UPWIND MACCORMACK EULER SOLVER WITH NON-EQUILIBRIUM CHEMISTRY

Scott E. Sherer James N. Scott
Department of Aeronautical & Astronautical Engineering
The Ohio State University, Columbus, Ohio 43210

ABSTRACT

A computer code, designated UMPIRE, is currently under development to solve the Euler equations in two dimensions with non-equilibrium chemistry. UMPIRE employs an explicit MacCormack algorithm with dissipation introduced via Roe's flux-difference split upwind method. The code also has the capability to employ a point-implicit methodology for flows where stiffness is introduced through the chemical source term. A technique consisting of diagonal sweeps across the computational domain from each corner is presented, which is used to reduce storage and execution requirements. Results depicting one-dimensional shock tube flow for both calorically perfect gas and thermally perfect, dissociating Nitrogen are presented to verify current capabilities of the program. Also, computational results from a chemical reactor vessel with no fluid dynamic effects are presented to check the chemistry capability and to verify the point-implicit strategy.

INTRODUCTION

The role of Computational Fluid Dynamics (CFD) for engineering applications has become widespread in various disciplines within technology. This growth can be attributed to the development of advanced solution algorithms and computer architectures. One area of recent particular interest is the application of CFD codes with non-equilibrium chemistry to high-performance propulsion systems such as ramjets and scramjets.[1,2] The use of CFD in such situations is especially appealing due to the challenges associated with obtaining experimental data. CFD can provide substantial information regarding the physics of flows associated with these systems that may be either impractical or impossible to obtain from ground or flight-based experiments.

The Euler equations which govern inviscid fluid dynamics provide a good initial point for the development of computational methods. They describe significant features in the physics of fluid dynamics, but are easier to work with than the full Navier-Stokes equations. In the past, the standard method for solving the Euler equations numerically was to use central differences to evaluate the spatial derivatives with second-order accuracy. This approach produces good results everywhere except in the vicinity of discontinuities such as shock waves, slip lines, or contact discontinuities. Near these features, central differencing generates spurious oscillations, resulting in corrupted, non-physical solutions. Numerical dissipation is usually introduced through artificial viscosity, but this method requires repeated "knob-turning" to determine satisfactory amounts of dissipation necessary for different applications.

Recently, upwind schemes have become popular in dealing with flows possessing discontinuities. Upwind schemes exhibit the ability to reduce spurious oscillations by incorporating physical characteristics of the flow into the discretization process. They are naturally dissipative, and thus the "knob-turning" required by central differencing schemes is unnecessary. One of the standard upwind-type schemes is the flux-difference split method of Roe. Formulated first for a calorically-perfect gas,[3] it has been modified for both equilibrium and non-equilibrium chemically-reacting gases.[4,5,6] Roe's flux-difference split method utilizes exact solutions to a series of local, approximate Riemann problems at computational cell interfaces. Roe's method in its basic form results in a spatially first-order accurate scheme, since the state of the fluid is assumed constant across the entire cell. It can be extended to second-order accuracy through a array of techniques that use either variable extrapolation[7] or flux extrapolation.[8,9] In general, however, by raising the accuracy to second-order, some oscillations will be produced.[10] This difficultly can usually be overcome through the application of flux limiters in the algorithm to smooth oscillations without additional smearing of any discontinuities.[11]

As part of an effort to acquire the capability to model high-speed, reacting flows, a computer code was developed to solve the Euler equations including non-equilibrium chemistry. The code, designated UMPIRE (for Upwind MacCormack Point-Implicit) uses a scheme similar to the one presented by White, et.al.[12] This method is based on the explicit, predictor-corrector MacCormack scheme with upwind dissipation terms introduced through Roe's flux-difference split method and extended to second-order accuracy using the Szema-Chakravarthy method[8] and a minmod flux limiter. The point-implicit method presented by Bussing and Murman[13] for dealing with a stiff chemical source term is also utilized to speed convergence for steady-state applications. The motivation behind UMPIRE is to create a code that can accurately and efficiently model inviscid flows with non-equilibrium chemistry for a wide variety of different conditions, and to have the code to serve as a basis for future development of more advanced codes to solve the Parabolized or Full Navier-Stokes equations. Also, the educational benefits and practical experience obtained in developing such a code from the ground up cannot be overlooked.

GOVERNING EQUATIONS

Fluid Dynamics Model

The governing fluid dynamic equations currently utilized in the UMPIRE code are the two-dimensional, time-dependent, compressible Euler equations in chemical non-equilibrium. They may be written in compact vector form for Cartesian coordinates as

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + H = 0 . \qquad (1)$$

The dependent vector Q, the inviscid flux vectors E and F, and the chemical source term H are given by

$$Q = \begin{bmatrix} \rho f_1 \\ \rho f_2 \\ \vdots \\ \rho f_N \\ \rho u \\ \rho v \\ \varepsilon_o \end{bmatrix} \qquad E = \begin{bmatrix} \rho f_1 u \\ \rho f_2 u \\ \vdots \\ \rho f_N u \\ \rho u^2 + p \\ \rho u v \\ u(\varepsilon_o + p) \end{bmatrix}$$
$$H = \begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \vdots \\ \dot{\omega}_N \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad F = \begin{bmatrix} \rho f_1 v \\ \rho f_2 v \\ \vdots \\ \rho f_N v \\ \rho v u \\ \rho v^2 + p \\ v(\varepsilon_o + p) \end{bmatrix} , \qquad (2)$$

where ρ is the density, f_i is the mass fraction of specie i, $\dot{\omega}_i$ is the production rate of specie i, p is the pressure, u and v are the x- and y-components of the velocity, respectively, and ε_o is the total energy per unit volume. In order to close the set of equations, additional relationships are required. For a mixture consisting of independent, thermally perfect gases in thermodynamic equilibrium, the equations for pressure, specie enthalpy, and total energy may be defined as

$$p = \rho \Re T \sum_{i=1}^{N} \frac{f_i}{M_i} \qquad (3)$$

$$h_i = (\Delta h_f)_i^{T_s} + \int_{T_s}^{T} c_{p,i}(\tau)\, d\tau \qquad (4)$$

$$e_o = \rho \sum_{i=1}^{N} h_i f_i - p + \frac{\rho}{2}(u^2+v^2)\,, \qquad (5)$$

where $\mathfrak{R}$ is the universal gas constant, $(\Delta h_f)_i^{T_s}$ is the formation enthalpy of specie i at reference temperature T_s, T is the temperature, and $c_{p,i}$ is the specific heat at constant pressure of specie i. The final equation necessary to complete the system is given by

$$\sum_{i=1}^{N} f_i = 1\,. \qquad (6)$$

Equations 1 and 2 are best suited for solving on an orthogonal grid with constant spacing. Since the Cartesian x-y coordinate system does not provide such a grid for most physical applications, Equations 1 and 2 were transformed into a general computational ξ-η coordinate system possessing the above qualities. Carrying out this transformation and manipulating the resulting equations into conservation-law form yields[14]

$$\frac{\partial \bar{Q}}{\partial t} + \frac{\partial \bar{E}}{\partial \xi} + \frac{\partial \bar{F}}{\partial \eta} + \bar{H} = 0\,, \qquad (7)$$

where the transformed vectors are given by

$$\bar{Q} = \frac{1}{J}\begin{bmatrix} \rho f_1 \\ \rho f_2 \\ \vdots \\ \rho f_N \\ \rho u \\ \rho v \\ e_o \end{bmatrix} \qquad \bar{E} = \frac{1}{J}\begin{bmatrix} \rho f_1 U \\ \rho f_2 U \\ \vdots \\ \rho f_N U \\ \rho u U + \xi_x p \\ \rho v V + \xi_y p \\ U(e_o + p) \end{bmatrix}$$
$$\bar{H} = \frac{1}{J}\begin{bmatrix} \dot{\omega}_1 \\ \dot{\omega}_2 \\ \vdots \\ \dot{\omega}_N \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \bar{F} = \frac{1}{J}\begin{bmatrix} \rho f_1 V \\ \rho f_2 V \\ \vdots \\ \rho f_N V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ V(e_o + p) \end{bmatrix} \qquad (8)$$

In Equation 8, J represents the Jacobian of the transformation, ξ_x, ξ_y, η_x, and η_y are the transformation metrics, and U and V are defined as

$$U = \xi_x u + \xi_y v \qquad V = \eta_x u + \eta_y v\,. \qquad (9)$$

Equations 7 and 8 are the equations that are actually differenced and solved by the UMPIRE code. For the remainder of this paper, the bar over the transformed vectors will be dropped and it will be assumed that they are being used unless otherwise noted.

Thermodynamic Model

The thermodynamic model utilized by UMPIRE consists of a fourth-order polynomial for the specific heat at constant pressure,

$$c_{p,i} = A_i + B_i T + C_i T^2 + D_i T^3 + E_i T^4\,. \qquad (10)$$

The coefficients A_i, B_i, C_i, D_i, and E_i are found for each specie of interest using a least-squares curve-fitting routine and thermodynamic data up to 6000 K as given in the JANAF tables[15].

A thermodynamic quantity required for the calculation of the chemical equilibrium constant is the Gibbs free energy per mole of specie i at one atmosphere pressure. From its definition, the Gibbs free energy may be found from

$$g_i^{p=1} = A_i(T - T\ln T) - \frac{B_i}{2}T^2 - \frac{C_i}{6}T^3 - \frac{D_i}{12}T^4 - \frac{E_i}{20}T^5 - F_i + G_i T\,, \qquad (11)$$

where the coefficients F_i and G_i are functions of T_s, $(\Delta h_f)_i^{T_s}$, A_i, B_i, C_i, D_i, E_i, and the specie entropy at T_s. All of this information may be easily obtained for the species from the JANAF tables.

Chemistry Model

To obtain the specie production rates required in the calculation of chemical source vector, it was assumed that the reaction method employed consists of J chemical reactions involving N species, and that the jth reaction may be written in the generic form

$$\sum_{i=1}^{N} \nu'_{i,j}\, X_i \rightarrow \sum_{i=1}^{N} \nu''_{i,j}\, X_i \,. \qquad (12)$$

In the above equation, $\nu_{i,j}$ is the stoichiometric coefficient for the ith species in the jth chemical reaction, where i = 1, 2,..., N, and j = 1, 2,..., J. The source term for specie i is found by summing its production rate over all reactions. If the forward and backward rate expressions are defined as

$$\sigma^f_{i,j} = k_{f,j}\prod_{m=1}^{N}\left(\frac{\rho f_m}{M_m}\right)^{\nu'_{i,j}}$$
$$\sigma^b_{i,j} = k_{b,j}\prod_{m=1}^{N}\left(\frac{\rho f_m}{M_m}\right)^{\nu''_{i,j}} \qquad (13)$$

then the source term becomes

$$\dot{\omega}_i = M_i \sum_{j=1}^{J} (\nu''_{i,j}-\nu'_{i,j})\,[\sigma^f_{i,j}-\sigma^b_{i,j}] \qquad (14)$$

The forward rate constant of reaction j, $k_{f,j}$, is calculated using the Arrhenius model equation

$$k_{f,j} = A_j T^{N_j} \exp\left(-\frac{E_j}{\Re T}\right) \qquad (15)$$

where A_j, N_j, and E_j are empirically determined constants for each reaction. The backwards reaction rate and equilibrium constants are

$$k_{b,j} = \frac{k_{f,j}}{k_{eq,j}} \qquad (16)$$

$$k_{eq,j} = \left(\frac{101,325}{\Re T}\right)^{\Delta n} \exp\left(\frac{\Delta G_j^{p-1}}{\Re T}\right) \qquad (17)$$

where

$$\Delta G_j^{p-1} = \sum_{i=1}^{N} (\nu''_{i,j} - \nu'_{i,j})\, g_i^{p-1} \qquad (18)$$

and

$$\Delta n = \sum_{i=1}^{N} (\nu''_{i,j} - \nu'_{i,j}) \,. \qquad (19)$$

NUMERICAL METHODS

Roe's Flux-Difference Split Method

The upwinding present in the UMPIRE code is introduced through Roe's flux-difference split method. For a one-dimensional situation in Cartesian coordinates, the approximate Riemann problem

$$\frac{\partial Q}{\partial t} + [A'] \frac{\partial Q}{\partial x} = 0 \qquad (20)$$

with the initial conditions given by

$$Q(x,0) = Q_L \quad for \quad x \le 0$$
$$Q(x,0) = Q_R \quad for \quad x > 0 \qquad (21)$$

is solved at each cell interface to yield a new distribution of the dependant variables across a computational cell. The matrix [A'] is a special form of the flux Jacobian matrix, $\partial E/\partial Q$, that is assumed to be a constant function of Q_R and Q_L over a computational cell and must satisfy certain properties as shown by Roe[3]. For the non-reacting, calorically perfect gas case, these properties are satisfied by the "Roe-averaging" of the flow variables, given for a general flow variable q as

$$\hat{q}_{i+\frac{1}{2}} = \frac{q_{i+1}\sqrt{\rho_{i+1}} + q_i\sqrt{\rho_i}}{\sqrt{\rho_{i+1}} + \sqrt{\rho_i}} \qquad (22)$$

For the non-equilibrium reacting gas case, this averaging process becomes more involved due to the added complexities of the system of equations.

Grossman and Cinnella[4] presented Roe-averaged expressions for a reacting gas that satisfy the conditions prescribed by Roe exactly, and these are the relationships used by UMPIRE.

In Roe's flux-difference split method, the first-order numerical flux at the cell interfaces in the ξ-direction may be defined by[16]

$$\begin{aligned} E^*_{i+\frac{1}{2},j} &= E_{i,j} + de^-_{i+\frac{1}{2},j} \\ &= E_{i+1,j} - de^+_{i+\frac{1}{2},j} \end{aligned} \qquad (23)$$

where

$$de^\pm_{i+\frac{1}{2},j} = (\hat{S}_\xi \hat{\Lambda}^\pm_\xi \hat{S}^{-1}_\xi)_{i+\frac{1}{2},j}(Q_{i+1,j} - Q_{i,j}) \qquad (24)$$

In Equation 23, S_ξ is the matrix whose columns consist of the right eigenvectors of ∂E/∂Q, S_ξ^{-1} is the inverse of S_ξ, $\Lambda^\pm_\xi$ is a matrix consisting either of the positive or negative eigenvalues of ∂E/∂Q on its diagonal, and the ^ represents evaluation at the Roe-averaged state. The first-order numerical flux for the interface at i-½,j is defined similarly as

$$\begin{aligned} E^*_{i-\frac{1}{2},j} &= E_{i-1,j} + de^-_{i-\frac{1}{2},j} \\ &= E_{i,j} - de^+_{i-\frac{1}{2},j} \end{aligned} \qquad (25)$$

where

$$de^\pm_{i-\frac{1}{2},j} = (\hat{S}_\xi \hat{\Lambda}^\pm_\xi \hat{S}^{-1}_\xi)_{i-\frac{1}{2},j}(Q_{i,j} - Q_{i-1,j}) \qquad (26)$$

For two-dimensional calculations, one-dimensional flux-difference splitting may be applied in each direction independently, and the results then combined. This method is fairly straight-forward, but does have some limitations when waves oblique to the computational grid are present.[6,17] The first-order numerical fluxes in the η-direction at the interfaces i,j±½ are given by equations similar to Equations 23 through 26.

Upwind MacCormack Method

The numerical algorithm used by UMPIRE is the MacCormack predictor-corrector scheme with flux-difference split, upwind dissipation terms as presented by White, et.al.[12] The derivation of the upwind MacCormack scheme begins with the spatially and temporally second-order accurate form of the modified Euler method for the Euler equations

$$\begin{aligned} \Delta Q^{\tau+1}_{i,j} = -\Delta t [& (E^\tau_{i+\frac{1}{2},j} - E^\tau_{i-\frac{1}{2},j}) \\ & + (F^\tau_{i,j+\frac{1}{2}} - F^\tau_{i,j-\frac{1}{2}}) + H^\tau_{i,j} \end{aligned} \qquad (27)$$

where τ+1 = p and τ = n for the predictor step, and τ+1 = c and τ = p for the corrector step. The change in Q for the entire time step from n to n+1 is then given by

$$\begin{aligned} \Delta Q^{n+1}_{i,j} &= \frac{1}{2}(\Delta Q^p_{i,j} + \Delta Q^c_{i,j}) \\ Q^{n+1}_{i,j} &= Q^n_{i,j} + \Delta Q^{n+1}_{i,j} \end{aligned} \qquad (28)$$

The numerical fluxes at the cell interfaces (± ½) are calculated by Roe's flux-difference split method as represented in Equations 23 through 26. Since there are two expressions that yield the numerical flux at each interface, the expression $\mathbf{E}_{i+½,j}$ - $\mathbf{E}_{i-½,j}$ may be written in four different ways. The two ways that are of interest here are

$$\begin{aligned} E_{i+\frac{1}{2},j} - E_{i-\frac{1}{2},j} &= E_{i+1,j} - de^+_{i+\frac{1}{2},j} - \\ & \quad (E_{i,j} - de^+_{i-\frac{1}{2},j}) \qquad (29) \\ &= \Delta_\xi E_{i,j} - (de^+_{i+\frac{1}{2},j} - de^+_{i-\frac{1}{2},j}) \end{aligned}$$

and

$$E_{i+\frac{1}{2},j} - E_{i-\frac{1}{2},j} = E_{i,j} + de^{-}_{i+\frac{1}{2},j} - (E_{i-1,j} + de^{-}_{i-\frac{1}{2},j}) \quad (30)$$

$$= \nabla_{\xi} E_{i,j} + (de^{-}_{i+\frac{1}{2},j} - de^{-}_{i-\frac{1}{2},j})$$

Similar equations may also be written for $\mathbf{F}_{i,j+½}$ - $\mathbf{F}_{i,j-½}$. Inserting these expressions into Equation 27 yields the MacCormack predictor-corrector scheme with upwind terms to provide dissipation. For example, if forward differences are inserted into the predictor step while backwards differences are inserted into the corrector step, the resulting equation is

$$\Delta Q^{p}_{i,j} = -\Delta t[\Delta_{\xi} E^{n}_{i,j} - (de^{+\,n}_{i+\frac{1}{2},j} - de^{+\,n}_{i-\frac{1}{2},j}) + \Delta_{\eta} F_{i,j} - (df^{+\,n}_{i,j+\frac{1}{2}} - df^{+\,n}_{1,j-\frac{1}{2}}) + H^{n}_{i,j}]$$

$$\Delta Q^{c}_{i,j} = -\Delta t[\nabla_{\xi} E_{i,j} + (de^{-\,p}_{i+\frac{1}{2},j} - de^{-\,p}_{i-\frac{1}{2},j}) + \nabla_{\eta} F_{i,j} + (df^{-\,p}_{i,j+\frac{1}{2}} - df^{-\,p}_{i,j-\frac{1}{2}}) + H^{p}_{i,j}] \quad (31)$$

In the UMPIRE code, the forward and backward differences for the predictor and corrector steps are cycled according to Table 1[14] to prevent build-up of directional bias. This method is stable for a CFL number ≤ 1, as is the standard MacCormack scheme when applied to the Euler equations.

Second Order Terms

Thus far, the upwind MacCormack scheme as presented is only first-order accurate in space because the state of the fluid is assumed to be constant across a computational cell. There are several methods available for extending the spatial accuracy of the upwind MacCormack scheme, but UMPIRE currently uses the Szema-Chakravarthy method[8], based on the work of Lawerance, et.al.[19]

First, in the ξ-direction, intermediate variables α are defined as

$$\alpha_{1,i+\frac{1}{2},j} = (S_{\xi}^{-1})_{i+\frac{1}{2},j}(Q_{i,j} - Q_{i-1,j})$$

$$\alpha_{2,i+\frac{1}{2},j} = (S_{\xi}^{-1})_{i+\frac{1}{2},j}(Q_{i+1,j} - Q_{i,j}) \quad .$$

$$\alpha_{3,i+\frac{1}{2},j} = (S_{\xi}^{-1})_{i+\frac{1}{2},j}(Q_{i+2,j} - Q_{i+1,j}) \quad (32)$$

Then, these vectors are limited relative to each other using a minmod flux limiter function in order to reduce spurious oscillations. The resulting equations are given by

$$\mathbf{A}_{1,i+\frac{1}{2},j} = mm[(\alpha_1)_{i+\frac{1}{2},j}, \beta(\alpha_2)_{i-\frac{1}{2},j}]$$

$$\mathbf{A}_{2,i+\frac{1}{2},j} = mm[(\alpha_2)_{i+\frac{1}{2},j}, \beta(\alpha_1)_{i-\frac{1}{2},j}]$$

$$\mathbf{A}_{3,i+\frac{1}{2},j} = mm[(\alpha_2)_{i+\frac{1}{2},j}, \beta(\alpha_3)_{i-\frac{1}{2},j}]$$

$$\mathbf{A}_{4,i+\frac{1}{2},j} = mm[(\alpha_3)_{i+\frac{1}{2},j}, \beta(\alpha_2)_{i-\frac{1}{2},j}] \quad (33)$$

where the minmod function of two values x and y is defined as

$$mm = sgn(x)\max[0, \min\{|x|, y\,sgn(x)\}] \quad (34)$$

and

$$\beta = \frac{(3 - \phi)}{(1 - \phi)} \, . \quad (35)$$

ϕ is known as the accuracy parameter, and was assumed equal to -1 throughout this study, corresponding to a fully-upwind method.[19]

The intermediate vectors are then multiplied by the eigenvalues and eigenvectors to obtain the limited upwind fluxes

$$de^{+}_{1,i+\frac{1}{2},j} = S_{i+\frac{1}{2},j}\Lambda^{+}_{i+\frac{1}{2},j}\mathrm{A}_{1,i+\frac{1}{2},j}$$

$$de^{+}_{2,i+\frac{1}{2},j} = S_{i+\frac{1}{2},j}\Lambda^{+}_{i+\frac{1}{2},j}\mathrm{A}_{2,i+\frac{1}{2},j} \quad . \quad (36)$$

$$de^{-}_{1,i+\frac{1}{2},j} = S_{i+\frac{1}{2},j}\Lambda^{-}_{i+\frac{1}{2},j}\mathrm{A}_{3,i+\frac{1}{2},j}$$

$$de^{-}_{2,i+\frac{1}{2},j} = S_{i+\frac{1}{2},j}\Lambda^{-}_{i+\frac{1}{2},j}\mathrm{A}_{4,i+\frac{1}{2},j}$$

Finally, the total second-order contribution to the numerical flux is defined by

$$dee_{i-\frac{1}{2},j} = \frac{1-\phi}{4}de^{+}_{1,i+\frac{1}{2},j} + \frac{1+\phi}{4}de^{+}_{2,i+} - \frac{1+\phi}{4}de^{-}_{1,i+\frac{1}{2},j} - \frac{1-\phi}{4}de^{-}_{2,i+\frac{1}{2}} \quad (37)$$

Similar second-order terms may be found for the i-½,j interface as well as for the interfaces in the η-direction.

The second-order upwind MacCormack scheme is obtained by adding Equation 37 and its counterparts at i-½,j and i,j±½ evaluated at time level n to their corresponding first-order numerical fluxes defined by Equations 23 through 26 in the corrector step only.[12,16] For example, to raise Equation 31 to second-order spatial accuracy, the predictor step would remain unchanged while the corrector step would become

$$\begin{aligned}\Delta Q^{c}_{i,j} = -\Delta t[&\nabla_{\xi}E_{i,j} + (de^{-\,p}_{i+\frac{1}{2},j} - de^{-\,p}_{i-\frac{1}{2},j}) \\ &+ (dee^{n}_{i+\frac{1}{2},j} - dee^{n}_{i-\frac{1}{2},j}) \\ &+ \nabla_{\eta}F_{i,j} + (df^{-\,p}_{i,j+\frac{1}{2}} - df^{-\,p}_{i,j-\frac{1}{2}}) \\ &+ (dff^{n}_{i,j+\frac{1}{2}} - dff^{n}_{1,j-\frac{1}{2}}) \\ &+ H^{p}_{i,j}]\end{aligned} \quad (38)$$

Point-Implicit Treatment of Source Vector

When dealing with numerical solutions of chemically reacting systems, the problem of stiffness often arises. Stiffness in a chemically reacting system is a result of the widely varying characteristic time scales between the chemical and the fluid dynamic processes being modelled. If left untreated, stiff problems require prohibitively long solution times, due to the fact that the solution must be advanced at its smallest time step to remain stable. A popular method for treating the chemical source term is to evaluate it implicitly, which introduces the chemical source Jacobian as a premultiplying matrix to the left-hand side of the predictor and corrector equations. Continuing with the forward predictor, backwards corrector example in both coordinate directions, Equation 38 becomes

$$\left[I + \Delta t\left(\frac{\partial H}{\partial Q}\right)^{n}\right]\Delta Q^{p}_{i,j} = RHS\ Eq.\ 38 \quad (39)$$

$$\left[I + \Delta t\left(\frac{\partial H}{\partial Q}\right)^{p}\right]\Delta Q^{c}_{i,j} = RHS\ Eq.\ 38$$

Thus, at each grid point, an N+3 system of equations must be solved. Bussing and Murman[13] have shown that by evaluating the source term implicitly, the disparity between the characteristic times is removed, thus allowing each process to advance towards the steady-state at its own rate. Point-implicit capability has been included in the UMPIRE code for use in steady-state problems which contain stiff chemical source terms.

Coding of Upwind MacCormack Method

The coding of the upwind MacCormack method was given much consideration while constructing UMPIRE. A technique where the computational domain is swept diagonally in varying directions was felt to be the most efficient application of the upwind MacCormack method. By examining Equation 44 and its counterparts, it can be seen that $\Delta Q_{i,j}{}^{p}$ and $\Delta Q_{i,j}{}^{c}$ each rely on information from two of the four surrounding grid points and all four of the surrounding cell interfaces. The two grid points providing information depend upon whether forward or backward differences are currently being employed for each of the coordinate directions. For example, in the predictor step of Equation 38, which will be referred to as an FF step since the finite differences in both directions are forward, the grid points i+1,j and i,j+1 are used in the calculation. By inspecting Figure 1, a model 6 x 6 grid in computational space, it can be seen that, for an FF step, a diagonal line of grid points (Line A) relies on the information found along the next diagonal line of grid points to the upper right (Line B) and the two surrounding diagonal lines of cell interfaces (Lines F and G). Similarly, for a BB step, Line A relies on the information from Line C and Lines F and G, for a FB step, Line D and Lines H and I, and for a BF step, Line E and Lines H and I. Thus, for any combination of differences, only four lines of data need be stored, two for the fluxes at the grid points, and two for the numerical fluxes at the cell interfaces. This reduces storage requirements when compared with calculating and storing data for each grid point and interface in the domain, and it reduces execution time when compared to calculating data at each node and interface as needed and discarding, especially when considering the computational effort necessary to calculate the interface fluxes. There are some additional coding requirements as well as an increase in code complexity with this method, but it was felt that these were insignificant when compared to the disadvantages associated with either of the other methods. Also, this method should improve vectorization ability when the code is ported to vector machines such as the Cray Y-MP.

For each predictor or corrector step, the code first determines the category of the current step; FF, FB, or BF, BB (see Table 1). With this information, an initial starting grid point is defined; lower left for FF, upper right for BB, upper left for FB, and lower right for BF. The code then sweeps towards the corner of the computational domain opposite to its starting point, stepping one diagonal at a time. The same lines of code are used for all of the categories, with the only difference being the value of a few integer variables to make the distinction between sweep directions. Only the data for the current grid point and interface diagonal and for the previous grid point and interface diagonal are stored in the computer's memory. If the step is a predictor step, then the second-order terms from Equation 37 are calculated and stored for the entire domain, while if the step is a corrector step, the previously calculated second-order terms are added to the interface flux as indicated Equation 38. Care must be taken when calculating and adding the second-order flux terms using this method to insure that all of the signs are correct when performing both forward and backward sweeps.

RESULTS

Three test cases that were solved using the UMPIRE code are presented here; a shock tube containing calorically perfect gas, a well-stirred chemical reactor with dissociating Nitrogen, and a shock tube containing dissociating Nitrogen.

Shock Tube (Calorically Perfect Gas)

Shown in Figure 2 are the density profiles obtained by UMPIRE for a shock tube with calorically perfect gas. This example was run until the time was equal to 5 x 10^{-4} seconds, with γ = 1.4, a CFL number of 0.5, a pressure ratio of 2:1, and a temperature ratio of 1:1. There were 101 grid points taken along the length of the tube, and, even though the problem is essentially one-dimensional, 5 grid points were taken along the width of the tube to test the code's basic structure in two dimensions. Figure 2 was generated by running UMPIRE in four different modes; no upwinding (MacCormack scheme with no added dissipation), first-order upwinding, second-order upwinding with no flux limiters, and second-order upwinding with minmod flux limiters. A modest

2:1 pressure ratio was selected here to contrast the results from the standard MacCormack scheme with no dissipation to those from the MacCormack scheme with upwind dissipation, since at higher pressure ratios, the former produces such wild oscillations that negative pressures and temperatures develop. The effect of the upwind dissipation on the oscillations present in the standard MacCormack scheme can be easily seen. Also, the second-order scheme appears to resolve the shock and contact discontinuity better that the first-order scheme. In this case, the difference between the limited and unlimited second-order scheme are minimal because of the low pressure ratio, which causes only minor oscillations to be produced by the unlimited scheme. Other pressure and temperature ratios were examined for the case of a shock tube with calorically perfect gas, and they yielded results that similarly matched those obtained exactly from one-dimensional gas-dynamics[20].

Chemical Reactor (Dissociating Nitrogen)

The second case presented here is the chemical reactor containing dissociating Nitrogen. In this model reactor, it is assumed that there are no fluid dynamic effects present, and that the only change in the system is caused by the presence of chemical reactions. This case was investigated to check the ability of the code to handle a simple reaction mechanism decoupled from the fluid dynamics, and to examine some of the features of the point-implicit method.

The reactor was assumed to be a square box, and a simple 5 x 5 grid was used. An initial temperature and pressure were selected, and values for the mass fractions of N_2 and N were chosen so as to not correspond with the equilibrium conditions. For the case shown here, these values were taken to be 4000 K, 10 MPa, .9, and .1, respectively. UMPIRE was then used to march the solution forward in time to a steady-state, equilibrium condition, using both the point-implicit and non-point-implicit methods and a variety of CFL numbers. The reaction used to drive this system and its Arrhenius coefficients are shown in Table 2[21]. The results are shown in Figure 3, which are graphs of the length of time and the number of steps required to reduce the norm of the residual vector to 10^{-8}. When the point implicit method is not used, it can be seen that the length of time required to reach the equilibrium point is about the same no matter what CFL number is used. This represents the physical time required for the dissociation reaction of Nitrogen to equilibrate. However, the results for the point-implicit method show no such constant time is found for different CFL numbers. The time in the point-implicit method is no longer a physical time, but a "psuedotime"[13] used to advance towards the steady-state. The point-implicit method was found to converge to an equilibrium value for CFL numbers as high as 0.9, while for the non-point-implicit method, CFL numbers under 0.01 were required for stability. This example demonstrates the potential of the point-implicit method for solving equations where the chemical and fluid dynamic time scales vary widely. The point-implicit method has allowed the reaction to be separated from its physical time scale and instead be associated with a psuedotime scale, which will be of the same order as the fluid time scale. It should also be noted that, for the initial conditions presented earlier, the code converged to practically the same equilibrium point no matter what CFL number or whether the point-implicit or the non-point-implicit was used. For the given initial conditions, this equilibrium point is given by T = 6067.8 K, p = 1,426,811 Pa, f_{N2} = 0.965355, and f_N = 0.034644.

Shock Tube (Dissociating Nitrogen)

Once some confidence was established for the chemistry capabilities of UMPIRE for the simple Nitrogen dissociation reaction, it was applied to the shock tube case. In this case, a pressure ratio of 8:1 was used (10 MPa : 1.25 MPa), with a temperature ratio of 1:1 (T = 4000

K). Again, the grid was 101 x 5 and a CFL of 0.5 was used. The code was marched up to 1 x 10^{-4} seconds, and the second-order, limited, upwind MacCormack method was used to solve the equations. The profile of the mass fractions for the two species of Nitrogen are shown in Figure 4. At 4000 K, molecular Nitrogen is just starting to dissociate, and thus we do not see much atomic Nitrogen present. As the shock wave propagates to the right into the lower pressure gas, the mass fraction of the molecular Nitrogen drops while the mass fraction of atomic Nitrogen rises. The non-equilibrium effects are clearly seen in the overshoots and undershoots present in the mass fractions immediately following the shock. Downstream of the shock, the chemical composition is given the chance to equilibrate, and returns to its equilibrium composition.

CONCLUSIONS

In this study, a computer code named UMPIRE has been presented for solving the Euler equations with non-equilibrium chemistry. The code employees a MacCormack predictor-corrector scheme with upwind dissipation terms provided by Roe's flux-difference split method to smooth out spurious numerical oscillations. A technique involving diagonal sweeps across the computational domain that is especially suited for use with this method is used to reduce storage and speed execution time. Three test cases have been investigated to verify the current capabilities of UMPIRE; a shock tube with calorically perfect gas, a chemical reactor with dissociating Nitrogen, and a shock tube with dissociating Nitrogen. All three cases compare well with exact results or results from other sources. Current work involving UMPIRE includes examining true two-dimensional cases, such as inlets and blunt bodies, and incorporating more advanced reaction mechanisms, such as reacting air, or hydrogen-air mixtures.

REFERENCES

[1] White, M.E., Drummond, J.P., and Kumar, A., "Evolution and Application of CFD Techniques for Scramjet Engine Analysis", *Journal of Propulsion and Power*, Vol. 3, No. 5, 1987, pp. 423-439.

[2] Barber, T.J., and Cox, Jr., G.B., "Hypersonic Vehicle Propulsion: A CFD Application Case Study", AIAA Paper 88-0475, January 1988.

[3] Roe, P.L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes", *Journal of Computational Physics*, Vol. 43, 1981, pp. 357-372.

[4] Grossman, B., and Cinnella, P., "Flux-Split Algorithms for Flows with Non-Equilibrium Chemistry and Vibrational Relaxation", *Journal of Computational Physics*, Vol. 88, 1990, pp. 131-168.

[5] Vinokur, M., and Montagne, J.-L., "Generalized Flux-Vector Splitting and Roe Averaging for an Equilibrium Real Gas", *Journal of Computational Physics*, Vol. 89, 1990, pp. 276-300.

[6] Shuen, J.-S., Liou, M.-S., and van Leer, B., "Inviscid Flux-Splitting Algorithms for Real Gases with Non-equilibrium Chemistry", *Journal of Computational Physics*, Vol. 90, 1990, pp. 371-395.

[7] van Leer, B., "Towards the Ultimate Conservation Difference Scheme V, A Second-Order Sequel to Godunov's Method", *Journal of Computational Physics*, Vol. 32, 1979, pp.101-136.

[8] Chakravarthy, S.R., and Szema, K.Y., "An Euler Solver for Three-Dimensional

Supersonic Flows with Subsonic Pockets", AIAA Paper 85-1703, July 1985.

[9] Chakravarthy, S.R., and Osher, S., "A New Class of High Accuracy TVD Schemes for Hyperbolic Conservation Laws", AIAA Paper 85-0363, January 1985.

[10] Engquist, B. and Osher, S., "One-sided Difference Approximations for Nonlinear Conservation Laws", *Mathematics of Computation*, Vol. 36, pp. 321-352.

[11] Scott, J.N., and Niu, Y.-Y., "Comparison of Limiters in Flux-Split Algorithms for Euler Equations", AIAA Paper 93-0068, January 1993.

[12] White, J.A., Korte, J.J., and Gaffney, Jr., R.L., "Flux-Difference Split Parabolized Navier-Stokes Algorithm for Nonequilibrium Chemically Reacting Flows", AIAA Paper 93-0534, January 1993.

[13] Bussing, T.R.A., and Murman, E.M., "Finite-Volume Method for the Calculation of Compressible Chemically Reacting Flows", *AIAA Journal*, Vol. 26, No. 9, 1988, pp. 1070-1078.

[14] Anderson, D.A., Tannehill, J.C., Pletcher, R.H., *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corporation, New York, 1984.

[15] Chase, M.W., et.al., "JANAF Thermochemical Tables, Third Edition, Part II, Cr-Zr", *Journal of Physical and Chemical Reference Data*, Vol. 14, Supplement 1, American Chemical Society and American Institute of Physics, 1985.

[16] Hirsch, C., *Numerical Computation of Internal and External Flows, Volume 2: Computational Methods for Inviscid and Viscous Flows*, John Wiley and Sons, New York, 1988.

[17] Leck, C.L., and Tannehill, J.C., "A New Rotated Upwind Difference Scheme for the Euler Equations", AIAA Paper 93-0066, January 1993.

[18] Lawerance, S.L., Tannehill, J.C., and Chausee, D.S., "Upwind Algorithm for the Parabolized Navier-Stokes Equations", *AIAA Journal*, Vol. 27, No. 9, 1989, pp. 1175-1183.

[19] Yee, H.C., and Shinn, J.L., "Semi-Implicit and Fully Implicit Shock Capturing Methods for Nonequilibrium Flows", *AIAA Journal*, Vol. 27, No. 3, 1989, pp. 299-307.

[20] Anderson, J.A., *Modern Compressible Flow With Historical Perspective*, McGraw-Hill Book Company, New York, 1982.

[21] Vincenti, W.G., and Kruger, Jr., C.H., *Introduction to Physical Gas Dynamics*, Robert E. Krieger Publishing Company, Malabar, Florida, 1965.

Cycle	Predictor		Corrector	
Direction	ξ	η	ξ	η
1	F	F	B	B
2	F	B	B	F
3	B	F	F	B
4	B	B	F	F

Table 1 - Cycling of MacCormack Scheme
F -> forward difference, B-> backward difference

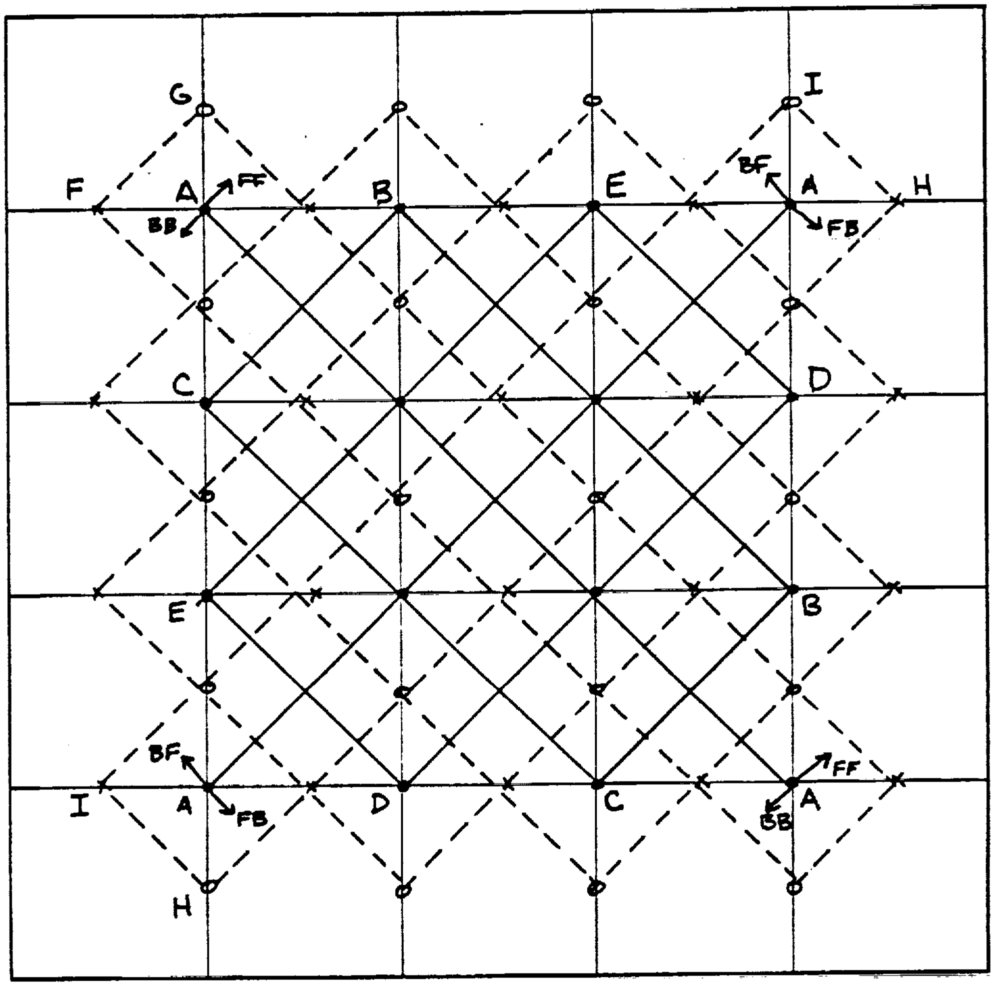

Figure 1 - Diagonal Sweeping of Computational Domain

$N_2 + N_2 \leftrightarrow 2N + N_2$	$A = 4.8x10^{14}$
	$N = -0.5$
	$E = 9.3951929x10^{8}$
$N_2 + N \leftrightarrow 2N + N$	$A = 4.1x10^{19}$
	$N = -1.5$
	$E = 9.3951929x10^{8}$

Table 2 - Reaction Mechanism for Dissociating N_2
k_f in units of $m^3/(kmol \cdot s)$

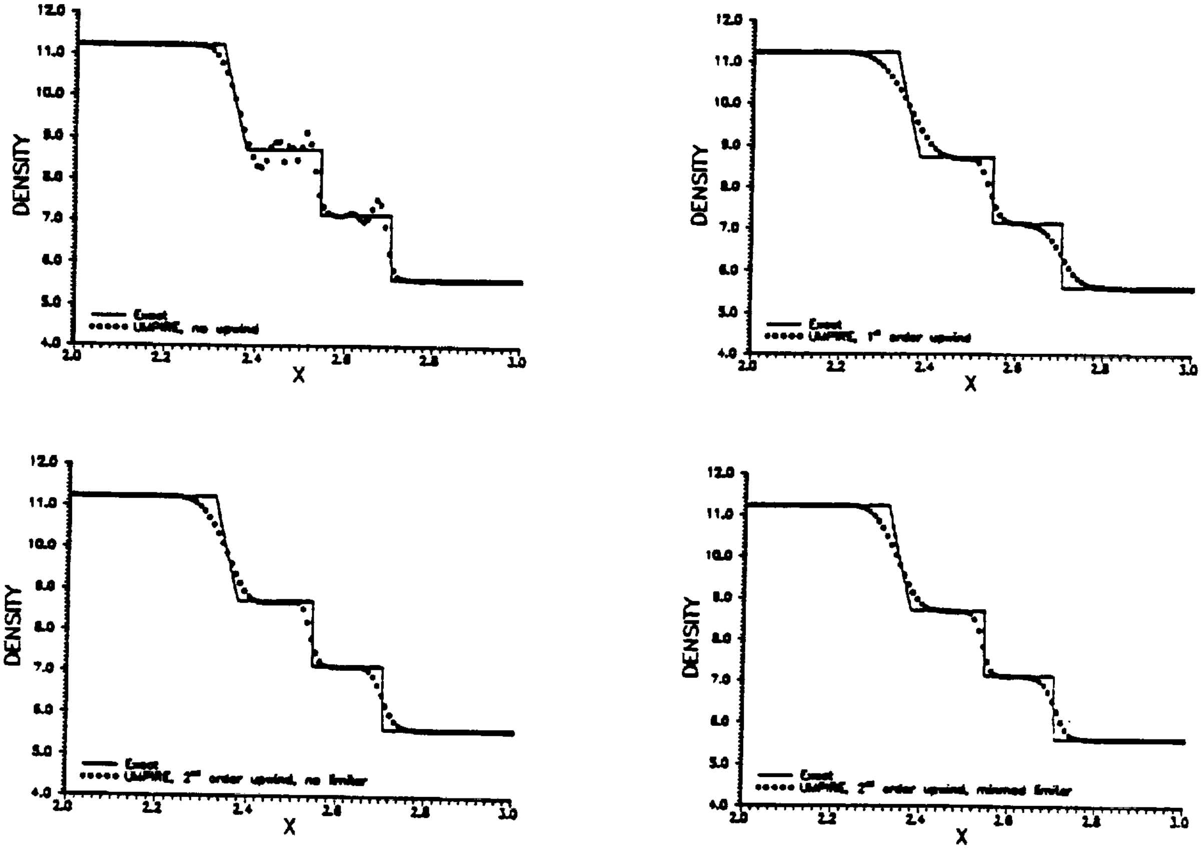

Figure 2 - Shock Tube, Calorically Perfect Gas

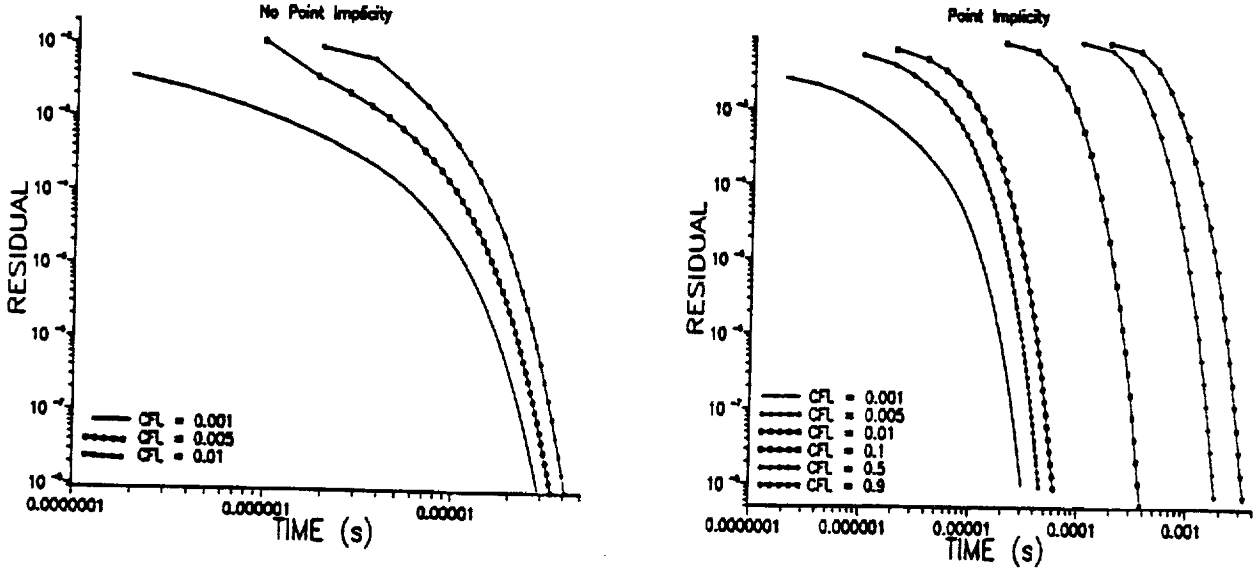

Figure 3 - Chemical Reactor, Dissociating Nitrogen

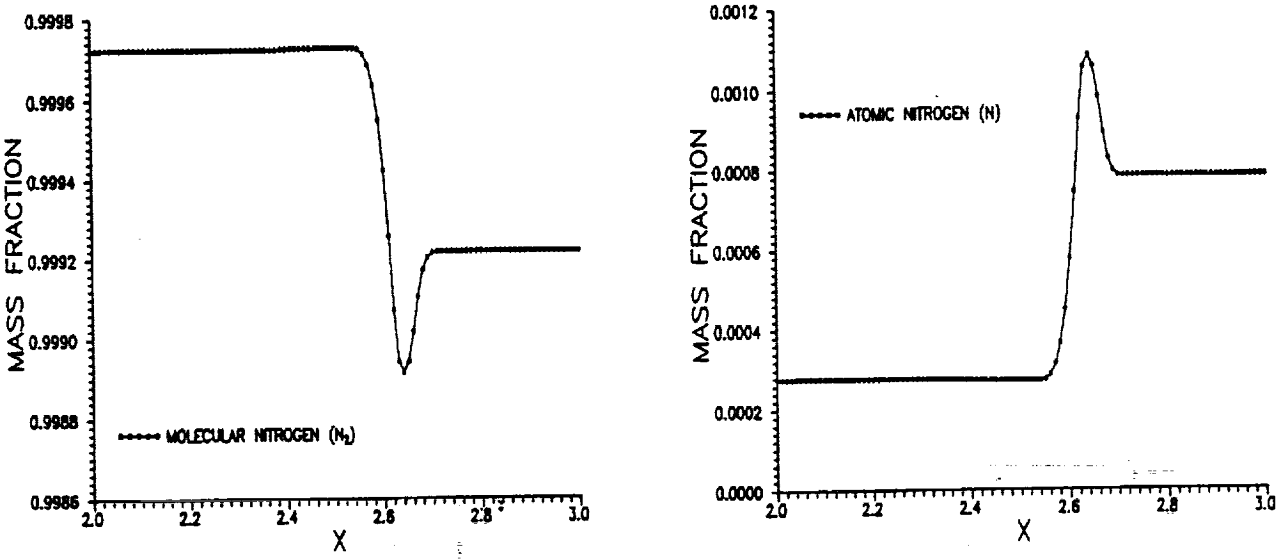

Figure 4 - Shock Tube, Dissociating Nitrogen

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE November 1993	3. REPORT TYPE AND DATES COVERED Conference Publication

4. TITLE AND SUBTITLE Fifth Annual Thermal and Fluids Analysis Workshop	5. FUNDING NUMBERS
6. AUTHOR(S)	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135–3191	8. PERFORMING ORGANIZATION REPORT NUMBER E–8094
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, D.C. 20546–0001	10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA CP–10122

11. SUPPLEMENTARY NOTES

Responsible person, Doug Darling, (216) 433–8273.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 01, 61, 64, and 34	12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

The Fifth Annual Thermal and Fluids Analysis Workshop was held at the Ohio Aerospace Institute, Brook Park, Ohio, cosponsored by NASA Lewis Research Center and the Ohio Aerospace Institute, August 16–20, 1993. The workshop consisted of classes, vendor demonstrations, and paper sessions. The classes and vendor demonstrations provided participants with the information on widely used tools for thermal and fluids analysis. The paper sessions provided a forum for the exchange of information and ideas among thermal and fluids analysts. Paper topics included advances and uses of established thermal and fluids computer codes (such as SINDA and TRASYS) as well as unique modeling techniques and applications.

14. SUBJECT TERMS Thermal simulation; Fluid mechanics; Computer programs	15. NUMBER OF PAGES 566
	16. PRICE CODE A24

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT

PRECEDING PAGE BLANK NOT FILMED