

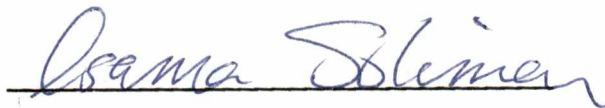
To the Graduate Council:

I am submitting herewith a thesis written by George Elliott Moore entitled "An Analysis of Factors Affecting the Convergence and Efficiency of an Implicit Finite Element Algorithm for Solution of the Quasi-One Dimensional Euler Equations." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Engineering Science.

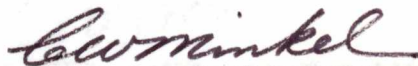


A. J. Baker, Major Professor

We have read this thesis
and recommend its acceptance:



Accepted for the Council:



The Graduate School



AN ANALYSIS OF FACTORS AFFECTING THE CONVERGENCE
AND EFFICIENCY OF AN IMPLICIT FINITE
ELEMENT ALGORITHM FOR SOLUTION OF
THE QUASI-ONE DIMENSIONAL
EULER EQUATIONS

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

George Elliott Moore

June 1984

ABSTRACT

The primary purpose of this study was to assess the effects of various terms in the formulation of a Newton matrix iteration Jacobian on the convergence rate of an implicit finite element algorithm for solution of the quasi-one dimensional Euler equations. An algorithm and computer code developed by A.J. Baker and M.O. Soliman was used as a basis for the study. In the original test of the algorithm only monotonic convergence was achieved.

A detailed check of the algorithm derivation exposed several errors in the original formulation and the omission of several terms in the original computer code was discovered. A Riemann shock tube problem was used as the primary test case with a DeLaval nozzle used to test area dependent terms. Checking the revisions individually gave some changes in the convergence histories of the dependent variables, but no significant change in the overall convergence rate. However, when the revisions were taken collectively, quadratic convergence was achieved with a concurrent 24% increase in efficiency.

The second portion of this study was to determine the applicability of a timestep optimization procedure to the fully revised algorithm. The optimization procedure is based on an approximately linear relationship between the size of the timestep and the maximum "Delta" values at that timestep.

The optimization procedure was tested on a DeLaval nozzle operating at both subsonic and transonic conditions. The

optimization procedure gave promising results when applied to smooth solutions with significant increase in computational efficiency and only minimal effects on solution accuracy. When applied to nonsmooth (shocked) solutions the optimization procedure produced poor results with decreases in efficiency or loss of convergence occurring in all test cases. These results indicate that the methods used in the optimization procedure do not model the given situation with sufficient accuracy to properly control the convergence.

TABLE OF CONTENTS

CHAPTER		PAGE
I.	INTRODUCTION	1
II.	THEORETICAL STATEMENT	4
III.	MATRIX FORM OF SOLUTION ALGORITHM	10
IV.	OPTIMIZATION SCHEME	19
V.	COMPUTER CODE	21
VI.	RIEMANN SHOCK TUBE	25
	Revisions to Euler Equation Terms	28
	Revisions to Dissipation Mechanism	33
	Efficiency and Accuracy--Fully Revised Formulation	42
VII.	DeLAVAL NOZZLE	50
	Revision to Algorithm	51
	Performance of Time Step Optimizer	52
VIII.	SUMMARY AND CONCLUSIONS	69
	LIST OF REFERENCES	73
	APPENDICES	76
	A. NOMENCLATURE	77
	B. DERIVATION OF MATRIX FORM OF F	81
	C. DERIVATION OF MATRIX FORM OF JACOBIAN	102
	D. ORIGINAL ALGORITHM	125
	E. LISTING OF COMPUTER CODE	129
	VITA	179

LIST OF TABLES

TABLE	PAGE
1. Accuracy and Efficiency, Riemann Shock Tube, Euler Equation Revisions, $M=100$, $k=1$, $\epsilon=0.001$	29
2. Newton Iteration Convergence in $\{\delta QI\}$, Riemann Shock Tube, Euler Equation Revisions, $M=100$, $k=1$, $\epsilon=0.0001$, $n\Delta t=0.0739\text{sec}$	31
3. Newton Iteration Convergence in $\{\delta QI\}$, Riemann Shock Tube, Euler Equation Revisions, $M=100$, $k=1$, $\epsilon=0.0001$, $n\Delta t=0.0755\text{sec}$	32
4. Newton Iteration Convergence in $\{\delta QI\}$, Riemann Shock Tube, Penalty Constraint Revisions, $M=100$, $k=1$, $\epsilon=0.001$, $n\Delta t=0.0739\text{sec}$	34
5. Newton Iteration Convergence in $\{\delta QI\}$, Riemann Shock Tube, Penalty Constraint Revisions, $M=100$, $k=1$, $\epsilon=0.001$, $n\Delta t=0.0755\text{sec}$	35
6. Accuracy and Efficiency, Riemann Shock Tube, Penalty Constraint Revisions	36
7. Newton Iteration Convergence in $\{\delta QI\}$, Riemann Shock Tube, Initial Time Steps, RMN07, $M=100$, $k=1$, $\epsilon=0.001$, $n\Delta t=0.0739\text{sec}$	39
8. Solution Norm $\left\ \left q_{\alpha}^h \right \right\ _{H^0}^2$, Riemann Shock Tube, $k=1$, $\epsilon=0.001$, Courant No. = .18	45
9. Solution Norm $\left\ \left q_{\alpha}^h \right \right\ _{L^1}^2$, Riemann Shock Tube, $k=1$, $\epsilon=0.001$, Courant No. = .18	46
10. Accuracy and Efficiency, Riemann Shock Tube, with Respect to Courant Number, $M=100$, $k=1$, $\epsilon=0.001$	47
11. Newton Iteration Convergence in $\{\delta QI\}$, Subsonic Converging Diverging Nozzle, Effects of Inclusion of Area Terms in [JGI]	53
12. Accuracy and Efficiency Summary, Converging Diverging Nozzle, Effects of Inclusion of Area Terms in [JGI]	54
13. Newton Iteration Convergence in $\{\delta Q\}$, Off Design Transonic Converging Diverging Nozzle, Effects of Inclusion of Area Terms in [JGI]	55

TABLE	PAGE
14. Accuracy and Efficiency Summary, Converging-Diverging Nozzle, Smooth Solution with Optimization	58
15. Accuracy and Efficiency Summary, Converging-Diverging Nozzle, Shocked Solutions with Optimization	63

LIST OF FIGURES

FIGURE	PAGE
1. Original Formulation of Algorithm, Riemann Shock Tube Solution	27
2. Revisions to Penalty Constraint Terms, $v_2 = v_0 \{3/4, 2, 1\}$, $v_0 = (15)^{1/2}$	37
3. Fully Revised Formulation with $V = V/V_{\max} $ in Penalty Constraint Terms, Riemann Shock Tube Solution, $v_2 = v_0 \{3/4, 2, 1\}$, $v_0 = (15)^{1/2}$	40
4. Fully Revised Formulation of Algorithm with $V = V/V_{\max} $ in Penalty Constraint Term, Riemann Shock Tube Solution, $v_2 = v_0 \{1/2, 3, 2, 3/4\}$, $v_0 = (15)^{-1/2}$	41
5. Penalty Constraint Derived From Conservative Form of Convective Derivatives, Riemann Shock Tube Solution . . .	43
6. Semi-discrete Approximation Accuracy and Convergence in $\ q^h\ _E$ and $\ q^h\ _{H^1}^2$, Fully Revised Algorithm, Riemann Shock Tube Solution	44
7. Fully Revised Algorithm, $\Delta t = 8x\Delta t$ Standard Riemann Shock Tube Solution	49
8. Subsonic Converging-Diverging Nozzle, Transient Solution, No Optimization	61
9. Subsonic Converging-Diverging Nozzle, Transient Solution, Second Level Optimization	62
10. Off Design Transonic Converging-Diverging Nozzle, Stationary Solution, Nodes 30-101	65
11. Off Design Transonic Converging-Diverging Nozzle, Transient Solution, Nodes 30-101	67

CHAPTER I

INTRODUCTION

The characterization of high speed fluid flow is an area of great interest, especially in the field of aerodynamics. Basic to these analyses are the Navier-Stokes equations. The Navier-Stokes equations are a set of highly complex nonlinear partial differential equations from which analytical solutions have been deduced only using restrictive simplifying assumptions and simple geometric boundaries. The analysis of transonic aerodynamic flows allows few simplifications in either the fluid properties or the geometric boundaries of the problem making an analytical solution difficult if not impossible. This has led to the development of numerical approximation techniques for solution of the required equation set. The use of discrete approximation methods has followed a corequisite increase in the development and usage of high speed computers. The most widely used discrete approximation methods have been the "MacCormack Algorithm" and similar finite difference algorithms [Roache, 1972]. In recent years the finite element method has been gaining wider acceptance due largely to advantages in the ease of application of irregular geometries and nonuniform meshes and imposition of appropriate boundary conditions.

A finite element algorithm for analysis of the three-dimensional Navier-Stokes equations has been presented by Baker

[1982]. This algorithm employs a tensor matrix product approximation for the Newton matrix iteration Jacobian in the ultimate non-linear algebraic equation statement. The algorithm also includes an artificial dissipation mechanism to control non-linearly magnified oscillations in the solution. The Newton matrix iteration procedure used for solution of the non-linear algebraic equation set has a theoretically predicted quadratic convergence rate [Burden, et al., 1981]. The algorithm presented by Baker [1982] achieved only linear convergence. The purpose of this analysis is to determine which factors in the development of the algorithm, particularly in the development of the Jacobian, exert dominant influence on the convergence rate.

Many of the performance characteristics of the algorithm can be observed using a simplified form of the basic equation set. For this analysis the quasi-one dimensional Euler equations have been used. The first step in this analysis was to check the formulation of the algorithm starting with the divergence form of the Euler equations and proceeding stepwise through the Galerkin weighted residual formulation and the derivation of the terms forming the F-vector and the Jacobian. The second step was revision of the original computer code to allow comparison of the original and revised algorithms. The code revisions included recasting of several variables into forms which allow direct comparison with the symbolic presentation of the algorithm. The last step in the analysis was the comparison of solutions reflecting changes in the algorithm and coding.

Three test cases are used to provide comparison data for several solution types. The primary test case is the Riemann shock tube which has steep gradients and discontinuities interspersed with plateaus in a solution which can be evaluated analytically, thus providing a good measure of solution stability and accuracy. The other test cases are an off design DeLaval nozzle, and a subsonic nozzle. These test cases provide discontinuous and smooth area dependent solutions. Several parameters were compared for each test. The parameters checked include the maximum "Delta" values for each iteration, the total number of iterations, and the CPU time for a given solution. Also, in order to determine the consistency of the solutions, a comparison was made of the solution integral norms and of the array of nodal values of solution variables.

In addition to the comparison of the algorithm formulations, a time step optimization procedure is developed wherein the step size is adjusted according to the maximum delta value from the first iteration and the number of iterations from the previous time step. With proper "fine tuning" the optimizer should provide time step sizes near the upper limit of convergence.

CHAPTER II

THEORETICAL STATEMENT

An algorithm has been established for solution of the multi-dimensional Navier-Stokes equations. The full algorithm by Baker [1982] was developed for the equation set governing flow of a viscous, compressible, heat conducting fluid. Many performance characteristics of this algorithm may be examined using a form of the algorithm which has been reduced by assumption of inviscid flow, i.e. the Euler equation set. The further assumption of quasi-one dimensional flow yields the hyperbolic conservation law system in the form [Liu, 1982].

$$L(\rho) = \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} [\rho u] + \rho u \frac{1}{A} \frac{dA}{dx} = 0 \quad (1)$$

$$L(\rho u) = \frac{\partial \rho u}{\partial t} + \frac{\partial}{\partial x} [u \rho u + p] + \rho u^2 \frac{1}{A} \frac{dA}{dx} = 0 \quad (2)$$

$$L(\rho e) = \frac{\partial \rho e}{\partial t} + \frac{\partial}{\partial x} [u \rho e + p u] + (u \rho e + p u) \frac{1}{A} \frac{dA}{dx} = 0 \quad (3)$$

where ρ is the density, u is velocity, p is pressure, e is the mass specific total energy and $A(x)$ is the flow cross sectional area. Closure of the equation set is attained by inclusion of the equation of state for a polytropic gas in the form

$$L(p) = p - (\gamma - 1)[\rho e - 1/2 \rho u \cdot u] = 0 \quad (4)$$

where γ is the specific heat ratio.

In order to provide a normalized solution the equation set was nondimensionalized. Reference conditions for the nondimensionalization are the stagnation conditions calculated from the initial inlet conditions (high pressure conditions for the Riemann shock tube). Two nondimensional groups occur in the final form of the equations. These are the Euler number $(p_0/\rho_0 a_0^2)$ and the Eckert number $(p_0/\rho_0 e_0)$. Making appropriate substitutions into the nondimensional group yields the Euler number $= \frac{1}{\gamma}$ and the Eckert number $= (\gamma - 1)$. The nondimensionalization leaves equations (1)-(4) in the form,

$$L(\rho) = \frac{\partial \rho}{\partial t} + \frac{\partial m}{\partial x} + mA' = 0 \quad (5)$$

$$L(m) = \frac{\partial m}{\partial t} + \frac{\partial}{\partial x} [um + \frac{1}{\gamma} p] + umA' = 0 \quad (6)$$

$$L(g) = \frac{\partial g}{\partial t} + \frac{\partial}{\partial x} [ug + (\gamma - 1)up] + (ug + (\gamma - 1)up)A' = 0 \quad (7)$$

$$L(p) = p - [g - \frac{1}{2} \gamma(\gamma - 1)mu] = 0 \quad (8)$$

where $m \equiv \rho u$, $g \equiv p e$, and $A' \equiv \frac{1}{A} \frac{dA}{dx}$. Details of the nondimensionalization are presented in Appendix B.

Representing the set of independent variables as $q_\alpha(x,t) \equiv \{\rho, m, g, p\}$, a compact form of equations (5)-(8) is

$$L(q_\alpha) = \frac{\partial q_\alpha}{\partial t} + \frac{\partial}{\partial x} [u q_\alpha + f_\alpha^1] + f_\alpha^2 = 0 \quad (9)$$

where f_α^1 are specified functions of the arguments q_α . The solution domain, over which equation (9) is applied, is

$$\Omega \equiv R^1 \times t = \{(x, t) : x \in R^1 \text{ and } t \in [t_0, t]\} \quad (10)$$

where R^1 is the one-dimensional Euclidean space spanned by the x coordinate. The boundaries of the solution domain are defined as $\partial\Omega = \partial R^1 \times t$ where ∂R^1 indicates the end points of the space spanned by the x coordinate. Problem specific boundary conditions applied on ∂R^1 are of the form.

$$\mathcal{L}(q_\alpha) = a_\alpha^1 q_\alpha + a_\alpha^2 \frac{\partial q_\alpha}{\partial x} + a_\alpha^3 = 0 \quad (11)$$

Definition of a_α^1 in equation (11) determines Dirichlet and/or Neumann conditions. The problem definition is completed by specification of an initial distribution $q_0 = q_\alpha(x, 0)$ [Kreyszig, 1972].

Application of the finite element method to this problem requires establishment of a semidiscrete approximation, $q_\alpha^h(x, t)$ to the unknown exact solution, $q_\alpha(x, t)$, of equation (9). The approximation may be constructed from members of any subspace of the infinite dimensional Hilbert space $H^1(\Omega)$ [Strang and Fix, 1977]. Defining the set of interior subdomains, R_e^1 , such that $\bigcup_{e=1}^M R_e^1 = R^1$, the approximate solution is

$$q_{\alpha}(x,t) \approx q_{\alpha}^h(x,t) = \sum_{e=1}^M q_{\alpha}^e(x,t) \quad (12)$$

and the elemental semi-discrete approximation is

$$q_{\alpha}^e(x,t) = \{N_k(x)\}^T \{QI(t)\}_e \quad (13)$$

where $\{QI(t)\}_e, 1 \leq I \leq 4$ is the column matrix of nodal values of q_{α}^h on R_e^1 evaluated at time t . The elements of $\{N_k(x)\}$ are polynomials on x , complete to degree k and defined on R_e^1 to form a cardinal basis. Specifically for R_e^1

$$\begin{aligned} \{N_1(x)\} &= \begin{Bmatrix} 1 - x/h_e \\ x/h_e \end{Bmatrix} \\ \{N_2(x)\} &= \begin{Bmatrix} (1 - x/h_e)(1 - 2x/h_e) \\ 4x/h_e(1 - x/h_e) \\ x/h_e(2x/h_e - 1) \end{Bmatrix} \end{aligned}$$

where h_e is the length of the subdomain (finite element) R_e^1 [Baker, 1983A].

The method of weighted residuals is used to minimize the error in an approximation of a given differential equation by forcing the residual, or error, to be zero in some average sense. This is accomplished by selecting a set of weighting functions and setting the integral of the weighted residual to zero. The Galerkin form of the method of weighted residuals, from which the finite element method is derived, uses the basis function which was used to form the approximation as the weighting function. Using this formulation, the error in equations (9) and (11) is required to be

orthogonal to the space of functions defining q_α^h [Botha and Pinder, 1983]. An additional constraint is included in the formulation to control nonlinearly magnified oscillations occurring in regions of nonsmoothness. The dissipation mechanism required to control these oscillations is generated by requiring the nonconservative form of the convective portion of equations (5)-(7), i.e., $L^c(\cdot) = \frac{\partial}{\partial t} + u \frac{\partial}{\partial x}$, to be orthogonal to the modified basis, $\beta^1 \frac{d}{dx} \{N_k\}$. The LaGrange multiplier set $\bar{\beta}_\alpha^1$ is defined as

$$\bar{\beta}_\alpha^1 \equiv h_e (v_\alpha^1 \delta t + v_\alpha^2 \text{sign } u \delta x) \hat{1} \quad (14)$$

where the v_α^i are artificial viscosity terms applied to the temporal ($i=1$) and the spatial ($i=2$) derivatives of the convective portions of the equations [Baker, 1982]. Combination of these linearly independent constraints yields the finite element solution algorithm statement in the form

$$\begin{aligned} \int_{R^1} \{N_k\} L(q_\alpha^h) dx + \beta_\alpha^1 \int_{R^1} \frac{d}{dx} \{N_k\} L^c(q_\alpha^h) dx + \\ \beta_\alpha^2 \int_{\partial R^1} \{N_k\} L(q_\alpha^h) dx \equiv \{0\} \end{aligned} \quad (15)$$

Upon defining k in equation (13), equation (15) becomes a large order system of non-linear, coupled ordinary differential equations on t of the form

$$[A] \frac{d\{QI\}}{dt} + \{B(QI)\} + \{C\} = \{0\} \quad (16)$$

which defines the temporal derivative of the fully discrete approximation. Using equation (16) in a Taylor series yields the ultimate matrix algebra problem statement.

$$\{FI\} \equiv \{QI\}_{j+1} - \{QI\}_j - \Delta t \{QI\} \Big|_{j+\theta} + \dots = \{0\} \quad (17)$$

for $1 \leq \theta \leq 4$. For $\theta \neq 0$, equation (17) is a non-linear system indicating the need for an implicit integration algorithm. Choosing $\theta = 1/2$ yields the trapezoidal rule for integration in the form

$$\{FI\} = \{QI\}_{j+1} - \{QI\}_j + \frac{\Delta t}{2} \left[\frac{d\{QI\}}{dt} \Big|_{j+1} + \frac{d\{QI\}}{dt} \Big|_j + \dots \right] = \{0\} \quad (18)$$

The Newton iteration algorithm for nonlinear systems applied to $\{FI\} = \{0\}$, equation (18), is of the form

$$[J(QI)]_{j+1}^p \{\delta QI\}_{j+1}^{p+1} = -\{FI\}_{j+1}^p \quad (19)$$

The superscript p denotes the iteration at time step t_{j+1} and $\{\delta QI\}$ is the iteration vector defining the $(p+1)^{st}$ refinement to the solution as

$$\{QI\}_{j+1}^{p+1} = \{QI\}_{j+1}^p + \{\delta QI\}_{j+1}^p \quad (20)$$

The Newton method Jacobian is defined as

$$[J(QI)] \equiv \frac{\partial \{FI\}}{\partial \{QI\}} \quad (21)$$

CHAPTER III

MATRIX FORM OF SOLUTION ALGORITHM

The matrix representation of the algorithm is presented in the hypermatrix symbology developed by Baker [1983B]. The hypermatrix construction of the spatial derivatives resulting from completion of the integrals in equation (15) is of the form $[AM \dots]$, where A is used to represent a one dimensional solution domain and M is the number of basis functions occurring in the integral. Following M are Boolean counters which identify each basis function as either differentiated (1) or not differentiated (0). As an example

$$[A3010]_e \equiv \int_0^{h_e} \{N_k\} \frac{d\{N_k\}}{dx} \{N_k\}^T dx .$$

Two additional matrices used in this representation are identified as $[A20]$ and $[A30]$. In these matrices, A and M are as previously defined, but the single trailing zero indicates that the basis functions have been modified so that the function exists only on the vertex nodes of the element and is directed along an outward pointing unit normal. A more detailed examination of the $[A20]$ and $[A30]$ matrices is given in Appendix B.

Denoting the elements of $\{QI\}$ and $\{FI\}$, $1 \leq I \leq 4$, as $\{R,M,G,P\}$ and $\{FR,FM,FG,FP\}$, the finite element algorithm statement, equation (18) is of the form,

$$\{FI\} = S_e [\{FI(h_e, [AM \dots], \{OI\}_e)\}_e] \quad (22)$$

where S_e is the matrix operator projecting elemental contributions to the global matrix statement. Evaluation of equations (5)-(8) in equation (15) yields the following expression for $\{FI\}_e$ in equation (22).

$$\begin{aligned} \{FR\}_e &= h_e \left[[A200] + v_R^1 [A210] \right] \{R\}_{j+1} + \\ &\frac{\Delta t}{2} \left[\left[[A20] - [A210] \right] \{M\} + v_R^2 \{\bar{V}\}^T [A3011] \{R\} + \right. \\ &\left. \{A\}^T ([A41000] \{V\}) \{R\} \right]_{j+1,j} = \{0\} \end{aligned} \quad (23)$$

$$\begin{aligned} \{FM\}_e &= h_e \left[[A200] + v_M^1 [A210] \right] \{M\}_{j+1} + \\ &\frac{\Delta t}{2} \left[\{V\} \left[[A30] - [A3010] \right] \{M\} + \right. \\ &\frac{1}{\gamma} \left[[A20] - [A210] \right] \{P\} + v_M^2 \{V\}^T [A3011] \{M\} + \\ &\left. \{A\}^T ([A41000] \{V\}) \{M\} \right]_{j+1,j} = \{0\} \end{aligned} \quad (24)$$

$$\begin{aligned}
\{FG\}_e &= h_e \left[[A200] + v_G^1 [A210] \right] \{G\}_{j+1} \\
&+ \frac{\Delta t}{2} \left[\{V\}^T \left[[A30] - [A3010] \right] \{ \{G\} + (\gamma - 1) \{P\} \} \right. \\
&+ \frac{v_G^2}{h} \{ \bar{V} \} [A3011] \{G\} + \{A\}^T ([A41000] \{V\}) \{ \{G\} \\
&\left. + (\gamma - 1) \{P\} \} \right]_{j+1,j} = \{0\} \quad (25)
\end{aligned}$$

$$\begin{aligned}
\{FP\}_e &= h_e [A200] \{P\} - h_e \left[[A200] \{G\} \right. \\
&\left. - \frac{1}{2} \gamma (\gamma - 1) \{V\}^T [A3000] \{M\} \right] = \{0\} \quad (26)
\end{aligned}$$

In equations (23)-(26) the subscript e has been suppressed on all matrices for clarity. The elements of $\{\bar{V}\}$ are the corresponding magnitudes of the convective velocity, m/ρ . The elements of $\{A\}$ are the nodal values of $\epsilon_n A(x)$. Other notations used in equations (23)-(26) are $\{\cdot\}_{j+1}^p \equiv \{\cdot\}_{j+1}^p - \{\cdot\}_j$ and $[\cdot]_{j+1,j} \equiv 1/2[\cdot]_{j+1}^p + 1/2[\cdot]_j$. Details of the derivation of equations (23)-(26) have been presented in Appendix B.

Comparison of equations (23)-(26) with the original formulation presented by Baker (see Appendix D) yields two basic differences.

The first of these differences arises from the nondimensionalization of equations (5)-(8). The Euler number and the Eckert number are formed during the nondimensionalization and appear as functions of the specific heat ratio, γ , on the pressure terms in equations (24) and (25) and in equation (26). The second difference is in the presentation of the non-area dependent spatial derivatives of equations (6)-(8) evaluated in the first constraint of equation (15). The Green Gauss divergence form of the integral in question contains a contour integral evaluated around the element assuming an outward directed unit normal vector [Kreyszig, 1972]. Upon evaluation of the integral (see Appendix B) the term disappears everywhere except at the vertex nodes of the element and, due to the outward directed unit normal, has opposing signs at the vertex nodes. Although these terms, involving $[AM_0]$, $M=2$ or 3 , did not appear in the original algorithm presentation, they were included in the original computer code.

The Newton matrix iteration Jacobian is constructed from equation (21) using equations (23)-(26). As in equation (22), the global Jacobian is constructed from assembly of elemental evaluations in the form

$$[J(QI)] \equiv S_e [J(h_e, [AM \dots], \{QI\})]_e \quad (27)$$

Evaluation of the derivatives of $\{FI\}_e$ with respect to $\{QJ\}_e$ yields.

$$[JRR]_e \equiv \frac{\partial \{FR\}_e}{\partial \{R\}_e} = h_e \left[[A200] + v_R^1 [A210] \right] + \frac{\Delta t}{2} \left[v_R^2 [\bar{V}] [A3011] \right.$$

$$\left. - \left| \{A10\}^T \left\{ \frac{M}{R^2} \right\} \right| \{R\}^T [A3110] \right] + \{A\}^T [A41000] \{V\}$$

$$\left. - \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{R\} \right]_{j+1}^P$$

$$[JRM]_e = \frac{\Delta t}{2} \left[[A20] - [A210] + v_R^2 \left| \{A10\}^T \left\{ \frac{1}{R} \right\} \right| \{R\}^T [A3110] \right.$$

$$\left. + \{A\}^T [A3100] \right]_{j+1}^P$$

$$[JRG]_e = [0]$$

$$[JRP]_e = [0] \quad (28)$$

$$[JMR]_e \equiv \frac{\partial \{FM\}_e}{\partial \{R\}_e} = - \frac{\Delta t}{2} \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{M\}^T \left[[A30] - [A3010] \right.$$

$$\left. + \text{SIGN}(u) v_M^2 [A3110] + [A40001] \{A\} \right]_{j+1}^P$$

$$\begin{aligned}
[JMM]_e &= h_e [A200] + v_M^1 [A210] + \frac{\Delta t}{2} \left[\{V\} \right. \\
&\quad + \{A10\}^T \left\{ \frac{1}{R} \right\} \{M\}^T [A30] - [A3010] + [A40001] \{A\} \\
&\quad \left. + v_M^2 \left[\{\bar{V}\} [A3011] + \left| \{A10\}^T \left\{ \frac{1}{R} \right\} \right| \{M\}^T [A3110] \right] \right]_{j+1}^P \\
[JMG]_e &= [0] \\
[JMP]_e &= \frac{\Delta t}{2} \frac{1}{\gamma} [A20] - [A210] \Big]_{j+1}^P \quad (29) \\
[JGR]_e &\equiv \frac{\partial \{FG\}}{\partial \{R\}} = - \frac{\Delta t}{2} \left[\{A10\}^T \left\{ \frac{M}{R^2} \right\} \{G\} + (\gamma - 1) \{P\}^T [A30] \right. \\
&\quad - [A3010] + [A40001] \{A\} \\
&\quad \left. + v_G^2 \left| \{A10\}^T \left\{ \frac{M}{R^2} \right\} \right| \{G\}^T [A3110] \right]_{j+1}^P \\
[JGM]_e &= \frac{\Delta t}{2} \left[\{A10\}^T \left\{ \frac{1}{R} \right\} \{G\} + (\gamma - 1) \{P\}^T [A30] - [A3010] \right. \\
&\quad \left. + [A40001] \{A\} + v_G^2 \left| \{A10\}^T \left\{ \frac{1}{R} \right\} \right| \{G\}^T [A3110] \right]_{j+1}^P
\end{aligned}$$

$$\begin{aligned}
[JGG]_e &= h_e [A200] + v_G^1 [A210] + \frac{\Delta t}{2} \left[\{V\}^T [A30] - [A3010] \right] \\
&+ v_G^2 \{\bar{V}\}^T [A3011] + \{V\}^T [A40001] \{A\} \Bigg]_{j+1}^P
\end{aligned}$$

$$\begin{aligned}
[JGP]_e &= \frac{\Delta t}{2} (\gamma - 1) \{V\}^T [A30] - [A3010] \\
&+ [A40001] \{A\} \Bigg]_{j+1}^P \quad (30)
\end{aligned}$$

$$[JPR]_e \equiv \frac{\partial \{FP\}}{\partial \{R\}}_e = -h_e \frac{\gamma(\gamma - 1)}{2} \left[\{A10\}^T \left\{ \frac{M}{R^2} \right\} \{M\} [A3000] \right]_{j+1}^P$$

$$[JPM]_e = h_e \frac{\gamma(\gamma - 1)}{2} \left[\{V\}^T [A3000] \right]$$

$$+ \{A10\}^T \left\{ \frac{1}{R} \right\} \{M\}^T [A3000] \Bigg]_{j+1}^P$$

$$[JPG]_e = -h_e [A200]$$

$$[JPP]_e = h_e [A200] \quad (31)$$

where the elements of the matrices $\{M/R^2\}$ and $\{1/R\}$ are the nodal value of m/ρ^2 and $1/\rho$. Using equations (23)-(26) and (28)-(31), the symbolic representation of the Newton algorithm is

$$S_e \begin{bmatrix} [JRR]_e & [JRM]_e & [0] & [0] \\ [JMR]_e & [JMM]_e & [0] & [JMP]_e \\ [JGR]_e & [JGM]_e & [JGG]_e & [JGP]_e \\ [JPR]_e & [JPM]_e & [JPG]_e & [JPP]_e \end{bmatrix} \begin{Bmatrix} \delta R \\ \delta M \\ \delta G \\ \delta P \end{Bmatrix} = - \begin{Bmatrix} FR \\ FM \\ FG \\ FP \end{Bmatrix} \quad (32)$$

Several differences can be noted between term (28)-(31) and the Jacobian formulation presented by Baker (See Appendix D). Three of these changes occur in the formulation of the derivatives and are not directly related to the changes in the formulation of $\{FI\}_e$ mentioned previously. The first of these changes is in the differentiation of velocity with respect to density and momentum, $\partial\{V\}/\partial\{R\}$ and $\partial\{V\}/\partial\{M\}$. The original form of these derivatives involved functions of the elemental averages of the given variables, i.e.

$$\frac{\partial\{V\}}{\partial\{M\}} = \frac{1}{\bar{\rho}} \quad , \quad \frac{\partial\{V\}}{\partial\{R\}} = - \frac{\bar{m}}{\bar{\rho}^2} \quad (33)$$

where the superscript bar indicates the elemental average of the given variable. A closer examination of these derivatives (See Appendix C) indicates the form should be the elemental average of the functions of the given variables, given as

$$\frac{\partial\{V\}}{\partial\{M\}} = \{A10\}^T \left\{ \frac{1}{R} \right\} \quad , \quad \frac{\partial\{V\}}{\partial\{R\}} = - \{A10\}^T \left\{ \frac{M}{R^2} \right\} \quad (34)$$

where $\left\{ \frac{1}{R} \right\}$ and $\left\{ \frac{M}{R^2} \right\}$ are the nodal values of the functions $1/\rho$ and m/ρ^2 respectively. The second change is in the derivative of the absolute value of the velocity which occurs in the diffusion terms. In the original formulation these derivatives were identical to

those taken of the actual value of the velocity. Experimentation with these derivatives indicates the best form for use in the dissipation mechanism to be the absolute value of the derivative. The third change occurs in the derivative of the diffusion term in $\{FR\}_e$ with respect to the momentum. This term in the original formulation was of the form $[A211]\{M\}$ which is the form that would arise in the conservative form of the convective derivative of the density were used. Evaluation of this term using the nonconservative convective derivative yields

$$\frac{\partial}{\partial \{M\}} \{\bar{V}\}^T [A3011] \{R\} = \left| \{A10\}^T \left\{ \frac{1}{R} \right\} \right| \{R\}^T [A3110] \quad (35)$$

Details of this derivation are given in Appendix C.

CHAPTER IV

OPTIMIZATION SCHEME

Observation of the convergence histories of transient solutions, such as a moving shock in an off design DeLaval nozzle, indicates a need for adjustment of the timestep size to allow for a more consistent convergence rate throughout the solution. A two step optimization scheme has been developed which should provide timesteps that are based on the amount of change in the solution with time at the given point in time.

The first level of the optimization provides an adjustment of the size of the timestep after the first iteration at a given timestep. In the Newton iteration algorithm, the values of $\{\delta QI\}_{j+1}^1$ at the step $j+1$ are based entirely on the converged values of $\{QI\}$ from step j . The values of $\{\delta QI\}_{j+1}^1$ are dependent solely on the value of Δt [Burden, et al., 1981]. Assuming an approximately linear relationship between Δt and $\{\delta QI\}_{j+1}^1$, the maximum value of $\{\delta QI\}_{j+1}^1$ is forced to match a given value called DQMAX. The percentage change between DQMAX and the actual maximum of $\{\delta QI\}_{j+1}^1$ is then applied to Δt and to the remaining values of $\{\delta QI\}_{j+1}^1$. The revised values of $\{\delta QI\}_{j+1}^1$ are used to calculate the values $\{QI\}_{j+1}^2$ which, along with the revised Δt , are used to calculate $\{\delta QI\}_{j+1}^2$. In the first level of optimization, the value of DQMAX and the initial value of Δt are input variables.

The second level of optimization varies the value of DQMAX at a given timestep based on the convergence history of the preceeding timestep. The number of iterations per timestep is given a range of acceptability. If the number of iterations is above the acceptable range, the value of DQMAX is decreased by a given factor. Likewise, if the number of iterations is below the acceptable range, the value of DQMAX is increased by a given factor. The range of acceptability and the adjustment factors for DQMAX are input variables.

CHAPTER V

COMPUTER CODE

The computer code used in this analysis is based on programs written by M.O. Soliman in 1979. Other than those reflecting revisions in the algorithm, the most notable changes have been a generalization of the input stream to allow one code to handle a variety of geometries and initial conditions and revisions of array names and methods of utilization of arrays to allow direct comparison of the code and the algorithm. The computer code and sample input files are presented in Appendix E.

The computer code is made up of a series of subroutines with the main program consisting of only two executable statements. The first sets the error message controls for the IBM 3081 and the second calls subroutine INPUT. In subroutine INPUT all variables required to define a specific problem are read into the program while the data which is read is used to develop the full description of the given problem.

The first set of data to be read in INPUT is the global solution parameters such as the degree of the basis function, the number of elements in the discretization, and the penalty constraint parameters.

The second set of data is discretization parameters used to develop the finite element grid and define the geometry of the problem. The discretization parameters define "macro-elements" each

of which are discretized into a specified number of elements of either uniform or nonuniform measure. The cross sectional area at each node is calculated from a quadratic curve fit of the cross section areas at the vertex nodes and one interior node of the given macro-element. The coefficients for the quadratic curve fit are calculated in subroutine CSAREA.

Specification for the initial conditions of the problem are read in the next section of input. The first variable read in this section (ICGEN) controls the mode of input and generation for the initial conditions. When ICGEN=1, a number of regions and a reference temperature are read. The values of density, momentum, and pressure are then read for the specified regions. The regional values of the energy is calculated from the input values of density, momentum, and pressure. The values of these variables are assigned to each node in the region and nondimensionalized in subroutine INIT. When ICGEN=2, a shock location and inlet value for temperature, pressure, and mass flow rate are read, and, if ICGEN=3, the inlet Mach number is read instead of the mass flow rate. The inlet variables are nondimensionalized and isentropic relations are used to calculate nodal values of the variables in subroutine ISNTRP. The last variable for this section is a flag for either accepting the calculated exit pressure or reading a different exit pressure.

The remaining data read by input are flags for setting either Dirichlet or Neumann boundary conditions on each variable at the inlet and outlet, time integration and optimization controls, and output controls. The last function of subroutine INPUT is to call

subroutine FESOLV. Specific variable names along with definitions and input formats are given in program comments in Appendix E.

With the termination of subroutine INPUT, no additional user interface with the program occurs. The following is a brief synopsis of the utility of all subroutines other than INPUT. A more complete idea of the flow of the program may be obtained from program comments in Appendix E.

FESOLV--called by INPUT. Controls stepwise time integration of the problem and the Newton method iteration at each timestep. Also controls timestep optimization and solution output.

FANDJ--called by FESOLV. Generates elemental matrices from nodal values of variables, assembles elemental matrices into block banded system matrices, applies specified boundary conditions to each system matrix, and assembles block banded system matrices into global matrices. Also locates maximum "Delta" values for each variable after solution.

EIMAT--called by FESOLV. Multiplies integer form of hypermatrices by appropriate coefficients and forms [A M--] matrices.

SORT--called by EIMAT. Rearranges matrix entries computed from block data in ELMAT into arrays directly usable in FANDJ.

OUTPUT--called by FESOLV. Calculates solution integral norms and prints output of solution in integer format. Also prints output into graphics file as specified in INPUT.

BANDET--called by FANDJ. Computes the LU decomposition of the non-symmetric banded global Jacobian matrix stored in diagonal format using partial pivoting.

BANDSL--called by FANDJ. Uses back substitution to compute the solution of the linear system using the LU decomposition of the coefficient matrix and the right handside F.

CSAREA--called by INPUT. Uses Kramer's rule to calculate coefficients for the quadratic curve fit of the input macro-element nodal cross-sectional areas.

DETERM--called by CSAREA. Calculates determinant of a 3x3 matrix.

INIT--called by INPUT. Generates initial condition nodal distributions of pressure, density, and momentum; calculates energy and stagnation reference states, and nondimensionalizes each variable array on the appropriate reference condition.

ISNTRP--called by INPUT. Computes initial condition distribution for air using isentropic/normal shock relationships with defined cross-sectional areas and either mass flow rate (ICGEN=2) or inlet flow mach number (ICGEN=3); calculates stagnation reference states, and generates nodal nondimensional variable distributions.

BLOCK DATA--stores finite element hypermatrix data.

CHAPTER VI

RIEMANN SHOCK TUBE

The Riemann Shock Tube Simulation was chosen as the primary test case for this analysis. The Riemann Shock Tube, cf. Shapiro (1953, p. 1007) and Schreier (1982, p. 234), provides a complex solution structure with steep gradients and discontinuities interspersed with planar plateau regions. The intricacies of the solution test the stability of the algorithm while the existence of analytical solutions allow estimation of the accuracy of the approximation. The Shock Tube problem has two shortcomings which necessitate the use of additional test cases (see CHAPTER VII). Firstly, the shock tube involves a constant area duct and, therefore, the solution is not area dependent. Secondly, the convergence rate is dominated by the area around the discontinuity.

The Riemann Shock Tube consists of two chambers, separated by a diaphragm, containing gases with differing stagnation conditions. The starting time for the problem ($t=0$) is marked by the sudden removal of the diaphragm sending a shock wave into the low pressure chamber and a rarefaction wave into the high pressure chamber. For this analysis the diaphragm is placed at the midpoint of a duct with a constant cross-sectional area of unity. The duct has a unit length which is discretized into $12 \leq M \leq 200$ finite elements R_e^1 of uniform length h_e . The initial conditions are $m(x)=0$, $p(x)=1=\rho(x)$ on $0 \leq x \leq 0.5$ and $p(x)=0.1$ and $\rho(x)=.125$ on $0.5 < x \leq 1.0$ with $\gamma=1.4$. The

stopping time for all tests is $n\Delta t = 0.14154$ seconds. At the specified stopping time the shock is centered at $x = 0.77$ and the contact discontinuity is centered at $x = 0.62$. Dirichlet boundary conditions are imposed on all variables on each end of the discretization.

The first run is a recreation of the solution from the original formulation in the nondimensional form used in the revised computer code. This run (RMN01) uses a $k=1$, $M=100$ discretization with the "numerically optimized" dissipation parameter set, $v_{\alpha}^1 = v\{3/8, 0, 1/4\}$ and $v_{\alpha}^2 = v\{3/4, 2, 1\}$, $v = (15)^{-1/2}$, presented by Baker (1982). Although not numerically comparable due to different forms of the basic equation set, the solution from RMN01 is proportionally identical to the solution from the original computer code. The RMN01 solution is presented graphically in Figure 1. The analytical solution [Sod, 1978] is also presented in Figure 1.

Four series of tests have been run in order to assess the effects of revisions on the performance characteristics of the algorithm. The first series checks the effect of changes in terms resulting directly from the Euler equations. The second series of tests uses the "best case" from the first series to check revisions in the terms of the artificial dissipation mechanism. The last two series of tests use the fully revised formulation of the algorithm. The third series of tests is used to determine solution accuracy and convergence, measured in several norms, as functions of grid refinement. The fourth series of tests measures solution accuracy as a function of Courant number.

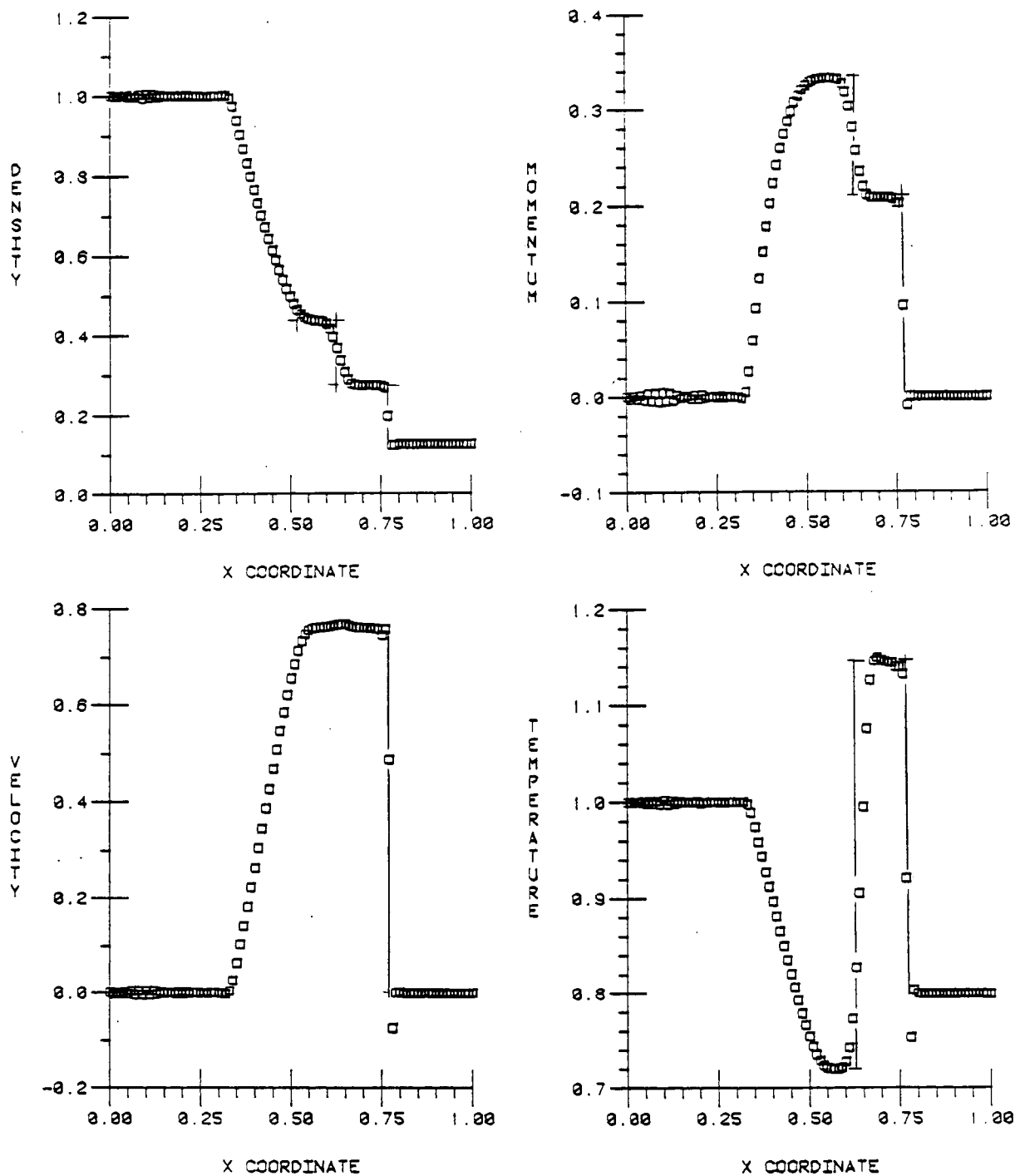


FIGURE 1. Original Formulation of Algorithm, Riemann Shock Tube Solution.

(—) analytical solution

Revisions to Euler Equation Terms

Four revisions have been made in terms which are derived from the Euler equations and the equation of state. Three of these revisions are checked individually in this series of tests. The fourth revision involves area dependent terms and is analyzed in CHAPTER VII. A fourth run has been made in this series to determine the effect of the three revisions taken collectively. The following is a brief description of the runs in this series.

RMN01 - original formulation

RMN02 - revision to the derivatives $\partial\{V\}/\partial\{R\}$ and $\partial\{V\}/\partial\{M\}$.

RMN03 - inclusion of the term $\{A10\}^T\{1/R\}_e\{M\}^T[A3000]$ in $[JPM]_e$, equation (31). This term had been omitted^e in original formulation.

RMN04 - inclusion of the spatial derivative term of $[JRR]_e$, equation (28), this term had been omitted in original formulation.

RMN05 - revisions from the three preceeding runs taken together.

The aim of the revisions to the algorithm is to increase the efficiency of the algorithm while maintaining or increasing the accuracy of the solution. The energy norm defined as

$$||q_\alpha^h||_E = \frac{1}{2} \left[\int_R (v q_\alpha^h)^2 dx \right] \quad (36)$$

can be used as a measure of the error in the solution [Becker, et al., 1981]. Using the energy norm as a measure of accuracy for this

series of tests indicates that the solution is unaffected by the revisions made in the algorithm and computer code. These data are presented in Table 1.

TABLE 1. Accuracy and Efficiency, Riemann Shock Tube, Euler Equation Revisions, $M=100$, $k=1$, $\varepsilon=0.0001$.

Run	Total Iterations	Go Step CPU (Normalized)	Energy Norm $ q_{\alpha}^h _E$			
			ρ	m	g	p
RMN01	363	1.000	1.55	1.65	2.79	2.44
RMN02	362	1.005	1.55	1.65	2.79	2.44
RMN03	363	1.005	1.55	1.65	2.79	2.44
RMN04	362	1.011	1.55	1.65	2.79	2.44
RMN05	299	0.829	1.55	1.65	2.79	2.44

Also presented in Table 1 are tabulations of the total number of iterations and the go step CPU required to reach the desired solution. These measures of efficiency indicate that the revisions in runs RMN02, RMN03 and RMN04 have little influence on the performance of the algorithm. However, the combination of these revisions in RMN05 produces a significant decrease in the computational effort required for a solution. More specifically, RMN05 requires

17.6% fewer iteration and 17.1% less CPU than the original formulation in RMN01.

A more accurate assessment of the impact of the revisions is obtained by observing the convergence history of the $\{\delta QI\}_{\max}$ values at a given time step. Tables 2 and 3 give the convergence history for the independent variables at two consecutive time steps.

Looking at the convergence history of RMN01 it can be seen that $\{R\}$ and $\{P\}$ are the variables which require the fourth iteration to meet the criteria for convergence. Little change is seen between the data from RMN01 and RMN02. A significant change occurs in the convergence history of $\{P\}$ in RMN03. The revision in RMN03 is in the term $[JPM]_e$ which is the derivative of the equation of state, defining the pressure, with respect to momentum. After this correction the requirement for a fourth iteration is solely on $\{R\}$. Similarly in RMN04, the revision is to the term $[JRR]_e$ which is the derivative of the continuity equation, defining the density, with respect to density. After this correction the fourth iteration is required by $\{P\}$ only.

The behavior exhibited in runs RMN03 and RMN04 provides possible explanations for the behavior of runs RMN02 and RMN05. The revision of RMN02 affects derivatives of the velocity with respect to density and momentum. The velocity is a dependent variable whose derivatives appear in Jacobian terms from all of the basic equations and, thus, the influence of the revision in RMN02 would be distributed among all of the dependent variables reducing the readily observable significance of the change. All three revisions are

TABLE 2. Newton Iteration Convergence in $\{\delta Q\}$, Riemann Shock Tube, Euler Equation Revisions, $M=100$, $k=1$, $\varepsilon=0.0001$, $n\Delta t=0.0739\text{sec}$.

Run	Iteration	$\{\delta R\}_{MAX}$	Node	$\{\delta M\}_{MAX}$	Node	$\{\delta G\}_{MAX}$	Node	$\{\delta P\}_{MAX}$	Node
RMN01	1	0.25155D-01	66	0.45649D-01	66	0.51752D-01	66	0.48420D-01	66
	2	0.28900D-02	66	-0.83826D-03	66	-0.22763D-02	66	-0.83336D-02	66
	3	-0.24727D-03	66	-0.13772D-04	65	-0.28020D-04	65	0.15224D-03	66
	4	-0.20320D-04	65	0.36155D-05	67	-0.22571D-05	64	0.35910D-05	65
RMN02	1	0.25110D-01	66	0.45805D-01	66	0.52141D-01	66	0.48599D-01	66
	2	0.29514D-02	66	-0.96752D-03	66	-0.26067D-02	66	-0.85095D-02	66
	3	-0.26251D-03	66	-0.34597D-04	66	-0.49125D-04	66	0.14027D-03	66
	4	-0.20044D-04	65	0.54770D-05	67	0.77142D-05	66	0.93605D-05	66
RMN03	1	0.25209D-01	66	0.45484D-01	66	0.51679D-01	66	0.43956D-01	66
	2	0.28159D-02	66	-0.69203D-03	66	-0.22139D-02	66	-0.37492D-02	66
	3	-0.22845D-03	66	-0.33513D-04	67	0.30271D-04	66	0.33040D-04	66
	4	0.21651D-04	66	0.53539D-05	67	0.27706D-05	65	0.11950D-05	67
RMN04	1	0.24695D-01	66	0.45650D-01	66	0.51753D-01	66	0.48404D-01	66
	2	0.31650D-02	66	-0.84935D-03	66	-0.22983D-02	66	-0.83526D-02	66
	3	-0.45894D-04	66	-0.13067D-04	65	0.30182D-04	66	0.18187D-03	66
	4	0.13816D-05	66	0.13661D-05	67	0.98065D-06	67	0.30974D-05	65
RMN05	1	0.24705D-01	66	0.45642D-01	66	0.52065D-01	66	0.43912D-01	66
	2	0.31491D-02	66	-0.83483D-03	66	-0.25653D-02	66	-0.37111D-02	66
	3	-0.40722D-04	66	-0.36802D-04	67	-0.45670D-04	67	-0.36979D-04	67

TABLE 3. Newton Iteration Convergence in $\{\delta QI\}$, Riemann Shock Tube, Euler Equation Revisions, $M=100$, $k=1$, $\epsilon=0.0001$, $n\Delta t=0.0755\text{sec}$.

Run	Iteration	$\{\delta R\}_{\max}$	Node	$\{\delta M\}_{\max}$	Node	$\{\delta C\}_{\max}$	Node	$\{\delta P\}_{\max}$	Node
RMN01	1	0.24596D-01	66	0.44012D-01	66	0.57411D-01	66	0.52972D-01	66
	2	0.23273D-02	66	-0.95866D-03	67	-0.28479D-02	66	-0.10174D-01	66
	3	0.23009D-03	65	-0.13671D-04	67	0.23851D-04	64	0.14704D-03	66
	4	-0.26446D-04	65	0.18016D-05	66	0.54984D-05	66	0.50693D-05	66
RMN02	1	0.24577D-01	66	0.43956D-01	66	0.57712D-01	66	0.52971D-01	66
	2	0.23089D-02	66	-0.14296D-02	67	-0.31112D-02	66	-0.10173D-01	66
	3	0.22566D-03	65	-0.33434D-04	67	-0.35934D-04	66	0.14109D-03	66
	4	-0.24409D-04	65	0.29278D-05	66	0.94740D-05	66	0.94514D-05	66
RMN03	1	0.24652D-01	66	0.44021D-01	66	0.57414D-01	66	0.46355D-01	66
	2	0.22762D-02	66	-0.61643D-03	66	-0.28519D-02	66	-0.34199D-02	66
	3	0.20181D-03	65	-0.18245D-04	67	0.18745D-04	64	0.24754D-04	67
	4	-0.23686D-04	65	0.11724D-05	67	0.35633D-05	66	0.35519D-05	66
RMN04	1	0.24027D-01	66	0.44005D-01	66	0.57390D-01	66	0.52910D-01	66
	2	0.27294D-02	66	-0.95589D-03	67	-0.28396D-02	66	-0.10147D-01	66
	3	-0.69398D-04	67	-0.21094D-04	67	0.19629D-04	66	0.18274D-03	66
	4	0.30214D-05	66	0.19619D-05	66	0.35678D-05	66	0.25769D-05	66
RMN05	1	0.24023D-01	66	0.43969D-01	66	0.57695D-01	66	0.46022D-01	66
	2	0.27087D-02	66	-0.10321D-02	67	-0.31010D-02	66	-0.31144D-02	66
	3	-0.81242D-04	67	-0.48713D-04	67	-0.41203D-04	67	0.38589D-04	66

incorporated in RMN05 and the effects of each revision is reflected in the solution. The fourth iteration in RMN01 is required by both $\{R\}$ and $\{P\}$. The revision RMN03 allows $\{P\}$ to converge in three steps and the revision RMN04 allows $\{R\}$ to converge in three steps, thus, in combination in RMN05 the convergence of the equation system is reduced to three steps and a significant decrease in computational effort is realized.

Revisions to Dissipation Mechanism

Three changes have been made to terms derived from the artificial dissipation mechanism. This series of tests begins with the fully revised formulation from the preceeding series and each revision is added to the previously included revisions. The following is a brief description of the runs in this series.

RMN05 - fully revised formulation from first series of tests

RMN06 - revision to the derivative $\partial\{\bar{V}\}^T[A3011]\{R\}/\partial\{M\}$ in [JRM].

RMN07 - revision to the derivatives $\partial\{\bar{V}\}/\partial\{R\}$ and $\partial\{\bar{V}\}/\partial\{M\}$.

RMN08 - normalization of velocity in v_2^α with respect to the

$$\text{maximum velocity, i.e. } v_i = |v_i/v_{\max}|.$$

RMN09 - adjustment of optimized values of v_2^α such that $v_2^\alpha =$

$$v\{\frac{1}{2}, \frac{3}{2}, \frac{3}{4}\}, \quad v = (15)^{-1/2}$$

Based on the experience of the first series of tests, RMN06 should produce a significant change in the $\{\delta R\}_{\max}$ terms. This is borne out by the data presented in Tables 4 and 5 which shows a noticable decrease in the values of $\{\delta R\}_{\max}$ at iterations 2 and 3

TABLE 4. Newton Iteration Convergence in $\{\delta Q\}$, Riemann Shock Tube, Penalty Constraint Revisions, $M=100$, $k=1$, $\epsilon=0.0001$, $n\Delta t=0.0739\text{sec}$.

Run	Iteration	$\{\delta R\}_{\max}$	Node	$\{\delta M\}_{\max}$	Node	$\{\delta G\}_{\max}$	Node	$\{\delta P\}_{\max}$	Node
RMN05	1	0.24705D-01	66	0.45642D-01	66	0.52065D-01	66	0.43912D-01	66
	2	0.31491D-02	66	0.83483D-03	66	-0.25653D-02	66	-0.37111D-02	66
	3	-0.40722D-04	66	-0.36802D-04	67	-0.45670D-04	67	-0.36979D-04	67
RMN06	1	0.27751D-01	66	0.45648D-01	66	0.52154D-01	66	0.44112D-01	66
	2	-0.17956D-03	67	-0.84077D-03	66	-0.26924D-02	66	-0.39000D-02	66
	3	-0.78932D-05	66	0.25194D-04	67	0.28547D-04	66	0.26043D-04	67
RMN07	1	0.27751D-01	66	0.45649D-01	66	0.52156D-01	66	0.44114D-01	66
	2	-0.17916D-03	67	-0.84161D-03	66	-0.26944D-02	66	-0.39017D-02	66
	3	-0.78498D-05	66	0.25046D-04	67	0.28739D-04	66	0.25635D-04	67
RMN08	1	0.23722D-01	66	0.38492D-01	66	0.47833D-01	66	0.39350D-01	66
	2	-0.22577D-03	67	-0.53293D-03	67	-0.24143D-02	66	-0.28880D-02	66
	3	0.37036D-05	66	0.19624D-04	66	0.34407D-04	66	0.30876D-04	66
RMN09	1	0.26881D-01	66	0.46938D-01	66	0.51253D-01	66	0.43061D-01	66
	2	-0.79290D-04	67	-0.99153D-03	66	-0.25631D-02	66	-0.39592D-02	66
	3	0.17900D-05	66	0.26051D-04	66	0.51759D-04	66	0.35482D-04	66

TABLE 5. Newton Iteration Convergence in $\{\delta QI\}$, Riemann Shock Tube, Penalty Constraint Revisions, $M=100$, $k=1$, $\varepsilon=0.0001$, $n\Delta t=0.0755$ sec.

Run	Iteration	$\{\delta R\}_{\max}$	Node	$\{\delta M\}_{\max}$	Node	$\{\delta G\}_{\max}$	Node	$\{\delta P\}_{\max}$	Node
RMN05	1	0.24023D-01	66	0.43969D-01	66	0.57695D-01	66	0.46022D-01	66
	2	0.27087D-02	66	-0.10321D-02	67	-0.31010D-02	66	-0.31144D-02	66
	3	-0.81242D-04	67	-0.48713D-04	67	-0.41230D-04	67	0.38589D-04	66
RMN06	1	0.26369D-01	66	0.44022D-01	66	0.57861D-01	66	0.46323D-01	66
	2	-0.38982D-03	67	-0.10562D-02	67	-0.33402D-02	66	-0.34271D-02	66
	3	-0.26704D-05	68	0.23520D-04	66	0.51382D-04	66	0.50508D-04	66
RMN07	1	0.26370D-01	66	0.44024D-01	66	0.57870D-01	66	0.46331D-01	66
	2	-0.39003D-03	67	-0.10530D-02	67	-0.33486D-02	66	-0.34349D-02	66
	3	-0.26639D-05	68	0.23507D-04	66	0.51440D-04	66	0.50509D-04	66
RMN08	1	0.20645D-01	66	0.32904D-01	66	0.46058D-01	66	0.36558D-01	66
	2	-0.20359D-03	67	-0.13963D-02	66	-0.32459D-02	66	-0.27333D-02	66
	3	0.60354D-05	67	0.20967D-04	67	0.51272D-04	66	0.33148D-04	67
RMN09	1	0.26537D-01	66	0.46259D-01	66	0.58519D-01	66	0.46265D-01	66
	2	0.11851D-02	66	-0.98948D-03	66	-0.15430D-02	66	-0.15431D-02	66
	3	0.48454D-05	66	0.20457D-04	66	0.20799D-04	66	0.26803D-04	66

while other variables are essentially unchanged giving a quadratic convergence rate in all variables. This change is also reflected in the computational effort as shown in Table 6 with an 8.0% reduction in the total number of iterations and a 2.4% reduction in CPU.

Comparing RMN06 to RMN01, the original formulation, yields overall reductions of 24.2% in the total number of iterations and 19.1% in CPU. While realizing the significant improvement in the convergence rate; the accuracy of the solution, as measured in the energy norm, is unchanged. Comparison of the graphical presentations of the solutions of RMN01 and RMN06, Figures 1 and 2 respectively, shows the solutions to be visually identical.

TABLE 6. Accuracy and Efficiency, Riemann Shock Tube, Penalty Constraint Revisions, $M=100$, $k=1$, $\epsilon=0.0001$

Run	Total Iterations	Go Step CPU (Normalized)	Energy Norm $\ q_{\alpha}^h \ _E$			
			ρ	m	g	p
RMN05	299	1.000	1.55	1.65	2.44	2.79
RMN06	275	.976	1.55	1.65	2.44	2.79
RMN07	275	.974	1.55	1.65	2.44	2.79
RMN08	275	.979	1.35	1.35	2.28	2.09
RMN09	278	.989	1.61	1.70	2.90	2.53

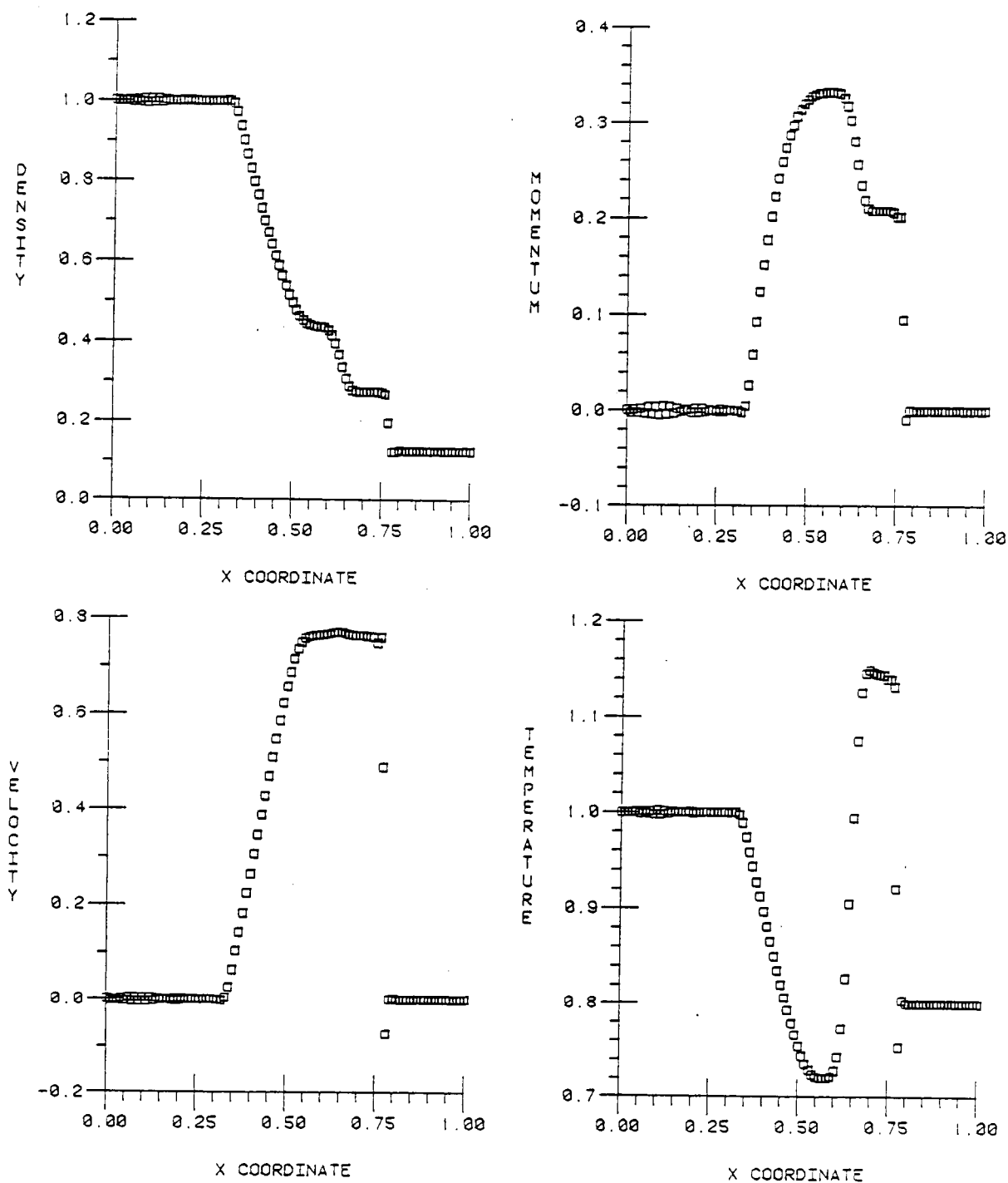


FIGURE 2. Revisions to Penalty Constraint Terms,
 $v_2 = v_0 \{3/4, 2, 1\}$, $v_0 = (15)^{1/2}$.

The revisions in RMN07 produce almost no change in either the convergence history or computational effort. This is expected for two reasons. First, the revisions are based on derivatives of the velocity in the dissipation mechanism which is attached to three of the four defining equations, thus any impact would be distributed between the equations lessening the apparent significance. Second and more importantly, quadratic convergence has already been attained except during the initial time steps (see Table 7) and the monotonic convergence of the first time steps is attributable to the discontinuity in the initial conditions [Chase, 1982 and Decker, et al., 1983].

The normalization of the velocity on the dissipation mechanism terms in RMN08 produces a degradation of the solutions as indicated by the energy norms shown in Table 6. Graphs of the solution, Figure 3, show that the shock has been "smeared" over several elements indicating that the normalization of the velocity has caused the equation system to be overdamped. In RMN09 the values of v^{α} have been reduced so that diffusion level is approximately the same as that in RMN06. The energy norms and the graphs of the solution, Table 6 and Figure 4, indicate that the solution from RMN06 is roughly equivalent to the solution from the original formulation.

In one manuscript Baker (1983a) presented the conservative form of the convective portion of the Euler equations, i.e.

$$L^C(\cdot) = \frac{\partial \cdot}{\partial t} + \frac{\partial}{\partial x}[u \cdot], \text{ in the definition of the dissipation}$$

mechanism. The terms presented in that manuscript are those

TABLE 7. Newton Iteration Convergence in $\{\delta Q\}$, Riemann Shock Tube, Initial Time Steps, RMN07, $M=100$, $k=1$, $\epsilon=0.0001$, $n\Delta t=0.0739\text{sec}$.

Time	Iteration	$\{\delta R\}$ MAX	Node	$\{\delta Q\}$ MAX	Node	$\{\delta G\}$ MAX	Node	$\{\delta P\}$ MAX	Node
0.157D-02	1	-0.19233D-01	52	0.76199D-01	53	0.49570D-01	53	0.49570D-01	53
0.157D-02	2	-0.25357D-02	53	0.37807D-02	54	-0.38228D-01	52	-0.39799D-01	52
0.157D-02	3	-0.10455D-03	52	0.40273D-03	53	0.10582D-02	52	0.10089D-02	52
0.157D-02	4	0.11456D-04	53	-0.25944D-04	53	-0.66607D-04	52	-0.69868D-04	52
0.315D-02	1	-0.45893D-01	52	0.67792D-01	53	0.10499D+00	53	0.90193D-01	53
0.315D-02	2	0.38830D-02	52	-0.34704D-02	53	-0.27238D-01	52	-0.26062D-01	52
0.315D-02	3	-0.10850D-03	52	0.24183D-03	54	0.11229D-02	52	0.97797D-03	52
0.315D-02	4	0.50956D-05	52	-0.27995D-04	54	-0.49255D-04	52	-0.39218D-04	52
0.472D-02	1	0.55420D-01	53	0.53571D-01	53	0.11488D+00	53	0.97745D-01	53
0.472D-02	2	-0.29335D-02	53	-0.31456D-02	54	-0.17901D-01	53	-0.13987D-01	53
0.472D-02	3	-0.82438D-04	52	0.12888D-03	53	0.60547D-03	53	0.36448D-03	53
0.472D-02	4	0.16767D-05	52	-0.64023D-05	53	-0.21106D-04	53	-0.14468D-04	53
0.629D-02	1	-0.47343D-01	52	0.48510D-01	54	0.78978D-01	53	-0.81135D-01	52
0.629D-02	2	0.17267D-02	52	-0.26638D-02	53	-0.12252D-01	53	-0.79143D-02	53
0.629D-02	3	0.34369D-04	54	0.15619D-03	54	0.31194D-03	53	0.20646D-03	53
0.629D-02	4	-0.18360D-05	54	-0.65478D-05	54	-0.11590D-04	53	-0.11288D-04	53
0.786D-02	1	-0.41314D-01	52	0.57983D-01	54	0.64952D-01	54	-0.63534D-01	52
0.786D-02	2	-0.10932D-02	53	-0.16402D-02	54	-0.56247D-02	53	-0.55381D-02	54
0.786D-02	3	-0.15211D-04	52	0.59433D-04	54	0.12501D-03	53	0.95926D-04	53
0.786D-02	4	-0.40717D-06	54	-0.30634D-05	54	-0.38699D-05	54	-0.34867D-05	53
0.944D-02	1	-0.32922D-01	52	0.52986D-01	54	0.71908D-01	54	0.55934D-01	54
0.944D-02	2	-0.75435D-03	53	-0.11502D-02	55	-0.34562D-02	54	-0.43769D-02	54
0.944D-02	3	0.12910D-04	53	0.23225D-04	54	0.32675D-04	54	0.78179D-04	54

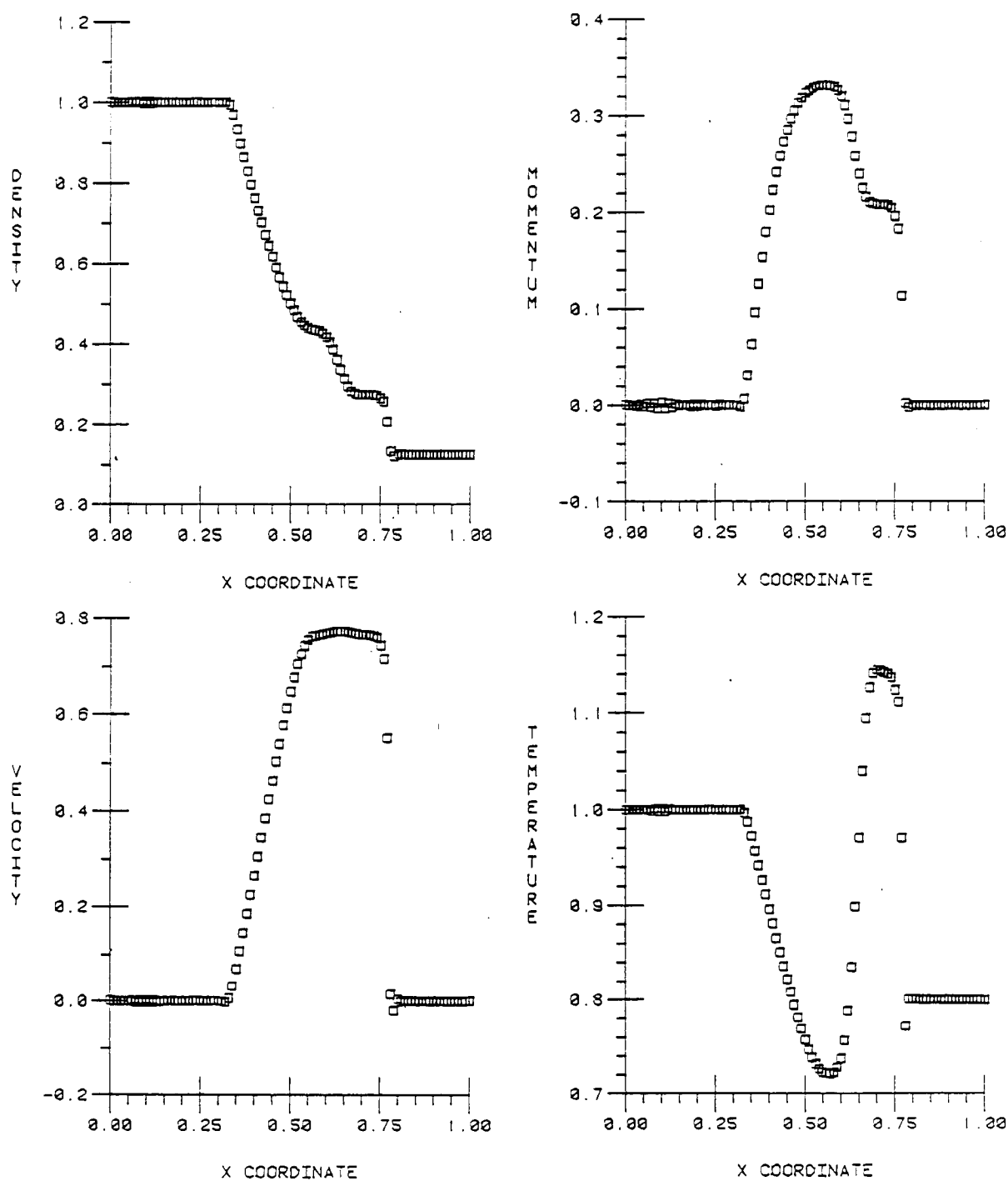


FIGURE 3. Fully Revised Formulation with $V = |V/V_{\max}|$ in Penalty Constraint Terms, Riemann Shock Tube Solution, $v_2 = v_0 \{3/4, 2, 1\}$, $v_0 = (15)^{1/2}$.

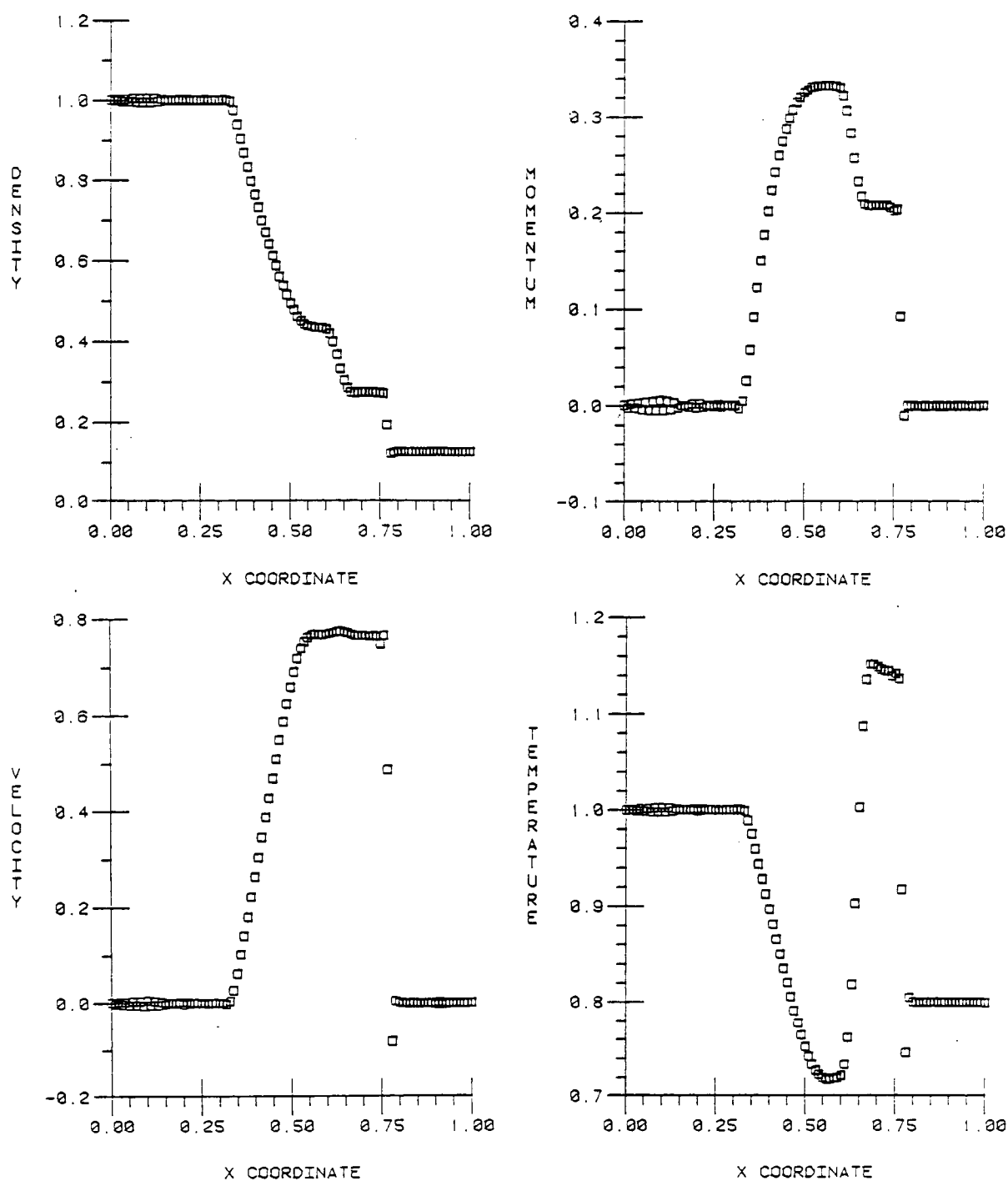


FIGURE 4. Fully Revised Formulation of Algorithm with $V = |V/V_{\max}|$ in Penalty Constraint Term, Riemann Shock Tube Solution, $v_2 = v_0\{1/2, 3/2, 3/4\}$, $v_0 = (15)^{-1/2}$

derived from the nonconservative form. The spatial derivative terms of the dissipation mechanism using the conservative convective derivatives are of the form

$$\{\bar{V}\}_e [A3011] \{q_\alpha\}_e + \{\bar{q}_\alpha\}_e [A3110] \{V\}_e \quad (37)$$

This form of the diffusion term is used in RMN10. The graphical presentation of the solution RMN10, Figure 5, shows extremely poor coherence of the solution in the low pressure region beyond the shock. The behavior of solution RMN10 indicates the nonconservative form of the convective derivative to be the correct form for the dissipation mechanism.

Efficiency and Accuracy--Fully Revised Formulation

Two series of tests have been performed to check the performance characteristics of the fully revised formulations. The first series of tests measures the change in the accuracy with respect to grid refinement. The second series measures the change in the efficiency with respect to the time increment (Courant number). The first series of tests applies a Courant number of 0.18 over a discretization range of $12 \leq M \leq 200$. A Richardson extrapolation step [Botha and Pinder, 1983] at the standard end time confirms that the digit of significance is unaffected by the temporal truncation error associated with the Courant number, as required for estimation of semi-discrete approximation error. Convergence characteristics, measured in H^1 and E , for $k=1$ solutions are presented in Figure 6. The data

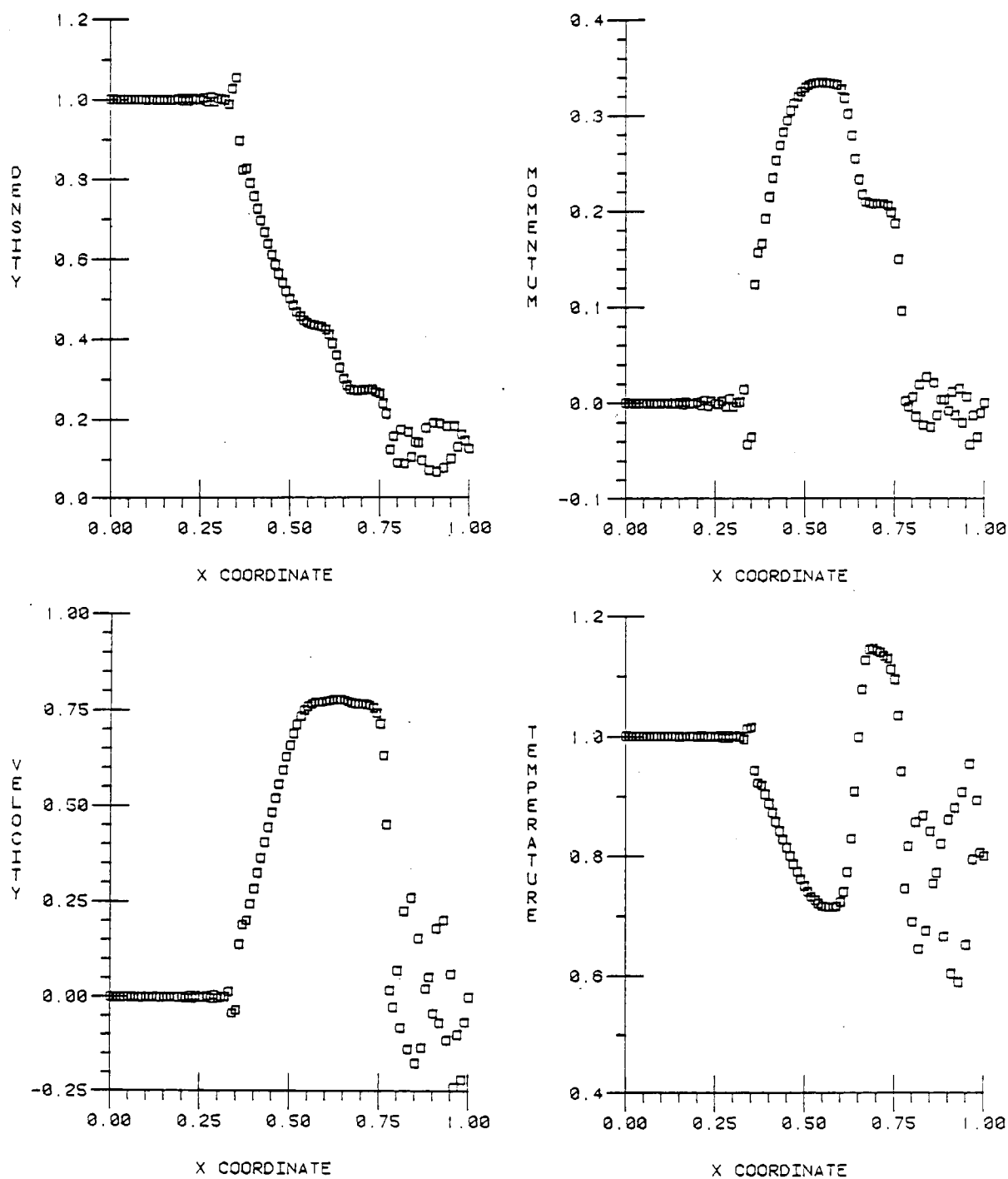


FIGURE 5. Penalty Constraint Derived from Conservative Form of Convective Derivatives, Riemann Shock Tube Solution.

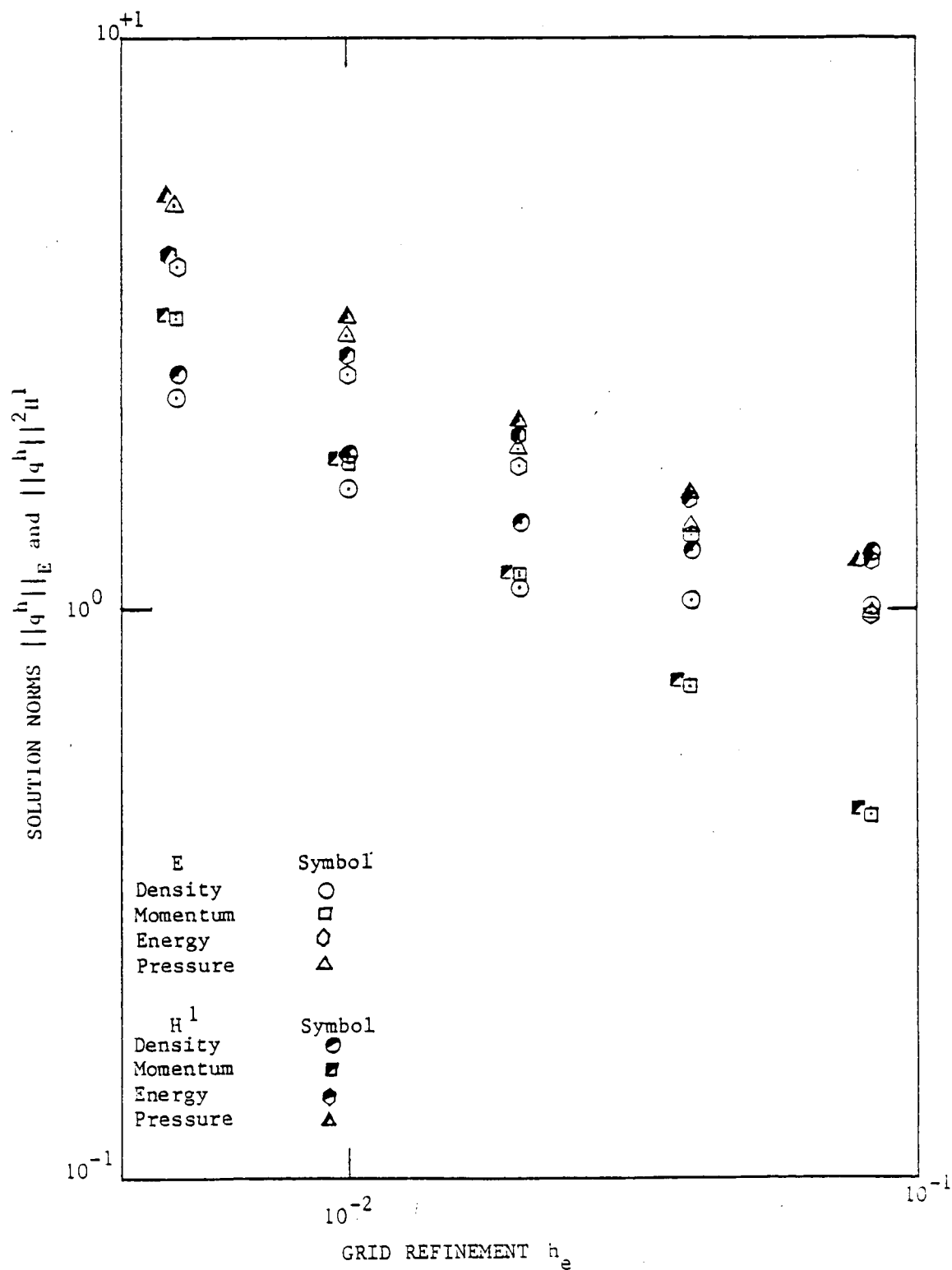


FIGURE 6. Semi-discrete Approximation Accuracy and Convergence in $||q^h||_E$ and $||q^h||_{H^1}$ Fully Revised Algorithm, Riemann Shock Tube Solution.

obtained in these tests are in substantial agreement with data obtained from the original formulation [Baker, 1982]. Assuming slope distinctions between dependent variables to be insignificant, the nominal convergence rate is approximately 2/3, indicating qualitative agreement with a p^{th} order accurate different scheme, or symbolically,

$$||q_{\alpha}^h|| < C_1 h^{p/p+1} + \dots \quad (38)$$

where $p=2$ and $c_1 \neq f(h)$ for sufficiently small h [Baker, 1982]. As possible additional methods for estimating solution accuracy, the H^0 and L^1 norms have been calculated. However, no discernable pattern has been found in these data, presented in Tables 8 and 9.

TABLE 8. Solution Norm $||q_{\alpha}^h||_{H^0}^2$, Riemann Shock Tube, $k=1$, $\epsilon=0.001$, Courant No. = .18.

Element Measure h_e	Dependent Variable			
	Density	Momentum	Energy	Pressure
5.0 E-03	0.2279	0.0149	0.2216	0.2142
1.0 E-02	0.2316	0.0146	0.2251	0.2178
2.0 E-02	0.229	0.0142	0.222	0.215
4.0 E-02	0.224	0.0134	0.216	0.209
8.33 E-02	0.244	0.012	0.235	0.228

TABLE 9. Solution Norm $\|q_u^h\|_L^2$, Riemann Shock Tube, $k=1$, $\epsilon=0.001$,
Courant No. = .18.

Element Measure h_e	Independent Variable			
	Density	Momentum	Energy	Pressure
5.0 E-03	0.5709	0.107	0.5586	0.5387
1.0 E-02	0.5779	0.107	0.5658	0.5462
2.0 E-02	0.5740	0.107	0.5618	0.5429
4.0 E-02	0.5659	0.107	0.5535	0.535

The second series of tests applies a range of time integration increments to the standard test case of $k=1$, $M=100$, $\epsilon=0.001$. Primary factors determining solution efficiency are the discretization, the integration time step, and the convergence level. Data from these tests, presented in Table 10 indicates computational effort can be cut approximately in half by doubling the Courant number (time step). After this point, diminishing returns are encountered as the number of Newton iterations per timestep to attain the desired level of convergence increases. Also as the Courant number increases the effect of the temporal truncation error increases and the accuracy, as measured in the energy norm, is degraded with the most serious deterioration of the solution

TABLE 10. Accuracy and Efficiency, Riemann Shock Tube with Respect to Courant Number,
 $M=100$, $k=1$, $\epsilon=0.001$.

Maximum Courant Number	Number of Timesteps	Total Iterations	Go Step CPU	Energy Norm $ q_a^h _E$			
				Density	Momentum	Energy	Pressure
.143	90	272	39.46	1.61	1.70	2.91	2.53
.287	45	142	21.24	1.55	1.63	2.74	2.40
.583	22	89	13.68	1.47	1.45	2.44	2.18
1.124	12	51	8.22	1.38	1.25	2.17	1.97

occurring in the region between the shock and the contact discontinuity (see Figure 7). Similar behavior was observed by Baker (1982) using the original formulation.

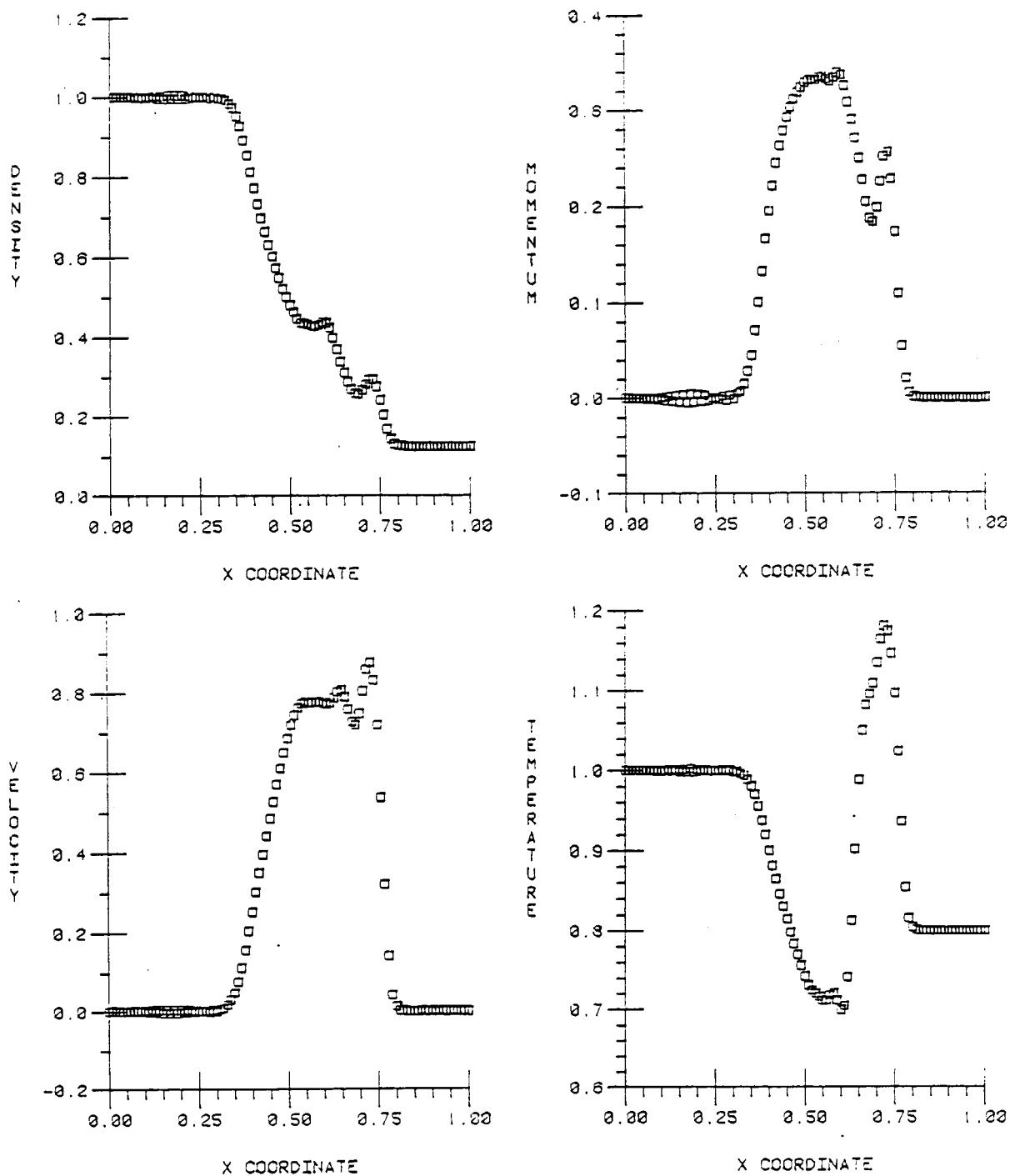


FIGURE 7. Fully Revised Algorithm, $\Delta t = 8 \times \Delta t$ Standard Riemann Shock Tube Solution.

CHAPTER VII

DeLAVAL NOZZLE

A second test case is required for analysis of the performance characteristics of the finite element algorithm due to three properties of the Riemann Shock tube problem. The first of these properties is the lack of area dependence of the shock tube. The shock tube is described over a constant area duct, thus after non-dimensionalization the tube has a unit area throughout. The area terms are of the form $f(q_\alpha)d\log A/dx$, which equals zero when A is constant. One revision to the algorithm is the inclusion of area terms omitted from the original algorithm but the impact of this revision would not be evident in the shock tube solution as the term would be zero. The second property of the solution is nonsmoothness over the time period of interest. A discontinuity is introduced as an initial condition and is maintained as a shock throughout the given interval. Although some smearing of the shock occurs, the convergence rate is dominated by the area surrounding the shock. The third property of the shock tube is that the solution does not converge toward steady state conditions in the time interval of interest. The second and third properties have no direct bearing on the analysis of algorithmic changes, but do make analysis of the timestep optimization technique difficult. In order to circumvent these properties a DeLaval nozzle, or converging-diverging duct, has been used [C.F. White, 1979, pp. 513-546].

The nozzle being analyzed is a 1.00 foot duct which has been discretized into $M=100$ finite element of uniform measure $h_e=0.01$ feet. The nozzle inlet is 0.1 feet long ($1 \leq M \leq 10$) with a constant nondimensional area of 5.82183. The throat (area=1.000) is located at $x=0.50$ feet. The nozzle outlet is 0.1 feet long ($91 \leq M \leq 100$) with a constant area of 1.16820. The inlet of the nozzle has been chosen such that, with choked flow, the inlet Mach number is 0.1. This provides an approximation of a reservoir, thus allowing stable subsonic calculations. For on design exit pressure, the outlet Mach number is 1.49.

Revision of the Algorithm

In the original formulation of the Jacobian, three terms dependent on pressure and area have been omitted. The terms in question are derived from the term $\{A\}^T([A41000]\{V\})\{P\}$ in $\{FG\}$. The terms in $[JGR]_e$ and $[JQM]_e$ were omitted from the original presentation of the algorithm [Baker, 1982]. The third term, in $[JGP]_e$, was included in the algorithm but was omitted from the original computer code.

Four sets of runs were made to determine the affects of inclusion of the area terms in the Jacobian. Two of the tests are subsonic with Dirichlet boundary conditions on the pressure, density, and energy at the inlet and on the pressure at the outlet. All other variables have Neumann constraints. Initial conditions for the two subsonic sets are the same with an inlet Mach number of 0.10, a throat Mach number of 1.0, and an outlet Mach number of 0.61. The

other two tests are off design supersonic, with Dirichlet conditions on all variables at the inlet and on the pressure at the outlet.

For inlet conditions, a shock is set at node $M=85$ ($M_a=1.4$).

The first set of runs (SSS010 and SSS020) are stationary solutions, i.e. the final solution should match the initial conditions. The convergence histories of the two runs in the set at two consecutive timesteps are presented in Table 11. These data indicate that the effect of inclusion of the area terms in the energy derivatives of the Jacobian is insignificant. The accuracy of the final solutions of the two runs, as measured in the energy norm, are identical (see Table 12).

The second set of runs MSS010 and MSS020 are transient solutions which use the same initial conditions as the first set except that the pressure on the outlet node is reduced by 5% at $t=0$ seconds. Convergence histories of the two solutions, presented in Table 13, again show an insignificant impact from inclusion of the terms in question. The accuracy of the final solutions is unchanged between the two runs (Table 12). Thus the data from the second set of runs supports the conclusion from the first set of runs. Data from the steady state and transient off design supersonic test cases, presented in Tables 12 and 13, exhibit the same behavior as the subsonic cases.

Performance of Timestep Optimizer

An assessment of the utility of the timestep optimization is based on data from five test cases, each requiring three solutions.

TABLE 11. Newton Iteration Convergence in $\{\delta O\}$, Subsonic Converging-Diverging Nozzle, Effects of Inclusion of Area Terms in $\{JGI\}$.

Time	Iteration	$\{\delta R\}MAX$	Node	$\{\delta M\}MAX$	Node	$\{\delta C\}MAX$	Node	$\{\delta P\}MAX$	Node
SSS010--Steady State Solution, Area Terms Not Included in $\{JGI\}$									
0.395D-02	1	0.21210D-03	57	0.41715D-04	26	0.20159D-03	56	0.24522D-03	57
0.395D-02	2	0.50179D-05	21	-0.32211D-05	63	-0.62331D-05	61	-0.47259D-05	61
0.400D-02	1	0.21052D-03	57	0.47295D-04	101	0.20207D-03	57	0.24490D-03	57
0.400D-02	2	0.51462D-05	23	-0.30094D-05	63	-0.58370D-05	61	-0.44251D-05	61
SSS020--Steady State Solution, Area Terms Included in $\{JGI\}$									
0.395D-02	1	0.21218D-03	57	0.41406D-04	26	0.20178D-03	56	0.24529D-03	57
0.395D-02	2	0.64086D-05	21	-0.32240D-05	63	-0.61574D-05	61	-0.46441D-05	61
0.400D-02	1	0.21047D-03	57	0.47487D-04	101	0.20197D-03	57	0.24484D-03	57
0.400D-02	2	0.65481D-05	23	-0.30210D-05	63	-0.57702D-05	61	-0.43532D-05	61
MSS010--Transient Solution, Area Terms Not Included in $\{JGI\}$									
0.395D-02	1	0.43402D-03	81	-0.15289D-03	101	0.48671D-03	81	0.56746D-03	81
0.395D-02	2	0.35134D-05	21	-0.59893D-05	92	-0.13876D-04	84	-0.11724D-04	84
0.400D-02	1	0.45378D-03	82	-0.10067D-03	87	0.51747D-03	82	0.59640D-03	82
0.400D-02	2	-0.32543D-05	25	-0.63131D-05	85	-0.14772D-04	83	-0.12558D-04	83
MSS020--Transient Solution, Area Terms Included in $\{JGI\}$									
0.395D-02	1	0.43270D-03	81	-0.14999D-03	101	0.48410D-03	81	0.56548D-03	81
0.395D-02	2	0.47162D-05	21	-0.60257D-05	92	-0.13885D-04	84	-0.11756D-04	84
0.400D-02	1	0.45286D-03	82	-0.10294D-03	87	0.51558D-03	82	0.59492D-03	82
0.400D-02	2	-0.43958D-05	21	-0.62890D-05	84	-0.14562D-04	83	-0.12353D-04	83

TABLE 12. Accuracy and Efficiency Summary, Converging Diverging Nozzle, Effects of Inclusion of Area Terms in [JGI].

Run	Time Steps	Total Iterations	Go Step CPU	Energy Norm $ q_u^h _E$			
				Density	Momentum	Energy	Pressure
SSS010	250	509	1:15.47	0.2039	0.4200	0.1607	0.3352
SSS020	250	509	1:15.60	0.2039	0.4200	0.1607	0.3352
SSH010	250	581	1:26.21	2.678	0.4671	2.389	3.903
SSH020	250	581	1:26.32	2.678	0.4671	2.389	3.903
MSS010	250	566	1:23.90	0.1781	0.4046	0.1563	0.2651
MSS020	250	567	1:24.12	0.1781	0.4046	0.1563	0.2651
MSH010	250	921	2:15.79	1.279	0.4335	1.104	1.839
MSH020	250	921	2:16.93	1.279	0.4335	1.104	1.839

TABLE 13. Newton Iteration Convergence in $\{\delta QI\}$, Off Design Transonic Converging Diverging Nozzle, Effects of Inclusion of Area Terms in $\{JGI\}$.

Time	Iteration	$\{\delta R\}MAX$	Node	$\{\delta M\}MAX$	Node	$\{\delta G\}MAX$	Node	$\{\delta P\}MAX$	Node
SSH010--Steady State Solution, Area Terms Not Included in $\{JGI\}$									
0.395D-02	1	0.13971D-03	85	0.16648D-03	98	0.20998D-03	85	0.21758D-03	85
0.395D-02	2	0.20152D-04	85	-0.16307D-04	87	-0.36442D-04	86	-0.27467D-04	86
0.400D-02	1	0.12169D-03	96	0.19139D-03	96	0.21089D-03	96	0.15595D-03	96
0.400D-02	2	0.19560D-04	85	-0.15559D-04	87	-0.35927D-04	86	-0.27796D-04	86
SSH020--Steady State Solution, Area Terms Included in $\{JGI\}$									
0.395D-02	1	0.13977D-03	85	0.16635D-03	98	0.20978D-03	85	0.21732D-03	85
0.395D-02	2	0.20362D-04	85	-0.16355D-04	87	-0.35952D-04	86	-0.26953D-04	86
0.400D-02	1	0.12170D-03	96	0.19138D-03	96	0.21084D-03	96	0.15591D-03	96
0.400D-02	2	0.19733D-04	85	-0.15644D-04	87	-0.35600D-04	86	-0.27415D-04	86

TABLE 13 continued.

Time	Iteration	(δR)MAX	Node	(δQ)MAX	Node	(δG)MAX	Node	(δP)MAX	Node
MSH010---Transient Solution, Area Terms Not Included in [JGI]									
0.395D-02	1	0.73460D-02	77	0.12114D-02	79	0.96134D-02	77	0.11342D-01	77
0.395D-02	2	0.97756D-03	77	-0.87879D-03	79	-0.21064D-02	78	-0.16532D-02	78
0.395D-02	3	0.18579D-03	77	-0.17721D-03	80	-0.41350D-03	78	-0.33506D-03	78
0.395D-02	4	0.33473D-04	77	-0.33762D-04	80	-0.79446D-04	78	-0.65524D-04	78
0.400D-02	1	0.71366D-02	77	0.12237D-02	79	0.94084D-02	77	0.11104D-01	77
0.400D-02	2	0.94093D-03	77	-0.85266D-03	79	-0.20567D-02	78	-0.16199D-02	78
0.400D-02	3	0.17796D-03	77	-0.16905D-03	80	-0.39721D-03	78	-0.32213D-03	78
0.400D-02	4	0.31926D-04	77	-0.31842D-04	80	-0.75484D-04	78	-0.62200D-04	78
MSH020---Transient Solution, Area Terms Included in [JGI]									
0.395D-02	1	0.73411D-02	77	-0.12138D-02	79	0.95999D-02	77	0.11326D-01	77
0.395D-02	2	0.98488D-03	77	-0.88190D-03	79	-0.20828D-02	78	-0.16276D-02	78
0.395D-02	3	0.18456D-03	77	-0.17736D-03	80	-0.42232D-03	78	-0.34414D-03	78
0.395D-02	4	0.32796D-04	77	-0.33429D-04	80	-0.80456D-04	78	-0.66790D-04	78
0.400D-02	1	0.73411D-02	77	0.12138D-02	79	0.95999D-02	77	0.11326D-01	77
0.400D-02	2	0.94795D-03	77	-0.85567D-03	79	-0.20340D-02	78	-0.15954D-02	78
0.400D-02	3	0.17672D-03	77	-0.16919D-03	80	-0.40577D-03	78	-0.33093D-03	78
0.400D-02	4	0.31263D-04	77	-0.31520D-04	80	-0.76431D-04	78	-0.63391D-04	78

The first of the three runs for each test case (XXX00) does not use the optimization procedure. The second run (XXX01) uses the first level of optimization in which the maximum value of $\{\delta QI\}$ is linearly extrapolated to a predetermined fixed value (DQMAX). The third run (XXX02) uses the second level of optimization in which the maximum value of $\{\delta QI\}$ is linearly extrapolated to a value which is adjusted according to the number of iterations from the previous timestep. For all test cases the adjustment parameters for the second level of optimization are a 10% increase in DQMAX if the number of iterations is less than three and a 33.3% decrease in DQMAX if the number of iterations is more than four. All test cases have been run to approximately steady state conditions.

The first two test cases are based on a DeLaval nozzle running at subsonic conditions. The first test case (SSS02I, $0 \leq I \leq 2$) is a stationary solution with the flow having a Mach number of one at the throat. The second test case (MSS02I, $0 \leq I \leq 2$) uses the same initial conditions as the first case except the exit pressure is increased by 5% at $t=0.0$ seconds. Data from these runs, presented in Table 14, indicate that a significant reduction in computational effort is realized by the implementation of the optimization procedures. In the stationary solution the value of DQMAX is 0.001 for the first level optimization, and the same value is used initially in the second level optimization. On the first level optimization the total number of iteration required to reach the desired solution (see Table 14) was decreased by 61%. Moving to the second level optimization, the decrease in total iterations was 715 in

TABLE 14. Accuracy and Efficiency Summary, Converging-Diverging Nozzle, Smooth Solutions with Optimization.

Run	Time Steps	Total Iterations	Go Step CPU	Energy Norm $ q_\alpha^h _e$				
				Density	Momentum	Energy	Pressure	
SSS020	250	516	1:17.14	0.1911	0.4164	0.1503	0.3157	
SSS021	93	202	30.82	0.187	0.415	0.147	0.308	
SSS022	50	149	23.21	0.187	0.415	0.147	0.309	
MSS020	250	567	1:24.20	0.097	0.3702	0.074	0.167	
MSS021	113	351	53.94	0.093	0.367	0.071	0.163	
MSS022	104	330	49.94	0.0922	0.366	0.0701	0.161	
SSD010	250	509	1:16.27	0.2676	0.4248	0.2221	0.4045	
SSD011	54	130	20.44	0.2676	0.4248	0.2221	0.4045	
SSD012	32	96	15.30	0.268	0.425	0.222	0.4048	

the transient solution with $DQMAX=0.005$, the number of iterations was reduced by 38% on the first level of optimization.

Implementation of the second level of optimization brought the total reduction in the number of iterations to 42%. The third test case (SS001I, $0 \leq I \leq 2$) is a DeLaval nozzle running on design. The value of $DQMAX$ is 0.001 for the first level of optimization and the same value is used initially in the second level. The first level of optimization decreases the number of iterations by 74%. With the second level of optimization the decrease in iterations becomes 81%.

In all cases the accuracy, as measured in the energy norm, remains approximately the same; however, a slight decrease in accuracy and a decrease in the number of significant digits, as measured in the Richardson step, occurs as the number of time step decreases due to increases in the temporal truncation errors [Strang and Fix, 1973]. Differences in the significant digits in MSS021 and MSS022 seems to indicate a departure in the trend toward fewer digits of significance with fewer total iterations, see Table 14. Explanation of this behavior is found by observation of the consistency of the number of iterations at individual time steps throughout the solution. The more consistent solution, MSS022, maintains three or four step convergence throughout most of the solution. The less accurate solution, MSS021, has a more variable rate of convergence with two up to eight steps required to achieve the desired level of convergence as the solution progresses.

The inconsistency in the convergence rate under the first level of optimization should only occur in transient solutions. By

introducing a discontinuity in the initial conditions, the solution is required to change significantly over a short period of time indicating a need for small timesteps. As the solution progresses toward steady state, the solution undergoes less change over a given time interval and larger timesteps may be used. Thus, while optimized solutions in steady state offer a trade off between the digits of significance in the measures of accuracy and the computational effort required for a solution, the optimized transient solution achieves the more consistent solution with the fewer total iterations. However, the differences in the accuracy are, at least for the given solutions, esoteric, as the results of the most accurate solution (no optimization) and the least accurate solution (level 2 optimization), presented in Figures 8 and 9, exhibit no visually discernable differences.

Two additional test cases for the optimization procedure are based on supersonic off-design nozzle flow. The first test case (SSH03I, $0 \leq I \leq 2$) is a stationary shock occurring at Mach number 1.40 with the exit pressure set to maintain the shock at that level. The second test case (MSH03I, $0 \leq I \leq 2$) uses the same initial conditions as the first test case, except the exit pressure is increased by 5.8% forcing the shock back to a Mach number of 1.3. Accuracy and efficiency data from these cases are presented in Table 15.

Data presented in Table 15 indicates no positive results for the off design supersonic test cases. For the subsonic test case, the unoptimized solution (SSS030) required 564 iterations over

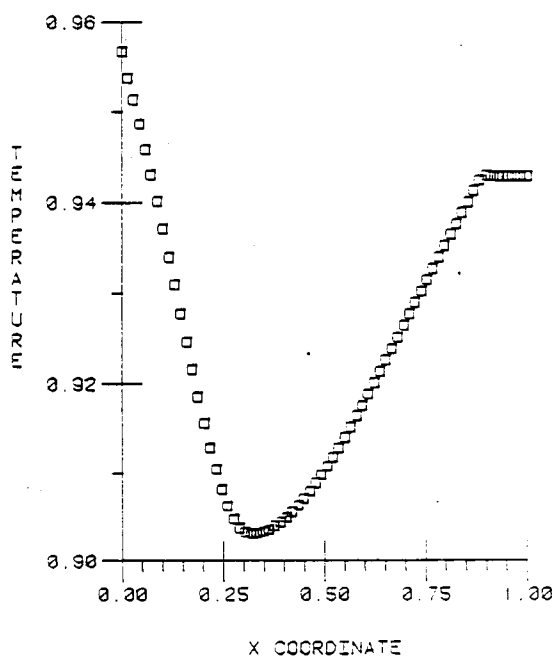
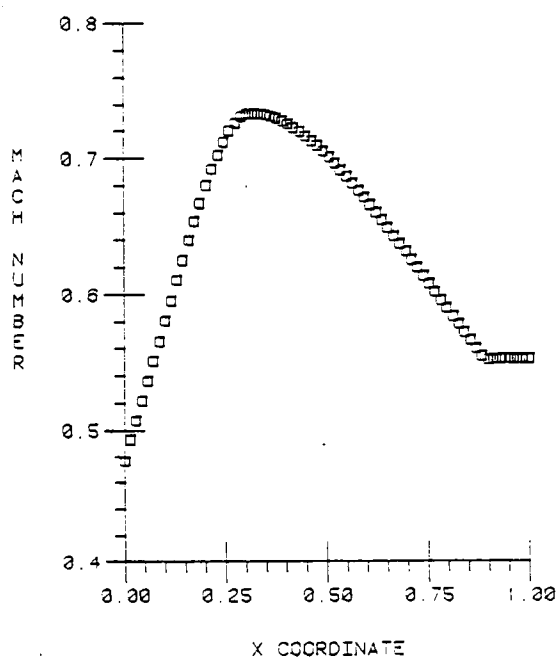
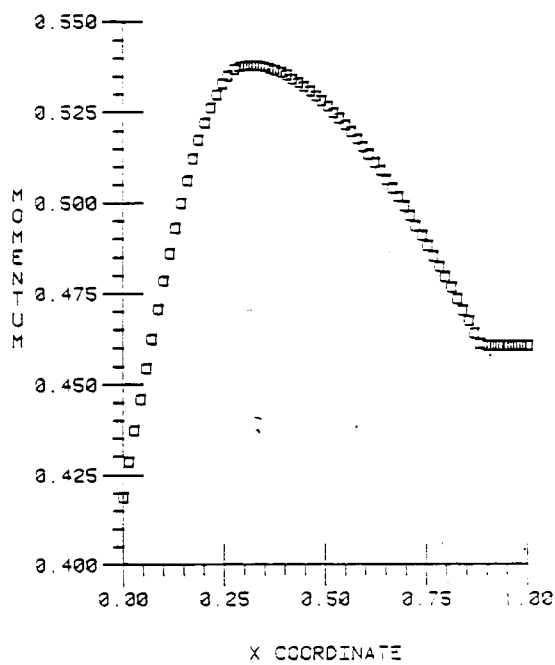
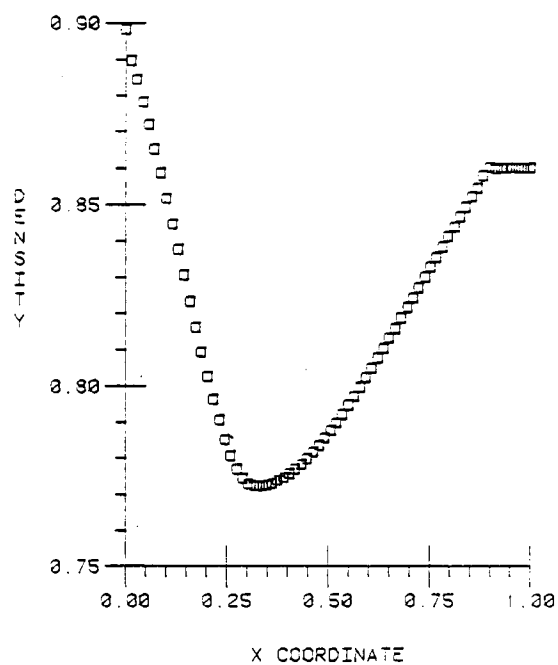


FIGURE 8. Subsonic Converging-Diverging Nozzle, Transient Solution, No Optimization.

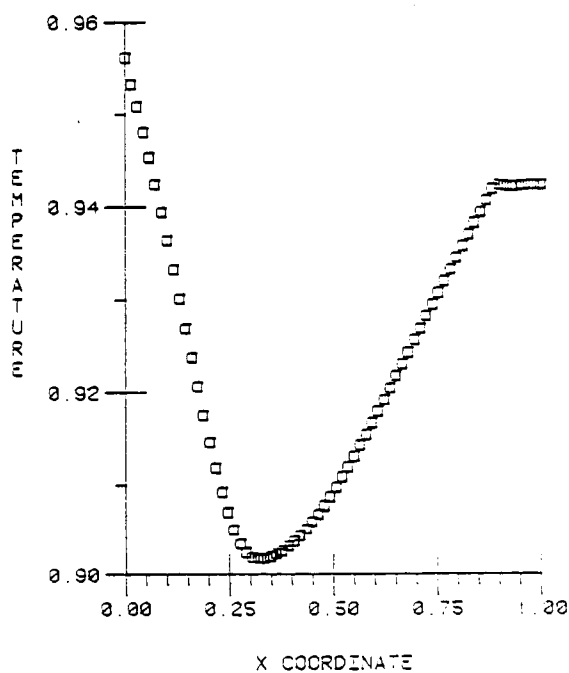
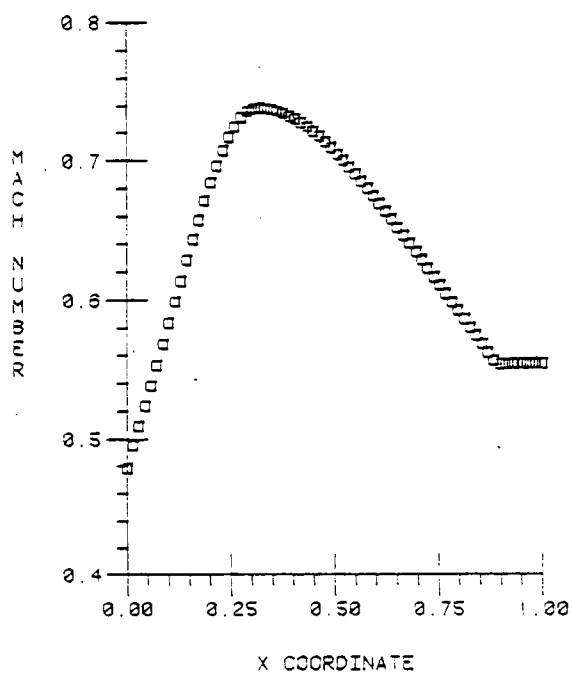
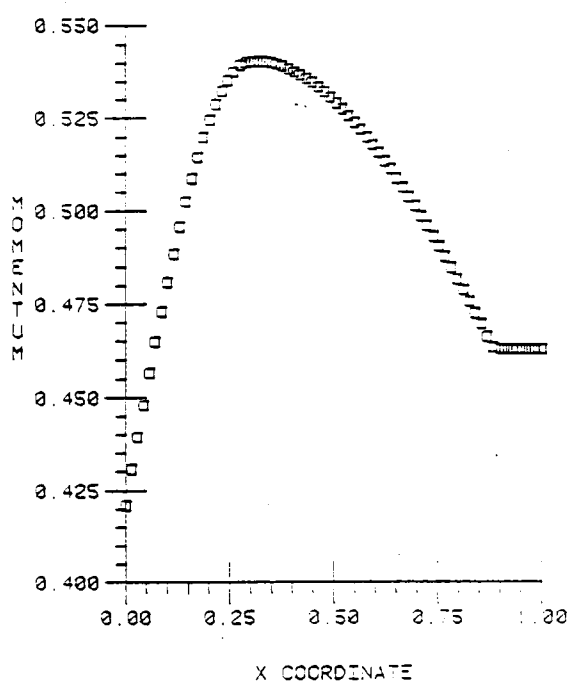
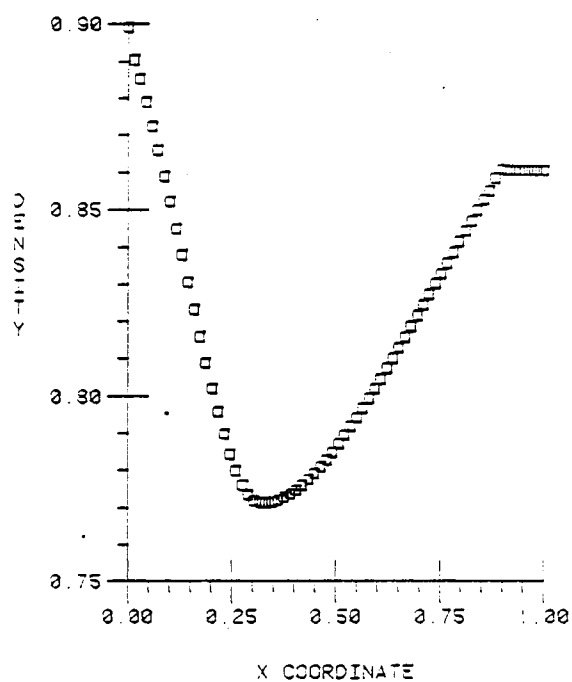


FIGURE 9. Subsonic Converging-Diverging Nozzle, Transient Solution, Second Level Optimization.

TABLE 15. Accuracy and Efficiency Summary, Converging-Diverging Nozzle, Shocked Solutions with Optimization.

Run	Time Steps	Total Iterations	Go Step CPU	Energy Norm $ q_{\alpha} ^h e$			
				Density	Momentum	Energy	Pressure
SSH030	250	564	1:23.86	1.9781	0.44703	1.7224	2.8382
SSH031	242	605	1:29.04	1.97(9) ^a	0.4470	1.72	2.84
SSH032	(Solution could not be obtained)						
MSH030	250	855	2:06.34	0.909	0.42776	0.773	1.314
MSH031	411	1312	3:12.46	0.91	0.4278	0.77	1.3
MSH032	220	866	2:08.72	0.90(4) ^a	0.4276	0.77	1.31

^aDigits have been affected by truncation error and are given for range only.

250 timesteps to reach an end time of 0.0125 seconds. Using the first level of optimization (SSS031), the solution required 605 iterations, an increase of 7.3%, even though a slight decrease, 3.2%, occurred in the number of time steps. The accuracy, as measured in the energy norms, remained unchanged except for a loss of one to two digits of significance in the optimized case. No solutions could be obtained from the second level of optimization. In the unoptimized run, the solution is essentially stabilized after 0.002 seconds which takes 40 time steps. The first level optimized run stabilizes at approximately the same time, but in run SSH031, 220 timesteps are required to reach 0.002 seconds. Graphical presentations of the stationary shock solutions were identical and results from SSS030 are plotted in Figure 10. The shock in Figure 10 is centered at the correct point ($x=0.73$) and is smeared over three nodes with some minor oscillations in the region upstream of the shock.

The unoptimized solution for the moving shock case (MSH030), required 250 time steps and 855 iterations. The first level of optimization (MSH031) produces a significant increase in computational effort with the number of timesteps increasing by 64% to 411 and the number of iterations increased by 53% to 1312. The second level of optimization (MSH032) reduced the number of timesteps by 12% from the unoptimized run to 220, but the number of iterations increased slightly to 866. The energy norms for the three solutions remain nominally the same except for changes in the number of significant digits. No discernable differences occur in the graphical presentations and results from MSH030 are plotted in

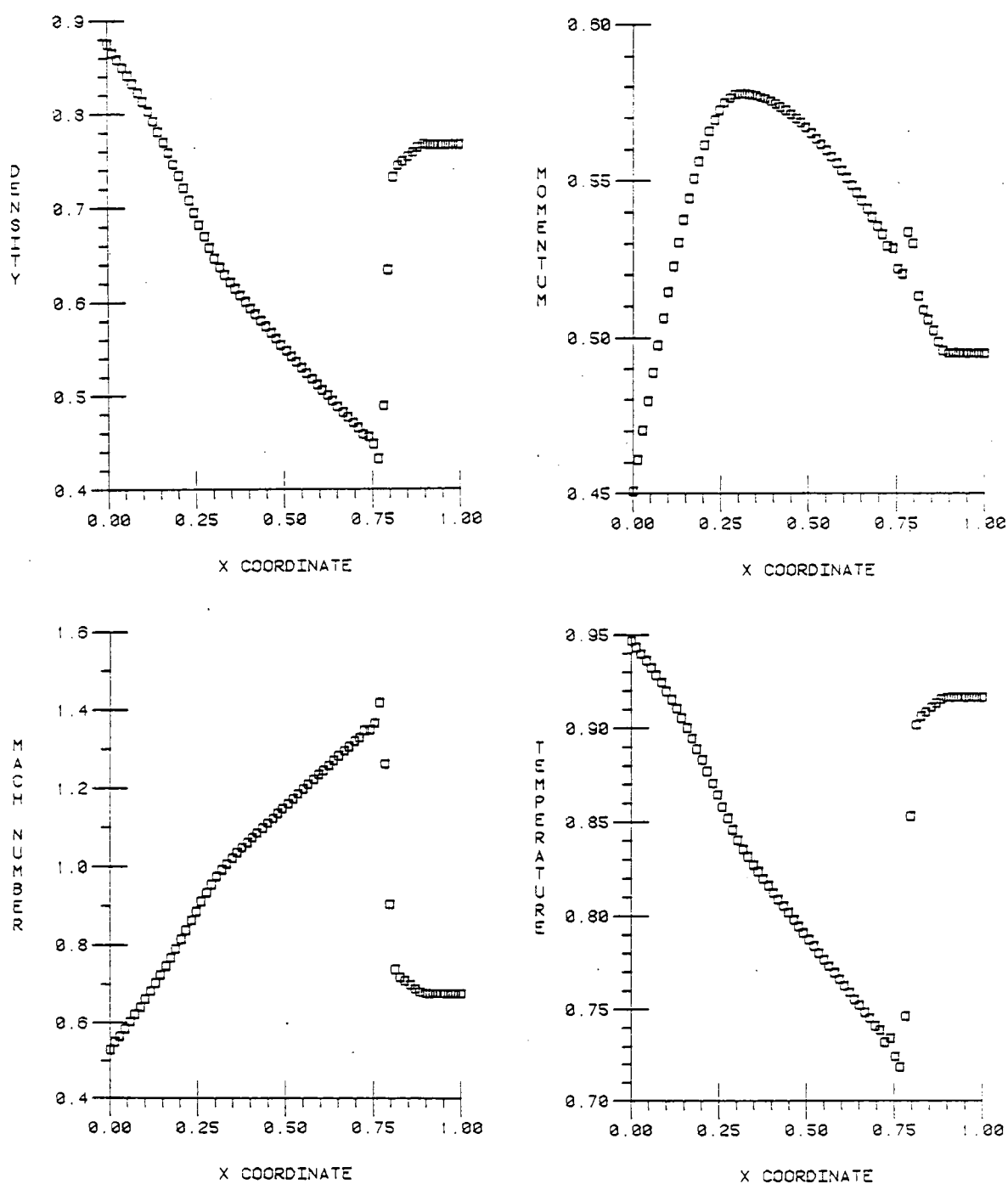


FIGURE 10. Off Design Transonic Converging-Diverging Nozzle, Stationary Solution, Nodes 30-101.

Figure 11. The maximum Mach number for this solution should be 1.3 with the shock centered at $x=0.63$. In Figure 11, the shock is centered correctly but is smeared over four element with a maximum Mach number of 1.24. The solution could probably be sharpened with adjustment of the penalty constraint parameters.

Several characteristics may be observed in the trends in the convergence histories of the individual timesteps. In the unoptimized solution the initial timesteps require a relatively large number of iterations with the number of iterations per timestep decreasing as the solution progresses toward steady state, thus a major controlling factor in selection of a fixed size timestep is the convergence of the first timestep. Using the first level of optimization the change in the number of timesteps follows a pattern exactly opposite of that in the unoptimized solution. With the maximum value of $\{DQI\}$ fixed (DQMAX) the number of iterations per timestep increases as the solution proceeds. The size of the timestep also increases with increasing time. Two interconnected problems are encountered in trying to use the first level optimizer. First, as the stepsize increases and the solution advances toward steady state the fixed value of DQMAX must be small because little change is occurring in the solution, and, at the large time steps, the collected data indicates a decreased convergence rate. Second, as the fixed value of DQMAX is decreased to accommodate conditions near steady state, the initial time steps must become very small, and the acceleration of the latter stages of the solution is negated by the excessive number of timesteps required for the initial stages

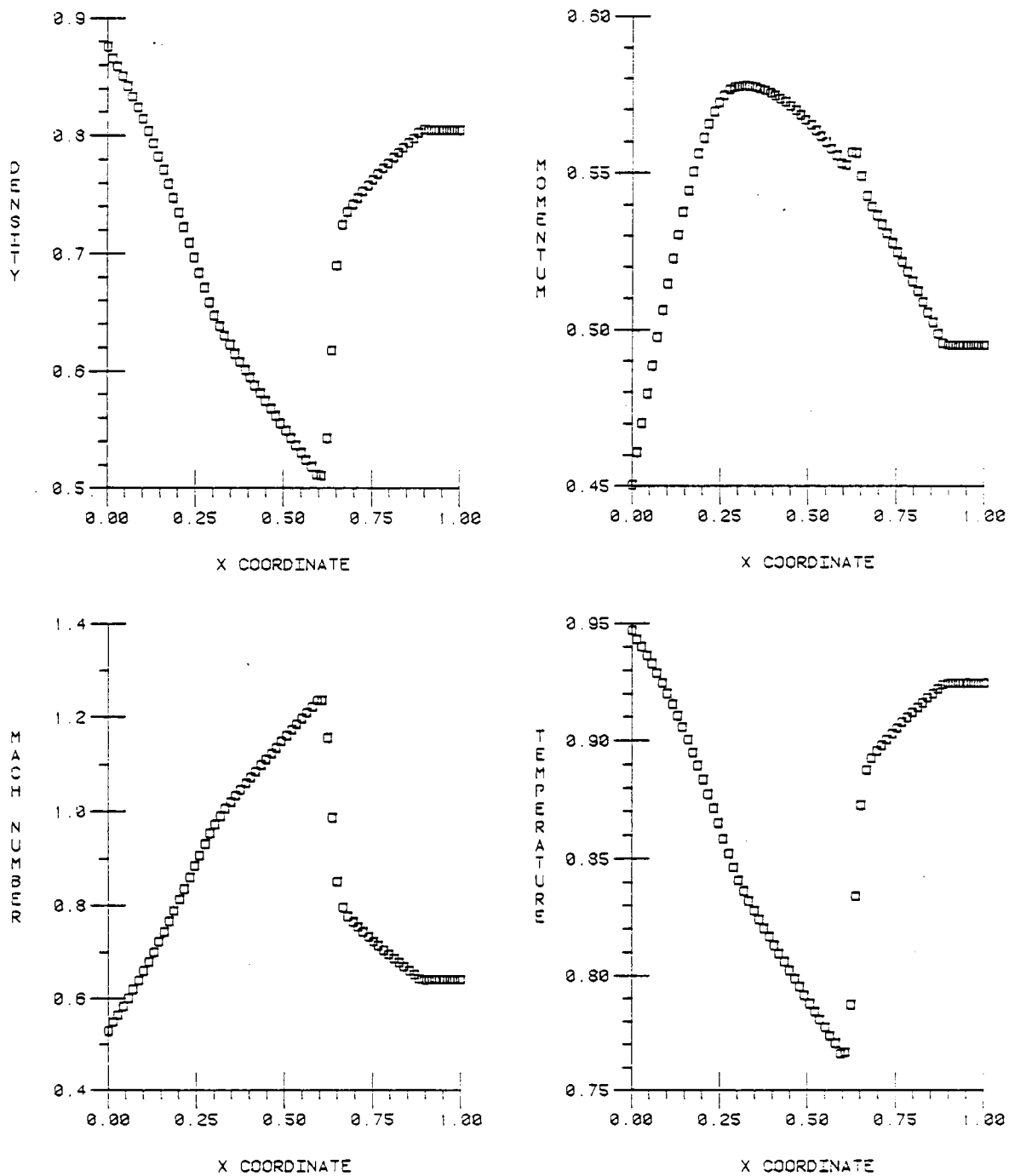


FIGURE 11. Off Design Transonic Converging-Diverging Nozzle, Transient Solution, Nodes 30-101.

of the solution. The second level of optimization has a framework which would appear to counteract the problems encountered in both the unoptimized case and the first level of optimization by adjusting the size of DQMAX and, thus, the size of the time steps. Based on the moving shock run (MSH032) the basic idea of the second level optimization appears to work with Courant numbers in excess of 10,000 being attained. In MSH032 a three step convergence is maintained throughout a major portion of the solution. As the solution nears steady state the number of iterations starts to increase and this causes DQMAX to be decreased controlling the stepsize and preserving convergence. The number of iterations per time step increases at a slow steady rate in MSH032 and the second level optimized works well; however, the stationary shock solution (SSH032) undergoes a rapid increase in the iterations per timestep, the value of DQMAX is not allowed to adjust, and convergence is lost.

From the presented data, the idea of an optimization scheme based on these concepts holds some promise. However, the scheme analyzed requires a priori knowledge of the solution behavior as well as experimentation with the input variables controlling the optimizer in order to obtain a solution. The solutions presented have offered savings in computation effort only for smooth solutions. The lack of applicability of the given optimization scheme indicates a need for development of a more exact relationship between timestep size and DQMAX before the idea can be implemented in generality.

CHAPTER VIII

SUMMARY AND CONCLUSIONS

Herein are presented a summarization of the major findings and conclusions of this analysis along with suggestions for further research.

1. An implicit finite element algorithm for solution of the quasi-one dimension Euler equations was developed by A.J. Baker and a computer code for verification of the utility of this algorithm was developed by M.O. Soliman. Although accurate solutions were obtained, the Newton matrix iteration method provided only monotonic convergence, while quadratic convergence is theoretically predicted.
2. After a step by step derivation of the algorithm, several errors were discovered in the original presentation of the terms forming the Jacobian. Also, examination of the original computer code exposed several terms which were omitted from the coding.
3. Taken individually, none of the revisions to the algorithm and code exert a significant influence on the overall convergence rate of the equation system. When the changes are applicable to a specific independent variable, the impact of the change is seen primarily on that variable. Changes to a dependent variable cause only minor changes in

convergence, but the changes are distributed to all independent variables.

4. Taken collectively, the revisions give a quadratic convergence rate for the system of equations and the efficiency of the algorithm is increased by 24%. This indicates that the overall accuracy of the Jacobian is of greater importance to the convergence than the accuracy of any particular term.
5. The accuracy of the solutions obtained from the fully revised formulations is equivalent to that obtained from the original formulation. Also, behavior of the accuracy and efficiency of the revised formulation with respect to changes in grid refinement and Courant number is unchanged.
6. No interpretable information has been obtained from the H^0 and L^1 solution norms.
7. The conservative form of the penalty constraint term yields unstable solutions.
8. Stable subsonic solutions for a converging-diverging nozzle under both steady state and transient solution were obtained. The subsonic solutions require an approximation to a reservoir at the inlet, with Dirichlet boundary conditions on density, energy, and pressure at the inlet and on pressure at the outlet. All other variables have Neumann boundary conditions.

9. Stable transonic solutions for a DeLaval nozzle were obtained for on-design conditions and both steady state and transient off-design conditions. Boundary conditions for transonic flows are Dirichlet on all variables at the inlet and Dirichlet on pressure at the outlet for off-design conditions. All other variables have Neumann boundary conditions.
10. Using optimized values of v_a^2 from the Riemann shock tube test case produces significant smearing of shocks and deterioration of peak values in solutions for the converging-diverging nozzle. This indicates a need for adjustment to the penalty constraints to the lower level of diffusion.
11. The timestep optimization procedure produces promising results when applied to smooth (subsonic and on design transonic) solutions, with Courant numbers in excess of 10,000 occurring under certain near steady state conditions.
12. When dealing with highly nonsmooth (off design transonic) solutions, implementation of the timestep optimization procedure is difficult, or in certain cases, impossible, and poor results are obtained from the cases in which convergence is maintained. Close examination of the cases which lose convergence show that the optimizer is behaving properly, but it cannot respond sufficiently to maintain convergence.

13. The basic framework of the timestep optimization procedure appears to be sound, but the method used for variation of the timesteps does not model the actual situation adequately.
14. Results of this analysis indicate a need for further studies in the following areas:
 - a. The revisions studied in this analysis should be projected into the multidimension algorithm from which the quasi-one dimensional algorithm was derived.
 - b. The penalty constraint parameters should be reevaluated to determine optimum values for the revised formulation. Also, additional studies should be made of the mechanism by which the penalty constraint operates.
 - c. A more exact model of the relationship between the delta values from the first Newton iteration and the size of the timestep should be developed for implementation in the timestep optimizer.
 - d. The fully revised formulation should be projected onto an adaptive grid.
 - e. The applicability of the H^0 and L^1 solution norm as a measure of accuracy should be determined.

LIST OF REFERENCES

LIST OF REFERENCES

- J.E. Akin, Application and Implementation of Finite Element Methods, Academic Press, New York, 1982.
- A.J. Baker. "Research on a Finite Element Numerical Algorithm for the Three Dimensional Navier Stokes Equations," USAF Report AFWAL-TR-82-3012, 1982.
- A.J. Baker, "A Finite Element Algorithm for High Speed Flow," Chapter 8 in Finite Elements in Fluids, Vol. 5, John Wiley, New York, 1984.
- A.J. Baker, Finite Element Computational Fluid Mechanics, McGraw-Hill/Hemisphere, New York, 1983.
- A.J. Baker, "Finite Element Analysis in Computational Fluid Dynamics," AIAA-83-1918CP, 1983.
- A.J. Baker and M.O. Soliman, "On the Accuracy and Efficiency of a Finite Element Algorithm for Hydrodynamic Flows," Presented at 4th International Conference on Finite Elements in Water Resources, 1982.
- E.B. Becker, G.F. Carey, and J.T. Oden, Finite Elements: An Introduction Volume I, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1981.
- G. Birkoff, "Numerical Fluid Dynamics," SIAM Review, Vol. 25 No. 1, pp. 1-34, 1983.
- J.F. Botha and G.F. Pinder, Fundamental Concepts in the Numerical Solution of Differential Equations, John Wiley & Sons, New York, 1983.
- R.L. Burden, J.D. Faires, and A.C. Reynolds, Numerical Analysis, 2nd ed., Prindle, Weber, & Schmidt, Boston, 1981.
- H. Chase, "A Note on the Convergence of Newton's Method," Applicable Analysis, Vol. 14, pp. 55-60, 1982.
- D.W. Decker, H.B. Keller, and C.T. Kelley, "Convergence Rates for Newton's Method at Singular Points," SIAM J. Numerical Analysis, Vol. 20, No. 2, pp. 55-60, 1982.
- C.S. Desai and J.F. Abel, Introduction to the Finite Element Method, Van Nostrand Reinhold Company, New York, 1972.

- E. Kreyszig, Advanced Engineering Mathematics, 3rd ed., John Wiley and Sons, Inc., New York, 1972.
- J.T. Oden, and J.N. Reddy, Mathematical Theory of Finite Elements, John Wiley and Sons, New York, 1976.
- P.J. Roache, Computation Fluid Dynamics, Hermosa Publishers, Albuquerque, NM, 1972.
- D.H. Rudy and J.C. Strikwerda, "Boundary Conditions for Subsonic Compressible Navier Stoke Calculations," Computers and Fluids, Vol. 9, pp. 327-338, 1981
- S. Schreier, Compressible Flow, John Wiley and Sons, New York, 1982.
- A.H. Shapiro, The Dynamics and Thermodynamics of Compressible Fluid Flow, Volume I, The Roland Press Company, New York, 1953.
- G.A. Sod, "A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws," Journal of Computational Physics, Vol. 27, pp. 1-31, 1978.
- G. Strang and G.J. Fix, An Analysis of the Finite Element Method, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973.
- V.L. Streeter, Fluid Mechanics, Fifth Ed., McGraw-Hill Book Company, New York, 1971.
- Tai-Pang Liu, "Transonic Gas Flow in A Duct of Varying Area," Archive for Ration Mechanics and Analysis, Vol. 23, No. 8, pp. 1-18, 1982.
- G.B. Thomas, Jr., Calculus and Analytic Geometry, Alternate Ed., Addison-Wesley Publishing Company, Inc., Philippines, 1972.
- R.S. Varga, Matrix Iterative Analysis, Prentice-Hall, New Jersey, 1962.
- F.M. White, Viscous Fluid Flow, McGraw-Hill Book Company, New York, 1974.
- F.M. White, Fluid Mechanics, McGraw-Hill Book Company, New York, 1979.

APPENDICES

APPENDIX A

NOMENCLATURE

a	boundary condition coefficient; sound speed
A	cross-sectional area; one-dimensional hypermatrix prefix
C	coefficient; matrix; Courant number
e	specific total energy
f	function of known argument
F	finite element matrix; discrete solution equation system
g	total energy
G	finite element matrix
h	element measure
H	Hilbert space
i	$\sqrt{-1}$; index
j	index
J	Jacobian matrix
k	cardinal basis degree
ℓ	summation index; differential operator
L	differential operator
m	momentum
M	number of finite elements spanning R^n
n	unit normal vector; dimension of space
N	finite element cardinal basis; discrete index
p	pressure; iteration index
q	generalized dependent variable
Q	generalized discrete dependent variable
R^n	spatial domain of differential operator
Re	Reynolds number
S_e	finite element assembly operator

t	time
u_i	velocity vector
u	convection velocity
x_i	Cartesian coordinate system
α	dependent variable reference
$\bar{\beta}_i$	Lagrange multiplier set
γ	ratio of specific heats; parameter
∂	partial derivative operator
∂R	boundary of solution domain R^n
δ	Kronecker delta; parameter
δQ	iteration variable
Δ	mesh measure; increment
ϵ	specific internal energy; isotropic dissipation function; parameter
ν	dissipation parameter
ρ	density
Σ	summation operator
Ω	solution domain

Superscripts:

h	solution semi-discrete approximation
o	initial condition reference, stagnation conditions
p	iteration index
T	matrix transpose
$1,2$	dissipation parameter reference
$\cdot$	ordinary derivative
$*$	nondimensional variable

Subscripts:

e	element reference
i,j,k	tensor indices
j	time step index
k	degree of polynomial
o	reference state
α	dependent variable index

Notation:

{ }	column matrix
[]	square matrix
$\cup$	union
$\cap$	intersection
ε	belongs to
$\ \quad \ $	norm

APPENDIX B

DERIVATION OF MATRIX FORM OF F

The following is a step by step derivation of the matrix form of the nonlinear algebraic equation set formed by substitution of the discrete approximation of equations (5)-(8) into the theoretical form of the finite element algorithm, equation (15).

Continuity Equation

The continuity equation is given in the form

$$L(\rho) = \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \rho u A' = 0 \quad (1B)$$

where $A' = \frac{d(\ln A)}{dx}$. The form of the continuity equation remains unchanged after nondimensionalization. The element semidiscrete approximation of the continuity equation is obtained by making the substitutions:

$$\rho_e(x,t) \equiv \{N_k(x)\}^T \{R(t)\}_e, \quad u_e(x,t) \equiv \{N_k(x)\}^T \{V(t)\}_e, \quad \text{and} \\ A_e(x) \equiv \{N_k(x)\}^T \{A\}_e, \quad \text{and is of the form}$$

$$L(\rho^h) = S_e \left[\frac{\partial}{\partial t} \{N_k\}^T \{R\}_e + \frac{\partial}{\partial x} (\{N_k\}^T \{R\}_e \{N_k\}^T \{V\}_e) + \right. \\ \left. \{N_k\}^T \{R\}_e \{N_k\}^T \{V\}_e \frac{d}{dx} \{N_k\}^T \{A\}_e \right] \quad (2B)$$

Identification of the independent variables has been dropped for convenience. Equation (2B) is then substituted into the first constraint of equation (15) yielding

$$\begin{aligned}
\int_{R^1} \{N_k\} L(\rho_e) dx &= \int_{R_e} \{N_k\} \left(\frac{\partial}{\partial t} \{N_k\}^T \{R\}_e \right. \\
&+ \frac{\partial}{\partial x} \{N_k\}^T \{R\}_e \{N_k\}^T \{V\}_e \\
&+ \{N_k\}^T \{R\}_e \{N_k\}^T \{V\}_e \frac{d}{dx} \{N_k\}^T \{A\} \left. \right) dx \quad (3B)
\end{aligned}$$

The following algebraic manipulations are performed on equation (3B).

$$\begin{aligned}
\int_{R^1} \{N_k\} L(\rho_e) dx &= \int_{R_e^1} \{N_k\} \left(\{N_k\}^T \frac{d}{dt} \{R\}_e \right) dx \\
&+ \int_{R_e^1} \{N_k\} \left(\{V\}^T \frac{d}{dx} (\{N_k\} \{N_k\}^T) \{R\} \right) dx
\end{aligned}$$

$$\text{with } \{V\}^T \{N_k\} \{N_k\}^T \{R\} = \{N_k\}^T \{M\}$$

$$\begin{aligned}
&= \int_{R_e^1} \{N_k\} \{N_k\}^T dx \{R\}_e + \int_{R_e^1} \{N_k\} \frac{d}{dx} \{N_k\}^T dx \{M\} \\
&+ \{A\}^T \int_{R_e^1} \left(\frac{d}{dx} \{N_k\} \right) \{N_k\}^T \{N_k\} \{N_k\}^T dx \{V\}_e \{R\}_e \quad (4B)
\end{aligned}$$

implementing the Greene-Gauss form of the divergence theorem

[Kreyszig, 1972] on the second term of (4B) gives

$$\begin{aligned}
&= \int_{R_e^1} \{N_k\} \{N_k\}^T dx \{R\}'_e + \left(\oint_{R_e^1} \frac{d}{dx} (\{N_k\} \cdot \{N_k\}^T) \cdot \hat{n} dx \right. \\
&\quad \left. - \int_{R_e^1} \left(\frac{d}{dx} \{N_k\} \right) \{N_k\} dx \right) \{M\}_e \\
&\quad + \{A\}^T \int_{R_e^1} \left(\frac{d}{dx} \{N_k\} \right) \{N_k\}^T \{N_k\} \{N_k\}^T dx \{V\}_e \{R\}_e \quad (5B)
\end{aligned}$$

Using Baker's hypermatrix nomenclature, equation (5B) reduces to

$$\begin{aligned}
\int_{R_e^1} \{N_k\} L(\rho e) dx &= h_e [A200] \{R\}'_e + \left[\oint_{R_e^1} \frac{d}{dx} (\{N_k\} \cdot \{N_k\}^T) \cdot \hat{n} dx \right. \\
&\quad \left. - [A210] \right] \{M\}_e + \{A\}^T ([A41000] \{V\}) \{R\} \quad (6B)
\end{aligned}$$

The second term in equation (6B) is a contour integral around a given element as shown in Figure A1. The integral can be written in the form

$$\begin{aligned}
\int_{R_e^1} \frac{d}{dx} (\{N_k\} \cdot \{N_k\}^T) \cdot n \, dx &= - \int_1^2 \frac{d}{dx} (\{N_k\} \{N_k\}^T) dx \\
&\quad + \int_2^3 \frac{d}{dx} (\{N_k\} \{N_k\}^T) dx + \int_3^4 \frac{d}{dx} (\{N_k\} \{N_k\}^T) dx \\
&\quad - \int_4^1 \frac{d}{dx} (\{N_k\} \{N_k\}^T) dx \quad (7B)
\end{aligned}$$

The definition of the problem as quasi-one dimensional dictates that no changes occur in the y-direction; therefore, the integrals 1-2

and 3-4 are equal, i.e. $\int_1^2 \frac{d}{dx}(\{N_k\}\{N_k\}^T)dx = \int_3^4 \frac{d}{dx}(\{N_k\}\{N_k\}^T)dx$, and equation (7A) thus reduces to

$$\oint_{R_e} \frac{d}{dx}(\{N_k\}\{N_k\}^T) \cdot \hat{n} dx = - \int_4^1 \frac{d}{dx}(\{N_k\}\{N_k\}^T) dx + \int_2^3 \frac{d}{dx}(\{N_k\}\{N_k\}^T) dx \quad (8B)$$

This equation indicates that the integral is evaluated only at the end points of the element. With the cardinal basis functions of degree 1 and 2 defined as,

$$\{N_1(x)\} = \begin{Bmatrix} 1-x/h_e \\ x/h_e \end{Bmatrix}, \quad \{N_2(x)\} = \begin{Bmatrix} (1-x/h_e)(1-2x/h_e) \\ 4x/h_e(x/h_e) \\ x/h_e(2x/h_e-1) \end{Bmatrix} \quad (9B)$$

evaluation the first integral at $x=0$ and the second integral at $x=h_e$ yields for $k=1$

$$\begin{aligned} \oint_{R_e} \frac{d}{dx}(\{N_1\}\{N_1\}^T) \cdot \hat{n} dx &= - \int_0 \frac{d}{dx} \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \{1,0\} dx \\ &+ \int_1 \frac{d}{dx} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \{0,1\} dx \\ &= - \int_0 \frac{d}{dx} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} dx + \int_1 \frac{d}{dx} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} dx \end{aligned}$$

$$\begin{aligned}
 &= - \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \equiv [A20]_1
 \end{aligned} \tag{10B-a}$$

A similar substitution using $k=2$ yields

$$\int_{R_e} \frac{d}{dx} (\{N_2\} \{N_2\}) \cdot \hat{n} dx = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \equiv [A20]_2 \tag{10B-b}$$

Substitutions (10B) into (6B) gives the symbolic form

$$\begin{aligned}
 \int_{R_e} \{N_k\} L(\rho_e) dx &= h_e [A200] \{R\}_e + \left[[A20] - [A210] \right] \{M\}_e \\
 &+ \{A\}_e^T ([A41000] \{V\}) \{R\}
 \end{aligned} \tag{11B}$$

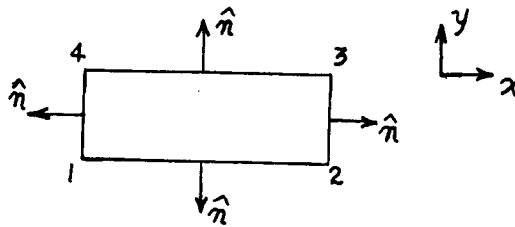


Figure A1

Evaluation of the second constraint of equation (15) requires the use of the nonconservative form of the convective derivative of the density defined as

$$L^c(\rho) = \frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} \tag{12B}$$

The elemental semi-discrete approximation of which is

$$L^c(\rho_e) = \frac{\partial}{\partial t} \{N_k\}^T \{R\}_e + \{N_k\}^T \{V\} \frac{\partial}{\partial t} \{N_k\}^T \{R\} \quad (13B)$$

Defining the dissipation parameter β_α as

$$\bar{\beta}_\rho \equiv h_e (v_R^1 \delta t + \text{sign } u \ v_R^2 \delta x) i \quad (14B)$$

and substituting (13B) and (14B) into the second constraint of equation (15) and simplifying yields.

$$\begin{aligned} \bar{\beta}_\rho \int_{R'_e} \nabla \{N_k\} L^c(\rho_e) dx &= h_e \ v_R^1 \int_{R_1} e \frac{d}{dx} \{N_k\} \frac{\partial}{\partial t} \{N_k\}^T \{R\}_e dx \\ &+ h_e \ v_R^2 \int_{R'_e} \frac{d}{dx} \{N_k\} \{N_k\}^T \{\bar{V}\} \frac{\partial}{\partial x} \{N_k\}^T \{R\} dx \\ &= h_e \ v_R^1 \int_{R'_e} \left(\frac{d}{dx} \{N_k\} \right) \{N_k\}^T dx \{R\}'_e \\ &+ h_e \ v_R^2 \{\bar{V}\}^T \int_{R'_e} \{N_k\} \frac{d}{dx} \{N_k\} \frac{d}{dx} \{N_k\}^T dx \{R\} \\ \bar{\beta}_\rho \int_{R'_e} \nabla \{N_k\} L^c(\rho_e) dx &= h_e \ v_R^1 [A210] \{R\}'_e \\ &+ v_R^2 \{\bar{V}\} [A3011] \{R\}_e \end{aligned} \quad (15B)$$

The summation of equations (11B) and (15B), using the summation in a Taylor series and using the trapezoid rule for integration yields the final form of the term $\{FR\}_e$ as

$$\begin{aligned}
\{R\}_e &= h_e \left[[A200] + v_R^1 [A210] \right] \{R\}_{j+1} \\
&+ \frac{\Delta t}{2} \left[\left[[A20] - [A210] \right] \{M\} + v_R^2 \{\bar{V}\} [A3011] \{R\} \right. \\
&\left. + \{A\}^T ([A41000] \{V\}) \{R\} \right]_{j+1,j} = \{0\} \quad (16B)
\end{aligned}$$

A reduction in computational effort can be achieved by the summation of frequently used matrix pairs prior to the assembly operation. One such pair exist in equation (16B). The matrix pair, $[A20] - [A210]$, has two possible steps which may be used to reduce the computation time. Firstly, the matrix $[A20]$ is of a form which will allow cancellation of all terms except those contributed by the terminal nodes of the discretized region. This allows a redefinition of the $[A20]$ matrix in the form

$$[A20]_1 \equiv \begin{bmatrix} -\delta_{ei} & 0 \\ 0 & \delta_{em} \end{bmatrix}, \quad [A20]_2 \equiv \begin{bmatrix} -\delta_{ei} & 0 & 0 \\ \delta_{ei} & 0 & 0 \\ 0 & 0 & \delta_{em} \end{bmatrix} \quad (17B)$$

where δ_{ei} is the Kronecker delta and e is the number of the element such that $1 \leq e \leq M$. Secondly, the summation of the matrix pair can be defined as

$$[D2MAT] = [A20] - [A210] \quad (18B)$$

Substituting equation (18B) into (16B) gives a symbolic representation of $\{FR\}_e$ in the form

$$\begin{aligned}
\{FR\}_e &= h_e \left[[A200] + v_R^1 [A210] \right] \{R\}_{j+1} \\
&+ \frac{\Delta t}{2} \left[[D2MAT] \{M\} + v_R^2 \{\bar{V}\}^T [A3011] \{R\} \right. \\
&\left. + \{A\}^T ([A41000] \{V\}) \{R\} \right]_{j+1,j} = \{0\}
\end{aligned} \tag{19B}$$

Momentum Equation

The momentum equation is given in the form

$$L(\rho u) = \frac{\partial}{\partial t} \rho u + \frac{\partial}{\partial x} [u \rho u + p] + \rho u^2 A \tag{20B}$$

This equation is nondimensionalized with reference to the initial stagnation conditions by making the following substitutions

$$\begin{aligned}
t^* &= t \frac{a_0}{L}, \quad x^* = \frac{x}{L}, \quad \rho^* = \frac{\rho}{\rho_0}, \quad u^* = \frac{u}{a_0}, \quad A^* = \frac{A}{A_r}, \\
\frac{\partial}{\partial t^*} &= \frac{L}{a_0} \frac{\partial}{\partial t}, \quad \frac{\partial}{\partial x^*} = L \frac{\partial}{\partial x}, \quad p^* = \frac{p}{p_0}, \quad e^* = \frac{e}{e_0}
\end{aligned} \tag{21B}$$

The nondimensional form of the momentum equation is

$$L(\rho^* u^*) = \frac{\partial \rho^* u^*}{\partial t^*} + \frac{\partial u^* \rho^* u^*}{\partial x^*} + \frac{p_0}{\rho_0 a_0^2} \frac{\partial p^*}{\partial x^*} + \rho^* u^{*2} A^* \tag{22B}$$

where $p_0/\rho_0 a_0^2$ is the Euler number. Given that $a_0^2 = \gamma p_0/\rho_0$, the Euler number reduces to $1/\gamma$. Upon substitution of the reduced form of the Euler number and dropping the superscript asterisk, equation (22B) is left in the form

$$L(\rho u) = \frac{\partial \rho u}{\partial t} + \frac{\partial}{\partial x} \left[u \rho u + \frac{1}{\gamma} p \right] + \rho u^2 A \quad (23B)$$

defining $m \equiv \rho u$ yields

$$L(m) = \frac{\partial m}{\partial t} + \frac{\partial}{\partial x} \left[u m + \frac{1}{\gamma} p \right] + u m A \quad (24B)$$

The elemental semidiscrete approximation of equation (24B) is

$$\begin{aligned} L(m_e) = & \frac{\partial}{\partial t} \{N_k\}^T \{M\}_e + \frac{\partial}{\partial x} \left[\{N_k\}^T \{V\}_e \{N_k\}^T \{M\}_e + \frac{1}{\gamma} \{N_k\}^T \{P\}_e \right] \\ & + \{N_k\}^T \{V\}_e \{N_k\}^T \{M\}_e \frac{d}{dx} \{N_k\}^T \{A\}_e \end{aligned} \quad (25B)$$

Substitution of equation (25A) into the first constraint of equation (15) and simplifying gives

$$\begin{aligned} \int_{R_e} \{N_k\} L(m_e) dx = & \int_{R_e} \{N_k\} \left(\frac{\partial}{\partial t} \{N_k\}^T \{M\}_e \right. \\ & + \frac{\partial}{\partial x} \left(\{V\}_e^T \{N_k\} \{N_k\}^T \{M\}_e + \frac{1}{\gamma} \{N_k\}^T \{P\}_e \right) \\ & \left. + \left(\frac{d}{dx} \{N_k\}^T \{A\}_e \right) \{N_k\}^T \{V\}_e \{N_k\}^T \{M\}_e \right) dx \end{aligned}$$

$$\begin{aligned}
&= \int_{R_e'} \{N_k\} \{N_k\}^T dx \{M\}_e' \\
&\quad + \int_{R_e'} \{N_k\} \{V\}^T \frac{d}{dx} (\{N_k\} \{N_k\}) \{M\}_e dx \\
&\quad + \frac{1}{Y} \int_{R_e'} \{N_k\} \frac{d}{dx} \{N_k\}^T dx \{P\} \\
&\quad + \{A\}_e^T \int_{R_e'} \left(\frac{d}{dx} \{N_k\} \right) \{N_k\} \{N_k\}^T dx \{V\}_e \{M\}_e \\
&= h_e [A200] \{M\}_e' + \{V\}^T \left(\oint \frac{d}{dx} (\{N_k\} \{N_k\} \{N_k\}^T) \cdot \hat{n} dx \right. \\
&\quad \left. - \int_{R_e'} \{N_k\} \left(\frac{d}{dx} \{N_k\} \right) \{N_k\} dx \right) \{M\}_e \\
&\quad + \left(\oint \frac{d}{dx} (\{N_k\} \{N_k\}) \cdot \hat{n} dx \right. \\
&\quad \left. - \int_{R_e'} \left(\frac{d}{dx} \{N_k\} \right) \{N_k\}^T dx \right) \{P\} \\
&\quad + \{A\}_e^T ([A41000] \{V\}_e) \{M\}_e \\
&= h_e [A200] \{M\}_e' + \{V\}^T \left(\oint \frac{d}{dx} (\{N_k\} \{N_k\} \{N_k\}^T) \cdot \hat{n} dx \right. \\
&\quad \left. - [A3010] \right) \{M\}_e + \left[[A20] - [A210] \right] \{P\}_e \\
&\quad + \{A\}_e^T ([A41000] \{V\}_e) \{M\}_e \tag{26B}
\end{aligned}$$

Using the same arguments presented for the contour integral in the continuity equation (7B), the second term of equation (26B) can be evaluated for $k=1$ in the form

$$\begin{aligned}
 \oint \frac{d}{dx} (\{N_1\} \{N_1\} \{N_1\}^T) \cdot \hat{n} dx &= - \int_0 \frac{d}{dx} \left(\begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}^T \right) dx \\
 &+ \int_1 \frac{d}{dx} \left(\begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}^T \right) dx \\
 &= - \int_0 \frac{d}{dx} \left[\begin{Bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \right] dx \\
 &+ \int_1 \frac{d}{dx} \left[\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{Bmatrix} \right] dx \\
 &= \left[\begin{Bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \right] + \left[\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{Bmatrix} \right] \\
 \oint \frac{d}{dx} (\{N_1\} \{N_1\} \{N_1\}^T) \cdot \hat{n} dx &= \left[\begin{Bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{Bmatrix} \right] = [A30]_1 \quad (27B-a)
 \end{aligned}$$

Similarly for $k=2$

$$\oint \frac{d}{dx} (\{N_2\} \{N_2\} \{N_2\}^T) \cdot \hat{n} dx = \left[\begin{Bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{Bmatrix} \right] = [A30]_2 \quad (27B-b)$$

Substituting (27B) into (26B) gives

$$\begin{aligned} \int_{R_1^e} \{N_k\} L(m_e) dx &= h_e [A200] \{M\}_e' + \{V\}_e^T \left[[A30] - [A3010] \right] \{M\}_e \\ &+ \left[[A20] - [A210] \right] \{P\}_e + \{A\}_e^T ([A41000] \{V\}_e) \{M\}_e \end{aligned} \quad (28B)$$

The convective derivative of the momentum equation is of the form

$$L^C(m) + \frac{\partial m}{\partial t} + u \frac{\partial m}{\partial x} \quad (29B)$$

The elemental semi discrete approximation is

$$L^C(m_e) = \frac{\partial}{\partial t} \{N_k\}^T \{M\}_e + \{N_k\}^T \{V\}_e \frac{\partial}{\partial x} \{N_k\} \{M\}_e \quad (30B)$$

Defining the dissipation parameter β_α as

$$\bar{\beta}_u \equiv h_e (v_M^1 \delta t + \text{sign } u \ v_M^2 \delta x) \hat{i} \quad (31B)$$

and substituting (30B) and (31B) into the second constraint of equation (15) and simplifying yields

$$\begin{aligned} \bar{\beta}_M \int_{R_e'} \nabla \{N_k\} L^C(m_e) dx &= h_e v_M^1 \int_{R_e'} \frac{d}{dx} \{N_k\} \frac{\partial}{\partial t} \{N_k\}^T \{M\}_e dx \\ &+ h_e v_M^2 \int_{R_e'} \left(\frac{d}{dx} \{N_k\} \right) \{N_k\}^T \{\bar{V}\}_e \frac{\partial}{\partial x} \{N_k\}^T \{M\}_e dx \end{aligned}$$

$$\begin{aligned}
&= h_e v_M^1 \int_{R_e'} \left(\frac{d}{dx} \{N_k\} \right) \{N_k\} dx \{M\}_e \\
&\quad + h_e v_M^2 \int_{R_e'} \left(\frac{d}{dx} \{N_k\} \right) \{\bar{V}\}_e \{N_k\}^T \frac{d}{dx} \{N_k\} dx \{M\}_e \\
&= v_M^1 [A210] \{M\}_e' \\
&\quad + h_e v_M^2 \{\bar{V}\}_e \int_{R_e'} \{N_k\}^T \frac{d}{dx} \{N_k\} \frac{d}{dx} \{N_k\} dx \{M\}_e \\
\bar{B}_M \int_{R_e'} \nabla \{N_k\} L^c(m_e) dx &= h_e v_M^1 [A210] \{M\}_e' \\
&\quad + v_M^2 \{\bar{V}\}_e [A3011] \{M\}_e \tag{32B}
\end{aligned}$$

The summation of equations (28B) and (32B), using the summation in a Taylor series and using the trapezoid rule for integration, yields $\{FM\}_e$ in the form

$$\begin{aligned}
\{FM\}_e &= h_e \left[[A200] + v^1 [A210] \right] \{M\}_{j+1}' + \frac{\Delta t}{2} \left[\{V\}^T [A30] \right. \\
&\quad \left. - [A3010] \right] \{M\} + \frac{1}{Y} \left[[A20] - [A210] \right] \{P\} \\
&\quad + v_M^2 \{\bar{V}\} [A3011] \{M\} + \{A\}^T ([A41000] \{V\}) \{M\} \Big]_{j+1,j} \tag{33B}
\end{aligned}$$

Using the definition in (18B) and similarly defining

$$[D3MAT] = [A30] - [A3010] , \quad (34B)$$

a more computationally efficient form of $\{FM\}_e$ is

$$\begin{aligned} \{FM\}_e = h_e \left[[A200] + v_M^1 [A210] \right] \{M\}_{j+1} + \frac{\Delta t}{2} \left[\{V\}^T [D3MAT] \{M\} \right. \\ \left. + \frac{1}{\gamma} [D2MAT] \{P\} + v_M^2 \{\bar{V}\}^T [A3011] \{M\} \right. \\ \left. + \{A\}^T ([A41000] \{V\}) \{M\} \right]_{j+1,j} \end{aligned} \quad (35B)$$

Energy Equation

The energy equation is given in the form

$$L(\rho e) = \frac{\partial \rho e}{\partial t} + \frac{\partial}{\partial x} [u \rho e + u p] + (u \rho e + u p) A = 0 \quad (36B)$$

Using the stagnation reference conditions specified in (21B), the nondimensionalized form of the energy equation is

$$\begin{aligned} L(\rho^* e^*) = \frac{\partial \rho^* e^*}{\partial t^*} + \frac{\partial}{\partial x^*} u^* \rho^* e^* + \frac{P_0}{\rho_0 e_0} \frac{\partial}{\partial x^*} u^* p^* + (u^* \rho^* e^* \\ + \frac{P_0}{\rho_0 e_0} u^* p^*) A^* \end{aligned} \quad (37B)$$

where $P_0/\rho_0 e_0$ is the Eckert number. Making the substitutions

$P_0/\rho_0 = R T_0$ and $e_0 = c_v T_0$, the Eckert number reduces to $(\gamma-1)$.

Substituting the reduced Eckert number, letting $g = \rho e$, and dropping the superscript asterisk leaves the energy equation in the form

$$L(g) = \frac{\partial g}{\partial t} + \frac{\partial}{\partial x} u g + (\gamma - 1) \frac{\partial}{\partial x} u p + (u g + (\gamma - 1) u p) A' \quad (38B)$$

The elemental semidiscrete approximation of equation (38B) is

$$\begin{aligned} L(g_e) = & \frac{\partial}{\partial t} \{N_k\}^T \{G\}_e + \frac{\partial}{\partial x} (\{N_k\}^T \{V\}_e \{N_k\}^T \{G\}_e) \\ & + (\gamma - 1) \frac{\partial}{\partial x} (\{N_k\}^T \{V\}_e \{N_k\}^T \{P\}_e) + (\{N_k\}^T \{V\}_e \{N_k\}^T \{G\}_e \\ & + (\gamma - 1) \{N_k\}^T \{V\}_e \{N_k\}^T \{P\}_e) \frac{d}{dx} \{N_k\}^T \{A\} \end{aligned} \quad (39B)$$

Substitution into the first constraint of equation (15) and simplifying gives

$$\begin{aligned} \int_{R_e} \{N_k\} L(g_e) dx = & \int_{R_e} \{N_k\} \left(\frac{\partial}{\partial t} \{N_k\}^T \{G\}_e + \frac{\partial}{\partial x} (\{N_k\}^T \{V\}_e \{N_k\}^T \{G\}_e) \right. \\ & + (\{N_k\}^T \{V\}_e \{N_k\}^T \{G\}_e \\ & \left. + (\gamma - 1) \{N_k\}^T \{V\}_e \{N_k\}^T \{P\}_e) \frac{d}{dx} \{N_k\}^T \{A\} \right) dx \end{aligned}$$

$$\begin{aligned}
&= \int_{R_e'} \{N_k\} \{N_k\} \frac{d}{dt} \{G\}_e' dx \\
&+ \int_{R_e'} \{N_k\} \{V\}^T \frac{d}{dx} (\{N_k\} \{N_k\}^T) \{ \{G\} + (\gamma-1) \{P\} \} dx \\
&+ \int_{R_e'} \{A\}_e^T \left(\frac{d}{dx} \{N_k\} \right) \{N_k\} \{N_k\}^T \{V\}_e \{N_k\}^T \{ \{G\}_e \\
&+ (\gamma-1) \{P\} \} dx \\
&= \int_{R_e'} \{N_k\} \{N_k\}^T dx \{G\}_e' \\
&+ \{V\}_e^T \left(\oint_{R_e'} \frac{d}{dx} (\{N_k\} \{N_k\} \{N_k\}^T) \cdot n dx \right. \\
&- \int_{R_e'} \{N_k\} \left(\frac{d}{dx} \{N_k\} \right) \{N_k\}^T dx \{ \{G\}_e + (\gamma-1) \{P\}_e \} \\
&+ \{A\}_e^T ([A41000] \{V\}_e) \{ \{G\}_e + (\gamma-1) \{P\}_e \} \\
&\int_{R_e'} \{N_k\} L(g_e) dx = h_e [A200] \{G\}_e' + \{V\}_e^T [[A30] - [A3010]] \{ \{G\}_e \\
&+ (\gamma-1) \{P\}_e \} + \{A\}_e^T ([A41000] \{V\}_e) \{ \{G\}_e + (\gamma-1) \{P\} \} \quad (40B)
\end{aligned}$$

The convective derivative of the total energy is of the form

$$L^c(g) = \frac{\partial g}{\partial t} + u \frac{\partial g}{\partial x} \quad (41B)$$

and the elemental semidiscrete approximation is

$$L^c(g_e) = \frac{\partial}{\partial t} \{N_k\}^T \{G\}_e + \frac{\partial}{\partial x} (\{N_k\}^T \{V\}_e \{N_k\}^T \{G\}_e) \quad (42B)$$

Defining the dissipation parameter $\bar{\beta}_\alpha$ as

$$\bar{\beta}_G = h_e (v_G^1 \delta t + \text{sign } u \ v_G^2 \delta x) i \quad (43B)$$

and substituting (42B) and (43B) into the second constraint of equation (15) and simplifying yields

$$\begin{aligned} \bar{\beta}_G \int_R \nabla \{N_k\} L^c(g_e) dx &= h_e v_G^1 \int_{R_e} \frac{d}{dx} \{N_k\} \frac{\partial}{\partial t} \{N_k\}^T \{G\}_e dx \\ &\quad + h_e v_G^2 \int_{R_e} \left(\frac{d}{dx} \{N_k\} \right) \{N_k\}^T \{V\}_e \frac{\partial}{\partial x} \{N_k\}^T \{G\}_e dx \\ &= h_e v_G^1 \int_{R_e} \left(\frac{d}{dx} \{N_k\} \right) \{N_k\}^T dx \{G\}_e' \\ &\quad + h_e v_G^2 \int_{R_e} \{\bar{V}\}_e^T \{N_k\} \frac{d}{dx} \{N_k\} \frac{d}{dx} \{N_k\}^T \{G\}_e dx \\ &= h_e v_G^1 [A210] \{G\}_e' \\ &\quad + v_G^2 \{\bar{V}\}_e^T \int_{R_e} \{N_k\} \frac{d}{dx} \{N_k\} \frac{d}{dx} \{N_k\}^T dx \{G\}_e \end{aligned}$$

$$\beta_G \int_{R_e} V\{N_k\} L^c(g_e) dx = h_e v_G^1 [A210] \{G\}'_e$$

$$+ v_G^2 \{\bar{V}\}_e^T [A3011] \{G\}_e \quad (44B)$$

The summation of equations (40B) and (44B), using the summation in a Taylor series and using the trapezoid rule for integration yields the final form of $\{FG\}_e$ as

$$\{FG\}_e = h_e \left[[A200] + v_G^1 [A210] \right] \{G\}'_{j+1}$$

$$+ \frac{\Delta t}{2} \left[\{V\} \left[[A30] - [A3010] \right] \{ \{G\} + (\gamma-1)\{P\} \} \right.$$

$$+ v_G^2 \{\bar{V}\}_e^T [A3011] \{G\} + \{A\}^T ([A41000] \{V\} \{G\}$$

$$\left. + (\gamma-1)\{P\} \right]_{j+1,j} \quad (45B)$$

Substituting the terms defined in (17B) and (34B) and factoring common terms yields the form

$$\{FG\}_e = h_e \left[[A200] + v_G^1 [A210] \right] \{G\}'_{j+1} + \frac{\Delta t}{2} \left[\{V\}^T [D3MAT] \right.$$

$$+ [A40001] \{A\} \left. \{ \{G\} + (\gamma-1)\{P\} \} \right.$$

$$\left. + v_G^2 \{\bar{V}\}_e^T [A3011] \{G\} \right]_{j+1,j} \quad (46B)$$

Equation of State

The equation of state is given in the form

$$L(p) = p - (\gamma-1)\left[\rho e - \frac{1}{2}\rho u \cdot u\right] = 0 \quad (47B)$$

Using stagnation reference conditions specified in (21B), the non-dimensionalized form of the equation of state is

$$L(p^*) = p^* - (\gamma-1)\left[\left(\frac{\rho_0 e_0}{p_0}\right)\rho^* e^* - \frac{1}{2}\left(\frac{\rho_0 a_0^2}{p_0}\right)\rho^* u^* \cdot u^*\right] \quad (48B)$$

where the nondimensional groups are the inverses of the Euler number and the Eckert number is previously defined. Making appropriate substitutions and dropping the superscript asterisk leave the equation of state in the form

$$L(p) = p - \left[g - \frac{1}{2}\gamma(\gamma-1)mu\right] \quad (49B)$$

The elemental semidiscrete approximation of equation (49B) is

$$\begin{aligned} L(p_e) &= \{N_k\}^T \{P\}_e - [\{N_k\}^T \{G\}_e \\ &\quad - \frac{1}{2}\gamma(\gamma-1)\{N_k\}^T \{V\}_e \{N_k\}^T \{M\}_e] \end{aligned} \quad (50B)$$

Substitution into the first constraint of equation (15) and simplifying gives

$$\begin{aligned}
\int_{R_e'} \{N_k\} L(p_e) dx &= \int_{R_e'} \{N_k\} (\{N_k\}^T \{P\}_e - [\{N_k\}^T \{P\}_e \\
&\quad - \frac{1}{2} \gamma(\gamma-1) \{N_k\}^T \{V\}_e \{N_k\}^T \{M\}_e] dx \\
&= \int_{R_e'} \{N_k\} \{N_k\}^T \{P\} dx - [\int_{R_e'} \{N_k\} \{N_k\}^T \{G\}_e dx - \\
&\quad - \frac{1}{2} \gamma(\gamma-1) \int_{R_e'} \{N_k\} \{V\}_e^T \{N_k\} \{N_k\}^T \{M\}_e dx] \\
&= \int_{R_e'} \{N_k\} \{N_k\}^T dx \{P\}_e - \int_{R_e'} \{N_k\} \{N_k\}^T dx \{G\}_e \\
&\quad + \frac{1}{2} \gamma(\gamma-1) \{V\}_e^T \int_{R_e'} \{N_k\} \{N_k\} \{N_k\}^T dx \{M\}_e \\
\int_{R_e'} \{N_k\} L(P_e) dx &= h_e [A200] \{P\}_e - h_e [A200] \{G\}_e \\
&\quad + \frac{h_e}{2} \gamma(\gamma-1) \{V\}_e^T [A3000] \{M\}_e \quad (51B)
\end{aligned}$$

Since there is no convective derivative involving pressure and no derivatives occur in the equation of state, the form of $\{FP\}_e$ is identical to (51B). Factorization of common terms gives the final form

$$\{FP\}_e = h_e \left[[A200] \{ \{P\}_e - \{G\}_e \} + \frac{1}{2} \gamma(\gamma-1) \{V\}_e^T [A3000] \{M\}_e \right] \quad (52B)$$

APPENDIX C

DERIVATION OF MATRIX FORM OF JACOBIAN

The following is a step by step formulation of the derivatives which form the Newton Matrix Iteration Jacobian. The derivatives of each of the equations, $\{FI\}_e$, are taken with respect to the independent variables, $\{QI\}_e$, yielding a total of 16 derivatives.

Continuity Equation

The following derivatives are developed using the form of $\{FR\}_e$ presented in equation (16B). The first derivative is taken with respect to density $\{R\}$.

$$\begin{aligned}
 [JRR]_e &\equiv \frac{\partial \{FR\}_e}{\partial \{R\}} = h_e \left[A200 \right] + v_R^1 [A210] \left[\frac{\partial \{R\}}{\partial \{R\}} \right]^{j+1} \\
 &+ \frac{\Delta t}{2} \left[\left[A02 \right] - [A210] \right] \frac{\partial \{M\}}{\partial \{M\}} + v_R^2 \frac{\partial}{\partial \{R\}} (\{\bar{V}\} [A3011] \{R\}) \\
 &+ \frac{\partial}{\partial \{R\}} (\{A\}^T ([A41000] \{V\}) \{R\}) \left[\right]_{j+1} \quad (1C)
 \end{aligned}$$

The values of $\{\cdot\}_j$ are fixed and thus their derivatives are zero; therefore, these terms are omitted in this presentation.

$$\begin{aligned}
 [JRR]_e &= h_e \left[A200 \right] + v_R^1 [A210] + \frac{\Delta t}{2} \left[\left[A20 \right] - [210] \right] \{0\} \\
 &+ v_R^2 \left[\{\bar{V}\} [A3011] \frac{\partial \{R\}}{\partial \{R\}} + \{0\} [A3110] \frac{\partial \{V\}}{\partial \{R\}} \right] \\
 &+ \{A\}^T \left([A41000] \frac{\partial \{V\}}{\partial \{R\}} \right) \{R\} + \{A\}^T [A41000] \{V\} \frac{\partial \{R\}}{\partial \{R\}} \left[\right]
 \end{aligned}$$

$$\begin{aligned}
&= h_e \left[[A200] + v_R^1 [A210] \right] + \frac{\Delta t}{2} \left[v_R^2 [V] [A3011] \right. \\
&\quad - \{A10\}^T \left| \left\{ \frac{M}{R^2} \right\} \right| [A3110] \left. \right] \\
&\quad + \{A\}^T [A41000] \left(\{A10\}^T \left\{ \frac{M}{R^2} \right\} \right) \{R\} + \{A\}^T [A41000] \{V\} \left. \right] \\
[JRR]_e &= h_e \left[[A200] + v_R^1 [A210] \right] + \frac{\Delta t}{2} \left[v_R^2 [V] [A3011] \right. \\
&\quad - \{A10\}^T \left| \left\{ \frac{M}{R^2} \right\} \right| \{R\}^T [A3110] \left. \right] + \{A\}^T [A41000] \{V\} \\
&\quad - \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{R\} \left. \right] \tag{2C}
\end{aligned}$$

The matrix derivatives used in the preceeding formulation are elemental averages of the given nodal functions. Three derivative forms are illustrated in the preceeding formulation. The first is the derivative of one independent variable, $\{R, M, G, P\}$, with respect to a different independent variable. The example in $\{JRR\}_e$ is $\partial\{M\}/\partial\{R\}$. Since momentum is not a function of density the derivative at each node is zero, and the elemental average is zero. The second form is the derivative of an independent variable with respect to itself. The derivative of any function with respect to itself is a constant [Thomas, 1972] taken to be one, thus the elemental average of $\partial\{R\}/\partial\{R\} = 1$. The third form is the derivative of a function with respect to an independent variable. The

example from $\{JRR\}_e$ is $\partial\{V\}/\partial\{R\}$. Where, for a given node, $u_n = m_n/\rho_n$. Thus taken at a specific node

$$\frac{\partial u_n}{\partial \rho_n} = \frac{\partial}{\partial \rho_n} (m_n / \rho_n)$$

$$= m_n \frac{\partial}{\partial \rho_n} \left(\frac{1}{\rho_n} \right)$$

$$\frac{\partial u_n}{\partial \rho_n} = - m_n / \rho_n^2 \quad (3C)$$

Therefore,

$$\frac{\partial\{V\}}{\partial\{R\}}_e = -\{A10\}^T \left\{ \frac{M}{R^2} \right\}_e \quad (4C)$$

where $\left\{ \frac{M}{R^2} \right\}$ is the vector of nodal values of the function m/ρ^2 and $\{A10\}$ is a column matrix which defines an elemental average, i.e. for $k=1,2$,

$$\{A10\}_1 = \frac{1}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}, \quad \{A10\}_2 = \frac{1}{6} \begin{Bmatrix} 1 \\ 4 \\ 1 \end{Bmatrix} \quad (5C)$$

Only one other nonzero derivative of the third form occurs in the formulation of the Jacobian. That is the derivative $\partial\{V\}/\partial\{M\}$. Again taking the function at a specific node

$$\frac{\partial u_n}{\partial m_n} = \frac{\partial}{\partial m_n} \frac{m_n}{\rho_n}$$

$$= \frac{1}{\rho_n} \frac{\partial m_n}{\partial m_n}$$

$$\frac{\partial u_n}{\partial m_n} = \frac{1}{\rho_n}$$

(6C)

Expanding to an elemental average,

$$\frac{\partial \{V\}_e}{\partial \{M\}_e} = \{A10\}^T \left\{ \frac{1}{R} \right\}_e \quad (7C)$$

With the preceeding explanations the formulation of the remaining terms in the Jacobian becomes a simple exercise.

The development of the term $\{JRM\}_e$ requires the use of an alternate form of one term of equation (16A). This substitution is

$$\{A\}^T [A3100] \{M\} = \{A\}^T ([A41000] \{V\}) \{R\} \quad (8C)$$

using the revised form, $\{JRM\}_e$ is derived as follows

$$\begin{aligned} [JRM]_e = \frac{\partial \{FR\}_e}{\partial \{M\}_e} &= h_e \left[[A200] + v_R^1 [A210] \right] \frac{\partial \{R\}}{\partial \{M\}} \\ &+ \frac{\Delta t}{2} \left[\left[[A20] - [A210] \right] \frac{\partial \{M\}}{\partial \{M\}} \right. \\ &\left. + v_R^2 \frac{\partial}{\partial \{M\}} (\{\bar{V}\}^T [A3011] \{R\}) + \{A\}^T [A3100] \frac{\partial \{M\}}{\partial \{M\}} \right] \end{aligned}$$

$$\begin{aligned}
&= [\bar{h}_e[A200] + v_R^1[A210]] \cdot 0 + \frac{\Delta t}{2} [A20] - [A210] \\
&\quad + v_R^2 [\bar{V}]^T [A3011] \frac{\partial \{R\}}{\partial \{M\}} + \{R\}^T [3110] \frac{\partial \{\bar{V}\}}{\partial \{M\}} \\
&\quad + \{A\}^T [A3100] \Big]
\end{aligned}$$

$$\begin{aligned}
[JRM]_e &= \frac{\Delta t}{2} [A20] - [A210] + v_R^2 \{A10\}^T \Big| \left\{ \frac{1}{R} \right\} \Big| \{R\}^T [A3110] \\
&\quad + \{A\}^T [A3100] \Big] \tag{9C}
\end{aligned}$$

Using definition (18B), equation (9C) reduces to

$$\begin{aligned}
[JRM]_e &= \frac{\Delta t}{2} [D2MAT] + v_R^2 \{A10\}^T \Big| \left\{ \frac{1}{R} \right\} \Big| \{R\}^T [A3110] \\
&\quad + \{A\}^T [A3100] \Big] \tag{10C}
\end{aligned}$$

The continuity equation does not involve the energy, $\{G\}_e$, or the pressure $\{P\}_e$; therefore, the derivatives of $\{FR\}_e$ with respect to these values are zero or,

$$[JRG]_e \equiv \frac{\partial \{FR\}_e}{\partial \{G\}_e} = [0] \tag{11C}$$

$$[JRP]_e \equiv \frac{\partial \{FR\}_e}{\partial \{P\}_e} = [0] \tag{12C}$$

Momentum Equation

The derivative of $\{FM\}_e$ with respect to $\{R\}_e$ is

$$\begin{aligned}
 [JMR]_e &\equiv \frac{\partial \{FM\}_e}{\partial \{R\}_e} = h_e \left[[A200] + v_M^1 [A210] \right] \frac{\partial \{M\}}{\partial \{R\}} \\
 &+ \frac{\Delta t}{2} \left[\frac{\partial}{\partial \{R\}} (\{V\}^T [A30] - [A3010]) \{M\} \right. \\
 &+ \frac{1}{\gamma} [A20] - [A210] \left. \right] \frac{\partial \{P\}}{\partial \{R\}} \\
 &+ v_M^2 \frac{\partial}{\partial \{R\}} (\{\bar{V}\}^T [A3011] \{M\}) \\
 &+ \frac{\partial}{\partial \{R\}} (\{A\}^T ([A41000] \{V\}) \{M\}) \left. \right] \\
 &= h_e \left[[A200] + v_M^1 [A210] \cdot 0 \right. \\
 &+ \frac{\Delta t}{2} \left[\{V\}^T [A30] - [A3010] \right] \frac{\partial \{M\}}{\partial \{R\}} \\
 &+ \{M\}^T [A30] - [A3010] \left. \right] \frac{\partial \{V\}}{\partial \{R\}} + \frac{1}{\gamma} [A20] \\
 &- [A210] \left. \right] \frac{\partial \{P\}}{\partial \{R\}} + v_M^2 \left[\{\bar{V}\} [A3011] \frac{\partial \{M\}}{\partial \{R\}} \right. \\
 &+ \{M\} [A3110] \frac{\partial \{\bar{V}\}}{\partial \{R\}} \left. \right] + \{A\}^T [A41000] \frac{\partial \{V\}}{\partial \{R\}} \{M\}
 \end{aligned}$$

$$\begin{aligned}
&= \frac{\Delta t}{2} \left[-\{A\}^T \left\{ \frac{M}{R^2} \right\} \{M\}^T [A30] - [A3010] \right] \\
&\quad - v_M^2 \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{M\}^T [A3110] \\
&\quad - \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{M\}^T [A4001] \{A\} \right]
\end{aligned}$$

Factoring Common Terms

$$\begin{aligned}
[JMR]_e &= - \frac{\Delta t}{2} \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{M\}^T [A30] - [A3010] \\
&\quad + v_M^2 \text{sign}(u) [A3110] + [A40001] \{A\}^T \quad (13C)
\end{aligned}$$

Further simplification using definition (34B) gives

$$\begin{aligned}
[JMR]_e &= - \frac{\Delta t}{2} \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{M\}^T [D3MAT] \\
&\quad + v_M^2 \text{sign}(u) [A3110] + [A40001] \{A\}^T \quad (14C)
\end{aligned}$$

The derivative of $\{FM\}_e$ with respect to $\{M\}_e$

$$\begin{aligned}
[JMM]_e &\equiv \frac{\partial \{FM\}}{\partial \{M\}}_e = h_e \left[[A200] + v_M^1 [A210] \right] \frac{\partial \{M\}}{\partial \{M\}} \\
&+ \frac{\Delta t}{2} \left[\frac{\partial}{\partial \{M\}} (\{V\}^T [A30] - [A3010]) \{M\} \right. \\
&+ \frac{1}{\gamma} [A20] - [A210] \frac{\partial \{P\}}{\partial \{M\}} \\
&+ v_M^2 \frac{\partial}{\partial \{M\}} (\{\bar{V}\}^T [A3011] \{M\}) \\
&\left. + \frac{\partial}{\partial \{M\}} (\{A\}^T [A41000] \{V\}) \{M\} \right] \\
&= h_e \left[[A200] + v_M^1 [A210] \right] + \frac{\Delta t}{2} \left[\{V\}^T [A30] \right. \\
&- [A3010] \frac{\partial \{M\}}{\partial \{M\}} + \{M\}^T [A30] - [A3010] \frac{\partial \{V\}}{\partial \{M\}} \\
&+ \frac{1}{\gamma} [A20] - [A210] \cdot 0 + v_M^2 [\{\bar{V}\}^T [A3011] \frac{\partial \{M\}}{\partial \{M\}} \\
&+ \{M\}^T [A3110] \frac{\partial \{\bar{V}\}}{\partial \{M\}} + \{A\}^T [A41000] \frac{\partial \{V\}}{\partial \{M\}} \{M\} \\
&\left. + \{A\}^T [A41000] \{V\} \frac{\partial \{M\}}{\partial \{M\}} \right]
\end{aligned}$$

$$\begin{aligned}
&= h_e \left[[A200] + v_M^1 [A210] \right] + \frac{\Delta t}{2} \left[\{V\}^T [A30] \right. \\
&\quad \left. - [3010] \right] + \{A10\}^T \left\{ \frac{1}{R} \right\} \{M\}^T [A30] - [A3010] \right] \\
&\quad + v_M^2 \left[\{\bar{V}\}^T [A3011] + \{A10\}^T \left| \left\{ \frac{1}{R} \right\} \right| \{M\}^T [A3110] \right] \\
&\quad + \{A\}^T [A41000] \{A10\}^T \left\{ \frac{1}{R} \right\} \{M\} \\
&\quad \left. + \{A\}^T [A41000] \{V\} \right]
\end{aligned}$$

Factoring common terms

$$\begin{aligned}
&= h_e \left[[A200] + v_M^1 [A210] \right] + \frac{\Delta t}{2} \left[\{V\} \right. \\
&\quad \left. + \{A10\}^T \left\{ \frac{1}{R} \right\} \{M\} \right\}^T [A30] - [A3010] \right] \\
&\quad + v_M^2 \left[\{\bar{V}\}^T [A3011] + \{A10\}^T \left| \left\{ \frac{1}{R} \right\} \right| \{M\}^T [A3110] \right] \\
&\quad \left. + \{V\} + \{A10\}^T \left\{ \frac{1}{R} \right\} \{M\} \right\}^T [A40001] \{A\} \right]
\end{aligned}$$

$$\begin{aligned}
[\text{JMM}]_e = h_e \left[[\text{A200}] + v_M^1 [\text{A210}] \right] + \frac{\Delta t}{2} \left[\{V\} \right. \\
+ \{A10\}^T \left\{ \frac{1}{R} \right\} \{M\}^T \left[[\text{A30}] - [\text{A3010}] + [\text{A40001}] \{A\} \right] \\
\left. + v_M^2 \left[\{\bar{V}\}^T [\text{A3011}] + \{A10\}^T \left\{ \frac{1}{R} \right\} \{M\}^T [\text{A3110}] \right] \right] \quad (15C)
\end{aligned}$$

Further simplification using definition (34B) gives

$$\begin{aligned}
[\text{JMM}]_e = h_e \left[[\text{A200}] + v_M^1 [\text{A210}] \right] + \frac{\Delta t}{2} \left[\{V\} \right. \\
+ \{A10\}^T \left\{ \frac{1}{R} \right\} \{M\}^T \left[[\text{D3MAT}] + [\text{A40001}] \{A\} \right] \\
\left. + v_M^2 \left[\{\bar{V}\}^T [\text{A3011}] + \{A10\}^T \left\{ \frac{1}{R} \right\} \{M\}^T [\text{A3110}] \right] \right] \quad (16C)
\end{aligned}$$

The term $\{FM\}_e$ does not depend on the energy, $\{G\}$, and therefore, the derivative of $\{FM\}_e$ with respect to $\{G\}$ is zero, i.e.

$$[\text{JMG}]_e = \frac{\partial \{FM\}}{\partial \{G\}} = [0] \quad (17C)$$

The derivative of $\{FM\}_e$ with respect to $\{P\}$ is

$$\begin{aligned}
[\text{JMP}]_e \frac{\partial \{FM\}_e}{\partial \{P\}_e} &= h_e \left[[A200] + v_M^1 [A210] \right] \frac{\partial \{M\}}{\partial \{P\}} \\
&+ \frac{\Delta t}{2} \left[\frac{\partial}{\partial \{P\}} (\{V\}^T [A30] - [A3010] \{M\}) \right. \\
&+ \frac{1}{\gamma} [A20] - [A210] \left. \right] \frac{\partial \{P\}}{\partial \{P\}} \\
&+ v_M^2 \frac{\partial}{\partial \{P\}} (\{V\}^T [A3011] \{M\}) \\
&+ \left. \frac{\partial}{\partial \{P\}} \{A\}^T ([A41000] \{V\}) \{M\} \right] \\
&= h_e \left[[A200] + v_M^1 [A210] \right] \cdot 0 + \frac{\Delta t}{2} \left[\{V\}^T [A30] \right. \\
&- [A3010] \left. \right] \frac{\partial \{M\}}{\partial \{P\}} + \{M\} [A30] - [A3010] \left. \right] \frac{\partial \{V\}}{\partial \{P\}} \\
&+ \frac{1}{\gamma} [A20] - [A210] \left. \right] \frac{\partial \{P\}}{\partial \{P\}} \\
&+ v_M^2 \left[\{V\}^T [A3011] \frac{\partial \{M\}}{\partial \{P\}} + \{M\}^T [A3110] \frac{\partial \{V\}}{\partial \{P\}} \right] \\
&+ \{A\}^T [A41000] \frac{\partial \{V\}}{\partial \{P\}} \{M\} + \{A\}^T [A41000] \{V\} \frac{\partial \{M\}}{\partial \{P\}} \left. \right] \\
&= [0] + \frac{\Delta t}{2} [0] + [0] + \frac{1}{\gamma} [A20] - [A210] \\
&+ v_M^2 [0] + [0] + [0] + [0]
\end{aligned}$$

$$[\text{JMP}]_e = \frac{\Delta t}{2\gamma} \left[[A20] - [A210] \right] \quad (21C)$$

Substituting (18B) yields

$$[\text{JMP}]_e = \frac{\Delta t}{2\gamma} [\text{D2MAT}] \quad (22C)$$

Energy Equation

The derivative of $\{\text{FG}\}_e$ with respect to $\{\text{R}\}$ is

$$\begin{aligned} [\text{JGR}]_e \equiv \frac{\partial \{\text{FG}\}}{\partial \{\text{R}\}} = & h_e \left[[A200] + v_G^1 [A210] \right] \frac{\partial \{\text{G}\}}{\partial \{\text{R}\}} \\ & + \frac{\Delta t}{2} \left[\frac{\partial}{\partial \{\text{R}\}} (\{\text{V}\}^T \left[[A30] - [A3010] \right] \{\text{G}\} \right. \\ & + (\gamma - 1) \{\text{P}\} + v_G^2 \frac{\partial}{\partial \{\text{R}\}} (\{\text{V}\}^T [A3011] \{\text{G}\}) \\ & \left. + \frac{\partial}{\partial \{\text{R}\}} (\{\text{A}\}^T ([A41000] \{\text{V}\}) \{\text{G}\} + (\gamma - 1) \{\text{P}\}) \right] \end{aligned}$$

$$\begin{aligned}
&= h_e \left[[A200] + v_G^1 [A210] \right] \cdot 0 + \frac{\Delta t}{2} \left[\{V\}^T [A30] \right. \\
&\quad \left. - [A3010] \right] \left(\frac{\partial \{G\}}{\partial \{R\}} + (\gamma-1) \frac{\partial \{P\}}{\partial \{R\}} \right) + \{G\} \\
&\quad + (\gamma-1) \{P\} \}^T [A30] - [A3010] \right] \frac{\partial \{V\}}{\partial \{R\}} \\
&\quad + v_G^2 \left[\{\bar{V}\}^T [A3011] \frac{\partial \{G\}}{\partial \{R\}} + \{G\}^T [A3110] \frac{\partial \{\bar{V}\}}{\partial \{R\}} \right] \\
&\quad + \{A\}^T [A41000] \frac{\partial \{V\}}{\partial \{R\}} \{ \{G\} + (\gamma-1) \{P\} \} \\
&\quad + \{A\}^T [A41000] \{V\} \left(\frac{\partial \{G\}}{\partial \{R\}} + (\gamma-1) \frac{\partial \{P\}}{\partial \{R\}} \right) \left. \right] \\
&= \frac{\Delta t}{2} \left[\{V\}^T [A30] - [A3010] \right] \cdot 0 - \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{ \{G\} \\
&\quad + (\gamma-1) \{P\} \}^T [A30] - [A3010] \\
&\quad + v_G^2 [\{V\} [A3011] \cdot 0 \\
&\quad - \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{ \{G\}^T [A3110] \right] \\
&\quad + -\{A10\}^T \left\{ \frac{M}{R^2} \right\} \{A\}^T [A41000] \{ \{G\} + (\gamma-1) \{P\} \} \\
&\quad + \{A\}^T [A41000] \{V\} \cdot 0
\end{aligned}$$

$$\begin{aligned}
& = -\frac{\Delta t}{2} \left[\{A10\}^T \left\{ \frac{M}{R^2} \right\} \{ \{G\} + (\gamma-1)\{P\} \}^T [A30] \right. \\
& \quad - [A3010] + v_g^2 \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{G\}^T [A3110] \\
& \quad \left. + \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{ \{G\} + (\gamma-1)\{P\} \}^T [A40001] \{A\} \right] \\
[JGR]_e & = -\frac{\Delta t}{2} \left[\{A10\}^T \left\{ \frac{M}{R^2} \right\} \{ \{G\} + (\gamma-1)\{P\} \}^T [A30] - [A3010] \right. \\
& \quad \left. + [A40001] \{A\} \right] + v_g^2 \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{G\}^T [A3110] \quad (23C)
\end{aligned}$$

Further simplification using definition (34B) yields

$$\begin{aligned}
[JGR]_e & = -\frac{\Delta t}{2} \left[\{A10\}^T \left\{ \frac{M}{R^2} \right\} \{ \{G\} + (\gamma-1)\{P\} \}^T [D3MAT] \right. \\
& \quad \left. + [A40001] \{A\} \right] \quad (24C)
\end{aligned}$$

The derivative of $\{FG\}_e$ with respect to $\{M\}$ is

$$\begin{aligned}
[JGM]_e &\equiv \frac{\partial \{FG\}}{\partial \{M\}} \frac{e}{e} = h_e \left[[A200] + v_G^1 [A210] \right] \frac{\partial \{G\}}{\partial \{M\}} \\
&+ \frac{\Delta t}{2} \left[\frac{\partial}{\partial \{M\}} (\{V\}^T [A30] - [A3010]) \{G\} \right. \\
&+ (\gamma - 1) \{P\} \} + v_G^2 \frac{\partial}{\partial \{M\}} (\{\bar{V}\}^T [A3011] \{G\}) \\
&\left. + \frac{\partial}{\partial \{M\}} (\{A\}^T [A41000] \{V\}) \{G\} + (\gamma - 1) \{P\} \right] \\
&= h_e \left[[A200] + v_G^1 [A210] \right] \cdot 0 + \frac{\Delta t}{2} \left[\{V\}^T [A30] \right. \\
&- [A3010] \left] \left(\frac{\partial \{G\}}{\partial \{M\}} + (\gamma - 1) \frac{\partial \{P\}}{\partial \{M\}} \right) + \{G\} \\
&- (\gamma - 1) \{P\} \}^T [A30] - [A3010] \left] \frac{\partial \{V\}}{\partial \{M\}} \right. \\
&+ v_G^2 (\{\bar{V}\} [A3011] \frac{\partial \{G\}}{\partial \{M\}} + \{G\} [A3110] \frac{\partial \{\bar{V}\}}{\partial \{M\}}) \\
&+ \{A\}^T [A41000] \frac{\partial \{V\}}{\partial \{M\}} \{G\} + (\gamma - 1) \{P\} \} \\
&\left. + \{A\}^T [A41000] \{V\} \left(\frac{\partial \{G\}}{\partial \{M\}} + (\gamma - 1) \frac{\partial \{P\}}{\partial \{M\}} \right) \right]
\end{aligned}$$

$$\begin{aligned}
&= \frac{\Delta t}{2} \left[\{V\}^T \left[[A30] - [A3010] \cdot 0 + \{A10\}^T \left\{ \frac{1}{R} \right\} \{G\} \right. \right. \\
&\quad + (\gamma-1)\{P\}\}^T \left[[A30] - [A3010] \right] \\
&\quad + v_G^2 \left(\{\bar{V}\} [A3011] \cdot 0 + \{A10\}^T \left\{ \frac{1}{R} \right\} \{G\}^T [A3110] \right. \\
&\quad + \{A10\}^T \left\{ \frac{1}{R} \right\} \{G\} + (\gamma-1)\{P\}\}^T [A40001] \{A\} \\
&\quad \left. \left. + \{A\}^T [A41000] \{V\} \cdot 0 \right] \right] \\
&= \frac{\Delta t}{2} \left[\{A10\}^T \left\{ \frac{1}{R} \right\} \{G\} + (\gamma-1)\{P\}\}^T \left[[A30] - [A3010] \right] \right. \\
&\quad + v_G^2 \{A10\}^T \left\{ \frac{1}{R} \right\} \{G\}^T [A3110] \\
&\quad \left. + \{A10\}^T \left\{ \frac{1}{R} \right\} \{G\} + (\gamma-1)\{P\}\}^T [A40001] \{A\} \right]
\end{aligned}$$

Factoring common terms

$$\begin{aligned}
[JGM]_e &= \frac{\Delta t}{2} \left[\{A10\}^T \left\{ \frac{1}{R} \right\} \{G\} + (\gamma-1)\{P\}\}^T \left[[A30] - [A3010] \right. \right. \\
&\quad \left. \left. + [A40001] \{A\} \right] + v_M^2 \{A10\}^T \left\{ \frac{1}{R} \right\} \{G\}^T [A3110] \right] \quad (25C)
\end{aligned}$$

The derivative of $\{FG\}_e$ with respect to $\{G\}$ is

$$\begin{aligned}
[JGG]_e &\equiv \frac{\partial \{FG\}_e}{\partial \{G\}_e} = h_e \left[[A200] + v_G^1 [A210] \right] \frac{\partial \{G\}}{\partial \{G\}} \\
&+ \frac{\Delta t}{2} \left[\frac{\partial}{\partial \{G\}} (\{V\}^T \left[[A30] - [A3010] \right] \{G\} \right. \\
&+ (\gamma-1) \{P\} \} + v_G^2 \frac{\partial}{\partial \{G\}} (\{\bar{V}\}^T [A3011] \{G\} \\
&+ \left. \frac{\partial}{\partial \{G\}} (\{A\}^T ([A41000] \{V\}) \{G\} + (\gamma-1) \{P\}) \right) \Big] \\
&= h_e \left[[A200] + v_G^1 [A210] \right] + \frac{\Delta t}{2} \left[\{V\}^T \left[[A30] \right. \right. \\
&- [A3010] \left. \right] \left(\frac{\partial \{G\}}{\partial \{G\}} + (\gamma-1) \frac{\partial \{P\}}{\partial \{G\}} \right) + \{G\} \\
&+ (\gamma-1) \{P\} \}^T \left[[A30] - [A3010] \right] \frac{\partial \{V\}}{\partial \{G\}} \\
&+ v_G^2 (\{\bar{V}\}^T [A3011] \frac{\partial \{G\}}{\partial \{G\}} + \{G\}^T [A3110] \frac{\partial \{\bar{V}\}}{\partial \{G\}}) \\
&+ \{A\}^T [A41000] \frac{\partial \{V\}}{\partial \{G\}} \{G\} + (\gamma-1) \{P\} \} \\
&+ \left. \{A\}^T [A41000] \{V\} \left(\frac{\partial \{G\}}{\partial \{G\}} + (\gamma-1) \frac{\partial \{P\}}{\partial \{G\}} \right) \right]
\end{aligned}$$

$$\begin{aligned}
&= h_e \left[[A200] + v_G^1 [A210] \right] + \frac{\Delta t}{2} \left[\{V\}^T [A30] \right. \\
&\quad \left. - [A3010] \right] (1+0) + \{G\} + (\gamma-1)\{P\} \}^T [A30] \\
&\quad - [A3010] \right] \cdot 0 + v_G^2 \{ \bar{V} \}^T [A3011] \\
&\quad + \{G\}^T [A3110] \cdot 0 + \{A\}^T [A41000] \cdot 0 \cdot \{G\} \\
&\quad + (\gamma-1)\{P\} + \{A\}^T [A41000] \{V\} (1+0) \left. \right] \\
&= h_e \left[[A200] + v_G^1 [A210] \right] + \frac{\Delta t}{2} \left[\{V\}^T [A30] \right. \\
&\quad \left. - [A3010] \right] + v_G^2 \{V\} [A3011] + \{V\}^T [A41000] \{A\} \left. \right]
\end{aligned}$$

Factoring common terms

$$\begin{aligned}
[JGG]_e &= h_e \left[[A200] + v_G^1 [A210] \right] + \frac{\Delta t}{2} \left[\{V\}^T [A30] - [A3010] \right. \\
&\quad \left. + [A40001] \{A\} \right] + v_G^2 \{ \bar{V} \}^T [A3011] \left. \right] \quad (26C)
\end{aligned}$$

Substitution of (34B) yields

$$\begin{aligned}
[JGG]_e &= h_e \left[[A200] + v_G^1 [A210] \right] + \frac{\Delta t}{2} \left[\{V\}^T [D3MAT] \right. \\
&\quad \left. + [A40001] \{A\} \right] + v_G^2 \{V\}^T [A3011] \left. \right] \quad (27C)
\end{aligned}$$

The derivative of $\{FG\}_e$ with respect to $\{P\}$

$$\begin{aligned}
 [JGP]_e &\equiv \frac{\partial \{FG\}_e}{\partial \{P\}_e} = h_e \left[[A200] + v_G^1 [A210] \right] \frac{\partial \{G\}}{\partial \{P\}} \\
 &\quad + \frac{\Delta t}{2} \left[\frac{\partial}{\partial \{P\}} (\{V\} \left[[A30] - [A3010] \right] \{G\} \right. \\
 &\quad \left. + (\gamma-1)\{P\} \right) + v_G^2 \frac{\partial}{\partial \{P\}} (\{V\}^T [A3011] \{G\}) \\
 &\quad \left. + \frac{\partial}{\partial \{P\}} (\{A\}^T ([A41000] \{V\}) (\{G\} + (\gamma-1)\{P\})) \right] \\
 &= h_e \left[[A200] + v_G^1 [A210] \right] \cdot 0 + \frac{\Delta t}{2} \left[\{V\}^T \left[[A30] \right. \right. \\
 &\quad \left. \left. - [A3010] \right] \left(\frac{\partial \{G\}}{\partial \{P\}} + (\gamma-1) \frac{\partial \{P\}}{\partial \{P\}} \right) + \{G\} \right. \\
 &\quad \left. + (\gamma-1)\{P\} \right]^T \left[[A30] - [A3010] \right] \frac{\partial \{V\}}{\partial \{P\}} \\
 &\quad + \{A\}^T [A41000] \frac{\partial \{V\}}{\partial \{P\}} (\{G\} + (\gamma-1)\{P\}) \\
 &\quad \left. + \{A\}^T [A41000] \{V\} \left(\frac{\partial \{G\}}{\partial \{G\}} + (\gamma-1) \frac{\partial \{P\}}{\partial \{P\}} \right) \right]
 \end{aligned}$$

$$\begin{aligned}
&= \frac{\Delta t}{2} \left[\{V\}^T \left[[A30] - [A3010] \right] (0 + (\gamma-1)) + \{G\} \right. \\
&\quad + (\gamma-1)\{P\}^T \left[[A30] - [A3010] \right] \cdot 0 \\
&\quad + \{A\}^T [A4100] \cdot 0 \cdot \{G\} + (\gamma-1)\{P\} \\
&\quad \left. + \{A\}^T [A41000] \{V\} (0 + (\gamma-1)) \right] \\
&= \frac{\Delta t}{2} \left[(\gamma-1)\{V\}^T \left[[A30] - [A3010] \right] \right. \\
&\quad \left. + (\gamma-1)\{A\}^T [A41000] \{V\} \right]
\end{aligned}$$

Factoring common terms

$$[JGP]_e = \frac{\Delta t}{2} (\gamma-1) \{V\}^T \left[[A30] - [A3010] + [A40001] \{A\} \right] \quad (28C)$$

Substitution of (34B) yields

$$[JGP]_e = \frac{\Delta t}{2} (\gamma-1) \{V\}^T \left[[D3MAT] + [A40001] \{A\} \right] \quad (29C)$$

Equation of State

The derivative of $\{FP\}_e$ with respect to $\{R\}$ is

$$\begin{aligned}
[JPR]_e &\equiv \frac{\partial \{FP\}_e}{\partial \{R\}_e} = h_e [A200] \frac{\partial \{P\}}{\partial \{R\}} - h_e [A200] \frac{\partial \{G\}}{\partial \{R\}} \\
&\quad + \frac{1}{2} h_e \gamma (\gamma-1) \frac{\partial}{\partial \{R\}} (\{V\}^T [A3000] \{M\})
\end{aligned}$$

$$\begin{aligned}
&= h_e [A200] \cdot 0 - h_e [A200] \cdot 0 \\
&+ \frac{1}{2} h_e \gamma (\gamma - 1) (\{V\}^T [A3000] \frac{\partial \{M\}}{\partial \{R\}} \\
&+ \{M\} [A3000] \frac{\partial \{V\}}{\partial \{R\}}) \\
&= \frac{1}{2} h_e \gamma (\gamma - 1) (\{V\}^T [A3000] \cdot 0 \\
&- \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{M\}^T [A3000])
\end{aligned}$$

$$[JPR]_e = - \frac{1}{2} h_e \gamma (\gamma - 1) \{A10\}^T \left\{ \frac{M}{R^2} \right\} \{M\}^T [A3000] \quad (30C)$$

The derivative of $\{FP\}_e$ with respect to $\{M\}$ is

$$\begin{aligned}
[JPM]_e &\equiv \frac{\partial \{FP\}_e}{\partial \{M\}} = h_e [A200] \frac{\partial \{P\}}{\partial \{M\}} - h_e [A200] \frac{\partial \{G\}}{\partial \{M\}} \\
&+ \frac{1}{2} h_e \gamma (\gamma - 1) \frac{\partial}{\partial \{M\}} (\{V\}^T [A3000] \{M\}) \\
&= h_e [A200] \cdot 0 - h_e [A200] \cdot 0 \\
&+ \frac{1}{2} h_e \gamma (\gamma - 1) (\{V\}^T [A3000] \frac{\partial \{M\}}{\partial \{M\}} \\
&+ \{M\}^T [A3000] \frac{\partial \{V\}}{\partial \{M\}}) \\
&= \frac{1}{2} h_e \gamma (\gamma - 1) (\{V\}^T [A3000] + \{A10\}^T \left\{ \frac{1}{R} \right\} \{M\}^T [A3000])
\end{aligned}$$

$$[JPM]_e = \frac{1}{2} h_e \gamma (\gamma - 1) (\{V\} + \{A10\}^T \left\{ \frac{1}{R} \right\} \{M\})^T [A3000] \quad (31C)$$

The derivative of $\{FP\}_e$ with respect to $\{G\}$ is

$$\begin{aligned}
 [JPG]_e &= \frac{\partial \{FP\}_e}{\partial \{G\}_e} = h_e[A200] \frac{\partial \{P\}}{\partial \{G\}} - h_e[A200] \frac{\partial \{G\}}{\partial \{G\}} \\
 &\quad + \frac{1}{2} \gamma(\gamma-1) \frac{\partial}{\partial \{G\}} (\{V\}^T [A3000] \{M\}) \\
 &= h_e[A200] \cdot 0 - h_e[A200] \\
 &\quad + \frac{1}{2} \gamma(\gamma-1) (\{V\}^T [A3000] \frac{\partial \{M\}}{\partial \{G\}} + \{M\}^T [A3000] \frac{\partial \{V\}}{\partial \{G\}})
 \end{aligned}$$

APPENDIX D

ORIGINAL ALGORITHM

Presented here are the specific matrix forms of the solution algorithm for the dimensional Euler equations presented by Baker [1982, pp. 41-43].

$$\begin{aligned}
 \{FR\}_e &= [h_e[A200] + v_R^1[A210]] \{R\}_{j+1} \\
 &+ \frac{\Delta t}{2} [A210] \{M\} + v_R^2 \{\bar{V}\}^T [A3011] \{R\} \\
 &+ \{A\}^T ([A41000] \{V\}) \{R\}]_{j+1,j} = \{0\}
 \end{aligned} \tag{1D}$$

$$\begin{aligned}
 \{FM\}_e &= [h_e[A200] + v_M^1[A210]] \{M\}_{j+1} \\
 &+ \frac{\Delta t}{2} [\{V\}^T [A3010] \{M\} + [A210] \{P\} + v_M^2 \{\bar{V}\}^T [A3011] \{M\} \\
 &+ \{A\}^T ([A41000] \{V\}) \{M\}]_{j+1,j} = \{0\}
 \end{aligned} \tag{2D}$$

$$\begin{aligned}
 \{FG\}_e &= [h_e[A200] + v_G^1[A210]] \{G\}_{j+1} \\
 &+ \frac{\Delta t}{2} [\{V\}^T [A3010] (\{G\} + \{P\}) + v_G^2 \{\bar{V}\}^T [A3011] \{G\} \\
 &+ \{A\}^T ([A41000] \{V\}) (\{G\} + \{P\})]_{j+1,j} = \{0\}
 \end{aligned} \tag{3D}$$

$$\begin{aligned}
 \{FP\}_e &= h_e[A200] \{P\} - h_e(\gamma-1) [A200] \{G\} \\
 &- \frac{1}{2} \{V\}^T [A3000] \{M\}] = \{0\}
 \end{aligned} \tag{4D}$$

The elements of the Newton algorithm Jacobian are derived as

$$\begin{aligned}
\frac{\partial \{FR\}}{\partial \{R\}} \equiv [JRR] &= h_e [A200] + v_R^1 [A210] + \frac{\Delta t}{2} \left[v_R^2 (\{\bar{V}\})^T [A3011] \right. \\
&\quad \left. - \frac{\bar{m}}{\rho^2} \{R\}^T [A3110] \right) + \{A\}^T [A41000] (\{V\} \\
&\quad \left. - \frac{\bar{m}}{\rho^2} \{R\}) \right] \quad (5D)
\end{aligned}$$

$$[JRM] = \frac{\Delta t}{2} \left[[A210] + v_R^2 [A211] + \{A\}^T [A3100] \right]^P_{j+1} \quad (6D)$$

$$[JRG] = 0 \quad (7D)$$

$$[JRP] = 0 \quad (8D)$$

$$\begin{aligned}
\frac{\partial \{FM\}}{\partial \{R\}} \equiv [JMR] &= - \frac{\Delta t}{2} \left(\frac{\bar{m}}{\rho^2} \right) \{M\}^T \left[[A3010] + v_M^2 [A3110] \right. \\
&\quad \left. + [A40001] \{A\} \right]^P_{j+1} \quad (9D)
\end{aligned}$$

$$\begin{aligned}
[JMM] &= h_e [A200] + v_M^1 [A210] + \frac{\Delta t}{2} \{V\}^T [A3010] \\
&\quad + v_M^2 [A3011] + [A40001] \{A\} \right]^P_{j+1} \\
&\quad + \frac{\Delta t}{2} \left(\frac{1}{\rho} \right) \{M\}^T \left[[A3010] + v_M^2 [A3110] \right. \\
&\quad \left. + [A40001] \{A\} \right]^P_{j+1} \quad (10D)
\end{aligned}$$

$$[JMG] = [0] \quad (11D)$$

$$[JMP] = \frac{\Delta t}{2} [A210] \quad (12D)$$

$$\begin{aligned} \frac{\partial \{FG\}}{\partial \{R\}} \equiv [JGR] = & -\frac{\Delta t}{2} \left(\frac{\bar{m}}{\rho^2} \right) \left[\{G\}^T ([A3010] + v_G^2 [A3110] \right. \\ & \left. + [A40001] \{A\}) + \{P\}^T [A3010] \right]_{j+1}^P \end{aligned} \quad (13D)$$

$$\begin{aligned} [JGM] = & \frac{\Delta t}{2} \left(\frac{1}{\rho} \right) \left[\{G\}^T ([A3010] + v_G^2 [A3110] + [A40001] \{A\}) \right. \\ & \left. + \{P\}^T [A3010] \right]_{j+1}^P \end{aligned} \quad (14D)$$

$$\begin{aligned} [JGG] = & h_e [A200] + v_G^1 [A210] + \frac{\Delta t}{2} \{V\}^T [A3010] \\ & + v_G^2 [A3011] + [A40001] \{A\} \right]_{j+1}^P \end{aligned} \quad (15D)$$

$$[JGP] = \frac{\Delta t}{2} \{V\}^T [A3010] + [A40001] \{A\} \right]_{j+1}^P \quad (16D)$$

$$\frac{\partial \{FP\}}{\partial \{R\}} \equiv [JPR] = -h_e \left(\frac{\gamma-1}{2} \right) \left[\left(\frac{\bar{m}}{\rho^2} \right) \{M\}^T [A3000] \right]_{j+1}^P \quad (17D)$$

$$[JPM] = h_e \frac{\gamma-1}{2} \left[\{V\}^T [A3000] + \left(\frac{1}{\rho} \right) \{M\}^T [A3000] \right]_{j+1}^P \quad (18D)$$

$$[JPG] = -h_e (\gamma-1) [A200] \quad (19D)$$

$$[JPP] = h_e [A200] \quad (20D)$$

In equations (5D)-(20D) the superscript bar on the terms $\bar{m}$ and $\bar{\rho}$ indicate elemental average values of the given variables.

APPENDIX E

LISTING OF COMPUTER CODE


```

C      NELE1 = NUMBER OF FINITE ELEMENTS IN DISCRETIZATION      X
C      (MAX=200)                                                X
C      NPR   = NUMBER OF MACRO-ELEMENTS IN DISCRETIZATION      X
C      IDS   = SWITCH FOR FORMULATION OF PENALTY CONSTRAINT      X
C      0 -- NONCONSERVATIVE                                     X
C      1 -- CONSERVATIVE                                         X
C                                                                X
C      INPUT FORMAT (15I5)                                       X
C                                                                X
C      ANU1QI = INITIAL-VALUE PENALTY CONSTRAINT MULTIPLIERS    X
C      ANU2QI = BOUNDARY-VALUE PENALTY CONSTRAINT MULTIPLIERS    X
C      SPECIFY VALUES OF ANU1 FOR QF, Q=DENSITY,                X
C      MOMENTUM, AND ENERGY                                     X
C      SPECIFY VALUES OF ANU1 FOR QJ, Q=DENSITY,                X
C      MOMENTUM,ENERGY                                           X
C      SPECIFY VALUES OF ANU2 FOR QF, Q=DENSITY,                X
C      MOMENTUM,ENERGY                                           X
C      SPECIFY VALUES OF ANU2 FOR QJ, Q=DENSITY,                X
C      MOMENTUM, ENERGY                                         X
C                                                                X
C      INPUT FORMAT (5D15.5)                                     X
C                                                                X
CX-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X
C
      WRITE(6,130)
      READ(5,310)K,NELE1,NPR,IDS
      WRITE(6,140)
      WRITE(6,310)K,NELE1,NPR,IDS
      WRITE(6,250)
      READ(5,330)ANU1RF,ANU1MF,ANU1GF
      WRITE(6,330)ANU1RF,ANU1MF,ANU1GF
      WRITE(6,270)
      READ(5,330)ANU1RJ,ANU1MJ,ANU1GJ
      WRITE(6,330)ANU1RJ,ANU1MJ,ANU1GJ
      WRITE(6,260)
      READ(5,330)ANU2RF,ANU2MF,ANU2GF
      WRITE(6,330)ANU2RF,ANU2MF,ANU2GF
      WRITE(6,280)
      READ(5,330)ANU2RJ,ANU2MJ,ANU2GJ
      WRITE(6,330)ANU2RJ,ANU2MJ,ANU2GJ
C
CX-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X
C
      READ DISCRETIZATION PARAMETERS                             X
C                                                                X
C      XP = MACRO-ELEMENT VERTEX NODE COORDINATES              X
C      YP = FLOW AREA AT MACRO-ELEMENT VERTEX NODE             X
C      XM = MACRO-ELEMENT INTERIOR NODE COORDINATE              X
C      YM = FLOW AREA AT MACRO-ELEMENT INTERIOR NODES          X
C      PRG= GEOMETRIC PROGRESSION FACTOR TO DEFINE A            X
C      NON-UNIFORM DISCRETIZATION WITHIN EACH                  X
C      MACRO-ELEMENT                                             X

```

```

C                                     X
C      INPUT FORMAT (5D15.5)         X
C                                     X
C      NEM = NUMBER OF FINITE ELEMENTS TO BE GENERATED WITHIN X
C      EACH MACRO-ELEMENT           X
C                                     X
C      INPUT FORMAT (I5)             X
C                                     X
CX-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X
C
      NNODE=K*NELE1+1
      WRITE(9,100)NNODE
      INODE=NELE1+1
      NBAND=2*K+1
      READ(5,330)(XP(J),J=1,NPR)
      READ(5,330)(YP(J),J=1,NPR)
      READ(5,330)(XM(J),J=2,NPR)
      READ(5,330)(YM(J),J=2,NPR)
      READ(5,330)(PRG(J),J=2,NPR)
      READ(5,310)(NEM(J),J=2,NPR)
      WRITE(6,350)
      WRITE(6,330)(XP(I),I=1,NPR)
      WRITE(6,360)
      WRITE(6,330)(YP(I),I=1,NPR)
      WRITE(6,370)
      WRITE(6,330)(XM(I),I=2,NPR)
      WRITE(6,380)
      WRITE(6,330)(YM(I),I=2,NPR)
      WRITE(6,390)
      WRITE(6,330)(PRG(I),I=2,NPR)
      WRITE(6,400)
      WRITE(6,310)(NEM(I),I=2,NPR)
      NEL1=1
      X(1)=XP(1)
      AREA(1)=YP(1)
      DO 40 IPR=2,NPR
      CALL CSAREA(XP(IPR-1),XM(IPR),XP(IPR),YP(IPR-1),YM(IPR),
      *YP(IPR),C1,C2,C3)
      SUM=1.0D00
      INM=K*NEM(IPR)
      DO 10 I=2,INM
10  SUM=SUM+PRG(IPR)**(I-1)
      RSCALE=(XP(IPR)-XP(IPR-1))/SUM
      NEL1=NEL1+1
      NEL2=NEL1+K*NEM(IPR)
      J=1
      DO 30 I=NEL1,NEL2
      J=J+1
20  X(I)=X(I-1)+RSCALE*PRG(IPR)**(J-2)
30  AREA(I)=C1+X(I)*C2+X(I)*X(I)*C3
40  NEL1=NEL2-1
      IF(K.EQ. 1)GO TO 60

```

```

      DO 50 I=1,INODE
50  X(I)=X(2*I-1)
60  ASTAR=AREA(1)
      DO 70 I=2,NNODE
70  IF(AREA(I) .LT. ASTAR)ASTAR=AREA(I)
      DO 80 I=1,NNODE
80  AREA(I)=AREA(I)/ASTAR
C
CX-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X
C
C      INITIAL CONDITION GENERATION
C
C      ICGEN = INITIAL CONDITION DEFINITION MODE
C
C      INPUT FORMAT (15I5)
C
C      1 -- READ INITIAL CONDITIONS FOR EACH REGION OF
C          DISCRETIZATION
C          NREG  =NUMBER OF REGIONS IN DISCRETIZATION
C              (MAX, 4)
C          TEMPIN=REFERENCE TEMPERATURE
C
C          FORMAT(15,3D15.5)
C
C          ISTART=FIRST NODE IN REGION
C          ISTOP  =LAST NODE IN REGION
C          RHO    =DENSITY IN REGION
C          XMOM   =MOMENTUM IN REGION
C          PRS    =PRESSURE IN REGION
C
C          FORMAT(2I5,3D15.5)
C
C      2 -- CALCULATE INITIAL CONDITIONS GIVEN INLET
C          MASS FLOWRATE AND INLET TEMPERATURE AND
C          PRESSURE
C
C      3 -- CALCULATE INITIAL CONDITIONS GIVEN INLET
C          MACH NO. AND INLET TEMPERATURE AND PRESSURE
C          ISHK  =NODAL LOCATION OF SHOCK
C              FOR SUPERSONIC ON DESIGN USE LAST NODE
C              FOR SUBSONIC ON DESIGN USE THROAT NODE
C          TEMPIN=TEMPERATURE AT INLET
C          PRSIN  =PRESSURE AT INLET
C          XMDOT  =INLET MASS FLOW RATE(ICGEN=2)
C          XMN(1)=INLET FLOW MACH NO.(ICGEN=3)
C
C          FORMAT(15,3D15.5)
C
C      IPEXIT = FLAG FOR EXIT PRESSURE SPECIFICATION(ICGEN=1,2)X
C          0 -- USE EXIT PRESSURE CALCULATED BY
C              I. C. GENERATOR
C          1 -- READ SPECIFIC EXIT PRESSURE

```

```

C
C                               INPUT FORMAT (5D15,5)
C
C
CX-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X
C
      READ(5,310)ICGEN
      WRITE(6,430)
      WRITE(6,310)ICGEN
      GO TO (525,530,535),ICGEN
525  READ(5,470)NREG,TEMPIN
      WRITE(6,490)
      WRITE(6,470)NREG,TEMPIN
      WRITE(6,500)
      DO 540 I=1,NREG
      READ(5,480)ISTART(I),ISTOP(I),RHO,XMOM,PRS
      WRITE(6,480)ISTART(I),ISTOP(I),RHO,XMOM,PRS
      RHOO(ISTART(I))=RHO
      XMOMO(ISTART(I))=XMOM
540  PRSO(ISTART(I))=PRS
      CALL INIT
      IPEXIT=0
      GO TO 550
530  IF(ICGEN .EQ. 2)READ(5,470)ISHK,TEMPIN,PRSIN,XMDOT
      WRITE(6,510)
      WRITE(6,470)ISHK,TEMPIN,PRSIN,XMDOT
      GO TO 545
535  IF(ICGEN .EQ. 3)READ(5,470)ISHK,TEMPIN,PRSIN,XMN(1)
      WRITE(6,520)
      WRITE(6,470)ISHK,TEMPIN,PRSIN,XMN(1)
545  CALL ISNTRF
      READ(5,100)IPEXIT
      WRITE(6,560)
      WRITE(6,100)IPEXIT
550  PEXIT=PRSO(NNODE)
      IF(IPEXIT .EQ. 1)READ(5,330)PEXIT
      PRSO(NNODE)=PEXIT
      NNOD1=NNODE-1
      WRITE(6,340)
      WRITE(6,330)PEXIT
      IF(XMN(NNOD1) .GT. 1.0D 00 .AND. PRSO(NNOD1) .GT. PEXIT)
      * GO TO 90
C
CX-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X
C
C                               BOUNDARY CONDITION DEFINITION
C
C                               SET INLET BOUNDARY CONDITIONS
C
C                               NINBCR = DENSITY
C                               NINBCM = MOMENTUM
C                               NINBCG = ENERGY

```

[illegible]

```

C      IOPT   = FLAG FOR TIME STEP EXTREMIZER                X
C      0 -- EXTREMIZER OFF                                  X
C      1 -- DQMAX FIXED                                      X
C      2 -- DQMAX ADJUSTED BY CONVERGENCE RATE              X
C      NITMAX = MAXIMUM ALLOWED NUMBER OF ITERATIONS AT ANY  X
C      TIME STEP (PRINT ERROR MESSAGE AND EXIT)            X
C      NITLO  = NUMBER OF ITERATIONS FOR INCREASING DQMAX   X
C      NITHI  = NUMBER OF ITERATIONS FOR DECREASING DQMAX   X
C                                                         X
C      INPUT FORMAT (15I5)                                   X
C                                                         X
CX-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X
C
      READ(5,330)CHIO,CHIF,DQMAX,DQFCTU,DQFCTD
      WRITE(6,240)
      WRITE(6,330)CHIO,CHIF,DQMAX,DQFCTU,DQFCTD
      READ(5,330)THETA,DCIMP,EPS,CHIMIN
      WRITE(6,300)
      WRITE(6,330)THETA,DCIMP,EPS,CHIMIN
      READ(5,310)IOPT,NITMAX,NITLO,NITHI
      WRITE(6,290)
      WRITE(6,310)IOPT,NITMAX,NITLO,NITHI
      CHIO=CHIO/TIMZRO
      CHIF=CHIF/TIMZRO
      DCIMP=DCIMP/TIMZRO
      CHIMIN=CHIMIN/TIMZRO
C
CX-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X
C
      DEFINE OUTPUT CONTROLS                                X
C                                                         X
C      IPRMP  = NO. OF INTEGRATION STEPS BETWEEN OUTPUT PRINTS X
C      IPRNT  = PRINT FLAG                                    X
C      0 -- PRINT ALL INTERMEDIATE STEPS                     X
C      1 -- PRINT DATA REFLECTION AND SOLUTION ONLY         X
C      IMPLT  = OUTPUT FLAG FOR MULTI TIME STEP PLOTS       X
C      0 -- OUTPUT FOR FINAL STEP ONLY                       X
C      1 -- OUTPUT FOR INTERMEDIATE STEPS ALSO               X
C                                                         X
C      INPUT FORMAT (15I5)                                   X
C                                                         X
CX-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X-X
C
      READ(5,310)IPRMP,IPRNT,IMPLT
      WRITE(6,320)
      WRITE(6,310)IPRMP,IPRNT,IMPLT
      WRITE(6,410)(X(I),I=1,INODE)
      WRITE(6,420)(AREA(I),I=1,NNODE)
      WRITE(6,440)TITLE,K,NNODE,THETA,DQMAX
      IF(IPRNT .EQ. 0)   WRITE(6,450)
      CALL FESOLV

```

C

```

C      FOLLOWING TERMS REQUIRED FOR USE ON DEC-10
C
C      CLOSE (UNIT=5)
C      CLOSE (UNIT=6)
C      CLOSE (UNIT=9)
      RETURN
90 WRITE(6,460)PEXIT,PRS0(NNOD1),XMN(NNOD1)
      STOP
100 FORMAT(I5)
110 FORMAT(20A4)
120 FORMAT(////,28X,20A4,/,28X,20A4)
130 FORMAT(/,10X,'** INPUT DATA REFLECTION **',/)
140 FORMAT(2X,'K,NELE1,NPR,IDS')
150 FORMAT(/,'NINBCR NINBCM NINBCG NINBCP')
155 FORMAT(/,'NOUBCR NOUBCM NOUBCG NOUBCP')
160 FORMAT(/,2X,'IDVDRM,IDS,IAR,JPM,JRR,JRM')
170 FORMAT(/,1X,'NFI')
180 FORMAT(/,2X,'IFRTQ(I)')
220 FORMAT(/,3X,'RHO1(I)')
230 FORMAT(/,3X,'PHI(I)')
240 FORMAT(/,4X,'TIME INITIAL',4X,'TIME FINAL',6X,'DQMAX',9X,
1      'DQFCTU',9X,'DQFCTD')
250 FORMAT(/,6X,'ANU1RF',10X,'ANU1MF',10X,'ANU1GF')
260 FORMAT(/,6X,'ANU2RF',10X,'ANU2MF',10X,'ANU2GF')
270 FORMAT(/,6X,'ANU1RJ',10X,'ANU1MJ',10X,'ANU1GJ')
280 FORMAT(/,6X,'ANU2RJ',10X,'ANU2MJ',10X,'ANU2GJ')
290 FORMAT(/,1X,'IOPT,NITMAX,NITLO,NITHI')
300 FORMAT(/,5X,'THETA',11X,'DCIMP',12X,'EPS',9X,'CHIMIN')
310 FORMAT(15I5)
320 FORMAT(/,' IPRMP,IPRNT,IMPLT')
330 FORMAT(SD15.5)
340 FORMAT(/,6X,'PEXIT')
350 FORMAT(/,6X,'XP')
360 FORMAT(/,6X,'YP')
370 FORMAT(/,6X,'XH')
380 FORMAT(/,6X,'YM')
390 FORMAT(/,6X,'PRG')
400 FORMAT(/,6X,'NEM')
410 FORMAT(1H1,10X,'AXIAL DISTANCE',/,(1X,SD16.6))
420 FORMAT(/,8X,'C. S. AREA',/,(1X,SD16.6))
430 FORMAT(/,1X,'ICGEN')
440 FORMAT(1H1,28X,20A4,/,
1      35X,'** DEGREE OF FINITE ELEMENT BASIS = ',I2,/,
2      35X,'** NUMBER OF NODES IN DISCRETIZATION = ',I5,/,
3      35X,'** THETA = ',F6.4,/,35X,'** DQMAX = ',D12.4,/)
450 FORMAT(/,5X,'NODAL COORDINATES',/)
460 FORMAT(////,1X,'ONE DIMENSIONAL ASSUMPTIONS VIOLATED.',
1/,1X,'GIVEN EXIT PRESSURE,',D15.6,
2/,1X,'IS LESS THAN THE CALCULATED EXIT PRESSURE,',D15.6,/,/,
3/'WITH SUPERSONIC OUTLET CONDITIONS, MACH NUMBER =',D15.6)
470 FORMAT(I5,3D15.5)
480 FORMAT(2I5,3D15.5)

```

```
490 FORMAT(/2X,'NREG,TEMPIN')
500 FORMAT(/2X,'ISTART,ISTOP,RHO,XMOM,PRS')
510 FORMAT(/2X,'ISHK,TEMPIN,PRSIN,XMDOT')
520 FORMAT(/2X,'ISHK,TEMPIN,PRSIN,XMN(1)')
560 FORMAT(/2X,'IPEXIT')
    END
```

SUBROUTINE FESOLV

C
C
C
C

NEWTON ITERATION (J*DELTAQ = -F) AT TIME STATION J+1

IMPLICIT REAL*8(A-H,O-Z)

COMMON/DATA/ CHIO,CHIF,THETA,DCIMP,EPS,CHIMIN,DQMAX,DQFCTU,

1 DQFCTD,PEXIT,

2 ANU2RF,ANU2MF,ANU2GF,

3 ANU1RF,ANU1MF,ANU1GF,

4 ANU2RJ,ANU2MJ,ANU2GJ,

5 ANU1RJ,ANU1MJ,ANU1GJ

COMMON/ARRAYS/ RH01(201),XMDM1(201),G1(201),PRS1(201),

1 RH00(201),XMDM0(201),G0(201),PRS0(201),

2 QMAT(9),FR(201),FM(201),FP(201),FG(201),RHT(804)

COMMON/MULT/ X(201),XP(10),YP(10),XM(10),YM(10),

1 FRG(10),XMN(201),AREA(201),ISTART(4),ISTOP(4)

COMMON/ARRAYI/ IPRTQ(201),NEM(10),NFI(2)

COMMON/IARRAY/NDOFEL,NTMAT,NTMATR,NELE1,ISRC

1 NINEBCR,NINEBCM,NINEBCG,NINEBCP,

2 NOUBCR,NOUBCM,NOUBCG,NOUBCP

COMMON/IDAT/ NBAND,NNUDE,INODE,K,IPRNT,IMPLOT,XI,NPR,ICGEN,

1 NITMAX,IPRMP,NBAND1,IRMX,IMMX,IPMX,IGMX,IOPT,NITLO,

2 NITHI,H,VMAX,NREG,ISHK

COMMON/SWITCH/IDVDRM,IDS,IAR,JPM,JRR,JRM

COMMON/NONDIM/TZERO,RHOZRO,PZERO,GZERO,CSNDO,TIMZRO,ASTAR,

1 XMZERO,XMDOT0,TEMPIN,PRSIN,XMDOT

COMMON/UBRMAX/ UB10MX,UB11MX

COMMON/MAT/ A200E(9),A201E(9),A210E(9),A211E(9),

1 A3000E(27),A3001E(27),A3010E(27),A3100E(27),

2 A3011E(27),A3110E(27),A3101E(27),A30E(9),

3 A40000(16),A40001(16),A40010(16),A41000(16),

4 Q40001(81),A10E(3)

DIMENSION QRHO(201),UXMDM(201),UPRS(201),UG(201)

DIMENSION QQRHO(201),UUXMDM(201),UUPRS(201),UUG(201)

DIMENSION UVEL(201),TEMP(201)

IPMULT=1000

NDOFEL=K+1

C
C
C
C

OBTAIN INTEGER HYPERMATRIX DATA FROM BLOCK DATA AND

DIVIDE BY COMMON DENOMINATOR

CALL ELMAT

C
C
C

INITIALIZE SOLUTION PARAMETERS

TO=CHIO

TF=CHIF

H=DCIMP

LXYZ=-1

JFIM=0

KFIM=0

```

      NITOTL=0
      IRCHDS=0
      DO 10 I=1,NNODE
      UUXMOM(I)=XMOMO(I)
      UUPRS(I)=FRS0(I)
      UUG(I)=G0(I)
      QQRHO(I)=RH00(I)
      UXMOM(I)=XMOMO(I)
      UFRS(I)=FRS0(I)
      UG(I)=G0(I)
      QRHO(I)=RH00(I)
      RHO1(I)=RH00(I)
      XMOM1(I)=XMOMO(I)
      G1(I)=G0(I)
10  PRS1(I)=FRS0(I)
      NBAND1=NBAND-1
      IF(IPRNT .EQ. 0) WRITE(6,220)
C     IF(IPRMP .GT. 2) WRITE(6,1190)
      TCHI=CHID*TIMZRO
      TDCIMP=DCIMP*TIMZRO
      WRITE(6,200)TCHI,TDCIMP,TIMZRO
C
C     OUTPUT DISCRETE SOLUTION INITIAL CONDITIONS
C
      CALL OUTPUT
      IF(IOPT .EQ. 2)DQMAX=DQMAX/DQFCTU
C
C     SET UP FOR FIRST ITERATION AT TIME STEP J+1
C
20  T1=T0+H
      TIME1=T1*TIMZRO
      HTIME=H*TIMZRO
      JFIM=JFIM+1
      KPIM=KPIM+1
      VMAX=0.0D00
      DO 30 I=1,NNODE
      VEL=XMOMO(I)/RH00(I)
      IF(VMAX .LT. DABS(VEL))VMAX=VEL
      XMOM1(I)=XMOMO(I)
      FRS1(I)=FRS0(I)
      G1(I)=G0(I)
30  RH01(I)=RH00(I)
      IF(VMAX .LT. 0.1D00)VMAX=1.0D00
      IF(IOPT .NE. 2) GO TO 35
      IF(NIT .LT. NITLO)DQMAX=DQMAX*DQFCTU
      IF(NIT .GT. NITHI)DQMAX=DQMAX/DQFCTD
35  NIT=1
C
C     EXECUTE FIRST NEWTON ITERATION TO DETERMINE DELTAQ AT J+1
C
      NITOTL=NITOTL+1
      CALL FANDJ(H,LXYZ,DELRMX,DELMX,DELFMX,DELGMX,T1,NIT)

```

```

        WRITE(6,40)
40  FORMAT(1X)
        IF(IDPT .EQ. 0)GO TO 60
        DELQ=DMAX1(DABS(DELRMX),DABS(DELMX),
        *          DABS(DELGMX),DABS(DELPMX))
        T1=T1-H
        QDEL=DQMAX/DELQ
        H=H*QDEL
        T1=T1+H
        TIME1=T1*TIMZRO
        HTIME=H*TIMZRO
        DELRMX=DELRMX*QDEL
        DELMX=DELMX*QDEL
        DELGMX=DELGMX*QDEL
        DELPMX=DELPX*QDEL
        J=-3
        DO 50 I=1,NNODE
        J=J+4
        RHO1(I)=RHO0(I)+QDEL*RHT(J)
        XMOM1(I)=XMOM0(I)+QDEL*RHT(J+1)
        G1(I)=G0(I)+QDEL*RHT(J+2)
50  PRS1(I)=PRS0(I)+QDEL*RHT(J+3)
60  WRITE(6,190)TIME1,HTIME,NIT,DELRMX,IRMX,DELMX,IMMX,DELGMX,
        *          IGMX,DELPX,IPMX
C
C          EXECUTE REMAINING NEWTON ITERATIONS TO NITMAX OR EPS
C
        DO 80 I=2,NITMAX
        NIT=I
        NITOTL=NITOTL+1
        CALL FANDJ(H,LXYZ,DELRMX,DELMX,DELPX,DELGMX,T1,NIT)
        WRITE(6,190)TIME1,HTIME,NIT,DELRMX,IRMX,DELMX,IMMX,DELGMX,
        *          IGMX,DELPX,IPMX
        IF(DABS(DELRMX) .GT. EPS)      GO TO 85
        IF(DABS(DELMX) .GT. EPS)      GO TO 85
        IF(DABS(DELPX) .GT. EPS)      GO TO 85
        IF(DABS(DELGMX) .GT. EPS)     GO TO 85
        GO TO 90
85  IF(I .EQ. NITMAX)GO TO 240
80  CONTINUE
90  CONTINUE
C
C          NEWTON HAS CONVERGED TO EPS. PRINT OUTPUT IF DESIRED
C
        IF(T1 .GT. TF)      GO TO 100
        IF(DABS(TF-T1) .LT. CHIMIN)  GO TO 100
        IF(JPIM .LT. IPRMP)  GO TO 120
        JPIM=0
100  IF(IPRMP .GT. 2)      WRITE(6,230)
        XMUMAX=0.0
        DO 110 I=1,NNODE
        IF(DABS(XMOM1(I)/RHO1(I)) .GT. XMUMAX) XMUMAX=

```

```

      1DABS(XMOM1(I)/RHO1(I))
110 CONTINUE
      WRITE(6,210)TIME1,HTIME,TIMZRO,NITOTL,KPIM
C
C      COMPUTE NORMS AND OUTPUT DISCRETE SOLUTION
C
      CALL OUTPUT
120 CONTINUE
      IF(T1 .GT. TF)      GO TO 150
      IF(DABS(TF-T1) .LE. CHIMIN)      GO TO 150
      IF(H .LT. (TF-T1))GO TO 130
      H=TF-T1
      IOPT=0
      IMPLOT=1
C
C      SOLUTION STEP COMPLETE. SET UP ARRAYS FOR NEXT TIME STEP
C
130 T0=T1
      DO 140 I=1,NNODE
      UUXMOM(I)=UXMOM(I)
      UUPRS(I)=UPRS(I)
      UUG(I)=UG(I)
      QQRHO(I)=QRHO(I)
      UXMOM(I)=XMOM0(I)
      UPRS(I)=PRSO(I)
      UG(I)=G0(I)
      QRHO(I)=RH00(I)
      XMOM0(I)=XMOM1(I)
      PRSO(I)=PRS1(I)
      G0(I)=G1(I)
140 RH00(I)=RHO1(I)
      GO TO 20
C
C      RETURN TO FIRST ITERATION AT NEXT TIME STEP
C
150 CONTINUE
      IF(IRCHDS .GT. 0) GO TO 180
C
C      IF FINAL TIME IS REACHED, HALVE TIME STEP AND
C      CALCULATE RICHARDSON STEP SOLUTION
C
160 H=H/2.D00
      DO 170 I=1,NNODE
      XMOM0(I)=UXMOM(I)
      PRSO(I)=UPRS(I)
      G0(I)=UG(I)
      RH00(I)=QRHO(I)
      XMOM1(I)=UXMOM(I)
      PRS1(I)=UPRS(I)
      G1(I)=UG(I)
170 RHO1(I)=QRHO(I)
      IRCHDS=IRCHDS+1

```

```
IMPL0T=0
JPIH=0
KPIH=0
NIOTL=0
IPRNF=2
GO TO 20
180 CONTINUE
RETURN
240 WRITE(6,250)KPIH
STOP
190 FORMAT(2X,D10.3,2X,D10.3,I3,4(D15.5,I5))
200 FORMAT(1X,'TIME = ',D10.3,10X,'STEP SIZE = ',D8.3,
110X,'TIME ZERO = ',D8.3)
210 FORMAT(1H1,01X,'TIME = ',D10.3,6X,'STEP SIZE = ',D8.3,
16X,'TIME ZERO = ',D8.3,6X,'ITERATIONS = ',I3,6X,'STEPS = ',
2I3)
220 FORMAT(1H1)
230 FORMAT(///)
250 FORMAT(////////10X,'MAXIMUM NUMBER OF ITERATION EXCEEDED AT',
* 'TIME STEP NO. ',I5)
END
```

```

SUBROUTINE FANDJ(DCHI,LXYZ,DELRMX,DELMX,DELPX,DELGX,
1              T1,NIT)
C
C      FORMATION AND ASSEMBLY OF NEWTON ALGORITHM RIGHT HAND
C      SIDE(-F) AND BLOCK BANDED JACOBIAN(J)
C
      IMPLICIT REAL*8(A-H,D-Z)
      COMMON/DATA/ CHID,CHIF,THETA,DCIMP,EPS,CHIMIN,DQMAX,DQFCTU,
1      DQFCTD,PEXIT,
2      ANU2RF,ANU2MF,ANU2GF,
3      ANU1RF,ANU1MF,ANU1GF,
4      ANU2RJ,ANU2MJ,ANU2GJ,
5      ANU1RJ,ANU1MJ,ANU1GJ
      COMMON/ARRAYS/ RHO1(201),XMOM1(201),G1(201),PRS1(201),
1      RHO0(201),XMOM0(201),G0(201),PRS0(201),
2      QMAT(9),FR(201),FM(201),FP(201),FG(201),RHT(804)
      COMMON/MULT/ X(201),XF(10),YP(10),XM(10),YM(10),
1      PRG(10),XMN(201),AREA(201),ISTART(4),ISTOP(4)
      COMMON/ARRAYI/ IPRTQ(201),NEM(10),NFI(2)
      COMMON/IARRAY/ NDOFEL,NTMAT,NTMATR,NELE1,ISRC
1      NINBCR,NINBCM,NINBCG,NINBCF,
2      NOUBCR,NOUBCM,NOUBCG,NOUBCF
      COMMON/IDAT/ NBAND,NNODE,INODE,K,IPRNT,IMPLOT,XI,NPR,ICGEN,
1      NITMAX,IPRMP,NBAND1,IRMX,IMMX,IPMX,IGMX,IOPT,NITLO,
2      NITHI,H,UMAX,NREG,ISHK
      COMMON/SWITCH/ IDVDRM,IDS,IAR,JPM,JRR,JRM
      COMMON/NONDIM/ TZERO,RHOZRO,PZERO,GZERO,CSNDO,TIMZRO,ASTAR,
1      XMZERO,XMDOTO,TEMPIN,PRSIN,XMDOT
      COMMON/UBRMAX/ UB10MX,UB11MX
      COMMON/MAT/ A200E(9),A201E(9),A210E(9),A211E(9),
1      A300E(27),A301E(27),A310E(27),A311E(27),
2      A301E(27),A311E(27),A310E(27),A30E(9),
3      A4000(16),A4001(16),A40010(16),A41000(16),
4      Q40001(81),A10E(3)
      DIMENSION QRHO(201),UXMOM(201),UPRS(201),UG(201)
      DIMENSION QQRHO(201),UUXMOM(201),UUPRS(201),UUG(201)
      DIMENSION UVEL(201),TEMP(201)
      DIMENSION DSMAT(8844),INT(804),SMAT(18492)
      DIMENSION SJRR(1005),SJMM(1005),SJGG(1005),SJPP(1005)
      DIMENSION SJRM(1005),SJMR(1005),SJMP(1005)
      DIMENSION SJGR(1005),SJGM(1005),SJGP(1005)
      DIMENSION SJPR(1005),SJPM(1005),SJPG(1005)
      DIMENSION XHOMJ1(3),XHOMJ0(3),ALGARE(3)
      DIMENSION AVELJ1(3),AVELJ0(3)
      DIMENSION SXMJ0(3),SXMJ1(3)
      DIMENSION PRSJ0(3),PRSJ1(3),VELJ0(3),VELJ1(3)
      DIMENSION RHOEJ0(3),RHOEJ1(3),GELJ0(3),GELJ1(3)
      DIMENSION AR401(27)
      DIMENSION D2MAT(9),D3MAT(27),X210E(9)
      DIMENSION TFR(3),XFR(3),TFM(3),XFM(3),TFG(3),XFG(3),EFF(3)
      DIMENSION EJRR(9),EJRM(9),EJMR(9),EJMM(9),EJMP(9)
      DIMENSION EJGR(9),EJGM(9),EJGG(9),EJGP(9)

```

```

DIMENSION EJPR(9),EJPM(9),EJPG(9),EJPP(9)
DIMENSION DFRJO(9),DFRJ1(9),DFMJ0(9),DFMJ1(9),DFF(9)
DIMENSION DFGJOP(9),DFGJ1P(9),DFGJOG(9),DFGJ1G(9)
DIMENSION XJRR(9),XJRM(9),XJMR(9),XJMMV(9),XJMMH(9)
DIMENSION XJGG(9),XJGR(9),XJGM(9)

```

```

C
C      SWITCHES FOR CHANGING FORMULATION OF JACOBIAN
C
C      IDVDRM = SWITCH FOR FORMULATION OF DV/DR AND DV/DM
C              0 -- ORIGINAL
C              1 -- REVISED
C      IAR      = SWITCH FOR INCLUSION OF AREA TERMS IN JGP,
C              JGR, AND JGM
C              0 -- NOT INCLUDED
C              1 -- INCLUDED
C      JPM      = SWITCH FOR INCLUSION OF MOMENTUM TERM IN JPM
C              0 -- NOT INCLUDED
C              1 -- INCLUDED
C      JRR      SWITCH FOR INCLUSION OF SPATIAL DERIVATIVE
C              TERMS IN JRR
C              0 -- NOT INCLUDE
C              1 -- INCLUDED

```

```

C      SET JACOBIAN PARAMETERS AT CORRECT VALUES
C
C

```

```

IDVDRM=1
IAR=1
JPM=1
JRR=1

```

```

C
C      SET UP ASSEMBLY PARAMETERS
C

```

```

DUDRM=FLOAT(IDVDRM)
DCHETA=DCHI*THETA
DCOMTA=DCHI*(1.0D00-THETA)
DELRMX=0.0D00
DELMX=0.0D00
DELPX=0.0D00
DELGX=0.0D00
NROWB=NBAND*NNODE
DO 10 I=1,NTMAT
  U3MAT(I)=A30E(I)-A3010E(I)
10 CONTINUE

```

```

C
C      ZERO OUT PREVIOUS DATA
C

```

```

DO 20 I=1,NROWB
  SJRR(I)=0.0D00
  SJMM(I)=0.0D00
  SJGG(I)=0.0D00
  SJRM(I)=0.0D00

```

```

      SJMR(I)=0.0D00
      SJMP(I)=0.0D00
      SJGR(I)=0.0D00
      SJGM(I)=0.0D00
      SJGP(I)=0.0D00
      SJPR(I)=0.0D00
      SJPM(I)=0.0D00
      SJPG(I)=0.0D00
20  SJPP(I)=0.0D00
      DO 30 I=1,NNODE
      FM(I)=0.0D00
      FP(I)=0.0D00
      FG(I)=0.0D00
30  FR(I)=0.0D00

C
C      START BASIC ELEMENT LOOP TO FORM -F AND J
C
      DO 150 IEL=1,NELE1
      JKOUNT=(IEL-1)*K
      DELTXE=X(IEL+1)-X(IEL)
      BARM=0.0D00
      BARR=0.0D00
      BM2=0.0D00
      B1R=0.0D00

C
C      INITIALIZE ELEMENT WORKING ARRAYS
C
      DO 40 K1=1,NDOFEL
      JK=JKOUNT+K1
      RHOEJO(K1)=RHOO(JK)
      RHOEJ1(K1)=RH01(JK)
      IF(K .EQ. 1) BARR=BARR+0.5D00*RHOEJ1(K1)*(1.D00-DVDRM)+
1  A10E(K1)/RHOEJ1(K1)*DVDRM
      IF(K .EQ. 2) BARR=BARR+A10E(K1)*RHOEJ1(K1)*(1.D00-DVDRM)+
1  A10E(K1)/RHOEJ1(K1)*DVDRM
      XMOMJO(K1)=XMOMO(JK)
      XMOMJ1(K1)=XMOM1(JK)
      IF(K .EQ. 1) BARM=BARM+0.5D00*XMOMJ1(K1)*(1.D00-DVDRM)+
1  A10E(K1)*XMOMJ1(K1)*DVDRM/(RHOEJ1(K1)*RHOEJ1(K1))
      IF(K .EQ. 2) BARM=BARM+A10E(K1)*XMOMJ1(K1)*(1.D00-DVDRM)+
1  A10E(K1)*XMOMJ1(K1)*DVDRM/(RHOEJ1(K1)*RHOEJ1(K1))
      VELJO(K1)=XMOMO(JK)/RHOO(JK)
      VELJ1(K1)=XMOM1(JK)/RH01(JK)
      ALGARE(K1)=DLOG(AREA(JK))
      GELJO(K1)=G0(JK)
      GELJ1(K1)=G1(JK)
      PRSJO(K1)=PRS0(JK)
      PRSJ1(K1)=PRS1(JK)
      SXMJ1(K1)=DSIGN(1.0D00,XMOMJ1(K1))
      SXMJO(K1)=DSIGN(1.0D00,XMOMJO(K1))
40  CONTINUE
      BAR2=BARR*BARR

```

```

      BM2=BARM/BAR2*(1.D00-DVDRM)+BARM*DVDRM
      B1R=(1.D00-DVDRM)/BARR+BARR*DVDRM
      SEM2=DSIGN(1.0D00,BM2)
      SB1R=DSIGN(1.0D00,B1R)
      UBRFC1=0.25D00
      IXI=1
      DO 50 K1=1,NDOFEL
      AVELJ0(K1)=DABS(VELJ0(K1)/VMAX)
      IF(AVELJ0(K1).LT. 0.1D00) IXI=0
      AVELJ1(K1)=DABS(VELJ1(K1)/VMAX)
      IF(AVELJ1(K1).LT. 0.1D00) IXI=0
50 CONTINUE
C
C      PERFORM HYPERMATRIX MULTIPLICATION [LOG(AREA)]*CA41000]
C
      L=0
      IF(K.EQ. 2) GO TO 70
      DO 60 I=1,NTMAT
      AR401(I)=0.0D00
      DO 60 J=1,NDOFEL
      L=L+1
      AR401(I)=AR401(I)+A40001(L)*ALGARE(J)
60 CONTINUE
      GO TO 90
70 DO 80 I=1,NTMAT
      AR401(I)=0.0D00
      DO 80 J=1,NDOFEL
      L=L+1
      AR401(I)=AR401(I)+Q40001(L)*ALGARE(J)
80 CONTINUE
90 CONTINUE
      IF(IEL.EQ.NELE1)D3MAT(NTMAT)=D3MAT(NTMAT)+1.0D00
C
C      ZERO ELEMENT ASSEMBLY MATRICES
C
      DO 100 I=1,NTMATR
      IK=(I-1)*NDOFEL
      DFRJ0(I)=0.0D00
      DFRJ1(I)=0.0D00
      DFMJ0(I)=0.0D00
      DFMJ1(I)=0.0D00
      DFGJ0P(I)=0.0D00
      DFGJ1P(I)=0.0D00
      DFGJ0G(I)=0.0D00
      DFGJ1G(I)=0.0D00
      DFP(I)=0.0D00
      XJRR(I)=0.0D00
      XJRM(I)=0.0D00
      XJHR(I)=0.0D00
      XJMMV(I)=0.0D00
      XJMMH(I)=0.0D00
      EUPM(I)=0.0D00

```

```

EJFR(I)=0.0D00
EJGF(I)=0.0D00
XJGG(I)=0.0D00
XJGR(I)=0.0D00
XJGM(I)=0.0D00
D2MAT(I)=0.0D00
DO 100 J=1,NDOFEL
  K1=IK+J

```

C
C
C

PERFORM PREMULIPLICATION OF [A3---] MATRICES

```

D2MAT(I)=D2MAT(I)+D3MAT(K1)
DFRJ0(I)=DFRJ0(I)+ANU2RF*(A3011E(K1)*AVELJ0(J)+
*           IDS*A3110E(K1)*VELJ0(J))+
*           AR401(K1)*VELJ0(J)
DFRJ1(I)=DFRJ1(I)+ANU2RF*(A3011E(K1)*AVELJ1(J)+
*           IDS*A3110E(K1)*VELJ1(J))+
*           AR401(K1)*VELJ1(J)
DFMJ0(I)=DFMJ0(I)+ANU2MF*(A3011E(K1)*AVELJ0(J)+
*           SXMJ0(J)*IDS*A3110E(K1)*VELJ0(J))+
*           (D3MAT(K1)+AR401(K1))*VELJ0(J)
DFMJ1(I)=DFMJ1(I)+ANU2MF*(A3011E(K1)*AVELJ1(J)+
*           SXMJ1(J)*IDS*A3110E(K1)*VELJ1(J))+
*           (D3MAT(K1)+AR401(K1))*VELJ1(J)
DFGJ0P(I)=DFGJ0P(I)+(D3MAT(K1)+AR401(K1))*VELJ0(J)
DFGJ1P(I)=DFGJ1P(I)+(D3MAT(K1)+AR401(K1))*VELJ1(J)
DFGJ0G(I)=DFGJ0G(I)+ANU2GF*(A3011E(K1)*AVELJ0(J)+
*           IDS*A3110E(K1)*VELJ0(J))+
*           (D3MAT(K1)+AR401(K1))*VELJ0(J)
DFGJ1G(I)=DFGJ1G(I)+ANU2GF*(A3011E(K1)*AVELJ1(J)+
*           IDS*A3110E(K1)*VELJ1(J))+
*           (D3MAT(K1)+AR401(K1))*VELJ1(J)
DFP(I)=DFP(I)+0.28D00*DELTXE*VELJ1(J)*A3000E(K1)
XJRK(I)=XJRK(I)+(ANU2RJ*(AVELJ1(J)*A3011E(K1)+
*           IDS*VELJ1(J)*A3110E(K1))+AR401(K1)*VELJ1(J)-
*           BM2*RH0EJ1(J)*((SBM2*A3110E(K1)+
*           IDS*A3011E(K1))*ANU2RJ+AR401(K1)))*JRR
XJRM(I)=XJRM(I)+A3100E(K1)*ALGARE(J)
*           +(1-IDS)*B1R*SB1R*A3110E(K1)*RH0EJ1(J)*ANU2RJ
XJMR(I)=XJMR(I)-BM2*((D3MAT(K1)+AR401(K1))*XMOMJ1(J)+
*           ANU2MJ*(SBM2*A3110E(K1)*XMOMJ1(J)+
*           IDS*A3011E(K1)*DABS(XMOMJ1(J))))
XJMMV(I)=XJMMV(I)+(D3MAT(K1)+AR401(K1))*VELJ1(J)+
*           ANU2MJ*(IDS*A3110E(K1)*VELJ1(J)+
*           A3011E(K1)*AVELJ1(J))
XJMMM(I)=XJMMM(I)+B1R*((D3MAT(K1)+AR401(K1))*XMOMJ1(J)+
*           ANU2MJ*(SB1R*A3110E(K1)*XMOMJ1(J)+
*           IDS*A3011E(K1)*DABS(XMOMJ1(J))))
EJPM(I)=EJPM(I)+DELTXE*0.28D00*(VELJ1(J)+
*           JPM*B1R*XMOMJ1(J))*A3000E(K1)
EJPR(I)=EJPR(I)-DELTXE*0.28D00*BM2*XMOMJ1(J)*A3000E(K1)
EJGP(I)=EJGP(I)+DCHETA*0.4D00*VELJ1(J)*(D3MAT(K1)+

```

```

*       IAR*AR401(K1))
XJGG(I)=XJGG(I)+(D3MAT(K1)+AR401(K1))*VELJ1(J)+
*       ANU2GJ*(IDS*A3110E(K1)*VELJ1(J)+A3011E(K1)*AVELJ1(J))
XJGR(I)=XJGR(I)+(D3MAT(K1)+AR401(K1)+ANU2GJ*(A3110E(K1)*SBM2+
*       IDS*A3011E(J)))*GELJ1(J)+
*       0.4D00*PRSJ1(J)*(D3MAT(K1)+IAR*AR401(K1))
XJGM(I)=XJGM(I)+(D3MAT(K1)+AR401(K1)+ANU2GJ*(A3110E(K1)*SB1R+
*       IDS*A3011E(J)))*GELJ1(J)+
*       0.4D00*PRSJ1(J)*(D3MAT(K1)+IAR*AR401(K1))
100 CONTINUE
DO 110 I=1,NTMATR
X210E(I)=-D2MAT(I)
110 CONTINUE
DO 120 I=1,NDOFEL

C
C       ZERO OUT ELEMENT ASSEMBLY MATRICES
C
TFR(I)=0.0D00
XFR(I)=0.0D00
TFM(I)=0.0D00
XFM(I)=0.0D00
TFG(I)=0.0D00
XFG(I)=0.0D00
EFP(I)=0.0D00
DO 120 J=1,NDOFEL
K1=(J-1)*NDOFEL+I
JK=JKOUNT+J

C
C       PERFORM POST MULTIPLICATION OF [A2--]
C       AND PREMULTIPLIED [A3---] MATRICES
C
TFR(I)=TFR(I)-DELTXE*(A200E(K1)+IXI*ANU1RF*X210E(K1))*
*       (RH01(JK)-RH00(JK))
XFR(I)=XFR(I)-DCHETA*(D2MAT(K1)*(XMM1(JK)+XMM0(JK))+
*       DFRJ1(K1)*RH01(JK)+DFRJ0(K1)*RH00(JK))
TFM(I)=TFM(I)-DELTXE*(A200E(K1)+
*       IXI*ANU1MF*X210E(K1))*(XMM1(JK)-XMM0(JK))
XFM(I)=XFM(I)-DCHETA*(D2MAT(K1)*(PRSJ1(J)+PRSJ0(J))/1.4D00+
*       DFMJ1(K1)*XMM1(JK)+DFMJ0(K1)*XMM0(JK))
TFG(I)=TFG(I)-DELTXE*(A200E(K1)+
*       IXI*ANU1GF*X210E(K1))*(G1(JK)-G0(JK))
XFG(I)=XFG(I)-DCHETA*(DFGJ0G(K1)*G0(JK)+DFGJ1G(K1)*G1(JK)+
*       (DFGJ0F(K1)*PRSJ0(J)+DFGJ1F(K1)*PRSJ1(J))*0.4D00)
EFP(I)=EFP(I)-DELTXE*A200E(K1)*PRS1(JK)+
*       DELTXE*G1(JK)*A200E(K1)-DFP(K1)*XMM1(JK)
EJRR(K1)=DELTXE*(A200E(K1)+IXI*ANU1RJ*X210E(K1))+
*       DCHETA*XJRR(K1)
EJRM(K1)=DCHETA*(D2MAT(K1)+IDS*ANU2RJ*A211E(K1)+XJRM(K1))
EJMP(K1)=DCHETA*D2MAT(K1)/1.4D00
EJMR(K1)=DCHETA*XJMR(K1)
EJMM(K1)=DELTXE*(A200E(K1)+IXI*ANU1MF*X210E(K1))+
*       DCHETA*(XJMMV(K1)+XJMMM(K1))

```

```

EJFF(K1)=DELTXE*A200E(K1)
EJFG(K1)=-DELTXE*A200E(K1)
EJGG(K1)=DELTXE*(A200E(K1)+IXI*ANU1GJ*X210E(K1))+
*      DCHETA*XJGG(K1)
EJGR(K1)=-DCHETA*BM2*XJGR(K1)
EJGM(K1)=DCHETA*B1R*XJGM(K1)
120 CONTINUE
DO 130 I=1,NDOFEL

C
C      ASSEMBLE BLOCK BANDED GLOBAL JACOBIAN
C
      JI=JKOUNT+I
      KISS=JI*NBAND1-K
      DO 130 J=1,NDOFEL
      IS=KISS+JKOUNT+J
      IE=I+(J-1)*NDOFEL
      SJRR(IS)=SJRR(IS)+EJRR(IE)
      SJMM(IS)=SJMM(IS)+EJMM(IE)
      SJGG(IS)=SJGG(IS)+EJGG(IE)
      SJRM(IS)=SJRM(IS)+EJRM(IE)
      SJMR(IS)=SJMR(IS)+EJMR(IE)
      SJMP(IS)=SJMP(IS)+EJMP(IE)
      SJGR(IS)=SJGR(IS)+EJGR(IE)
      SJGM(IS)=SJGM(IS)+EJGM(IE)
      SJGP(IS)=SJGP(IS)+EJGP(IE)
      SJPR(IS)=SJPR(IS)+EJPR(IE)
      SJPM(IS)=SJPM(IS)+EJPM(IE)
      SJPG(IS)=SJPG(IS)+EJPG(IE)
      SJPP(IS)=SJPP(IS)+EJPP(IE)
130 CONTINUE
      DO 140 I=1,NDOFEL

C
C      ASSEMBLE BLOCK FORMATION OF RHS
C
      JI=JKOUNT+I
      FM(JI)=FM(JI)+TFM(I)+XFM(I)
      FP(JI)=FP(JI)+EFP(I)
      FG(JI)=FG(JI)+TFG(I)+XFG(I)
      FR(JI)=FR(JI)+TFR(I)+XFR(I)
140 CONTINUE
      D3MAT(1)=-A3010E(1)
150 CONTINUE

C
C      ASSEMBLY OF BLOCK MATRICES INTO GLOBAL MATRICES J AND -F
C      AND APPLICATION OF DIRICHLET BOUNDARY CONDITIONS
C
      IF (NINBCR .EQ. 0) GO TO 170

C
C      SET DIRICHLET CONDITONS ON DENSITY
C      AT THE INLET AS SPECIFIED
C
      DO 160 I=1,NBAND

```

```

      SJRM(I)=0.0D 00
      IF (I .EQ. (K+1)) GO TO 160
      SJRR(I)=0.0D 00
160  CONTINUE
      FR(1)=0.0D 00
170  IF (NINBCM .EQ. 0) GO TO 171
C
C      SET DIRICHLET CONDITONS ON MOMENTUM
C      AT THE INLET AS SPECIFIED
C
      DO 161 I=1,NBAND
      SJMR(I)=0.0D 00
      SJMP(I)=0.0D 00
      IF (I .EQ. (K+1)) GO TO 161
      SJMM(I)=0.0D 00
161  CONTINUE
      FM(1)=0.0D 00
171  IF (NINBCG .EQ. 0) GO TO 172
C
C      SET DIRICHLET CONDITONS ON ENERGY
C      AT THE INLET AS SPECIFIED
C
      DO 162 I=1,NBAND
      SJGR(I)=0.0D 00
      SJGM(I)=0.0D 00
      SJGP(I)=0.0D 00
      IF (I .EQ. (K+1)) GO TO 162
      SJGG(I)=0.0D 00
162  CONTINUE
      FG(1)=0.0D 00
172  IF (NINBCP .EQ. 0) GO TO 173
C
C      SET DIRICHLET CONDITONS ON PRESSURE
C      AT THE INLET AS SPECIFIED
C
      DO 163 I=1,NBAND
      SJPR(I)=0.0D 00
      SJPM(I)=0.0D 00
      SJPG(I)=0.0D 00
      IF (I .EQ. (K+1)) GO TO 163
C      SJPP(I)=0.0D 00
163  CONTINUE
      FP(1)=0.0D 00
173  IF (NOUBCD .EQ. 0) GO TO 190
C
C      SET DIRICHLET CONDITIONS ON DENSITY
C      AT THE OUTLET AS SPECIFIED
C
      DO 180 I=1,NBAND
      NBD=NROWB-I+1
      SJRM(NBD)=0.0D 00
      IF (I .EQ. (K+1)) GO TO 180

```

```

      SJRR(NBD)=0.0D 00
180  CONTINUE
      FR(NNODE)=0.0D 00
190  IF(NOUSCM .EQ. 0) GO TO 191
      DO 181 I=1,NBAND
        NBD=NROWB-I+1
        SJMR(NBD)=0.0D00
        SJMP(NBD)=0.0D00
        IF(I .EQ. (K+1)) GO TO 181
        SJMM(NBD)=0.0D00
181  CONTINUE
      FM(NNODE)=0.0D00
191  IF(NOURCG .EQ. 0) GO TO 192
      DO 182 I=1,NBAND
        NBD=NROWB-I+1
        SJGR(NBD)=0.0D00
        SJGM(NBD)=0.0D00
        SJMP(NBD)=0.0D00
        IF(I .EQ. (K+1)) GO TO 182
        SJGG(NBD)=0.0D00
182  CONTINUE
      FG(NNODE)=0.0D00
192  IF(NOURCP .EQ. 0) GO TO 193
C
C      SET DIRICHLET CONDITIONS ON PRESSURE
C      AT OUTLET AS SPECIFIED
C
      DO 183 I=1,NBAND
        NBD=NROWB-I+1
        SJPR(NBD)=0.0D 00
        SJPM(NBD)=0.0D 00
        SJPG(NBD)=0.0D 00
        IF (I .EQ. (K+1)) GO TO 183
        SJPP(NBD)=0.0D 00
183  CONTINUE
      FP(NNODE)=0.0D 00
193  NNODIT=4*NNODE
      NBNDTT=15
      IF(K .EQ. 2) NBNDTT=23
      NBNDTH=(NBNDTT-1)/2
      NROWBT=NBNDTT*NNODIT
C
C      ZERO OUT GLOBAL JACOBIAN MATRIX
C
      DO 210 I=1,NROWBT
210  SMAT(I)=0.0D00
C
C      ASSEMBLE GLOBAL JACOBIAN MATRIX
C
      DO 220 I=1,NNODE
        K1=4*NBNDTT*(I-1)

```

```

      I1=NBAND*(I-1)
      DO 220 J=1,NBAND
        J1=I1+J
        K1=K1+4
        SMAT(K1)=SJRR(J1)
        SMAT(K1+1)=SJRM(J1)
        SMAT(K1+NBNDTT-1)=SJMR(J1)
        SMAT(K1+NBNDTT)=SJMM(J1)
        SMAT(K1+NBNDTT+2)=SJMP(J1)
        SMAT(K1+2*NBNDTT-2)=SJGR(J1)
        SMAT(K1+2*NBNDTT-1)=SJGM(J1)
        SMAT(K1+2*NBNDTT)=SJGG(J1)
        SMAT(K1+2*NBNDTT+1)=SJGP(J1)
        SMAT(K1+3*NBNDTT-3)=SJPR(J1)
        SMAT(K1+3*NBNDTT-2)=SJPM(J1)
        SMAT(K1+3*NBNDTT-1)=SJPG(J1)
        SMAT(K1+3*NBNDTT)=SJPP(J1)
220  CONTINUE
C
C      ASSEMBLE GLOBAL RHS (-F)
C
      K1=-3
      DO 230 I=1,NNODE
        K1=K1+4
        RHT(K1)=FR(I)
        RHT(K1+1)=FM(I)
        RHT(K1+2)=FG(I)
        RHT(K1+3)=FP(I)
230  CONTINUE
C
C      WRITE STATEMENTS USED FOR DEBUGGING
C
C      IF(NIT.NE.2)GO TO 260
C      WRITE(6,240)(FR(L),L=1,NNODE)
C      WRITE(6,240)(FM(L),L=1,NNODE)
C      WRITE(6,240)(FP(L),L=1,NNODE)
C      WRITE(6,240)(FG(L),L=1,NNODE)
240  FORMAT(/,(2X,10D12.3))
C      WRITE(6,240)SJRR
C      WRITE(6,240)SJMM
C      WRITE(6,240)SJGG
C      WRITE(6,240)SJPP
C      WRITE(6,240)SJRM
C      WRITE(6,240)SJMR
C      WRITE(6,240)SJMP
C      WRITE(6,240)SJGR
C      WRITE(6,240)SJGM
C      WRITE(6,240)SJGP
C      WRITE(6,240)SJPR
C      WRITE(6,240)SJPM
C      WRITE(6,240)SJPG
C      WRITE(6,250)SMATT

```

```

250 FORMAT(///,(1X,15D8.1))
260 CONTINUE
C
C      ASSEMBLY REARRANGEMENTS COMPLETE.
C      CALL SOLVER FOR LINEAR ALGEBRAIC EQUATION SYSTEM
C
CALL BANDET(DSMAT,INT,SMAT,NNODIT,NNODIT,NBNDTH,NBNDTT)
CALL BANDSL(DSMAT,INT,RHT,SMAT,NNODIT,NNODIT,NBNDTH,NBNDTT)
C
C      FIND VALUE AND NODAL LOCATION OF MAXIMUM DELTAQ(I)
C
      J=-3
      DO 270 I=1,NNODE
      J=J+4
      IF(DABS(RHT(J)) .GT. DABS(DELRMX))      DELRMX=RHT(J)
      IF(DABS(RHT(J)) .EQ. DABS(DELRMX))      IRMX=I
      IF(DABS(RHT(J+1)) .GT. DABS(DELMX))      DELMX=RHT(J+1)
      IF(DABS(RHT(J+1)) .EQ. DABS(DELMX))      IMX=I
      IF(DABS(RHT(J+3)) .GT. DABS(DELPMX))     DELPMX=RHT(J+3)
      IF(DABS(RHT(J+3)) .EQ. DABS(DELPMX))     IPMX=I
      IF(DABS(RHT(J+2)) .GT. DABS(DELGMX))     DELGMX=RHT(J+2)
      IF(DABS(RHT(J+2)) .EQ. DABS(DELGMX))     IGMX=I
C
C      CALCULATE VALUES OF QI(P+1,J+1)
C
      XMOM1(I)=XMOM1(I)+RHT(J+1)
      PRS1(I)=PRS1(I)+RHT(J+3)
      G1(I)=G1(I)+RHT(J+2)
270 RHQ1(I)=RHQ1(I)+RHT(J)
C      PRS1(NNODE)=PEXIT
280 CONTINUE
      RETURN
      END

```

SUBROUTINE ELMAT

GENERATION OF FINITE ELEMENT HYPER-MATRIX REAL DATA FROM
INTEGER ARRAYS STORED IN BLOCK DATA

```

C
C
C
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/AL/A3000L(8),A3011L(8),A3001L(8),A3101L(8),A3010L(8)
1      ,A3100L(8),A3110L(8),A3003L(8)
2      ,P40000(16),P40001(16),P40010(16),P41000(16)
      COMMON/AQ/A3000Q(27),A3011Q(27),A3001Q(27),A3010Q(27)
1      ,A3100Q(27),A3110Q(27),A3003Q(27)
2      ,S40001(81)
      COMMON/IARRAY/NDOFEL,NTMAT,NTMATR,NELE1,ISRC
1      ,NINBCR,NINBCM,NINBCG,NINBCP,
2      ,NOUBCR,NOUBCM,NOUBCG,NOUBCP
      COMMON/IOAT/ NBAND,NNODE,INODE,K,IPRNT,IMPLOT,XI,NPR,ICGEN,
1      ,NITMAX,IPRMP,NBAND1,IRMX,IMMX,IPMX,IGMX,IOPT,NITLO,
2      ,NITHI,H,UMAX,NREG,ISHK
      COMMON/MAT/ A200E(9),A201E(9),A210E(9),A211E(9),
1      ,A3000E(27),A3001E(27),A3010E(27),A3100E(27),
2      ,A3011E(27),A3110E(27),A3101E(27),A30E(9),
3      ,A40000(16),A40001(16),A40010(16),A41000(16),
4      ,Q40001(81),A10E(3)
      DIMENSION DMAT(27),AMAT(27)
      NS1=K+1
      NS2=NS1*NS1
      NTMATR=NDOFEL*NDOFEL
      NTMAT=NTMATR*NDOFEL
      A30E(1)=-1.0D00
      DO 10 I=2,NTMAT
10  A30E(I)=0.0D00
C
C
C      SUBROUTINE BRANCHES ACCORDING TO DEGREE OF BASIS
C
C      IF(K .EQ. 2)GO TO 170
C
C      FOR LINEAR BASIS FORMULATION
C
C      DO 20 I=1,NDOFEL
20  A10E(I)=0.5D00
C
C      DO 30 I=1,NTMAT
30  AMAT(I)=A3000L(I)/12.D00
C
C      SUBROUTINE SORT REARRANGES STORAGE FOR CONVENIENT
C      MULTIPLICATION LAYER
C
C      CALL SORT(AMAT,DMAT,NS1,NS2)
      DO 40 I=1,NTMAT
40  A3000E(I)=DMAT(I)
C

```

```

      DO 50 I=1,NTMAT
50  AMAT(I)=A3011L(I)/2.D00
      CALL SORT(AMAT,DMAT,NS1,NS2)
      DO 60 I=1,NTMAT
60  A3011E(I)=DMAT(I)
C
      DO 70 I=1,NTMAT
70  AMAT(I)=A3001L(I)/6.D00
      CALL SORT(AMAT,DMAT,NS1,NS2)
      DO 80 I=1,NTMAT
80  A3001E(I)=DMAT(I)
C
      DO 90 I=1,NTMAT
90  AMAT(I)=A3101L(I)/2.D00
      CALL SORT(AMAT,DMAT,NS1,NS2)
      DO 100 I=1,NTMAT
100 A3101E(I)=DMAT(I)
C
      DO 110 I=1,NTMAT
110 AMAT(I)=A3010L(I)/6.D00
      CALL SORT(AMAT,DMAT,NS1,NS2)
      DO 120 I=1,NTMAT
120 A3010E(I)=DMAT(I)
C
      DO 130 I=1,NTMAT
130 AMAT(I)=A3110L(I)/2.D00
      CALL SORT(AMAT,DMAT,NS1,NS2)
      DO 140 I=1,NTMAT
140 A3110E(I)=DMAT(I)
C
      DO 150 I=1,NTMAT
150 AMAT(I)=A3100L(I)/6.D00
      CALL SORT(AMAT,DMAT,NS1,NS2)
      DO 160 I=1,NTMAT
160 A3100E(I)=DMAT(I)
C
      GO TO 350
C
C      FOR QUADRATIC BASIS FORMULATION
C
170 A10E(1)=1.D00/6.D00
      A10E(2)=4.D00/6.D00
      A10E(3)=1.D00/6.D00
C
      DO 180 I=1,NTMAT
180 AMAT(I)=A3000Q(I)/420.D00
      CALL SORT(AMAT,DMAT,NS1,NS2)
      DO 190 I=1,NTMAT
190 A3000E(I)=DMAT(I)
C
200 DO 210 I=1,NTMAT
210 AMAT(I)=A3011Q(I)/30.D00

```

```

      CALL SORT(AMAT,DMAT,NS1,NS2)
      DO 220 I=1,NTMAT
220  A3011E(I)=DMAT(I)
C
230  DO 240 I=1,NTMAT
240  AMAT(I)=A3001Q(I)/90.D00
      CALL SORT(AMAT,DMAT,NS1,NS2)
      DO 250 I=1,NTMAT
250  A3001E(I)=DMAT(I)
C
260  DO 270 I=1,NTMAT
270  AMAT(I)=A3010Q(I)/30.D00
      CALL SORT(AMAT,DMAT,NS1,NS2)
      DO 280 I=1,NTMAT
280  A3010E(I)=DMAT(I)
C
290  DO 300 I=1,NTMAT
300  AMAT(I)=A3110Q(I)/30.D00
      CALL SORT(AMAT,DMAT,NS1,NS2)
      DO 310 I=1,NTMAT
310  A3110E(I)=DMAT(I)
C
320  DO 330 I=1,NTMAT
330  AMAT(I)=A3100Q(I)/30.D00
      CALL SORT(AMAT,DMAT,NS1,NS2)
      DO 340 I=1,NTMAT
340  A3100E(I)=DMAT(I)
350  CONTINUE
      DO 360 I=1,NTMATR
      A200E(I)=0.0D00
      A201E(I)=0.0D00
      A211E(I)=0.0D00
      A210E(I)=0.0D00
      DO 360 J=1,NDOFEL
      K1=(I-1)*NDOFEL+J
      A200E(I)=A200E(I)+A3000E(K1)
      A201E(I)=A201E(I)+A3001E(K1)
      A211E(I)=A211E(I)+A3011E(K1)
      A210E(I)=A210E(I)+A3010E(K1)
360  CONTINUE
      DO 370 I=1,16
      A40000(I)=F40000(I)/60.D00
      A40001(I)=F40001(I)/12.D00
      A40010(I)=F40010(I)/12.D00
370  A41000(I)=F41000(I)/12.D00
      DO 380 I=1,81
380  Q40001(I)=S40001(I)/420.D00
      RETURN
      END

```

```
SUBROUTINE SORT(AMAT,DMAT,NS1,NS2)

C
C      REARRANGE MATRIX ENTRIES COMPUTED FROM BLOCK DATA IN
C      ELMAT INTO ARRAYS DIRECTLY USABLE IN SUBROUTINE FANDJ
C

  IMPLICIT REAL*8(A-H,O-Z)
  DIMENSION AMAT(27),DMAT(27)
  DO 10 I=1,NS1
    DO 10 J=1,NS2
      K1=(J-1)*NS1+I
      K2=(I-1)*NS2+J
      DMAT(K2)=AMAT(K1)
10  CONTINUE
  RETURN
  END
```

SUBROUTINE OUTPUT

C
C
C
C
C
C

CALCULATION OF SOLUTION INTEGRAL NORMS
INCLUDES HO,H1,ENERGY, AND L1 NORMS
OUTPUT OF SOLUTION IN INTEGER FORMAT

```

IMPLICIT REAL*8(A-H,O-Z)
COMMON/DATA/ CHIO,CHIF,THETA,DCIMP,EPS,CHIMIN,DQMAX,DQFCTU,
1      DQFCTD,PEXIT,
2      ANU2RF,ANU2MF,ANU2GF,
3      ANU1RF,ANU1MF,ANU1GF,
4      ANU2RJ,ANU2MJ,ANU2GJ,
5      ANU1RJ,ANU1MJ,ANU1GJ
COMMON/ARRAYS/ RH01(201),XMDM1(201),G1(201),PRS1(201),
1      RH00(201),XMDM0(201),G0(201),PRS0(201),
2      QMAT(9),FR(201),FM(201),FF(201),FG(201),RHT(804)
COMMON/MULY/ X(201),XP(10),YP(10),XM(10),YM(10),
1      PRG(10),XMN(201),AREA(201),ISTART(4),ISTOP(4)
COMMON/ARRAYI/ IPRTQ(201),NEM(10),NFI(2)
COMMON/IARRAY/ NDOFEL,NTMAT,NTMATR,NELE1,ISRC
1      NINBCR,NINBCM,NINBCG,NINBCP,
2      NOUBCR,NOUBCM,NOUBCG,NOUBCP
COMMON/IDAT/ NBAND,NNODE,INODE,K,IPRNT,IMPLOT,XI,NPR,ICGEN,
1      NITMAX,IPRMP,NBAND1,IRMX,IMMX,IPMX,IGMX,IOPT,NITLO,
2      NITHI,H,VMAX,NREG,ISHK
COMMON/SWITCH/ IDVDRM,IDS,IAR,JPM,JRR,JRM
COMMON/NONDIM/ TZERO,RH0ZRO,PZERO,GZERO,CSNDO,TIMZRO,ASTAR,
1      XMZERO,XMDOTO,TEMPIN,FRSIN,XMDOT
COMMON/UBRMAX/ UB10MX,UB11MX
COMMON/MAT/ A200E(9),A201E(9),A210E(9),A211E(9),
1      A300E(27),A3001E(27),A3010E(27),A3100E(27),
2      A3011E(27),A3110E(27),A3101E(27),A30E(9),
3      A40000(16),A40001(16),A40010(16),A41000(16),
4      A40001(81),A10E(3)
DIMENSION QRHO(201),UXMOM(201),UPRS(201),UG(201)
DIMENSION QQRHO(201),UUXMOM(201),UUPRS(201),UUG(201)
DIMENSION UVEL(201),TEMP(201)
IPMULT=1000

```

C
C
C

ZERO OUT SOLUTION NORM LOCATION

```

ENGRHO=0.0D00
ENGMDM=0.0D00
ENGPRS=0.0D00
ENGENG=0.0D00
HORHO=0.0D00
HOMDM=0.0D00
HOPRS=0.0D00
HOENG=0.0D00
SL1RHO=0.0D00
SL1MDM=0.0D00

```

```
SL1ENG=0.0D00
SL1PRS=0.0D00
```

C
C
C

CALCULATE SOLUTION INTEGRAL NORMS

```
DO 10 IEL=1,NELE1
K1=K*(IEL-1)
DOX=X(IEL+1)-X(IEL)
ODX=1.D00/DOX
DO 10 J=1,NDOFEL
M1=K1+J
SL1RHO=SL1RHO+DOX*A10E(J)*RHO1(M1)
SL1MOM=SL1MOM+DOX*A10E(J)*XMOM1(M1)
SL1ENG=SL1ENG+DOX*A10E(J)*G1(M1)
SL1PRS=SL1PRS+DOX*A10E(J)*PRS1(M1)
DO 10 L=1,NDOFEL
JL=(L-1)*NDOFEL+J
M2=K1+L
ENGRHO=ENGRHO+DOX*RHO1(M1)*A200E(JL)*RHO1(M2)
ENGMOM=ENGMOM+DOX*XMOM1(M1)*A200E(JL)*XMOM1(M2)
ENGPRS=ENGPRS+DOX*PRS1(M1)*A200E(JL)*PRS1(M2)
ENGENG=ENGENG+DOX*G1(M1)*A200E(JL)*G1(M2)
HORHO=HORHO+ODX*RHO1(M1)*A211E(JL)*RHO1(M2)
HOMOM=HOMOM+ODX*XMOM1(M1)*A211E(JL)*XMOM1(M2)
HOPRS=HOPRS+ODX*PRS1(M1)*A211E(JL)*PRS1(M2)
10 HOENG=HOENG+ODX*G1(M1)*A211E(JL)*G1(M2)
ENGRHO=0.5D0*DABS(ENGRHO)
ENGMOM=0.5D0*DABS(ENGMOM)
ENGPRS=0.5D0*DABS(ENGPRS)
ENGENG=0.5D0*DABS(ENGENG)
HORHO=0.5D0*DABS(HORHO)
HOMOM=0.5D0*DABS(HOMOM)
HOPRS=0.5D0*DABS(HOPRS)
SL1RHO=0.5D00*SL1RHO
SL1MOM=0.5D00*SL1MOM
SL1ENG=0.5D00*SL1ENG
SL1PRS=0.5D00*SL1PRS
HOENG=0.5D0*DABS(HOENG)
HIRHO=ENGRHO+HORHO
HIMOM=ENGMOM+HOMOM
HIPRS=ENGPRS+HOPRS
HIENG=ENDENG+HOENG
```

C
C
C

PRINT OUT DISCRETE SOLUTION ARRAYS IN INTEGER FORM

```
DO 20 K1=1,NNODE
20 IPRTQ(K1)=IPMULT*RHO1(K1)
WRITE(6,260)RHOZRO
30 WRITE(6,250)(IPRTQ(L),L=1,NNODE)
DO 40 K1=1,NNODE
UVEL(K1)=XMOM1(K1)/RHO1(K1)
40 IPRTQ(K1)=IPMULT*UVEL(K1)
```

```

WRITE(6,270)CSND0
50 WRITE(6,250)(IPRTQ(L),L=1,NNODE)
DO 60 K1=1,NNODE
60 IPRTQ(K1)=IPMULT*XMOM1(K1)
WRITE(6,280)XMZERO
70 WRITE(6,250)(IPRTQ(L),L=1,NNODE)
DO 80 K1=1,NNODE
80 IPRTQ(K1)=IPMULT*PRS1(K1)
WRITE(6,290)PZERO
90 WRITE(6,250)(IPRTQ(L),L=1,NNODE)
DO 100 K1=1,NNODE
100 IPRTQ(K1)=IPMULT*G1(K1)
WRITE(6,300)GZERO
110 WRITE(6,250)(IPRTQ(L),L=1,NNODE)
DO 120 K1=1,NNODE
TEMP(K1)=G1(K1)/RHO1(K1)-0.28D00*UVEL(K1)*UVEL(K1)
120 IPRTQ(K1)=IPMULT*TEMP(K1)
WRITE(6,340)TZERO
130 WRITE(6,250)(IPRTQ(L),L=1,NNODE)
C
C      INLET BOUNDARY CONDITION FOR MOMENTUM IS
C      AUTOMATICALLY SET AS DIRICHLET IF
C      MACH NO. IS ANYWHERE GREATER THAN 1
C
NINBCM=0
DO 140 I=1,NNODE
CSOUND=DSQRT(PRS1(I)/RHO1(I))
XMN(I)=UVEL(I)/CSOUND
140 IF(XMN(I) .GE. 1.0D00)NINBCM=1
DO 150 K1=1,NNODE
150 IPRTQ(K1)=IPMULT*(XMN(K1))/10.D00
WRITE(6,310)
160 WRITE(6,250)(IPRTQ(L),L=1,NNODE)
DO 170 K1=1,NNODE
170 IPRTQ(K1)=100*AREA(K1)
WRITE(6,320)ASTAR
180 WRITE(6,250)(IPRTQ(L),L=1,NNODE)
ICRMAX=0
DCX=H/(X(2)-X(1))
DO 190 K1=1,NNODE
IPRTQ(K1)=10*DCX*XMOM1(K1)/RHO1(K1)
IF(IPRTQ(K1) .LE. ICRMAX) GO TO 190
ICRMAX=IPRTQ(K1)
190 CONTINUE
C      IF(ICRMAX .GE. 80) CHIFCT=1.0D00
WRITE(6,350)
200 WRITE(6,250)(IPRTQ(L),L=1,NNODE)
DO 210 K1=1,NNODE
210 IPRTQ(K1)=IPMULT*RHO1(K1)*AREA(K1)*UVEL(K1)
WRITE(6,330)XMDOTO
WRITE(6,250)(IPRTQ(L),L=1,NNODE)
C

```

```

C      PRINT OUT SOLUTION NORMS
C
WRITE(6,370)
WRITE(6,380)HORHO,ENGRHO,H1RHO,SL1RHO
WRITE(6,390)HOMOM,ENGMOM,H1MOM,SL1MOM
WRITE(6,400)HOPRS,ENGPRS,H1PRS,SL1PRS
WRITE(6,410)HOENG,ENGENG,H1ENG,SL1ENG
IF(IRCHDS .EQ. 0)WRITE(6,420)
IF(IMPLOT .EQ. 0)GO TO 230
IF(IRCHDS .GT. 0)GO TO 220

C
C      WRITE DISCRETE SOLUTION ARRAYS INTO GRAPHICS FILE
C
WRITE(9,240)(RHO1(I),I=1,NNODE)
WRITE(9,240)(XMOM1(I),I=1,NNODE)
WRITE(9,240)(PRS1(I),I=1,NNODE)
WRITE(9,240)(G1(I),I=1,NNODE)
WRITE(9,240)(XMN(I),I=1,NNODE)
WRITE(9,240)(UVEL(I),I=1,NNODE)
WRITE(9,240)(TEMP(I),I=1,NNODE)
220 WRITE(9,240)HORHO,H1RHO
WRITE(9,240)HOMOM,H1MOM
WRITE(9,240)HOPRS,H1PRS
WRITE(9,240)HOENG,H1ENG
230 CONTINUE
RETURN
240 FORMAT(50I5.5)
250 FORMAT(1X,25I5)
260 FORMAT(/,10X,'RHO VECTOR * 1000',8X,'RHO ZERO =',D15.6)
270 FORMAT(/,10X,'VELOCITY VECTOR * 1000',3X,'AZERO =',D15.6)
280 FORMAT(/,10X,'MOMENTUM VECTOR * 1000',3X,'MOMENTUM ZERO =',
  *D15.6)
290 FORMAT(/,10X,'PRESSURE VECTOR * 1000',3X,'PRESSURE ZERO =',
  *D15.6)
300 FORMAT(/,10X,'ENERGY VECTOR * 1000',4X,'ENERGY ZERO =',
  *D15.6)
310 FORMAT(/,10X,'MACH NO. * 100')
320 FORMAT(/,10X,'AREA VECTOR * 100',9X,'AREA THROAT =',D15.6)
330 FORMAT(/,10X,'MASS FLOW RATE *1000',5X,'MDDTO =',D15.6)
340 FORMAT(/,10X,'TEMPERATURE VECTOR * 1000',3X,'TEMPERATURE ZERO =',
  *'TEMPERATURE ZERO =',D15.6)
350 FORMAT(/,10X,'COURANT NUMBER * 10')
360 FORMAT(//)
370 FORMAT(1H1,9X,'NORMS',12X,'ENERGY',9X,'H-ZERO**2',7X,
  *'H-ONE**2',9X,'L-ONE',/)
380 FORMAT(10X,'DENSITY',5X,4D15.6,/)
390 FORMAT(10X,'MOMENTUM',4X,4D15.6,/)
400 FORMAT(10X,'PRESSURE',4X,4D15.6,/)
410 FORMAT(10X,'ENERGY',6X,4D15.6,/)
420 FORMAT(/,5X,'TIME',7X,'DELTIME'2X,'NIT',6X,'DELRMX',4X,
  * 'NODE',6X,'DELMX',4X,'NODE',6X,'DELGX',4X,'NODE',6X,
  * 'DELPX',4X,'NODE')
END

```

```

SUBROUTINE BANDET (S,INT,A,N,MAXN,M1,M3)
C
C      COMPUTE THE LU DECOMPOSITION OF A NONSYMMETRIC BANDED
C      MATRIX STORED IN DIAGONAL FORMAT USING PARTIAL PIVOTING
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION S(MAXN,M1),A(M3,MAXN),INT(MAXN)
      L=M1
      DO 30 I=1,M1
        K2=M1+2-I
        DO 10 J=K2,M3
10    A(J-L,I)=A(J,I)
        L=L-1
        K2=M3-L
        DO 20 J=K2,M3
20    A(J,I) = 0.0 D00
30    CONTINUE
      L=M1
      DO 130 K=1,N
        X=A(1,K)
        I=K
        K2=K+1
        IF(L .LT. N)      L=L+1
        IF(L .LT. K2)      GO TO 60
        DO 50 J=K2,L
          IF ( DABS(A(1,J)) .LE. DABS(X) )      GO TO 50
40    X=A(1,J)
        I=J
50    CONTINUE
60    INT(K)=I
        IF(X)      70,140,70
70    IF(I-K)      80,100,80
80    DO 90 J=1,M3
          X=A(J,K)
          A(J,K)=A(J,I)
90    A(J,I)=X
100   IF(L .LT. K2)      GO TO 130
        DO 120 J=K2,L
          X = 1.0 D00
          IF ( A(1,K) .NE. 0.0 D00)      X = A(1,J) / A(1,K)
          S(K,J-K) = X
          DO 110 JJ=2,M3
110    A(JJ-1,J)=A(JJ,J)-A(JJ,K)*X
120    A(M3,J) = 0.0 D00
130   CONTINUE
      RETURN
C
C      ZERO PIVOT HAS BEEN FOUND. PRINT ERROR MESSAGE AND STOP
C
140  CONTINUE
      WRITE(6,200)K,LEL,JEL,LXYZ,K,I,K2,L,J,A(1,K),A(1,J),X
      GO TO 190

```

```

      NNN=(M3-1)/2
      GO TO (150,170),NNN
150  WRITE(6,160)((A(L,K),L=1,M3),K=1,MAXN)
160  FORMAT(5X,3D15.5)
      GO TO 190
170  WRITE(6,180)((A(L,K),L=1,M3),K=1,MAXN)
180  FORMAT(5X,5D15.5)
190  STOP
200  FORMAT ( 1H1 /// 5X, 30H<<<< ZERO PIVOT IN BANDET >>>>/,
      *10X, 'K, LEL, LXYZ, K, I, K2, L, J, A(1,K), A(1,J), X',
      */ ,10X,9I5,3D15.5)
      END

```

C
C
C
C

```

      SUBROUTINE BANDSL (S,INT,F,A,N,MAXN,M1,M3)

```

C
C
C
C
C

```

      COMPUTE THE SOLUTION OF THE LINEAR SYSTEM USING THE LU
      DECOMPOSITION OF THE COEFFICIENT MATRIX AND OTHER
      INFORMATION PROVIDED BY BANDET AND THE RIGHT HAND SIDE F

```

```

      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION S(MAXN,M1),F(MAXN),A(M3,MAXN),INT(MAXN)
      NAS      =  M3 * MAXN
      L=M1
      DO 40 K=1,N
      I=INT(K)
      IF(I-K) 10,20,10
10  X=F(K)
      F(K)=F(I)
      F(I)=X
20  K2=K+1
      IF(L.LT. N)  L=L+1
      IF(L.LT. K2)  GO TO 40
      DO 30 I=K2,L
      X=S(K,I-K)
30  F(I)=F(I)-X*F(K)
40  CONTINUE
      L=1
      DO 90 II=1,N
      I=N+1-II
      X=F(I)
      M=I-1
      IF(L-1) 50,70,50
50  DO 60 K=2,L
      X=X-A(K,I)*F(K+M)
70  F(I)=X/A(1,I)
      IF(L-M3) 80,90,90
80  L=L+1
90  CONTINUE
      RETURN
      END

```

```
SUBROUTINE CSAREA(X1,X2,X3,Y1,Y2,Y3,C1,C2,C3)
```

```
C
C
C
C
C
C
```

```
    CALCULATE COEFFICIENTS FOR QUADRATIC CURVE FIT OF
    THE INPUT MACRO-ELEMENT NODAL CROSS SECTIONAL AREAS
```

```
    IMPLICIT REAL*8(A-H,O-Z)
    DIMENSION F(9),XP(3),YP(3)
    XP(1)=X1
    XP(2)=X2
    XP(3)=X3
    YP(1)=Y1
    YP(2)=Y2
    YP(3)=Y3
    DO 10 I=1,3
    F(I)=1.0D00
    F(I+3)=XP(I)
10  F(I+6)=XP(I)*XP(I)
    CALL DETERM(F,C0)
    DO 20 I=1,3
20  F(I)=YP(I)
    CALL DETERM(F,C1)
    C1=C1/C0
    DO 30 I=1,3
    F(I)=1.000
30  F(I+3)=YP(I)
    CALL DETERM(F,C2)
    C2=C2/C0
    DO 40 I=1,3
    F(I+3)=XP(I)
40  F(I+6)=YP(I)
    CALL DETERM(F,C3)
    C3=C3/C0
    RETURN
    END
```

```
C
```

```
    SUBROUTINE DETERM(F,C)
```

```
C
```

```
    CALCULATE DETERMINANT OF 3X3 MATRIX
```

```
C
C
```

```
    IMPLICIT REAL*8(A-H,O-Z)
    DIMENSION F(9)
    C=F(1)*F(5)*F(9)+F(3)*F(4)*F(8)+F(2)*F(6)*F(7)-
1  (F(1)*F(6)*F(8)+F(2)*F(4)*F(9)+F(3)*F(5)*F(7))
    RETURN
    END
```

SUBROUTINE INIT

C
C
C
C
C
C
C
C

GENERATE INITIAL CONDITION NODAL DISTRIBUTIONS OF
PRESSURE, DENSITY, AND MOMENTUM. CALCULATE ENERGY AND
STAGNATION REFERENCE STATES. NONDIMENSIONALIZE EACH
VARIABLE ARRAY ON APPROPRIATE REFERENCE STATE.

```

      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/ARRAYS/ RHO0(201),XMM0(201),G0(201),PRS0(201),
1      RHO0(201),XMM0(201),G0(201),PRS0(201),
2      OMAT(9),FR(201),FM(201),FP(201),FG(201),RHT(804)
      COMMON/MULT/ X(201),XP(10),YP(10),XM(10),YM(10),
1      PRG(10),XMN(201),AREA(201),ISTART(4),ISTOP(4)
      COMMON/IDAT/ NBAND,NNODE,INODE,K,IPRNT,IMPLOT,XI,NPR,ICGEN,
1      NITMAX,IPRMP,NBAND1,IRMX,IMMX,IPMX,IGMX,IOPT,NITLO,
2      NITHI,H,VMAX,NREG,ISHK
      COMMON/NONDIM/TZERO,RHOZRO,PZERO,GZERO,CSNDO,TIMZRO,ASTAR,
1      XMZERO,XMDOTO,TEMPIN,PRSIN,XMDOT
      WRITE(6,50)

```

C
C
C
C

GENERATE NODAL VALUES FOR DENSITY, MOMENTUM,
AND PRESSURE

```

      DO 60 I=1,NREG
      JSTART=ISTART(I)
      JSTOP=ISTOP(I)
      DO 60 J=JSTART,JSTOP
      RHO0(J)=RHO0(ISTART(I))
      XMM0(J)=XMM0(ISTART(I))
60 PRS0(J)=PRS0(ISTART(I))

```

C
C
C

CALCULATE STAGNATION REFERENCE STATES

```

      VELIN=XMM0(1)/RHO0(1)
      TZERO=TEMPIN+VELIN*VELIN/6009.5D00
      TRAT=TZERO/TEMPIN
      PZERO=PRS0(1)*TRAT**3.5D00
      RHOZRO=RHO0(1)*TRAT**2.5D00
      CSNDO=DSQRT(1.4D00*TRAT*PRS0(1)/RHO0(1))
      GZERO=4292.5D00*RHOZRO*TZERO
      XMZERO=RHOZRO*CSNDO
      XMDOTO=RHOZRO*CSNDO*ASTAR
      XL=X(INODE)-X(1)
      TIMZRO=XL/CSNDO

```

C
C
C

CALCULATE ENERGY AND NONDIMENSIONALIZE VARIABLES

```

      DO 10 I=1,NNODE
      G0(I)=(2.5D00*PRS0(I)+
*      0.5D00*XMM0(I)*XMM0(I)/RHO0(I))/GZERO

```

```
CSOUND=DSQRT(1.4D00*PRS0(I)/RH00(I))
XMN(I)=XMOM0(I)/RH00(I)/CSOUND
RH00(I)=RH00(I)/RHOZRO
XMOM0(I)=XMOM0(I)/RHOZRO/CSND0
10 PRS0(I)=PRS0(I)/PZERO
DO 20 I=1,INODE
20 X(I)=X(I)/XL
RETURN
30 FORMAT(15I5)
40 FORMAT(5D15.5)
50 FORMAT(/,1X,'GENERATE NODAL VALUES OF',/,1X,
1'DENSITY, PRESSURE, AND MOMENTUM',/)
END
```

SUBROUTINE ISNTRP

```

C
C
C      COMPUTE INITIAL CONDITION DISTRIBUTION FOR AIR USING
C      ISENTROPIC/NORMAL SHOCK SOLUTIONS WITH DEFINED CROSS-
C      SECTIONAL AREAS AND EITHER INLET MASS FLOW RATE(ICGEN=2)
C      OR INLET FLOW MACH NO.(ICGEN=3).
C      CALCULATE STAGNATION REFERENCE STATES.
C      GENERATE NODAL NONDIMENSIONAL VARIABLE DISTRIBUTIONS.
C      INITIAL CONDITION NODAL LOCATION OF SHOCK SPECIFIED
C      BY ISHK.
C
C      IMPLICIT REAL*8(A-H,O-Z)
C      COMMON/ARRAYS/ RHO1(201),XDM1(201),G1(201),PRS1(201),
1      RHO0(201),XDM0(201),G0(201),PRS0(201),
2      DMAT(9),FR(201),FM(201),FP(201),FG(201),RHT(804)
C      COMMON/MULT/ X(201),XP(10),YP(10),XM(10),YM(10),
1      PRG(10),XMN(201),AREA(201),ISTART(4),ISTOP(4)
C      COMMON/IDAT/ NBAND,NNODE,INODE,K,IPRNT,IMPLT,XI,NPR,ICGEN,
1      NITMAX,IPRMP,NBAND1,IRMX,IMMX,IPMX,IGMX,IDPT,NITLO,
2      NITHI,H,VMAX,NREG,ISHK
C      COMMON/NONDIM/TZERO,RHOZRO,PZERO,GZERO,CSNDO,TIMZRO,ASTAR,
1      XMZERO,XMDOTO,TEMPIN,PRSIN,XMDOT
C      IF(ICGEN .EQ. 3)GO TO 10
C
C      IF ICGEN = 2, CALCULATE
C      INLET VELOCITY, STAGNATION TEMPERATURE, AND T/TZERO
C
C      VELIN=1717.0000*TEMPIN*XMDOTO*ASTAR/PRSIN/AREA(1)
C      TZERO=TEMPIN+VELIN*VELIN/6009.5000
C      TRAT=TZERO/TEMPIN
C      GO TO 20
C
C      IF ICGEN = 3, CALCULATE
C      STAGNATION TEMPERATURE AND T/TZERO
C
10  TRAT=1.0000+0.2000*XMN(1)*XMN(1)
C      TZERO=TEMPIN*TRAT
C
C      CALCULATE STAGNATION REFERENCE CONDITIONS
C
20  FZERO=PRSIN*TRAT**3.5000
C      RHOZRO=FZERO/1717.0000/TZERO
C      CSNDO=DSQRT(2403.8000*TZERO)
C      GZERO=4292.5000*RHOZRO*TZERO
C      XMZERO=RHOZRO*CSNDO
C      XMDOTO=RHOZRO*CSNDO*ASTAR
C      XL=X(INODE)-X(1)
C      TIMZRO=XL/CSNDO
C      SWI=1.0000
C

```

C CALCULATE NODAL MACH NO. BASED ON AREA RATIOS FOR NODE 1 170
 C THROUGH ISHK. CURVE FIT SOLUTIONS BASED ON GAMMA = 1.4
 C IS USED TO CALCULATE MACH NO.
 C

```

DO 40 I=1,ISHK
IF(I.EQ. 1)GO TO 30
IF(DABS(AREA(I)-ASTAR) .LE. .0001)SWI=0.0D00
XMN(I)=1.0D00+(1.0D00-SWI)*1.2D00*DSQRT(AREA(I)-1.0D00)
1        -SWI*0.88D00*((DLOG(AREA(I)))*0.45D00)
30 XMNK=1.0D00/(1.0D00+0.2D00*XMN(I)*XMN(I))

```

C
 C CALCULATE INITIAL CONDITION NODAL DISTRIBUTIONS FOR
 C NODUES 1 THROUGH ISHK USING ISENTROPIC RELATIONS
 C

```

PRS0(I)=XMNK**3.5D00
RH00(I)=XMNK**2.5D00
CSOUND=DSQRT(PRS0(I)/RH00(I))
XMUM0(I)=XMN(I)*CSOUND*RH00(I)
PRS=PRS0(I)*PZERO
VEL=CSND0*CSOUND*XMN(I)
RHO=RH00(I)*RHOZRO
GO(I)=(2.5D00*PRS+0.5D00*VEL*VEL*RHO)/GZERO
40 CONTINUE

```

C
 C CALCULATE RATIO OF ASTAR UPSTREAM OF SHOCK TO ASTAR
 C DOWNSTREAM OF SHOCK USING NORMAL SHOCK RELATIONS
 C

```

IF(ISHK.EQ. NNODE)GO TO 70
SHKRAT=((2.4D00*XMN(ISHK)*XMN(ISHK)/
1        (2.0D00+0.4D00*XMN(ISHK)*XMN(ISHK))**3.5D00)*
2        ((2.4D00/(2.8D00*XMN(ISHK)*XMN(ISHK)-0.4D00))**2.5D00)
IF(XMN(ISHK) .GT. 1.0D00)XMN(ISHK)=
*        DSQRT((.4D00*XMN(ISHK)*XMN(ISHK)+2.0D00)/
*        (2.8D00*XMN(ISHK)*XMN(ISHK)-0.4D00))
ASTAR2=AREA(ISHK)*XMN(ISHK)*1.728D00/
*        ((1.0D00+0.2D00*XMN(ISHK)*XMN(ISHK))**3.0D00)
DO 60 I=ISHK,NNODE
IF(I.EQ. ISHK)GO TO 50

```

C
 C CALCULATE NODAL MACH NO. BASED ON AREA RATIOS FOR ISHK
 C THROUGH NNODE. CURVE FIT SOLUTION BASED ON GAMMA = 1.4
 C IS USED TO CALCULATE MACH NO.
 C

```

XMN(I)=1.0D00-0.88D00*((DLOG(AREA(I)*ASTAR/ASTAR2))*0.45D00)
50 XMNK=1.0D00/(1.0D00+0.2D00*XMN(I)*XMN(I))

```

C
 C CALCULATE INITIAL CONDITION NODAL DISTRIBUTIONS FOR
 C ISHK THROUGH NNODE USING ISENTROPIC RELATIONS
 C

```

PRS0(I)=SHKRAT*XMNK**3.5D00
RH00(I)=SHKRAT*XMNK**2.5D00
CSOUND=DSQRT(PRS0(I)/RH00(I))

```

```
      XMOM0(I)=XMN(I)*CSOUND*RHO0(I)
      PRS=PRS0(I)*PZERO
      VEL=CSND0*CSOUND*XMN(I)
      RHO=RHO0(I)*RHOZRO
60    G0(I)=(2.5D00*PRS+0.5D00*VEL*VEL*RHO)/GZERO
70    DO 80 I=1,INODE
80    X(I)=X(I)/XL
      RETURN
90    FORMAT(15I5)
100   FORMAT(5D15.5)
110   FORMAT(/,1X,'READ INLET VALUES OF PRESSURE AND TEMPERATURE'
1      ,/,1X,'AND EITHER MASS FLOW RATE OR MACH NUMBER.',/,/,6X,
2      'PRSIN',10X,'TEMPIN')
120   FORMAT(/,6X,'ISHK')
130   FORMAT(/,6X,'XMDOT')
140   FORMAT(/,6X,'XMN(1)')
      END
```

BLOCK DATA

```

C
C      FINITE ELEMENT HYPER-MATRIX INTEGER DATA
C      REAL DATA ARRAYS ARE COMPUTED IN SUBROUTINE ELMAT
C
C      IMPLICIT REAL*8(A-H,O-Z)
C
C      COMMON/AL/A3000L(8),A3011L(8),A3001L(8),A3101L(8),A3010L(8)
1          ,A3100L(8),A3110L(8),A3003L(8)
2          ,P40000(16),P40001(16),P40010(16),P41000(16)
C      COMMON/AQ/A3000Q(27),A3011Q(27),A3001Q(27),A3010Q(27)
1          ,A3100Q(27),A3110Q(27),A3003Q(27)
2          ,S40001(81)
C
C      DATA A3011L /
1      1.0 D 00 , -1.0 D 00
2      , 1.0 D 00 , -1.0 D 00
3      , -1.0 D 00 , 1.0 D 00
4      , -1.0 D 00 , 1.0 D 00
5      /
C
C      DATA A3001L /
1      -2.0 D 00 , 2.0 D 00
2      , -1.0 D 00 , 1.0 D 00
3      , -1.0 D 00 , 1.0 D 00
4      , -2.0 D 00 , 2.0 D 00
5      /
C
C      DATA A3000L /
1      3.0 D 00 , 1.0 D 00
2      , 1.0 D 00 , 1.0 D 00
3      , 1.0 D 00 , 1.0 D 00
4      , 1.0 D 00 , 3.0 D 00
5      /
C
C      DATA A3101L /
1      1.0 D 00 , -1.0 D 00
2      , -1.0 D 00 , 1.0 D 00
3      , 1.0 D 00 , -1.0 D 00
4      , -1.0 D 00 , 1.0 D 00
5      /
C
C      DATA A3010L /
1      -2.0 D 00 , -1.0 D 00
2      , -1.0 D 00 , -2.0 D 00
3      , 2.0 D 00 , 1.0 D 00
4      , 1.0 D 00 , 2.0 D 00
5      /

```

C

```

DATA A3110L /
1      1.0    D 00    ,    1.0    D 00
2      , -1.0    D 00    ,   -1.0    D 00
3      , -1.0    D 00    ,   -1.0    D 00
4      ,  1.0    D 00    ,    1.0    D 00
5      /

```

C

```

DATA A3003L /
1      1.0    D 00    ,    0.0    D 00
2      ,  0.0    D 00    ,    0.0    D 00
3      ,  0.0    D 00    ,    0.0    D 00
4      ,  0.0    D 00    ,   -1.0    D 00
5      /

```

```

DATA A3100L /
1      -2.0    D 00    ,   -1.0    D 00
2      ,  2.0    D 00    ,    1.0    D 00
3      , -1.0    D 00    ,   -2.0    D 00
4      ,  1.0    D 00    ,    2.0    D 00
5      /

```

C

```

DATA A3000R /
1      39.0    D 00    ,  20.0    D 00    , -3.0    D 00    ,
2      20.0    D 00    ,  16.0    D 00    , -8.0    D 00    ,
3      -3.0    D 00    , -8.0    D 00    , -8.0    D 00    ,
4      20.0    D 00    ,  16.0    D 00    , -8.0    D 00    ,
5      16.0    D 00    , 192.0    D 00    ,  16.0    D 00    ,
6      -8.0    D 00    ,  16.0    D 00    ,  20.0    D 00    ,
7      -3.0    D 00    , -8.0    D 00    , -3.0    D 00    ,
8      -8.0    D 00    ,  16.0    D 00    ,  20.0    D 00    ,
9      -3.0    D 00    ,  20.0    D 00    ,  39.0    D 00
1      /

```

C

```

DATA A3001R /
1      -30.0    D 00    ,  36.0    D 00    , -6.0    D 00    ,
2      -18.0    D 00    ,  24.0    D 00    , -6.0    D 00    ,
3       3.0    D 00    ,  0.0    D 00    , -3.0    D 00    ,
4      -18.0    D 00    ,  24.0    D 00    , -6.0    D 00    ,
5      -48.0    D 00    ,  0.0    D 00    , 48.0    D 00    ,
6       6.0    D 00    , -24.0    D 00    , 18.0    D 00    ,
7       3.0    D 00    ,  0.0    D 00    , -3.0    D 00    ,
8       6.0    D 00    , -24.0    D 00    , 18.0    D 00    ,
9       6.0    D 00    , -36.0    D 00    , 30.0    D 00
1      /

```

C

```

DATA A3011Q /
1  37.0  D 00 , -44.0  D 00 ,  7.0  D 00 ,
2  36.0  D 00 , -32.0  D 00 , -4.0  D 00 ,
3  -3.0  D 00 , -4.0  D 00 ,  7.0  D 00 ,
4 -44.0  D 00 , 48.0  D 00 , -4.0  D 00 ,
5 -32.0  D 00 , 64.0  D 00 , -32.0  D 00 ,
6  -4.0  D 00 , 48.0  D 00 , -44.0  D 00 ,
7   7.0  D 00 , -4.0  D 00 , -3.0  D 00 ,
8  -4.0  D 00 , -32.0  D 00 , 36.0  D 00 ,
9   7.0  D 00 , -44.0  D 00 , 37.0  D 00
1 /

```

C

```

DATA A3010Q /
1 -10.0  D 00 , -6.0  D 00 ,  1.0  D 00 ,
2  -6.0  D 00 , -16.0  D 00 ,  2.0  D 00 ,
3   1.0  D 00 ,  2.0  D 00 ,  2.0  D 00 ,
4  12.0  D 00 ,  8.0  D 00 ,  0.0  D 00 ,
5   8.0  D 00 ,  0.0  D 00 , -8.0  D 00 ,
6   0.0  D 00 , -8.0  D 00 , -12.0  D 00 ,
7  -2.0  D 00 , -2.0  D 00 , -1.0  D 00 ,
8  -2.0  D 00 , 16.0  D 00 ,  6.0  D 00 ,
9  -1.0  D 00 ,  6.0  D 00 , 10.0  D 00
1 /

```

C

```

DATA A3110Q /
1  37.0  D 00 , 36.0  D 00 , -3.0  D 00 ,
2 -44.0  D 00 , -32.0  D 00 , -4.0  D 00 ,
3   7.0  D 00 , -4.0  D 00 ,  7.0  D 00 ,
4 -44.0  D 00 , -32.0  D 00 , -4.0  D 00 ,
5  48.0  D 00 , 64.0  D 00 , 48.0  D 00 ,
6  -4.0  D 00 , -32.0  D 00 , -44.0  D 00 ,
7   7.0  D 00 , -4.0  D 00 ,  7.0  D 00 ,
8  -4.0  D 00 , -32.0  D 00 , -44.0  D 00 ,
9  -3.0  D 00 , 36.0  D 00 , 37.0  D 00
1 /

```

C

```

DATA A3100Q /
1 -10.0  D 00 , -6.0  D 00 ,  1.0  D 00 ,
2  12.0  D 00 ,  8.0  D 00 ,  0.0  D 00 ,
3  -2.0  D 00 , -2.0  D 00 , -1.0  D 00 ,
4  -6.0  D 00 , -16.0  D 00 ,  2.0  D 00 ,
5   8.0  D 00 ,  0.0  D 00 , -8.0  D 00 ,
6  -2.0  D 00 , 16.0  D 00 ,  6.0  D 00 ,
7   1.0  D 00 ,  2.0  D 00 ,  2.0  D 00 ,
8   0.0  D 00 , -8.0  D 00 , -12.0  D 00 ,
9  -1.0  D 00 ,  6.0  D 00 , 10.0  D 00
1 /

```

C FORTRAN VARIABLES FOR 2ND DEGREE HYPERMATRICES

C A40000L

DATA P40000 /	12.0	D00, 3.0	D00, 3.0	D00, 2.0	D00,
1	3.0	D00, 2.0	D00, 2.0	D00, 3.0	D00,
2	3.0	D00, 2.0	D00, 2.0	D00, 3.0	D00,
3	2.0	D00, 3.0	D00, 3.0	D00, 12.0	D 00
4 /					

C A40001L

DATA P40001 /	-3.0	D00, 3.0	D00, -1.0	D00, 1.0	D00,
1	-1.0	D00, 1.0	D00, -1.0	D00, 1.0	D00,
2	-1.0	D00, 1.0	D00, -1.0	D00, 1.0	D00,
3	-1.0	D00, 1.0	D00, -3.0	D00, 3.0	D 00
4 /					

C A40010L

DATA P40010 /	-3.0	D00, -1.0	D00, -1.0	D00, -1.0	D00,
1	-1.0	D00, -1.0	D00, -1.0	D00, -3.0	D00,
2	3.0	D00, 1.0	D00, 1.0	D00, 1.0	D00,
3	1.0	D00, 1.0	D00, 1.0	D00, 3.0	D 00
4 /					

C A41000L

DATA P41000 /	-3.0	D00, -1.0	D00, 3.0	D00, 1.0	D00,
1	-1.0	D00, -1.0	D00, 1.0	D00, 1.0	D00,
2	-1.0	D00, -1.0	D00, 1.0	D00, 1.0	D00,
3	-1.0	D00, -3.0	D00, 1.0	D00, 3.0	D 00
4 /					

C A40001Q

C A40001Q

DATA S40001 /	
1	-105.D0, 132.D0, -27.D0, -44.D0, 48.D0, -4.D0, 9.D0, -12.D0, 3.D0,
2	-44.D0, 8.D0, -4.D0, -48.D0, 64.D0, -16.D0, 8.D0, 0.D0, -8.D0,
3	9.D0, -12.D0, 3.D0, 8.D0, 0.D0, -8.D0, -3.D0, 12.D0, -9.D0,
4	-44.D0, 48.D0, -4.D0, 64.D0, 64.D0, -16.D0, 8.D0, 0.D0, -8.D0,
5	-48.D0, 64.D0, -16.D0, -192.D0, 0.D0, 192.D0, 16.D0, -64.D0, 48.D0,
6	8.D0, 0.D0, -8.D0, 16.D0, -64.D0, 48.D0, 4.D0, -48.D0, 44.D0,
7	9.D0, -12.D0, 3.D0, 8.D0, 0.D0, -8.D0, -3.D0, 12.D0, -9.D0,
8	8.D0, 0.D0, -8.D0, 16.D0, -64.D0, 48.D0, 4.D0, -48.D0, 44.D0,
9	-3.D0, 12.D0, -9.D0, 4.D0, -48.D0, 44.D0, 27.D0, -132.D0, 105.D0
1	/

END

RIEMANN SHOCK TUBE(100 ELEMENTS)

```

1 100 2 0
0.96830D-01 0.00000D 00 0.64550D-01
0.96830D-01 0.00000D 00 0.64550D-01
0.12910D 00 0.38730D 00 0.19365D 00
0.12910D 00 0.38730D 00 0.19365D 00
0.0 D 00 0.10000D 01
1.00000D 00 1.00000D 00
0.50000D 00
1.00000D 00
1.00000D 00
100
1
2 5.82411D-04
0 52 0.10000D+01 0.00000D+00 0.10000D+01
53 101 0.12500D+00 0.00000D+00 0.10000D+00
1 1 1 1
1 1 1 1
0.0 D 00 0.14154D 00 0.5 D-01 0.10500D+01 0.12500D+01
0.5 D 00 0.15727D-02 0.1 D-03 0.1 D-05
0 8 3 5
20 1 0

```

STATIONARY SHOCK(100 ELEMENTS)

```

1  100  8  0
0.96820D-01  0.00000D 00  0.64550D-01
0.96820D-01  0.00000D 00  0.64550D-01
0.12910D 00  0.38730D 00  0.19365D 00
0.12910D 00  0.38730D 00  0.19365D 00
0.0  D 00  0.10000D 00  0.30000D 00  0.50000D 00  0.63000D 00
0.76000D 00  0.90000D 00  0.10000D+01
5.82183D 00  5.82183D 00  1.25495D 00  1.00000D 00  1.01943D 00
1.07371D 00  1.16820D 00  1.16820D 00
0.50000D 00  0.28000D 00  0.48000D 00  0.57000D 00  0.69000D 00
0.82000D 00  0.94000D 00
5.82183D 00  1.33039D 00  1.00173D 00  1.00581D 00  1.04041D 00
1.10963D 00  1.16820D 00
1.00000D 00  1.00000D 00  1.00000D 00  1.00000D 00  1.00000D 00
1.00000D 00  1.00000D 00
10  20  20  13  13  14  10
3
75  0.42000D+03  1.44682D+04  0.10000D+00
0
1  0  1  1
0  0  0  1
0.0  D 00  0.12500D-01  0.1  D-02  0.110  D+01  0.150  D+00
0.5  D 00  0.50  D-04  0.1  D-03  0.1  D-06
0  8  3  4
20  1  0

```

MOVING SHOCK(100 ELEMENTS)

```

1  100  8  0
0.96820E-01  0.00000E 00  0.64550E-01
0.96820E-01  0.00000E 00  0.64550E-01
0.12910E 00  0.38730E 00  0.19365E 00
0.12910E 00  0.38730E 00  0.19365E 00
0.0  D 00  0.10000E 00  0.30000E 00  0.50000E 00  0.63000E 00
0.76000E 00  0.90000E 00  0.10000E+01
5.82183E 00  5.82183E 00  1.25495E 00  1.00000E 00  1.01943E 00
1.07371E 00  1.16820E 00  1.16820E 00
0.50000E 00  0.28000E 00  0.48000E 00  0.57000E 00  0.69000E 00
0.82000E 00  0.94000E 00
5.82183E 00  1.33039E 00  1.00173E 00  1.00581E 00  1.04041E 00
1.10963E 00  1.16820E 00
1.00000E 00  1.00000E 00  1.00000E 00  1.00000E 00  1.00000E 00
1.00000E 00  1.00000E 00
10  20  20  13  13  14  10
3
85  0.42000E+03  1.44682E+04  0.10000E+00
1
0.74374E+00
1  0  1  1
0  0  0  1
0.0  D 00  0.12500E-01  0.1  D-02  0.110  D+01  0.150  D+00
0.5  D 00  0.50  D-04  0.1  D-03  0.1  D-06
0  8  3  4
20  1  0

```

VITA

George Elliott Moore was born in Bridgeport, Alabama on July 26, 1954. He attended elementary schools in South Pittsburg, Tennessee and Soddy Daisy, Tennessee and was graduated from Soddy Daisy High School in June 1972. He entered The University of Tennessee in June 1972. In June, 1977 he received a Bachelor of Science degree in Engineering Science. After graduation, he was employed as a water resources engineer with the Tennessee Department of Conservation. In March 1982 he entered the Graduate School of The University of Tennessee, Knoxville. He received a Master of Science with a major in Engineering Science in June 1984. Mr. Moore will be employed by the Westinghouse Bettis Atomic Power Laboratory after graduation.