. 36 except that A^- is zero, and B_T still retains its full viscous form of Eqn. 32. As for the supersonic inviscid flow calculation, iterations at each line are converged before moving downstream so that the converged solution is obtained in a single sweep.

Euler Equation Boundary Conditions

Boundary conditions are specified in exactly the same method as described in Appendix C. We begin with boundary conditions for the Euler equations. For the Euler equations, we rely on the Method of Characteristics to ensure we enforce both the correct number of conditions and types of conditions that are physically and mathematically acceptable. In general, this means that the four unknowns (corresponding to the four equations in two-dimensional flow) are obtained from a combination of boundary conditions and specific subsets of the equations of motion.

At a subsonic inflow boundary, we specify three quantities as boundary conditions and obtain the last from the equations of motion. For supersonic inflow, we specify all four quantities. Since we are primarily interested in supersonic nozzle flows which are started downstream of the throat of a converging-diverging nozzle, this latter choice is the one of primary interest here. Consequently, we give no details as to the MOC procedure for subsonic inflow. These can, however, be obtained from an extension of the MOC procedures for wall boundary conditions, or from Appendix C. In the case of supersonic inflow, the necessary two lines of information are obtained from the transonic solution (see Appendix C).

At the outflow boundary, the solution of Euler equations for a diverging nozzle is wholly supersonic. Therefore, all the information at the downstream end is determined from the flowfield itself. No downstream boundary condition is needed or allowed. On the axis of symmetry, y = 0, therefore,

$$Q = 0 \qquad (39)$$

because y is contained inside Q (see Eqn. 1). After Q is determined, the primitive variables are determined by a second order extrapolation. If f denotes a primitive variable, then on the axis $\partial f/\partial y = 0$ and,

$$f_{i,1} = \frac{\xi_y f_{i-1} - \frac{1}{2}\eta_y \; (4f_{1,2} - f_{i,3})}{\xi_y + 3/2\eta_y} \qquad (40)$$

Since v = 0 on the axis, there are only three primitive variables to be extrapolated. The real gas variables are $f = \rho$, u, e_o.

For inviscid flows, slip wall boundary conditions are enforced by means of the Methods of Characteristics and auxiliary information corresponding to in-running characteristics.

From an eigenvalue analysis of the Jacobian, $B = \partial F/\partial \eta$, one gets,

$$\Lambda_B = \text{diag}\ (V,\ V,\ V+C,\ V-C)$$

where V is the contravariant velocity component, $V = \eta_x u + \eta_y v$, and C is the transformed speed of sound, $C = c(\eta_x^2 + \eta_y^2)^{1/2}$.

On a wall, the first three eigenvalues are positive or zero and the fourth is negative. Accordingly, three quantities on the boundary are enforced by outgoing characteristic relations and the fourth is obtained from the specified boundary condition, V = 0 which forces the flow to be tangent to the wall.

The three characteristic equations are derived from the discretized Euler equations (Eqns. 30) by premultiplying both the left- and right-hand sides by the product of a selection matrix, L, and an eigenvector matrix, $T_B^{-1}P^{-1}$, where the matrix L, L = diag (1, 1, 1, 0), acts as a selection operator that selects only the information (equations) corresponding to the outgoing characteristics. The modal matrix, $T_B^{-1}P^{-1}$ is given in Appendix A.

The single boundary condition which combines with the three characteristic relations given above can be expressed in vector form as,

$$\Omega(Q) = [0,\ 0,\ 0,\ V]^T = 0.$$

In order to express this boundary condition in delta form, we expand Ω in a Taylor's series,

$$\Omega^{n+1} = \Omega^n + \frac{\partial \Omega}{\partial Q}\,\Delta Q = 0 \qquad (41)$$

where $\partial\Omega/\partial Q$ is a Jacobian matrix.

By combining the one relation in Eqn. 41 with the three conditions in Eqn. 40, we obtain four coupled equations which determine the four unknowns on the nozzle wall.

Viscous Boundary Conditions For The TLNS Equations

For the thin-layer Navier-Stokes equations, two boundary conditions, namely those at the upstream end and those along the axis, are identical to those applied to the Euler equations. Additional detail is not given here. Both the outflow boundary conditions and the solid wall boundary conditions are different. The outflow conditions are still treated in an inviscid manner by the Method of Characteristics, but because of the nozzle wall boundary layer, portions of the flow at the nozzle exit are subsonic, so both subsonic and supersonic conditions are needed there. At the wall, the viscous no slip conditions are specified. Details of these implementations are given here.

The flow at the downstream boundary consists of two parts -- a major supersonic part extending from the axis to the vicinity of the wall, and a small subsonic part adjacent to the wall corresponding to the subsonic part of the boundary layer.

Quantities on these two parts of the boundary line are determined differently because of the different nature of supersonic and subsonic flows. The fourth eigenvalue of the Jacobian, $A = \partial E/\partial Q$, changes sign from positive to negative as the flow changes from supersonic to subsonic.

From an eigenvalue analysis of the Jacobian A, one gets,

$$\Omega_A = \text{diag}\ (U,\ U,\ U{+}C,\ U{-}C)$$

where U is the contravariant velocity component, $U = \xi_x u + \xi_y v$, and C is the transformed speed of sound, $C = R(\xi_x^2 + \xi_y^2)^{1/2}$. In the supersonic region, all eigenvalues are positive. Therefore, the boundary conditions are determined by the outgoing flow which is solved by the forward sweep step. In the subsonic region, the first three eigenvalues are positive and the fourth is negative. This means that three of the four unknowns are determined by the outgoing characteristics while the fourth must be imposed as a boundary condition.

The three characteristic equations are derived from the discretized TLNS equations (Eqn. 33) upon premultiplying by $LT_A^{-1}P^{-1}$, in the same manner as described for the inviscid wall boundary condition. The difference is that here the selection matrix is given by L = diag (1, 1, 1, 0) so that only the information (equations) corresponding to the outgoing characteristics is retained. A second difference is that the modal matrix, $T_A^{-1}P^{-1}$ (given in Appendix A) is the one corresponding to the Jacobian A.

The boundary condition that is used to augment these three Method of Characteristic relations is a specification of the nozzle back pressure, p_b, over those portions of the flow that are subsonic. In vector notation, this becomes,

$$\Omega(Q) = [0,\ 0,\ 0,\ p - p_b]^T = 0 \tag{42}$$

Then, taking the Jacobian of d with respect to Q, we obtain as before,

$$\frac{\partial \Omega}{\partial Q}\ \Delta Q = p_b - p \tag{43}$$

The right-hand side of Eqn. 43 can either be left as is to prevent small drifts of the pressure at the exit plane, or can be set to zero as Eqn. 42 suggests.

In addition to this mathematically and physically correct treatment of the subsonic portion of the outflow, we also use a second option of extrapolating the subsonic pressure from inside the flowfield. This second procedure is not correct mathematically, but it is frequently imposed in problems of this type. Its success rests with the smallness of the subsonic region. Choice of the extrapolation condition eliminates the possibility of determining the effect of perturbations in the back pressure on nozzle performance, an issue that is sometimes of considerable interest.

To apply the extrapolated pressure condition, we use the three characteristic relations along with a first order extrapolation of pressure, namely,

$$p_{I,j}^{n+1} = p_{I-1,j}^{n+1} \qquad (44)$$

where I is the last grid line. In vector form, this is expressed as,

$$\Omega_I - \Omega_{I-1} = 0 \qquad (45)$$

where,

$$\Omega_i = (0,\ 0,\ 0,\ p_{ij})^T \qquad (46)$$

Finally, we note that in using the specified boundary condition, p_B, only minor deviations from the ideally expanded nozzle can be accommodated. A typical procedure is to first determine the "ideally expanded" back pressure by the extrapolation condition. Then other back pressures in the neighborhood of this condition can be specified. Generally speaking, the back pressure can only be varied by a factor of two or three from the value determined by the extrapolation procedure. This variation is, however, enough to show a decided thinning of the subsonic portion of the boundary layer when the back pressure is lowered, as well as a thickening of the subsonic region and the establishment of a small recirculating region when the back pressure is increased. When re-entry flow is encountered, inflow conditions must be added for those points on the exit boundary that experience inflow. In general, the procedure fails when the inflow region becomes large enough that the point of demarcation between inflow and outflow begins to jump back and forth during the iteration. To go beyond these limits, it is necessary to expand the flow domain to extend outside the nozzle. Nevertheless, the present procedure can be very effective in understanding the physics of non-ideally expanded, viscous supersonic flows and in assessing performance shifts when the nozzle back pressure is nearly, but not exactly, matched to the flow.

For the viscous wall boundary condition, we specify three boundary conditions. The no-slip boundary condition implies that both components of the contravariant velocity are zero, $u = V = 0$. The third boundary condition is one on the energy flux at the wall. This can either take the form of a specified wall temperature, $(T_w = T_w(\xi)$, or a specified heat flux, $\partial T/\partial y = f(\xi)$, where f is an arbitrary function.

The fourth condition for the viscous wall boundary condition comes by solving the radial momentum equation for the normal pressure gradient at the wall, and enforcing this momentum equation in conjunction with the three boundary conditions. The normal gradient of the pressure in a non-orthogonal coordinate system is given by,

$$n \bullet \nabla p = (\eta_x \xi_x - \eta_y \xi_y)\ p_\xi + (\eta_x{}^2 + \eta_y{}^2)\ p_\eta \qquad (47)$$

where n is the unit normal to the wall. An analogous relation is used for specifying the temperature gradient when the heat flux boundary condition is imposed. In differencing Eqn. 47, second order accurate one-sided differences are used in η and central differences are used in ξ.

Boundary Conditions For The PNS Procedure

Boundary conditions for the PNS procedure are identical to those for the TLNS equations except that no downstream conditions are allowed and no downstream information is allowed in computing normal derivatives on the walls. When normal derivatives of pressure and temperature are obtained, similar procedures are used except the ξ derivatives are evaluated by a second-order backward difference.

Computing The Rate Of Mass Flow And Thrust

For an axisymmetric nozzle, the conservation of mass flow is verified by integrating along arbitrary cross-sections at various streamwise positions. With no mass addition, the rate of mass flow must be equal at each of these sections. The nozzle thrust is determined by integrating across the cross-section at the exit plane.

For simplicity, cross-sections along constant values of ξ are chosen to verify the conservation of mass flow and to compute the thrust.

The rate of mass flow, $d\dot{m}$, through an elemental surface, ds, is given by,

$$d\dot{m} = \rho U_\xi ds \tag{48}$$

where,

$$U_\xi = \frac{U}{(\xi_x{}^2 + \xi_y{}^2)^{1/2}} = \frac{\xi_x u + \xi_y v}{(\xi_x{}^2 + \xi_y{}^2)^{1/2}} \tag{49}$$

is the component of the velocity vector normal to the surface, $\xi = \xi(x,y)$, and $ds = 2\pi y dy$, is the elemental area at a distance y from the axis.

In Cartesian notation, $d\eta = ((dx)^2 + (dy)^2)^{1/2}$, and the rate of mass flow is given by,

$$d\dot{m} = 2\pi(\rho uydy - \rho vydx) \tag{50}$$

The integration of $d\dot{m}$ is performed numerically between j=1 and j=JL for each i using Simpson's rule.

In the real gas TLNS equations, the fluxes are defined at the midpoint $\tilde{E}_{i+1/2j}$ (Eqn. 25). Then, for second order accuracy (the index j is omitted for simplicity),

$$(\rho U_{\xi})_{i+\frac{1}{2}} = \frac{1}{2}\,[(\rho U_{\xi})_{i} + (\rho U_{\xi})_{i+1}]$$

$$-\ \frac{1}{2}\,[\Delta|\,(\rho U_{\xi})|_{\,i+\frac{1}{2}} - \Delta(\rho U_{\xi})^{+}_{i-\frac{1}{2}} + \Delta(\rho U)^{-}_{i+3/2}] \qquad (51)$$

The differential thrust, dt, in the axial direction, is given by,

$$d_{T} = d\dot{m}u + (P - P_{b})\ 2\pi y dy \qquad (52)$$

where P_b is the prescribed pressure at the exit.

USERS MANUAL FOR THE CODES

Introduction

The following discussion is intended to be an overview that explains briefly how to run the codes. A total of four different codes are described that solve the perfect or real gas flow for the Euler or thin-layer Navier-Stokes equations in an axisymmetric nozzle. In particular, codes AXI2DSA, AXI2DSE and AXI2DSF solve the Euler equations, while the codes DDADIPBC, DDADIPBE and DDADIPBG solve the TLNS equations.

The perfect gas flow is solved by the codes AXI2DSA, AXI2DSE, DDADIPBC, and DDADIPBE. The real gas flow is solved by the codes AXI2DSG and DDADIPBG. The real gas codes are demonstrated for equilibrium air flow, however, by changing the input data and the relevant data subroutines, the user is able to solve any other equilibrium real gas flow. These modifications are explained later in this section.

The Euler solver, AXI2DSA, and the TLNS solver, DDADIPBC, are designed to solve perfect gas flows. In these codes, the molecular weight, M_w, is constant, and the internal energy, e, is given by,

$$e = c_v T \tag{53}$$

where c_v, the specific heat coefficient in constant volume, is constant, and T is the temperature. In the TLNS solvers the laminar viscosity coefficient, μ, may be defined by one of the following three options:

a) constant value, $\mu = \mu_o$ (54a)

b) Sutherland law,

$$\mu = \mu_o \frac{T_o + 196}{T + 196} \left(\frac{T}{T_o}\right)^{1.5} \tag{54b}$$

c) Power law,

$$\mu = \mu_o \left(\frac{T}{T_o}\right)^{\omega} \tag{54c}$$

where T_o and μ_o are reference values and ω is a constant ranging between $0.5 \leq \omega \leq 1.0$.

The Euler code AXI2DSC and the TLNS code DDADIPBG are the most generalized solvers. In these codes, the molecular weight, M_w, and the internal energy, e, are defined as functions of two variables.

$$M_w = M_w\ (\rho, T) \tag{55}$$

and

$$e = e(\rho,T),\ \text{or}\ T = T(\rho,e) \tag{56}$$

In literature these relations appear as tables[6,7]. The viscosity and thermal conductivity coefficients, μ and k, are also specified as functions of two variables,

$$\mu = \mu(\rho, e)$$
$$k = k(\rho, e)$$

in the real gas TLNS code. Simplified curve fits for μ and k for air were suggested by Srinivasan et al.[8].

In order to run the codes, the user must furnish the transformed coordinates for the flowfield and define the gas properties. The real gas properties, excluding the viscosity and thermal conductivity, are read into the Euler and TLNS codes. The viscosity and thermal conductivity data are generated by the TLNS codes during the computation process.

In the Euler codes, AXI2DSA and AXI2DSG, the flowfield initial guess at each i and i = 1,2,3..., JL is obtained from the previous converged solution at i-1. Since the Euler solution converges relatively rapidly, it is usually obtained in a single run. Consequently, the constants that monitor the number of iterations and the restart for the next run, although defined in namelist INPUT, are not in use.

In the TLNS codes, DDADIPBC and DDADIPBG, the flowfield solution is obtained by alternate forward and backward sweeps. The flowfield initial guess is evaluated by two ways. In DDADIPBC, the initial guess is obtained by a PNS single-sweep procedure which is an integrated part of the code. In DDADIPBG, the flowfield initial guess is derived from the isentropic relations throughout the flow domain. The above-described PNS initial condition represents a cost savings of two or more and its implementation in this code is recommended.

The iterative solution of the TLNS codes requires a large amount of computer time. Therefore, the user should be cautious in defining the constants that monitoring the number of iterations and continuation of the next runs.

Table 2 presents a summary of the codes' executing data, which includes the number of mesh points, the recommended number of iterations, and the computing time per iteration per mesh point on the NOS-BE system. In this table, the number of mesh points denotes the maximum amount allowed by the core-memory of the CYBER 180/840, NOS-BE system. As indicated before, these stringent limitations are removed when running under NOS-VE. The converged solution is obtained when the average relative error, Δ, defined as,

$$\Delta = \frac{1}{JL \bullet IL} \sum_{j=1}^{JL} \sum_{i=1}^{IL} \left| \frac{\Delta Q_{ilj}}{Q_{ilj}} \right| \qquad (57)$$

is less than a specified value. In the present runs, this was set at $\Delta = 10^{-18}$ for the Euler codes and $\Delta = 10^{-6}$ for the TLNS codes.

The units used in the codes are the International System of Units (SI units). These SI units are as given in Table 3. The universal gas constant that is consistent with these units is R = 8314.3 J/(kgmole)•K.

In the energy equation, the units of the internal energy, e, are given by $(m/sec)^2$.

Additional Information Concerning Real Gas Flows

Calculations of real gas functions require the molecular weight, the temperature, the viscosity and thermal conductivity and several derivatives of these quantities to be known as functions of two thermodynamic variables. As primary dependent variables, we choose the properties density, ρ, and internal energy, e. A number of FUNCTION subroutines have been written to obtain these quantities from tabular input data. These functions are described in Table 4. In addition to these Function Subprograms, subroutines UGAS3 and UGAS4 are called by Functions FZMU and FZK.

The Perfect Gas Euler Code, AXI2DSA

The code AXI2DSA solves the Euler equations for a perfect gas flow. The present version has all dimensional arrays set to (70 x 44), i.e., 70 points in the ξ direction and 44 points in the η direction. These are the maximum grid points allowed by the CYBER NOS-BE memory. These limitations can easily be reset by a recompilation of the code. Only the DIMENSION statements need be changed, all other FORTRAN statements will execute properly if these numbers are increased.

Input Data Description

1. Namelist INPUT. Namelist INPUT specifies the flow constants. These constants are read from file TAPE1 by subroutine INITIA. The quantities in this NAMELIST are as follows:

IL - Number of mesh points in the ξ direction (Integer).
JL - Number of mesh points in the η direction (Integer).
ITR1 - Not used; set to 1.
ITRN - Not used; set to 1.
ITRS - = ITRN - ITR1 + 1 (Integer).
ISVP - Not used; set to 0.
CFL - Courant-Friedrichs-Levy number (typical ranges 100-500).
ITIME - Time step control (Integer):
ITIME = 0 for constant Δt;
ITIME = 1 for constant CFL (normal value).
THETA - Crank-Nicholson constant, always set to 1.0.
RM1 - The Mach number at the throat, RM1 > 1.0.
RM2 - The exit Mach number; approximate value, only used for set-up.
AIN - Not used.
AEX - Not used.
RL - Not used.
PO - Upstream stagnation pressure, n/m^2 (uniform across the stream).
TO - Upstream stagnation temperature, K (uniform across the stream).
CP - Specific heat in a constant pressure (constant).

CV - Specific heat in a constant volume (constant).

GAMMA - Ratio of specific heats, CP/CV.

GM1 - (GAMMA-1).

INER - Maximum Number of iterations for a converged solution (Integer).

IB1 - Grid line in the ξ direction at which the iterative process starts. Usually IB1 = 2 (Integer).

IBL - Grid line in the u direction at which the iterative process ends. Usually IBL = IL (Integer).

NORD - Control variable for specifying the order of accuracy in the ξ direction (Integer);
NORD = 0 - first order accurate
NORD = 1 - second order accurate (normal value).

OMEGAX- Artificial dissipation in the ξ direction. (Real variable, generally less than 0.5).

OMEGAY- Artificial dissipation in η direction. (Real variable generally less than 0.5).

IREAD - Not used; set to 0.

IWRT - Output print control (Integer):
IWRT = 0 - Brief printout
IWRT = 1 - Full printout

IRVN - Not used; set to 1.

2. Mesh Points Coordinates. The flowfield mesh points coordinates, X(I,J), Y(C,J) for I=1,2,3,...,IL, J=1,2,3,...,JL, are read from file TAPE3 by subroutine INITIA. In this input data the inflow boundary conditions are defined on I=1, J=1,2,3,...,JL, the outflow on I=IL, J=1,2,3,...,TL, the axis of symmetry on J=1, I=1,2,3,...,IL, and the wall on J=JL, I=1,2,3,...,IL. The reading format is,

```
      READ (3,401)((X(I,J), Y(I,J), I=1,IL), J=1,JL)
  501 FORMAT (E17.9, 4E16.9)
```

Output Data Description

1. Printed Results. For the short form printout, IWRT is set to 0 in namelist INPUT. With this printout choice, the convergence rate for each equation from subroutine SUPPLY (ENTRY CHECK), and the downstream rate of mass flow from subroutine SUPPLY (ENTRY MASS) is printed at each iteration.

When IWRT=1 a detailed printout of the flowfield results is printed by subroutine SUPPLY (ENTRY OUTPUT). For each mesh point (I,J) it includes the coordinates, (X,Y), the velocity components (u,v), the pressure, density, temperature, stagnation internal energy, entropy, and the Mach number.

The printout is stored on TAPE6 and printed automatically at the end of the run.

2. Conservative Variables. For further evaluation such as plotting, the iteration constants, ITR1, ITRN, the local time increment, DELTAU(I,J) (for Δt_{ilj}), and the conservative variables, RHO(I,J), RHOU(I,J), RHOV(I,J) and EO(I,J) (for ρ_{iij}, $(\rho u)_{iij}$, $(\rho v)_{iij}$, and $(e_o)_{iij}$, respectively), are written on TAPE8 by subroutine SUPPLY (ENTRY OUTPUT), as follows,

```
      WRITE (8,504) ITR1, ITRN
  504 FORMAT (IG, IG)
      WRITE (8,503) ((DELTAV(I,J), I=1,IL), J=1,JL)
  503 FORMAT (E17.9, 4E16.9)
      WRITE (8,503) ((RHO(I,J), RHOU(I,J), RHOV(I,J),
        E(I,J), I=1,IL), J=1,JL)
```

The Real Gas Euler Code, AXI2DSG

This code solves the Euler equations for a real gas flow. The molecular weight, M_w, and the internal energy, e, are functions of two variables, ρ and T. For air, these relations are implemented by the code INPLA and then are read as an input by the mail code AXI2DSG.

In the present version of AXI2DSG, the number of mesh points in the ξ and η directions are set at 70 and 44, respectively.

Input Data Description

1. Namelist INPUT. This namelist is identical to that for the perfect gas code described in the previous section except that the specific heat inputs CP and CV along with the specific heat ratios GAMMA and GMI are omitted. In their place, the following gas constant inputs are added:

RG - Universal Gas Constant (8314.3).
AMWO - Gas molecular weight at inlet stagnation conditions (uniform across stream).
GAMMAO-Ratio of specific heats at inlet stagnation conditions (uniform across stream).

2. Namelist DINPL. This namelist reads in the real gas properties for the gas of interest. The data are read from TAPE9 by subroutine INITIA.

X1(I) - Array of derivatives for molecular weight.
Y1(J) - Array of temperatures for molecular weight.
F1(I,J)- Array of molecular weights at (X1(I), Y1(J)).

X2(I) - Density array for molecular weight derivative.
Y2(J) - Temperature array for molecular weight derivative.
F2(I,J)- Array of molecular weight derivatives, $(\partial M_w/\partial \rho)_T$ at (X2(I), Y2(J)).

X3(I) - Density array for $\partial M_w/\partial T$.
Y3(J) - Temperature array for $(\partial M_w/\partial T)$.
F3(I,J)- Array of the derivatives $(\partial M_w/\partial T)_\rho$ at (X3(I), Y3(J)).

XY(F) - Density array for internal energy.
Y4(J) - Temperature array for internal energy.
F4(I,J)- Array of internal energy values, e, at (XY(I), Y4(J)).

X5(I) - Density array for temperature.
Y5(J) - Internal energy array for temperature.

F5(I,J)- Array of temperatures at (X5(I), Y5(J).

X6(I) - Density array for $\partial T/\partial \rho$
Y6(J) - Internal energy array for $\partial T/\partial \rho$
F6(I,J)- Array of derivatives, $(\partial T/\partial \rho)_e$ at (X6(I), Y6(J)).

X7(I) - Density array for $\partial T/\partial e$.
Y7(J) - Internal energy array for $\partial T/\partial e$.
F7(I,J)- Array of derivatives, $(\partial T/\partial e)_\rho$ at (X7(I), Y7(I)).

3. Mesh Points Coordinates. The transformed coordinates are read in the same manner as for the perfect gas code. See the section on The Perfect Gas Euler Code, AXI2DSA.

Output Data Description

1. Printed Results. For the shortform printout, IWRT=0, the namelists INPUT and DINPL, the iterated values of the molecular weight, M_w, and the effective γ at the throat are printed from subroutine INITIA. In addition, the convergence rate for each equation is printed from subroutine SUPPLY (ENTRY CHECK), and the downstream rate of mass flow is printed from subroutine SUPPLY (ENTRY MASS).

The print for IWRT=1 gives all flowfield quantities listed in Output Data Description for the perfect gas Euler code. In addition, it prints the molecular weight and the entropy. The printout is again stored on TAPE6 and printed automatically at the end of the run.

2. Conservative Variables. Plotting information is also given on TAPE8 in the same format and for the same variables as listed in Output Data Description for the perfect gas Euler code.

The Perfect Gas TLNS Code, DDADIPBC

This code solves the TLNS equations for a perfect gas flow. The laminar viscosity, μ, is determined by the relations in Eqns. 54a-c and the turbulent viscosity, μ_f, is determined by the Baldwin-Lomax model of turbulence[9]. The molecular thermal conductivity, k, and the turbulent thermal conductivity, k_T, are expressed in terms of the laminar turbulent viscosities by means of the Prandtl numbers, Pr, and Pr_T, respectively, as defined in Eqn. 13.

In the perfect gas viscous code, the maximum number of mesh points as limited by the NOS-BE system are (70 x 44), the same as for the perfect gas Euler code. Because of the additional run time required (see Table 2), the viscous code is designed to be stopped and restarted.

Input Data Description

1. Namelist INPUT. As for the inviscid codes, this namelist specifies the flow constants. The namelist is read from TAPE1 by subroutine INITIA. Most of the variables are analogous to

those described in Input Data Description for the perfect gas Euler code. These have the same definitions as given there. Additional quantities for specifying the viscous flow are:

NBEG - The initial iteration number. Generally set to 1.

NEDN - Total number of iterations to be performed. If NEND=1, the PNS solution is obtained.

NITER - Number of iterations in the run (Integer).

CFL1 - CFL number for the PNS solution.

IVISC - An integer specifying the nature of the flow:
IVISC=0 inviscid flow (with IVISC=0, this code should duplicate the results of AXI2DSA);
IVISC=1 viscous flow.

IWALL - An integer specifying the wall boundary condition;
IWALL=0 an adiabatic wall,
IWALL=1 specified wall temperature.

PRN - Laminar Prandtl number (constant).

PRNT - Turbulent Prandtl number (constant).

TREF - The reference temperature for the viscosity in Eqn. 54, K.

ZMUO - Reference viscosity at T=TREF (SI units).

OMEGA - Exponent for the power law form of the viscosity, Eqn. 54c.

TWALL - Value of the wall temperature when IWALL=1 (TWALL=0.0 when IWALL=0), K.

PB - The back pressure at the nozzle exit. When PB=0, the subsonic pressure at the exit is extrapolated from interior.

IRVN - Control variable for restart (Integer):
IRVN=0, for the first run only,
IRVN=A, for restart from previous runs.

2. Mesh Points Coordinates. These are read in the same format as for the Euler code, Input Data Description.

3. Convergence Rate, Previous Run. (Ignored if not a restart case.) For the case of restart files, additional (formatted) input is placed on TAPE1 to specify the iteration number (NDUM+1) at which the restart run begins. This input is also used to create a continuous file of the convergence of the TLNS equations. This input is placed on TAPE1 behind the namelist INPUT data. It is also read by subroutine INITIA. The reading format is,

```
 70 READ (9, 502, END=65) NDUM, (SS(K), K=1,Y)
502 FORMAT (I5, 3X, 4(1X, E14.7)
    .
    .
    .
    GO TO 70
 65 CONTINUE
```

4. Restart Field For Continuation Runs. (Ignored if not a restart case.) Flowfield information for restarting a computation is obtained by reading the conservative variables data on TAPE7. This tape contains the local time step, DELTAU(I,J) and the

conservative variables, RHO(I,J), RHOU(I,J), RHOV(I,J), and E(I,J) for each grid point. This file is written upon completion of any run and is then read from TAPE7 by subroutine INITIA for use in restarting. The format used is:

```
               READ(7) ((DELTAU(I,J), I=1,IL), J=1,JL)
               READ(7) ((RHO(I,J), RHOU(I,J), RHOV(I,J),
                         E(I,J), I=1,IL), J=1,JL)
```

Output Data Description

1. Printed Results. The printed output is the same as for the perfect gas Euler code described in Output Data Description.

2. Namelist INPUT. (If not a restart case, namelist INPUT is written out to TAPE2.) In this output, the value of IRVN is changed by the code from 0 to 1 to prepare for the restart. This change enables the next run to read the convergence rate from TAPE9 and the conservative variables from TAPE7 as determined by the previous run. The conservative variables are then used as the initial condition for the succeeding run.

3. Convergence Rate. This output presents the convergence rate of the TLNS equations, SS(K), K=1,4 for each iteration. The convergence rates from the previous run are stored on TAPE10 by subroutine INITIA, so that a continuous concatenation of the convergence will be available. The format is:

```
               WRITE (10, 502) NDUM, (SS(K), K=1,4)
           502 FORMAT (I5, 3X, 4(1X, E14.7))
```

The convergence levels from the current run are added to this same file by subroutine SUPPLY (ENTRY CHECK).

4. Conservative Variables. The conservative variables used for restart are written on TAPE8 in the same manner described in Output Data Description for the perfect gas Euler code.

Choosing the Laminar Viscosity

The laminar viscosity is chosen from among the three choices in Eqn. 54 by subroutine MU (ENTRY MULAN (I1)). One of the following three lines of FORTRAN must be included by recompilation. For the constant viscosity case, set,

```
             ZMU(J) = ZMU0,
```

For the Sutherland law case, set,

```
             ZMU(J) = ZMUO*TOS/TTS*(TT/TREF)**1.5,
```

and for the power law case, set,

```
             ZMU(J) = ZMUO*(TT/TREF)**OMEGA
```

Viscous Flow of Real Gases, Code DDADIPBG

The code DDADIPBG solves the TLNS equations for a real gas flow. The molecular weight, M_W, and the internal energy, e, are functions of the density, and the temperature as noted before, and representative properties for air have been used for verification purposes as stated earlier. The laminar viscosity, μ, and the laminar thermal conductivity, k, are given as simplified curve fit functions of the density, ρ, and the internal energy, e. The turbulent viscosity, μ_+, is determined by the Baldwin-Lomax model of turbulence[9]. The turbulent thermal conductivity, k_T, is again determined from the turbulent viscosity using the turbulent Prandtl number.

In this version of the code, the number of mesh points allowed under the NOS-BE system is limited to 60 points in the ξ direction by 40 points in the η direction. The code is designed with restart capability so that converged solutions can be obtained in sequential runs.

Input Data Description

1. Namelist INPUT. The variables in this namelist are identical to those for the perfect gas TLNS code (version DDADIPBC, Input Data Description for the perfect gas Euler code) except that the real gas variables, RG, AMWO, and GAMMAO which were used in the real gas Euler code (version AXI2DSG, Input Data Description for the real gas Euler code) are also included here.

2. Namelist DINPL. This namelist is used to read in the real gas properties. It is identical to the namelist used in the real gas Euler code, version AXI2DSG, in Input Data Description for the real gas Euler code.

3. Mesh Point Coordinates. The transformed coordinates are again read in in the same fashion as before. The input is described in Input Data Description for the perfect gas version of the Euler code.

4. Convergence Rate Summary for Restart Cases. The table of convergence levels from previous runs is again used as input for restart cases so a single concaterated file of the error at each iteration is available. Input details are given in Input Data Description for the perfect gas version of the TLNS code.

5. Conservative Variables Input for Restart. Restarting the code is accomplished by reading in the conservative variables from the last time step of the previous run. Details of this output file are given in Output Data Description of the real gas version of the Euler code. Comments as to re-entering this data for restart purposes are given in Input Data Description of the perfect gas version of the TLNS code.

Output Data Description

Output from the real gas version of the TLNS code is composed of four parts. Flowfield results are available in either long or short form and are written to TAPE6 as described in Input Data Description (real gas version of the perfect gas code). A printout of Namelist INPUT is written to TAPE9 as described in Output Data Description of the perfect gas TLNS code, with the value of IRVN reset as necessary. The convergence rate table including a summary of all runs to date is written to TAPE7 for plotting or restarting, and the conservative variables for the last time step computed are written to TAPE8 for similar purposes. Format details and contents of these tapes are described in Output Data Description of the perfect gas Euler code.

EXAMPLE CALCULATIONS

A broad spectrum of perfect gas calculations are given in Appendix D along with some careful validations of the code predictions against independent standards. The perfect gas examples include inviscid calculations for the transonic flow in the converging and throat sections of various nozzle geometries, as well as inviscid solutions for the divergence, supersonic portion of various nozzles. In addition, similar perfect gas results are presented to illustrate TLNS solutions in both the subsonic-transonic and the supersonic sections of the nozzle. These results include both laminar and turbulent conditions. In addition, Appenuix D gives both inviscid and viscous solutions for swirling flow in the divergent section of various nozzles. Results concerning the effect of back pressure on the flowfield structure near the nozzle lip are presented for moderate variations in back pressure showing, in particular, the manner in which recirculating regions are set up near the nozzle wall. Finally, examples showing the coupled effect of nozzle wall cooling on wall temperature and heat flux are presented.

In the present section, we summarize some real gas calculations for one particular nozzle geometry and we present some complete subsonic-transonic-supersonic solutions for an entire converging-diverging nozzle.

Real Gas Calculations For The R-S Nozzle

The coordinates of the supersonic section of a specific contoured nozzle are given in Table 5. This nozzle geometry was used for the real gas calculations given in this section. The Namelist INPUT data needed to run the cases described herein is given in Figs. 1 through 4 for the perfect gas Euler version of the code (Fig. 1), the real gas Euler version (Fig. 2), the perfect gas TLNS version (Fig. 3), and the real gas TLNS version (Fig. 4). These calculations were run on the AFAL CYBER 180/840 machine under the NOS-BE operating system. This available core size with this system is very limited so that maximum allowable grid sizes are quite sparse. Nevertheless, the data give an indication of the code's validity. The real gas calculations were all done for equilibrium air as noted earlier.

Figure 5 shows the nozzle geometry and the 70 x 44 grid that was used for the Euler equation solutions. Figure 6 shows Mach number and u-velocity component contour plots for the perfect gas constant specific heat calculation. Comparisons of calculations made with the perfect gas code of Appendix D and the real gas Euler code with property tables that correspond to constant specific heats were identical indicating the real gas routines were working properly. These latter perfect gas calculations are shown in Fig. 7. The effects of real gas properties (for equilibrium air) are shown on Fig. 8. As would be expected, real gas properties drastically diminish the Mach numbers, but the exit velocity profiles are slightly higher for the real gas case partially because of the low value of γ used for the perfect gas calculations.

Thin-layer Navier-Stokes calculations are shown on Figs. 9 through 11. Figure 9 shows the 60 x 40 grid used for these calculations (refer to Table 2), while Fig. 10 shows TLNS results based on perfect gas properties and Fig. 11 shows results for the real gas calculation. Again, the effect of real gas properties is to reduce the expansion Mach number.

Additional Perfect Gas Calculations For The Converging-Diverging Nozzle

As an additional calculation demonstrating the coupled transonic-supersonic codes, we have computed inviscid and viscous solutions for a complete converging-diverging nozzle. Figure 12 shows the converging portion of the nozzle and the 120 x 80 grid used for inviscid calculations. Figure 13 shows the Mach number contours in the subsonic portion of the nozzle. This transonic flow solution was used to obtain starting information for the supersonic portion of the grid which is shown in Fig. 14. In the diverging section, we used a 145 x 80 grid. Figure 15 shows the Mach number contours in the supersonic portion of the nozzle, while Fig. 16 shows the Mach number contours in the entire nozzle. The presence of a fairly strong shock in this nozzle gives rise to wiggles in the Mach number contours. These could be minimized by the addition of a second order viscosity in the vicinity of the shock or a TVD limiter. These have not been added because nozzle flows typically contain only weak shocks. Corresponding pressure contours for the inviscid case are given on Fig. 17. These calculations are performed for a specific heat ratio $\gamma = 1.17$.

The subsonic portion of the grid used for the viscous calculations was increased to 120 x 100 to improve resolution in the boundary layer. This viscous grid is shown on Fig. 18, and the corresponding flow Mach number contours are given on Fig. 19. (Note, these viscous calculations are all for the Baldwin-Lomax turbulence model.)

The viscous grid used in the supersonic portion of the nozzle included a 145 x 100 grid as shown in Fig. 20. The corresponding Mach number contours in the supersonic section are given in Fig. 21, while Mach number contours for the complete nozzle are given in Fig. 22. Note, that comparison of Figs. 16 and 22 shows that the inviscid flow expands slightly more than the viscous case as would be expected. These calculations are for a throat Reynolds number of 5×10^5. Corresponding static pressure contours for the entire nozzle are given in Fig. 23.

Finally, the convergence of the viscous TLNS calculation is presented in Fig. 24. This figure shows that the RMS errors in the solution were reduced by more than five orders of magnitude in 40 iterations.

REFERENCES

1. Chang, C.-L., "Time-Iterative Solutions of Viscous Supersonic Flows", Ph.D. Dissertation, Department of Mechanical Engineering, The Pennsylvania State University, December 1988.

2. Chang, C.-L., Kronzon, Y. and Merkle, C. L., "Time-Iterative Solutions of Viscous Supersonic Nozzle Flows", AIAA Journal, Vol. 26, No. 10, October 1988, pp. 1208-1215.

3. Chang, C.-L. and Merkle, C. L., "The Relation Between Flux Vector Splitting and Parabolized Schemes", Journal of Computational Physics, Vol. 80, No. 2, February 1989, pp. 344-361.

4. Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes", Journal of Computational Physics, Vol. 43, 1983, pp. 357-372.

5. Vigneron, Y. C., Rakich, J. V., and Tannehill, J. L., "Calculation of Supersonic Viscous Flow Over Delta Wings with Subsonic Leading Edges", AIAA Paper 78-1137, 1978, Seattle, WA.

6. Zucrow, M. J., Hoffman, J. D., Gas Dynamics, Vol. 1, John Wiley & Sons, Inc., 1976, pp. 53-63.

7. Brahinsky, H. S. and Neal, C. A., "Tables of Equilibrium Thermodynamic Propertier of Air", Vol. 1, Constant Temperature, ARO, Inc., AEDS-TR-69-89, Vol. 1, April 1987.

8. Srinivasan, S., Tannehill, J. L., and Weilmuenster, K. J., "Simplified Curve Fits for the Transport Properties of Equilibrium Air", ERI Report 88405, Iowa State University, Ames, IA, September 1987.

9. Baldwin, B. S. and Lomax, H., "Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows", AIAA Paper 78-0257, Huntsville, AL, January 1978.

Table 1. SUMMARY OF THE VARIOUS VERSIONS OF THE CODE

Code Name	Governing Equations	Gas	Molecular Weight M_W	Internal Energy e	Viscosity	Thermal Conductivity k^*
AXI2DSA	Euler	Perfect	Const.	C_VT	-	-
AXI2DSG	Euler	Real	$M_W(\rho,T)$	$e(\rho,T)$	-	-
DDADIPBC	TLNS	Perfect	Const.	C_VT	Power Law or Sutherland's law	-
DDADIPBG	TLNS	Real	$M_W(\rho,T)$	$e(\rho,T)$	$\mu(\rho,e)$	$k(\rho,e)$

*Related to viscosity by Prandtl number.

Table 2. CHARACTERISTICS OF CODE EXECUTION

Code Name	Number of Mesh Points*	Number of Iterations	Executing Time on CYBER 180/840 Per Iteration Per Mesh Point (Milliseconds)
AXI2DSA	70 x 44	30	3.851
AXI2DSG	70 x 44	30	7.412
DDADIPBC	70 x 44	400	27.211
DDADIPBG	60 x 44	2000	121.912

*Maximum storage allowed by NOS-BE system. NOS-VE allows much denser grids.

Table 3. UNITS FOR INPUTS

Physical Quantity	Unit	Symbol
Length	meter	m
Mass	kilogram	kg
Density	kilogram per cubic meter	kg/m^3
Time	second	sec
Temperature	kelvin	K
Force	newton	N
Pressure	newton per square meter	n/m^3
Energy	joule	J
Molecular weight	kg per kilogram mole	kg/kg mole
Viscosity	newton-second per square meter	$(n \cdot sec)/m^2$
Thermal conductivity	joule kelvin-meter-second	$J/(k \cdot m \cdot sec)$

Table 4. FUNCTION SUBPROGRAMS FOR REAL GAS PROPERTY EVALUATION

Function Subprogram Name and Arguments	Property Computed	Comments
FAMW (RHO,T)	$M_W(\rho,T)$	Molecular weight
FE (RHO, T)	$E(\rho,T)$	Internal energy
FT (RHO, E)	$T(\rho,e)$	Temperature
FDMDRT (RHO, T)	$(\partial M_W/\partial \rho)_T$	Molecular weight derivative as function of ρ and T
FDMDTR (RHO, T)	$(\partial M_W/\partial T)_\rho$	Molecular weight derivative as function of ρ and T
FDTDRE (RHO, E)	$(\partial T/\partial \rho)_e$	Temperature derivative; function of ρ and e
FDTDER (RHO, E)	$(\partial T/\partial e)_\rho$	Temperature derivative; function of ρ and e
FZMU (RHO, E)	$\mu(\rho,e)$	Viscosity
FZK (RHO, E)	$k(\rho,e)$	Thermal conductivity; computed from Prandtl number
FDMUDRE (RHO, E)	$(\partial \mu/\partial \rho)_e$	Viscosity derivative; function of ρ and e
FDMUDER (RHO, E)	$(\partial \mu/\partial e)_\rho$	Viscosity derivative; function of ρ and e
FDKDRE (RHO, E)	$(\partial k/\partial \rho)_e$	Thermal conductivity derivative
FDKDER (RHO, E)	$(\partial k/\partial e)_\rho$	Thermal conductivity derivative

Table 5a. CONTOUR POINTS OF THE RS NOZZLE (70, 44)

I	YC	XC	ALPHA	I	YC	XC	ALPHA
1	.01006	0.00000	0.00000	41	.07105	.11669	21.64860
2	.01009	.00093	3.36523	42	.07377	.12362	21.08610
3	.01018	.00191	7.10866	43	.07654	.13090	20.52531
4	.01035	.00295	11.07978	44	.07936	.13857	19.96692
5	.01061	.00403	15.31548	45	.08224	.14662	19.41090
6	.01097	.00517	19.86431	46	.08518	.15509	18.85670
7	.01147	.00637	24.79635	47	.08818	.16400	18.30537
8	.01213	.00763	30.20430	48	.09123	.17336	17.75549
9	.01301	.00896	36.34891	49	.09433	.18321	17.20873
10	.01403	.01035	36.33356	50	.09748	.19356	16.66520
11	.01511	.01182	36.27180	51	.10068	.20444	16.12458
12	.01623	.01336	36.11491	52	.10393	.21588	15.58597
13	.01741	.01498	35.89578	53	.10723	.22792	15.05187
14	.01864	.01668	35.62613	54	.11056	.24056	14.52032
15	.01992	.01847	35.30239	55	.11395	.25386	13.99123
16	.02124	.02035	34.93930	56	.11736	.26785	13.46773
17	.02261	.02233	34.54235	57	.12081	.28255	12.94699
18	.02404	.02442	34.11646	58	.12429	.29801	12.42823
19	.02551	.02660	33.66482	59	.12780	.31426	11.91538
20	.02703	.02890	33.19136	60	.13133	.33135	11.40440
21	.02859	.03132	32.70377	61	.13487	.34931	10.89921
22	.03021	.03387	32.20513	62	.13841	.36820	10.39807
23	.03188	.03654	31.69175	63	.14197	.38806	9.89894
24	.03360	.03935	31.16775	64	.14552	.40894	9.40745
25	.03537	.04231	30.63482	65	.14907	.43090	8.91692
26	.03719	.04542	30.09359	66	.15258	.45398	8.43359
27	.03906	.04869	29.54643	67	.15609	.47825	7.95193
28	.04099	.05212	28.99329	68	.15957	.50377	7.48025
29	.04297	.05573	28.43651	69	.16286	.53060	7.00226
30	.04500	.05953	27.87618	70	.16596	.55880	6.52500
31	.04709	.06353	27.31309				
32	.04923	.06772	26.74888				
33	.05143	.07214	26.18210				
34	.05368	.07678	25.61507				
35	.05599	.08166	25.04716				
36	.05836	.08679	24.47881				
37	.06078	.09219	23.91089				
38	.06326	.09786	23.34445				
39	.06580	.10382	22.77762				
40	.06840	.11009	22.21202				

Note: XC and YC are axial and radial coordinates in meters;
ALPHA is wall angle in degrees.

Table 5b. CONTOUR POINTS OF THE RS NOZZLE (60, 40)

T	YC	XC	ALPHA
1	.01006	0.00000	0.00000
2	.01009	.00093	3.36523
3	.01018	.00193	7.15475
4	.01036	.00298	11.22513
5	.01063	.00411	15.62255
6	.01102	.00531	20.41094
7	.01157	.00658	25.67889
8	.01232	.00794	31.56032
9	.01332	.00938	36.34023
10	.01445	.01092	36.32034
11	.01565	.01255	36.20025
12	.01692	.01429	35.99863
13	.01826	.01615	35.71517
14	.01966	.01812	35.36682
15	.02114	.02021	34.96643
16	.02269	.02245	34.51934
17	.02431	.02482	34.03213
18	.02600	.02735	33.51066
19	.02777	.03004	32.95256
20	.02961	.03291	32.39225
21	.03152	.03596	31.80246
22	.03351	.03920	31.19516
23	.03557	.04266	30.57319
24	.03772	.04633	29.93819
25	.03994	.05025	29.29224
26	.04225	.05441	28.63803
27	.04463	.05884	27.97653
28	.04710	.06356	27.30897
29	.04966	.06858	26.63766
30	.05230	.07392	25.96185
31	.05502	.07960	25.28350
32	.05784	.08565	24.60259
33	.06074	.09209	23.92056
34	.06373	.09895	23.23896
35	.06681	.10624	22.55594
36	.06998	.11400	21.87451
37	.07324	.12227	21.19345
38	.07659	.13106	20.51365
39	.08003	.14042	19.83620
40	.06356	.15038	19.16112
41	.08717	.16098	18.48817
42	.09087	.17226	17.81855
43	.09466	.18427	17.15137
44	.09852	.19705	16.48805
45	.10246	.21065	15.82853
46	.10647	.22513	15.17335
47	.11056	.24053	14.52154
48	.11470	.25693	13.87519
49	.11891	.27438	13.23191
50	.12317	.29295	12.59448
51	.12747	.31272	11.96280
52	.13181	.33376	11.33653
53	.13617	.35615	10.71358
54	.14055	.37998	10.09854
55	.14492	.40534	9.49067
56	.14929	.43233	8.88584
57	.15363	.46106	8.29007
58	.15796	.49163	7.70414
59	.16210	.52417	7.11456
60	.16596	.55880	6.52500

Note: XC and YC are axial and radial coordinates in meters;
ALPHA is wall angle in degrees.

```
-$INPUT
 IL        = 70,
 JL        = 44,
 ITR1      = 1,
 ITRN      = 1,
 ITRS      = 1,
 ISUP      = 0,
 CFL       = 5.,
 ITIME     = 1,
 THETA     = 1.,
 RM1       = 1.02,
 RM2       = 8.,
 AIN       = 0.05,
 AEX       = 0.275,
 RL        = 0.8485,
 PO        = 3509431.4,
 TO        = 3497.8,
 CP        = 1308.79,
 CV        = 1021.77,
 GAMMA     = 1.2809,
 GM1       = .2809,
 INER      = 30,
 IB1       = 2,
 IB2       = 70,
 NORD      = 1,
 OMEGAX    = 0.0,
 OMEGAY    = .5,
 IREAD     = 1,
 IWRT      = 1,
 IRUN      = 1,
 $END
```

Fig. 1 Sample Input For AXI2DSA; Namelist INPUT.

```
-$INPUT
 IL          = 70,
 JL          = 44,
 ITR1        = 1,
 ITRN        = 1,
 ITRS        = 1,
 ISUP        = 0,
 CFL         = 5.,
 ITIME       = 1,
 THETA       = 1.,
 RM1         = 1.02,
 RM2         = 8.,
 AIN         = 0.01,
 AEX         = 0.11,
 RL          = 0.245,
 PO          = 3509431.4,
 TO          = 3497.8,
 RG          = 8314.,
 AMWO        = 28.9670,
 GAMMAO      = 1.2809,
 INER        = 30,
 IB1         = 2,
 IB2         = 70,
 NORD        = 1,
 OMEGAX      = 0.0,
 OMEGAY      = .5,
 IREAD       = 1,
 IWRT        = 1,
 IRUN        = 1,
 $END
```

Fig. 2 Sample Input For AXI2DSG; Namelist INPUT.

```
-$INPUT
 IL          = 70,
 JL          = 44,
 NBEG        = 1,
 NEND        = 1,
 NITER       = 100,
 THETA       = 1.0,
 NORD        = 1,
 CFL         = 4.0,
 CFL1        = 1.0,
 ITIME       = 1,
 OMEGAX      = 0.0,
 OMEGAY      = 0.5,
 AIN         = 0.05,
 AEX         = 0.275,
 RL          = 0.8485,
 FST         = 0.0,
 FSTY        = 0.0,
 RM1         = 1.02,
 RM2         = 8.0,
 IVISC       = 1,
 IWALL       = 0,
 GAMMA       = 1.2809,
 CP          = 1308.79,
 REN         = 1.0,
 PRN         = 0.7,
 PRNT        = 0.9,
 TREF        = 3109.55,
 ZMUO        = 1.02319E-05,
 OMEGA       = 0.8130,
 PO          = 3509431.4,
 TO          = 3497.8,
 TWALL       = 0.0,
 PB          = 0.0,
 IREAD       = 1,
 IWRT        = 0,
 IRUN        = 0,
 $END
```

Fig. 3 Sample Input For DDADIPBC; Namelist INPUT.

```
-$INPUT
 IL          = 60,
 JL          = 40,
 NBEG        = 1,
 NEND        = 100,
 NITER       = 100,
 THETA       = 1.0,
 NORD        = 0,
 CFL         = 0.5,
 CFL1        = 1.0,
 ITIME       = 0,
 OMEGAX      = 0.0,
 OMEGAY      = 0.08,
 AIN         = .5E-01,
 AEX         = .275E+00,
 RL          = .8485E+00,
 FST         = 0.0,
 FSTY        = 0.0,
 RM1         = 1.02,
 RM2         = 5.0,
 IVISC       = 1,
 IWALL       = 0,
 RG          = 8314.,
 AMWO        = 28.967,
 GAMMAO      = 1.2809,
 CP          = 1308.79,
 REN         = .1E+01,
 PRN         = 0.7,
 PRNT        = 0.9,
 TREF        = 1.0,
 ZMUO        = .102319E-06,
 OMEGA       = .838551,
 PO          = 3509431.4,
 TO          = 3497.8,
 TWALL       = 0.0,
 PB          = 0.0,
 IREAD       = 1,
 IWRT        = 0,
 IRUN        = 0,
 $END
```

Fig. 4 Sample Input For DDADIPBG; Namelist INPUT.

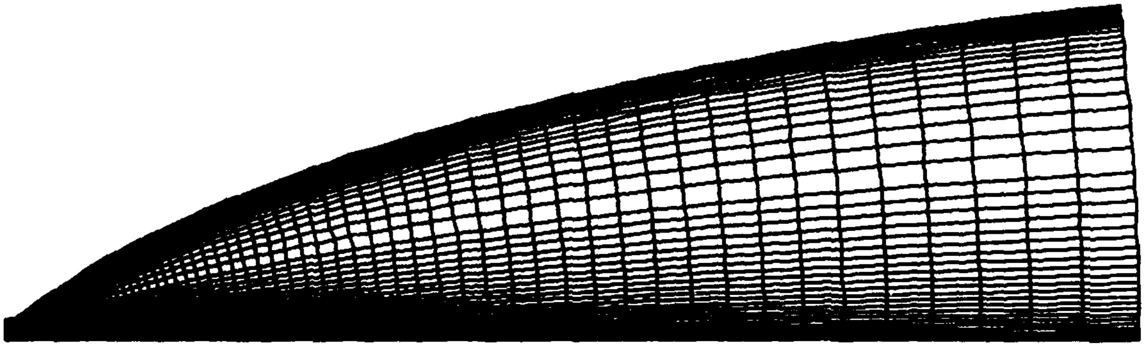
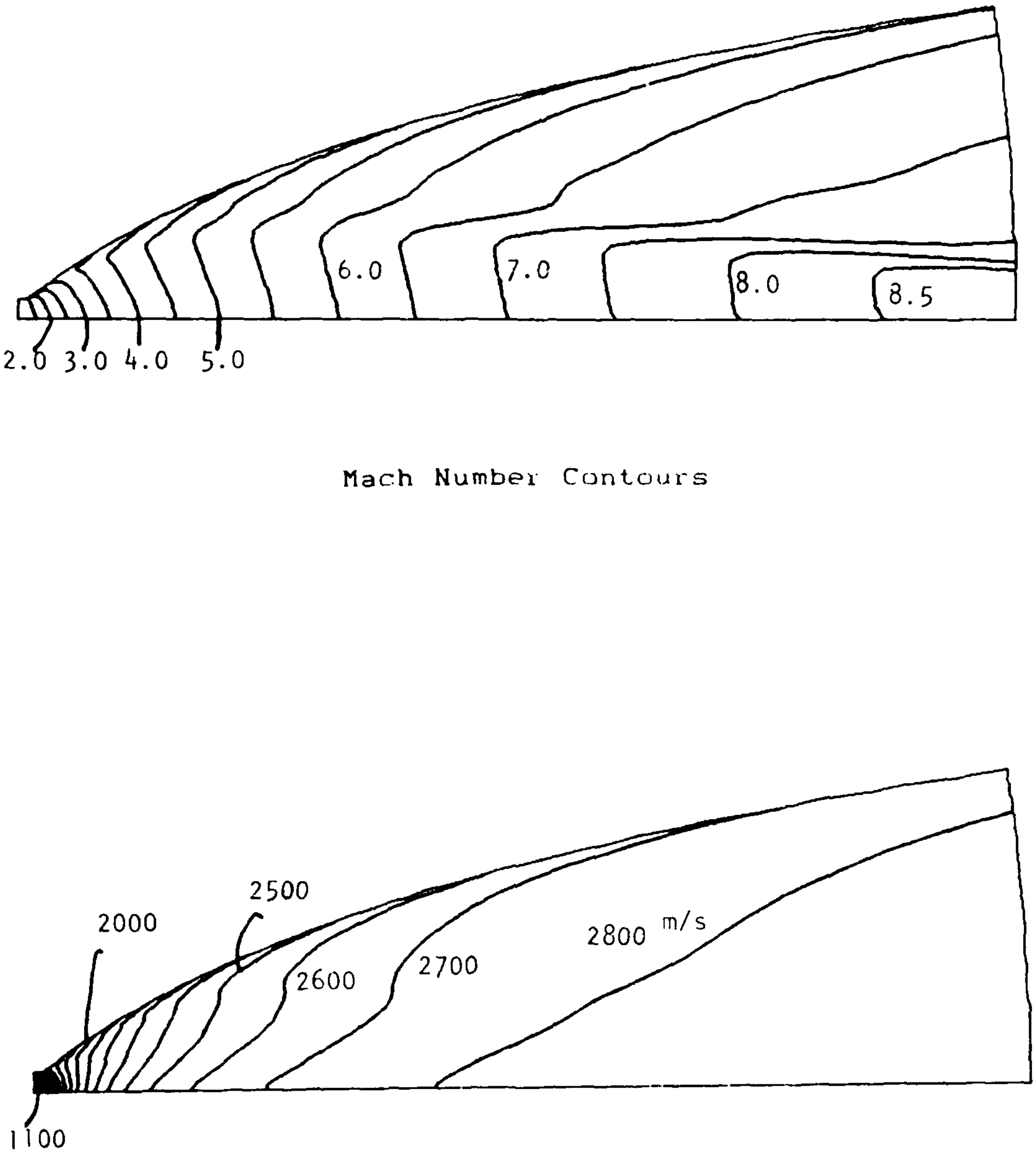


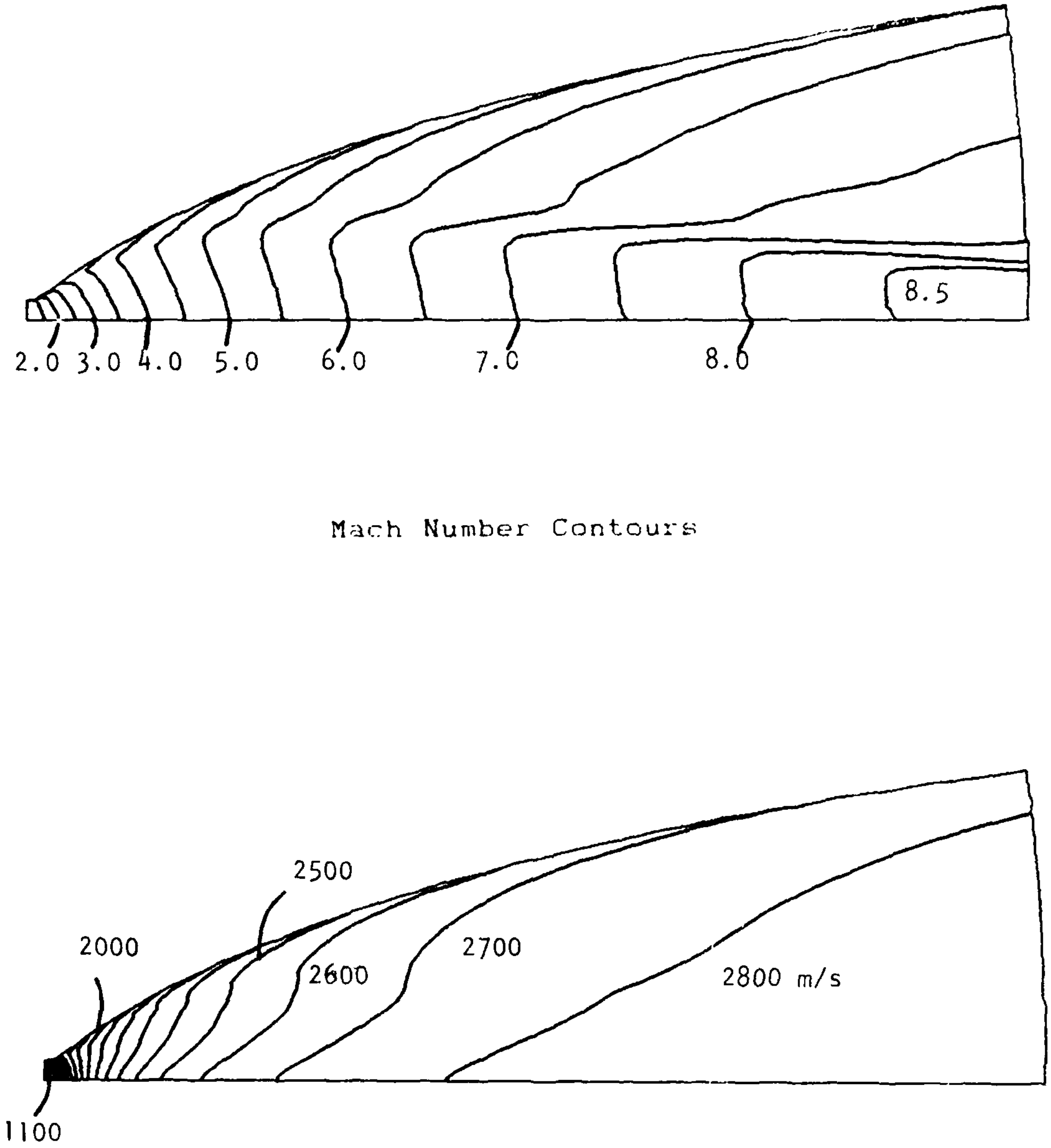
Fig. 5 Nozzle Geometry and 70 x 44 Grid Used For Euler Equation Calculations.



Mach Number Contours

u-Velocity Contours

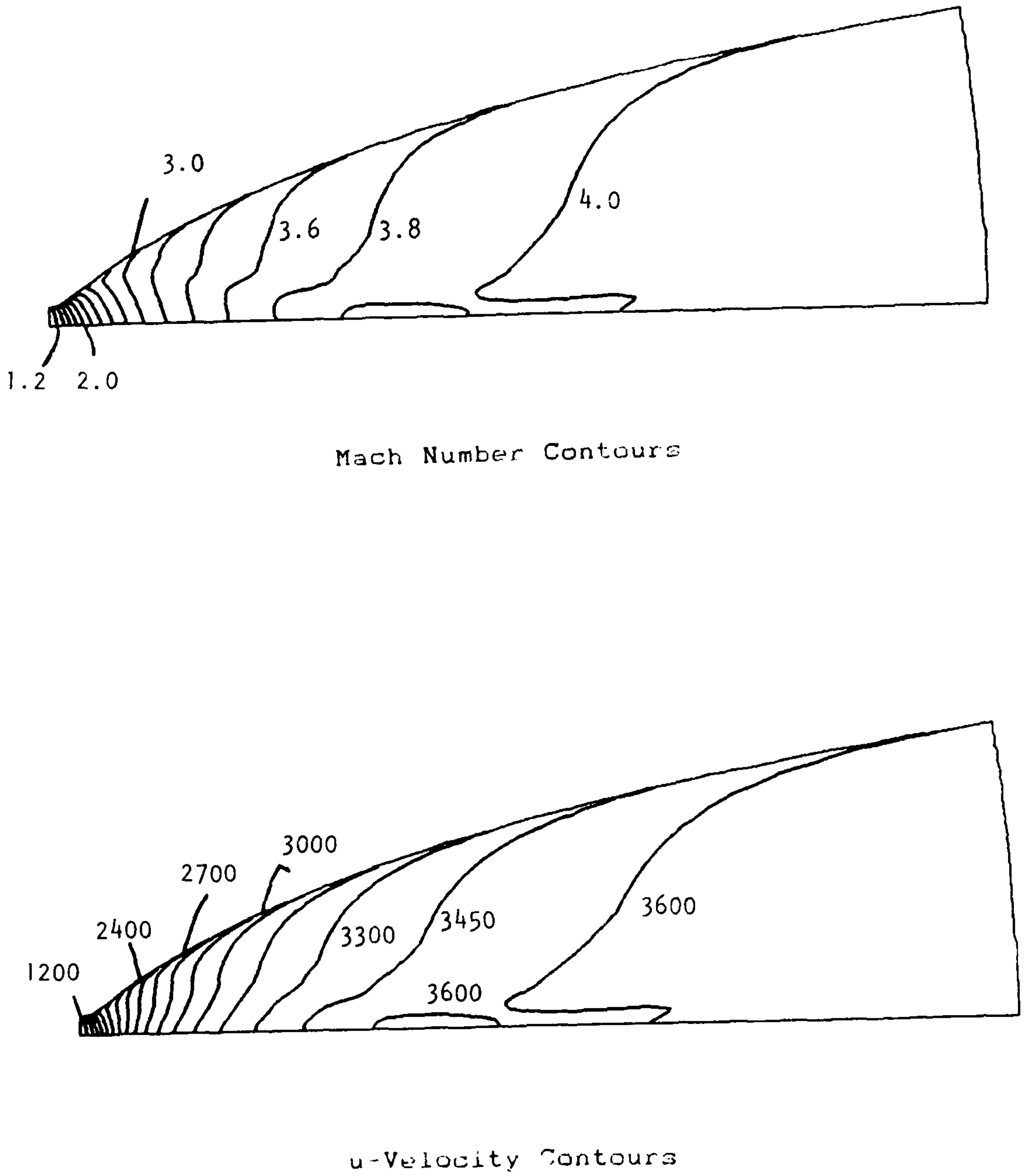
Fig. 6 Contours of Mach Number (Top) and U-Velocity (Bottom) For Perfect Gas Solution of Euler Equations.



Mach Number Contours

u-Velocity Contours

Fig. 7 Contours of Mach Number (Top) and U-Velocity (Bottom) For Simulated Real Gas Properties Based on Tabular Values That Give Constant Specific Heats.



Mach Number Contours

u-Velocity Contours

Fig. 8 Contours of Mach Number (Top) and U-Velocity (Bottom) For Real Gas Calculations of Euler Equations Based on Equilibrium Air.

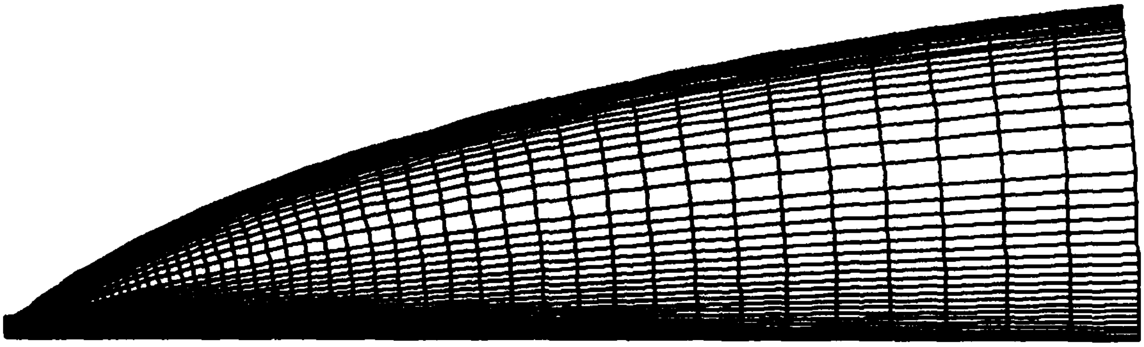
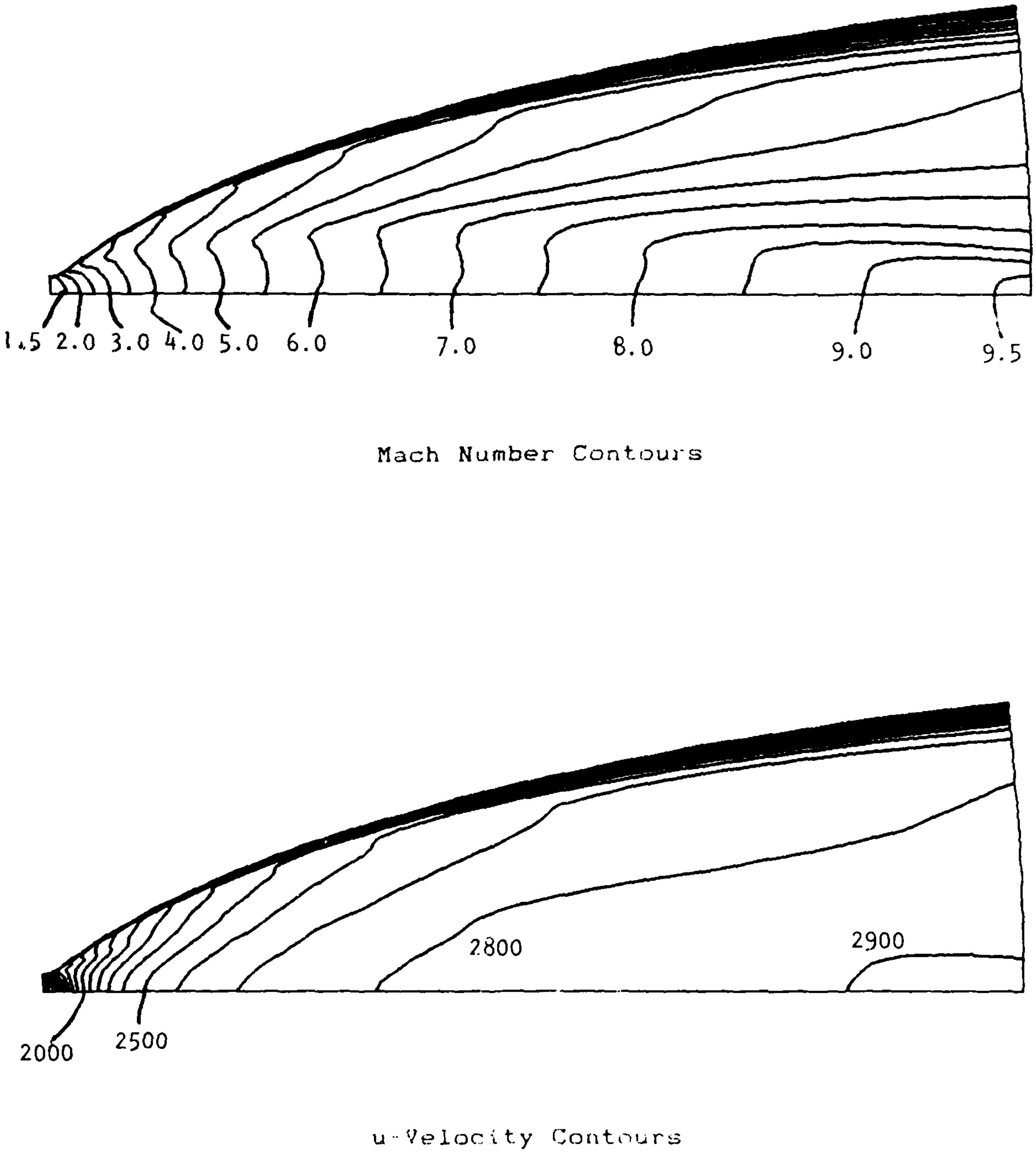


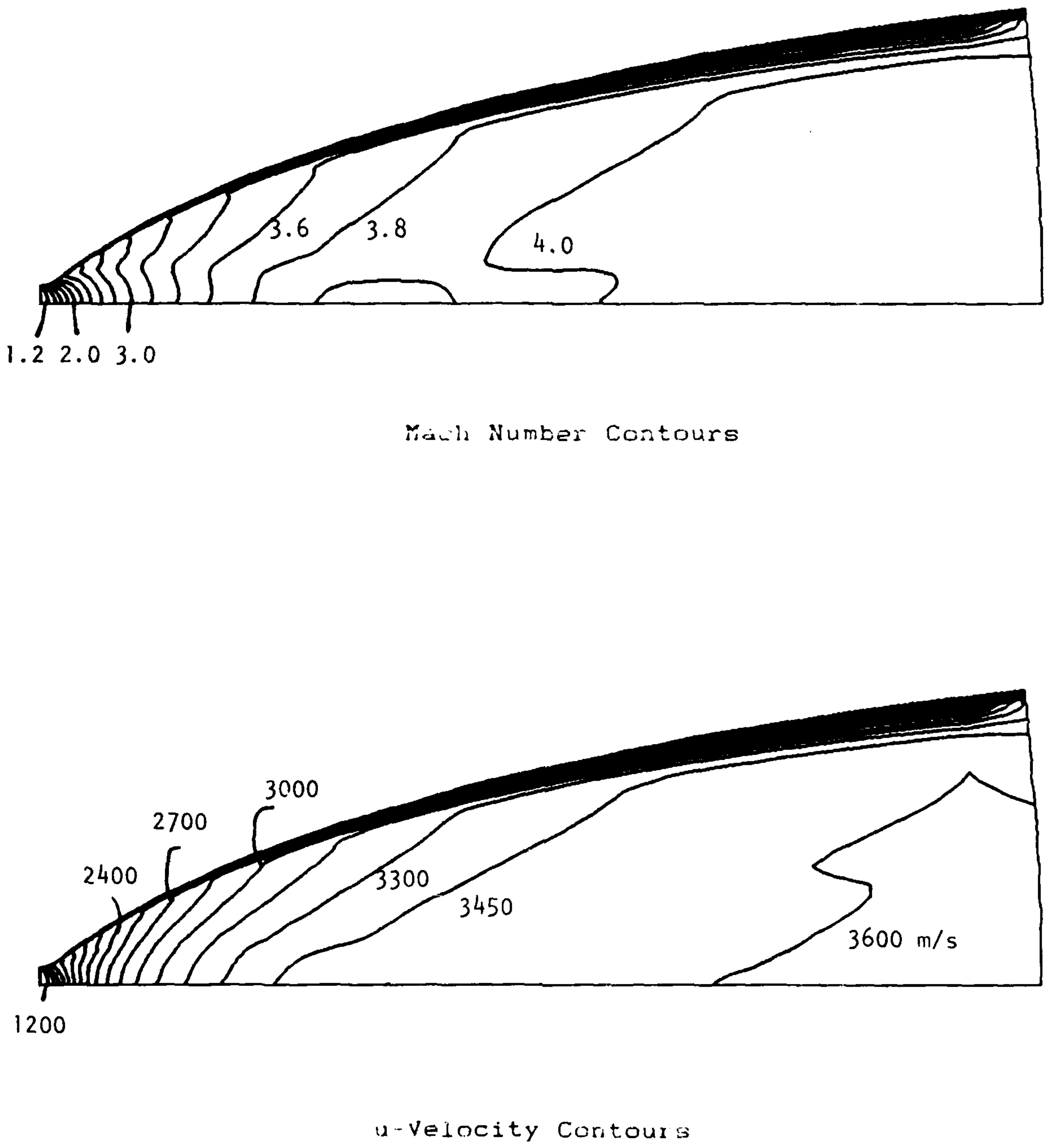
Fig. 9 Nozzle Geometry and 60 x 40 Grid Used for Thin-Layer Navier-Stokes Calculations.



Mach Number Contours

u-Velocity Contours

Fig. 10 Contours of Mach Number (Top) and U-Velocity (Bottom) For Perfect Gas Calculations of Thin-Layer Navier-Stokes Equations.



Mach Number Contours

u-Velocity Contours

Fig. 11 Contours of Mach Number (Top) and U-Velocity (Bottom) For Real Gas Calculations of Thin-Layer Navier-Stokes Equations Using Properties of Equilibrium Air.

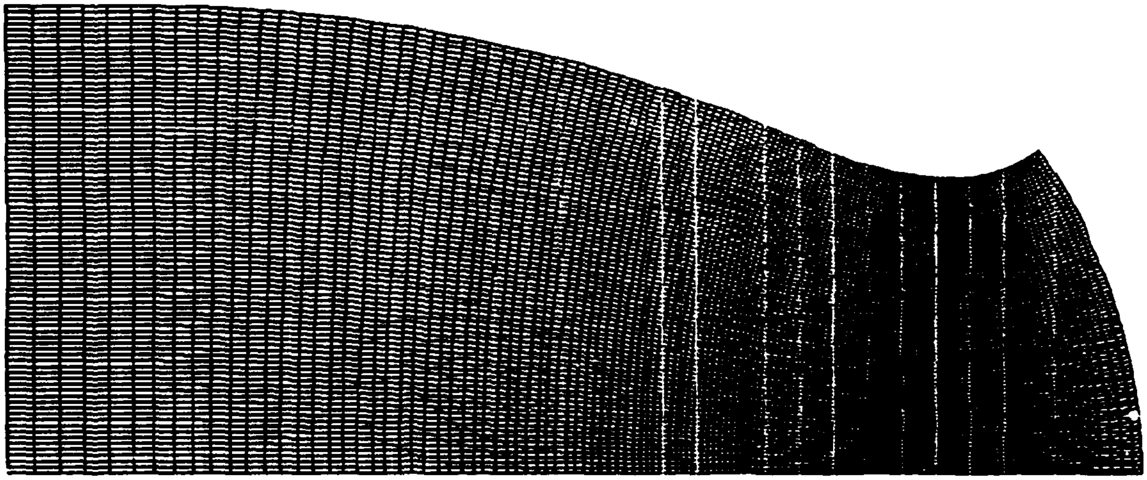


Fig. 12 120 x 80 Grid For Inviscid Transonic Flow Calculation.

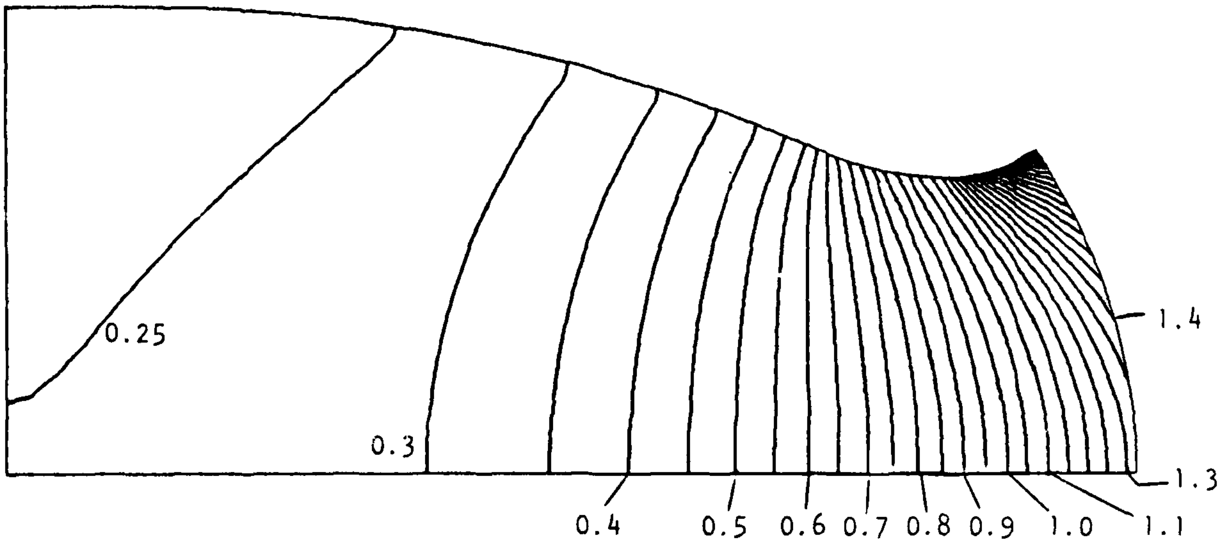


Fig. 13 Mach Number Contours of Inviscid Transonic Flow.

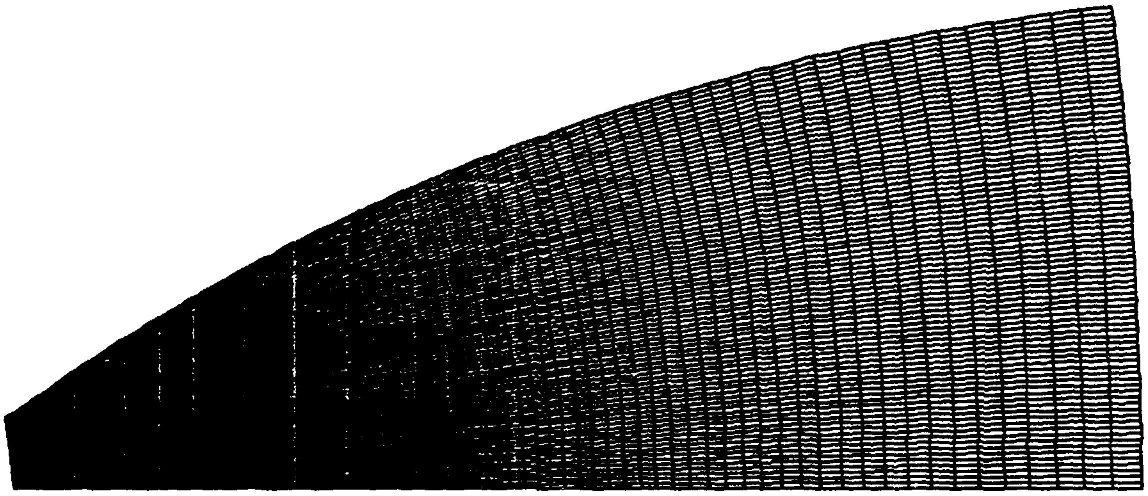


Fig. 14 145 x 80 Grid For Inviscid Supersonic Flow Calculation.

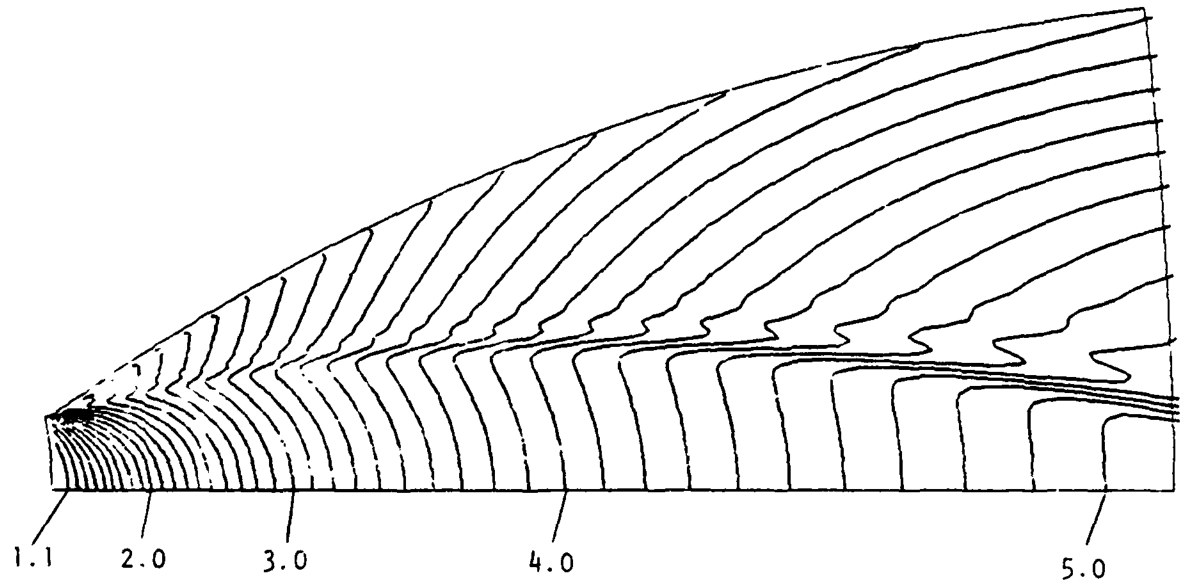


Fig. 15 Mach Number Contours of Inviscid Supersonic Flow.

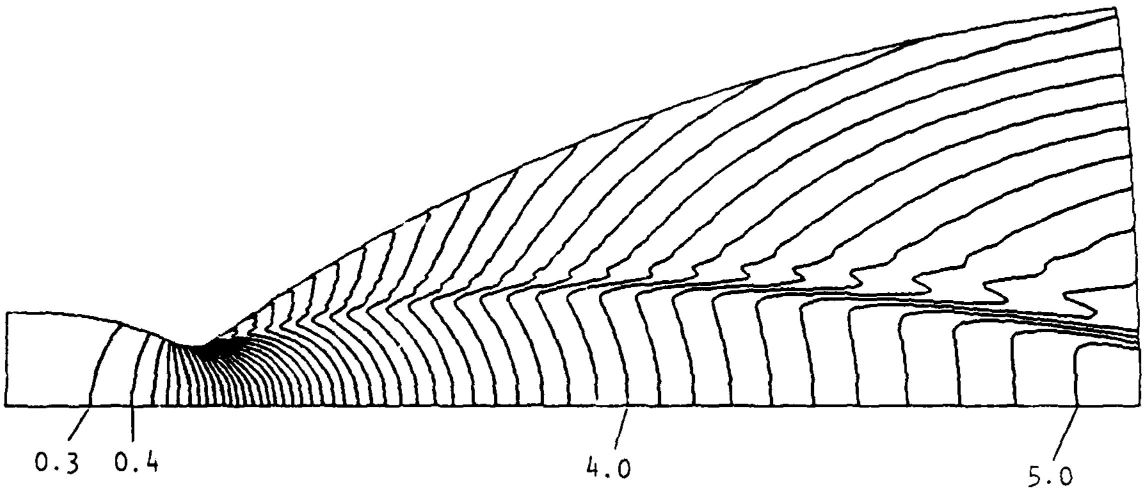


Fig. 16 Mach Number Contours of Inviscid Calculations.

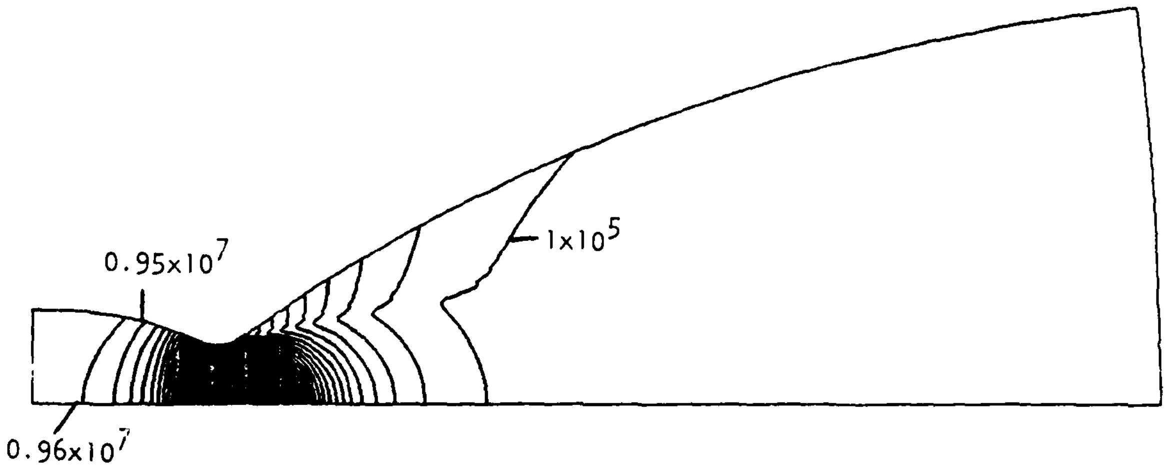


Fig. 17 Pressure Contours of Inviscid Calculations.

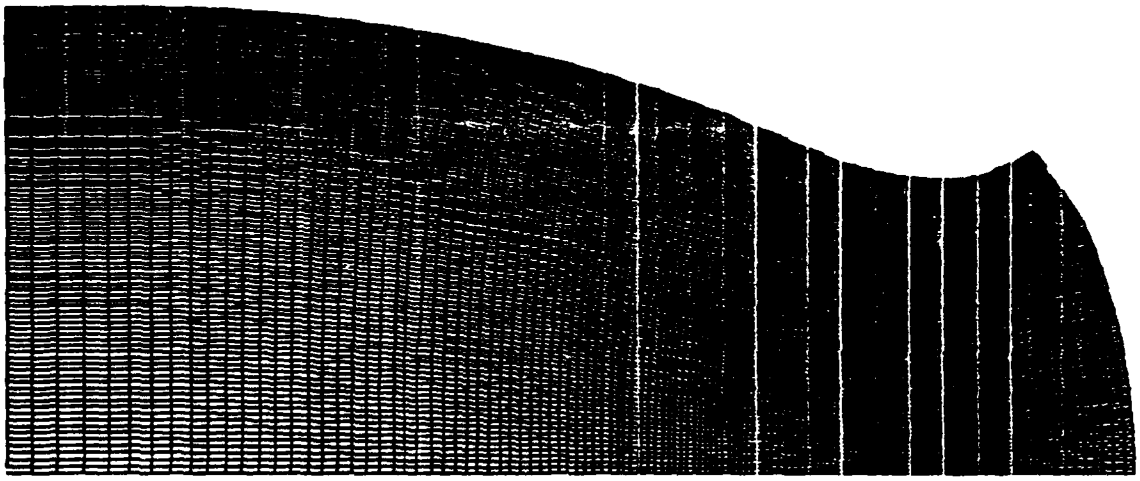


Fig. 18 120 x 100 Grid for Viscous Transonic Flow Calculations.

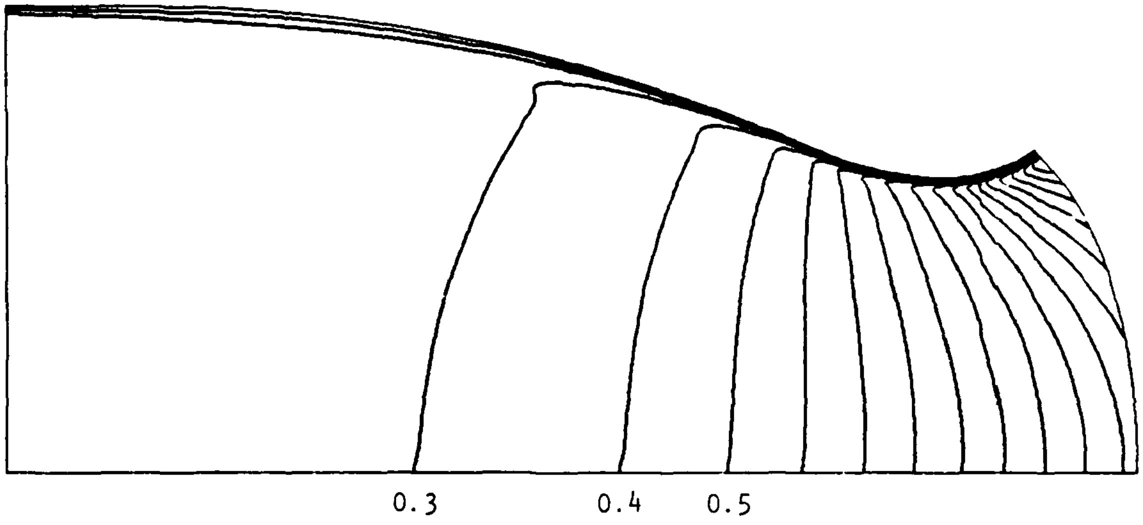


Fig. 19 Mach Number Contours of Viscous Transonic Flow.

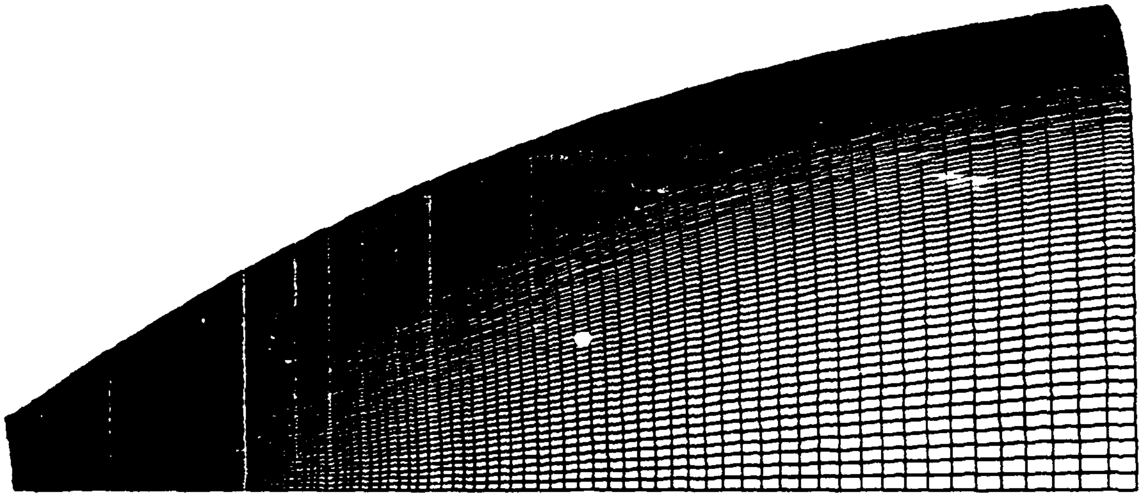


Fig. 20 145 x 100 Grid for Viscous Supersonic Flow Calculations.

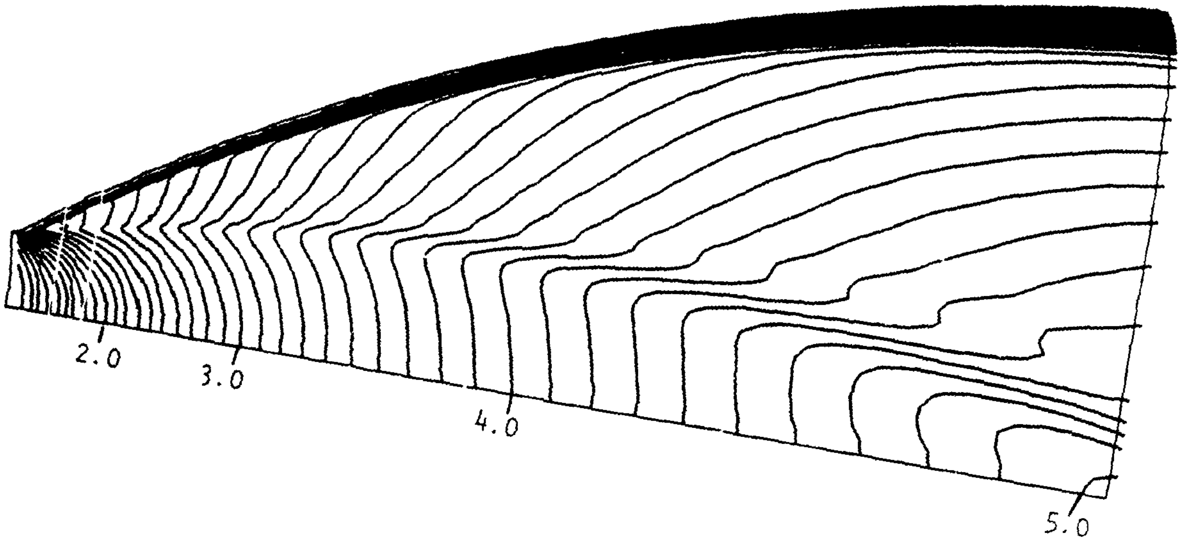


Fig. 21 Mach Number Contours of Viscous Supersonic Flow.

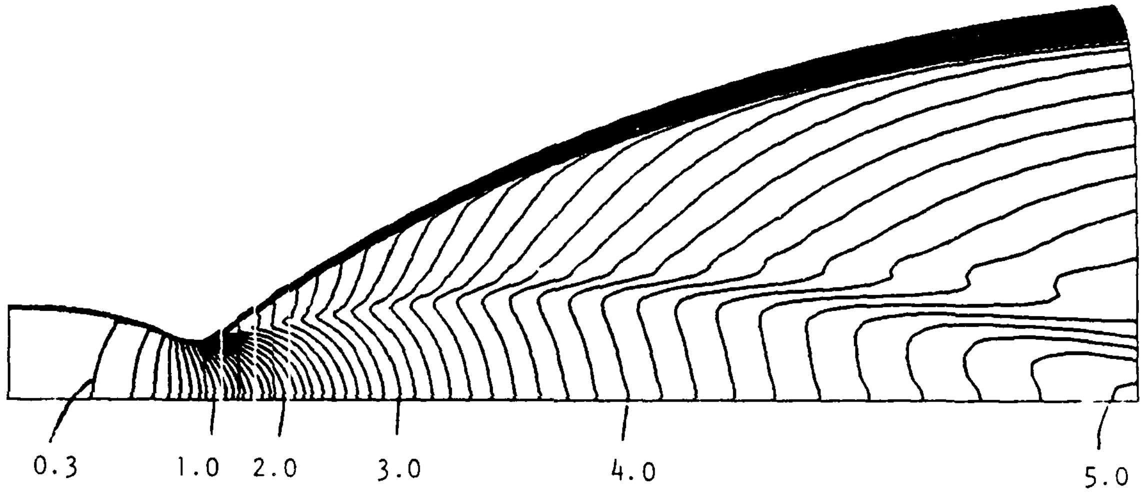


Fig. 22 Mach Number Contours of Viscous Calculations.

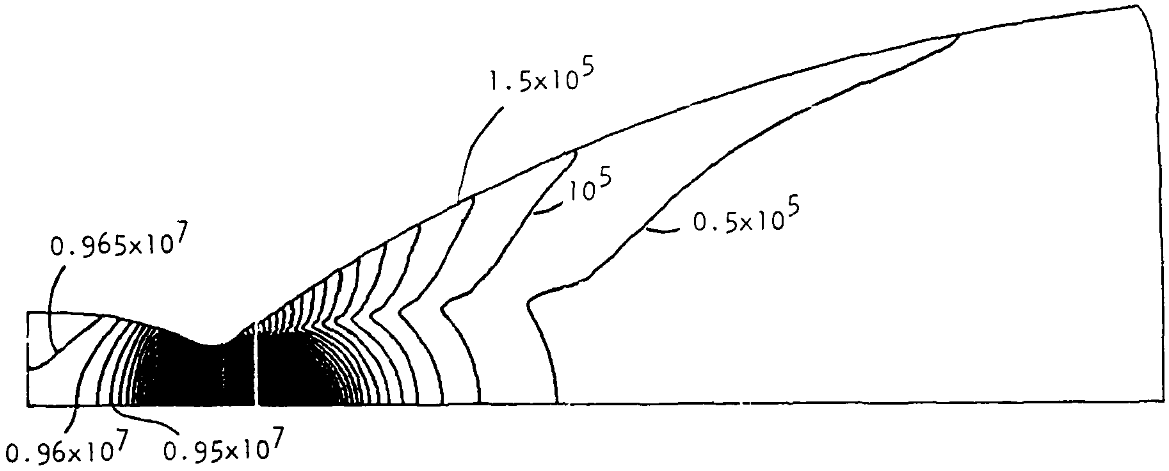


Fig. 23 Pressure Contours of Viscous Calculations.

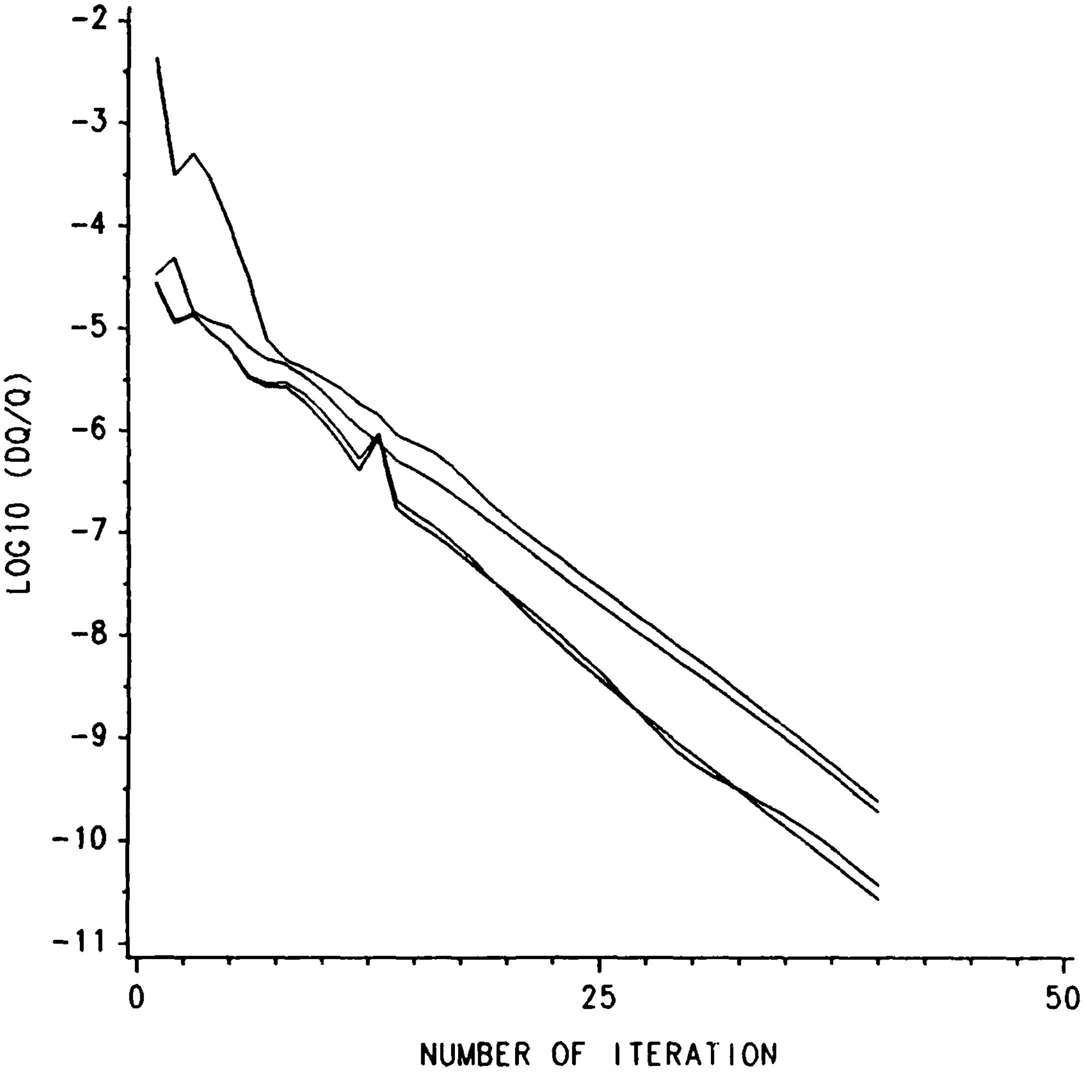


Fig. 24 Convergence for Viscous Supersonic Flow Calculations.

APPENDIX A

JACOBIAN MATRICES AND ASSOCIATED ALGEBRA FOR REAL GAS EQUATIONS

APPENDIX A

JACOBIAN MATRICES AND ASSOCIATED ALGEBRA FOR REAL GAS EQUATIONS

The present Appendix summarizes a number of important algebraic quantities for the real gas equations. In all cases, we indicate how these quantities simplify for the perfect gas case. We start by specifying the Jacobian matrices of the inviscid flux vectors E and F. These are defined right below Eqn. 1 in the main body of the report. These Jacobians are respectively, A = ∂E/∂Q and B = ∂F/∂Q. They are composed of the following terms:

$$
A \text{ or } B = \left\{ \begin{array}{cccc}
0 & K_x & K_y & 0 \\
-uU+K_x \frac{\partial P}{\partial P} & U+K_x(u+\frac{\partial P}{\partial(\rho u)}) & K_y u+K_x \frac{\partial P}{\partial(\rho v)} & K_x \frac{\partial P}{\partial e_o} \\
-vU+K_y \frac{\partial P}{\partial P} & K_x V+K_y \frac{\partial P}{\partial(\rho u)} & U+K_y(v+ \frac{\partial P}{\partial(\rho v)}) & K_y \frac{\partial P}{\partial e_o} \\
U(\frac{\partial P}{\partial \rho}-\frac{e_o+P}{\rho}) & U\partial \frac{\partial P}{(\rho u)}+K_x\frac{e_o+P}{\rho} & U\partial \frac{\partial P}{(\rho v)}+K_y\frac{e_o+P}{\rho} & U(1+\frac{\partial P}{\partial e_o})
\end{array} \right\} \tag{A.1}
$$

where U = $K_x U + K_y V$, and K = ξ, U = U for A, while K = η, U = V for B.

For the real gas case, the derivatives of pressure are given by:

$$
\begin{aligned}
\frac{\partial P}{\partial \rho} &= A_\rho - A_e \frac{1}{\rho}(\frac{e_o}{\rho} - u^2 - v^2) \\
\frac{\partial P}{\partial \rho u} &= - A_e \frac{u}{\rho} \\
\frac{\partial P}{\partial \rho v} &= - A_e \frac{v}{\rho} \\
\frac{\partial P}{\partial e_o} &= A_e \frac{1}{\rho}
\end{aligned} \tag{A.2}
$$

where,

$$
\begin{aligned}
A_\rho &= \frac{P}{\rho} B_P + \frac{P}{T} B_T \left(\frac{\partial T}{\partial \rho}\right)_e \\
A_e &= \frac{P}{T} B_T \left(\frac{\partial T}{\partial e}\right)_\rho
\end{aligned} \tag{A.3}
$$

and,

$$
B_\rho = 1 - \frac{\rho}{M_w} \left(\frac{\partial M_w}{\partial \rho}\right)_T , \quad B_T = 1 - \frac{T}{M_w} \left(\frac{\partial M_w}{\partial T}\right)_\rho \tag{A.4}
$$

For the perfect gas case, M_W is a constant, so $B\rho = B_T = 1$. In addition, $(\partial T/\partial\rho)_e = 0$, while $(\partial T/\partial e)\rho = 1/C_V$. Thus, for the perfect gas case, we have,

$$A\rho = p/\rho \tag{A.5}$$

and,

$$A_e = (\gamma-1)\rho \tag{A.6}$$

Substituting these values into the expressions for the derivatives of pressure and inserting in the matrices A and B gives the standard, perfect gas form of these Jacobians.

The modal matrices $T^{-1}P^{-1}$ and $T_B{}^{-1}P^{-1}$ are: $T^{-1}P^{-1} =$

$$\left\{\begin{array}{cccc}
1-\frac{1}{c}\frac{\partial P}{\partial\rho} & -\frac{1}{c^2}\frac{\partial P}{\partial(\rho u)} & -\frac{1}{c}\frac{\partial P}{\partial(\rho v)} & -\frac{1}{c^2}\frac{\partial P}{\partial e_o} \\
-(K_y^1 u-K_x^1 v)\ \frac{1}{\rho} & K_y^1\ \frac{1}{\rho} & -K_x^1\ \frac{1}{\rho} & 0 \\
\frac{1}{\sqrt{2}\rho}(-(K_x^1 u+K_y^1 v)+\frac{1}{c}\frac{\partial P}{\partial\rho}) & \frac{1}{\sqrt{2}\rho}(K_x^1+\frac{1}{c}\frac{\partial\rho}{\partial(\rho u)} & \frac{1}{\sqrt{2}\rho}(K_y^1+\frac{1}{c}\frac{\partial P}{\partial(\rho v)}) & \frac{1}{\sqrt{2}\rho c}\frac{\partial P}{\partial e_o} \\
\frac{1}{\sqrt{2}\rho}((K_x^1 u+K_y^1 v)+\frac{1}{c}\frac{\partial P}{\partial\rho}) & \frac{1}{\sqrt{2}\rho}(-K_x^1+\frac{1}{c}\frac{\partial P}{(\rho u)}) & \frac{1}{\sqrt{2}\rho}(-K_x^1+\frac{1}{c}\frac{\partial P}{\partial(\rho\times)}) & \frac{1}{\sqrt{2}\rho c}\frac{\partial P}{\partial e_o}
\end{array}\right\} \tag{A.7}$$

where $K = \xi$ for T_A, $K = \eta$ for T_B,

$$K_x^1 = \frac{K_x}{(K_x{}^2 + K_y{}^2)^{1/2}}\ , \text{ and } \ K_y^1 = \frac{K_y}{(K_x{}^2 + K_y{}^2)^{1/2}}$$

The inverses of these matrices, PT_A and PT_B are given by, $PT =$

$$\left\{\begin{array}{llcc}
1 & 0 & \frac{\rho}{\sqrt{2}c} & \frac{P}{\sqrt{2}c} \\
u & K_y^1\rho & \frac{\rho}{\sqrt{2}}(\frac{u}{c} + K_x^1) & \frac{\rho}{\sqrt{2}}(\frac{u}{c} - K_x^1) \\
v & -K_x^1\rho & \frac{\rho}{\sqrt{2}}(\frac{v}{c} + K_y^1) & \frac{\rho}{\sqrt{2}}(\frac{v}{c} - K_y^1) \\
\alpha & \rho(K_y^1 u+K_x^1 v) & \sqrt{2}c\ \alpha+\frac{\rho c}{\sqrt{2}}\frac{1}{\frac{\partial P}{\partial e_o}}+\frac{\rho}{\sqrt{2}}(K_x^1 u+K_y^1 v) & \frac{\rho}{\sqrt{2}c}\alpha+\frac{\rho c}{\sqrt{2}}\frac{1}{\frac{\partial P}{\partial e_o}}-\frac{P}{\sqrt{2}}(K_x^1 u+K_y^1 v)
\end{array}\right\} \tag{A.8}$$

where,

$$\alpha = \frac{1}{\frac{\partial P}{\partial e_o}}\ (u\frac{\partial P}{\partial(\rho u)} + v\frac{\partial P}{\partial(\rho v)} + \frac{\partial P}{\partial\rho})$$

Again, the derivatives of pressure are as given above, and the limiting values for perfect gases again give the familiar results of Appendix C.

For the right-hand side terms, we need the Jacobians D′ and D′′.

For the source term, the Jacobian of the algebraic (non-differentiated) terms in H is,

$$D' = \frac{1}{y} \left\{ \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ D'_{31} & D'_{32} & D'_{33} & D'_{34} \\ 0 & 0 & 0 & 0 \end{matrix} \right\} \tag{A.9}$$

where,

$$D'_{31} = \frac{\partial P}{\partial \rho} - \frac{4}{3}\frac{1}{y} v \frac{\partial \mu}{\partial \rho} + \frac{4}{3}\frac{1}{y} \mu \frac{v}{\rho} \tag{A.10a}$$

$$D'_{32} = \frac{\partial P}{\partial(\rho u)} - \frac{4}{3}\frac{1}{y} v \frac{\partial \mu}{\partial(\rho u)} \tag{A.10b}$$

$$D'_{33} = \frac{\partial P}{\partial(\rho v)} - \frac{4}{3}\frac{1}{y} v \frac{\partial \mu}{\partial(\rho v)} - \frac{4}{3}\frac{1}{y} \mu \frac{1}{\rho} \tag{A.10c}$$

$$D'_{34} = \frac{\partial P}{\partial e_o} - \frac{4}{3}\frac{1}{y} v \frac{\partial \mu}{\partial e_o} \tag{A.10d}$$

where, again the pressure derivatives are given above for both real and perfect gas variables.

Similarly, the Jacobian of the differentiated (first order) terms in H is,

$$D'' = \left\{ \begin{matrix} 0 & 0 & 0 & 0 \\ D''_{21} & D''_{22} & D''_{23} & D''_{24} \\ D''_{31} & D''_{32} & D''_{33} & D''_{34} \\ D''_{41} & D''_{42} & D''_{43} & D''_{44} \end{matrix} \right\} \tag{A.11}$$

where,

$$D''_{21} = \frac{1}{J} [- \frac{2}{3} \eta_x \frac{\partial}{\partial\eta} (v \frac{J}{y} \frac{\partial\mu}{\partial\rho} - \mu\frac{J}{y} \frac{v}{\rho})] \qquad (A.12a)$$

$$D''_{22} = \frac{1}{J} [- \frac{2}{3} \eta_x \frac{\partial}{\partial\eta} (v \frac{J}{y} \frac{\partial\mu}{\partial(\rho u)})] \qquad (A.12b)$$

$$D''_{23} = \frac{1}{J} [- \frac{2}{3} \eta_x \frac{\partial}{\partial\eta} (v \frac{J}{y} \frac{\partial\mu}{\partial(\rho v)} + \mu \frac{J}{y} \frac{1}{\rho})] \qquad (A.12c)$$

$$D''_{24} = \frac{1}{J} [- \frac{2}{3} \eta_x \frac{\partial}{\partial\eta} (v \frac{J}{y} \frac{\partial\mu}{\partial e_o})] \qquad (A.12d)$$

$$D''_{31} = \frac{1}{J} [\frac{2}{3} \eta_x \frac{J}{y} \frac{\partial\mu}{\partial\rho} \frac{\partial u}{\partial\eta} - \frac{2}{3} \eta_x \mu \frac{\partial}{\partial\eta} (\frac{J}{y} \frac{u}{\rho})$$

$$+ \frac{2}{3} \eta_y \frac{J}{y} \frac{v}{\rho} \frac{\partial\mu}{\partial\eta} - \frac{2}{3} \eta_y v \frac{\partial}{\partial\eta} (\frac{J}{y} \frac{\partial\mu}{\partial\rho})] \qquad (A.12e)$$

$$D''_{32} = \frac{1}{J}[\frac{2}{3} \eta_x \frac{J}{y} \frac{\partial\mu}{\partial(\rho u)} \frac{\partial u}{\partial\eta} + \frac{2}{3} \eta_x \mu\frac{\partial}{\partial\eta}(\frac{J}{y} \frac{1}{\rho}) - \frac{2}{3} \eta_y v\frac{\partial}{\partial\eta}(\frac{J}{y} \frac{\partial\mu}{\partial(\rho u)})] \quad (A.12f)$$

$$D''_{33} = \frac{1}{J} [\frac{2}{3} \eta_x \frac{J}{y} \frac{\partial\mu}{\partial(\rho v)} \frac{\partial u}{\partial\eta} - \frac{2}{3} \eta_y \frac{J}{y} \frac{1}{\rho} \frac{\partial\mu}{\partial\eta} - \frac{2}{3} \eta_y v\frac{\partial}{\partial\eta} (\frac{J}{y} \frac{\partial\mu}{\partial(\rho v)})] \quad (A.12g)$$

$$D''_{34} = \frac{1}{J} [\frac{2}{3} \eta_x \frac{J}{y} \frac{\partial\mu}{\partial e_o} \frac{\partial u}{\partial\eta} - \frac{2}{3} \eta_y v\frac{\partial}{\partial\eta} (\frac{J}{y} \frac{\partial\mu}{\partial e_o})] \qquad (A.12h)$$

$$D''_{41} = \frac{1}{J} [- \frac{2}{3} \eta_x \frac{\partial}{\partial\eta} (\frac{J}{y} \frac{\partial u}{\partial\rho} uv - 2\mu\frac{J}{y} \frac{uv}{\rho}) - \frac{2}{3} \eta_y \frac{\partial}{\partial\eta}(\frac{J}{y} \frac{\partial\mu}{\partial\rho}v^2 - 2\mu \frac{J}{y} \frac{v^2}{\rho}) \qquad (A.12i)$$

$$D''_{42} = \frac{1}{J} [- \frac{2}{3} \eta_x \frac{\partial}{\partial\eta} (\frac{J}{y} \frac{\partial\mu}{\partial(\rho u)} uv + \mu\frac{J}{y} \frac{v}{\rho}) - \frac{2}{3} \eta_y \frac{\partial}{\partial\eta} (\frac{J}{y} \frac{\partial\mu}{\rho u)} v^2)] \quad (A.12j)$$

$$D''_{43} = \frac{1}{J}[- \frac{2}{3} \eta_x \frac{\partial}{\partial\eta} (\frac{J}{y} \frac{\partial\mu}{\partial(\rho v)} uv + \mu\frac{J}{y} \frac{u}{\rho}) - \frac{2}{3} \eta_y \frac{\partial}{\partial\eta} (\frac{J}{y} \frac{\partial\mu}{\partial(\rho v)} v^2 + 2\mu\frac{J}{y} \frac{v}{\rho}) \qquad (A.12k)$$

$$D''_{44} = \frac{1}{J} [- \frac{2}{3} \eta_x \frac{\partial}{\partial\eta} (\frac{J}{y} \frac{\partial\mu}{\partial e_o} uv) - \frac{2}{3} \eta_y \frac{\partial}{\partial\eta} (\frac{J}{y} \frac{\partial\mu}{\partial e_o} v^2)] \qquad (A.12L)$$

The derivatives of μ (and later k and T) with respect to Q require some comment. By taking $\mu = \mu(\rho,e)$ and $k = k(\rho,e)$ and using the chain rule, we obtain,

$$d\mu = (\frac{\partial\mu}{\partial\rho})_e \, d\rho + (\frac{\partial\mu}{\partial e})_\rho \, de \qquad (A.13)$$

with a similar expression for the changes in k and T. Derivatives of μ, k, or T are then defined in terms of the derivatives $(\partial\mu/\partial\rho)_e$ and $(\partial\mu/\partial e)_\rho$ etc. which are obtained from the property tables. (Note, the $\partial\mu/\partial\rho$) appearing, for example, in Eqn. A.12a is not equal to $(\partial\mu/\partial\rho)_e$ because the term in Eqn. A.12a is obtained with Q as the independent variable, whereas the latter considers ρ and e as independent variables.

The Jacobians for the viscous terms, B_{v1} and B_{v2}, are:

$$B_{v1} = \frac{\partial Q_1}{\partial Q} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -u/\rho & 1/\rho & 0 & 0 \\ -v/\rho & 0 & 1/\rho & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{A.14}$$

while the matrix B_{v2} is given by,

$$B_{v2} = \frac{\partial Q_2}{\partial Q} = \begin{pmatrix} \frac{\partial T}{\partial \rho} & \frac{\partial T}{\partial \rho u} & \frac{\partial T}{\partial \rho v} & \frac{\partial T}{\partial e_o} \\ -2u^2/\rho & 2u/\rho & 0 & 0 \\ -2v^2/\rho & 0 & 2v/\rho & 0 \\ -2uv/\rho & v/\rho & u/\rho & 0 \end{pmatrix} \tag{A.15}$$

The derivatives of temperature are first expanded by the chain rule expression given in Eqn. A.13 above. Because temperature is a primary variable of interest, we re-express this here as,

$$dT = \left(\frac{\partial T}{\partial \rho}\right)_e d\rho + \left(\frac{\partial T}{\partial e}\right)_\rho de \tag{A.16a}$$

and we introduce the coefficients c_p and c_e such that,

$$dT = c_\rho d_\rho + c_e de \tag{A.16b}$$

In terms of these variables, we can express the four derivatives of temperature as:

$$\frac{\partial T}{\partial \rho} = c_\rho + c_e \frac{1}{\rho} \left(- \frac{e_o}{\rho} + (u^2 + v^2)\right) \qquad \text{(A.17a)}$$

$$\frac{\partial T}{\partial (\rho u)} = - c_e \frac{u}{\rho} \qquad \text{(A.17b)}$$

$$\frac{\partial T}{\partial (\rho v)} = - c_e \frac{v}{\rho} \qquad \text{(A.17c)}$$

$$\frac{\partial T}{\partial e_o} = - c_e \frac{1}{\rho} \qquad \text{(A.17d)}$$

For completeness, we also define the speed of sound. The speed of sound, c, is given by:

$$c^2 = \frac{P}{\rho} B_\rho + \frac{P}{T} B_T \left(\frac{\partial T}{\partial \rho} + \frac{P}{\rho^2} \frac{\partial T}{\partial e}\right) = A_\rho + \frac{A_e}{\rho^2} P \qquad \text{(A.18)}$$

APPENDIX B

ROE AVERAGING

APPENDIX B
ROE AVERAGING

Roe averaging is used to ensure conservation even when some terms are expressed in non-conservative form. Details of the procedure are given in Ref. 4. In brief, Roe averaging is given by,

$$\rho_{i+1/2} = \frac{\rho_{i+1}\sqrt{\rho_{i+1}} + \rho_i\sqrt{\rho_i}}{\sqrt{\rho_{i+1}} + \sqrt{\rho_i}} \tag{B.1}$$

$$u_{i+1/2} = \frac{u_{i+1}\sqrt{\rho_{i+1}} + u_i\sqrt{\rho_i}}{\sqrt{\rho_{i+1}} + \sqrt{\rho_i}} \tag{B.2}$$

$$v_{i+1/2} = \frac{v_{i+1}\sqrt{\rho_{i+1}} + v_i\sqrt{\rho_i}}{\sqrt{\rho_{i+1}} + \sqrt{\rho_i}} \tag{B.3}$$

$$(h_f)_{i+1/2} = \frac{(h_f)_{i+1}\sqrt{\rho_{i+1}} + (h_f)_i\sqrt{\rho_i}}{\sqrt{\rho_{i+1}} + \sqrt{\rho_i}} \tag{B.4}$$

The total internal energy, e_o, is related to the stagnation enthalpy by,

$$\rho h_o = e_o + p \tag{B.5}$$

where p is the pressure. P can be evaluated from,

$$P = \frac{\overline{\gamma} - 1}{\overline{\gamma}} (\rho h_o - 1/2\rho(u^2 + v^2) \tag{B.6}$$

and assuming an arithmetic averaging for which is given by,

$$\overline{\gamma} = 1 + \frac{R/M_W}{e/T} \tag{B.7}$$

APPENDIX C

TIME-ITERATIVE SOLUTIONS OF VISCOUS SUPERSONIC FLOWS

(This document is bound separately)

APPENDIX D

COMPUTER CODE LISTINGS

(This document is bound separately)