AD-A045 703 BALLISTIC RESEARCH LABS ABERDEEN PROVING GROUND MD F/G 20/4
A FINITE ELEMENT METHOD AND CORRESPONDING PILOT COMPUTER CODE F--ETC(U)
SEP 77 J A SCHMITT

UNCLASSIFIED BRL-2017 NL

1 OF 1
ADA
045703

END
DATE
FILMED
11-77
DDC

(12)

AD

BRL-R-2017

BRL

AD A045703

REPORT NO. 2017

A FINITE ELEMENT METHOD AND CORRESPONDING PILOT COMPUTER CODE FOR HYPERBOLIC SYSTEMS OF EQUATIONS IN TWO SPATIAL DIMENSIONS AND TIME APPLIED TO UNSTEADY GAS FLOWS

James A. Schmitt

September 1977

Approved for public release; distribution unlimited.

USA ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
USA BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: BRL REPORT NO. 2017	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): A Finite Element Method and Corresponding Pilot Computer Code for Hyperbolic Systems of Equations in Two Spatial Dimensions and Time Applied to Unsteady Gas Flows.		5. TYPE OF REPORT & PERIOD COVERED: Final rept.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Dr. James A. Schmitt		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: USA Ballistic Research Laboratory, Aberdeen Proving Ground, MD 21005		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 1W161102AH43
11. CONTROLLING OFFICE NAME AND ADDRESS: US Army Materiel Development & Readiness Command, 5001 Eisenhower Avenue, Alexandria, VA 22333		12. REPORT DATE: SEPTEMBER 1977
		13. NUMBER OF PAGES: 80
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 72p.		15. SECURITY CLASS. (of this report): Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution u[illegible]d.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

FINITE ELEMENT METHOD, COMPUTER CODE, PLANAR SHOCK, LEAST SQUARES RESIDUAL, COMPRESSIBLE FLOW, BLAST WAVE, HYPERBOLIC SYSTEMS, GAS DYNAMICS, UNSTEADY GAS FLOW, NUMERICAL SOLUTION

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report contains a discussion of a numerical method for solving systems of first order time dependent hyperbolic equations in two spatial variables. This scheme which combines the finite element methodology and the properties of a hyperbolic system of differential equations is applied to unsteady gas flow problems. The formulation is based on the elementwise least squares minimization of the differential residual error and on the construction of the finite elements in both space and time. The corresponding computer program is

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

050 750

LB

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Block #20 - Abstract

listed. Numerical experiments involving both smooth and shocked flows are discussed. Areas of possible future code development are proposed.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Although the finite element method is a proven effective method for obtaining numerical solutions of solid mechanics problems[1], its impact on computational fluid dynamics has been felt only in the past few years[2]. Because of the diverse applications of this method in continuum mechanics, many departures from the original method used in structural analysis have been made. Our adaptation of the finite element method for direct application to unsteady gas flows in two spatial variables is based, in part, on Lynn and Arya's least squares formulation[3,4] and on Polk's one dimensional study[5]. Lynn and Arya's approach is based on the elementwise least squares minimization of the differential residual error which allows a direct finite element formulation from the governing differential equations. Furthermore, since the governing equations are hyperbolic, the finite elements can be constructed in both space and time so that they approximate the domain of determinancy associated with hyperbolic problems. Polk combined these two concepts and applied them to the unsteady isentropic flow of an inviscid gas expanding behind a piston. Using both linear and quadratic approximations to the dependent variables, he showed good agreement between the finite element results and the exact solution for the smooth portion of the flow. Near the gradient discontinuities which occurred in the flow, Polk constructed the finite elements so that a side of the element and the locus of discontinuities coincided. Using this special construction, good agreement was obtained everywhere in the flow.

This report is concerned with a finite element method for unsteady inviscid compressible flows in two spatial dimensions, a corresponding pilot computer code and some resulting numerical experiments.

---

[1] *O.C. Zienkiewicz, The Finite Element Method in Engineering Science, McGraw-Hill, 1971.*

[2] *J.T. Oden, O.C. Zienkiewicz, R.H. Gallagher, C. Taylor, eds., Finite Element Methods in Flow Problems, University of Alabama in Huntsville Press, 1974.*

[3] *P.P. Lynn and S.K. Arya, "Use of the Least Squares Criterion in the Finite Element Formulation," Int. J. Num. Meth. Engng., 6, 75-88, 1973.*

[4] *P.P. Lynn and S.K. Arya, "Finite Elements Formulated by the Weighted Discrete Least Squares Method," Int. J. Num. Meth. Enging., 8, 71-90, 1974.*

[5] *J. Polk, "A Least Squares Finite Element Approach to Unsteady Gas Dynamics," BRL Report No. 1885, May 1976.* (AD #A026531)

The system of governing equations are nonlinear hyperbolic equations. We take advantage of this fact to simplify the general finite element methodology. This is contrary to the parabolic regularization method of Oden, et. al.[6] for hyperbolic problems. In the techniques of Oden, et al., certain terms which depend on the discretization parameters are appended to the equations so that they become parabolic. This parabolic problem is then solved by a finite element technique. It can be shown for a class of problems that the solution of the parabolic problem converges to the original hyperbolic solution in the limit as the mesh size tends to zero. On the other hand, our formulation deals directly with the hyperbolic equations.

Our construction of the finite elements is reminiscent of Polk's construction in that they are in both space and time but differ in that they enclose, not coincide with, the domain of dependence. It will be shown that this construction simplifies the necessary integration routines while satisfying a Courant condition. No special construction of the finite element is provided near steep gradients and discontinuities in the flow. Consequently, no special knowledge of the solution is required and all interior nodes in the calculation are treated identically. Furthermore, by applying Lynn and Arya's least squares minimization to each finite element, we can avoid the large matrices generally associated with the finite element method for elliptic problems while still retaining the essential advantages of the method.

The general methodology; the construction of the finite elements, the approximations to the flow variables and the formulation in terms of the least squares error criterion, is explained in Section II. Section III contains a brief discussion of the computer code, the form of the governing equations and certain approximations used within the code. The results of two numerical experiments are given and discussed in Section IV. Section V contains a summary of the method and areas in which future work is required.

II. GENERAL METHODOLOGY

The first step in the finite element methodology is to divide the solution region into elements. We divide the computational domain for a given time (a two dimensional region) into triangular elements. The vertices of the triangles are called nodes. We assume that the boundaries are stationary so that the triangular divisions remain unchanged

---

[6] *J.T. Oden, L.C. Wellford, and C.T. Reddy, "A Study of Convergence and Stability of Finite Element Approximation of Shock and Acceleration Waves in Nonlinear Materials," U.S. Army Research Office Report P-11860-M/DAAG29-76-G-C022, August 1976.*

with time. The system of governing equations for compressible fluid flow include coupled nonlinear partial differential equations which express conservation of mass, momentum, and energy plus an algebraic equation of state. Because the governing equations are hyperbolic, the solution at a point $(\bar{x},\bar{y},\bar{t}+\Delta t)$ in the solution domain depends only on the value of the flow variables within the intersection of the domain of dependence (the mach cone) from the point $(\bar{x},\bar{y},\bar{t}+\Delta t)$ and the $(x,y,\bar{t})$ plane. Consequently, given a union of triangles in the $(x,y,\bar{t})$ plane with a vertex at the point $(\bar{x},\bar{y},\bar{t})$ and the values of the flow variables at the corresponding nodes, we can compute a value of Δt such that the mach cone from the point $(\bar{x},\bar{y},\bar{t}+\Delta t)$ lies within the union of triangles. Since the computed value of Δt will vary from node to node, we take the minimum value of Δt over all nodes as the next time step. This value allows a systematical advance in time.

We now define our finite element at a point $(\bar{x},\bar{y},\bar{t}+\Delta t)$ as the union of all prisms with a base vertex at the point $(\bar{x},\bar{y},\bar{t})$ and with a uniform height Δt. See Figure 1. The present one offers several desirable simplifications over other possible constructions of the finite element. From the discussion above it is clear that the values of the flow variables at a node are independent of the values at the other nodes at the same time level. Thus, the interconnection of the nodal values which is characteristic of finite element formulations of elliptic problems and which result in the manipulation of large matrices can and will be avoided. We will solve for the central nodal values at the new time level by considering only the finite element at the central node. Futhermore, any necessary time integrations over the finite element are simplified, since the sides of the elements are independent of time.

The next step is to choose the interpolating or trial function over the finite element. Let ω be a flow variable; that is, a dependent variable computed directly from the governing differential equations, not the equation of state. The interpolating function for ω is

$$\omega(x,y,t) = \omega_o(x,y,\bar{t}) + t\cdot(a_i x + a_{i+1} y + a_{i+2}), \qquad (1)$$

where the points (x,y,t) lie within the finite element at $(\bar{x},\bar{y},\bar{t}+\Delta t)$ and the parameters a_i, a_{i+1}, a_{i+2} are to be determined. The function $\omega_o(x,y,\bar{t})$ represents the flow variable ω at time $\bar{t}$ and is assumed known. These interpolation functions form over each finite element a linear approximation in space to the time derivative of ω.

To complete the model, a technique to determine the parameters a_i and thus the flow variables, is needed. A basic idea in the finite element methodology is to minimize the errors arising from the residual of the governing differential equations in terms of the interpolating functions. Thus, the finite element method approximates the minimum residual whereas the finite difference method approximates the differ-

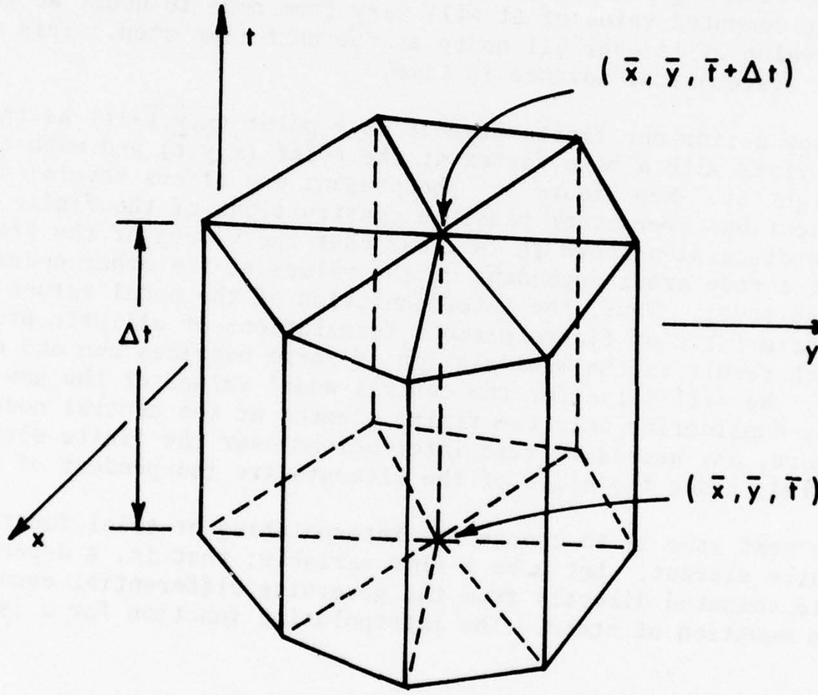

Figure 1. Typical Finite Element at an Interior Node $(\bar{x},\bar{y},\bar{t}+\Delta t)$

ential equation. Since the proposed analysis is elementwise, we desire a minimization technique for each element. To this end, we choose the elementwise least squares minimization of the differential residual error employed by Lynn and Arya.

We substitute the interpolating functions into the k^{th} governing differential equation, make the result dimensionless and denote the resulting residual by $D_k(x,y,t;\vec{a})$, where $\vec{a}$ is the vector of unknown parameters a_i. Each residual D_k is dimensionless so that no individual D_k will numerically dominate the least squares sum of all the residuals because of dimensional disparities. The D_k's are explicit algebraic functions of the independent variables x,y,t and the parameters $\vec{a}$. The total error in the sense of least squares over a particular finite element is denoted by $E(\vec{a})$ and is given by

$$E(\vec{a}) = \iiint_V \sum_{k=1}^{NOEQ} D_k^2(x,y,t;\vec{a})\ dxdydt, \tag{2}$$

where NOEQ is the number of governing equations and V is the volume of the finite element. We wish to minimize $E(\vec{a})$ with respect to the a_i's. A necessary condition for the existence of a minimum is $\partial E/\partial a_i = 0$ or

$$\iiint_V \sum_{k=1}^{NOEQ} D_k \frac{\partial D_k}{\partial a_i}\ dxdydt = 0\ ,\ \text{for each } i \tag{3}$$

Note that the derivatives $\partial D_k/\partial a_i$ can be explicitly calculated. By solving the nonlinear algebraic system of equations (3) for the unknowns a_i, we can determine the values of the flow variables at the nodal point $(\overline{x},\overline{y},\overline{t}+\Delta t)$. We repeat this process for each interior node in the solution domain.

For a boundary node, the above procedure is slightly altered. We rewrite the given boundary condition(s) at the boundary node in terms of the interpolating functions (1). We then solve for the unknown parameters a_i in equation (3) at the boundary node subject to the rewritten boundary conditions. Thus, at a boundary node we no longer have a pure minimization problem as at an interior node, but rather a constrained minimization problem.

III. PILOT COMPUTER PROGRAM

The pilot computer code is written in a modular structure fashion in order to clarify the logic of the program and to allow changes in the form of the governing equations, the interpolating functions, the method selected to solve the nonlinear system, etc.. Such flexibility is highly desirable for this pilot code. For example, in the finite difference techniques, the formulation of the equations have a profound affect on a method's performance (see, for example, Moretti[7]). Similarly, the form of the equations may affect the performance of the finite element method. Different forms of the equations for unsteady compressible flows are listed in Appendix A.

The code has three major components. The first component, subroutine START, accepts the geometric and control parameters. Furthermore, this section accepts and/or generates the nodal positions within the x-y solution subdomain and necessary initial and boundary value data. The second component, subroutine TIMESTEP, calculates the time increment (subroutine DELTAT) and the flow variables' values at each node at the new time (subroutine CALCI for interior nodes and CALCB for boundary nodes). The "heart" of the pilot code is clearly the TIMESTEP routine, since the general method outlined in Section II is implemented in this portion of the code. The third component, subroutine DISPLAY, provides the output and graphics capabilities for the program. The code as listed in Appendix D uses the non-conservation inviscid form of the governing equations in Cartesian coordinates, where body forces, heat absorption and heat fluxes are neglected (see equation system (A1) of Appendix A), the Newton-Raphson iteration method and certain approximations. We briefly discuss this specific situation below.

The governing equations in dimensional variables are:

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + \rho\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0 \ , \tag{4}$$

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) + \frac{\partial p}{\partial x} = 0, \tag{5}$$

$$\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right) + \frac{\partial p}{\partial y} = 0, \tag{6}$$

$$\rho\left(\frac{\partial e}{\partial t} + u\frac{\partial e}{\partial x} + v\frac{\partial e}{\partial y}\right) + p\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) = 0, \tag{7}$$

$$\rho = \rho(p,e) \ . \tag{8}$$

---

[7] *Gino Moretti, "A Pragmatical Analysis of Discretization Procedures for Initial - and Boundary - Value Problems in Gas Dynamics and Their Influence on Accuracy or Look Ma, No Wiggles!," Polytechnic Institute of New York Report No. 74-15, September 1974.*

Here the spatial coordinates are x and y, t is the time, u is the x-component of the velocity, v is the y-component of the velocity, p is the pressure, ρ is the density and e is the internal energy per unit mass. The functional form of the equation of state is given by equation (8). The particular equation of state used in a given calculation is specified in the subroutine EQNST.

In the actual calculations for a given finite element at the point $(\bar{x},\bar{y},\bar{t}+\Delta t)$, we translate the origin of the coordinate system to the point $(\bar{x},\bar{y},\bar{t})$ (see subroutine TRANSL). Conceptually, the translation enables each finite element to have its base centered at the same point $(x,y,t) = (0,0,0)$ and practically, it simplifies the calculations. The interpolating functions for the variables u, v, p, e within a finite element are:

$$u(x,y,t) = u_o(x,y,0) + t \cdot (a_1x + a_2y + a_3), \tag{9}$$

$$v(x,y,t) = v_o(x,y,0) + t \cdot (a_4x + a_5y + a_6), \tag{10}$$

$$p(x,y,t) = p_o(x,y,0) + t \cdot (a_7x + a_8y + a_9), \tag{11}$$

$$e(x,y,t) = e_o(x,y,0) + t \cdot (a_{10}x + a_{11}y + a_{12}), \tag{12}$$

where the variables x, y, t are in the translated finite element, the variables with subscript zero denote the variables at the known time level t=0, and the a_i's, i=1, 2, ..., 12 are the unknown parameters. We define the dimensionless residuals $D_k(x,y,t;\vec{a})$, k=1,2,3,4 as the quantities obtained by substituting equations (8)-(12) into the four differential equations (4)-(7), respectively, and then by dividing the first result by a ratio of a characteristic density to Δt, the second and third results by a ratio of a characteristic velocity to Δt, and the fourth by a ratio of a characteristic internal energy per mass to Δt. The values of the characteristic quantities are the density, sound speed and internal energy per unit mass at the center node and Δt=0. The derivatives of ρ with respect to the variables x, y and t can be obtained by the chain rule. The subsequent partial derivatives of ρ with respect to p and e are calculated directly from the equation of state and are coded into the subroutine EQNST. The functions u_o, v_o, p_o, e_o are estimated by a linear approximation on each triangle based on their values at the vertices. Although other approximations are possible, the present one is easily computable and is independent of the number of prisms in the finite element. The partial derivatives of u_o, v_o, p_o, e_o, required in $D_k(x,y,t;\vec{a})$ vary from triangle to triangle and consequently equation (3) must be rewritten as

$$F_i(\vec{a}) = \frac{\partial E(\vec{a})}{\partial a_i} = \sum_{\ell=1}^{NUMTRI} \iiint_{\ell\text{th prism}} \sum_{k=1}^{4} D_k \frac{\partial D_k}{\partial a_i} \, dxdydt$$

$$= 0 \ , \ i=1,2,...12, \qquad (13)$$

where NOEQ is now four and the finite element contains NUMTRI prisms (or triangles).

The Newton-Raphson method is used to solve the system of nonlinear algebraic equations (13) (see subroutine SØLVER) since the terms $D_k(x,y,t;\vec{a})$ are known functions of the parameter $\vec{a}$. The second partial derivatives of the least squares error can be explicitly calculated and are given by

$$\frac{\partial F_i(\vec{a})}{\partial a_j} = \frac{\partial}{\partial a_j}\left(\frac{\partial E(\vec{a})}{\partial a_i}\right) = \sum_{\ell=1}^{NUMTRI} \left\{ \sum_{k=1}^{4} \iiint_{\ell\text{th prism}} \left[D_k \frac{\partial^2 D_k}{\partial a_i \partial a_j} + \frac{\partial D_k}{\partial a_i}\frac{\partial D_k}{\partial a_j} \right] dxdydt \right\}, \quad i,j=1,2,...,12, \qquad (14)$$

(see subroutine FU). The initial values for a_3, a_6, a_9, a_{12} in the Newton-Raphson scheme for the finite element at $(0,0,\Delta t)$ are taken as the previously computed values of these parameters for the finite element at (0,0,0), (see subroutine INITZR). However, at the first time step a special calculation is needed to find the appropriate initial values (see subroutine INITNG). These parameters are determined by solving for $\partial/\partial t$ in each of the equations (4)-(7), by approximately the spatial dependence of the variables at the initial time by linear functions on each triangle, by finding the corresponding values of $\partial/\partial t$ at each vertex, by equating these to the time partials of equations (9) - (12), by averaging the resulting values of the a_i's over the triangles and by translating the result. In all cases, the initial values of a_1, a_2, a_4, a_5, a_7, a_8, a_{10}, and a_{11} are set to zero. In the sample problems, the iteration converged faster with the zero values than with the more obvious choice $a_i \neq 0$, $i \neq 3, 6, 9, 12$. The integrals of $D_k \cdot (\partial^2 D_k/\partial a_i \partial a_j)$, $D_k \cdot (\partial D_k/\partial a_i)$ and $(\partial D_k/\partial a_i) \cdot (\partial D_k/\partial a_j)$ are approximated by a two point Gaussian quadrature in time and by a product of first order functions in space. For example,

$$\iiint_{\ell\text{th prism}} \frac{\partial D_k}{\partial a_i} \frac{\partial D_k}{\partial a_j} \, dxdydt \approx \frac{\Delta t}{2} \sum_{m=1}^{2} \left| \iint_{\ell\text{th triangle}} \frac{\partial D_k}{\partial a_i} (x,y,t_m;\vec{a}) \cdot \frac{\partial D_k}{\partial a_j} (x,y,t_m;\vec{a}) \, dxdy \right| , \quad (15)$$

where t_m, $m = 1,2$, are the two Gaussian quadrature points between zero and Δt. In the spatial integrals at each Gaussian time, both factors are approximated by first order functions in x and y which are then multiplied and integrated exactly over the desired triangle (see subroutines DSUB, EVALPR and AREAIN).

Once the values of the parameters a_i are known, the desired values of the flow variables at the center node at the new time ($t = \Delta t$) can be determined easily from equations (9) - (12), (see subroutine VALNEW).

The above discussion applies equally to interior and boundary type nodes. However, as noted at the end of Section II, the minimization at a boundary node is a constrained minimization. The type of constraints will depend on the type of boundary condition imposed at a given node. As an example, the zero normal velocity boundary condition imposed at a solid wall is incorporated into the equation system (13) in Appendix B.

IV. NUMERICAL EXPERIMENTS

In this section the results of two non-steady flow calculations, the flow behind a cylindrical blast wave and the flow across a propagating normal shock, are presented. These two examples are computed in order to ascertain the scheme's characteristics on interior nodes for a smooth and discontinuous flow, respectively. Both of these time-dependent flows are essentially one dimensional in space, however, they will be treated as two dimensional problems. Furthermore, since closed-form solutions are known for each problem, the accuracy of the finite element formulation can be precisely evaluated.

The similarity solution of the blast wave problem is discussed by Sedov[8] and the formulas for the cylindrical case are summarized in Appen-

---

[8] *L.I. Sedov, Similarity and Dimensional Methods in Mechanics, Academic Press, 1959.*

dix C. The flow behind a cylindrical blast wave rather than a planar wave is computed because of its associated circular solution domain. The computational domain for a given time consists of an annulus from radius r_a to radius r_b which is the position of the shock front at the initial time t_o. Perhaps the most important advantage of the finite element method is its capability of dealing with complex geometrical shapes by using arbitrarily shaped simple elements. The ease by which this cylindrical problem is handled by the Cartesian finite element program, especially the circular boundaries, demonstrates this advantage in an elementary manner. In Figure 2 the first quadrant of a computational domain for a given time is drawn, where $r_a = 2.2$[m], $r_b = 3.0$[m], the nodes are equally spaced at four degree intervals on a given circle and the radial divisions are computed so that approximate equilateral triangles result. Consequently, the size of the triangles increases with the radial distance from the origin. Note the good approximation to the arc boundaries by the series of straight lines forming the base of the triangles. A finite difference method in Cartesian coordinates either would approximate the arc boundaries by a series of horizontal and vertical lines or would require a transform of the solution domain. In both cases special treatment of the boundaries would be required. On the other hand, no special programming is needed to treat the arc in the finite element code.

Numerical calculations were performed for the case $r_a = 2.2$[m], $r_b = 3.0$[m] and the initial time $t_o = 3.85342034\text{x}10^{-3}$[s]. On the initial time plane the values of flow variables are calculated from the exact solution (see Appendix C). For the inflow condition at r_a and outflow condition at r_b we again assign the exact value to the flow variables. The computed ratios of p/p_s, v_r/v_{r_s}, e/e_s and ρ/ρ_s (subscripts denotes value at the shock front) are compared to Sedov's exact values (solid line) in Figures 3, 4, 5, and 6, respectively. The symbols ⋈ and ⧗ denote the computed value on the triangular finite element mesh with nodes spaced at two degree intervals (the computed radial subdivisions range from 0.07[m] to 0.09[m]) and at four degree intervals (the computed radial subdivisions range from 0.14[m] to 0.18[m]), respectively. Both sets of values are at the same time ($4.12516608\text{x}10^{-3}$ [s]). The former used four timesteps ($\Delta t \approx 0.68\text{x}10^{-4}$[s]) and the latter two timesteps ($\Delta t \approx 1.36\text{x}10^{-4}$[s]) to reach the termination time. The maximum absolute value of the percent relative error for the ratios of pressure, radial velocity, internal energy and density are 1.15%, 0.12%, 0.10% and 1.24%, respectively, for the finer mesh and 2.68%, 0.30%, 0.36% and 3.04%, respectively, for the coarser mesh. We recall that the density is computed from the equation of state once the pressure and energy are calculated from the differential equations. Hence, the error in the density includes both the pressure and energy errors. We note that the maximum relative error in each flow variable occurs where the finite elements are the largest; that is, near r_b. At the opposite end, near r_a, the finite elements are the smallest and the absolute value of the percent relative error of the pressure, radial

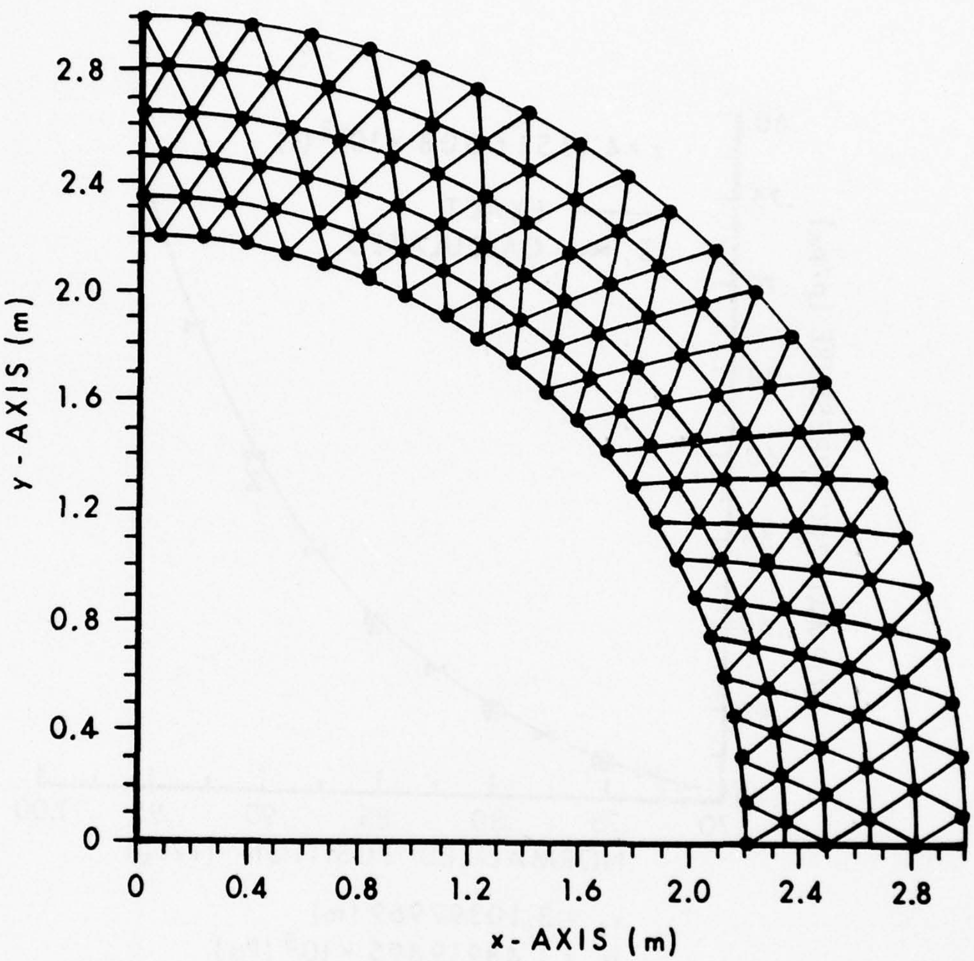

Figure 2. First Quadrant of Finite Element Mesh for Cylindrical Blast Wave Problem with Nodes at Four Degree Intervals

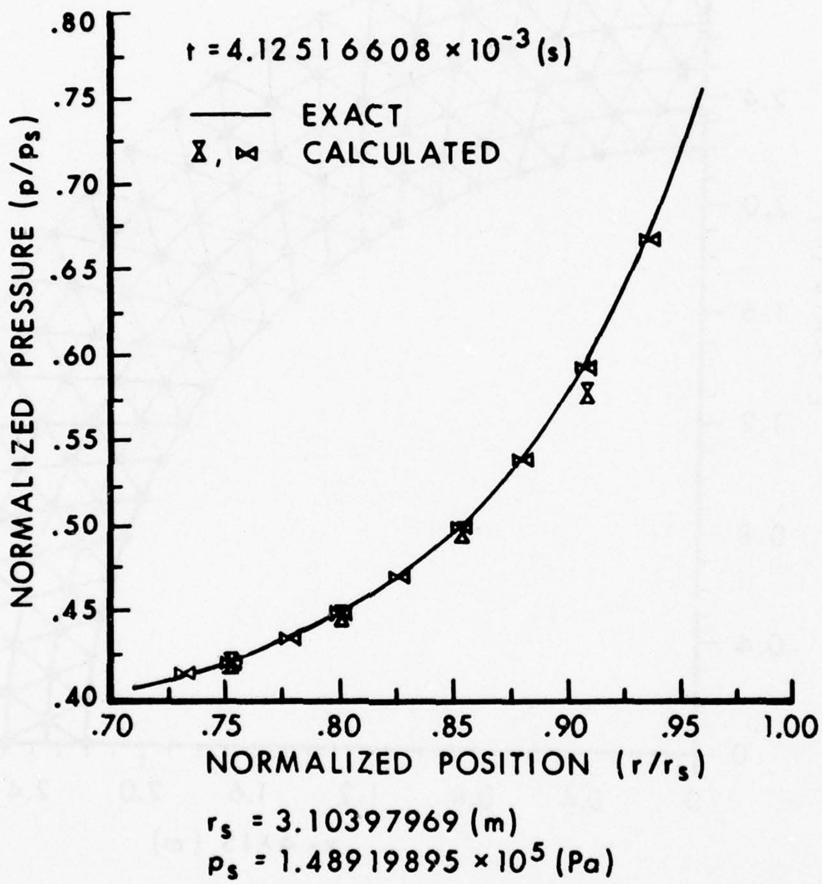

Figure 3. Comparison of Exact Normalized Pressure for Cylindrical Blast Wave [8] with Computed Values for Nodes Spaced at 4° (⧗) and 2° (⋈) Intervals

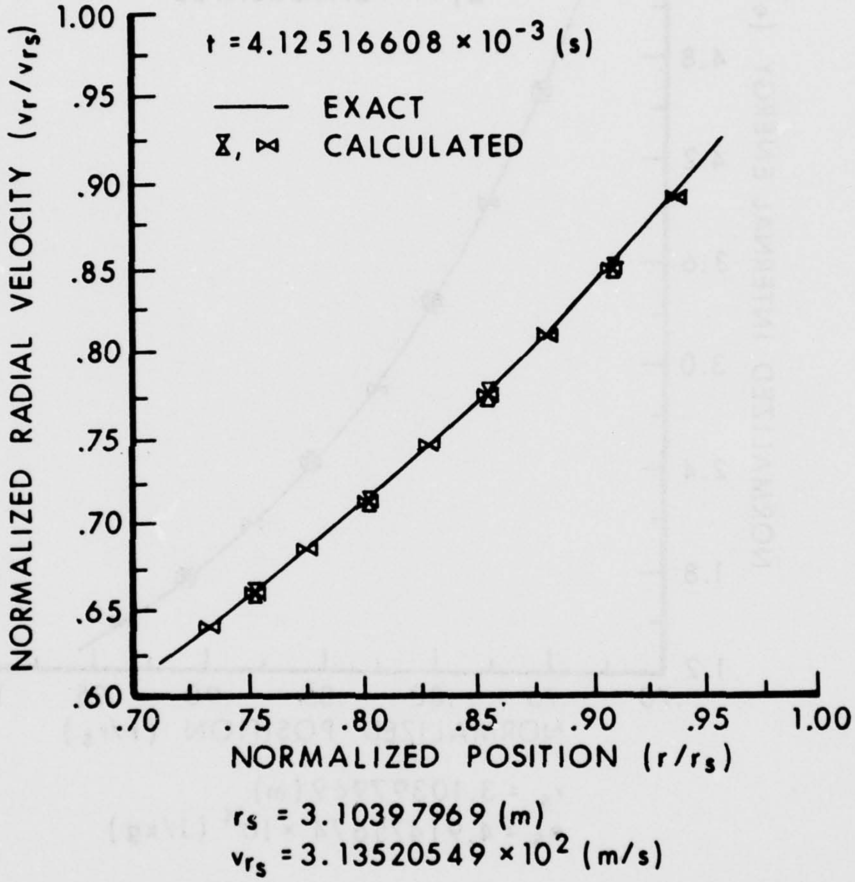

Figure 4. Comparison of Exact Normalized Radial Velocity for Cylindrical Blast Wave [8] with Computed Values for Nodes Spaced at 4° (⧗) and 2° (⋈) Intervals

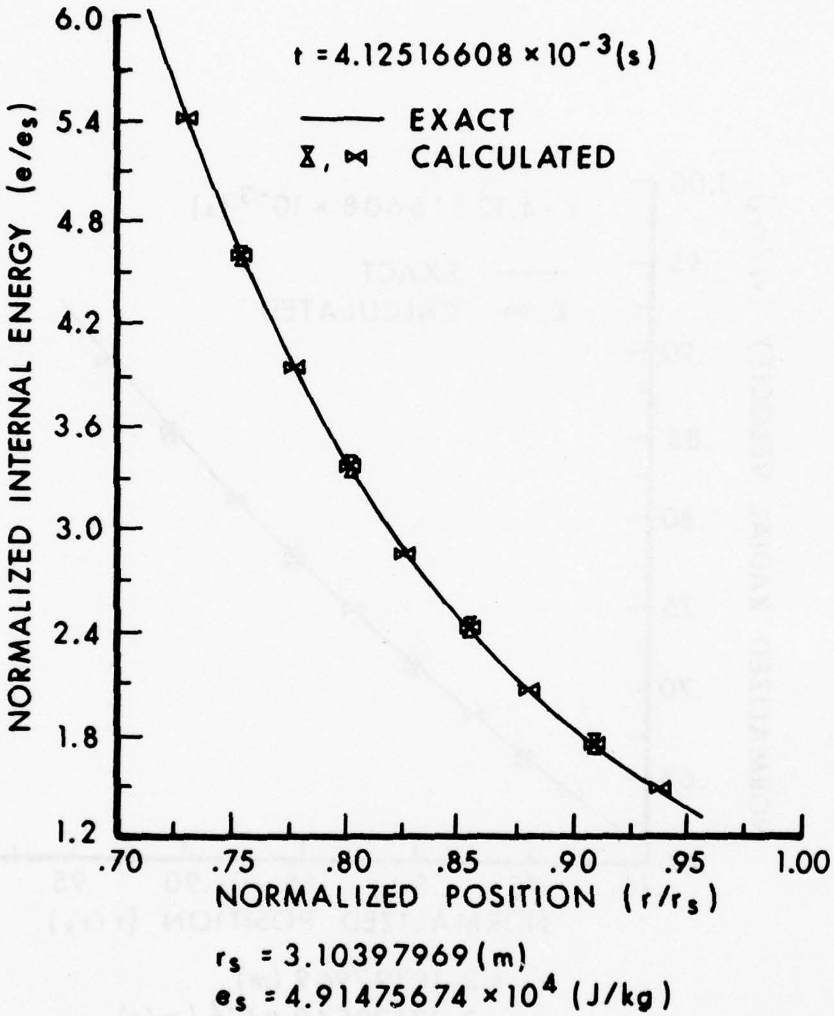

Figure 5. Comparison of Exact Normalized Internal Energy for Cylindrical Blast Wave [8] with Computed Values for Nodes Spaced at 4° (⧗) and 2° (⋈) Intervals

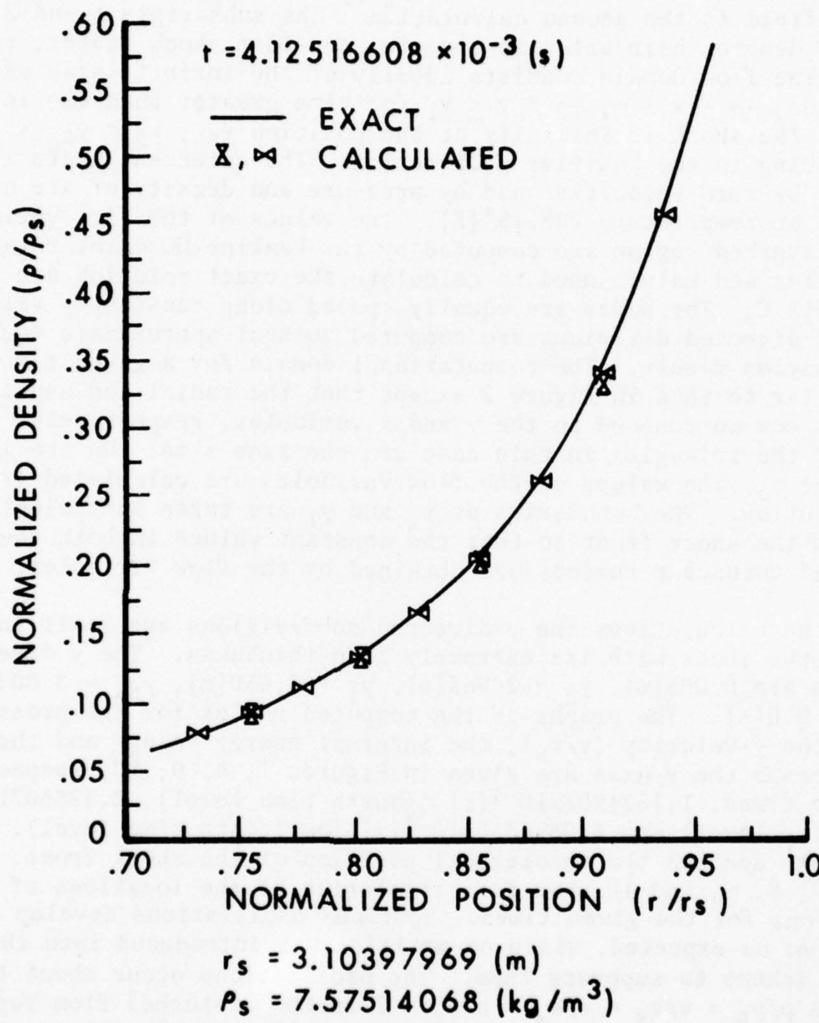

Figure 6. Comparison of Exact Normalized Density for Cylindrical Blast Wave [8] with Computed Values for Nodes Spaced at 4° (⧗) and 2° (⋈) Intervals

velocity, internal energy and density ratios for the finer and courser mesh are 0.11%, 0.01%, 0.02%, 0.14%, and 0.44%, 0.18%, 0.19%, 0.64%, respectively. Hence, overall and despite the relatively large size of the triangular elements, the agreement is fairly good.

A normal shock of strength five ($p_2/p_1 = 5$) propagating into a rectangular field is the second calculation. The subscripts 1 and 2 on the variables denote their value in the pre- and post shock states, respectively. The flow domain consists ideally of the infinite slab of the (x,y) plane, $-\infty < x < \infty$, $y_2 \leq y \leq y_1$ for time greater than the initial time t_o. The shock is initially at the position y_{s_o}, $y_2 < y_{s_o} < y_1$, and is moving in the positive y-direction. The quiescent state is characterized by zero velocities and by pressure and density of air at sea level and at temperature 288.15°[K]. The values of the flow variables in the disturbed region are computed by the Rankine-Hugoniot relations. The formulas and values used to calculate the exact solution are listed in Appendix C. The nodes are equally spaced along constant y values and the y directed divisions are computed so that approximate equilateral triangles result. The computational domain for a given time is very similar to that in Figure 2 except that the radial and angular variables now correspond to the y and x variables, respectively. However, all the triangles in this case are the same size. On the initial time plane t_o, the values of the flow variables are calculated from the exact solution. The boundaries at y_2 and y_1 are taken sufficiently far away from the shock front so that the constant values in both the disturbed and quiescent regions are obtained by the flow variables.

For the calculations the y directed subdivisions are small in order to model the shock with its extremely thin thickness. The y directed divisions are 0.005[m], $y_2 = 2.965$[m], $y_1 = 3.050$[m], $y_{s_o} = 3.0025$[m] and $t_o = 0.0$[s]. The graphs of the computed ratios for the pressure (p/p_s), the y-velocity (v/v_s), the internal energy (e/e_s) and the density (ρ/ρ_s) versus the y-axis are given in Figures 7, 8, 9, 10, respectively, for three times; $1.1524902\text{x}10^{-4}$[s] (fourth time level), $2.5986071\text{x}10^{-4}$[s] (ninth time level) and $4.0654310\text{x}10^{-4}$[s](fourteenth time level). The symbol $\bowtie$ denotes the theoretical position of the shock front. The Figures 7, 8, 9, and 10 show fair resolution of the locations of the shock front for the given times. Spurious oscillations develop at the shock front as expected, since no artifice was introduced into the equations or scheme to suppress them. The oscillations occur about the exact solutions $p/p_s = v/v_s = e/e_s = \rho/\rho_s = 1$ in the disturbed flow region. A major objective of future work is to curtail these oscillations (see Section V of this report).

We close this section with a discussion of the calculation time. The times needed to calculate the flow variables both for an entire time level and at an individual interior node are given to help ascertain the time characteristics of the scheme. All the calculations were done on the BRLESC computer facility at the Ballistic Research Laboratory. The calculations of the flow behind a cylindrical blast wave took approximately 1.3 and 0.5 minutes for the finer and coarser grids, respectively,

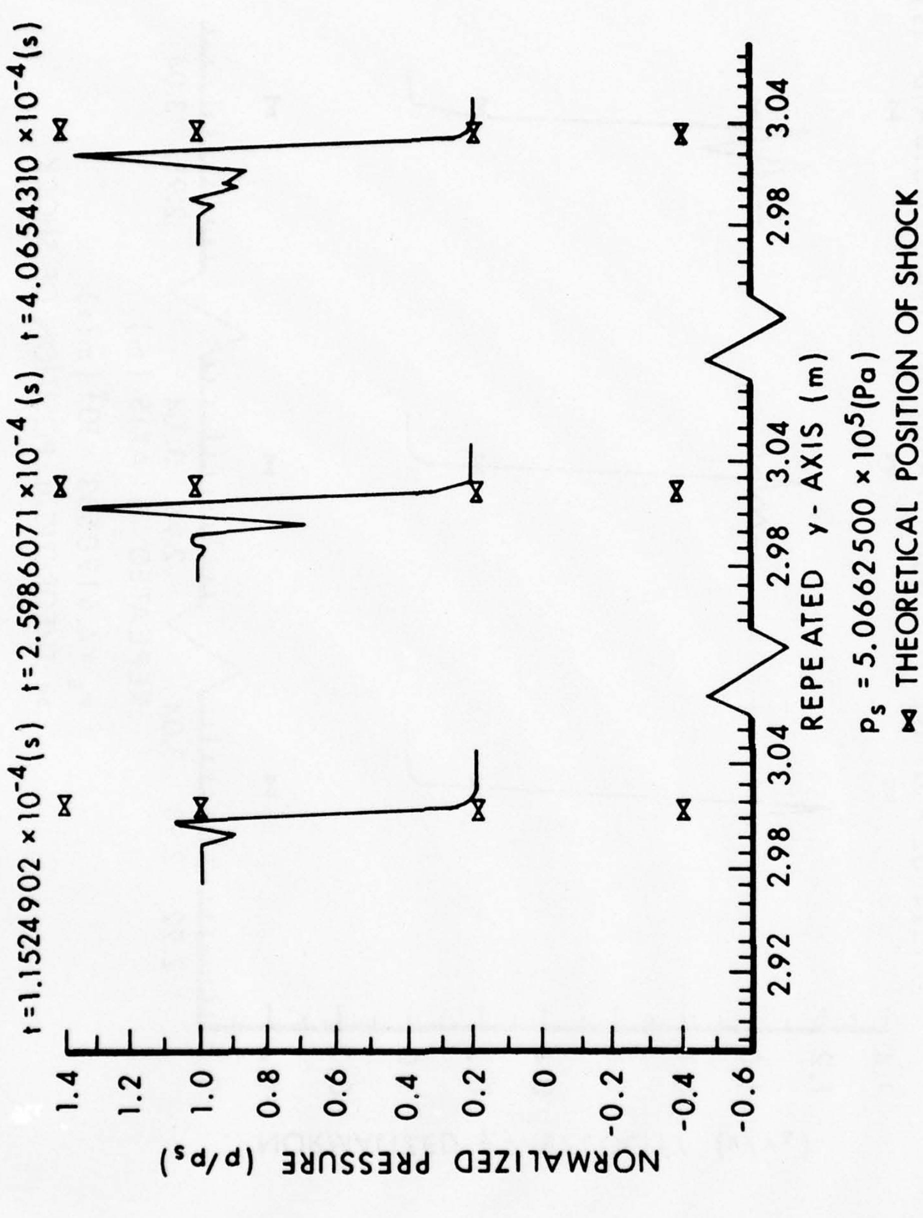

Figure 7. Variation of Normalized Pressure with Distance for Normal Shock of Strength Five at Three Times

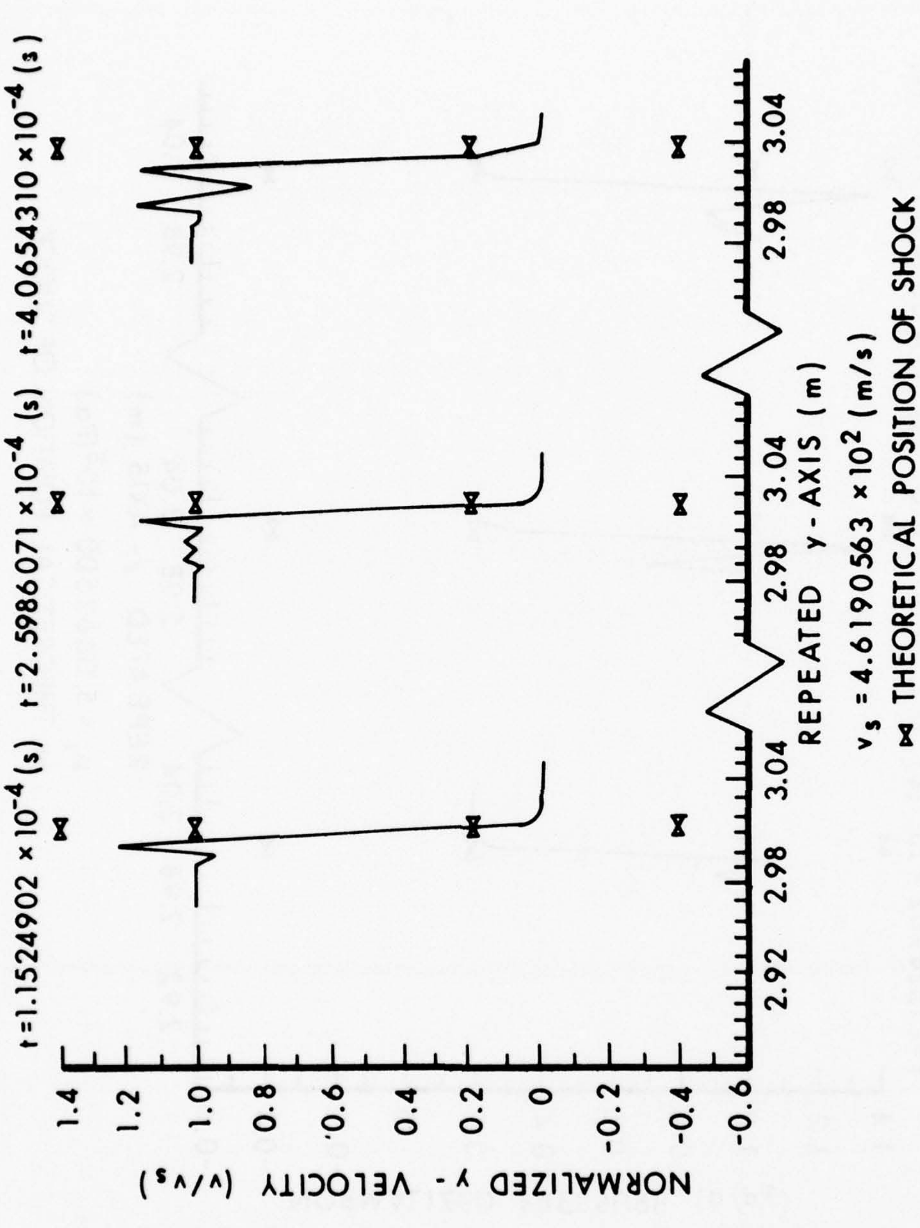

Figure 8. Variation of Normalized y Component of Velocity with Distance for Normal Shock of Strength Five at Three Times

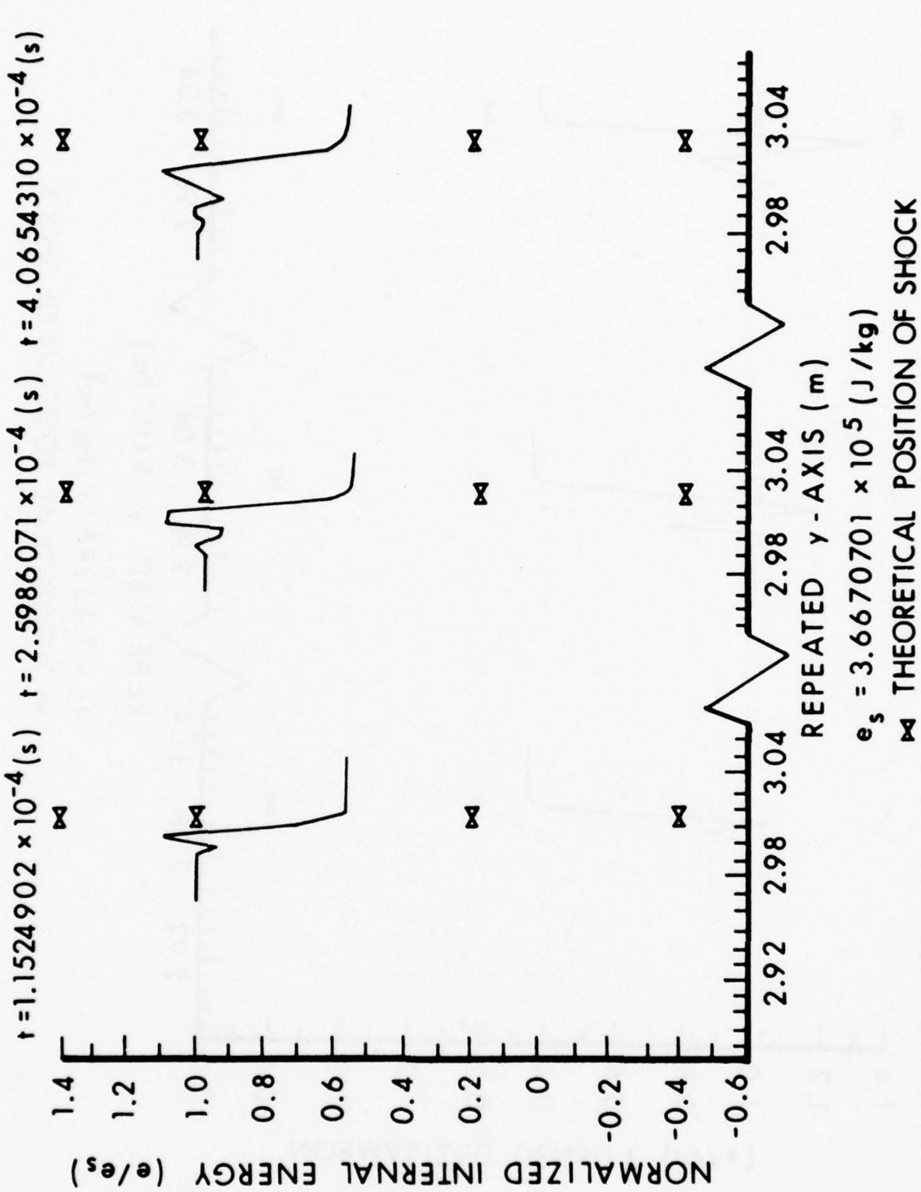

Figure 9. Variation of Normalized Internal Energy with Distance for Normal Shock of Strength Five at Three Times

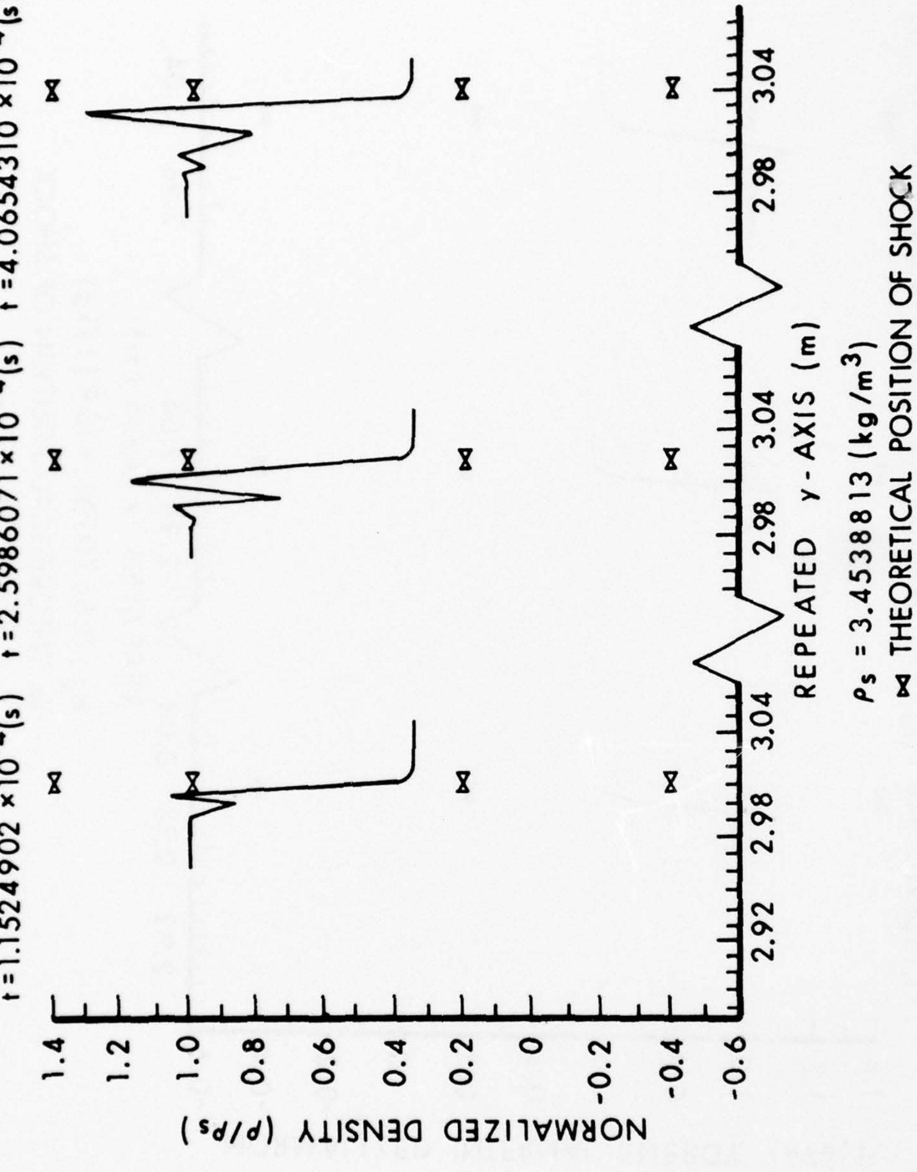

Figure 10. Variation of Normalized Density with Distance for Normal Shock of Strength Five at Three Times

of run time per time level and approximately eight seconds per interior node for both grids. These times include, on the average, two iterations per node in the Newton-Raphson method. The shock propagation calculations took approximately 4.5 minutes per time level and approximately 16 seconds per interior node. The shock propagation problem typically required 4 iterations per node in order to find convergent values of the unknown parameters a_i. Although the above times are not as small as one might hope, they can be reduced substantially by simple modifications of the algorithm (see Section V). Such simple optimizations must be investigated in future work.

V. SUMMARY

We have presented a numerical scheme for solving time dependent hyperbolic equations in two space dimensions. This method merges the concepts of the finite element method and the properties of hyperbolic systems of equations and is applied to unsteady gas flows. The corresponding hydrocode is summarized and listed. Finally, numerical calculations involving both smooth and shocked flows are given and discussed.

The purpose of this report is to give a summary of the work already accomplished rather than a definitive description of the quality and usefulness of this numerical scheme. However, several positive aspects of the method can already be seen. The formulation of the method is straight-forward and avoids the large matrices associated with the finite element method. The method handles different geometrically shaped boundaries easily and accurately. Even for relatively large mesh sizes, the results for a smooth flow are accurate. Finally in a propagating shock problem, the shock's position can easily be discerned.

From the example calculations in Section IV, it is clear that improvements must be made in several areas before the method's potential can be accurately ascertained. Listed below, not necessarily in order of importance or difficulty, are several such areas.

1. Curtail the spurious oscillations near shocks. Several methods; such as the flux-corrected transport techniques of Boris, et al,[9] and the well-known artificial viscosity method (see Roache[10]), can be

---

[9] *D.L. Book, J.P. Boris, and K. Hain "Flux-Control Transport II: Generalizations of the Method,"* <u>*J. Comp. Phys.*</u>*, 18, pp. 248-83, July 1975.*

[10] *P.J. Roache,* <u>*Computational Fluid Dynamics*</u>*, Hermosa Publishers, P. O. Box 8172, Albuquerque, New Mexico, 1976.*

applied. Although the latter is simpler to use, the former technique may hold more promise because the oscillations do occur about the correct solution and the flux correction will not significantly reduce the resolution of the shock as does artificial viscosity.

2. Shorten computing time. The run time can be significantly reduced by simple alterations in the algorithm. Recall that the spatial derivatives of the variables at the known time level (u_o, v_o, p_o, e_o) were computed on every triangular base of the finite element which resulted in the repeated calculation (up to NUMTRI times) of the residuals D_k and their partial derivatives at the nodes (see equations (13) - (14)). If these derivatives at the nodes were computed once for a given finite element, the time reduction would be a factor of 0.5. The run time could be further reduced by an order of magnitude if the calculations were done on a machine similar to the CDC 7600.

3. Rerun examples with different equation formulations. Certain formulations may increase the accuracy of the computed results and decrease the oscillations due to shocks. See Appendix A for different form of the governing equations.

4. Apply the method to actual numerical problem. By applying this method to a problem with complicated boundaries, not only could the method's treatment of boundaries be tested, but also the entire method could be compared to another numerical solution technique.

5. Extend the method with respect to "infinite" strength shocks and moving boundaries. Once these extensions are incorporated into the method, this scheme could hopefully be used to finally develop an adequate model of the severe transitional ballistics environment.

ACKNOWLEDGEMENT

The author wishes to thank Dr. Aivars Celmiņš for his continual help and guidance during the course of this investigation.

REFERENCES

1. O.C. Zienkiewicz, The Finite Element Method in Engineering Science, McGraw-Hill, 1971.

2. J.T. Oden, O.C. Zienkiewicz, R.H. Gallagher, C. Taylor eds., Finite Element Methods in Flow Problems, University of Alabama in Huntsville Press, 1974.

3. P.P. Lynn and S.K. Arya, "Use of the Least Squares Criterion in the Finite Element Formulation," Int. J. Num. Meth. Engng., 6, 75-88, 1973.

4. P.P. Lynn and S.K. Arya, "Finite Elements Formulated by the Weighted Discrete Least Squares Method," Int. J. Num. Meth. Engng., 8, 71-90, 1974.

5. J. Polk, "A Least Square Finite Element Approach to Unsteady Gas Dynamics," BRL Rep. No 1885, May 1976. (AD #A026531)

6. J.T. Oden, L.C. Wellford, and C.T. Reddy, "A Study of Convergence and Stability of Finite Element Approximation of Shock and Acceleration Waves in Nonlinear Materials," U.S. Army Research Office Report P-11860-M/DAAG29-76-G-0022, August 1976.

7. Gino Moretti, "A Pragmatical Analysis of Discretization Procedures for Initial - and Boundary - Value Problems in Gas Dynamics and Their Influence on Accuracy or Look Ma, No Wiggles!," Polytechnic Institute of New York Report No. 74-15, September 1974.

8. L.I. Sedov, Similarity and Dimensional Methods in Mechanics, Academic Press, 1959.

9. D.L. Book, J.P. Boris, and K. Hain, "Flux-Corrected Transport II: Generalizations of the Method," J. Comp. Phys., 18, pp. 248-83, July 1975.

10. P.J. Roache, Computational Fluid Dynamics, Hermosa Publishers, P.O. Box 8172, Albuquerque, New Mexico, 1976.

11. H. W. Liepmann and A. Roshko, Elements of Gas Dynamics, Wiley and Sons, 1957.

APPENDIX A

VARIOUS FORMULATIONS OF THE GOVERNING EQUATIONS

The system of governing equations for fluid flow consists of differential equations which express the conservation of mass (continuity equation), momentum (Navier-Stokes equations) and internal energy (energy equation) plus an algebraic equation of state. We list four different formulations of these time dependent differential equations in two spatial variables. In the first three subsections the following definitions are used: the first and second coefficients of viscosity, heat absorption, the x and y components of the body force and heat flux are denoted by $\mu(x,y,t)$ [kg/(m·s)], $\lambda(x,y,t)$ [kg/(m·s)], $H(x,y,t)$ [J/(kg·s)], $B_1(x,y,t)$[N/kg], $B_2(x,y,t)$[N/kg], $q_1(x,y,t)$ [J/(m^2·s)], and $q_2(x,y,t)$[J/(m^2·s)], respectively.

1. Independent Cartesian coordinates are x,y,t and dependent valuables are u, v, p, ρ, e.

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

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) = \rho B_1 - \frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left[(\lambda+2\mu)\frac{\partial u}{\partial x}\right] + \frac{\partial}{\partial x}\left(\lambda\frac{\partial v}{\partial y}\right) + \frac{\partial}{\partial y}\left[\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right]$$

$$\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right) = \rho B_2 - \frac{\partial p}{\partial y} + \frac{\partial}{\partial y}\left[(\lambda+2\mu)\frac{\partial v}{\partial y}\right] + \frac{\partial}{\partial y}\left(\lambda\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial x}\left[\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right] \qquad \text{(A1)}$$

$$\rho\left(\frac{\partial e}{\partial t} + u\frac{\partial e}{\partial x} + v\frac{\partial e}{\partial y}\right) = \rho Q - \left(\frac{\partial q_1}{\partial x} + \frac{\partial q_2}{\partial y}\right) - p\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)^2 + 2\mu\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right] + \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2$$

2. Independent Cartesian variables are x,y,t and dependent variables are $\bar{u}$, $\bar{v}$, $\bar{e}$, ρ, p. The variables $\overline{u} = \rho u$, $\overline{v} = \rho v$, $\overline{B}_1 = \rho B_1$, $\overline{B}_2 = \rho B_2$, $\overline{e} = \rho e$, $\overline{H} = \rho H$ are the x and y components of the momentum and body force, internal energy and heat absorption per unit volume, respectively.

$$\frac{\partial \rho}{\partial t} + \frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y} = 0$$

$$\frac{\partial \bar{u}}{\partial t} + \frac{\partial}{\partial x}\left(\frac{\bar{u}^2}{\rho}\right) + \frac{\partial}{\partial y}\left(\frac{\bar{v}\bar{u}}{\rho}\right) = \bar{B}_1 - \frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left[\frac{\lambda+2\mu}{\rho}\left(\frac{\partial \bar{u}}{\partial x} - \frac{\bar{u}}{\rho}\frac{\partial \rho}{\partial x}\right)\right]$$

$$+ \frac{\partial}{\partial x}\left[\frac{\lambda}{\rho}\left(\frac{\partial \bar{v}}{\partial y} - \frac{\bar{v}}{\rho}\frac{\partial \rho}{\partial y}\right)\right]$$

$$+ \frac{\partial}{\partial y}\left[\frac{\mu}{\rho}\left(\frac{\partial \bar{u}}{\partial y} + \frac{\partial \bar{v}}{\partial x} - \frac{\bar{u}}{\rho}\frac{\partial \rho}{\partial y} - \frac{\bar{v}}{\rho}\frac{\partial \rho}{\partial x}\right)\right]$$

$$\frac{\partial \bar{v}}{\partial t} + \frac{\partial}{\partial x}\left(\frac{\bar{u}\bar{v}}{\rho}\right) + \frac{\partial}{\partial y}\left(\frac{\bar{v}^2}{\rho}\right) = \bar{B}_2 - \frac{\partial p}{\partial y} + \frac{\partial}{\partial y}\left[\frac{\lambda+2\mu}{\rho}\left(\frac{\partial \bar{v}}{\partial y} - \frac{\bar{v}}{\rho}\frac{\partial \rho}{\partial y}\right)\right]$$

$$+ \frac{\partial}{\partial y}\left[\frac{\lambda}{\rho}\left(\frac{\partial \bar{u}}{\partial x} - \frac{\bar{u}}{\rho}\frac{\partial \rho}{\partial x}\right)\right] \qquad \text{(A2)}$$

$$+ \frac{\partial}{\partial x}\left[\frac{\mu}{\rho}\left(\frac{\partial \bar{v}}{\partial x} + \frac{\partial \bar{u}}{\partial y} - \frac{\bar{v}}{\rho}\frac{\partial \rho}{\partial x} - \frac{\bar{u}}{\rho}\frac{\partial \rho}{\partial y}\right)\right]$$

$$\frac{\partial \bar{e}}{\partial t} + \frac{\partial}{\partial x}\left(\frac{\bar{e}\bar{u}}{\rho}\right) + \frac{\partial}{\partial y}\left(\frac{\bar{e}\bar{v}}{\rho}\right) = \bar{H} - \left(\frac{\partial q_1}{\partial x} + \frac{\partial q_2}{\partial y}\right) - \frac{p}{\rho}\left[\frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y} - \frac{\bar{u}}{\rho}\frac{\partial \rho}{\partial x} - \frac{\bar{v}}{\rho}\frac{\partial \rho}{\partial y}\right]$$

$$+ \frac{\lambda+\mu}{\rho^2}\left[\frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y} - \frac{\bar{u}}{\rho}\frac{\partial \rho}{\partial x} - \frac{\bar{v}}{\rho}\frac{\partial \rho}{\partial y}\right]^2$$

$$+ \frac{2\mu}{\rho^2}\left[\left(\frac{\partial \bar{u}}{\partial x} - \frac{\bar{u}}{\rho}\frac{\partial \rho}{\partial x}\right)^2 + \left(\frac{\partial \bar{v}}{\partial y} - \frac{\bar{v}}{\rho}\frac{\partial \rho}{\partial y}\right)^2\right]$$

3. Independent Cartesian variables are x,y,t and dependent variables are v_r, θ, S, ρ, p. The variables $v_r(x,y,t)$[m/s] given by $v_r = (u^2 + v^2)^{1/2}$, θ(x,y,t)[rad] given by tan θ = v/u, S[J/K], and T[K] are the modulus of the velocity vector, the argument of the velocity vector, entropy and temperature, respectively.

$$\frac{\partial \rho}{\partial t} + \cos\Theta \left[\frac{\partial}{\partial x}(\rho v_r) + \rho v_r \frac{\partial \Theta}{\partial y}\right] + \sin\Theta \left[\frac{\partial}{\partial y}(\rho v_r) - \rho v_r \frac{\partial \Theta}{\partial x}\right] = 0$$

$$\rho \left[\cos\Theta \left(\frac{\partial v_r}{\partial t} + v_r \frac{\partial v_r}{\partial x}\cos\Theta - v_r^2 \frac{\partial \Theta}{\partial x}\sin\Theta\right)\right.$$

$$\left. - v_r \sin\Theta \left(\frac{\partial \Theta}{\partial t} + v_r \frac{\partial \Theta}{\partial y}\sin\Theta - \frac{\partial v_r}{\partial y}\cos\Theta\right)\right]$$

$$= \rho B_1 - \frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left[(\lambda+2\mu)\left(\frac{\partial v_r}{\partial x}\cos\Theta - v_r \frac{\partial \Theta}{\partial x}\sin\Theta\right)\right]$$

$$+ \frac{\partial}{\partial x}\left[\lambda\left(v_r \frac{\partial \Theta}{\partial y}\cos\Theta + \frac{\partial v_r}{\partial y}\sin\Theta\right)\right]$$

$$+ \frac{\partial}{\partial y}\left\{\mu\left[\left(\frac{\partial v_r}{\partial x} - v_r \frac{\partial \Theta}{\partial y}\right)\sin\Theta + \left(\frac{\partial v_r}{\partial y} + v_r \frac{\partial \Theta}{\partial x}\right)\cos\Theta\right]\right\}$$

(A3)

$$\rho \left[v_r \cos\Theta \left(\frac{\partial \Theta}{\partial t} + v_r \frac{\partial \Theta}{\partial x}\cos\Theta + \frac{\partial v_r}{\partial x}\sin\Theta\right)\right.$$

$$\left. + \sin\Theta \left(\frac{\partial v_r}{\partial t} + v_r^2 \frac{\partial \Theta}{\partial y}\cos\Theta + v_r \frac{\partial v_r}{\partial y}\sin\Theta\right)\right]$$

$$= \rho B_2 - \frac{\partial p}{\partial y} + \frac{\partial}{\partial y}\left[(\lambda+2\mu)\left(v_r \frac{\partial \Theta}{\partial y}\cos\Theta + \frac{\partial v_r}{\partial y}\sin\Theta\right)\right]$$

$$+ \frac{\partial}{\partial y}\left[\lambda\left(\frac{\partial v_r}{\partial x}\cos\Theta - v_r \frac{\partial \Theta}{\partial x}\sin\Theta\right)\right]$$

$$+ \frac{\partial}{\partial x}\left\{\mu\left[\left(v_r \frac{\partial \Theta}{\partial x} + \frac{\partial v_r}{\partial y}\right)\cos\Theta + \left(\frac{\partial v_r}{\partial x} - v_r \frac{\partial \Theta}{\partial y}\right)\sin\Theta\right]\right\}$$

$$\rho T\left[\frac{\partial S}{\partial t} + v_r \frac{\partial S}{\partial x}\cos\Theta + v_r \frac{\partial S}{\partial y}\sin\Theta\right] = \rho H - \left(\frac{\partial q_1}{\partial x} + \frac{\partial q_2}{\partial y}\right)$$

$$+ \lambda \left[\left(\frac{\partial v_r}{\partial x} + v_r \frac{\partial \Theta}{\partial y}\right)\cos\Theta\right.$$

$$\left. + \left(\frac{\partial v_r}{\partial y} - v_r \frac{\partial \Theta}{\partial x}\right)\sin\Theta\right]^2 + 2\mu\left\{\left[\left(\frac{\partial v_r}{\partial x}\right)^2 + v_r^{\,2}\left(\frac{\partial \Theta}{\partial y}\right)^2\right]\cos^2\Theta\right.$$

$$\left. - 2v_r\left[\frac{\partial \Theta}{\partial x}\frac{\partial v_r}{\partial x} - \frac{\partial \Theta}{\partial y}\frac{\partial v_r}{\partial y}\right]\sin\Theta\cos\Theta + \left[v_r^{\,2}\left(\frac{\partial \Theta}{\partial x}\right)^2 + \left(\frac{\partial v_r}{\partial y}\right)^2\right]\sin^2\Theta\right\}$$

$$+ \mu\left[\left(\frac{\partial v_r}{\partial y} + v_r \frac{\partial \Theta}{\partial x}\right)\cos\Theta + \left(\frac{\partial v_r}{\partial x} - v_r \frac{\partial \Theta}{\partial y}\right)\sin\Theta\right]^2$$

4. Independent variables are r,z,t and dependent variables are u_r, u_Θ, w, ρ^*, p^*, I. In the governing equations listed below, the following definitions apply: r is the radial distance ($r = (x^2 + y^2)^{1/2}$) [m], z is the axial distance [m], t is the time [s], $u_r(r,z,t)$ is the radial velocity [m/s], $u_\Theta(r,z,t)$ is the swirl velocity [m/s], $w(r,z,t)$ is the axial velocity [m/s], $\rho^*(r,z,t)$ is the density [kg/m^3], $p^*(r,z,t)$ is the pressure [Pa] and I is the internal energy per unit mass [J/kg]. Furthermore, the first and second coefficients of viscosity, the radial and axial components of the heat flux vector, the heat absorption term and the radial, angular and axial components of the body force are denoted by $\nu(r,z,t)$[kg/(m·s)], $\eta(r,z,t)$[kg/(m·s)], $h_r(r,z,t)$(J/(m^2·s)], $h_z(r,z,t)$[J/(m^2·s)], $H^*(r,z,t)$[J/(kg·s)], $B_r(r,z,t)$[N/kg], $B_\Theta(r,z,t)$ [N/kg], $B_z(r,z,t)$[N/kg], respectively. In this formulation, the flow is three dimensional and axially symmetric; that is, the six unknowns depend <u>only</u> on the three variables, r, z, and t.

$$\frac{\partial \rho^*}{\partial t} + u_r \frac{\partial \rho^*}{\partial r} + w\frac{\partial \rho^*}{\partial z} + \rho^*\left[\frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{\partial w}{\partial z}\right] = 0$$

$$\rho^*\left[\frac{\partial u_r}{\partial t} + u_r\frac{\partial u_r}{\partial r} - \frac{u_\Theta^{\,2}}{r} + w\frac{\partial u_r}{\partial z}\right] = + \rho^* B_r - \frac{\partial p^*}{\partial r} + \frac{\partial}{\partial r}\left[(\eta + 2\nu)\frac{\partial u_r}{\partial r}\right]$$

$$+ \frac{\partial}{\partial r}\left[\eta\left(\frac{u_r}{r} + \frac{\partial w}{\partial z}\right)\right] + \frac{2\nu}{r}\left(\frac{\partial u_r}{\partial r} - \frac{u_r}{r}\right)$$

$$+ \frac{\partial}{\partial z}\left[\nu\left(\frac{\partial u_r}{\partial z} + \frac{\partial w}{\partial r}\right)\right]$$

$$\rho^*\left[\frac{\partial u_\Theta}{\partial t} + u_r\frac{\partial u_\Theta}{\partial r} - \frac{u_r u_\Theta}{r} + w\frac{\partial u_\Theta}{\partial z}\right] = \rho^* B_\Theta + 2\nu\left[\frac{1}{r}\frac{\partial u_\Theta}{\partial r} + \frac{u_\Theta}{r^2}\right] - \frac{\partial}{\partial z}\left[\nu\frac{\partial u_\Theta}{\partial z}\right]$$

$$\rho^*\left[\frac{\partial w}{\partial t} + u_r\frac{\partial w}{\partial r} + w\frac{\partial w}{\partial z}\right] = \rho^* B_z - \frac{\partial p^*}{\partial z} + \frac{\partial}{\partial r}\left[\nu\left(\frac{\partial w}{\partial r} + \frac{\partial u_r}{\partial z}\right)\right] + \frac{\nu}{r}\left[\frac{\partial w}{\partial r} + \frac{\partial u_r}{\partial z}\right]$$

$$+ \frac{\partial}{\partial z}\left[(\eta + 2\nu)\frac{\partial w}{\partial z}\right] + \frac{\partial}{\partial z}\left[\eta\left(\frac{\partial u_r}{\partial r} + \frac{u_r}{r}\right)\right] \qquad \text{(A4)}$$

$$\rho^*\left[\frac{\partial I}{\partial t} + u_r\frac{\partial I}{\partial r} + w\frac{\partial I}{\partial z}\right] = \rho^* H^* - \left[\frac{\partial h_r}{\partial r} + \frac{h_r}{r} + \frac{\partial h_z}{\partial z}\right] - p^*\left[\frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{\partial w}{\partial z}\right]$$

$$+ \eta\left[\frac{\partial u_r}{\partial r} + \frac{u_r}{r} + \frac{\partial w}{\partial z}\right]^2$$

$$+ 2\nu\left[\left(\frac{\partial u_r}{\partial r}\right)^2 + \left(\frac{u_r}{r}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2 - \frac{u_\Theta}{r}\frac{\partial u_\Theta}{\partial r}\right]$$

$$+ \nu\left[\left(\frac{\partial u_\Theta}{\partial r}\right)^2 + \left(\frac{u_\Theta}{r}\right)^2 + \left(\frac{\partial u_r}{\partial z} + \frac{\partial w}{\partial r}\right)^2 + \left(\frac{\partial u_\Theta}{\partial z}\right)^2\right]$$

APPENDIX B

BOUNDARY NODE FORMULATION FOR A ZERO NORMAL VELOCITY BOUNDARY CONDITION

Let Q be a node at a stationary wall where the tangent is defined. The normal velocity being zero at Q implies that

$$u(\bar{x},\bar{y},t)\ \sin\alpha - v(\bar{x},\bar{y},t)\ \cos\alpha = 0 \tag{B1}$$

where $(\bar{x},\bar{y})$ are the spatial coordinates of node Q and α is the angle between the tangent line at Q and the x-axis. Since the wall is stationary α is a constant and equation (B1) holds for all times. By translating the axes to $(\bar{x},\bar{y},\bar{t})$, (the plane corresponding to $\bar{t}$ is the initial plane), and using the interpolating functions for u and v, equations (9) and (10), respectively, equation (B1) becomes

$$[u_o(0,0,0,) + a_3 t]\ \sin\alpha - [v_o(0,0,0) + a_6 t]\ \cos\alpha = 0, 0 \leq t \leq \Delta t. \tag{B2}$$

Since the solid wall is stationary, equation (B2) reduces to

$$a_3\ \sin\alpha - a_6\ \cos\alpha = 0\ . \tag{B3}$$

To minimize the least square residual error over the finite element at the point $(\bar{x},\bar{y},\bar{t} + \Delta t)$ subject to equation (B3), we use Lagrange multipliers to obtain

$$\cos\alpha \left\{ \sum_{\ell=1}^{NUMTRI} \iiint_{\ell\text{th prism}} \left(\sum_{k=1}^{4} D_k \frac{\partial D_k}{\partial a_3} \right) dxdydt \right\}$$

$$+\sin\alpha \left\{ \sum_{\ell=1}^{NUMTRI} \iiint_{\ell\text{th prism}} \left(\sum_{k=1}^{4} D_k \frac{\partial D_k}{\partial a_6} \right) dxdydt \right\} = 0, \tag{B4}$$

$$a_3 \sin\alpha - a_6\ \cos\alpha = 0. \tag{B5}$$

In the minimization of the residual error over the boundary type element at which zero normal velocity is imposed, equations (B4) and (B5) replace equations $\partial E(\vec{a})/\partial a_3 = 0$ and $\partial E(\vec{a})/\partial a_6 = 0$ of equation system (13), respectively.

APPENDIX C

EXACT SOLUTIONS OF TEST PROBLEMS

Cylindrical Blast Wave. Sedov's solution[8] (pp. 219-20) for a cylindrical blast wave generated by the instantaneous release of a finite amount of energy proportional to E_o into a gas with initial density ρ_o is summarized below. For the calculations, the specific heat ratio of a perfect gas γ and two constants E_o and ρ_o are:

$$\gamma = 1.40,$$

$$E_o = 6.887025656E{+}06 \text{ [J/m]}, \tag{C1}$$

$$\rho_o = 1.262523446 \text{ [kg/m3]} .$$

The similarity solution which uses the Rankine-Hugoniot strong shock conditions can be expressed in terms of a parameter β as follows:

$$r_s = 48.32791537\cdot(t)^{1/2} \text{ [m]} ,$$

$$v_{r_s} = 973.1614183/r_s \text{ [m/s]} ,$$

$$\rho_s = 7.575140676 \text{ [kg/m}^3\text{]} ,$$

$$p_s = 1.434797011\text{x}10^6/r_s^2 \text{ [Pa]} , \tag{C2}$$

$$r/r_s = 0.9562806477(14\beta - 5)^{1/7} (5\beta - 7\beta^2)^{-1/2},$$

$$v_r/v_{r_s} = 2.40 \cdot \beta \cdot r/r_s ,$$

$$\rho/\rho_s = 2.512993254\text{x}10^{-4} \cdot (14\beta - 5)^{5/7} [(5-7\beta)/(1-2\beta)]^{10/3},$$

$$p/p_s = 6.618423368\text{x}10^{-3} \cdot \beta \cdot [(5-7\beta)/(1-2\beta)]^{7/3} ,$$

where the subscript s denotes the value at the shock front and the parameter β satisfies the inequalities $5/14 \leq \beta \leq 5/12$. Note that $\beta = 5/14$ corresponds to the origin of the blast, a singularity for the energy and sound speed and $\beta = 5/12$ corresponds to the position of the shock front at a given time. The internal energy for a perfect gas of specific heat ratio $\gamma = 1.4$ is given by $e = 2.5\ p/\rho$. The x and y components of the velocity are easily computed by the formulas $v_r\cos\theta$ and $v_r\sin\theta$, respectively, where the $\tan\theta = y/x$.

Propagating Normal Shock. The quiescent state into which the normal shock is propagating is characterized by

$$
\begin{aligned}
u_1 &= v_1 = 0.0 \text{ [m/s]} , \\
p_1 &= 1.01325\text{x}10^5 \text{ [Pa]} , \\
\rho_1 &= 1.225570786 \text{ [kg/m}^3\text{]} , \\
e_1 &= 2.06689408\text{x}10^5 \text{ [J/kg]} ,
\end{aligned}
\tag{C3}
$$

where e_1 is computed from the perfect gas formula $e = 2.5p/\rho$ for $\gamma = 1.4$. The Rankine-Hugoniot relations for a normal shock wave in a perfect gas[11] imply:

$$\frac{\rho_2}{\rho_1} = \frac{1 + \frac{\gamma+1}{\gamma-1}\frac{p_2}{p_1}}{\frac{\gamma+1}{\gamma-1} + \frac{p_2}{p_1}} ,$$

$$v_2 = \sqrt{\frac{p_1}{\gamma\rho_1}}\left(\frac{p_2}{p_1} - 1\right)\left[\frac{\frac{2\gamma}{\gamma+1}}{\frac{p_2}{p_1} + \frac{\gamma-1}{\gamma+1}}\right]^{1/2} , \tag{C4}$$

$$c_s = \left[\frac{p_1(\gamma-1)}{2\rho_1} + \frac{p_2(\gamma+1)}{2\rho_1}\right]^{1/2} ,$$

$$e_2 = \frac{1}{\gamma-1}\frac{p_2}{\rho_2} ,$$

---

[11] *H.W. Liepman and A. Roshko, Elements of Gas Dynamics, Wiley and Sons, 1967, pp. 62-65.*

where the subscript 1 and 2 denote the value of the variables in the quiescent and disturbed states, respectively, c_s is the shock front velocity and $\gamma = 1.4$. In particular for a shock of strength (p_2/p_1) five and a quiescent state characterized by equations (C3), we have

$$
\begin{aligned}
v_2 &= 4.6190563\text{x}10^2 \text{ [m/s]} , \\
u_2 &= 0.0 \text{ [m/s]} , \\
p_2 &= 5.06625\text{x}10^5 \text{ [Pa]} , \\
\rho_2 &= 3.4538813 \text{ [kg/m}^3\text{]}, \\
e_2 &= 3.6670701\text{x}10^5 \text{ [J/kg]} .
\end{aligned}
\qquad \text{(C5)}
$$

APPENDIX D

LISTING OF COMPUTER PROGRAM

```
C**** THE FOLLOWING PILOT HYDRO-CODE USING A FINITE-ELEMENT
C      METHOD TO SOLVE THE UNSTEADY GAS FLOW EQUATIONS.  THE FOLLOWING LISTING
C      CORRESPONDS TO THE VERSION OF THE CODE WHICH PRODUCED FIGURES 7,8,9,10 OF
C      THIS BRL REPORT
C CONTENTS OF COMMON BLOCKS BLOK1,BLOK2 AND BLOK3 ARE
C      XY(1,J)=ABSCISSA OF NODE J
C      XY(2,J)=ORDINATE OF NODE J
C      K=KOLD OR KNEW-TIME LEVEL AT WHICH THE ABOVE VARIABLES ARE KNOWN
C      OR UNKNOWN
C      VAR(K,1,J)=ABSCISSA VELOCITY COMPONENT AT NODE J
C      VAR(K,2,J)=ORDINATE VELOCITY COMPONENT AT NODE J
C      VAR(K,3,J)=PRESSURE AT NODE J
C      VAR(K,4,J)=SPECIFIC INTERNAL ENERGY AT NODE J
C      VART(L,J)=PARTIAL DERIVATIVE OF VAR(KNEW,L,J) WITH RESPECT TO TIME
C      AT NODE J
C      NR(M,J)=NUMBER OF THE NEIGHBORING NODE M TO NODE J
C DEFINITION OF VARIABLES
C      NUMBER=TOTAL NUMBER OF NODES IN THE PROBLEM,.LE.2000
C      NUMOND=TOTAL NUMBER OF NEIGHBORS FOR A GIVEN NODE
C      NOEQ=NUMBER OF DIFFERENTIAL EQUATIONS TO BE SOLVED, MOST OFTEN 4
C      MUA=NUMBER OF UNKNOWN PARAMETERS A(I) I=1,2,...MUA,MOST OFTEN 12
C      ORGTIM=ORIGINAL OR INITIAL TIME
C      MXNTMS=MAXIMUM NUMBER OF TIME STEPS
C
      COMMON/BLOK1/XY(2,2000),NR(9,2000)/BLOK2/VAR(2,4,2000)/BLOK3/VART(
     14,2000)
C
C FOR SHOCK PROBLEM, THE SHOCK STRENGTH (RATIO OF PRESSURE) MUST BE
C SET IN BOTH EXTSOL AND GRAPHIT **MANUALLY***
C CERTAIN OTHER PARAMETERS MUST BE SET MANUALLY IN EXTSOL

      CALL START(NUMBER,NUMOND,NOEQ,MUA,KOLD,KNEW,ORGTIM,MXNTMS,BEGPOS)
      INDEX=0
      OTIME=ORGTIM
      TIME=ORGTIM
   19 INDEX=INDEX+1
      CALL TIMESTEP(NUMBER,NUMOND,NOEQ,MUA,KOLD,KNEW,OTIME,TIME,INDEX)
      CALL DISPLAY(NUMBER,KOLD,KNEW,TIME,INDEX,ORGTIM,BEGPOS)
      IF(INDEX .LT. MXNTMS) GO TO 19
      PRINT 52
   52 FORMAT(1H,13X,'PROGRAM IS FINISHED-STOP')
      STOP
      END

      SUBROUTINE START(NUMBER,NUMOND,NOEQ,MUA,KOLD,KNEW,ORGTIM,MXNTMS,
     1 BEGPOS)
C    **START-GENERATES ENTRIES FOR ARRAYS XY,NR,VAR(KNEW,4,2000),VART
C           -ASSIGNS NUMBERS TO ITS ARGUMENT LIST
      COMMON/BLOK1/XY(2,2000),NR(9,2000)/BLOK2/VAR(2,4,2000)/BLOK3/VART(
     14,2000)
C   ** THE FOLLOWING ARE GEOMETRIC PARAMETERS FOR A NORMAL SHOCK PROBLEM
C      Y1=INITIAL VALUE OF Y IN DOMAIN
      Y1=2.965
C      Y2=FINAL VALUE OF Y IN DOMAIN
      Y2=3.050
C      DY=SIZE OF MESH IS Y DIRECTION
      DY=0.005
C      NOXD=NUMBER OF DIVISIONS IN X-DIRECTION
      NOXD=4
C      BEGPOS IS BEGINNING POSITION OF SHOCK
```

```
      BEGPOS=3.0025
C     XBEGIN=INITIAL X POSITION
      XBEGIN=0.0
      ORGTIM=0.0
      MXNTMS=14
      CALL MESGEN(NUMBER,Y1,Y2,DY,NCXD,XBEGIN)
      PRINT 103
  103 FORMAT( 1H,//,20X,' MATRICES XY(2,2000) AND NR(9,2000) ARE',//)
      PRINT 104
  104 FORMAT  (1H,5X,'NODE',13X,'CORRDINATES',25X,'NEIGHBORING NODES(1-8
     1)',24X,'NODE ')
      PRINT 105
  105 FORMAT (1H,105X,'TYPE')
      PRINT 101
  101 FORMAT (1H,6X,'I',11X,'X',17X,'Y',13X,'1',6X,'2',6X,'3',6X,'4',6X,
     1'5',6X,'6',6X,'7',6X,'8',6X,'9')
      DO 51, I=1,NUMBER
   51 PRINT 102,I,XY(1,I),XY(2,I),(NR(J,I),J=1,9)
  102 FORMAT (1H, 2X,I5,3X,1PE15.8,3X,1PE15.8,1X, 9(2X,I5))
      NUMONC=9
      NOEC=4
      KNEW=1
      KOLD=2
      MUA=12
      DO 10 NONUM=1,NUMBER
      X=XY(1,NONUM)
      Y=XY(2,NONUM)
      T=ORGTIM
      CALL EXTSOL(X,Y,T,UV,VV,P,E,SPECV,IBAD)
      IF(IBAD.NE.0)GOTO 20
      VAR(KNEW,1,NONUM)=UV
      VAR(KNEW,2,NONUM)=VV
      VAR(KNEW,3,NONUM)=P
      VAR(KNEW,4,NONUM)=E
   10 CONTINUE
      CALL INITNG(NUMBER,NUMOND,NOEQ,KNEW,MUA)
      CALL OUTNO4(NUMBER,KNEW,ORGTIM)
      RETURN
   20 PRINT 21,NONUM
   21 FORMAT(1H,/,'  TROUBLE WAS INCURRED IN FINDING   EXACT  VALUE AT
     1ODE=',I6,/,'  PROGRAM IS STOPPED BY PROGRAMMER')
      STOP
      RETURN
      END

      SUBROUTINE TIMESTEP(NUMBER,NUMOND,NOEQ,MUA,KOLD,KNEW,OTIME,TIME,
     1 INDEX)
C  COMPUTES VALUES OF FLOW VARIABLES AT INITIAL TIME+DT AT ALL NODES
C KOLD INDICATES LEVEL OF KNOWN INFO IN VAR
C KNEW INDICATES LEVEL OF UNKNOWN INFO IN VAR
      COMMON/BLOK1/XY(2,2000),NR(9,2000)/BLOK2/VAR(2,4,2000)/BLOK3/VART(
     14,2000)
C    **THE FOUR KEY SUBROUTINES OF CALCI ARE TRANSL,SOLVER,INITZR,VALNEW
      EXTERNAL TRANSL,SOLVER,INITZR,VALNEW
C SWITCH INDICATORS FOR CALCULATION AT NEW TIME LEVEL
      OTIME=TIME
      ISAVE=KNEW
      KNEW=KOLD
      KOLD=ISAVE
C  COMPUTE TIME INCREMENT
```

```
      CALL DELTAT(NUMBER,NUMOND,NOEQ,KOLD,DT)
C   COMPUTE NEW TIME
      TIME=OTIME+DT
      PRINT 4000,OTIME,DT ,TIME
 4000 FORMAT (1H,/,7X,'SUB TIMESTEP    OLD TIME=',1PE15.8,/,22X,'DELTAT='
     1,1PE15.8,/,22X,'TIME TO BE USED IN CALCULATIONS=',1PE15.8)
C   LOOP OVER NODES
C     **BECAUSE ALL THE VALUES OF THE FLOW VARIABLES ARE EQUAL FOR CONSTANT
C       Y VALUES ONLY ONE VALUE AT A GIVEN Y MUST BE CALCULATED
      NONUM=3
   28 CONTINUE
C   IF NR(NUMOND,NONUM)=1,USE CALCI(NON-BDY NODE),ELSE CALCB(BDY NODE)
      IF(NR(NUMOND,NONUM).EQ.1)GOTO 25
      CALL CALCB(NONUM,KNEW,TIME,INDEX)
      GOTO 20
   25 CALL CALCI(NUMBER,NUMOND,NOEQ,MUA,KOLD,KNEW,TIME,DT,NONUM,TRANSL,S
     1OLVER,INITZR,VALNEW)
C     **USE THE VALUE AT THE NODE(NONUM) TO COMPUTE VAR AND VART ENTRIES FOR
C       A GIVEN Y VALUE
   20 CHKRD=XY(2,NONUM)
      RADVEL=VAR(KNEW,2,NONUM)
      N=NONUM-3
   26 N=N+1
      IF(  N    .GT. NUMBER) RETURN
      RD=XY(2,N)
      IF(RD .GE. (CHKRD+0.000001)) GO TO 27
      VAR(KNEW,1,N)=VAR(KNEW,1,NONUM)
      VAR(KNEW,2,N)=RADVEL
      VAR(KNEW,3,N)=VAR(KNEW,3,NONUM)
      VAR(KNEW,4,N)=VAR(KNEW,4,NONUM)
      VART(1,N)=VART(1,NONUM)
      VART(2,N)=VART(2,NONUM)
      VART(3,N)=VART(3,NONUM)
      VART(4,N)=VART(4,NONUM)
      GO TO 26
   27 NONUM=N+2
      GO TO 28
      END

      SUBROUTINE DISPLAY(NUMBER,KOLD,KNEW,TIME,INDEX,ORGTIM,BEGPOS)
C     **DISPLAY-GENERATES OUTPUT AFTER THE FLOW VARIABLES ARE COMPUTED AT A NEW
C       TIME
      COMMON/BLOK2/VAR(2,4,2000)/BLCK1/XY(2,2000),NR(9,2000)
C     **OUTN06 GIVES LISTING OF VAR AND DENSITY AT NEW TIME
      CALL OUTN06(NUMBER,TIME,KNEW)
C     **GRAPHIT USES THE CALCOMP PLOTTER TO GRAPH PRESSURE,Y-VELOCITY, INTERNAL
C       ENERGY PER UNIT MASS AND DENSITY
      CALL GRAPHIT (TIME,KNEW,NUMBER,INDEX,ORGTIM,BEGPOS)
      RETURN
      END

      SUBROUTINE INITNG(NUMBER,NUMOND,NOEQ,KNEW,MUA)
C COMPUTES INITIAL VALUES OF THE PARAMETERS OF FLOW VARIABLES AT THE INITIAL
C TIME LEVEL ONLY
      COMMON/BLOK1/XY(2,2000),NR(9,2000)/BLOK2/VAR(2,4,2000)/BLOK3/VART(
     14,2000)
      DIMENSION ALPHA(12),RHOINT(3,9),X(3),Y(3),GT(4,3),G(4,3),DOG(4,2),
     1AI(12),RHOT(3),TRHOIN(3,3)
C INITNG IS WRITTEN FOR MUA=12. CHANGES MUST BE MADE IF INITNG IS NOT 12. INITNG
C CHECKS FOR MUA=12.
```

```
      IF(MUA.NE.12)GOTO 101
C THE NUMBER OF NODES IN DOMAIN(NUMBER) CANNOT BE GREATER THAN 2000. INITNG
C CHECKS NUMBER
      IF(NUMBER.GT.2000)GOTO 201
C LOOP OVER ALL NODES
      DO 20 NONUM=1,NUMBER
      DO 25 J=1,MUA
   25 ALPHA(J)=0.0
C COUNT NUMBER OF NODES PER PRISM --NDPPRM
      NDPPRM=1
      NONM1=NUMOND-1
      DO 30 J=1,NONM1
      IF(NR(J,NONUM).EQ.0)GOTO 35
   30 NDPPRM=NDPPRM+1
C COMPUTE VARIABLE FROM EQUATION OF STATE AND ITS FIRST DERIVATIVES AT PRISM
C NODES
   35 CALL INARES(NUMOND,NOEQ,KNEW,NONUM,NDPPRM,RHOINT)
C TRANSFER CENTER NODE DATA TO WORKING ARRAYS
      X(1)=XY(1,NONUM)
      Y(1)=XY(2,NONUM)
      DO 40 I=1,NOEQ
   40 G(I,1)=VAR(KNEW,I,NONUM)
C DETERMINE NUMBER OF TRIANGLES PER PRISM NDPPRM-1 FOR NON-BDY AND NDPPRM-2 FOR
C BDY NODE
      NUMTRI=NDPPRM-1
      IF(NR(NUMOND,NONUM).NE.1)NUMTRI=NDPPRM-2
C LOOP OVER THE TRIANGLES OF THE PRISM
      RNT=NUMTRI
      DO 45 ITRI=1,NUMTRI
C THIRD VERTEX OF TRIANGLE IS
      IV3=ITRI+1
C EXCEPT WHEN CENTER NODE IS NOT ON BOUNDARY AND LAST NGHB. NODE OF PRISM IS
C CONSIDERED
      IF(ITRI.EQ.NDPPRM-1)IV3=1
C TRANSFER DATA OF NEIGHBORING NODES TO WORKING ARRAYS
      JSUB2=NR(ITRI,NONUM)
      JSUB3=NR(IV3,NONUM)
      X(2)=XY(1,JSUB2)
      X(3)=XY(1,JSUB3)
      Y(2)=XY(2,JSUB2)
      Y(3)=XY(2,JSUB3)
C ABSOLUTE VALUE OF DET IS THE AREA OF THE TRIANGLE/2
      DET=X(1)*Y(3)-X(3)*Y(1)+X(2)*Y(1)-X(1)*Y(2)+X(3)*Y(2)-X(2)*Y(3)
      IF(DET.EQ.0.)GOTO 19
C DOG(I,J)=PARTIAL DERIVATIVE OF VAR(KNEW,I,NONUM) WITH RESPECT TO X(J=1) OR
C     Y(J=2) ON THE TRIANGLE WITH CENTER AT NONUM
      DO 50 I=1,NOEQ
      G(I,2)=VAR(KNEW,I,JSUB2)
      G(I,3)=VAR(KNEW,I,JSUB3)
      DOG(I,1)=( (G(I,1)-G(I,2))*Y(3)-(G(I,1)-G(I,3))*Y(2)+(G(I,2)-G(I,3
    1 ))*Y(1))/DET
   50 DOG(I,2)=(-(G(I,1)-G(I,2))*X(3)+(G(I,1)-G(I,3))*X(2)-(G(I,2)-G(I,3
    1 ))*X(1))/DET
C TRANFER INFO FROM RHOINT TO TRHOIN AND COMPUTE APPROXIMATE DERIVATIVES OF U,V
C AND INT ENG WITH RESPECT TO TIME BY SOLVING FOR TIME DERIVATIVES IN THE
C     GOVERNING EQUATIONS
      DO 55 I=1,3
      TRHOIN(I,1)=RHOINT(I,1)
      TRHOIN(I,2)=RHOINT(I,ITRI+1)
      TRHOIN(I,3)=RHOINT(I,IV3+1)
```

```
      GT(1,I)=-(G(1,I)*DOG(1,1)+G(2,I)*DOG(1,2)+DOG(3,1)/TRHOIN(1,I))
      GT(2,I)=-(G(1,I)*DOG(2,1)+G(2,I)*DOG(2,2)+DOG(3,2)/TRHOIN(1,I))
   55 GT(4,I)=-(G(1,I)*DOG(4,1)+G(2,I)*DOG(4,2)+G(3,I)*(DOG(1,1)+DOG(2,2
     1))/TRHOIN(1,I))
C COMPUTE INITIAL VALUES OF PARAMETERS FOR U,V, INT. ENG.(A(1-6,10-12))
C SOLVE FOR A(1),A(2),A(3) BY EQUATING A(1)*X(I)+A(2)*Y(I)+A(3)=GT(1,I),I=1,2,3
      DO 60 I=1,NOEQ
      IF(I.EQ.3)GOTO 60
      J=I*3
      JM1=J-1
      JM2=J-2
      AI(J)=(GT(I,1)*(X(3)*Y(2)-X(2)*Y(3))+GT(I,2)*(X(1)*Y(3)-Y(1)*X(3))
     1      +GT(I,3)*(X(2)*Y(1)-X(1)*Y(2)))/DET
      AI(JM1)=(-GT(I,1)*(X(3)-X(2))+GT(I,2)*(X(3)-X(1))-GT(I,3)*(X(2)-X(
     11))))/DET
      AI(JM2)=(GT(I,1)*(Y(3)-Y(2))-GT(I,2)*(Y(3)-Y(1))+GT(I,3)*(Y(2)-Y(1
     1))))/DET
   60 CONTINUE
C COMPUTE DERIVATIVES OF DENSITY AND PRESSURE WITH RESPECT TO TIME AND
C PARAMETERS A(7-9) FOR THE PRESSURE
      DO 65 I=1,3
C RHOT IS THE TIME DERIVATIVE OF DENSITY
      RHOT(I)=-(G(1,I)*(TRHOIN(2,I)*DOG(3,1)+TRHOIN(3,I)*DOG(4,1))
     1          +G(2,I)*(TRHOIN(2,I)*DOG(3,2)+TRHOIN(3,I)*DOG(4,2))
     2          +TRHOIN(1,I)*(DOG(1,1)+DOG(2,2)))
C GT(3,I) IS THE TIME DERIVATIVE OF PRESSURE AT VERTICES
   65 GT(3,I)=(RHOT(I)-TRHOIN(3,I)*(AI(10)*X(I)+AI(11)*Y(I)+AI(12)))/TRH
     1OIN(2,I)
      AI(7)=(GT(3,1)*(Y(3)-Y(2))-GT(3,2)*(Y(3)-Y(1))+GT(3,3)*(Y(2)-Y(1))
     1)/DET
      AI(8)=(-GT(3,1)*(X(3)-X(2))+GT(3,2)*(X(3)-X(1))-GT(3,3)*(X(2)-X(1)
     1))/DET
      AI(9)=(GT(3,1)*(X(3)*Y(2)-X(2)*Y(3))+GT(3,2)*(X(1)*Y(3)-Y(1)*X(3))
     1      +GT(3,3)*(X(2)*Y(1)-X(1)*Y(2)))/DET
C SUM THE A(I)'S OVER THE TRIANGLES IN THE GIVEN PRISM
      DO 70 I=1,MUA
   70 ALPHA(I)=ALPHA(I)+AI(I)
   45 CONTINUE
C THE AVERAGE OF THE A(I)'S IS THE INITIAL VALUE OF THE A(I)'S FOR THE PRISM
      DO 75 I=1,MUA
   75 AI(I)=ALPHA(I)/RNT
C STORE IN MATRIX VART THE INITIAL VALUES OF THE PARAMETERS AT THE CENTER NODE
C IN THE TRANSLATED COORDINATE SYSTEM. WE TAKE A(1,2,4,5,7,8,10 AND 11)=0
C AND A(I*3)=VART(I,NONUM)
      DO 80 I=1,NOEQ
   80 VART(I,NONUM)=AI(I*3-2)*XY(1,NONUM)+AI(I*3-1)*XY(2,NONUM)+AI(I*3)
   20 CONTINUE
      RETURN
  101 PRINT 102,MUA
      STOP
  201 PRINT 202,NUMBER
      STOP
   19 PRINT 5,NONUM
      STOP
    5 FORMAT(1H,//,' DET=0.0 IN SUB. INITNG',/,'  ONE OF THE TRIANGLES
     1ASSOCIATED WITH NODE',I5,' HAS ZERO AREA',/,' CHECK THE ORDERING A
     1ND NUMBERING OF NGHB. NODES',/,'  STOP')
  102 FORMAT (1H,' MUA=',I5,'  BUT SUBROUTINE INITNG IS WRITTEN FOR MUA=
     112,STOP')
  202 FORMAT (1H, ' NUMBER=',I7,' BUT INITNG IS WRITTEN FOR NUMBER NOT
```

```
     1GREATER THAN 2000. STOP')
      END

      SUBROUTINE MESGEN(NONUMF,R1,R2,DY,NOTD,THETAB)
C    THIS SET OF SUBROUTINES  GENERATE A FINITE ELEMENT MESH IN CARTESIAN
C    COORDINATES(X,Y)  THEY WERE ADAPTED FROM  THE CLYDRICAL COOR. PROGRAM
C    R IS THE Y COORDINATE AND THETA THE X COORDINATE
C*****FINITE ELEMENT GENERATING PACKAGE MAY BE CONFUSING IN ITS LOGIC
C      BECAUSE IT HAS BEEN ADAPTED FROM POLAR TO RECTANGULAR
C      GEOMETRY
      COMMON/BLOK1/XY(2,2000),NR(9,2000)
      DIMENSION RADII(200)
C NOTD IS NUMBER OF DIVISIONS IN THE X DIRECTION
      NOTDP1=NOTD+1
C THETAB IS INITIAL X POSITION
C DY IS INCREMENT IN Y-DIRECTION
C R1 IS INITIAL VALUE OF Y
C R2 IS FINAL VALUE OF Y
C DR IS INCREMENT IN X-DIRECTION TO INSURE EQUILATERAL TRIANGLES
      DR=1.154700538*DY
      THETAE=THETAB+NOTD*DR
C NODPOS INDICATES TYPE OF POSITION THE FIRST NODE WILL HAVE-SEE AUTGEN
      NODPOS=1
      NONUMB=NOTD+2
      RADII(1)=R1
      I=1
   23 I=I+1
      RADII(I)=RADII(I-1)+DY
      IF( (RADII(I)+0.00001) .LT. R2) GO TO 23
      PRINT 24,(RADII(IKL),IKL=1,I)
   24 FORMAT (1H,10X,'THE Y SUBDIVISIONS FROM SUBPROGRAM MESGEN FOR X-Y
     1 FIN. EL. MESH ARE',(/,13X,1PE15.8))
C AUTGEN GENERATES GRID FOR Y.GT.Y1=R1
      CALL AUTGEN (RADII,I,NODPOS,NONUMB,NOTD,THETAB,THETAE,NONUMF)
C GENERATE GRID FOR Y=Y1=R1
      DO 20 I=1,NOTDP1
      NR(9,I)=7
      DO 20 J=5,8
   20 NR(J,I)=0
      DO 21 I=2,NOTDP1
      NR(1,I)=I-1
      NR(2,I)=I+NOTD+1
      NR(3,I)=I+NOTD+2
   21 NR(4,I)=I+1
      NR(1,1)=NOTD+2
      NR(2,1)=NOTD+3
      NR(3,1)=2
      NR(4,1)=0
      NR(4,16)=0
      DTHETA=(THETAE-THETAB)/NOTD
      DO 22 I=1,NOTDP1
      RI=I-1
      THETA=THETAB+DTHETA*RI
      XY(1,I)=THETA
   22 XY(2,I)=R1
      RETURN
      END
```

```
      SUBROUTINE AUTGEN(RADII,N,NODPOS,NONUMB,NOTD,THETAB,THETAE,NONUMF)
C   THE AUTOMATIC GENERATION OF THE FINITE ELEMENT GRID STARTS WITH R1+
C   CELTAR1=RADII(2)
C   NOTC IS NUMBER OF THETA DIVISICNS,   NONUMF=TOTAL NUMBER OF NODES (NONUMB+
C   THOSE GENERATEC
      COMMON/BLOK1/XY(2,2000),NR(9,2000)
      CIMENSION RACII(200)
      PI=3.14159265358979324
C   NONUMB=THE BEGINNING NODE NUMBER AT RADII(2).
      NONUM=NONUMB-1
      NDP1=NOTD+1$ NDP2=NOTC+2
      DTHETA=(THETAE-THETAB)/NOTD
C   NODPOS=0 IF NODE NONUMB IS A VERTEX OF A COMPLETE EQUILATERAL TRIANGLE,
C   IF NOT NODPOS=1
      KNOW=NODPOS
      KK=KNOW+1
      KBEFOR=XMCDF(KK,2)
      CO 20 I=2,N
      IF(KNOW.NE.1)GOTO 60
C   VERTEX OF AN INCOMPLETE EQUILATERAL TRIANGLE IS ON THE RIGHT BOUNDARY
      IF(I.EQ.N)GOTO 30
C   R1.LT.RADII.LT.R2 AND THETA=THETAB
      NONUM=NONUM+1
      XY(1,NONUM)=THETAB
      XY(2,NONUM)=RADII(I)
      NR(9,NONUM)=2
      NR(1,NONUM)=NONUM-NDP1
      NR(2,NONUM)=NONUM+1
      NR(3,NONUM)=NONUM+NDP2
      CO 36 J=4,8
   36 NR(J,NONUM)=0
      THETA=THETAB+0.5*DTHETA
C   R1.LT.RADII.LT.R2 AND THETA=THETAB
C   R1.LT.RADII.LT.R2 AND THETAB.LT.THETA.LT.THETAE
   40 NONUM=NONUM+1
      XY(1,NONUM)=THETA
      XY(2,NONUM)=RADII(I)
      NR(9,NONUM)=1
      NR(1,NONUM)=NONUM-1
      NR(2,NONUM)=NONUM+NDP2-1
      NR(3,NONUM)=NONUM+NDP2
      NR(4,NONUM)=NONUM+1
      NR(5,NONUM)=NONUM-NDP1
      NR(6,NONUM)=NONUM-NDP1-1
      NR(7,NONUM)=0
      NR(8,NONUM)=0
      THETA=THETA+DTHETA
      IF(THETA.LE.THETAE)GOTO 40
C   R1.LT.RADII.LT.R2 AND THETA=THETAE
      NONUM=NONUM+1
      XY(1,NONUM)=THETAE
      XY(2,NONUM)=RADII(I)
      NR(9,NONUM)=3
      NR(1,NONUM)=NONUM-NDP2
      NR(2,NONUM)=NONUM-1
      NR(3,NONUM)=NONUM+NDP1
      CO 39 J=4,8
   39 NR(J,NONUM)=0
      GOTO 50
C   VERTEX OF A COMPLETE EQUILATERAL TRIANGLE IS ON THE RIGHT BOUNDARY
```

```
   60 IF(I.EQ.N)GOTO 35
      NONUM=NONUM+1
C     R1.LT.RADII.LT.R2 AND THETA=THETAB
      XY(1,NONUM)=THETAB
      XY(2,NONUM)=RADII(I)
      NR(9,NONUM)=2
      NR(1,NONUM)=NONUM-NDP2
      NR(2,NONUM)=NONUM-NDP2+1
      NR(3,NONUM)=NONUM+1
      NR(4,NONUM)=NONUM+NDP1+1
      NR(5,NONUM)=NONUM+NDP1
      NR(6,NONUM)=0
      NR(7,NONUM)=0
      NR(8,NONUM)=0
      THETA=THETAB+DTHETA
C     R1.LT.RADII.LT.R2 AND THETAB.LT.THETA.LT.THETAE
   44 NONUM=NONUM+1
      XY(1,NONUM)=THETA
      XY(2,NONUM)=RADII(I)
      NR(9,NONUM)=1
      NR(1,NONUM)=NONUM-1
      NR(2,NONUM)=NONUM+NDP1
      NR(3,NONUM)=NONUM+NDP1+1
      NR(4,NONUM)=NONUM+1
      NR(5,NONUM)=NONUM-NDP2+1
      NR(6,NONUM)=NONUM-NDP2
      NR(7,NONUM)=0
      NR(8,NONUM)=0
      THETA=THETA+DTHETA
      IF(THETA.LT.(THETAE-DTHETA/2.))GOTO 44
C     R1.LT.RADII.LT.R2 AND THETA=THETAE
      NONUM=NONUM+1
      XY(1,NONUM)=THETAE
      XY(2,NONUM)=RADII(I)
      NR(9,NONUM)=3
      NR(1,NONUM)=NONUM-NDP1
      NR(2,NONUM)=NONUM-NDP1-1
      NR(3,NONUM)=NONUM-1
      NR(4,NONUM)=NONUM+NDP2-1
      NR(5,NONUM)=NONUM+NDP2
      NR(6,NONUM)=0
      NR(7,NONUM)=0
      NR(8,NONUM)=0
   50 ISAVE=KNOW
      KNOW=KBEFOR
      KBEFOR=ISAVE
      GOTO 20
   30 NONUM=NONUM+1
C     RADII=R2 AND THETA=THETAB
      XY(1,NONUM)=THETAB
      XY(2,NONUM)=RADII(I)
      NR(9,NONUM)=5
      NR(1,NONUM)=NONUM+1
      NR(2,NONUM)=NONUM-NDP1
      DO 51 J=3,8
   51 NR(J,NONUM)=0
      THETA=THETAB+0.5*DTHETA
   62 NONUM=NONUM+1
C     RADII=R2 AND THETAB.LT.THETA.LT.THETAE
      XY(1,NONUM)=THETA
```

```
      XY(2,NONUM)=RADII(I)
      NR(9,NONUM)=4
      NR(1,NONUM)=NONUM+1
      NR(2,NONUM)=NONUM-NDP1
      NR(3,NONUM)=NONUM-NDP1-1
      NR(4,NONUM)=NONUM-1
      DO 52 J=5,8
   52 NR(J,NONUM)=0
      THETA=THETA+DTHETA
      IF(THETA.LE.THETAE)GOTO 62
      NONUM=NONUM+1
C  RADII=R2 AND THETA=THETAE
      XY(1,NONUM)=THETAE
      XY(2,NONUM)=RADII(I)
      NR(9,NONUM)=6
      NR(1,NONUM)=NONUM-NDP2
      NR(2,NONUM)=NONUM-1
      DO 53 J=3,8
   53 NR(J,NONUM)=0
      GOTO 20
   35 NONUM=NONUM+1
C  RADII=R2 AND THETA=THETAB
      XY(1,NONUM)=THETAB
      XY(2,NONUM)=RADII(I)
      NR(9,NONUM)=5
      NR(1,NONUM)=NONUM+1
      NR(2,NONUM)=NONUM-NDP2+1
      NR(3,NONUM)=NONUM-NDP2
      DO 54 J=4,8
   54 NR(J,NONUM)=0
      THETA=THETAB+DTHETA
   49 NONUM=NONUM+1
C  RADII=R2 AND THETAB.LT.THETA.LT.THETAE
      XY(1,NONUM)=THETA
      XY(2,NONUM)=RADII(I)
      NR(9,NONUM)=4
      NR(1,NONUM)=NONUM+1
      NR(2,NONUM)=NONUM-NDP2+1
      NR(3,NONUM)=NONUM-NDP2
      NR(4,NONUM)=NONUM-1
      DO 55 J=5,8
   55 NR(J,NONUM)=0
      THETA=THETA+DTHETA
      IF(THETA.LT.(THETAE-DTHETA/2.))GOTO 49
      NONUM=NONUM+1
C  RADII=R2 AND THETA=THETAE
      XY(1,NONUM)=THETAE
      XY(2,NONUM)=RADII(I)
      NR(9,NONUM)=6
      NR(1,NONUM)=NONUM-NDP1
      NR(2,NONUM)=NONUM-NDP1-1
      NR(3,NONUM)=NONUM-1
      DO 56 J=4,8
   56 NR(J,NONUM)=0
   20 CONTINUE
      NONUMF=NONUM
      RETURN
      END

      SUBROUTINE INARES(NUMOND,NOEQ,KNEW,NONUM,NDPPRM,RHOINT)
```

```
C COMPUTES VARIABLE FROM EQUATION OF STATE AND ITS FIRST DERIVATIVES AT PRISM
C NODES
      COMMON/BLOK1/XY(2,2000),NR(9,2000)/BLOK2/VAR (2,4,2000)
      DIMENSION RHOINT(3,NUMOND),R(3)
      CALL INEQST(VAR(KNEW,3,NONUM),VAR(KNEW,4,NONUM),R)
      DO 20 I=1,3
   20 RHOINT(I,1)=R(I)
      DO 25 K=2,NDPPRM
      JSUB=NR(K-1,NONUM)
      CALL INEQST(VAR(KNEW,3,JSUB),VAR(KNEW,4,JSUB),R)
      DO 25 I=1,3
   25 RHOINT(I,K)=R(I)
C IF NUMBER OF TRIANGLES IS LESS THAN MAX NUMBER ALLOWED IN A PRISM, SET
C REMAINDER OF RHOINT TO ZERO.
      IF(NDPPRM.EQ.NUMOND)RETURN
      NPPP1=NDPPRM+1
      DO 30 J=NPPP1,NUMOND
      DO 30 I=1,3
   30 RHOINT(I,J)=0.0
      RETURN
      END

      SUBROUTINE INEQST(P,E,R)
C COMPUTES THE DENSITY AND ITS DERIVATIVE WITH RESPECT TO PRESSURE AND INT.
C ENERGY FOR A NOBEL-ABEL GAS.
C GAMMA AND ETA ARE TWO PARAMETERS IN THE NOBEL-ABEL EQUATION OF STATE
C TWO PARAMETERS ARE GAMMA AND ETA (UNITS FOR ETA ARE M**3/KG)
      DIMENSION R(3)
      DATA GAMMA,ETA/1.40,0.0/
      G=GAMMA-1.0
      D=G*E+ETA*P
      DD=D*D
C R(1)=DENSITY
      R(1)=P/D
C R(2)=FIRST DERIVATIVE OF R(1) WRT PRESSURE
      R(2)=G*E/DD
C R(3)=FIRST DERIVATIVE OF R(1) WRT INTERNAL ENERGY
      R(3)=-G*P/DD
      RETURN
      END

      SUBROUTINE OUTNO4(NUMBER,KNEW,ORGTIM)
C   ** OUTNO4-LISTS THE VALUE OF MATRICES VAR AT KNEW LEVEL AND VART
      COMMON/BLOK2/VAR(2,4,2000)/BLOK3/VART(4,2000)
      PRINT 399,ORGTIM
  399 FORMAT (1H,/,1X,'THE VALUES OF U,V,P,E AND RHO AT THE INITIAL ',
     1 'TIME  (',1PE16.8,')  ARE',/)
      PRINT 397
  397 FORMAT(1H,5X,'NODE',10X,'U',18X,'V',18X,'P',18X,'E',17X,'RHO')
      DO 11 I=1,NUMBER
      RHO=VAR(KNEW,3,I)/(0.4*VAR(KNEW,4,I))
   11 PRINT 402,I,(VAR(KNEW,J,I),J=1,4),RHO
  402 FORMAT(1H,3X,I5,5(4X,1PE15.8))
      PRINT 398,ORGTIM
      PRINT 405,(I,(VART(J,I),J=1,4),I=1,NUMBER)
  398 FORMAT(1H,/'  INITIAL VALUES OF D/DT OF U,V,P,E ARE AT TIME=',
     11PE15.8,//,5X,'NODE',8X,'DU/DT',14X,'DV/DT',14X,'DP/DT',14X,'DE/PT
     2')
  405 FORMAT(1H,(2X,I6,4(4X,1PE15.8)  ))
      RETURN
```

```
      ENC

      SUBROUTINE EXTSOL(X,Y,T,UV,VV,P,E,SPECV,IBAD)
C  ** EXTSOL-FOR A GIVEN X,Y,T,EXTSOL COMPUTES THE X AND Y VELOCITY COMPONENTS,
C             THE PRESSURE,INTERNAL ENERGY BEHIND A NORMAL SHOCK PROPOGATING INTO
C             A CUIESCENT FIELD.  PERFECT GAS IS ASSUMED ON EITHER SIDE
C  ** VEL1,P1,R1,T1 ARE Y-VELOCITY,PRESSURE,DENSITY,TEMPERATURE OF QUIESCENT
C     STATE
      CATA VEL1,P1,R1,T1/0.0,1.01325E5,1.225570786,288.15/
C  ** ORGSHP,ORGSTM,SHKSTH ARE INITIAL POSITION,TIME,STRENGTH OF SHOCK
C  ** G IS RATIO OF SPECIFIC HEATS
      CATA ORGSHP,ORGSTM,G/3.0025,0.0,1.4/
      CATA SHKSTH/ 5.0/
C  ** SPECV AND IBAD ARE UNUSED PARAMETERS
      SPECV=1.0  $  IBAC=0
C  ** A1=SOUNC SPEED
      A1=SQRT(G*P1/R1)
      COM=(G+1.0)/(G-1.0)
C  ** CS=SPEEC OF SHOCK FRONT
      CS=A1*SQRT((G-1.0)/(2.0*G)+(G+1.0)*SHKSTH/(2.0*G))
      POS=Y
      SHKPOS=ORGSHP+CS*(T-ORGSTM)
      IF (POS .GT. SHKPOS) GO TO 15
      UP=(A1/G)*(SHKSTH-1.0)*SQRT (((2.0*G)/(G+1.0))/(SHKSTH+1.0/COM))
      UV=0.0
      VV=UP
      P=SHKSTH*P1
      R=R1*(1.0+COM*SHKSTH)/(SHKSTH+COM)
      E=2.5*P/R
      RETURN
   15 UV=0.0
      VV=0.0
      P=P1
      E=2.5*P/R1
      RETURN
      ENC

      SUBROUTINE CELTAT(NUMBER,NUMOND,NOEQ,KOLD,DT)
C COMPUTES THE TIME INCREMENT FOR THE ENTIRE DOMAIN BY USING BI-CHAR. AND VALUES
C OF THE FLOW VARIABLES AT TIME=OLD TIME
      COMMON/BLOK1/XY(2,2000),NR(9,2000)/BLOK2/VAR(2,4,2000)
      INTEGER START,SKIP,START2
C START IS STARTING NODE  SKIP IS NUMBER OF NODES TO BE SKIPPED
      CATA START,SKIP/1, 1/
C FINC MINIMUM LENGTH(RADII) OF THE FIRST PRISM AND SOUND SPEED AT START NODE
      JSUB=NR(1,START)
      XR=(XY(1,JSUB)-XY(1,START))**2
      YR=(XY(2,JSUB)-XY(2,START))**2
      RMIN=XR+YR
      NONM1=NUMONC-1
      DO 20 I=2,NONM1
      JSUB=NR(I,START)
C IF NR(I,START)=0, THEN THERE IS NO MORE NGHB. NODES TO THE START NODE,ELSE
C THERE ARE
      IF(JSUB .EQ. 0) GO TO 25
      XR=(XY(1,JSUB)-XY(1,START))**2
      YR=(XY(2,JSUB)-XY(2,START))**2
      R=XR+YR
   20 RMIN=MIN1F(RMIN,R)
   25 RMIN=SQRTF(RMIN)
```

```
      CALL SNDSPD(VAR(KOLD,3,START),VAR(KOLD,4,START),A)
C INCLUDE POSSIBILITY FOR UNKNOWN (SWIRL VELOCITY IN POLAR COORD)
      SWRLSP=0.0
      IF(NOEQ.EQ.5)SWRLSP=VAR(KOLD,5,START)
C FIND THE FIRST VALUE OF DELTA T
      TMIN=RMIN/(SQRTF(VAR(KOLD,1,START)**2+VAR(KOLD,2,START)**2
     1            +SWRLSP*SWRLSP)+A)
C INCREMENT START BY SKIP AND PROCEED THROUGH THE FINITE ELEMENT NET
      START2=START+SKIP
      DO 30 NONUM=START2,NUMBER,SKIP
      JSUB=NR(1,NONUM)
      XR=(XY(1,JSUB)-XY(1,NONUM))**2
      YR=(XY(2,JSUB)-XY(2,NONUM))**2
      RMIN=XR+YR
      NONM1=NUMOND-1
      DO 35 I=2,NONM1
      JSUB=NR(I,NONUM)
C IF NR(I,NONUM)=0, THEN THERE IS NO MORE NGHB. NODES TO THE START NODE,ELSE
C THERE ARE
      IF(JSUB.EQ.0)GOTO 40
      XR=(XY(1,JSUB)-XY(1,NONUM))**2
      YR=(XY(2,JSUB)-XY(2,NONUM))**2
      R=XR+YR
   35 RMIN=MIN1F(RMIN,R)
   40 RMIN=SQRTF(RMIN)
      CALL SNDSPD(VAR(KOLD,3,NONUM),VAR(KOLD,4,NONUM),A)
C INCLUDE POSSIBILITY FOR UNKNOWN (SWIRL VELOCITY IN POLAR COORD)
      SWRLSP=0.0
      IF(NOEQ.EQ.5)SWRLSP=VAR(KOLD,5,NONUM)
      TM=RMIN/(SQRTF(VAR(KOLD,1,NONUM)**2+VAR(KOLD,2,NONUM)**2+
     1SWRLSP*SWRLSP)+A)
C FINAL VALUE OF DELTA T IS TMIN
   30 TMIN=MIN1F(TMIN,TM)
      DT=TMIN
      RETURN
      END

      SUBROUTINE CALCI(NUMBER,NUMOND,NOEQ,MUA,KOLD,KNEW,TIME,DT,
     1                 NONUM,TRANSL,SOLVER,INITZR,VALNEW)
C COMPUTES FOR A NON-BOUNDARY NODE NEW INITIAL VALUES FOR THE PARAMETERS AND
C VALUES OF THE FLOW VARIABLES AT THE OLD TIME + DT (KNEW) LEVEL
      EXTERNAL TRANSL,SOLVER,INITZR ,VALNEW,FU,FUSPEC
      COMMON/BLOK1/XY(2,2000),NR(9,2000)/BLOK2/VAR(2,4,2000)
     1      /BLOK3/VART(4,2000)
C THE VECTOR OF UNKNOWN PARAMETERS A IS OFTEN REDEFINED AS B
C THE VECTORS X AND Y ARE XY(1,J),XY(2,J) FOR J CORRESPONDING TO NEIGHBORING AND
C     CENTER NODE OF NODE NUMBER(NONUM).
C MATRIX VALS CONTAIN VAR DATA FOR NEIGHBORING NODES
      DIMENSION B(12),X(9),Y(9),VALS(4,9)
C NDPPRM=NUMBER OF NODES PER PRISM CENTERED AT NONUM (.LE.9)
      CALL TRANSL(NUMOND,NOEQ,MUA,KOLD,NONUM,NDPPRM,X,Y,VALS,B)
      CALL SOLVER(NUMOND,NOEQ,MUA,TIME,DT,NDPPRM,NONUM,X,Y,VALS,B,FU,FUS
     1PEC)
      CALL INITZR(NOEQ,MUA,NONUM,B)
      CALL VALNEW(NUMOND,NOEQ,MUA,KNEW,DT,NONUM,VALS,B)
      RETURN
      END

      SUBROUTINE CALCB(NONUM,KNEW,TIME,INDEX)
      COMMON/BLOK1/XY(2,2000),NR(9,2000)/BLOK2/VAR(2,4,2000)/BLOK3/VART(
```

```
     14,2000)
      X=XY(1,NONUM)
      Y=XY(2,NONUM)
      T=TIME
      CALL EXTSOL(X,Y,T,UV,VV,P,E,SPECV,IBAD)
      IF(IBAC.NE.0)GOTO 20
      VAR(KNEW,1,NONUM)=UV
      VAR(KNEW,2,NONUM)=VV
      VAR(KNEW,3,NONUM)=P
      VAR(KNEW,4,NONUM)=E
      RETURN
   20 PRINT 21,NONUM
   21 FORMAT(1H,/,'  TROUBLE WAS INCURRED IN FINDING BOUNDARY VALUE AT N
     1ODE=',I6,/,'  PROGRAM IS STOPPED BY PROGRAMMER')
      STOP
      END

      SUBROUTINE TRANSL(NUMOND,NOEQ,MUA,KOLD,NONUM,NDPPRM,X,Y,VALS,B)
      COMMON/BLOK1/XY(2,2000),NR(9,2000)/BLOK2/VAR(2,4,2000)/BLOK3/VART(
     14,2000)
C TRANSFERS INFO OF A GIVEN PRISM INTO WORKING ARRAYS AND FORMS INITIAL GUESS
C TO A(I).
      DIMENSION X(NUMOND),Y(NUMOND),B(MUA),VALS(NOEQ,NUMOND)
C DETERMINE NUMBER OF NODES OF PRISM--NDPPRM
      NDPPRM=1
      MK=NUMOND-1
      DO 20 J=1,MK
      IF(NR(J,NONUM).EQ.0)GOTO 25
   20 NDPPRM=NDPPRM+1
C TRANSFER DATA INTO X,Y,VALS FROM XY,VAR AND SET TO ZERO UNUSED PORTION OF
C X,Y,VALS
C CENTER NODE INFO IN COORDINATE SYSTEM WITH ORIGIN AT CENTER NODE
   25 X(1)=0.0
      Y(1)=0.0
      DO 35 I=1,NOEQ
C KOLD REFERS TO INFO AT KNOWN INITIAL LEVEL
   35 VALS(I,1)=VAR(KOLD,I,NONUM)
C NEIGHBORING NODE INFO IN COORDINATE SYSTEM WITH ORIGIN AT CENTER NODE
      DO 30 K=2,NDPPRM
      JSUB=NR(K-1,NONUM)
      X(K)=XY(1,JSUB)-XY(1,NONUM)
      Y(K)=XY(2,JSUB)-XY(2,NONUM)
      DO 30 I=1,NOEQ
   30 VALS(I,K)=VAR(KOLD,I,JSUB)
C IF NUMBER OF TRIANGLES IS LESS THAN MAX NUMBER ALLOWED IN A PRISM, SET
C REMAINDER OF X,Y,VALS TO ZERO
      IF(NDPPRM.EQ.NUMOND)GOTO 40
      JIM=NDPPRM+1
      DO 45 K=JIM,NUMOND
      X(K)=0.0
      Y(K)=0.0
      DO 45 I=1,NOEQ
   45 VALS(I,K)=0.0
C ASSUMPTION MUA/NOEQ IS AN INTEGER(PARAMETERS A ARE EQUALLY SPACED AMONG
C UNKOWNS)
   40 MDIV=MUA/NOEQ
C FORM INITIAL GUESS OF PARAMETERS A(I)
C IN THE TRANSLATED COORDINATE SYSTEM. WE TAKE A(1,2,4,5,7,8,10 AND 11)=0
C AND A(I*3)=VART(I,NONUM)
      MDIVM1=MDIV-1
```

```
      DO 50 I=1,NOEQ
      B(MDIV*I)=VART(I,NONUM)
      DO 50 J=1,MDIVM1
   50 B(MDIV*I-J)=0.0
      RETURN
      END

      SUBROUTINE SOLVER(NUMOND,NOEQ,MUA,TIME,DT,NDPPRM,NONUM,X,Y,VALS,B,
     1FU,FUSPEC)
C     THIS SOLVER IS THE BASIC NEWTON-RAPHSON SOLVER
C COMPUTES THE FINAL VALUE OF THE PARAMETERS A(I) BY THE NEWTON-RAPHSON METHOD
C WITH THE TERMINATION CRITERIA GIVEN BY THE SUBROUTINE TERMIN
C FU AND FUSPEC ARE SUBROUTINES TO COMPUTE NECESSARY QUANTITIES FOR NEWTON-
C     RAPHSON ITERATION PROCEDURE
      EXTERNAL FU,FUSPEC
      LOGICAL INDICT,PROCED
      DIMENSION B(MUA),X(NUMOND),Y(NUMOND),VALS(NOEQ,NUMOND)
      DIMENSION CAJVNI(12,12),F(12),SAVE(12),A(12),WTS(12),DMCRFA(12)
C MXITPJ IS THE MAX NUMBER OF ITERATIONS ALLOWED PER JACOBIAN EVALUATION
C MXNOJE IS THE MAX NUMBER OF JACOBIAN EVALUATIONS ALLOWED.
C IF PROCED=T, PROGRAM WILL CONTINUE EVEN THOUGH CONVERGENCE CRITERIA IS NOT
C MET AFTER ALL ALLOWABLE ITERATIONS.
C IF PROCED=F, PROGRAM WILL STOP.
      DATA MXITPJ,MXNOJE/1,10/, PROCED/.FALSE./
      DO 20 I=1,MUA
      SAVE(I)=B(I)
   20 A(I)=B(I)
C THE NEWTON-RAPHSON METHOD
C***THE MAX NORM IS USED FOR THE CONVERGENCE CRITERIA SINCE A(I) ARE
C     DIMENSIONAL, DMCRFA(I) IS USED TO MAKE THEM NON-DIMENSIONAL FOR TESTING
C     CONVERGENCE
      CALL SCALFA(NUMOND,NOEQ,MUA,DT,VALS,DMCRFA)
      DO 30 L=1,MXNOJE
C COMPUTE THE VECTOR F AND JACOBIAN
      CALL FU(NUMOND,NOEQ,MUA,TIME,DT,NDPPRM,X,Y,VALS,A,F,CAJVNI,RES)
C COMPUTE THE INVERSE OF JAC-CAJVNI
C MATINV IS SUBPROGRAM AVAILABLE ON BRLESC SYSTEM
      CALL MATINV(CAJVNI,MUA,F,MUA,0,DET)
      IF (DET .EQ. 0.0) GO TO 35
C ITERATE MXITPJ TIMES BEFORE EVALUATING JACOBIAN AGAIN
      DO 30 K=1,MXITPJ
C CALCULATE NEW VALUES OF PARAMETERS
      DO 45 I=1,MUA
      DO 45 J=1,MUA
   45 A(I)=A(I)-CAJVNI(I,J)*F(J)
C CHECK FOR CONVERGENCE
C STPCRT,RELERR,INDICT,SAVE,WTS ARE DEFINED IN TERMIN
      CALL TERMIN (NUMOND,NOEQ,MUA,DT,NDPPRM,INDICT,STPCRT,RELERR,VALS,
     1A,SAVE,WTS,DMCRFA,NONUM)
C IF A(I)'S ARE WITHIN TOLERANCES, INDICT=TRUE
      IF(INDICT) GO TO 50
C ITERATE AGAIN
      DO 55 I=1,MUA
   55 SAVE(I)=A(I)
C **FUSPEC COMPUTES ONLY F OF NEWTON-RAPHSON, NOT THE JACOBIAN
      IF(K.LT.MXITPJ) CALL FUSPEC(NUMOND,NOEQ,MUA,TIME,DT,NDPPRM,X,Y,VAL
     1S,A,F,RES)
   30 CONTINUE
C NO CONVERGENCE
      NOITER=MXITPJ*MXNOJE
```

```
      PRINT 80
      PRINT 81,NONUM,TIME,DT,NOITER,STPCRT,RELERR,PROCED
      PRINT 82
      PRINT 83,(B(I),A(I),WTS(I),I=1,MUA)
      IF(.NOT.PROCED)STOP
C CONVERGENCE IS OBTAINED
   50 DO 60 I=1,MUA
   60 B(I)=A(I)
      PRINT 92,NONUM,L
   92 FORMAT(1H,15X,'NODE',I5,3X,I4,'ITERATIONS')
      RETURN
C TROUBLE WITH FINDING THE INVERSE JACOBIAN
   35 PRINT 91
      PRINT 85, NONUM,TIME,DT,L
      PRINT 86
      PRINT 87, (X(I),Y(I),I=1,NUMOND)
      PRINT 90
      PRINT 88
      DO 47 I=1,NUMOND
   47 PRINT 89,I,(VALS(J,I),J=1,NOEQ)
      STOP
   80 FORMAT(1H,//,'  NO CONVERGENCE')
   81 FORMAT(1H,' NUMBER OF NODE IS',I5,' TIME=',F12.7,'  DELTA T=',F8.5
     1,//,' NUMBER OF ITERATIONS=',I4,' CRITERION FOR CONVERGENCE IS',E1
     24.5,// '  LAST COMPUTED RELATIVE ERROR IS',E14.5,  / ,'  PROGRAM
     3WILL NOT STOP--',L3)
   82 FORMAT (1H,' INITIAL A(I) VALUES',5X,'LAST A(I) VALUES',7X,
     1'WEIGHTS')
   83 FORMAT  (1H, 3X,E14.5,8X,E14.5,4X,E14.5)
   85 FORMAT(1H,' NUMBER OF NODE IS',I5,' TIME=',E15.8,'  DEL T=',E15.8
     1,/' JACOBIAN WAS BEING EVALUATED THE',I3,'TH TIME')
   86 FORMAT (1H, 3X,'X-VALUES',10X,'Y-VALUES')
   87 FORMAT (1H, E14.5,4X,E14.5)
   88 FORMAT (1H,' NODE NUMBER',3X,'U-VELOCITY',3X,'V-VELOCITY',4X,'PRES
     1SURE',4X,'INT.-ENERGY',3X,'SWIRL VELOCITY')
   89 FORMAT (1H, 5X,I3,6X,E12.5,1X,E12.5,1X,E12.5,2X,E12.5,2X,E12.5)
   90 FORMAT (1H,' VALUES AT INITIAL SURFACE (DT=0) ARE ')
   91 FORMAT (1H, //,' STOP IS DUE TO DET=0.0 IN MATINV-SOLVER')
      END

      SUBROUTINE TERMIN(NUMOND,NOEQ,MUA,DT,NDPPRM,INDICT,STPCRT,RELERR,
     1                   VALS,A,SAVE,WTS,DMCRFA,NONUM)
C USES MAX NORM OVER I=1,...,MUA TO DETERMINE CONVERGENCE
      LOGICAL INDICT
      DIMENSION VALS(NOEQ,NUMOND),A(MUA),SAVE(MUA),WTS(MUA)
      DIMENSION  DMCRFA(12),EXT(12)
C WTS(I) IS THE WEIGHT ASSOCIATED WITH A(I)
C WTS ALLOWS MODIFICATION OF MAX NORM TO A WEIGHTED MAX NORM
C STPCRT IS UPPER BOUND OF ALLOWABLE RELATIVE ERROR
C THIS PROGRAM HAS A DATA STATEMENT. CHANGE DATA TO AGREE WITH DIMENSION
C OF ARRAY WTS(MUA).
      DATA  EXTSTC/1.E-5/,(EXT(I),I=1,12)/0.0,0.0,1.0,0.0,0.0,1.0,0.0,
     10.0,1.0,0.0,0.0,1.0/
      STPCRT=EXTSTC
      DO 20 I=1,MUA
   20 WTS(I)=EXT(I)
C COMPUTE THE MAX RELATIVE ERROR
      RELERR=(DMCRFA(1)* WTS(1)*(A(1)-SAVE(1)))**2
      DO 40 I=2,MUA
      ST=(DMCRFA(I)*WTS(I)*(A(I)-SAVE(I)))**2
```

```
C RELERR=RELATIVE ERROR
   40 RELERR=MAX1F(ST,RELERR)
      RELERR=SQRTF(RELERR)
C COMPARISON
C INDICT HAS VALUE TRUE IF CONVERGENCE, ELSE FALSE
      INDICT=RELERR.LT.STPCRT
      RETURN
      END

      SUBROUTINE SCALFA(NUMOND,NOEQ,MUA,DT,VALS,SF)
C COMPUTES DIMENSIONAL CORRECTION FACTORS FOR EACH A(I) SQUARED
C**** FOR EXAMPLE U=U0+(A(1)*X+A(2)*Y+A(3))*T UNITS FOR A(1),A(2) AND
C     A(3) ARE 1/(SEC*SEC), 1/(SEC*SEC), METRES/(SEC*SEC) SO THE
C     EVERY TERM HAS UNIT (METRES PER SECOND)
      DIMENSION VALS(NOEQ,NUMOND),SF(MUA)
C CHARACTERISTIC VELOCITY IS SOUND SPEED AT DT=0 AND CENTER NODE
      CALL SNDSPD(VALS(3,1),VALS(4,1),REFVEL)
C CHARACTERISTIC LENGTH IS DT*SNDSPD
      REFLNG=DT*REFVEL
C CHARACTERISTIC PRESSURE AND INTERNAL ENERGY ARE THOSE VALUES AT CENTER
C NODE AND DT=0
      REFPR=VALS(3,1)
      REFIE=VALS(4,1)
C COMPUTE THE DIMENSIONAL CORRECTION FACTORS FOR THE A(I)'S
      SF(1)=DT *DT
      SF(2)=SF(1)
      SF(3)= DT/REFVEL
      SF(4)=SF(1)
      SF(5)=SF(2)
      SF(6)=SF(3)
      F= DT*REFLNG
      SF(7)=F/REFPR
      SF(8)=SF(7)
      SF(9)=DT /REFPR
      SF(10)=F/REFIE
      SF(11)=SF(10)
      SF(12)=DT /REFIE
      IF(NOEQ.NE.5)GOTO 30
      SF(13)=SF(1)
      SF(14)=SF(2)
      SF(15)=SF(3)
   30 RETURN
      END

      SUBROUTINE FU(NUMOND,NOEQ,MUA,TIME,DT,NDPPRM,X,Y,VALS,B,F,FA,RES)
C COMPUTES TRIPLE INTEGRAL(F(I),I=1-MUA) OVER A PRISM OF THE SUM OF TERMS D(K)*
C D(K,I),K=1-NOEQ AND ITS FIRST PARTIAL DERIVATIVES (FA(I,J)),IF DESIRED
C D(K) IS THE RESIDUAL CORRESPONDING TO THE KTH EQUATION
      DIMENSION X(NUMOND),Y(NUMOND),B(MUA),VALS(NOEQ,NUMOND),F(MUA),
     1FA(MUA,MUA)
      DIMENSION RHO(9,10),PF(12),PFA(12,12)
C**** IN SOLVING A(NEW ITERATE)=A(OLD ITERATE)-INVERSE(JACOBIAN)*F, UPDATE
C     JACOBIAN AND F IF INDI=0, ONLY F IF INDI=1
      DO 20 I=1,MUA
      DO 20 J=1,MUA
   20 FA(I,J)=0.0
      INDI=0
C**** RES IS THE LEAST SQUARE RESIDUAL THAT IS TO BE MINAMIZED
      RES=0.0
      GOTO 25
```

```
      ENTRY FUSPEC(NUMOND,NOEQ,MUA,TIME,DT,NDPPRM,X,Y,VALS,B,F,RES)
      INDI=1
      RES=0.0
   25 DO 30 I=1,MUA
   30 F(I)=0.0
C APPROXIMATE TIME INTEGRAL BY A TWO PT. GAUSSIAN QUADRATURE
      JUDGE=0
C FIRST GAUSSIAN TIME POINT
      GAUST=0.2113248654*DT
C CALCULATE VARIABLE FROM EQUATION OF STATE AND ITS DERIVATIVES AT PRISM NODES
   35 CALL AREQST (NUMOND,NOEQ,MUA,NDPPRM,GAUST,X,Y,VALS,B,RHO)
C CALCULATE SPATIAL INTEGRALS OVER PRISM FOR A GIVEN GAUSSIAN TIME
      DO 40 K=1,NOEQ
C**** PF AND PFA CORRESPOND TO THE SPATIAL INTEGRALS ASSOCIATED WITH F AND FA AT
C     A GIVEN GAUSSIAN TIME
      CALL EVALPR(NUMOND,NOEQ,MUA,NDPPRM,INDI,GAUST,K,X,Y,VALS,B,RHO,PF,
     1PFA,PRES)
      RES=RES+PRES
      DO 40 L=1,MUA
      F(L)=F(L)+PF(L)
C IF INDI=1 DO NOT CALCULATE FA
      IF (INDI .EQ. 1) GOTO 40
      DO 40 J=1,MUA
      FA(L,J)=FA(L,J)+PFA(L,J)
   40 CONTINUE
C SECOND AND LAST GAUSSIAN POINT
      IF (JUDGE .NE. 0) GOTO 45
      JUDGE=1
      GAUST=0.7886751346*DT
      GOTO 35
C MULTIPLY SUMS BY CONSTANT WEIGHT FACTOR 0.5*DT
      RES=0.5*DT*RES
   45 DO 50 L=1,MUA
      F(L)=0.5*DT*F(L)
C IF INDI=1 DO NOT CALCULATE FA
      IF (INDI .EQ. 1) GOTO 50
      DO 55 J=1,MUA
   55 FA(L,J)=0.5*DT*FA(L,J)
   50 CONTINUE
      RETURN
      END

      SUBROUTINE EVALPR(NUMOND,NOEQ,MUA,NDPPRM,INDI,GAUST,K,X,Y,VALS,B,
     1                  RHO,PF,PFB,PRES)
C COMPUTES SPATIAL INTEGRALS OVER A PRISM FOR A GIVEN K RESIDUAL TERM AND
C     GAUSSIAN TIME
      COMMON/INARRS/XTRI(3),YTRI(3),U(3),V(3),P(3),E(3),SV(3),RHOTRI(3,1
     1        0)/OUTARRS/D(3),DA(12,3),DAA(12,12,3)
C IN COMMON/OUTARRS/,WE HAVE MUA=12. WHEN MUA IS NOT 12, MANUALLY CHANGE
C DIMENSIONS IN OUTARRS
      DIMENSION X(NUMOND),Y(NUMOND),VALS(NOEQ,NUMOND),RHO(NUMOND,10),B(M
     1        UA),PF(MUA),PFB(MUA,MUA),RR(3),RR2(3),RR3(3),RR4(3)
C SV DENOTES SWIRL VELOCITY (NOEQ=5)
      PRES=0.0
      DO 10 I=1,MUA
   10 PF(I)=0.0
C IF INDI=1,THEN JACOBIAN IS NOT TO BE CALCULATED
      IF(INDI .EQ. 1) GO TO 15
      DO 16 I=1,MUA
      DO 16 J=1,MUA
```

```
   16 PFB(I,J)=0.0
C COLLECT INFO FOR A GIVEN TRIANGLE INTO XTRI,YTRI,U,V,P,E,SV,RHOTRI
   15 XTRI(1)=X(1)
      YTRI(1)=Y(1)
      U(1)=VALS(1,1)
      V(1)=VALS(2,1)
      P(1)=VALS(3,1)
      E(1)=VALS(4,1)
      SV(1)=0.0
      IF(NOEQ.EQ.5)SV(1)=VALS(5,1)
      NUMTRI=NDPPRM-1
C LOOP OVER TRIANGLES COMPOSING THE PRISM
      DO 20 ITRI=1,NUMTRI
C COLLECT REMAINING DATA FOR A GIVEN TRIANGLE
      IV2=ITRI+1
      IV3=ITRI+2
      IF(ITRI.EQ.NUMTRI)IV3=2
      XTRI(2)=X(IV2)
      XTRI(3)=X(IV3)
      YTRI(2)=Y(IV2)
      YTRI(3)=Y(IV3)
      U(2)=VALS(1,IV2)
      U(3)=VALS(1,IV3)
      V(2)=VALS(2,IV2)
      V(3)=VALS(2,IV3)
      P(2)=VALS(3,IV2)
      P(3)=VALS(3,IV3)
      E(2)=VALS(4,IV2)
      E(3)=VALS(4,IV3)
      IF(NOEQ.EQ.5)GOTO 25
      SV(2)=0.0
      SV(3)=0.0
      GOTO 26
   25 SV(2)=VALS(5,IV2)
      SV(3)=VALS(5,IV3)
   26 DO  27 J=1,10
      RHOTRI(1,J)=RHO(1,J)
      RHOTRI(2,J)=RHO(IV2,J)
   27 RHOTRI(3,J)=RHO(IV3,J)
C COMPUTE QUANTITY PROPORTIONAL TO TRIANGLES AREA
      DET=DETERM(XTRI,YTRI)
C DSUB COMPUTES KTH RESIDUAL (D) AND ITS DERIVATIVES AT VERTICES OF GIVEN
C       TRIANGLE (DA,DAA)
      CALL DSUB(MUA,GAUST,INDI,K,DET,B)
C PROCEED TO CALCULATE THE SPATIAL INTEGRALS
      CALL AREAIN(D,D,DET,SOROS)
      PRES=PRES+SOROS
      DO 30 I=1,MUA
      DO 35 J=1,3
   35 RR(J)=DA(I,J)
      CALL AREAIN(D,RR,DET,SXY)
   30 PF(I)=PF(I)+SXY
C IF INDI=1 THEN JACOBIAN IS NOT TO BE CALCULATED INCREMENT ITRI.
      IF(INDI.EQ.1)GOTO 20
      DO 40 I=1,MUA
      DO 42 L=1,3
   42 RR4(L)=DA(I,L)
      DO 40 J=I,MUA
      DO 47 L=1,3
      RR2(L)=DA(J,L)
```

```
 47 RR3(L)=DAA(I,J,L)
    CALL AREAIN(D,RR3,DET,SXY1)
    CALL AREAIN(RR4,RR2,DET,SXY2)
 40 PFB(I,J)=PFB(I,J)+SXY1+SXY2
 20 CONTINUE
C REFLECT CALCULATED UPPER TRIANGULAR PORTION OF PFB TO OBTAIN PFB,IF NECESSARY
    IF (INDI .EQ. 1)  GO TO 101
    MMUA=MUA-1
    DO 50 I=1,MMUA
    III=I+1
    DO 50 J=III,MUA
 50 PFB(J,I)=PFB(I,J)
101 RETURN
    END

    SUBROUTINE DSUB(MUA,T,INDI,KSPEC,DET,A)
    DIMENSION A(12)
    DIMENSION WTS(4)
    COMMON/INARRS/AX(3),AY(3),AU(3),AV(3),AP(3),AE(3),ASV(3),ARHO(3,10
   1)/OUTARRS/D(3),DA(12,3),DAA(12,12,3)
C WTS ALLOW FOR A WEIGHTED LEAST SQUARES RESIDUAL FORMULATION IF DESIRED
    DATA  (WTS(I),I=1,4) /1.0,1.0,1.0,1.0/
C DSUB COMPUTES THE VALUE OF THE RESIDUALS D AND ITS DERIVATIVES (DA,DAA) WRT
C A(J) AT THE VERTICES OF THE TRIANGLE BEING CONSIDERED.
C TO MAKE THE INDIVIDUAL TERMS OF THE LEAST SQUARES RESIDUALS NON-DIMENSIONAL
C DIVIDE EACH TERM BY AN APPROPRIATE FACTOR-DMCF SEE THE END OF EACH SUBSECTION
C KSPEC=1,2,3,4
C DSUB IS CODED PRESENTLY FOR MUA=12 AND K=1,2,3, OR 4. DSUB CHECKS FOR THESE
C VALUES
    IF(MUA.NE.12)GOTO 101
C ON EACH TRIANGLE USE A LINEAR APPROXIMATION TO U,V,P,I AT TIME LEVEL CORRES-
C PONDING TO DT=0. BELOW ARE FACTORS NEEDED FOR THIS APPROX.
    FX1=AX(3)-AX(2)$ FX2=AX(3)-AX(1)$ FX3=AX(2)-AX(1)

    FY1=AY(3)-AY(2)$ FY2=AY(3)-AY(1)$ FY3=AY(2)-AY(1)
    IF (KSPEC .EQ. 1) GO TO 15
    IF (KSPEC .EQ. 2) GO TO 30
    IF (KSPEC .EQ. 3) GO TO 45
    IF (KSPEC .EQ. 4) GO TO 60
    PRINT 99,KSPEC
    STOP

C BEGIN CALCULATIONS FOR K=1 (CONTINUITY EQN.)
 15 UOX=( (AU(2)-AU(3))*AY(1)-(AU(1)-AU(3))*AY(2)+(AU(1)-AU(2))*AY(3))
   1 /DET
    EOX=( (AE(2)-AE(3))*AY(1)-(AE(1)-AE(3))*AY(2)+(AE(1)-AE(2))*AY(3))
   1 /DET
    POX=( (AP(2)-AP(3))*AY(1)-(AP(1)-AP(3))*AY(2)+(AP(1)-AP(2))*AY(3))
   1 /DET
    VOY=(-(AV(2)-AV(3))*AX(1)+(AV(1)-AV(3))*AX(2)-(AV(1)-AV(2))*AX(3))
   1 /DET
    EOY=(-(AE(2)-AE(3))*AX(1)+(AE(1)-AE(3))*AX(2)-(AE(1)-AE(2))*AX(3))
   1 /DET
    POY=(-(AP(2)-AP(3))*AX(1)+(AP(1)-AP(3))*AX(2)-(AP(1)-AP(2))*AX(3))
   1 /DET
    PX=POX+T*A(7)$ PY=POY+T*A(8)
    EX=EOX+T*A(10)$ EY=EOY+T*A(11)
    CR=UOX+VOY+T*(A(1)+A(5))
    TTT=T*T
    DO 20 I=1,3
```

```
      X=AX(I)$ Y=AY(I)
      R=ARHO(I,1)$ RP=ARHO(I,2)$ RI=ARHO(I,3)
      RPP=ARHO(I,4)$ RII=ARHO(I,5)$ RPI=ARHO(I,6)
      RPPP=ARHO(I,7)$ RIII=ARHO(I,8)$ RPPI=ARHO(I,9)$ RPII=ARHO(I,10)
      UT=A(1)*X+A(2)*Y+A(3)
      VT=A(4)*X+A(5)*Y+A(6)
      U=AU(I)+T*UT
      V=AV(I)+T*VT
      PT=A(7)*X+A(8)*Y+A(9)
      ET=A(10)*X+A(11)*Y+A(12)
      CRP=PT+U*PX+V*PY
      CRI=ET+U*EX+V*EY
C THE FIRST RESIDUAL D AT THE VERTICES OF THE TRIANGLE (I=1,2,3)IS
      D(I)=RP*CRP+RI*CRI+R*CR
C THE DERIVATIVE OF THE 1ST RESIDUAL WRT A(K) IS DA(K,I) AT THE VERTICES
C     (I=1,2,3)
      RTT=R*T
      XPTTU=X+T*U
      YPTTV=Y+T*V
      STOR=(RP*PX+RI*EX)*T
      DA(1,I)=STOR*X+RTT$ DA(2,I)=STOR*Y$ DA(3,I)=STOR
      STOR=(RP*PY+RI*EY)*T
      DA(4,I)=STOR*X$ DA(5,I)=STOR*Y+RTT$ DA(6,I)=STOR
      STOR=(RPP*CRP+RPI*CRI+RP*CR)*T
      DA(7,I)=RP*XPTTU+STOR*X$ DA(8,I)=RP*YPTTV+STOR*Y$ DA(9,I)=RP+STOR
      STOR=(RPI*CRP+RII*CRI+RI*CR)*T
      DA(10,I)=RI*XPTTU+STOR*X$DA(11,I)=RI*YPTTV+STOR*Y$DA(12,I)=RI+STOR
C 2ND DERIVATIVES OF D (DAA(L,K,I)) ARE NOT TO BE COMPUTED IF INDI=1
      IF(INDI.EQ.1) GO TO 20
C DAA(L,K,I) IS THE DERIVATIVE OF THE 1ST RESIDUAL D WRT A(L) AND A(K) AT THE
C     VERTICES (I=1,2,3)
      XTT=X*T$ XTT2=XTT*T
      YTT=Y*T$ YTT2=YTT*T
      RPTT2=RP*TTT$ RITT2=RI*TTT
      XPTUTT=XPTTU*T$ YPTVTT=YPTTV*T
      SPITXU=RPI*XPTUTT$ SPITYV=RPI*YPTVTT
      GPXT=(RPP*PX+RPI*EX)*TTT$ GIXT=(RPI*PX+RII*EX)*TTT
      GPYT=(RPP*PY+RPI*EY)*TTT$ GIYT=(RPI*PY+RII*EY)*TTT
      DO 24 L=1,6
      DO 24 M=L,6
   24 DAA(L,M,I)=0.0
      STORA=GPXT*X+RPTT2
      STORB=GIXT*X+RITT2
      DAA(1,7,I)=(STORA+RPTT2)*X$ DAA(1,8,I)=STORA*Y$ DAA(1,9,I)=STORA
      DAA(1,10,I)=(STORB+RITT2)*X$ DAA(1,11,I)=STORB*Y$DAA(1,12,I)=STORB
      STOR=GPXT*Y
      DAA(2,7,I)=STORA*Y$ DAA(2,8,I)=STOR*Y$ DAA(2,9,I)=STOR
      STOR=GIXT*Y
      DAA(2,10,I)=STORB*Y$ DAA(2,11,I)=STOR*Y$ DAA(2,12,I)=STOR
      DAA(3,7,I)=STORA$ DAA(3,8,I)=GPXT*Y$ DAA(3,9,I)=GPXT
      DAA(3,10,I)=STORB$ DAA(3,11,I)=STOR$ DAA(3,12,I)=GIXT
      STORA=GPYT*Y+RPTT2
      STORB=GIYT*Y+RITT2
      STOR=GPYT*X
      DAA(4,7,I)=STOR*X$ DAA(4,8,I)=STORA*X$ DAA(4,9,I)=STOR
      STOR=GIYT*X
      DAA(4,10,I)=STOR*X$ DAA(4,11,I)=STORB*X$ DAA(4,12,I)=STOR
      DAA(5,7,I)=STORA*X$ DAA(5,8,I)=(STORA+RPTT2)*Y$ DAA(5,9,I)=STORA
      DAA(5,10,I)=STORB*X$ DAA(5,11,I)=(STORB+RITT2)*Y$DAA(5,12,I)=STORB
```

```
      DAA(6,7,I)=GPYT*X$ DAA(6,8,I)=STORA$ DAA(6,9,I)=GPYT
      DAA(6,10,I)=STORS$ DAA(6,11,I)=STORB$ DAA(6,12,I)=GIYT
      STORC=RPPP*CRP+RPPI*CRI+RPP*CR
      STORD=RPPI*CRP+RPII*CRI+RPI*CR
      STORA=RPP*XPTUTT
      STORB=STORC*XTT2
      DAA(7,7,I)=(2.*STORA+STORB)*X
      DAA(7,8,I)=(STORA+STORB)*Y+RPP*YPTVTT*X
      DAA(7,9,I)=STORA+STORB+RPP*XTT
      STORA=SPITXU
      STORB=STORD*XTT2
      DAA(7,10,I)=(2.*STORA+STORB)*X
      DAA(7,11,I)=(STORA+STORB)*Y+SPITYV*X
      DAA(7,12,I)=STORA+STORB+RPI*XTT
      STORA=RPP*YPTVTT
      STORB=STORC*YTT2
      DAA(8,8,I)=(2.*STORA+STORB)*Y
      DAA(8,9,I)=STORA+STORB+RPP*YTT
      STORA=RPI*YPTVTT
      STORB=STORD*YTT2
      DAA(8,10,I)=(STORA+STORB)*X+SPITXU*Y
      DAA(8,11,I)=(2.*STORA+STORB)*Y
      DAA(8,12,I)=STORA+STORB+RPI*YTT
      DAA(9,9,I)=(2.0*RPP+STORC*T)*T
      STORA=(STORD*T+RPI)*T
      DAA(9,10,I)=STORA*X+SPITXU
      DAA(9,11,I)=STORA*Y+SPITYV
      DAA(9,12,I)=STORA+RPI*T
      STORE=RPII*CRP+RIII*CRI+RII*CR
      STORA=RII*XPTUTT
      STORB=STORE*XTT2
      DAA(10,10,I)=(2.0*STORA+STORB)*X
      DAA(10,11,I)=(STORA+STORB)*Y+X*RII*YPTVTT
      DAA(10,12,I)=STORA+STORB+RII*XTT
      STORA=YPTVTT*RII
      STORB=STORE*TTT
      DAA(11,11,I)=(2.*STORA+STORB*Y)*Y
      DAA(11,12,I)=STORA+STORB*Y+RII*YTT
      DAA(12,12,I)=2.0*RII*T+STORB
      DO 28 M=1,11
      KP1=M+1
      DO 28 L=KP1,12
   28 DAA(L,M,I)=DAA(M,L,I)
   20 CONTINUE
C FOR CONTINUITY EQUATION THE CHARACTERISTIC QUANTITY(DMCF) IS DENSITY AT
C      T=0 AND CENTER NODE
      DMCF=ARHO(1,1)
      DO 200 IC=1,3
      D(IC)=WTS(1)*D(IC)/DMCF
      DO 200 JC=1,MUA
      DA (JC,IC)=WTS(1)*DA(JC,IC)/DMCF
      DO 200 KC=1,MUA
  200 DAA(KC,JC,IC)=WTS(1)*DAA(KC,JC,IC)/DMCF
      RETURN

C BEGIN CALCULATIONS FOR K=2 (X-MOMENTUM EQN.)
   30 UOX=( (AU(2)-AU(3))*AY(1)-(AU(1)-AU(3))*AY(2)+(AU(1)-AU(2))*AY(3))
     1 /DET
      UOY=(-(AU(2)-AU(3))*AX(1)+(AU(1)-AU(3))*AX(2)-(AU(1)-AU(2))*AX(3))
     1 /DET
```

```
      POX=( (AP(2)-AP(3))*AY(1)-(AP(1)-AP(3))*AY(2)+(AP(1)-AP(2))*AY(3))
     1 /CET
      UY=UOY+A(2)*T$ UX=UOX+A(1)*T
      UYTTT=UY*T*T
      CR1=1.0+T*UX
      CR1TT=CR1*T
      CO 35 I=1,3
      X=AX(I)$ Y=AY(I)
      R=ARHO(I,1)$ RP=ARHO(I,2)$ RI=ARHO(I,3)
      RFP=ARHO(I,4)$ RII=ARHO(I,5)$ RPI=ARHO(I,6)
      UT=A(1)*X+A(2)*Y+A(3)
      VT=A(4)*X+A(5)*Y+A(6)
      U=AU(I)+T*UT
      V=AV(I)+T*VT
      CR=UT+U*UX+V*UY
C THE SECOND RESIDUAL D AT THE VERTICES OF THE TRIANGLE (I=1,2,3) IS
      C(I)=R*CR+POX+A(7)*T
C THE CERIVATIVE OF THE 2ND RESIDUAL WRT A(K) IS DA(K,I) AT THE VERTICES
C (I=1,2,3)
      STORA=R*CR1
      STORB=R*T
      CA(1,I)=STORA*X+STORB*U$ DA(2,I)=STORA*Y+STORB*V$ DA(3,I)=STORA
      STOR=STORB*UY
      CA(4,I)=STOR*X$ DA(5,I)=STOR*Y$ DA(6,I)=STOR
      STORA=CR*T
      STOR=STORA*RP
      CA(7,I)=STOR*X+T$ DA(8,I)=STOR*Y$ DA(9,I)=STOR
      STOR=STORA*RI
      CA(10,I)=STOR*X$ CA(11,I)=STOR*Y$ DA(12,I)=STOR
C 2ND CERIVATIVES OF D (DAA(L,K,I)) ARE NOT TO BE COMPUTED IF INDI=1
      IF(INCI.EC.1) GO TO 35
C DAA(L,K,I) IS THE CERIVATIVE OF THE 2ND RESIDUAL D WRT A(L) AND A(K) AT THE
C VERTICES (I=1,2,3)
      RTT2=R*T*T
      CRTT2=CR*T*T
      DAA(1,1,I)=2.0*RTT2*X$ DAA(1,2,I)=RTT2*Y$ DAA(1,3,I)=RTT2
      CAA(1,4,I)=0.0$ DAA(1,5,I)=0.0$ DAA(1,6,I)=0.0
      STORA=(CR1*X+T*U)*T
      STOR=RP*STORA
      CAA(1,7,I)=STOR*X$ DAA(1,8,I)=STOR*Y$ DAA(1,9,I)=STOR
      STOR=RI*STORA
      CAA(1,10,I)=STOR*X$ DAA(1,11,I)=STOR*Y$ DAA(1,12,I)=STOR
      DAA(2,2,I)=0.0$ DAA(2,3,I)=C.C
      CAA(2,4,I)=RTT2*X$ DAA(2,5,I)=RTT2*Y$ DAA(2,6,I)=RTT2
      STORA=(CR1*Y+T*V)*T
      STOR=RP*STORA
      CAA(2,7,I)=STOR*X$ DAA(2,8,I)=STOR*Y$ DAA(2,9,I)=STOR
      STOR=RI*STORA
      CAA(2,10,I)=STOR*X$ DAA(2,11,I)=STOR*Y$ DAA(2,12,I)=STOR
      DAA(3,3,I)=0.0$ DAA(3,4,I)=0.0$ DAA(3,5,I)=0.0$ DAA(3,6,I)=0.0
      STOR=CR1TT*RP
      DAA(3,7,I)=STOR*X$ DAA(3,8,I)=STOR*Y$ DAA(3,9,I)=STOR
      STOR=CR1TT*RI
      DAA(3,10,I)=STOR*X$ DAA(3,11,I)=STOR*Y$ DAA(3,12,I)=STOR
      CAA(4,4,I)=0.0$ DAA(4,5,I)=0.0$ DAA(4,6,I)=0.0
      STORA=UYTTT*X
      STOR=RP*STORA
      CAA(4,7,I)=STOR*X$ DAA(4,8,I)=STOR*Y$ DAA(4,9,I)=STOR
      STOR=RI*STORA
      DAA(4,10,I)=STOR*X$ DAA(4,11,I)=STOR*Y$ DAA(4,12,I)=STOR
```

```
      CAA(5,5,I)=0.0$ DAA(5,6,I)=0.0
      STORA=UYTTT*Y
      STOR=RP*STORA
      DAA(5,7,I)=STOR*X$ DAA(5,8,I)=STOR*Y$ DAA(5,9,I)=STOR
      STOR=RI*STORA
      CAA(5,10,I)=STOR*X$ DAA(5,11,I)=STOR*Y$ DAA(5,12,I)=STOR
      CAA(6,6,I)=0.0
      STOR=RP*UYTTT
      CAA(6,7,I)=STOR*X$ DAA(6,8,I)=STOR*Y$ DAA(6,9,I)=STOR
      STOR=RI*UYTTT
      DAA(6,10,I)=STOR*X$ DAA(6,11,I)=STOR*Y$ DAA(6,12,I)=STOR
      STORA=CRTT2*X
      STOR=RPP*STORA
      CAA(7,7,I)=STOR*X$ DAA(7,8,I)=STOR*Y$ DAA(7,9,I)=STOR
      STOR=RPI*STORA
      CAA(7,10,I)=STOR*X$ DAA(7,11,I)=STOR*Y$ DAA(7,12,I)=STOR
      STORA=CRTT2*Y
      STOR=STORA*RPP
      CAA(8,8,I)=STOR*Y$ DAA(8,9,I)=STOR
      STOR=STORA*RPI
      CAA(8,10,I)=STOR*X$ DAA(8,11,I)=STOR*Y$ DAA(8,12,I)=STOR
      CAA(9,9,I)=RPP*CRTT2
      STOR=RPI*CRTT2
      CAA(9,10,I)=STOR*X$ DAA(9,11,I)=STOR*Y$ DAA(9,12,I)=STOR
      STORA=CRTT2*RII
      STOR=X*STORA
      CAA(10,10,I)=STOR*X$ DAA(10,11,I)=STOR*Y$ DAA(10,12,I)=STOR
      STOR=Y*STORA
      CAA(11,11,I)=STOR*Y$ DAA(11,12,I)=STOR
      CAA(12,12,I)=STORA
      CO 40 M=1,11
      KP1=M+1
      CO 40 L=KP1,12
   40 CAA(L,M,I)=DAA(M,L,I)
   35 CONTINUE
C FOR X-MOMENTUM EQUATION THE CHARACTERISTIC QUANTITY(DMCF) IS VELOCITY
C     --TAKE SOUND SPEED AT T=0 AND CENTER NODE
      CALL SNDSPD(AP(1),AE(1),DMCF)
      CO 300 IC=1,3
      C(IC)=WTS(2)*D(IC)/DMCF
      CO 300 JC=1,MUA
      CA (JC,IC)=WTS(2)*DA(JC,IC)/DMCF
      CO 300 KC=1,MUA
  300 CAA(KC,JC,IC)=WTS(2)*DAA(KC,JC,IC)/DMCF
      RETURN

C BEGIN CALCULATIONS FOR K=3 (Y-MOMENTUM EQN.)
   45 VOX=( (AV(2)-AV(3))*AY(1)-(AV(1)-AV(3))*AY(2)+(AV(1)-AV(2))*AY(3))
     1 /CET
      VOY=(-(AV(2)-AV(3))*AX(1)+(AV(1)-AV(3))*AX(2)-(AV(1)-AV(2))*AX(3))
     1 /CET
      POY=(-(AP(2)-AP(3))*AX(1)+(AP(1)-AP(3))*AX(2)-(AP(1)-AP(2))*AX(3))
     1 /CET
      VX=VOX+A(4)*T$ VY=VOY+A(5)*T
      TTT=T*T
      CR5=1.0+T*VY
      CR5TT=CR5*T
      CO 50 I=1,3
      X=AX(I)$ Y=AY(I)
      R=ARHO(I,1)$ RP=ARHO(I,2)$ RI=ARHO(I,3)
```

```
      RFP=ARHO(I,4)$ RII=ARHO(I,5)$ RPI=ARHO(I,6)
      UT=A(1)*X+A(2)*Y+A(3)
      VT=A(4)*X+A(5)*Y+A(6)
      U=AU(I)+T*UT
      V=AV(I)+T*VT
      CR=VT+U*VX+V*VY
C THE THIRD RESIDUAL D AT THE VERTICES OF THE TRIANGLE (I=1,2,3) IS
      D(I)=R*CR+POY+A(8)*T
C THE DERIVATIVE OF THE 3RD RESIDUAL WRT A(K) IS DA(K,I) AT THE VERTICES
C (I=1,2,3)
      STOR=R*VX*T
      DA(1,I)=STOR*X$ DA(2,I)=STOR*Y$ DA(3,I)=STOR
      STORA=R*CR5
      STORB=R*T
      DA(4,I)=X*STORA+STORB*U$ DA(5,I)=Y*STORA+V*STORB$ DA(6,I)=STORA
      STORA=CR*T
      STOR=RP*STORA
      DA(7,I)=STOR*X$ DA(8,I)=STOR*Y+T$ DA(9,I)=STOR
      STOR=RI*STORA
      DA(10,I)=STOR*X$ DA(11,I)=STOR*Y$ DA(12,I)=STOR
C 2ND DERIVATIVES OF D (DAA(L,K,I)) ARE NOT TO BE COMPUTED IF INDI=1
      IF(INDI.EQ.1) GO TO 50
C DAA(L,K,I) IS THE DERIVATIVE OF THE 3RD RESIDUAL D WRT A(L) AND A(K) AT THE
C VERTICES (I=1,2,3)
      RTT2=R*TTT
      STORB=TTT*X
      DAA(1,1,I)=0.0$ DAA(1,2,I)=0.0$ DAA(1,3,I)=0.0
      DAA(1,4,I)=R*STORB$ DAA(1,5,I)=0.0$ DAA(1,6,I)=0.0
      STORA=VX*STORB
      STOR=RP*STORA
      DAA(1,7,I)=STOR*X$ DAA(1,8,I)=STOR*Y$ DAA(1,9,I)=STOR
      STOR=RI*STORA
      DAA(1,10,I)=STOR*X$ DAA(1,11,I)=STOR*Y$ DAA(1,12,I)=STOR
      STORB=TTT*Y
      DAA(2,2,I)=0.0$ DAA(2,3,I)=0.0
      DAA(2,4,I)=R*STORB$ DAA(2,5,I)=0.0$ DAA(2,6,I)=0.0
      STORA=VX*STORB
      STOR=RP*STORA
      DAA(2,7,I)=STOR*X$ DAA(2,8,I)=STOR*Y$ DAA(2,9,I)=STOR
      STOR=RI*STORA
      DAA(2,10,I)=STOR*X$ DAA(2,11,I)=STOR*Y$ DAA(2,12,I)=STOR
      DAA(3,3,I)=0.0
      DAA(3,4,I)=RTT2$ DAA(3,5,I)=0.0$ DAA(3,6,I)=0.0
      STORA=VX*TTT
      STOR=RP*STORA
      DAA(3,7,I)=STOR*X$ DAA(3,8,I)=STOR*Y$ DAA(3,9,I)=STOR
      STOR=RI*STORA
      DAA(3,10,I)=STOR*X$ DAA(3,11,I)=STOR*Y$ DAA(3,12,I)=STOR
      DAA(4,4,I)=0.0$ DAA(4,5,I)=RTT2*X$ DAA(4,6,I)=0.0
      STORA=X*CR5TT+TTT*U
      STOR=RP*STORA
      DAA(4,7,I)=STOR*X$ DAA(4,8,I)=STOR*Y$ DAA(4,9,I)=STOR
      STOR=RI*STORA
      DAA(4,10,I)=STOR*X$ DAA(4,11,I)=STOR*Y$ DAA(4,12,I)=STOR
      DAA(5,5,I)=2.0*RTT2*Y$ DAA(5,6,I)=RTT2
      STORA=Y*CR5TT+TTT*V
      STOR=RP*STORA
      DAA(5,7,I)=STOR*X$ DAA(5,8,I)=STOR*Y$ DAA(5,9,I)=STOR
      STOR=RI*STORA
      DAA(5,10,I)=STOR*X$ DAA(5,11,I)=STOR*Y$ DAA(5,12,I)=STOR
```

```
      DAA(6,6,I)=0.0
      STOR=CR5TT*RP
      DAA(6,7,I)=STOR*X$ DAA(6,8,I)=STOR*Y$ DAA(6,9,I)=STOR
      STOR=CR5TT*RI
      DAA(6,10,I)=STOR*X$ DAA(6,11,I)=STOR*Y$ DAA(6,12,I)=STOR
      STORB=CR*T*T
      STORA=STORB*X
      STOR=RPP*STORA
      DAA(7,7,I)=STOR*X$ DAA(7,8,I)=STOR*Y$ DAA(7,9,I)=STOR
      DAA(7,10,I)=STOR*X$ DAA(7,11,I)=STOR*Y$ DAA(7,12,I)=STOR
      STORA=STORB*Y
      STOR=STORA*RPP
      DAA(8,8,I)=STOR*Y$ DAA(8,9,I)=STOR
      STOR=RPI*STORA
      DAA(8,10,I)=STOR*X$ DAA(8,11,I)=STOR*Y$ DAA(8,12,I)=STOR
      DAA(9,9,I)=RPP*STORB
      STOR=RPI*STORB
      DAA(9,10,I)=STOR*X$ DAA(9,11,I)=STOR*Y$ DAA(9,12,I)=STOR
      STORA=RII*STORB
      STOR=STORA*X
      DAA(10,10,I)=STOR*X$ DAA(10,11,I)=STOR*Y$ DAA(10,12,I)=STOR
      STOR=STORA*Y
      DAA(11,11,I)=STOR*Y$ DAA(11,12,I)=STOR
      DAA(12,12,I)=STORA
      DO 55 M=1,11
      KP1=M+1
      DO 55 L=KP1,12
   55 DAA(L,M,I)=DAA(M,L,I)
   50 CONTINUE
C FOR Y-MOMENTUM EQUATION THE CHARACTERISTIC QUANTITY(DMCF) IS VELOCITY
C       --TAKE SOUND SPEED AT T=0 AND CENTER NODE
      CALL SNDSPD(AP(1),AE(1),DMCF)
      DO 400 IC=1,3
      D(IC)=WTS(3)*D(IC)/DMCF
      DO 400 JC=1,MUA
      DA (JC,IC)=WTS(3)*DA(JC,IC)/DMCF
      DO 400 KC=1,MUA
  400 DAA(KC,JC,IC)=WTS(3)*DAA(KC,JC,IC)/DMCF
      RETURN

C BEGIN CALCULATIONS FOR K=4 (INTERNAL ENERGY EQN.)
   60 UOX=( (AU(2)-AU(3))*AY(1)-(AU(1)-AU(3))*AY(2)+(AU(1)-AU(2))*AY(3))
     1 /DET
      VOY=(-(AV(2)-AV(3))*AX(1)+(AV(1)-AV(3))*AX(2)-(AV(1)-AV(2))*AX(3))
     1 /DET
      EOY=(-(AE(2)-AE(3))*AX(1)+(AE(1)-AE(3))*AX(2)-(AE(1)-AE(2))*AX(3))
     1 /DET
      EOX=( (AE(2)-AE(3))*AY(1)-(AE(1)-AE(3))*AY(2)+(AE(1)-AE(2))*AY(3))
     1 /DET
      EX=EOX+A(10)*T$ EY=EOY+A(11)*T
      UXPVY=UOX+VOY+T*(A(1)+A(5))
      TTT=T*T
      DO 65 I=1,3
      X=AX(I)$ Y=AY(I)
      R=ARHO(I,1)$ RP=ARHO(I,2)$ RI=ARHO(I,3)
      RPP=ARHO(I,4)$ RII=ARHO(I,5)$ RPI=ARHO(I,6)
      U=AU(I)+T*(A(1)*X+A(2)*Y+A(3))
      V=AV(I)+T*(A(4)*X+A(5)*Y+A(6))
      P=AP(I)+T*(A(7)*X+A(8)*Y+A(9))
      CR=A(10)*X+A(11)*Y+A(12)+U*EX+V*EY
```

```
      CRTT=CR*T
      CR10=X+T*U
      CR11=Y+T*V
C THE FOURTH RESIDUAL D AT THE VERTICES OF THE TRIANGLE (I=1,2,3) IS
      D(I)=R*CR+P*UXPVY
C THE DERIVATIVE OF THE 4TH RESIDUAL WRT A(K) IS DA(K,I) AT THE VERTICES
C (I=1,2,3)
      STORB=P*T
      STORA=R*T
      STOR=STORA*EX
      DA(1,I)=STOR*X+STORB$ DA(2,I)=STOR*Y$ DA(3,I)=STOR
      STOR=STORA*EY
      DA(4,I)=STOR*X$ DA(5,I)=STOR*Y+STORB$ DA(6,I)=STOR
      STOR=RP*CRTT+UXPVY*T
      DA(7,I)=STOR*X$ DA(8,I)=STOR*Y$ DA(9,I)=STOR
      STOR=RI*CRTT
      DA(10,I)=STOR*X+R*CR10$ DA(11,I)=STOR*Y+R*CR11$ DA(12,I)=STOR+R
C 2ND DERIVATIVES OF D(DAA(L,K,I)) ARE NOT TO BE COMPUTED IF INDI=1
      IF(INDI.EQ.1) GO TO 65
C DAA(L,K,I) IS THE DERIVATIVE OF THE 4TH RESIDUAL D WRT A(L) AND A(K) AT THE
C VERTICES (I=1,2,3)
      RTT2=R*TTT$ RTT2TY=RTT2*Y$ RTT2TX=RTT2*X
      RPTT=RP*T$ RITT=RI*T
      CR11TT=CR11*T
      CRTT2=CRTT*T
      CR10RPT=CR10*T*RP$ CR11TRP=CR11TT*RP$ CR11RIT=CR11TT*RI
      STORB=TTT*EX
      STORA=STORB*X
      STOR=TTT+STORA*RP
      DAA(1,1,I)=0.0$ DAA(1,2,I)=0.0$ DAA(1,3,I)=0.0
      DAA(1,4,I)=0.0$ DAA(1,5,I)=0.0$ DAA(1,6,I)=0.0
      DAA(1,7,I)=STOR*X$ DAA(1,8,I)=STOR*Y$ DAA(1,9,I)=STOR
      STOR=STORA*RI
      DAA(1,10,I)=RTT2TX+STOR*X$ DAA(1,11,I)=STOR*Y$ DAA(1,12,I)=STOR
      DAA(2,2,I)=0.0$ DAA(2,3,I)=0.0
      DAA(2,4,I)=0.0$ DAA(2,5,I)=0.0$ DAA(2,6,I)=0.0
      STORA=Y*STORB
      STOR=RP*STORA
      DAA(2,7,I)=STOR*X$ DAA(2,8,I)=STOR*Y$ DAA(2,9,I)=STOR
      STOR=RI*STORA
      DAA(2,10,I)=STOR*X+RTT2TY$ DAA(2,11,I)=STOR*Y$ DAA(2,12,I)=STOR
      DAA(3,3,I)=0.0
      DAA(3,4,I)=0.0$ DAA(3,5,I)=0.0$ DAA(3,6,I)=0.0
      STOR=STORB*RP
      DAA(3,7,I)=STOR*X$ DAA(3,8,I)=STOR*Y$ DAA(3,9,I)=STOR
      STOR=STORB*RI
      DAA(3,10,I)=STOR*X+RTT2$ DAA(3,11,I)=STOR*Y$ DAA(3,12,I)=STOR
      DAA(4,4,I)=0.0$ DAA(4,5,I)=0.0$ DAA(4,6,I)=0.0
      STORB=TTT*EY
      STORA=STORB*X
      STOR=STORA*RP
      DAA(4,7,I)=STOR*X$ DAA(4,8,I)=STOR*Y$ DAA(4,9,I)=STOR
      STOR=STORA*RI
      DAA(4,10,I)=STOR*X$ DAA(4,11,I)=STOR*Y+RTT2TX$ DAA(4,12,I)=STOR
      DAA(5,5,I)=0.0$ DAA(5,6,I)=0.0
      STORA=STORB*Y
      STOR=STORA*RP+TTT
      DAA(5,7,I)=STOR*X$ DAA(5,8,I)=STOR*Y$ DAA(5,9,I)=STOR
      STOR=STORA*RI
      DAA(5,10,I)=STOR*X$ DAA(5,11,I)=RTT2TY+STOR*Y$ DAA(5,12,I)=STOR
```

```
      DAA(6,6,I)=0.0
      STOR=STORB*RP
      DAA(6,7,I)=STOR*X$ DAA(6,8,I)=STOR*Y$ DAA(6,9,I)=STOR
      STOR=STORB*RI
      DAA(6,10,I)=STOR*X$ DAA(6,11,I)=RTT2+STOR*Y$ DAA(6,12,I)=STOR
      STORA=CRTT2*X
      STOR=STORA*RPP
      DAA(7,7,I)=STOR*X$ DAA(7,8,I)=STOR*Y$ DAA(7,9,I)=STOR
      STOR=STORA*RPI
      DAA(7,10,I)=(STOR+CR10RPT)*X$ DAA(7,11,I)=STOR*Y+CR11TRP*X
      DAA(7,12,I)=STOR+RPTT*X
      STORA=CRTT2*Y
      STOR=STORA*RPP
      DAA(8,8,I)=STOR*Y$ DAA(8,9,I)=STOR
      STOR=STORA*RPI
      DAA(8,10,I)=STOR*X+CR10RPT*Y$ DAA(8,11,I)=(STOR+CR11TRP)*Y
      DAA(8,12,I)=STOR+RPTT*Y
      DAA(9,9,I)=CRTT2*RPP
      STOR=CRTT2*RPI
      DAA(9,10,I)=STOR*X+CR10RPT$ DAA(9,11,I)=STOR*Y+CR11TRP
      DAA(9,12,I)=STOR+RPTT
      STORA=RII*CRTT2
      STOR=STORA*X+RITT*CR10
      DAA(10,10,I)=(STOR+RITT*CR10)*X$ DAA(10,11,I)=STOR*Y+CR11RIT*X
      DAA(10,12,I)=STOR+RITT*X
      STOR=STORA*Y
      DAA(11,11,I)=STOR*Y+2.0*CR11RIT*Y$DAA(11,12,I)=STOR+CR11RIT+RITT*Y
      DAA(12,12,I)=STORA+2.0*RITT
      DO 70 M=1,11
      KP1=M+1
      DO 70 L=KP1,12
   70 DAA(L,M,I)=DAA(M,L,I)
   65 CONTINUE
C FOR ENERGY EQUATION THE CHARACTERISTIC QUANTITY(DMCF) IS INTERNAL ENERGY
C     AT T=0 AND CENTER NODE
      DMCF=AE(1)
      DO 500 IC=1,3
      D(IC)=WTS(4)*D(IC)/DMCF
      DO 500 JC=1,MUA
      DA (JC,IC)=WTS(4)*DA(JC,IC)/DMCF
      DO 500 KC=1,MUA
  500 DAA(KC,JC,IC)=WTS(4)*DAA(KC,JC,IC)/DMCF
      RETURN
  101 PRINT 98,MUA
      STOP
   98 FORMAT(1H,' MUA=',I5,'BUT SUBROUTINE DSUB IS WRITTEN FOR MUA=12,ST
     1OP')
   99 FORMAT ( 1H,' K=',I5,'BUT SUBROUTINE DSUB IS WRITTEN FOR K=1,2,3 O
     1R 4. STOP')
      END

      SUBROUTINE AREQST(NUMOND,NOEQ,MUA,NDPPRM,GAUST,X,Y,VALS,B,RHO)
C COMPUTES DENSITY AND ITS DERIVATIVES AT ALL NODES OF A PRISM
C AT A GIVEN GAUSSIAN TIME
      DIMENSION X(NUMOND),Y(NUMOND),VALS(NOEQ,NUMOND),B(MUA),RHO(NUMOND,
     110),R(10)
      DO 50 K=1,NDPPRM
      P=VALS(3,K)+GAUST*(B(7)*X(K)+B(8)*Y(K)+B(9))
      E=VALS(4,K)+GAUST*(B(10)*X(K)+B(11)*Y(K)+B(12))
      CALL EQNST(P,E,R)
```

```
      DO 50 I=1,10
   50 RHO(K,I)=R(I)
C IF NUMBER OF TRIANGLES IS LESS THAN MAX NUMBER ALLOWED IN A PRISM, SET
C REMAINDER OF RHO TO ZERO
      IF(NDPPRM.EQ.NUMOND) GO TO 101
      J=NDPPRM+1
      DO 60 K=J,NUMOND
      DO 60 I=1,10
   60 RHO(K,I)=0.0
  101 CONTINUE
      RETURN
      END

      SUBROUTINE EQNST(P,E,R)
C COMPUTES DENSITY FOR A NOBEL ABEL GAS AND ITS FIRST 9 PARTIAL DERIVATIVES
C AT A GIVEN GAUSSIAN TIME
C HAS TWO PARAMETERS GAMMA AND ETA (UNITS FOR ETA ARE M**3/KG)
      DIMENSION R(10)
      DATA GAMMA,ETA/1.40,0.0/
      G=GAMMA-1.0
      D=G*E+ETA*P
      IF(D.EQ.0.0)GOTO 101
      DD=D*D
      DDD=D*D*D
      DDDD=DD*DD
C R(1)=DENSITY
      R(1)=P/D
C R(2)=FIRST DERIVATIVE OF R(1) WRT PRESSURE
      R(2)=G*E/DD
C R(3)=FIRST DERIVATIVE OF R(1) WRT INTERNAL ENERGY
      R(3)=-G*P/DD
C R(4)=DERIVATIVE OF R(2) WRT PRESSURE
      R(4)=-2.0*ETA*G*E/DDD
C R(5)=DERIVATIVE OF R(3) WRT INTERNAL ENERGY
      R(5)=2.0*G*G*P/DDD
C R(6)=DERIVATIVE OF R(2) WRT INTERNAL ENERGY OR OF R(3) WRT PRESSURE
      R(6)=-G*(G*E-ETA*P)/DDD
C R(7) IS THE 3RD PARTIAL OF DENSITY WRT PRESSURE
      R(7)=6.0*ETA*ETA*G*E/DDDD
C R(8) IS THE 3RD PARTIAL OF DENSITY WRT INT. ENERGY
      R(8)=-6.0*G*G*G*P/DDDD
C R(9) IS THE 3RD PARTIAL OF DENSITY WRT INT. ENERGY AND PRESSURE SQUARED
      R(9)=2.0*ETA*G*(2.0*G*E-P*ETA)/DDDD
C R(10) IS THE 3RD PARTIAL OF DENSITY WRT PRESSURE AND INT. ENERGY SQUARED
      R(10)=2.0*G*G*(G*E-2.0*ETA*P)/DDDD
      RETURN
  101 PRINT 102
  102 FORMAT(1H,//,2X,'EQUATION OF STATE,DENSITY(PRESSURE,INTERNAL ENERG
     1Y),IS WRONG SINCE DENOMINATOR IS ZERO.   STOP  SUBR. EQNST')
      STOP
      END

      SUBROUTINE SNDSPD(P,E,SS)
C COMPUTES SOUND SPEED FOR A NOBEL-ABEL GAS WITH PARAMETERS GAMMA,ETA
C UNITS FOR ETA ARE M**3/KG
      DATA GAMMA,ETA/1.40,0.0/
      G=GAMMA-1.0
      S=GAMMA*(E*G+ETA*P)**2/(E*G)
      SS=SQRTF(S)
      RETURN
```

```
      ENC

      SUBROUTINE AREAIN(A,B,DET,VALUE)
C COMPUTES INTEGRAL OVER A TRIANGLE OF A PRODUCT OF TWO FIRST DEGREE POLYNOMIALS
      CIMENSION A(3),B(3)
      C=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)
      D=A(1)*B(2)+A(2)*B(1)
      E=A(1)*B(3)+A(3)*B(1)
      F=A(2)*B(3)+A(3)*B(2)
      VALUE=ABSF(DET)*(C/12.0+(D+E+F)/24.0)
      RETURN
      ENC

      FUNCTION DETERM(X,Y)
C COMPUTES (1,1,1)DOT(Y CROSS X)
      CIMENSION X(3),Y(3)
      DETERM=X(1)*(Y(3)-Y(2))+Y(1)*(X(2)-X(3))+(X(3)*Y(2)-X(2)*Y(3))
      RETURN
      ENC

      SUBROUTINE INITZR(NOEQ,MUA,NONUM,A)
C REPLACES INITIAL DATA USED AT TIME=OLD TIME WITH INITIAL DATA TO BE USED AT
C TIME=OLD TIME+DT
      COMMON/BLOK3/VART(4,2000)
      CIMENSION A(MUA)
C ASSUMPTION  PARAMETERS A(I) ARE EQUALLY SPACED AMONG UNKNOWNS
      MDIV=MUA/NOEQ
      DO 20 K=1,NOEQ
C SAVE ONLY LAST PARAMETER IN EXPRESSION FOR EACH FLOW VARIABLE--INITIAL VALUE
C OF THE FLOW VARIABLE AT NEXT TIME STEP
   20 VART(K,NONUM)=A(MDIV*K)
      RETURN
      ENC

      SUBROUTINE VALNEW(NUMOND,NOEQ,MUA,KNEW,DT,NONUM,VALS,A)
C SUBSTITUTE NEW VALUES OF THE FLOW VARIABLES AT TIME=OLD TIME+DT INTO MAIN INFO
C MATRIX VAR AT LEVEL KNEW
      COMMON/BLOK2/VAR(2,4,2000)
      CIMENSION VALS(NOEQ,NUMOND),A(MUA)
C ASSUMPTION  PARAMETERS A(I) ARE EQUALLY SPACED AMONG UNKNOWNS
      MDIV=MUA/NOEQ
      DO 20 K=1,NOEQ
   20 VAR(KNEW,K,NONUM)=VALS(K,1)+DT*A(MDIV*K)
      RETURN
      ENC

      SUBROUTINE OUTNO6 (NUMBER,TIME,KNEW)
C OUTNO6 IS AN OUTPUT ROUTINE WHICH PRINTS THE KNEW LEVEL OF VAR AND THE
C      DENSITY
      COMMON /BLOK2/VAR(2,4,2000)
      PRINT 400
  400 FORMAT (1H,//,20X,'MATRIX VAR, AND DENSITY')
      PRINT 401
  401 FORMAT (1H,/,5X,'NODE',9X,'X VEL',14X,'Y VEL',14X,'PRESSURE',14X,
     1  'INT ENG',11X,'DENSITY')
      DO 15 J=1,NUMBER
      DENS=2.5*VAR(KNEW,3,J)/VAR(KNEW,4,J)
   15 PRINT 402,J,VAR(KNEW,1,J),VAR(KNEW,2,J),        VAR(KNEW,3,J),
     1  VAR(KNEW,4,J),DENS
  402 FORMAT(1H,3X,I5,5(4X,1PE16.8))
```

```
      RETURN
      END

      SUBROUTINE GRAPHIT (T,KNEW,NUMBER,INDEX,ORGTIM,BEGPOS)
C  SUBROUTINE GRAPHIT TAKES THE VALUES FROM VAR(KNEW,I,J)  NORMALIZES THEM
C  BY THEIR CORRESPONDING SHOCK VALUES, PRINTS THEIR VALUES, AND CALLS
C  SUBROUTINE PLOTTO TO PLOT THEIR GRAPHS
C  THE CURRENT GRAPHIT IS WRITTEN FOR NODPOS=1, NOTD=4
      DIMENSION R(200), DATP(4,200)
      COMMON/BLOK1/XY(2,2000),NR(9,2000)/BLOK2/VAR(2,4,2000)
C  COMPUTE SHOCK VALUES
C     SUBSCRIPT 1 DENOTES QUIESCENT STATE
C     FORMULAS ARE FOR GAMMA=1.4
      SHKSTH= 5.0
      OTIME=ORGTIM
      OPOS=BEGPOS
      V1=0.0
      P1=1.01325E5
      R1=1.225570786
      T1=288.15
      A1=SQRT(1.4*P1/R1)
      SHVV=A1*SQRT((1.0+6.0*SHKSTH)/7.0)
      SHKPOS=OPOS+SHVV*(T-OTIME)
      SHKV=5.0*A1*(SHKSTH-1.0)/SQRT(42.0*SHKSTH+7.0)
      SHKR=R1*(1.0+6.0*SHKSTH)/(SHKSTH+6.0)
      SHKP=P1*SHKSTH
      SHKE=2.5*SHKP/SHKR
      NODPOS=1$ I=0
      NOTD=4
      NICR=NOTD/2
C  RIGHTISH POSITION
   51 IF(NICR.GT.NUMBER)GOTO 34
      I=I+1
      R(I)=XY(2,NICR)
      DATP (1,I)=VAR(KNEW,3,NICR)/SHKP
      DATP (2,I)=VAR(KNEW,2,NICR)/SHKV
      DATP (4,I)=VAR(KNEW,4,NICR)/SHKE
      DATP (3,I)=    DATP(1,I)/DATP(4,I)
C  CENTER POSITION
      NICC=NICR+NOTD+2
      IF(NICC.GT.NUMBER)GOTO 34
      I=I+1
      R(I)=XY(2,NICC)
      DATP (1,I)=VAR(KNEW,3,NICC)/SHKP
      DATP (2,I)=VAR(KNEW,2,NICC)/SHKV
      DATP (4,I)=VAR(KNEW,4,NICC)/SHKE
      DATP (3,I)=    DATP(1,I)/DATP(4,I)
      NICR=NICC+NOTD+1
      GOTO 51
   34 PRINT 198,T
  198 FORMAT(1H,///,10X,'TIME=',1PE16.8,10X,'PRESSURE,VELOCITY,DENSITY,
     1ENERGY ARE NORMALIZED WRT THEIR SHOCK VALUES')
      PRINT 197
  197 FORMAT(1H,//,4X,'  Y      POSITION',7X,'PRESSURE',
     19X,'  Y     VELOCITY',8X,'DENSITY',9X,'INT. ENG.')
      PRINT 199,(R(K),(DATP (J,K),J=1,4),K=1,I)
  199 FORMAT(1H,(5(3X,1PE16.8),/))
      CALLPLOTTO(T,I,DATP,R,SHKPOS,SHKP,SHKV,SHKR,INDEX)
      RETURN
      END
```

```
      SUBROUTINE PLOTTO (T,NODP,DATAP,X,SHKPOS,PV,VV,RV,INDEX)
C PLOTTO USES THE COMPLOT ROUTINES TO PLOT INFO IN DATAP
C SEE TECHNICAL REPORT ARDC TR6 FOR DEFINITIONS OF VARIABLES AND SUBROUT
C     INES
      DIMENSION SCF(4),TTLB(10)
      DIMENSION TTTLB(10)
      DIMENSION TLBB(10)
      DIMENSION SX(20),SY(20)
      DIMENSION BUFF(3000),DATAP(4,200),X(200),TLB(10),SLA(4),ARR(200)
      DATA (SLA(I),I=1,4),/' PRESSURE>',' Y  VEL>',
     1 ' DENSITY>',' I ENERGY>' /
      SCF(1)=PV
      SCF(2)=VV
      SCF(3)=RV
      SCF(4)=2.5*SCF(1)/SCF(3)
      XMIN=2.8
      XMAX=3.3
      YMIN=-0.6
      YMAX=1.6
      XD=12.5
      YD=11.0
      XS=0.04
      YS=0.2
      XB=10.0
      YB=-11.0
      XOR=2.8
      YOR=-0.6
      DX=0.01
      DY=0.1
      HT=0.2
      XFAC=1.0
      YFAC=1.0
      XPAGE=27.0
      PRINT 199, XMIN,XMAX,YMIN,YMAX,XD,YD,XS,YS,HT,XOR,YOR,XPAGE
  199 FORMAT (1H,3X,'XMIN=',E15.8,3X,'XMAX=',E15.8,3X,'YMIN=',E15.8,3X,
     1 'YMAX=',E15.8,/,3X,'XD=',E15.8,3X,'YD=',E15.8,3X,'XS=',E15.8,3X,
     2  'YS=',E15.8,/ ,3X,'HT=',E15.8,3X,'XOR=',E15.8,3X,'YOR=',E15.8,
     3  3X,'XPAGE=',E15.8)
      SX(1)=SHKPOS
      SX(2)=SHKPOS
      SX(3)=SHKPOS
      SX(4)=SHKPOS
      SY(1)=1.4
      SY(2)=1.0
      SY(3)=0.2
      SY(4)=-0.4
      M=2
      CALL PLTCCB(XPAGE,M, BUFF(1),BUFF(3000))
      DO 224 JKL=1,4
      DO 50 JMN=1,NODP
   50 ARR(JMN)=DATAP(JKL,JMN)
      YB=YB+18.0
      CALL PLTCCS (XB,YB,XOR,YOR,XS,YS)
      MM=4
      CALL PLTCCA (DX,DY,XMIN,XMAX,YMIN,YMAX,MM)
      SINA=1.0
      COSA=0.0
      XY=2.72
      YX=0.2
```

```
      CALL PLTCCT(HT,SLA(JKL),SINA,COSA,XY,YX)
C     INDICATE TIME VALUE
      ENCODE(50,99,TLB) T
   99 FORMAT (' TIME= ',E15.8,1H>)
      SINA=0.0
      COSA=1.0
      XY=2.76
      YX=-0.95
      CALL PLTCCT (HT,TLB(1),SINA,COSA,XY,YX)
C     INDICATE THE NORMALIZING FACTOR
      SPSQ=SCF(JKL)
      ENCODE(50,111,TTLB) SPSQ
  111 FORMAT (' NORMALIZING FACTOR =',E15.8,1H> )
      YX=-1.15
      CALL PLTCCT(HT,TTLB(1),SINA,COSA,XY,YX)
C     INDICATE THE TIME LEVEL
      ENCODE (50,105,TTTLB) INDEX
  105 FORMAT( ' TIME LEVEL=',I4,1H> )
      YX=-0.85
      CALL PLTCCT(HT,TTTLB(1),SINA,COSA,XY,YX)
      ENCODE (50,112,TLBB)
  112 FORMAT ( ' ABSCISSA IS Y-COORDINATE IN METRES ',1H>)
      YX=-1.05
      CALL PLTCCT(HT,TLBB(1),SINA,COSA,XY,YX)
      CALL LABELA(DX,DY,XMIN,XMAX,YMIN,YMAX,XFAC,YFAC)
      MMM=1 $NS=0  $ IC=0
      CALL PLTCCD(MMM,NS,X(1),ARR(1)       ,NODP,IC, XMIN,XMAX,YMIN,
     1 YMAX)
      NS=10$ M=2 $ N=4 $ IC=0
      CALL PLTCCD (M,NS,SX(1),SY(1),N, IC, XMIN,XMAX,YMIN,YMAX)
  224 CONTINUE
      CALL PLTCCP
      RETURN
      END

*     COMPILE DISC,LABELA,ALL
```

LIST OF SYMBOLS

$\vec{a}$	vector of unknown parameters
a_i	i^{th} component of vector $\vec{a}$
$e(x,y,t)$	internal energy per unit mass [J/kg]
$p(x,y,t)$	pressure [Pa]
r	polar radial coordinate [m]
r_a, r_b	radii of annular region for blast wave calculation [m]
t	time [s]
$\bar{t}$	given value of time [s]
Δt	time increment [s]
t_m	Gaussian quadrature points used in time integration [s]
$u(x,y,t)$	velocity component in x direction [m/s]
$v(x,y,t)$	velocity component in y direction [m/s]
$v_r(x,y,t)$	velocity component in the radial direction [m/s]
x	Cartesian spatial coordinate [m]
$\bar{x}$	given value of the x coordinate [m]
y	Cartesian spatial coordinate [m]
$\bar{y}$	given value of the y coordinate [m]
$D_k(x,y,t;\vec{a})$	nondimensional residual error from the k^{th} differential equation at a point (x,y,t)
$E(\vec{a})$	total residual least squares error over a finite element
$F_i(\vec{a})$	the first partial derivative of $E(\vec{a})$ with respect to a_i

NOEQ	number of differential equations to be solved simultaneously
NUMTRI	number of prisms composing a finite element
V	volume of a finite element [m^3]
$\rho(x,y,t)$	density [kg/m^3]
$\omega(x,y,t)$	generic flow variable

SUBSCRIPTS

o	corresponds to known value at a given time
s	corresponds to value at the shock front
1	corresponds to value in the pre-shock state
2	corresponds to value in the post-shock state

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Documentation Center ATTN: DDC-TCA Cameron Station Alexandria, VA 22314
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDMA-ST 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Aviation Research and Development Command ATTN: DRSAV-E 12th and Spruce Streets St. Louis, MO 63166
1	Director US Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Electronics Command ATTN: DRSEL-RD Fort Monmouth, NJ 07703
1	Commander US Army Missile Research and Development Command ATTN: DRDMI-R Redstone Arsenal, AL 35809
1	Commander US Army Tank Automotive Development Command ATTN: DRDTA-RWL Warren, MI 48090
2	Commander US Army Mobility Equipment Research & Development Cmd ATTN: Tech Docu Cen, Bldg. 315 DRSME-RZT Fort Belvoir, VA 22060
1	Commander US Army Armament Materiel Readiness Command Rock Island, IL 61202
1	Commander US Army Armament Research and Development Command ATTN: DRDAR-SCF, S. Goodman Dover, NJ 07801
1	Commander US Army Harry Diamond Labs ATTN: DRXDO-TI 2800 Powder Mill Road Adelphi, MD 20783
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL White Sands Missile Range NM 88002
1	Commander US Army Research Office ATTN: CRD-AA-IP P. O. Box 12211 Research Triangle Park NC 27709
1	RADC (ETSR-Sty-30) ATTN: Mr. John G. Grath Hanscom AFB, MA 01731

DISTRIBUTION LIST

No. of Copies	Organization	No. of Copies	Organization
1	Director Lawrence Livermore Laboratory ATTN: Tech Info Div P. O. Box 808 Livermore, CA 94550	1	Texas Insitute for Computational Mechanics University of Texas ATTN: Prof. J. T. Oden Austin, TX 78712
1	Director Los Alamos Scientific Lab ATTN: Repts Lib P. O. Box 1663 Los Alamos, NM 87544	1	University of Alabama in Huntsville Dept of Mechanical Engineering ATTN: Prof. J. J. Chung Huntsville, AL 35807
1	U.S. Steel Research Laboratory MS 99 ATTN: Dr. Harry Morrison Monroeville, PA 15146	1	University of Colorado Department of Civil and Environmental Engineering ATTN: Prof. P. P. Lynn Boulder, CO 80302
1	Friends University Department of Mathematics ATTN: Prof. John Cheng Wichita, KS 67213	1	University of Tennessee Knoxville Department of Engineering Science and Mechanics ATTN: Prof. A. J. Baker Knoxville, TN 37916
1	Oregon State College Department of Mathematics ATTN: Prof. D. Jesperson Corvallis, OR 97331		**Aberdeen Proving Ground** Marine Corps Ln Ofc Dir, USAMSAA