FINAL REPORT

COMBUSTION INSTABILITY ANALYSIS

NAS8-36955

Period of Performance
June 1, 1989 to May 30, 1990

T.J. Chung
Principal Investigator
Department of Mechanical Engineering
The University of Alabama in Huntsville
A new theory and computer program for combustion instability analysis are presented herein. The basic theoretical foundation resides in the concept of entropy--controlled energy growth or decay. Third order perturbation expansion is performed on the entropy--controlled acoustic energy equation to obtain the first order integrodifferential equation for the energy growth factor in terms of the linear, second, and third order energy growth parameters. These parameters are calculated from Navier–Stokes solutions with time averages performed on as many Navier–Stokes time steps as required to cover at least one peak wave period.

Applications are made for one–dimensional Navier–Stokes solution for the SSME thrust chamber with cross section area variations taken into account. It is shown that instability occurs when the mean pressure is raised to 2000 psi with 30% disturbances. Instability also arises when the mean pressure is set at 2935 psi with 20% disturbances. The system with mean pressures and disturbances more adverse than these cases has been shown to be unstable.

The present theory has a great potential and all avenues of further studies will prove to be fruitful.
# TABLE OF CONTENTS

Abstract

Nomenclature

1. Introduction

2. Governing Equations
   2.1 Navier–Stokes Equations
   2.2 Entropy–Controlled Stability Equation

3. Solution Procedures

4. Applications

5. Conclusions

6. Recommendations

Acknowledgements

References

Appendices

A. Derivation of Energy Gradients in Terms of Entropy Gradients

B. Derivation of Entropy Perturbation

C. Derivation of Integrodifferential Equation for Energy Growth Factor from Entropy–Controlled Acoustic Energy Equation

D. Integrand of $E_1$, $E_2$, $E_3$, $I_1$, $I_2$, and $I_3$

E. Listing of Computer Program (ECI–1)
**NOMENCLATURE**

\[
\begin{align*}
    c_p &= \text{Specific heat at constant pressure} \\
    B &= \text{Body force vector} \\
    D &= \text{Mass diffusivity} \\
    e &= \text{Internal energy density} \\
    E &= \text{Stagnation energy} \\
    f_{ki} &= \text{Body force} \\
    F_j &= \text{Convective flux vector} \\
    G_j &= \text{Dissipative vector} \\
    H_k &= \text{Total enthalpy} \\
    p &= \text{Pressure} \\
    R &= \text{Gas constant} \\
    S &= \text{Entropy} \\
    U &= \text{Time dependent variable vector} \\
    v_i &= \text{Velocity} \\
    Y_k &= \text{Mass fraction} \\
    \alpha_1, \alpha_2 &= \text{Energy growth rate parameters of first order, second order, and third order, respectively.} \\
    \alpha_3 &= \text{} \\
    \gamma &= \text{Specific heat ratio} \\
    \epsilon &= \text{Energy growth factor} \\
    \lambda &= \text{Thermal conductivity} \\
    \mu &= \text{Viscosity} \\
    \rho &= \text{Density} \\
    \sigma_{ij} &= \text{Total stress tensor} \\
    \tau_{ij} &= \text{Viscous stress tensor} \\
    \omega_k &= \text{Reaction rate}
\end{align*}
\]

**Subscripts and Superscripts**

- ' Fluctuation
- $ -$ Time averaged mean quantity
- $ o $ Reference state
1. INTRODUCTION

Unstable waves may exhibit a linear behavior initially under the low mean pressure, but tend to oscillate nonlinearly as the mean pressure increases, resulting possibly in sawtooth wave forms. Multidimensional effects become significant as transverse modes contribute to instability. Chemical reactions, atomization, vaporization, and turbulent flow environments must also be considered. With these complications affecting the overall stability behavior, we come to the question: What is the most rigorous method of determining combustion instability?

If time-dependent Navier-Stokes solutions for combustion capable of generating both linear and nonlinear wave oscillations are available, this information alone may provide qualitative interpretation of instability as to the tendency of possible energy growth or decay. However, they do not provide quantitative data for instability. Will there be, then, a "measure" of instability? In fact, there have been many attempts in seeking such data, the so-called "growth rate parameter" [1–5]. Unfortunately, they are normally limited to linear instability.

In order to accommodate nonlinear behavior, multidimensionality, and complex flowfield phenomena, we introduce a new approach, the Entropy-Controlled-Instability (ECI) method. The concept is similar to Flandro [6] in which the energy balance method was used in deriving the expression for energy growth from the acoustic energy equation. The focal point of the present study is the entropy-controlled energy equation which automatically takes into account shock wave oscillations in determining energy growth for instability. The asymptotic perturbation expansions of all acoustic energy terms lead to the entropy-controlled-energy equation. Applying the Green-Gauss theorem and taking time averages, we derive the stability integrodifferential equation for the energy growth factor. This factor is solved in terms of growth rate parameters which are determined from the Navier-Stokes solution.
The advantage of the present method is to provide stability information during any time period of Navier–Stokes solutions. Stability prediction capability is, therefore, limited only by the Navier–Stokes solver.

In the following, we shall describe the governing equations, derivation of stability integrodifferential equation, solution procedure, and one-dimensional example problems for validation of the theory. Extension to multidimensions and more complex flow fields is achieved simply by adopting an appropriate Navier–Stokes solver. The present formulation of stability analysis remains unchanged.

2. GOVERNING EQUATIONS

2.1 Navier–Stokes Equations

The most general conservation form of Navier Stokes equations is given by

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_j}{\partial x_j} + \frac{\partial \mathbf{G}_j}{\partial x_j} = \mathbf{B}$$

(1)

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho \mathbf{v}_i \\ \rho E \\ \rho Y_k \end{bmatrix}, \quad \mathbf{F}_j = \begin{bmatrix} \rho \mathbf{v}_j \\ \rho \mathbf{v}_i \mathbf{v}_j + p \delta_{ij} \\ \rho \mathbf{E} \mathbf{v}_j + p \mathbf{v}_j \\ \rho Y_k \mathbf{v}_j \end{bmatrix},$$

$$\mathbf{G}_j = \begin{bmatrix} 0 \\ -\tau_{ij,j} \\ -\tau_{ij} \mathbf{v}_i + q_j \\ \rho \mathbf{D} \mathbf{Y}_{k,j} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ \rho \sum_{k=1}^{N} Y_k f_{ki} \\ \rho \sum_{k=1}^{N} Y_k f_{ki} \mathbf{v}_i \\ \omega_k \end{bmatrix},$$

where $\tau_{ij}$ is the viscous stress tensor

$$\tau_{ij} = \mu (\mathbf{v}_{i,j} + \mathbf{v}_{j,i} - \frac{2}{3} \mathbf{v}_{kk} \delta_{ij}).$$
and $E$ is the stagnation energy

$$E = e + \frac{1}{2} v_i v_i = c_p T - \frac{P}{\rho} + \frac{1}{2} v_i v_i$$

and $f_{ki}$ is the body force and $q_j$ is the heat flux vector.

$$q_j = -\lambda T_{ij} + \rho D \sum_{k=1}^{N} H_k Y_{k,j}$$

Here, $\lambda$ and $D$ are the thermal conductivity and mass diffusivity, respectively. $H_k$ is the total enthalpy of species $k$, $Y_k$ is the mass fraction for the species $k$, and $\omega_k$ is the reaction rate for the species $k$. Example problems in this report do not include reacting flows.

Solution of the Navier–Stokes equations is obtained using the Taylor–Galerkin finite element method. Details of the solution procedure are found in [7].

### 2.2 Entropy-Controlled Stability Equation

Suppose that the Navier–Stokes solution has been obtained with the results exhibiting sawtooth waves. Our objective is to determine whether such waves are stable or unstable. To this end we examine the conservation form of the energy equation,

$$\frac{\partial}{\partial t} (\rho E) + (\rho E v_i - \sigma_{ij} v_j)_i = 0 \quad (2)$$

where the comma implies partial derivatives and $\sigma_{ij}$ is the stress tensor,

$$\sigma_{ij} = -P \delta_{ij} + \mu \left( v_{ij} + v_{ji} - \frac{2}{3} v_j v_{ji} \right) \quad (3)$$

From thermodynamic relations it can be shown (appendix A) that

$$\rho E_{v_j} = \frac{P}{\rho} \rho v_i + \frac{P}{\rho} S_{v_j} + \rho v_j v_{ji} \quad (4)$$

where $S$ is the specific enthalpy per unit mass. Substituting (4) into (2) yields

$$\frac{\partial}{\partial t} (\rho E) + E(\rho v_i)_i + v_i \left[ \frac{P}{\rho} \rho v_i + \frac{P}{\rho} S_{v_j} + \rho v_j v_{ji} \right] - (\sigma_{ij} v_j)_i = 0 \quad (5)$$

This is the entropy-controlled-energy equation, instrumental in determining the nonlinear instability.
Assuming that the Navier–Stokes solutions for density \( \rho \), pressure \( p \), and velocity \( v_i \) represent the sum of mean and fluctuation parts, we write

\[
\begin{align*}
\rho &= \bar{\rho} + \rho' \quad (6) \\
p &= \bar{p} + p' \quad (7) \\
v_i &= \bar{v}_i + v'_i \quad (8)
\end{align*}
\]

where the symbols, bar and prime, denote the mean and perturbation quantities, respectively.

From thermodynamic relations we may write the entropy difference in the form

\[
S - S_0 = R \ln \left[ \left( 1 + \frac{\rho'}{\rho_0} \right)^{\gamma-1} \left( 1 + \frac{p'}{p_0} \right)^{-\frac{\gamma}{\gamma-1}} \right]
\]

or

\[
S = R \left[ S_{(1)} + S_{(2)} + S_{(3)} + S_{(4)} \right] + S_0 \quad (9)
\]

where \( S_0 \) represents the entropy at the initial state and \( S_{(i)} \) are given in Appendix B.

Our objective is to establish quantitative criteria whether the system is stable or unstable when we are provided with the Navier–Stokes solution exhibiting wave oscillations during unsteady motions. To this end, let \( \epsilon \) be the energy growth factor, \( \epsilon \geq 0 \) with \( \epsilon = 1 \) indicating the neutral stability. We then substitute (6) through (9) into (5), expand each term of the energy equation in terms of \( \epsilon \), integrate by parts (or using Green–Gauss theorem), and take time averages.

Writing (5) in an integral form

\[
\left\langle \int_{\Omega} \left[ \frac{\partial}{\partial t} (\rho E) + E(\rho v_i)_i + v_i \left( \frac{p}{\rho} \right)_i + \frac{P}{k} S_{,i} + \rho v_j v_j, i - (\sigma_{ij} v_j)_i \right] \, d\Omega \right\rangle = 0
\]

Integrating (10) by parts,

\[
\left\langle \int_{\Omega} \frac{\partial}{\partial t} (\rho E) \, d\Omega + \int_{\Gamma} \left[ E_{,i} v_i n_i + v_i \left( \frac{p}{\rho} \right)_i + \frac{P}{k} S n_i + \rho v_j v_j n_i - \sigma_{ij} v_j n_i \right] \, d\Gamma \right. \\
- \left. \int_{\Omega} \left[ E_{,i} \rho v_i + (v_i \frac{p}{\rho})_i + (v_i \frac{P}{k})_i S + (\rho v_j v_j)_i v_j \right] \, d\Omega \right\rangle = 0
\]

(11)
where \( \langle \cdot \rangle \) implies time averages. A typical term in (11) for multiples of two or more variables appears in the form

\[
\left\langle \int_{\Omega} (\cdot) \, d\Omega \right\rangle = \left\langle \int_{\Omega} (\delta_0 + \epsilon \delta_1 + \epsilon^2 \delta_2 + \epsilon^3 \delta_3 + \ldots) \, d\Omega \right\rangle
\]  

(12)

Here \( \delta_0 \) term contains only the mean quantity, \( \delta_1 \), the first order perturbation, \( \delta_2 \), the second order perturbation, etc. See detailed derivations in Appendix C.

It follows from (12) that the perturbed acoustic equation takes the form

\[
\frac{\partial}{\partial t} (\epsilon^2 \mathbf{E}_1 + \epsilon^3 \mathbf{E}_2 + \epsilon^4 \mathbf{E}_3) = \epsilon^2 \mathbf{I}_1 + \epsilon^3 \mathbf{I}_2 + \epsilon^4 \mathbf{I}_3
\]  

(13)

Thus, finally, the entropy-controlled stability equation becomes (See Appendix C)

\[
\frac{d\epsilon}{dt} - \alpha_1 \epsilon - \alpha_2 \epsilon^2 - \alpha_3 \epsilon^3 = 0
\]  

(14)

where \( \alpha_1, \alpha_2, \) and \( \alpha_3 \) are growth rate parameters of first, second and third order, respectively.

\[
\alpha_1 = \frac{1}{2 \mathbf{E}_1} \mathbf{I}_1
\]  

(15a)

\[
\alpha_2 = \frac{1}{2 \mathbf{E}_1} (\mathbf{I}_2 - \frac{3 \mathbf{E}_2}{2 \mathbf{E}_1} \mathbf{I}_1)
\]  

(15b)

\[
\alpha_3 = \frac{1}{2 \mathbf{E}_1} \left\{ \mathbf{I}_3 - \frac{3 \mathbf{E}_2}{2 \mathbf{E}_1} + \left[ \frac{9}{4} (\frac{\mathbf{E}_2}{\mathbf{E}_1})^2 - \frac{2 \mathbf{E}_3}{\mathbf{E}_1} \right] \mathbf{I}_1 \right\}
\]  

(15c)

with

\[
\mathbf{E}_1 = \left\langle \int_{\Omega} a^{(1)} \, d\Omega \right\rangle
\]  

(16a)

\[
\mathbf{E}_2 = \left\langle \int_{\Omega} a^{(2)} \, d\Omega \right\rangle
\]  

(16b)

\[
\mathbf{E}_3 = \left\langle \int_{\Omega} a^{(3)} \, d\Omega \right\rangle
\]  

(16c)

\[
\mathbf{I}_1 = \left\langle \int_{\Omega} b^{(1)} \, d\Omega \right\rangle - \left\langle \int_{\Gamma} c^{(1)} \, n_i \, d\Gamma \right\rangle
\]  

(17a)
The basic ingredients of integrands in Eq. (15) are the data from Navier-Stokes solutions. The mean quantities are obtained as time averages of Navier-Stokes solutions within suitable time segments and the fluctuation (perturbation) quantities are the differences between the Navier-Stokes solutions and their time averages.

To gain an insight into a solution of Eq. (14), we may neglect the last two terms of the left hand side of Eq. (14) and write
\[
\frac{d\epsilon}{dt} - \alpha_1 \epsilon = 0
\]  
(18)

which yields a solution in the form
\[
\ln \epsilon = \alpha_1 t + c_1
\]  
(19)

To establish an initial condition we assume a neutral stability \( \epsilon = 1 \) at \( t = 0 \). This gives \( c_1 = 0 \). Thus, the solution takes the form
\[
\epsilon = e^{\alpha_1 t}
\]  
(20)

Under this initial condition, there exists a unique solution for any given \( \alpha_1 \) with \( t > 0 \). It then follows that for stability we have \( 0 \leq \epsilon \leq 1 \) for \( -\infty \leq \alpha_1 \leq 0 \); for instability \( 1 < \epsilon < \infty \) for \( 0 < \alpha_1 < \infty \).
Although these criteria are not applicable for the nonlinear equation (Eq. 14), similar initial conditions as postulated above can be used. That is, there exists a unique solution $\epsilon$ for any given $\alpha_1$, $\alpha_2$, and $\alpha_3$ with $t > 0$.

Solutions of the nonlinear equation (Eq. 14) may be obtained using Newton–Raphson iterations. To this end, the residual of Eq. (14) is written as

$$R_{n+1,r} = \epsilon_{n+1,r} - \epsilon_n - \frac{\Delta t}{2} \left[ \alpha_1 (\epsilon_{n+1,r} + \epsilon_{n,r}) + \alpha_2 (\epsilon_{n+1,r}^2 + \epsilon_{n,r}^2) + \alpha_3 (\epsilon_{n+1,r}^3 + \epsilon_{n,r}^3) \right]$$

(21)

The Newton–Raphson process for Eq. (24) takes the form

$$J_{n+1,r} \Delta \epsilon_{n+1,r+1} = - R_{n+1,r}$$

(22)

where the Jacobian $J_{n+1,r}$ becomes

$$J_{n+1,r} = \frac{\partial R_{n+1,r}}{\partial \epsilon_{n+1,r}} = 1 - \frac{\Delta t}{2} (\alpha_1 + 2\alpha_2 \epsilon_{n+1,r} + 3\alpha_3 \epsilon_{n+1,r}^2)$$

(23)

and

$$\Delta \epsilon_{n+1,r+1} = \epsilon_{n+1,r+1} - \epsilon_{n+1,r}$$

(24)

Thus for each iterative step, we have

$$\epsilon_{n+1,r+1} = \epsilon_{n,r} + \Delta \epsilon_{n+1,r+1}$$

(25)

The initial value for $\epsilon$ begins with $\epsilon_{n,r} = 0$ and $\epsilon_{n+1,r} = 1$. Iterations continue until convergence.

3. SOLUTION PROCEDURE

To solve the nonlinear ordinary differential equation (14), we proceed as follows:

(1) With appropriate boundary and initial conditions, solve the Navier–Stokes equations using a numerical scheme capable of handling shock
discontinuities. Obtain $p$, $v_i$, and $\rho$. The Taylor–Galerkin Finite Element method is used in this study.

(2) Advance time steps ($\Delta t$) of Navier–Stokes solutions to obtain wave oscillations to cover at least one wave period.

(3) Take time averages for the period $n\Delta t$ (the range of $n$ is approximately, $15 < n < 150$, depending on frequencies $f$, $n$ is small if $f$ is high), corresponding to $\bar{p}$, $\bar{v}_i$, and $\bar{\rho}$.

(4) Calculate the fluctuation quantities as $p' = p - \bar{p}$, $v'_i = \bar{v}_i - v_i$, etc., where $p$, $\bar{v}_i$, and $v_i$ represent Navier–Stokes solutions.

(5) Calculate the growth rate parameters $\alpha_1$, $\alpha_2$, and $\alpha_3$ from (13a,b,c).

(6) Solve the nonlinear ordinary differential equation (14) using the Newton–Raphson method with a suitable initial guess for $\epsilon$. Ideally begin with $\epsilon = 1$, neutral stability.

(7) Repeat steps 1 through 4 until the desired length of time has been advanced.

Note that for each time–average period in step 4, above, instability and stability are determined by $\epsilon > 1$ and $\epsilon < 1$, respectively, with $\epsilon = 1$ being the neutral stability. If the system is found to be unstable, then it is not necessary to proceed to the next time step. However, for the entire ranges of time for which Navier–Stokes solutions are available, the stability analysis may be performed if desired, even if instability has been found in previous time steps. This is so because Navier–Stokes solutions are independent of the stability analysis as formulated here. Rather, the stability analysis in this formulation determines the state of stability or instability based on the current flowfield as calculated from the Navier–Stokes solution.
4. APPLICATIONS

Our objective here is to prove validity of the present theory for combustion instability analysis. To this end, one dimensional nonreacting flow has been chosen for the geometry of SSME thrust chamber with cross section area variations taken into account (Fig. 1).

The initial and boundary conditions for the Navier–Stokes solution consists of:

**Pressure**

\[ p = \bar{p} + d \bar{p} \sin (\omega t + \theta_0) \]

% disturbance, \( d = 10, 20, 30\% \)

mean pressure, \( \bar{p} = 500, 2,000, 2,935 \text{ psi} \)

frequency, \( \omega = 2\pi f \geq a/2L \) (\( L \) = distance between inlet and nozzle throat)

**Velocity (inlet)**

\( u = Ma \) (\( M = 0.2 \))

**Temperature**

\( T = 1000^\circ R \) for \( p = 500 \text{ psi} \)

\( T = 4000^\circ R \) for \( p = 2000 \text{ psi} \)

\( T = 6550^\circ R \) for \( p = 2935 \text{ psi} \)

Other constants used in this analysis are:

**Specific heat ratio** \( \gamma = 1.2 \)

**Mesh size** \( \Delta x = 1.685 \times 10^{-2} \text{ m} \)

**Courant Number** \( \text{C.N.} = 0.6 \)

The computational time increment \( \Delta t \) is calculated at each time step of Navier–Stokes solution from the Courant number. Time averages for calculation of energy growth rate parameters \( \alpha_1, \alpha_2, \) and \( \alpha_3 \) are calculated over 15 to 120 intervals of Navier–Stokes \( \Delta t \)'s to cover at least an average of one peak at any grid point. For simplicity, viscosity is ignored in this example problem. The computer program listing is given in Appendix E.

The Navier–Stokes solutions were obtained using the Taylor–Galerkin finite elements. Formulations of this method have been well documented and accuracy verified in the literature [7].
In Figs. 2 through 15, for each mean pressure and each % disturbance, the pressure and velocity oscillations are shown at various locations, \( x = -0.31, 0.05 \) m, and 3.05 m, along with the corresponding energy growth factors versus time.

In Figs. 2 and 3, \( (\bar{p} = 500 \text{ psi}, d = 10\%, T = 1000^\circ R) \), we note that it takes 0.0163 sec. for the pressure at \( x = 3.06 \) m to begin decreasing and for the velocity to increase from zero. Notice that shock waves develop around \( t = 0.14 \) sec., but stability is maintained \((\epsilon < 1)\) throughout since the mean pressure and disturbance are small.

The response due to \( \bar{p} = 500 \text{ psi}, d = 30\%, T = 1000^\circ R \), Figs. 4 and 5, is very similar to the case for \( d = 10\% \). Although the shock waves grow in magnitude and the energy growth factors increase, the system is still stable \((\epsilon < 1)\).

In Figs. 6 and 7, \( (p = 1000 \text{ psi}, d = 20\%, T = 4000^\circ R) \), shock waves grow and the energy growth factors reach almost the level of neutral stability. But, instability has not been observed.

The first instability has arrived at \( p = 2000 \text{ psi}, d = 30\%, T = 4000^\circ R \), Figs. 8 and 9, in the time interval, \( 0.045 < t < 0.6 \) sec, where sawtooth type shock waves at \( x = 3.06 \) m are prominent.

In Figs. 10 and 11, in which the pressure is raised to \( p = 2935 \text{ psi} \) with \( T = 6550^\circ R \), but disturbances are lowered to \( d = 10\% \), the system recovers stability.

With the disturbances raised to \( d = 20\% \), \( p = 2935 \text{ psi} \), however, Figs. 12 and 13, notice that the energy growth factor rises sharply at \( t = 0.05 \) sec, where pressure decreases to almost zero, but shock waves rise rapidly. However, the instability for this case is not as severe as when pressure was lower (2000 psi) but disturbance was large (30\%), as seen in Figs. 8 and 9.

The most severe instability occurs when the disturbances are raised to \( d = 30\% \), with \( \bar{p} = 2935 \text{ psi} \), Figs. 14 and 15. Notice that instability is spread over the wide time
range $0.045 < t < 0.06$ sec., rather than a single peak for the case of $d = 20\%$ above. Similar situation existed for $d = 30\%$ with $\bar{p} = 2000$ psi. It appears that instability is more sensitive to the increase in \% disturbances than mean pressure.

In Figs. 16 through 19, the energy growth rate parameters $\alpha_1$, $\alpha_2$, and $\alpha_3$ versus time are shown. When stable, the sum of $\alpha_1$, $\alpha_2$, and $\alpha_3$ is negative for the case of $\bar{p} = 500$ psi and $d = 10\%$ (Fig. 16). If unstable, however, the sum is positive for cases of $\bar{p} = 2000$ psi and $d = 30\%$ (Fig. 17), $\bar{p} = 2935$ psi and $d = 20\%$ (Fig. 18), and $\bar{p} = 2935$ psi and $d = 30\%$ (Fig. 19). Notice that as pressure increases the distribution of energy growth parameters become oscillatory. It is important to realize, however, that $\alpha_1$ represents a linear instability whereas $\alpha_2$ and $\alpha_3$ contribute to the nonlinear instability as controlled by entropy.

5. CONCLUSIONS

To our knowledge, the full scale Navier–Stokes solutions combined with rigorous determinations of stability or instability during any time step of unsteady Navier–Stokes solutions have been carried out for the first time. The key to this success lies in the fact that the entropy is induced in the acoustic energy equation. It is shown that entropy is calculated automatically, contributing to the shock waves and instability. For small disturbances and low pressures the effect of entropy is negligible whereas it is activated freely when the mean pressures and disturbances are increased.

To demonstrate the validity of the theory, the space shuttle main engine thrust chamber geometry was adopted for one dimensional flow but with cross section area variations taken into account. The computational results indicate that instability ($\epsilon > .1$) arises first when the mean pressure is raised to $2000$ psi with $30\%$ disturbances. Instability also arises when the mean pressure is set at $2935$ psi with $20\%$ disturbances. The system with mean pressures and disturbances more adverse than these quantities are shown to be unstable.
6. RECOMMENDATIONS

Based on the studies reported herein the following recommendations are provided:

(1) Extend the calculations to two-dimensional, axisymmetric cylindrical, and three dimensional geometries.

(2) Investigate effects of chemical kinetics.

(3) Investigate effects of Reynolds number (viscosity).

(4) Investigate effects of atomization, vaporization, and spray droplet combustion.

(5) Investigate effects of radiative heat transfer.

In summary, it is the opinion of this principal investigator that the present theory has a great potential and all avenues of further studies will prove to be fruitful.
Acknowledgement

Dr. Y. M. Kim contributed to the Navier–Stokes solution. Derivations of explicit forms of the stability integrals and computer programs for stability analysis were carried out by Mr. W. S. Yoon. Discussions of technical developments with Klaus Gross and John Hutt, contract monitor, are appreciated.
REFERENCES


Fig. 1  Geometry for one-dimensional Navier-Stokes solutions - SSME thrust chamber with variations of cross-section area taken into account.
Fig. 2 Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $p = 500$ psi, $d = 10\%$, $T = 1000^\circ R$. 
Fig. 3 Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for \( P = 500 \text{ psi}, \ d = 10\%, \ T = 1000^\circ\text{R}. \)
Fig. 4  Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $\bar{p} = 500$ psi, $d = 30\%$, $T = 1000^\circ$R.
Fig. 5  Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $\bar{p} = 500$ psi, $d = 30\%$, $T = 1000^\circ R$. 
Fig. 6 Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for \( p = 2000 \) psi, \( d = 20\% \), \( T = 4000^\circ R \).
Fig. 7  Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $p = 2000$ psi, $d = 20\%$, $T = 4000^\circ R.$
Fig. 8 Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $p = 2000$ psi, $d = 30\%$, $T = 4000^\circ R$. 

(a) $x = 0.31m$

(b) $x = 0.05m$

(c) $x = 3.06m$

(d)
Fig. 9 Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $\bar{p} = 2000$ psi, $d = 30\%$, $T = 4000^\circ R$. 
Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $P = 2935$ psi, $d = 10\%$, $T = 6550^\circ R$. 

![Figure 10](image)
Fig. 11 Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $p = 2935$ psi, $d = 10\%$, $T = 6550^\circ R$. 
Fig. 12 Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $\bar{p} = 2935$ psi, $d = 20\%$, $T = 6550^\circ R$. 
Fig. 13 Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $\overline{p} = 2935$ psi, $d = 20\%$, $T = 6550^\circ R$. 
Fig. 14 Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (ε) versus time for \( p = 2935 \) psi, \( d = 30\% \), \( T = 6550^\circ R \).
Fig. 15 Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors (ε) versus time for $p = 2935$ psi, $d = 30\%$, $T = 6550^\circ R$. 
Fig. 1 Geometry for one-dimensional Navier-Stokes solutions - SSME thrust chamber with variations of cross-section area taken into account.
Fig. 2 Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for \( \bar{p} = 500 \text{ psi}, d = 10\%, T = 1000^\circ\text{R.} \)
Fig. 3 Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $p = 500$ psi, $d = 10\%$, $T = 1000^\circ R$. 
Fig. 4 Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (ε) versus time for $\bar{p} = 500$ psi, $d = 30\%$, $T = 1000^\circ$R.
Fig. 5  Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $p = 500$ psi, $d = 30\%$, $T = 1000^\circ R$. 
Fig. 6 Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $\bar{p} = 2000$ psi, $d = 20\%$, $T = 4000^\circ R$. 
Fig. 7 Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors ($\epsilon$) versus time for $p = 2000$ psi, $d = 20\%$, $T = 4000^\circ R$. 
Fig. 8 Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (ε) versus time for $\bar{p} = 2000$ psi, $d = 30\%$, $T = 4000^\circ$R.
Fig. 9 Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors versus time for $p = 2000$ psi, $d = 30\%$, $T = 4000°$R.
Fig. 10 Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (ε) versus time for p = 2935 psi, d = 10%, T = 6550°R.
Fig. 11  Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $\bar{p} = 2935$ psi, $d = 10\%$, $T = 6550^\circ$R.
Fig. 12 Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (ε) versus time for $\bar{p} = 2935$ psi, $d = 20\%$, $T = 6550^\circ R$. 
Fig. 13 Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors (ε) versus time for \( p = 2935 \) psi, \( d = 20\% \), \( T = 6550^\circ R \).
Fig. 14 Navier-Stokes solutions for pressure (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $\bar{p} = 2935$ psi, $d = 30\%$, $T = 6550^\circ R$. 
Fig. 15 Navier-Stokes solutions for velocity (a), (b), (c) at various locations and (d) energy growth factors (e) versus time for $\bar{p} = 2935$ psi, $d = 30\%$, $T = 6550^\circ R$. 
Fig. 16  Energy growth rate parameters $\alpha_1$, $\alpha_2$, $\alpha_3$ versus time, $p = 500$ psi, $d = 10\%$, stable system ($\varepsilon < 1$), sum of $\alpha_1$, $\alpha_2$, $\alpha_3$ less than zero.
Fig. 17 Energy growth rate parameters $\alpha_1$, $\alpha_2$, $\alpha_3$ versus time, $\bar{p} = 200$ psi, $d = 30\%$, unstable system ($\epsilon > 1$), sum of $\alpha_1$, $\alpha_2$, $\alpha_3$ larger than zero.
Fig. 18 Energy growth rate parameters $\alpha_1$, $\alpha_2$, $\alpha_3$ versus time, $\bar{p} = 2935$ psi, $d = 20\%$, unstable system ($\varepsilon > 1$), sum of $\alpha_1$, $\alpha_2$, $\alpha_3$ larger than zero.
Fig. 19 Energy growth rate parameters $\alpha_1$, $\alpha_2$, $\alpha_3$, versus time, $\bar{p} = 2935$ psi, $d = 30\%$, unstable system ($\varepsilon > 1$), sum of $\alpha_1$, $\alpha_2$, $\alpha_3$ larger than zero.
APPENDIX A

DERIVATION OF ENERGY GRADIENTS IN TERMS OF ENTROPY GRADIENTS

From an ideal gas law

\[ \frac{P}{P_o} = \left( \frac{\rho}{\rho_o} \right)^\gamma \exp \left( \frac{S-S_o}{c_v} \right) \]

or

\[ \ln \left( \frac{P}{P_o} \right) = \ln \left( \frac{\rho}{\rho_o} \right)^\gamma + \frac{S-S_o}{c_v} \]

Differentiating

\[ \frac{1}{p} p_{,i} = \frac{1}{\rho} (\rho \gamma)_{,i} + \frac{1}{c_v} S_{,i} \]

or

\[ p_{,i} = c_o^2 \rho_{,i} + \frac{\rho c_o^2}{c_p} S_{,i} \]  \hspace{1cm} (A.1)

Now the gradient of the stagnation energy becomes

\[ E_{,i} = (c_p T - \frac{P}{\rho} + \frac{1}{2} v_j v_j)_{,i} \]

or

\[ E_{,i} + \frac{c_v}{R \rho} p_{,i} - \frac{c_v}{R} \frac{P}{\rho} \rho_{,i} + v_j v_{j,i} \]  \hspace{1cm} (A.2)

Substituting (A.1) into (A.2), we obtain

\[ \rho E_{,i} = \frac{P}{\rho} \rho_{,i} + \frac{P}{R} S_{,i} + \rho v_j v_{j,i} \]  \hspace{1cm} (A.3)
Thus

\[ S - S_o = R \ln \left[ \left(1 + \frac{P'}{\rho} \right)^{\frac{1}{\gamma - 1}} \left(1 + \frac{\rho'}{\rho} \right)^{\frac{\gamma}{\gamma - 1}} \right] \]

\[ = R \left[ \frac{1}{\gamma - 1} \ln \left(1 + \frac{P'}{\rho} \right) - \frac{\gamma}{\gamma - 1} \ln \left(1 + \frac{\rho'}{\rho} \right) \right] \]

\[ = R \left\{ \frac{1}{\gamma - 1} \left\{ \frac{P'}{\rho} - \frac{1}{2} \left( \frac{P'}{\rho} \right)^2 + \frac{1}{6} \left( \frac{P'}{\rho} \right)^3 - \frac{1}{24} \left( \frac{P'}{\rho} \right)^4 \ldots \right\} \right\} \]

\[ - \frac{1}{\gamma - 1} \left\{ \frac{\rho'}{\rho} - \frac{1}{2} \left( \frac{\rho'}{\rho} \right)^2 + \frac{1}{6} \left( \frac{\rho'}{\rho} \right)^3 - \frac{1}{24} \left( \frac{\rho'}{\rho} \right)^4 \ldots \right\} \]

\[ = R \left\{ \left( \frac{1}{\gamma - 1} \frac{P'}{\rho} - \frac{\gamma}{\gamma - 1} \frac{\rho'}{\rho} \right) \right\} \]

\[ - \frac{1}{2} \left[ \frac{1}{\gamma - 1} \left( \frac{P'}{\rho} \right)^2 - \frac{\gamma}{\gamma - 1} \left( \frac{\rho'}{\rho} \right)^2 \right] \]

\[ + \frac{1}{6} \left[ \frac{1}{\gamma - 1} \left( \frac{P'}{\rho} \right)^3 - \frac{\gamma}{\gamma - 1} \left( \frac{\rho'}{\rho} \right)^3 \right] \]

\[ - \frac{1}{48} \left[ \frac{1}{\gamma - 1} \left( \frac{P'}{\rho} \right)^4 - \frac{\gamma}{\gamma - 1} \left( \frac{\rho'}{\rho} \right)^4 \right] \]  

(B.1)

Thus

\[ S = R \left[ S_{(1)} + S_{(2)} + S_{(3)} + S_{(4)} \right] + S_o \]  

(B.2)

where

\[ S_{(1)} = \left[ \frac{1}{\gamma - 1} \frac{P'}{\rho} - \frac{1}{\gamma - 1} \frac{\rho'}{\rho} \right] \]

\[ S_{(2)} = -\frac{1}{2} \left[ \frac{1}{\gamma - 1} \left( \frac{P'}{\rho} \right)^2 - \frac{1}{\gamma - 1} \left( \frac{\rho'}{\rho} \right)^2 \right] \]

\[ S_{(3)} = \frac{1}{6} \left[ \frac{1}{\gamma - 1} \left( \frac{P'}{\rho} \right)^3 - \frac{1}{\gamma - 1} \left( \frac{\rho'}{\rho} \right)^3 \right] \]

\[ S_{(4)} = -\frac{1}{48} \left[ \frac{1}{\gamma - 1} \left( \frac{P'}{\rho} \right)^4 - \frac{1}{\gamma - 1} \left( \frac{\rho'}{\rho} \right)^4 \right] \]  

(B.3)
APPENDIX C

DERIVATION OF INTEGRODIFFERENTIAL EQUATION FOR ENTROPY INDUCED ENERGY GROWTH

From Eq. (11) and Eq. (A.3) the energy equation takes the form

\[
\frac{\partial}{\partial t} \left( \rho E \right) = - E \left( \sigma v_i \right)_i - v_i \left[ \frac{P}{\rho} \rho_i + \frac{P}{\rho R} S_{ii} + \rho v_i v_j v_{ji} \right] + (\sigma_{ij} v_j)_i \\
= - E(\rho v_i)_i - \frac{P v_i}{\rho R} S_{ii} - \frac{P v_i}{\rho} \rho_i - \rho v_i v_j v_{ji} + (\sigma_{ij} v_j)_i \\
= - \left( E \rho v_i \right)_i + \rho v_i E_{ii} - \frac{1}{R} (P v_i S)_i + \frac{1}{R} S (P v_i)_i - \frac{v_i}{\rho} \rho_i \\
- \rho v_i v_j v_{ji} + (\sigma_{ij} v_j)_i \\
= \left[ - \left( E \rho v_i \right)_i - \frac{1}{R} (P v_i S)_i + (\sigma_{ij} v_j)_i \right] \\
+ \left[ \rho v_i E_{ii} + \frac{1}{R} S (P v_i)_i - \frac{P v_i}{\rho} \rho_i - \rho v_i v_j v_{ji} \right]
\]

Integrating the above over the domain \( \Omega \) and boundary \( \Gamma \) and taking the time averages

\[
\left\langle \int_{\Omega} \frac{\partial}{\partial t} \left( \rho E \right) d\Omega \right\rangle = \left\langle \int_{\Omega} \left[ \rho v_i E_{ii} + \frac{1}{R} S (P v_i)_i - \frac{P v_i}{\rho} \rho_i - \rho v_i v_j v_{ji} \right] d\Omega \right\rangle \\
+ \left\langle \int_{\Gamma} \left[ - \rho v_i E - \frac{1}{R} P v_i S + \sigma_{ij} v_j \right] S_i d\Gamma \right\rangle
\]

(C.1)

where \( \langle \rangle \) denotes the time average. That is

\[
\langle \rangle = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \text{dt}
\]

Note also that

\[
\frac{P}{\rho} = \frac{\bar{P} + \bar{P}'}{\bar{\rho} + \bar{\rho}'} = \frac{\bar{P} + \bar{P}'(\bar{\rho} - \rho')}{\bar{\rho} + \bar{\rho}'(\bar{\rho} + \rho')} \\
= \frac{\bar{P} - \bar{\rho} \rho' + \bar{\rho} \rho - \rho' \rho'}{\bar{\rho}^2 - \rho'^2} \\
= \frac{\bar{P} - \bar{\rho} \rho' + \bar{\rho} \rho - \rho' \rho'}{\bar{\rho}^2 - \rho'^2} (\bar{\rho}^2 + \rho'^2)
\]

(C.2)
The numerator becomes

\[(\bar{p}^2 - p^2) (\bar{p}^2 + p^2) = \bar{p}^4 - p^4\]

Neglecting \(p^4\) (small) we have

\[
\frac{p}{\bar{p}} = \frac{1}{\bar{p}^4} \left[ \bar{p} \bar{p}^3 + (+ \bar{p}^3 p' - \bar{p} \bar{p}^2 p'') + (- \bar{p}^2 p'\rho + \bar{p} \bar{p} \rho') + (+ \bar{p} p' \rho^2 - \bar{p} \rho^3) + (- p' \rho^3) \right]
\]

(C.4)

Thus

\[
E = \frac{1}{\gamma - 1} \frac{p}{\rho} + \frac{1}{2} v_j v_j
\]

\[
= \bar{e} + e(1) + e(2) + e(3) + e(4)
\]

(C.5)

where

\[
\bar{e} = \frac{1}{\gamma - 1} \frac{\bar{p}}{\bar{p}} + \frac{1}{2} \bar{v}_j \bar{v}_j
\]

\[
e(1) = \frac{1}{\gamma - 1} \left( \frac{p'}{\bar{p}} - \bar{p}^3 p' \rho + \bar{v}_j v_j \right)
\]

\[
e(2) = - \frac{1}{\gamma - 1} \left( \frac{1}{\rho^2} p' \rho' - \frac{\bar{p}}{\bar{p}^3} \rho^2 \right) + \frac{1}{2} v_j v_j
\]

\[
e(3) = \frac{1}{\gamma - 1} \left( \frac{p' \rho^2}{\rho^3} - \frac{\bar{p}}{\bar{p}^4} \rho^3 \right)
\]

\[
e(4) = - \frac{1}{\gamma - 1} \frac{p' \rho^3}{\rho^4}
\]

(C.6)

It follows from (C.2) through (C.6) that

\[
\rho E = (\bar{p} + p') \left( \bar{e} + e(1) + e(2) + e(3) + e(4) \right)
\]

\[
\rho E = \bar{p} \bar{e} + e(\bar{p} e(1) + p' \bar{e}) + \epsilon^2(\bar{p} e(2) + p' \bar{e}(1))
\]

\[
+ \epsilon^3 (\bar{p} e(3) + p' \bar{e}(2)) + \epsilon^4(\bar{p} e(4) + p' \bar{e}(3))
\]

\[
= \frac{1}{\gamma - 1} \bar{p} + \frac{1}{2} \bar{p} \bar{v}_j \bar{v}_j
\]

\[
+ \epsilon \left[ \frac{1}{\gamma - 1} \left( \frac{p' - \bar{p}}{\bar{p}} \rho' \right) + \bar{p} \bar{v}_j v_j + \frac{1}{\gamma - 1} \frac{\bar{p}}{\rho} \rho' + \frac{\bar{p}'}{2} \bar{v}_j v_j \right]
\]

\[
+ \epsilon^2 \left[ - \frac{1}{\gamma - 1} \left( \frac{1}{\rho} p' \rho - \bar{p} \bar{p}^2 \rho' \right) + \bar{p} v_j v_j + \frac{1}{\gamma - 1} \frac{p' \rho'}{\rho^2} \rho' \right] + \rho' \bar{v}_j v_j
\]
where the energy growth factor $\epsilon$ was introduced with powers corresponding to the number of multiples of perturbed variables.

Similarly,

$$\rho v_{ij_E} = (\bar{v}_i + v_i) (\tilde{\rho} + \rho')(\bar{e} + e_{(1)} + e_{(2)} + e_{(3)} + e_{(4)})_i$$

$$= \bar{v}_i \bar{\rho} \bar{e}_i$$

$$+ \epsilon \left[ \bar{\rho} \bar{e}_i v_i + \bar{v}_i (\bar{\rho} e_{(1)} + \rho' \bar{e}_i) \right]$$

$$+ \epsilon^2 \left[ v_i (\bar{\rho} e_{(1)} + \rho' \bar{e}_i) + \bar{v}_i (\bar{\rho} e_{(2)} + \rho' e_{(1)}) \right]$$

$$+ \epsilon^3 \left[ v_i (\bar{\rho} e_{(2)} + \rho' e_{(1)}) + \bar{v}_i (\bar{\rho} e_{(3)} + \rho' e_{(2)}) \right]$$

$$+ \epsilon^4 \left[ v_i (\bar{\rho} e_{(3)} + \rho' e_{(2)}) + \bar{v}_i (\bar{\rho} e_{(4)} + \rho' e_{(3)}) \right]$$

$$= \epsilon (\bar{v}_i \bar{\rho}) S_{(1)}$$

$$+ \epsilon^2 \left[ S_{(1)} (\bar{v}_i \bar{\rho}) + S_{(2)} (\bar{v}_i \bar{\rho}) + S_{(1)} (p' \bar{v}_i) \right]$$

$$+ \epsilon^3 \left[ S_{(3)} (\bar{v}_i \bar{\rho}) + S_{(2)} (\bar{v}_i \bar{\rho}) + S_{(1)} (p' \bar{v}_i) \right]$$

$$+ \epsilon^4 \left[ S_{(4)} (\bar{v}_i \bar{\rho}) + S_{(3)} (\bar{v}_i \bar{\rho}) + S_{(2)} (p' \bar{v}_i) \right]$$
\[
\frac{p_i}{\rho} v_i \rho_{i1} = \left[ \frac{p}{\rho} + \left( \frac{p_i'}{\rho} - \frac{p_i}{\rho^2} \rho' \right) + \left( - \frac{p_i'}{\rho^2} + \frac{p_i}{\rho^3} \rho'' \right) \rho' \right]

+ \left( \frac{p_i'}{\rho^3} - \frac{p_i}{\rho^4} \rho^3 \right) \left[ \bar{v}_i \rho_{i1} + v_i' \rho_{i1} + v_i \rho_{i1} + v_i' \rho_{i1} \right]

+ \epsilon \left[ \frac{p}{\rho} (\bar{v}_i \rho_{i1} + v_i' \rho_{i1}) + v_i \rho_{i1} \left( \frac{p_i'}{\rho^2} - \frac{p_i}{\rho^3} \rho' \right) \right]

+ \epsilon^2 \left[ \frac{p}{\rho} (v_i' \rho_{i1}) + \left( \frac{p_i'}{\rho^2} - \frac{p_i}{\rho^3} \rho' \right) (v_i' \rho_{i1} + v_i \rho_{i1}) \right]

+ \epsilon^3 \left[ \left( \frac{p_i'}{\rho^2} + \frac{p_i}{\rho^3} \rho'' \right) \bar{v}_i \rho_{i1} \right]

+ \epsilon^4 \left[ \left( - \frac{p_i'}{\rho^2} + \frac{p_i}{\rho^3} \rho'' \right) v_i' \rho_{i1} + \left( \frac{p_i'}{\rho^3} - \frac{p_i}{\rho^4} \rho^3 \right) (v_i' \rho_{i1} + v_i \rho_{i1}) \right]

+ \left( - \frac{p_i'}{\rho^4} \bar{v}_i \rho_{i1} \right] \right)

(\text{C.11})

\rho v_i v_j v_{j1} = [\rho + \rho'] [\bar{v}_i + v_i] [v_j + v_j] [v_{j1} + v_{j1}] = \left[ \bar{p} \bar{v}_i + \rho' \bar{v}_i + \bar{p} v_i' + \rho' v_i' \right] [v_j \bar{v}_{j1} + v_j v_{j1} + v_j v_{j1} + v_j v_{j1}]

= \bar{p} \bar{v}_i v_j v_{j1} + \epsilon \left[ \bar{p} \bar{v}_i (v_j \bar{v}_{j1} + v_j v_{j1}) + (\rho \bar{v}_i + \rho v_i') v_j \bar{v}_{j1} \right]

+ \epsilon^2 \left[ \bar{p} v_i v_j v_{j1} + (\rho \bar{v}_i + \rho v_i') [v_j \bar{v}_{j1} + v_j v_{j1}] + \rho v_i v_j \bar{v}_{j1} \right]

+ \epsilon^3 \left[ (\rho' \bar{v}_i + \rho v_i') v_i v_j v_{j1} + \rho' v_i v_j \bar{v}_{j1} \right]

+ \epsilon^4 \left[ p' v_i v_{j1} \right]
\[ \rho v_i E = \bar{\rho} \bar{v}_i + \epsilon [\bar{\rho} \bar{v}_i' + \bar{v}_i (\bar{\rho} e_{(1)} + \rho' \bar{e})] \\
+ \epsilon^2 [v_i' (\bar{\rho} e_{(2)} + \rho' \bar{e}) + \bar{v}_i (\bar{\rho} e_{(2)} + \rho' e_{(1)})] \\
+ \epsilon^3 [v_i' (\bar{\rho} e_{(2)} + \rho' e_{(1)}) + \bar{v}_i (\bar{\rho} e_{(3)} + \rho' e_{(2)})] \\
+ \epsilon^4 [v_i' (\bar{\rho} e_{(3)} + \rho' e_{(2)}) + \bar{v}_i (\bar{\rho} e_{(4)} + \rho' e_{(3)})] \]  
(C.12)

\[ p v_i S = \epsilon (\bar{p} \bar{v}_i S_{(1)}) \\
+ \epsilon^2 [S_{(1)} \bar{p} v_i' + S_{(2)} \bar{p} v_i + S_{(1)} p' \bar{v}_i] \\
+ \epsilon^3 [S_{(2)} \bar{p} v_i + S_{(2)} \bar{p} v_i' + S_{(1)} p' v_i + \bar{v}_i p'] \\
+ \epsilon^4 [S_{(2)} \bar{p} v_i + S_{(3)} \bar{p} v_i' + S_{(1)} p' v_i + \bar{v}_i p'] \]  
(C.13)

\[ p v_i = (\bar{p} + p)(\bar{v}_i + v_i) = \bar{p} \bar{v}_i + \epsilon (\bar{p} v_i' + p' \bar{v}_i) + \epsilon^2 p' v_i' \]  
(C.14)

\[ \sigma_{ij} = \bar{\sigma}_{ij} + \sigma'_{ij} \]  
(C.15)

where

\[ \bar{\sigma}_{ij} = -\bar{p} \delta_{ij} + \mu (\bar{v}_{i,j} + \bar{v}_{j,i}) - \frac{2}{3} \mu \bar{v}_{k,k} \delta_{ij} \]  
(C.16)

\[ \sigma'_{ij} = -p' \delta_{ij} + \mu (v_{i,j} + v_{j,i}) - \frac{2}{3} \mu v_{k,k} \delta_{ij} \]  
(C.17)

Thus,

\[ \sigma_{ij} v_j = (\bar{\sigma}_{ij} + \sigma'_{ij})(v_j + v_j') = \bar{\sigma}_{ij} \bar{v}_j + \epsilon (\sigma_{ij} v_j' + \sigma'_{ij} \bar{v}_j) + \epsilon^2 \sigma'_{ij} v_j' \]  
(C.18)

Substituting the above relations into (C.1) yields

\[ \frac{\partial}{\partial t} [\epsilon^2 E_1 + \epsilon^3 E_2 + \epsilon^4 E_3] = \epsilon^2 I_1 + \epsilon^3 I_2 + \epsilon^4 I_2 \]  
(C.19)

where

\[ E_i = \left\{ \int_{\Omega} \left[ \frac{\bar{p}}{2} v_j v_j' + \rho' \bar{v}_j v_j' \right] d\Omega \right\} \]  
(C.20)
\[ E_2 = \langle \int_{\Omega} \left[ \frac{1}{2} \rho' v'_i v'_j \right] \, d\Omega \rangle \]

\[ E_3 = \langle \int_{\Omega} \left[ -\frac{\bar{p}}{\rho^4} \rho'^4 \right] \, d\Omega \rangle \]

\[ I_1 = \langle \int_{\Omega} \left[ \bar{v}_i (\bar{\rho} e_{(1)}),, _{1} + \rho' \bar{e}_i, + \bar{v}_i (\bar{\rho} e_{(2)}), _{1} + \rho' e_{(1) i} \right] \]

\[ + S_{(1)} (\bar{p} v_i')_{,1} + S_{(2)} (\bar{p} v_i'), \rangle_{,1} + S_{(1)} (p' v_i'), + \left\{ \frac{\bar{p}}{\rho} (v_i' \rho ,, i) \right\} \]

\[ + \left( \frac{p' \bar{p}}{\rho^2} \rho', (v_i' \rho',, i + \bar{v}_i \bar{\rho},, i) \right) + \left( \frac{p' \rho'}{\rho^2} + \frac{\bar{p}}{\rho} \rho'^2 \right) \bar{v}_i \bar{\rho},, i \right\} \]

\[ - \left\{ \bar{p} v_i v_j',, i + (\rho \bar{v}_i + \bar{p} v_i') (v_j v_i',, i + \bar{v}_j v_i') + \rho' \bar{v}_i v_j v_i',, i \right\} \rangle \]

\[ + \langle \int_{\Gamma} \left[ - \left\{ v_i' (\bar{\rho} e_{(1)} + \rho' \bar{e}) + \bar{v}_i (\bar{\rho} e_{(2)} + \rho' e_{(1)}) \right\} \right] \]

\[ - \left\{ S_{(1)} \bar{p} v_i' + S_{(2)} \bar{p} v_i' \right\} + \left\{ - p' \delta_{ij} + \mu (v_{i,i} + v_{j,i}) - \frac{2}{3} \mu v_{k,k} \delta_{ij} \right\} \cdot v_{i,j} \rangle n_i \, d\Gamma \]

\[ I_2 = \langle \int_{\Omega} \left[ \{ v_i' (\bar{\rho} e_{(2)})',, _{1} + \rho' \bar{e}_{(2)} ,, _{1} \} + \bar{v}_i (\bar{\rho} e_{(3)}),, _{1} + \rho' e_{(2) i} \right] \}

\[ + \left\{ S_{(3)} (\bar{p} v_i'),, _{1} + S_{(2)} (\bar{p} v_i'),, _{1} + S_{(1)} (p' v_i'),, _{1} \right\} \]

\[ - \left\{ (-\frac{p'}{\rho} \rho') v_i',, _{1} + (-\frac{p' \rho'}{\rho^2} + \frac{\bar{p}}{\rho} \rho'^2)(\bar{v}_i \rho',, _{1} + v_i \bar{\rho},, _{1}) \right\} \]

\[ + \left\{ \frac{p' \rho'^2}{\rho^3} - \frac{\bar{p}}{\rho} \rho'^3 \right\} \bar{v}_i \bar{\rho},, _{1} \]

\[ - \left\{ (\rho' \bar{v}_i + \bar{p} v_i') v_j v_j',, _{1} + p' v_i' (v_j v_j',, _{1} + \bar{v}_j v_j') \right\} \rangle \}

\[ + \langle \int_{\Gamma} \left[ - \left\{ v_i' (\bar{\rho} e_{(2)} + \rho' \bar{e}) + \bar{v}_i (\bar{\rho} e_{(3)} + \rho' e_{(2)}) \right\} \right] \]

\[ - \left\{ S_{(3)} \bar{p} v_i' + S_{(2)} \bar{p} v_i' + S_{(1)} p' v_i' \right\} \rangle n_i \, d\Gamma \]

(C.21)

(C.22)

(C.23)

(C.24)
\[ I_3 = \left\langle \int_{\Omega} \left\{ v'_i (\bar{p} \epsilon_{(3)} i + \rho' e_{(2)} i) + \bar{v}_i (\bar{p} \epsilon_{(4)} i + \rho' e_{(3)} i) \right\} \right. \]

\[ + \left\{ S_{(4)} (\bar{p} \bar{v}_i) i + S_{(3)} (\bar{p} v'_i) i + S_{(2)} (p' v'_i) i \right\} \]

\[ - \left\{ \frac{p' \rho'}{\rho^2} + \frac{\bar{p}}{\rho^3} \rho' v'_i i + \left( \frac{p' \rho' \rho}{\rho^3} - \frac{\bar{p}}{\rho^4} \rho^3 \right) (\bar{v}_i \rho'_i i + v'_i \bar{\rho}_i i) \right. \]

\[ + \left. \left( \frac{p' \rho'}{\rho^4} \bar{v}_i \bar{\rho}_i i \right) - [p' v'_i; v'_j i] \right\} d\Omega \]

\[ + \left\langle \int_{\Gamma} \left\{ - v'_i (\bar{p} \epsilon_{(3)} i + \rho' e_{(2)} i) + \bar{v}_i (\bar{p} \epsilon_{(4)} i + \rho' e_{(3)} i) \right\} \right. \]

\[ - \left\{ S_{(4)} \bar{p} \bar{v}_i + S_{(3)} \bar{p} v'_i + S_{(2)} p' v'_i \right\} n_i d\Gamma \right\rangle \]

(C.25)

Performing the differentiation as implied in (C.19), we obtain

\[ \frac{\partial \epsilon}{\partial t} = \frac{\epsilon^2 I_1 + \epsilon^3 I_2 + \epsilon^4 I_3}{2 \epsilon E_1 + 3 \epsilon^2 E_2 + 4 \epsilon^3 E_3} = \left( \epsilon I_1 + \epsilon^2 I_2 + \epsilon^3 E_3 \right) \frac{1}{2 E_1} \left( 1 - \epsilon \right) \frac{3 E_2}{2 E_1} \]

\[ + \epsilon^2 \left[ \frac{9}{4} \left( \frac{E_2}{E_1} \right) - \frac{2 E_3}{E_1} \right] \]

(C.26)

where higher order terms and those terms much smaller than unity have been neglected.

Thus, finally, we obtain

\[ \frac{d \epsilon}{dt} - \alpha_1 \epsilon - \alpha_2 \epsilon^2 - \alpha_3 \epsilon^3 = 0 \]

(C.27)
APPENDIX D

INTEGRANDS OF $E_1, E_2, E_3, I_1, I_2, I_3$

\[ a^{(1)} = \frac{\bar{p}}{2} v_j v'_j + \rho \bar{v}_j v'_j \]

\[ a^{(2)} = \frac{1}{2} \rho' v_j v'_j \]

\[ a^{(3)} = \frac{\bar{p}}{\bar{\rho}^4} \rho'^4 \]

\[ b^{(1)} = (v_i \bar{p} + \rho' \bar{v}_i) \left( \frac{1}{\gamma - 1} \left( \frac{\bar{p}' - \bar{p}}{\bar{\rho}^2} \rho' \right) + \bar{v}_j v'_j \right)_{,i} + v_j \rho' + \left( \frac{1}{\gamma - 1} \frac{\bar{p}}{\bar{\rho}} + \frac{1}{2} \bar{v}_j \bar{v}_j \right)_{,i} \]

\[ + \bar{p} \bar{v}_i \left( - \frac{1}{\gamma - 1} \left( \frac{\bar{p}' - \bar{p}}{\bar{\rho}^2} \rho' \right) + \frac{1}{2} v_j v'_j \right)_{,i} + (p v_i' + p v_i)_{,i} \left( \frac{1}{\gamma - 1} \frac{\bar{p}'}{\bar{\rho}} - \frac{\gamma}{\gamma - 1} \frac{\bar{p}'}{\bar{\rho}} \right) \]

\[ + (\bar{p} v_i)'_{,i} (- \frac{1}{2} \left( \frac{\bar{p}'}{\bar{\rho}^2} - \frac{\gamma}{\gamma - 1} \frac{\bar{p}' - \bar{p}}{\bar{\rho}^2} \rho' \right) + \frac{1}{2} v_j v'_j)_{,i} + (p v_i' + p v_i)_{,i} - (\frac{1}{\gamma - 1} \frac{\bar{p}'}{\bar{\rho}} - \frac{\gamma}{\gamma - 1} \frac{\bar{p}'}{\bar{\rho}} \rho') (v_i v_i'_{,i} + \bar{v}_i v_j v_j'_{,i}) \]

\[ + v_i' \bar{v}_i_{,i} - (\frac{1}{\gamma - 1} \frac{\bar{p}'}{\bar{\rho}^2} - \frac{1}{2} \frac{\bar{p}'}{\bar{\rho}^3} \rho' \bar{\rho}'_{,i} - \bar{p} \bar{v}_i v_j v_j'_{,i} - (\rho' \bar{v}_i + p v_i') (v_j v_j'_{,i} + \bar{v}_j v_j'_{,i}) \]

\[ - \rho' v'_j \bar{v}_j v_j'_{,i} \]

\[ c^{(1)} = (v_i \bar{p} + \rho' \bar{v}_i) \left( \frac{1}{\gamma - 1} \left( \frac{\bar{p}' - \bar{p}}{\bar{\rho}^2} \rho' \right) + \bar{v}_j v'_j \right) + v_j \rho' + \left( \frac{1}{\gamma - 1} \frac{\bar{p}}{\bar{\rho}} + \frac{1}{2} \bar{v}_j \bar{v}_j \right) \]

\[ + \bar{p} \bar{v}_i \left( - \frac{1}{\gamma - 1} \left( \frac{\bar{p}' - \bar{p}}{\bar{\rho}^2} \rho' \right) + \frac{1}{2} v_j v'_j \right)_{,i} + (p v_i' + p v_i)_{,i} \left( \frac{1}{\gamma - 1} \frac{\bar{p}'}{\bar{\rho}} \right) \]

\[ - \frac{\gamma}{\gamma - 1} \frac{\bar{p}'}{\bar{\rho}} + (\bar{p} v_i)'_{,i} (- \frac{1}{2} \left( \frac{\bar{p}'}{\bar{\rho}^2} - \frac{\gamma}{\gamma - 1} \frac{\bar{p}' - \bar{p}}{\bar{\rho}^2} \rho' \right) + \frac{1}{2} p' \delta_{ij} - \mu (v_i v_j' + v_j v_j'_{,i}) \]

\[ + \frac{2}{3} \mu v_k v_k v_j' n_j \]
\[ b^{(2)} = (v_i \bar{v} + \rho \bar{v} \nu_i)(- \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \nu^i)}{\rho^2} + \frac{1}{2} v_j \nu_j) + \rho \nu_i \left( \frac{1}{\gamma - 1} \frac{(\nu^i - \bar{v}^i \rho)}{\rho^2} \right) + \bar{v}_j \nu_j \nu_i + \bar{v} \nu_i \left( \frac{1}{\gamma - 1} \frac{(\nu^i - \bar{v}^i \rho)}{\rho^3} \right) + \frac{1}{6} \left[ \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^3} \right] \bar{v} \nu_i \nu_i \]

\[ \frac{1}{2} \left[ \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^2} \right] (\bar{p} \nu_i + \nu_i \bar{p}) \nu_i + \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^3} \bar{v} \nu_i \nu_i \]

\[ - (\rho \bar{v} \nu_i + \bar{v} \nu_i \nu_i - \rho \nu_i \nu_i) \nu_j \nu_i + \nu_j \nu_i \]

\[ c^{(2)} = (v_i \bar{v} + \rho \bar{v} \nu_i)(- \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^2} + \frac{1}{2} v_j \nu_j) + \rho \nu_i \left( \frac{1}{\gamma - 1} \frac{(\nu^i - \bar{v}^i \rho)}{\rho^2} \right) + \bar{v}_j \nu_j \nu_i \]

\[ + \bar{v} \nu_i \left( \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^3} \right) + \frac{1}{6} \left[ \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^3} \right] \bar{v} \nu_i \]

\[ \frac{1}{2} \left[ \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^2} \right] \nu_i \nu_i + \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^3} \bar{v} \nu_i \nu_i \]

\[ b^{(3)} = (v_i \bar{v} + \bar{v} \nu_i)(- \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^2} + \frac{1}{2} v_j \nu_j) + \rho \nu_i \left( \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^2} \right)

+ \frac{1}{2} v_j \nu_j \nu_i + \bar{v} \nu_i \left( \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^3} \right) + \frac{1}{6} \left[ \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^3} \right] \bar{v} \nu_i \nu_i \]

\[ - \frac{1}{2} \left[ \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^2} \right] \nu_i \nu_i + \frac{1}{6} \left[ \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^3} \right] \bar{v} \nu_i \nu_i \]

\[ - (\frac{\nu^i \nu^i - \bar{v}^i \rho}{\rho^2} \nu_i \nu_i + \frac{1}{48} \left[ \frac{1}{\gamma - 1} \frac{(\nu^i \nu^i - \bar{v}^i \rho)}{\rho^4} \right] \nu_i \nu_i \]

\[ - (\nu^i \nu^i - \bar{v}^i \rho \nu_i \nu_i + \nu_i \bar{v} \nu_i) \]

\[ - \nu_i \nu_i \nu_j ] \]

\[ - \nu_i \nu_i \nu_j ] \]

\[ - \nu_i \nu_i \nu_j ] \]
\[ c^{(3)} = (v_i \bar{\rho} + \bar{v}_i \rho') \frac{1}{\gamma - 1} \left( \frac{\rho' \rho^2}{\rho^3} - \frac{\bar{\rho} \rho^3}{\rho^4} \right) + \rho' v_i^i \left( \left( -\frac{1}{\gamma - 1} \right) \left( \frac{1}{\rho^2} p' \rho' - \frac{\bar{\rho}}{\rho^3} \rho'^2 \right) \right) \]

\[ + \frac{1}{2} v_i^i v_j^j - \bar{\rho} \bar{v}_i^i \frac{1}{\gamma - 1} \left( \frac{\rho' \rho^3}{\rho^4} \right) + \frac{1}{8} \left[ \frac{1}{\gamma - 1} \left( \frac{\rho'}{\bar{\rho}} \right)^3 - \frac{\bar{\rho}}{\rho^3} \left( \bar{\rho} \rho_i^i \right) \right] \bar{v}_i^i + p' \bar{v}_i^i \right),_i \]

\[ - \frac{1}{2} \left( \frac{1}{\gamma - 1} \left( \frac{\rho'}{\bar{\rho}} \right)^2 - \frac{\bar{\rho}}{\rho^3} \left( \bar{\rho} \rho_i^i \right) \right) - (p' \bar{v}_i^i) - \frac{1}{48} \left[ \frac{1}{\gamma - 1} \left( \frac{\rho'}{\bar{\rho}} \right)^4 - \frac{\bar{\rho}}{\rho^3} \left( \bar{\rho} \rho_i^i \right) \rho_n^i \right] \]
APPENDIX E
Listing of Computer Program (ECI-1)

PROGRAM TGID

PARAMETER (NELEM=200,NPOIN=201)
CALL DINPUT
CALL LPMASS
CALL ITERAT

STOP
END

SUBROUTINE DINPUT
PARAMETER (NELEM=200,NPOIN=201)
PARAMETER (NNODP=2,NGAUS=2,NCONS=3)
COMMON/AAAA/A(NPOIN),DA(NPOIN)
COMMON/DOMA/DO,UO,EO,PO
COMMON/INIT/DENSY(NPOIN),UVELY(NPOIN),ENEGY(NPOIN),PRESY(NPOIN)
COMMON/COOR/XX(NPOIN),LNODS(NELEM,NNODP)
COMMON/PRTY/CAPAV,CAPAP,CGAM,CONDT,VISCY
COMMON/TIME/CFLNB,DTIME,ITMAX
DIMENSION XI(100),AI(100)
DIMENSION EA(NELEM),EDA(NELEM)

READ IN FLOW PROPERTIES AND TEMPORAL PARAMETERS
READ(19,*) ITMAX
IRE_D=2
CGAM=1.22
CAF_V=1.0
CO*JT=0.0
VISCY=0.0
CF,NB=0.6
DTIME=0.025
WRITE(6,2010) CGAM,CAPAV,CONDT,VISCY
WRITE(6,2020) CFLNB,DTIME,ITMAX

READ IN NODAL CONNECTIVITIES
WRITE(6,2030)
DO 10 I=1,NELEM
LNODS(I,1)=I
LNODS(I,2)=I+1
WRITE(6,1000) I,LNODS(I,1),LNODS(I,2)
10 CONTINUE
1000 FORMAT(1X,I5,5X,2E15)

READ IN NODAL COORDINATES
CORD=0.0254*5.1527
XTH=0.0
ATH=3.14*CORD**2
IN=82
DO 151 I=1,IN
READ(17,152) II,XI(I),AI(I)
PRINT*,II,XI(I),AI(I)
XI(I)=CORD*XI(I)
AI(I)=ATH*AI(I)**2
PRINT*,I,XI(I),AI(I)
151 CONTINUE
152 FORMAT(5X,I5,2X,2E12.5)
DX=(XI(IN)-XI(1))/FLOAT(NELEM)
XX(1)=XI(1)
A(1)=AI(1)
XX(NPOIN)=XI(IN)
A(NPOIN)=AI(IN)
DO 20 I=1,NPOIN-1
XX(I+1)=XX(I)+DX
20 CONTINUE
DO 153 I=2,NPOIN-1
XA=XX(I)
DO 154 J=I,NPOIN-I
IF(XA.GE.XI(J).AND.XA.LE.XI(J+1)) THEN
SLOPE=(AI(J+1)-AI(J))/((XI(J+1)-XI(J))
A(I)=AI(J)+SLOPE*(XA-XI(J))
ENDIF
154 CONTINUE
153 CONTINUE
DO 5003 I=1,NELEM
EA(I)=0.5*(A(I)+A(I+1))
DXX=XX(I+1)-XX(I)
EDA(I)=(A(I+1)-A(I))/DXX
5003 CONTINUE
DO 5004 I=2,NPOIN-1
DA(I)=0.5*(EDA(I)+EDA(I-1))
5004 CONTINUE
DA(1)=EDA(1)
DA(NPOIN)=EDA(NELEM)
DO 85 I=1,NPOIN
C WRITE(6,2500) I,XX(I),A(I),DA(I)
85 CONTINUE
2500 FORMAT(1X,15,F10.5,3E15.5)
C READ IN INITIAL CONDITIONS
C C AIR PROPERTIES @ T=1000 K
REC=2.67E+5
CGAM=1.2
CAPAV=1.7
C CAPAV=1000.
READ(19,*) APRES,ATEMP
PATM=APRES/14.7
PSTG=PATM*9.8E4
CTEM=(ATEMP-460.)*5./9.+273
CMACH=0.2
CGAS=1.987*1000.*4.184
CGRAV=9.8
CWGT=0.79*28.+0.21*32.
CSND=SQR(CGAM*CTEM*CGAS/CWGT)
CVEL=CMACH*CSND
CSQR=0.5*CVEL**2
CAPAV=CAPAV*CGAS/CWGT
CENG=CAPAV*CTEM+CSQR
CPRE=PSTG
CRHO=CPRE/((CGAM-1.)*(CENG-CSQR))
CENT=CENG+CPRE/CRHO
C PRINT*,CSND,CVEL,CRHO,CENG,CPRE,CAPAV
C $TOP
C CAPAP=CGAM*CAPAV
DO=CRHO  
PO=CPRE  
UO=CVEL  
VO=0.0  
TO=CTEM  
EO=CENG  
HO=CENT  
CAPAP=CGAM*CAPAV  
DO 30 I=1,NPOIN  
PRESY(I)=PO  
UVELY(I)=0.0  
ENERGY(I)=CAPAV*TO+0.5*UVELY(I)**2  
DENSITY(I)=PRESY(I)/((CGAM-1.0)*(ENERGY(I)-0.5*UVELY(I)**2))  
30 CONTINUE  
PRESY(1)=PO  
UVELY(1)=UO  
ENERGY(1)=EO  
DENSITY(1)=DO  
C  
C RESTART PROCEDURES  
C  
IF(IREAD.EQ.1) THEN  
READ(II,1060) XX(I),I=1,NPOIN  
READ(II,1060) A(I),I=1,NPOIN  
READ(II,1060) DENSITY(I),I=1,NPOIN  
READ(II,1060) UVELY(I),I=1,NPOIN  
READ(II,1060) ENERGY(I),I=1,NPOIN  
READ(II,1060) PRESY(I),I=1,NPOIN  
ENDIF  
1060 FORMAT(5(200(4EI5.5,/)))  
C  
C WRITE OUT COORDINATES AND INITIAL CONDITIONS  
C  
C RETURN  
C  
2010 FORMAT(/'PHYSICAL PROPERTY',/' ***************' //  
- ' CGAM =',F7.4,4X,' CAPAV =',F7.4,4X,  
- ' CONDT =',F7.4,4X,' VISCY =',F7.4)  
2020 FORMAT(/'INITIAL TIME STEP',/' ***************' //  
- ' CFLNB =',F7.4,4X,' DTIME =',F7.4,4X,  
- ' ITMAX =',I5)  
2030 FORMAT(/'ELEMENT TOPOLOGY',/' ***************',//  
- ' ELEMENT',6X,'NODE NUMBERS')  
2040 FORMAT(/'NODE POINT DATA',/' ***************',//  
- ' 5X,'NODE',1X,'X',10X,'DENSY',5X,  
- 'UVELY',5X,'ENERGY',5X,'PRESY')  
2045 FORMAT(5X,14,3X,5(F7.4,3X))  
C  
C SUBROUTINE LPMASS  
PARAMETER (NELEM=200,NPOIN=201)  
PARAMETER (NNODP=2)  
COMMON/COOR/XX(NPOIN),LNODS(NELEM,NNODP)  
COMMON/MASS/GMASS(NPOIN)  
DIMENSION FI(2),POSGP(2),WEIGP(2)  
C  
C INITIALIZATION OF LUMPED MASS  
C
DO 10 I=1,NPOIN
   GMASS(I)=0.0
10 CONTINUE

C*** SET UP POSITIONS AND WEIGHTS FOR 2 POINT GAUSS RULE
C
   POSGP(1)=0.5773502691
   POSGP(2)=-POSGP(1)
   WEIGP(1)=1.0000000000
   WEIGP(2)=WEIGP(1)

C ASSEMBLE LUMPED MASS
C
DO 100 IELEM=1,NELEM
   SLETH=ABS(XX(LNODS(IELEM,2))-XX(LNODS(IELEM,1)))

C INTEGRATIONS
C
DO 90 IGAUS=1,2
   DJA=0.50*SLETH*WEIGP(IGAUS)
   XI=POSGP(IGAUS)
   FI(1)=0.50*(1.0-XI)
   FI(2)=0.50*(1.0+XI)

DO 30 I=1,NNODP
   K=LNODS(IELEM,I)
   SHAPX=FI(I)
   DO 30 J=1,NNODP
      SHAPY=FI(J)
      GMASS(K)=GMASS(K)+SHAPX*SHAPY*DJA
30 CONTINUE
90 CONTINUE
100 CONTINUE

C STORE IN THE OUTER-CORE MEMORY
C
RETURN
END

SUBROUTINE SDTIME
PARAMETER (NELEM=200,NPOIN=201)
PARAMETER (NNODP=2)
COMMON/AAAA/A(NPOIN),DA(NPOIN)
COMMON/AREA/AREAL(NELEM)
COMMON/COOR/XX(NPOIN),LNODS(NELEM,NNODP)
COMMON/INIT/DENSY(NPOIN),UVELY(NPOIN),ENEGY(NPOIN),PRESY(NPOIN)
COMMON/PRTY/CAPAV,CAPAP,CGAM,CONDT,VISCY
COMMON/TIME/CFLNB,DTIME,ITMAX
DIMENSION TIMEL(NELEM)
DIMENSION DENS(NNODP),PRES(NNODP),UVEL(NNODP),VVEL(NNODP)

C EVALUATE TIME STEP IN EACH ELEMENT
C
DO 10 ILELEM=1,NELEM
DO 20 J=1,NNODP
   K=LNODS(IELEM,J)
   DENS(J)=DENSY(K)
   UVEL(J)=UVELY(K)
   PRES(J)=PRESY(K)
20 CONTINUE
10 CONTINUE
20 CONTINUE
   DABSM=0.0
   UABSM=0.0
   PABSM=0.0
   AA=0.0
   DO 30 I=1,NNODP
      DABSM=DABSM+0.5*DENSM(I)
      UABSM=UABSM+0.5*UVELM(I)
      PABSM=PABSM+0.5*PRESM(I)
      AA=AA+0.5*A(LNODS(IELEM,I))
30 CONTINUE

C SLETH=ABS(XX(LNODS(IELEM,2))-XX(LNODS(IELEM,1)))
   UVABS=ABS(UABSM)
   CSPED=SQRT(CGAM*ABS(PABSM)/ABS(DABSM))
   TIMEL(IELEM)=CFLNB*SLETH/(UVABS+CSPED)
   TIMEL(IELEM)=CFLNB*SQRT(AREAL(IELEM))/(UVABS+CSPED)
10 CONTINUE

C FIND MINIMUM TIME STEP

C DTIME=TIMEL(1)
   CFLLL=TIMEL(1)
   DO 40 IELEM=2,NELEM
      IF(TIMEL(IELEM).LT.DTIME) DTIME=TIMEL(IELEM)
      IF(TIMEL(IELEM).GT.CFLLL) CFLLL=TIMEL(IELEM)
40 CONTINUE
   PRINT*,'CFLNUMBER == ',CFLLL
   RETURN
END

C SUBROUTINE MATRIX(IITER,IEQS,IELEM)
PARAMETER (NELEM=200,NPOIN=201)
PARAMETER (NNODP=2,NEQNS=3,NGAUS=2)
COMMON/AAA.A/A(NPOIN),DA(NPOIN)
COMMON/DOMA/DO,UO,EO,PO
COMMON/BCBC/DI,UI,EI,P1
COMMON/AREA/AREAL(NELEM)
COMMON/COOR/XX(NPOIN),LNODS(NELEM,NNODP)
COMMON/PRTY/CAPAV,CAPAP,CGAM,COND,T, VISCY
COMMON/TIME/CFLNB,DTIME,ITMAX
COMMON/INIT/DENSY(NPOIN),UVELY(NPOIN),E'EGY(NPOIN),PRESY(NPOIN)
COMMON/HALF/DENSH(NELEM),UVELH(NELEM),E'EGH(NELEM),PRESH(NELEM)
COMMON/EQNS/EQRHR(NPOIN),EQRHU(NPOIN),EQRHE(NPOIN)
DIMENSION POSGP(NGAUS),WEIGP(NGAUS),FI(2),DX(2)
DIMENSION UHALF(NEQNS),FHALF(NEQNS),FLUXH(NEQNS),RHALF(NEQNS)

C*** SET UP POSITIONS AND WEIGHTS FOR 2 POINT GAUSS RULE
C
   AH=0.5*(A(LNODS(IELEM,1)))+A(LNODS(IELEM,2))
   DAH=0.5*(DA(LNODS(IELEM,1))+DA(LNODS(IELEM,2)))
   POSGP(1)=0.5773502691
   POSGP(2)=-POSGP(1)
   WEIGP(1)=1.0000000000
   WEIGP(2)=WEIGP(1)

C LOOP TO CARRY OUT GAUSS INTEGRATION
C NOTE : PERFORMED JUST ONCE IN EACH TEMPORAL ITERATION
C
IF (IEQNS .NE. 1) GO TO 20

C AREAL(IELEM) = ABS(XX(LNODS(IELEM, 2)) - XX(LNODS(IELEM, 1)))
DO 10 J = 1, NEQNS
RHALLF(J) = 0.0
10 CONTINUE

C EVALUATE INTEGRATIONS IN THE RIGHT-HAND SIDE OF HALF STEP

C DO 70 IGAUS = 1, NGAUS

DJA = 0.50 * AREAL(IELEM) * WEIGP(IGAUS)
DTA = 0.50 * AREAL(IELEM)
XI = POSGP(IGAUS)
FI(1) = 0.50 * (1.0 - XI)
FI(2) = 0.50 * (1.0 + XI)
DX(1) = -0.50 / DTA
DX(2) = 0.50 / DTA

C EVALUATE PREVIOUS VARIABLES AND FLUXES AT GAUSS POINTS

C IN THE HALF STEP

C SOURCE = 0.0
DO 40 J = 1, NEQNS
UHF_LJF(J) = 0.0
FH_LJF(J) = 0.0
40 CONTINUE
DO 50 I = 1, NNODP
K = LNODS(IELEM, I)
UHALF(1) = UHALF(1) + DENSY(K) * A(K) * FI(I)
UHALF(2) = UHALF(2) + DENSY(K) * UVELY(K) * A(K) * FI(I)
UHALF(3) = UHALF(3) + DENSY(K) * ENEGY(K) * A(K) * FI(I)
FH_LJF(1) = FHALF(1) + DENSY(K) * UVELY(K) * A(K) * DX(I)
FHALF(2) = FHALF(2) + (DENSY(K) * UVELY(K) ** 2 + PRESY(K)) * A(K) * DX(I)
FHALF(3) = FHALF(3) + UVELY(K) * A(K) ** 2 + PRESY(K)) * A(K) * DX(I)
- *DX(I)
SOURCE = SOURCE + PRESY(K) * DA(K) * FI(I)
50 CONTINUE

C ORGANIZE RIGHT-HAND SIDE OF HALF STEP

C RHALF(1) = RHALF(1) + DJA * (UHALF(1) - 0.5 * DTIME * FHALF(1))
RHALF(2) = RHALF(2) + DJA * (UHALF(2) + 0.5 * DTIME * (SOURCE - FHALF(2))
RHALF(3) = RHALF(3) + DJA * (UHALF(3) - 0.5 * DTIME * FHALF(3))
70 CONTINUE

C CALCULATE EACH VARIABLE AT THE HALF STEP

DENSH(IELEM) = RHALF(1) / (AREAL(IELEM) * AH)
UVELH(IELEM) = RHALF(2) / (DENSH(IELEM) * AREAL(IELEM) * AH)
ENEGH(IELEM) = RHALF(3) / (DENSH(IELEM) * AREAL(IELEM) * AH)
PRESH(IELEM) = (CGAM - 1.0) * DENSH(IELEM)
- * (ENEGH(IELEM) - 0.5 * UVELH(IELEM) * UVELH(IELEM))
20 CONTINUE

C CALCULATE FLUX TERMS AT THE HALF STEP

FLUXH(1) = DENSH(IELEM) * UVELH(IELEM) * AH
FLUXH(2) = (DENSH(IELEM) * UVELH(IELEM) * UVELH(IELEM) + PRESH(IELEM)) * AH
FLUXH(3) = UVELH(IELEM) * (DENSH(IELEM) * ENEGH(IELEM) + PRESH(IELEM)) * AH
SORCH = PRESH(IELEM) * DAH

C EVALUATE INTEGRATIONS IN THE RIGHT-HAND SIDE AT THE FULL STEP
C
DO 80 IGAUS = 1, NGAUS
  DJA = 0.50 * AREAL(IELEM) * WEGP(IGAUS)
  DTA = 0.50 * AREAL(IELEM)
  XI = POSGP(IGAUS)
  FI(1) = 0.50 * (1.0 - XI)
  FI(2) = 0.50 * (1.0 + XI)
  DX(1) = -0.50 / DTA
  DX(2) = 0.50 / DTA
C
C EVALUATE RIGHT-HAND SIDE AT THE FULL STEP
C
DO 110 I = 1, NNODP
  K = LNODS(IELEM, I)
  CARXI = DX(I) * DJA * DTIME
  GO TO (111, 112, 113), IEQNS
111 EQRHR(K) = EQRHR(K) + FLUXH(1) * CARXI
  GO TO 110
112 EQRHU(K) = EQRHU(K) + FLUXH(2) * CARXI + SORCH * FI(I) * DJA * DTIME
  GO TO 110
113 EQRHE(K) = EQRHE(K) + FLUXH(3) * CARXI
110 CONTINUE
80 CONTINUE

RETURN
END

SUBROUTINE BDFLUX(EIQNS)
PARAMETER (NELEM = 200, NPOIN = 201, NNODP = 2)
COMMON/AAAA/A(NPOIN), DA(NPOIN)
COMMON/COOR/XX(NPOIN), LNODS(NELEM, NNODP)
COMMON/PRTY/CAPAV, CAPAP, CGAM, CONDT, VISCY
COMMON/TIME/CFLNB, DTIME, ITMAX
COMMON/INIT/DENSY(NPOIN), UVELY(NPOIN), ENEGY(NPOIN), PRESY(NPOIN)
COMMON/HALF/DENSH(NELEM), UVELH(NELEM), ENEGH(NELEM), PRESH(NELEM)
COMMON/EIQNS/EQRHR(NPOIN), EQRHU(NPOIN), EQRHE(NPOIN)

C
C EVALUATE AVERAGES INSIDE DOMAIN ELEMENT
C
K11 = LNODS(1, 1)
K12 = LNODS(1, 2)
KN1 = LNODS(NELEM, 1)
KN2 = LNODS(NELEM, 2)

DRDA1 = 0.5 * DENSY(K11) * UVELY(K11) * A(K11)
- 0.5 * DENSY(K12) * UVELY(K12) * A(K12)
DUDA1 = 0.5 * (DENSY(K11) * UVELY(K11) + UVELY(K11) + PRESY(K11)) * A(K11)
- 0.5 * (DENSY(K12) * UVELY(K12) + UVELY(K12) + PRESY(K12)) * A(K12)
DED1 = 0.5 * (DENSY(K11) * ENEGY(K11) + PRESY(K11)) * UVELY(K11) * A(K11)
- 0.5 * (DENSY(K12) * ENEGY(K12) + PRESY(K12)) * UVELY(K12) * A(K12)

DRDA2 = 0.5 * DENSY(KN1) * UVELY(KN1) * A(KN1)
- 0.5 * DENSY(KN2) * UVELY(KN2) * A(KN2)
DUDA2 = 0.5 * (DENSY(KN1) * UVELY(KN1) + UVELY(KN1) + PRESY(KN1)) * A(KN1)
- 0.5 * (DENSY(KN2) * UVELY(KN2) + UVELY(KN2) + PRESY(KN2)) * A(KN2)
DED2 = 0.5 * (DENSY(KN1) * ENEGY(KN1) + PRESY(KN1)) * UVELY(KN1) * A(KN1)
- 0.5 * (DENSY(KN2) * ENEGY(KN2) + PRESY(KN2)) * UVELY(KN2) * A(KN2)
- \[ -0.5 \times (DENSY(KN2) \times ENEGY(KN2) + PRESY(KN2)) \times UVELY(KN2) \times A(KN2) \]

**Evaluate Boundary Terms at the Half Step**

- \[ AH1 = 0.5 \times (A(K11) + A(K12)) \]
- \[ AHN = 0.5 \times (A(KN1) + A(KN2)) \]
- \[ DRDH1 = DENSH(1) \times UVELH(1) \times AH1 \]
- \[ DUDH1 = (DENSH(1) \times UVELH(1) \times UVELH(1)) + PRESH(1)) \times AH1 \]
- \[ DEDH1 = (DENSH(1) \times ENEGH(1) + PRESH(1)) \times UVELH(1) \times AH1 \]

- \[ DRDNS = DENSH(NELEM) \times UVEL(NELEM) \times AHN \]
- \[ DUDNS = (DENSH(NELEM) \times UVEL(NELEM) \times UVEL(NELEM)) + PRESY(NELEM)) \times AHN \]
- \[ DEDNS = (DENS(NELEM) \times ENEGY(NELEM) + PRESY(NELEM)) \times UVEL(NELEM) \times AHN \]

**Zero-Th Time Step**

- \[ DRDNI = DENSY(1) \times UVELY(1) \times A(1) \]
- \[ DUDNI = (DENSY(1) \times UVELY(1) \times UVELY(1)) + PRESY(1)) \times A(1) \]
- \[ DEDNI = (DENSY(1) \times ENEGY(1) + PRESY(1)) \times UVELY(1) \times A(1) \]

- \[ DRDNN = DENSY(NPOIN) \times UVELY(NPOIN) \times A(NPOIN) \]
- \[ DUDNN = (DENSY(NPOIN) \times UVELY(NPOIN) \times UVELY(NPOIN)) + PRESY(NPOIN)) \times A(NPOIN) \]
- \[ DEDNN = (DENSY(NPOIN) \times ENEGY(NPOIN) + PRESY(NPOIN)) \times UVELY(NPOIN) \times A(NPOIN) \]

**Include Boundary Gradient Terms into RHS Vector**

- \[ EQHR(1) = EQRHR(1) - DTIMEX(-DRDN1-DRDH1+DRDA1) \]
- \[ EQRHR(NPOIN) = EQRHR(NPOIN) + DTIMEX(-DRDNS-DRDNS+DRDNS) \]

- \[ EQRHU(1) = EQRHU(1) - DTIME*(-DUDN1-DUDH1+DUDA1) \]
- \[ EQRHU(NPOIN) = EQRHU(NPOIN) + DTIME*(-DUDNN-DUDHN+DUDN) \]

- \[ EQRHE(1) = EQRHE(1) - DTIME*(-DEDN1-DEDH1+DEDA1) \]
- \[ EQRHE(NPOIN) = EQRHE(NPOIN) + DTIME*(-DEDNN-DEDHN+DEDN) \]

**CONTINUE**

**RETURN**

**END**

**SUBROUTINE SOLVER(IEQNS,DELTA)**

PARAMETER (NELEM=200, NPOIN=201)

PARAMETER (NNODP=2, NCONS=3)

COMMON/MASS/GMASS(NPOIN)

COMMON/AREA/AREAL(NELEM)

COMMON/COOR/XX(NPOIN),LNODS(NELEM,NNODP)

COMMON/EQNS/EQRHR(NPOIN),EQRHU(NPOIN),EQRHE(NPOIN)

DIMENSION DELTA(NPOIN),EQHR(NPOIN),EQRHU(NPOIN),EQRHE(NPOIN)

DIMENSION POSGP(2),WEIGP(2),FI(2)

**SET UP POSITIONS AND WEIGHTS FOR 2 POINT GAUSS RULE**

- \[ POSGP(1) = 0.5773502691 \]
- \[ POSGP(2) = -POSGP(1) \]
- \[ WEIGP(1) = 1.0 \]
- \[ WEIGP(2) = -WEIGP(1) \]

**GO TO (1,2,3), IEQNS**
1 DO 5 I=1,NPOIN
2 EQRHS(I)=EQRHR(I)
3 GO TO 9
4 DO 6 I=1,NPOIN
5 EQRHS(I)=EQRHU(I)
6 GO TO 9
7 DO 7 I=1,NPOIN
8 EQRHS(I)=EQRHE(I)
9 CONTINUE

C C READ LUMPED MASS FROM STORED TAPE
C C SOLUTION PROCEDURE OF ALGEBRAIC EQUATIONS USING EXPLICIT
C C METHOD
C C - LUMPED MASS
C
IF(NCONS.EQ.1) THEN
  DO 200 I=1,NPOIN
  DELTA(I)=EQRHS(I)/GMASS(I)
200 CONTINUE
ENDIF

C C - JACOBI ITERATIONS
C
IF(NCONS.EQ.3) THEN
  DO 100 ICONS=1,NCONS
    IF(ICONS.NE.1) O0 TO 20
  DO I=1,NPOIN
    GDUMY(I)=0.0
20 CONTINUE
  DO 30 I=1,NPOIN
    CDUMY(I)=0.0
30 CONTINUE
ENDIF

C C COMPUTATION OF M*DU
C
DO 80 IELEM=1,NELEM

C LOOP TO CARRY OUT GAUSS INTEGRATION
C
DO 70 IGAUS=1,2

DJA=0.50*AREAL(IELEM)*WEIGP(IGAUS)
DFA=0.50*AREAL(IELEM)
XI=POSGP(IGAUS)
FI(1)=0.50*(1.0*XI)
FI(2)=0.50*(1.0+XI)

GINTP=0.0
DO 50 I=1,NNODP
  K=LNODS(IELEM,I)
  GINTP=GINTP+GDUMY(K)*FI(I)
50 CONTINUE
DO 60 I=1,NNODP
  K=LNODS(IELEM,I)
  CDUMY(K)=CDUMY(K)+GINTP*FI(I)*DJA
60 CONTINUE
70 CONTINUE
80 CONTINUE
C
C CALCULATION OF DELTA IN EVERY ITERATION
C
DO 90 I=1,NPOIN
   DELTA(I)=(EQRHS(I)-CDU_Pf(I))/GMASS(I)+GD_(I)
90 CONTINUE
C
DO 110 I=1,NPOIN
110 GDUMY(I)=DELTA(I)
C
100 CONTINUE
ENDIF
RETURN
END

C
SUBROUTINE LAPDUS(IEQNS)
PARAMETER (NELEM=200,NPOIN=201)
PARAMETER (NNODP=2)
COMMON/AAAA/A(NPOIN),DA(NPOIN)
COMMON/AREA/AREAL(NELEM)
COMMON/INIT/DENSY(NPOIN),UVELY(NPOIN),ENEGY(NPOIN),PRESY(NPOIN)
COMMON/EQNS/EQRHR(NPOIN),EQRHU(NPOIN),EQRHE(NPOIN)
COMMON/COOR/XX(NPOIN),LNODS(NELEM,NNODP)
COMMON/TIME/CFLNB,DTIME,ITMAX
DIMENSION X(2),U(2),FI(2),DX(2),POSGP(2),WEIGP(2)

C*** SET UP POSITIONS AND WEIGHTS FOR 2 POINT GAUSS RULE
C
POSGP(1)=0.5773502691
POSGP(2)=-POSGP(1)
WEIGP(1)=1.000000000
WEIGP(2)=WEIGP(1)

C COMPUTATION OF ARTIFICIAL VISCOSITIES USING 'APIDUS' CONCEPT
C
DO 100 IELEM=1,NELEM
C
C ARTIFICIAL VISCOSITIES
C
DO 10 I=1,NNODP
   K=LNODS(IELEM,I)
   X(I)=XX(K)
   U(I)=UVELY(K)
10 CONTINUE
C
DUDXA=ABS((U(2)-U(1))/(X(2)-X(1)))
C
C LOOP TO CARRY OUT GAUSS INTEGRATION
C
DO 100 IGAUS=1,2
C
DJA=0.50*AREAL(IELEM)*WEIGP(IGAUS)
DTA=0.50*AREAL(IELEM)
XI=POSGP(IGAUS)
FI(1)=0.50*(1.0-XI)
FI(2)=0.50*(1.0+XI)
DX(1)=-0.50/DTA
DX(2)= 0.50/DTA
C K=LNODS(IELEM,I)
DORDX=DDRDX+DENSY(K)*A(K)*DX(I)
DRUDX=DRUDX+DENSY(K)*UVELY(K)*A(K)*DX(I)
DREDX=DREDX+DENSY(K)*ENEGY(K)*A(K)*DX(I)
40 CONTINUE
C
C ARTIFICIAL VISCOSITY
C
CONSX=1.0*AREAL(IELEM)*AREAL(IELEM)*ABS(DUDX)
C
C EVALUATE RIGHT-HAND SIDE
C
DO 50 I=1,NNODP
K=LNODS(IELEM,I)
CARXI=DX(I)*DJA*CONSX*DTIME
EQHR(K)=EQHRK)-DDRDX*CARXI
EQHU(K)=EQHU(K)-DRUDX*CARXI
EQHE(K)=EQHE(K)-DREDX*CARXI
50 CONTINUE
100 CONTINUE
C
RETURN
END

C SUBROUTINE WRITER(IITER,RMSER,TSAVE)
C PAF_METER (NELEM=200,NPOIN=201)
C PARAMETER (NNODP=2)
C COMMON/INIT/DENSY(NPOIN),UVELY(NPOIN),ENEGY(NPOIN),PRESY(NPOIN)
C COMMON/COOR/XX(NPOIN),LNODS(NELEM,NNODP)
C COMMON/AAAA/A(NPOIN),DA(NPOIN)
C COMMON/TIME/CFLNB,DTIME,ITMAX
C DIMENSION PB(NPOIN),UB(NPOIN),RB(NPOIN)
C
C WRITING PROCEDURES
C
IF(IITER.EQ.1) THEN
READ(19,*) NUM
INUM=ITMAX[NUM
ENDIF
IF(IITER/200.*200.EQ.IITER) WRITE(6,1000) IITER,TSAVE,RMSER
write(18,1000) IITER,TSAVE,RMSER
NPIC1=2
NPIC2=NPOIN/2
NPIC3=NPOIN-1
NPIC=23
NPIC3=200
write(18,1010) TSAVE,PRESY(NPIC1),PRESY(NPIC2),PRESY(NPIC3),
& PRESY(NPIC4)
write(28,1010) TSAVE,UVELY(NPIC1),
1 UVELY(NPIC2),UVELY(NPIC3),UVELY(NPIC4)
1010 FORMAT(5E12.5)
IF(IITER.EQ.1) ICONT=NUM
IF(IITER.EQ.ICONT) THEN
ICONT=INUM
DO 333 I=1,NPOIN
PB(I)=0.0
UB(I)=0.0
RB(I)=0.0
TDIST=0.0
333 CONTINUE
ENDIF
DO 444 I=1,NPOIN
PB(I)=PB(I)+PRESY(I)*DTIME
UB(I)=UB(I)+UVELY(I)*DTIME
RB(I)=RB(I)+DENSY(I)*DTIME
444 CONTINUE
TDIST=TDIST+DTIME
IF(IITER.EQ.ICONT) THEN
DO i0 I=1,NPOIN
PB(I)=PB(I)/TDIST
UB(I)=UB(I)/TDIST
RB(I)=RB(I)/TDIST
WRITE(16,1020) I,PB(I),UB(I),RB(I)
i0 CONTINUE
DO 15 I=1,NPOIN
PB(I)=0.0
UB(I)=0.0
RB(I)=0.0
15 CONTINUE
TDIST=0.0
ICONT=IITER+INUM
ENDIF
C
C WRITE AT EACH ICONT-TH ITERATION
C
C
C WRITE IF SOLUTIONS ARE CONVERGED
C
IF(RMSER.GT.1.0E-05) GO TO 20
IF(IITER.EQ.1) GO TO 20
DO 30 I=1,NPOIN
WRITE(6,1020) I,XX(I),DENS_Y(I),UVELY(I),ENEGY(I),PRESY(I)
30 CONTINUE
WRITE(13,1060) (XX(I),I=1,NPOIN)
WRITE(13,1060) (A(I),I=1,NPOIN)
WRITE(13,1060) (DENS_Y(I),I=1,NPOIN)
WRITE(13,1060) (UVELY(I),I=1,NPOIN)
WRITE(13,1060) (ENEGY(I),I=1,NPOIN)
WRITE(13,1060) (PRESY(I),I=1,NPOIN)
STOP
20 CONTINUE
C
C WRITE IF IITER EQUALS TO ITMAX
C
IF(IITER.EQ.ITMAX) THEN
DO 40 I=1,NPOIN
WRITE(6,1020) I,XX(I),DENS_Y(I),UVELY(I),ENEGY(I),PRESY(I)
40 CONTINUE
WRITE(13,1060) (XX(I),I=1,NPOIN)
WRITE(13,1060) (A(I),I=1,NPOIN)
WRITE(13,1060) (DENS_Y(I),I=1,NPOIN)
WRITE(13,1060) (UVELY(I),I=1,NPOIN)
WRITE(13,1060) (ENEGY(I),I=1,NPOIN)
WRITE(13,1060) (PRESY(I),I=1,NPOIN)
C
WRITE(13,1060) (PRESY(I), I=1,NPOIN)
END
C
RETURN
1000 FORMAT(5X,I5,2X,(2(E10.5,1X)))
1020 FORMAT(5X,I5,F10.5,4E15.5)
1060 FORMAT(5(200(4E15.5,/)1))
END
C
SUBROUTINE ITERAT
PARAMETER (NELEM=200,NPOIN=201)
PARAMETER (NNODP=2,NEQNS=3)
PARAMETER (NTIME=2000)
PARAMETER (NTHNN=20,NTHEE=NTHNN-1)
PARAMETER (NTHNN=201,NTHEE=NTHNN-1)
COMMON/MASS/GMASS(NPOIN)
COMMON/AAAA/A(NPOIN),DA(NPOIN)
COMMON/DOMA/DO,UO,EO,PO
COMMON/BCBC/D1,U1,E1,P1
COMMON/AREA/AREAL(NELEM)
COMMON/INIT/DENSY(NPOIN),UVELY(NPOIN),ENEGY(NPOIN),PRESY(NPOIN)
COMMON/COOR/XX(NPOIN),LNODS(NELEM,NNODP)
COMMON/PRTY/CAPAV,CAPAP,CGAM,CONDT,VISCY
COMMON/TIME/CFLNB,DTIME,ITMAX
COMMON/EQNS/EQRHR(NPOIN),EQRHU(NPOIN),EQRHE(NPOIN)
DIMENSION DELTR(NPOIN),DELTU(NPOIN),DELTE(NPOIN),
-DENST(NPOIN),UVELT(NPOIN),ENEGT(NPOIN)
DIMENSION EQRHS(NPOIN)
DIMENSION DDS(NTIME,NTHNN),PPS(NTIME,NTHNN),UUS(NTIME,NTHNN)
DIMENSION NNSS(NTHEE,NNODP),XXS(NTHNN),AS(NTHNN)
DIMENSION PSTA(NTHNN),USTA(NTHNN),DSTEP(NTIME)
DIMENSION PBAR(NTHNN),UBAR(NTHNN),RBAR(NTHNN)
DIMENSION RSTA(NTHNN)
C
OPEN(16,FILE='stl6.dat')
NELS=NTHEE
NXS=NTHNN
NTS=NTIME
DO 441 I=1,NELS
DO 441 J=1,2
NNSS(I,J)=LNODS(I,J)
441 CONTINUE
DO 442 I=1,NXS
XXS(I)=XX(I)
AS(I)=A(I)
PSTA(I)=0.
USTA(I)=0.
RSTA(I)=0.
442 CONTINUE
RR1=DENSY(1)
UUI=UVELY(1)
EE1=ENEGY(1)
PPI=PRESY(1)
TT1=PPI/RR1/(CGAM-1.)*CAPAV
ASOUND=SQRT(CGAM*PPI/RR1)
AMACH=UUI/ASOUND
THLEN=XX(20)-XX(1)
C
THLEN=XX(NPOIN)-XX(1)
FREQ=3.14*ASOUND/THLEN
PRINT*, ASOUND, AMACH, FREQ

C SET UP ITERATION COUNTER AND LOOP ADDRESS
C
READ(19, *) CPERT
IITER = 0
ICOUN = 0
C
DDTM = 0.
TSAVE = 0.0
10 CONTINUE
IITER = IITER + 1
C
C SET UP TIME STEP (VARIABLE TIME STEP)
C
IF (IITER.EQ.1) TSAVE = DTIME
IF (IITER.GE.1) CALL SDTIME
DTIME = 1.0E-5
IF (IITER.GE.1) TSAVE = TSAVE + DTIME
C
RCOS = SIN (FREQ*TSAVE)
RCOS = 1.0
PERT = CPERT*PP1
NTRIG = 1000
IF (IITER.GE.NTRIG) PERT = CPERT*PP1
TPRE = PP1 + PERT*RCOS
CIGAM = 1./CGAM
PRRR = TPRE/PP1
PRINT*, I, RAD, RCOS, TPRE
ITEM = TTI/PRRR**CIGAM
TUUU = TUU
TSQR = 0.5*TUUU**2
TENG = CAPAV*TITEM+TSQR
TRHO = TPRE/((CGAM-1.)*(TENG-TSQR))
TENT = TENG+TPRE/TRHO
DENSY(1) = TRHO
UVELY(1) = TUUU
ENERGY(1) = TENG
PRESSY(1) = TPRE
C
C INITIALIZATIONS
C
GO TO (21, 22, 23). IEQNS
21 DO 25 I = 1, NPOIN
25 EQRHR(I) = 0.0
GO TO 20
22 DO 26 I = 1, NPOIN
26 EQRHU(I) = 0.0
GO TO 20
23 DO 27 I = 1, NPOIN
27 EQRHE(I) = 0.0
GO TO 20
20 CONTINUE
C
C ASSEMBLE CONTRIBUTIONS OF EACH ELEMENT TO THE RIGHT-HAND
C SIDE VECTORS
C
C
C DOMA\N CONTRIBUTIONS
C
DO 30 IELEM = 1, NELEM
CALL MATRIX(IITER,1,IELEM)
30 CONTINUE

C SURFACE CONTRIBUTIONS
C
CALL BDFLUX(1)

C MAIN MATRIX SOLVER USING ITERATIVE SCHEME
C
DO 90 IEQNS=1,NEQNS
   IF(IEQNS.EQ.1) CALL SOLVER(IEQNS,DELTR)
   IF(IEQNS.EQ.2) CALL SOLVER(IEQNS,DELTU)
   IF(IEQNS.EQ.3) CALL SOLVER(IEQNS,DELTE)
90 CONTINUE

C UPDATE SOLUTIONS
C
DO 100 I=1,NEQNS
   DENST(I)=DENSY(I)*A(I)+DELTR(I)
   UVELT(I)=UVELY(I)*A(I)+DELTU(I)
   ENEGT(I)=ENEGY(I)*A(I)+DELTE(I)
100 CONTINUE

DO 110 I=1,NPOIN
   DENST(I)=DENSY(I)/A(I)
   UVALY(I)=UVELT(I)/DENST(I)
   ENEGY(I)=ENEGT(I)/DENST(I)
   PRESY(I)=(CGAM-1.0)*DENSY(I)*(ENEGY(I)-0.5*UVELY(I)**2)
   PRINT*,I,DENSY(I),UVELY(I),ENEGY(I),PRESY(I)
110 CONTINUE

C CHECK THE CONVERGENCE
C
SUMUP=0.0
SUMDN=0.0
DO 115 I=1,NPOIN
   SUMUP=SUMUP+DELTR(I)**2+DELTU(I)**2+DELTE(I)**2
   SUMDN=SUMDN+DENST(I)**2+UVELT(I)**2+ENEGT(I)**2
115 CONTINUE
RMSR=SQRT(SUMUP/SUMDN)

C APPLY LAPIDUS' ARTIFICIAL VISCOSITY
C
DO 190 IEQNS=1,NEQNS
   GO TO (121,122,123), IEQNS
121 DO 125 I=1,NPOIN
125 EQRHR(I)=0.0
   GO TO 120
122 DO 126 I=1,NPOIN
126 EQRHU(I)=0.0
   GO TO 120
123 DO 127 I=1,NPOIN
127 EQRHE(I)=0.0
120 CONTINUE

   CALL LAPDUS(1)
   GO TO (140,150,160), IEQNS
140 DO 145 I=1,NPOIN
   DENST(I)=DENSY(I)*A(I)+EQRHR(I)/GMASS(I)
145 CONTINUE
DO 155 I=1,NPOIN
155 UVELT(I)=DENSY(I)*UVELY(I)*A(I)+EQRHU(I)/GMASS(I)
GO TO 180

DO 165 I=1,NPOIN
165 ENEGT(I)=DENSY(I)*ENEGY(I)*A(I)+EQRHE(I)/OMASS(I)
180 CONTINUE
190 CONTINUE

COMPUTE FINAL SOLUTIONS AT EACH TIME STEP

DO 200 I=1,NPOIN
200 DENSY(I)=DENST(I)/A(I)
UVELY(I)=UVELT(I)/DENST(I)
ENEYO(I)=ENEGT(I)/DENST(I)
PRESY(I)=(CGAM-I.0)*DENSY(I)*(ENEGY(I)-0.5*UVELY(I)**2)
CONTINUE

--- CASE A
D1=DO
UI=UO
EI=EO
P1=PO

--- CASE C
HO=(CGAM/(CGAM-I.0))*PO/DO+0.5*UO*UO
CGAM1=CGAM-1.0
R3=PO/DO**CGAM
D1=((CGAM1/CGAM)**(HO-0.5*UVELY(I)**2)/R3)**(1./CGAM1)
UI=UVELY(I)
P1=3*D1**CGAM
EI=UVELY(I)=U1
PRESY(I)=P1
ENEY(I)=EI

--- CASE C
SUBSONIC OUTLET BOUNDARY CONDITIONS
PRESY(NPOIN)=0.704
PRESY(NPOIN)=0.61845265

CALL WRITEER TO OUTPUT ITERATION RESULTS
CALL WRITEER(IITER,RMSER,TSAVE)
DDTM=DDTM+DTIME
NONE=ITMAX/2000
IA=IITER/NONE
IB=NONE*IA
IF(IITER.EQ.IB) THEN
ICOUN=ICOUN+1
DSTEP(ICOUN)=DDTM
DDTM=0.
DO 957 I=1,NXS
PP5(ICOUN,I)=PRESY(I)
UUS(ICOUN,I)=UVELY(I)
DSS(ICOUN,I)=DENSY(I)
957 CONTINUE
ENDIF
DO 443 I=1,NXS
   PSTA(I)=PSTA(I)+DTIME*PRESY(I)
   USTA(I)=USTA(I)+DTIME*UVELY(I)
   RSTA(I)=RSTA(I)+DTIME*DENSY(I)
443 CONTINUE
C
IF(IITER.LT.ITMAX) GO TO 10
DO 871 I=1,NXS
   PSTA(I)=PSTA(I)/TSAVE
   USTA(I)=USTA(I)/TSAVE
   RSTA(I)=RSTA(I)/TSAVE
871 CONTINUE
C
REWIND(16)
C
CALL STAB(NELS,NXS,NTS,XXS,AS,PPS,UUS,DDS,NNSS,DSTEP,
-PSTA,USTA,RSTA,PBAR,UBAR,RBAR)
RETURN
END
SUBROUTINE STAB(NELE,NX,NT,X,A,P,U,R,NEL,DSTEP,
-PSTA,USTA,RSTA,PBAR,UBAR,RBAR)
PARAMETER (NINT=2,L=2,NF=12)
COMMON/PARAMT/SOUND,GAMMA,ADMI,ADMO
COMMON/PVALUE/XLENG,TLENG,IPERT,PI,EPSI,VISCO
C
DIMENSION NEL(NELE,L)
DIMENSION X(NX),A(NX),P(NT,NX),U(NT,NX),R(NT,NX)
DIMENSION XI(NINT),WI(NINT),PHI(L,NINT)
DIMENSION DSTEP(NT),PSTA(NX),RSTA(NX),USTA(NX)
DIMENSION PBAR(NX),UBAR(NX),RBAR(NX)
C
IPERT=0
PI=3.141592654
GAMMA=1.2
ADMI=0.0
ADMO=0.0
VISCO=0.0
EPSI=0.2
C
XLENG=X(NX)-X(1)
TLENG=0.05
CALL GAUSS(NINT,XI, WI)
C
CALL SHAPE(NINT,XI, PHI)
C
CALL ABC(NELE,L,NF,NX,NINT,NT,NELS,NX,A,R,P,U,
- WI,PHI,AA,BB,CC,DSTEP,PSTA,USTA,RSTA,PBAR,UBAR,RBAR)
C
RETURN
END
C
C
SUBROUTINE ABC(NELE,L,NF,NX,NINT,NT,NELS,NX,A,R,P,U, WI, PHI, AA, BB, CC,
- DSTEP,PSTA,USTA,RSTA,PBAR,UBAR,RBAR)
COMMON/PARAMT/SOUND,GAMMA,ADMI,ADMO
COMMON/PVALUE/XLENG,TLENG,IPERT,PI,EPSI,VISCO
DIMENSION NEL(NELE,L),WI(NINT),PHI(L,NINT)
DIMENSION X(NX),A(NX),P(NT,NX),U(NT,NX),R(NT,NX)
DIMENSION E(3),G(3),ETA(3),EE(3),GG(3),ET(3)

C
DIMENSION PBAR(NX),UBAR(NX),RBAR(NX),DSTEP(NT)
DIMENSION PSTA(NX),RSTA(NX),USTA(NX)
DIMENSION ICAL(10000),IDAL(10000)

C
NXS=NX
NTS=NT

C
READ(19,*), NUM
NUM=2000/INUM
DO 941 I=1,NT
ICAL(I)=0
IA=I/NUM
IB=NUM*IA
IF(I.EQ.IB) THEN
ICAL(I)=I
ENDIF
CONTINUE
ICAL(1)=I
ICAL(NT)=0
AAA=0.0
BBB=0.0
CCC=0.0
ITER=0
ITER=ITER+1
PRINT*, 'ITER = ', ITER
DO 102 I=1,3
EE(I)=0.0
GG(I)=0.0
ET(I)=0.0
CONTINUE
 NT=100
STEP=TLENG/NT
DO 200 I=1,NT
T=(I-1)*STEP
PRINT*, (DSTEP(I), I=1,NT)
SOUND=0.0
RHHH=0.0
EZZZ=0.0

C
T=0.0
TZERO=T
NTRIG=600
DO 100 IITER=1,NT
IF(ICAL(IITER).EQ.1) THEN
DO 660 I=1,NX
READ(16,*), NUM,PSTA(I),USTA(I),RSTA(I)
CONTINUE
ENDIF
XSND=0.0
XRRR=0.0
XJUU=0.0
DO 443 I=1,NXS
PBAR(I)=P(IITER,I)
UBAR(I) = U(IITER, I)  
RBAR(I) = R(IITER, I)  
XRRR = XRRR + PBAR(I)  
XUUU = XUUU + UBAR(I)  
XSND = XSND + SQRT((GAMMA * PBAR(I) / RBAR(I)))

CONTINUE
C  
SOUND = SOUND + XSND / FLOAT(NXS * NTS)
C  
RHHH = RHHH + XRRR / FLOAT(NXS * NTS)
SOUND = SOUND / FLOAT(NXS)
RHHH = RHHH / FLOAT(NXS)
XPPP = XPPP / FLOAT(NXS)

PRINT*, SOUND, RHHH, XPPP
C  
XUUU = XUUU / FLOAT(NXS)
STEP = DSTEP(IITER)
T = T + STEP
C  
CALL DOMAIN(NEL, NINT, NEL, NX, WI, PHI, T, X, A, - PBAR, UBAR, RBAR, ETA, PSTA, USTA, RSTA)

DO 250 J = 1, 3
EE(J) = EE(J) - STEP * E(J)
GG(J) = GG(J) + STEP * G(J)
250 CONTINUE
C  
CALL BOUND(NX, T, X, A, PBAR, UBAR, RBAR, ETA, PSTA, USTA, RSTA)

DO 350 J = 1, 3
ET(J) = ET(J) + STEP * ETA(J)
350 CONTINUE
C  
IF(ICAL(IITER).EQ.1) THEN
ONE = ET(1) + GG(1)
TWO = ET(2) + GG(2)
THR = ET(3) + GG(3)
CONE = 0.5 / EE(1)
CTWO = EE(2) / EE(1)
CTHR = EE(3) / EE(1)
AA = ONE * CONE
BB = (TWO - 1.5 * ONE * CTWO) * CONE
CC = (THR - 1.5 * TWO * CTWO + (2.25 * CTWO**2 - 2.0 * CTHR) * ONE) * CONE
WRITE(24, 1031) T, AA, BB, CC
C  
CALL EULER(AA, BB, CC, TZERO, TEND)
DO 107 IJ = 1, 3
EE(IJ) = 0.0
GG(IJ) = 0.0
ET(IJ) = 0.0
107 CONTINUE
ENDIF
100 CONTINUE
1031 FORMAT(2X, 4E14.5)
C  
PRINT*, IITER, EE(1), EE(2), EE(3)
PRINT*, EE(1), EE(2), EE(3)
PRINT*, CONE, CTWO, CTHR
PRINT*, AA, BB, CC
TEND = T
C
DDA = ABS(AA - AAA)
DDB = ABS(BB - BBB)
DDC = ABS(CC - CCC)
RMA = AA**2 + BB**2 + CC**2
RMB = DDA**2 + DDB**2 + DDC**2
RMC = SQRT(RMB / RMA)
AAA = AA
BBB = BB
CCC = CC
TZERO = TEND
PRINT*, ITER, DDA, DDB, DDC, RMC
IF (RMC .GT. 1.0E-4 .AND. ITER .LT. 1) GO TO 101
PRINT*, ITER, RMC
RETURN
END

SUBROUTINE EULER(AA, BB, CC, TZERO, TEND)
COMMON/PARAMT/SOUND, GAMMA, ADMI, ADMO
COMMON/PVALUE/XLENG, TLENG, IPERT, PI, EPSI, VISCO

NT = 100
EPSI = 1.0
STEP = (TEND - TZERO) / (NT - 1)
DO 100 I = 1, NT
T = TZERO + (I - 1) * STEP
EZERO = EPSI
EEND = 0.0
1000 CONTINUE
CONE = STEP / 2.0 * (AA + 2.0 * BB * EEND + 3.0 * EEND**2)
CTWO = EEND - STEP / 2.0 * (AA * EEND + BB * EEND + CC * EEND**3)
CTHR = -EZERO - STEP / 2.0 * (AA * EZERO + BB * EZERO**2 + CC * EZERO**3)
DELTE = -(CTWO + CTHR) / (1.0 - CONE)
EEND = EEND + DELTE
IF (ABS(DELTE) .LT. 1.0E-5) GOTO 1000
WRITE(26, 11) T, EEND
FORMAT (2E15.3)
CONTINUE
RETURN
END

SUBROUTINE DOMAIN(NELE, L, NINT, NEL, NX, WI, PHI, T, X, A, P, U, R, E, G, -PSTA, USTA, RSTA)
COMMON/PARAMT/SOUND, GAMMA, ADMI, ADMO
COMMON/PVALUE/XLENG, TLENG, IPERT, PI, EPSI, VISCO

DIMENSION NEL(NELE, L), X(NX), A(NX), P(NX), U(NX), R(NX)
DIMENSION WI(NINT), PHI(L, NINT), DPDS(2), EE(5)
DIMENSION QX(201), QA(201), QU(201), QR(201)
DIMENSION QPP(201), QUR(201), QRR(201)
DIMENSION E(3), G(3), ENT(5), DE(201, 5)
DIMENSION PSTA(NX), RSTA(NX), USTA(NX), S(4)
DIMENSION RV(5), DEX(5), DPVX(5), PR(5), PRV(5), DRX(5)
DIMENSION RVV(5),DVX(5)

C
DPDS(1)=-0.5
DPDS(2)=0.5
DO 50 I=1,3
E(I)=0.0
Q(I)=0.0
50 CONTINUE

C
DO 100 I=1,NX
QX(I)=X(I)
QA(I)=A(I)
QP(I)=PSTA(I)
QPR(I)=P(I)-PSTA(I)
QU(I)=USTA(I)
QUR(I)=U(I)-USTA(I)
QR(I)=RSTA(I)
QRR(I)=R(I)-RSTA(I)
QP=QP(I)
PPR=QPR(I)
UPR=QUR(I)
RR=QR(I)
PRR=QRR(I)
CALL ENTHAL(P,PPR,UPR,RR,PRR,EE)
DO 150 J=1,5
DP(I,J)=EE(J)
150 CONTINUE
100 CONTINUE

C
:00 I=1,NELE
:50 J=1,NINT
XX=0.0
AA=0.0
PP=0.0
PPR=0.0
UU=0.0
UPR=0.0
RR=0.0
PRR=0.0
DO 280 JONE=1,5
ENT(JONE)=0.0
280 CONTINUE

C
:300 K=1,L
NM=NEL(I,K)
CON=PHI(K,J)
XX=XX+QX(NUM)*CON
AA=AA+QA(NUM)*CON
UU=UU+QU(NUM)*CON
UPR=UPR+QUR(NUM)*CON
PP=PP+QP(NUM)*CON
PPR=PPR+QPR(NUM)*CON
RR=RR+QR(NUM)*CON
PRR=PRR+QRR(NUM)*CON
DO 350 KONE=1,5
ENT(KONE)=ENT(KONE)+DE(NUM,KONE)*CON
350 CONTINUE
300 CONTINUE

C
CJCB = -0.5 * X(NEL(I,1)) + 0.5 * X(NEL(I,2))

CONE = CJCB * WI(J) * AA

E(1) = E(1) + (RR * ENT(3) + RPR * ENT(2)) * CONE
E(2) = E(2) + (RR * ENT(4) + RPR * ENT(3)) * CONE
E(3) = E(3) + (RR * ENT(5) + RPR * ENT(4)) * CONE

DO 400 KONE = 1, 5
   DEX(KONE) = 0.0
DO 450 KTWO = 1, L
   NUM = NEL(I, KTWO)
   DEX(KONE) = DEX(KONE) + DE(NUM, KONE) * DPDS(KTWO)
450 CONTINUE
   DEX(KONE) = DEX(KONE) / CJCB
400 CONTINUE

DO 480 K = 1, 3
   DPVX(K) = 0.0
480 CONTINUE

DO 550 KTWO = 1, L
   NUM = NEL(I, KTWO)
   DPVX(1) = DPVX(1) + QP(NUM) * QU(NUM) * DPDS(KTWO)
   DPVX(2) = DPVX(2) + (QP(NUM) * QUR(NUM) + QPR(NUM) * QU(NUM)) * DPDS(KTWO)
   DPVX(3) = DPVX(3) + QPR(NUM) * QUR(NUM) * DPDS(KTWO)
550 CONTINUE

DO 600 KONE = 1, 3
   DPVX(KONE) = DPVX(KONE) / CJCB
600 CONTINUE

DO 650 KONE = 1, 2
   DRX(KONE) = 0.0
650 CONTINUE

DO 700 KONE = 1, 2
   NUM = NEL(I, KONE)
   DRX(1) = DRX(1) + QR(NUM) * DPDS(KONE)
   DRX(2) = DRX(2) + QRR(NUM) * DPDS(KONE)
700 CONTINUE

DO 750 KONE = 1, 2
   DRX(KONE) = DRX(KONE) / CJCB
750 CONTINUE

DO 810 KONE = 1, 2
   DVX(KONE) = 0.0
810 CONTINUE

DO 820 KONE = 1, 2
   NUM = NEL(I, KONE)
   DVX(1) = DVX(1) + QU(NUM) * DPDS(KONE)
   DVX(2) = DVX(2) + QUR(NUM) * DPDS(KONE)
820 CONTINUE

DO 830 KONE = 1, 2
   DVX(KONE) = DVX(KONE) / CJCB
830 CONTINUE

RV(1) = RR * UU
RV(2) = RR * UPR + RPR * UU
RV(3) = RPR * UPR

CA = 1.0 / (GAMMA - 1.0)
CB = -GAMMA / (GAMMA - 1.0)
PC = PPR / PP
RC = RPR / RR
S(1) = CA * PC + CB * RC
S(2) = -1.0 / 2.0 * (CA * PC ** 2 + CB * RC ** 2)
S(3) = 1.0 / 6.0 * (CA * PC ** 3 + CB * RC ** 3)
S(4) = -1.0 / 48.0 * (CA * PC ** 4 + CB * RC ** 4)

CON = 1.0 / RR ** 4
PR(1) = PP * RR ** 3 * CON
PR(2) = (-PP * RR ** 2 * RPR + RR ** 3 * PPR) * CON
PR(3) = (PP * RR * RPR ** 2 - RR ** 2 * PPR * RPR) * CON
PR(4) = (-PP * RPR ** 3 + RR * PPR * RPR ** 2) * CON
PR(5) = -PP * RPR ** 3 * CON

PRV(1) = PR(1) * UU
PRV(2) = PR(1) * UPR + PR(2) * UU
PRV(3) = PR(2) * UPR + PR(3) * UU
PRV(4) = PR(3) * UPR + PR(4) * UU
PRV(5) = PR(4) * UPR + PR(5) * UU

RW(1) = RR * UU ** 2
RW(2) = 2.0 * RR * UU * UPR + UU ** 2 * RPR
RW(3) = RR * UPR ** 2 + 2.0 * UU * UPR * RPR
RW(4) = RPR * UPR ** 2

G(1) = G(1) + (RV(1) * DEX(3) + RV(2) * DEX(2) + RV(3) * DEX(1)) * CONE
G(2) = G(2) + (RV(1) ** 3 + RV(1) * DEX(2) + RV(1) * DEX(3) + RV(3) * DEX(1)) * CONE
G(3) = G(3) + (RV(2) ** 2 + RV(1) * DEX(2) + RV(2) * DEX(4) + RV(3) * DEX(1)) * CONE

G(1) = G(1) + (DPVX(1) * S(2) + DPVX(2) * S(1)) * CONE
G(2) = G(2) + (DPVX(1) ** 3 + DPVX(1) * S(3) + DPVX(2) * S(2) + DPVX(3) * S(1)) * CONE
G(3) = G(3) + (DPVX(1) ** 2 + DPVX(1) * S(4) + DPVX(2) ** 2 + DPVX(3) * S(2)) * CONE

G(1) = G(1) - (DRX(1) * PRV(3) + DRX(2) * PRV(2)) * CONE
G(2) = G(2) - (DRX(1) ** 3 + DRX(2) * PRV(3)) * CONE
G(3) = G(3) - (DRX(1) ** 2 + DRX(2) ** 2 + DRX(4) * PRV(4)) * CONE

G(1) = G(1) - (DVX(1) * RVV(3) + DVX(2) * RVV(2)) * CONE
G(2) = G(2) - (DVX(1) ** 3 + DVX(1) * RVV(3) + DVX(2) ** 2 + DVX(3) * RVV(2)) * CONE
G(3) = G(3) - (DVX(2) ** 2 + DVX(2) * RVV(3) + DVX(4) ** 2) * CONE

CONTINUE
CONTINUE
RETURN
END

SUBROUTINE BOUND(NX, TX, AX, P, U, R, ETA, PSTA, USTA, RSTA)
COMMON/PARAM/SOUND, GAMMA, ADM, ADMO
COMMON/PVALUE/XLENG, TLENG, IPERT, PI, EPSI, VISCO
DIMENSION X(NX), A(NX), P(NX), U(NX), R(NX)
DIMENSION EE(5), ETA(3)
DIMENSION PSTA(NX), RSTA(NX), USTA(NX), S(4)
DIMENSION RV(5), PV(5)

C DO 1201 I = 1, 3
ETA(I) = 0.0
1201 CONTINUE
EONE = 0.0
ETWO = 0.0
ETHR = 0.0

C XX = X(1)
PP = PSTA(I)
UU = USTA(I)
AA = A(1)
RR = RSTA(I)

C PPR = P(1) - PSTA(1)
UPR = U(1) - USTA(1)
RPR = R(1) - RSTA(1)

C CALL ENTHAL(PP, PPR, UU, UPR, RR, RPR, EE)

C RV(1) = RR*UU
RV(2) = RR*UPR + RPR*UU
RV(3) = RPR*UJPR

C CONE = RV(1) * EE(3) + RV(2) * EE(2) + RV(3) * EE(1)
CTWO = RV(1) * EE(4) + RV(2) * EE(3) + RV(3) * EE(2)
CTHR = RV(1) * EE(5) + RV(2) * EE(4) + RV(3) * EE(3)

C EONE = EONE - ONE
ETWO = ETWO - TWO
ETHR = ETHR - THR

C PV(1) = PP*UU
PV(2) = PP*UPR + PPR*UU
PV(3) = PPR*UJPR

C CA = 1.0 / (GAMMA - 1.0)
CB = - GAMMA / (GAMMA - 1.0)
PC = PPR/PP
RC = RPR/RR
S(1) = CA*PC*CB*RC
S(2) = - 1.0/8.0*(CA*PC**2 + CB*RC**2)
S(3) = 1.0/8.0*(CA*PC**3 + CB*RC**3)
S(4) = - 1.0/8.0*(CA*PC**4 + CB*RC**4)

C EONE = EONE - PV(1)*S(2) + PV(2)*S(1) - PPR*UPR
ETWO = ETWO - PV(1)*S(3) + PV(2)*S(2) + PV(3)*S(1)
ETHR = ETHR - (PV(1)*S(4) + PV(2)*S(3) + PV(3)*S(2))

C ETA(1) = EONE * AA
ETA(2) = ETWO * AA
ETA(3) = ETHR * AA

C EONE = 0.0
ETWO = 0.0
ETHR = 0.0

C XX = X(NX)
CALL ENTHAL(PP, PPR, UU, UPR, RR, RPR, EE)

RV(1) = RR*UU
RV(2) = RR*UPR + RPR*UU
RV(3) = RPR*UPR

CONE = RV(1)*EE(3) + RV(2)*EE(2) + RV(3)*EE(1)
CTWO = RV(1)*EE(4) + RV(2)*EE(3) + RV(3)*EE(2)
CTHR = RV(1)*EE(5) + RV(2)*EE(4) + RV(3)*EE(3)

EONE = EONE - CONE
ETWO = ETWO - CTWO
ETHR = ETHR - CTHR

CA = 1.0/(GAMMA - 1.0)
CB = -GAMMA/(GAMMA - 1.0)
PC = PPR/PP
RC = RPR/RR
S(1) = CA*PC + CB*RC
S(2) = -0.5*(CA*PC**2 + CB*RC**2)
S(3) = 1.0/6.0*(CA*PC**3 + CB*RC**3)
S(4) = -1.0/48.0*(CA*PC**4 + CB*RC**4)

EONE = EONE - (PV(1)*S(2) + PV(2)*S(1)) - PPR*UPR
ETWO = ETWO - (PV(1)*S(3) + PV(2)*S(2) + PV(3)*S(1))
ETHR = ETHR - (PV(1)*S(4) + PV(2)*S(3) + PV(3)*S(2))

ETA(1) = ETA(1) + EONE*AA
ETA(2) = ETA(2) + ETWO*AA
ETA(3) = ETA(3) + ETHR*AA

RETURN
END

SUBROUTINE ENTHAL(PP, PPR, UU, UPR, RR, RPR, EE)

COMMON/PARAMT/SOUND, GAMMA, ADM1, ADMO
COMMON/PVALUE/XLENG, TLENG, IPERT, PI, EPSI, VISCO

DIMENSION EE(5)

CON = (GAMMA - 1.0)*RR**4
EE(1) = PP*RR**3/CON + 0.5*UU**2
EE(2) = (-PP*RR**2*PPR + RR**3*PPR)/CON + UU*UPR
EE(3) = (PP*RR**2*PPR**2 - RR**2*PPR*RPP)/CON + 0.5*UPR**2
EE(4) = (-PP*PPR**3 + RR*PPR*RPR**2)/CON
EE(5) = -PPR*RPR**3/CON
SUBROUTINE GAUSS(NINT,SAMP,WEIGHT)

COMMON/PARAMT/SOUND,GAMMA,ADMI,ADMO
COMMON/PVALUE/XLENG,TLENG,IPERT,PI,EPSI,VISCO

DIMENSION SAMP(NINT),WEIGHT(NINT)

N=NINT
M=(N+1)/2
EI=N*(N+1)
DO 1 I=1,M
T=(4*I-1)*PI/(4*N+2)
XO=(1.0-(1.0-1.0)/N/(8.0*N*N))*COS(T)
PKM1=1.0
PK=XO
DO 3 K=2,N
T1=XO*PK
PK1=T1-PKM1-(T1-PKM1)/K+T1
PKM1=PK
3 PK=PK1

DEN=1.-XO*XO
D1=N*(PKM1-XO*PK)
DPN=D1/DEN
D2PN=(2.0*XO*DPN-EI*PK)/DEN
D3PN=(4.0*XO*D2PN+(2.0-EI)*DPN)/DEN
D4PN=(6.0*XO*D3PN+(6.0-EI)*D2PN)/DEN
U=PK/DPN
V=D2PN/DPN
CH=-U*(1.0+0.5*U*(V+U*(V-V*3.0*DPN)))/DEN
P=PK+CH*(DPN+0.5*CH*(D2PN+CH/3.0*(D3PN+0.25*CH*D4PN)))/DEN
DP=DPN+CH*(D2PN+0.5*CH*(D3PN+CH*D4PN/3.0))
:
WEIGHT(I)=2.*(1.0-SAMP(I)*SAMP(I))/(CFX*CFX)
MM=N/2
DO 25 J=1,MM
IF(2*M.EQ.N) GOTO 22
SAMP(M+J)=-SAMP(M-J)
WEIGHT(M+J)=WEIGHT(M-J)
GOTO 25
22 SAMP(M+J)=-SAMP(M+J)
IF(M+M.GT.N) SAMP(M)=0.0
CONTINUE
RETURN
END

SUBROUTINE SHAPE(NINT,XI,PHI)

COMMON/PARAMT/SOUND,GAMMA,ADMI,ADMO

RETURN
END
COMMON/PVALUE/XLENG,TLENG,IPERT,PI,EPSI,_VISCO

DIMENSION XI(NINT),PHI(2,NINT)

DO 100 I=1,NINT
PHI(1,I)=0.5*(1.0-XI(I))
PHI(2,I)=0.5*(1.0+XI(I))
100 CONTINUE

RETURN
END

SUBROUTINE PRIME(PZERO,UZERO,XX,TT,PPRIME,UPRIME)

COMMON/PARAMT/SOUND,GAMMA,ADMI,ADMO
COMMON/PVALUE/XLENG,TLENG,IPERT,PI,EPSI,VISCO

PPRIME=0.0
UPRIME=0.0

DO 100 I=1,12
CON=I*PI
CN=2.0*SQRT(1.0+CON**2)/CON**2
PHASE=-ATAN(1.0/CON)
CKN=CON/XLENG
OMEGA=CON*SOUND/XLENG
XCON=CKN*XX
TCON=OMEGA*TT-PHASE
PPRIME=PPRIME+CN*COO(XCON)*SIN(TCON)
UPRIME=UPRIME+CN*SIN(XCON)*COS(TCON)
100 CONTINUE

PPRIME=EPSI*PZERO*PPRIME
UPRIME=EPSI*SOUND*UPRIME

RETURN
END