NASA-CR-191713

Development of Generalized Pressure Velocity Coupling Scheme for the Analysis of Compressible and Incompressible Combusting Flows

Grant Number NAG8-128
Final Report
Period of Performance: May 19, 1989 - May 18, 1992

by C. P. Chen and S. T. Wu

Contents

Abstract

The objective of this investigation has been to develop an algorithm (or algorithms) for the improvement of the accuracy and efficiency of the computer fluid dynamics (CFD) models to study the fundamental physics of combustion chamber flows, which are necessary ultimately for the design of propulsion systems such as SSME and STME. During this three year study (May 19, 1989 - Mary 18, 1992), a unique algorithm was developed for all speed flows. This newly developed algorithm basically consists of two pressure-based algorithms (i.e. PISOC and MFICE). This PISOC is a non-iterative scheme and the FICE is an iterative scheme where PISOC has the characteristic advantages on low and high speed flows and the modified FICE has shown its efficiency and accuracy to compute the flows in the transonic region. A new algorithm is born from a combination of these two algorithms.

This newly developed algorithm has general application in both time-accurate and steady state flows, and also was tested extensively for various flow conditions, such as turbulent flows, chemically reacting flows, and multiphase flows. A list of publications and doctoral dissertations resulting from this effort is provided in Appendix 1.

1 CURRENT NUMERICAL METHODS FOR COMPRESSIBLE AND INCOMPRESSIBLE FLOWS

The objective of this study is to improve the accuracy and efficiency of numerical simulation of combustion chamber flows for better understanding of the physics of these complicated flow conditions. To achieve this objective, a three year study has been initiated to develop an algorithm (or algorithms) for incorporation of complex physical phenomena such as turbulence, compressibility, chemistry and multiphase effects into existing CFD codes. This report summarizes the achievements of developing a unique algorithm during the last three years. Two pressure-based algorithms have been identified and extensively tested. In the following, we will describe the methodology and testing results.

The unsteady compressible Navier-Stokes (N-S) equations are a mixed set of hyperbolic-parabolic partial differential equations (PDE's) while the unsteady incompressible N-S equations are a mixed set of elliptic-parabolic PDE's. Traditionally, different numerical techniques have been used to solve the N-S equations in the compressible and incompressible flow regimes. If the unsteady terms are dropped from the equation set for compressible flows, the resulting equations become a mixed set of hyperbolic-elliptic equations which are difficult to solve because of the differences in numerical techniques required for hyperbolic and elliptic type equations. Consequently, nearly all successful solutions of the compressible N-S equations have employed the unsteady form of the governing equations. This strategy will also be adopted in this study for the developed algorithm to obtain a more general application in both time-accurate transient and steady state applications. With this approach, the steady state solution is obtained by marching the solution in time until convergence is achieved.

1.1 Density-Based Methods

To date, numerical methods utilizing the primitive variables: density, pressure and velocities (in contrast to stream function-vorticity formulation) for solving the unsteady N-S equation, can be largely classified into two schemes: density-velocity scheme and pressure-velocity scheme. Most density-velocity methods have their origins in external aerodynamics problems. In these problems, it is natural to choose density as a primary dependent variable for the continuity equation, whereas the pressure is calculated from the equation of state.

For most density-based methods the equations governing continuity, momentum, energy and other scalars are solved in a coupled manner. For inviscid flow calculations, explicit methods [1] are often used for simplicity and storage considerations. However, the explicit methods suffer from the limitation of small time steps due to stability requirements, and their application to viscous flow problems is costly. Thus implicit methods are employed for most of the compressible viscous flow calculations. The most widely used implicit schemes for viscous flows are the methods of Beam and Warming [2], Briley and McDonald [3] and MacCormack [4]. Since the governing equations are solved in a coupled manner, the characteristics of the equation system are easily obtained. The fluxes at cell faces can be calculated by the so called flux-vector splitting techniques or by solving the Rieman problem. The advantage of this approach is that the jump (discontinuity) conditions are satisfied, therefore good results for discontinuous problems, such as flows with shocks, are expected. One approach, using the flux-vector splitting techniques, is the Steger and Warming [5] method, in which the flux vector F(U) is split into two parts, F^+ and F^- such that all the eigenvalues of $\frac{\partial F^+}{\partial U_j}$ are non-negative, and the eigenvalues $\frac{\partial F^-}{\partial U_j}$ are non-positive. The spatial derivatives of F^+ are backward differenced and that of F^- are forward differenced. The method involving solutions of the Rieman problem was originally proposed by Godunov [6]. Since an exact solution of the Rieman problem is expensive and unnecessary, several flux-difference split schemes have been developed, for example by Colella [7], Dukowicz [8], and Roe [9], for which the exact solution of the Rieman problem is replaced by an approximate solution. Since the use of central differencing for convective terms at high Reynolds numbers resulted in spatial oscillations in the early study, these oscillation are suppressed now by adding artificial damping, usually fourth-order damping terms are used by density-based methods. In the last few years, significant progress has been made in the high resolution numerical schemes based on the Total Variation Diminishing (TVD) principle, introduced by Harten [10] to develop oscillation-free schemes. The TVD schemes are very robust for transient problems and shocks capturing [11]. Further details of these schemes can be found in the recent book by Hirsh [12].

Although the density-based coupling schemes have been successfully applied to compressible flows, the methods have a disadvantage when in the limit the incompressible flow is approached, and the linkage between pressure and density weakens in the low Mach number range [13]. In fact, Mach number zero constitutes a singularity in the compressible form of the equations. Any tiny disturbances of density are enough to destroy the stability of solution in the low Mach number regime. Numerical experiments [14] confirmed the low Mach number inefficiency and instability of the density based methods. To avoid such a problem, a "pseudo" (artificial) compressibility term can be added into the continuity equation [15, 16]. This new parameter, called "pseudo", should speed its use. As a result, the convergence rate highly depends on the choice of this value. This method is not efficient for unsteady simulations since many inner iterations are required to obtain divergence free solutions at each time step. Many existing methods, developed specifically for incompressible flows, surmount this problem by treating the pressure as a primary dependent variable. These pressure-velocity coupling schemes are equally valid for compressible flows.

1.2 Pressure-Based Methods

The earliest development of primitive variable schemes based on pressure-velocity coupling was the semi-implicit transient Marker-and-Cell method (MAC) [17] and Simplified MAC [18] by the Los Alamos group. Since there is no explicit governing equation for the pressure field, these methods basically derive a working pressure equation through joint manipulation of the momentum and continuity equations. Existing pressure-velocity coupling methods can be divided into two categories, namely, semi-implicit and full implicit schemes. Because of their reliance on explicit differencing, semi-implicit schemes have a disadvantage in time-dependent computations, since the time-step size, necessary to maintain stability of such methods may drastically impair the efficiency of the algorithm especially when applied to the calculation of steady-state flows. Implicit methods on the other hand, do not suffer from time step restrictions.

The most popular method using pressure-velocity coupling schemes for solving incompressible flows is the SIMPLE algorithm of Patankar and Spalding [19] and its variants: SIMPLER by Patankar [20], SIMPLEC by Van Doormaal and Raithby [21], SIMPLEX by Van Doormaal and Raithby [22] and SIMPLEST by Sha [23]. The advantages gained by the implicit differencing of the SIMPLE method, which is based on a pressure correction procedure, are offset by the use of an iteration, which makes time dependent calculations rather expensive as iteration is required at each time step. The SIMPLE algorithms can be extended to handle compressible flow calculations as shown by Van Doormaal *et al.* [24]. This method accounts for additional variations in density through an equation of state based pressure -density coupled correction scheme. Although applicable to a wide variety of flows, there are certain flow situations in which this method is inappropriate and fails to yield acceptable results. Recognised and addressed by Gosman and Watkins [25], these flows are the ones in which the temperature is strongly coupled with the pressure and velocity, such as chemically reacting flows.

Another method for handling the pressure-velocity coupling of implicitly differenced fluid flow equations is the non-iterative PISO algorithm of Issa [26]. This method splits the process of the solution into a series of predictor and corrector steps that, at each step, a simplified set of equations in terms of a single unknown variable is obtained. The PISO algorithm has exhibited a very efficient and robust nature when applied to a variety of flows as shown by Benodekar [27] and Issa [28].

The operator splitting technique can be described briefly as follows: We first describe the N-S in a fully implicit way such that the N-S equation

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_i}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\tau_{ij} + S_i \qquad (1)$$

becomes

$$\frac{1}{\Delta t}\left\{(\rho u_i)^{n+1} - (\rho u_i)^n\right\} = H\left(u_i^{n+1}\right) - \Delta_i p^{n+1} + S_i \qquad (2)$$

where Δ is the discrete ∇, u_i^{n+1}, p_i^{n+1} denote i^{th} grid point values at $(n+1)^{th}$ time level. H( ) is a linearized convection diffusion operator. H( ) in principle is a time-dependent operator in which H(u^{n+1}) represents a fully implicit formulation while H(u^n) represents a fully explicit formulation. The starting point of the operator-splitting technique in this research is the one originally proposed by Issa [26], namely, the Pressure Implicit by Splitting of Operator (PISO) algorithm. In this algorithm, each marching time step is further divided into a sequence of predictor-corrector steps. In the first momentum predictor step the operator is split in such a way,

$$H(u_i^*) = A_o u_i^* + H\prime(u_i^*) \qquad (3)$$

in which A_o is the diagonal part of the original matrix operator H. u_i^* is the first predictor velocity value (unknown). The implicit momentum predictor step then can be rewritten as:

$$\left(\frac{1}{\delta t} - \frac{A_o}{\rho^n}\right)\rho^n u_i^* = H\prime(u_i^*) - \delta_i p^n + S_i + \frac{\rho^n u_i^n}{\delta t} \qquad (4)$$

Note that the pressure value used in equation (4) is the "old" value (time step n). Thus, the solution of (4) will not satisfy continuity at time step $n+1$.

Corrector steps are devised to derive a pressure governing equation, driving the velocity field to satisfy continuity. This is done by splitting the operator for first correcting the velocity field such that

$$H(u_i^{**}) = A_o u_i^{**} + H'(u_i^{*}) \tag{5}$$

thus the discretized momentum equation becomes

$$\left(\frac{1}{\delta t} - \frac{A_o}{\rho^n}\right)\rho^* u_i^{**} = H'(u_i^*) - \Delta_i p^* + S_i + \frac{\rho^n u_i^n}{\delta t} \tag{6}$$

Note that equation (6) now has two unknowns: u_i^{**} and p^*.

By subtracting equation (4) from (6), we obtain a momentum increment equation

$$\rho^* u^{**} - \rho^n u_i^* = -\left(\frac{1}{\delta t} - \frac{A_o}{\rho_n}\right)^{-1} \Delta_i (p^* - p^n) \tag{7}$$

By taking divergence of this equation and involving continuity constraints, a pressure increment equation is obtained

$$\left[\Delta_i\left\{\left(\frac{1}{\delta t} - \frac{A_o}{\rho^n}\right)^{-1}\Delta_i\right\} - \frac{\Phi(P^n T^n)}{\delta t}\right](p^* - p^n) = \Delta_i(\rho^n u_i^*) \tag{8}$$

where $\Phi()$ is the equation of state linking pressure and density. This one -step corrector scheme resembles the SIMPLE algorithm, if iterations between first predictor and first corrector were executed. To reduce this iteration procedure, a second corrector, based on further operator-splitting for second corrector velocity field, is used:

$$H(u_i^{***}) = \frac{A_o \rho^{**}}{\rho^*} u_i^{***} + H'(u_i^{**}) \tag{9}$$

In this operation, the momentum and continuity then are simultaneously satisfied. Summaries of the PISO procedure are listed in Table 1.

Here, we compare the PISO algorithm to another established pressure based method. The newly developed FICE scheme of Hu and Wu [29] based on the earlier ICE scheme [30] is chosen. However, a modification of the

FICE scheme in parallel with the operator-splitting idea is developed as part of the research effort of this task in section 2. This newly developed algorithm is called MFICE. In the FICE scheme, the discretized momentum equation is written as

$$\left(\frac{1}{\delta t} + A_o\right)u_i^{k+1} = H'(u_i^n) - \Delta_i p^n + S_i + \frac{\rho^n u_i^n}{\delta t} \tag{10}$$

The main difference between this equation and equation (4) is the split operator $H'()$. FICE used an explicit scheme for the predictor step while the PISO used an implicit scheme. Another difference lies in the way the pressure equation is set up. Instead of deriving a pressure-correction equation, the FICE scheme directly takes divergence of the momentum equation (in contrast to the momentum increment equation) and invoking the continuity equation. This scheme is essentially a one-step predictor-corrector scheme and requires iteration.

1.3 Relationship Between Pressure and Density Based Methods

Recently, the PISO algorithm has been rearranged in a vector formulation for direct comparison with other density based algorithms. The findings [31] indicate that the PISO algorithm alters the sonic speed so that the equations stay well conditioned in the limit of low Mach numbers. In particular, the PISO algorithms is very closely related to the preconditioning algorithm developed by the Penn State group [32]. The philosophy of the preconditioning technique is to cause the density-based method to appear pressure-based at low speeds but to remain density-based at high speeds. Originally, preconditioning methods were used as a means of circumventing the disparity in the eigen values at low Mach numbers. This technique alters the time derivatives of the equation of motion with the acoustic speed scaled down to the level of the fluid velocity, such that the local CFL number (which controls the marching time-steps) is approximately of the same order for viscous and inviscid terms. The extensions of the predicting technique have recently been carried out for viscous chemical reacting flows involving chemical source terms. However, how preconditioning may be applied to improve convergence and robustness in the calculation of multiphase and turbulent flow (involving higher-order turbulent models) remains to be seen.

2 NEW NUMERICAL FORMULATIONS FOR ALL-SPEED REGIMES BASED ON PRESSURE METHODS

2.1 Formulation of PISOC and MFICE

The basic ideas of the operator-splitting technique have been described in 1.2. The algorithm used was the PISO algorithm first proposed by Issa [26]. The major difference between the PISO and the previously used pressure-based schemes, such as the SIMPLE-family schemes, is the use of momentum per unit volume as the resulting variable of the momentum equations, rather than velocity (which is momentum per unit mass). The advantages are twofold. First, the time-dependent equations give directly the change in a property per unit volume, whereas the SIMPLE pressure correction algorithm must divide the time change by density in order to calculate the new velocity. Second, the change in momentum can be reinterpreted as a change in mass flux. This gives a linkage between pressure and mass flux; the mass conservation equation then only contains density in the time-dependent term. The pressure correction equations thus obtained become the momentum correction equation:

$$\rho^* u_i^{**} - \rho^n u_i^* = -\left(\frac{1}{\delta t} - \frac{A_o}{\rho_n}\right)^{-1} [\Delta_i(p^* - p^n)] \qquad (11)$$

thus the pressure increment equation is obtained by taking divergence of (11) and involving the continuity equation

$$\left[\Delta_i\left\{\left(\frac{1}{\delta t} - \frac{A_o}{\rho^n}\right)^{-1} \Delta_i\right\} - \frac{\Phi(p_i^n T_i^n)}{\delta t}\right](p^* - p^n) = \Delta_i(\rho^n u_i^*) \qquad (12)$$

where $\Phi(p^n, T^n) = p^*/\rho^*$

Note that the pressure-momentum linkage equation as described in (12) remains essentially elliptic at all flow speeds and cannot mimic the hyperbolic behavior of the system of equations, when flows are transonic and supersonic. To remedy this defect, we propose to split the operator, consistent with the compressibility effect of the density correction, such that the first momentum corrector step becomes:

$$\left(\frac{1}{\delta t} - \frac{A_o}{\rho^*}\right)\rho^* u_i^{**} = H'(u_i^*) - \Delta_i p^* + S_i + \frac{\rho^n u_i^n}{\delta t} \tag{13}$$

the operator splitting procedure used is

$$H(u_i^{**}) = H'(u_i^*) + A_o \frac{\rho^*}{\rho^*} u_i^{**} \tag{14}$$

This new algorithm is named PISOC (Consistent) due to its consistent operator splitting procedure. Compared to the original PISO algorithm where

$$H(u_i^{**}) = H'(u_i^*) + A_o \frac{\rho^*}{\rho^n} u_i^{**} \tag{15}$$

the new momentum correction equation becomes

$$\rho^* u_i^{**} - \rho^n u_i^* = -\left(\frac{1}{\delta t} - \frac{A_o}{\rho^n}\right)^{-1}\left[\Delta_i(p^* - p^n) + A_o \frac{\rho'}{\rho^n} u_i^*\right] \tag{16}$$

where $\rho\prime = \rho^* - \rho^n$, and the pressure correction equation is

$$\begin{aligned} &\left[\Delta_i\left(\frac{1}{\delta t} - \frac{A_o}{\rho n}\right)^{-1}\Delta_i - \frac{\Phi}{\rho_n}\right](p^* - p^n) \\ -\Delta_i&\left[1 - \left(\frac{1}{\delta t} - \frac{A_o}{\rho_n}\right)^{-1}\right]\frac{1}{\delta t} u_i^* \Phi (p^* - p^n) = \Delta_i(\rho^n u_i^*) \end{aligned} \tag{17}$$

The capability to solve compressible flow with shocks by the PISOC algorithm is achieved by the convection incremental pressure term, the second term in the left hand side of equation (17). This term properly takes into account the hyperbolic nature of supersonic flows.

The PISOC algorithm is a pressure correction method (PCM) in which the pressure correction equations are solved. On the other hand, the MFICE scheme is a pressure substitute method (PSM) in which the Poisson pressure equations are directly solved in place of the continuity equation. In the MFICE algorithm, the equation set is discretized as

$$\frac{1}{\delta t}(\rho^{n+1} - \rho^{n}) + \Delta_i(\rho u_i)^{n+1} = 0 \tag{18}$$

$$\frac{(\rho u_i)^{n+1} - 1(\rho u_i)^{n}}{\delta t} - A_o(\rho u_i)^{n+1} = H\prime(u_i^{n+1}) - \Delta_i p^{n+1} + S_i^{n+1} \tag{19}$$

To solve the fully implicit equations (18), (19), the MFICE scheme employs the iterative solution procedure as shown below:

i. $k = 1, (\rho, p, \rho u_i)^k = (\rho, \rho u_i)^n$

ii. substituting (19) into (18) (also via equation of state p = ρRT), we obtain the pressure equation in the form

$$\frac{1}{\delta t}\frac{p^{k+1}}{RT^{k}} - \Delta_i\left[\left(\frac{1}{\delta t} - A_o\right)^{-1}\Delta_i p^{k+1}\right] = \frac{1}{\delta t}\rho^{n} - \Delta_i\left\{\left(\frac{1}{\delta t} - A_o\right)^{-1}\right.$$
$$\left.\left[\frac{1}{\delta t}(\rho u_i)^{n} + H\prime(u_i^{k}) + S_i^{k}\right]\right. \tag{20}$$

which yields p^{k+1} and thus solves for $(\rho u_i)^{k+1}$ by

$$\left(\frac{1}{\delta t} - A_o\right)(\rho u_i)^{k+1} = \frac{1}{\delta t}(\rho u_i)^{n} + H'(u_i^{k}) + S_i^{k} \tag{21}$$

iii. if $\mid p^{k+1} = p^{k} \mid < \varepsilon$ then $(\rho, p, \rho u_i)^{n+1} = (\rho, p, \rho u_i)^{k=1}$, goes to the next time step or else $(\rho, p, \rho u_i)^{k} = (\rho, p, \rho u_i)^{k+1}, k = k + 1$, goes to ii)

This MFICE algorithm has the following characteristics:

- The MFICE, as compared to its original FICE, holds now the operator-splitting (i.e, write H = A_o + H$\prime$) scheme to enhance the convergence of the iteration procedure.

- We may write H in the form H_n, or H_k, where the former is a linearized operator (depending on the n^{th} time level results), and the latter is in a non-linear form, which is updated during the $k \rightarrow k+1$ iteration. In MFICE H_k is used.

- The technique of introducing convection terms to the pressure equation, to mimic the hyperbolic property when supersonic phenomena are involved, does not have much significance to MFICE, since the term $\partial\rho/\partial t$ in the continuity equation (1) is retained, which recovers the hyperbolic property automatically. Therefore, MFICE does not use this technique.

2.2 Boundary Conditions

The implementation of boundary conditions is one of the most complex problems in computational fluid dynamics. The difficulties are due to various possibilities of combining different boundary conditions in a general CFD problem. In addition a few special results are known about the mathematical representation of boundary conditions to ensure existence or uniqueness of the solution. For these reasons a discussion of general boundary conditions of CFD problems is not undertaken here. In this section only the most frequently encountered boundary conditions in fluid flow problems and the treatment, necessary to incorporate them in the discretised equations, are described.

2.2.1 Inlet Boundary Condition

At the inlet boundary, the values of all dependent variables are normally known. These are usually obtained by reference to experimental data, analysis or estimation. These values can be substituted into the discretization equations for the boundary control volumes and thus nothing special needs to be done. Since the inlet boundary mass fluxes are generally known in incompressible flows (here only the velocity component normal to the boundary is of concern), all intermediate values of u_i at the boundary, namely, u_i^{**} and u_i^{***}, are set to the given boundary value. This is equivalent to prescribing zero gradient boundary conditions for the pressure correction equation. It is readily implemented by setting the coefficient of equation for the boundary node to zero. The same treatment also applies for outlet, symmetry, and wall boundaries. The pressure at the boundary is obtained by linear extrapolation from inner points.

In contrast to incompressible flows, the pressure or stagnation pressure is often fixed at inlet boundary for compressible flows. The number of variables that can be specified at the inflow boundary depends on the number of incoming flow characteristics. For subsonic inflow, this requires the specification of three variables. Whereas, if the inflow is supersonic, all variables must be fixed. For the case of subsonic inflow there is a considerable choice as to which variables should be specified. For example, both the velocity components and the pressure may be specified, or both the velocity component and temperature may be specified. In internal flows, it may be convenient to specify the stagnation temperature, stagnation pressure, and transverse component of velocity or the inlet flow angles. Other variables that are required at the inflow boundary are obtained by extrapolation from the interior. If the pressure at the boundary is specified, then $p^* - p^n$ and $p^{**} - p^*$ are set to zero, which serves as the boundary condition for the pressure-increment equations. The normal velocities at the boundary are updated by the continuity equation, and the tangential velocities are obtained by linear extrapolation. If the stagnation pressure at boundary is specified, the velocities are obtained by the same procedure described above. From this velocity, the given stagnation pressure and the stagnation temperature, static pressure and temperature can be calculated. So, it is basically the same as the specification of a pressure boundary condition.

2.2.2 Outlet Boundary Condition

For incompressible flows the value of the dependent variables are generally unknown. The outlet boundary should be placed sufficiently far downstream from the region of interest. As a result, any inaccuracy in estimating the outlet conditions will not propagate far upstream. In this study two conditions are used to get velocities at the outlet. The first one is that the velocity profile at the outlet is similar to the velocity at the first internal point from the outlet. The second one is that the overall continuity must be satisfied that is, the sum of outlet mass fluxes equals the sum of inlet mass fluxes.

$$u^{o} = \alpha u^{I} \text{ and } \mathrm{v}^{o} = \alpha \mathrm{v}^{\mathrm{I}} \tag{22}$$

$$\dot{m}_{out} = \sum_{o}(u^{o}\delta y_{o} - v^{o}\delta x_{o}) = \sum_{I,O}\alpha(u^{I}\delta yo - v^{I}\delta x_{o}) = \dot{m}_{in} \tag{23}$$

$$\alpha = \frac{\dot{m}_{in}}{\sum_{I,O} (u^{I}\delta y_{o} - v^{I}\delta x_{o})} \tag{24}$$

where α is a scale constant. O denotes values at the outlet. I denotes the first interior point from the outlet. When the new velocity is obtained at internal points, equation (24) is employed to get α, then the velocities at the outlet are updated by using equation (23). The other scalar variables are obtained by setting all coefficients corresponding to the outlet boundary nodes to zero. It is, therefore, appropriate to evaluate outlet boundary values by extrapolation from upstream.

If flow is compressible, the number of variables that can be specified is similar to the inlet boundary. It is equal to the number of incoming characteristics. For a subsonic outflow, this requires specification of one boundary condition. The variable that is usually fixed is the static pressure. It is called the back pressure. The values of other variables are obtained by extrapolation from the interior domain. For supersonic outflow, all the variables at the outlet are determined by upwind information, no boundary conditions should be specified, and all values of variables at the outlet are obtained by extrapolation.

2.2.3 Symmetry Boundary Condition

The symmetry boundary condition implies two contents, no flow crosses the symmetry line (or plane), and diffusive flux at the direction normal to the symmetry line is zero. The first one can be satisfied by setting the contravariant velocity (velocity normal to the symmetry line) to zero. The second one can be met by setting all the coefficients corresponding to the symmetry boundary points to zero. The values of variables at the symmetry boundary are obtained by $\frac{\partial\phi}{\partial n} = 0$,

$$\left(x_{n}^{2} + y_{n}^{2}\right)\phi_{\eta} - \left(x_{\xi}x_{n} + y_{\xi}y_{\eta}\right)\phi_{\eta} = 0 \tag{25}$$

2.2.4 Wall Boundary Condition

For incompressible flow the no-slip condition is applied. The velocity of the fluid at the wall equals the velocity of the wall. These conditions are easy to impose. Because the velocity at the wall is known, the implementation of wall boundary conditions is the same as inlet boundary conditions. There are two types of boundary conditions: the temperature and other scalars, fixed wall value and fixed wall flux.

For compressible flow there are two possibilities of wall boundary condition, no-slip wall and slip wall. The no-slip wall is for viscous compressible flows. The implementation is the same as for the incompressible wall boundary. The slip wall is for inviscid compressible flows. This condition is imposed by setting the normal component of contravariant velocity to zero. The velocity at the boundary is obtained by a projection of interior points along the wall.

3 EXTENSION OF THE NEW METHOD

3.1 Turbulent Flows

Implementation of a two-equation model ($k - \varepsilon$), to include turbulence effects into this combined algorithm, has been developed for subsonic flows. In the following, we discuss the new method and the implementation procedure.

The equations of fluid flows are written in general form:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0 \tag{26}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\tau_{ij} + S_i \tag{27}$$

In the equation above ρ is the density, u_i the velocity components, p is the pressure, S_i are body forces and τ_{ij} are the components of a deviatoric stress tensor:

$$\tau_{ij} = \mu\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] \tag{28}$$

As mentioned in the previous section, the PISOC algorithm is a pressure correction method (PCM) in which the pressure correction equations are solved. On the other hand, the MFICE scheme is a pressure substitute method (PSM) in which the Poisson pressure equation is directly solved in place of the continuity equation. The key issue is to include the momentum source calculations and to ensure continuity (Eq. (26)) for a prescribed number of corrector steps on a pressure equation. Here we seek a split operator in a time domain such that the splitting error of the finite-difference form of Eq. (2) is less than the truncation error of the temporal finite differencing.

The splitting procedure described above is extended for solving other scalar transport equations. In simulating turbulent flows, the well-known $k-\varepsilon$ model of turbulence requires solving transport equations for the turbulence kinetic energy, k, and its rate of dissipation ε. These equations are strongly coupled, especially through the source terms. The splitting procedure presented here does away with iterations, however, a non-iterative scheme must also be developed to deal with the other equations, such that the accuracy and stability of the overall scheme are preserved. It is often the case that the poor resolution of these scalar fields (including the species equation for chemical-reacting flows) undermines the integrity of the overall solution procedure.

The $k-\varepsilon$ model in differential form is:

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho u_j k) = \frac{\partial}{\partial x_j}\left(\mu_k \frac{\partial}{\partial x_j}\right) + S_k \tag{29}$$

and

$$\frac{\partial}{\partial t}(\rho \epsilon) + \frac{\partial}{\partial x_j}(\rho u_j \epsilon) = \frac{\partial}{\partial x_j}\left(\mu_\epsilon \frac{\partial}{\partial x_j} + S_\epsilon\right) \tag{30}$$

where

$$S_k = G - \rho\epsilon \tag{31}$$

and

$$S_\epsilon = \frac{\epsilon}{k}[C_1 G - C_2 \rho\epsilon] \tag{32}$$

and

$G = -\overline{u_i\prime\, u_j\prime}\frac{\partial U_i}{\partial x_j}$ is the generation term. The eddy diffusion coefficients μ_k and μ_ϵ are related to eddy viscosity by

$$\mu_k = \frac{\mu_t}{\delta_k}$$

$$\mu_\epsilon = \frac{\mu_t}{\delta_\epsilon}$$

$$\text{resulting in the eddy viscosity } \mu_t = C_u\rho\frac{k^2}{\varepsilon} \tag{33}$$

The splitting procedure used in this study is to reconstruct the source terms such that

$$S_k = G - \frac{\rho^2 C_\mu}{\mu_t}k^2 \tag{34}$$

and

$$S_\epsilon = C_1 C_u \frac{G}{\mu_t}\rho k - C_2 C_\mu \rho^2 \frac{k}{\mu_t}\epsilon \tag{35}$$

In doing so, the differential equations can be split into the following:

$$\left(\frac{\rho}{\delta t} - K_o + \frac{\rho^2 C_\mu}{\mu_t^n}k^n\right)k^* = K\prime(k^*) + G^n + \frac{\rho^n K^n}{\delta t} \tag{36}$$

and **Predictor:**

$$\left(\frac{\rho}{\delta t} - L_o + C_2 C_\mu \rho^2 \frac{k^*}{\mu_t^n}\right)\epsilon^* = L\prime(\epsilon^*) + C_1 \frac{G}{\mu_t^n} C\mu\rho k^* + \frac{\rho^n \epsilon^n}{\delta t} \tag{37}$$

Corrector:

$$\left(\frac{\rho}{\delta t} - K_o + \frac{\rho^2 C_\mu}{\mu_t^*}k^*\right)k^{**} = K\prime(k^*) + \mu_t^* G + \frac{\rho^n k^n}{\delta t} \tag{38}$$

and

$$\left(\frac{\rho}{\delta t} - L_o + C_2 C_\mu \rho^2 \frac{k^{**}}{\mu_t^*}\right)\epsilon^{**} = L\prime(\epsilon^*) + C_1 \frac{G}{\mu_t^*} C_\mu \rho k^{**} + \frac{\rho^n \epsilon^n}{\delta t} \tag{39}$$

It can be shown that k^{**} and ε^{**} are second order approximations in δt to the original equations (25) and (26). Thus, further corrections in k and ε are not needed.

3.2 Chemical Reacting Flows

The pressure-velocity procedure has also been extended to compute chemically reacting flows. In reacting flows, very large density gradients arise, leading to strong non-linear coupling of the equations. The incorporation of the species and energy equations demands significant restructuring of the predictor and corrector steps in the algorithm.

The governing equations for multiple species undergoing chemical reactions are the continuity, momentum, energy, and species equations. In generalised tensor notation they can be written:

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{40}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}\mu\frac{\partial u_k}{\partial x_k}\right] \tag{41}$$

$$\frac{\partial}{\partial t}(\rho h) + \frac{\partial}{\partial x_j}(\rho u_j h) = \frac{\partial}{\partial x_j}\left[\Gamma\frac{\partial h}{\partial x_j}\right] + \sum_{i=1}^{N}\dot{R}_i h_{if} + \frac{\partial p}{\partial t} \tag{42}$$

$$\frac{\partial}{\partial t}(\rho f_i) + \frac{\partial}{\partial x_j}(\rho u_j f_i) = \dot{R}_i + \frac{\partial}{\partial x_j}\left(\Gamma\frac{\partial f_i}{\partial x_j}\right), \quad i = 1, 2, \ldots, N \tag{43}$$

where ρ is the density; u_i the velocity; p the pressure; μ the effective viscosity; h the static enthalpy; Γ the effective diffusion coefficient; f_i the mass fraction of chemical species; and $\dot{R}_i$ the chemical source term.

In addition to the equations described above, expressions are required for the thermodynamic quantities. In the present study, the JANNAF data bank [33] and the CEC76 program were incorporated for chemical equilibrium and thermodynamic property estimations. In the CEC76 program, the minimization of the Gibbs free energy method was used to calculate the composition of chemical equilibrium species. The static enthalpy and the fluid density are then obtained by:

$$h = \sum_{i=1}^{N} f_i h_i(T) \tag{44}$$

and

$$\rho = p / R_o T \sum_{i=1}^{N} \frac{f_i}{Mw_i} \tag{45}$$

where R_o is the universal gas constant; and Mw_i, the molecular weight of species.

For finite rate chemistry, the production rates for each of the species ($\dot{R}_i$) required in Eq. (43) are obtained using the laws of mass action. For a general homogeneous chemical reaction, which may proceed in both the forward and reverse directions, the stoichiometric equation can be written as

$$\sum_{i=1}^{N} \nu'_{ji} \Lambda_i \underset{k_{bj}}{\overset{k_{fj}}{\rightleftharpoons}} \sum_{i=1}^{N} \nu''_{ji} \Lambda_i, \quad j = 1, 2, \ldots, N_R \tag{46}$$

where N_R is the number of reaction steps.

The law of mass continuity states that the net production of species i by reaction j is:

$$\left(\frac{dC_i}{dt}\right)_j = \left(\nu''_{ij} - \nu'_{ij}\right) \left[k_{fj} \prod_{ij}^{N} C_i^{\nu'_{i,j}} - k_{bj} \prod_{i=1}^{N} C_i^{\nu''_{ij}}\right] \tag{47}$$

where C_i is the concentration of species. The net rate of change in concentration of specie i by all reactions is found by summing the contribution from each reaction considered

$$\left(\frac{dC_i}{dt}\right) = \sum_{j=1}^{N_R} \left(\frac{dC_i}{dt}\right)_j \tag{48}$$

In reacting flows, a coupled implicit solution procedure of a chemical kinetics/fluid dynamics problem would require the inversion of a complex system of matrices. In the present methodology, the chemical kinetics and the fluid dynamic solutions are decoupled in performing the integration by using the operator-splitting technique [35, 36]. This procedure is embedded in the previously described predictor-corrector sequence with special treatment of the species equation. Using the operator representation, lets the governing equations be written in the following form:

$$\frac{\partial \rho f_i}{\partial t} + C(f_i) + D(f_i) = \dot{R}_i \tag{49}$$

where $C()$ and $D()$ are convective and diffusive operators for species f_i respectively.

To facilitate the splitting technique, the chemical kinetic solution only involves a time dependent term during the predictor step. Thus, the equation

$$\frac{d}{dt}\rho f_i = \dot{R}_i(f_1, f_2, \cdots, f_n) \tag{50}$$

is integrated in a fully implicit fashion. The effective chemical source terms are then determined by dividing the increment of the chemical species by the fluid residence time.

Subsequently, the convection and the diffusion part of the species equations are then implicitly integrated in the correction step. The corrector step integrates the fluid dynamic part with the effective chemical source terms from the predictor step as follows:

$$C(F_i) + D(f_i) = \left(\frac{d\rho f_i}{dt}\right)_{eff} \tag{51}$$

3.3 Multiphase Flows

The pressure velocity procedure is extended to include the dynamical equations for spray droplets. Due to the strong coupling between two phases in terms of momentum, heat and mass exchange, the incorporation of the dispersed droplet phase requires considerable extension of the corrector steps in the algorithm. These new formulations are described in this subsection.

The general approach used for gas/droplet flows in this study is the so-called "tracking" approach. This approach requires formulating droplets dynamics in a Lagrangian frame within the continuum media for which an Eulerian formulation is utilized. This approach has the flexibility of handling a poly-dispersed spray system and is ready for extension to dense spray effects, as compared to the ”two-fluid” approach. In the Eulerian-Lagrangian two phase approach, the governing equations for gas phase are:

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x_i}(\rho U_i) = \dot{S}_m \tag{52}$$

$$\frac{\partial \rho U_i}{\partial t} + \frac{\partial}{\partial x_j}(\rho U_j U_i) = -\frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j}(\Gamma_{ij}) + \bar{F}_i \tag{53}$$

and for the particles:

$$\frac{dx_i}{dt} = v_i \tag{54}$$

$$\frac{dv_i}{dt} = F_{pi} + g_i \tag{55}$$

Since the formulation here is essentially a statistical approach, each computational parcel represents a large number of droplets having equal location, velocity, size, and temperature. The two-way coupling between the two phases is accounted for by the interaction terms, where

$$F_{pi} = \frac{U_i + u_i' - v_i}{\tau} \tag{56}$$

$$\dot{S}_m = \sum_{p=1}^{Np} N_p \dot{m}_{ev,p}/dV \tag{57}$$

for evaporating spray

$$F_i = \sum_{p=1}^{Np} \left[N_p \dot{m}_{ev,p} \left(v_i\right)_p - \frac{4}{3}\pi \rho_d r_p^3 N_p \left(\frac{dv_i}{dt}_p\right)\right] /dV \tag{58}$$

in which dV denotes the computational cell, and the effective relaxation time $\tau = t_*/f$, with $t_* = \rho_d d_p^2/18$ and $f = C_D Re_p/24$.

The goal of the present method is to build the coupling procedure of the two- phase interactions in a $non - iterative, time - accurate$ frame which eliminates the conventional *global iterations* between the two-phases as depicted in the Particle-Source-In-Cell (PSIC) [37] methodology. The PSIC method first proposed by Crowe at Washington State University in the late

1970's, is still the state-of-the-science methodology and is widely used in industry. Our method arranges the two-phase coupling into a sequence of predictor-corrector steps, utilizing the operator-splitting technique, and thus does not require global iterations. This method is described briefly as follows: We seek the finite difference form of the governing equations (49) and (51) as follows

$$\left(\frac{\rho}{\delta t} - A_o\right) U_i^{n+1} = H\prime(U_i^{n+1}) - \delta_i P^{n+1} + S_i + F_i^{n+1} \qquad (59)$$

and

$$\frac{v_i^{n+1} - v_i^n}{\delta t} = \frac{U_i^{n+1} + U_i' - v_i^{n+1}}{\tau^{**}} + g_i \qquad (60)$$

The superscripts n and n+1 denote time events t^n and t^{n+1} respectively. Operator A_o and $H'(\ )$ are constructed from the second-order upwind scheme for the convection term and the central difference scheme for the diffusion term, respectively. The effective relaxation time τ is evaluated at the second corrector level (**), to be defined later. The key step in our method is the separation of the interaction term as follows:

$$F_i^{n+1} = -S_u^{**} U_i^{n+1} + R_u^{**}; \qquad (61)$$

$$F_i^{**} = -S_u^{*} U_i^{**} + R_u^{*}; \qquad (62)$$

and so on.
In the above splitting procedure, the terms S_u^{**} and R_u^{**} are obtained by rearranging equation (56):

$$S_u^{**} = \frac{1}{dV} \sum_{p=1}^{Np} N_p M_p / \left(\delta t + \tau_p^{*}\right) \qquad (63)$$

and

$$R_u^{**} = \frac{1}{dV} \sum_{p=1}^{Np} N_p M_p / \left(\delta t + \tau_p^{*}\right) \left(V_i^n - u_i' + g_i \delta t\right)_p \qquad (64)$$

We now divide the predictor-corrector procedure as follows:

Predictor Step

$$\left(\frac{\rho}{\delta t} - A_o\right) U_i^* = H'(U_i^*) - \delta_i p^n + S_i - S_u^n U_i^* + R_u^n \tag{65}$$

$$\frac{V_i^* - V_i^n}{\delta t} = g_i + \frac{(U_i')^n - V_i^*}{\tau^n} \tag{66}$$

To strongly couple the interaction terms, we use the solved V_i^* and v_i^* to evaluate τ^*, S_u^*, and R_u^*, such that a second approximation to the gas velocity can be performed:

$$\left(\frac{p}{\delta t} - A_o\right) U_i^{*T} = H'(U_i^*) - \delta_i p^n + S_i - S_n^* U_i^{*T} + R_u^* \tag{67}$$

This is the essential step which eliminates the global iterations. A similar procedure was derived for the first corrector and the second corrector step involving droplet source terms in the pressure correction equations [38]. Turbulence effects then are added during the corrector steps. This approach is a unique approach, including a non-iterative feature which is consistent for all the physics involved. The algorithm is time accurate and requires no under-relaxation for all the steps involved.

4 VALIDATION STUDIES

We have carried out detailed linear stability analyses based on model equations and von-Newman Fourier mode technique. Both algorithms are shown to be unconditionally stable with second order accuracy on time domain discretisation. Detailed discussions can be found in the Ph. D. dissertations of Zhou [39] and Jiang [40]. Both algorithms have been successfully coded into existing CFD codes. Specifically, the PISOC algorithm is implemented in the MAST computer code for the first predictor and the remaining corrector steps were established to utilize the concept of MFICE algorithm. This iteration sometimes is required especially for transonic flow calculations. Various cases including both steady-state and time-dependent incompressible flows, subsonic, transonic, supersonic and hypersonic flows involving turbulence effects, chemistry effects as well as multiphase flows were studied and validated against relevant experimental data or other CFD results in the literature. In the following, highlights of these results will be summarized. Other details involving grid set-ups, grid sensitivity studies, convergence histories and boundary condition set-ups can be found in the three Ph.D. dissertations. [39, 40, 41].

4.1 Driven Cavity Flow

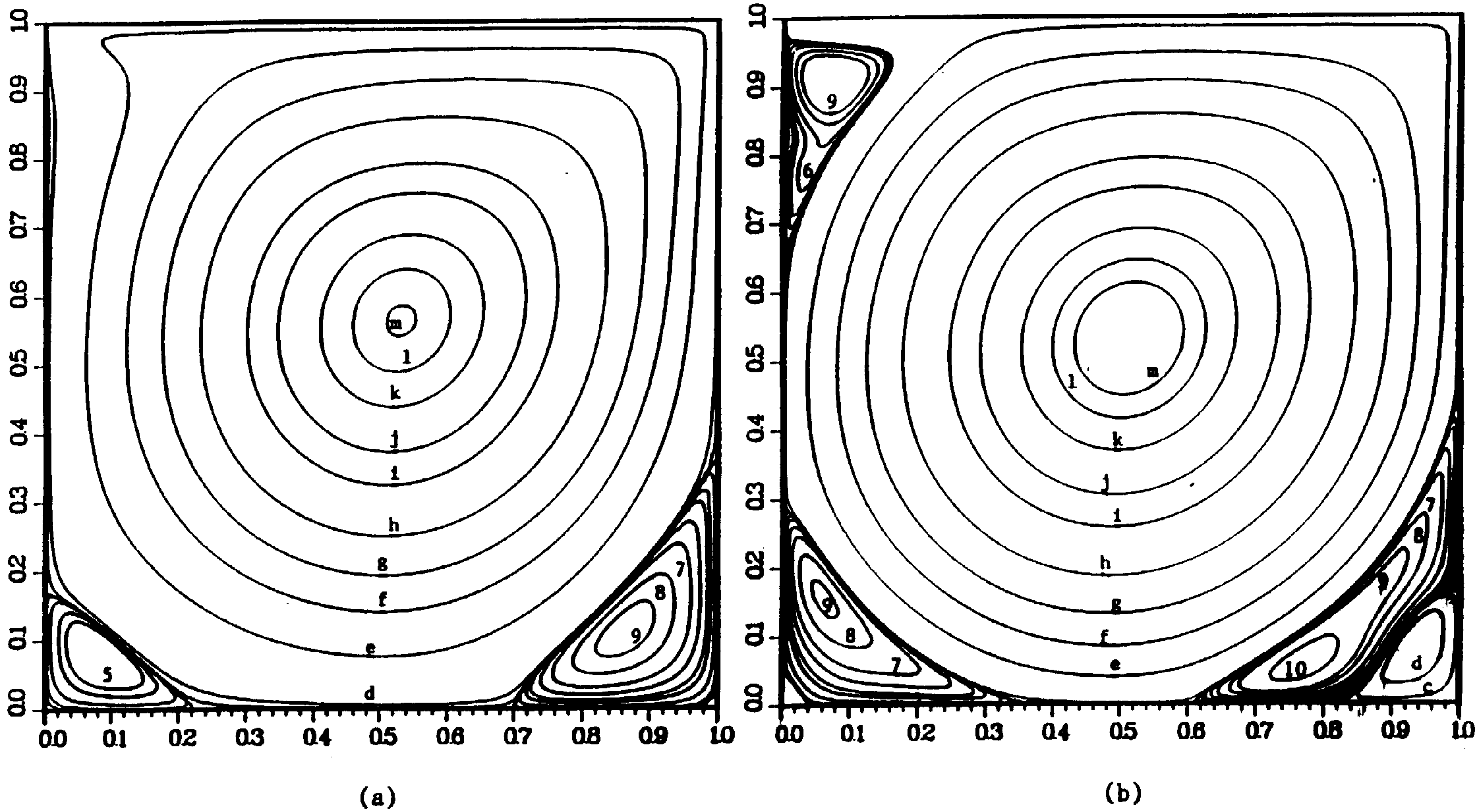
A square two-dimensional cavity with the top wall moving at a constant speed is calculated for the Reynolds number 1000 and 10,000. this case converges in one run in 80 CPU (CRAY/XMP) seconds with $\delta t = 1.0$ and a 51×51 grid system. Figure 1(a&b) shows the computed streamline contours for $Re = 1000$ and $10,000$ respectively. The contours are plotted with the same level of Ghia [42], who used a much finer grid system and long run time. This study demonstrates the accuracy of the current method and insensivity of the convergent rate due to grid refinement.

4.2 Two-Dimensional Circular Cylinder

The flow over a two-dimensional circular cylinder is an example of the external unsteady flow when the Reynolds number is not too small. A 41×41 "0"-grid was algebraically generated. A fixed velocity was set at the outer boundary and a cyclic boundary condition was set at the front center line. Instantaneous stream function plot at several time steps are shown in Figure 2. More detailed calculations and comparisons of this case can be found in [43]. 500 Steps of this calculation take 175 CPU seconds and 0.2 M words core memory. This study demonstrates the time-accurate aspect of the current algorithm.

4.3 Backward Facing Step Flows

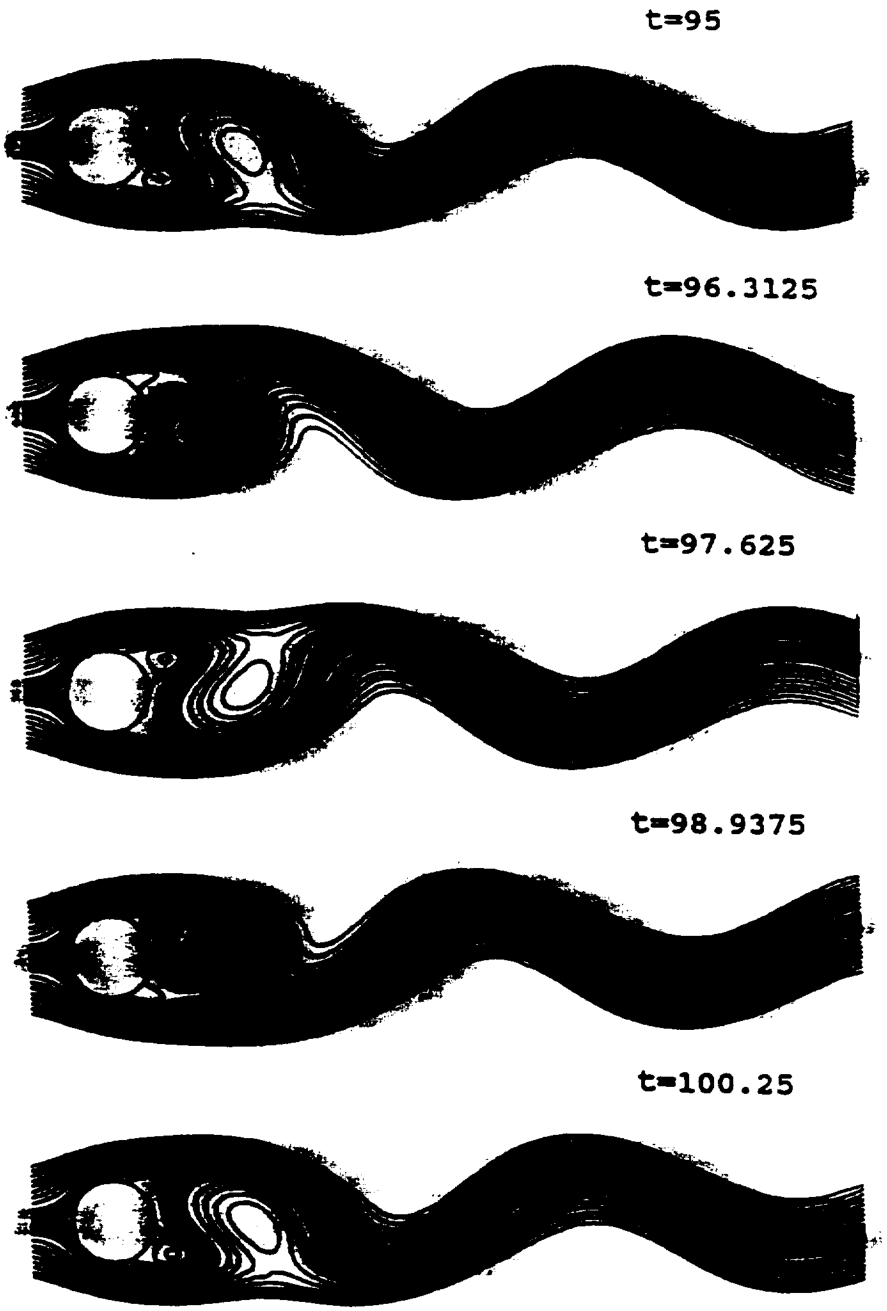
Both benchmark cases of the laminar and turbulenct viscosity flows over a two-dimensional backward facing step were tested. For the laminar case, a critical case of $Re = 800$ is selected and the standard $k - \varepsilon$ model is implemented for calculating the turbulent flow case. A 81×51 uniform grid was used and the time step $\delta t = 0.5$ was taken for the laminar flow case. 350 time steps are used to reach steady-state solutions with the reattachment length at 11.5 step heights which compares favorably with



contour	ψ value	contour	ψ value	contour	ψ value	contour	ψ value
b	-1.0E-7	g	-0.0500	l	-0.1150	5	1.0E-4
c	-1.0E-5	h	-0.0700	m	-0.1175	6	2.5E-4
d	-1.0E-4	i	-0.0900	2	1.0E-6	7	5.0E-4
e	-0.0100	j	-0.1000	3	1.0E-5	8	1.0E-3
f	-0.0300	k	-0.1100	4	5.0E-5	9	1.5E-3

Figure 1a. Streamlines of cavity flow (*Re* = 1000, *grid* = 51 × 51).
b. Streamlines of cavity flow (*Re* = 1000, *grid* = 81 × 81).

ORIGINAL PAGE IS
OF POOR QUALITY



t=95

t=96.3125

t=97.625

t=98.9375

t=100.25

Figure 2. Instantaneous streamline pattern for flow over a cylinder at Re = 200.

INS3D results. A 47 × 33 non-uniform grid is used for the turbulent flow case. The calculated reattachment length is 6.18 step heights; in good agreement with other CFD results [44]. The turbulent flow case took 60 CPU seconds for 90 time steps ($\delta t = 0.6$). The calculated streamlines are shown in Figure 3 and 4 for the laminar and turbulent flow case respectively. This study demonstrates that the extention to include turbulence models does not affect the efficiency of the algorithm.

4.4 Quasi-One-Dimensional Inviscid Flow

The above mentioned algorithms were then implemented into the MAST code for one -dimensional benchmark case calculations to cover a wide range of Mach numbers. First, a one-dimensional symmetric nozzle flow as calculated for subsonic, supersonic and shock flows. The calculations were performed on a uniform spacing of 101 grid points. Initial conditions for all variables were set over the whole domain as being constant and equal to the known analytical solution at the inlet with the exception of the exit pressure which was set at a fixed value. The inlet boundary condition consisted of extrapolating the static pressure from the interior which, together with the given values of total pressure and total temperature, allows an isentropic calculation of the other variables. At the outlet the static pressure was held fixed, and any unknown variable was found from the extrapolation.

Figure 5 shows the steady state Mach distributions for subsonic, chocked and supersonic flows. This study demonstrates the most important features of the results, which are the capabilities of low to high speed flow calculations, the sharpness of shock capturing, accuracy of shock positioning and speed of computation. The efficiency assessments of the algorithms are summarized in Table 1. Examination of Table 1 shows that the algorithms gave satisfactory shock capturing results with large CFL number and converged fast for all special flow regimes.

4.5 Compressible Channel Flows Over a Circular Bump

Validations of the PISOC algorithm for all-speed capabilities were further tested for two-dimensional flows. All the calculations were performed using a normalized time step of 0.1 for 100 time steps to reach steady state

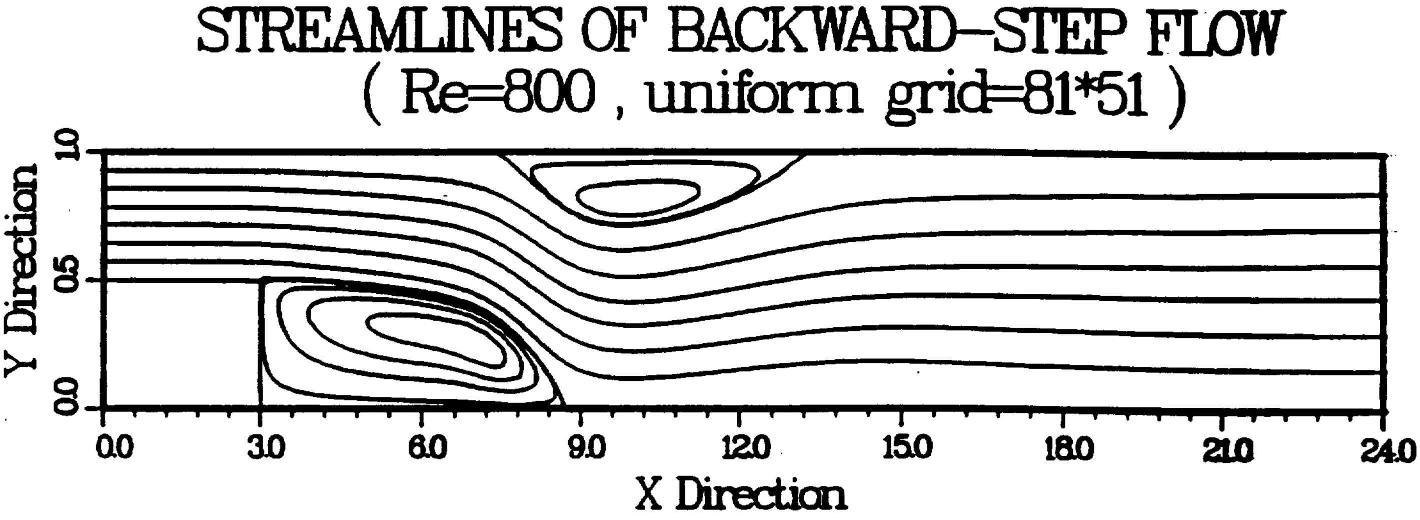


Figure 3. Streamline of laminar flow over backward facing step.

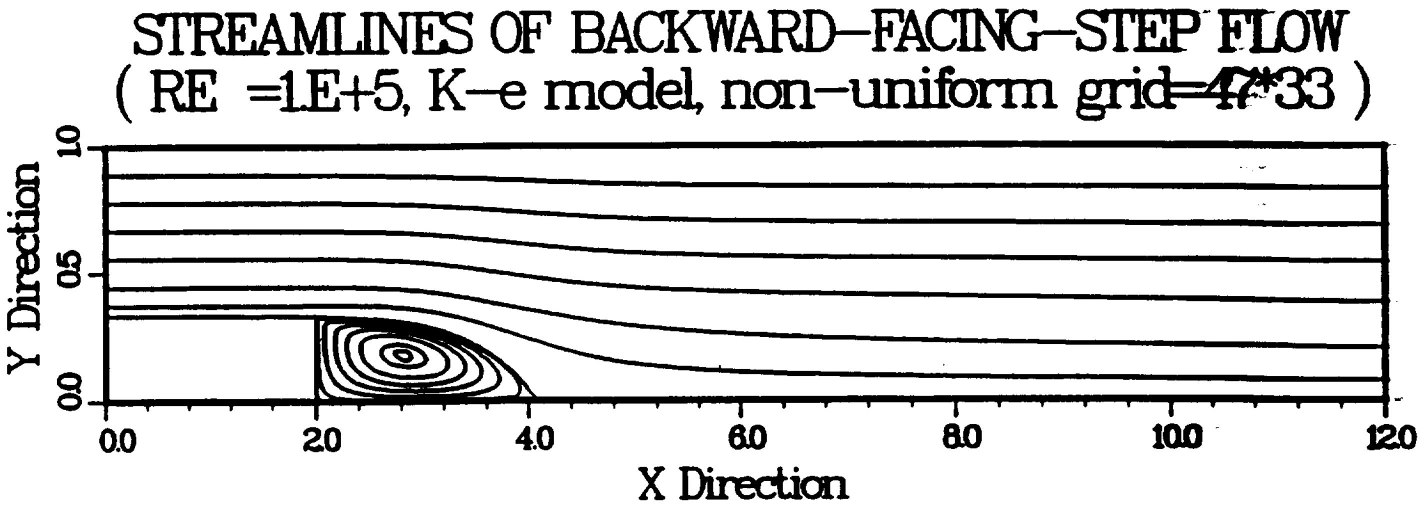


Figure 4. Streamline of turbulent flow over backward facing step.

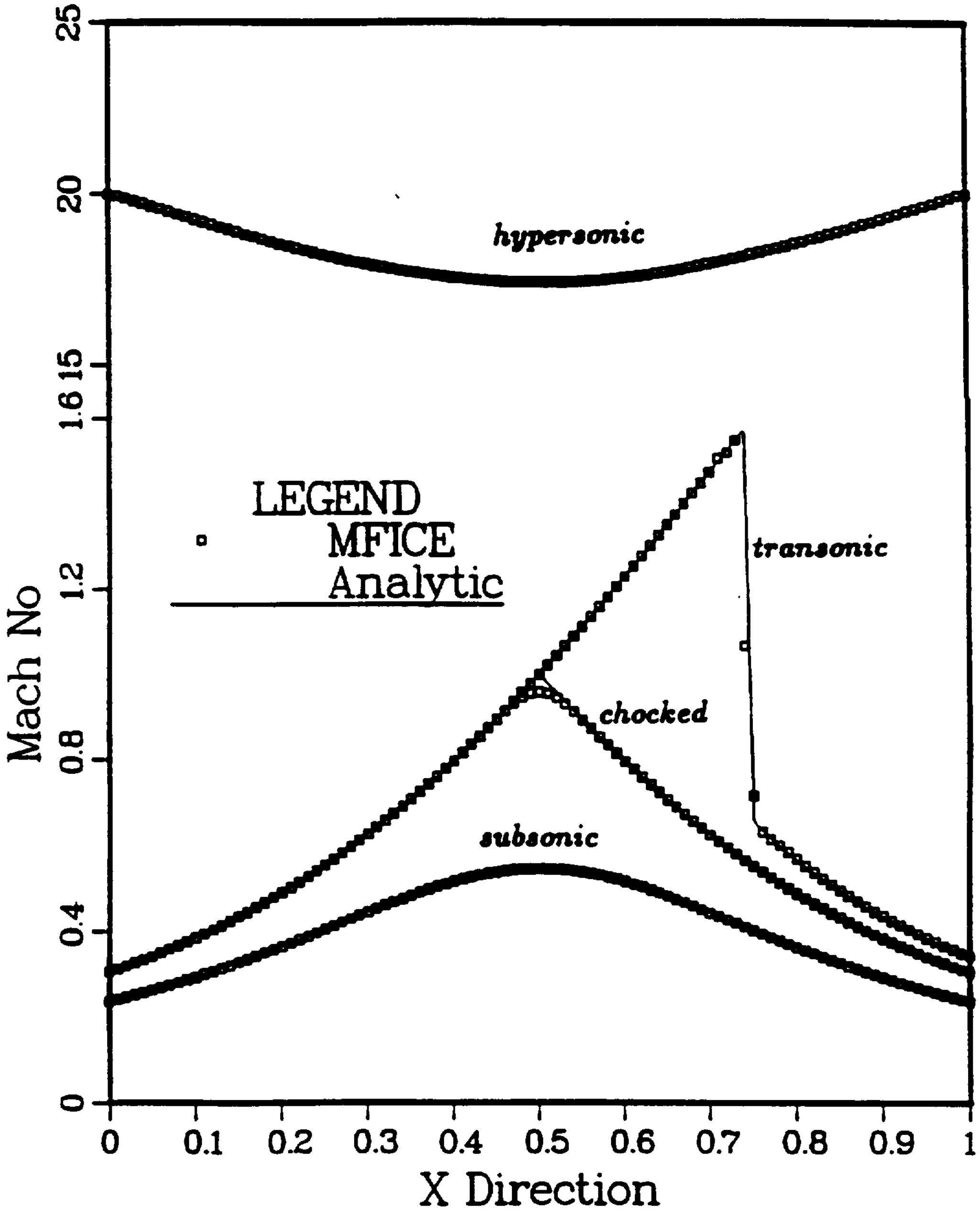
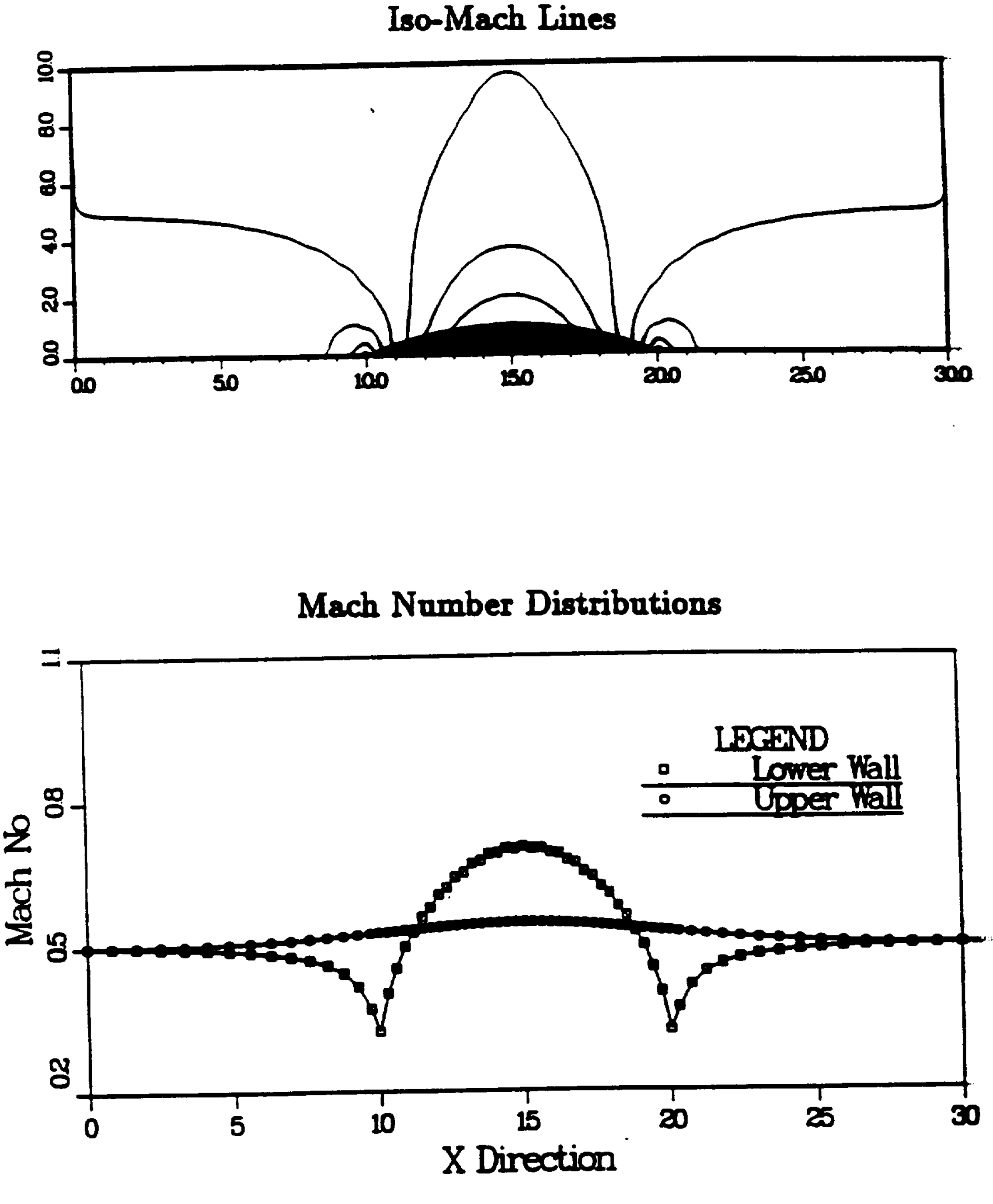


Figure 5. Quasi-one-dimensional inviscid.nozzel flow.

Case	Grid	Δt	CFL_{max}	Step#/Error	CPU (sec) Cray-XMP	$\frac{\text{CPU}}{Grid \times Step\#}$	Scheme
subsonic	101	0.02	1.E+3	32 / 1.E-6	0.135	4.2E-5	C-O ($\varphi = 1.0$)
chocked	101	0.01	1.E+3	79 / 1.E-6	0.32	4.0E-5	C-O ($\varphi = 1.0$)
transonic	101	5.0	1.E+5	200 / 1.E-4	0.71	3.5E-5	C-O ($\varphi = 0.0$)
hypersonic	101	5.E-4	35	250 / 1.E-6	0.93	3.7E-5	C-O ($\varphi = -\frac{1}{3}$)

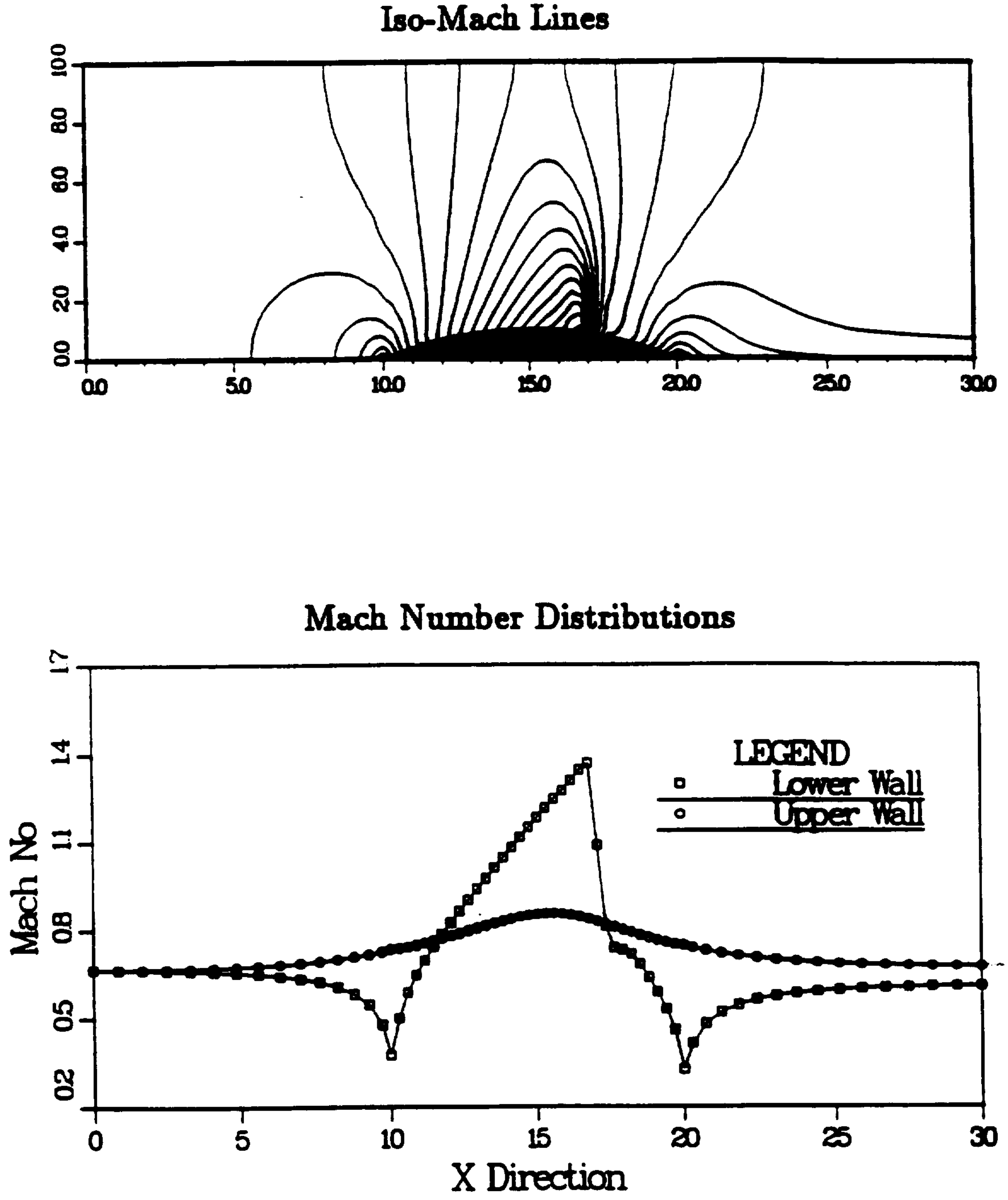
† C-O = *Chakravarthy – Osher* TVD '85 .

Table 1. Numerical parameters for quasi-one-dimensional nozzle flows


Iso-Mach Lines

Mach Number Distributions

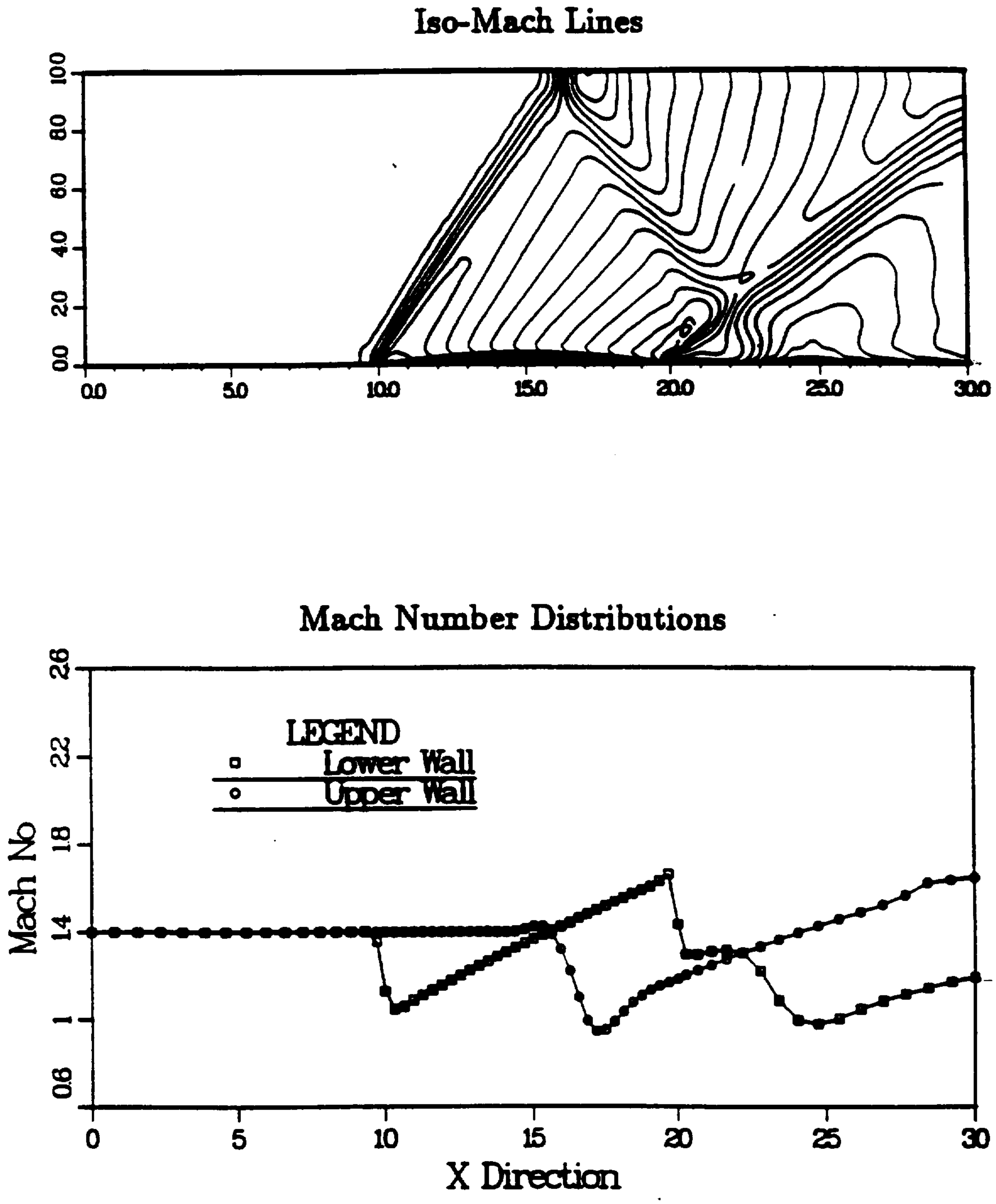
Figure 6. Results of bumped channel flow ($M_{-\infty} = 0.5$).



Iso-Mach Lines

Mach Number Distributions

Figure 7. Results of bumped channel flow ($M_{-\infty} = 0.675$).



Iso-Mach Lines

Mach Number Distributions

Figure 8. Results of bumped channel flow ($M_{-\infty} = 1.4$).

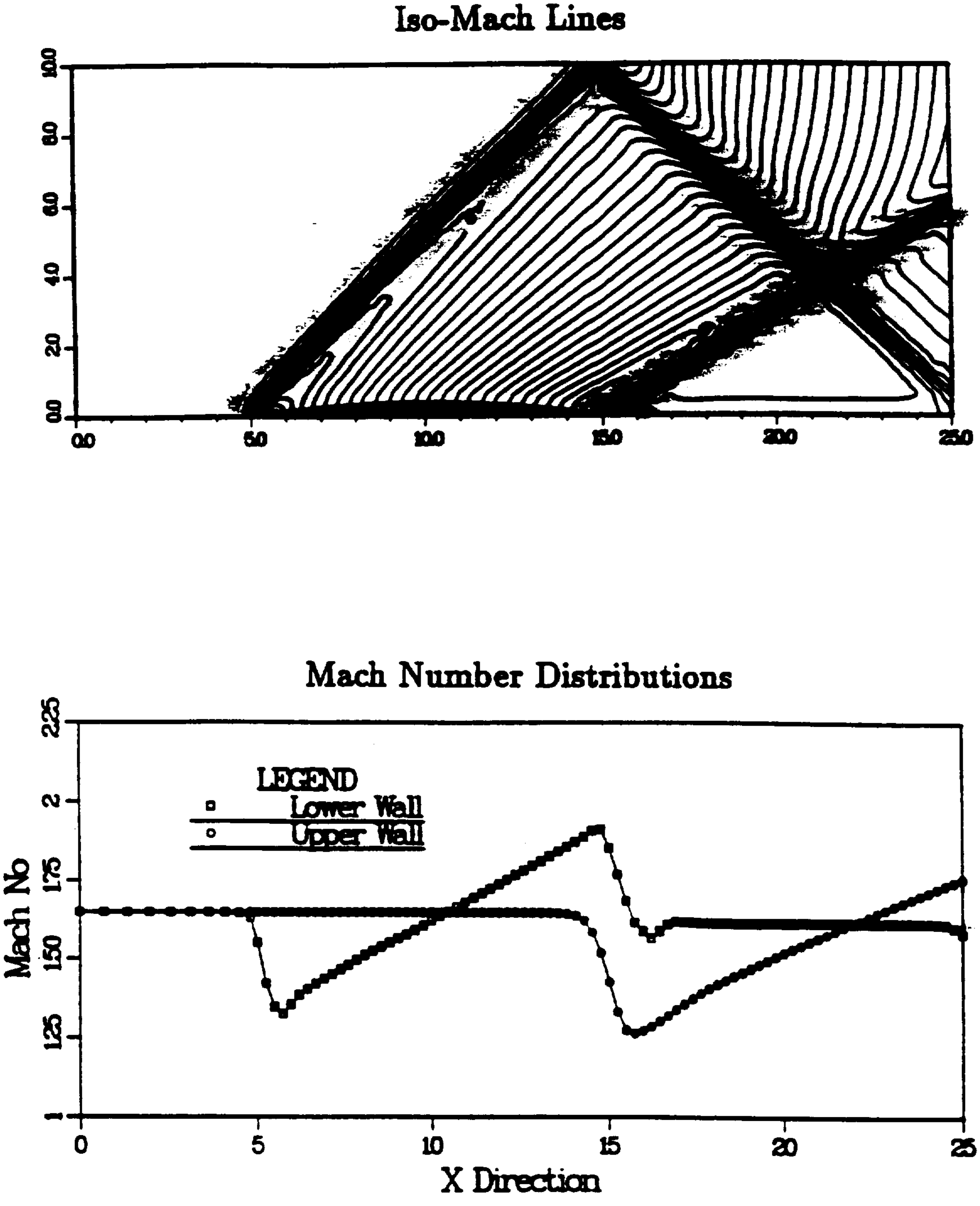


Figure 9. Results of bumped channel flow ($M_{-\infty} = 1.65$).

solutions. Initial values for all variables were set constant over the whole domain for the flows tested, ranging from subsonic to hypersonic conditions.

The calculated Mach number contours and surface Mach number profiles are shown in Figure 6-9 for subsonic, transonic and supersonic features of the results showing the capabilities of low to high speed flow calculations, the sharpness of shock capturing, accuracy of shock positioning, and speed of computation. The efficiency of the algorithms is very consistent. For all cases tested the CPU is 0.18 ms per grid point per time step, which is a very fast convergence rate among the available pressure-correction methods. An axisymmetric inviscid flow striking a blunt body leading edge with a free stream mach number up to $Ma_\infty = 10$ has also been tested, and the calculated static Mach lines and the static temperature along the axisymmetric line are shown in Figure 10.

4.6 Chemical Reacting Flows

A careful validation of the present MAST code with both equilibrium chemistry and finite rate chemistry has been carried out. First, an equilibrium chemistry example involving hypersonic viscous flow, with free stream Mach number of 10.0 past a two-dimensional blunt body with circullar nose is tested. The Reynolds number is 8,600 and the free stream temperature is assumed to be 300 K. Figure 11 illustrates the iso- Mach contours of both the ideal gas case and the equilibrium air case. In Figure 12, the calculated static temperatures are plotted along the flow symmetric line. The effects of equilibrium chemistry on the static temperature jump and shock location are significant in these calculations, which agree well with other calculations in the literature.

The second case was the low-speed Burke-Schuman diffusion flame. The geometry and inlet conditions are illustrated in Figure 13. A global two-step Methane/Oxygen finite rate reaction model of [45] is employed for this case. The calculated flame temperature contours and mass fractions along the centerline are shown in Figure 14. The results compare very well with the analytical solutions reported in [47]. This study demonstrates XXX.

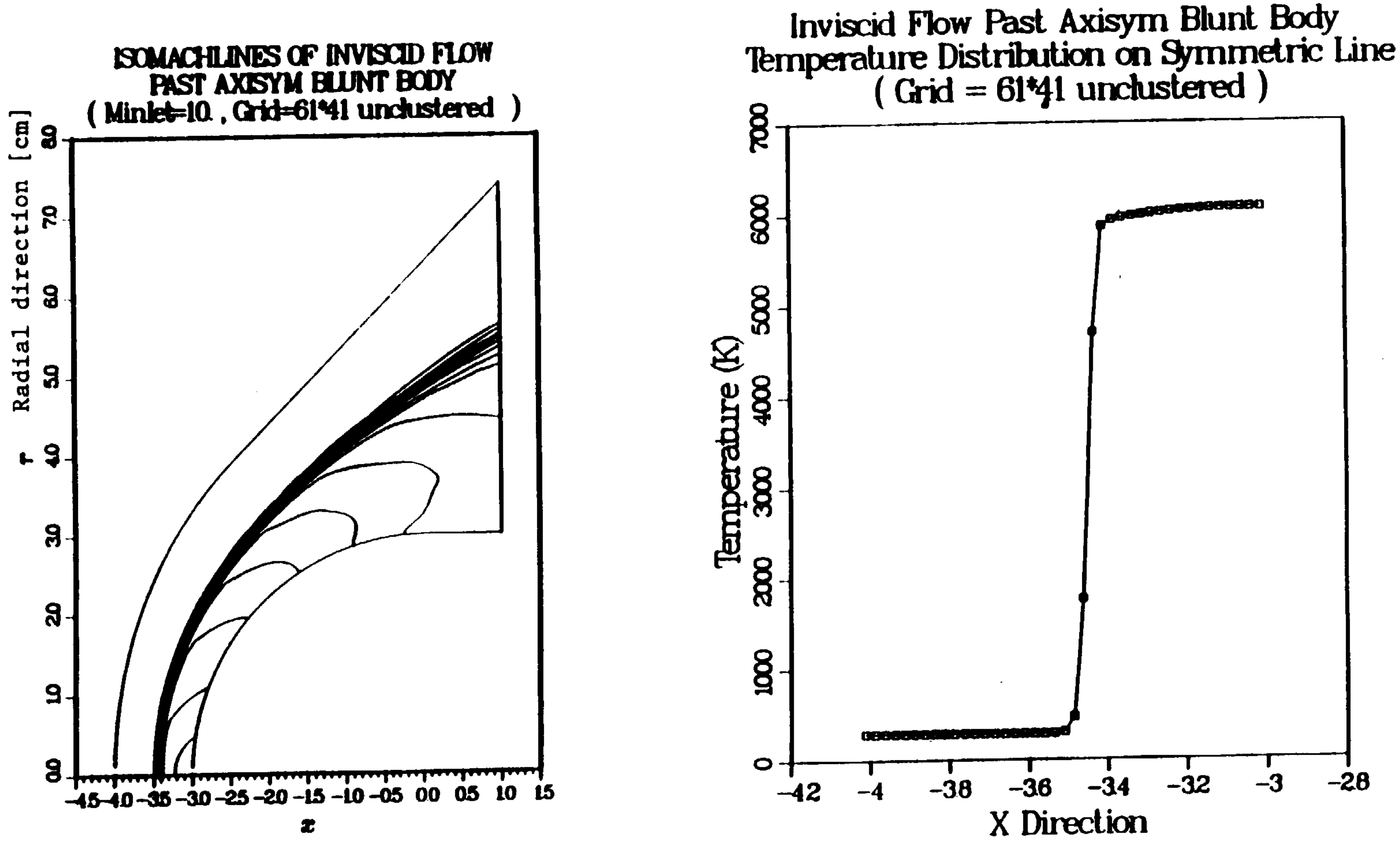


Figure 10. Results of hypersonic flow over blunt body.

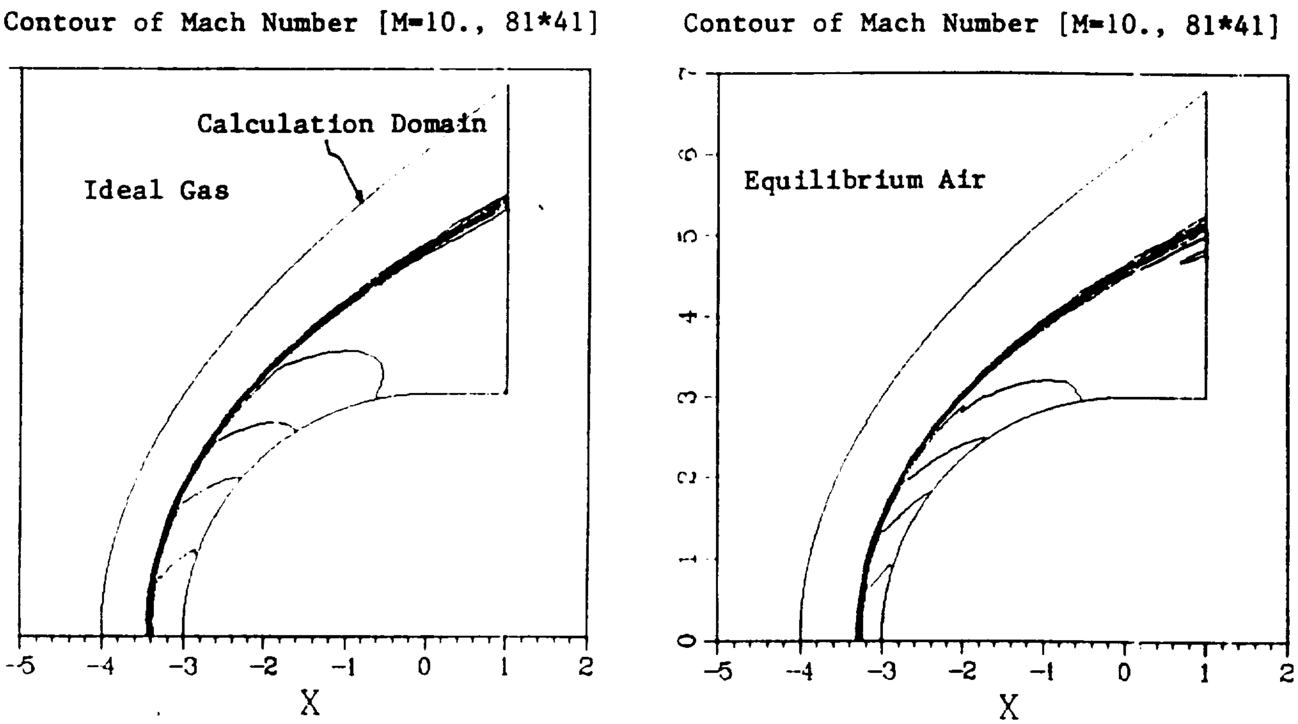


Figure 11. Viscous hypersonic flow over a blunt body, 81 × 41, (a) Ideal gas air, (b) Equilibrium chemistry air.

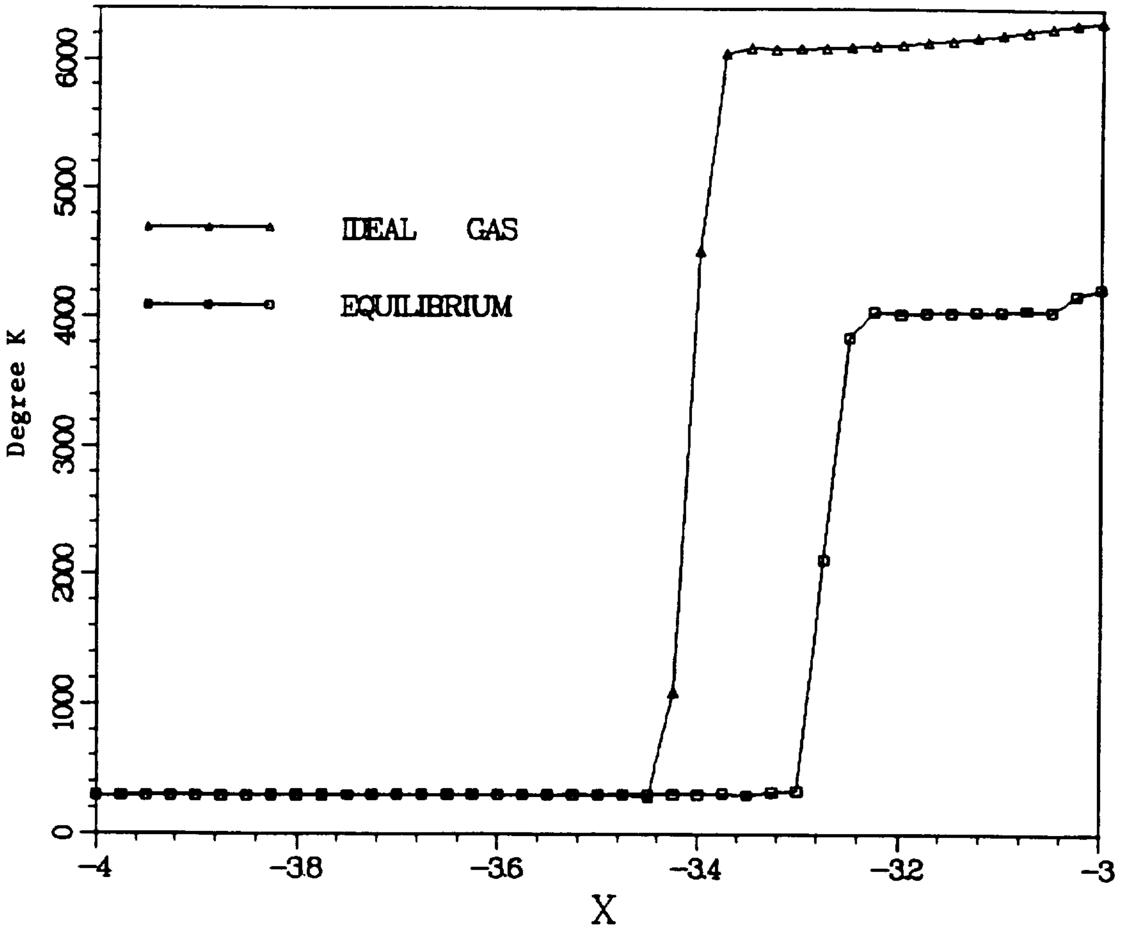


Figure 12. Temperature profiles along the forward center line.

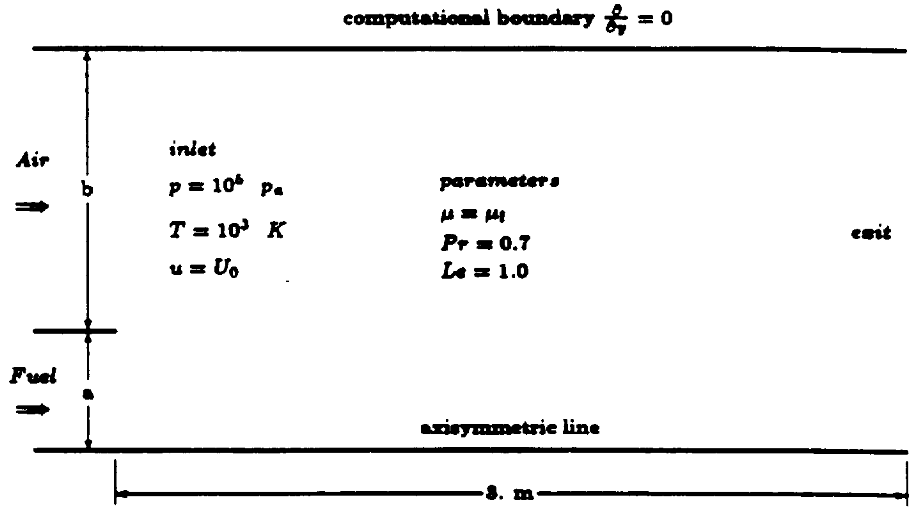
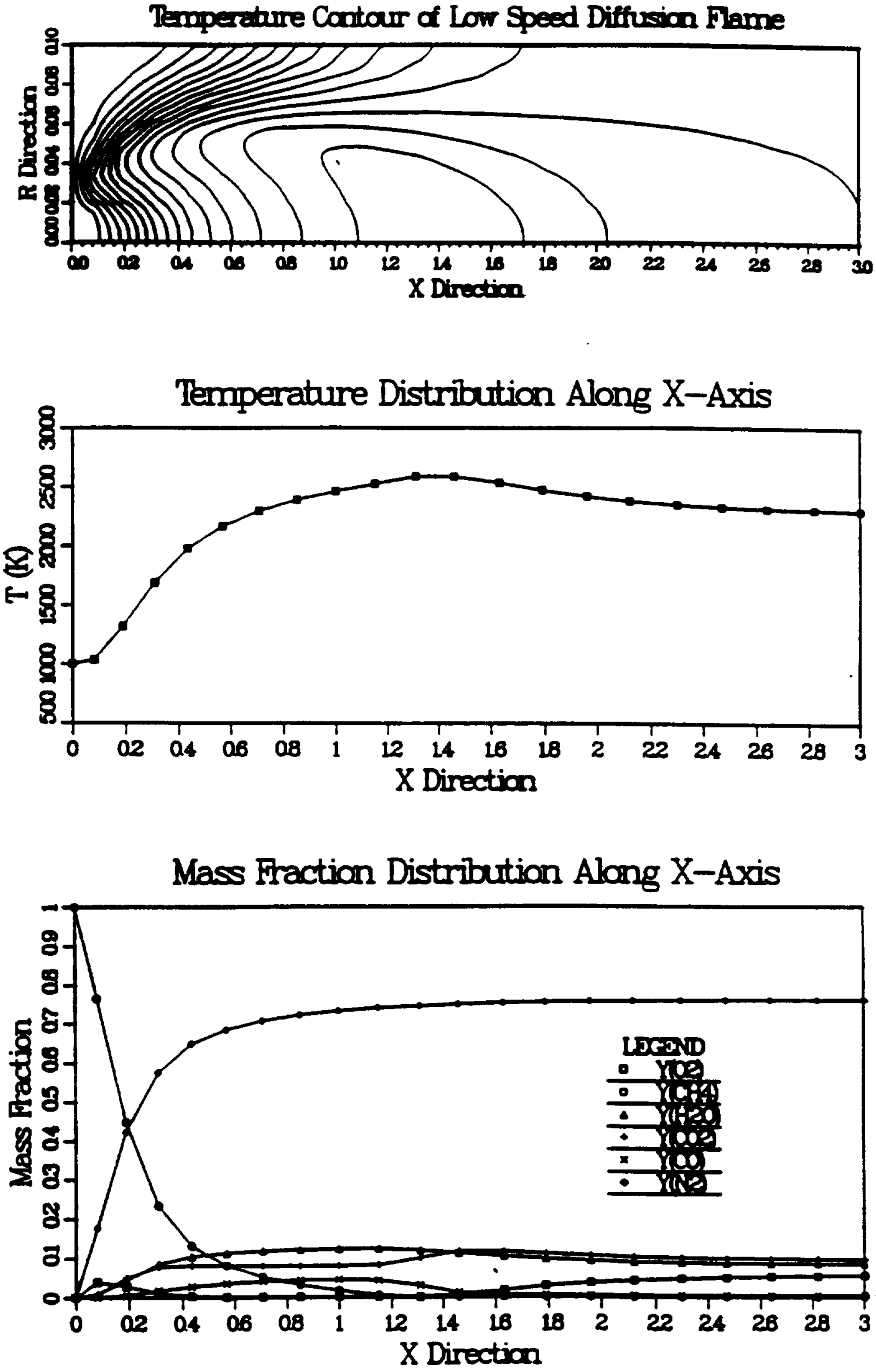


Figure 13. Schematics of low speed diffusion flame.



Temperature Contour of Low Speed Diffusion Flame

Temperature Distribution Along X-Axis

Mass Fraction Distribution Along X-Axis

Figure 14. Results of low speed CH_4/Air diffusion flame.

4.7 Multi-phase Flows

For multi-phase flows, we first applied the current method for the prediction of particle dispersion controlling the turbulence with the mounting of different screens. The experimental set up of this case was conducted by Snyder and Lumley in 1971 [46]. Particle densities and sizes are chosen to examine the dispersion of light and small (46.5 μm diameter hollow glass), as well as heavy (87.0 μm solid glass) particles. Five thousand computational particles were sampled to calculate the mean squared dispersion with respect to time. Comparison of the predicted and measured particle dispersion are shown in Figure 15. The agreement is considered quite good. The current method has the flexibility of taking into account both the gravity (crossing trajectory effect) and the non-stokian drag law as compared to the continuum approach and time accuracy.

A poly-dispersed pulsed hollow-cone spray case of practical importance is also chosen for the test condition listed in Table 2. The calculation starts at 5 mm downstream of the nozzle, and the information of particle size distribution and velocity distribution is directly taken from the measurements. Figure 16 shows the particle distribution plot and the gas velocity vectors for a 30 deg spray. With the back pressure of 1 atm, the intersection between the gas and the droplets trajectories is quite strong. The shape of the spray is no longer conical even for a very short time, and the spray penetration is suppressed due to the interactions of the droplets with the induced air flow. These flow patterns and spray shapes compared quite favorably with the experimental results.

The efficiency assessment of the present numerical method is shown in Table 3 for the hollow cone spray case. The CPU time on a CRAY X/MP using the MAST code utilizing the present method for the transient spray calculations with $\delta t = 0.1ms$ is given. It can be seen that the amount of CPU time is reduced about one order of magnitude by using the present method. Also, the present method is rather particle number independent. This is due to the fact that the particles at each time step, and the source terms in the continuous passes, are updated for all the particles at each Eulerian control volume. In contrast for the TEACH/PSIC method, all the particles have to be tracked, and the continuous phase flow field is held frozen between the global interactions at each time step.

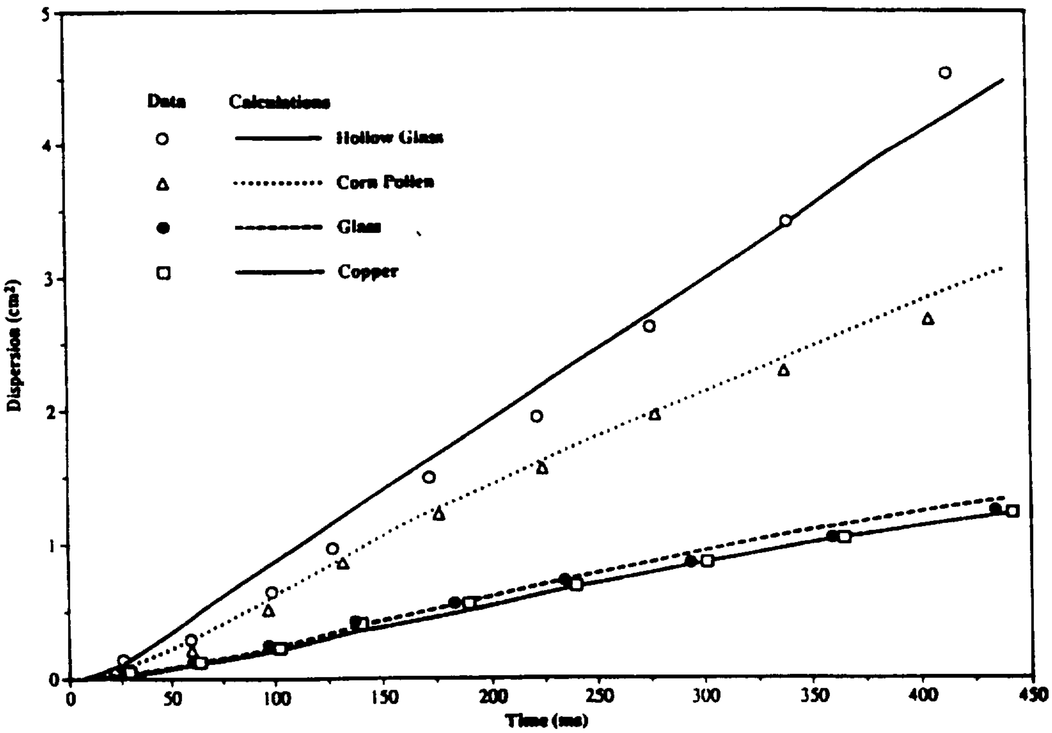


Figure 15. Particle dispersion in a flow with mechanically produced turbulence.

Chamber gas pressure (atm)	Injection velocity ($m\,s^{-1}$)	Injection angle (deg)	Gas density ($kg\,m^{-3}$)	Mass flow rate ($kg\,s^{-1}$)	Sauter mean radius (SMR) (μm)
1	20·0	30	1·123	4×10^{-4}	2·5

Table 2. Hollow cone spray

	MAST 2D		TEACH/PSIC	
	Particles	CPU time (s)	Particles	CPU time (s)
Single-orifice spray				
41 × 61 grid	600	126·9	800	1420
300 time steps	1200	135·7		
Hollow cone spray				
31 × 31 grid	400	74·9	800	934
200 time steps	1000	88·3		

Table 3. Efficiency assessment of numerical method.

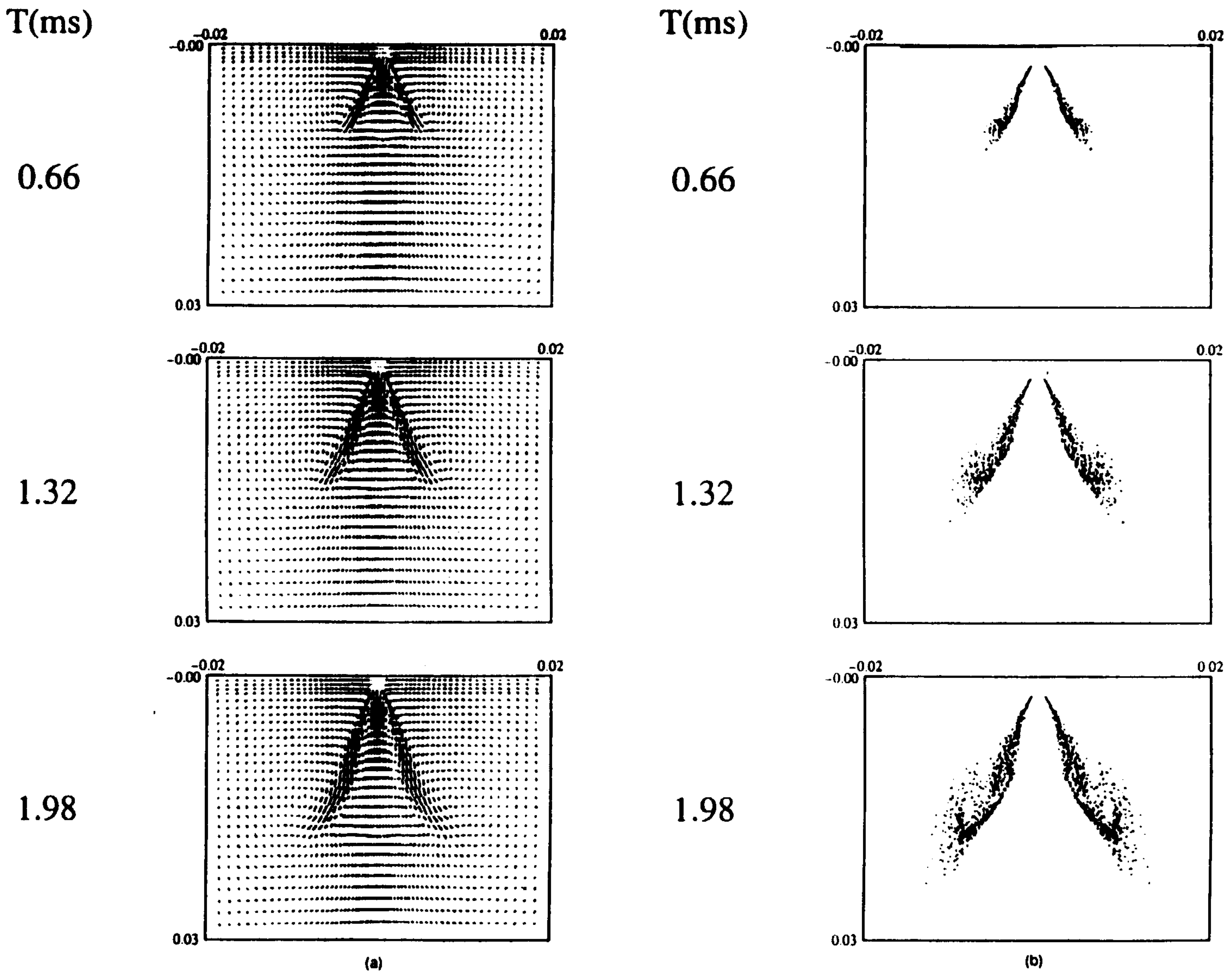


Figure 16. (a) Velocity vectors and (b) particle plot of a 30° hollow cone spray.

The pressure-velocity algorithm developed in this study can be implemented in the framework of finite-volume, finite difference, or finite-element formulations. All the extensions of the current algorithm, including turbulence models, chemistry models (equilibrium and finite rate), and particle tracking subroutines were incorporated into the MAST code in a modular form. These submodel modules are stand-alone solvers and can be transferred from one code to another with few modifications. For example, the chemistry module originally tested in the MAST code was transferred to the MFICE code which utilized staggered grids and iterative procedures with sucessful applications within one month period.

5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Summaries

Efficient pressure-velocity coupling procedures have been developed and investigated for the calculation of fluid flows at all speeds. Two algorithms, PISOC and MFICE were studied, and a combined algorithm was implemented into an existing CFD code, MAST, with physical submodels including two-equation turbulence models, equilibrium and finite-rate chemical reaction mechanisms and gas-droplet multiphase models. Some important conclusions are summarized as follows:

1. A physics-consistent pressure equation is derived to emcompass flows in all Mach number regimes. This pressure equation is implemented in the PISOC algorithm as a pressure correction method and in the MFICE algorithm as a pressure substitution method.

2. Both PISOC and MFICE algorithms are non-iteration algorithms and are unconditionally stable, based on linearized stability analyses.

3. The pressure-velocity algorithm can be extended to include extra physical submodels, including turbulence, chemistry, and multiphase flows without invoking iteration procedures.

4. The current algorithm can be implemented on both staggered and non- staggered grid systems without deteriorating the computational efficiency.

5. Time accuracy can be achieved within prescribed predictor corrector steps without invoking iterations.

6. Extensions for arbitrary body-fitted coordinates do not affect the computational efficiency.

5.2 Algorithm Limitations

Although the current method is developed for general time-accurate transient and steady state flow calculations, the time accurate aspect of the current method for high speed transient calculations is not firmly established. This aspect is strongly coupled with the spatial discretization of the governing equations. In addition, the finite rate chemistry procedure, currently implemented, is only loosely coupled with the fluid dynamics, based on the operator-splitting method. To accurately account for the aero-thermal phenomena, transient chemical reacting flows, such as the ignition processes, and the thermal property changes due to chemistry, have to be strongly coupled with the fluid dynamics through the continuity equation. With the present operator splitting, the current pressure-velocity algorithm cannot be used for time-accurate high-speed and chemical reacting flows involving complex chemistry.

5.3 Recommendations

The future work only concerns the numerical aspects of the pressure-based method.

1. The pressure-velocity coupling procedure should be further explored to include density and temperature effects, due to complex chemistry and other sources, to establish strong coupling among fluid dynamics, thermal energy, and species to achieve time accuracy with arbitrary chemical kinetics.

2. More complex flows will result in more sparsity in time scales associated with various physics. Although the current method is stable using rather large time steps, adaptive time step adjustments should be devised, to match the solution based on optimal time steps associated with the relevant physics.

3. Due to the capability of the current method to cover both incompressible and compressible flow regimes simultaneously, the pressure-velocity algorithm should be extended to include volume displacement effects due to the co-existence of liquid and gas with the same calculation grid. This will involve reformulating the momentum and mass fluxes across the grid boundary and modifying the "pressure" equation within the grid.

4. Currently, the solution method, such as the conjugate gradient square method used here, for solving the pressure equation, is still inefficient. Multigrid methods or multi-level error propagation methods should be included in the future, to speed up solving the pressure equation.

5. For complex configurations, the pressure-velocity coupling procedure should be extended for implementation into multi-zone methodologies. Coupling between calculation zones, involving different boundary conditions, should be developed to smoothly transfer information back-and-forth in the pressure-velocity coupling mode especially for unmatched grid patching.

6 References

1. D. A. Anderson, J. C. Tannehill and R. H. Pletcher, Computational Fluid Mechanics and Heat Transfer, Hemisphere Pub., N. Y., 1984.

2. R. M. Beam and R. F. Warming, "An Implicit Finite-Differnce Algorithm for Hyperbolic Systems in Conservation Law Form", *J. Comp. Phys.*, **22**, 87, 1976.

3. W. R. Briley and H. McDonald, "Solution of the Multi-dimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method", *J. Comp. Phys.*, **24**, 372, 1977.

4. R. W. McCormack, "The Effect of Viscosity in Hypervelocity Impact Cratering", AIAA paper 69-354, 1969.

5. J. L. Steger and R. F. Warming, "Flux Vector Splitting of the Inviscid Gas-Dynamic Equations with Applications to Finite Difference Methods", *J. Comp. Phys.*, **40**, 263, 1981.

6. S. K. Godunov, "A Difference Scheme for Numerical Computation of Discontinuous Solution of Hydrodynamic Equations", *Math. Shornik*, **47**, 271, 1969.

7. S. Eidelman, P. Colella and R. P. Shreeve, "Application of the Godunov Method and Its Second-Order Extension to Cascade Flow Modeling", *AIAA J.*, **22**, 1609, 1984.

8. J. Dukowicz, "A General, Non-Iterative Rieman Solver for Godunov's Method", *J. Comp. Phys.*, **61**, 119, 1985.

9. P. L. Roe, "Approximate Rieman Solvers, Parameter Vectors and Difference Schemes", *J. Comp. Phys.*, **43**, 357, 1981.

10. A. Harten, "On A Class of High Resolution Total Variation Stable Finite Difference Scheme", *SIAM J. Numer. Analy.*, **21**, 1, 1984.

11. H. C. Yee, "Construction of Explicit and Implicit Symmetric TVD Schemes and Their Applications", *J. Comp. Phys.*, **68**, 151, 1987.

12. C. Hirsh, Numerical Computation of Internal and External Flows, Vol. 2, John Wiley and Sons, Inc., New York, 1990.

13. C. L. Merkle and Y. H. Choi, "Computation of Compressible Flow at Very Low Mach Numbers", AIAA-86-0351, 1986.

14. J. Feng and C. L. Merkle, "Evaluation of Preconditioning Methods for Time- Marching Systems", AIAA-90-0016, 1990.

15. J. L. C. Chang and D. Kwok, "A Three-Dimensional Incompressible Flow Simulation Method and Its Application to the SSME", AIAA-85-0175, 1985.

16. S. E. Rogers and D. Kwok, "An Upwind Differencing Scheme for the Time Accurate Incompressible Navier-Stokes Equations", AIAA-88-2583, 1988.

17. J. E. Welch, F. H. Harlow, J. P. Shannon and B. J. Daly, The MAC Method, Los Alamos Report, LA-3425, 1966.

18. A. A. Amsden and F. H. Harlow, Los Alamos Scientific Report, LA-4370, 1970.

19. S. V. Patankar and D. B. Spalding, "A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows", *Int. J. Heat & Mass Transfer*, **15**, 1787, 1972.

20. S. V. Patankar, Numerical Heat Tranfer and Fluid Flow, Hemisphere, New York, 1980.

21. J. P. Van Doormaal and G. D. Raithby, "Enhancement of the SIMPLE Method for Predicting Incompressible Fluid Flows", *Num. Heat Transfer*, **7**, 147, 1984.

22. J. P. Van Doornaal, G. D. Raithby and D. H. McDonald, "The Segregated Approach to Predicting Viscous Compressible Fluid Flows", *J. Turbo ASME*, **109**, 268, 1987.

23. V. L. Sha, *et al.* "COMMIX-1B Code", Argonne National Lab., NUREG/CR-4348, 1985.

24. K. C. Karki and S. V. Patankar, "Pressure Based Calculation Procedure for Viscous Flows at all Speeds in Arbitrary Configurations", *AIAA J.* **27**, 1667, 1989.

25. R. I. Issa, A. D. Gosman and A. P. Watkins, "The Computation of Compressible and Incompressible Flows by a Non-Iterative Implicit Scheme", *J. Comp. Phys.*, **63**, 66, 1986.

26. R. I. Issa, "Solution of the Implicit Discretised Fluid Flow Equations by Operator-Splitting", *J. Comp. Phys.*, **62**, 40, 1985.

27. G. M. Neely and R. W. Claus, "Acclelerated Convergence for Incompressible Flow Calculations", AIAA paper 85-0058, 1985.

28. A. Wanik and U. Schnell, "Some Remarks on the PISO and SIMPLE Algorithm for Steady Turbulent Flow Problems", *Int. J. Numer. Method for Fluids*, **12**, 555, 1989.

29. Y. Q. Hu and S. T. Wu, "A Full-Implicit-Continuous-Eulerian Scheme for Multi-dimenisonal MHD Flows", *J. Comp. Phys.*, **55**, 33, 1984.

30. F. H. Harlow, and A. A. Amsden, "A Numerical Fluid Dynamics Calculation Method for All Flow Speeds", *J. Comp. Phys.*, **8**, 197, 1971.

31. C. L. Merkle, S. Venkateswaran, P. E. O. Buelow, "The Relationship Between Pressure-based and Density-based Algorithms", AIAA Paper 92-0425, 1992.

32. C. L. Merkle, P. E. O. Buelow and S. Venkateswaran, "Comparison Betweent the PISO Algorithm and Preconditioning Methods for Compressible Flow", 10th Workshop for Computational Fluid Dynamics Application in Rocket Propulsion, NASA/MSFC, April 28-30, 1992.

33. D. R. Stall and H. Prophet, "JANNAF Thermochemical Tables", 2nd ed., NSRDS -5B5 37, 1971.

34. S. Gordon and B. J. McBride, "Computer Program for Calculation of Complex Chemical Equilibrium", NASA SP-273, 1976.

35. J. B. Greenburg, "A New Reliable Family of Split-Operator Method for Computing Reacting Flow", *Int. J. Numer. Method Fluid*, **4**, 653, 1984.

36. Z. J. Chen, C. P. Chen, and Y. S. Chen, "A Pressure-Corrector Method for the Calculation of Compressible Chemical Reacting Flows", AIAA-92-3032, 1992.

37. C. T. Crowe, M. P. Sharma and D. E. Stock, "The Particle-Source-In Cell (PSIC) Model for Gas-Droplet Flows", *J. Fluid Eng.*, **99**, 325, 1977.

38. C. P. Chen, H. M. Shang and Y. Jiang, "An Efficient Pressure-Velocity Procedure for Gas Droplet Two-Phase Flow Calculations", *Int. J. Num. Method Fluids*, **15**, 233, 1992.

39. N. Zhou, A New Pressure-Velocity Method for All-Speed Flows with Arbitrary Geometric Boundaries, Ph.D. Dissertation, The University of Alabama in Huntsville, Nov. 1992.

40. Y. Jiang, A New Pressure-Velocity Coupling Procedure for Inviscid and Viscous Flows at All Speed, Ph.D. Disseration, The University of Alabama in Huntsville, Dec. 1992.

41. H. M. Shang, Numerical Studies of Spray Combustion in Liquid-Fueled Engines, Ph.D. Dissertation, The University of Alabama in Huntsville, Nov. 1992.

42. U. Ghia, K. N. Ghia and C. T. Shiu, "High-Resolution for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method", *J. Comp. Phys.*, **48**, 387, 1982.

43. Y. Jiang, C. P. Chen and P. K. Tucker, "Multigrid Solution of Unsteady Navier Stokes Equation Using a Pressure Method", *Numerical Heat Tranfer, A*, **20**, 81, 1991.

44. T. Wang and Y. S. Chen, "A Unified Navier-Stokes Flowfield and Performance Analysis of Liquid Rocket Engines", AIAA-90-2494, 1990.

45. C. K. Westbrook and F. L. Dryer, "Simplified Reaction Mechanism for the Oxidation of Hydrocarbon Fuel Flames", *Combustion Science and Technology*, **27**, 31, 1981.

46. W. H. Snyder and J. L. Lumley "Some Measurements of Particle Velocity Autocorreltation Function in a Turbulent Flow", **J. Fluid Mechanics**, **48**, 41, 1971.

47. C. P. Chen, *et al.*, "A Computer Code for Multi-phase All-Speed Transient Flows in Complex Geometries", MAST Version 1.0", October, 1991.

48. T. S. Wang and Y. S. Chen "A Unified Navier-Stokes Flow Field and Performance Analysis of Liquid Rocket Engines" AIAA Paper 90-2494.

49. R. C. Rogers and W. Chinitz, "Using a Global Hydrogen-Air Combustion Model in Turbulent Reacting Flow Calculations" *AIAA J.* **21** 581, 1983.

50. K. Gross, NASA/MSFC, Minutes of 27th JANNAF Combustion Subcommittee Meeting, "CFD Code Survey for Thrust Chamber Application", Nov. 5, 1990, Cheyenne, Wyoming

7 Appendix 1

List of Publications Resulting from this Investigation

1. A New Pressure-Velocity Method for All-Speed Flows with Arbitrary Geometric Boundaries: Modified FICE Method, Ning Zhou, Dissertation Thesis, Department of Mechanical Engineering, The University of Alabama in Huntsville, 1992.

2. A New Pressure-Velocity Method for All-Speed Flows with Arbitrary Geometric Boundaries, Y. Jiang, Ph.D. Dissertation Thesis, Department of Chemical Engineering, The University of Alabama in Huntsville, 1991.

3. Numerical Studies of Spray Combustion in Liquid-Fueled Engine, H.-M. Shang, Ph.D. Dissertation Thesis, Department of Chemical Engineering, The University of Alabama in Huntsville, 1991.

4. Modified Full-Implicit-Continuous-Eulerian (MFICE) Method for All-Speed Flows, N. Zhou, S. T. Wu, and C. P. Chen, *J. Comp. Phys.*, 1991 (submitted)

5. A New Pressure-Based Numerical Method for All-Speed Flows in Arbitrary Configurations, N. Zhou, S. T. Wu, and C. P. Chen, *Computers and Fluids*, 1991 (submitted).

6. Multigrid Solution of Unsteady Navier-Stokes Equations Using a Pressure- Method, Y. Jiang, C. P. Chen, and K. Tucker, *Int. J. Numerical Heat Transfer*, **20**, 88-93, 1991

7. Turbulence Modulation Effects on Evaporating Spray Characteristics, H.-M. Shang, C. P. Chen, and Y. Jiang, AIAA Paper 90-2442, 1990

8. An Efficient Pressure-Velocity Coupling Method for Two-Phase Gas Droplet Flows, C. P. Chen. H.-M. Shang and Y. Jiang, *Int. J. Numerical Heat Transfer*, **15**, 8233-245, 1992

9. A New Pressure-Velocity Coupling Procedure for Inviscid and Viscous Flows at All-Speeds, Y. Jiang, C. P. Chen, and H.-M. Shang, 4th International Symposium on Computational Fluid Dynamics, Vol. 1, 545-550, 1991; submitted to AIAA J., (1992)

10. A Pressure Correction Method for the Calculation of Compressible Chemical Reaction Flows, Z. T. Chen, C. P. Chen, and Y. S. Chen, AIAA Paper 92-3032; submitted to J. Propulsion (1992).

8 Appendix 2

Sample Inputs and Results

Appendix 2
Sample Inputs and Results

A. General Input File

The MAST family computer programs consists of a set of subroutines controlled by a short main program. The fundamental structure can be found in the MAST user's manual version 1.0 [47]. The updated capabilities, resulting from the current study, are summarized in Table A.1. In the following, the updated input file descriptions and the handling of boundary conditions for the 2-D/Axisymmetric MAST code are described.

The Input structure used in the MAST code utilizes blocks of optional namelists under several keywords. Each of the keywords and associated namelists are described here. These "keywords" are optional, i.e., skip if not needed.

There are nine keywords built into the current MAST code. These are : GRID, BOUND, SOLV, PROPERTY, TURBULEN, SPRAY, REACTION, RUN and ENDJOB. In addition, there is one extra block called CONTROL which is used for identifying numeric options in the code. The CONTROL block is usually created first in the input file to be read into the program through Logic Unit 1. All entries are optional. If certain entries require numerical values or logical values, they are entered after the entry names, separated by at least one blank space. SI unit is used for required numerical values. Block name, entry names, the possible range of values, and a short description of the variable are described as follow.

1. CONTROL Block : This block is always put at the top of the input file.

RESTART: It activates the usage of a restart file as initial conditions by reading through LU=4.

SWIRL: It activates the swirl velocity calculations.

IMON,JMON: It specifies the monitor grid point at 2-D map (IMON,JMON), this allows the user to monitor the convergence progress for the selected monitor variable solved.
The default value is (2,2).

MONU,MONV,
MONP, MONTEMP,
MONTK,MONTE: Only one variable can be specified as a monitor variable tracking on screen. Such as velocity at (x,y) location, pressure, temperature, turbulent kinetic energy, dissipation rate. The default is MONU.

ERRCG: Convergency criterion for conjugate gradient Matrix solver. The default value is 0.01.

NCGM: Maximum Conjugate Gradient solver iteration number. The default number is 100.

ERRM: Termination criterion for steady state solutions using the time marching scheme. The default value is 0.0001.

INCOMP: Incompressible flow calculations (=1).

COMPRES: (INCOMP=0) Compressible flow options.

OMGD: 1 for supersonic flows, 0 (default) for low speed flows.

NCRT: Number of corrector steps (default=2).

OMGF: Interpolation parameter for face velocities
1 for interpolation and 0 for averaging.
The default value is 1.

OMGT: For supersonic temperature field relaxation.

PHI: Parameter in the finite difference limiter.
1 --- central, 1/2 --- no name, 1/3 --- 3rd order upwind,
0 --- Fromm schem, -1/2 --- no name, -1 --- 2nd order upwind.

OMGPHI: Weighting parameter for upwind scheme and other scheme. Numerical value of OMGPHI ranges from 0. to 1. The resulting scheme is weighted according to:
PHI*(1-OMGPHI) + 1st order upwind scheme*OMGPHI

END: End of Block input.

For example, a typical CONTROL block for running a laminar backward-facing step flow using a second order upwind scheme and monitoring the convergence of U velovity on grid point (17,4) would have the following input block:

```
CONTROL  INCOMP  OMEGD 0 PHI -1.0 OMGPHI 0.0 ERRCG 1.E-1 ERRM 1.E-4
         IMON 17 JMON 4 MONU  OMGF 0.2
```

A typical CONTROL input for compressible flow calculations is given in Figure A.1.

2. GRID Block:

NX: Number of grid points in the x direction (>3).

NY: Number of grid points in the y-direction (>3).

XLEN: It identifies the length in SI unit of the entire physical calculation domain in the x direction.

YLEN: The length of the entire calculation domain in y direction.

READXY: It activated reading of an externally generated grid point X(I,J), and Y(I,J) from input file "sgrid.d". The input format is given as:

```
        READ(4,9910) NX,NY
        READ(4,9920) ((X(I,J),I=1,NX),J=1,NY)
        READ(4,9920) ((Y(I,J),I=1,NX),J=1,NY)
    9910 FORMAT(2I5)
    9920 FORMAT(10E10.4)
```

This input format can be modified by the user.

UNIFORM: It specifies a uniform grid system.

XDIR,YDIR: It specifies an X-direction grid (only dependent on I) or a Y-direction grid (only dependent on J). Either one must be specified at the beginning of a line.

IST: The grid cell starts from IST index

IEND: The grid cell ends with IEND index

DST: The grid cell starts from DST

DEND: The grid cell ends with DEND

EXP: Grid space stretching factor (see the footnote *) using the power law formula.

DELT: First grid cell size. EXP and DELT can be specified only once.

*----"power law" formula. Such a grid generation can be described as follows:

(a) Explicit "power law":

For example, XDIR IST 10 IEND 25 DST 3.2 DEND 5.6 EXP 1.5, the power-law formulation for such a grid distribution internally calculated as:

$$X(I,J) = DST + \left(\frac{I-IST}{IEND-IST}\right)^{EXP} (DEND-DST)$$

I=IST,........IEND for EXP>0

$$X(I,J) = DEND - \left(\frac{IEND-I}{IEND-IST}\right)^{|EXP|} (DEND-DST)$$

I=IST,........IEND for EXP<0

. If EXP = ±1, grid is uniform

. If EXP > 1, grid is compressed close to DST and expanded close to DEND.

. If EXP < -1, grid is expanded close to DST and compressed close to DEND.

. EXP >1 and exp < -1 can be used to give a symmetric grid distribution.

(b) Implicit "power-law" :

If the user specified first grid cell size DELT instead of EXP, the grid generator automatically computes the stretching factor, EXP, and distributes the grid maintaining the first grid size DELT = X(IST +1, J) -X(IST, J) for DELT > 0 or the last grid size |DELT| =X(IEND, J)-X(IEND-1, J) for DELT < 0.

For example,a typical grid system used for a driven cavity flow in a square domain of 1 meter by 1 meter with grid clustering near the wall regions and top driven lid region can be specifies using the power law formula with a streching factor 1.5 as:

```
GRID NX 51 NY 51
  XDIR  IST  1  IEND 26  DST 0.0  DEND 0.5  EXP  1.5
  XDIR  IST 26  IEND 51  DST 0.5  DEND 1.0  EXP -1.5
  YDIR  IST  1  IEND 26  DST 0.0  DEND 0.5  EXP  1.5
  YDIR  IST 26  IEND 51  DST 0.5  DEND 1.0  EXP -1.5
```

3. Property block (PROP)

VISCOS: It specifies the fluid viscosity. The default value is 1.

DENGAS: It specifies the fluid density. The default value is 1.

CPGAS: It specifies the fluid specific heat. The default value is 1.

KGAS: It specifies the fluid thermal conductivity.The default value is 1.

PRG: It specifies the fluid Prandtl number. The default value is 0.74.

TIN: It specifies the flow field initial temperature. The default value is 0.

PIN: It specifies the flow field initial pressure. The default value is 0.

UIN: It specifies the flow field initial U velocity. The default value is 0.

VIN: It specifies the flow field initail V velocity. The default value is 0.

OMEGA: Angular momentum

GAMMA $\gamma = \frac{C_P}{C_V}$

PSTAG: It specifies the flow field total pressure at the inlet. The default value is 0.

TSTAG: It specifies the flow field total temperature at the inlet. The default value is 0.

Typical examples of using this input block to idendity invisid and viscous flow calculations are given in Figure A.1 and A.2.

4.Runtime block (RUN)

DT: It specifies time step.

DTMIN: It specifies the minimum time step.

DTMAX: It specifies the maximum time step.

CFLN: It specifies the CFL number.
The default value is 1.
The real DT is calculated based on CFLN. If it is less than DTMIN, DT=DTMIN.

NSTEP: Maximum time step number to be computed.

NPR1: It specifies the screen monitor output frequency.

NPR2: It specifies the monitor output restart file frequency.

NEX: It specifies the example number. The user can code in SUBROUTINE EXAMPL. If this option is used, no other input data is required unless a change is desired.

LPFAC: It specifies the monitor to output control volume face quantities.
The default value is 0.

LPGEO: It specifies the monitor to output the Jacobian coiefficients.
The default value is 0.

STOP: If this keyword is specified, MAST code only checks the input data and no execution is done.

For example, to run a job using time step 0.1 second for 300 time steps and to monitor the convergence progess every 10 time steps and save results every 100 time steps requires:

```
RUN   DT 0.1  NSTEP  300 NPR1  10  NPR2 100
```

5. Variable solution block (SOLV)

U,V: It idendify the velocity component to be solved.

P: Pressure will be solved.

TEMP: Temperature will be solved.

TK, TE: Turbulence kinetic energy and dissipation rate will be solved.

SW: Swirl velocity will be solved. (SWIRL must be activated in Selection block).

PATC: Particle tracking is active.

Examples using this input block for running laminar and turbulent flows are given in Figures A.1 and A.2.

6. Turbulent block (TURBU): CT1, CT2, CMU, SME, SMK : Turbulence model constants

CT1 = 1.44,
CT2 = 1,92,
CMU =0.09,
SME =1.3,
SMK =1.0,

TKIN, TEIN : They specify the initial value of turbulence kinetic energy and the dissipation rate. The inlet TKIN is usually estimated based on TKIN=0.01xUINxUIN and the measured TEIN should be used. If no information is known, the dissipation rate is calculated based of some estimated turbulence length scale SCALE.

SCALE: If SCALE specified, TEIN is calculated as $TEIN = CMU \times TKIN^{1.5} / SCALE$

7. Spray block (SPRAY)

SMR: It specifies the particle Sauter mean radius.
If SMR < 0, it specifies a constant particle radius.

X-SQR: It specifies the built-in Chi-square droplet size distribution.

DENPT: It specifies particle density.

TEMP: It specifies particle temperature.

IST, IEND,
JST, JEND: Particles are injected from a interval specified from grid (IST, JST) to (IEND, JEND)

FLOWP : It specifies spray flow rate.

VINJ: It specified particle injection velocity.

NPTS : It specifies particle parcels per time step.

The following example illustrate a solid-cone spray due to a point injection:

```
SPRAY  SMR 150.E-6 X-SQR DENPT 840 TEMP 298
     IST 2 IEND 2 JST 2 JEND 2 NTPS 5
     VINJ 86.41 FLOWP 5.13E-3
```

8. Boundary block (BOUND):

Boundary conditions must be specified in a patched form:

IST, IEND, JST, JEND, B.C. type, variable names.

The variables are defined as

IST, IEND: The starting and finishing points in the I direction grids.

JST, JEND: The starting and finishing points in the J direction grids.
If IST=IEND or JST=JEND, then a line boundary condition is used.

The following keywords used to specify B.C. type :

INLET: Inlet boundary condition, followed by variable names to be described:

OUTLET: Outlet boundary condition.

WALL: No slip wall boundary condition, followed by variables. Wall function of turbulence model activated.

SYMMETRY : It specifies a symmetric b.c.

CYCLE : It specifies a cyclic b.c.

BLOCK : It specifies a blockage. Wall functions of turbulence model activated on block face.

Variable name keywords (specify b.c.)

U : x-direction velocity.

V : y-direction velocity.

TK : Turbulence kinetic energy in INLET.

TE : Turbulence dissipation rate in INLET.

TEMP : Temperature at INLET, WALL or BLOCK.

Q: It specifies the wall heat fluxes. The default value is 0.

YH2, YO2, YHO, YH2O, YH, YO, YHO2, YH2O2,YO3 :
Mass fraction at inlet.
The default values are 0.

Examples for this block can be illustrated by the calculations of a incompressible turbulent backward-facing step flow as follows:

```
BOUND
  IST  1 IEND 59 JST  1 JEND  1 WALL
  IST  1 IEND 59 JST 35 JEND 35 WALL
  IST  1 IEND  1 JST 16 JEND 35 INLET U 1.
  IST 59 IEND 59 JST 1  JEND 35 OUTLET
  IST  1 IEND 11 JST 1  JEND 15 BLOCK
```

9. Reaction block (RCHEM)

EQLM: It specifies the equilibrium chemical reaction.
This option requires another input file 'chem.in'.
The default value is 0.

The input file "chem.in" requires the following keyword:

ns: Number of species to be calculated

ne: Numver of elements to be considered.

nf: Number of species to be frozen if 'frozen' is activated (T).

b0: Element convervation constant, a set of 'ne' numbers.
For example, air 1O2+3.76N2 has
number of element O=1x2=2
number of element H=3.76x2=7.52,
thus tne conservation constants are 2,7.52 or 1,3.76.

chemical species

chemical element

frozen T to activate the frozen chemistry option

A typical example is given in Figure A.4

FINIR: It specifies the finite rate chemical reaction.
The default value is 0.

LCHEM: It specifies the the finite rate reaction model.
The default is 0.
LCHEM = 1 : 2-step H2 + O2 model.
LCHEM = 2 : 8-step H2 + O2 model.

INISPE: It specifies the species be given initail values, the default value is 0.

YH2, YO2, YHO, YH2O, YH, YO, YH2O2, YO3, YHO2:
It specifies the initial mass fraction of species.
The default values are 0

RATIO: It specifies the equivalence ratio in 2-step H2+O2 model.

Examples for chemistry block can be seen in Figures A.3-A.5.

10. ENDJOB: ENDJOB identifies the end of the input file.

B. Sample Calculations

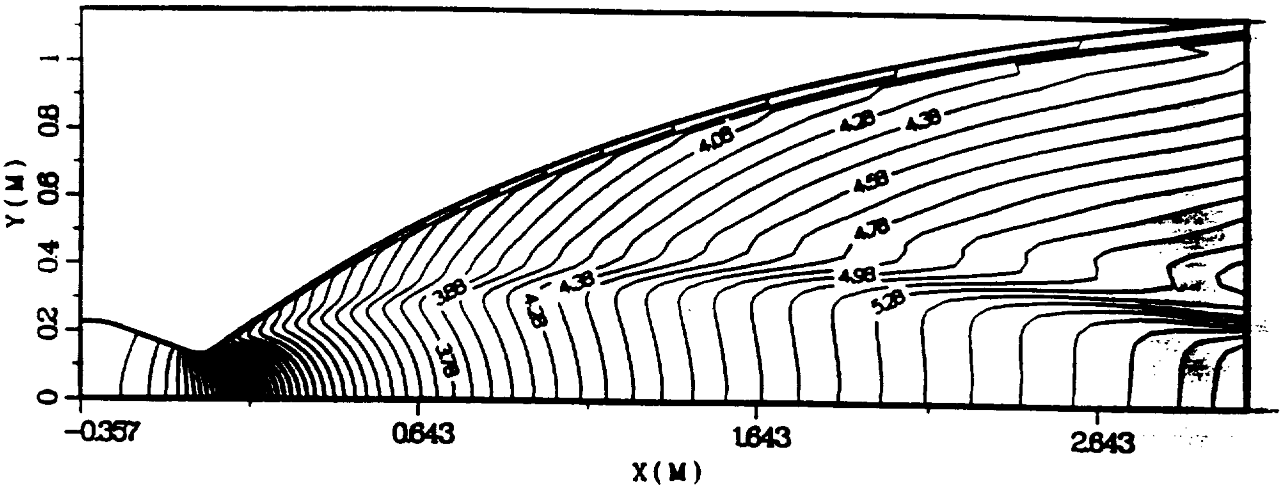
In this section, five sample calculations of SSME thrust chamber flows are presented. The input files used for the calculations are shown based on the geometry and inlet conditions of [48] as well as the calculated iso-Mach contours and temperature contours, in Figures A.1-A.5. The five cases chosen are for inviscid, turbulent non-reacting calculations with the k-epsilon model, turbulent reacting flow with equilibrium chemistry, turbulent reacting flow with a 2-step reaction kinetics model according to [49], and turbulent reacting flows with a 8-step reaction kinetics model [49]. The calculated vacuum specific impulse values are also shown in the figures, which may be compared with experimental data of 453.3 seconds.

Finally, Table A.1 exhibits an updated MAST code capability status from the previous survey date of October 1990 shown in Ref. [50].

SAMPLE CASE 1

```
CONTROL    COMPRES NCRT 3  OMGM 1  NCGM 50
           OMGD 1.0 PHI -1.0 OMGPHI 0.00  OMGT 0.50 OMGF 1.00
           ERRCG 1.0E-2 ERRM 5.0E-6 IMON    81 JMON 10 MONU
;RESTART
GRID  NX  81  NY  41  AXISYM  READXY
BOUND
    IST   1  IEND   1  JST  1  JEND  41  INLET
    IST   1  IEND  81` JST 41  JEND  41  SLIP   Q 0.
    IST   1  IEND  81  JST  1  JEND   1  SYMMETRY
    IST 81 IEND    81  JST  1  JEND  41  OUTLET
TURBULENT   TKIN 3.072 TEIN 10000.
PROPERTY  VISCOS   0.0
         PSTAG 20240946.90  TSTAG 3637.   GAMMA 1.2  GMW 10.18
SOLV  U    V  P  TEMP   TK  TE
RUN   DT 1.E-5 DTMIN 1.0E-7 DTMAX 1.E-2   CFLN 1.00 NSTEP 100
        NPR1 1   NPR2 100   NEX 18 ; LPFAC ; LPGEO
ENDJOB
```

CONTOUR OF MACH NUMBER



CONTOUR OF TEMPERATURE

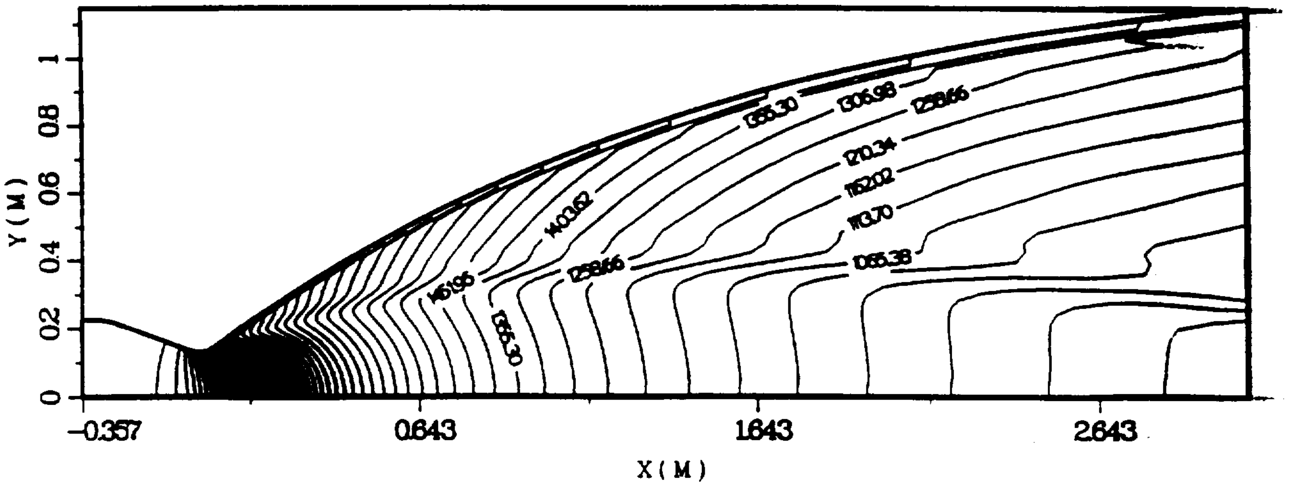
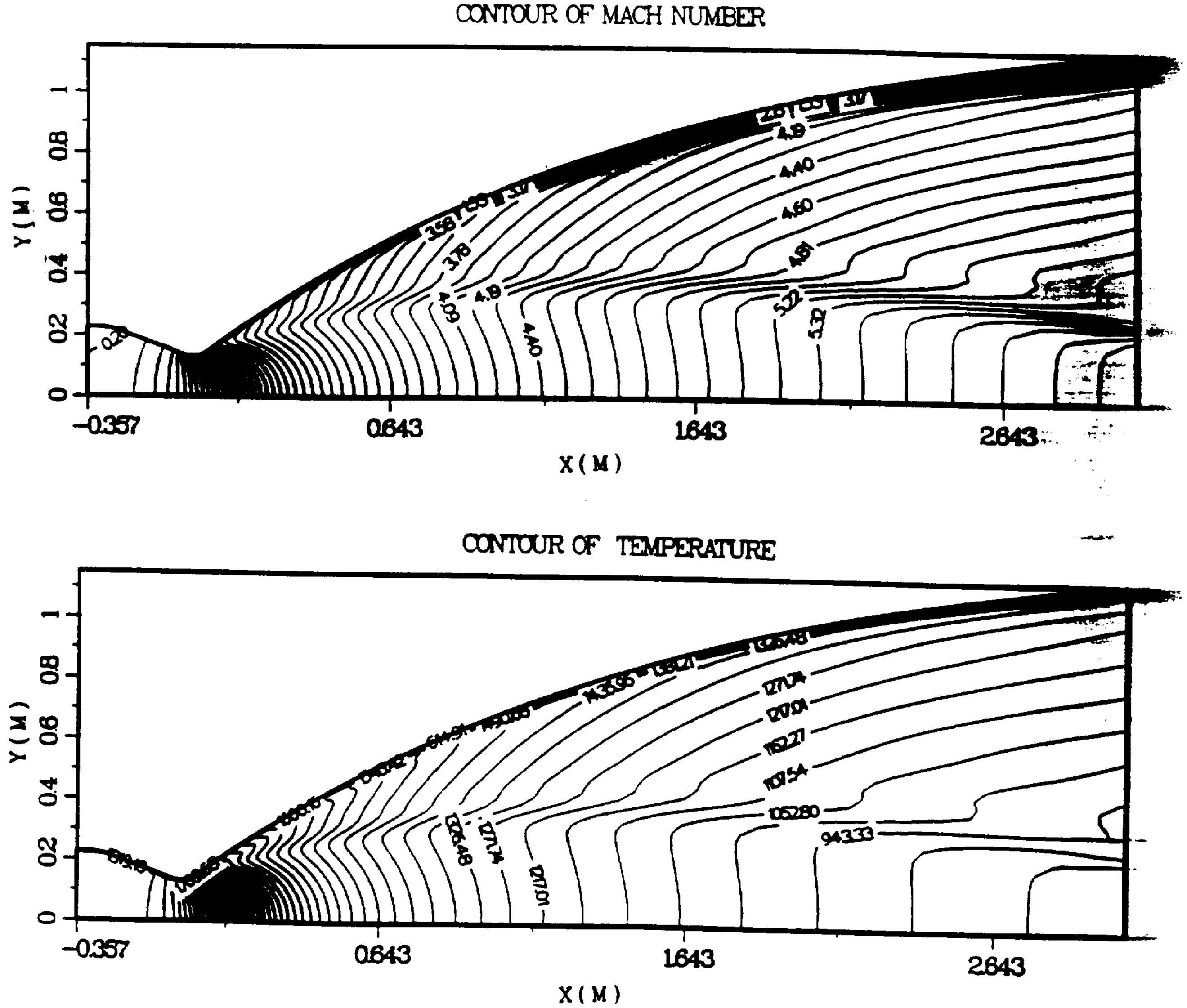


Figure A.1 Sample SSME Nozzle Flow Inputs and Results --- Inviscid.
ISP = 524.44 sec.

SAMPLE CASE 2

```
CONTROL   COMPRES NCRT 2  OMGM 1  NCGM 50
          OMGD 1.0 PHI -1.0 OMGPHI 0.00  OMGT 0.50 OMGF 1.00
          ERRCG 1.0E-2 ERRM 1.0E-5 IMON    81 JMON 10 MONU
;RESTART
GRID  NX  81  NY  41  AXISYM  READXY
BOUND
    IST   1  IEND   1  JST  1  JEND  41  INLET
    IST   1  IEND  81  JST 41  JEND  41   WALL U 0. V 0.0
    IST   1  IEND  81  JST  1  JEND   1  SYMMETRY
    IST 81 IEND    81  JST  1  JEND  41  OUTLET
TURBULENT   TKIN 3.072 TEIN 10000.
PROPERTY  VISCOS  9.05E-5
        PSTAG 20240946.90  TSTAG 3637.   GAMMA 1.2  GMW 10.18
SOLV  U    V  P  TEMP    TK  TE
RUN    DT 1.E-5 DTMIN 1.0E-5 DTMAX 1.E-2   CFLN 4.00 NSTEP 100
         NPR1 1   NPR2 100   NEX 18 ; LPFAC ; LPGEO
ENDJOB
```

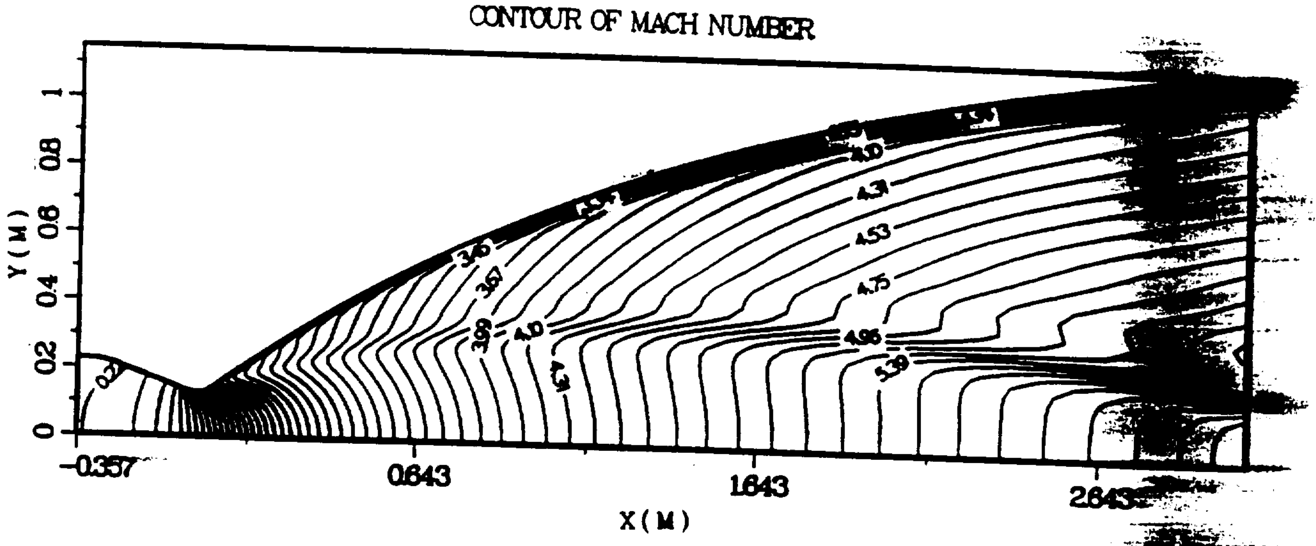

CONTOUR OF MACH NUMBER

CONTOUR OF TEMPERATURE

Figure A.2 Sample SSME Nozzle Flow Inputs and Results --- Turbulent, Non-reacting.
ISP = 513.06 sec.

SAMPLE CASE 3

```
CONTROL    COMPRES NCRT 3  OMGM 1  NCGM 50
           OMGD 1.0 PHI -1.0 OMGPHI 0.00  OMGT 0.50 OMGF 1.00
           ERRCG 1.0E-2 ERRM 1.0E-5 IMON    81 JMON 10 MONU
;RESTART
GRID  NX  81  NY  41  AXISYM  READXY
BOUND
    IST   1  IEND   1  JST  1  JEND  41  INLET
    IST   1  IEND  81  JST 41  JEND  41   WALL U 0. V 0.0
    IST   1  IEND  81  JST  1  JEND   1  SYMMETRY
    IST 81 IEND    81  JST  1  JEND  41  OUTLET
TURBULENT   TKIN 3.072 TEIN 10000.
REACTION
 EQLM
PROPERTY  VISCOS  9.05E-5
       PSTAG 20240946.90  TSTAG 3637.   GAMMA 1.2  GMW 10.18
SOLV  U    V  P  TEMP    TK  TE
RUN   DT 1.E-5 DTMIN 1.0E-5 DTMAX 1.E-2   CFLN 4.00 NSTEP 100
        NPR1 1   NPR2 100   NEX 18 ; LPFAC ; LPGEO
ENDJOB
```



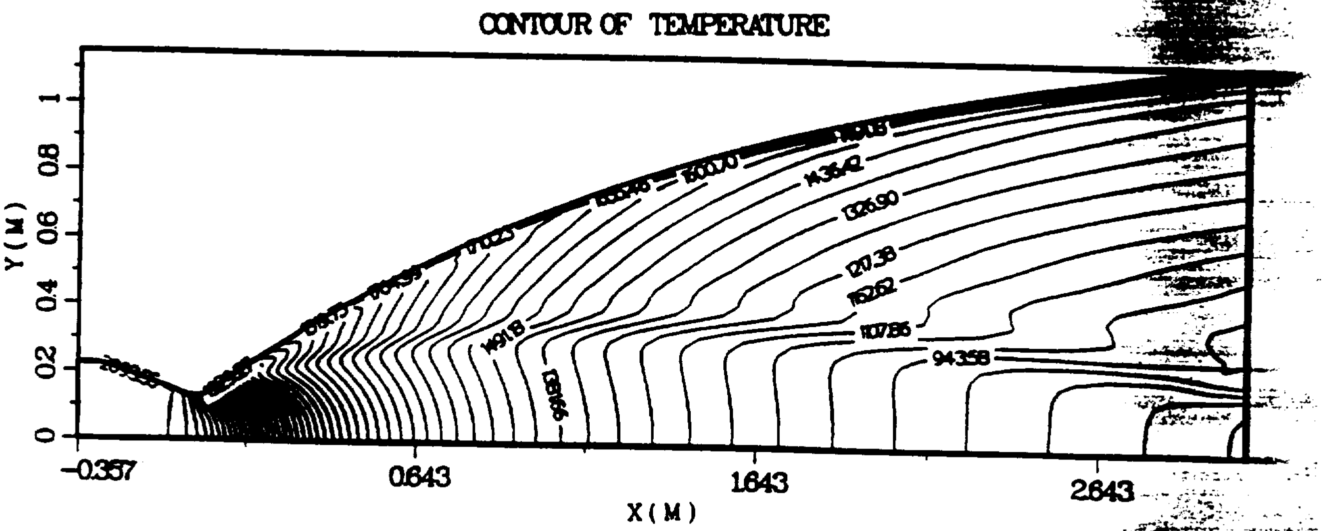


Figure A.3 Sample SSME Nozzle Flow Inputs and Results --- Turbulent, Equilibrium reaction.
ISP = 459.86 sec.

ORIGINAL PAGE IS
OF POOR QUALITY

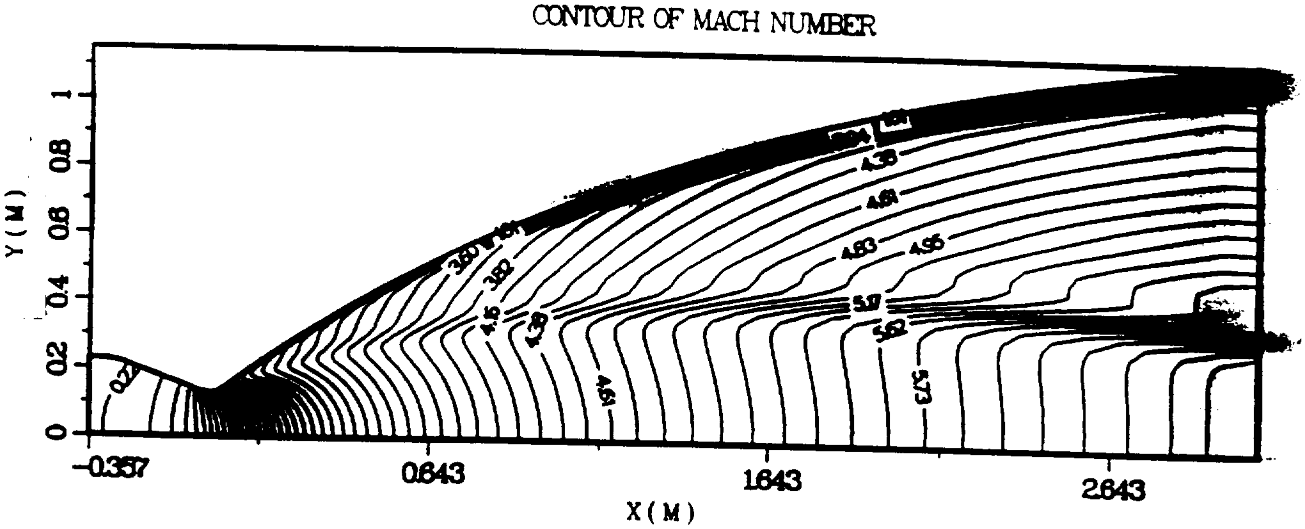
SAMPLE INPUT FILE

"chem.in"

```
specis # ns, elements # ne, frosen # nf
 6             2              2
chemical element
O H
chemical species
O2 H2 H2O HO O H ;
elements conservation b0
3.,8.
frosen ?
F
```

SAMPLE CASE 4

```
CONTROL    COMPRES NCRT 3  OMGM 1  NCGM 50
           OMGD 1.0 PHI -1.0 OMGPHI 0.00  OMGT 0.50 OMGF 1.00
           ERRCG 1.0E-2 ERRM 5.0E-6 IMON    81 JMON 10 MONU
;RESTART
GRID  NX  81  NY  41  AXISYM  READXY
BOUND
    IST   1  IEND   1  JST  1  JEND  41  INLET
    YH2 0.143 YO2 0.857 YHO 0. YH2O 0.
    IST   1  IEND  81  JST 41  JEND  41  WALL U 0.0 V 0.0
    IST   1  IEND  81  JST  1  JEND   1  SYMMETRY
    IST 81 IEND    81  JST  1  JEND  41  OUTLET
TURBULENT   TKIN 3.072 TEIN 10000.
REACTION
     FINIR LCHEM 1  INISPE
      YH2 0.143 YO2 0.857 YHO 0. YH2O 0.
PROPERTY  VISCOS  9.05E-5
        PSTAG 20240946.90  TSTAG 3637.   GAMMA 1.2  GMW 10.18
SOLV  U    V  P  TEMP   TK  TE
RUN   DT 1.E-5 DTMIN 1.0E-5 DTMAX 1.E-2   CFLN 4.00 NSTEP 100
        NPR1 1   NPR2 100   NEX 18 ; LPFAC ; LPGEO
ENDJOB
```



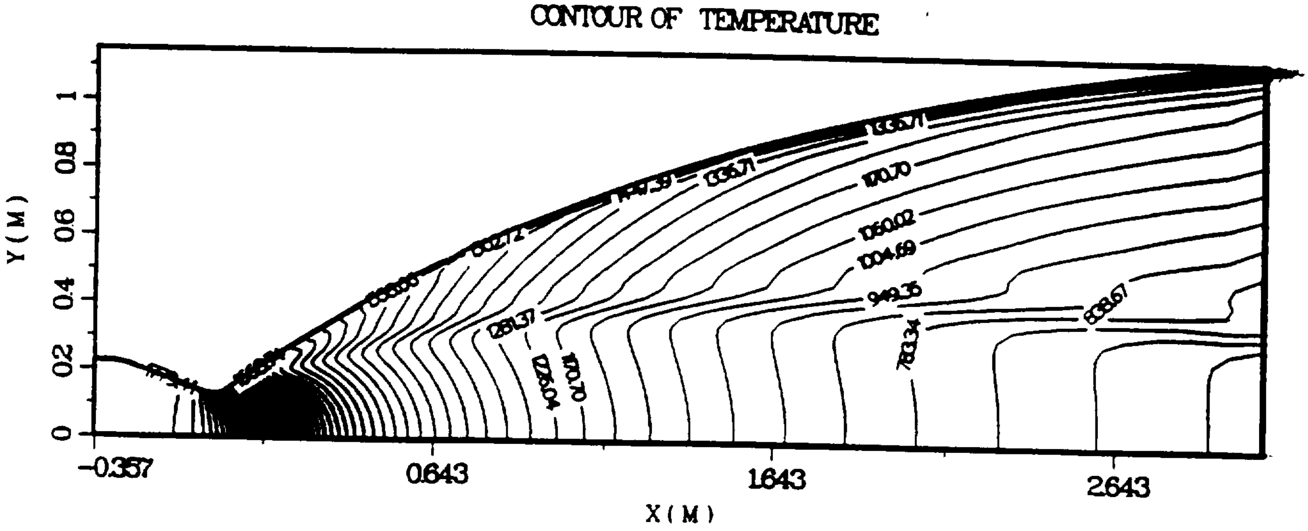
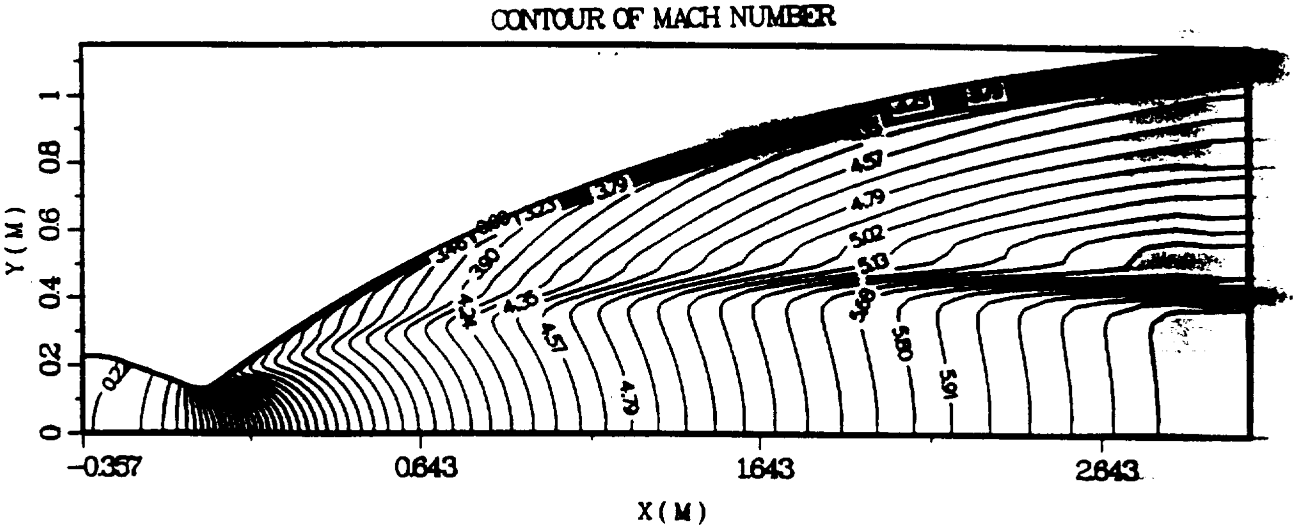


Figure A.4 Sample SSME Nozzle Flow Inputs and Results --- Turbulent, 2-Step Kinetics.
ISP = 442.01 sec.

SAMPLE CASE 5

```
CONTROL   COMPRES NCRT 3  OMGM 1  NCGM 50
          OMGD 1.0 PHI -1.0 OMGPHI 0.00  OMGT 0.50 OMGF 1.00
          ERRCG 1.0E-2 ERRM 1.0E-5 IMON    81 JMON 10 MONU
;RESTART
GRID  NX  81  NY  41  AXISYM  READXY
BOUND
    IST   1  IEND   1  JST  1  JEND  41  INLET
    YH2 0.143 YO2 0.857 YO 0. YH 0. YHO 0. YH2O 0.
    IST   1  IEND  81  JST 41  JEND  41   WALL U 0. V 0.0 ;
    IST   1  IEND  81  JST  1  JEND   1  SYMMETRY
    IST 81 IEND    81  JST  1  JEND  41  OUTLET
TURBULENT   TKIN 3.072 TEIN 10000.
REACTION     -
 FINIR LCHEM 2   INISPE
 YH2 0.143 YO2 0.857 YO 0. YH 0. YHO 0.0 YH2O 0.0
PROPERTY  VISCOS  9.05E-5
        PSTAG 20240946.90  TSTAG 3637.   GAMMA 1.2  GMW 10.18
SOLV  U    V  P  TEMP    TK  TE
RUN   DT 1.E-5 DTMIN 1.0E-5 DTMAX 1.E-2   CFLN 4.00 NSTEP 100
       NPR1 1   NPR2 100   NEX 18 ; LPFAC ; LPGEO
ENDJOB
```



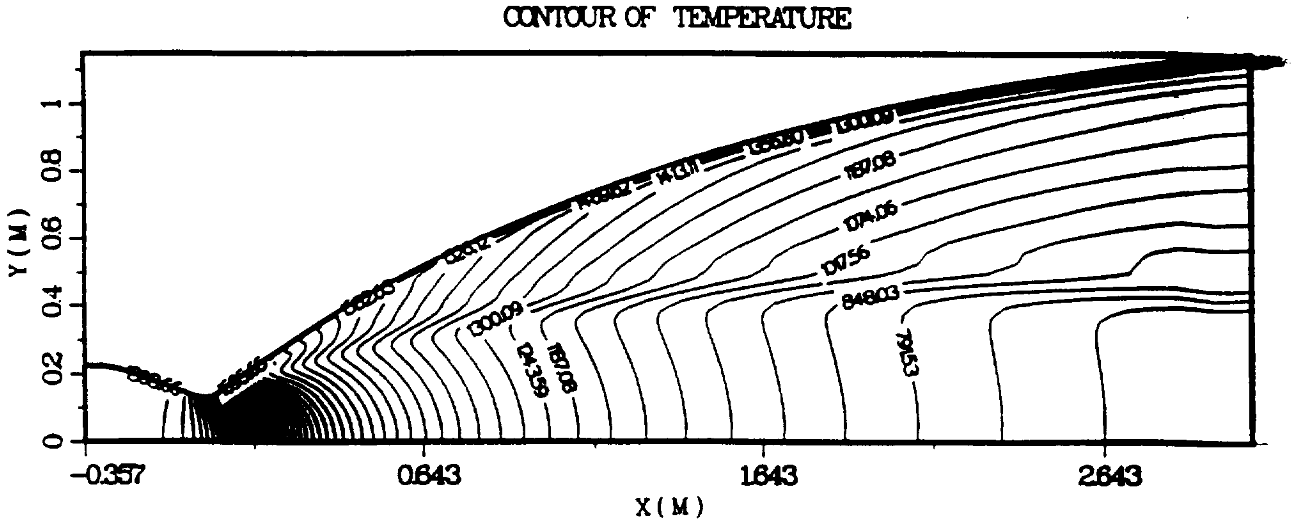


Figure A.5 Sample SSME Nozzle Flow Inputs and Results --- Turbulent, 8-Step Kinetics.
ISP = 452.78 sec.

ORGANIZATION			UAH	ARGONNE	[illegible]	CHAM	CRE
RESPONSIBLE PERSON			C.P. Chen	G. Berry	[illegible]	[illegible]	Z.
CODE NAME			MAST	[illegible]	[illegible]	[illegible]	FL
DIMENSIONS			2D Axi	2D	[illegible]	[illegible]	2
COORDINATES	CARTESIAN/BODY FIT		Bdy Fit		[illegible]	[illegible]	Car(
TIME PROBLEMS	STEADY/UNSTEADY		Stdy/Unstdy	Unsteady	[illegible]	[illegible]	St
TYPE OF FLOW	INVISCID/VISCOUS	[illegible]	Invis/Vis (L/T)	Vis(T)	[illegible]	[illegible]	Invi
	INCOMP/COMPRESSIBLE	[illegible]	Inc/Comp	[illegible]	[illegible]	[illegible]	[illegible]
EQUATIONS	CONSERV/NONCONSERV		Consv		[illegible]	Conserv	C
	MOMENTUM	[illegible]	N-S Eq		[illegible]	N-S	
	ENERGY		Yes	Yes	[illegible]	Yes	
	SPECIES	[illegible]	Yes	Yes(T)	[illegible]	[illegible]	Ye
	MULTI-PHASE TRACKING	[illegible]	E-L Stat	E-E	[illegible]	[illegible]	
	EQUATIONS OF STATE	[illegible]	I G/R G	I.G,R.G[illegible]	[illegible]	I.G,R.G	
	TRANSPORT PROPERTIES	[illegible]	Eqn	Eqn	[illegible]	[illegible]	T
PHYSICAL PROCESS	TURBULENCE MODELING	[illegible]			[illegible]	[illegible]	
		[illegible]	Ke, ASM	K-ε	[illegible]	K-E	
		[illegible]	Multi-scale		[illegible]		A
	ATOMIZATION MODEL	[illegible]	Calculated	Assumed	[illegible]		
	VAPORIZATION MODEL	[illegible]	Drop (S/H) (Sb)	Yes(T)	[illegible]	[illegible]	Dr
	CHEMISTRY MODEL	[illegible]	EQ.FR	FR	[illegible]	EQ,FR	
	RADIATION MODEL	[illegible]			[illegible]	[illegible]	Di
ROCKET PROPELLANT	TYPE	[illegible]	HC, H_2/O_2		[illegible]	[illegible]	
	PHASES(FUEL/OX)	[illegible]	Two (G/S, G/L)	Two[illegible]	[illegible]	Two[illegible]	S
DISCRETIZATION	FDM/FVM/FEM/SPECTRAL/ETC.		FVM	FVM	[illegible]	FVM	
NUMERICAL SCHEME	VARIABLES BASED	[illegible]	p-V	P-V	[illegible]	P-V	
	DIFF. ACCURACY(TIME/SPATIAL)	[illegible]	T(2nd) S/(2nd)		[illegible]	[illegible]	T(
	MULTI STEP/FACTORIZATION	[illegible]	MS(Yes)/FACT(No)		[illegible]	[illegible]	[illegible]
	EXPLICIT	[illegible]			[illegible]	No	
	IMPLICIT	[illegible]		Fully implicit	[illegible]	Implicit	
	OTHERS: SPECIFY	[illegible]			[illegible]		
		[illegible]			[illegible]		
		[illegible]	PISOC	SIMPLER	[illegible]	[illegible]	
	MULTIGRID CAPABILITY	Yes/No	Yes		[illegible]	No	
MATRIX SOLVER	DIRECT METHOD	[illegible]		TDMA	[illegible]	[illegible]	
	ITERATIVE METHOD	[illegible]		Line Iter	[illegible]		
		[illegible]	CCM(CCS)		[illegible]		
GRID	SEPARATE (NAME)	[illegible]			[illegible]	[illegible]	
	INTERNAL	[illegible]	Northo	Ortho	[illegible]	[illegible]	S
		[illegible]	Struct	Adapt	[illegible]	[illegible]	
	TECHNIC USED	[illegible]	Alge		[illegible]	[illegible]	
INLET/WALL	INLET CONDITIONS	[illegible]	Const, Tim Var		[illegible]	[illegible]	C
BOUNDARY COND.	WALL BOUNDARY	[illegible]	Nslp,MOV	Nslp	[illegible]	[illegible]	Sl
		[illegible]	Const.Tim Var	Adiab	[illegible]	[illegible]	A
		[illegible]	Surf SMOOTH	[illegible]	[illegible]	[illegible]	
		[illegible]	AdH,Bounce		[illegible]		Ad
PROGRAM CAPABILITY	INCOMPRESSIBLE FLOWS	[illegible]	10-10E7		[illegible]	All	
(Experienced)	COMPRESSIBLE FLOWS	[illegible]	0-20		[illegible]	All	
	INTERNAL FLOW	[illegible]	Separ, Recir		[illegible]	All	R
		[illegible]	Combustor, Nozz	Spray combustion	[illegible]	[illegible]	Co
	INJECTION	[illegible]	Coax		[illegible]		C
		[illegible]	Vap.Mix		[illegible]		
	PERFORMANCE PREDICTION	[illegible]	All	Thrust,Dyn pump	[illegible]	All	
	MISC.	[illegible]			[illegible]	[illegible]	
CODE VALIDATION	UNIQUE CASES	[illegible]	25		[illegible]	~200	
	CASES PUBLISHED	[illegible]	N10		[illegible]	~25	
	THRUST CHMBR RELATED	[illegible]	1(SSME)		[illegible]	~3	
PRE/POST PROCESSOR	DEVELOPED	[illegible]	Inhouse	N/A	[illegible]	Inhouse,Others	
	OPERATION	[illegible]	Batch		[illegible]	Interact	I
	GRAPHICS	[illegible]	DISPL	N/A	[illegible]	[illegible]	
PROGRAM	DOCUMENTATION	[illegible]	Eng.Prog	TBD	[illegible]	[illegible]	E
	AVAILABILITY	[illegible]	Pub domain	Pub(Argonne)	[illegible]	Sale	
COMPUTER SYSTEM	MAIN FRAME	[illegible]	Cray	Cray,IBM	[illegible]	[illegible]	
	MINI COMPUTER	[illegible]		Vax	[illegible]	Convex,Alliant,Vax	Co
	WORKSTATION	[illegible]			[illegible]	[illegible]	S
MISCELLANEOUS	CODE EXPERIENCE	No. of years Used	Developing (4 years)	4 Yrs	[illegible]	9 Yrs	
	CODE ORIGIN	[illegible]	Inhouse	Inhouse	[illegible]	Inhouse	
	VECTORIZATION	[illegible]	Partially		[illegible]	Yes	

Survey Date : October, 1990 (Except **)

None

Table A.1. Updated features in the MAST code. (Dec., 1992)