1982

Numerical Modeling of the Internal Ballistics of a Gas Gun.

Ameziane Harhad
Louisiana State University and Agricultural & Mechanical College

8229502

Harhad, Ameziane

NUMERICAL MODELING OF THE INTERNAL BALLISTICS OF A GAS GUN

The Louisiana State University and Agricultural and Mechanical Col. PH.D. 1982

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __√__.

1. Glossy photographs or pages ______
2. Colored illustrations, paper or print ______
3. Photographs with dark background ______
4. Illustrations are poor copy ______
5. Pages with black marks, not original copy ______
6. Print shows through as there is text on both sides of page ______
7. Indistinct, broken or small print on several pages ______
8. Print exceeds margin requirements ______
9. Tightly bound copy with print lost in spine ______
10. Computer printout pages with indistinct print __√__
11. Page(s) ____________ lacking when material received, and not available from school or author.
12. Page(s) ____________ seem to be missing in numbering only as text follows.
13. Two pages numbered ____________. Text follows.
14. Curling and wrinkled pages ______
15. Other ____

NUMERICAL MODELING OF THE INTERNAL BALLISTICS OF A GAS GUN

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Mechanical Engineering

by
Ameziane Harhad
B.S., The University of Algiers, 1976
M.S., Louisiana State University, 1978
August, 1982

ACKNOWLEDGEMENT

The author wishes to express his sincere appreciation to Dr. Robert Courter for serving as his major professor and for his helpful suggestions, cooperation and guidance in the course of preparing this manuscript.

The author acknowleges the help, encouragement and advice given him by Dr. O. A. Arnas, Dr. P. H. Miller and Dr. D. W. Yannitell

The author extends special thanks to professors Dr. J. A. Hildebrant and Dr. R. G. Hussey for their time and consideration as members of his graduate committee.

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

u	axial velocity (M/S)
v	radial velocity (M/S)
P	Pressure (PA)
ρ	density (KG/M^3)
T	Temperature ($^\circ$K)
x	axial coordinate (M)
y	radial coordinate (M)
t	time (S)
a	speed of sound (M/S)
γ	specific heat ratio
R	gas constant (J/MOLE.$^\circ$K)
μ	first coefficient of viscosity (PA $\cdot$ S)
λ	second coefficient of viscosity (PA $\cdot$ S)
(ξ,η,τ)	coordinates in the computational space
Mp	piston mass (KG)
Xp	piston displacement (M)
Ap	piston area (M^2)
D	friction force (N)
Yw(x)	wall shape
P_c	charge pressure (PA)
Ri, Re	inlet radius, outlet radius (in the discharge chamber)
σ	parameter defined in the grid generation of the discharge chamber
W	vector defined in the method of characteristic (Appendix A)

Λ, F and ψ	scalars defined in the barrel chamber flow
QROT, QUT, QVT, and QPT	scalars defined in the main chamber flow

ABSTRACT

A numerical procedure is applied to three principal flow models of a gas gun, the discharge chamber, the main chamber, and the barrel chamber. In the first two models, an explicit method, the second order MacCormack scheme, is used to solve the Navier-Stokes equations for two-dimensional (axisymmetric), time dependent, compressible flow. The fluid is assumed to be a perfect gas. The flow boundaries may be arbitrary curved and time dependent. Transformations have been used to map the physical space into a computational space with uniform grid spacing. A grid generation surface has been achieved for various wall shapes. The third model investigates a detailed one-dimensional time dependent flow. In this case, the method of characteristics is used to solve the quasi-linear non-homogeneous partial differential equations of the first order. The numerical results obtained from the present solution are compared with the one-dimensional solution. The present technique shows better correlation with the experimental data.

CHAPTER I

INTRODUCTION

The two fields of compressible flow theory and numerical solution of partial differential equations were combined more than thirty years ago when Richtmyer reported a study of one dimensional waves (Ref. [1]). Since that time, the hybrid field of computational gas dynamics has undergone much growth. At the present time methods are available for the analysis of three-dimensional, unsteady, compressible, viscous flows involving complicated geometries, both internal and external. Still, the application of the numerical methods to various problems is somewhat of an art, and each solution achieved is evidence of ingenuity and perception on the part of the researcher. In other words, we are not yet to the point where an engineer can just push a button on an all-encompassing fluid dynamics computer program and expect a unique and correct result.

The field of ballistics is particularly amenable to the methods of computational gasdynamics. Several years ago the author performed a one dimensional analysis of the internal ballistics of a dual piston gas gun (Ref. [2]). That work involved reducing the gun equations to a system of ordinary differential equations which could be solved simultaneously for the gun parameters. The results of the analysis were very encouraging, but they provided no details of the complicated gasdynamic processes at work within the gun chambers. In the present study a two-dimensional (axisymmetric) study of the transient behavior of the compressible viscous gases within the gun chamber is performed. However, the main purpose of the study is not to determine the flow characteristics for a specific gun, but rather to develop and demonstrate methods for dealing with internal chamber flows which involve time dependent boundaries, variable inlet streams and various wall shapes.

The studies in this work have been divided into three principal flow models. In the first model we have used a finite difference method to solve the governing partial differential equations for a discharge chamber flow. The method of characteristics is used to solve for the flow properties at the boundaries. Transformations have been used to map the physical space into a computational space with uniform grid spacing. Some of the walls are vertical. In order to eliminate the singularity special functions have been used to fit the shape of the wall. A complete surface generation has been achieved. The second flow model, representative of a main piston chamber flow, incorporates a two-dimensional, time-dependent flow for which the boundaries are also time dependent. In this model a surface generation technique with a grid which is both spatially variable and time-dependent is required to handle the computations in the variable-inflow regions and at the moving boundaries. The third model investigates a detailed one-dimensional time dependent flow representative of a barrel flow. The method of characteristics, which is a very accurate method for solving hyperbolic partial differential equations, is used. This numerical technique presents another advantage in the treatment of the shocks. The shock wave is treated as a moving boundary separating two regions of continuous flow. The method of characteristics combined with the Rankine-Hugoniot equations are combined to solve for the flow properties.

Throughout the analysis it has been necessary to utilize certain devices which help to ensure the accuracy and stability of the numerical methods. An explicit artificial viscosity has been used to stabilize regions of large pressure rise such as shock waves. In regions of rapid expansion resulting in a change of sign of velocity components a velocity averaging technique has been employed. Finally, grid generation procedures are employed to improve the accuracy of the calculations.

CHAPTER II

MATHEMATICAL MODEL OF INTERNAL CHAMBER FLOWS

The analysis of gas guns encompasses a variety of internal flow problems. A typical arrangement consists of three parts; namely, the discharge chamber, the main chamber, and the barrel chamber. A schematic representation is shown in Figure 1. The gas flows through the discharge chamber and the main chamber are treated as - two-dimensional, time dependent flows. The flow within the barrel chamber is treated as one dimensional and time dependent.

A. DISCHARGE CHAMBER FLOW

1. Governing Equations

In the absence of body force, the two-dimensional equations for a viscous, non heat conducting flow in cylindrical coordinates with axial symmetry and zero azimuthal velocity may be written as follows:

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} + \rho \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + \rho \frac{v}{y} = 0 \qquad (1)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{1}{\rho} \frac{\partial p}{\partial x} = \mu \left[\frac{\partial^2 u}{\partial y^2} + \frac{4}{3} \frac{\partial^2 u}{\partial x^2} + \frac{1}{3} \frac{\partial^2 v}{\partial y \, \partial x} \right]$$

$$+ \frac{\mu}{\rho y} \left[\frac{1}{3} \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right] \qquad (2)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{1}{\rho}\frac{\partial P}{\partial y} = \mu\ [\ \frac{\partial^2 v}{\partial x^2} + \frac{4}{3}\frac{\partial^2 v}{\partial y^2} + \frac{1}{3}\frac{\partial^2 U}{\partial y\,\partial x}\ ]$$

$$+ \frac{4}{3}\frac{\mu}{\rho}\ [\ \frac{\partial v}{\partial y} - \frac{v}{y}\ ] \tag{3}$$

$$\frac{\partial P}{\partial t} + u\frac{\partial P}{\partial x} + v\frac{\partial P}{\partial y} - a^2(\ \frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y}\ )$$

$$= (\gamma - 1)\mu\{\frac{4}{3}[\ (\ \frac{\partial u}{\partial x}\ )^2 + (\ \frac{\partial v}{\partial y}\ )^2\ ] + [\ (\frac{\partial u}{\partial y})^2 + (\frac{\partial v}{\partial x})^2\ ]$$

$$+ [\ 2\frac{\partial u}{\partial y}\frac{\partial v}{\partial x} - \frac{4}{3}\frac{\partial u}{\partial x}\frac{\partial v}{\partial y}] + [\frac{4}{3}\frac{v^2}{y^2} - \frac{4}{3}\frac{v}{y}\ (\frac{\partial v}{\partial y} + \frac{\partial u}{\partial x})\ ]\} \tag{4}$$

$$P = \rho R T, \quad a^2 = \gamma P/\rho \tag{5}$$

where ρ is the density, u is the velocity in the axial direction, v is the velocity in the radial direction, P is the pressure, t is the time, a is the speed of sound, R is the engineering gas constant, γ is the ratio of the specific heats, y is the radial distance from the axis of symmetry, and μ is the first coefficient of viscosity. The bulk viscosity in the viscous terms is taken to be zero, that is, $\lambda = -\frac{2}{3}\mu$, where λ is the second coefficient of viscosity. The system of equations (1) - (5) is closed, in the sense that there are as many equations as unknowns. The basic unknowns are u, v, ρ, P, and T.

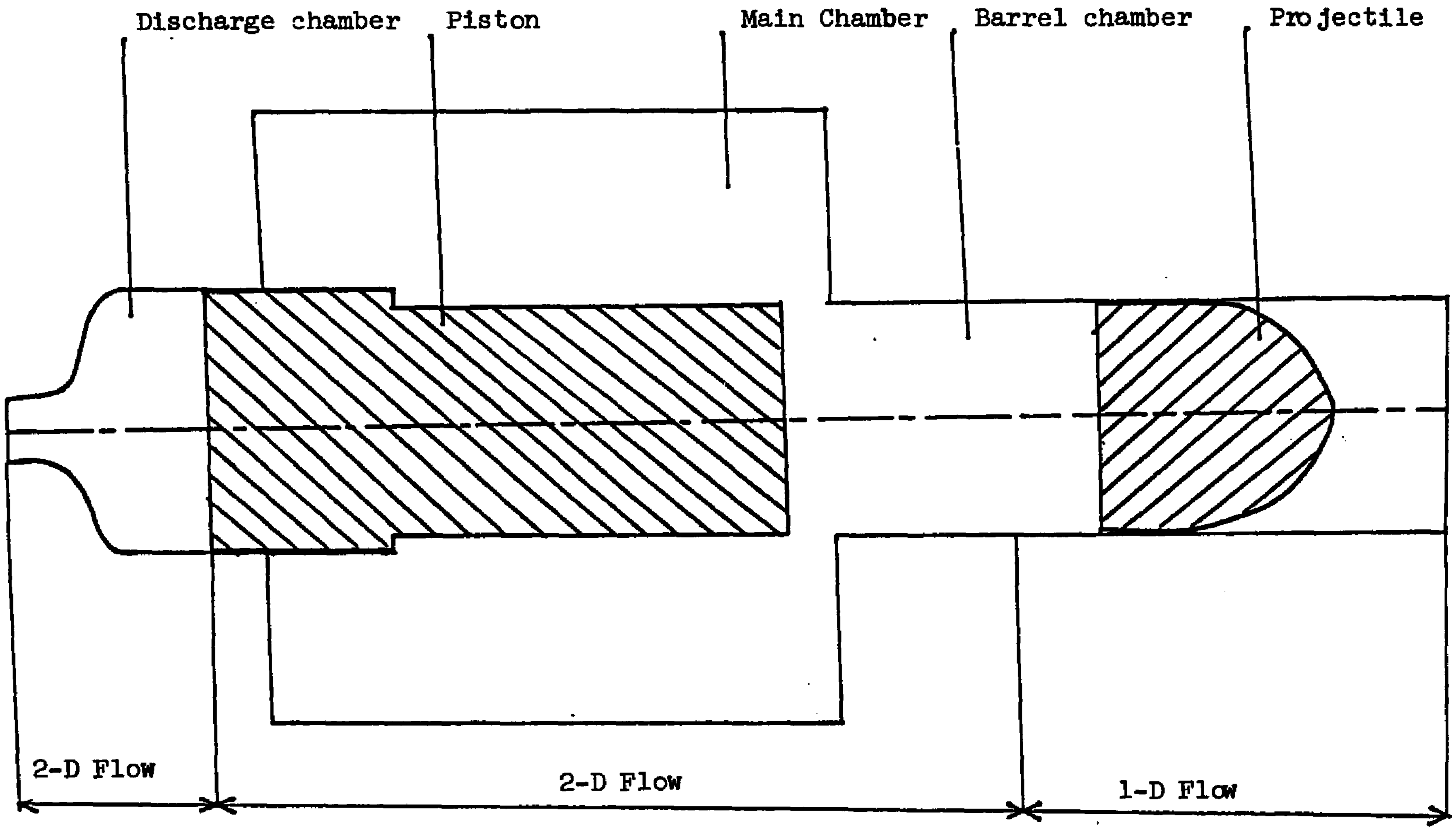


Figure 1. Schematic illustration of a gas gun.

2. Physical and Computational Flow Spaces.

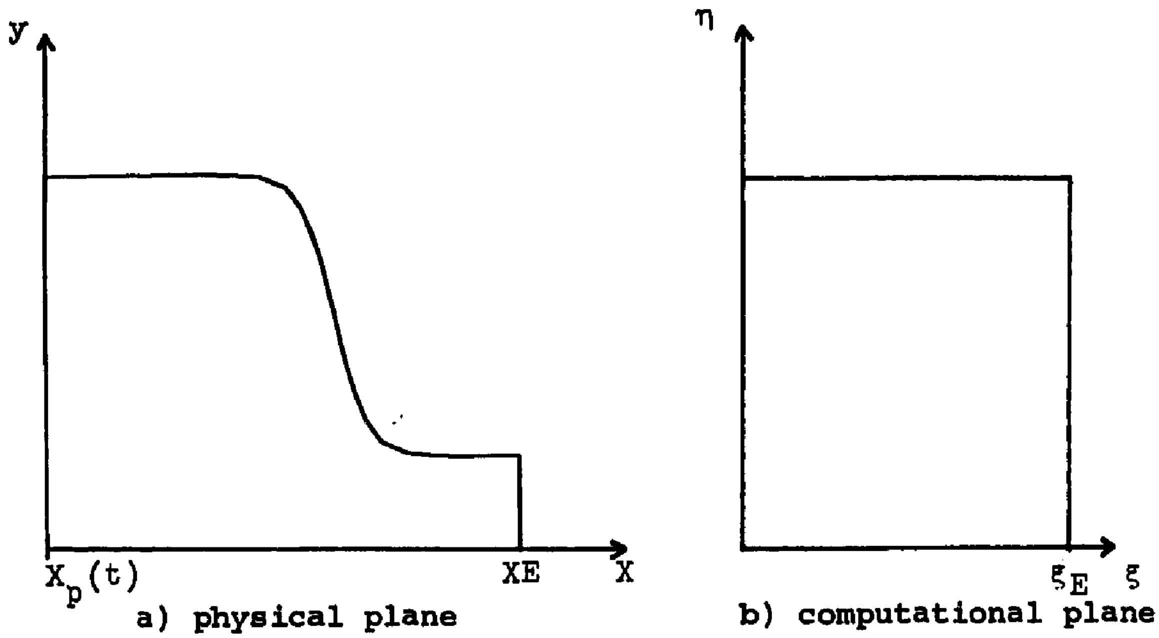


Figure 2. Schematic illustration of the discharge chamber.

The flow region enclosed by curved boundaries may be mapped into a rectangle by means of the following transformations, as shown in Figure 2.

$$\xi(x,t) = \frac{1}{\xi_E} \int_{X_{p(t)}}^{x} \sqrt{1 + y_w'^2(s)}\ ds \tag{6}$$

$$\eta(x,y) = \frac{y}{Y_{w(x)}} \tag{7}$$

$$\tau = t \tag{8}$$

where $y_w(x)$ is the wall shape as a function of x, $x_p(t)$ is the piston displacement as a function of t, and

$$\xi_E = \int_{x_p(t)}^{xE} \sqrt{1 + y_w'^2 (x)} \; dx$$

The transformation ξ stretches the axial coordinate x, such that a finite difference grid with uniform $\Delta \xi$ will concentrate the grid in regions of large curvatures. The functions ξ and η vary from 0 to 1.

The spatial and the time partial derivatives are then,

$$\frac{\partial}{\partial x} = w \frac{\partial}{\partial \xi} + \alpha \frac{\partial}{\partial \eta} \tag{9}$$

$$\frac{\partial}{\partial y} = \beta \frac{\partial}{\partial \eta} \tag{10}$$

$$\frac{\partial}{\partial t} = \delta \frac{\partial}{\partial \xi} + \frac{\partial}{\partial \tau} \tag{11}$$

Where $w = \dfrac{\sqrt{1 + y_w'^2 (x)}}{\xi_E}$, $\alpha = -\eta \beta \dfrac{dy_w}{dx}$

$\beta = \dfrac{1}{Y_w(x)}$, and $\delta = - \sqrt{1 + Y_w'^2 (X_p)} \dfrac{d X_p}{dt}$

3. Transformed Governing Equations.

The governing Equations (1) - (4) can be written in the transformed plane as follows

$$\frac{\partial \rho}{\partial \tau} + u^* \frac{\partial \rho}{\partial \xi} + v^* \frac{\partial \rho}{\partial \eta} + \rho \left(w \frac{\partial u}{\partial \xi} + \alpha \frac{\partial u}{\partial \eta}\right) = -\left[\rho \beta \frac{\partial v}{\partial \eta} + \rho \frac{v}{\eta}\right] \tag{12}$$

$$\frac{\partial u}{\partial \tau} + u^*\frac{\partial u}{\partial \xi} + v^*\frac{\partial u}{\partial \eta} + \frac{1}{\rho}\left(w\,\frac{\partial P}{\partial \xi} + \alpha\,\frac{\partial P}{\partial \eta}\right) = \frac{\mu}{\rho}\,[\beta\,\frac{\partial}{\partial \eta}\left(\beta\,\frac{\partial v}{\partial \eta}\right)$$

$$+ \frac{4}{3}\left(w\,\frac{\partial}{\partial \xi} + \alpha\,\frac{\partial}{\partial \eta}\right)\left(w\frac{\partial u}{\partial \xi} + \alpha\,\frac{\partial u}{\partial \eta}\right) + \frac{1}{3}\left(w\,\frac{\partial}{\partial \xi} + \alpha\,\frac{\partial}{\partial \eta}\right)\left(\beta\,\frac{\partial v}{\partial \eta}\right)]$$

$$+ \frac{\mu}{\rho\bar{\eta}}\left[\frac{1}{3}\left(w\,\frac{\partial v}{\partial \xi} + \alpha\,\frac{\partial v}{\partial \eta}\right) + \beta\,\frac{\partial u}{\partial \eta}\right] \tag{13}$$

$$\frac{\partial v}{\partial \tau} + u^*\frac{\partial v}{\partial \xi} + v^*\frac{\partial v}{\partial \eta} + \frac{\beta}{\rho}\,\frac{\partial P}{\partial \eta} = \frac{\mu}{\rho}\,[\left(w\,\frac{\partial}{\partial \xi} + \alpha\,\frac{\partial}{\partial \eta}\right)\left(\frac{\partial v}{\partial \xi} + \alpha\,\frac{\partial v}{\partial \eta}\right)$$

$$+ \frac{4}{3}\,\beta\,\frac{\partial}{\partial \eta}\left(\beta\,\frac{\partial v}{\partial \eta}\right) + \frac{1}{3}\left(w\,\frac{\partial}{\partial \xi} + \alpha\,\frac{\partial}{\partial \eta}\right)\left(\beta\,\frac{\partial u}{\partial \eta}\right)]$$

$$+ \frac{4}{3\rho}\,\frac{\mu}{\bar{\eta}}\left(\beta\,\frac{\partial v}{\partial \eta} - \frac{v}{\bar{\eta}}\right) \tag{14}$$

$$\frac{\partial P}{\partial \tau} + u^*\frac{\partial P}{\partial \xi} + v^*\frac{\partial P}{\partial \eta} - a^2\left(\frac{\partial \rho}{\partial t} + u^*\frac{\partial \rho}{\partial \xi} + v^*\frac{\partial \rho}{\partial \eta}\right)$$

$$= (\gamma - 1)\mu\{\frac{4}{3}\,[\left(w\,\frac{\partial u}{\partial \xi} + \alpha\,\frac{\partial u}{\partial \eta}\right)^2 + \left(\beta\,\frac{\partial v}{\partial \eta}\right)^2] + [2\beta\,\frac{\partial u}{\partial \eta}\left(w\,\frac{\partial v}{\partial \xi} + \alpha\,\frac{\partial v}{\partial \eta}\right)$$

$$- \frac{4}{3}\left(w\,\frac{\partial u}{\partial \xi} + \alpha\,\frac{\partial u}{\partial \eta}\right)\left(\beta\,\frac{\partial v}{\partial \eta}\right)] + [\left(w\,\frac{\partial v}{\partial \xi} + \alpha\,\frac{\partial v}{\partial \eta}\right)^2 + \left(\beta\,\frac{\partial u}{\partial \eta}\right)^2]$$

$$+ [\frac{4}{3}\,\frac{v^2}{\bar{\eta}^2} - \frac{4}{3}\,\frac{v}{\bar{\eta}}\left(\beta\,\frac{\partial v}{\partial \eta} + \left(w\,\frac{\partial u}{\partial \xi} + \alpha\,\frac{\partial u}{\partial \eta}\right)\right)]\} \tag{15}$$

where,

$u^* = \delta + w\,u,$

$v^* = \beta\,v + \alpha\,u,$

$\bar{\eta} = \dfrac{\eta}{\beta}$

The transformed Navier-Stokes equations are somewhat more complicated than the original ones, because new coefficients are introduced, but these equations offer several advantages. The main advantage is that the physical boundary surface can be mapped into a rectangular domain surface in the transformed plane. Another significant aspect of the transformation is that the grid can be concentrated in regions that experience rapid changes in flow field gradients or variables.

4. Numerical Method.

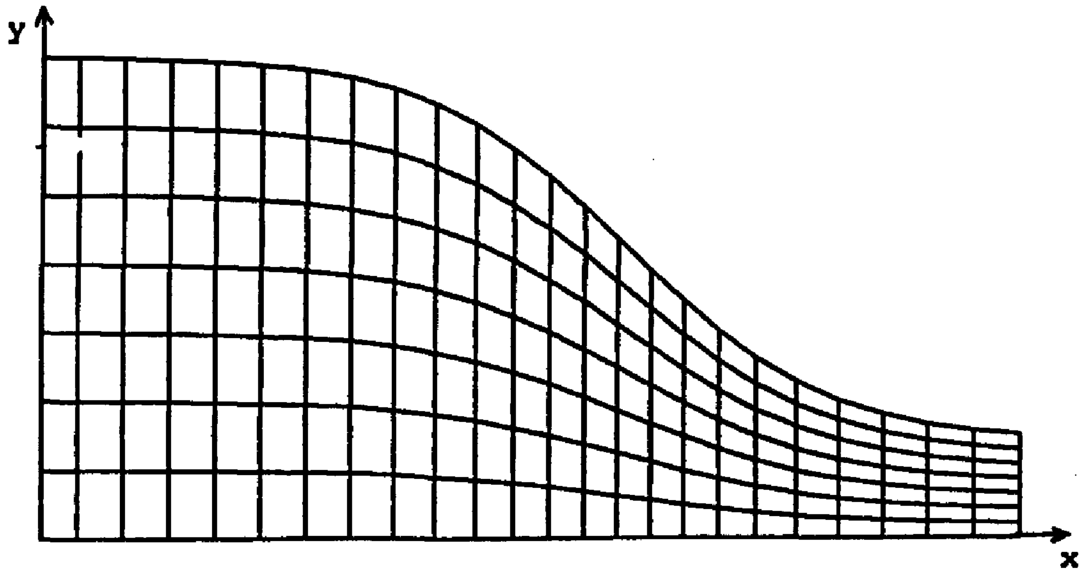
At the present time, four numerical methods are applicable to the solution of Navier-Stokes equations. Namely, they are the method of finite differences, the method of characteristics, the method of integral relations, and the fluid-in-cell method. The first two methods are used in this problem. The method of finite differences consists of replacing derivatives by finite differences, such that the discretized forms are consistent with the physics of the Navier Stokes equations. If the physical parameters in the neighborhood of any point in the flow field are Taylor-expandable and their Taylor series are rapidly convergent, then the discretization errors can be evaluated and kept within tolerable limits by considering a sufficiently small neighborhood. However, in real problems, the flow parameters may not be Taylor-expandable. In this case a special treatment is needed, one such is the artificial viscosity. Sometimes even if the flow parameters are Taylor-expandable, their Taylor series start diverging and begin converging only when certain terms are taken into account. Such terms are the fourth order dissipation, their order being higher than the second order of the most finite difference schemes. These fourth order damping terms have been found helpful in the case of problems with non-smooth initial data.

The method of characteristics is to determine the characteristic curves and the corresponding compatibility equations which can be solved simultaneously at grid points for flow properties. A characteristic curve is a curve in the solution plane in which the governing partial differential equations may be combined to form a set of ordinary differential equations.

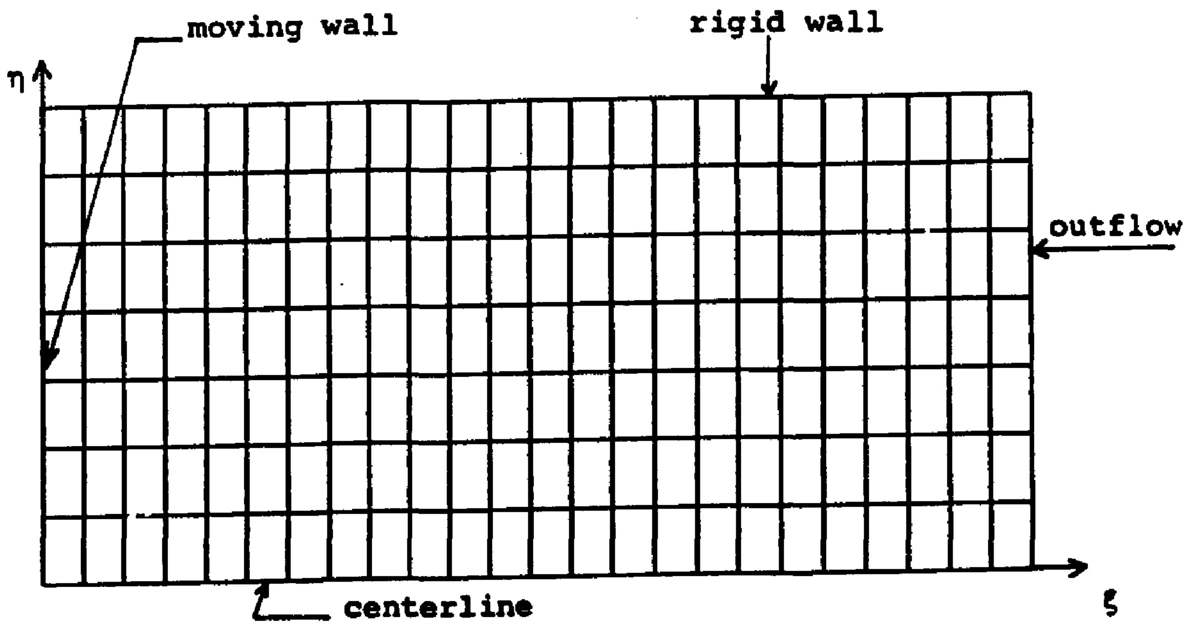
A numerical algorithm for determining the solution at a given point in the transformed plane can now be presented. The computational plane, shown in Figure 3, is divided into two sets of mesh points, the interior grid points and the boundary grid points.

4.1 Interior Grid Points

The interior grid points are computed using the explicit predictor-corrector finite difference method presented by MacCormack[3] in 1969. MacCormack's method has been used successfully by a number of investigators. The finite difference approximates the governing equations to second-order accuracy in space and time, are simple to program, but are conditionally stable. In this case



a) Physical computational mesh.



b) Rectangular computational mesh

Figure 3. Physical and computational domains

it seemed advantageous to use the explicit scheme. The governing equations are left in non conservative form in order to apply the boundary conditions more easily. As an example, the finite difference equations approximating Equation (13) are,

$$\Delta U_L = -\Delta t\,[\,(\delta^N_{L,M} + w_{L,M}\,U^N_{L,M})\,\frac{U^N_{L,M}\quad U^N_{L-1,M}}{\Delta x} + (\beta_{L,M}\,V^N_{L,M}$$

$$+ \alpha_{L,M}\,U^N_{L,M})\,\frac{U^N_{L,M} - U^N_{L,M-1}}{\Delta y} - \frac{1}{\rho^N_{L,M}}\,(w_{L,M}\,\frac{P^N_{L,M} - P^N_{L-1,M}}{\Delta x}$$

$$+ \alpha_{L,M}\,\frac{P^N_{L,M} - P^N_{L,M-1}}{\Delta y}\,)] + \mu\,\frac{\beta_{L,M}}{\rho^N_{L,M}}\,\frac{\Delta t}{\Delta y}\,[\beta_{L,M}\,\frac{V^N_{L,M+1} - V^N_{L,M}}{\Delta y}$$

$$- \beta_{L,M}\,\frac{V^N_{L,M} - V^N_{L,M-1}}{\Delta y} + \frac{4}{3}\,\frac{\mu \cdot w_{L,M}}{\rho_{L,M}}\,\frac{\Delta t}{\Delta x}\,[\,(w_{L,M}\,\frac{U^N_{L+1,M} - U_{L,M}}{\Delta x}$$

$$+ \frac{\alpha_{L,M}}{2}\,\frac{U^N_{L+1,M+1} - U^N_{L+1,M-1}}{2\Delta y} - w_{L,M} \cdot \frac{U^N_{L,M} - U^N_{L-1,M}}{\Delta x}$$

$$- \frac{\alpha_{L,M}}{2} \cdot \frac{U^N_{L-1,M+1} - U^N_{L-1,M-1}}{2\Delta y}\,] + \frac{4}{3}\,\frac{\mu \cdot \alpha_{L,M}}{\rho_{L,M}}\,\frac{\Delta t}{\Delta y}$$

$$[\frac{w_{L,M}}{2}\,\frac{U^N_{L+1,M+1} - U_{L-1,M+1}}{2\Delta x} + \alpha_{L,M}\,\frac{U^N_{L,M+1} - U^N_{L,M}}{\Delta y}$$

$$- \frac{w_{L,M}}{2} \cdot \frac{U^N_{L+1,M-1} - U^N_{L-1,M-1}}{2\Delta x} - \alpha_{L,M} \cdot \frac{U^N_{L,M} - U^N_{L,M-1}}{\Delta y}\,]$$

$$+ \frac{1}{3}\,\frac{\mu \cdot w_{L,M}}{\rho_{L,M}}\,\frac{\Delta t}{\Delta x}\,[\frac{\beta_{L,M}}{2}\,\frac{V^N_{L+1,M+1} - V^N_{L+1,M-1}}{2\Delta y}$$

$$- \frac{\beta_{L,M}}{2} \frac{V^N_{L-1,M+1} - V^N_{L-1,M-1}}{2\Delta y}] + \frac{1}{3}\frac{\mu \cdot \alpha_{L,M}}{\rho_{L,M}} \frac{\Delta t}{\Delta y}[\beta_{L,M} \cdot \frac{V^N_{L,M+1} - V^N_{L,M}}{\Delta y}$$

$$- \beta_{L,M} \cdot \frac{V^N_{L,M} - V^N_{L,M-1}}{\Delta y}] + \frac{\mu}{N\rho_{L,M}} \frac{\Delta t}{\Delta y} [\frac{1}{3}(w_{L,M} \cdot \frac{V^N_{L+1,M} - V^N_{L-1,M}}{2\Delta x}$$

$$+ \alpha_{L,M} \frac{V^N_{L,M+1} - V^N_{L,M+1}}{2\Delta y}] + \beta_{L,M} \frac{U^N_{L,M+1} - U^N_{L,M-1}}{2\Delta y}] \tag{16}$$

For the predictor step, we have

$$\overline{U}^{N+1}_{L,M} = U_{L,M} + \Delta U_{L,M} \tag{17}$$

And,

$$\overline{\Delta U}^{N+1}_{L,M} = - \Delta t[(\delta^N_{L,M} + w_{L,M} \overline{U}^{N+1}_{L,M}) \frac{\overline{U}^{N+1}_{L+1,M} - \overline{U}^{N+1}_{L,M}}{\Delta x} + (\beta_{L,M}\overline{V}^{N+1}_{L,M}$$

$$+ \alpha_{L,M} \overline{U}^{N+1}_{L,M}) \frac{U^{N+1}_{L,M+1} - \overline{U}^{N+1}_{L,M}}{\Delta y} - \frac{1}{\rho_{L,M}}(w_{L,M} \frac{\overline{P}^{N+1}_{L+1,M} - \overline{P}^{N+1}_{L,M}}{\Delta x}$$

$$+ \alpha_{L,M} \frac{\overline{P}^{N+1}_{L,M+1} - \overline{P}^{N+1}_{L,M}}{\Delta y})] + QUT \tag{18}$$

For the corrector step we have,

$$U^{N+1}_{L,M} = \frac{1}{2} (U^N_{L,M} + \overline{U}^{N+1}_{L,M} + \overline{\Delta U}^{N+1}_{L,M}) \tag{19}$$

where L = subscript for the axial mesh point

M = subscript for the radial mesh point

N = time step

QUT = viscous terms computed in the predictor step

The first step predicts a new solution at time $t = (N+1)\Delta t$ at each mesh point (L, M) from the known solution at time $t = N\Delta t$, and uses backward differences to approximate the first derivatives and centered differences to approximate the second derivatives; that is, the viscous terms.

The second step corrects the predicted values with forward differences for the first derivatives. Note that the viscous terms are not computed in the corrector step for reasons given by Cline[4]. This method is stable if it satisfies the following condition given in the transformed plane,

$$\Delta t \leq \frac{A}{|U| w/\Delta\xi + |V|\beta/\Delta\eta + a\sqrt{w^2/\Delta\xi^2 + \beta^2/\Delta\eta^2}} \qquad (20)$$

where A is a coefficient close to 1, but not greater than 1 for inviscid flows, and much less than 1, ($A \simeq 0.4$ or less), for viscous flows and flows with shocks.

4.2 Boundary Grid Points

The boundary conditions for the present analysis are divided into 4 categories,

- centerline grid points
- wall grid points
- moving wall grid points
- exit grid points

4.2.1. Center Line Grid Points

The boundary condition at the centerline is that of symmetry, i.e. $v = 0$. However, since there is a singular term at the axis ($y = 0$), it is replaced by its limiting. form as y goes to zero, that is, $\lim_{y \to 0} \frac{v}{y} = \frac{\partial v}{\partial y}$.

4.2.2. Wall Grid Points

(1) Free slip walls

The boundary condition for free-slip walls is that the flow is tangent to the wall. This can be written as $v\,dx - u\,dy_w = 0$ (i.e. wall is a streamline)
The boundary conditions for no-slip walls are that the velocity components vanish at the wall.

The numerical treatment of the boundary conditions for free-slip walls is to prescribe one flow condition, and then calculate the remaining flow parameters by using one of the following schemes:

(1) Extrapolation (zeroth order or first order)

(2) Reflection principle

(3) One sided difference (first order or second order accurate

(4) Reference plane characteristic

The methods (1) - (3) can cause some problems in some cases, such as in subsonic inflow and outflow (Ref.[5].) where errors made in prescribing the boundaries propagate in all directions. The method (4) has been originally proposed by Moretti[5] and used by a number of fluid dynamicists, notably, Serra[6], and Cline[4]. Cline has developed the method for all the boundaries (entrance, exit, and walls). For this problem the methods (1), (2), and (4) have been used. Better solutions have been obtained with the method (4). Method(2) has failed to converge (see Figure 9). The reference plane characteristic method is derived in Appendix C, and it follows closely the work done by Cline[4]. Three characteristic relations that relate the interior flow to the wall boundary with the given boundary condition form a system of four coupled equations for u, v, ρ, and P. This can be written as,

$$\beta\, du - \alpha\, dv = (\beta \Psi_2 - \alpha \Psi_3)\, d\tau \tag{22}$$

$$d\,P - a^2 d\rho = \Psi_4\, d\tau \tag{23}$$

along the characteristic $d\eta = v^* d\tau$

and,

$$d\,P + \rho\, a\, \alpha\, du/\alpha^* + \rho\beta a\, dv/\alpha^* = (\Psi_4 + a^2 \Psi_1 + \rho\, \alpha\, a\, \Psi_2/\alpha^*$$

$$+ \rho\, \beta\, a\, \Psi_3/\alpha^*)\, d\tau \tag{24}$$

along the characteristic $d\eta = (V^* + \alpha^* a)\, d\tau$

The boundary condition is: $v dx - u\, d\, y_w = 0.$ (25)

The coefficients Ψ_1, Ψ_2, Ψ_3, and Ψ_4 are given in Appendix C. The prescribed boundary condition is the function representing the shape of the wall, i.e. $y_w(x)$. Equations (22) - (25) are written in finite difference form. The differentials du, dv, $d\rho$, and dP are replaced by differences along the characteristic curves. The coefficients are evaluated by the MacCormack scheme.

(2) No slip walls

The numerical treatment of the boundary conditions for no-slip walls is much simpler. The velocity components u and v vanish at the wall. The remaining variables P and ρ are calculated from the energy equation and the continuity equation, respectively.

4.2.3. Moving Wall Grid Points

The boundary condition at the moving wall is the velocity of the piston. The velocity is obtained from the equation of motion of the moving piston. This can be written

$$du_P - \frac{1}{m_P}\left[\frac{2\pi}{\beta^2}\int_0^1 (P_{LHS} - P_{RHS})\,\eta\,d\eta - D\right] dt = 0 \tag{26}$$

In this equation m_P is the piston mass, P_{LHS} is the pressure at the left hand side of the piston, P_{RHS} is the pressure at the right hand side of the piston, and D is the friction force. The characteristic expression that relates the interior points to the moving boundary can be written as,

$$dP - \rho\, a\, du = (\Psi_4 + a^2\Psi_1 - \rho\, a\, \Psi_2)\; d\tau \tag{27}$$

along the characteristic $d\xi = (u^* - aw)\; d\tau$

The prescribed flow parameters are $v = 0$, ρ is extrapolated from the adjacent grid. The pressure P, and the velocity u are obtained from the solution of Equations (26) and (27). Equations (26) and (27) are written in finite difference form. If the pressure is obtained by the method (2), then the solution diverges. (see Figure 9.) The finite difference form for Equation (26) can be written as,

$$U^{N+1}(L,M) = U^{N}(L,M) + \frac{\Delta t}{M_P}\left[(\overline{P}_{LHS} - \overline{P}_{RHS})A_P - D\right] \qquad (28)$$

the overbar denotes the average values.

4.2.4. Exit Grid Points

The boundary condition at the exit plane can be supersonic outflow or a subsonic outflow. For supersonic outflow, the normal velocity at the exit is greater than the speed of sound. Then no signals from outside can propagate into the discharge chamber. In this case the flow variables are evaluated from the interior values by a simple linear extrapolation, and the error generated from the extrapolation is not expected to propagate back and affect the upstream results. For subsonic outflow a special treatment is needed because any arbitrariness in the choice of the boundary values sends arbitrary signals inside the computational region. It can reflect pressure disturbances which are damped only by dissipative-effects. A non-reflecting outflow boundary condition has been used for this purpose by Serra[5], Cline[4], C. K. Chu and Aron Sereny[7] and Rudy[8]. It seems advantageous to use this form in the present analysis. The prescribed boundary condition is the ambient pressure, $P = P_A$. In this case the problem is well posed, i.e., the solution is unique and continuous at the boundary. The characteristic relations that relate the interior points to the exit boundary points can be written as,

$$dP - a^2 d\rho = \dot{\Psi}_4 \, d\tau \tag{29}$$

$$dv = \Psi_3 \, d\tau$$

along the characteristic $d\xi = u^* d\tau$,

$$d P + \rho a \, du = (\dot{\Psi}_4 + a^2 \dot{\Psi}_{.} + \rho a \Psi_2) \, d\tau \tag{30}$$

along the characteristic $d \xi = (u^* + aw) \, d\tau$

Equations (29) - (30) are solved by the same methods used in Part C .

5. Results

The function y_w describing the shape of the wall can now be approximated by the following equation,

$$y_w(x) = (R_i + R_e)/2 - (R_i - R_e) \tanh \sigma (x - x_0) \tag{31}$$

where R_i and R_e are, respectively, the radius at the entrance and at the exit of the discharge chamber, σ is a parameter that controls the curvature of the wall, and x_0 is the point about which the grid is concentrated. The mapping functions given by the Equations (6) - (8) become,

$$\xi(x,t) = \frac{\int_{x_\rho(t)}^{x} \sqrt{1 + \frac{(R_i-R_e)^2\sigma^2}{\cos h^4 \sigma(x-x_0)}}\, dx}{\int_{x_\rho(t)}^{xE} \sqrt{1 + \frac{(R_i-R_e)^2\sigma^2}{\cos h^4 \sigma(x-x_0)}}\, dx} \tag{32}$$

$$\eta(y,x) = \frac{y}{(R_i + R_e)/2 - (R_i-R_e)\ \tan h\ \sigma(x-x_0)} \tag{33}$$

$$\tau = t \tag{34}$$

The coefficients β, w, α, and δ are also obtained by substituting the value of $y_w(x)$ given in Equation (31). They are computed in the mapping subroutine.

The inversion of the function ξ is done by interpolation of tabular values. The grid generated in the discharge chamber for four different wall shapes is, respectively, shown in Figures 4a, 4b, 4c, and 4d. In each case, the physical space grid consists of 50 by 16 grid points. The inlet and outlet radii are, respectively, set equal to 6.35 and 1.27 CM . The coefficient, σ, that controls the wall curvature is, respectively, set equal to 1.0, 6.0, 10.0, and 15.0. Therefore,a grid can be generated from Equations (32) and (33) to handle the rapid change along the steep wall of the discharge chamber.

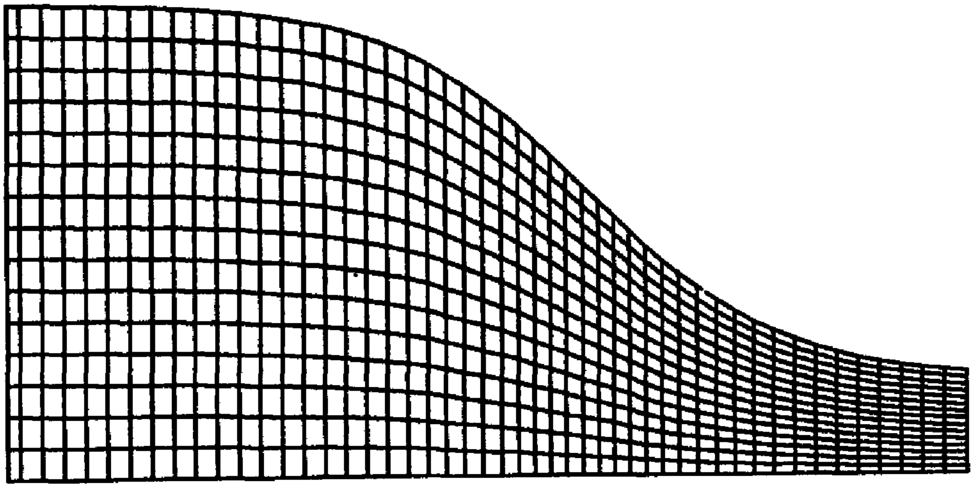


Figure 4a. Grid generated in the discharge chamber, ($\sigma = 1.0$)

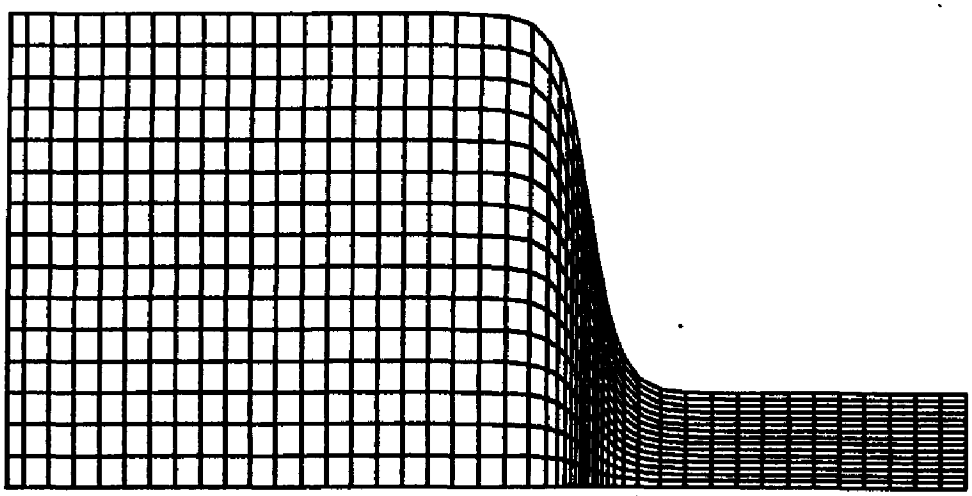


Figure 4b. Grid generated in the discharge chamber, ($\sigma = 6.0$)

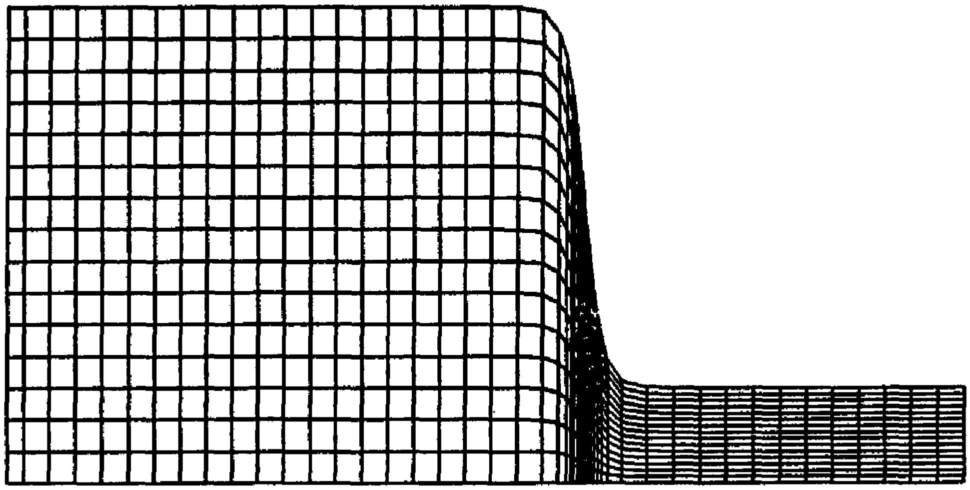


Figure 4c. Grid generated in the discharge chamber, (σ = 10.0)

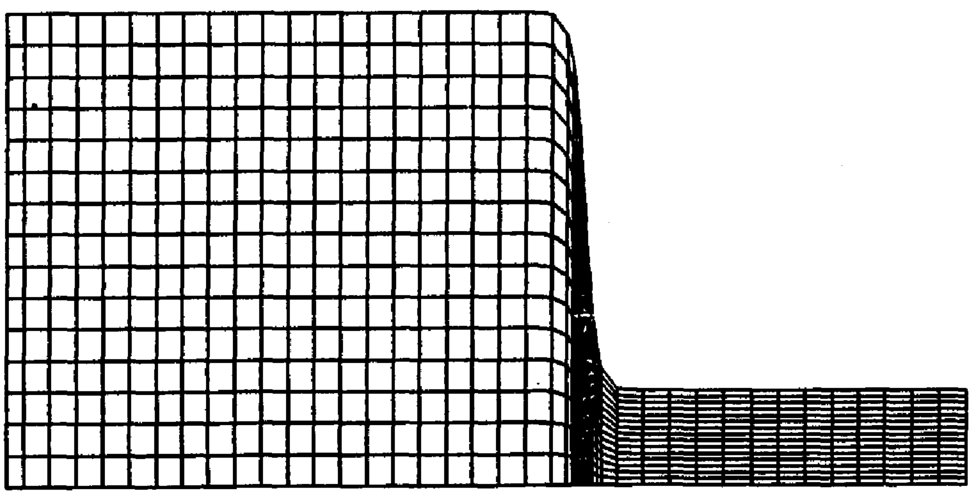


Figure 4d. Grid generated in the discharge chamber, (σ = 15.0)

Following the same technique described above, one can generate a grid in the discharge chamber used in Pakarat[9]'s experiment, Figure 5. For this case the shape of the wall is,

$$y_w(x) = \begin{cases} \dfrac{R_1 + R_2}{2} - \dfrac{R_1 - R_2}{2} \tanh \sigma_1 (x - x_1) & \text{if } x < \dfrac{x_1 + x_2}{2} \\ \dfrac{R_2 + R_3}{2} - \dfrac{R_2 - R_3}{2} \tanh \sigma_2 (x - x_2) & \text{if } x \geq \dfrac{x_1 + x_2}{2} \end{cases} \tag{35}$$

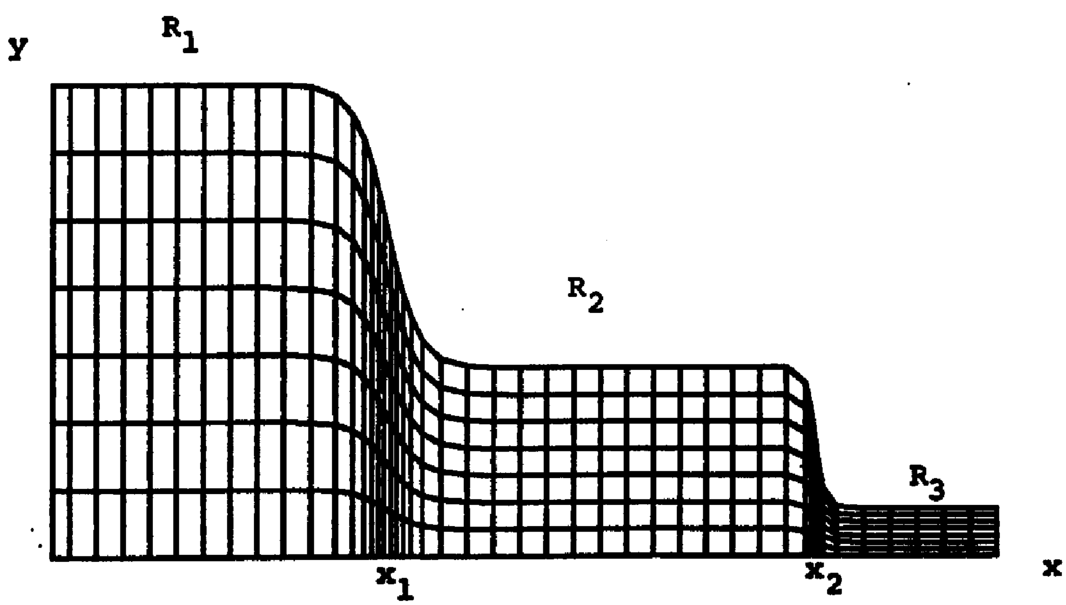


Figure 5. Grid generated in the discharge chamber used in Pakarat's experiment.

A grid generation can also be extended to the discharge chamber whose wall contains n different curvatures, Figure 6. In this case the shape of the wall is given by Equation (36).

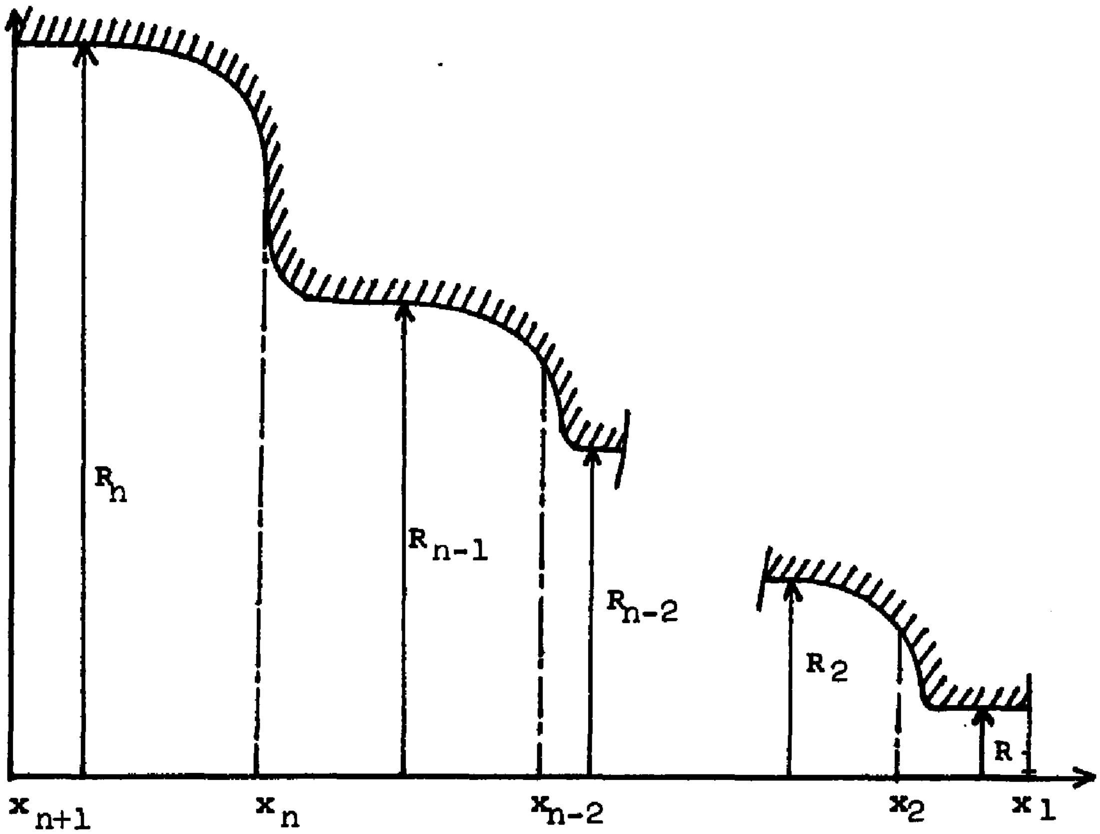


Figure 6. Discharge chamber in the case where the wall has n different curvatures.

$$Y_w(x) = \sum_{j=2}^{n} \chi_K(x) \left(\frac{R_j + R_{j-1}}{2} - \frac{R_j - R_{j-1}}{2}\right) \tanh \sigma_j (x - x_j) \quad (36)$$

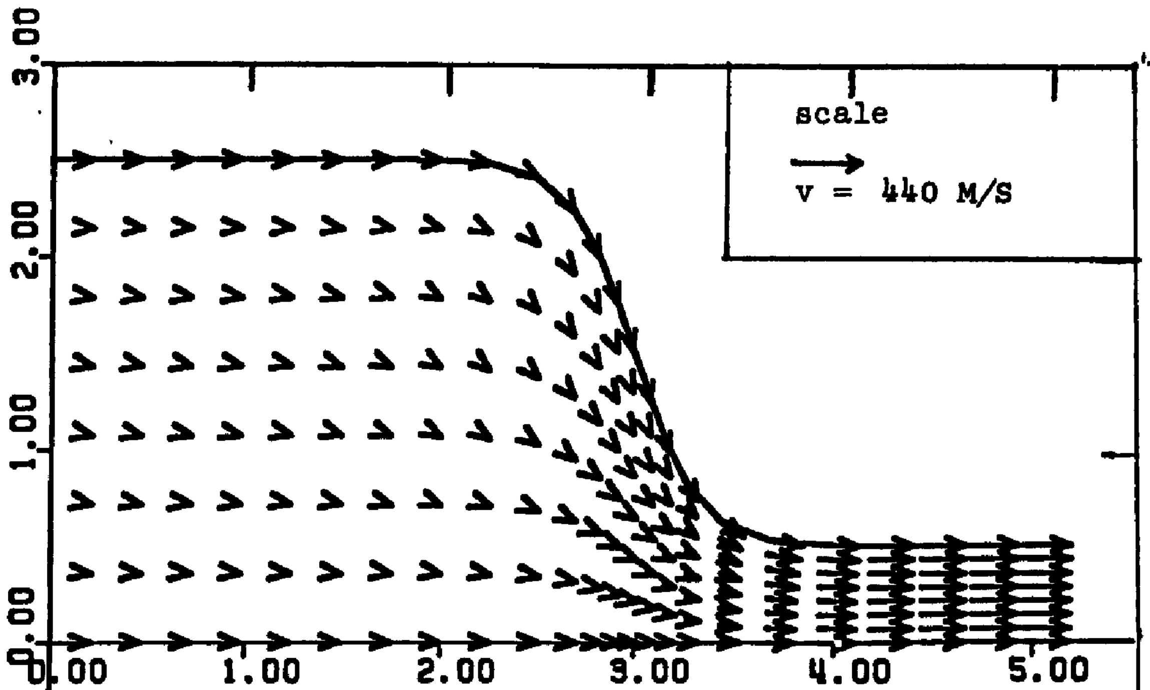
where $K = [\frac{x_{j+1} + x_j}{2} , \frac{x_j + x_{j-1}}{2}]$

Figures 7-9 show the velocity vector plots of the flow in the discharge chamber. The physical space grid consists of 25 by 8 grid points. The discharge pressure, P_C, is set equal to 986.KPA. The coefficient, σ, that controls the wall curvature is set equal to 6.0. The inlet and outlet radii are, respectively, set equal to 6.35 and 1.27 CM, X_0 = 7.62 CM, and finally the maximum number of time steps, N MAX, is set equal to 15000. It can be seen in Figures 7-8 that the transient flow occurring during the discharge phenomenon is accurately simulated. The velocity vector plots are shown every 800 time steps. Figures 7a and 7b show that the flow is sonic at the exit if the pressure in the discharge chamber is above 200.KPA. Then as the pressure gets below 170 KPA, the flow is subsonic (see Figure 7C).

Figure 7d demonstrates a case of an inflow. This phenomenon can occur if the ambient pressure is higher than the pressure in the discharge chamber. It is seen from Figure 13, that the pressure varies only slightly with position. The same results have been observed by Pakarat[9] in his experimental study. Comparison of the pressure distributions calculated by the present technique with the one dimensional solution are shown in Figure 10. This recent work shows improvements in the mathematical model of the discharge chamber flow. The one-dimensional solution used earlier (Ref. [2]) predicts a discharge time of 5 ms. The present solution indicates a discharge time of 13 ms. However it requires a computational time of 11 min and 29 s.

This numerical model has also been applied to the discharge chamber used by Pakarat[9] in his experimental study. The experimental time for the discharge chamber is about 160 ms. In this case the computational time is expected to be much larger because the grid surface generation has produced more mesh points and smaller grid sizes. The model simulates the discharge cycle only for a short time. The results are shown in Figures 14 through 17. These results are in better correlation with the experimental data.

In conclusion the Navier-Stokes equations can be solved numerically as a means of predicting the flow fields generated during the discharge. However it is too expensive to be accepted as an engineering solution.



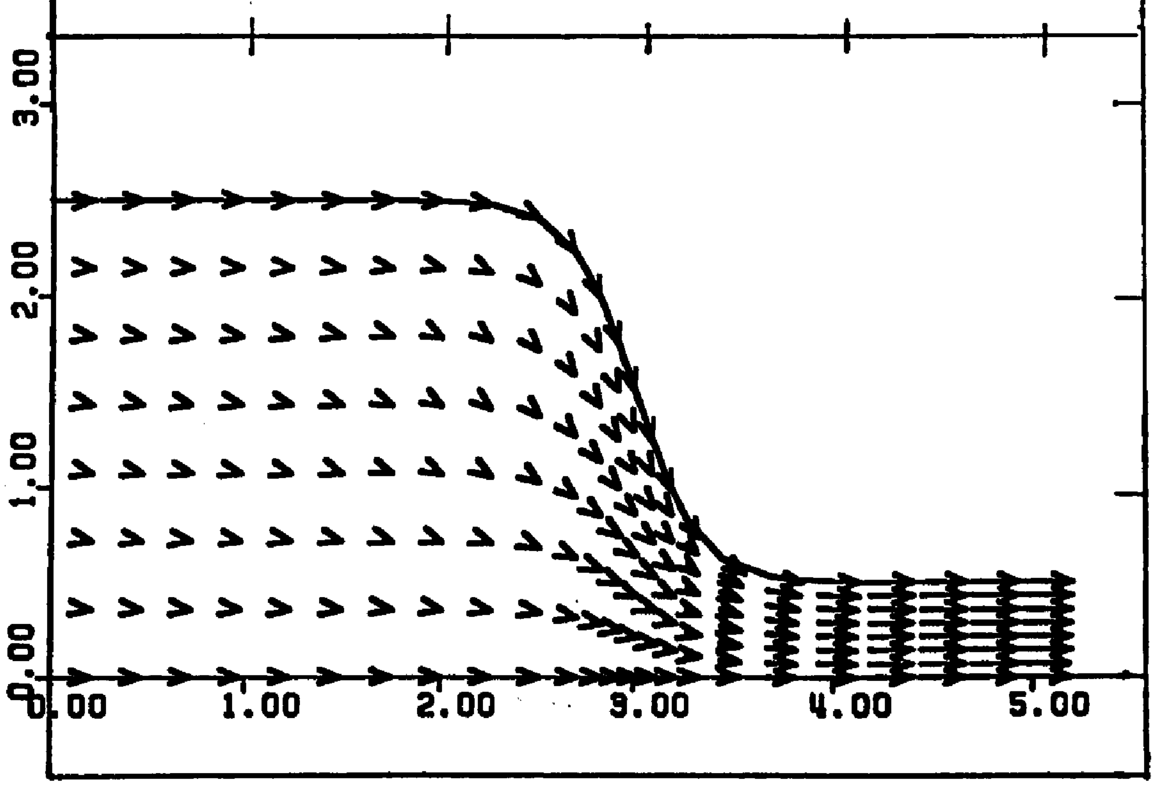


Figure 7. Velocity vector plots of the flow in the discharge chamber.

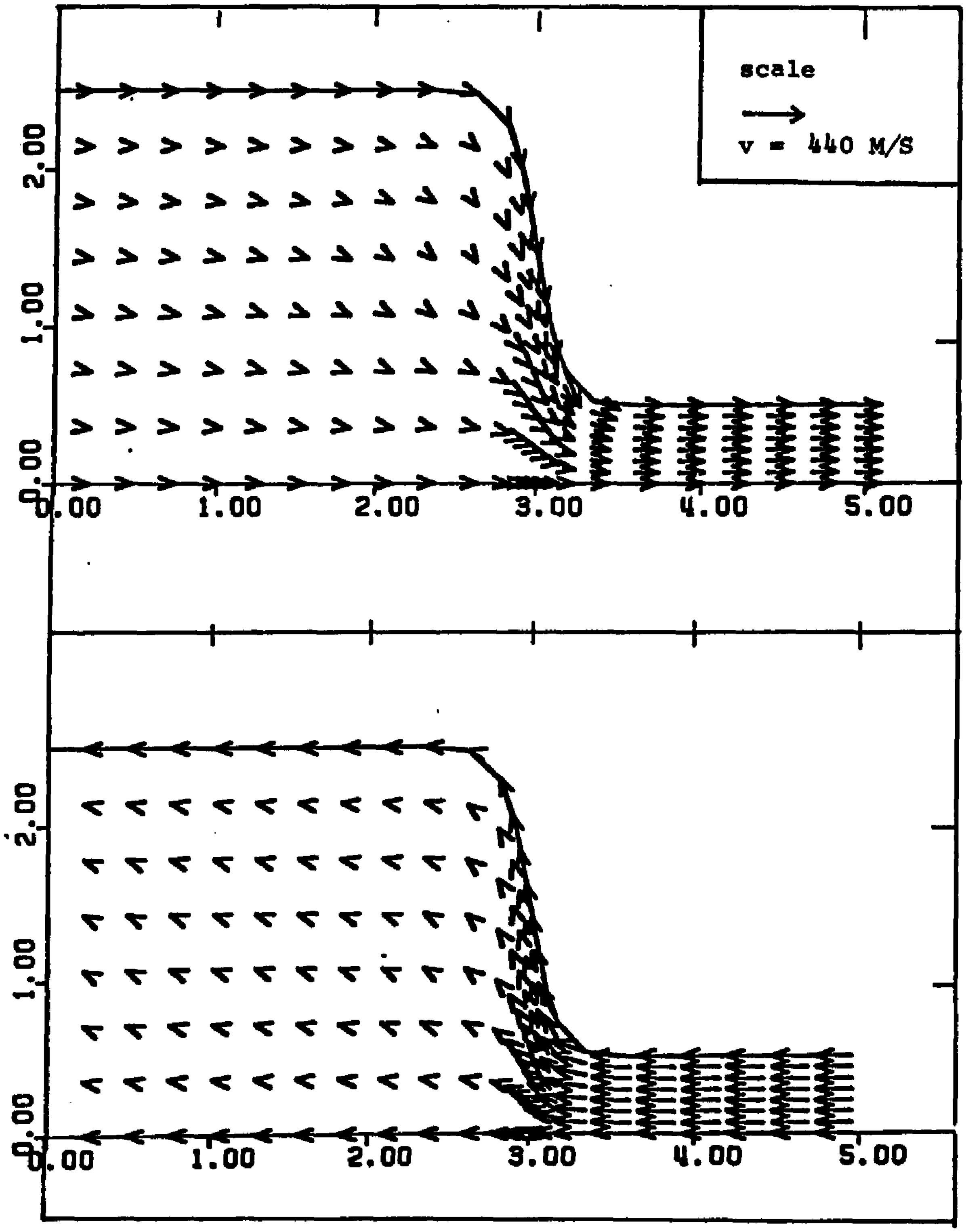


Figure 7 cont.

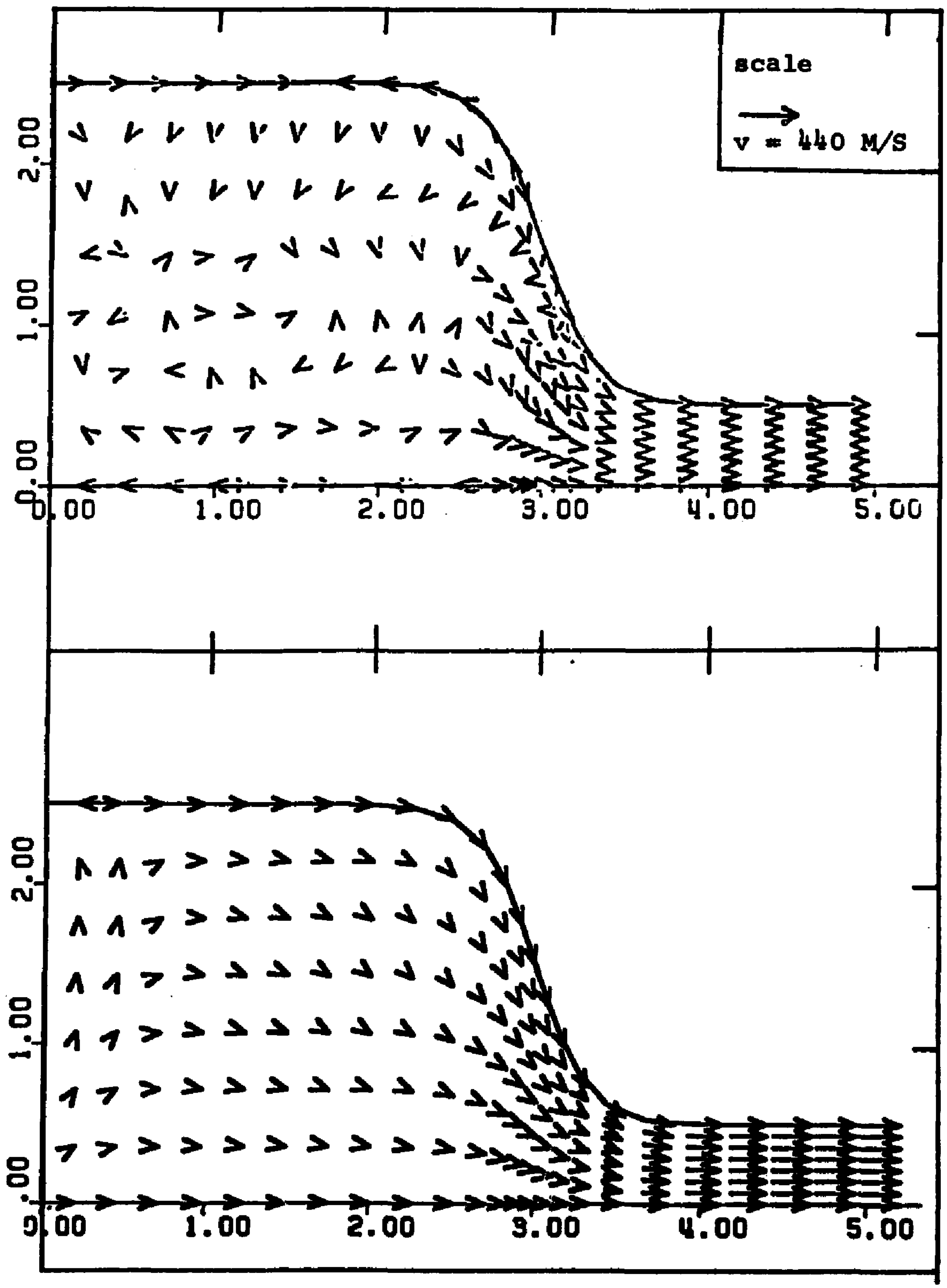


Figure 8. Velocity vector plots of the flow when smoothing devices are used.

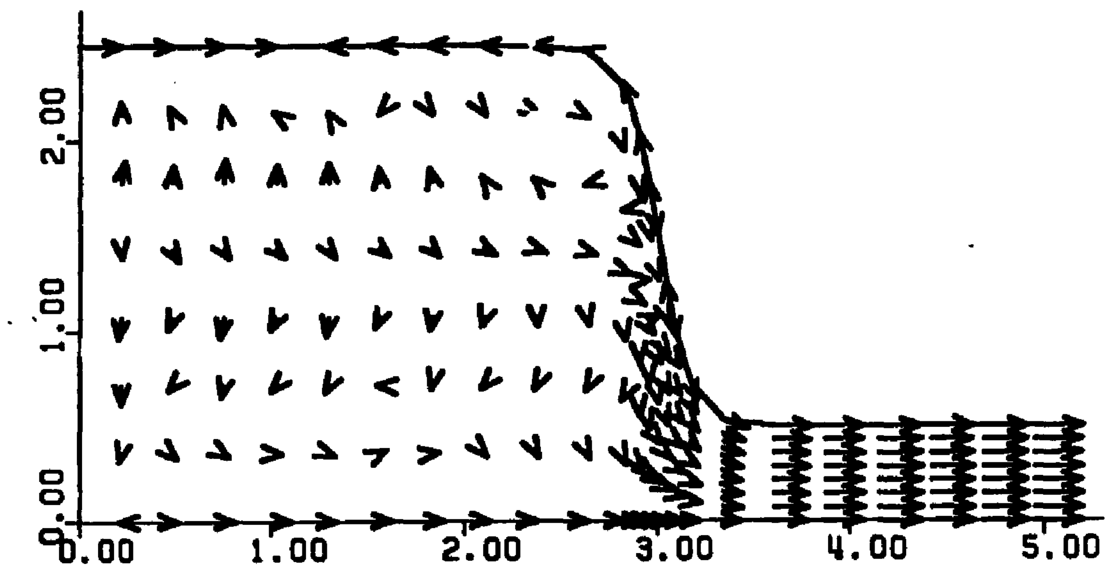


Figure 9. Velocity vector plots of the flow when the method of characteristics is not used at the boundaries.

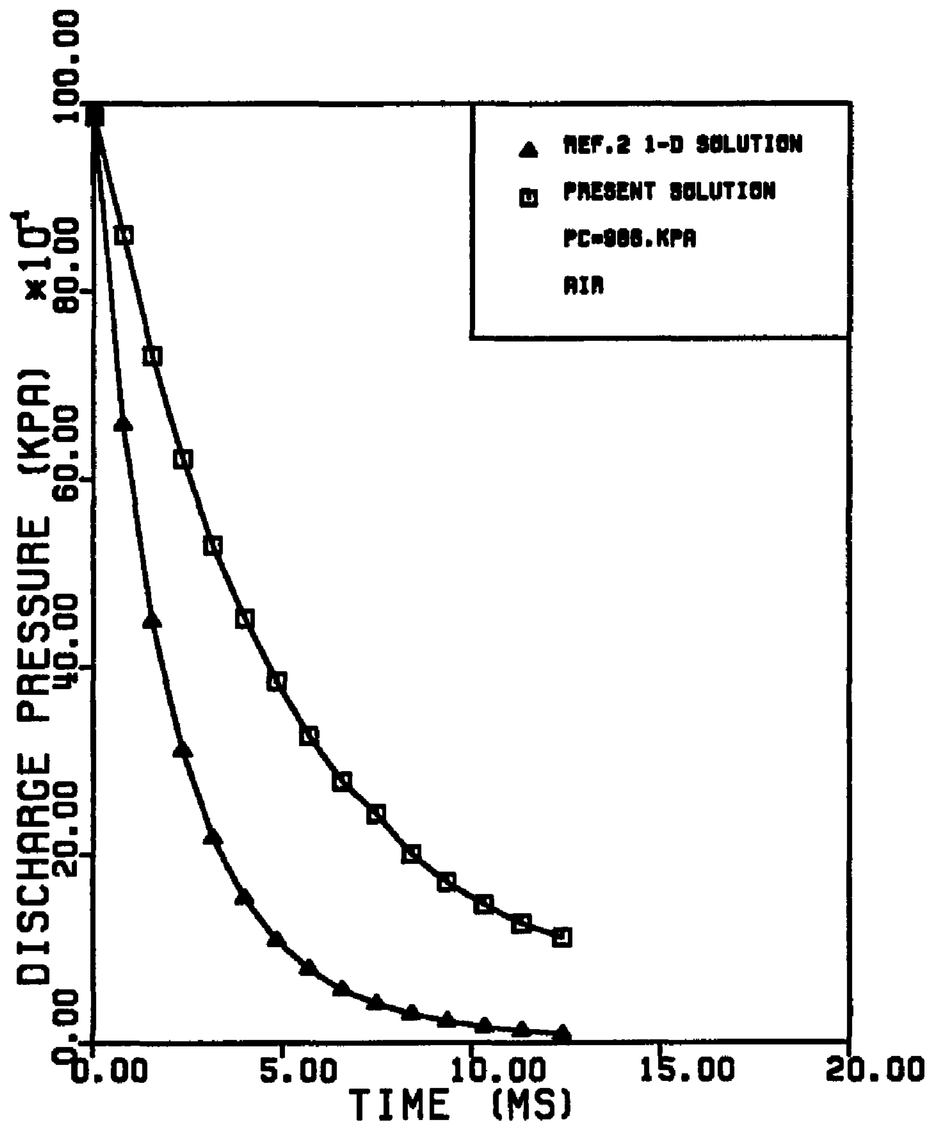


FIGURE 10.DISCHARGE PRESSURE VS TIME

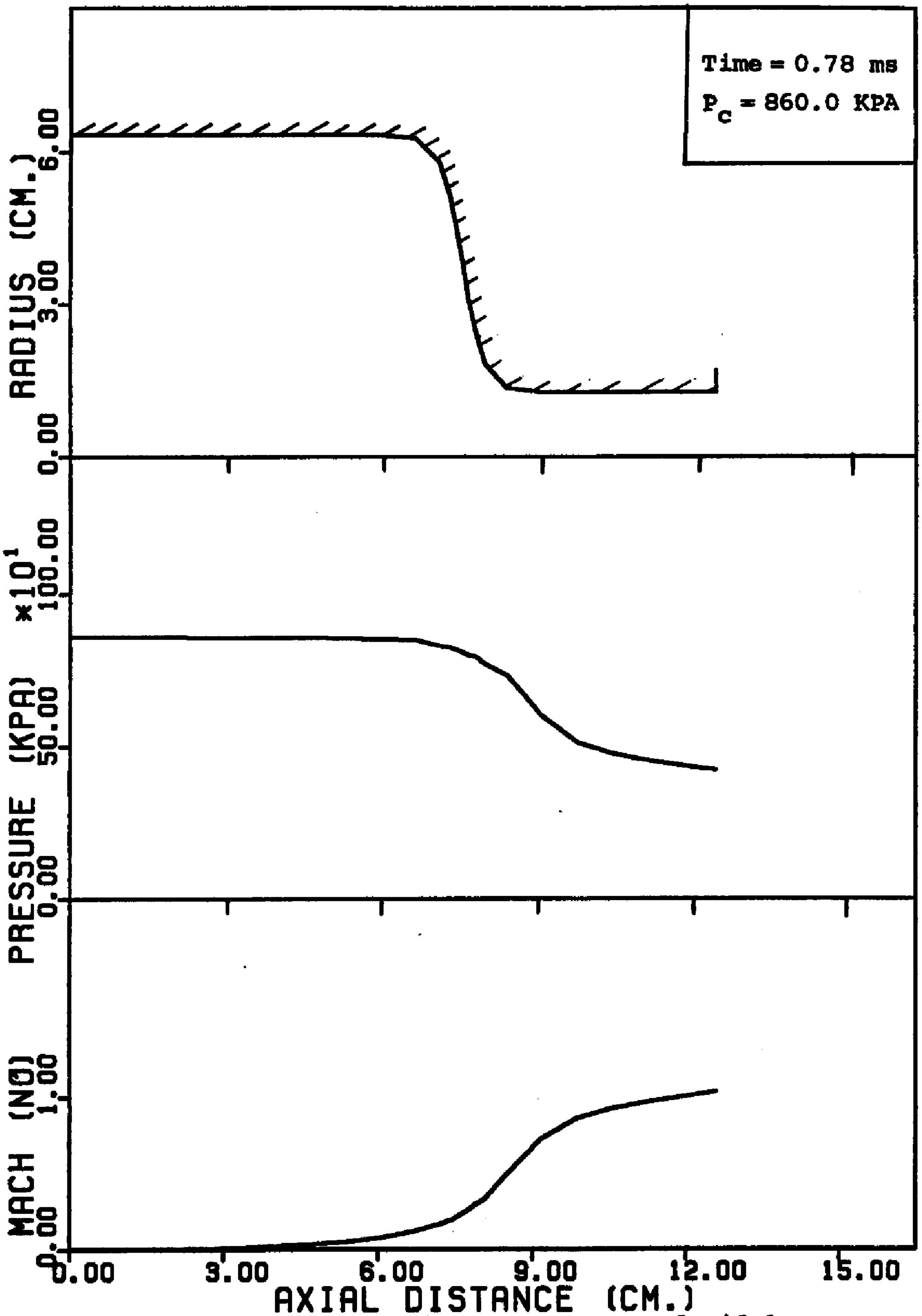


Figure 11. Discharge chamber geometry, and midplane pressure and Mach number for inviscid flow.

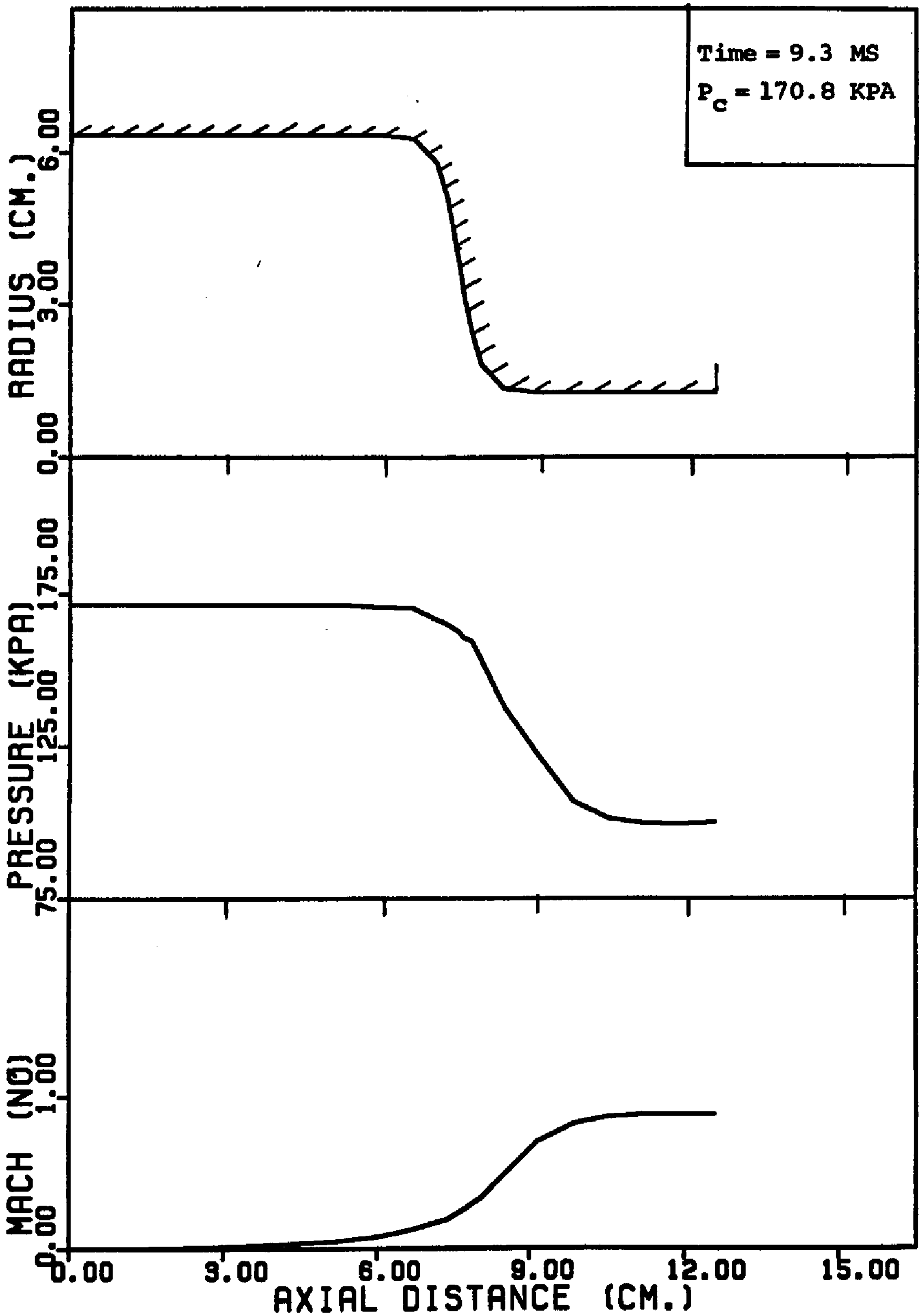


Figure 12. Discharge chamber geometry, and midplane pressure and Mach number for inviscid flow.

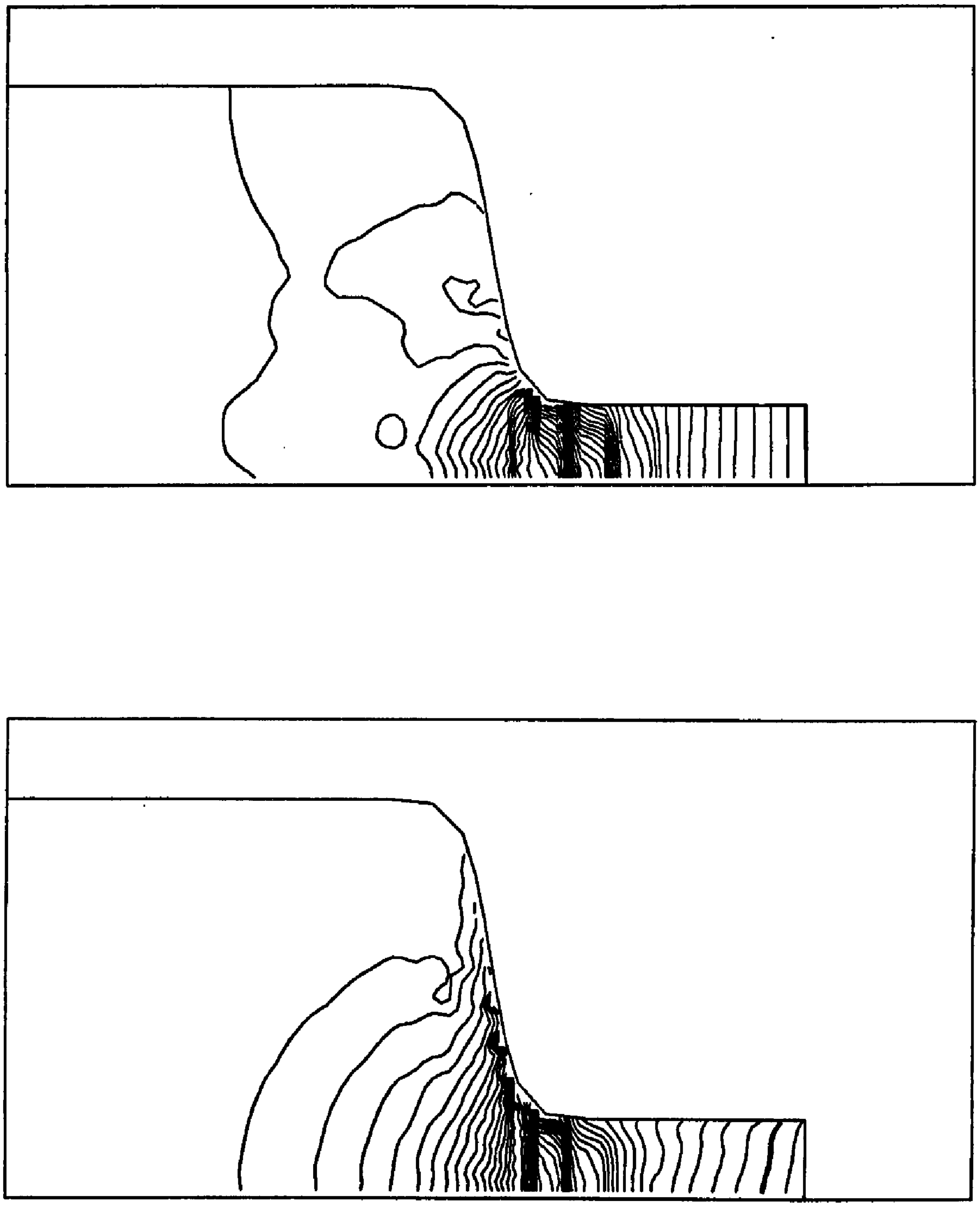


Figure 13. Pressure (top) and Mach number contours for inviscid flow.

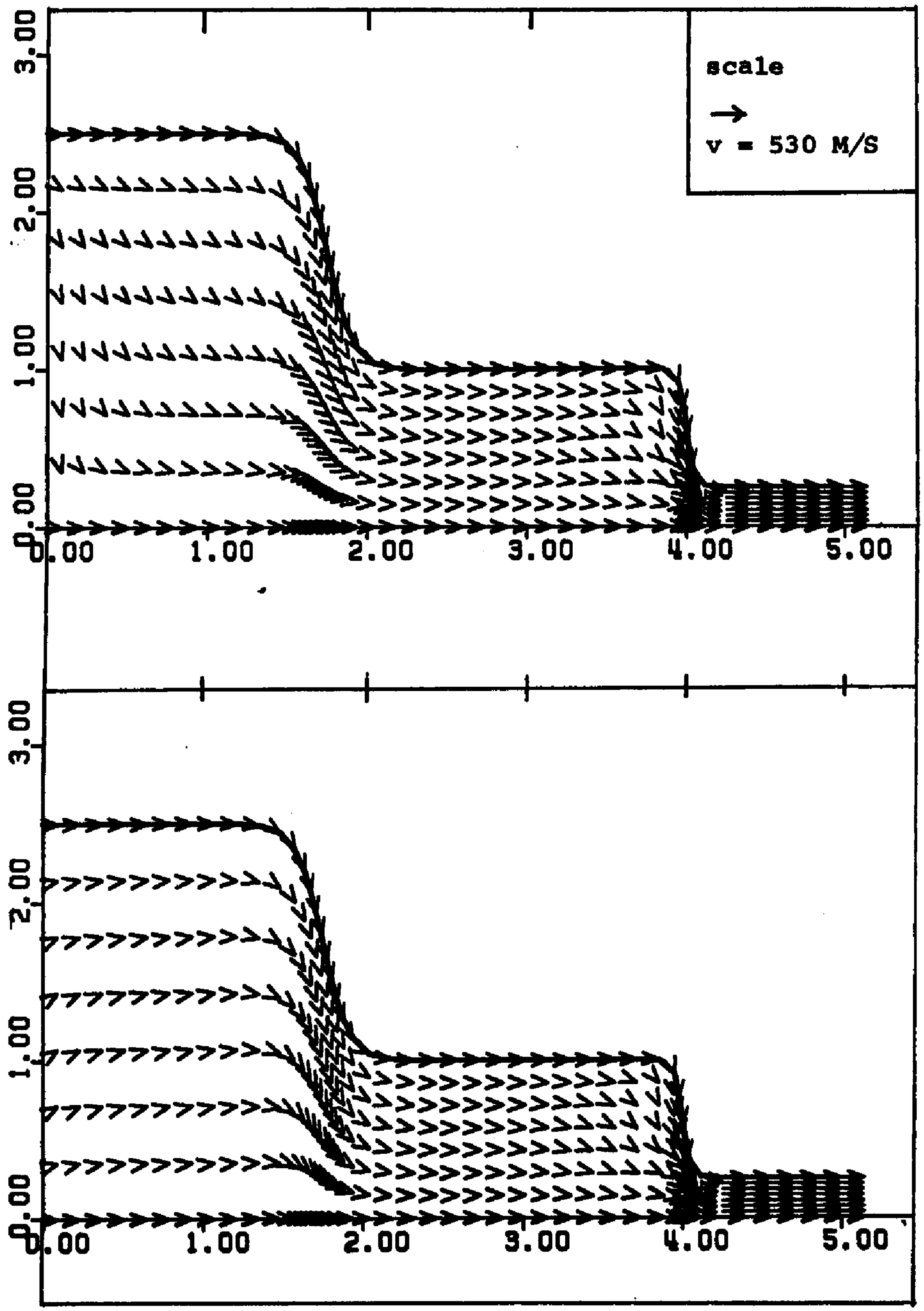


Figure 14. Velocity vector plots of the flow in the discharge chamber used in Pakarat s experiment

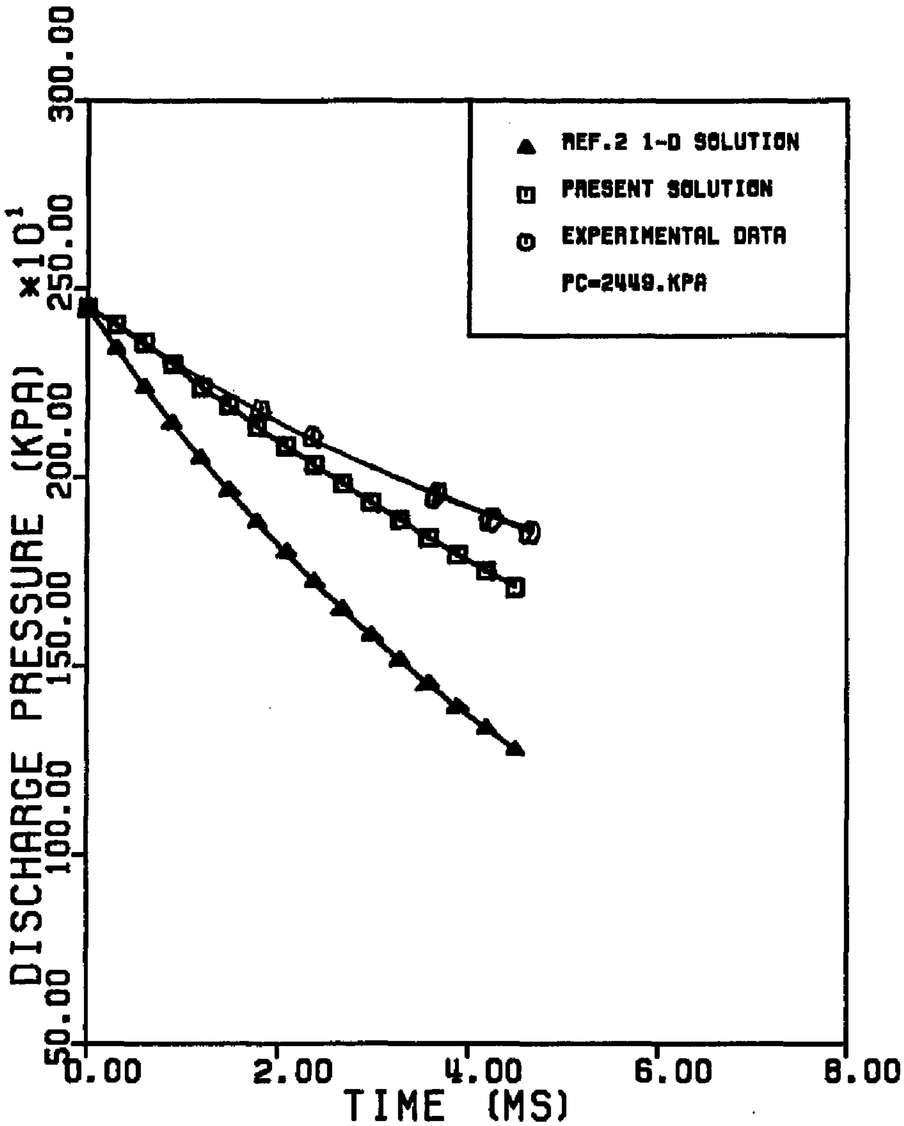


FIGURE 15. DISCHARGE PRESSURE VS TIME

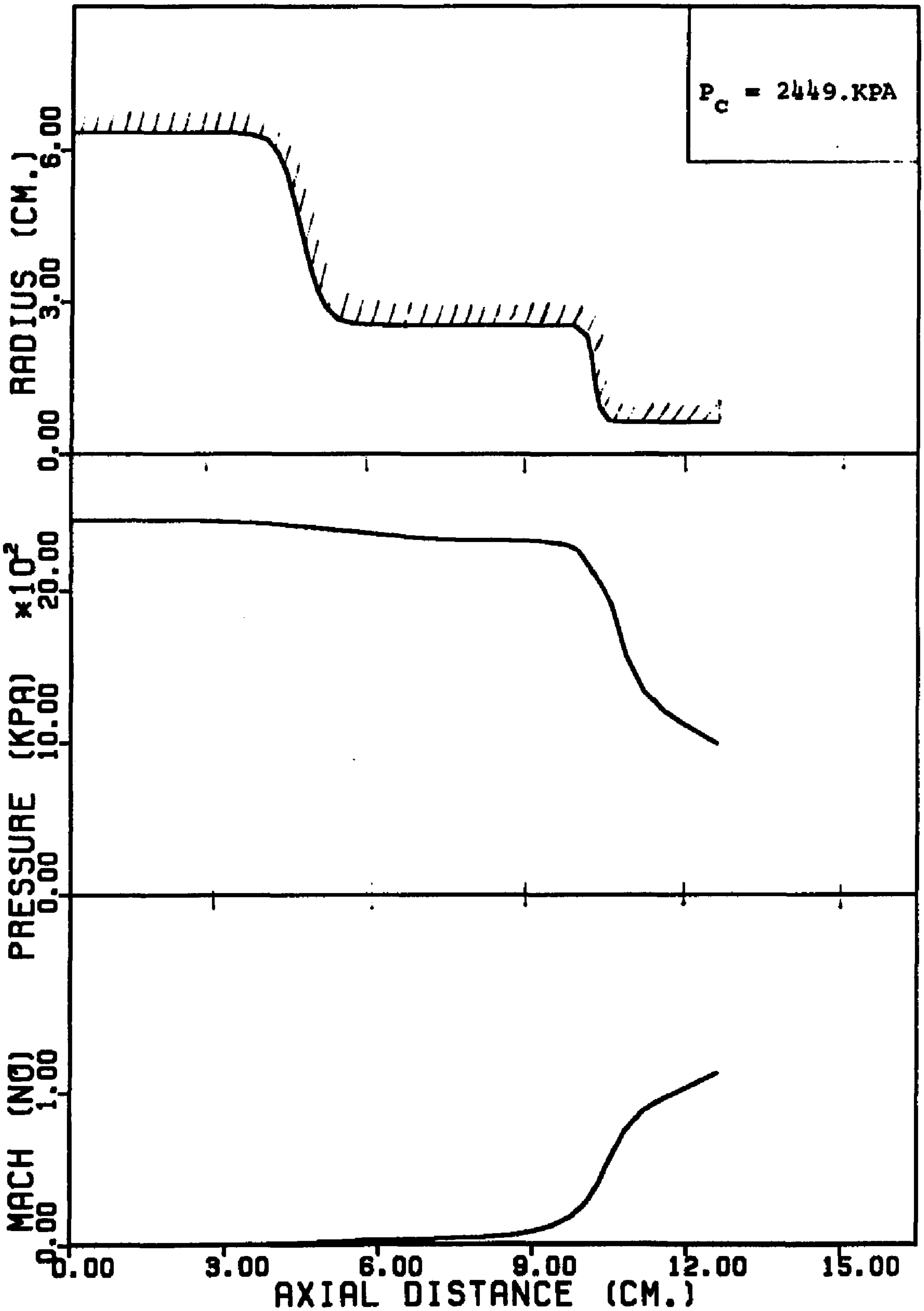


Figure 16. Discharge chamber geometry, and midplane pressure and Mach number for inviscid flow.

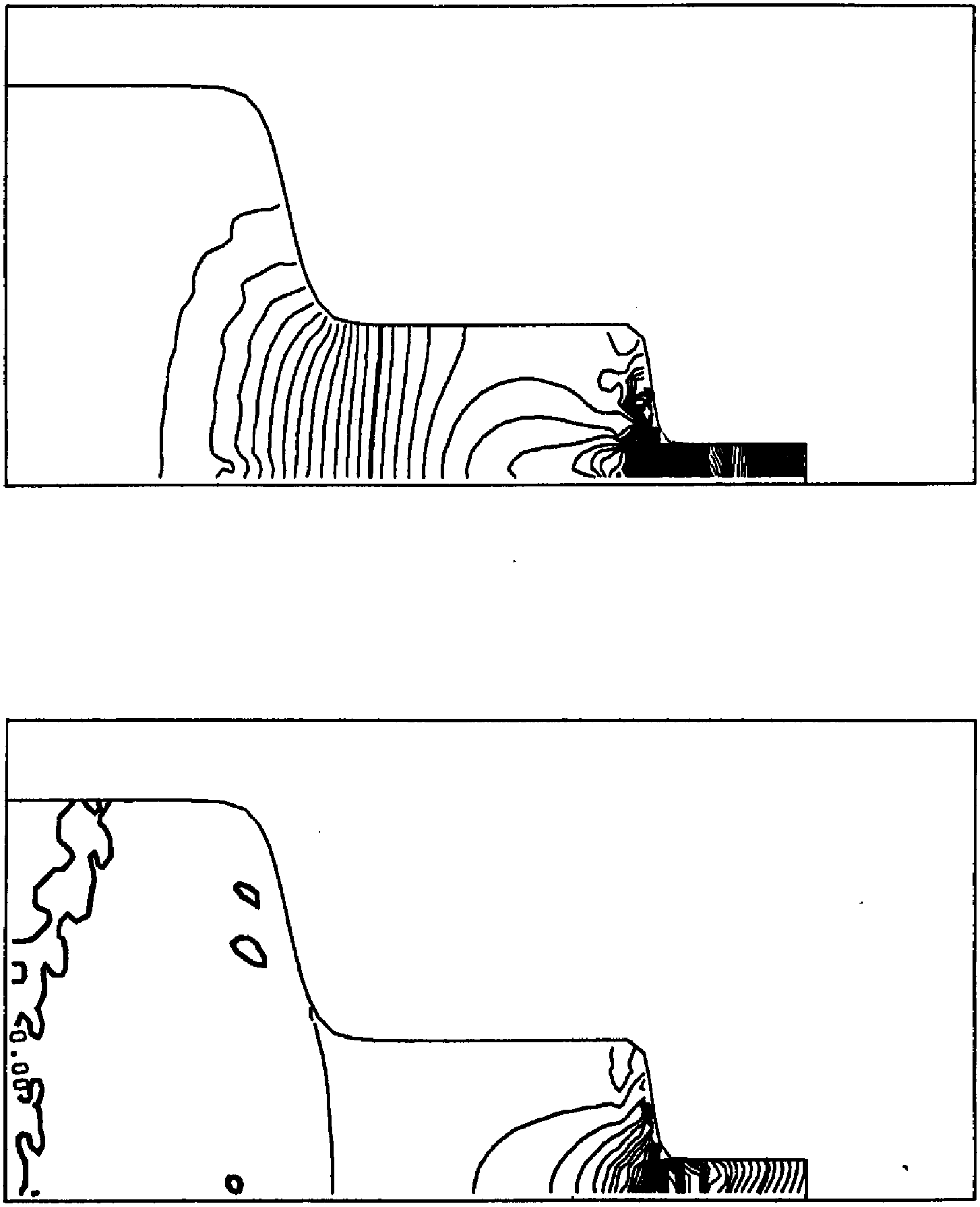


Figure 17. Pressure (top) and Mach number contours for inviscid flow.

B. BARREL CHAMBER FLOW

1. Governing Equations

The mathematical model simulates the gas flow through a chamber confined on one side by a moving boundary. An application of this model is in the barrel chamber of a gun where the moving boundary is the projectile. The model assumes a one dimensional tube flow, because the axial component of flow there is one of principal interest. Therefore, the flow properties depend only on the axial coordinate x and the time t. The model takes into account the effect of gas friction. However it neglects the effect of turbulence. The governing equations for this flow can be written as,

$$\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + \rho \frac{\partial u}{\partial x} = \Lambda \tag{37}$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{1}{\rho} \frac{\partial p}{\partial x} = F \tag{38}$$

$$\frac{\partial P}{\partial t} + u \frac{\partial P}{\partial x} - a^2 \left(\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} \right) = \Upsilon \tag{39}$$

$$\frac{P}{\rho} = RT , \quad a^2 = \gamma P/\rho \tag{40}$$

In the above equations the dependent variables are the cross-sectional-averages of density ρ, axial velocity u, the general term, Λ, contains the effect of mass addition and area change for the quasi-one dimensional flow. The general term, F, contains the effect of friction. The general term, Ψ, contains the effect of heat transfer and viscous dissipation.

2. Numerical Method

The mathematical model discussed in the previous section consists of a system of quasi-linear non-homogeneous partial differential equations of the first order. A quasi-linear partial differential equation of the first order is one that may be non-linear in the dependent variables but is linear in the first order partial derivatives. The system of governing Equations (37) - (39) is hyperbolic. There are several numerical methods which are applicable to this mathematical model. One of the most accurate numerical techniques for solving hyperbolic partial differential equations is the method of characteristics. In the present investigation the method of characteristics is used. A complete numerical algorithm can be developed. The development presented herein closely follows the technique presented by Zucrow and Hoffman[10]. A characteristic is defined mathematically as a curve along which the governing partial differential equation reduce to an interior operator; that is, a total differential equation known as the compatibility equation.

The dependent variables may not be specified arbitrarily on a characteristic curve, but they must be compatible with the interior operator. From a physical point of view, a characteristic is defined as the path of propagation of a physical disturbance.

2.1 Characteristic and Compatibility Equations

The equations specifying the characteristic curves and the corresponding compatibility equations can now be written. There are three families of characteristic curves, two families of Mach lines, and a family of pathlines.

Along the pathline

$$dx = u\,d\,t \tag{41}$$

$$dP - a^2 d\,\rho = \Psi\,dt \tag{42}$$

Along the Mach lines

a) $$dx = (u + a)\ dt \tag{43}$$

$$dP + \rho\,a\,du = (\Psi + a^2 \Lambda + \rho\,a\,F) \tag{44}$$

b) $$dx = (u - a)\ dt \tag{45}$$

$$dP - \rho\,a\,du = (\Psi + a^2 \Lambda - \rho\,a\,F) \tag{46}$$

These ordinary differential equations are derived in Appendix A.

2.2 Numerical Integration Scheme

The characteristic and compatibility Equations (41) - (45) are ordinary differential equations. These ordinary differential equations may be integrated by simple numerical integration techniques which are applicable to ordinary differential equations, such as Runge-Kutta, Taylor, Euler etc. The second order modified Euler predictor-corrector numerical integrator can be used for the integration of the compatibility equations. The basic features of this method are summarized below.

Consider the ordinary differential equation,

$$dy = f(x,y)\,dx \quad \text{with initial conditions } (x_0, y_0) \tag{47}$$

The integration of Equation (46) can be obtained in two steps.

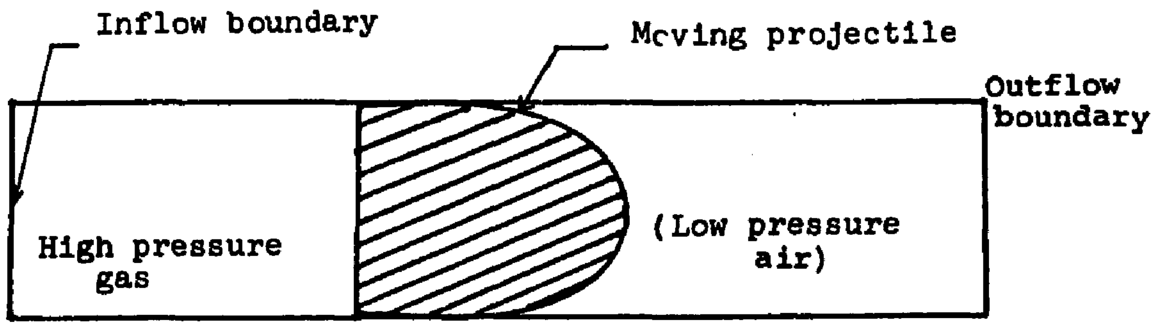
For the first step we have,

$$Y_{i+1}^{P} = Y_i + f(x_i, y_i)\ \Delta x \tag{48}$$

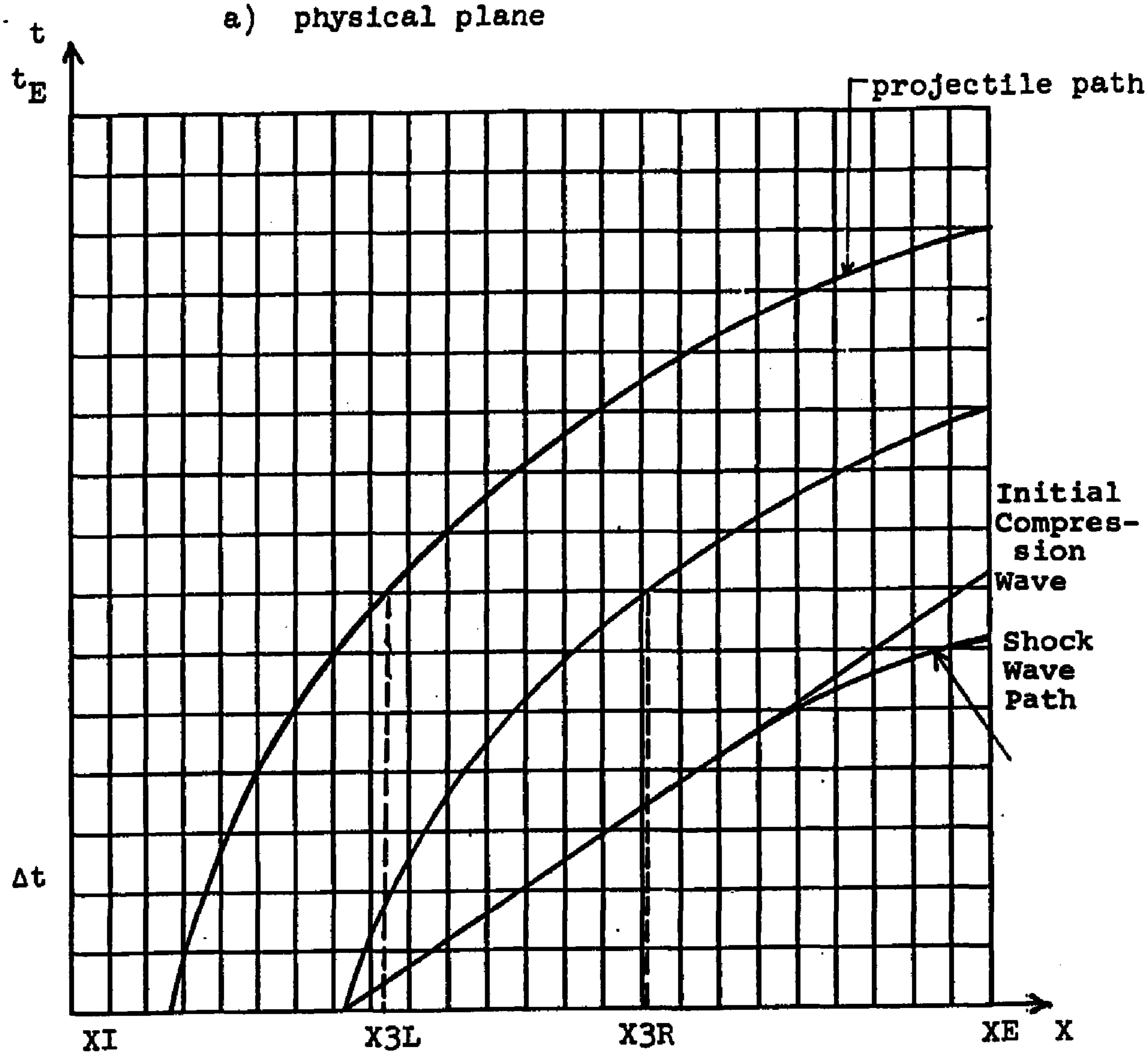
For the second step (with iterations) we have

$$Y_{i+1}^{C} = Y_i + f\left(x_i + \frac{\Delta x}{2},\ \frac{y_i + y_{i+1}^{C}}{2}\right)\ \Delta x \tag{49}$$

Two different types of overall marching algorithm may be constructed in the numerical method of characteristics, the direct marching method in which the characteristics are projected forward (in time) to determine the location of the solution point, and the inverse marching method in which the solution points are prespecified in some manner, usually on a rectangular grid, and the characteristics are projected rearward (in time) to determine the initial-data points. The latter method is employed in the numerical integration schemes presented in this work. Figure 18 shows a typical computational grid. The solution is obtained on lines of constant t. The physical



a) physical plane



b) Computational flow plane.

Figure 18 . Physical and computational flow planes.

plane is subdivided into a number of fixed grid points. The time step Δt between successive solutions lines is choosen to satisfy the CFL stability criterion developed by Courant, Friedrichs, and Lewy. The CFL stability criterion states that the differential domain of dependence of the solution point must be contained within the finite difference domain of dependence of the solution point. This numerical integration scheme is, therefore, conditionally stable. The physical interpretation of this stability criterion is shown in Figure 19. In that figure the solution point (4) is influenced by the solution of each of the grid points at previous time steps contained within the two diagonals (4A) and (4C). Thus the region (4AC4) is the domain of dependence of point (4) in the finite difference domain.

If (4-1) and (4-2) are the backward characteristics of slope $(dx/dt)_{\pm} = u \pm a$ passing through the solution point (4), the CFL stability requires that line segment(1-2) fall completely within line segment (AC), as shown in Figure 19. However, from the theory of characteristics it is known that point (4) can receive signals only from the region 4-1-2-4 which is the domain of dependence of the differential equation, that is, the physical domain of dependence of the point (4). The computational grid shown in Figure 18 is divided into 5 sets of mesh points.

(1) interior grid points

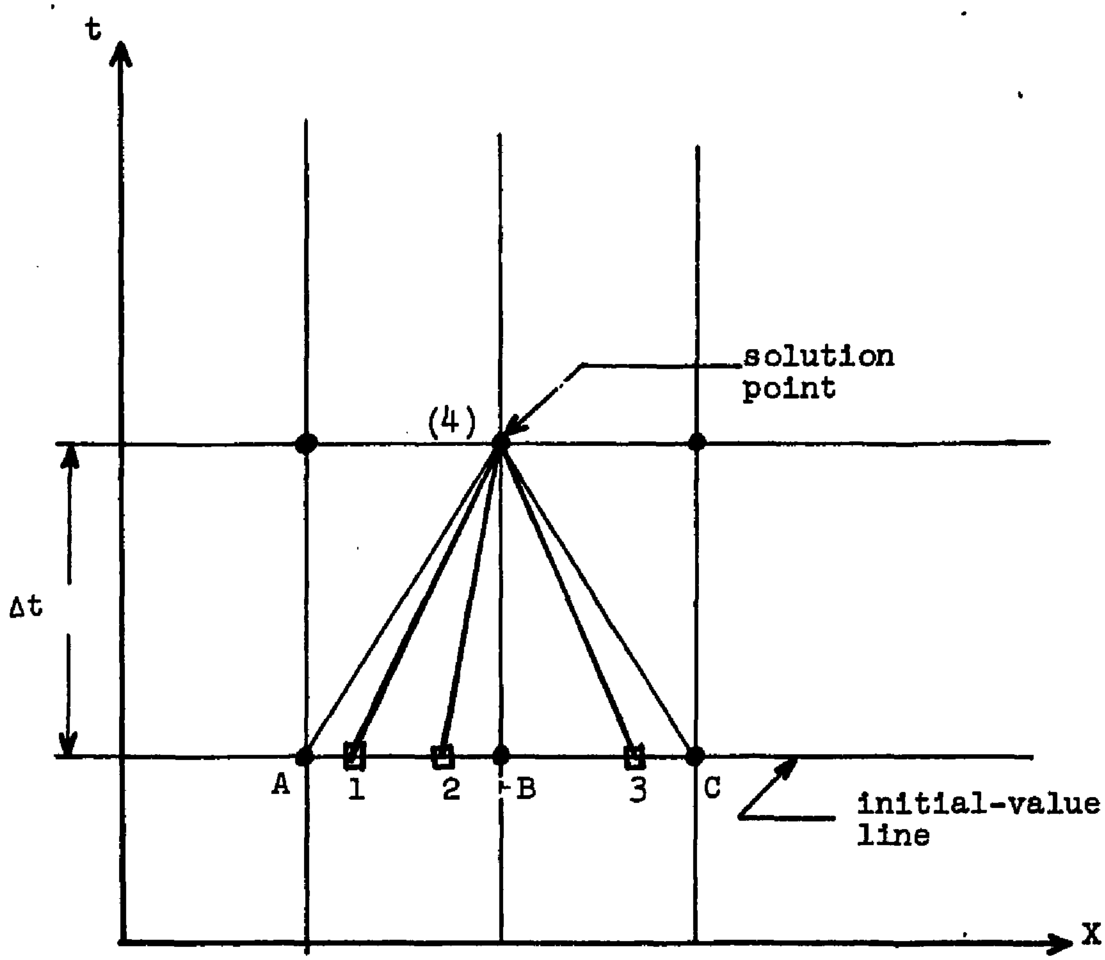


Figure 19. Physical interpretation of the CFL stability criterion.

(2) inflow grid point

(3) outflow grid point

(4) projectile grid point

(5) shock wave grid point

A numerical algorithm for each type of mesh point can now be presented.

2.2.1 Interior Grid Point

The finite difference grid for the interior point is shown in Figure 19. The flow properties at the solution point (4) are determined by numerically integrating the compatibility equations, Equations (40) - (46), along the pathline and the Mach lines, respectively.

Equations (40) - (45) are written in finite difference form. The differentials du, dρ and dP are replaced by differences along the characteristic curves. These can be written as follows:

$$(P_4 - P_3) - (a^2)_{34}(P_4 - P_3) = \bar{\Psi}_{34}\ \Delta t \tag{50}$$

$$(P_4 - P_1) + \overline{(\rho a^2)}_{14}(u_4 - u_1) = \overline{(\rho a F + a^2\Lambda + \Psi)}_{14}\ \Delta t \tag{51}$$

$$(P_4 - P_2) - \overline{(\rho a^2)}_{24}(u_4 - u_2) = \overline{(-\rho a F + a^2\Lambda + \Psi)}_{24}\ \Delta t \tag{52}$$

where the overbar denotes average values along the characteristic curves. If the locations and the flow properties at the points (1), (2), and (3) are known, then the above equations can be solved simultaneously for u_4, ρ_4, and P_4. The locations of these points are obtained by numerically integrating the characteristic curves passing through the solution point (4).

These can be written as follows,

$$(x_4 - x_3) = \overline{u_{34}}\ \Delta t \tag{53}$$

$$(x_4 - x_1) = (\overline{u+a})_{14}\ \Delta t \tag{54}$$

$$(x_4 - x_2) = (\overline{u-a})_{24}\ \Delta t \tag{55}$$

The initial values of $\overline{u_{34}}$, $(\overline{u+a})_{14}$, and $(\overline{u-a})_{24}$ are set equal respectively to u_B, $u_A + a_A$, $u_C - a_C$. Then an iterative technique is used to determine the locations and the flow parameters as follows,

Predictor algorithm.

<u>step</u> 1 Calculate x_1, x_2, and x_3 from Equations (53) - (55)

<u>step</u> 2 Determine the flow properties at each point found in step 1 by using a simple linear interpolation.

<u>step</u> 3 Set $\overline{u_{34}} = u_3$, $(\overline{u+a})_{14} = u_1$, and $(\overline{u-a})_{24} = u_2$.

<u>step</u> 4 Repeat steps 1-3 until the changes in locations are within a given tolerance.

<u>step</u> 5 Solve the system of equations (50) - (52) for u_4, ρ_4, and P_4.

At this point the Euler predictor algorithm is complete. This algorithm may be repeated a given number of times if desired by the corrector algorithm.

Corrector algorithm.

The corrector algorithm employs the predictor algorithm by taking average values along each characteristic curve. That is,

$$\overline{U_{34}} = (u_3 + u_4)/2,$$

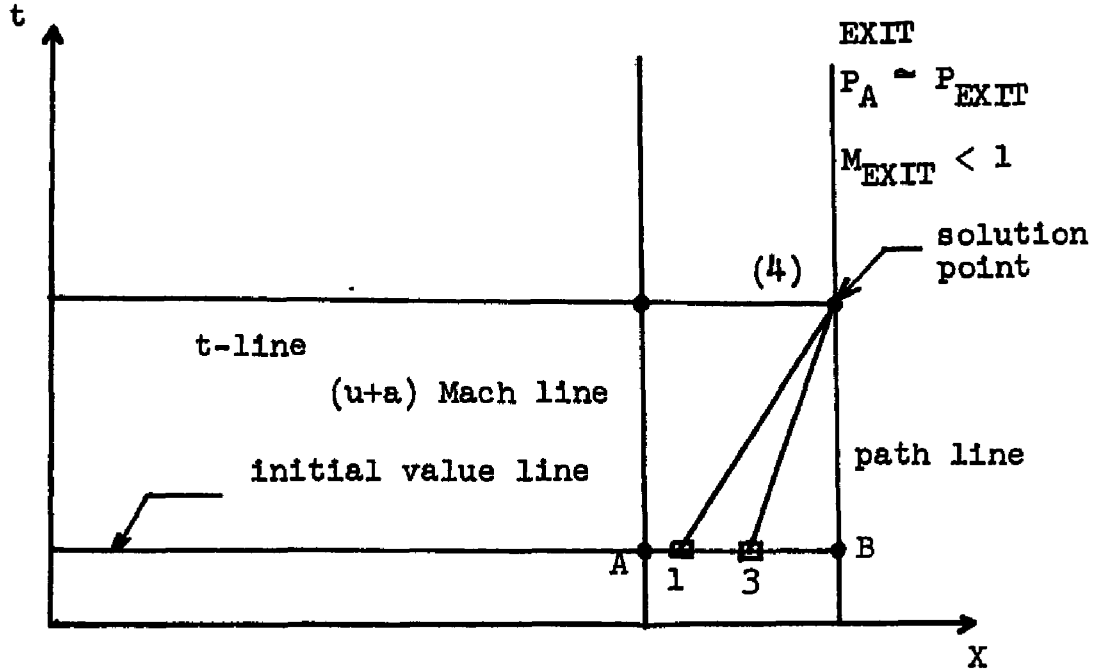
$$\overline{(u+a)}_{14} = \frac{(u_1 + u_4 + a_1 + a_4)}{2}, \text{ and}$$

$$\overline{(u-a)}_{24} = \frac{(u_2 + u_4 - a_2 - a_4)}{2}.$$

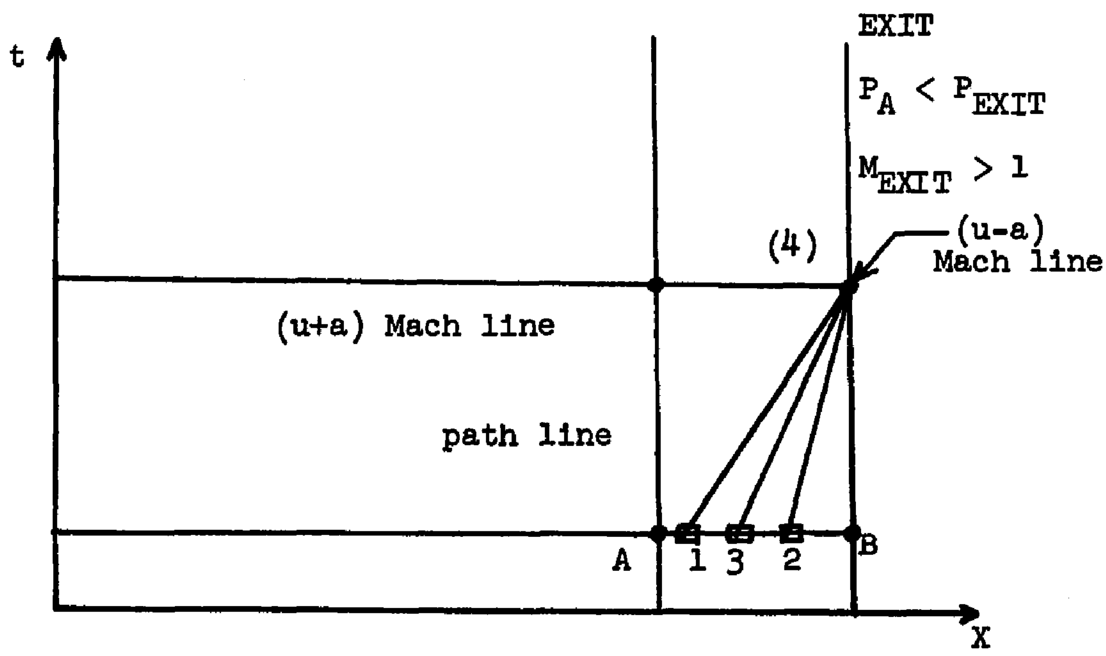
2.2.2 Outflow Grid Point.

The flow at the exit of the barrel can be a supersonic outflow or a subsonic outflow. For supersonic outflow, the finite difference grid is shown in Figure 20 b. Both characteristics originate within the physical domain. Therefore, point (4) can be considered as an interior point. In this case a reasonable approximation is to extrapolate from the interior values, since the error generated from the extrapolation can not affect the upstream conditions. For subsonic outflow, the finite difference is shown in Figure 20a. In this case the boundary condition is the specification of the ambient pressure, $P_4 = P_A$. The remaining flow properties are determined by numerically integrating the compatibility equations, Equations (42) and (44), along the path line and the Mach line, respectively.

The finite difference equations of Equations (42) and (44) can be written as follows:



a) subsonic outflow



b) supersonic outflow

Figure 20. Finite difference grid for an outflow boundary.

$$P_4 = P_A$$

$$(x_4 - x_1) = (\overline{u+a})_{14}\ \Delta t \tag{56}$$

$$(x_4 - x_3) = (\overline{u})_{34}\ \Delta t \tag{57}$$

$$(P_4 - P_3) - (a^2)_{34}(\rho_4 - \rho_3) = \overline{\Psi}_{34}\ \Delta t \tag{58}$$

$$(P_4 - P_1) + (\overline{\rho a^2})_{14}(u_4 - u_1) = (\overline{\rho a F + a^2 \Lambda + \Psi})_{14}\ \Delta t \tag{59}$$

Using the numerical algorithm described in section 2.2.1. the properties at the solution point (4) can be obtained.

2.2.3 Inflow Grid Point

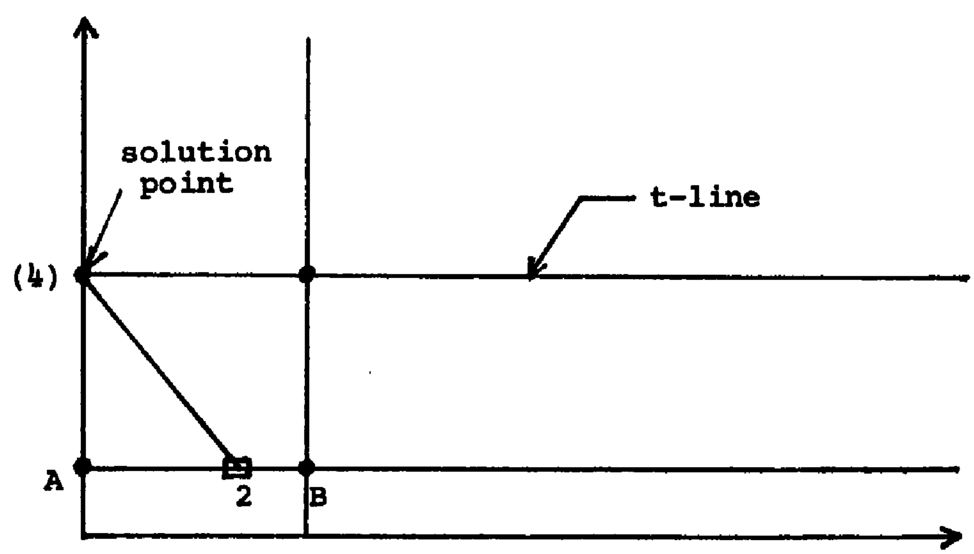


Figure 21 . Finite difference grid for a subsonic inflow

The flow at the barrel entrance can be a supersonic inflow or a subsonic inflow. For supersonic inflow all flow parameters must be specified. For subsonic inflow, the finite difference grid is shown in Figure 21 . In this case the specified flow parameters are the density and the velocity, i.e., u_4 and ρ_4. The pressure, P_4, can be determined by numerically integrating the compatibility equation, Equation (46), along the Mach line, Equation (45). In finite difference form this can be written,

$$(x_4 - x_2) = (\overline{u-a})_{24}\,\Delta t \tag{60}$$

$$(P_4 - P_3) - (\overline{\rho a^2})_{24}(u_4 - u_2) = (\overline{-\rho aF + a^2\Lambda\Psi})_{24}\,\Delta t \tag{61}$$

P_4 is then calculated by the algorithm presented in Section 2.2.1.

2.2.4 Projectile Grid Point

The finite difference grid for a moving projectile is shown in Figure 22 . The unspecified parameters are the flow properties at the left hand side and the right hand side of the projectile. The number of unknowns is equal to five. Five equations can now be written.

-Along the $(u+a)$-Mach line

$$dP + \rho a^2\,du = (\rho aF + a^2\Lambda + \Psi)\,dt \tag{62}$$

-Along the path line

$$dP - a^2 d\rho = \Psi\,dt \tag{63}$$

At the right hand side of the projectile we have,

-Along the $(u-a)$-Mach line

$$dP - \rho a^2\,du = (\Psi + a^2\Lambda - \rho aF)\,dt \tag{64}$$

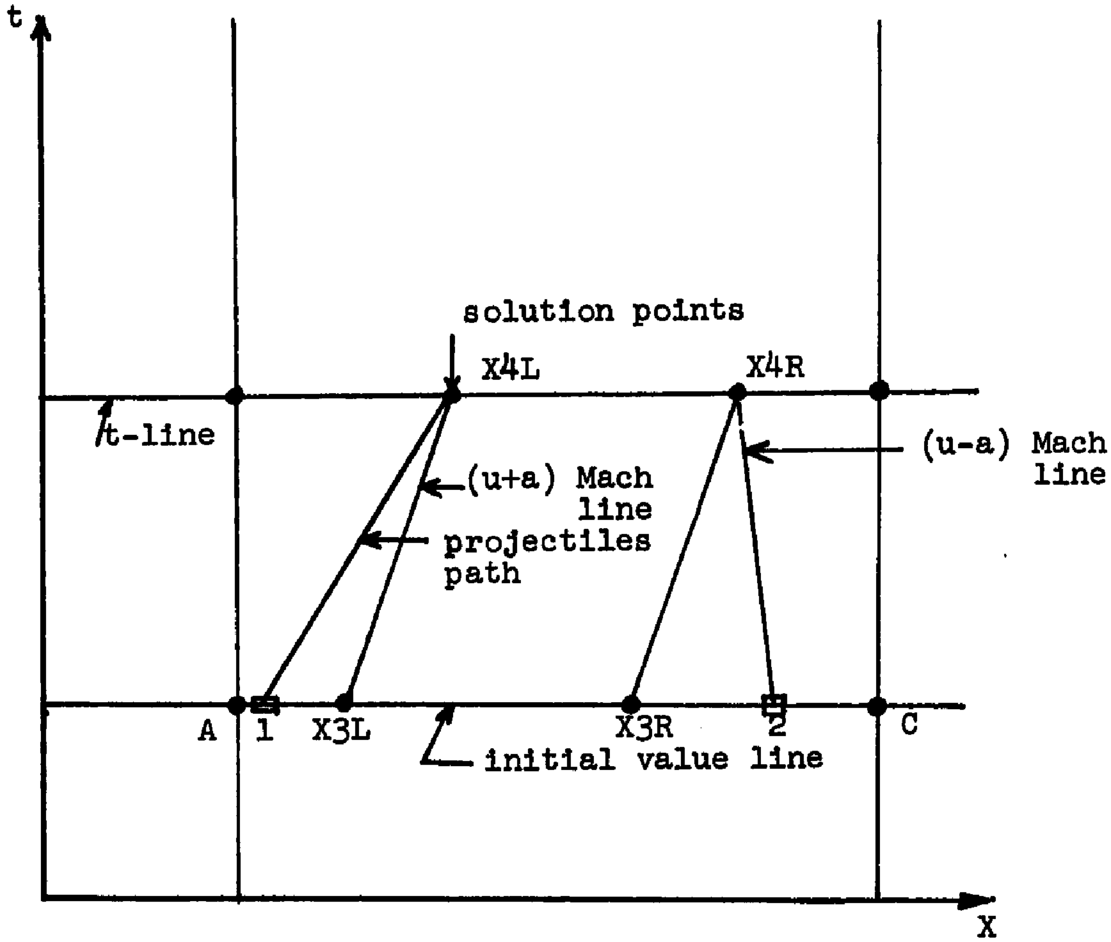


Figure 22 . Finite difference grid for a moving projectile.

-Along the path line

$$dP - a^2 d\rho = \Psi dt \tag{65}$$

Projectile's Motion.

$$du_s - [(P_{3L} - P_{3R})A_s - D_s]\ dt/m_s = 0 \tag{66}$$

In Equation (62) m_s is the projectile mass, P_{3L} is the pressure at the left hand side of the projectile, P_{3R} is the pressure at the right hand side of the projectile, A_s is the projectile's cross sectional area, and D_s is the friction force.

Equations (62) - (66) are written in finite difference form by the same methods presented in Section 2.2.1. Then they are solved simultaneously for u_4, P4L, R4L, R4R, and P4R.

2.2.5 Shock Wave Grid Point

The finite difference grid for a shock wave point is shown in Figure 23. The flow properties are discontinuous at any shock wave point. However the governing equations (37) - (40) fail to apply across discontinuities. The shock wave will be treated as a moving boundary separating two regions of continuous flow. The boundary is defined by two points, one the first computational region (shown in Figure 23a. as X3L) and one the second (shown in Figure 23b. as X3R). The unknown flow parameters are the values of velocity, density, and pressure on the left-hand side and on the right-hand side of the projectile, and the position and velocity of the shock at every time step. This problem is solved by combining the Rankine-Hugoniot equations and the method of characteristics as follows:

1. Detection of a shock wave.

A shock wave is detected if the pressure gradient, $\frac{\partial P}{\partial x}$, becomes infinite. However, such a detection is not possible. Instead, a shock is assumed to appear when $\frac{\partial x}{\partial P}$ vanishes. For this problem the prediction of the shock is given analytically by Moretti[11], and also by Thompson[12]. This can be written as,

$$t_s = \frac{2a_0}{(\gamma+1)\ \ddot{x}_s(0)} \tag{67}$$

$$x_s = a_0 t_s \quad , \quad w_s = a_0 \tag{68}$$

where x_s, w_s, $\ddot{x}_s(0)$, and a are, respectively, the position of the shock, velocity of the shock, initial acceleration of the projectile, and initial speed of sound.

2. Use of an iterative algorithm to solve for the flow properties.

<u>step</u> 1 Assume velocity of shock and gas in the right-hande side of the shock.

<u>step</u> 2 Determine the location of the shock.

<u>step</u> 3 Use the numerical algorithm for the interior point to determine the flow properties at point (see Figure 23b)

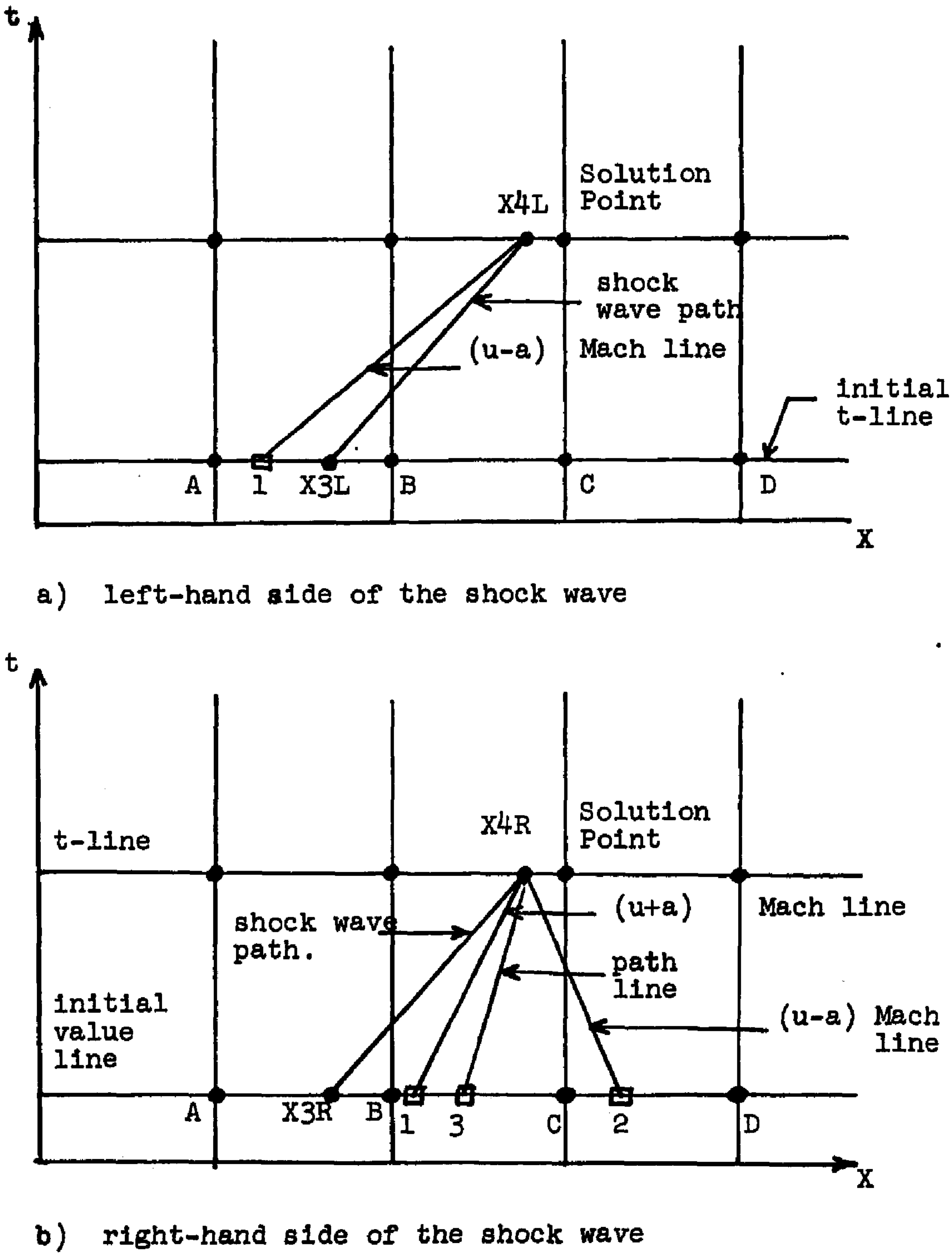


Figure 23 . Finite difference grid for a shock wave point.

step 4 Use Rankine-Hugoniot Equations across the shock wave.

step 5 Determine the flow properties on the left-hand side of the projectile by numerically integrating the compatibility Equation (44).

3. Results

Figures 24 through 27 show the numerical results obtained from the mathematical model used for the barrel flow. Projectile displacement, velocity, and pressure as functions of time are shown for three different cases. In the first case, the charge pressure, P_c, is set equal to 8,000.KPA. From Figure 24, it is seen that the projectile reaches the end of the barrel at time, t = 8.1 ms. At this time all calculations were stopped. Projectile muzzle velocity is obtained from Figure 24, (velocity = 400 m/s) at time t = 8.1 ms. The pressure at the base of the projectile versus time is also shown in Figure 24. Similar plots are shown in Figure 25. In this case the charge pressure is set equal to 5,000.KPA. The muzzle velocity obtained, is equal to 300 m/s and the time it occurs is, t = 10 ms.

Finally, in the third case, the charge pressure is set equal to 50,000 KPA. A shock is predicted to occur inside the barrel. The shock is assumed to form at time, t = 0, and at the origin, x = 0. From Figure 26, it is seen that the projectile leaves the tube with a muzzle velocity of 700 m/s at time, t = 4 ms. However referring to Figure 27, it is seen that, the shock reaches the barrel in less than 2.5 ms.

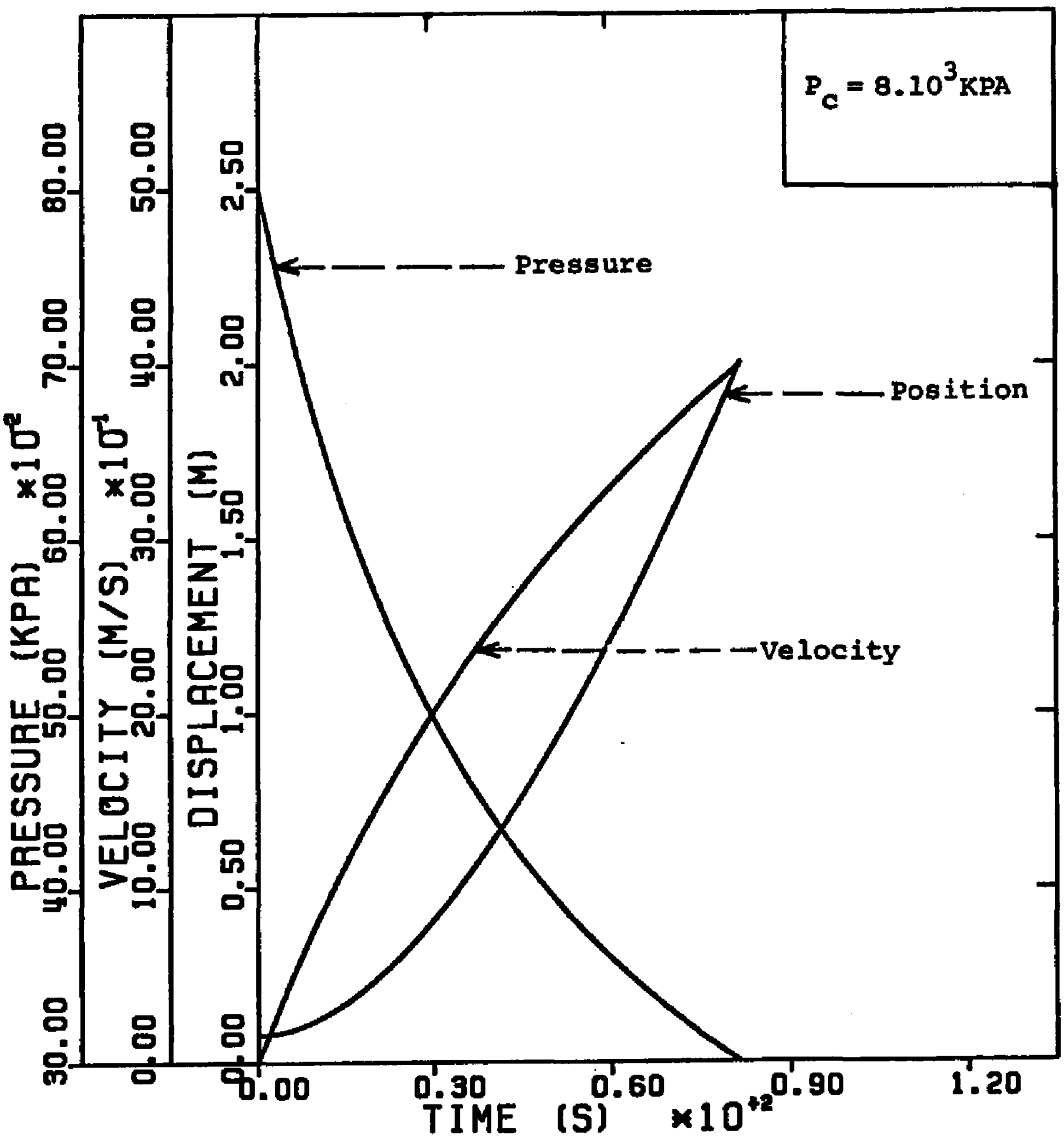


Figure 24. Projectile base pressure, velocity, and displacement as a function of time.

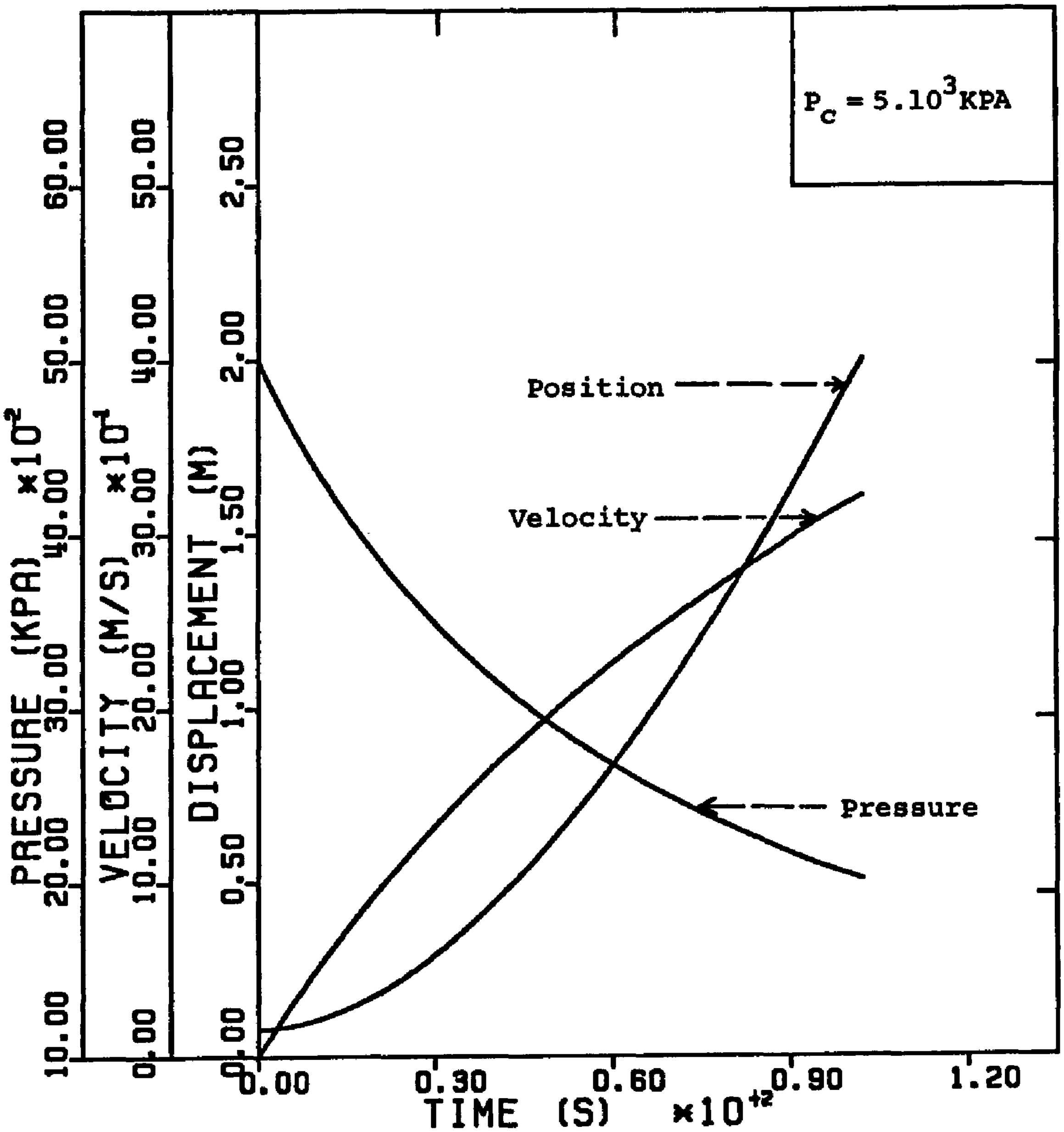


Figure 25. Projectile base pressure, velocity, and displacement as a function of time.

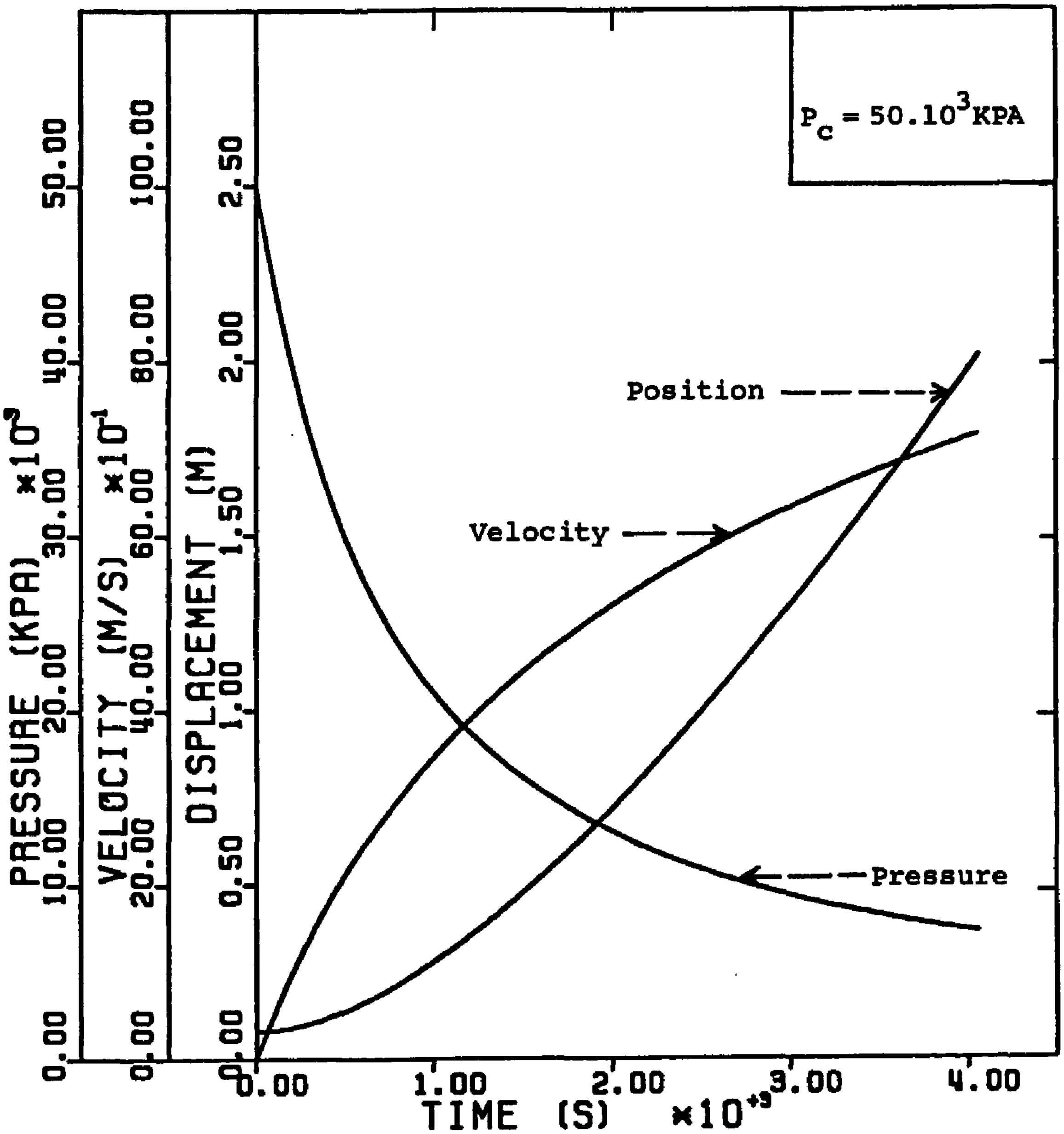


Figure 26. Projectile base pressure, velocity, and displacement as a function of time.

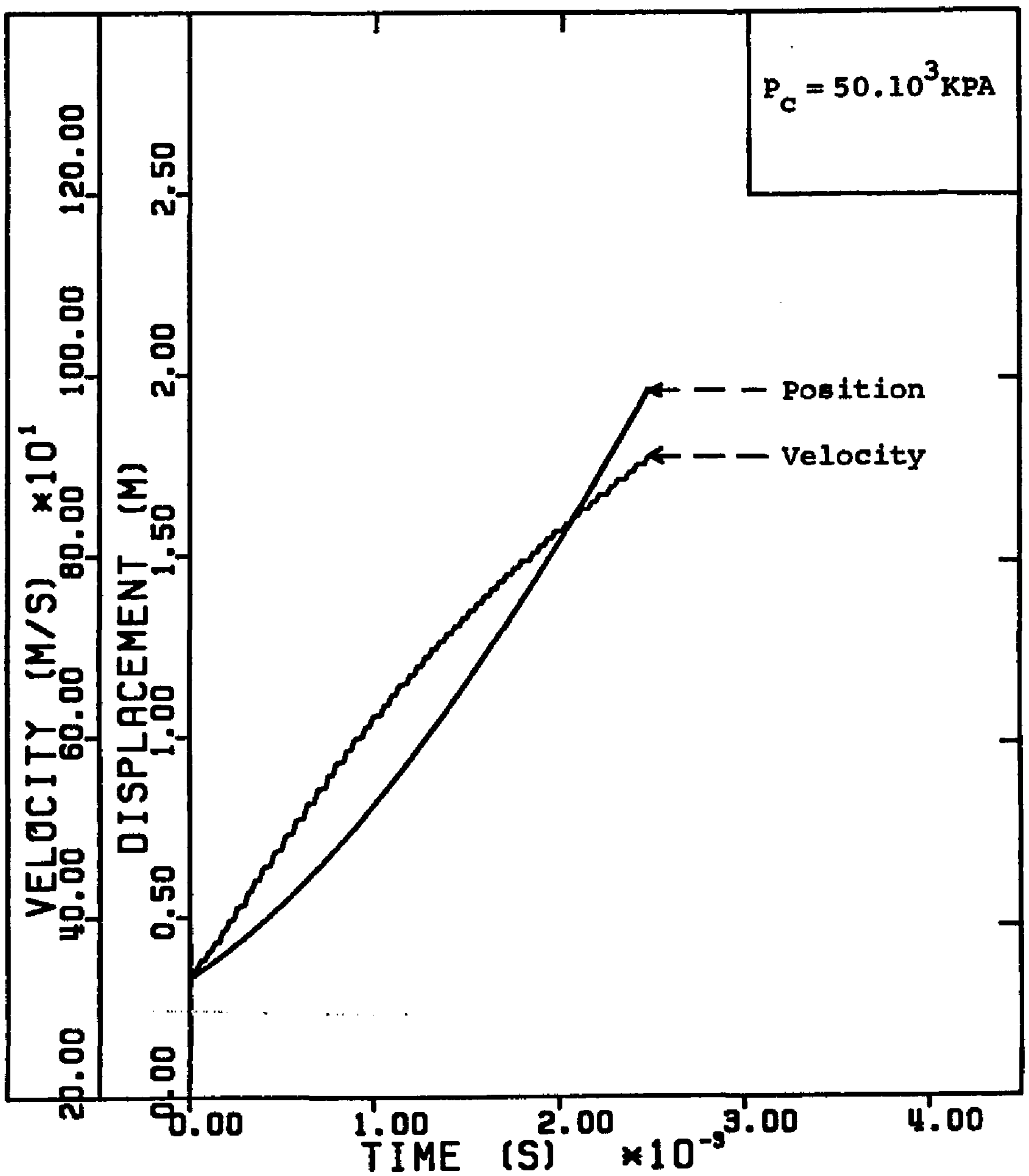


Figure 27. Shock position and velocity as a function of time.

C. MAIN CHAMBER FLOW

The flow in the main chamber is treated as a two-dimensional, time dependent flow. The governing equations, Equations (1) - (4), presented in Part A are then applicable for this case. However the boundary conditions and the geometry are more complex, see Figure 28 . The left and the right boundaries are both time dependent. The gas flows from the main chamber to the barrel through a time dependent area. Therefore, the model should be able to generate not only a variable grid but a time dependent one. Initially there is a very large pressure gradient at the barrel entrance. For this purpose an explicit artificial viscosity is used to stabilize the numerical method for shock wave calculations.

1. Physical and Computational Flow Spaces.

The physical space shown in Figure 28 is mapped into a computational space with uniform grid spacing by means of the following transformations

$$\xi(x,t) = \int_0^x \sqrt{1 + \left(\frac{\partial f}{\partial s}\right)^2 (s,t)} \quad ds \tag{69}$$

$$\eta(Y) = \int_0^Y \sqrt{1 + \left(\frac{dg}{ds}\right)^2} \quad ds \tag{70}$$

$$\tau = t \tag{71}$$

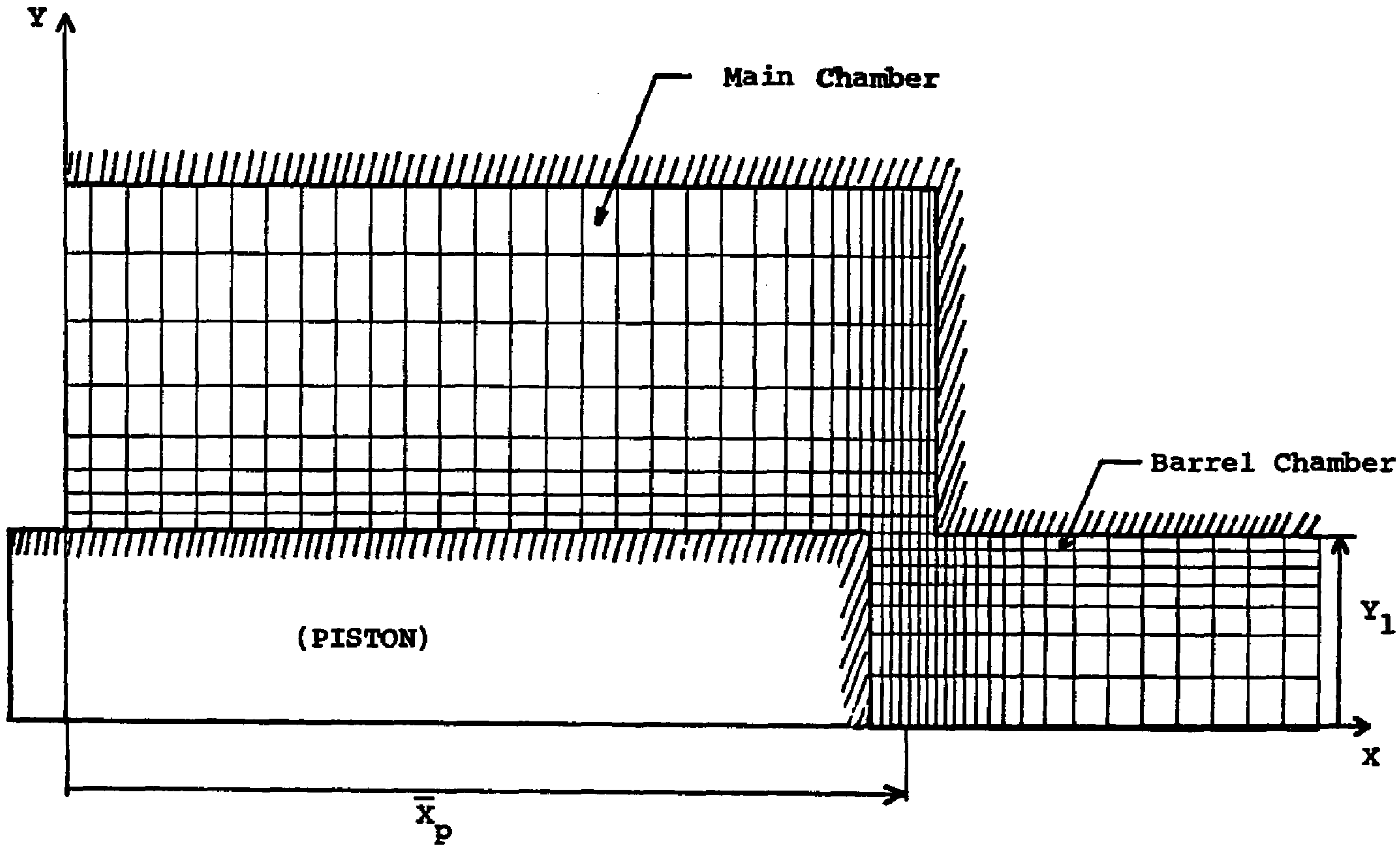


Figure 28 . Schematic illustration of the main chamber.

where,

$$f(x,t) = R_x \tanh (x - \overline{x_p}(t))$$

$$g(y) = R_y \tanh (y - Y_1)$$

R_x and R_y are parameters that control the distribution of the grid $(\overline{x}_p(t), Y_1)$ denotes the location of the point about which the grid is concentrated. Note that the first coordinate is a function of time.

The spatial and the time partial derivatives become,

$$\frac{\partial}{\partial x} = \omega_x \frac{\partial}{\partial \xi} \tag{73}$$

$$\frac{\partial}{\partial y} = \omega_y \frac{\partial}{\partial \eta} \tag{74}$$

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \tau} + \alpha \frac{\partial}{\partial \xi} \tag{75}$$

where $$\omega_x = \sqrt{1 + \frac{R_x^2}{\cosh^4 (x - x_p(t))}} \, , \tag{76}$$

$$\omega_y = \sqrt{1 + \frac{R_y^2}{\cosh^4 (y - y_1)}} \, , \tag{77}$$

and $$\alpha = \int_0^x \frac{\partial f}{\partial s} [1 + (\frac{\partial f}{\partial s})^2]^{-\frac{1}{2}} \frac{\partial^2 f}{\partial s \partial t} \, ds \tag{78}$$

2. Transformed Governing Equations

Using Equations (73) - (75), the governing equations, Equations (1) - (4), can now be written in the transformed space as follows:

$$\frac{\partial \rho}{\partial \tau} + u*\frac{\partial \rho}{\partial \xi} + v*\frac{\partial \rho}{\partial \eta} + \rho\, \omega_x \frac{\partial u}{\partial \xi} + \rho\, \omega_y \frac{\partial v}{\partial \eta} + \rho \frac{v}{Y} = QROT \quad (79)$$

$$\frac{\partial u}{\partial \tau} + u*\frac{\partial u}{\partial \xi} + v*\frac{\partial u}{\partial \eta} + \frac{\omega x}{\rho} \frac{\partial P}{\partial \xi} = QUT \quad (80)$$

$$\frac{\partial v}{\partial \tau} + u*\frac{\partial v}{\partial \xi} + v*\frac{\partial v}{\partial \eta} + \frac{\omega y}{\rho} \frac{\partial P}{\partial \eta} = QVT \quad (81)$$

$$\frac{\partial P}{\partial \tau} + u*\frac{\partial P}{\partial \xi} + v*\frac{\partial P}{\partial \eta} - a^2\left(\frac{\partial \rho}{\partial \tau} + u*\frac{\partial \rho}{\partial \xi} + v*\frac{\partial \rho}{\partial \eta}\right) = QPT \quad (82)$$

where,

$u* = \alpha + \omega_x U$

$v* = \omega_y v$

QROT = Artificial viscosity added to the Continuity Equation.

QUT = Viscous terms in the ξ-Momentum.

QVT = Viscous terms in the η-Momentum.

QPT = Viscous terms in the Energy Equation.

3. Numerical Method

The numerical methods used to integrate Equations (79) - (82) are similar to the numerical techniques described in Part A of Chapter II. That is, the computational space is divided into two sets of mesh points, the interior grid points and the boundary grid points. The numerical algorithms are described in detail in Part A. In this case the MacCormack scheme is stable if it satisfies the following condition given in the transformed space,

$$\Delta t \leq \frac{A}{|u|\omega_x/\Delta\xi + |v|\omega_y/\Delta\eta + a\sqrt{\omega_x^2/\Delta\xi^2 + \omega_y^2/\Delta\eta^2}} \qquad (83)$$

To determine the location of the piston, ie.e. $\overline{x_p}(t)$, and the velocity at the left boundary. The dynamics of the piston motion is needed. This can be written as,

$$Mp\,\frac{dv_p}{dt} = \int_{Ap} (P_c - P_g)\, dA - D \qquad (84)$$

where,

P_c = Pressure in the Main Chamber

P_g = Pressure in the Discharge Chamber

Mp = Piston Mass

Ap = Piston Area

D = Frictional Drag

The finite difference approximation to Equation (84) is used to update the velocity and the displacement of the piston so that the grid can be redistributed according to the piston displacement, i.e. $\overline{x_p}(t)$. Recall that $\overline{x_p}(t)$ is the coordinate of the point about which the grid is concentrated. This can be written as,

$$V_p^{N+1} = V_p^N + \frac{\Delta t}{Mp} (\overline{Pc} - \overline{Pg})\ Ap \tag{85}$$

$$X_p^{N+1} = X_p^N + V_p^N \Delta t + \frac{\Delta t^2}{2Mp} (\overline{Pc} - \overline{Pg})\ Ap \tag{86}$$

The overbar denotes the average values.

Now it is important to describe the artificial smoothing devices which can be used to stabilize the numerical method. To stabilize the numerical method for shock wave calculations, an explicit artificial viscosity is included. This can be written as,

$$QROT = \frac{C\rho}{\rho}[\ \omega_x \frac{\partial}{\partial \xi}\ (\mu_A\ \omega_x \frac{\partial \rho}{\partial \xi}\ ) + \ \omega_y\ \frac{\partial \rho}{\partial \xi}\ (\mu_A\ \omega_y \frac{\partial \rho}{\partial \eta}$$

$$+ \frac{\mu_A}{y}\ \omega_y\ \frac{\partial \rho}{\partial \eta}\ ] \tag{87}$$

where $$\mu_A = \frac{C_\mu\, C\, C\lambda\ \ \Delta\xi\, \Delta\eta\, \rho}{\omega_x \omega_y}\ [\ \omega_x \frac{\partial u}{\partial \xi} + w_y \frac{\partial v}{\partial \eta}\ + \frac{v}{y}\ ] \tag{88}$$

C, C_μ, $C\lambda$, and $C\rho$ are constants.

These artificial quantities, QROT and μ_A, are included into the model only when significant compressions are going to occur. The prediction of shock waves is detected by checking at each cell the divergence of the velocity. If it is less than zero, a compression wave is indicated. The computation is then automatic. If the divergence is greater than zero, then the artificial viscosity is set equal to zero. Other smoothing devices, fourth order damping terms and relaxation, are also used. It is very helpful to retain the fourth order damping terms to the second order MacCormack scheme in order to eliminate instabilities. Since discontinuities and errors made in the initial data can propagate and cause the solution to diverge. A fourth order term can be written as,

$$\frac{\partial^4 Q}{\partial \xi^4} = C_1 \frac{Q_{L+2} - 4Q_{L+1} + 6Q_L - 4Q_{L-1} + Q_{L-2}}{(\Delta\xi)^4} \tag{89}$$

where Q is a dependent variable. The dependent variables can also be smoothed by the following formula:

$$Q_{L,M} = SMP\ Q_{L,M} + (1 - SMP)(Q_{L+1,M} + Q_{L,M+1} + Q_{L-1,M} + Q_{L,M-1})/4.0 \tag{90}$$

where SMP is a constant and must be between 0.0 and 1.0. Equation (90), obtained from Reference 4. , is found to be very helpful for this numerical method. These are the basic differences with the numerical integration presented in Part A of this Chapter.

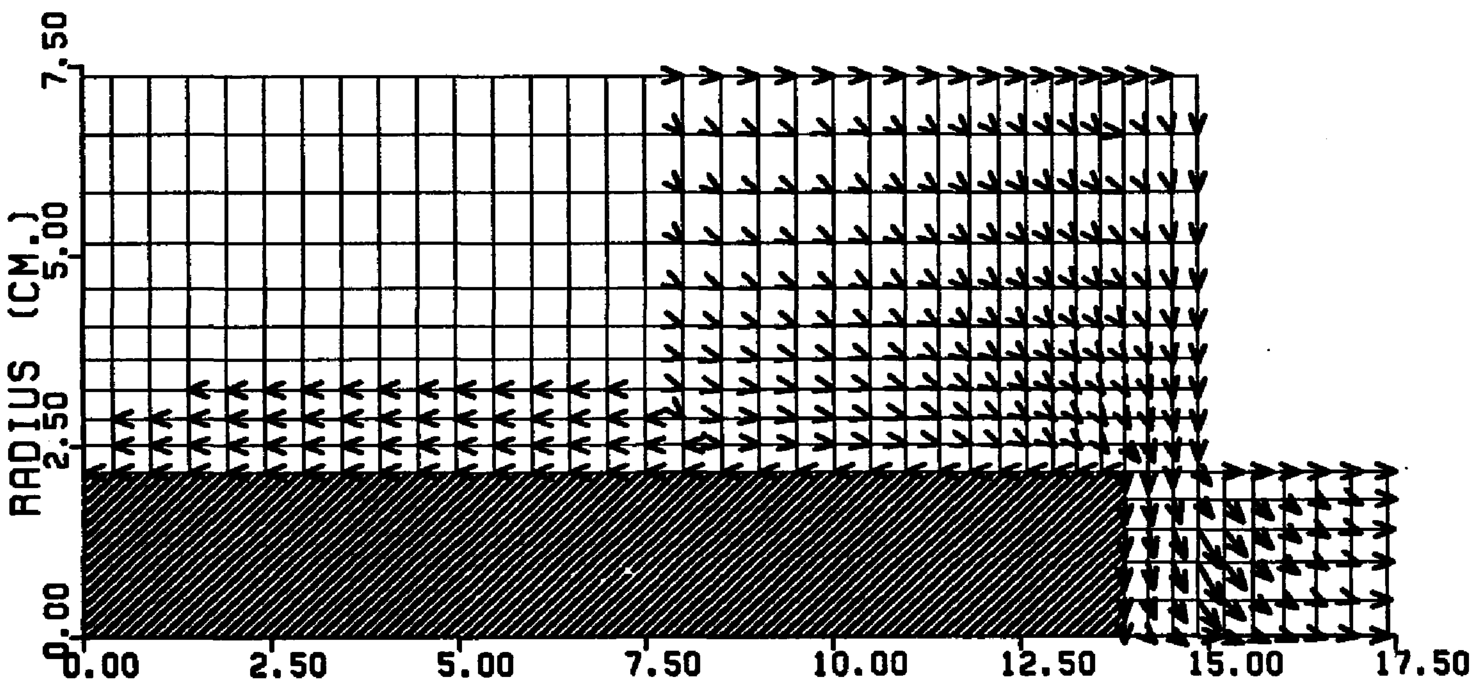
4. Results

Figures 29 through 34 show the velocity vector plots of the flow fields generated during main chamber operation. The flow development is shown for full travel of piston, (displacement = 3.81 CM). Piston displacement and velocity as functions of time, are also shown in Figures 29-34. The maximum velocity obtained in such a short distance is about 41 m/s. The full travel of piston lasted less than 1 ms. Therefore the role of the piston may be thought in some cases as a sudden opening of a valve. In these cases the motion of the piston will have a little effect on the muzzle velocity of the projectile. The assumption of a sudden opening valve may be then acceptable.

The model has generated a variable grid, but not time dependent. In this case, we let the piston grid points move through the grid lines. The flow properties at the piston boundaries are extrapolated from neighboring points.

TIME= 0.02 (MS) PISTON VELOCITY= 0.31 (M/S)

CHARGE PRESSURE=5,000.KPA

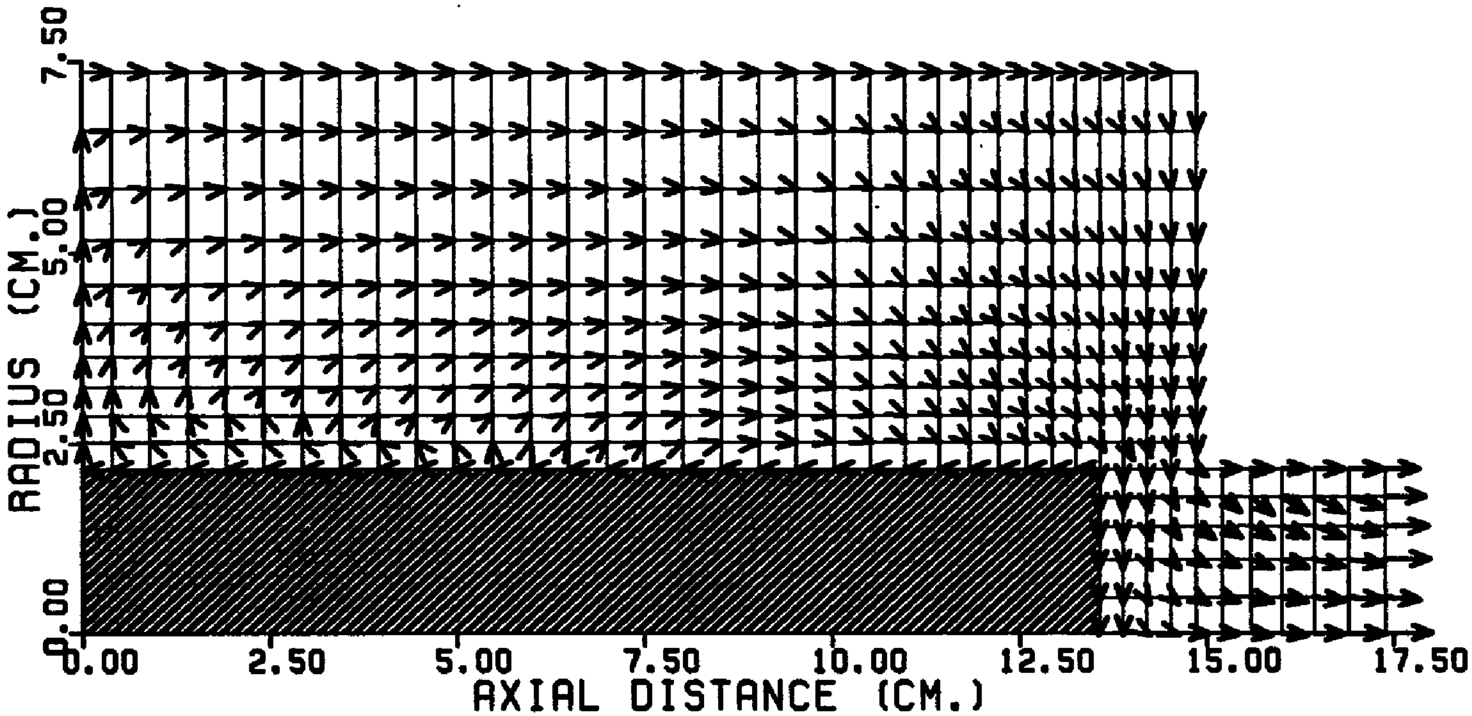


TIME STEP 20

FIGURE 29. VELOCITY DISTRIBUTION IN THE MAIN CHAMBER

TIME= 0.44 (MS) PISTON VELOCITY= 15.90 (M/S)

CHARGE PRESSURE=5,000.KPA



TIME STEP 339

FIGURE 30. VELOCITY DISTRIBUTION IN THE MAIN CHAMBER

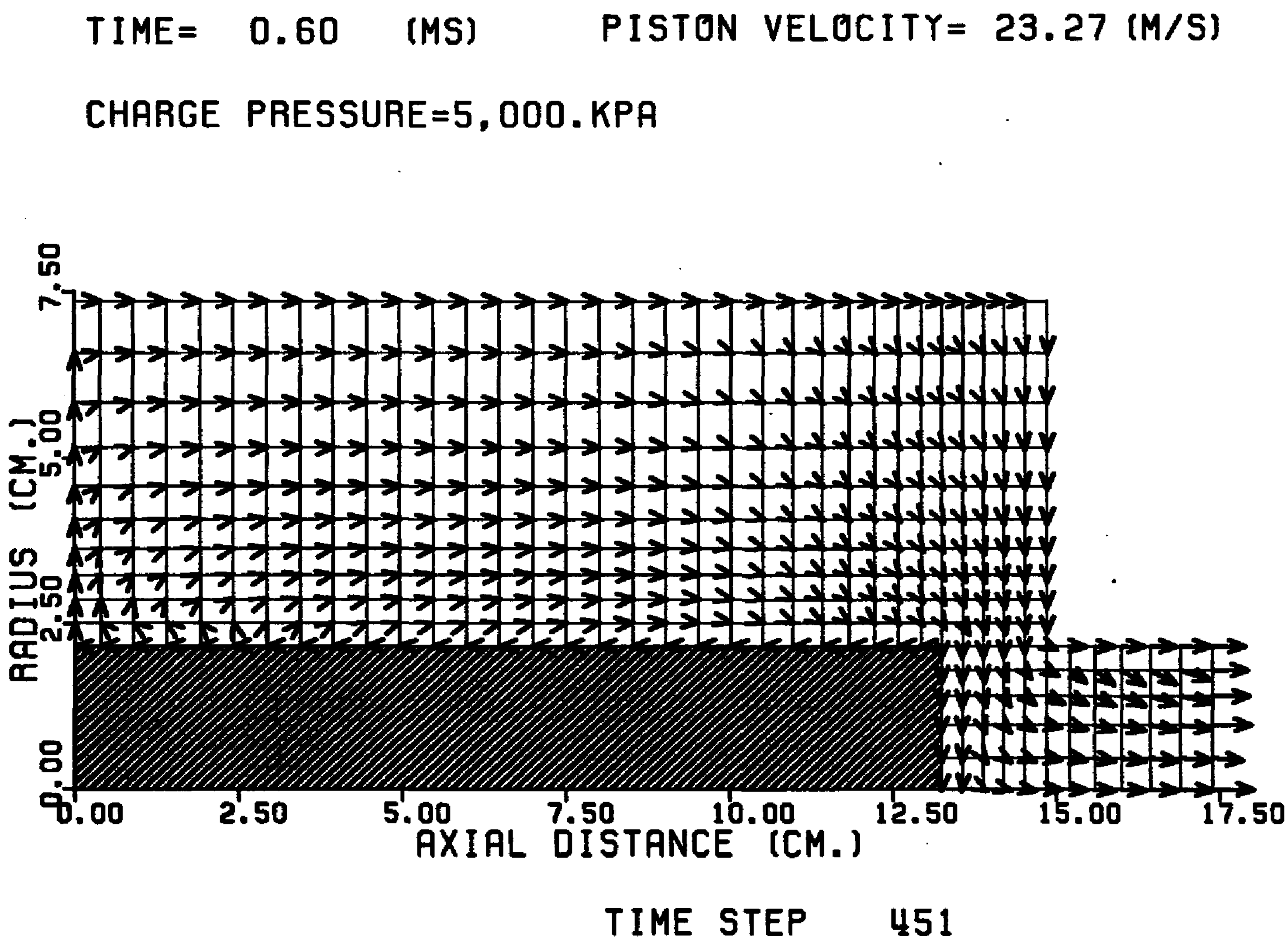
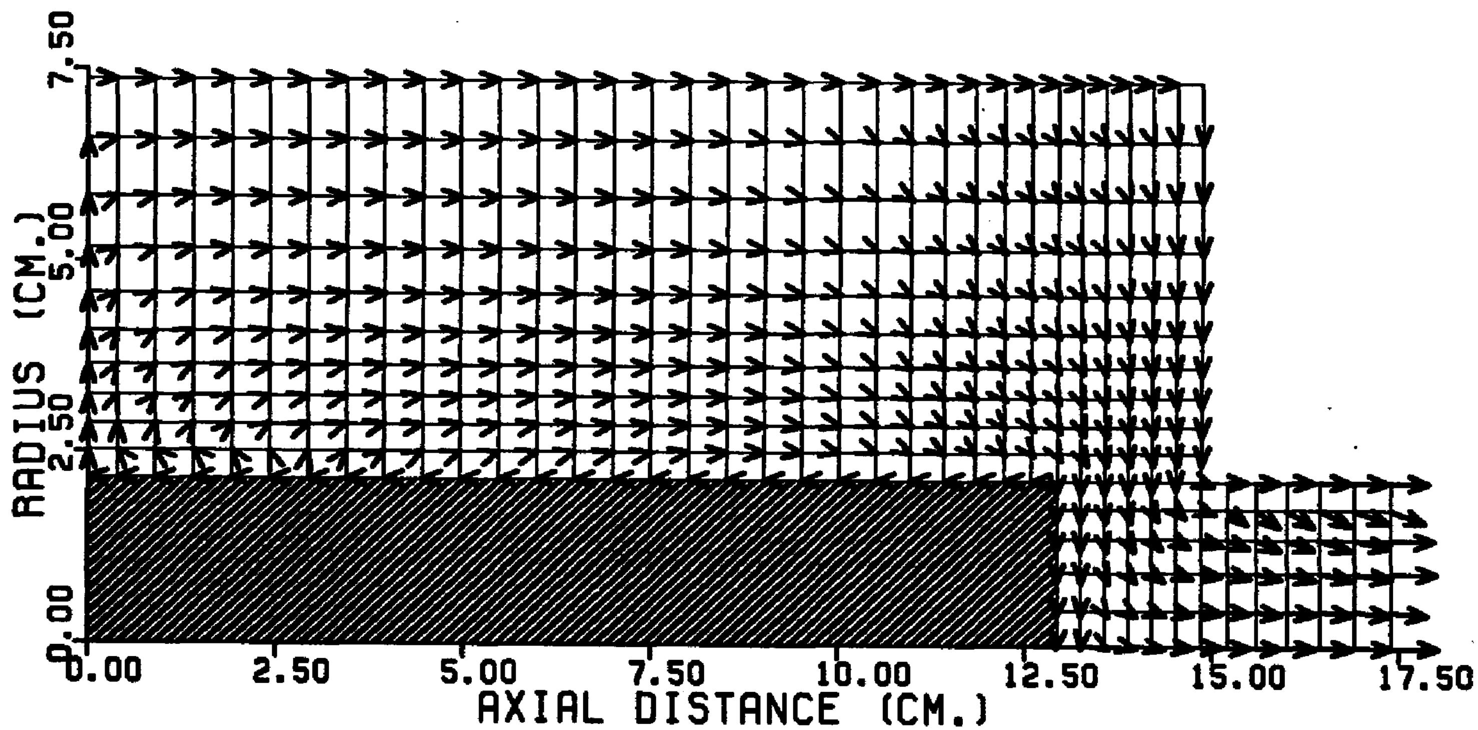


FIGURE 31. VELOCITY DISTRIBUTION IN THE MAIN CHAMBER

TIME= 0.72 (MS) PISTON VELOCITY= 29.42 (M/S)

CHARGE PRESSURE=5,000.KPA

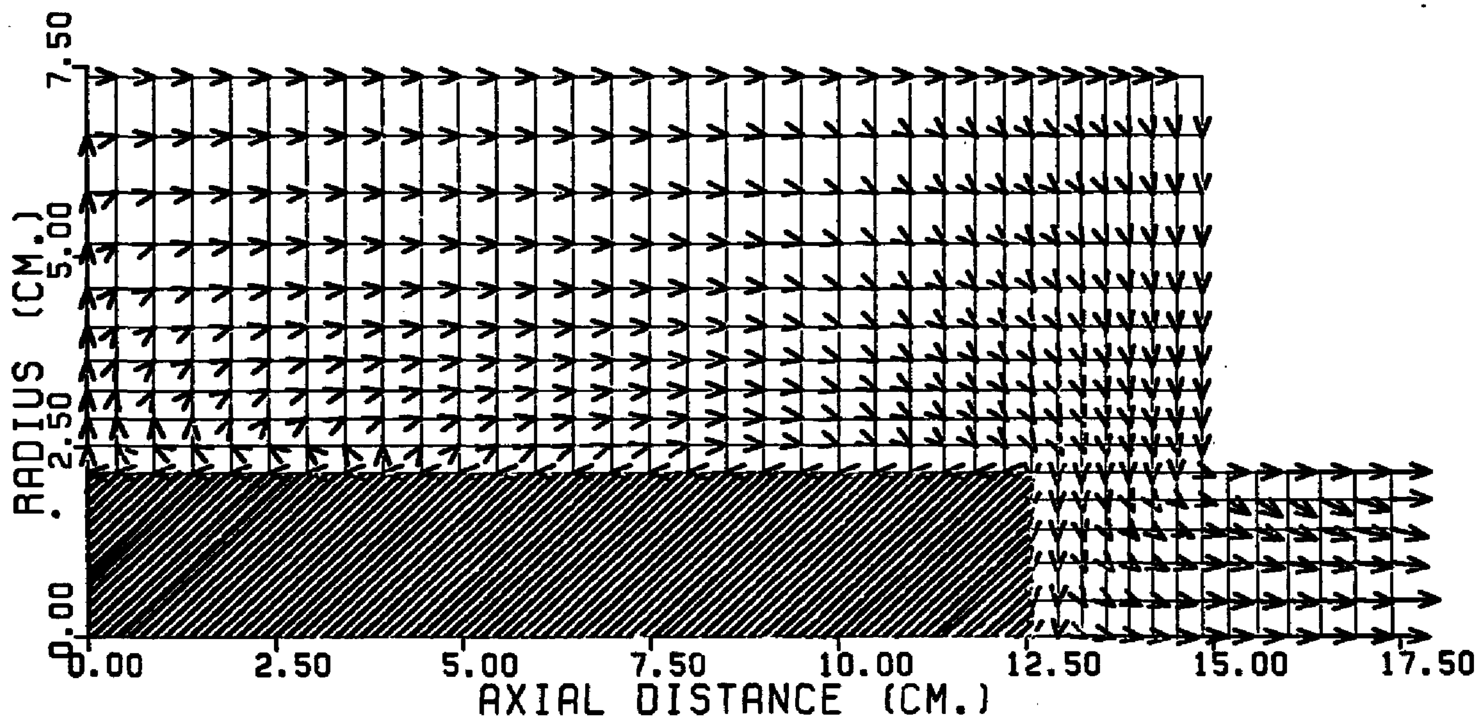


TIME STEP 533

FIGURE 32. VELOCITY DISTRIBUTION IN THE MAIN CHAMBER

TIME= 0.83 (MS) PISTON VELOCITY= 35.04 (M/S)

CHARGE PRESSURE=5,000.KPA

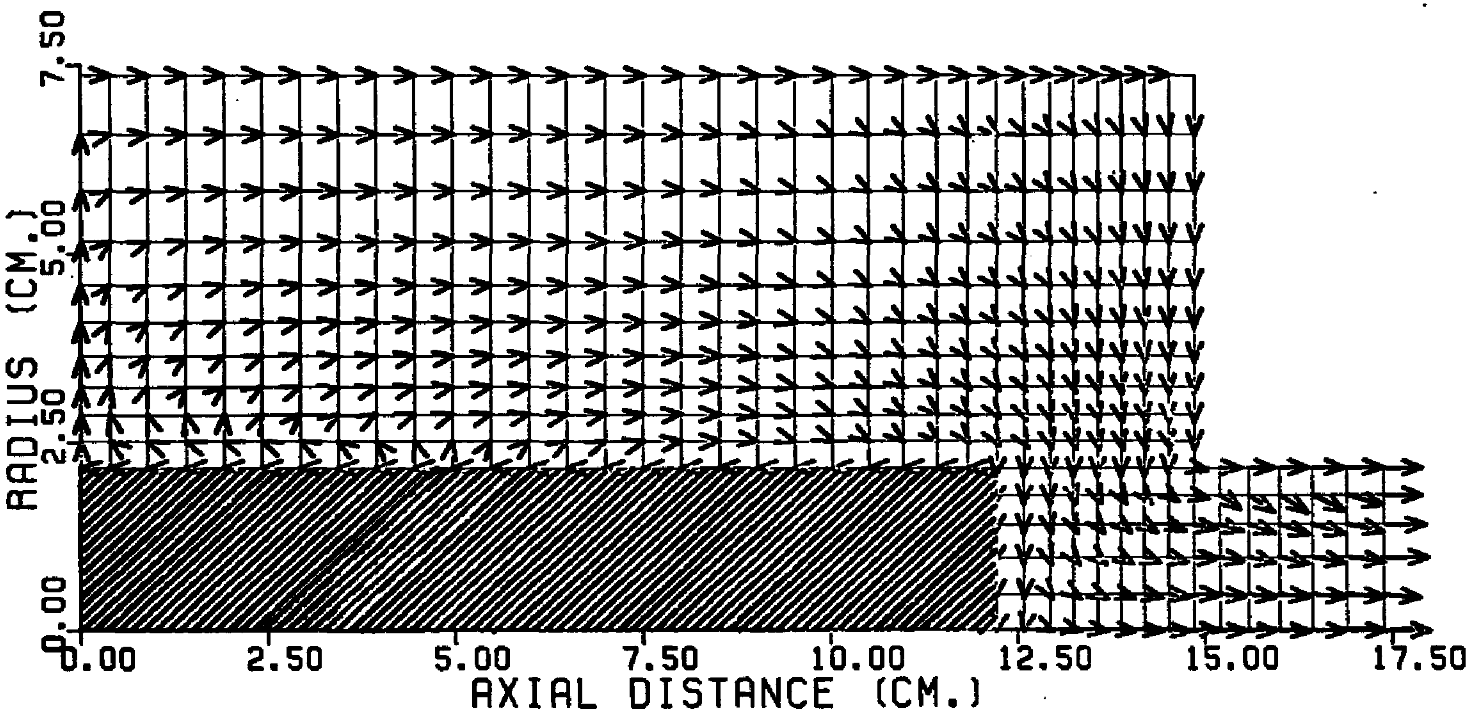


TIME STEP 604

FIGURE 33. VELOCITY DISTRIBUTION IN THE MAIN CHAMBER

TIME= 0.93 (MS) PISTON VELOCITY= 40.45 (M/S)

CHARGE PRESSURE=5,000.KPA



TIME STEP 671

FIGURE 34. VELOCITY DISTRIBUTION IN THE MAIN CHAMBER

REFERENCES

1. Richtmyer, R.D., "A Method for the Numerical Calculation of Hydrodynamics Shocks", Journal of Applied Physics, Vol. 21, p. 232, 1950.
2. Harhad, A., "Theoretical Analysis of a Gas Gun", Master thesis, Louisiana State University, December 1978.
3. MacCormack, R.W., "The Effect of Viscosity in Hypervolicity, Impact Cratering, "AIAA Paper No. 69-354, April, 1969.
4. Cline, M.C., "VNAP: A Computer Program for Computation of Two-Dimensional, Time-Dependent, Compressible, Viscous, Internal Flow", Los Alamos Scientific Laboratory LA-7326, November 1978.
5. Moretti, G., "The choice of a Time-Dependent Technique in Gas Dynamics", PIBAL Report No. 69-26, 1969.
6. Serra, R.A., "Determination of Internal Gas Flows by a Transient-Numerical Technique", AIAA Journal, Vol. 10, p. 603, 1972.
7. Chu, C.K., "Boundary Conditions in Finite Difference Fluid Dynamics Codes", Journal of Computional Physics, Vol. 15, p. 476, 1974.
8. Rudy, D.A., "A Non Reflecting Outflow Boundary Condition for Subsonic Navier-Stokes Calculations", Journal of Computational Physics, Vol.36, p. 55, 1980.

9. Pakarat , A., "Pressure Variation in a Gas Gun Chamber", Master thesis, Louisiana State University, August 1980.

10. Zucrow, M.J., and Hoffman, J.D., Vol. II Gas Dynamics, John Wiley and Sons, Inc., New York 1976.

11. Moretti, G., "A Pragmatical of Discretization Procedures For Initial-and-Boundary-Value Problems in Gas Dynamics and Their Influence on Accuracy." Poly-AE/AM Report No. 74-15, 1974.

12. Moretti, G., "Thoughts and Afterthoughts about Shock Computations", PIBAL Report No. 72-37.

13. Thompson, P., Compressible-Fluid Dynamics, McGraw-Hill Book Company, 1972.

BIBLIOGRAPHY

1. Smith, G. D., Numerical Solution of Partial Differential Equations, Clarendon Press, Oxford, London, 1978.
2. Mitchell, A. R., and Griffiths, D. F., The Finite Difference Method in Partial Differential Equations, John Wiley and Sons, Inc., New York, 1980.
3. Krier, H., Interior Ballistics of Guns, Volume 66, Progress in Astronautics and Aeronautics, AIAA 1979.
4. Yoshihara, H., "A Survey of Computational Methods For Two-dimensional and Three-dimensional Transonic Flows with Shocks, AGARD-LS-64, February, 1973 .

APPENDIX A

APPLICATION OF THE METHOD OF CHARACTERISTICS FOR UNSTEADY ONE-DIMENSIONAL FLOW

I. GOVERNING EQUATIONS

The governing equations (37) - (40), can be written as

$$\rho_t + u\rho_x + \rho u_x = \Lambda \tag{A-1}$$

$$u_t + u\,u_x + P_x/\rho = F \tag{A-2}$$

$$P_t + u\,P_x - a^2(\rho_t + u\rho_x) = \Psi \tag{A-3}$$

where the t and x subscripts denote the partial derivatives with respect to those variables.

II. CHARACTERISTIC CURVES

A linear combination of the equations of motion can be formed by multiplying Eqs.(A-1) - (A-3) by $\ell_i, (i=1,2,3)$ respectively, and then summing them. This linear combination can be written as

$$\ell_1(\rho_t + u\,\rho_x + \rho\,u_x - \Lambda) + \ell_2(u_t + u\,u_x + P_x/\rho - F)$$
$$+ \ell_3\ (P_t + u\,P_x - a^2\rho_t - a^2 u\,\rho_x - \Psi) = 0 \tag{A-4}$$

Rearrangement of Eq. (A-4) yields

$$(u\,\ell_1 - a^2 u\,\ell_3)\,\rho_x + (\ell_1 - a^2\ell_3)\,\rho_t + (u\,\ell_2 + \ell_1)\,u_x + \ell_2\,u_t$$
$$+ (\ell_2/\rho + u\,\ell_3)\,P_x + \ell_3 P_t = (\ell_1\Lambda + \ell_2 F + \ell_3\Psi) \tag{A-5}$$

Equation (A-5) may be interpreted as the sum of directional derivatives of u, P, and ρ in the directions of vectors $\overline{W_1}$, $\overline{W_2}$, and $\overline{W_3}$, respectively, where the components of the vectors $W_i(i=1,2,3)$ are the coefficients of the x and t

derivatives in Eq. (A-5). Thus

$$\overline{W_1} = (u\ell_1 - a^2 u\ell_3 , \ell_1 - a^2\ell_3) \tag{A-6}$$

$$\overline{W_2} = (u\ell_2 + \rho\ell_1 , \ell_2) \tag{A-7}$$

$$\overline{W_3} = (\ell_2/\rho + u\ell_3 , \ell_3) \tag{A-8}$$

Therefore, Eq. (A-5) may be written as

$$\nabla\rho \cdot W_1 + \nabla u \cdot W_2 + \nabla P \cdot W_3 = (\ell_1\Lambda + \ell_2 F + \ell_3\Psi) \qquad ((A-9)$$

where ∇ is the gradient vector; that is, $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial t})$

Equation (A-9) may be written as,

$$d_{\overline{W}_1}\rho + d_{\overline{W}_2}u + d_{\overline{W}_3}P = (\ell_1\Lambda + \ell_2 F + \ell_3\Psi) \tag{A-10}$$

where $d_{\overline{W}_1}\rho = \nabla\rho \cdot W_1$ is the derivative of ρ in the direction of the vector W_1.

A question is now posed: can the arbitrary factors ℓ_1, ℓ_2, and ℓ_3 be chosen so that the vectors $\overline{W_i}(i = 1,2,3)$ all lie along a curve in the xt plane. If such $\ell_i(i = 1,2,3)$ do exist, the curve that contains the vectors W_i is called the characteristic curve. Let N be the normal to that curve. The vector N, which is called the characteristic normal, is shown in Figure (A-1).

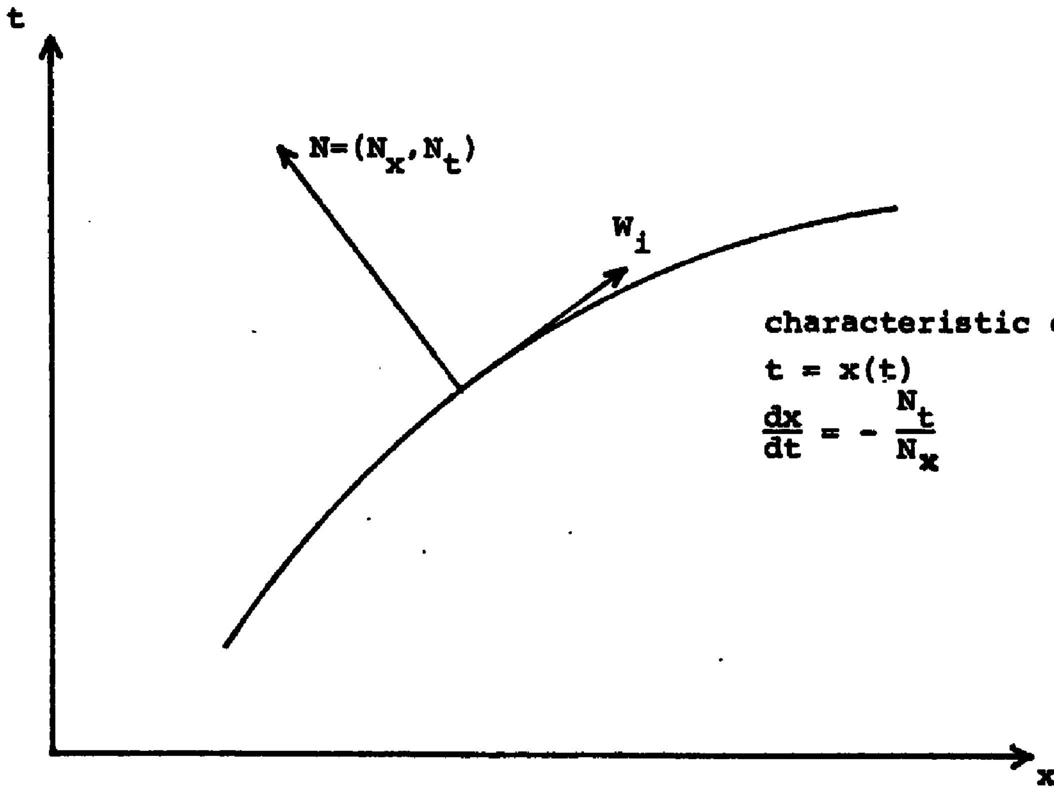


Figure A-1. Relationship between the vectors W_i and N, and the characteristic curve.

Since N is normal to $\overline{W_i}$ $(i = 1,2,3)$, we obtain

$$N \cdot \overline{W_i} = 0 \quad (i = 1,2,3) \qquad \text{(A-11)}$$

Equation (A-11) may be written as a system of three equations

$$\begin{aligned} (uN_x + N_t)\,\ell_1 - a^2(uN_x + N_t)\ell_3 &= 0 \\ \rho N_x \ell_1 + (uN_x + N_t)\ell_2 &= 0 \\ N_x/\ell \;\; \ell_2 + (uN_x + N_t)\ell_3 &= 0 \end{aligned} \qquad \text{(A-12)}$$

In matrix form Eqs. (A-12) become

$$\begin{vmatrix} u N_x + N_t & 0 & -a^2(u N_x + N_t) \\ \rho N_x & u N_x + N_t & 0 \\ 0 & N_x/\rho & (u N_x + N_t) \end{vmatrix} \begin{vmatrix} \ell_1 \\ \ell_2 \\ \ell_3 \end{vmatrix} = 0 \qquad \text{(A-13)}$$

For the system of equations specified by Eqs. (A-13) to have any solution other than the trivial solution $\ell_1 = \ell_2 = \ell_3 = 0$, the determinant must equal zero. Setting the determinant equal to zero, we obtain

$$(u N_x + N_t)[(u N_x + N_t)^2 - a^2 N_x^{\ 2}] = 0 \qquad \text{(A-14)}$$

Equation (A-14), which is called the characteristic equation, contains the information required to determine the components of the characteristic nommal, N, and hence, the characteristic curves.

Setting the first factor of Eq. (A-14) equal to zero, we obtain

$$u N_x + N_t = 0 \qquad \text{(A-15)}$$

Setting the second factor of Eq. (A-14) equal to zero, we obtain,

$$u N_x + N_t = \pm a N_x \qquad \text{(A-16)}$$

Eqs. (A-15) and (A-16) can be written as

$$\frac{dx}{dt} = u \qquad \text{(A-17)}$$

$$\frac{dx}{dt} = u \pm a \qquad \text{(A-18)}$$

Equation (A-17) represents the flow pathlines. Equation (A-18) represents the Mach lines.

III. SOLUTION FOR THE ℓ_i's

The compatibility equation that is applicable on a characteristic curve is determined from Eq. (A-10). First, the values of the ℓ_i's must be determined along the characteristic curves. Consider first the characteristic curve given by Eq. (A-17). Recall that $u N_x + N_t = 0$ along the pathline. Thus, Eqs. (A-13) become

$$\begin{vmatrix} 0 & 0 & 0 \\ \rho N_x & 0 & 0 \\ 0 & N_x/\rho & 0 \end{vmatrix} \begin{vmatrix} \ell_1 \\ \ell_2 \\ \ell_3 \end{vmatrix} = 0 \tag{A-19}$$

Since the rank of the matrix is two in Eq. (A-19), one of the ℓ_i's is a free variable.

Thus a solution may be written,

$$\ell_1 = 0, \ \ell_2 = 0, \ \ell_3 = 1 \tag{A-20}$$

Consider next the characteristic curve given by Eq. (A-18). Recall that $u N_x + N_t = \pm a N_x$ along the Mach lines. Thus, Eqs. (A-13) become

$$\begin{vmatrix} \pm a N_x & 0 & -a^3 N_x \\ \rho N_x & \pm a N_x & 0 \\ 0 & N_x/\rho & \pm a N_x \end{vmatrix} \begin{vmatrix} \ell_1 \\ \ell_2 \\ \ell_3 \end{vmatrix} = 0 \tag{A-21}$$

The rank of the coefficient matrix of Eq. (A-21) is two. There is one independent solution for ℓ_i. A solution can now be written

$$\ell_1 = a^2$$

$$\ell_2 = \pm \rho a \tag{A-22}$$

$$\ell_3 = 1$$

IV. COMPATIBILITY EQUATIONS

Substituting Eq. (A-20) into Eq. (A-4) yields

$$(P_t + u P_x) - a^2(\rho_t + u \rho_x) = \Psi \tag{A-23}$$

Substituting Eq. (A-22) into Eq. (A-4) yields

$$a^2(\rho_t + u \rho_x + \rho u_x - \Lambda) \pm \rho a(u_t + u u_x + P_x/\rho - F)$$
$$+ (P_t + u P_x - a^2\rho_t - a^2 u \rho_x - \Psi) = 0 \tag{A-24}$$

Equation (A-24) may be rearranged to yield

$$[P_t + (u \pm a) P_x] \pm \rho a[u_t \pm (u \pm a) U_x]$$
$$= (\Psi + a^2\Lambda \pm \rho a F) \tag{A-25}$$

Equations (A-23) and (A-25) can be written as

Along pathlines

$$dx = u\, dt$$
$$dP - a^2 d\rho = \Psi\, dt \tag{A-26}$$

Along Mach lines

$$dx = (u \pm a)dt$$

$$dP \pm \rho\, a\, du = (\Psi + a^2\Lambda \pm \rho\, a\, F) \qquad \text{(A-27)}$$

APPENDIX B

CONSTANT ETA REFERENCE PLANE CHARACTERISTIC RELATIONS

I. GOVERNING EQUATIONS

The governing equations (12) - (15) can be written as

$$\rho_\tau + u^*\rho_\xi + \rho\, w\, u_\xi = -v^*\rho_\eta - \rho\,\alpha\, u_\eta - \rho\, v/\bar{\eta} - \rho\,\beta\, v_\eta \qquad \text{(B-1)}$$

$$u_\tau + u^*u_\xi + w\, P_\xi/\rho = -v^*u_\eta - \alpha\, P_\eta + QuT \qquad \text{(B-2)}$$

$$v_\tau + u^*v_\xi = -v^*v_\eta - \beta\, P_\eta/\rho + QVT \qquad \text{(B-3)}$$

$$P_\tau + u^*\, P - a^2(\rho_\tau + u^*\rho_\xi) = -v^*P_\eta + a^2v^*\rho_\eta + QPT \qquad \text{(B-4)}$$

where the τ, ξ, and η subscripts denote partial derivatives with respect to those variables. QuT, QVT, and QPT denote, respectively, the viscous terms in the ξ-momentum, η-momentum, and energy equations. Letting Ψ_1, Ψ_2, Ψ_3, and Ψ_4 be, respectively, the right hand side of Eqs (B-1) - (B-4). Then Eqs (B-1) - (B-4) become

$$\rho_\tau + u^*\rho_\xi + \rho\, w\, u_\xi = \Psi_1 \qquad \text{(B-5)}$$

$$u_\tau + u^*u_\xi + w\, P_\xi/\rho = \Psi_2 \qquad \text{(B-6)}$$

$$v_\tau + u^*v_\xi = \Psi_3 \qquad \text{(B-7)}$$

$$P_\tau + u^*P_\xi - a^2(\rho_\tau + u^*\rho_\xi) = \Psi_4 \qquad \text{(B-8)}$$

II. CHARACTERISTIC CURVES

A linear combination of the equations of motion can be formed by multiplying Eqs. (B-5) - (B-8) by P_i, ($i = 1, 2, 3, 4$) and then summing them. This linear combination can be written as

$$\ell_1(\rho_\tau + u^*\rho_\xi + \rho w u_\xi - \Psi_1) + \ell_2(u_\tau + u^* u_\xi + w P_\xi/\rho - \Psi_2)$$

$$+ \ell_3(v_\tau + u^* v_\xi - \Psi_3) + \ell_4(P_\tau + u^* P_\xi - a^2\rho_\tau - a^2\rho \; - a^2 u^*\rho_\xi$$

$$- \Psi_4) = 0 \qquad \text{(B-9)}$$

Rearrangement of Eq. (B-9) yields

$$(u^*\ell_1 - a^2 u^*\ell_4)\rho_\xi + (\ell_1 - a^2\ell_4)\rho_\tau + (\rho w \ell_1 + u^*\ell_2)u_\xi$$

$$+ \ell_2 u_\tau + u^*\ell_3 v_\xi + \ell_3 v_\tau + (\ell_2 w/\rho + u^*\ell_4)P_\xi + \ell_4 P_\tau$$

$$= \sum_{i=1}^{4} \ell_i \Psi_i \qquad \text{(B-10)}$$

Equation (B-10) may be interpreted as the sum of directional derivatives of ρ, u, v, and P in the directions of vectors $\bar{W}_1$, $\bar{W}_2$, $\bar{W}_3$, and $\bar{W}_4$, respectively, where the components of the vectors W_i ($i = 1, 2, 3, 4$) are the coefficients of the ξ and τ derivatives in Eq. (B-10). Thus

$$\bar{W}_1 = (u^*\ell_1 - a^2 u^*\ell_4, \; \ell_1 - a^2\ell_4 \qquad \text{(B-11)}$$

$$\bar{W}_2 = (\rho w \ell_1 + u^*\ell_2 \, , \; \ell_2) \qquad \text{(B-12)}$$

$$\bar{W}_3 = (u^*\ell_3 \, , \; \ell_3) \qquad \text{(B-13)}$$

$$W_4 = (\ell_2 w/\rho + u^*\ell_4 \, , \; \ell_4) \qquad \text{(B-14)}$$

Therefore Eq. (B-10) may be written as

$$\nabla\rho \cdot W_1 + \nabla u \cdot W_2 + \nabla_V \cdot W_3 + \nabla P \cdot W_4 = \sum_{i=1}^{4} \ell_i \Psi_i \qquad \text{(B-15)}$$

A question is now posed: Can the arbitrary factors ℓ_1, ℓ_2, ℓ_3, and ℓ_4 be chosen so that the vectors $\bar{W}_i (i = 1,2,3,4)$ all lie along a curve in the ξ-τ plane? If such $\ell_i (i = 1,2,3,4)$ do exist, the curve that contains the vectors $\bar{W}_i$ is called the characteristic curve. Let N be the characteristic normal to that curve. Since N is normal to $\bar{W}_i (i = 1,4)$, we obtain.

$$N \cdot \bar{W}_i = 0 \quad (i = 1,4) \qquad \text{(B-16)}$$

Equation (B-16) may be written as a system of equations

$$(u^*\ell_1 - a^2 u^*\ell_4)N_\xi + (\ell_1 - a^2\ell_4)N_\tau = 0 \qquad \text{(B-17)}$$

$$(\rho w \ell_1 + u^*\ell_2)N_\xi + \ell_2 N_\tau = 0 \qquad \text{(B-18)}$$

$$u^*\ell_3 N_\xi + \ell_2 N_\tau = 0 \qquad \text{(B-19)}$$

$$\ell_2 w/\rho + u^*\ell_4)N_\xi + \ell_4 N_\tau = 0 \qquad \text{(B-20)}$$

In matrix form Eqs (B-17) - (B-20) become

$$\begin{vmatrix} u^*N_\xi+N_\tau & 0 & 0 & -a^2(u^*N_\xi+N_\tau) \\ \rho w N_\xi & u^*N_\xi+N_\tau & 0 & 0 \\ 0 & 0 & u^*N_\xi+N_\tau & 0 \\ 0 & 0 & 0 & u^*N_\xi+N_\tau \end{vmatrix} \begin{vmatrix} \ell_1 \\ \ell_2 \\ \ell_3 \\ \ell_4 \end{vmatrix} = 0 \qquad \text{(B-21)}$$

For the system of equations specified by Eqs. (B-21) to have any solution other than the trivial solution

$\ell_1 = \ell_2 = \ell_3 = 0$, the determinant must equal zero. Setting the determinant equal to zero, we obtain

$$(u^*N_\xi + N_\tau)^2[(u^*N_\xi + N_\tau)^2 - a^2w^2N_\xi^{\ 2}] = 0 \qquad \text{(B-22)}$$

Equation (B-22), which is called the characteristic equation, contains the information required to determine the components of the characteristic normal N and hence the characteristic curves.

Setting the first factor of Eq. (B-22) equal to zero, we obtain

$$u^*N_\xi + N_\tau = 0 \qquad \text{(B-23)}$$

Setting the second factor of Eq. (B-22) equal to zero, we obtain

$$u^*N_\xi + N_\tau = \pm a\,w\,N_\xi \qquad \text{(B-24)}$$

Equations (B-23) and (B-24) can be written

$$d\xi/d \quad = u^* \qquad \text{(B-25)}$$

$$d\xi/d \quad = u^* \pm a\,w \qquad \text{(B-26)}$$

Equation (B-24) represents the projection of the flow pathlines on the η = constant planes. Equation (B-26) represents the projection of the Mach cones on the η constant planes.

III. SOLUTION FOR THE ℓ'_i s

The compatibility equation that is applicable on a characteristic curve is determined from Eq. (B-15).

First the values of the ℓ_i' s must be determined along the characteristic curves. Consider first the characteristic curve given by Eq. (B-23). Substituting Eq. (B-23) into Eqs.(B-21) yields

$$\begin{vmatrix} 0 & 0 & 0 & \pm a^3 wN_\xi \\ \rho wN_\xi & \pm awN_\xi & 0 & 0 \\ 0 & 0 & \pm awN_\xi & 0 \\ 0 & wN_\xi/\rho & 0 & \pm awN_\xi \end{vmatrix} \begin{vmatrix} \ell_1 \\ \ell_2 \\ \ell_3 \\ \ell_4 \end{vmatrix} = 0 \qquad \text{(B-28)}$$

since the rank of the coefficient matrix of Eq. (B-28) is two. Thus,there are two free variables. Therefore, two solutions are possible

$$(0,\ 0,\ 1,\ 0) \quad \text{and} \quad (0,\ 0,\ 0,\ 1) \qquad \text{(B-27)}$$

Consider next the characteristic curve given by Eq. (B-24) Substituting Eq. (B-24) into Eq. (B-21) yields

$$\begin{vmatrix} \pm awN_\xi & 0 & 0 & \pm a^3 wN_\xi \\ \rho wN_\xi & \pm awN_\xi & 0 & 0 \\ 0 & 0 & \pm awN_\xi & 0 \\ 0 & wN_\xi/\rho & 0 & \pm awN_\xi \end{vmatrix} \begin{vmatrix} \ell_1 \\ \ell_2 \\ \ell_3 \\ \ell_4 \end{vmatrix} = 0 \qquad \text{(B-28)}$$

The rank of the coefficient matrix of Eq. (B-28) is three. There is only one free variable. Therefore,the solution is

$$(a^2,\ \pm\rho a,\ 0,\ 1) \qquad \text{(B-29)}$$

IV. COMPATIBILITY EQUATIONS

Substituting Eq. (B-27) into Eq. (B-10) yields

$$P_\tau + u^*P_\xi - a^2(\rho_\tau + u^*\rho_\xi) = \Psi_4 \qquad \text{(B-30)}$$

$$u^*V_\xi + V_\tau = \Psi_3 \qquad \text{(B-31)}$$

For $d\xi = u^* d\tau$.

Substituting Eq. (B-29) into Eq. (B-10) yields

$$dP \pm \rho a\, du = (\Psi_4 + a^2\Psi_1 \pm \rho a\, \Psi_2)d\tau \qquad \text{(B-32)}$$

For $d\xi = (u^* \pm aw)d\tau$.

APPENDIX C

CONSTANT ZETA REFERENCE PLANE CHARACTERISTIC RELATIONS

I. GOVERNING EQUATIONS

The governing equations (12) - (15), can be written as

$$\rho_\tau + v^*\rho_\eta + \rho\alpha u_\eta + \rho\beta v_\eta = -u^* - \rho w u_\xi - \rho v/\bar{\eta} \qquad \text{(C-1)}$$

$$u_\tau + v^* u_\eta + \alpha P_\eta/\rho = -u^* u_\xi - w P_\xi/\rho + QUT \qquad \text{(C-2)}$$

$$v_\tau + v^* v_\eta + \beta P_\eta/\rho = -u^* v_\xi + QVT \qquad \text{(C-3)}$$

$$P_\tau + v^* P_\eta - a^2(\rho_\tau + v^*\rho_\eta) = -u^* P_\xi + a^2 u^* \rho_\xi + QPT \qquad \text{(C-4)}$$

where the τ, ξ, and η subscripts denote partial derivatives with respect to those variables. Letting Ψ_1, Ψ_2, Ψ_3, and Ψ_4 be respectively the right hand side of Eqs. (C-1) - (C-4). Then Eqs. (C-1) - (C-4) become

$$\rho_\tau + v^*\rho_\eta - \rho\alpha u_\eta + \rho\beta v_\eta = \Psi_1 \qquad \text{(C-5)}$$

$$u_\tau + v^* u_\eta + P_\eta/\rho = \Psi_2 \qquad \text{(C-6)}$$

$$v_\tau + v^* v_\eta + \beta P_\eta/\rho = \Psi_3 \qquad \text{(C-7)}$$

$$P_\tau + v^* P_\eta - a^2(\rho_\tau + v^*\rho_\eta) = \Psi_4 \qquad \text{(C-8)}$$

II. CHARACTERISTIC CURVES

Using the same method as that described in Appendix B, we have

$$\nabla\rho \cdot \overline{W}_1 + \nabla u \cdot \overline{W}_2 + \nabla v \cdot \overline{W}_3 + \nabla P \cdot \overline{W}_4 = \sum_{i=1}^{4} \ell_i \Psi_i \qquad \text{(C-9)}$$

where

$$\begin{aligned}
\overline{W}_1 &= (\rho\ell_1 v^* - a^2 v^* \ell_4 \,,\, \ell_1 - a^2 \ell_4) \\
\overline{W}_2 &= (\rho\alpha\ell_1 + v^*\ell_2 \,,\, \ell_2) \\
\overline{W}_3 &= (\rho\beta\ell_1 + v^*\ell_3 \,,\, \ell_3) \\
\overline{W}_4 &= (\alpha\ell_2/\rho + \beta\ell_2/\rho + v^*\ell_4 ,\, \ell_4)
\end{aligned} \qquad \text{(C-10)}$$

The matrix form of Eq. $N \cdot W_i = 0 \quad i = 1,4$ can be written as follows.

$$\begin{vmatrix} v^*N_\eta + N_\tau & 0 & 0 & -a^2(v^*N_\eta + N_\tau) \\ \rho\,\alpha\,N_\eta & v^*N_\eta + N_\tau & 0 & 0 \\ \rho\,\beta\,N_\eta & 0 & v^*N_\eta & 0 \\ 0 & \alpha N_\eta/\rho & \beta N_\eta/\rho & v^*N_\eta + N_\tau \end{vmatrix} \begin{vmatrix} \ell_1 \\ \ell_2 \\ \ell_3 \\ \ell_4 \end{vmatrix} = 0 \qquad \text{(C-11)}$$

Setting the determinant equal to zero yields,

$$v^*N_\eta + N_\tau = 0 \qquad \text{(C-12)}$$

$$v^*N_\eta + N_\tau = \pm a\sqrt{\alpha^2 + \beta^2}\, N_\eta \qquad \text{(C-13)}$$

This can also be written as,

$$\frac{d\xi}{d\tau} = v^* \qquad \text{(C-14)}$$

$$\frac{d\xi}{d\tau} = v^* \pm a\alpha^* \quad \text{where } \alpha^* = \sqrt{\alpha^2 + \beta^2} \tag{C-15}$$

Equations (C-14) and (C-15) are the characteristic curves.

III. SOLUTIONS FOR THE ℓ_i' s.

1) Substituting Eq. (C-12) into Eq. (C-11) yields,

$$\ell_1 = 0, \ \ell_2 = \beta/\alpha, \ \ell_3 = 1, \ \ell_4 = 0. \tag{C-16}$$

and $\ell_1 = \ell_2 = \ell_3 = 0, \ \ell_4 = 1$ (C-17)

2) Substituting Eq. (C-13) into Eq. (C-11) yields,

$$\ell_1 = a^2, \ \ell_2 = \pm\rho\alpha\, a/\alpha^*, \ \ell_3 = \pm\rho\beta\, a/\alpha^*, \ \ell_4 = 1 \tag{C-18}$$

IV. COMPATIBILITY EQUATIONS

1) Substituting Eqs. (C-16) and (C-17) into Eq. (C-9) yields

$$\beta\, du - \alpha\, dv = (\beta\,\Psi_2 - \alpha\,\Psi_3)\, d\tau \tag{C-19}$$

$$dP - a^2 d\rho = \Psi_4\, d\tau \tag{C-20}$$

along the characteristic $d_\eta = v^* d\tau$

2) Substituting Eq. (C-18) into Eq. (C-9) yields

$$dP \pm \rho\, a\, du/\alpha^* \pm \rho\,\beta a\, dv/\alpha^* = (\Psi_4 + a^2\Psi_1 \pm \rho\alpha a\,\Psi_2/\alpha^*$$

$$\pm \rho\beta a\,\Psi_3/\alpha^*)\, d\tau \tag{C-21}$$

along the characteristics $d\eta = (v^* \pm \alpha^* a)d\tau$.

* APPENDIX D *

****

FORTRAN IV LISTING OF THE PROGRAM
FOR THE DISCHARGE CHAMBER FLOW

****

```
                        * MAIN PROGRAM *

              ***********************************

              THIS COMPUTER PROGRAM IS FOR THE
              COMPUTATION OF TWO DIMENSIONAL,
              TIME DEPENDENT FLOW,
              REPRESENTATIVE OF A DISCHARGE
              CHAMBER FLOW. THE DISCARGE CHAMBER
              WALL MAY BE ARBITRARY CURVED.

              ***********************************

      DIMENSION KSI(1000),XP(1000),XP1(50,20),YP1(50,20)
      DIMENSION TIME(1502),PRES1(1502),PRES2(1502),PRES3(1502)
      COMMON/ONESID/UD(4),VD(4),PD(4),ROD(4)
      COMMON/SOLUTN/U(50,20,2),V(50,20,2),P(50,20,2),
     $ RO(50,20,2)
      COMMON/CNTRLC/LMAX,MMAX,NMAX,TCONV,FDT,GAMMA,RGAS,GAM1,
     $ GAM2,GAM3,L1,L2,M1,M2,DX,DY,DT,ICHAR,N1D,IERR,IUI,IUO,
     $ DXR,DYR,RSTAR,N,N1,N3,RSTARS,G,PC,TC,LC,PLOW,ROLOW,RG,
     $ DAX,XEI
      COMMON/GEMTRL/XW(50),YW(50),NXNY(50),LT,XT,XI,XE,NDIM,
     $ XWI(50),YWI(50)
      COMMON/BCC/PT(20),TT(20),PE(20),IEXTRA,TW(20),ISUPER,
     $ UI(20),VI(20),PI(20),ROI(20),IEX
      COMMON/AV/CAV,NST,LSS,XMU,XLA,RKMU,XRO,QUT(50,20),
     $ QVT(50,20),QPT(50,20),QROT(50,20)
      COMMON/RV/CMU,CLA,CK,EMU,ELA,EK,CHECK,ITM,TML
      REAL MN3,NXNY,LC,KSI
C
C     SET DEFAULT VALUES
C
      RSTARS=0.0
      SMACH=0.0
      PE(1)=0.
      TT(1)=0.0
      PE(1)=14.7
      LC=12.0
      XMU=0.0
      XLA=1.0
      RKMU=0.7
      NCOUNT=1
      XRO=0.6
      TW(1)=-1.0
      PE(2)=-1
C
      TCONV=0.003
      FDT=0.5
      CHU=0.0
```

```
      T=0.0
      NPOLT=800
      JCOUNT=1
      N1=1
      N3=2
      E=1.
      NDIM=1
      GAMMA=1.4
      RGAS=53.35
      G=32.174
      PC=144.0
      GAM1=GAMMA/(GAMMA-1.0)
      GAM2=(GAMMA-1.0)/2.0
      GAM3=(GAMMA+1.0)/(GAMMA-1.0)
      RG=RGAS*G
      PLOW=0.01*PC
      ROLOW=0.0001/G
C
C     ARTIFICIAL VISCOSITY
C
      CAV=0.0
      XMU=.4
      XLA=1.0
      RKMU=.7
      XRO=.6
      NST=0
      SMP=0.95
      LSS=2
      SMACH=0.0
      IAV=1
C
C     PARAMETERS DEFINING THE MOLECULAR VISCOSITY
C
      CMU=0.0000009643
      CLA=-0.0000006429
      CK=0.00127
      EMU=0.5
      ELA=0.5
      EK=0.5
      LSS=2
      CAV=0.4
      SMACH=0.0
      ITM=0.0
      CMU=CMU/47.88/1.88**EMU
      CLA=CK*0.125/1.8**EK
      CMU=CMU*LC
      CLA=CLA*LC
      CK=CK*LC
      NOSLIP=1
      NOSLIP=0
```

```
      CHECK=ABS(CMU)+ABS(CLA)+ABS(CK)
      CHECK=0.0
      BI=1000000.
      BI=2500000.0
      PT(1)=BI
      PE(1)=14.7*PC
      TT(1)=540.
      IUI=1
      IUO=1
      IEXTRA=0
      IEX=1
      ISUPER=0
      N1D=1
      MMAX=6
      LMAX=48
C
C     INITIAL CONDITIONS
C
      DO 120 M=2,MMAX
      PT(M)=PT(1)
      TT(M)=TT(1)
      PE(M)=PE(1)
  120 CONTINUE
C
C     SET INDICIES AND ZERO VISCOUS TERM ARRAYS
C
      DO 320 M=1,MMAX
      DO 320 L=1,LMAX
      QUT(L,M)=0.0
      QVI(L,M)=0.0
      QPT(L,M)=0.0
      QROT(L,M)=0.0
  320 CONTINUE
      LJET=23
      JFLAG=1
      N1D=0
      ISUPER=-1
C
C     GRID GENERATION IN THE DISCHARGE CHAMBER
C
      RI=2.5
      RE=0.5
      XO=1.75
      RF=0.25
      XF=4.0
      XI=0.0
      XE=5.0
      GA=6.0
      GF=10.
      GF=20.
```

```
      LMAX=48
      MMAX=6
      KMAX=1000
      L1=LMAX-1
      M1=MMAX-1
      K1=KMAX-1
      DS=XE/FLOAT(K1)
      XP(1)=0.0
      KSI(1)=0.0
      SLOPE1=0.0
      S=0.0
      X=0.0
      DO 2 L=2,KMAX
      X=X+DS
      IF ( X.GT.3.0) GO TO 144
      SLOPE2=-(RI-RE)*GA*0.5/(COSH(GA*(X-XO))*COSH(GA*(X-XO)))
      GO TO 145
  144 SLOPE2=-(RE-RF)*GF*0.5/(COSH(GF*(X-XF))*COSH(GF*(X-XF)))
  145 S1=SQRT(1.+SLOPE1*SLOPE1)
      S2=SQRT(1.+SLOPE2*SLOPE2)
      S=S+DS*0.5*(S1+S2)
      KSI(L)=S
      XP(L)=X
      SLOPE1=SLOPE2
    2 CONTINUE
      XE1=KSI(KMAX)
      DKSI=XE1/FLOAT(L1)
      J=1
      SS=-DKSI
      DO 3 L=1,LMAX
      SS=SS+DKSI
      DO 4 K=J,KMAX
      IF ( ABS(SS-KSI(K)).LT.0.05) GO TO 5
    4 CONTINUE
    5 J=K
      XW(L)=XP(K)
      X=XW(L)
      IF ( X.GT.3.0) GO TO 146
      NXNY(L)=+(RI-RE)*GA*0.5/(COSH(GA*(X-XO))*COSH(GA*(X-XO)))
      YW(L)=(RI+RE)*0.5-(RI-RE)*TANH(GA*(X-XO))*0.5
      GO TO 147
  146 NXNY(L)=+(RE-RF)*GF*0.5/(COSH(GF*(X-XF))*COSH(GF*(X-XF)))
      YW(L)=(RE+RF)*0.5-(RE-RF)*TANH(GF*(X-XF))*0.5
  147 WRITE(6,100) KSI(K),XW(L),YW(L),NXNY(L)
    3 CONTINUE
C
C     PLOT THE PHYSICAL GRID
C
      CALL IDENT
      CALL PLOT (2.0,6.0,-3)
```

```
      CALL VECTOR(-1.93,7.88,6.67,7.88)
      CALL VECTOR(6.67,7.88,6.67,-3.12)
      CALL VECTOR (6.67,-3.12,-1.93,-3.12)
      CALL VECTOR (-1.93,-3.12,-1.93,7.88)
      DO 6 L=1,LMAX
      DYIO=YW(L)/FLOAT(M1)
      Y=-DYIO
      DO 6 M=1,MMAX
      Y=Y+DYIO
      XP1(L,M)=XW(L)
      YP1(L,M)=Y
    6 CONTINUE
      DO 55 L=1,LMAX
      CALL VTHICK(3)
      CALL VTHICK(3)
      CALL VECTOR(XP1(L,1),YP1(L,1),XP1(L,MMAX),YP1(L,MMAX))
   55 CONTINUE
      DO 44 L=1,L1
      DO 44 M=1,MMAX
      CALL VTHICK(3)
      CALL VTHICK(3)
      CALL VECTOR (XP1(L,M),YP1(L,M),XP1(L+1,M),YP1(L+1,M))
   44 CONTINUE
      ME=MMAX
      L1=LMAX-1
      L2=LMAX-2
      L3=LMAX-3
      M1=MMAX-1
      M2=MMAX-2
      DY=1./FLOAT(MMAX-1)
      DX=XE1/FLOAT(LMAX-1)
      DXR=1./DX
      DXX=DXR
      DYR=1./DY
      DXRS=DXR*DXR
      DYRS=DYR*DYR
      PA=103000.
      ROA=1.22
      COI=ROA*(BI/PA)**(0.714)
      BI=BI*PC/6894.8
      COA=1.22/(G*16.02)
      COI=COI/(G*16.02)
      DO 14 L=1,LMAX
      DO 14 M=1,MMAX
      U(L,M,N1)=0.0001
      V(L,M,N1)=0.0001
      P(L,M,1)=BI
      RO(L,M,1)=COI
   14 CONTINUE
      ISUPER=-1
```

```
      UD(1)=U(LJET-1,MMAX,N1)
      VD(1)=V(LJET-1,MMAX,N1)
      PD(1)=P(LJET-1,MMAX,N1)
      ROD(1)=RO(LJET-1,MMAX,N1)
      UD(2)=UD(1)
      VD(2)=VD(1)
      PD(2)=PD(1)
      ROD(2)=ROD(1)
      DO 131 L=1,LMAX
      XWI(L)=XW(LJET-1)
      YWI(L)=YW(L)
  131 CONTINUE
      NMAX=15000
      J=1
      TIME(1)=0.0
      PRES1(1)=P(1,1,1)/PC
      PRES2(1)=P(1,MMAX,1)/PC
      PRES3(1)=P(5,MMAX,1)/PC
      B=14.7*1.05
C
C     ENTER THE TIME STEP INTEGRATION
C
      DO 660 N=1,NMAX
      DO 450 L=1,LMAX
      W=SQRT(1.+NXNY(L)*NXNY(L))
      CALL MAP (0,L,MMAX,AL,BE,DE,LD1,AL1,BE1,DE1)
      DXDY=DXRS*W*W+DYRS*BE*BE
      DO 450 M=1,MMAX
      CALL EOS  (1,P(L,M,N1),RO(L,M,N1),TEMP,AS,D2,D3)
      UPA=ABS(U(L,M,N1))*W*DXR+ABS(V(L,M,N1))*BE*DYR+SQRT(AS*DXDY)
      IF (L.EQ.1.AND.M.EQ.1) UPAM=UPA
      IF (UPA.GT.UPAM) UPAM=UPA
  450 CONTINUE
      DT=FDT/UPAM
      DTT=DT/LC
      T=T+DTT
      ICHAR=1
      CALL INTER
      CALL YWALL
      DO 132 M=1,MMAX
      U(1,M,N3)=0.0
      V(1,M,N3)=0.0
      RO(1,M,N3)=RO(2,M,N3)
  132 CONTINUE
      CALL INLET
      CALL EXIT
      ICHAR=2
      CALL INTER
      CALL YWALL
      DO 133 M=1,MMAX
```

```
      U(1,M,N3)=0.0
      V(1,M,N3)=0.0
      RO(1,M,N3)=RO(2,M,N3)
  133 CONTINUE
      CALL INLET
      CALL EXIT
      CALL SMOOTH
      IF ( JCOUNT.EQ.400 ) GO TO 85
      JCOUNT=JCOUNT+1
      GO TO 84
   85 JCOUNT=1
C
C     PRINT THE SOLUTIONS
C
      WRITE(6,88) DTT,T
      DO 580 L=1,LMAX
      X=XW(L)
      DYIO=DY*YW (L)
      CALL MAP(0,L,1,AL,BE,DE,LD1,AL1,BE1,DE1)
      DYIO=DY/(BE)
      Y=-DYIO
      DO 580 M=1,MMAX
      Y=Y+DYIO
      VELMAG=SQRT (U(L,M,N3)**2+V(L,M,N3)**2)
      CALL EOS(5,P(L,M,N3),RO(L,M,N3),TEMP,AS,D2,D3)
      XMACH=VELMAG/SQRT(AS)
      PRES=P(L,M,N3)/PC
      RHO=RO(L,M,N3)*G
      XX=X
      YP=Y
      XP1(L,M)=XX
      YP1(L,M)=YP
      UP=U(L,M,N3)
      VP=V(L,M,N3)
      WRITE(6,920) L,M,XX,YP,UP,VP,PRES,RHO,VELMAG,XMACH
  580 CONTINUE
      CALL GRAPH(XP1,YP1,N3)
   84 NNN=N1
      N1=N3
      N3=NNN
  660 CONTINUE
C
C     FORMAT STATEMENTS
C
   88 FORMAT ( /,5X,'DT=',F12.9,3X,'T=',F12.9,/)
  100 FORMAT(5X,4F14.5)
  910 FORMAT(1H,25X,4H(CM),7X,4H(CM),6X,5H(M/S),7X,5H(M/S),7X,
     $ 6H  (KPA),7X,7H(KG/M3),5X,5H(M/S),10X,2HNO)
  920 FORMAT(1H,7X,2I5,4F12.4,F13.5,F12.6,2F12.4)
 1190 FORMAT(1H,10X,'------------------------------------'
```

```
$,'---------------------------------------------------------'
$,'--------')
 CALL EOJOB
  STOP
  END
```

```
      SUBROUTINE MAP(IP,L,M,AL,BE,DE,LD1,AL1,BE1,DE1)
C
C     ****************************************************************
C
C     THIS SUBROUTINE CALCULATES THE MAPPING FUNCTIONS
C
C     ****************************************************************
C
      DIMENSION KSI(1000),XP(1000),XP1(50,20),YP1(50,20)
      DIMENSION TIME(1502),PRES1(1502),PRES2(1502),PRES3(1502)
      COMMON/ONESID/UD(4),VD(4),PD(4),ROD(4)
      COMMON/SOLUTN/U(50,20,2),V(50,20,2),P(50,20,2),
     $ RO(50,20,2)
      COMMON/CNTRLC/LMAX,MMAX,NMAX,ICONV,FDT,GAMMA,RGAS,GAM1,
     $ GAM2,GAM3,L1,L2,M1,M2,DX,DY,DT,ICHAR,N1D,IERR,IUI,IUO,
     $ DXR,DYR,LSTAR,N,N1,N3,LSTARS,G,PC,TC,LC,PLOW,ROLOW,RG,
     $ DXX,XE1
      COMMON/GEMTRL/XW(50),YW(50),NXNY(50),LT,XT,XI,XE,NDIM,
     $ XWI(50),YWI(50)
      COMMON/BCC/PT(20),TT(20),PE(20),IEXTRA,TW(20),ISUPER,
     $ UI(20),VI(20),PI(20),ROI(20),IEX
      COMMON/AV/CAV,NST,LSS,XMU,XLA,RKMU,XRO,QUT(50,20),
     $ QVT(50,20),QPT(50,20),QROT(50,20)
      COMMON/RV/CMU,CLA,CK,EMU,ELA,EK,CHECK,ITM,TML
      REAL MN3,NXNY,LC
      BE=1.0/YW(L)
      IF(IP.EQ.0) RETURN
      Y=FLOAT(M-1)*DY
      AL=BE*Y*NXNY(L)
      DE=0.
      IF(IP.EQ.1) RETURN
      BE1=1.0/YW(LD1)
      AL1=BE1*Y*NXNY(LD1)
      DE1=0.
      RETURN
      END
```

```
      SUBROUTINE EOS(I1,PRESS,RHO,TEMP,D1,D2,MM3)
C
C     ****************************************************************
C
C      THIS SUBROUTINE CALCULATES THE EQUATION OF STATE
C     QUANTITIES
C
C     ****************************************************************
C
      DIMENSION KSI(1000),XP(1000),XP1(50,20),YP1(50,20)
      DIMENSION TIME(1502),PRES1(1502),PRES2(1502),PRES3(1502)
      COMMON/ONESID/UD(4),VD(4),PD(4),ROD(4)
      COMMON/SOLUTN/U(50,20,2),V(50,20,2),P(50,20,2),
     $ RO(50,20,2)
      COMMON/CNTRLC/LMAX,MMAX,NMAX,TCONV,FDT,GAMMA,RGAS,GAM1,
     $ GAM2,GAM3,L1,L2,M1,M2,DX,DY,DT,ICHAR,N1D,IERR,IUI,IUO,
     $ DXR,DYR,RSTAR,N,N1,N3,RSTARS,G,PC,TC,LC,PLOW,ROLOW,RG,
     $ DXX,XE1
      COMMON/GEMTRL/XW(50),YW(50),NXNY(50),LT,XT,XI,XE,NDIM,
     $ XWI(50),YWI(50)
      COMMON/BCC/PT(20),TT(20),PE(20),IEXTRA,TW(20),ISUPER,
     $ UI(20),VI(20),PI(20),ROI(20),IEX
      COMMON/AV/CAV,NST,LSS,XMU,XLA,RKMU,XRO,QUT(50,20),
     $ QVT(50,20),QPT(50,20),QROT(50,20)
      COMMON/RV/CMU,CLA,CK,EMU,ELA,EK,CHECK,ITM,TML
      REAL MM3,NXNY,LC
       GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,
     $ 150),I1
C
C      CALCULATE THE SOUND SPEED SQUARRED(D1=AS)
C
   10  D1=GAMMA*PRESS/RHO
       RETURN
C
C      CALCULATE THE TEMPERATURE
C
   20  TEMP=PRESS/(RHO*RG)
      RETURN
C
C     CALCULATE THE PRESSURE
C
   30 PRESS=TEMP*RHO*RG
       RETURN
C
C      CALCULATE THE DENSITY
C
   40  RHO=PRESS/(TEMP*RG)
      RETURN
C
C     CALCULATE THE TEMPERATURE AND SOUND SPEED SQUARED
```

```
C        ND1=AS)
C
   50   TEMP=PRESS/(RHO*RG)
        D1=GAMMA*PRESS/RHO
        RETURN
C
C       CALCULATE THE DENSITY AND SOUND SPEED SQUARED ND1=AS)
C
   60 RHO=PRESS/(TEMP*RG)
      D1=GAMMA*PRESS/RHO
        RETURN
C
C       CALCULATE THE SOUND SPEED SQUARED FROM THE TEMPERATURE
C       ND1=AS)
C
   70   D1=GAMMA*RG*TEMP
        RETURN
C
C       CALCULATE THE PRESSURE AND TEMPERATURE FROM THE
C      STAGNATION
C      CONDITIONS ND1=PT,D2=TT,NN3=MACH NO)
C
   80   DEM=1.0+GAM2*MN3*MN3
      PRESS=D1/DEM**GAM1
        TEMP=D2/DEM
        RETURN
C
C       CALCULATE THE STAGNATION TEMPERATUTE FROM THE
C      STATIC CONDITIONS ND1=PT,D2=TT,MN3=MACH NO)
C
   90   DEM=1.0+GAM2*MN3*MN3
        D1=PRESS*DEM**GAM1
        D2=TEMP*DEM
        RETURN
C
C       CALCULATE THE MACH NUMBER ND2=TT,MN3=MACH NO)
C
  100 MN3=SQRT((D2/TEMP-1.0)/GAM2)
        RETURN
C
C       CALCULATE THE DENSITY FOR THE UNDER-EXPANDED JET(D1=PD
C      ,D2=ROD)
C
  110   RHO=D2*(PRESS/D1)**(1.0/GAMMA)
      RETURN
C
C      CALCULATE THE DENSITY FOR THE OVER-EXPANDED JET  (ND1=PO
C      ,D2=ROD)
C
  120   PRO=PRESS/D1
```

```
      RHO=D2*(GAM3*PRO+1.0)/(PRO*GAM3)
      RETURN
C
C      CALCULATE THE SQUARE ROOT OF GAM3 FOR THE SHARP
C     EXPANSION CORNER
C
  130  D1=SQRT(GAM3)
       RETURN
C
C      CALCULATE GAMMM=1.0 FOR THE INTERNAL ENERGY EQUATION R.M.S
C     R,M,S
C
  140  D1=GAMMA-1.0
       RETURN
C
C     CALCULATE THE TERM FOR THE ARTIFICIAL CONDUCTIVITY
C
  150 D1=GAM1*RG/RKMU
       RETURN
       END
```

```
      SUBROUTINE INTER
C
C     ***********************************************************
C
C     THIS SUBROUTINE CALCULATES THE INTERIOR MESH POINTS
C
C     ***********************************************************
C
C
      DIMENSION KSI(1000),XP(1000),XP1(50,20),YP1(50,20)
      DIMENSION TIME(1502),PRES1(1502),PRES2(1502),PRES3(1502)
      COMMON/ONESID/UD(4),VD(4),PD(4),ROD(4)
      COMMON/SOLUTN/U(50,20,2),V(50,20,2),P(50,20,2),
     $ RO(50,20,2)
      COMMON/CNTRLC/LMAX,MMAX,NMAX,TCONV,FDT,GAMMA,RGAS,GAM1,
     $ GAM2,GAM3,L1,L2,M1,M2,DX,DY,DT,ICHAR,N1D,IERR,IUI,IUO,
     $ DXR,DYR,RSTAR,N,N1,N3,RSTARS,G,PC,TC,LC,PLOW,ROLOW,RG,
     $ DXX,XL1
      COMMON/GEMTRL/XW(50),YW(50),NXNY(50),LT,XT,XI,XE,NDIM,
     $ XWI(50),YWI(50)
      COMMON/BCC/PT(20),TT(20),PE(20),IEXTRA,TW(20),ISUPER,
     $ UI(20),VI(20),PI(20),ROI(20),IEX
      COMMON/AV/CAV,NST,LSS,XMU,XLA,RKMU,XRO,QUT(50,20),
     $ QVT(50,20),QPT(50,20),QROT(50,20)
      COMMON/RV/CMU,CLA,CK,EMU,ELA,EK,CHECK,ITM,TML
      REAL MN3,NXNY,LC
      DP=0.0
      ATERM=0.
      IF(ICHAR.NE.1) GO TO 40
C
C     COMPUTE THE TENTATIVE SOLUTION AT T+DT
C
      DO 30 L=2,L1
      W=SQRT(1.+NXNY(L)*NXNY(L))
      DO 30 M=1,M1
      CALL MAP(1,L,M,AL,BE,DE,LD1,AL1,BE1,DE1)
      UB=U(L,M,N1)
      VB=V(L,M,N1)
      PB=P(L,M,N1)
      ROB=RO(L,M,N1)
C
      CALL EOS(1,PB,ROB,T,ASB,D2,D3)
C
      U1B=W*UB+DP
      IF(M.NE.1) GO TO 10
C
C     COMPUTATION OF THE MIDPLANE MESH POINTS
C
C
C     COMPUTATION OF THE DERIVATIVES WITH RESPECT TO X
```

```
C
      DUDX=(UB-U(L-1,M,N1))*DXR
      DPDX=(PB-P(L-1,M,N1))*DXR
      DRODX=(ROB-RO(L-1,M,N1))*DXR
C
C     COMPUTAION OF THE DERIVATIVES WITH RESPECT TO X
C
      DVDY=(4.0*V(L,2,N1)-V(L,3,N1))*.5*DYR
      V(L,M,N3)=0.0
      URHS=-U1B*DUDX-W*DPDX/ROB+QUT(L,M)
      RORHS=-U1B*DRODX-ROB*W*DUDX-FLOAT(1+NDIM)*ROB*DVDY*BE+
     $ QROT(L,M)
      PRHS=-U1B*DPDX+ASB*(RORHS+U1B*DRODX)+QPT(L,M)
      GO TO 20
   10 IF (NDIM.EQ.1) ATERM=ROB*VB/(FLOAT(M-1)*DY/BE)
C
C     COMPUTAION OF THE DERIVATIVES WITH RESPECT TO X
C
      UVB=UB*AL+VB*BE+DE
      DUDX=(UB-U(L-1,M,N1))*DXR
      DVDX=(VB-V(L-1,M,N1))*DXR
      DPDX=(PB-P(L-1,M,N1))*DXR
      DRODX=(ROB-RO(L-1,M,N1))*DXR
      DUDY=(UB-U(L,M-1,N1))*DYR
      DVDY=(VB-V(L,M-1,N1))*DYR
      DPDY=(PB-P(L,M-1,N1))*DYR
      DRODY=(ROB-RO(L,M-1,N1))*DYR
      URHS=-U1B*DUDX-UVB*DUDY-(W*DPDX+AL*DPDY)/ROB+QUT(L,M)
      VRHS=-U1B*DVDX-UVB*DVDY-BE*DPDY/ROB+QVT(L,M)
      RORHS=-U1B*DRODX-UVB*DRODY-ROB*(W*DUDX+AL*DUDY+BE*DVDY)
     $ -ATERM+QROT(L,M)
      PRHS=-U1B*DPDX-UVB*DPDY+ASB*(RORHS+U1B*DRODX+UVB*DRODY)
     $ + QPT(L,M)
      V(L,M,N3)=V(L,M,N1)+VRHS*DT
   20 U(L,M,N3)=U(L,M,N1)+URHS*DT
      P(L,M,N3)=P(L,M,N1)+PRHS*DT
      RO(L,M,N3)=RO(L,M,N1)+RORHS*DT
      IF(P(L,M,N3).LE.0.0) P(L,M,N3)=PLOW
      IF(RO(L,M,N3).LE.0.0) RO(L,M,N3)=ROLOW
   30 CONTINUE
      RETURN
C
C     COMPUTE THE FINAL SOLUTION AT T+DT
C
   40 DO 70 L=2,L1
      W=SQRT(1.+NXNY(L)*NXNY(L))
      DO 70 M=1,M1
      CALL MAP(1,L,M,AL,BE,DE,LD1,AL1,BE1,DE1)
      UB=U(L,M,N3)
      VB=V(L,M,N3)
```

```
      PB=P(L,M,N3)
      ROB=RO(L,M,N3)
C
      CALL EOS(1,PB,ROB,T,ASB,D2,D3)
C
      U1B=W*UB+DP
      IF(M.NE.1) GO TO 50
      DUDX=(U(L+1,M,N3)-UB)*DXR
      DVDX=(V(L+1,M,N3)-VB)*DXR
      DPDX=(P(L+1,M,N3)-PB)*DXR
      DRODX=(RO(L+1,M,N3)-ROB)*DXR
      DVDY=(4.0*V(L,2,N3)-V(L,3,N3))*.5*DYR
      V(L,M,N3)=0.0
      URHS=-U1B*DUDX-W*DPDX/ROB+QUT(L,M)
      RORHS=-U1B*DRODX-ROB*W*DUDX-FLOAT(1+NDIM)*ROB*DVDY*BE+
     $  QROT(L,M)
      PRHS=-U1B*DPDX+ASB*(RORHS+U1B*DRODX)+QPT(L,M)
      GO TO 60
C
   50 IF (NDIM.EQ.1) ATERM=ROB*VB/(FLOAT(M-1)*DY/BE)
      UVB=UB*AL+VB*BE+DE
      DUDX=(U(L+1,M,N3)-UB)*DXR
      DPDX=(P(L+1,M,N3)-PB)*DXR
      DVDX=(V(L+1,M,N3)-VB)*DXR
      DRODX=(RO(L+1,M,N3)-ROB)*DXR
C
      DUDY=(U(L,M+1,N3)-UB)*DYR
      DVDY=(V(L,M+1,N3)-VB)*DYR
      DPDY=(P(L,M+1,N3)-PB)*DYR
      DRODY=(RO(L,M+1,N3)-ROB)*DYR
C
      URHS=-U1B*DUDX-UVB*DUDY-(W*DPDX+AL*DPDY)/ROB+QUT(L,M)
      VRHS=-U1B*DVDX-UVB*DVDY-BE*DPDY/ROB+QVT(L,M)
      RORHS=-U1B*DRODX-UVB*DRODY-ROB*(W*DUDX+AL*DUDY+BE*DVDY)
     $ -ATERM+QROT(L,M)
      PRHS=-U1B*DPDX-UVB*DPDY+ASB*(RORHS+U1B*DRODX+UVB*DRODY)
     $ + QPT(L,M)
      V(L,M,N3)=(V(L,M,N1)+V(L,M,N3)+VRHS*DT)*.5
   60 U(L,M,N3)=(U(L,M,N1)+U(L,M,N3)+URHS*DT)*.5
      P(L,M,N3)=(P(L,M,N1)+P(L,M,N3)+PRHS*DT)*.5
      RO(L,M,N3)=(RO(L,M,N1)+RO(L,M,N3)+RORHS*DT)*.5
      IF(P(L,M,N3).LE.0.0) P(L,M,N3)=PLOW
      IF(RO(L,M,N3).LE.0.0) RO(L,M,N3)=ROLOW
   70 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INLET
C
C     ************************************************************
C
C     THIS SUBROUTINE CALCULATES THE INLET MESH POINTS
C
C     ************************************************************
C
      DIMENSION KSI(1000),XP(1000),XP1(50,20),YP1(50,20)
      DIMENSION TIME(1502),PRES1(1502),PRES2(1502),PRES3(1502)
      COMMON/ONESID/UD(4),VD(4),PD(4),ROD(4)
      COMMON/SOLUTN/U(50,20,2),V(50,20,2),P(50,20,2),
     $ RO(50,20,2)
      COMMON/CNTRLC/LMAX,MMAX,NMAX,TCONV,FDT,GAMMA,RGAS,GAM1,
     $ GAM2,GAM3,L1,L2,M1,M2,DX,DY,DT,ICHAR,N1D,IERR,IUI,IUO,
     $ DXR,DYR,RSTAR,N,N1,N3,RSTARS,G,PC,TC,LC,PLOW,ROLOW,RG,
     $ DXX,XE1
      COMMON/GEMTRL/XW(50),YW(50),NXNY(50),LT,XT,XI,XE,NDIM,
     $ XWI(50),YWI(50)
      COMMON/BCC/PT(20),TT(20),PE(20),IEXTRA,TW(20),ISUPER,
     $ UI(20),VI(20),PI(20),ROI(20),IEX
      COMMON/AV/CAV,NST,LSS,XMU,XLA,RKMU,XRO,QUT(50,20),
     $ QVT(50,20),QPT(50,20),QROT(50,20)
      COMMON/EV/CMU,CLA,CK,EMU,ELA,EK,CHECK,ITM,TML
      REAL MN3,NXNY,LC
C
      L=1
      XI=0.0
      X3=XI
      ATERM2=0.
      ATERM3=0.
      ATERM2=0.0
      ATERM3=0.0
      DO 240 M=1,MMAX
      CALL MAP (2,1,M,AL,BE,DE,2,AL1,BE1,DE1)
      U2=U(L,M,N1)
      CALL EOS(1,P(L,M,N1),RO(L,M,N1),T,AS,D2,D3)
      A2=SQRT(AS)
      IF (ICHAR.NE.1) GO TO 40
      IF(ISUPER.EQ.-1) GO TO 30
      U(L,M,N3)=U2
      V(L,M,N3)=V(L,M,N1)
   30 A3=A2
C
C     CALCULATE THE PROPERTY INTERPOLATING POLYNOMIAL
C     COEFFICIENTS
C
   40 BU=(U(L+1,M,N1)-U(L,M,N1))*DXR
      BV=(V(L+1,M,N1)-V(L,M,N1))*DXR
      BP=(P(L+1,M,N1)-P(L,M,N1))*DXR
```

```
      BRO=(RO(L+1,M,N1)-RO(L,M,N1))*DXR
      BAL=(AL1-AL)*DXR
      BBE=(BE1-BE)*DXR
      CU=U(L,M,N1)-BU*X3
      CV=V(L,M,N1)-BV*X3
      CP=P(L,M,N1)-BP*X3
      CRO=RO(L,M,N1)-BRO*X3
      CAL=AL-BAL*X3
      CBE=BE-BBE*X3
C
C     CALCULATE THE CROSS DERIVATIVE INTERPOLATING POLYNOMIAL
C     COEFFICIENTS
C
      IF(M.EQ.1)  GO TO 50
      DU=(U(L+1,M,N1)-U(L+1,M-1,N1))*DYR
      DV=(V(L+1,M,N1)-V(L+1,M-1,N1))*DYR
      DP=(P(L+1,M,N1)-P(L+1,M-1,N1))*DYR
      DRO=(RO(L+1,M,N1)-RO(L+1,M-1,N1))*DYR
      DU1=(U(L,M,N1)-U(L,M-1,N1))*DYR
      DV1=(V(L,M,N1)-V(L,M-1,N1))*DYR
      DP1=(P(L,M,N1)-P(L,M-1,N1))*DYR
      DRO1=(RO(L,M,N1)-RO(L,M-1,N1))*DYR
      GO TO 70
   50 DU=0.
      DV=(4.0*V(2,2,N1)-V(2,3,N1))*0.5*DYR
      DP=0.
      DRO=0.0
      DU1=0.0
      DV1=(4.0*V(L,2,N1)-V(L,3,N1))*0.5*DYR
      DP1=0.0
      DRO1=0.0
   70 BDU=(DU-DU1)*DXR
      BDV=(DV-DV1)*DXR
      BDP=(DP-DP1)*DXR
      BDRO=(DRO-DRO1)*DXR
      CDU=DU1-BDU*X3
      CDV=DV1-BDV*X3
      CDP=DP1-BDP*X3
      CDRO=DRO1-BDRO*X3
C
C     CALCULATE X2
C
      IF(ICHAR.EQ.1) GO TO 80
      CALL EOS(1,P(L,M,N3),RO(L,M,N3),T,AS,D2,D3)
      A3=SQRT(AS)
   80 DO 90 IL=1,2
      X2=X3-(U(L,M,N3)-A3+U2-A2)*0.5*DT
C
C     INTERPOLATE FOR THE PROPERTIES
C
```

```
      U2=BU*X2+CU
      P2=BP*X2+CP
      RO2=BRO*X2+CRO
      CALL  EOS(1,P2,RO2,T,AS,D2,D3)
      A2=SQRT(AS)
   90 CONTINUE
      V2=BV*X2+CV
      AL2=BAL*X2+CAL
      BE2=BBE*X2+CBE
      UV2=U2*AL2+V2*BE2
C
C     INTERPOLATE FOR THE CROSS DERIVATIVES
C
      DU2=BDU*X2+CDU
      DV2=BDV*X2+CDV
      DP2=BDP*X2+CDP
      DRO2=BDRO*X2+CDRO
C
C     CALCULATE THE PSI TERMS
C
      IF(M.EQ.1)  GO TO 100
      ATERM2=RO2*V2/(DY*FLOAT(M-1)/BE2)
      GO TO 110
  100 ATERM2=RO2*BE2*DV2
  110 PSI12=-UV2*DRO2-RO2*AL2*DU2-RO2*BE2*DV2-ATERM2
      PSI22=-UV2*DU2-AL2*DP2/RO2
      PSI42=-UV2*DP2+A2*A2*UV2*DRO2
      IF(ICHAR.EQ.1)  GO TO 170
C
C     CALCULATE THE CROSS DERIVATIVES AT THE SOLUTION POINT
C
      IF(M.EQ.1)  GO TO 120
      IF  (M.EQ.MMAX)  GO TO 130
      DU3=(U(L,M+1,N3)-U(L,M,N3))*DYR
      DV3=(V(L,M+1,N3)-V(L,M,N3))*DYR
      DP3=(P(L,M+1,N3)-P(L,M,N3))*DYR
      DRO3=(RO(L,M+1,N3)-RO(L,M,N3))*DYR
      GO TO 140
  120 DU3=0.0
      DV3=(4.0*V(L,2,N3)-V(L,3,N3))*0.5*DYR
      DP3=0.0
      DRO3=0.0
      GO TO 140
  130 DU3=(U(L,MMAX,N3)-U(L,M1,N3))*DYR
      DV3=(V(L,MMAX,N3)-V(L,M1,N3))*DYR
      DP3=(P(L,MMAX,N3)-P(L,M1,N3))*DYR
      DRO3=(RO(L,MMAX,N3)-RO(L,M1,N3))*DYR
C
C     CALCULATE THE PSI TERM AT THE SOLUTION POINT
C
```

```
  140 IF (M.EQ.1) GO TO 150
      ATERM3=RO(L,M,N3)*V(L,M,N3)/(DY*FLOAT(M-1)/BE)
      GO TO 160
  150 ATERM3=RO(L,M,N3)*BE*DV3
  160 UV3=U(L,M,N3)*AL+V(L,M,N3)*BE
      PSI13=-UV3*DRO3-RO(L,M,N3)*AL*DU3-RO(L,M,N3)*BE*DV3-
     $ ATERM3
      PSI23=-UV3*DU3-AL*DP3/RO(L,M,N3)
      PSI43=-UV3*DP3+A3*A3*UV3*DRO3
      GO TO 180
  170 PSI23=PSI22
      PSI43=PSI42
      PSI13=PSI12
  180 PSI1B=0.5*(PSI12+PSI13)
      PSI2B=0.5*(PSI22+PSI23)
      PSI4B=0.5*(PSI42+PSI43)
C
C     SOLVE THE COMPATIBILITY EQUATION FOR P
C
      IF(ISUPER.EQ.0) GO TO 190
      ROAB=0.5*(RO2*A2+RO(L,M,N3)*A3)
      AB=0.5*(A2+A3)
      P(L,M,N3)=P2+ROAB*(U(L,M,N3)-U2)+(PSI4B-ROAB*PSI2B+AB*
     $ AB*PSI1B)*DT
      U(1,M,N3)=0.0
      V(1,M,N3)=0.0
      RO(1,M,N3)=RO(2,M,N3)
      GO TO 240
C
C     SOLVE THE COMPATIBILITY EQUATIONS FOR U,V,P,AND RO
C
  190 MN3=SQRT(U(L,M,N3)*U(L,M,N3)+V(L,M,N3)*V(L,M,N3))/A3
      CALL EOS(2,P2,RO2,T2,AS,D2,D3)
      DO 220 ITER=1,20
      CALL EOS (6,P(L,M,N3),RO(L,M,N3),T3,PT(M),TT(M),MN3)
      PB=(P2+P(L,M,N3))*0.5
      TB=(T2+T3)*0.5
      CALL EOS (6,PB,ROB,TB,AS,D2,D3)
      U(L,M,N3)=U2+DT*PSI2B+(P(L,M,N3)-P2-(PSI4B+AS*PSI1B)*DT)
     $ /(ROB*SQRT(AS))
      OMN3=MN3
      CALL EOS(7,PB,ROB,T3,AS,D2,D3)
      MN3=SQRT((U(L,M,N3)**2+V(L,M,N3)**2)/AS)
      IF(OMN3.NE.0.0) GO TO 210
      IF (ABS(MN3-OMN3).LE.0.0001) GO TO 230
      GO TO 220
  210 IF(ABS((MN3-OMN3)/OMN3).LE.0.001) GO TO 230
  220 CONTINUE
C
  230 CALL EOS (4,P(L,M,N3),RO(L,M,N3),T3,AS,D2,D3)
```

```
C
  240 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EXIT
C
C     ************************************************************
C
C     THIS SUBROUTINE CALCULATES THE BOUNDARY MESH POINTS AT
C     THE EXIT PLANE
C
C     ************************************************************
C
C
      DIMENSION KSI(1000),XP(1000),XP1(50,20),YP1(50,20)
      DIMENSION TIME(1502),PRES1(1502),PRES2(1502),PRES3(1502)
      COMMON/ONESID/UD(4),VD(4),PD(4),ROD(4)
      COMMON/SOLUTN/U(50,20,2),V(50,20,2),P(50,20,2),
     $ RO(50,20,2)
      COMMON/CNTRLC/LMAX,MMAX,NMAX,TCONV,FDT,GAMMA,RGAS,GAM1,
     $ GAM2,GAM3,L1,L2,M1,M2,DX,DY,DT,ICHAR,N1D,IERR,IUI,IUO,
     $ DXR,DYR,RSTAR,N,N1,N3,RSTARS,G,PC,TC,LC,PLOW,ROLOW,RG,
     $ DXX,XE1
      COMMON/GEMTRL/XW(50),YW(50),NXNY(50),LT,XT,XI,XE,NDIM,
     $ XWI(50),YWI(50)
      COMMON/BCC/PT(20),TT(20),PE(20),IEXTRA,TW(20),ISUPER,
     $ UI(20),VI(20),PI(20),ROI(20),IEX
      COMMON/AV/CAV,NST,LSS,XMU,XLA,RKMU,XRO,QUT(50,20),
     $ QVT(50,20),QPT(50,20),QROT(50,20)
      COMMON/RV/CMU,CLA,CK,EMU,ELA,EK,CHECK,ITM,TML
      REAL MN3,NXNY,LC
      ME=MMAX
      X3=XE1
      ATERM2=0.0
      ATERM3=0.0
      DO 180 M=1,ME
      IF(IEXTRA.EQ.1) GO TO 10
      CALL EOS(1,P(LMAX,M,N1),RO(LMAX,M,N1),T,AS,D2,D3)
      A1=SQRT(AS)
      IF (IEXTRA.EQ.2) GO TO 20
      Q=SQRT(U(LMAX,M,N1)*U(LMAX,M,N1)+V(LMAX,M,N1)*
     $ V(LMAX,M,N1))
      IF(Q/A1.LT.1.0) GO TO 20
   10 U(LMAX,M,N3)=U(L1,M,N3)+FLOAT(IEX)*(U(L1,M,N3)-
     $ U(L2,M,N3))
      V(LMAX,M,N3)=V(L1,M,N3)+FLOAT(IEX)*(V(L1,M,N3)-
     $ V(L2,M,N3))
      P(LMAX,M,N3)=P(L1,M,N3)+FLOAT(IEX)*(P(L1,M,N3)-
     $ P(L2,M,N3))
      RO(LMAX,M,N3)=RO(L1,M,N3)+FLOAT(IEX)*(RO(L1,M,N3)-
     $ RO(L2,M,N3))
      GO TO 180
   20 CALL MAP(2,LMAX,M,AL,BE,DE,L1,AL1,BE1,DE1)
      U1=U(LMAX,M,N1)
```

```
      U2=U1
      A2=A1
      IF(ICHAR.NE.1)  GO TO 30
      U(LMAX,M,N3)=U1
      RO(LMAX,M,N3)=RO(LMAX,M,N1)
      A3=A1
C
C     CALCULATE THE PROPERTY INTERPOLATION POLYNOMIAL
C     COEFFICIENTS
C
   30 BU=(U(LMAX,M,N1)-U(L1,M,N1))*DXR
      BV=(V(LMAX,M,N1)-V(L1,M,N1))*DXR
      BP=(P(LMAX,M,N1)-P(L1,M,N1))*DXR
      BRO=(RO(LMAX,M,N1)-RO(L1,M,N1))*DXR
      BAL=(AL-AL1)*DXR
      BBE=(BE-BE1)*DXR
      BDE=(DE-DE1)*DXR
      CU=U(LMAX,M,N1)-BU*X3
      CV=V(LMAX,M,N1)-BV*X3
      CP=P(LMAX,M,N1)-BP*X3
      CRO=RO(LMAX,M,N1)-BRO*X3
      CAL=AL-BAL*X3
      CBE=BE-BBE*X3
      CDE=DE-BDE*X3
C
C     CALCULATE THE CROSS DERIVATIVE INTERPOLATING POLYNOMIAL
C     COEFFICIENTS
C
      IF(M.EQ.1)  GO TO 40
      DU=(U(LMAX,M,N1)-U(LMAX,M-1,N1))*DYR
      DV=(V(LMAX,M,N1)-V(LMAX,M-1,N1))*DYR
      DP=(P(LMAX,M,N1)-P(LMAX,M-1,N1))*DYR
      DRO=(RO(LMAX,M,N1)-RO(LMAX,M-1,N1))*DYR
      DU1=(U(L1,M,N1)-U(L1,M-1,N1))*DYR
      DV1=(V(L1,M,N1)-V(L1,M-1,N1))*DYR
      DP1=(P(L1,M,N1)-P(L1,M-1,N1))*DYR
      DRO1=(RO(L1,M,N1)-RO(L1,M-1,N1))*DYR
      GO TO 60
   40 DU=0.0
      DV=(4.0*V(LMAX,2,N1)-V(LMAX,3,N1))*0.5*DYR
      DP=0.0
      DRO=0.0
      DU1=0.0
      DV1=(4.0*V(L1,2,N1)-V(L1,3,N1))*0.5*DYR
      DP1=0.0
      DRO1=0.0
   60 BDU=(DU-DU1)*DXR
      BDV=(DV-DV1)*DXR
      BDP=(DP-DP1)*DXR
      BDRO=(DRO-DRO1)*DXR
```

```
      CDU=DU-BDU*X3
      CDV=DV-BDV*X3
      CDP=DP-BDP*X3
      CDRO=DRO-BDRO*X3
C
C     CALCULATE X1 AND X2
C
      IF(ICHAR.EQ.1) GO TO 70
      CALL EOS(1,P(LMAX,M,N3),RO(LMAX,M,N3),T,AS,D2,D3)
      A3=SQRT(AS)
   70 DO 80 IL=1,2
      X1=X3-(U(LMAX,M,N3)+U1)*0.5*DT
      X2=X3-(U(LMAX,M,N3)+A3+U2+A2)*0.5*DT
C
C     INTERPOLATE FOR THE PROPERTIES
C
      U1=BU*X1+CU
      U2=BU*X2+CU
      P2=BP*X2+CP
      RO2=BRO*X2+CRO
      CALL EOS(1,P2,RO2,T2,AS,D2,D3)
      A2=SQRT(AS)
   80 CONTINUE
      V1=BV*X1+CV
      P1=BP*X1+CP
      RO1=BRO*X1+CRO
      AL1=BAL*X1+CAL
      BE1=BBE*X1+CBE
      DE1=BDE*X1+CDE
      UV1=U1*AL1+V1*BE1+DE1
      CALL EOS(1,P1,RO1,T1,AS,D2,D3)
      A1=SQRT(AS)
      V2=BV*X2+CV
      AL2=BAL*X2+CAL
      BE2=BBE*X2+CBE
      DE2=BDE*X2+CDE
      UV2=U2*AL2+V2*BE2+DE2
C
C     INTERPOLATE FOR THE CROSS DERIVATIVES
      DV1=BDV*X1+CDV
      DP1=BDP*X1+CDP
      DRO1=BDRO*X1+CDRO
      DU2=BDU*X2+CDU
      DV2=BDV*X2+CDV
      DP2=BDP*X2+CDP
      DRO2=BDRO*X2+CDRO
C
C     CALCULATE THE PSI TERMS
C
      IF(NDIM.EQ.0) GO TO 100
```

```
      IF(M.EQ.1)  GO TO 90
      ATERM2=RO2*V2/(DY*FLOAT(M-1)/BE2)
      GO TO 100
   90 ATERM2=RO2*BE2*DV2
  100 PSI31=-UV1*DV1-(DP1/RO1)*BE1
      PSI41=-UV1*DP1+A1*A1*UV1*DRO1
      PSI12=-UV2*DRO2-RO2*DV2*BE2-ATERM2-RO2*AL2*DU2
      PSI22=-UV2*DU2-AL2*DP2/RO2
      PSI42=-UV2*DP2+A2*A2*UV2*DRO2
      IF(ICHAR.EQ.1) GO TO 160
C
C     CALCULATE THE CROSS DERIVATIVES AT THE SOLUTION POINT
C
      IF(M.EQ.1)  GO TO 110
      IF(M.EQ.ME) GO TO 120
      DU3=(U(LMAX,M+1,N3)-U(LMAX,M,N3))*DYR
      DV3=(V(LMAX,M+1,N3)-V(LMAX,M,N3))*DYR
      DP3=(P(LMAX,M+1,N3)-P(LMAX,M,N3))*DYR
      DRO3=(RO(LMAX,M+1,N3)-RO(LMAX,M,N3))*DYR
      GO TO 130
  110 DU3=0.0
      DV3=(4.0*V(LMAX,2,N3)-V(LMAX,3,N3))*0.5*DYR
      DP3=0.0
      DRO3=0.0
      GO TO 130
  120 MB=ME-1
      DU3=(U(LMAX,ME,N3)-U(LMAX,MB,N3))*DYR
      DV3=(V(LMAX,ME,N3)-V(LMAX,MB,N3))*DYR
      DP3=(P(LMAX,ME,N3)-P(LMAX,MB,N3))*DYR
      DRO3=(RO(LMAX,ME,N3)-RO(LMAX,MB,N3))*DYR
C
C     CALCULATE THE PSI TERMS AT THE SOLUTION POINT
C
  130 IF (NDIM.EQ.0) GO TO 150
      IF(M.EQ.1)  GO TO 140
      ATERM3=RO(LMAX,M,N3)*V(LMAX,M,N3)/(DY*FLOAT(M-1)/BE)
      GO TO 150
  140 ATERM3=RO(LMAX,1,N3)*BE*DV3
  150 UV3=V(LMAX,M,N3)*BE+U(LMAX,M,N3)*AL+DE
      PSI13=-UV3*DRO3-RO(LMAX,M,N3)*(DV3*BE+AL*DU3)-ATERM3
      PSI23=-UV3*DU3-AL*DP3/RO(LMAX,M,N3)
      PSI33=-UV3*DV3-(DP3/RO(LMAX,M,N3))*BE
      PSI43=-UV3*DP3+A3*A3*UV3*DRO3
      PSI31B=(PSI31+PSI33)*0.5+QVT(LMAX,M)
      PSI41B=(PSI41+PSI43)*0.5+QPT(LMAX,M)
      PSI12B=(PSI12+PSI13)*0.5+QROT(LMAX,M)
      PSI22B=(PSI22+PSI23)*0.5+QUT(LMAX,M)
      PSI42B=(PSI42+PSI43)*0.5+QPT(LMAX,M)
      GO TO 170
  160 PSI31B=PSI31+QVT(LMAX,M)
```

```
      PSI41B=PSI41+QPT(LMAX,M)
      PSI12B=PSI12+QROT(LMAX,M)
      PSI22B=PSI22+QUT(LMAX,M)
      PSI42B=PSI42+QPT(LMAX,M)
C
C     SOLVE THE COMPATIBILITY EQUATIONS FOR U,V,P, AND RO
C
  170 P(LMAX,M,N3)=PE(M)
      AB=0.5*(A2+A3)
      ROB=0.5*(RO2+RO(LMAX,M,N3))
      RO(LMAX,M,N3)=RO1+2.0*(P(LMAX,M,N3)-P1-DT*PSI41B)/
     $ (A3*A3+A1*A1)
      IF(RO(LMAX,M,N3).LE.0.0) RO(LMAX,M,N3)=ROLOW
      U(LMAX,M,N3)=U2+((PSI42B+ROB*AB*PSI22B+AB*AB*PSI12B)*DT
     $ -(P(LMAX,M,N3)-P2))/(ROB*AB)
      V(LMAX,M,N3)=V1+DT*PSI31B
C
C     CHECK FOR INFLOW AND IF SO SET INFLOW BOUNDARY
C     CONDITIONS
C
      IF(U(LMAX,M,N3).GE.0.0) GO TO 180
      V(LMAX,M,N3)=0.0
      RO(LMAX,M,N3)=0.5*(RO(LMAX,1,N1)+RO(LMAX,MMAX,N1))
  180 CONTINUE
      V(LMAX,MMAX,N3)=-U(LMAX,MMAX,N3)*NXNY(LMAX)
      RETURN
      END
```

```
      SUBROUTINE SHOOTH
C
C     ************************************************************
C
C     THIS SUBROUTINE IS USED TO SMOOTH THE FLOW VARIABLES
C
C
C     ************************************************************
C
      DIMENSION KSI(1000),XP(1000),XP1(50,20),YP1(50,20)
      DIMENSION TIME(1502),PRES1(1502),PRES2(1502),PRES3(1502)
      COMMON/ONESID/UD(4),VD(4),PD(4),ROD(4)
      COMMON/SOLUTN/U(50,20,2),V(50,20,2),P(50,20,2),
     $ RO(50,20,2)
      COMMON/CNTRLC/LMAX,MMAX,NMAX,TCONV,FDT,GAMMA,RGAS,GAM1,
     $ GAM2,GAM3,L1,L2,M1,M2,DX,DY,DT,ICHAR,N1D,IERR,IUI,IUO,
     $ DXR,DYR,RSTAR,N,N1,N3,RSTARS,G,PC,TC,LC,PLOW,ROLOW,RG,
     $ DXX,XE1
      COMMON/GEMTRL/XW(50),YW(50),NXNY(50),L1,XT,AI,XE,NDIM,
     $ XWI(50),YWI(50)
      COMMON/BCC/PT(20),TT(20),PE(20),IEXTRA,TW(20),ISUPER,
     $ UI(20),VI(20),PI(20),ROI(20),IEX
      COMMON/AV/CAV,NST,LSS,XMU,XLA,RKMU,XRO,QUT(50,20),
     $ QVT(50,20),QPT(50,20),QROT(50,20)
      COMMON/RV/CMU,CLA,CK,EMU,ELA,EK,CHECK,ITM,TML
      REAL MN3,NXNY,LC,KSI
C
      SMP=0.95
      SMP4=0.25*(1.0-SMP)
      KMAX=MMAX
      J=1
      K1=KMAX-1
      MS=J+1
      DO 20 L=2,L1
      U(L,KMAX,N3)=SMP4*(U(L-1,KMAX,N3)+U(L+1,KMAX,N3)+2.0*
     $ U(L,K1,N3))+SMP*U(L,KMAX,N3)
      V(L,KMAX,N3)=-U(L,KMAX,N3)*NXNY(L)
      P(L,KMAX,N3)=SMP4*(P(L-1,KMAX,N3)+P(L+1,KMAX,N3)+2.0*
     $ P(L,K1,N3))+SMP*P(L,KMAX,N3)
      RO(L,KMAX,N3)=SMP4*(RO(L-1,KMAX,N3)+RO(L+1,KMAX,N3)+2.0
     $ *RO(L,K1,N3))+SMP*RO(L,KMAX,N3)
      U(L,J,N3)=SMP4*(U(L-1,J,N3)+U(L+1,J,N3)+2.0*U(L,MS,N3))
     $ +SMP*U(L,J,N3)
      V(L,J,N3)=SMP4*(V(L-1,J,N3)+V(L+1,J,N3)+2.0*V(L,MS,N3))
     $ +SMP*V(L,J,N3)
      P(L,J,N3)=SMP4*(P(L-1,J,N3)+P(L+1,J,N3)+2.0*P(L,MS,N3))
     $ +SMP*P(L,J,N3)
      RO(L,J,N3)=SMP4*(RO(L-1,J,N3)+RO(L+1,J,N3)+2.0*
     $ RO(L,MS,N3))+SMP*RO(L,J,N3)
      DO 20 M=2,M1
```

```
      U(L,M,N3)=SMP4*(U(L-1,M,N3)+U(L+1,M,N3)+U(L,M-1,N3)+
     $ U(L,M+1,N3))+SMP*U(L,M,N3)
      V(L,M,N3)=SMP4*(V(L-1,M,N3)+V(L+1,M,N3)+V(L,M-1,N3)+
     $ V(L,M+1,N3))+SMP*V(L,M,N3)
      P(L,M,N3)=SMP4*(P(I-1,M,N3)+P(L+1,M,N3)+P(L,M-1,N3)+
     $ P(L,M+1,N3))+SMP*P(L,M,N3)
      RO(L,M,N3)=SMP4*(RO(L-1,M,N3)+RO(L+1,M,N3)+RO(L,M-1,N3)
     $ +RO(L,M+1,N3))+SMP*RC(L,M,N3)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE YWALL
C
C     ***********************************************************
C
C     TIS SUBROUTINE CALCULATES THE BOUNDARY MESH POINTS AT
C     THE WALL
C
C     ***********************************************************
C
      DIMENSION KS1(1000),XP(1000),XP1(50,20),YP1(50,20)
      DIMENSION TIME(1502),PRES1(1502),PRES2(1502),PRES3(1502)
      COMMON/ONESID/UD(4),VD(4),PD(4),ROD(4)
      COMMON/SOLUTN/U(50,20,2),V(50,20,2),P(50,20,2),
     $ RO(50,20,2)
      COMMON/CNTRLC/LMAX,MMAX,NMAX,TCONV,FDT,GAMMA,RGAS,GAM1,
     $ GAM2,GAM3,L1,L2,M1,M2,DX,DY,DT,ICHAR,N1D,IERR,IUI,IUO,
     $ DXR,DYR,RSTAR,N,N1,N3,RSTARS,G,PC,TC,LC,PLOW,ROLOW,RG,
     $ DXX,XE1
      COMMON/GEMTRL/XW(50),YW(50),NXNY(50),LT,XT,XI,XE,NDIM,
     $ XWI(50),YWI(50)
      COMMON/BCC/PT(20),TT(20),PE(20),IEXTRA,TW(20),ISUPER,
     $ UI(20),VI(20),PI(20),ROI(20),IEX
      COMMON/AV/CAV,NST,LSS,XMU,XLA,RKMU,XRO,QUT(50,20),
     $ QVT(50,20),QPT(50,20),QROT(50,20)
      COMMON/RV/CMU,CLA,CK,EMU,ELA,EK,CHECK,ITM,TML
      REAL MN3,NXNY,LC
      ATERM2=0.
      ATERM3=0.
      LJET=23
      JFLAG=0
      IF ( N.EQ.1.AND.JFLAG.NE.0) DELY=0.0001*YW(LJET-1)
      SIGN=1.0
      YJ=1.0
      DYS=SIGN*DYR
      DO 510 L=2,L1
      W=SQRT(1.+NXNY(L)*NXNY(L))
      DXB=DXX*W
      DXR=DXB
      CALL MAP(1,L,MMAX,AL,BE,DE,LD1,AL1,BE1,DE1)
C
C     CALCULATE THE DEPENDENT VARIABLES FOR FREE-SLIP WALLS
C
      IF ( JFLAG.EQ.0) GO TO 120
      XWID=XWI(L)
      IF ( ICHAR.EQ.1)  GO TO 100
C
      IF (L.NE.LJET-2) GO TO 100
      U(L+1,MMAX,N3)=UD(3)
      V(L+1,MMAX,N3)=VD(3)
      P(L+1,MMAX,N3)=PD(3)
```

```
      RO(L+1,MMAX,N3)=ROD(3)
      GO TO 120
  100 IF (L.NE.LJET-1) GO TO 110
      IF ( ICHAR.EQ.1) UOLD=U(L,MMAX,N1)
      U(L,MMAX,N1)=UD(1)
      V(L,MMAX,N1)=VD(1)
      P(L,MMAX,N1)=PD(1)
      RO(L,MMAX,N1)=ROD(1)
      GO TO 120
  110 IF (L.NE.LJET) GO TO 120
      U(L-1,MMAX,N1)=UD(2)
      V(L-1,MMAX,N1)=VD(2)
      P(L-1,MMAX,N1)=PD(2)
      RO(L-1,MMAX,N1)=ROD(2)
C
  120 U1=U(L,MMAX,N1)
      V1=V(L,MMAX,N1)
      P1=P(L,MMAX,N1)
      RO1=RO(L,MMAX,N1)
      U2=U1
      V2=V1
      CALL EOS(1,P1,RO1,T,AS,D2,D3)
      A1=SQRT(AS)
      A2=A1
      IF(ICHAR.NE.1) GO TO 130
      U3=U1
      V3=V1
      P3=P1
      RO3=RO1
      A3=A1
      GO TO 140
  130 U3=U(L,MMAX,N3)
      V3=V(L,MMAX,N3)
      P3=P(L,MMAX,N3)
      RO3=RO(L,MMAX,N3)
      CALL EOS(1,P3,RO3,T,AS,D2,D3)
      A3=SQRT(AS)
C
C     CALCULATE THE PROPERTY INTERPOLATING POLYNOMIAL
C     COEFFICIENTS
C
  140 BU=(U1-U(L,M1,N1))*DYR
      BV=(V1-V(L,M1,N1))*DYR
      BP=(P1-P(L,M1,N1))*DYR
      BRO=(RO1-RO(L,M1,N1))*DYR
      CU=U1-BU*Y3
      CV=V1-BV*Y3
      CP=P1-BP*Y3
      CRO=RO1-BRO*Y3
C
```

```
C     CALCULATE THE CROSS DERIVATIVE INTERPOLATING POLYNOMIAL
C     COEFFICIENTS
C
      DU=(U1-U(L-1,MMAX,N1))*DXR
      DV=(V1-V(L-1,MMAX,N1))*DXR
      DP=(P1-P(L-1,MMAX,N1))*DXR
      DRO=(RO1-RO(L-1,MMAX,N1))*DXR
      DU1=(U(L,M1,N1)-U(L-1,M1,N1))*DXR
      DV1=(V(L,M1,N1)-V(L-1,M1,N1))*DXR
      DP1=(P(L,M1,N1)-P(L-1,M1,N1))*DXR
      DRO1=(RO(L,M1,N1)-RO(L-1,M1,N1))*DXR
      BDU=(DU-DU1)*DYR
      BDV=(DV-DV1)*DYR
      BDP=(DP-DP1)*DYR
      BDRO=(DRO-DRO1)*DYR
      CDU=DU-BDU*Y3
      CDV=DV-BDV*Y3
      CDP=DP-BDP*Y3
      CDRO=DRO-BDRO*Y3
C
C     CALCULATE Y2
C
      ALS=SQRT(AL*AL+BE*BE)
      UV3=U3*AL+V3*BE+DE
      DO 170 ILL=1,3
      UV2=U2*AL+V2*BE+DE
      Y2=Y3-(UV2+SIGN*ALS*A2+UV3+SIGN*ALS*A3)*DT*0.5
C
C     INTERPOLATE FOR PROPERTIES
C
      U2=BU*Y2+CU
      V2=BV*Y2+CV
      P2=BP*Y2+CP
      RO2=BRO*Y2+CRO
      CALL EOS(1,P2,RO2,T,AD,D2,D3)
      IF(AD.GT.0.0) GO TO 160
      IERR=1
      RETURN
  160 A2=SQRT(AD)
  170 CONTINUE
C
C     INTERPOLATE FOR THE CROSS DERIVATIVES
C
      DU1=DU
      DV1=DV
      DP1=DP
      DRO1=DRO
      DU2=BDU*Y2+CDU
      DV2=BDV*Y2+CDV
      DP2=BDP*Y2+CDP
```

```
      DRO2=BDRO*Y2+CDRO
      ATERM2=RO2*V2/(Y2/BE)
      PSI21=-U1*DU1-DP1/RO1
      PSI31=-U1*DV1
      PSI41=-U1*DP1+A1*A1*U1*DRO1
      PSI12=-U2*DRO2-RO2*DU2-ATERM2
      PSI22=-U2*DU2-DP2/RO2
      PSI32=-U2*DV2
      PSI42=-U2*DP2+A2*A2*U2*DRO2
      IF(ICHAR.EQ.1) GO TO 240
C
C     CALCULATE THE CROSS DERIVATIVES AT THE SOLUTION POINT
C
      IF ( JFLAG.EQ.0) GO TO 220
      IF( L.EQ.2) GO TO 220
      IF ( L.NE.LJET-1) GO TO 220
      GO TO 230
  220 DU3=(U(L+1,MMAX,N3)-U3)*DXR
      DV3=(V(L+1,MMAX,N3)-V3)*DXR
      DP3=(P(L+1,MMAX,N3)-P3)*DXR
      DRO3=(RO(L+1,MMAX,N3)-RO3)*DXR
      GO TO 240
  230 DU3=(U3-U(L-1,MMAX,N3))*DXR
      DV3=(V3-V(L-1,MMAX,N3))*DXR
      DP3=(P3-P(L-1,MMAX,N3))*DXR
      DRO3=(RO3-RO(L-1,MMAX,N3))*DXR
C
C     ENTER THE FREE-JET BOUNDARY ITERATION LOOP
C
  240 YWI(L)=YW(L)
      DO 390 NJ=1,10
      IF ( ICHAR.EQ.1) GO TO 340
      IF ( JFLAG.LE.0) GO TO 300
      IF ( L.LT.LJET) GO TO 300
      IF ( NJ.EQ.1) GO TO 290
      IF ( NJ.GT.2) GO TO 270
  250 YWOLD=YW(L)
      POLD=P(L,MMAX,N3)
      IF ( P(L,MMAX,N3).LT.PE(MMAX)) GO TO 260
      YW(L)=YW(L)+DELY
      GO TO 280
  260 YW(L)=YW(L)-DELY
      GO TO 280
  270 IF ( P(L,MMAX,N3).EQ.POLD) GO TO 250
      DYDP=(YW(L)-YWOLD)/(P(L,MMAX,N3)-POLD)
      YWNEW=YW(L)+DYDP*(PE(MMAX)-P(L,MMAX,N3))
      YWOLD=YW(L)
      POLD=P(L,MMAX,N3)
      YW(L)=YWNEW
  280 IF (YW(L).LT.(1.0-DYW)*YWOLD) YW(L)=(1.0-DYW)*YWOLD
```

```
      IF  (YW(L).GT.(1.0+DYW)*YWOLD)  YW(L)=(1.0+DYW)*YWOLD
  290 NANY(L)=-(YW(L)-YW(L-1))*DXR
      XWI(L)=(YW(L)-YWI(L))/DT
      XWID=XWI(L)
      CALL MAP ( 1,L,MMAX,AL,BE,DE,LD1,AL1,BE1,DE1)
      ALS=SQRT(AL*AL+BE*BE)
C
C     CALCULATE THE PSI TERMS AT THE SOLUTION POINT
C
  300 IF ( NDIM.EQ.0) GO TO 330
      ATERM3=RO3*V3/(1.0/BE)
  330 PSI13=-U3*DRO3-RO3*DU3-ATERM3
      PSI23=-U3*DU3-DP3/RO3
      PSI33=-U3*DV3
      PSI43=-U3*DP3+A3*A3*U3*DRO3
  340 ABR=NANY(L)
      ALB=AL/ALS
      BEB=BE/ALS
      A1B=(A1+A3)*0.5
      A2B=(A2+A3)*0.5
      RO2B=(RO2+RO3)*0.5
      IF (ICHAR.EQ.1) GO TO 350
      PSI21B=(PSI21+PSI23)*0.5+QUT(L,MMAX)
      PSI31B=(PSI31+PSI33)*0.5+QVT(L,MMAX)
      PSI41B=(PSI41+PSI43)*0.5+QPT(L,MMAX)
      PSI12B=(PSI12+PSI13)*0.5+QROT(L,MMAX)
      PSI22B=(PSI22+PSI23)*0.5+QUT(L,MMAX)
      PSI32B=(PSI32+PSI33)*0.5+QVT(L,MMAX)
      PSI42B=(PSI42+PSI43)*0.5+QPT(L,MMAX)
      GO TO 360
  350 PSI21B=PSI21+QUT(L,MMAX)
      PSI31B=PSI31+QVT(L,MMAX)
      PSI41B=PSI41+QPT(L,MMAX)
      PSI12B=PSI12
      PSI22B=PSI22
      PSI32B=PSI32
      PSI42B=PSI42
C
C     SOLVE THE COMPATIBILITY EQUATIONS FOR FREE SLIP WALLS
C
  360 XWID=0.0
      U(L,MMAX,N3)=(U1-ABR*(V1-XWID)+(PSI21B-ABR*PSI31B)*DT)/
     $ (1.+ABR*ABR)
      V(L,MMAX,N3)=-U(L,MMAX,N3)*ABR+XWID
      P(L,MMAX,N3)=P2-SIGN*RO2B*A2B*(ALB*(U(L,MMAX,N3)-U2)+
     $ BEB*(V(L,MMAX,N3)-V2))+(PSI42B+A2B*A2B*PSI12B+SIGN*
     $ RO2B*A2B*(ALB*PSI22B+BEB*PSI32B))*DT
      IF(P(L,MMAX,N3).LT.0.0)  P(L,MMAX,N3)=PLOW*PC
      RO(L,MMAX,N3)=RO1+(P(L,MMAX,N3)-P1-PSI41B*DT)/(A1B*A1B)
      IF(RO(L,MMAX,N3).LE.0.0)  RO(L,MMAX,N3)=ROLOW/G
```

```
      IF ( JFLAG.EQ.0) GO TO 510
      IF ( L.LT.LJET-1) GO TC 510
      IF ( L.EQ.LJET-1) GO TO 400
      IF ( ICHAR.EQ.1) GO TO 510
      IF ( JFLAG.EQ.-1.AND.L.NE.LJET) GO TO 510
      IF ( JFLAG.EQ.-1.AND.L.EQ.LJET) GO TO 500
      DELP=ABS ((P (L,MMAX,N3) -PE (MMAX))/PE (MMAX))
      IF ( DELP.LE.0.001.AND.L.NE.LJET) GO TO 510
      IF ( DELP.LE.0.001.AND.L.EQ.LJET) GO TO 500
  390 CONTINUE
      IF ( L.EQ.LJET) GO TO 500
      GO TO 510
  400 UD(3)=U(L,MMAX,N3)
      VD(3)=V(L,MMAX,N3)
      PD(3)=P(L,MMAX,N3)
      ROD(3)=RO(L,MMAX,N3)
      PD(4)=PE(MMAX)
      CALL EOS (5,PD(3),ROD(3),TD,AS,D2,D3)
      XM1=SQRT((UD(3)*UD(3)+VD(3)*VD(3))/AS)
      CALL EOS( 9,PD(3),ROD(3),TD,PTD,TTD,XM1)
      IF ( PE(MMAX).GT.PD(3).AND.XM1.GE.1.0) GO TO 450
      CALL EOS ( 11,PE(MMAX),ROD(4),TD,PD(3),ROD(3),D3)
      GO TO 460
C
  450 CALL EOS (12,PE(MMAX),ROD(4),TD,PD(3),ROD(3),D3)
  460 CALL EOS (2,PE(MMAX),ROD(4),TE,AS,D2,D3)
      CALL EOS ( 10,PE(MMAX),ROD(4),TE,PTD,TTD,XM2)
  470 CALL EOS (1,PD(4),ROD(4),T,AS,D2,D3)
      VMAG=XM2*SQRT(AS)
      UD(4)=VMAG/SQRT(1.0+NXNY(LJET)*NXNY(LJET))
      VD(4)=-UD(4)*NXNY(LJET)
      IF ( JFLAG.EQ.-1) GO TO 510
      IF ( XM1.GE.1.0) GO TO 510
      XMB=(XM1+XM2)/2.0
      IF ( XMB.GE.1.0) GO TO 480
      DPL=1.0
      DPR=1.0
      GC TO 490
  480 DPL=XM2-1.0
      DPR=1.0-XM1
      XMB=1.0
  490 DPLR=DPR+DPL
      CALL EOS (8 ,P(L,MMAX,N3),RO(L,MMAX,N3),TEMP,PTD,TTD,
     $ XMB)
      CALL EOS ( 6,P(L,MMAX,N3),RO(L,MMAX,N3),TEMP,AS,D2,D3)
      QA=XMB*SQRT(AS)
      DNXNY=(DPR*NXNY(LJET)+DPL*NXNY(L))/DPLR
      U(L,MMAX,N3)=QA/SQRT(1.0+DNXNY*DNXNY)
      V(L,MMAX,N3)=-U(L,MMAX,N3)*DNXNY
      GO TO 510
```

```
500  UD(1)=UD(3)
     VD(1)=VD(3)
     PD(1)=PD(3)
     ROD(1)=ROD(3)
     UD(2)=UD(4)
     VD(2)=VD(4)
     PD(2)=PD(4)
     ROD(2)=ROD(4)
510  CONTINUE
     DXR=DXX
     IF ( JFLAG.EQ.0) RETURN
     IF ( ICHAR.EQ.1) RETURN
     U(LJET-1,MMAX,N1)=UOLD
     IF ( JFLAG.EQ.-1) RETURN
     YW1(LMAX)=YW(LMAX)
     YW(LMAX)=2.0*YW(L1)-YW(L2)
     NXNY(LMAX)=-(YW(LMAX)-YW(L1))*DXR
     XW1(LMAX)=(YW(LMAX)-YW1(LMAX))/DT
     RETURN
     END
```

```
      SUBROUTINE GRAPH(XP1,YP1,NP)
C
C     ************************************************************
C
C     THIS SUBROUTINE PLOTS THE VELOCITY VECTORS
C
C     ************************************************************
C
      DIMENSION KSI(1000),XP(1000),XP1(50,20),YP1(50,20)
      DIMENSION TIME(1502),PRES1(1502),PRES2(1502),PRES3(1502)
      COMMON/ONESID/UD(4),VD(4),PD(4),ROD(4)
      COMMON/SOLUTN/U(50,20,2),V(50,20,2),P(50,20,2),
     $ RO(50,20,2)
      COMMON/CNTRLC/LMAX,MMAX,NMAX,TCONV,FDT,GAMMA,RGAS,GAM1,
     $ GAM2,GAM3,L1,L2,M1,M2,DX,DY,DT,ICHAR,N1D,IERR,IUI,IUO,
     $ DXR,DYR,RSTAR,N,N1,N3,RSTARS,G,PC,TC,LC,PLOW,ROLOW,RG,
     $ DXX,XE1
      COMMON/GEMTRL/XW(50),YW(50),NXNY(50),LT,XT,XI,XE,NDIM,
     $ XWI(50),YWI(50)
      COMMON/BCC/PT(20),TT(20),PE(20),IEXTRA,TW(20),ISUPER,
     $ UI(20),VI(20),PI(20),ROI(20),IEX
      COMMON/AV/CAV,NST,LSS,XMU,XLA,RKMU,XRO,QUT(50,20),
     $ QVT(50,20),QPT(50,20),QROT(50,20)
      COMMON/RV/CMU,CLA,CK,EMU,ELA,EK,CHECK,ITM,TML
      REAL MN3,NXNY,LC
      XW(LMAX+1)=0.0
      YW(LMAX+1)=0.0
      XW(LMAX+2)=1.0
      YW(LMAX+2)=1.0
      CALL VTHICK(2)
      CALL AXIS (0.0,0.0,'AXIAL DISTANCE',-15,6.0,0.0,
     $ XW(LMAX+1),XW(LMAX+2))
      CALL AXIS(0.0,0.0,'RADIUS',7,3.0,90.0,YW(LMAX+1),
     $ YW(LMAX+2))
      CALL VTHICK(4)
      CALL LINE (XW,YW,LMAX,1,0,0)
      CALL SYMBOL(0.75,-1.25,.14,'VELOCITY DISTRIBUTION IN
     $ THE DISCHARGE CHAMBER',0.0,46)
     $CHAMBER',0.0,46)
      EPSIL=0.000001
      DO 2 L=1,LMAX
      DO 2 M=1,MMAX
      Q=U(L,M,NP)*U(L,M,NP)+V(L,M,NP)*V(L,M,NP)
      Q1=SQRT(Q)
      IF(L.EQ.1.AND.M.EQ.1) QMAX=Q1
      IF(Q1.GT.QMAX) QMAX=Q1
    2 CONTINUE
      DO 3 L=1,LMAX
      DO 3 M=1,MMAX
      IF(L.EQ.1.AND.M.EQ.1) GO TO 3
```

```
      Y1=YP1(L,M)
      X1=XP1(L,M)
      XT1=U(L,M,NP)*0.25/QMAX
      YT1=V(L,M,NP)*0.25/QMAX
      XT2=ABS(XT1)
      YT2=ABS(YT1)
      IF(XT2.LT.EPSIL.AND.YT2.LT.EPSIL) GO TO 3
      XT=XT1+X1
      YT=YT1+Y1
      CALL AROHD(X1,Y1,XT,YT,.1,0.075,14)
    3 CONTINUE
      CALL VTHICK(1)
      CALL EOPLOT(1)
      RETURN
      END
```

```
      SUBROUTINE  SHOW(TIME,PRES1,PRES2,PRES3,N)
C
C     ************************************************************
C
C     THIS SUBROUTINE PLOTS A FLOW VARIABLE VERSUS TIME
C     AND LOCATION
C
C     ************************************************************
C
      DIMENSION KSI(1000),XP(1000),XP1(50,20),YP1(50,20)
      DIMENSION TIME(1502),PRES1(1502),PRES2(1502),PRES3(1502)
      COMMON/ONESID/UD(4),VD(4),PD(4),ROD(4)
      COMMON/SOLUTN/U(50,20,2),V(50,20,2),P(50,20,2),
     $ RO(50,20,2)
      COMMON/GEMTRL/XW(50),YW(50),NXNY(50),LT,XT,XI,XE,NDIM,
     $ XWI(50),YWI(50)
      COMMON/BCC/PT(20),TT(20),PE(20),IEXTRA,TW(20),ISUPER,
     $ UI(20),VI(20),PI(20),ROI(20),IEX
      COMMON/AV/CAV,NST,LSS,XMU,XLA,RKMU,XRO,QUT(50,20),
     $ QVT(50,20),QPT(50,20),QRCT(50,20)
      COMMON/RV/CMU,CLA,CK,EMU,ELA,EK,CHECK,ITM,TML
      REAL MN3,NXNY,LC
      CALL VTHICK(2)
      CALL NSCALE (TIME,5.0,N,1,1)
      CALL NSCALE (PRES1,5.,N,1,1)
      CALL NSCALE (PRES2,5.,N,1,1)
      CALL NSCALE (PRES3,5.,N,1,1)
      CALL GRID (0.,0.,2,5.,1.,2,5.,1.)
      CALL AXIS (0.,0.,'TIME IN SECONDS',-15,5.,0.,TIME(N+1),
     $ TIME(N+2))
      XMAX=AMAX1(PRES1(N+2),PRES2(N+2),PRES3(N+2))
      XMIN=AMIN1(PRES1(N+1),PRES2(N+1),PRES3(N+1))
      CALL AXIS (0.,0.,'DISCHARGE PRESSURE IN PSI',25,5.,90.,
     $ XMIN,XMAX)
      CALL VTHICK(3)
      CALL LINE(TIME,PRES1,N,1,10,1)
      CALL LINE(TIME,PRES2,N,1,10,2)
      CALL LINE(TIME,PRES3,N,1,10,3)
      CALL EOPLOT(1)
      RETURN
      END
```

- APPENDIX E -

```
C*************************************
FORTRAN IV LISTING OF THE PROGRAM FOR
THE MAIN AND BARREL CHAMBER FLOW
C*************************************
```

```
C                         MAIN PROGRAM
C
C     ---------------------------------------------------------
C
C     THIS COMPUTER PROGRAM IS FOR THE COMPUTATION OF ONE AND
C     TWO DIMENSIONAL,TIME DEPENDENT FLOW REPRESENTATIVE OF
C     A BARREL AND MAIN CHAMBER FLOW OF A GAS GUN
C
C     ---------------------------------------------------------
C
      COMMON/CONTRL/ ICOR,IE,E1,E2,E3,E4,GO,GL,GM,G1,G2,RG1,RG2
      COMMON /D0/ XBRL(70),UBRL(70,2),PBRL(70,2),RBRL(70,2)
      COMMON /D1/ X4,U4,P4,R4,X5,U5,P5,R5,X6,U6,P6,R6,X7,U7,P7
     $ ,R7,PA,ME
      COMMON /D2/ W3,W4,X8,U8,P8,R8,X3L,U3L,P3L,R3L,X3R,U3R,
     $ P3R,R3R,X4L,U4L,P4L,R4L,X4R,U4R,P4R,XP,AREA,MP
      COMMON/D3/DPBRL(70),DUBRL(70),DRBRL(70),NT1T1(8),NT1T2(8)
     $ ,NC(14)
      COMMON /D4/ X1L,X1R,U1L,U1R,P1L,P1R,R1L,R1R,R2L
      COMMON /D5/ XPROJ(1010),UPROJ(1010),PBPROJ(1010),
     $ XBPROJ(1010),XSHOCK(1010),USHOCK(1010),TIME(1010),
     $ XCAR(1010),XFPROJ(1010)
      COMMON/D6/ U(50,20,2),V(50,20,2),P(50,20,2),RO(50,20,2)
      COMMON/D7/ LMAX,MMAX,MCB,LCB,LBR,N,N1,N3,L1,L2,M1,M2,
     $ MCB1,MCB2,LCB1,LCB2,LBR1,LBR2,GAMMA,DX,DY,DT,ICHAR,DXR,
     $ DYR,RSTAR,RSTARS,G,PC,LC,PLOW,ROLOW,RG
      COMMON/D8/ XP1(50,20),YP1(50,20),KSI(2000),ETA(2000),
     $ PX(2000),PY(2000),XW(50),YW(20),MXMY(20),XE1,
     $ MDIM,PDS,AP
      COMMON/D9/CAV,XMU,XLA,RKMU,XRO,QUT(50,20),QVT(50,20),
     $ QPT(50,20),QROT(50,20),PE(20)
      COMMON /D15/ PREF1(1000),PREF2(1000),NCOUNT
      REAL LP,LM,LO,MUL,MPL,MRL,MUR,MPR,MRR,MP
      REAL MXMY,MYMX,LC,KSI
C
C     SET DEFAULT VALUES
C
      MAIN=2
      MPC=-1
      XSTEP=0.0
      JCOUNT=1
      NMAX=1000
      N1=1
      N3=2
      LCOUNT=1
      TSHOCK=1.0
      TB=0.0
      IH=-1
      BRLE=2.0
      NMAX=5000
```

```
      RSTARS=0.0
      SMACH=0.0
      LC=12.0
      NDIM=1
      GAMMA=1.4
      RGAS=53.35
      G=32.174
      PC=144.0
      RG=RGAS*G
      PLOW=0.01*PC
      ROLOW=0.0001/G
      PE(1)=14.7*PC
      T=0.0
      FDT=0.5
      PCHARG=50.E+5
      RCHARG=25.0
C     BOUNDARY CONDITIONS
      BI=PCHARG
      COI=RCHARG
      PA=103000.
      UB=-1.0
      ROA=1.22
      BI=BI*PC/6894.8
      COA=1.22/(G*16.02)
      COI=COI/(G*16.02)
      XI=0.0
      NPLOT=1
      PDS=5.0
      AP=0.0254*0.0254*3.1415*0.25
      UB=0.0
      XPNEW=0.0
C
C     GEOMETRY OF THE PHYSICAL DOMAIN
C
      XE=7.0
      LCB=30
      LCB=31
      LBR=34
      MCB=6
      MMAX=16
      LMAX=40
      L1=LMAX-1
      LCB1=LCB-1
      LCB2=LCB+1
      LBR1=LBR-1
      LBR2=LBR+1
      M1=MMAX-1
      MCB1=MCB-1
      MCB2=MCB+1
      Y2=3.0
```

```
      IF ( NPC.EQ.1 ) GO TO 432
C
      CALL GMETY
C
      DXRS=DXR*DXR
      DYRS=DYR*DYR
      LSS=LCB1
C     INITIAL CONDITIONS
      DO 19 L=1,LMAX
      DO 19 M=1,MMAX
      U(L,M,N1)=0.0
      U(L,M,N3)=0.0
      V(L,M,N1)=0.0
      V(L,M,N3)=0.0
   19 CONTINUE
      DO 15 L=1,LBR
      DO 15 M=MCB,MMAX
      U(L,M,N1)=0.0001
      U(L,M,N3)=0.0001
      V(L,M,N3)=0.0001
      V(L,M,N1)=0.0001
      P(L,M,N1)=BI
      P(L,M,N3)=BI
      RO(L,M,N1)=COI
      RO(L,M,N3)=COI
      QUT(L,M)=0.0
      QVT(L,M)=0.0
      QPT(L,M)=0.0
      QROT(L,M)=0.0
   15 CONTINUE
      DO 16 L=LCB,LMAX
      DO 16 M=1,MCB
      U(L,M,N1)=0.0001
      U(L,M,N3)=0.0001
      V(L,M,N3)=0.0001
      V(L,M,N1)=0.0001
      P(L,M,N1)=14.7*PC
      P(L,M,N3)=14.7*PC
      RO(L,M,N1)=COA
      RO(L,M,N3)=COA
      QUT(L,M)=0.0
      QVT(L,M)=0.0
      QPT(L,M)=0.0
      QROT(L,M)=0.0
   16 CONTINUE
C
C     GEOMETRY AND DEFAULT VALUES FOR THE BARREL
C
  432 FAR=0.0
      G1=1.4
```

```
      G2=G1
      RG1=287.0
      RG2=287.0
      GO=1.0
      GL=1.0
      GN=1.0
      ICOR=15
      E1=0.0001
      E2=0.0001
      E3=0.0001
      E4=0.0001
      DB=4.897*0.0254
      TB=0.0
      DTB=2.E-5
      KMAX=70
      MP=1.107
      LP=0.2
      AREA=(3.1415*DB*DB)/4.
      X3L=0.08
      X3R=X3L+LP
      P3R=1.013E+5
      R3R=1.22
      U3R=0.0001
      U3L=U3R
      P3L=P3R
      R3L=R3R
C
C     INITIAL CONDITIONS
C
      DO 71 K=1,KMAX
      UBRL(K,1)=0.0001
      UBRL(K,2)=0.0001
      PBRL(K,1)=1.013E+5
      PBRL(K,2)=1.013E+5
      RBRL(K,1)=1.22
      RBRL(K,2)=1.22
   71 CONTINUE
      PBRL(KMAX,1)=1.012E+5
      RBRL(KMAX,1)=1.15
      DS=2.0/(FLOAT(KMAX-1))
      X=0.0-DS
      DO 700 K=1,KMAX
      X=X+DS
      XBRL(K)=X
  700 CONTINUE
      AZ=340.
      XE2DX=BRLE-2.0*DS
      IF ( NPC.NE.1 ) GO TO 430
      ACC=PCHARG*AREA/MP
      TSHOCK=2.*AZ/((G1+1.)*ACC)
```

```
      USH=AZ
      XSH=TSHOCK*AZ
      XSH=XSH+X3R
      XSH=X3R+2.*DS
      TSHOCK=0.0
      TSHOCK=1.0
      XSHOCK(1)=XSH
      USHOCK(1)=USH
      WRITE( 6,10 ) TSHOCK,XSH,USH
      DO 72 K=1,10
      PBRL(K,1)=PCHARG
      PBRL(K,2)=PCHARG
      RBRL(K,1)=RCHARG
      RBRL(K,2)=RCHARG
   72 CONTINUE
      P3L=PCHARG
      R3L=RCHARG
      U3L=0.0001
      U3R=U3L
  430 FARID=0.0
      NPLOT=1
      XPROJ(1)=X3L
      UPROJ(1)=U3L
      PBPROJ(1)=P3L
      TIME(1)=0.0
      LCOUNT=1
      X1L=X3L
      X1R=X3R
      U1L=U3L
      U1R=U3R
      P1L=P3L
      P1R=P3R
      R1R=R3R
      R1L=R3L
      R2L=R1R
C
C     ENTER STEP TIME INTEGRATION
C
      DO 660 N=1,NMAX
      IF ( NPC.EQ.1 ) GO TO 76
      DO 450 L=LCB,LMAX
      WX=SQRT(1.+NXNY(L)*NXNY(L))
      DO 450 M=1,MCB
      WY=SQRT(1.+NYNX(M)*NYNX(M))
      AS=P(L,M,N1)*GAMMA/RO(L,M,N1)
      DXDY=WX*WX*DXRS+WY*WY*DYRS
      UPA=ABS(U(L,M,N1))*WX*DXR+ABS(V(L,M,N1))*WY*DYR+
     $ SQRT(AS*DXDY)
      IF (L.EQ.LCB.AND.M.EQ.1) UPAM=UPA
      IF (UPA.GT.UPAM) UPAM=UPA
```

```
  450 CONTINUE
      DO 451 L=1,LBR
      WX=SQRT(1.+NXNY(L)*NXNY(L))
      DO 451 M=MCB,MMAX
      WY=SQRT(1.+NYNX(M)*NYNX(M))
      AS=P(L,M,N1)*GAMMA/RO(L,M,N1)
      DXDY=WX*WX*DXRS+WY*WY*DYRS
      UPA=ABS(U(L,M,N1))*WX*DXR+ABS(V(L,M,N1))*WY*DYR+
     $ SQRT(AS*DXDY)
      IF (UPA.GT.UPAM) UPAM=UPA
  451 CONTINUE
      DT=FDT/UPAM
      DTT=DT/LC
      T=T+DTT
      TB=TB+DTT
C
C     FIRST STEP INTEGRATION
C
      ICHAR=1
      CALL INTER(2,LCB,MCB2,N1)
      CALL INTER(LBR,L1,1,MCB1)
      CALL INTER(LCB2,LBR1,1,M1)
      CALL BDRY ( UB )
      PE(1)=PBRL(2,N1)*PC/6894.2
      PINB=PE(1)*47.876
      CALL EXIT ( 0 )
C
C     SECOND STEP INTEGRATION
C
      ICHAR=2
      CALL INTER(2,LCB,MCB2,M1)
      CALL INTER(LBR,L1,1,MCB1)
      CALL INTER(LCB2,LBR1,1,M1)
      CALL BDRY ( UB )
      CALL SMOOTH(2,LCB1,MCB2,MMAX)
      CALL SMOOTH(LBR2,L1,2,MCB)
      CALL SMOOTH(LCB2,LBR1,2,MMAX)
      CALL BDRY ( UB )
      DO 21 L=1,LCB
      U(L,MCB,N3)=-UB
   21 CONTINUE
      MCB3=MCB-2
      U(LBR,MCB,N3)=0.25*(U(LBR1,MCB,N3)+2.0*U(LBR,MCB1,N3)+
     $ U(LBR2,MCB,N3))
      V(LBR,MCB,N3)=0.25*(V(LBR1,MCB,N3)+2.0*V(LBR,MCB1,N3)+
     $ V(LBR2,MCB,N3))
      P(LBR,MCB,N3)=0.25*(P(LBR1,MCB,N3)+2.0*P(LBR,MCB1,N3)+
     $ P(LBR2,MCB,N3))
      RO(LBR,MCB,N3)=0.25*(RO(LBR1,MCB,N3)+2.0*RO(LBR,MCB1,N3)+
     $ RO(LBR2,MCB,N3))
```

```
      CALL EXIT ( 0 )
      IF ( XPNEW.GT.0.021 ) GO TO 27
      XPOLD=XPNEW
      VPOLD=UB*3.208
      CALL PISTON ( VPOLD,XPOLD,VPNEW,XPNEW )
      UB=VPNEW/3.208
      IF ( N.EQ.20 ) CALL GRAPH ( N3,T,VPNEW )
      IF ( ABS ( XPNEW-XSTEP).LT.0.0254*(XW(LCB)-XW(LCB1)))
     $ GO TO 28
      LCB=LCB1
      LCB1=LCB-1
      LCB2=LCB+1
      DO 452 M=1,MCB
      U(LCB,M,N1)=U(LCB2,M,N1)
      V(LCB,M,N1)=V(LCB2,M,N1)
      P(LCB,M,N1)=P(LCB2,M,N1)
      RO(LCB,M,N1)=RO(LCB2,M,N1)
      U(LCB,M,N3)=U(LCB2,M,N3)
      V(LCB,M,N3)=V(LCB2,M,N3)
      P(LCB,M,N3)=P(LCB2,M,N3)
      RO(LCB,M,N3)=RO(LCB2,M,N3)
  452 CONTINUE
      XSTEP=XPNEW
      WRITE(6,10) PINB,DTT,XPNEW,VPNEW
      CALL GRAPH ( N3,T,VPNEW )
      GO TO 28
   27 UB=0.0
      VPNEW=0.0
   28 HAR=0.0
      IF ( NPLOT.EQ.70 ) CALL GRAPH ( N3,T,VPNEW )
      IF ( MAIN.EQ.1 ) GO TO 24
      IF ( N.EQ.1 ) GO TO 79
      IF ( TB.LT.DTB ) GO TO 89
      TB=0.0
   79 HAR=0.0
      UAV=(U(LMAX,1,N3)+U(LMAX,MCB,N3))*0.5/3.208
      ROAV=(RO(LMAX,1,N3)+RO(LMAX,MCB,N3))*0.5*32.194*16.02
      PAV=(P(LMAX,1,N3)+P(LMAX,MCB,N3))*0.5*6894.2/PC
      UBRL(1,N3)=UAV
      RBRL(1,N3)=ROAV
      PBRL(1,N3)=PAV
   76 IF ( NPC.EQ.1 ) T=T+DTB
      CALL BARREL ( DS,IH,NPC,DTB,NPLOT,N1,N3 )
      DO 1 K=1,KMAX
      DPBRL(K)=(PBRL(K,N3)-PBRL(K,N1))/DTB
      DUBRL(K)=(UBRL(K,N3)-UBRL(K,N1))/DTB
      DRBRL(K)=(RBRL(K,N3)-RBRL(K,N1))/DTB
    1 CONTINUE
      NPLOT=NPLOT+1
      TIME(NPLOT)=T
```

```
      X5L=XPROJ(NPLOT)
      X5R=X5L+LP
      IF ( X5L.GE.BRLE ) GO TO 890
      IF ( T.GE.TSHOCK ) IH=2
      IF ( XSHOCK(NPLOT).GE.XE2DX) IH=-1
      IF ( X5R.GT.XE2DX ) IH=1
      IF ( JCOUNT.EQ.200 ) GO TO 77
      IF ( NPC.EQ.1 ) GO TO 84
      GO TO 89
   77 JCOUNT=1
      LCOUNT=1
      WRITE(6,88) DT,T
      DO 74 K=1,KMAX
      X=XBRL(K)*100.
      VEL=UBRL(K,N3)
      PRES=PBRL(K,N3)*0.001
      PRESS=PRES*1000.
      RHO=RBRL(K,N3)
      CALL THERMO ( PRESS,RHO,A)
      XMACH=VEL/A
      WRITE (6,110) X,VEL,PRES,RHO,XMACH
   74 CONTINUE
      WRITE(6,103)
   89 HAR=0.0
      IF ( NPC.EQ.1 ) GO TO 84
      UAV=(U(LMAX,1,N3)+U(LMAX,MCB,N3))*0.5/3.208
      ROAV=(RO(LMAX,1,N3)+RO(LMAX,MCB,N3))*0.5*32.194*16.02
      PAV=(P(LMAX,1,N3)+P(LMAX,MCB,N3))*0.5*6894.2/PC
      UBRL(1,N3)=UAV
      RBRL(1,N3)=ROAV
      PBRL(1,N3)=PAV
      DO 22 L=2,KMAX
      UBRL(L,N3)=UBRL(L,N1)+DUBRL(L)*DTT
      PBRL(L,N3)=PBRL(L,N1)+DPBRL(L)*DTT
      RBRL(L,N3)=RBRL(L,N1)+DRBRL(L)*DTT
   22 CONTINUE
C
   24 HAR=0.0
      IF ( JCOUNT.EQ.10000 ) GO TO 85
      JCOUNT=JCOUNT+1
      GO TO 84
   85 JCOUNT=1
      WRITE(6,88) DTT,T
      DO 580 L=1,LBR
      DO 580 M=MCB,MMAX
      VELMAG=SQRT(U(L,M,N3)**2+V(L,M,N3)**2)
      XMACH=VELMAG/SQRT(P(L,M,N3)*GAMMA/RO(L,M,N3))
      DRES=P(L,M,N3)/PC
      RHO=RO(L,M,N3)*G
      UP=U(L,M,N3)
```

```
      VP=V(L,M,N3)
      XP=XP1(L,M)
      YP=YP1(L,M)
      WRITE(6,920) L,M,XP,YP,UP,VP,DRES,RHO,VELMAG,XMACH
  580 CONTINUE
      DO 581 L=LCB,LMAX
      DO 581 M=1,MCB
      VELMAG=SQRT(U(L,M,N3)**2+V(L,M,N3)**2)
      XMACH=VELMAG/SQRT(P(L,M,N3)*GAMMA/RO(L,M,N3))
      DRES=P(L,M,N3)/PC
      RHO=RO(L,M,N3)*G
      UP=U(L,M,N3)
      VP=V(L,M,N3)
      XP=XP1(L,M)
      YP=YP1(L,M)
      WRITE(6,920) L,M,XP,YP,UP,VP,DRES,RHO,VELMAG,XMACH
  581 CONTINUE
C     PRINT THE SOLUTION
      WRITE (6,103)
      WRITE (6,101)
      WRITE (6,102)
   75 PARID=0.0
   84 NNN=N1
      N1=N3
      N3=NNN
  660 CONTINUE
C     PRINT THE SOLUTIONS
  890 WRITE(6,88) DT,T
      DSS=AZ*DTB
      XFPROJ(1)=0.28
      XCAR(1)=0.28
      XL1=0.28
      DX5=.20
      XJ4=0.0
      DO 992 L=2,NPLOT
      IF ( XL1.GE.2.0 ) DSS=0.0
      XL1=XL1+DSS
      XCAR(L)=XL1
      IF ( XJ4.GE.2.0 ) GO TO 993
      XJ4=XPROJ(L)+DX5
      XFPROJ(L)=XJ4
      GO TO 992
  993 XFPROJ(L)=XFPROJ(L-1)
  992 CONTINUE
      WRITE (6,10) XCAR(NPLOT),XCAR(1)
      DO 994 L=1,NPLOT
      PBPROJ(L)=PBPROJ(L)*0.001
  994 CONTINUE
      CALL IDENT
      CALL PLOT(4.0,4.0,-3)
```

```
      DO 1112 L=1,NPLOT
      PREF1(L)=PREF1(L)*0.001
      PREF2(L)=PREF2(L)*0.001
 1112 CONTINUE
      N=NPLOT
      CALL VTHICK(3)
      CALL NSCALE(PREF1,5.0,N,1,1)
      CALL NSCALE(PREF2,5.0,N,1,1)
      CALL NSCALE(TIME,4.0,N,1,1)
      TIME(N+1)=0.000
      TIME(N+2)=0.003
      CALL AXIS(0.0,0.0,'TIME (S)',-8,4.0,0.0,TIME(N+1),
     $ TIME(N+2))
      XMAX=AMAX1(PREF1(N+2),PREF2(N+2))
      XMIN=AMIN1(PREF1(N+1),PREF1(N+1))
      CALL AXIS(0.0,0.0,'PRESSURE (KPA)',14,5.0,90.0,XMIN,XMAX)
      CALL LINE(TIME,PREF1,N,1,20,2)
      CALL LINE(TIME,PREF2,N,1,20,0)
      CALL VECTOR(0.0,0.0,0.0,6.0)
      CALL VECTOR(0.0,6.0,5.0,6.0)
      CALL VECTOR(5.0,6.0,5.0,0.0)
      CALL VECTOR(5.0,0.0,0.0,0.0)
      CALL EOPLOT(1)
      CALL SHOW ( NPLOT )
      IF ( TSHOCK.NE.1) CALL SHOW2 ( NPLOT )
      CALL EOJOB
C
C     FORMAT STATEMENTS
C
   88 FORMAT ( /,5X,'DT=',F12.9,3X,'T=',F12.9,/)
  101 FORMAT (10X,'LOCATION',14X,'VELOCITY',13X,'PRESSURE',
     $ 8X,'DENSITY',8X,'MACH NUMBER',/)
  102 FORMAT ( 15X,'(CM)',15X,'(M/S)',15X,'(KPA)',10X,'(KG/M3
     $ )',15X,'(NO)',/)
  103 FORMAT (/,11X,'X4L',11X,'U4L',11X,'P4L',11X,'R4L',11X,'
     $ 'X4R',11X,'U4R',13X,'P4R',11X,'R4R',/)
  920 FORMAT(1H,7X,2I5,4F12.4,F13.5,F12.6,2F12.4)
   10 FORMAT (5X,9E14.5)
  110 FORMAT (10X,F8.3,14X,F8.3,13X,3(F8.3,8X))
  910 FORMAT(1H,25X,4H(CM),7X,4H(CM),6X,5H(M/S),7X,5H(M/S),7X,
     $ 6H (KPA),7X,7H(KG/M3),5X,5H(M/S),10X,2HNO)
  100 FORMAT (5X,4E14.5)
 1190 FORMAT(1H,10X,'-----------------------------------------'
     $,'---------------------------------------------------'
     $,'-------')
       STOP
       END
```

```
      SUBROUTINE BARREL ( DS,IH,NPC,DTB,NPLOT,N1,N3 )
C
C     ----------------------------------------------------------
C
C     THIS SUBROUTINE IS USED FOR THE CALLING OF THE DIFFERENT
C     SUBROUTINES NEEDED TO COMPUTE THE FLOW VARIABLES
C     THROUGHOUT THE BARREL
C
C     ----------------------------------------------------------
C
      COMMON/CONTRL/ ICOR,IE,E1,E2,E3,E4,GO,GL,GN,G1,G2,RG1,RG2
      COMMON /D0/ XBRL(70),UBRL(70,2),PBRL(70,2),RBRL(70,2)
      COMMON /D1/ X4,U4,P4,R4,X5,U5,P5,R5,X6,U6,P6,R6,X7,U7,P7
     $ ,R7,PA,ME
      COMMON /D2/ W3,W4,X8,U8,P8,R8,X3L,U3L,P3L,R3L,X3R,U3R,
     $ P3R,R3R,X4L,U4L,P4L,R4L,X4R,U4R,P4R,XP,AREA,MP
      COMMON/D3/DPBRL(70),DUBRL(70),DRBRL(70),NT1T1(8),NT1T2(8)
     $ ,NC(14)
      COMMON /D4/ X1L,X1R,U1L,U1R,P1L,P1R,R1L,R1R,R2L
      COMMON /D5/ XPROJ(1010),UPROJ(1010),PBPROJ(1010),
     $ XBPROJ(1010),XSHOCK(1010),USHOCK(1010),TIME(1010),
     $ XCAR(1010),XFPROJ(1010)
      COMMON /D15/ PREF1(1000),PREF2(1000),NCOUNT
      REAL LP,LM,LO,MUL,MPL,MRL,MUR,MPR,MRR,MP
      KMAX=70
      T=FLOAT(NPLOT)*DTB
      USH=USHOCK(NPLOT)
      XSH=XSHOCK(NPLOT)
      X1=X1L-DS
      X3RDX=X1R+DS
      XL1=X1L
      XR1=X1R
      XX1=XL1+0.2
      XSHDS=XSH-DS
      JCOUNT=1
      IH2=-1
      IH1=0
      DO 840 K=1,KMAX
      IF ( IH1.EQ.1 ) GO TO 96
      IF ( IH2.EQ.1 ) GO TO 99
      IF ( K.EQ.1) GO TO 840
      IF ( IH.NE.-1 ) GO TO 94
      IF ( K.EQ.KMAX) GO TO 810
   94 FAR=0.0
C
C     COMPUTE THE PROPERTIES OF THE POINTS  LOCATED AT THE LHS
C     OF THE PROJECTILE
C
      IF ( XBRL(K).GT.XBRL(1).AND.XBRL(K).LE.X1) GO TO 730
      GO TO 740
```

```
  730 X5=XBRL(K-1)
      X6=XBRL(K)
      X4=X6
      X7=XBRL(K+1)
      U5=UBRL(K-1,N1)
      U6=UBRL(K,N1)
      U7=UBRL(K+1,N1)
      P5=PBRL(K-1,N1)
      P6=PBRL(K,N1)
      P7=PBRL(K+1,N1)
      R5=RBRL(K-1,N1)
      R6=RBRL(K,N1)
      R7=RBRL(K+1,N1)
C
      CALL POINT  (DTB)
C
      UBRL(K,N3)=U4
      PBRL(K,N3)=P4
      RBRL(K,N3)=R4
      IF (K.NE.2)  GO TO 840
      IF ( NPC.NE.1 ) GO TO 840
      UBRL(1,N3)=UBRL(2,N3)
      PBRL(1,N3)=PBRL(2,N3)
      RBRL(1,N3)=RBRL(2,N3)
      GO TO 840
  740 IF ( XBRL(K).GT.X1.AND.XBRL(K).LT.XL1)  GO TO 750
      IF ( IH.NE.-1 ) GO TO 92
      GO TO 760
  750 K1=K
      IF ( IH.NE.-1 ) GO TO 92
      GO TO 840
C
C     POINTS LOCATED BETWEEN THE LHS AND THE RHS OF THE
C     THE PROJECTILE
C
  760 IF (XBRL(K).GT.XL1.AND.XBRL(K).LT.XR1)  GO TO 840
C
C     LOCATE THE POINT TO BE USED FOR THE RHS OF THE
C     PROJECTILE
C
      IF ( XBRL(K).GT.XR1.AND.XBRL(K).LT.X3RDX)  GO TO 780
      GO TO 790
  780 K2=K
      X5=XBRL(K1-1)
      U5=UBRL(K1-1,N1)
      P5=PBRL(K1-1,N1)
      R5=RBRL(K1-1,N1)
      X7=XBRL(K2+1)
      U7=UBRL(K2+1,N1)
      P7=PBRL(K2+1,N1)
```

```
      R7=RBRL(K2+1,N1)
      X3L=X1L
      X3R=X1R
      U3L=U1L
      U3R=U1R
      P3L=P1L
      P3R=P1R
      R3L=R1L
      R3R=R1R
C
      CALL MISILE ( DTB,2 )
C
      P4L1=P4L*1.0E-5
      P4R1=P4R*1.0E-5
      X41=X4L*100.
      X42=X4R*100.
      WRITE ( 6,10 ) NPLOT,X41,U4L,P4L1,R4L,P4R1,R4R,T,IH
   10 FORMAT ( 5X,I3,5X,7E14.5,I3 )
      NCOUNT=NPLOT
C
      CALL PSABOT(U4L,P4R,P3R,IH)
C
      WRITE(6,1111) PREF1(NCOUNT),PREF2(NCOUNT)
 1111 FORMAT(50X,2E14.5)
      XPROJ(NPLOT+1)=X4L
      UPROJ(NPLOT+1)=U4L
      PBPROJ(NPLOT+1)=P4L
      X1L=X4L
      X1R=X4R
      U1L=U4L
      U1R=U4R
      P1L=P4L
      P1R=P4R
      R1R=R4R
      R1L=R4L
      GO TO 840
  790 IF ( XBRL(K).GT.X3RDX) GO TO 800
      GO TO 840
  800 X5=XBRL(K-1)
      X6=XBRL(K)
      X4=X6
      X7=XBRL(K+1)
      U5=UBRL(K-1,N1)
      P5=PBRL(K-1,N1)
      R5=RBRL(K-1,N1)
      U6=UBRL(K,N1)
      P6=PBRL(K,N1)
      R6=RBRL(K,N1)
      U7=UBRL(K+1,N1)
      P7=PBRL(K+1,N1)
```

```
      R7=RBRL(K+1,N1)
      CALL POINT (DTB)
      UBRL(K,N3)=U4
      PBRL(K,N3)=P4
      RBRL(K,N3)=R4
      GO TO 840
  810 UBRL(K1,N3)=0.5*(U4L+UBRL(K1-1,N3))
      PBRL(K1,N3)=0.5*(P4L+PBRL(K1-1,N3))
      RBRL(K1,N3)=0.5*(R4L+RBRL(K1-1,N3))
      UBRL(K2,N3)=0.5*(U4R+UBRL(K2+1,N3))
      PBRL(K2,N3)=0.5*(P4R+PBRL(K2+1,N3))
      RBRL(K2,N3)=0.5*(R4R+RBRL(K2+1,N3))
      M3=K1+1
      M4=K2-1
      DO 73 M=M3,M4
      UBRL(M,N3)=U4L
      PBRL(M,N3)=P4L
      RBRL(M,N3)=R4L
   73 CONTINUE
      X5=XBRL(K-1)
      U5=UBRL(K-1,N1)
      P5=PBRL(K-1,N1)
      R5=RBRL(K-1,N1)
      X6=XBRL(K)
      U6=UBRL(K,N1)
      P6=PBRL(K,N1)
      R6=RBRL(K,N1)
      CALL OPEN (DTB)
      UBRL(K,N3)=U4
      PBRL(K,N3)=P4
      RBRL(K,N3)=R4
      GO TO 840
   92 HAR=0.0
      IF ( IH.EQ.2 ) GO TO 95
C
C     THIS STEP IS EXCUTED IF THE RHS OF THE PROJECTIE IS OUT
C
      X3L=X1L
      X3R=X1R
      U3L=U1L
      U3R=U1R
      P3L=P1L
      P3R=P1R
      R3L=R1L
      R3R=R1R
      X5=XBRL(K1-1)
      U5=UBRL(K1-1,N1)
      P5=PBRL(K1-1,N1)
      R5=RBRL(K1-1,N1)
C
```

```
      CALL MISILE ( DTB,1 )
C
      P4L1=P4L*1.0E-5
      P4R1=P4R*1.0E-5
      X41=X4L*100.
      X42=X4R*100.
      WRITE ( 6, 10 ) NPLOT,X41,U4L,P4L1,R4L,P4R1,R4R,T,IH
      XPROJ(NPLOT+1)=X4L
      UPROJ(NPLOT+1)=U4L
      PBPROJ(NPLOT+1)=P4L
      X1L=X4L
      X1R=X4R
      U1L=U4L
      U1R=U4R
      P1L=P4L
      P1R=P4R
      R1R=R4R
      R1L=R4L
      IH1=1
   96 HAR=0.0
      UBRL(K,N3)=U1L
      PBRL(K,N3)=P1L
      RBRL(K,N3)=R1L
      GO TO 840
C     THE SHOCK HAS OCCURED INSIDE THE BARREL
   95 HAR=0.0
      IF (JCOUNT.EQ.2) GO TO 111
      X3L=X1L
      X3R=X1R
      U3L=U1L
      U3R=U1R
      P3L=P1L
      P3R=P1R
      R3L=R1L
      R3R=R1R
      X5=XBRL(K1-1)
      U5=UBRL(K1-1,N1)
      P5=PBRL(K1-1,N1)
      R5=RBRL(K1-1,N1)
C
      CALL MISILE ( DTB, 1 )
C
      P4L1=P4L*1.0E-5
      P4R1=P4R*1.0E-5
      X41=X4L*100.
      X42=X4R*100.
      WRITE ( 6,10 ) NPLOT,X41,U4L,P4L1,R4L,P4R1,R4R,T,IH
      XPROJ(NPLOT+1)=X4L
      UPROJ(NPLOT+1)=U4L
      PBPROJ(NPLOT+1)=P4L
```

```
      X1L=X4L
      X1R=X1L+0.2
      U1L=U4L
      U1R=U1L
      P1L=P4L
      P1R=P4R
      R1L=R4L
      R1R=R4R
      JCOUNT=2
      GO TO 840
  111 HAR=0.0
C
C     LOCATE POINTS BETWEEN THE LHS AND THE RHS OF THE
C     PROJECTILE
C
      IF ( XBRL(K).GT.XX1 ) GO TO 97
      UBRL(K,N3)=U1L
      PBRL(K,N3)=P1L
      RBRL(K,N3)=R1L
      GO TO 840
   97 HAR=0.0
      IF ( XBRL(K).GT.XSHDS ) GO TO 98
      UBRL(K,N3)=U1R
      PBRL(K,N3)=P1R
      RBRL(K,N3)=R1R
      GO TO 840
   98 IF ( XBRL(K).GT.XSH ) GO TO 99
      KS8=K-1
      KS5=K+1
      KS6=K+2
      KS7=K+3
      X5=XBRL(KS5)
      X6=XBRL(KS6)
      X7=XBRL(KS7)
      X8=XBRL(KS8)
      P5=PBRL(KS5,N1)
      P6=PBRL(KS6,N1)
      P7=PBRL(KS7,N1)
      P8=PBRL(KS8,N1)
      R5=RBRL(KS5,N1)
      R6=RBRL(KS6,N1)
      R7=RBRL(KS7,N1)
      R8=RBRL(KS8,N1)
      U5=UBRL(KS5,N1)
      U6=UBRL(KS6,N1)
      U7=UBRL(KS7,N1)
      U8=UBRL(KS8,N1)
      X3L=XSH
      X3R=X3L
      X3=X3L
```

```
      U3=USH
      U3L=U8
      P3L=P8
      R3L=R8
      U3R=U5
      P3R=P5
      R3R=R5
C
      CALL SHOCK  ( DTB,X3 )
C
      IH2=1
      XSH=X4
      USH=W4
      U2R=U4R
      P2R=P4R
      R2R=R4R
      U2L=U4L
      P2L=P4L
      R2L=R4L
      UBRL(KS8+1,N3)=U4L
      PBRL(KS8+1,N3)=P4L
      RBRL(KS8+1,N3)=R4L
      XSHOCK(NPLOT+1)=XSH
      USHOCK(NPLOT+1)=USH
      R4R=R2L
      P4R=P2L
      WRITE(6,10)  NPLOT,XSH,USH ,P2L,P2R,R2L,R2R
      GO TO 840
C     POINTS LOCATED IN FONT OF THE SHOCK
   99 HAR=0.0
      UBRL(K,N3)=U2R
      PBRL(K,N3)=P2R
      RBRL(K,N3)=R2R
  840 CONTINUE
      XSHOCK(NPLOT+1)=XSH
      USHOCK(NPLOT+1)=USH
      X3DS=2.0-3.0*DS
      IF ( XSH.GE.X3DS )  IH=-1
      RETURN
      END
```

```
      SUBROUTINE PSABOT(U4L,P4R,P3R,IH)
C
C     -----------------------------------------------------------
C
C     THIS SUBROUTINE CALCULATES THE PRESSURE AHEAD OF THE
C     PROJECTILE
C
C     -----------------------------------------------------------
C
      COMMON /D15/ PREF1(1000),PREF2(1000),NCOUNT
      PA=1.031E+5
      XMA=U4L/340.
      XMAI=1./XMA
      SPART=SQRT(2.4*2.4*0.25*0.25+XMAI*XMAI)
      P4RI=PA*(1.+1.4*XMA*XMA*(2.4*0.25+SPART))
      P4RT=PA*(1.+0.2*XMA)**7.0
      PREF1(NCOUNT)=P4RI
      PREF2(NCOUNT)=P4RT
      IF(IH.EQ.-1) RETURN
      P3R=P4RT
      P4R=P3R
      RETURN
      END
```

```
      SUBROUTINE  OPEN (DT)
C
C     ----------------------------------------------------------
C
C     SUBROUTINE OPEN CALCULATES THE SOLUTION AT AN OPEN END
C
C     ----------------------------------------------------------
C
      REAL LP,LM,LO,MUL,MPL,MRL,MUR,MPR,MRR,MP
      COMMON/CONTRL/ ICOR,IE,E1,E2,E3,E4,GO,GL,GM,G1,G2,RG1,RG2
      COMMON /D1/ X4,U4,P4,R4,X5,U5,P5,R5,X6,U6,P6,R6,X7,U7,P7
     $ ,R7,PA,ME
C
C     DEFINE INITIAL PROPERTIES AND DETERMINE INTERPOLATING
C     POLYNOMIALS
C
      ITER=0
      IE=0
      P4=P6
      PA=P6
      X1S=X5
      X3S=X6
      X4=X6
      U1=U5
      P1=P5
      R1=R5
      U3=U6
      P3=P6
      R3=R6
      DX=X5-X6
      MUL=(U5-U6)/DX
      BUL=U6-MUL*X6
      MPL=(P5-P6)/DX
      BPL=P6-MPL*X6
      MRL=(R5-R6)/DX
      BRL=R6-MRL*X6
C
C     LOCATE POINT 1 AND DETERMINE COEFFICIENTS ALONG LINE 14
C
   10 IF (ITER.GT.0) GO TO 20
      U4=U1
      P4=P1
      R4=R1
   20 U=0.5*(U1+U4)
      P=0.5*(P1+P4)
      R=0.5*(R1+R4)
      CALL THERMO (P,R,A)
      LP=1.0/(U+A)
      X1=X4-DT*3N/LP
      IF (ABS(X1-X1S).LT.0.0001) GO TO 30
```

```
      X1S=X1
      U1=MUL*X1+BUL
      P1=MPL*X1+BPL
      R1=MRL*X1+BRL
      GO TO 10
   30 QP=R*A/GO
      TP=GL*P1+QP*U1
C
C     LOCATE POINT 3 AND DETERMINE COEFFICIENTS ALONG LINE 34
C
   40 IF (ITER.GT.0) GO TO 50
      U4=U3
      P4=P3
      R4=R3
   50 U=0.5*(U3+U4)
      P=0.5*(P3+P4)
      R=0.5*(R3+R4)
      CALL THERMO (P,R,A)
      LO=1.0/U
      X3=X4-DT*GN/LO
      IF (ABS(X3-X3S).LT.0.0001) GO TO 60
      X3S=X3
      U3=MUL*X3+BUL
      P3=MPL*X3+BPL
      R3=MRL*X3+BRL
      GO TO 40
   60 AO=A**2/GO
      TO=GL*P3-AO*R3
C
C     CALCULATE THE PROPERTIES AT POINT 4,AND TEST FOR
C     CONVERGENCE
C
      IF (IR.EQ.0) GO TO 90
      I=1
      IF ( ITER.EQ.0) P4=P6
   70 U4=(TP-GL*P4)/QP
      R4=(GL*P4-TO)/AO
      CALL THERMO(P4,R4,AO)
      M4=U4/A4
      FF=FLOAT(M4-ME)
      IF(ABS(FF).LT.0.00001) GO TO 100
      IF (I.GT.1) GO TO 80
      I=I+1
      P4I=P4
      M4I=M4
      P4=1.01*P4
      GO TO 70
   80 SL=(M4-M4I)/(P4-P4I)
      P4I=P4
      M4I=M4
```

```
      P4=P4+(RE-R4)/SL
      GO TO 70
   90 P4=PA
      U4=(TP-GL*P4)/QP
      R4=(GL*P4-TO)/AO
  100 IF (ITER.EQ.ICOR) RETURN
      IF ( ITER.EQ.0) GO TO 110
      IF(( ABS(U4-UD).LT.E1*UD).AND.(ABS(R4-RD).LT.E3*RD))
     $ RETURN
  110 ITER=ITER+1
      UD=U4
      RD=R4
      GO TO 10
      END
```

```
      SUBROUTINE MISILE (DT,IH)
C
C     ------------------------------------------------------------
C
C     SUBROUTINE MISSILE CALCULATES PROPERTIES OF A MOVING
C     PROJECTILE
C
C     ------------------------------------------------------------
C
      REAL LP,LM,LO,MUL,MPL,MRL,MUR,MPR,MRR,MP
      COMMON/CONTRL/ ICOR,IE,E1,E2,E3,E4,GO,GL,GN,G1,G2,RG1,RG2
      COMMON /D1/ X4,U4,P4,R4,X5,U5,P5,R5,X6,U6,P6,R6,X7,U7,P7
     $ ,R7,PA,ME
      COMMON /D2/ W3,W4,X8,U8,P8,R8,X3L,U3L,P3L,R3L,X3R,U3R,
     $ P3R,R3R,X4L,U4L,P4L,R4L,X4R,U4R,P4R,XP,AREA,MP
C
C     DEFINE INITIAL PROPERTIES AND DETERMINE INTERPOLATING
C     POLYNOMIALS
C
      X1S=X5
      U1=U5
      P1=P5
      R1=R5
      P4L=P3L
      DX=X5-X3L
      MUL=(U5-U3L)/DX
      BUL=U3L-MUL*X3L
      MPL=(P5-P3L)/DX
      BPL=P3L-MPL*X3L
      MRL=(R5-R3L)/DX
      BRL=R3L-MRL*X3L
      IF ( IH.EQ.1 ) GO TO 11
      X2S=X7
      U2=U7
      P2=P7
      R2=R7
      P4R=P3R
      DX=X3R-X7
      MUR=(U3R-U7)/DX
      BUR=U7-MUR*X7
      MPR=(P3R-P7)/DX
      BPR=P7-MPR*X7
      MRR=(R3R-R7)/DX
      BRR=R7-MRR*X7
      ITER=0
C
C     CALCULATE ACCELERATION,VELOCITY,AND LOCATION OF THE
C     PROJECTILE
C
   11 IF ( IH.EQ.1 ) CALL PSABOT(U4L,P4R,P3R,IH)
```

```
   10 AP=0.5*(P3L+P4L-P3R-P4R)*AREA*GO/MP
      UMISLE=U3L+AP*DT
      U4L=UMISLE
      U=0.5*(U3L+U4L)
      LO=1.0/U
      DX=DT*GN/LO
       X4L=X3L+DX
      IF (IH.EQ.1 ) GO TO 20
      X4R=X3R+DX
C
C     LOCATE POINT 1 AND DETERMINE COEFFICIENTS ALONG LINE 14
C
   20 G=G1
      IF ( ITER.GT.0) GO TO 30
      U4L=U1
      P4L=P1
      R4L=R1
   30 U=0.5*(U1+U4L)
      P=0.5*(P1+P4L)
      R=0.5*(R1+R4L)
      CALL THERMO (P,R,A)
      LP=1.0/(U+A)
      X1=X4L-DT*GN/LP
      IF (ABS(X1-X1S).LT.0.0001) GO TO 40
      X1S=X1
      U1=MUL*X1+BUL
      P1=MPL*X1+BPL
      R1=MRL*X1+BRL
      GO TO 20
   40 QP=R*A/GO
      TP=GL*P1+QP*U1
      IF ( IH.EQ.1 ) GO TO 70
C
C     LOCATE POINT 2 AND DETERMINE COEFFICIENTS ALONG LINE 24
C
   50 G=G2
      IF ( ITER.GT.0) GO TO 60
      U4R=U2
      P4R=P2
      R4R=R2
   60 U=0.5*(U2+U4R)
      P=0.5*(P2+P4R)
      R=0.5*(R2+R4R)
      CALL THERMO (P,R,A)
      LM=1.0/(U-A)
      X2=X4R-DT*GN/LM
      IF (ABS(X2-X2S).LT.0.0001) GO TO 70
      X2S=X2
      U2=MUR*X2+BUR
      P2=MPR*X2+BPR
```

```
      R2=ARR*X2+BRR
      GO TO 50
   70 QM=R*A/GO
      TM=GL*P2-QM*U2
C
C     DETERMINE COEFFICIENTS ALONG LINE 34
C
      IF ( ITER.GT.0) GO TO 80
      P4L=P3L
      R4L=R3L
      IF ( IH.EQ.1 ) GO TO 80
      P4R=P3R
      R4R=R3R
   80 PL=0.5*(P3L+P4L)
      RL=0.5*(R3L+R4L)
      G=G1
      CALL THERMO (PL,RL,AL)
      AOL=AL**2/GO
      TOL=GL*P3L-AOL*R3L
      PR=0.5*(P3R+P4R)
      RR=0.5*(R3R+R4R)
      IF ( IH.EQ.1 ) GO TO 95
      G=G2
      CALL THERMO (PR,RR,AR)
      AOR=AR**2/GO
      TOR=GL*P3R-AOR*R3R
C
C     CALCULATE THE PROPERTIES AT POINT 4 AND TEST FOR
C     CONVERGENCE
C
      U4R=UNISLE
      P4R=(TM+QM*U4R)/GL
      R4R=(GL*P4R-TOR)/AOR
   95 FAR=0.0
      U4L=UNISLE
      P4L=(TP-QP*U4L)/GL
      R4L=(GL*P4L-TOL)/AOL
      IF ( ITER.EQ.ICOR) RETURN
      IF ( ITER.EQ.0) GO TO 90
      IF ((ABS(X4L-XD).GT.E4).OR.(ABS(U4L-UD).GT.E1*UD))
     $ GO TO 90
      IF ((ABS(P4L-PDL).GT.E2*PDL).OR.(ABS(R4L-RDL).GT.E3*RDL))
     $ GO TO 90
      IF ( IH.EQ.1 ) GO TO 90
      IF ((ABS(R4R-RDR).LT.E2*RDR).AND.(ABS(R4L-RDL).LT.E3*RDL
     $ )) RETURN
   90 ITER=ITER+1
      XD=X4L
      UD=U4L
      PDL=P4L
```

```
RDL=R4L
IF ( IH.EQ.1 ) GO TO 10
PDR=P4R
RDR=R4R
GO TO 10
END
```

```
      SUBROUTINE SHCK
C
C     ----------------------------------------------------------------
C
C     THIS SUBROUTINE CALCULATES THE PROPERTIES ACROSS
C     A SHOCK WAVE
C
C     ----------------------------------------------------------------
C
C     DEFINE QUASI-STEADY SHOCK WAVE PROPERTIES
      REAL LP,LM,LO,MUL,MPL,MRL,MUR,MPR,MRR,MP
      COMMON/CONTRL/ ICOR,IE,E1,E2,E3,E4,GO,GL,GM,G1,G2,RG1,RG2
      COMMON /DO/ XBRL(70),UBRL(70,2),PBRL(70,2),RBRL(70,2)
      COMMON /D1/ X4,U4,P4,R4,X5,U5,P5,R5,X6,U6,P6,R6,X7,U7,P7
     $ ,R7,PA,ME
      COMMON /D2/ W3,W4,X8,U8,P8,R8,X3L,U3L,P3L,R3L,X3R,U3R,
     $ P3R,R3R,X4L,U4L,P4L,R4L,X4R,U4R,P4R,XP,AREA,MP
      G=G1
      V1=W4
      CALL THERMO ( P4R,R4R,A1 )
      M1=V1/A1
C     CALCULATE PROPERTIES RATOS ACROSS SHOCK WAVE  FOR A
C     PERFECT GAS
      V2V1=(2.0+(G-1.0)*M1**2)/((G+1.0)*M1**2)
      R2R1 =1.0/V2V1
      P2P1=2.0*G*M1**2/(G+1.0)-(G-1.0)/(G+1.0)
C     CALCULATES PROPERTIES BEHIND QUASI -STEADY SHOCK WAVE
      V2=V2V1*V1
      P4L=P2P1*P4R
      R4L=R2R1*R4R
      U4L=W4+U4R-V2
      RETURN
      END
```

```
      SUBROUTINE SHOCK ( DT, X3 )
C
C     ----------------------------------------------------------
C
C     THIS SUBROUTINE CALCULATES THE FLOW PROPERTIES AT THE
C     LEFT AND THE RIGHT SIDES OF A SHOCK WAVE
C
C     ----------------------------------------------------------
C
      REAL LP,LM,LO,MUL,MPL,MRL,MUR,MPR,MRR,MP
      COMMON/CONTRL/ ICOR,IE,E1,E2,E3,E4,GO,GL,GM,G1,G2,RG1,RG2
      COMMON /D0/ XBRL(70),UBRL(70,2),PBRL(70,2),RBRL(70,2)
      COMMON /D1/ X4,U4,P4,R4,X5,U5,P5,R5,X6,U6,P6,R6,X7,U7,P7
     $ ,R7,PA,ME
      COMMON /D2/ W3,W4,X8,U8,P8,R8,X3L,U3L,P3L,R3L,X3R,U3R,
     $ P3R,R3R,X4L,U4L,P4L,R4L,X4R,U4R,P4R,XP,AREA,MP
C     DEFINE INITIAL PROPERTIES AND DETERMINE INTERPOLATING
C     POLYNOMIALS
      ITER=0.0
      X1S=X8
      U1=U8
      P1=P8
      R1=R8
      DX=X8-X3
      MUL=(U8-U3L)/DX
      BUL=U3L-MUL*X3
      MPL=(P8-P3L)/DX
      BPL=P3L-MPL*X3
      MRL=(R8-R3L)/DX
      BRL=R3L-MRL*X3
C     LOCATE POINT 4
   10 IF ( ITER.EQ.0 ) W4=W3
      W=0.5*(W3+W4)
      LO=1./W
      X4=X3+DT*GM/LO
C     CALL INTER AND SHCK TO DETERMINE THE SOLUTION AT POINTS
C     4L AND 4R
      CALL POINT ( DT )
      U4R=U4
      P4R=P4
      R4R=R4
      CALL SHCK
C     LOCATE POINT 1 AND DETERMINE THE COEFFICIENTS ALONG
C     LINE 14
   30 U=0.5*(U4L+U1)
      P=0.5*(P4L+P1)
      R=0.5*(R4L+R1)
      CALL THERMO ( P,R,A )
      LP=1.0/(U+A)
      X1=X4-DT*GM/LP
```

```
      IF ( ABS(X1-X1S).LT.0.0001) GO TO 40
      X1S=X1
      U1=AUL*X1+BUL
      P1=APL*X1+BPL
      R1=ARL*X1+BRL
      GO TO 30
   40 QP=R*A/GO
      TP =GL*P1+QP*U1
C     CALCULATE P4RRO AND TEST FOR CONVERGENCE
      P4RRO= (TP-QP*U4L)/GL
      DP=P4L-P4RRO
      IF ( ABS (DP ).LT.E2*P4L) RETURN
      ITER=ITER+1
      IF ( ITER.GT.ICOR ) RETURN
      IF ( ITER.GT.1 ) GO TO 50
      W4I=W4
      DPI=DP
      W4=1.01*W4
      GO TO 10
   50 SL= (DP-DPI)/(W4-W4I)
      W4I=W4
      DPI=DP
      W4=W4-DP/SL
      GO TO 10
      END
```

```
      SUBROUTINE THERMO (P,R,A)
C
C     ----------------------------------------------------------
C
C     SUBROUTINE THERMO CALCULATES THE SPEED OF SOUND FOR
C     A PERFECT GAS
C
C     ----------------------------------------------------------
C
      COMMON/CONTRL/ ICOR,IE,E1,E2,E3,E4,G0,GL,GN,G1,G2,RG1,RG2
      A=SQRT(1.4*P/R)
      RETURN
      END
```

```
      SUBROUTINE POINT ( DT )
C
C     ----------------------------------------------------------------
C
C     SUBROUTINE INTER CALCULATES THE SOLUTION AT AN INTERIOR
C     POINT
C
C     ----------------------------------------------------------------
C
C
      REAL LP,LM,IO,MUL,MPL,MRL,MUR,MPR,MRR,MP
      COMMON/CONTRL/ ICOR,IE,E1,E2,E3,E4,GO,GL,GN,G1,G2,RG1,RG2
      COMMON /D1/ X4,U4,P4,R4,X5,U5,P5,R5,X6,U6,P6,R6,X7,U7,P7
     $ ,R7,PA,ME
C     DEFINE INITIAL PROPERTIES AND DETERMINE INTERPOLATING
C     POLYNOMIALS
      ITER=0
      X1S=X5
      X2S=X7
      X3S=X6
      U1=U5
      P1=P5
      R1=R5
      U2=U7
      P2=P7
      R2=R7
      U3=U6
      P3=P6
      R3=R6
      DX=X5-X6
      MUL=(U5-U6)/DX
      BUL=U6-MUL*X6
      MPL=(P5-P6)/DX
      BPL=P6-MPL*X6
      MRL=(R5-R6)/DX
      BRL=R6-MRL*X6
      DX=X6-X7
      MUR=(U6-U7)/DX
      BUR=U7-MUR*X7
      MPR=(P6-P7)/DX
      BPR=P7-MPR*X7
      MRR=(R6-R7)/DX
      BRR=R7-MRR*X7
C
C     LOCATE POINT 1 AND DETERMINE COEFFICIENTS ALONG LINE 14
C
   10 IF ( ITER.GT.0) GO TO 20
      U4=U1
      P4=P1
      R4=R1
```

```
   20 U=0.5*(U1+U4)
      P=0.5*(P1+P4)
      R=0.5*(R1+R4)
      CALL THERMO (P,R,A)
      LP=1.0/(U+A)
      X1=X4-DT*GN/LP
      IF(ABS(X1-X1S).LT.0.0001) GO TO 40
      X1S=X1
      IF (X1.GT.X6) GO TO 30
      U1=MUL*X1+BUL
      P1=MPL*X1+BPL
      R1=MRL*X1+BRL
      GO TO 10
   30 U1=MUR*X1+BUR
      P1=MPR*X1+BPR
      R1=MRR*X1+BRR
      GO TO 10
   40 QP=R*A/GO
      TP=GL*P1+QP*U1
C
C     LOCATE POINT 2 AND DETERMINE COEFFICIENTS ALONG 24
C
   50 IF (ITER.GT.0) GO TO 60
      U4=U2
      P4=P2
      R4=R2
   60 U=0.5*(U2+U4)
      P=0.5*(P2+P4)
      R=0.5*(R2+R4)
      CALL THERMO (P,R,A)
      LM=1.0/(U-A)
      X2=X4-DT*GN/LM
      IF (ABS(X2-X2S).LT.0.0001) GO TO 80
      X2S=X2
      IF (X2.GT.X6) GO TO 70
      U2=MUL*X2+BUL
      P2=MPL*X2+BPL
      R2=MRL*X2+BRL
      GO TO 50
   70 U2=MUR*X2+BUR
      P2=MPR*X2+BPR
      R2=MRR*X2+BRR
      GO TO 50
   80 QM=R*A/GO
      TM=GL*P2-QM*U2
C
C     LOCATE POINT 3 AND DETERMINE COEBFICIENTS ALONG LINE 34
C
   90 IF ( ITER.GT.0) GO TO 100
      U4=U3
```

```
      P4=P3
      R4=R3
  100 U=0.5*(U3+U4)
      P=0.5*(P3+P4)
      R=0.5*(R3+R4)
      CALL THERMO (P,R,A)
      IF (ABS(U).GT.1.0E-6) GO TO 110
      LO=0.0
      X3=X4
      GO TO 120
  110 LO=1.0/U
      X3=X4-DT*GM/LO
  120 IF (ABS (X3-X3S).LT.0.0001) GO TO 140
      X3S=X3
      IF(X3.GT.X6) GO TO 130
      U3=MUL*X3+BUL
      P3=MPL*X3+BPL
      R3=MRL*X3+BRL
      GO TO 90
  130 U3=MUR*X3+BUR
      P3=MPR*X3+BPR
      R3=MRR*X3+BRR
      GO TO 90
  140 AO=A**2/GO
      TO=GL*P3-AO*R3
C
C     CALCULATE THE PROPERTIES AT POINT 4,AND TEST FOR
C     CONVERGENCE
C
      U4=(TP-TM)/(QP+QM)
      P4=(TP-QP*U4)/GL
      R4=(GL*P4-TO)/AO
      IF ( ITER.EQ.ICOR) RETURN
      IF ( ITER.EQ.0) GO TO 150
      IF ((ABS(U4-UD).GT.E1*UD).OR.(ABS(P4-PD).GT.E2*PD))
     $ GO TO 150
      IF ( ABS(R4-RD).LT.E3*RD) RETURN
  150 ITER=ITER+1
      UD=U4
      PD=P4
      RD=R4
      GO TO 10
      END
```

```
      SUBROUTINE SHOW2 ( N )
C
C     ------------------------------------------------------
C
C     THIS SUBROUTINE PLOTS THE PROJECTILE CHARACTERISTICS
C
C     ------------------------------------------------------
C
      COMMON /D5/ XPROJ(1010),UPROJ(1010),PBPROJ(1010),
     $ XBPROJ(1010),XSHOCK(1010),USHOCK(1010),TIME(1010),
     $ XCAR(1010),XFPROJ(1010)
      CALL VTHICK(3)
      CALL VECTOR(0.0,0.0,0.0,6.0)
      CALL VECTOR(-0.5,0.0,-0.5,6.0)
      CALL VECTOR(-1.0,0.0,-1.0,6.0)
      CALL VECTOR(-1.0,6.0,4.5,6.0)
      CALL VECTOR(4.5,6.0,4.5,0.0)
      CALL VECTOR(4.5,0.0,-1.0,0.0)
      CALL NSCALE(TIME,4.0,N,1,1)
      CALL NSCALE(XSHOCK,5.0,N,1,1)
      TIME(N+1)=0.000
      TIME(N+2)=0.001
      CALL AXIS(0.0,0.0,'TIME (S)',-8,4.0,0.0,TIME(N+1),
     $ TIME(N+2))
      CALL AXIS(0.0,0.0,'DISPLACEMENT (M)',16,5.0,90.0,
     $ XSHOCK(N+1),XSHOCK(N+2))
      CALL LINE(TIME,XSHOCK,N,1,0,0)
      CALL VTHICK(0)
      CALL NSCALE(TIME,4.0,N,1,1)
      CALL NSCALE(USHOCK,5.0,N,1,1)
      TIME(N+1)=0.000
      TIME(N+2)=0.001
      CALL AXIS(0.0,0.0,'TIME (S)',-8,4.0,0.0,TIME(N+1),
     $ TIME(N+2))
      CALL VTHICK(3)
      CALL AXIS(-0.5,0.0,'VELOCITY (M/S)',14,5.0,90.0,
     $ USHOCK(N+1),USHOCK(N+2))
      CALL LINE(TIME,USHOCK,N,1,0,0)
      CALL EOPLOT(1)
      RETURN
      END
```

```
      SUBROUTINE SHOW ( N )
C
C     ----------------------------------------------------------------
C
C     THIS SUBROUTINE PLOTS A PROJECTILE VARIABLE VESUS TIME
C
C     ----------------------------------------------------------------
C
      COMMON /D5/ XPROJ(1010),UPROJ(1010),PBPROJ(1010),
     $ XBPROJ(1010),XSHOCK(1010),USHOCK(1010),TIME(1010),
     $ XCAR(1010),XFPROJ(1010)
      CALL VTHICK(3)
      CALL VECTOR(0.0,0.0,0.0,6.0)
      CALL VECTOR(-0.5,0.0,-0.5,6.0)
      CALL VECTOR(-1.0,0.0,-1.0,6.0)
      CALL VECTOR(-1.0,6.0,4.5,6.0)
      CALL VECTOR(4.5,6.0,4.5,0.0)
      CALL VECTOR(4.5,0.0,-1.0,0.0)
      CALL NSCALE(TIME,4.0,N,1,1)
      CALL NSCALE(XPROJ,5.0,N,1,1)
      TIME(N+1)=0.000
      TIME(N+2)=0.001
      TIME(N+2)=0.003
      CALL AXIS(0.0,0.0,'TIME (S)',-8,4.0,0.0,TIME(N+1),
     $ TIME(N+2))
      CALL AXIS(0.0,0.0,'DISPLACEMENT (M)',16,5.0,90.0,
     $ XPROJ(N+1),XPROJ(N+2))
      CALL LINE(TIME,XPROJ,N,1,0,0)
      CALL VTHICK(0)
      CALL NSCALE(TIME,4.0,N,1,1)
      CALL NSCALE(UPROJ,5.0,N,1,1)
      TIME(N+1)=0.000
      TIME(N+2)=0.001
      TIME(N+2)=0.003
      CALL AXIS(0.0,0.0,'TIME (S)',-8,4.0,0.0,TIME(N+1),
     $ TIME(N+2))
      CALL VTHICK(3)
      CALL AXIS(-0.5,0.0,'VELOCITY (M/S)',14,5.0,90.0,
     $ UPROJ(N+1),UPROJ(N+2))
      CALL LINE(TIME,UPROJ,N,1,0,0)
      CALL VTHICK(0)
      CALL NSCALE(TIME,4.0,N,1,1)
      CALL NSCALE(PBPROJ,5.0,N,1,1)
      TIME(N+1)=0.000
      TIME(N+2)=0.001
      TIME(N+2)=0.003
      CALL AXIS(0.0,0.0,'TIME (S)',-8,4.0,0.0,TIME(N+1),
     $ TIME(N+2))
      CALL VTHICK(3)
      CALL AXIS(-1.0,0.0,'PRESSURE (KPA)',14,5.0,90.0,
```

```
$ PBPROJ(N+1),PBPROJ(N+2))
$ PBPROJ(N+2))
 CALL LINE(TIME,PBPROJ,N,1,0,0)
 CALL EOPLOT(1)
 RETURN
 END
```

```
      SUBROUTINE INTER(LS,LF,MS,MF)
C
C     ---------------------------------------------------
C
C     THIS SUBROUTINE CALCULATES THE INTERIOR MESH POINTS
C
C     ---------------------------------------------------
C
      COMMON/D6/ U(50,20,2),V(50,20,2),P(50,20,2),RO(50,20,2)
      COMMON/D7/ LMAX,MMAX,MCB,LCB,LBR,N,N1,N3,L1,L2,M1,M2,
     $ MCB1,MCB2,LCB1,LCB2,LBR1,LBR2,GAMMA,DX,DY,DT,ICHAR,DXR,
     $ DYR,RSTAR,RSTARS,G,PC,LC,PLOW,ROLOW,RG
      COMMON/D8/ XP1(50,20),YP1(50,20),KSI(2000),ETA(2000),
     $ PX(2000),PY(2000),XW(50),YW(20),NXNY(20),XE1,
     $ NDIM,PDS,AP
      COMMON/D9/CAV,XMU,XLA,RKMU,XRO,QUT(50,20),QVT(50,20),
     $ QPT(50,20),QROT(50,20),PE(20)
      REAL NXNY,NYNX,LC,KSI
      AL=0.0
      BE=1.0
      DE=0.0
      DP=0.0
      ATERM=0.
      IF(ICHAR.NE.1) GO TO 40
C
C     COMPUTE THE TENTATIVE SOLUTION AT T+DT
C
      DO 30 L=LS,LF
      W=SQRT(1.+NXNY(L)*NXNY(L))
      DO 30 M=MS,MF
      WY=SQRT(1.+NYNX(M)+NYNX(M))
      UB=U(L,M,N1)
      VB=V(L,M,N1)
      PB=P(L,M,N1)
      ROB=RO(L,M,N1)
C
      ASB=P(L,M,N3)*GAMMA/RO(L,M,N3)
C
      U1B=W*UB+DP
      IF(M.NE.1) GO TO 10
C
C     COMPUTATION OF THE MIDPLANE MESH POINTS
C
C
C     COMPUTAION OF THE DERIVATIVES WITH RESPECT TO X
C
      DUDX=(UB-U(L-1,M,N1))*DXR
      DPDX=(PB-P(L-1,M,N1))*DXR
      DRODX=(ROB-RO(L-1,M,N1))*DXR
C
```

```
C     COMPUTAION OF THE DERIVATIVES WITH RESPECT TO X
C
      DVDY= (4.0*V (L,2,N1) -V (L,3,N1))*.5*DYR
      V (L,M,N3)=0.0
      URHS=-U1B*DUDX-W*DPDX/ROB+QUT (L,M)
      RORHS=-U1B*DRODX-ROB*W*DUDX-FLOAT(1+NDIM) *ROB*DVDY
     $ *BE+QROT(L,M)
      PRHS=-U1B*DPDX+ASB* (RORHS+U1B*DRODX) +QPT (L,M)
      GO TO 20
   10 IF (NDIM.EQ.1) ATERM=ROB*VB/YW(M)
C
C     COMPUTAION OF THE DERIVATIVES WITH RESPECT TO X
C
      UVB=UB*AL+ VB*WY+DE
      DUDX= (UB-U (L-1,M,N1)) *DXR
      DVDX= (VB-V (L-1,M,N1)) *DXR
      DPDX=(PB-P (L-1,M,N1)) *DXR
      DRODX= (ROB-RO(L-1,M,N1))*DXR
      DUDY= (UB-U (L,M-1,N1)) *DYR
      DVDY=(VB-V (L,M-1,N1)) *DYR
      DPDY= (PB-P (L,M-1,N1)) *DYR
      DRODY=(ROB-RO(L,M-1,N1)) *DYR
      URHS=-U1B*DUDX-UVB*DUDY-(W*DPDX+AL*DPDY)/ROB+QUT (L,M)
      VRHS=-U1B*DVDX-UVB*DVDY-WY*DPDY/ROB+QVT (L,M)
      RORHS=-U1B*DRODX-UVB*DRODY-ROB*(W*DUDX+AL*DUDY+WY*DVDY)
     $ -ATERM+QROT (L,M)
      PRHS=-U1B*DPDX-UVB*DPDY+ASB* (RORHS+U1B*DRODX+UVB*DRODY)
     $ +QPT(L,M)
      V(L,M,N3)=V(L,M,N1) +VRHS*DT
   20 U(L,M,N3)=U(L,M,N1) +URHS*DT
      P (L,M,N3)=P(L,M,N1) +PRHS*DT
      RO(L,M,N3) =RO(L,M,N1) +RORHS*DT
      IF(P(L,M,N3).LE.0.0)  P(L,M,N3)=PLOW
      IF(RO(L,M,N3).LE.0.0)  RO(L,M,N3)=ROLOW
   30 CONTINUE
      RETURN
C
C     COMPUTE THE FINAL SOLUTION AT T+DT
C
   40 DO 70 L=LS,LF
      W=SQRT(1.+WXWY(L)*WXWY(L))
      DO 70 M=MS,MF
      WY=SQRT(1.+WYWX(M) *WYWX(M))
      UB=U(L,M,N3)
      VB=V(L,M,N3)
      PB=P (L,M,N3)
      ROB=RO(L,M,N3)
C
      ASB=P(L,M,N3)*GAMMA/RO (L,M,N3)
C
```

```
      U1B=W*UB+DP
      IF(M.NE.1)  GO TO 50
      DUDX=(U(L+1,M,N3)-UB)*DXR
      DVDX=(V(L+1,M,N3)-VB)*DXR
      DPDX=(P(L+1,M,N3)-PB)*DXR
      DRODX=(RO(L+1,M,N3)-ROB)*DXR
      DVDY=(4.0*V(L,2,N3)-V(L,3,N3))*.5*DYR
      V(L,M,N3)=0.0
      URHS=-U1B*DUDX-W*DPDX/ROB+QUT(L,M)
      RORHS=-U1B*DRODX-ROB*W*DUDX-FLOAT(1+NDIM)*ROB*DVDY
     $ *BE+QROT(L,M)
      PRHS=-U1B*DPDX+ASB*(RORHS+U1B*DRODX)+QPT(L,M)
      GO TO 60
C
   50 IF  (NDIM.EQ.1)  ATERM=ROB*VB/YN(M)
      UVB=UB*AL+VB*NY+DE
      DUDX=(U(L+1,M,N3)-UB)*DXR
      DPDX=(P(L+1,M,N3)-PB)*DXR
      DVDX=(V(L+1,M,N3)-VB)*DXR
      DRODX=(RO(L+1,M,N3)-ROB)*DXR
C
      DUDY=(U(L,M+1,N3)-UB)*DYR
      DVDY=(V(L,M+1,N3)-VB)*DYR
      DPDY=(P(L,M+1,N3)-PB)*DYR
      DRODY=(RO(L,M+1,N3)-ROB)*DYR
C
      URHS=-U1B*DUDX-UVB*DUDY-(W*DPDX+AL*DPDY)/ROB+QUT(L,M)
      VRHS=-U1B*DVDX-UVB*DVDY-NY*DPDY/ROB+QVT(L,M)
      RORHS=-U1B*DRODX-UVB*DRODY-ROB*(W*DUDX+AL*DUDY+NY*DVDY)
     $ -ATERM+QROT(L,M)
      PRHS=-U1B*DPDX-UVB*DPDY+ASB*(RORHS+U1B*DRODX+UVB*DRODY)
     $ +QPT(L,M)
      V(L,M,N3)=(V(L,M,N1)+V(L,M,N3)+VRHS*DT)*.5
   60 U(L,M,N3)=(U(L,M,N1)+U(L,M,N3)+URHS*DT)*.5
      P(L,M,N3)=(P(L,M,N1)+P(L,M,N3)+PRHS*DT)*.5
      RO(L,M,N3)=(RO(L,M,N1)+RO(L,M,N3)+RORHS*DT)*.5
      IF(P(L,M,N3).LE.0.0)  P(L,M,N3)=PLOW
      IF(RO(L,M,N3).LE.0.0)  RO(L,M,N3)=ROLOW
   70 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BDRY ( UB )
C
C     ------------------------------------------------------------
C
C     THIS SUBROUTINE CALCULATES THE BOUNDARY MESH POINTS
C
C     ------------------------------------------------------------
C
      COMMON/D6/ U(50,20,2),V(50,20,2),P(50,20,2),RO(50,20,2)
      COMMON/D7/ LMAX,MMAX,MCB,LCB,LBR,N,N1,N3,L1,L2,M1,M2,
     $ MCB1,MCB2,LCB1,LCB2,LBR1,LBR2,GAMMA,DX,DY,DT,ICHAR,DXR,
     $ DYR,RSTAR,RSTARS,G,PC,LC,PLOW,ROLOW,RG
      COMMON/D8/ XP1(50,20),YP1(50,20),KSI(2000),ETA(2000),
     $ PX(2000),PY(2000),XW(50),YW(20),NXNY(20),XE1,
     $ NDIM,PDS,AP
      COMMON/D9/CAV,XMU,XLA,RKMU,XRO,QUT(50,20),QVT(50,20),
     $ QPT(50,20),QROT(50,20),PE(20)
      REAL NXNY,NYNX,LC,KSI
      DO 22 L=LCB2,LBR1
      U(L,MCB,N3)=0.0
   22 CONTINUE
      DO 1 M=MCB,M1
      U(1,M,N3)=0.0
      V(1,M,N3)=V(2,M,N3)
      P(1,M,N3)=P(2,M,N3)
      RO(1,M,N3)=RO(2,M,N3)
    1 CONTINUE
      DO 3 M=1,MCB
      U(LCB,M,N3)=-UB
      V(LCB,M,N3)=V(LCB2,M,N3)
      P(LCB,M,N3)=P(LCB2,M,N3)
      RO(LCB,M,N3)=RO(LCB2,M,N3)
    3 CONTINUE
      DO 4 M=MCB2,MMAX
      U(LBR,M,N3)=0.0
      V(LBR,M,N3)=V(LBR1,M,N3)
      P(LBR,M,N3)=P(LBR1,M,N3)
      RO(LBR,M,N3)=RO(LBR1,M,N3)
    4 CONTINUE
      DO 5 L=2,LCB1
      U(L,MCB,N3)=-UB
      V(L,MCB,N3)=0.0
      P(L,MCB,N3)=P(L,MCB2,N3)
      RO(L,MCB,N3)=RO(L,MCB2,N3)
    5 CONTINUE
      DO 6  L=1,LBR
      U(L,MMAX,N3)=U(L,M1,N3)
      V(L,MMAX,N3)=0.0
      P(L,MMAX,N3)=P(L,M1,N3)
      RO(L,MMAX,N3)=RO(L,M1,N3)
```

```
    6 CONTINUE
      DO 7 L=LBR,L1
      U(L,MCB,N3)=U(L,MCB1,N3)
      P(L,MCB,N3)=P(L,MCB1,N3)
      RO(L,MCB,N3)=RO(L,MCB1,N3)
      V(L,MCB,N3)=0.0
    7 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SMOOTH(LS,LF,MS,KMAX)
C
C     --------------------------------------------------------
C
C     THIS SUBROUTINE IS USED TO SMOOTH THE FLOW VARIABLES
C
C     --------------------------------------------------------
C
      COMMON/D6/ U(50,20,2),V(50,20,2),P(50,20,2),RO(50,20,2)
      COMMON/D7/ LMAX,MMAX,MCB,LCB,LBR,N,N1,N3,L1,L2,M1,M2,
     $ MCB1,MCB2,LCB1,LCB2,LBR1,LBR2,GAMMA,DX,DY,DT,ICHAR,DXR,
     $ DYR,RSTAR,RSTARS,G,PC,LC,PLOW,ROLOW,RG
      COMMON/D8/ XP1(50,20),YP1(50,20),KSI(2000),ETA(2000),
     $ PX(2000),PY(2000),XW(50),YW(20),NXNY(20),XE1,
     $ NDIM,PDS,AP
      COMMON/D9/CAV,XMU,XLA,RKMU,XRO,QUT(50,20),QVT(50,20),
     $ QPT(50,20),QROT(50,20),PE(20)
      REAL NXNY,NYNX,LC,KSI
C
      K1=KMAX-1
      J=MS-1
      SMP=0.50
      SMP=0.15
      SMP=0.95
      SMP=0.25
      SMP4=0.25*(1.0-SMP)
      DO 20 L=LS,LF
      U(L,KMAX,N3)=SMP4*(U(L-1,KMAX,N3)+U(L+1,KMAX,N3)
     $ +2.0*U(L,K1,N3))+SMP*U(L,KMAX,N3)
      V(L,KMAX,N3)=-U(L,KMAX,N3)*NXNY(L)
      P(L,KMAX,N3)=SMP4*(P(L-1,KMAX,N3)+P(L+1,KMAX,N3)
     $ +2.0*P(L,K1,N3))+SMP*P(L,KMAX,N3)
      RO(L,KMAX,N3)=SMP4*(RO(L-1,KMAX,N3)+RO(L+1,KMAX,N3)
     $ +2.0*RO(L,K1,N3))+SMP*RO(L,KMAX,N3)
      U(L,J,N3)=SMP4*(U(L-1,J,N3)+U(L+1,J,N3)+2.0*U(L,MS,N3))
     $ +SMP*U(L,J,N3)
      V(L,J,N3)=SMP4*(V(L-1,J,N3)+V(L+1,J,N3)+2.0*V(L,MS,N3))
     $ +SMP*V(L,J,N3)
      P(L,J,N3)=SMP4*(P(L-1,J,N3)+P(L+1,J,N3)+2.0*P(L,MS,N3))
     $ +SMP*P(L,J,N3)
      RO(L,J,N3)=SMP4*(RO(L-1,J,N3)+RO(L+1,J,N3)+2.0
     $ *RO(L,MS,N3))+SMP*RO(L,J,N3)
      DO 20 M=MS,K1
      U(L,M,N3)=SMP4*(U(L-1,M,N3)+U(L+1,M,N3)+U(L,M-1,N3)
     $ +U(L,M+1,N3))+SMP*U(L,M,N3)
      V(L,M,N3)=SMP4*(V(L-1,M,N3)+V(L+1,M,N3)+V(L,M-1,N3)
     $ +V(L,M+1,N3))+SMP*V(L,M,N3)
      P(L,M,N3)=SMP4*(P(L-1,M,N3)+P(L+1,M,N3)+P(L,M-1,N3)
     $ +P(L,M+1,N3))+SMP*P(L,M,N3)
      RO(L,M,N3)=SMP4*(RO(L-1,M,N3)+RO(L+1,M,N3)+RO(L,M-1,N3)
```

```
      $ RO(L,M+1,N3))+SMP*RO(L,M,N3)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PISTON ( VPOLD,XPOLD,VPNEW,XPNEW )
C
C     ---------------------------------------------------------
C
C     THIS SUBROUTINE CALCULATES THE PISTON VARIABLES
C
C     ---------------------------------------------------------
C
      COMMON/D6/ U(50,20,2),V(50,20,2),P(50,20,2),RO(50,20,2)
      COMMON/D7/ LMAX,MMAX,MCB,LCB,LBR,N,N1,N3,L1,L2,M1,M2,
     $ MCB1,MCB2,LCB1,LCB2,LBR1,LBR2,GAMMA,DX,DY,DT,ICHAR,DXR,
     $ DYR,RSTAR,RSTARS,G,PC,LC,PLOW,ROLOW,RG
      COMMON/D8/ XP1(50,20),YP1(50,20),KSI(2000),ETA(2000),
     $ PX(2000),PY(2000),XW(50),YW(20),MXNY(20),XE1,
     $ NDIM,PDS,AP
      COMMON/D9/CAV,XMU,XLA,RKMU,XRO,QUT(50,20),QVT(50,20),
     $ QPT(50,20),QROT(50,20),PE(20)
      REAL MXNY,NYMX,LC,KSI
      PA=1.031E+5
      PAV=P(LCB,1,N3)
      DO 1 M=2,MCB
      PAV=PAV+P(LCB,M,N3)
    1 CONTINUE
      PAV=PAV*6894.2/FLOAT(MCB)
      DELTAV=AP*(PAV-PA)*DT/(PDS*LC)
C     DETERMINE THE PISTON VELOCITY
      VPNEW=VPOLD+DELTAV
C     DETERMINE THE PISTON DISPLACEMENT
      XPNEW=XPOLD+VPOLD*DT/LC+DELTAV*DT/(2.0*LC)
      RETURN
      END
```

```
      SUBROUTINE GRAPH ( NP,T,VPNEW )
C
C     ----------------------------------------------------------
C
C     THIS SUBROUTINE PLOTS THE VELOCITY VECTORS
C     AND PLOTS THE PHYSICAL GRID
C     ----------------------------------------------------------
C
      COMMON/D6/ U(50,20,2),V(50,20,2),P(50,20,2),RO(50,20,2)
      COMMON/D7/ LMAX,MMAX,MCB,LCB,LBR,N,N1,N3,L1,L2,M1,M2,
     $ MCB1,MCB2,LCB1,LCB2,LBR1,LBR2,GAMMA,DX,DY,DT,ICHAR,DXR,
     $ DYR,RSTAR,RSTARS,G,PC,LC,PLOW,ROLOW,RG
      COMMON/D8/ XP1(50,20),YP1(50,20),KSI(2000),ETA(2000),
     $ PX(2000),PY(2000),XW(50),YW(20),MXNY(20),XE1,
     $ NDIM,PDS,AP
      COMMON/D9/CAV,XMU,XLA,RKMU,XRO,QUT(50,20),QVT(50,20),
     $ QPT(50,20),QROT(50,20),PE(20)
      REAL MXNY,NYNX,LC,KSI
      CALL RECT(-2.45,-2.55,8.5,11.0,0.0,3)
      DO 220 L=1,LMAX
      IF ( L.LT.LCB) GO TO 6
      IF ( L.LE.LBR ) GO TO 8
      J1=1
      JMAX=MCB
      GO TO 7
    8 MAR=0.0
      J1=1
      JMAX=MMAX
      GO TO 7
    6 MAR=0.0
      J1=MCB
      JMAX=MMAX
    7 CALL VECTOR ( XP1(L,J1),YP1(L,J1),XP1(L,JMAX),
     $ YP1(L,JMAX))
  220 CONTINUE
      DO 230 L=1,LBR1
      DO 230 M=MCB,MMAX
      CALL VECTOR ( XP1(L,M),YP1(L,M),XP1(L+1,M),YP1(L+1,M))
  230 CONTINUE
      DO 240 L=LCB,L1
      DO 240 M=1,MCB
      CALL VECTOR ( XP1(L,M),YP1(L,M),XP1(L+1,M),YP1(L+1,M))
  240 CONTINUE
      CALL RECT(0.0,0.0,YW(MCB),XW(LCB),0.0,3)
      CALL PATERN(0.0,0.0,YW(MCB),XW(LCB),0.03,3)
      FPN=FLOAT(N)
      FPN1=T*1000.
      XW(LMAX+1)=0.0
      YW(MMAX+1)=0.0
      XW(LMAX+2)=2.50
```

```
      YW(MMAX+2)=2.50
      CALL VTHICK(2)
      CALL AXIS ( 0.0,0.0,'AXIAL DISTANCE (CM.)',-20,7.0,0.0,
     $ XW(LMAX+1),XW(LMAX+2))
      CALL AXIS ( 0.0,0.0,'RADIUS (CM.)',12,3.0,90.0,
     $ YW(MMAX+1),YW(MMAX+2))
      CALL SYMBOL(0.00,-1.40,.14,' FIGURE   VELOCITY
     $ DISTRIBUTION IN THE MAIN CHAMBER',0.0,51)
      CALL SYMBOL(3.25,-.875,.14,'TIME STEP',0.0,9)
      CALL NUMBER(5.00,-.875,.14,FPN,0.0,-1)
      CALL SYMBOL(0.1,4.5,.14,'TIME=',0.0,5)
      CALL SYMBOL(2.0,4.5,.14,' (MS)',0.0,4)
      CALL NUMBER(1.1,4.5,.14,FPN1,0.0,2)
      CALL SYMBOL(0.1,5.0,.14,'PISTON VELOCITY=',0.0,16)
      CALL NUMBER(2.5,5.0,.14,VPNEW,0.0,2)
      CALL SYMBOL(3.25,5.0,.14,'(M/S)',0.0,5)
      CALL SYMBOL(0.1,4.0,.14,'CHARGE PRESSURE=5,000.KPA',
     $ 0.0,25)
      EPSIL=0.000001
      DO 2 L=1,LMAX
      DO 2 M=1,MMAX
      Q=U(L,M,NP)*U(L,M,NP)+V(L,M,NP)*V(L,M,NP)
      Q1=SQRT(Q)
      IF(L.EQ.1.AND.M.EQ.1) QMAX=Q1
      IF(Q1.GT.QMAX) QMAX=Q1
    2 CONTINUE
      DO 3 L=1,LMAX
      DO 3 M=1,MMAX
      IF(L.EQ.1.AND.M.EQ.1) GO TO 3
      Y1=YP1(L,M)
      X1=XP1(L,M)
      XT1=U(L,M,NP)*0.25/QMAX
      YT1=V(L,M,NP)*0.25/QMAX
      XT2=ABS(XT1)
      YT2=ABS(YT1)
      IF(XT2.LT.EPSIL.AND.YT2.LT.EPSIL) GO TO 3
      XT=XT1+X1
      YT=YT1+Y1
      CALL VTHICK(3)
      CALL AROHD(X1,Y1,XT,YT,.1,0.075,14)
    3 CONTINUE
      CALL VTHICK(1)
      CALL EOPLOT(1)
      RETURN
      END
```

```
      SUBROUTINE EXIT ( IEXTRA )
C
C     ----------------------------------------------------------
C
C     THIS SUBROUTINE CALCULATES THE EXIT MESH POINTS
C
C     ----------------------------------------------------------
C
      COMMON/D6/ U(50,20,2),V(50,20,2),P(50,20,2),RO(50, 20,2)
      COMMON/D7/ LMAX,MMAX,MCB,LCB,LBR,N,N1,N3,L1,L2,M1,M2,
     $ MCB1,MCB2,LCB1,LCB2,LBR1,LBR2,GAMMA,DX,DY,DT,ICHAR,DXR,
     $ DYR,RSTAR,RSTARS,G,PC,LC,PLOW,ROLOW,RG
      COMMON/D8/ XP1(50,20),YP1(50,20),KSI(2000),ETA(2000),
     $ PX(2000),PY(2000),XW(50),YW(20),MXNY(20),XE1,
     $ NDIM,PDS,AP
      COMMON/D9/CAV,XMU,XLA,RKMU,XRO,QUT(50,20),QVT(50,20),
     $ QPT(50,20),QROT(50,20),PE(20)
      REAL MXNY,NINX,LC,KSI
      ME=MCB
      AL=0.0
      BE=1.0
      DE=0.0
      AL1=0.0
      BE1=1.0
      DE1=0.0
      IEX=0.0
      X3=XE1
      DO 1 M=1,ME
      PE(M)=PE(1)
    1 CONTINUE
      ATERM2=0.0
      ATERM3=0.0
      DO 180 M=1,ME
      IF(IEXTRA.EQ.1) GO TO 10
      AS=P(LMAX,M,N1)*GAMMA/RO(LMAX,M,N1)
      A1=SQRT(AS)
      IF (IEXTRA.EQ.2) GO TO 20
      Q=SQRT (U(LMAX,M,N1)*U(LMAX,M,N1)+V(LMAX,M,N1)*
     $ V(LMAX,M,N1))
      IF(Q/A1.LT.1.0) GO TO 20
   10 U(LMAX,M,N3)=U(L1,M,N3)+FLOAT(IEX)*(U(L1,M,N3)-
     $ U(L2,M,N3))
      V(LMAX,M,N3)=V(L1,M,N3)+FLOAT(IEX)*(V(L1,M,N3)-
     $ V(L2,M,N3))
      P(LMAX,M,N3)=P(L1,M,N3)+FLOAT(IEX)*(P(L1,M,N3)-
     $ P(L2,M,N3))
      RO(LMAX,M,N3)=RO(L1,M,N3)+FLOAT(IEX)*(RO(L1,M,N3)-
     $ RO(L2,M,N3))
      GO TO 180
   20 MAR=0.0
```

```
      U1=U(LMAX,M,N1)
      U2=U1
      A2=A1
      IF(ICHAR.NE.1) GO TO 30
      U(LMAX,M,N3)=U1
      RO(LMAX,M,N3)=RO(LMAX,M,N1)
      A3=A1
C
C     CALCULATE THE PROPERTY INTERPOLATION POLYNOMIAL
C     COEFFICIENTS
C
   30 BU=(U(LMAX,M,N1)-U(L1,M,N1))*DXR
      BV=(V(LMAX,M,N1)-V(L1,M,N1))*DXR
      BP=(P(LMAX,M,N1)-P(L1,M,N1))*DXR
      BRO=(RO(LMAX,M,N1)-RO(L1,M,N1))*DXR
      BAL=(AL-AL1)*DXR
      BBE=(BE-BE1)*DXR
      BDE=(DE-DE1)*DXR
      CU=U(LMAX,M,N1)-BU*X3
      CV=V(LMAX,M,N1)-BV*X3
      CP=P(LMAX,M,N1)-BP*X3
      CRO=RO(LMAX,M,N1)-BRO*X3
      CAL=AL-BAL*X3
      CBE=BE-BBE*X3
      CDE=DE-BDE*X3
C
C     CALCULATE THE CROSS DERIVATIVE INTERPOLATING POLYNOMIAL
C     COEFFICIENTS
C
C
      IF(M.EQ.1) GO TO 40
      DU=(U(LMAX,M,N1)-U(LMAX,M-1,N1))*DYR
      DV=(V(LMAX,M,N1)-V(LMAX,M-1,N1))*DYR
      DP=(P(LMAX,M,N1)-P(LMAX,M-1,N1))*DYR
      DRO=(RO(LMAX,M,N1)-RO(LMAX,M-1,N1))*DYR
      DU1=(U(L1,M,N1)-U(L1,M-1,N1))*DYR
      DV1=(V(L1,M,N1)-V(L1,M-1,N1))*DYR
      DP1=(P(L1,M,N1)-P(L1,M-1,N1))*DYR
      DRO1=(RO(L1,M,N1)-RO(L1,M-1,N1))*DYR
      GO TO 60
   40 DU=0.0
      DV=(4.0*V(LMAX,2,N1)-V(LMAX,3,N1))*0.5*DYR
      DP=0.0
      DRO=0.0
      DU1=0.0
      DV1=(4.0*V(L1,2,N1)-V(L1,3,N1))*0.5*DYR
      DP1=0.0
      DRO1=0.0
   60 BDU=(DU-DU1)*DXR
      BDV=(DV-DV1)*DXR
```

```
      BDP= (DP-DP1) *DXR
      BDRO=(DRO-DRO1) *DXR
      CDU=DU-BDU*X3
      CDV=DV-BDV*X3
      CDP=DP-BDP*X3
      CDRO=DRO-BDRO*X3
C
C     CALCULATE X1 AND X2
C
      IF(ICHAR.EQ.1)  GO TO 70
      AS=P(LMAX,M,N3)*GAMMA/RO(LMAX,M,N3)
      A3=SQRT(AS)
   70 DO 80 IL=1,2
      X1=X3-(U(LMAX,M,N3)+U1)*0.5*DT
      X2=X3-(U(LMAX,M,N3)+A3+U2+A2)*0.5*DT
C
C     INTERPOLATE FOR THE PROPERTIES
C
      U1=BU*X1+CU
      U2=BU*X2+CU
      P2=BP*X2+CP
      RO2=BRO*X2+CRO
      AS=P2*GAMMA/RO2
      A2=SQRT(AS)
   80 CONTINUE
      V1=BV*X1+CV
      P1=BP*X1+CP
      RO1=BRO*X1+CRO
      AL1=BAL*X1+CAL
      BE1=BBE*X1+CBE
      DE1=BDE*X1+CDE
      UV1=U1*AL1+V1*BE1+DE1
      AS=P1*GAMMA/RO1
      A1=SQRT(AS)
      V2=BV*X2+CV
      AL2=BAL*X2+CAL
      BE2=BBE*X2+CBE
      DE2=BDE*X2+CDE
      UV2=U2*AL2+V2*BE2+DE2
C
C     INTERPOLATE FOR THE CROSS DERIVATIVES
      DV1=BDV*X1+CDV
      DP1=BDP*X1+CDP
      DRO1=BDRO*X1+CDRO
      DU2=BDU*X2+CDU
      DV2=BDV*X2+CDV
      DP2=BDP*X2+CDP
      DRO2=BDRO*X2+CDRO
C
C     CALCULATE THE PSI TERMS
```

```
C
      IF(NDIM.EQ.0) GO TO 100
      IF(M.EQ.1) GO TO 90
      ATERM2=RO2*V2/(DY*FLOAT(M-1)/BE2)
      GO TO 100
   90 ATERM2=RO2*BE2*DV2
  100 PSI31=-UV1*DV1-(DP1/RO1)*BE1
      PSI41=-UV1*DP1+A1*A1*UV1*DRO1
      PSI12=-UV2*DRO2-RO2*DV2*BE2-ATERM2-RO2*AL2*DU2
      PSI22=-UV2*DU2-AL2*DP2/RO2
      PSI42=-UV2*DP2+A2*A2*UV2*DRO2
      IF(ICHAR.EQ.1) GO TO 160
C
C     CALCULATE THE CROSS DERIVATIVES AT THE SOLUTION POINT
C
      IF(M.EQ.1) GO TO 110
      IF(M.EQ.ME) GO TO 120
      DU3=(U(LMAX,M+1,N3)-U(LMAX,M,N3))*DYR
      DV3=(V(LMAX,M+1,N3)-V(LMAX,M,N3))*DYR
      DP3=(P(LMAX,M+1,N3)-P(LMAX,M,N3))*DYR
      DRO3=(RO(LMAX,M+1,N3)-RO(LMAX,M,N3))*DYR
      GO TO 130
  110 DU3=0.0
      DV3=(4.0*V(LMAX,2,N3)-V(LMAX,3,N3))*0.5*DYR
      DP3=0.0
      DRO3=0.0
      GO TO 130
  120 MB=ME-1
      DU3=(U(LMAX,ME,N3)-U(LMAX,MB,N3))*DYR
      DV3=(V(LMAX,ME,N3)-V(LMAX,MB,N3))*DYR
      DP3=(P(LMAX,ME,N3)-P(LMAX,MB,N3))*DYR
      DRO3=(RO(LMAX,ME,N3)-RO(LMAX,MB,N3))*DYR
C
C     CALCULATE THE PSI TERMS AT THE SOLUTION POINT
C
  130 IF (NDIM.EQ.0) GO TO 150
      IF(M.EQ.1) GO TO 140
      ATERM3=RO(LMAX,M,N3)*V(LMAX,M,N3)/(DY*FLOAT(M-1)/BE)
      GO TO 150
  140 ATERM3=RO(LMAX,1,N3)*BE*DV3
  150 UV3=V(LMAX,M,N3)*BE+U(LMAX,M,N3)*AL+DE
      PSI13=-UV3*DRO3-RO(LMAX,M,N3)*(DV3*BE+AL*DU3)-ATERM3
      PSI23=-UV3*DU3-AL*DP3/RO(LMAX,M,N3)
      PSI33=-UV3*DV3-(DP3/RO(LMAX,M,N3))*BE
      PSI43=-UV3*DP3+A3*A3*UV3*DRO3
      PSI31B=(PSI31+PSI33)*0.5+QVT(LMAX,M)
      PSI41B=(PSI41+PSI43)*0.5+QPT(LMAX,M)
      PSI12B=(PSI12+PSI13)*0.5+QROT(LMAX,M)
      PSI22B=(PSI22+PSI23)*0.5+QUT(LMAX,M)
      PSI42B=(PSI42+PSI43)*0.5+QPT(LMAX,M)
```

```
      GO TO 170
  160 PSI31B=PSI31+QVT(LMAX,M)
      PSI41B=PSI41+QPT(LMAX,M)
      PSI12B=PSI12+QROT(LMAX,M)
      PSI22B=PSI22+QUT(LMAX,M)
      PSI42B=PSI42+QPT(LMAX,M)
C
C     SOLVE THE COMPATIBILITY EQUATIONS FOR U,V,P, AND RO
C
  170 P(LMAX,M,N3)=PE(M)
      AB=0.5*(A2+A3)
      ROB=0.5*(RO2+RO(LMAX,M,N3))
      RO(LMAX,M,N3)=RO1+2.0*(P(LMAX,M,N3)-P1-DT*PSI41B)/
     $ (A3*A3+A1*A1)
      IF(RO(LMAX,M,N3).LE.0.0) RO(LMAX,M,N3)=ROLOW
      U(LMAX,M,N3)=U2+((PSI42B+ROB*AB*PSI22B+AB*AB*PSI12B)*DT
     $ -(P(LMAX,M,N3)-P2))/(ROB*AB)
      V(LMAX,M,N3)=V1+DT*PSI31B
      V(LMAX,M,N3)=0.0
C
C     CHECK FOR INFLOW AND IF SO SET INFLOW BOUNDARY
C     CONDITIONS
C
      IF(U(LMAX,M,N3).GE.0.0) GO TO 180
      V(LMAX,M,N3)=0.0
      RO(LMAX,M,N3)=0.5*(RO(LMAX,1,N1)+RO(LMAX,MCB,N1))
  180 CONTINUE
      V(LMAX,MCB,N3)=-U(LMAX,MCB,N3)*NXNY(LMAX)
      V(LMAX,MCB,N3)=0.0
      RETURN
      END
```

```
      SUBROUTINE GRETY
C
C     ----------------------------------------------------------
C
C     THIS SUBROUTINE CALCULATES AND PLOTS THE PHYSICAL GRID
C
C     ----------------------------------------------------------
C
      COMMON/D6/ U(50,20,2),V(50,20,2),P(50,20,2),RO(50,20,2)
      COMMON/D7/ LMAX,MMAX,MCB,LCB,LBR,N,N1,N3,L1,L2,M1,M2,
     $ MCB1,MCB2,LCB1,LCB2,LBR1,LBR2,GAMMA,DX,DY,DT,ICHAR,DXR,
     $ DYR,RSTAR,RSTARS,G,PC,LC,PLOW,ROLOW,RG
      COMMON/D8/ XP1(50,20),YP1(50,20),KSI(2000),ETA(2000),
     $ PX(2000),PY(2000),XW(50),YW(20),NXNY(20),XE1,
     $ NDIM,PDS,AP
      COMMON/D9/CAV,XMU,XLA,RKMU,XRO,QUT(50,20),QVT(50,20),
     $ QPT(50,20),QROT(50,20),PE(20)
      REAL NXNY,NYNX,LC,KSI
      RX=1.0
      RX=2.
      XX=5.5
      XE=7.0
      KMAX=2000
      K1=KMAX-1
      YE=3.0
      RY=RX
      YY=1.0
      GX=6.0
      GY=6.0
      GX=2.0
      GY=2.0
      GX=1.0
      GY=1.0
      DSY=YE/FLOAT(K1)
      PY(1)=0.0
      DS=XE/FLOAT(K1)
      PX(1)=0.0
      KSI(1)=0.0
      ETA(1)=0.0
      SLPY1=0.0
      SY=0.0
      Y=0.0
      SLOPE1=0.0
      S=0.0
      X=0.0
      DO 2 L=2,KMAX
      Y=Y+DSY
      X=X+DS
      SLOPE2=RX*GX/(COSH(GX*(X-XX))*COSH(GX*(X-XX)))
      SLPY2=RY*GY/(COSH(GY*(Y-YY))*COSH(GY*(Y-YY)))
```

```
      SY1=SQRT(1.+SLPY1*SLPY1)
      SY2=SQRT(1.+SLPY2*SLPY2)
      S1=SQRT(1.+SLOPE1*SLOPE1)
      S2=SQRT(1.+SLOPE2*SLOPE2)
      SY=SY+DSY*0.5*(SY1+SY2)
      S=S+DS*0.5*(S1+S2)
      ETA(L)=SY
      PY(L)=Y
      KSI(L)=S
      PX(L)=X
      SLPY1=SLPY2
    2 CONTINUE
      SLOPE1=SLOPE2
      YE1=ETA(KMAX)
      DETA=YE1/FLOAT(M1)
      JY=1
      SSY=-DETA
      XE1=KSI(KMAX)
      DKSI=XE1/FLOAT(L1)
      J=1
      SS=-DKSI
      WRITE(6,100) KSI(KMAX),ETA(KMAX)
      DO 3 L=1,LMAX
      SS=SS+DKSI
      DO 4 K=J,KMAX
      IF ( ABS ( SS-KSI(K)).LT.0.05) GO TO 5
    4 CONTINUE
    5 J=K
      XW(L)=PX(K)
      X=XW(L)
      NXNY(L)=RX*GX/(COSH(GX*(X-XX))*COSH(GX*(X-XX)))
      WRITE(6,100) XW(L),SS,NXNY(L)
    3 CONTINUE
  100 FORMAT ( 5X,4F14.5)
      DO 33 M=1,MMAX
      SSY=SSY+DETA
      DO 44 KY=JY,KMAX
      IF ( ABS ( SSY-ETA(KY)).LT.0.05) GO TO 55
   44 CONTINUE
   55 JY=KY
      YW(M)=PY(KY)
      Y=YW(M)
      NYNX(M)=RY*GY/(COSH(GY*(Y-YY))*COSH(GY*(Y-YY)))
      WRITE(6,100) YW(M),SSY,NYNX(M)
   33 CONTINUE
      DX=XE1/FLOAT(L1)
      DY=YE1/FLOAT(M1)
      DXR=1./DX
      DYR=1./DY
      DO 210 L=1,LMAX
```

```
      DO 210 M=1,MMAX
      XP1(L,M)=XW(L)
      YP1(L,M)=YW(M)
  210 CONTINUE
      RETURN
      END
```

VITA

Ameziane Harhad was born July 6, 1952 in Algiers, Algeria. He received his secondary education at Technical High School in Algiers, Algeria, and was graduated in June, 1971. He received his diploma of Engineering in June, 1976 from the University of Algiers (Algeria). He came to the United States and enrolled for graduate study in Mechanical Engineering at Louisiana State University in August, 1977. He completed the Master of science degree in December, 1978. He has also earned a master of science degree in Mathematics in August, 1981. Now he is a candidate for the degree of Doctor of Philisophy in the Department of Mechanical Engineering to be awarded at the convocation to be held on August 7, 1982.

EXAMINATION AND THESIS REPORT

Candidate: Ameziane Harhad

Major Field: Mechanical Engineering

Title of Thesis: Numerical Modeling of the Internal Ballistics of a Gas Gun

Approved:

Major Professor and Chairman

Dean of the Graduate School

EXAMINING COMMITTEE:

Date of Examination:

May 28, 1982