Computation of Multi-Dimensional Viscous Supersonic Jet Flow

Y. N. Kim, R. C. Buggeln, and H. McDonald

CONTRACT NAS3-22759
OCTOBER 1986
Computation of Multi-Dimensional Viscous Supersonic Jet Flow

Y. N. Kim, R. C. Buggeln, and H. McDonald

Scientific Research Associates, Inc.
Glastonbury, Connecticut

Prepared for
Lewis Research Center
under Contract NAS3-22759
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>LIST OF SYMBOLS</td>
<td>10</td>
</tr>
<tr>
<td>ANALYSIS</td>
<td>13</td>
</tr>
<tr>
<td>SOLUTION OF THE GOVERNING EQUATIONS</td>
<td>21</td>
</tr>
<tr>
<td>TEST CASES</td>
<td>24</td>
</tr>
<tr>
<td>DISCUSSION AND CONCLUSIONS</td>
<td>30</td>
</tr>
<tr>
<td>USER'S MANUAL</td>
<td>31</td>
</tr>
<tr>
<td>Flow Diagram</td>
<td>31</td>
</tr>
<tr>
<td>PEPSIN Subroutines</td>
<td>36</td>
</tr>
<tr>
<td>Logical File Units Utilized by PEPSIN Computer Code</td>
<td>40</td>
</tr>
<tr>
<td>PEPSIN Input</td>
<td>42</td>
</tr>
<tr>
<td>Namelist Input Description</td>
<td>45</td>
</tr>
<tr>
<td>Error Conditions in the PEPSIN Computer Code</td>
<td>59</td>
</tr>
<tr>
<td>PEPSIN FORTRAN Variables</td>
<td>63</td>
</tr>
<tr>
<td>Sample Input and Sample Output</td>
<td>85</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>87</td>
</tr>
<tr>
<td>FIGURES</td>
<td>90</td>
</tr>
</tbody>
</table>

PRECEDING PAGE BLANK NOT FILMED
SUMMARY

A new method has been developed for two and three-dimensional computations of viscous supersonic flows with embedded subsonic regions adjacent to solid boundaries. The approach employs a reduced form of the Navier-Stokes equations which allows solution as an initial-boundary value problem in space, using an efficient noniterative forward marching algorithm. Numerical instability associated with forward marching algorithms for flows with embedded subsonic regions is avoided by approximation of the reduced form of the Navier-Stokes equations in the subsonic regions of the boundary layers. Supersonic and subsonic portions of the flow field are simultaneous calculated by a consistently split linearized block implicit computational algorithm. The results of computations for a series of test cases relevant to internal supersonic flow is presented and compared with data. Comparison between data and computation are in general excellent thus indicating that the computational technique has great promise as a tool for calculating supersonic flow with embedded subsonic regions. Finally, a User's Manual is presented for the computer code used to perform the calculations.
INTRODUCTION

The interaction of a supersonic jet with an external stream has received considerable attention due to its critical role in determining both the thermodynamic and acoustic signatures from a vehicle producing a supersonic jet. The physics of supersonic jet flow is characterized by the formation of shock and expansion waves, the development of viscous mixing layers and the interaction of these phenomena (Figs. 1 and 2). As a consequence, a complicated multiple shock cell flow structure comes into existence in the jet. The detailed flow structure of a given jet including a shape of jet interface is determined primarily by the conditions of an ambient flow. At the jet interface, these two different flows form a turbulent mixing layer which strongly interacts with both shock and expansion waves. In the presence of interactions with waves, the turbulence structure in the mixing layer may be strongly affected. Both compressibility effects and pressure gradients in the jet can also influence the turbulence structure in the mixing layer. Under certain operational conditions, Mach disc(s) may occur and complicate the turbulent jet flow structure (Fig. 2). The mixing process that is present downstream of the disc has wake-like characteristics, and the entrainment process induces large streamline displacement which can appreciably alter pressure levels from those occurring in the inviscid limit. Because of the strong viscous/inviscid interactions as discussed above, inviscid theoretical methods have been found to be inadequate for predicting jet/external flow interactions. Therefore, very costly experimental testing has been the primary means of determining the complex flow patterns associated with these flows. However, in recent years, advances in computational fluid dynamics have demonstrated that the dependence on the experimental analysis of such a complex flow can be reduced by utilizing computational analysis. Unlike the experimental testing, computational analysis is not necessarily restricted by Reynolds number and other flow or testing conditions. Furthermore, experimental costs are steadily increasing, whereas the computational costs have been and are expected to decrease. Thus, the goal of the current work was to develop an efficient and reliable numerical approach for the prediction of the detailed flow structure of a supersonic jet exhausted into the external flow.
Early computational viscous solutions of the nozzle flow field region consisted of viscous-inviscid interaction techniques called patching methods (for example, Refs. 1-3). The techniques consist of dividing the various viscous and inviscid regions of the flow according to the physical nature and mathematical behavior of the equations applicable to each region. Each of these regions was then analyzed independently and was iteratively coupled using appropriate matching conditions consistent with weak-interaction theory. Although these solutions gave reasonable results for specific data sets, the required matching limited the usefulness of this method as a predictive technique. It is quite natural to consider solving the full time-dependent, compressible Navier-Stokes equations as an alternative approach for such a complex flow (for example, Refs. 4-5). However, the widely disparate length scales and flow characteristics in the various regions involved could lead to perhaps unacceptable computer time and storage requirements in achieving the results of adequate resolution, especially in three dimensions. In this study, it was decided that such a procedure should be used only if no suitable alternative exists. A compromise approach would possess the general three-dimensional, viscous nature of the Navier-Stokes equations, but would take advantage of realistic physical approximations to limit the computer running time and storage requirements associated with the solution of the complete three-dimensional Navier-Stokes equations.

During the past two decades much effort has been expended in developing such numerical procedures which can be used as an alternative to solving the full Navier-Stokes equations for certain classes of problems. These procedures treat a reduced form of the steady state Navier-Stokes equations, often referred to as the parabolized Navier-Stokes equations, as an initial boundary value problem that can be solved by spatial forward marching. The ability to forward march the governing equations from an initial streamwise location to some desired downstream location rather than perform a simultaneous solution of the governing equations at all streamwise locations, as is required for the solution of the full Navier-Stokes equations, results in a considerable reduction of computational time. Although the amount of reduction will depend on the problem considered, the efficiency of the solution procedures and numerous other variables, this reduction has been the primary motivation for the development of marching procedures.
To devise a set of governing equations suitable for the spatial forward marching of supersonic flows, three steps must be taken. First, a nominally primary flow direction must be identified. Second, a coordinate system must be constructed with one of its coordinate directions aligned with the primary flow direction. Third, all diffusion in the primary flow direction must be neglected. These steps when applied to the steady state Navier-Stokes equations produce a set of governing equations which is well posed for the spatial forward marching of supersonic flows (e.g. Ref. 6). The introduction of no-slip surfaces into a supersonic flow usually results in the formation of embedded subsonic regions adjacent to these surfaces. Also, under certain operating conditions of the vehicle supersonic flow coexists with the subsonic flow as can be found in the case of supersonic jet exhaust into the subsonic ambient flow. When the set of reduced equations is forward marched with embedded or coflowing subsonic regions, the governing equations are not well posed and hence the solution procedure may become unstable. This instability, which is often referred to as the branching phenomenon, has been the subject of much research (e.g. Refs. 7-8) and the techniques used to suppress this instability are a convenient way to differentiate between procedures for solving the reduced form of the Navier-Stokes equations for supersonic flow with subsonic regions. In one of the earliest works in this area Garvine (Ref. 7) demonstrated (for a model problem) the existence of exponentially growing (divergent) terms in the solution of the reduced form of the Navier-Stokes equations when applied to a viscous (inviscid supersonic flow interacting with a viscous boundary layer) interaction problem. The author concluded that for this problem the reduced form of the Navier-Stokes equations was improperly set as an initial value problem, because the interaction dynamics contained upstream "elliptic" influence. In the model problem if the upstream conditions are not precisely set to cause the divergent terms to be multiplied by zero, the exponentially growing terms will cause the streamwise pressure gradient terms to grow exponentially large resulting in unrestrained acceleration or deceleration of the flow. In general it is not possible to pick the upstream conditions to negate the exponentially growing modes, hence several investigators have attempted to suppress the unstable (or branching) behavior by further modification of the reduced form of the Navier-Stokes equations for supersonic flows with embedded subsonic regions.
Much of the early work on the solution of the reduced form of the Navier-Stokes equations is based on the work of Rudman and Rubin (Ref. 8). Rudman and Rubin solved the equations for the hypersonic flow over slender bodies with sharp leading edges. Based on an order of magnitude analysis they demonstrated that for this class of problems the streamwise pressure gradient term was negligible when compared with the inertia and viscous terms of the streamwise momentum equation. Neglecting the streamwise pressure gradient term in this equation prevented the branching behavior in their calculation. Although this approach does yield a set of equations that is well posed for spatial forward marching, its assumption of negligible streamwise pressure gradient limits the cases which can be considered. In a later work Lubard and Helliwell (Ref. 9) proposed a method for preventing branching that involved explicit spatially lagged evaluation of the streamwise pressure gradient term. The above authors found that in addition to the frequently encountered problem of instability associated with exceeding some marching direction step size, a further instability was encountered when the step size was reduced below some limit. By examining the eigenvalues of a model set of equations (Ref. 12) they were able to develop a criterion for this minimum step size. Numerical experimentation with their computer code demonstrated reasonable correlation with their criterion. Several test cases using this method have been successfully run (mainly for cone flow cases) by the authors of Ref. 9 and others (Refs. 13-14) and in these cases evidently the restriction on the minimum marching step size was not a problem in allowing sufficiently accurate results to be obtained. However, the restriction on minimum marching step size is in principle not a desirable feature. Since it does prevent arbitrary mesh refinement, and thereby the assurance that an accurate unique solution has been obtained. In at least one case (Ref. 9) this minimum step size restriction prevented the successful running of a case. In a later technique developed by Vigneron, Rakich and Tannehill (Ref. 10) a variant of the technique of Lubard and Helliwell was used to prevent branching. In this particular variant the streamwise pressure gradient term was evaluated by an implicit backward difference in the supersonic portion of the flow. However, in the subsonic region only that portion of the streamwise pressure gradient term that could be included without causing branching was evaluated implicitly. The stability analysis through the investigation of the
eigenvalues of a model set of equations also produces a restriction on the minimum allowable step size. When that portion of the subsonic pressure gradient excluded from the implicit evaluation is evaluated explicitly by a lagged technique similar to Lubard and Helliwell, the investigators noted that the scheme became unstable. Thus, in order to achieve stability this technique neglected the explicit portion of the streamwise pressure gradient term in the subsonic region. Schiff and Steger (Ref. 11) treat the subsonic streamwise pressure gradient term by either a first- or second-order streamwise extrapolation technique in the subsonic regions. The first order technique is equivalent to setting the streamwise pressure gradient term equal to zero in the subsonic region while the second order technique is equivalent to the explicit evaluation of the streamwise pressure term (as was done by Lubard and Helliwell). As with the two previously discussed techniques, these authors also report a restriction on the minimum marching step size that they may take and still retain a stable calculation. On the other hand, Rubin and Lin (Ref. 15) have developed a global relaxation procedure for solving the reduced form of the Navier-Stokes equations. This technique was primarily developed for application to cases where upstream influence is strong. To obtain the upstream influence with the reduced form of the Navier-Stokes equations requires a global iteration or relaxation procedure. The above authors do this by approximating the streamwise pressure gradient term by a forward difference. When marching the solution from the i\textsuperscript{th} to the i+1\textsuperscript{st} station the pressure gradient term is evaluated in terms of the pressure at the i+1\textsuperscript{st} (current) and i+2\textsuperscript{nd} (unknown) station (i+1\textsuperscript{st} station is the implicit station; all other streamwise derivatives are backward differenced between i+1 and i). Initially the pressure at the unknown i+2\textsuperscript{nd} station is guessed; during subsequent global iterations the previously calculated value is used. Global iteration of the governing equations is continued until the solution converges. Rubin and Lin report that convergence is typically obtained in five to ten iterations for cases with small streamwise pressure gradients (cases run to date have been limited to flow over cones). The authors also report that there is no minimum marching step size requirement with their approach.

The supersonic jet exhausted from a nozzle is essentially turbulent and has associated with it large streamwise pressure gradients, e.g., turbulent mixing layer interacting with multiple shock wave-cells. It is expected that one would desire to take a small streamwise marching step to accurately
resolve the phenomenon, for example, the near-field plume-mixing interactions.

As reviewed above, the methods of Refs. 8-11 all make an attempt to consider the effects of streamwise pressure gradient in the embedded subsonic regions. However, they give only an approximate treatment to this possibly dominant term, and all of those methods have a minimum marching step size limitation which, in the cases of interest in this study, may not either allow for an accurate or in some cases even a minimally acceptable solution. Although there is no minimum marching step size requirement with the global relaxation procedure developed by Rubin and Lin (Ref. 15), the computation time would increase considerably in more complex flow situations. If the number of iterations were to increase much beyond the five to ten iterations it took to converge a small streamwise pressure gradient case, the computation time would then become comparable to that required for a solution of the full Navier-Stokes equations.

Hence, a noniterative approach is sought with a consequent reduction in computer cost relative to either the global iteration approach to solving the reduced Navier-Stokes equations or solution of the full Navier-Stokes equation. Further as a prerequisite it is required that there exist no numerical limitation on the minimum marching step and it is desired to keep to a minimum any approximation to the streamwise pressure gradient term.

Such an approach which meets these requirements has been developed by the present authors as reported elsewhere (Ref. 16). A detailed explanation of the techniques will be presented in the next analysis section. Unlike other noniterative methods, the streamwise pressure gradient term is maintained and evaluated by an implicit backward difference in the supersonic as well as the subsonic region. However, approximations were made on both the normal (to the wall usually) momentum and continuity equation in the subsonic region(s) to suppress the branching behavior of solution. In the (thin) subsonic flow region, the normal momentum equation is approximated by the conventional boundary layer form of that equation, and the continuity equation is approximated by integrating the continuity equation from the wall to an arbitrary point in the subsonic portion of the boundary layer. This approximation of continuity equation is a major difference between the
present technique and other noniterative methods. As discussed in Ref. 16, numerical experimentation with various cases indicates no restriction on the minimum step size. Thus, an accurate resolution of the flow fields has been successfully obtained without being limited by the requirement of minimum step size. Since no approximations were made on the streamwise pressure gradient term, the predictions for the flow field under the condition of non-negligible streamwise pressure gradient agreed very well with the data as discussed in Ref. 16.

Thus, application of this approach to the analysis of steady three-dimensional supersonic jet interacting with an ambient flow was a major concern in this study. The flow field calculation by the existing computer program, called PEPSIS, was limited to a computational domain without any embedded solid objects except for the wall boundary. Thus, the first task of this study was to extend a capability of PEPSIS so that the flow field could be analyzed in a computational domain with one or more embedded solid bodies such as nozzle walls. The extended PEPSIS computer program, which is called PEPSIN, is the same as PEPSIS as far as its basic assumptions and approximations are concerned. One of the basic assumptions for PEPSIS approach was the thinness of subsonic layer within a boundary layer. As a result of this assumption, approximations were made on the governing equations within this thin, subsonic layer based on the fundamental properties of a boundary layer. However, analysis of supersonic jet interactions under certain low conditions is expected to encounter subsonic regions of a different nature. One of such subsonic regions may be present when a Mach disc appears in the jet. Another type of subsonic region may be encountered when a supersonic jet exhausts into an ambient subsonic stream. First of all, these two kinds of subsonic regions may not be necessarily thin. Furthermore, since these subsonic regions are not located in the vicinity of a wall boundary, approximations based on boundary layer properties become questionable. Therefore, it was decided to exclude such subsonic regions from consideration in the present study. Those subsonic regions require further separate investigations before they can be incorporated in the PEPSIN approach. Thus, as will be discussed later, test cases considered in the present case were limited to the analysis of a supersonic jet interacting with supersonic external stream.
Two other investigators have utilized forward marching techniques to solve the jet/external flow interactions. First, Dash and co-workers (Refs. 17 and 18) developed an explicit spatial marching method to predict both the inviscid plume structure and the viscous mixing layer. Their approach employs a multiple-domain solution algorithm to solve a reduced form of the Navier-Stokes equations. Their approach has been primarily applied to the two-dimensional axisymmetric cases, although they presented some results for nonaxisymmetric three-dimensional cases (rectangular cross sections of various aspect ratios) in a recent paper (Ref. 19). However, the marching step size in their procedure is limited by numerical stability conditions of explicit schemes. On the other hand, Vatsa and his co-workers (Ref. 18) developed a method to solve the problem of a slightly underexpanded supersonic jet in a subsonic co-flowing mainstream. First, they solved the inviscid problem which was formulated in terms of linearized perturbation potentials to estimate the streamline curvature. Then they obtained governing equations of parabolic type through the use of an approximate inviscid flow intrinsic coordinate system. Their solution based on a marching technique produces the class of weak interactions wherein the viscous interaction has a formally second-order effect. In this approach, prediction of streamline curvature, in general, may become as difficult as the viscous problem if the difference between jet and external flow conditions become large or if the problem is three-dimensional. A difficulty may arise also if the viscous effects become significant and have a strong influence on the secondary flow.

The present study is concerned with the application of an implicit forward marching procedure for the reduced form of the Navier-Stokes equations (Ref. 16) to the three-dimensional analysis of supersonic jet flow interacting with a co-flowing stream. The approach is noniterative and is not limited by a minimum step size criterion. Furthermore, being implicit the maximum allowable step size is expected to exceed that of explicit methods by a significant amount. The remainder of this report will consist of (1) a discussion of the analysis used in the study, (2) a discussion of the solution of the governing equations, (3) the results of a series of test cases run to demonstrate the applicability of the analysis and to exercise and to validate the resulting computer code and (4) a User's Manual for the computer code, termed PEPSIN.
LIST OF SYMBOLS

A
Square Matrix of coefficients

C_H
Nondimensional heat transfer coefficient

C_p
Specific heat

D
Column vector whose elements are spatial differential operators

D:D
Second invariant of the mean flow rate of deformation tensor

H
Column vector associated with marching direction terms

L
Linear differential operator

M
Mach number

P
Static pressure

Pr
Prandtl number

Re
Reynolds number

S
Source term

T
Static temperature

U
Streamwise velocity component

V
Transverse velocity component

W
Spanwise velocity component

X
Distance from leading edge

\dot{V}
Velocity

\Delta
Damping coefficient

h
Metric coefficient, enthalpy

l_m
Mixing length

\hat{n}
Unit vector normal
LIST OF SYMBOLS

q Heat transfer
r Radius
w Velocity component
x Coordinate direction
y Distance from a surface
$y^*$ Nondimensional distance from a surface

Greek Symbols

$\gamma$ Ratio of specific heats
$\delta_b$ Boundary layer thickness
$\kappa$ von Karman constant
$\mu$ Viscosity
$\rho$ Density
$\tau$ Viscous stress tensor
$\nabla$ Nabla operator
$\Delta x$ Marching direction step size
Subscripts

i  
ith direction

j  
jth direction, jet flow condition

l  
Laminar

n  
Normal direction

w  
Wall

E  
External stream condition

J  
Jet flow condition

R  
Reduced

S  
Sonic line

T  
Turbulent, tangential direction

0  
Stagnation condition

1  
Streamwise direction

2  
Cross plane direction

3  
Cross plane direction

∞  
Free stream condition

Superscripts

i  
ith streamwise station

T  
Transpose
II. ANALYSIS

Governing Equations

The steady state fluid dynamic conservation laws of mass, momentum and energy respectively can be written in nondimensional vector form as

\[ \nabla \cdot \rho \vec{V} = 0 \]  
(1)

\[ \nabla \cdot (\rho \vec{V} \vec{V}) + \nabla P - \frac{\nabla \cdot \tau}{Re} = 0 \]  
(2)

and

\[ \nabla \cdot (\rho h_o \vec{V}) - \nabla \left[ \frac{C_p}{Re} \left( \frac{\mu_t}{Pr_t} + \frac{\mu_T}{Pr_T} \right) \nabla T \right] - \frac{\nabla \cdot (\tau \cdot \vec{V})}{Re} = 0 \]  
(3)

This form of the governing equations, often referred to as the full Navier-Stokes equations, requires several auxiliary relations and models before these equations can be solved. In this study the stagnation enthalpy, \( h_o \), is related to the static temperature, \( T \), and the velocity \( \vec{V} \), through the relationship (assuming a calorically perfect gas)

\[ h_o = C_p T + \frac{\vec{V} \cdot \vec{V}}{2} \]  
(4)

while the temperature, \( T \), pressure \( P \), and density, \( \rho \), are related by means of the calorically perfect gas equation of state

\[ P = \frac{\gamma - 1}{\gamma} C_p \rho T \]  
(5)

The stress tensor, \( \tau \), is represented by the relationship

\[ \tau = \mu \left( \nabla \vec{V} + \nabla \vec{V}^T \right) - \frac{2}{3} \mu \nabla \cdot \vec{V} \]  
(6)

where the superscript \( T \) refers to the transpose of the tensor. The components of the velocity vector, \( \vec{V} \), are interpreted as the mass weighted mean velocity components and \( \rho, P \) and \( T \) are the ensemble-averaged
density, pressure and temperature (Ref. 21). Hence, these equations can be applied to both laminar and turbulent flows if the effective viscosity, \( \mu \), is interpreted as the sum of the laminar, \( \mu_L \), and turbulent, \( \mu_T \), viscosities, i.e.,

\[
\mu = \mu_L + \mu_T
\]  

(7)

It is assumed that the laminar viscosity can be computed from Sutherland's law and that the laminar and turbulent Prandtl numbers, \( Pr_L \) and \( Pr_T \) are constant. For this study an algebraic mixing length turbulence model of the form

\[
\mu_T = Re \rho l_m \sqrt{D:D}
\]  

(8)

was used where \( l_m \) is the algebraic mixing length and \( D:D \) is the second invariant of the mean flow rate of deformation tensor (Ref. 22). In this study the mixing length of McDonald and Camarata (Ref. 23) was used.

\[
l_m = 0.09 \delta_b \text{tanh} \left[ \kappa \bar{y} / (0.09 \delta_b) \right] \mathcal{D}
\]  

(9)

where \( \delta_b \) is the local boundary layer thickness, \( \kappa \) is the von Karman constant, \( \bar{y} \) is the distance to the nearest wall, and \( \mathcal{D} \) is the sublayer damping term of van Driest (Ref. 24).

To obtain what is often referred to as the reduced form or the 'parabolized' form of the Navier-Stokes equations involves approximation of the diffusion terms (both stress and Fourier heat conduction) of Eqs. (2), (3) and (6). This approximation neglects all derivatives of the stress tensor and the Fourier heat conduction terms in a selected 'marching' or 'streamwise' direction. In addition all streamwise derivatives of the velocity components of the stress tensor are neglected. Hence, the general reduced form of Eqs. (2) and (3) can be recast as

\[
\nabla \cdot (\rho \nabla \vec{V}) + \nabla P - \frac{(\nabla \cdot \tau)_R}{Re} = 0
\]  

(10)
where the subscript \( R \) refers to the approximated or reduced form of the noted term.

The reduced form of the Navier-Stokes equations, Eqs. (1), (10) and (11) is the starting point for the discussion of the governing equations to be used for this study. As discussed in detail in Ref. (16), this set of equations is not well posed for solution by spatial forward marching when applied to the class of problems considered in this study, i.e., supersonic flow with embedded subsonic layers. In this investigation, the same strategy as in Ref. (16) was adopted. The flow is divided into supersonic and subsonic flow regions and different sets of governing equations are utilized in each region. The reduced form of the Navier-Stokes equations in the supersonic region(s) of the flow and what can be considered to be a model set of equations in the subsonic region(s) are utilized. The model set of equations used in the subsonic region(s) is obtained by starting with the reduced form of the Navier-Stokes equation and making appropriate physical approximations in this region such that the coupled system of equations in both supersonic and subsonic flow are stable when solved as an initial value problem in space. There are several important features of the subsonic model set of governing equations. First, no approximation is made to either the streamwise or the tangential momenta equations, and second, the terms tangential and normal refer to some adjacent solid surface. Hence, the full effect of the pressure gradient terms will be felt in both the streamwise and tangential direction of the subsonic portion of the flow. The assumption needed to modify the normal momentum and continuity equations in the subsonic regions is the relatively unrestrictive condition that the subsonic layer is thin relative to the characteristic transverse dimension of the flow device. For the case of an impermeable wall this leads to the condition that within the viscous subsonic layer the normal velocity component is negligible. The importance of the specification of the normal velocity is that a mechanism can now be established to prevent the growing

\[
\nabla \cdot (\rho h_0 \overline{V}) - \left\{ \nabla \cdot \left[ \frac{C_p}{Re} \left( \frac{\mu_I}{Pr_I} + \frac{\mu_T}{Pr_T} \right) \nabla T \right] \right\}_R - \left[ \nabla \cdot \left( \frac{\tau \cdot \overline{V}}{Re} \right) \right]_R = 0
\]
mode caused by the interaction between the subsonic and supersonic layers, i.e., the branching phenomenon. This approximation can be obtained by integrating the continuity equation from the wall to an arbitrary point in the subsonic portion of the boundary layer \( X_s \). Thus,

\[
\left. h_1 n_T \rho w_n \right|_S = - \int_0^{X_s} \left[ \frac{\partial}{\partial x_1} (h_2 h_3 \rho w_n) + \frac{\partial}{\partial x_T} (h_1 n_T \rho w_n) \right] dx_1 + \left. h_1 n_T \rho w \right|_W
\] (12)

where the subscripts \( n \) and \( T \) refer to the crossflow direction normal and tangential to the walls, \( s \) refers to the evaluation at the arbitrary point in the subsonic region, and the subscript \( W \) refers to the evaluation at the wall. Restricting our attention to flows where the subsonic region is sufficiently thin allows the integral in Eq. (12) to be neglected and hence this equation can be approximated by

\[
\left. h_1 n_T \rho w_n \right|_S = \left. h_1 n_T \rho w_n \right|_W
\] (13)

As mentioned above, for the case of an impermeable wall Eq. (13) further reduces to

\[
\left. w_n \right|_S = 0
\] (14)

As a consequence of the thin subsonic layer assumption, the normal (to the wall) momentum equation then can be expressed as a balance only between the normal pressure gradient and the centrifugal (curvature) forces in the subsonic layer. For example, in general orthogonal coordinates, \( X_1, X_2, \) and \( X_3 \) with corresponding metric coefficients \( h_1, h_2, \) and \( h_3 \) and velocity components \( w_1, w_2, \) and \( w_3 \) this equation is expressed as

\[
\frac{\partial P}{\partial x_n} = (-1)^T n_T \frac{w_T}{h_T} \left( w_3 \frac{\partial h_3}{\partial x_2} - w_n \frac{\partial h_n}{\partial x_3} \right) + (-1)^n \frac{w_1}{h_1} \left( w_2 \frac{\partial h_1}{\partial x_n} - w_n \frac{\partial h_n}{\partial x_1} \right)
\] (15)

where \( X_n \) and \( X_T \) respectively refers to the appropriate cross-sectional direction normal and tangential to the wall (\( n \) and \( T \) have values of 2 or 3; direction \( l \) is the nominally streamwise direction). In summary, the new set
of governing equations consisting of the reduced form of the Navier-Stokes equations in the supersonic portion of the flow and the model set of equations in the subsonic regions of the flow (Eq. 13 instead of continuity equation and Eq. 15 for the normal momentum equation) has, on the bases of numerical experimentation (Ref. 16), been found to be well posed for solution by spatial forward marching for a wide range of practical problems.

**Initial and Boundary Conditions**

To uniquely define the problem of interest it is necessary to specify both initial and boundary conditions. For a spatial forward marching procedure, the initial conditions refer to the set of conditions that must be specified at the initial marching station. Boundary conditions must be set on the boundaries of the cross-sectional marching plane. For the calculation of the supersonic jet flow interacting with the coflowing stream, information must exist at an initial plane such that a reasonable approximation to a complete set of initial data can be constructed. In its most pure form this would be an initial plane where experimental or computational data was available such that all the initial conditions were known and consistent with the constitutive relations. In general, automatic generation of initial profiles is not straightforward for the present type of flow. However, if the flow is two-dimensional, a limited automation is possible. Usually a limited amount of information is available, where, for instance, freestream conditions with a boundary layer thickness and a skin friction coefficient on the nozzle surface might be known. In this case, a theoretical boundary layer profile of the pertinent variables (velocity components, temperature, pressure, etc.) can often be derived and matched with the flow outside the boundary layer. Analysis of the characteristics of the supersonic three-dimensional Euler equation shows that there are five characteristics entering the upstream boundary of the computational domain. It is argued that the reduced form of the Navier-Stokes equations used here would not require more boundary conditions than the Euler equations and this certainly appears reasonable at very high Reynolds numbers far from the solid walls. Thus, it is argued that five conditions must be set on the inflow boundary. In this study those conditions are chosen as the three velocity components, the pressure and the temperature. It is to be emphasized that the initial
conditions must in some sense be consistent with the governing equations in a hyperbolic system. In supersonic flow computations inconsistencies, perturbations, etc. can persist far downstream.

As used in this investigation the boundary conditions utilized on the boundaries of the cross-sectional plane can be divided into three categories: (1) wall conditions, (2) symmetry conditions and (3) external flow conditions. Analysis of the characteristics of the boundary layer equations shows that four conditions must be specified on walls. Arguing that in the limit as the present system approaches the wall it should approximate the boundary layer equations, the same four conditions are used here. For this study the no-slip conditions are used for the streamwise and tangential cross plane velocity components, i.e.,

\[ w_l = 0 \]  
\[ w_T = 0 \]

(16)  
(17)

where again the subscript \( l \) refers to the streamwise direction and the subscript \( T \) refers to the cross plane tangential velocity direction. For the cross plane normal velocity component either the normal velocity or the normal mass flux are specified, i.e.,

\[ w_n = w_W \]  
\[ (\rho w)_n = \rho_w w_W \]

(18)  
(19)

where the subscript \( W \) refers to the wall value (specified). The fourth condition (the thermal condition) used is to either specify an adiabatic wall
or to specify the wall temperature (a cold or hot wall). The condition can be specified respectively as

\[ \mathbf{n}_w \cdot \nabla T = 0 \]  \hspace{1cm} (20)

or

\[ T = T_w \]  \hspace{1cm} (21)

where in this case \( \mathbf{n}_w \) represents the unit vector normal to the wall. In addition, a fifth condition, not required by the characteristic analysis, is used to close the set of equations. The need for this fifth condition could be removed by the use of one-sided differencing or by applying one of the governing equations at the wall. In this study for convenience the second method was used and the boundary layer approximation to the normal momentum equation was applied at the wall. This can be expressed as

\[ \mathbf{n}_w \cdot \nabla P = 0 \]  \hspace{1cm} (22)

Studies have indicated that there is little difference between using this equation and the nonapproximated normal momentum equation.

The symmetry conditions are meant to be applied on a plane or axis of symmetry. The velocity conditions used require that the cross plane velocity component normal to the axis or plane of symmetry equals zero, i.e.,

\[ \mathbf{n}_s \cdot \mathbf{V} = 0 \]  \hspace{1cm} (23)

where \( \mathbf{n}_s \) is the unit vector normal to the axis or plane of symmetry and that the first derivatives of the remaining two velocity components equal zero. Two other conditions must be set on the axis or plane of symmetry. Usually the symmetry conditions on pressure and temperature are used, viz.

\[ \mathbf{n}_s \cdot \nabla T = 0 \]  \hspace{1cm} (24)
and

\[ \vec{n_s} \cdot \nabla T = 0 \]  \hspace{1cm} (25)

The final category of boundary conditions used in this investigation is the boundary conditions to be used on what can be considered to be external surfaces. Specifically these boundary conditions are meant to be used on the external boundary of the coflowing ambient stream. In this case a set of boundary conditions that will allow all disturbances which originate from within the computational domain to pass through the boundary without spurious reflection. The technique used in this investigation is based on the concept that in a simple wave region the flow properties remain constant along Mach lines. Thus the first derivatives of the flow variables in the direction of the Mach angle should be small and are here set equal to zero. The technique is termed Mach wave extrapolation and yields the boundary conditions

\[ \vec{n_m} \cdot \nabla V = 0 \]  \hspace{1cm} (26)

\[ \vec{n_m} \cdot \nabla P = 0 \]  \hspace{1cm} (27)

and

\[ \vec{n_m} \cdot \nabla T = 0 \]  \hspace{1cm} (28)

where \( \vec{n_m} \) is the unit vector in the direction of the local Mach angle. This technique requires computation of the Mach angle and has been successfully applied to a number of test cases both by the present authors and the authors of Ref. 25 and 26.
SOLUTION OF THE GOVERNING EQUATIONS

Numerical Methods

The governing equations in both the supersonic and the embedded subsonic portion of the flow are simultaneously solved by the consistently split linearized block implicit (LBI) technique described in detail in Refs. 27 and 28. This technique can be logically divided into three parts: (1) linearization of the governing equations (2) discretization of the resulting set of linearized equation by finite difference approximation of derivative terms and (3) simultaneous solution of the resultant set of linear coupled algebraic equations. Application of the LBI technique to a set of governing equations (and boundary conditions) that is well posed for forward marching is straightforward. The system of governing equations can be written in the following form:

\[ \frac{\partial H(\phi)}{\partial x} = D(\phi) + S(\phi) \]  

(29)

where \( \phi \) is the column vector of dependent variables \( (w_1, w_2, w_3, \rho, h_o) \), \( H \) and \( S \) are column vector algebraic functions of \( \phi \), and \( D \) is a column vector whose elements are the spatial differential operators.

When a solution at the \( i + 1 \)th station, at some distance \( \Delta x \) downstream, is desired after a solution is obtained at an arbitrary \( i \)th streamwise station, the solution procedure is based on the following implicit marching direction difference approximation of Eq. (29)

\[ \frac{(H_{i+1} - H_i)}{\Delta x} = (D^i + S_{i+1}) \]  

(30)

where, for example, \( H_{i+1} \) denotes \( H(\phi_{i+1}) \). A local spatial linearization (Taylor series expansion about \( \phi_i \)) of requisite formal accuracy is introduced, and this serves to define a linear differential operator \( L \) such that

\[ D_{i+1}^i = D^i + L^i (\phi_{i+1}^i - \phi_i^i) + O(\Delta x^2) \]  

(31)
Similarly
\[ H^{i+1} = H^i + \left( \frac{\partial H}{\partial \phi} \right)^i (\phi^{i+1} - \phi^i) + O(\Delta x^2) \]  
(32)
\[ S^{i+1} = S^i + \left( \frac{\partial S}{\partial \phi} \right)^i (\phi^{i+1} - \phi^i) + O(\Delta x^2) \]  
(33)

Eqs. (31-33) are inserted into Eq. (30) to obtain the following system which is linear in \( \phi^{i+1} \)

\[ (A - \Delta x L^i)(\phi^{i+1} - \phi^i) = \Delta x (D^i + S^i) \]  
(34)

and which is termed the linearized block implicit (LBI) scheme. Here A denotes a square matrix defined by

\[ A \equiv \left( \frac{\partial H}{\partial \phi} \right)^i - \Delta x \left( \frac{\partial S}{\partial \phi} \right)^i \]  
(35)

Eq. (35) is \( O(\Delta X) \) accuracy.

To obtain an efficient algorithm, the linearized system, Eq. (34) is split using ADI techniques. To obtain the split scheme, the multidimensional operator, \( L \), is rewritten as the sum of two 'one-dimensional' sub-operator \( L_i (i = 2, 3) \) each of which contains all terms having derivatives with respect to the \( i \)th-cross plane coordinate. The split form of Eq. (34) can be derived either as in Ref. 27 by following the procedure described by Douglas and Gunn (Ref. 29) in their generalization and unification of scalar ADI schemes, or using the approximate factorization as in Ref. 30. For the present system of equations, the split algorithm is given by

\[ (A - \Delta x L_1^i)(\phi^* - \phi^i) = \Delta x (D^i + S^i) \]  
(36)
\[ (A - \Delta x L_2^i)(\phi^{i+1} - \phi^i) = A(\phi^* - \phi^i) \]  
(37)

where \( \phi^* \) is the consistent intermediate solution (Ref. 28). If spatial derivatives in \( L_i \) and \( D \) are replaced by the difference formulae indicated previously, then each step in Eq. (36) and Eq. (37) can be solved by a block tridiagonal elimination.
Combining Eqs. (36) and (37) gives

\[(A - \Delta x L_i) A^{-1} (A - \Delta x L_2^i) (\phi^{i+1} - \phi^i) = \Delta x (D^i + S^i)\]

which approximates the unsplit scheme, Eq. (34) to \(O(\Delta x^2)\). Since the intermediate step is also a consistent approximation to Eq. (34), physical boundary conditions can be used for \(\phi^*\) (Refs. 28, 31).
TEST CASES

A primary objective of this study was to develop a computational procedure which is capable of analyzing a supersonic jet flow interacting with an external stream. Therefore, three three-dimensional and one axisymmetric test cases were considered to demonstrate the capability of the analysis and the associated computer program (PEPSIN). Three-dimensional cases deal with the underexpanded supersonic jet flow (i.e., $P_J > P_E$) exhausting from the rectangular nozzle of three different aspect ratios (1, 2 and 5 respectively). As discussed in the Introduction Section, a supersonic jet was assumed to interact with a supersonic external stream. However, since no data were available to validate the results obtained by PEPSIN analysis for these three-dimensional cases, it was decided to consider an axisymmetric case (Figure 3) first because of the existing computational data (Ref. 17) to validate the PEPSIN results. Compared to the three-dimensional case, the axisymmetric results are, in general, easy to interpret. Moreover, experience and insight obtained during the study of the axisymmetric case is useful for the study of three-dimensional cases. Most of the flow conditions utilized in the computation of the axisymmetric case were obtained from Ref. 17. Some of the unavailable conditions were either assumed or estimated. The analysis was performed in the same axisymmetric geometry as reported in Ref. 17. The Mach number of both jet and external stream was 2.0. The jet flow was hot ($T_J = 1500^\circ K$) and the static pressure ratio of jet with respect to the external stream was $P_J/P_E = 1.45$. The static temperature of the external stream was $240^\circ K$. The static pressure of the external flow was assumed as 2864 (Newton/m$^2$). Thus, the computed Reynolds number per unit length based on the ambient flow conditions was $1.6685 \times 10^6$. To analyze the interaction of the jet flow with the coflowing ambient supersonic flow, 98 mesh points were distributed in the radial direction. The grid points were tightly packed in the vicinity of estimated turbulent mixing layer region to achieve the maximum resolution. The continuity equation, two momenta equations and the energy equation were used as governing equations. For the present case, the subsonic regions associated with the nozzle boundary layer and the base of nozzle wall are embedded in the supersonic flow. However, these embedded subsonic regions appeared to be excluded from consideration in Ref. 17 because the technique
used in Ref. 17 was not capable of analyzing such regions. Therefore, since it was desired to exercise the PEPSIN code under the same conditions as in Ref. 17 for this test case, the solution was marched with the initial profiles devoid of subsonic flow in this study. However, it should be noted that proper treatment of the subsonic regions may be very important in simulating physical situations. Since Ref. 17 didn't specify explicitly how the initial profiles were generated, the shape of initial profiles in the present calculations was approximately represented by the profiles which would be expected downstream in the mixing layer. For instance, the resulting velocity profile at the initial station would consist of two different uniform streams connected by the smooth hyperbolic tangent curve which approximate the transverse distribution of streamwise velocity component in the mixing layer. The boundary conditions imposed on the axis of symmetry are symmetry conditions, while the conditions on the external boundary are Mach line extrapolations. The streamwise step size used for marching the solution was 0.005 of the transverse size of computational domain, i.e., 0.025 of jet radius (corresponding to a Courant number of 1.26).

The computed results by the PEPSIN code are compared with the calculation of Dash et al. (Ref. 17) in Figures 4(a) and 4(b). The figures show static pressure variation in the streamwise direction at two different radial locations (along the jet axis and \( r/r_j = 0.5 \) where \( r_j \) refers to the radius of axisymmetric jet). Considering the uncertainties of the initial flow conditions which were used in PEPSIN analysis, the agreement between two different calculations is reasonable. The PEPSIN analysis indicates a stronger streamwise damping of wave strengths compared to the computation (Ref. 17). This appears to be associated with the difference in turbulent dissipation which occurs as a result of using different turbulence models in the two separate computations. The PEPSIN analysis is based on a algebraic mixing-length turbulence model, whereas a two-equations model (KE model) was utilized by Dash and co-workers. However, both calculations show the absence of any noticeable wave structure beyond the first shock cell, which indicates that the reflected wave resulting from the interaction at the end of the first shock cell becomes very weak. The favorable comparison with the results of Ref. 17 indicates that both PEPSIN approach and its computer program are fundamentally sound.
After the PEPSIN code was successfully validated for an axisymmetric case, three-dimensional cases were considered. As discussed above, the underexpanded supersonic jet was assumed to be exhausted from a rectangular nozzle of aspect ratio 1, 2 and 5 respectively (Figure 5). An aspect ratio of a cross-section is defined by the ratio of spanwise dimension divided by the transverse dimension. Unlike the previously considered axisymmetric case, estimation of initial profiles for three-dimensional cases was not straightforward. Based upon our experience with the previous supersonic flow calculations, an initial profile must be not only smooth but also consistent with the governing equations. If not, perturbations, inconsistencies, etc. can persist far downstream. The difficulty of being consistent with the governing equations has been found to be further pronounced for three-dimensional cases. Therefore, in this study the problem was slightly modified such that initial profiles could be generated more easily. Thus, the jet was assumed to be exhausted from a short straight duct with a square or rectangular cross-section. In the present study we are primarily concerned with developing a computer code capable of predicting the interaction of a supersonic jet with an ambient supersonic flow downstream of exit plane. Since no specific nozzle geometry was defined in the present investigation, the length of duct was kept short mainly to reduce the computation time. The PEPSIN code is, however, capable of analyzing the flow field within a longer nozzle of more complex geometry as shown by the applications of the PEPSIS code to inlet flow systems (Ref. 16).

In this study the streamwise marching solution procedure began at a location upstream of the duct entrance. As an initial profile for this problem, the flow was assumed to consist of two coflowing uniform streams at different velocities as shown in Figure 5. One stream at faster speed moves into the duct, whereas the external flow moving at a slower speed moves around the duct. The initial conditions such as velocity ratio of two streams and corresponding pressure ratio were assumed to be the same as the previously considered axisymmetric case. As a consequence of the modification, the flow at the duct exit was expected to contain some extra physical features due to the development of flow near the duct entrance. The flow both within and outside the duct was complicated because of the shock waves generated due to the initial shock wave-boundary layer development on
the walls of the duct. In particular, the pressure field within the cross-section became non-uniform as one would expect in a rectangular duct flow.

The Mach number of both the initial streams was 2.0, while the Reynolds number per unit length based on the free stream (external flow) properties was $1.6685 \times 10^6$ (a jet exit Reynolds number was $2.847 \times 10^5$). In order to analyze both the internal and external flow of the duct as well as the jet flow interacting with the external stream, $50 \times 50$ mesh points for aspect ratio 1 and $50 \times 80$ mesh points for both aspect ratio 2 and 5 were distributed in each cross-sectional plane. Mesh points were packed in the region of the interaction. Because of the geometrical shape of both duct and jet flow, two planes of symmetry can be associated with this problem, and hence the flow need only be solved in the quarter plane. On the planes of symmetry, symmetry boundary conditions were utilized. On the other hand, three no-slip conditions for the velocity components, normal momentum equation and adiabatic wall conditions were imposed on the solid walls. On the external boundary of flow symmetry boundary conditions were used again. For the three-dimensional cases, the continuity equation, three momenta equations and the energy equation were used as governing equations.

The turbulence model utilized in this study was an algebraic mixing length model. This model was selected primarily because it is convenient to use at this point (demonstration of capability of new computer code was one of the primary goals in this study) and, furthermore, it can save some computer resources (time and storage). However, if a demonstrated need for advanced turbulence model exists, this can be easily accommodated within the existing code framework. The mixing length was assumed to be proportional to the half width of mixing region and, therefore, constant at each streamwise location. In the present study, for the purpose of the turbulence model the spreading angle of the mixing region was assumed to be $3^\circ$.

The computed results of three-dimensional cases are presented in Figs. 8 through 16. Since experimental data is not available for direct comparison, discussion will primarily focus on qualitative description of the essential physical features predicted by the computation. To assist the reader in interpreting the plots of various variables, both Figs. 6 and 7 are presented. Figure 6 illustrates a sideview of the computational domain.
on a vertical plane when the computational domain is cut through the duct vertically. Figure 7 provides a cross-sectional view of the domain.

As discussed before, three-dimensional test cases were run for three different aspect ratios of rectangular cross-section. Therefore, discussion will first concentrate on the description of the results for an aspect ratio equal to 1. Then, the effects of larger aspect ratios on the flow field will be considered.

Calculated results in the form of stagnation pressure and cross-sectional secondary velocity vectors for square nozzle case (aspect ratio 1) are presented in Figs. 8-9. Stagnation pressure isobars in the cross-sections at seven streamwise locations are presented to define a streamwise development of plume boundary (Fig. 8). Initial upstream contours were obtained at a distance of $X/H = 0.33$ from the exit plane of the duct where $H$ is a height of the duct. $H$ is chosen to be 0.4. Meanwhile, final downstream contours were obtained at a distance of $X/H = 22$ from the exit plane of the duct. The plume boundary experiences a severely distorted square shape before evolving into a smooth shape far downstream. This is probably due to strong three-dimensional effects of the flow. Compared to the usual supersonic jet exhausted from a nozzle, these three-dimensional effects may have been more pronounced by the use of a short straight duct as a means of avoiding the difficulty associated with generating three-dimensional initial profiles. As discussed before, a uniform flow was assumed to move into the duct entrance. As a consequence, the flow within the duct becomes highly three-dimensional due to the generation of shock waves associated with the leading edge. Combined with the three-dimensional effects, a highly non-uniform pressure develops in the cross-section of duct. In particular the pressure in the corner region is large. Thus, the largest pressure difference across the duct wall occurs in the corner region. Because of this non-uniform pressure distribution, a distortion of the plume boundary occurs. As can be seen in Fig. 8, the distortion is slightly asymmetric about the bisecting line (45° line) from the corner. The asymmetry was caused by the approximations used in the subsonic layer of the corner region of a nozzle. As discussed in the analysis section, both the continuity and normal momentum equations are approximated. The nearest solid wall from a grid point is identified to implement the subsonic approximations. However, the grid points located on the bisecting line have
two such surfaces. In this case the upper surface was arbitrarily chosen and, hence, a slight asymmetry introduced. It should be noted that asymmetry can be reduced by providing more grid points in the corner region, and as the flow proceeds downstream this effect significantly diminishes. A streamwise development of secondary velocity vectors in the cross-section is presented in Fig. 9. A strong secondary flow is seen in the corner region. As a result of the non-uniform pressure distribution, the flow develops two streamwise vortices which will eventually be diffused by the turbulent mixing action. To show the effects of jet aspect ratio on the flow field, Figs. 10 and 11 are presented. Both figures illustrate a streamwise development of stagnation pressure contours. Initial upstream contours are obtained at a distance of \(X/H = 0.675\) from the exit plane of the duct where \(H\) is a height of duct. \(H\) is taken to be 0.2. Final downstream contours are obtained at a distance of \(X/H = 44\).

The outer edge of the mixing region may be defined qualitatively based on these two figures. Examination of the plots indicates that the mixing layer spreads out more quickly in the corner region as discussed previously in the case of aspect ratio 1. Streamwise development of cross-sectional plume boundary is very similar to that for aspect ratio 1, although distortion of plume boundary is localized in the corner region at an aspect ratio 5. It appears that the rapid mixing in the corner region is related to the three-dimensional effects. However, as both figures show, the plume boundary gradually approaches a smooth elliptic or circular shape in far field. In the near field downstream of the nozzle exit, the high pressure in the corner region associated with the shock waves may have played a significant role in enhancing the mixing. It should be noted that similar mixing behavior may not be observed when the calculation is performed for a realistic nozzle under the initial profiles based on the exit flow conditions.
The primary objective of this investigation was to develop and validate a numerical procedure for the calculation of the interaction of three-dimensional supersonic jet with an external flow. The new computer code (named PEPSIN) was developed by extending the capability of existing code (PEPSIS) suitable for an analysis of jet flows. The major difference between these two codes lies in the new capability of calculating flow fields when the computational domain contains internally embedded solid bodies. To validate the code, one two-dimensional and three three-dimensional cases were computed. Comparison of the two-dimensional results with other computations was performed to validate the new code for this class of cases. Three-dimensional calculations were successfully performed, and essential physical features were predicted.
The PEPSIN Users' Manual is meant to serve as a guide in helping the user make successful runs with the PEPSIN computer program. The degree of success obtained by the user will depend on the skill of the user and his ability to correctly apply the code to his particular problem. The code will solve the governing equations, subject to the user supplied boundary conditions, however, meaningful results will only be obtained if the boundary conditions are appropriate to the problem. In addition, the user must specify viscosity models, initial conditions, a coordinate system and the location of grid points to adequately resolve the flow. The user with a good knowledge of the physics involved in his problem and how the code models the physics should with a moderate amount of experience be able to successfully apply the code to a wide variety of supersonic flow problems.

The Users' Manual is divided into eight parts consisting of: (1) a flow diagram, (2) a brief description of each subroutine and its use, (3) a list of the Fortran variables and a description of their meaning, (4) a description of the logical file units utilized by the PEPSIN computer code, (5) a detailed description of the input required by the PEPSIN computer code, (6) a description of the common error conditions that may be encountered during the execution of a PEPSIN run and the corrective action to be taken, (7) sample input for a three-dimensional case and (8) sample output for the corresponding case.

Flow Diagram

The purpose of the flow diagram is to help the user understand the basic flow of information within the PEPSIN computer code. Because of the size of the code (approximately 13,000 cards), a detailed flow diagram would be prohibitively large and probably be of little value to the user. Therefore, the flow diagram is intended only to give a general overview of the structure of the code. The interested user is urged to consult the program listing for details.
Flow Diagram for the PEPSIN Computer Code

Read Initial Data

No

Restart

Yes

INPUTS
JXS = 1
Read in Data

SETUP
Set up Computational Domain in Cross Section

GEOTRB
Calculate Geometry

PROF
Initial Profile

CROSEC
Calculate Derived Variables

A

RESTR
t
JXS = IRSTIN + 1
Read File

INPUTS
Reset Data

SET UP
Set up Computational Domain in Cross-Section

GEOTRB
Calculate Geometry

B
A

Yes

Plots

No

B

Plots

No

Yes

Position Plot File

Yes

No

D

C

Jdummy = Jxs

SETUP

Set up Computational Domain in Cross-Section

GEOTRB

Calculate Geometry

ADI

Solve Governing Eqs.

PLOTIN Initialize Plots

Initialize Plot File

PLOTIN Initialize Plots
流程图描述如下：

1. 开始（E）
2. JDUMMY = JDUMMY + 1
3. JDUMMY .LE. NS
   - 是（YES）：返回到 C
   - 否（NO）：结束（END）
<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDRES</td>
<td>Calculate addresses for finite difference representation of metric and fluid dynamic variables.</td>
</tr>
<tr>
<td>ADI</td>
<td>Master control subroutine for ADI procedure.</td>
</tr>
<tr>
<td>ADICP</td>
<td>Control subroutine for coupled equations.</td>
</tr>
<tr>
<td>ADIUN</td>
<td>Control subroutine for uncoupled equations.</td>
</tr>
<tr>
<td>AMARCH</td>
<td>Linearizes streamwise convective terms.</td>
</tr>
<tr>
<td>AMATRX</td>
<td>Linearizes all streamwise terms.</td>
</tr>
<tr>
<td>ARTVIS</td>
<td>Artificial dissipator subroutine.</td>
</tr>
<tr>
<td>AVRGG</td>
<td>Calculates averaged quantities in cross plane.</td>
</tr>
<tr>
<td>BC</td>
<td>Boundary condition subroutine.</td>
</tr>
<tr>
<td>BLKDATA</td>
<td>Stores default values of key variables.</td>
</tr>
<tr>
<td>BLT</td>
<td>Calculates boundary layer thickness.</td>
</tr>
<tr>
<td>BULEEVEV</td>
<td>Calculates Buleev mixing length.</td>
</tr>
<tr>
<td>CONVCT</td>
<td>Linearizes cross plane convective terms.</td>
</tr>
<tr>
<td>CORBND</td>
<td>Calculates geometry transformation information on boundaries.</td>
</tr>
<tr>
<td>CORTRN</td>
<td>Calculates geometry transformation information for interior points.</td>
</tr>
<tr>
<td>CROSEC</td>
<td>Control subroutine for calculation of derived variables.</td>
</tr>
<tr>
<td>CURVT</td>
<td>Linearizes curvature terms.</td>
</tr>
<tr>
<td>DATAS</td>
<td>Logical file control subroutine.</td>
</tr>
<tr>
<td>DELTX</td>
<td>Calculates transformation information for ICORD = 2 option.</td>
</tr>
<tr>
<td>DELTXZ</td>
<td>Calculates transformation information for ICORD = 3 option.</td>
</tr>
<tr>
<td>DIFF</td>
<td>Linearizes diffusion terms.</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Purpose</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DISFCN</td>
<td>Calculates dissipation function.</td>
</tr>
<tr>
<td>DIV</td>
<td>Calculates divergence of velocity.</td>
</tr>
<tr>
<td>DOP2</td>
<td>Control subroutine for linearization of Y-direction and source terms.</td>
</tr>
<tr>
<td>DOP3</td>
<td>Control subroutine for linearization of Z-direction terms.</td>
</tr>
<tr>
<td>ENDCAP</td>
<td>Control subroutine for endcap conditions.</td>
</tr>
<tr>
<td>EOS</td>
<td>Equation of state subroutine linearizes and updates pressure and temperature.</td>
</tr>
<tr>
<td>FGFUN</td>
<td>Calculates geometry groupings.</td>
</tr>
<tr>
<td>GAUSS</td>
<td>Solves uncoupled tridiagonal set of equations.</td>
</tr>
<tr>
<td>GENCBC</td>
<td>Control subroutine for coupled boundary conditions.</td>
</tr>
<tr>
<td>GENUBC</td>
<td>Control subroutine for uncoupled boundary conditions.</td>
</tr>
<tr>
<td>GEORD</td>
<td>Controls setup of computational domain and calculates geometry at each cross-section.</td>
</tr>
<tr>
<td>GEOTRB</td>
<td>Generates metric information.</td>
</tr>
<tr>
<td>INDIC</td>
<td>Determines if flow is subsonic or supersonic at grid points.</td>
</tr>
<tr>
<td>INPUTS</td>
<td>Input subroutine. Input data enters and is processed.</td>
</tr>
<tr>
<td>INTEBC</td>
<td>Performs a two-dimensional linear interpolation for wall transpiration rates.</td>
</tr>
<tr>
<td>LAMP</td>
<td>Calculates laminar profile.</td>
</tr>
<tr>
<td>LAW</td>
<td>Calculates nondimensional velocity, $U^+$ as a function of nondimensional distance $Y^+$.</td>
</tr>
<tr>
<td>LENGTH</td>
<td>Calculates mixing length.</td>
</tr>
<tr>
<td>LOOP</td>
<td>Determines the loop index at a grid point in both Y and Z-direction.</td>
</tr>
<tr>
<td>MAIN</td>
<td>Main control program.</td>
</tr>
<tr>
<td>MATPRT</td>
<td>Prints elements of block tridiagonal matrix.</td>
</tr>
<tr>
<td><strong>Subroutine</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>----------------</td>
<td>------------</td>
</tr>
<tr>
<td>MGAUSS</td>
<td>Control subroutine for solving block tridiagonal systems of equations.</td>
</tr>
<tr>
<td>MGERR</td>
<td>Calculates error associated with solving block tridiagonal system of equations.</td>
</tr>
<tr>
<td>NMLIST</td>
<td>Subroutine for printing namelist input information.</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>Control subroutine for printing out results on a cross-sectional (Y-Z) plane.</td>
</tr>
<tr>
<td>PLOT</td>
<td>Writes plot information on logical file unit JPLOT.</td>
</tr>
<tr>
<td>PLOTIN</td>
<td>Writes first record of general information on logical file unit JPLOT.</td>
</tr>
<tr>
<td>PROF</td>
<td>Generates initial profiles.</td>
</tr>
<tr>
<td>QUICK</td>
<td>Matrix elimination subroutine.</td>
</tr>
<tr>
<td>READZ</td>
<td>Prepares variables for printing.</td>
</tr>
<tr>
<td>RESTRT</td>
<td>Reads and writes restart information.</td>
</tr>
<tr>
<td>ROTATE</td>
<td>Rotates data from columns to rows and vice versa.</td>
</tr>
<tr>
<td>SETBVL</td>
<td>Updates boundary information a line at a time.</td>
</tr>
<tr>
<td>SETBVP</td>
<td>Updates boundary information a point at a time.</td>
</tr>
<tr>
<td>SETUP</td>
<td>Determines the extent of computational domain, number of loops and subsections at each cross-section</td>
</tr>
<tr>
<td>SHEAR</td>
<td>Control subroutine for the calculation of wall shear velocity.</td>
</tr>
<tr>
<td>SONIC</td>
<td>Determines the extent of supersonic and subsonic region.</td>
</tr>
<tr>
<td>SPREAD</td>
<td>Spreads two-dimensional data to three dimensions.</td>
</tr>
<tr>
<td>STORG</td>
<td>Control subroutine of the storage of temperature and density on boundaries for viscosity calculation.</td>
</tr>
<tr>
<td>SUB</td>
<td>Contain special subsonic logic.</td>
</tr>
<tr>
<td>SWITCH</td>
<td>Calculates streamwise location for switch of boundary condition.</td>
</tr>
<tr>
<td>TANHYP</td>
<td>Grid stretch subroutine.</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Purpose</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>TNDER</td>
<td>Calculates normal derivative of temperature at a wall.</td>
</tr>
<tr>
<td>TRANS</td>
<td>Transition model subroutine.</td>
</tr>
<tr>
<td>TURB</td>
<td>Turbulence model subroutine.</td>
</tr>
<tr>
<td>TURBP</td>
<td>Calculates turbulent profile based on theory of Maise-McDonald.</td>
</tr>
<tr>
<td>VISCOS</td>
<td>Constant and laminar viscosity subroutine.</td>
</tr>
<tr>
<td>WALLFN</td>
<td>Calculates wall shear velocity.</td>
</tr>
<tr>
<td>WHERE</td>
<td>Determines the surface number of four nearest boundaries for a given point.</td>
</tr>
<tr>
<td>WRMATR</td>
<td>Writes block tridiagonal dump information on logical file device NUNERR.</td>
</tr>
<tr>
<td>YCALC</td>
<td>Calculates Y and Z locations.</td>
</tr>
<tr>
<td>ZERO</td>
<td>Zeros out linearization arrays.</td>
</tr>
</tbody>
</table>
Logical File Units Utilized by PEPSIN Computer Code

The PEPSIN computer code utilizes up to eleven (11) logical file units during the execution of a run stream. In many cases not all eleven units are used, and hence in these cases there is no need to define all eleven units. All references to a logical file unit in the PEPSIN computer code is accomplished through the use of a FORTRAN name rather than through a specific unit number. Thus, if the user desires to change a logical file unit number, this can be done through the input file. A list of the logical file units utilized by the PEPSIN computer code, their FORTRAN name, default value unit number, and a brief description of the use of the unit is presented below. All units are sequential.

<table>
<thead>
<tr>
<th>FORTRAN Name</th>
<th>Default Unit Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIN</td>
<td>5</td>
<td>Input data unit.</td>
</tr>
<tr>
<td>MOUT</td>
<td>6</td>
<td>Printed output unit.</td>
</tr>
<tr>
<td>MASS1</td>
<td>8</td>
<td>First unit which stores dependent and derived variables either by rows or columns. Not needed for two-dimensional cases, i.e., when TWOD = .TRUE.</td>
</tr>
<tr>
<td>MASS2</td>
<td>9</td>
<td>Second unit which stores dependent and derived variables either by rows or by columns. Not needed for two-dimensional cases, i.e., when TWOD = .TRUE.</td>
</tr>
<tr>
<td>MSDD</td>
<td>15</td>
<td>Unit which stores dependent and derived variables by rows or columns. Not needed for two-dimensional cases, i.e., when TWOD = .TRUE.</td>
</tr>
<tr>
<td>JDRUM</td>
<td>11</td>
<td>Unit which contains output of ADD computer code. Only needed when IGEOM = 10 or 11.</td>
</tr>
<tr>
<td>FORTRAN Name</td>
<td>Default Unit Number</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>KDRUM</td>
<td>12</td>
<td>Unit which stores final metric information. Needed in all cases.</td>
</tr>
<tr>
<td>NUNERR</td>
<td>14</td>
<td>Unit which stores information concerning the block tridiagonal matrix inversion. Needed when MGDMP ≠ 0.</td>
</tr>
<tr>
<td>JPLT</td>
<td>15</td>
<td>Unit which stores plotting information. Needed when IPLOT ≠ 0.</td>
</tr>
<tr>
<td>JRSTIN</td>
<td>10</td>
<td>Input restart unit. This unit contains appropriate common block information and the values of the dependent and derived variables at each cross-sectional grid point at the restart streamwise station. Needed only when JRSTIN ≠ 0.</td>
</tr>
<tr>
<td>JRSTOT</td>
<td>10</td>
<td>Output restart unit. This unit contains appropriate common block information, and the values of the dependent and derived variables at each cross-sectional grid point at the restart streamwise station. Needed only when JRSTOT ≠ 0. Default is JRSTIN = JRSTOT; however, it is desired to have separate input restart and output restart files set JRSTOT = 17.</td>
</tr>
</tbody>
</table>
PEPSIN Input

Except for an initial title card and plot file input data, the entire PEPSIN input is entered by means of the NAMELIST format. There are two primary advantages to the use of the NAMELIST format: (1) that if the default values (defined in the block data subroutine) are acceptable, the user need not input that variable, and (2) the order (within a given NAMELIST) in which the variables are entered is irrelevant. There are seven NAMELIST input files in the PEPSIN code, $REST, $LIST1 through $LIST5 and $LISTR. The first file is read in the main program and enters restart information. The second through sixth NAMELIST files are read in subroutine INPUTS. Basically, the NAMELISTS $LIST1 through $LIST5 can be divided by function. $LIST1 enters information about the governing equations and appropriate boundary conditions, $LIST2 enters reference and free stream conditions, $LIST3 enters geometric information, $LIST4 enters viscosity model and initial profile information and $LIST5 enters file output information. The last NAMELIST file is read in subroutine CROSEC. $LISTR enters information needed to reset the grid array indicator, IFBW, for the computational domain in the cross-section. The PEPSIN computer code evolved from the PEPSIS computer code (Ref. 16) which was developed primarily for flows in supersonic inlets and for external flow situations. The primary difference between the two codes is in the ability of PEPSIN to consider cross-sectional geometric configuration that contain re-entrant corners as for example, might occur in the geometry illustrated in Fig. 12. In concept, the extension of the PEPSIS procedure to such geometric configurations is straightforward. However, to incorporate the re-entrant corner logic into the code in a general manner adds considerable complexity to the computer code. To minimize the effect of the complexities on the user while at the same time not detract from the generality that may be necessary at the present time or in the future, a very general procedure was incorporated into the PEPSIN code. This procedure utilizes the concept of a "grid array indicator" (FORTRAN variable IFBW) to type the grid points in the cross-sectional plane. Referring to Fig. 12, it can be seen that the idea is to type the grid points according to the function they serve. The convention is to type grid points as follows:
IFBW=1 Indicates that a grid point is an interior fluid point.
IFBW=2 Indicates that a grid point is an interior solid point, and thus does not effect the calculation.
IFBW=3 Indicates that a grid point is a noncorner boundary point.
IFBW=4 Indicates that a grid point is an inward corner point.
IFBW=5 Indicates that a grid point is a re-entrant corner point.

The FORTRAN variable is dimensioned IFBW(NN,NN). Based upon the information supplied by the user, subroutine SETUP sets up the necessary parameters to control the ADI procedure. Initially, the grid array indicator is introduced through NAMELIST LIST3. As the solution is marched downstream, the values of IFBW will remain unchanged. However, when it is desired to change the cross-sectional grid structure this can be accomplished by requesting that NAMELIST LISTR be read. This is determined by the input FORTRAN variable NBRKX which is read in NAMELIST LIST 3.
In addition to the explanation of the grid array indicator, it is convenient at this point to explain the surface number (KSURF) convention and direction (IDIR) convention utilized throughout the PEPSIN computer code. Several input variables, e.g. IBOUND(KSURF,IDIR) require a knowledge of this convention to correctly input data. The boundary surfaces are numbered according to a surface number relative to a cross-sectional computational direction. The convention is to allow values of IDIR=1 and 2 to correspond to the y and z physical directions, respectively. Referring to Fig. 12, it can be seen that the surfaces (with respect to y-direction) are numbered 1Y to 10Y. This corresponds to values of KSURF ranging from 1 to 10 and a value of IDIR corresponding to the y-direction i.e. IDIR=1. The analogous convention is also applied to the z-direction surfaces. Thus, the value of KSURF represents the surface number relative to a given cross-sectional direction, IDIR.

A description of all the PEPSIN input information will be given below.

Card 1

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
<th>Variable</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 24</td>
<td>6A4</td>
<td>TITLE(I)</td>
<td>Title Card</td>
</tr>
</tbody>
</table>

Card 2

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
<th>Variable</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 2</td>
<td>I12</td>
<td>ISYM</td>
<td>Reciprocal of Symmetry</td>
</tr>
<tr>
<td>3 - 12</td>
<td>1F10.0</td>
<td>SYSTEM</td>
<td>SYSTEM = 1 - Quasi-Cartesian Coordinates</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SYSTEM = 2 - Quasi-Cylindrical Coordinates</td>
</tr>
</tbody>
</table>
## Namelist Input Description

<table>
<thead>
<tr>
<th>Namelist or Variable Name</th>
<th>Description</th>
<th>Restart Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>IRSTIN</td>
<td>Marching station number where data is to be read in for restart case.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IRSTIN = 0: Dead start case.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IRSTIN ≠ 0: Restart case started at station IRSTIN. Default: 0.</td>
<td></td>
</tr>
<tr>
<td>IRSTOT</td>
<td>Interval for saving restart information.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IRSTOT = 0: No restart information is saved.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IRSTOT ≠ 0: Information is saved at each IRSTOTth station. Default: 0</td>
<td></td>
</tr>
<tr>
<td>JRSTIN</td>
<td>Logical file name of input restart file. Default: 10.</td>
<td></td>
</tr>
<tr>
<td>JRSTOT</td>
<td>Logical file name of output restart file. JRSTOT and JRSTIN do not have to be same file. Default: 10:</td>
<td></td>
</tr>
<tr>
<td>NFILE</td>
<td>File number on unit JRSTIN desired for restart. Default: 0.</td>
<td></td>
</tr>
<tr>
<td>NSAVED</td>
<td>Number of restart stations saved on JRSTOT. On a restart by setting JRSTOT = JRSTIN and NFILE = NSAVED, one file can be used for both reading and writing without destroying the information previously saved. Default: -1.</td>
<td></td>
</tr>
</tbody>
</table>
Namelist Input Description

**Namelist or Variable Name**

**REST**

**Description**

**Restart Options**

**ICOMP**

Flag for computer options:

- **ICOMP = 1**: Univac computer option.
- **ICOMP = 2**: CDC computer option.
- **ICOMP = 3**: IBM computer - virtual memory option.
- **ICOMP = 4**: Disk writing computer option.
  
  Default: 4.

**LIST1**

**IHSTAG**

- **IHSTAG = 0**: Energy equation formulated in terms of static enthalpy.
- **IHSTAG = 1**: Energy equation formulated in terms of stagnation enthalpy.
- **IHSTAG = 2**: Stagnation enthalpy is constant.
  
  Default: 1.

**IBOUND(KSURF,IDIR)**

Boundary characteristics (wall or non-wall) on surface KSURF in a computational domain.

- **IBOUND (KSURF,IDIR) = 1**: Solid wall boundary.
- **IBOUND (KSURF,IDIR) = 2**: Non-wall boundary.
  
  Default: 40*1.

**IEQBC(KSURF,IDIR,IEQ)**

Boundary condition of the governing equation IEQ on solid surface KSURF.

Default: 120*2, 40*16, 40*11, 80*2.

**JEQBC(KSURF,IDIR,IEQ)**

Boundary condition of the governing equation IEQ on non-wall surface KSURF.


Boundary condition options used for either IEQBC or JEQBC are as follows:

- $\phi$: Any dependent variable.
- $P$: Pressure.
- $T$: Temperature.
- $n$: Normal to boundary.
- $n$: Unit vector perpendicular to axis of symmetry.

  subscript C = Cartesian component.
### Equations and Boundary Conditions

<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\Delta \phi = 0$ (No change of $\phi$ at boundary)</td>
</tr>
<tr>
<td>2</td>
<td>$\phi = 0$</td>
</tr>
<tr>
<td>3</td>
<td>$V$ or $W$ known</td>
</tr>
<tr>
<td>4</td>
<td>$\rho V$ or $\rho W$ known</td>
</tr>
<tr>
<td>5</td>
<td>$\Delta P = 0$</td>
</tr>
<tr>
<td>6</td>
<td>$P = \text{PRESS(KSURF, IDIR)}$</td>
</tr>
<tr>
<td>7</td>
<td>$\Delta T = 0$</td>
</tr>
<tr>
<td>8</td>
<td>$T = \text{TWALL(KSURF, IDIR)}$</td>
</tr>
<tr>
<td>11</td>
<td>$\frac{\partial \phi}{\partial n} = 0$ (gradient of $\phi$ normal to boundary specified at boundary)</td>
</tr>
<tr>
<td>12</td>
<td>Mach Line Extrapolation using one-sided difference weights</td>
</tr>
<tr>
<td>13</td>
<td>Slip boundary condition for velocity using wall function</td>
</tr>
<tr>
<td>14</td>
<td>$\frac{\partial P}{\partial n} = 0$ (gradient of pressure normal to boundary)</td>
</tr>
<tr>
<td>15</td>
<td>$\frac{\partial P}{\partial n} = \text{curvature}$ (pressure gradient normal to boundary with curvature effects)</td>
</tr>
<tr>
<td>16</td>
<td>Momentum equation in direction normal to boundary</td>
</tr>
<tr>
<td>17</td>
<td>$\frac{\partial T}{\partial n} = 0$ (adiabatic condition for wall or symmetry condition for non-wall)</td>
</tr>
<tr>
<td>18</td>
<td>$\frac{\partial T}{\partial n} = \text{DTDN(KSURF, IDIR)}$</td>
</tr>
<tr>
<td>19</td>
<td>Wall function boundary condition for temperature</td>
</tr>
</tbody>
</table>
| 20    | $\frac{\partial \phi}{\partial n} = 0$ (same as 11, but applied at one grid point off the
### Namelist Input Description

#### Equations and Boundary Conditions

**Index** | **Description**
--- | ---
21 | Mach Line Extrapolation using central difference scheme at one point off the boundary
41 | \( \frac{\partial^2 \phi}{\partial n^2} = 0 \)
42 | \( \frac{\partial^2 p}{\partial n^2} = 0 \)
43 | \( \frac{\partial^2 T}{\partial n^2} = 0 \)
44 | \( \frac{\partial u}{\partial n} = 0 \)
45 | \( \frac{\partial v}{\partial n} = 0 \)
46 | \( n_c \partial \phi = 0 \)
47 | \( n_c \partial \theta = 0 \)

**TWALL(KSURF,IDIR)** | Specified temperature on surface KSURF.
Default: 40*1.0.

**PRESS(KSURF,IDIR)** | Specified pressure on surface KSURF.
Default: 40*0.0.

**DTDN(KSURF,IDIR)** | Specified temperature gradient on surface KSURF
Default: 40*0.0.

**ASW(KSURF,IDIR)** | Coefficient of a cubic polynomial fit for KSURFth surface to determine the axial location where boundary characteristics on surface KSURF should be changed from wall to non-wall or vice versa, i.e., IBOUND(KSURF,IDIR) automatically changed.
Namelist Input Description

**LIST1**

Equations and Boundary Conditions

**Description**

Default:

\[
\begin{align*}
\text{ASW} &= 40 \times 1.0 \times 10 \\
\text{BSW} &= 40 \times 0.0 \\
\text{CSW} &= 40 \times 0.0 \\
\text{DSW} &= 40 \times 0.0
\end{align*}
\]

**LIST2**

Freestream and Reference Conditions

**IUNITS**

Sentinel for units.

\[\text{IUNITS} = 1: \text{English units}\]

\[\text{IUNITS} = 2: \text{Metric units}\]

Default: 2.

**LREF**

Reference length. (ft or meters)

No default.

**REPL**

Reynolds number per unit length (ft\(^{-1}\) or m\(^{-1}\))

No default.

**MINF**

Freestream Mach number.

No default.

**PINF**

Freestream static pressure (lbf/ft\(^2\) or nt/m\(^2\)).

No default.

**PR**

Laminar Prandtl number.

Default: 0.74.

**PRT**

Turbulent Prandtl number.

Default: 1.0.

**PZERO**

Freestream stagnation pressure (lbf/ft\(^2\) or nt/m\(^2\)).

No default.
Namelist Input Description

LIST3

Geometric Options

ICORD
Flag for coordinate transformation.

ICORD = 1: Conformal coordinates
ICORD = 2: Nonorthogonal coordinates

\( X + X \)
\( Y + Y \)
\( Z + \zeta(X,Y) \)

ICORD = 3: Nonorthogonal coordinates

\( X + X \)
\( Y + \eta(X,Y,Z) \)
\( Z + Z \)

Default: 1.

NE(IDIR)
Number of grid points in the \( Y(IDIR = 1) \) and \( Z(IDIR = 2) \) directions.

No default values.

NS
Last streamwise station in each run.
(Solution is marched from \( IIRSTIN+1 \) to \( NS \)).

No default.

XENTR, DELX, IAP(ICOUNT), AP(ICOUNT), DXMIN(ICOUNT), DXMAX(ICOUNT)

XENTR is the initial streamwise location. DELX is the initial stepsize in the streamwise (marching) direction, i.e., \( X(2) = XENTR + DELX \). At streamwise station \( I \) the streamwise position is given by \( X(I) = X(I-1) + AP(X(I-1) - X(I-2)) \) where if \( AP \) is greater than 1.0, the streamwise step size will increase by \( (AP-1.0) \) percent each step. If \( AP \) is less than 1.0, the streamwise step size will decrease by \( (1.0-AP) \) percent each step. DXMIN and DXMAX are lower and upper overriding limits on the step size. AP, DXMIN and DXMAX are dimensional so that streamwise step size variation can be changed by the IAP parameter, IAP denoting the streamwise location where these variables change. Values of XENTR and DELX should normally only be set on the initial run as these variables are automatically calculated for restarts. Maximum ICOUNT is 10.
Namelist Input Description

**LIST3**

**Geometric Options**

Default:

- **XENTR, DELX**:  
  - Default: \( XENTR = 0.0 \)

- **IAP(ICOUNT, AP(ICOUNT), IAP = 1,9*1000000**
  - Default: \( AP = 10*1.0 \)

- **DXMIN(ICOUNT)**
  - Default: \( DXMIN = 10*0.0 \)

- **DXMAX(ICOUNT)**
  - (CONTINUED) 
  - Default: \( DXMAX = 10*1.0E + 06 \)

- **DELX**: No default.

**IGEOM**: Flag for coordinate options

- IGEOM = 1: Cartesian coordinates
- IGEOM = 2: Cylindrical coordinates
- IGEOM = 3: Polar coordinates
- IGEOM = 10: General orthogonal coordinates (Cartesian in cross plane)
- IGEOM = 11: General orthogonal coordinates (axisymmetric)

**TWOD**: Sentinel for two-dimensional option.

- If TWOD = .TRUE. \( \) TWO DIMENSIONAL
- If TWOD = .FALSE. \( \) THREE DIMENSIONAL

**Default**: .FALSE.

**TT1(IDIR)**: Grid distribution factor (lower surface (IDIR=1) - or, left surface (IDIR=2)). The closer the value is to 1.0, the tighter the packing.

- Default: 2*0.0.

**TT2(IDIR)**: Grid distribution factor (upper surface (IDIR=1) - or, right surface (IDIR=2)). The closer the value is to 1.0, the tighter the packing.

- Default: 2*0.0.
Namelist Input Description

**LIST3**

**T2(IDIR)**

Grid distribution factor (Y-direction (IDIR=1) or Z-direction (IDIR=2)). The larger the value is, the tighter the packing. Use in conjunction with XO. The value should not exceed 5.0.

Default: 2*0.0

**XO(IDIR)**

Clustering location of grid points within the computational domain (XO(1) is Y-location and XO(2) is Z-location). Use in conjunction with T2.

Default: 2*0.0

**YS(ILIM,IDIR)**

Define computational domain in Y-Z cross plane. ILIM defines either lower or upper limit, ILIM=1(LOWER), ILIM=2(UPPER).

YS(1,1) = 0.0 - lower limit Y-direction
YS(2,1) = 1.0 - upper limit Y-direction
YS(1,2) = 0.0 - lower limit Z-direction
YS(2,2) = 1.0 - upper limit Z-direction

Default: 0.0, 1.0, 0.0, 1.0.

**IFBW(JY,KZ)**

Grid array indicator needed to set up the computational domain in the cross plane at each streamwise station.

**IFBW(JY,KZ)** Notation

IFBW = 1 - Interior fluids point
IFBW = 2 - Interior solids point
IFBW = 3 - Conventional boundary point (fluids or solids)
IFBW = 4 - Boundary point outward corner
IFBW = 5 - Boundary point inward corner

No default.

**XBRKX(ICOUNT)**

Streamwise (marching direction) physical locations where an embedded solid body starts or ends. Maximum ICOUNT is 5.

Default: 5*1.0E+06.
Namelist Input Description

Geometric Options

LIST3

NBRKX
Total number of the streamwise locations where embedded solid bodies start or terminate.
Default: 0.

IBRMN(KZ,ICOUNT)
Minimum Y index of the grid points which change their characteristics at each spanwise (KZ) location of the cross-section. Maximum ICOUNT is NBRKX.
Default: 500*0.

IBRMX(KZ,ICOUNT)
Maximum Y index of the grid points which change their characteristics at each spanwise (KZ) location of the cross-section. Maximum ICOUNT is NBRKX.
Default: 500*0.

IEDGE
Flag which tells whether grid point changes its characteristics from fluids to solids or vice versa.

IEDGE = 0: From fluids to solids.
IEDGE = 1: From solids to fluids.
No default.

LIST4
Initial Profile, Turbulence Information

IBCP(LP)
Basic surface for initial profile generation for LPth loop.

Boundary Layer Profile in a loop

at lower surface 1  IBCP = 1
at upper surface 2  IBCP = 2
at left surface 3  IBCP = 3
at right surface 4  IBCP = 4

Default: 1.

DELTAP(IBCP,LP)
Boundary layer thickness on surface IBCP of LPth loop needed to generate the initial profile referenced to each surface.
No default.

CFP(IBCP,LP)
Skin friction coefficient on surface IBCP of LPth loop needed to generate the initial turbulent boundary layer profile.
No default.
**IPROF**

Flag for initial profile options.

- **IPROF = 1:** Freestream profiles
- **IPROF = 2:** Initial profiles supplied by user
- **IPROF = 3:** Boundary layer profiles based on necessary input
- **IPROF = 4:** Same as IPROF=3, but angular components are obtained for general orthogonal coordinates.

Default value is 1.
Namelist Input Description

<table>
<thead>
<tr>
<th>LIST4</th>
<th>Initial Profile, Turbulence Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMIXL</td>
<td>Flag for mixing length options.</td>
</tr>
<tr>
<td>IMIXL = 1: McDonald-Camarrata mixing length model based on prescribed boundary layer thickness (DELTAB). With wall shear value used to calculate nondimensional distance</td>
<td></td>
</tr>
<tr>
<td>IMIXL = 2: Buleev mixing length model</td>
<td></td>
</tr>
<tr>
<td>IMIXL = 3: McDonald-Camarrata mixing length model based on dynamically obtained boundary layer thickness with fixed wall shear</td>
<td></td>
</tr>
<tr>
<td>IMIXL = 4: Same as IMIXL = 1, but local shear is used to calculate nondimensional distance</td>
<td></td>
</tr>
<tr>
<td>IMIXL = 5: Same as IMIXL = 3, but local shear is used</td>
<td></td>
</tr>
<tr>
<td>IMIXL = 6: Mixing length model for jet flow</td>
<td></td>
</tr>
<tr>
<td>Default: 1.</td>
<td></td>
</tr>
</tbody>
</table>

BETA | Angle of attack in degrees. |
| Default: 0.0. |

YAW | Yaw angle in degrees. |
| Default: 0.0. |

IVISC | Flag for viscosity options. |
| IVISC = 1: Constant viscosity |
| IVISC = 2: Laminar viscosity obtained from Sutherland's relation |
| IVISC = 3: Turbulent viscosity is obtained from mixing length model |
| IVISC = 4: Turbulent viscosity obtained from TKE - mixing length model |
| Default: 1. |

DELTAB(KSURF,IDIR) | Specified boundary layer thickness on surface KSURF for mixing length model of turbulence. |
| No default values. |
Namelist Input Description

**LIST4**

**Initial Profile, Turbulence Information**

**ITRANS**
Flag which tells whether transition turbulence model logic is used.

- **ITRANS = 0**: No transitional model is used
- **ITRANS ≠ 0**: Transitional model is used

Default: 0.

**IBLT**
Flag which tells whether boundary layer thickness is input or calculated dynamically.

- **IBLT = 0**: Boundary layer thickness is input
- **IBLT ≠ 0**: Boundary layer thickness is dynamically calculated.

Default: 0.

**TKEINF**
Freestream turbulent kinetic energy.
Default: 0.0.

**NPROF(LP)**
Number of initial profiles generated for each loop in Y or Z-direction.
Default: 10*0.

**IDIRP**
Basic direction for initial profile generation.

- **IDIRP = 1**: Y-direction
- **IDIRP = 2**: Z-direction

Default: 2.

**LIST5**

**File Output Information**

**IVARPR(I)**
Index of variables to be printed. Needed only for three-dimensional flow.

- **IVARPR(I) = 0**: No print
- **IVARPR(I) = 1**: Print every IPRINT steps
- **IVARPR(I) = 2**: Print every JPRINT steps

- **I = 1**: UVEL
- **I = 2**: WVEL
- **I = 3**: Density
- **I = 4**: Enthalpy
- **I = 5**: Turbulent kinetic energy
- **I = 6**: Turbulent dissipation

Default: 5*1, 2*0, 3*1, 7*0, 1, 3*0, 1, 3*0.
Namelist Input Description

**LIST5**

**Initial Profile, Turbulence Information**

<table>
<thead>
<tr>
<th>I</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>Pressure</td>
</tr>
<tr>
<td>9</td>
<td>Temperature</td>
</tr>
<tr>
<td>10</td>
<td>Mach No.</td>
</tr>
<tr>
<td>11</td>
<td>Mach No. indicator</td>
</tr>
<tr>
<td>12</td>
<td>Stagnation temperature</td>
</tr>
<tr>
<td>13</td>
<td>Stagnation pressure</td>
</tr>
<tr>
<td>14</td>
<td>Pressure coefficient</td>
</tr>
<tr>
<td>15</td>
<td>Laminar viscosity</td>
</tr>
<tr>
<td>16</td>
<td>Mixing length</td>
</tr>
<tr>
<td>17</td>
<td>Turbulent viscosity</td>
</tr>
<tr>
<td>18</td>
<td>Effective viscosity</td>
</tr>
<tr>
<td>19</td>
<td>Dissipation function</td>
</tr>
<tr>
<td>20</td>
<td>Cell Reynolds number in Y-direction</td>
</tr>
<tr>
<td>21</td>
<td>Cell Reynolds number in Z-direction</td>
</tr>
<tr>
<td>22</td>
<td>ISS and JBOUND</td>
</tr>
<tr>
<td>23</td>
<td>Heat transfer coefficient, skin friction coefficient and heat transfer rate</td>
</tr>
<tr>
<td>24</td>
<td>Boundary layer thickness</td>
</tr>
<tr>
<td>25</td>
<td>Cross-sectional average of flow properties</td>
</tr>
</tbody>
</table>

**IPL0T**

Marching station interval for storage of plotting information.

- **IPL0T = 0:** No plotting
- **IPL0T ≠ 0:** Store plotting information every IPL0T station

Default: 0.

**IPR0NT**

Primary marching station interval for printing.

Default: 1.

**JPR0NT**

Secondary marching station interval for printing.

Default: 1.
### Namelist Input Description

**Reset Information for Cross-Section**

<table>
<thead>
<tr>
<th>LISTR</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFBW</td>
<td>See LIST3</td>
</tr>
<tr>
<td>IBOUND</td>
<td>See LIST1</td>
</tr>
<tr>
<td>IEQBC</td>
<td>See LIST1</td>
</tr>
<tr>
<td>JEQBC</td>
<td>See LIST1</td>
</tr>
<tr>
<td>DTDN</td>
<td>See LIST1</td>
</tr>
<tr>
<td>TWALL</td>
<td>See LIST1</td>
</tr>
<tr>
<td>PRESS</td>
<td>See LIST1</td>
</tr>
</tbody>
</table>
Error Conditions in the PEPSIN Computer Code

Failure of the PEPSIN computer code to successfully execute a runstream can occur because of either inconsistent or incorrect input data or because of an attempt to apply the PEPSIN code to a case where the physics violate the assumptions inherent in the code. This section will address only the former mode of failure. Avoidance of the latter failure mode is dependent primarily on the users understanding of the basic physics of the case he is going to run, and the degree to which the PEPSIN code can be expected to model the physics.

One method of discussing the inconsistent or incorrect input data mode of failure is by examining the possible failures in the various subroutines. Since the individual subroutines are responsible for separate tasks during the execution of a run, (e.g. overall control of the program geometry generation, etc.), this technique will in essence outline the possible failure modes as the tasks are performed. Discussion will occur in the same order as the run is executed.

SUBROUTINE RESTRT

There are two modes by which SUBROUTINE RESTRT can fail. Both involve improper use of the restart file. A message, RESTART INFORMATION REQUESTED AT (IRSTIN marching number) BUT STORED INFORMATION AT SEQUENCE (NFILE) IS AT STATION (Station number). This message occurs because the marching station number read off the NFILEth restart file does not match the input value of IRSTIN. The corrective action is to make NFILE and IRSTIN consistent with each other. Another possible mode of failure occurs when NFILE exceeds the number of files on the restart device, JRSTIN, in which case an END OF INFORMATION (or analogous statement) will appear in the day file. The corrective action is to recheck the input value of NFILE. If JRSTIN ≠ JRSTOT, the value of NFILE is the number of the restart on device JRSTIN.

SUBROUTINE INPUTS

There are two failure modes in SUBROUTINE INPUTS. On the first case, the message NS = (input value of NS) GREATER THAN NSMAX = (dimension of X vector) will be printed if the number of marching stations exceeds the dimensional value of X, the streamwise locations. The corrective action is
to lower the value of NS. The second failure mode occurs when the Buleev turbulence model is specified for a two-dimensional case. Since this model is not applicable to two-dimensional cases, the message CANNOT USE BULEEV TURBULENCE MODEL IN TWO-DIMENSIONAL FLOW is printed. The corrective action is to specify an alternate turbulence model.

SUBROUTINE SETUP

If grid point characteristics indicator, IFBW, is not correctly specified, SUBROUTINE SETUP can fail. When the message FAILURE IN SETUP DUE TO INCORRECT IFBW is printed, the corrective action is to recheck IFBW specified in INPUT DATA. IFBW should contain an information on the computational domain boundary which is specified by boundary indicator.

SUBROUTINE GEOTRB

At present SUBROUTINE GEOTRB is coded to calculate metric information for values of IGEOM = 1, 2, 3, 10 and 11. Values of IGEOM 4-9 are left for various coordinates that may be coded in the future. Input value of IGEOM = 4-9 will result in the message INVALID OPTION IN GEOTRB. The corrective action is to either change the value of IGEOM or to code in a new option. For IGEOM options 10 and 11 (conformal-Cartesian cross-section and conformal-axisymmetric cross-section) the metric information is externally generated by the ADD computer code. In this case, logical file units JDRUM and KDRUM must be defined. JDRUM contains the ADD code data which is then interpolated onto the PEPSIN mesh system. If the PEPSIN values of the streamwise coordinate is less than the first value of the ADD code streamwise coordinate no streamwise interpolation is possible and the message FAILURE IN GEOTRB - SQ12 = (PEPSIN position) SQ1 = (first ADD code position) SQ2 = (second ADD code position). The corrective action is to increase the value of XENTR (the first PEPSIN position) to a value greater than SQ1. On the other hand, if the value of a PEPSIN streamwise coordinate exceeds the last streamwise position generated by the ADD code and END OF INFORMATION message will appear in the day file. The corrective action is to either rerun the ADD code such that the maximum PEPSIN streamwise coordinate does not exceed the maximum ADD code streamwise coordinate or to reduce the maximum PEPSIN streamwise coordinate to an acceptable value.
SUBROUTINE INTEBC

SUBROUTINE INTEBC performs a two-dimensional linear interpolation of the transpiration schedules on both X-Y planes at X-Z planes. If the number of streamwise stations on a surface at which data is input exceeds 15 (the dimensional size of the data arrays) a message FAILURE IN INTEBC VALUE OF NPTSX (surface number IBC) = (value of NPTSX(IBC) EXCEEDS DIMENSION LIMITS OF 15 is printed. The corrective action is either to updimension NPTSX and associated variables or to decrease the value of NPTSX. Likewise, in the Y or Z direction data can be input at up to 15 locations. If the value of NPTSYZ exceeds 15, the message FAILURE IN INTEBC VALUE OF NPTSYZ (streamwise location, surface number) = (value of NPTSYZ) EXCEEDS DIMENSION LIMITS OF 15 is printed. The corrective action is either to updimension NPTSYZ and associated variables or to decrease the value of NPTSYZ.

SUBROUTINE QUICK

If the choice of boundary conditions is incorrectly made, it is possible that a singular matrix will result. This will manifest itself in SUBROUTINE QUICK in an attempt to divide by zero. The corrective action is to re-evaluate the choice of input boundary conditions to determine the source of the singularity. An example of an improper choice of a boundary condition set would be to choose as boundary conditions the three no-slip conditions for the three momenta equations, the normal pressure condition for the continuity equation and the normal momentum equation for the enthalpy equation. In this case, the enthalpy does not appear in any of the boundary conditions, and hence a singular matrix would result.

SUBROUTINE CROSEC

Often, if a case is not going to successfully run, the code will cease operation in SUBROUTINE CROSEC. This will occur because of the existence of a negative temperature in which case the Mach number calculation will fail in SQRT. There can be many reasons for this failure mode. Usually, however, it can be related to inadequate numerical resolution of the physical processes that are occurring. For instance, a lack of transverse grid points might lead to large oscillations in the pressure or too large a streamwise step in the region where a wall inclination is rapidly changing might result in a temperature becoming negative. Sometimes it is difficult to know a priori...
what grid resolution is necessary for a given problem. Usually, experimentation with two-dimensional cases can ironically provide some guidelines for three-dimensional cases. This in addition with the users' overall experience with the code and his understanding of the physical processes will usually provide the means of resolving the above problem.

**SUBROUTINE WHERE**

There are two modes by which SUBROUTINE WHERE can fail. For the first case, the message **PROBLEM IN SUBROUTINE WHERE MSECY = m JX=n1 JY=n2 KZ=n3** will be printed where n1, n2 and n3 refers to x, y and z location respectively if MSECY is not correctly specified, i.e., not consistent with the number of Y-perspective subsections. The corrective action is to increase the value of MSECY corresponding to the number of Y-perspective subsections. The second failure mode occurs when MSECZ is not consistent with the number of Z-perspective subsections. In this case, the message **PROBLEM IN SUBROUTINE WHERE MSECZ = m JX=u1 JY=n2 KZ=n3** is printed with n1, n2 and n3 referring to x, y and z location respectively. The corrective action is to increase the value of MSECZ corresponding to the number of Z-perspective subsections.
# PEPSIN FORTRAN Variables

<table>
<thead>
<tr>
<th>FORTRAN SYMBOL</th>
<th>COMMON BLOCK</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACON</td>
<td>LAWW</td>
<td>CONSTANT IN ARGUMENT OF EXPONENTIAL FUNCTION FOR TRANSITIONAL MODEL</td>
</tr>
<tr>
<td>AG(NN,9,2)</td>
<td>OPER</td>
<td>DIFFERENCE WEIGHTS IN PHYSICAL COORDINATES</td>
</tr>
<tr>
<td>AGEO</td>
<td>GEOM</td>
<td>COEFFICIENTS OF POLYNOMIAL FIT FOR BOUNDARY SHAPE</td>
</tr>
<tr>
<td>AG1D(9)</td>
<td>FGCOM</td>
<td>TEMPORARY STORAGE ARRAY OF METRIC INFORMATION</td>
</tr>
<tr>
<td>AG1P</td>
<td>FGCOM</td>
<td>TEMPORARY STORAGE ARRAY OF METRIC INFORMATION</td>
</tr>
<tr>
<td>AG2D(9)</td>
<td>FGCOM</td>
<td>TEMPORARY STORAGE ARRAY OF METRIC INFORMATION</td>
</tr>
<tr>
<td>AHP(5)</td>
<td>FGCOM</td>
<td>TEMPORARY STORAGE ARRAY OF METRIC INFORMATION</td>
</tr>
<tr>
<td>AH1D(5,9)</td>
<td>FGCOM</td>
<td>TEMPORARY STORAGE ARRAY OF METRIC INFORMATION</td>
</tr>
<tr>
<td>AIE(NN,7)</td>
<td>PREFILE</td>
<td>INITIAL PROFILE ARRAY</td>
</tr>
<tr>
<td>AM(NCPLD,3ANCPLD+1)COM</td>
<td></td>
<td>UTILITY MATRIX USED IN BLOCK MATRIX INVERSION</td>
</tr>
<tr>
<td>AMACRI</td>
<td>SUPER</td>
<td>MACH NUMBER CRITERION USED IN LOCATING SONIC LINE</td>
</tr>
<tr>
<td>AN(NEQS,NN)</td>
<td>LIN</td>
<td>STORAGE FOR LINEARIZATION COEFFICIENTS OF X - DERIVATIVES</td>
</tr>
<tr>
<td>AP(10)</td>
<td>GEOM</td>
<td>AMPLIFICATION RATE OF MARCHING STEP SIZE</td>
</tr>
<tr>
<td>APLUS</td>
<td>LAWW</td>
<td>CONSTANT IN ARGUMENT OF EXPONENTIAL FUNCTION FOR VAN DRIEST DAMPING FORMULA</td>
</tr>
<tr>
<td>ASW(20,2)</td>
<td>BOUND</td>
<td>COEFFICIENTS OF POLYNOMIAL FIT FOR SWITCHING THE BOUNDARY SURFACE TYPE</td>
</tr>
<tr>
<td>AVISC(2,NEQS)</td>
<td>VISC</td>
<td>COEFFICIENT USED IN ARTIFICIAL DAMPING</td>
</tr>
<tr>
<td>FEIA</td>
<td>REF</td>
<td>ANGLE OF ATTACK</td>
</tr>
<tr>
<td>BGEOM</td>
<td>GEOM</td>
<td>COEFFICIENTS OF POLYNOMIAL FIT FOR BOUNDARY SHAPE</td>
</tr>
<tr>
<td>BLOCK1(IADD3)</td>
<td></td>
<td>STORAGE FOR ADD CODE METRIC INFORMATION EQUIVALENCED TO C(1,1,1)</td>
</tr>
<tr>
<td>BLOCK2(IADD3)</td>
<td></td>
<td>STORAGE FOR ADD CODE METRIC INFORMATION EQUIVALENCED TO C(1,1,NH/2+1)</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>BLTH(NN,5,4)</td>
<td>LAWW</td>
<td>DYNAMICALLY DETERMINED BOUNDARY LAYER THICKNESS</td>
</tr>
<tr>
<td>BSW(20,2)</td>
<td>BOUND</td>
<td>COEFFICIENTS OF POLYNOMIAL FIT FOR SWITCHING THE BOUNDARY SURFACE TYPE</td>
</tr>
<tr>
<td>BWD</td>
<td>LIN</td>
<td>CRANK-NICHOLSON FACTOR</td>
</tr>
<tr>
<td>BWDI</td>
<td>LIN</td>
<td>INVERSE OF BWD</td>
</tr>
<tr>
<td>C(NN,NCPLD,NN)</td>
<td>CCOM</td>
<td>BLOCK DIAGONAL MATRIX ELEMENTS</td>
</tr>
<tr>
<td>CDUM(NDIM)</td>
<td></td>
<td>TEMPORARY STORAGE ARRAY TO ROTATE DATA FROM COLUMNS TO ROWS AND VICE VERSA - EQUIVALENCED TO C(1,1,1) NDIM = MZVAR A MLEVEL A NN</td>
</tr>
<tr>
<td>CFP(4,10)</td>
<td>PROFILE</td>
<td>SKIN FRICTION COEFFICIENT</td>
</tr>
<tr>
<td>CGEO</td>
<td>GEOM</td>
<td>COEFFICIENTS OF POLYNOMIAL FIT FOR BOUNDARY SHAPE</td>
</tr>
<tr>
<td>CHUINF</td>
<td>VISC</td>
<td>CONSTANT IN TURBULENT VISCOITY MODEL</td>
</tr>
<tr>
<td>CONGEO(11,NN)</td>
<td>GEOM</td>
<td>COORDINATE TRANSFORMATION INFORMATION</td>
</tr>
<tr>
<td>CONVDR</td>
<td>UNITS</td>
<td>CONVERSION FACTOR IN GOING FROM DEGREES TO RADIANS</td>
</tr>
<tr>
<td>CONVRD</td>
<td>UNITS</td>
<td>CONVERSION FACTOR IN GOING FROM RADIANS TO DEGREES</td>
</tr>
<tr>
<td>COOR(NN,4)</td>
<td>GEOM</td>
<td>PHYSICAL COORDINATE INFORMATION WITH RESPECT TO ABSOLUTE ORIGIN AT N+1ST STREAMWISE LOCATION</td>
</tr>
<tr>
<td>COORN(NN,4)</td>
<td>GEOM</td>
<td>PHYSICAL COORDINATE INFORMATION WITH RESPECT TO ABSOLUTE ORIGIN AT NTH STREAMWISE STATION</td>
</tr>
<tr>
<td>CPINF</td>
<td>FREE</td>
<td>FREE STREAM SPECIFIC HEAT</td>
</tr>
<tr>
<td>CPREF</td>
<td>REF</td>
<td>REFERENCE PRESSURE COEFFICIENT</td>
</tr>
<tr>
<td>CPREFI</td>
<td>REF</td>
<td>INVERSE OF REFERENCE PRESSURE COEFFICIENT</td>
</tr>
<tr>
<td>CRITU</td>
<td>OPER</td>
<td>CRITICAL VELOCITY USED FOR FLARE APPROXIMATION</td>
</tr>
<tr>
<td>CSOLN(NCPLD,NN)</td>
<td>CCOM</td>
<td>SOLUTION TO BLOCK TRI-DIAGONAL MATRIX INVERSION</td>
</tr>
<tr>
<td>CSW(20,2)</td>
<td>BOUND</td>
<td>COEFFICIENTS OF POLYNOMIAL FIT FOR SWITCHING THE BOUNDARY SURFACE TYPE</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>CTWO</td>
<td>REF</td>
<td>NUMERICAL CONSTANT IN GOVERNING EQUATION</td>
</tr>
<tr>
<td>CXI(NDIFM)</td>
<td>ADDR</td>
<td>STORAGE FOR FIRST DERIVATIVE DIFFERENCE WEIGHTS</td>
</tr>
<tr>
<td>CXXI(NDIFM)</td>
<td>ADDR</td>
<td>STORAGE FOR SECOND DERIVATIVE DIFFERENCE WEIGHTS</td>
</tr>
<tr>
<td>C1(NN)</td>
<td>CCOM</td>
<td>SUBDIAGONAL MATRIX ELEMENTS</td>
</tr>
<tr>
<td>C1SUTH</td>
<td>VISC</td>
<td>COEFFICIENT IN SUTHERLAND'S LAW OF LAMINAR VISCOSITY</td>
</tr>
<tr>
<td>C2(NN)</td>
<td>CCOM</td>
<td>DIAGONAL MATRIX ELEMENTS</td>
</tr>
<tr>
<td>C2SUTH</td>
<td>VISC</td>
<td>COEFFICIENT IN SUTHERLAND'S LAW OF LAMINAR VISCOSITY</td>
</tr>
<tr>
<td>C3(NN)</td>
<td>CCOM</td>
<td>SUPERDIAGONAL MATRIX ELEMENTS</td>
</tr>
<tr>
<td>C4(NN)</td>
<td>CCOM</td>
<td>VECTOR ELEMENTS</td>
</tr>
<tr>
<td>D</td>
<td>VAR</td>
<td>INDEX FOR DIVERGENCE</td>
</tr>
<tr>
<td>DELMAX(NEQS)</td>
<td>EQN</td>
<td>MAXIMUM VALUES OF THE DELTAS</td>
</tr>
<tr>
<td>DELTAB(20,2)</td>
<td>LAW</td>
<td>SPECIFIED BOUNDARY LAYER THICKNESS FOR MIXING LENGTH MODEL OF TURBULENCE</td>
</tr>
<tr>
<td>DELTAP(4,10)</td>
<td>PROFILE</td>
<td>BOUNDARY LAYER THICKNESS</td>
</tr>
<tr>
<td>DELX</td>
<td>GEOM</td>
<td>STEP SIZE IN MARCHING DIRECTION</td>
</tr>
<tr>
<td>DGE0</td>
<td>GEOM</td>
<td>COEFFICIENTS OF POLYNOMIAL FIT FOR BOUNDARY SHAPE</td>
</tr>
<tr>
<td>DIFOP(6)</td>
<td>OPER</td>
<td>DIFFERENCE WEIGHTS IN COMPUTATIONAL COORDINATES</td>
</tr>
<tr>
<td>DL</td>
<td>VAR</td>
<td>STORAGE LEVEL OF DIVERGENCE IN GENERAL PURPOSE STORAGE</td>
</tr>
<tr>
<td>DM1(NCPLD,NCPLD)</td>
<td>CCOM</td>
<td>GENERAL PURPOSE STORAGE ARRAYS FOR BLOCK TRI-DIAGONAL MATRIX INVERSION</td>
</tr>
<tr>
<td>DM2(NCPLD)</td>
<td>CCOM</td>
<td>GENERAL PURPOSE STORAGE ARRAYS FOR BLOCK TRI-DIAGONAL MATRIX INVERSION</td>
</tr>
<tr>
<td>DM3(NCPLD,NCPLD)</td>
<td>CCOM</td>
<td>GENERAL PURPOSE STORAGE ARRAYS FOR BLOCK</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>DS</td>
<td>VAR</td>
<td>INDEX FOR DISSIPATION FUNCTION</td>
</tr>
<tr>
<td>DSL</td>
<td>VAR</td>
<td>STORAGE LEVEL OF DISSIPATION FUNCTION IN GENERAL PURPOSE STORAGE</td>
</tr>
<tr>
<td>DSW(20,2)</td>
<td>BOUND</td>
<td>COEFFICIENTS OF POLYNOMIAL FIT FOR SWITCHING THE BOUNDARY SURFACE TYPE</td>
</tr>
<tr>
<td>D1DN(20,2)</td>
<td>BOUND</td>
<td>SPECIFIED TEMPERATURE GRADIENT NORMAL TO BOUNDARY</td>
</tr>
<tr>
<td>D1DNNW(nn,5,4)</td>
<td>BOUND</td>
<td>STORAGE ARRAY FOR TEMPERATURE GRADIENT NORMAL TO BOUNDARY</td>
</tr>
<tr>
<td>DX</td>
<td>OPER</td>
<td>STEP SIZE IN X-DIRECTION</td>
</tr>
<tr>
<td>DXI</td>
<td>OPER</td>
<td>INVERSE OF DX</td>
</tr>
<tr>
<td>DXMAX(10)</td>
<td>GEOM</td>
<td>MAXIMUM MARCHING STEP SIZE</td>
</tr>
<tr>
<td>DXMIN(10)</td>
<td>GEOM</td>
<td>MINIMUM MARCHING STEP SIZE</td>
</tr>
<tr>
<td>D1(nn,nd,nn)</td>
<td>LIN</td>
<td>STORAGE FOR LINEARIZATION COEFFICIENTS OF Y - DERIVATIVES</td>
</tr>
<tr>
<td>D1L(nn,nd)</td>
<td>LIN</td>
<td>STORAGE OF LINEARIZATION COEFFICIENT FOR TEMPERATURE AND PRESSURE COMPUTATION</td>
</tr>
<tr>
<td>D2(nn,nd,nn)</td>
<td>LIN</td>
<td>STORAGE FOR LINEARIZATION COEFFICIENTS OF Z - DERIVATIVES</td>
</tr>
<tr>
<td>E(nn,ncpld,nn)</td>
<td>ARRAY USED IN MGAUSS ERROR CHECK - EQUIVALENCED TO C(1,1,1)</td>
<td></td>
</tr>
<tr>
<td>EPS</td>
<td>VAR</td>
<td>INDEX FOR DISSIPATION OF TURBULENCE KINETIC ENERGY</td>
</tr>
<tr>
<td>EPSHWF</td>
<td>LAWW</td>
<td>CONVERGENCE CRITERION FOR WALL FUNCTION FORMULATION</td>
</tr>
<tr>
<td>F(14,nn)</td>
<td>ZPLOI</td>
<td>TEMPORARY STORAGE FOR PLOT INFORMATION</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>FACLM(20,2)</td>
<td>LAWW</td>
<td>MULTIPLICATION FACTOR TO BOUNDARY LAYER THICKNESS</td>
</tr>
<tr>
<td>FG(8,3,NN)</td>
<td>METRIC</td>
<td>STORAGE FOR METRIC COEFFICIENTS AND DERIVATIVES</td>
</tr>
<tr>
<td>GAM(3)</td>
<td>GEOM</td>
<td>COORDINATE TRANSFORMATION COEFFICIENT FOR BOUNDARY CONDITIONS</td>
</tr>
<tr>
<td>GAMMA</td>
<td>REF</td>
<td>RATIO OF SPECIFIC HEATS</td>
</tr>
<tr>
<td>GC</td>
<td>UNITS</td>
<td>GRAVITY CONSTANT</td>
</tr>
<tr>
<td>H</td>
<td>VAR</td>
<td>INDEX FOR ENTHALPY</td>
</tr>
<tr>
<td>HFORM</td>
<td>REF</td>
<td>HEAT OF FORMATION</td>
</tr>
<tr>
<td>HREF</td>
<td>FREE</td>
<td>FREE STREAM ENTHALPY</td>
</tr>
<tr>
<td>HREE</td>
<td>REF</td>
<td>REFERENCE ENTHALPY</td>
</tr>
<tr>
<td>HREFI</td>
<td>REF</td>
<td>INVERSE OF REFERENCE ENTHALPY</td>
</tr>
<tr>
<td>IA</td>
<td>CCOM</td>
<td>INDEX REFERRING TO SUBDIAGONAL MATRIX ELEMENTS</td>
</tr>
<tr>
<td>IADDO(NDIFM)</td>
<td>ADDR</td>
<td>ADDRESSES IN THE OPPOSITE DIRECTION</td>
</tr>
<tr>
<td>IADDP(NDIFM)</td>
<td>ADDR</td>
<td>ADDRESSES IN THE PRIMARY DIRECTION</td>
</tr>
<tr>
<td>IADDS1(NN)</td>
<td>ADDRF</td>
<td>ADDRESS FOR POINT LOGIC OF FLUID VARIABLES</td>
</tr>
<tr>
<td>IADDS2(NDIFM,NN)</td>
<td>ADDRF</td>
<td>ADDRESS FOR Y DERIVATIVE OF FLUID VARIABLES</td>
</tr>
<tr>
<td>IADDS3(NDIFM,NN)</td>
<td>ADDRF</td>
<td>ADDRESS FOR Z DERIVATIVE OF FLUID VARIABLES</td>
</tr>
<tr>
<td>IADDS4(NDIFM2,NN)</td>
<td>ADDRF</td>
<td>ADDRESS FOR MIXED DERIVATIVE OF FLUID VARIABLES</td>
</tr>
<tr>
<td>IADDS5(NDIFM2,NN)</td>
<td>ADDRF</td>
<td>ADDRESS FOR MIXED DERIVATIVE OF FLUID VARIABLES</td>
</tr>
<tr>
<td>IADD1</td>
<td>ADD</td>
<td>NO. OF GEOMETRIC VARIABLES USED IN ADD CODE</td>
</tr>
<tr>
<td>IADD2</td>
<td>ADD</td>
<td>NO. OF TRANSVERSE GRIDPOINT USED IN ADD CODE</td>
</tr>
<tr>
<td>IADD3</td>
<td>ADD</td>
<td>RECORD SIZE USED IN ADD CODE</td>
</tr>
<tr>
<td>IADI</td>
<td>SWEEP</td>
<td>ADI SWEEP DIRECTION</td>
</tr>
<tr>
<td>IADIM1</td>
<td>SWEEP</td>
<td>IADI - 1</td>
</tr>
<tr>
<td>IAP(10)</td>
<td>GEOM</td>
<td>MARCHING STEP INDEX AT WHICH AP,DXMIN,DXMAX</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IB</td>
<td>CCOM</td>
<td>INDEX REFERRING TO DIAGONAL MATRIX ELEMENTS</td>
</tr>
<tr>
<td>IBC</td>
<td>BOUND</td>
<td>INDEX FOR BOUNDARY SURFACE IDENTIFICATION</td>
</tr>
<tr>
<td>IBCP(10)</td>
<td>PROFILE</td>
<td>BASIC SURFACE FOR INITIAL PROFILE GENERATION</td>
</tr>
<tr>
<td>IBCPLP</td>
<td>PROFILE</td>
<td>BOUNDARY SURFACE NUMBER IN EACH LOOP WHERE INITIAL BOUNDARY LAYER PROFILE IS REQUIRED.</td>
</tr>
<tr>
<td>IBCPLP = 1 OR 2 IF IDIRP = 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBCPLP = 3 OR 4 IF IDIRP = 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IBLT</td>
<td>LAWW</td>
<td>FLAG WHICH TELLS WHETHER BOUNDARY LAYER THICKNESS IS INPUT OR CALCULATED DYNAMICALLY</td>
</tr>
<tr>
<td>IBOUND(20,2)</td>
<td>BOUND</td>
<td>BOUNDARY SURFACE TYPE INDICATOR</td>
</tr>
<tr>
<td>IBRKMX(NN,5)</td>
<td>BOUND</td>
<td>UPPER BOUNDARY OF AN EMBEDDED SOLID BODY EXPRESSED IN TERMS OF GRID POINT INDEX ON EACH Y-COORDINATE LINE AT UP TO 5 STREAMWISE LOCATIONS.</td>
</tr>
<tr>
<td>IEDGE</td>
<td>BOUND</td>
<td>SENTINEL WHICH SPECIFIES EITHER LEADING EDGE WHERE SOLID BODY STARTS OR TRAILING EDGE WHERE SOLID BODY TERMINATES.</td>
</tr>
<tr>
<td>IBRKMN(NN,5)</td>
<td>BOUND</td>
<td>LOWER BOUNDARY OF AN EMBEDDED SOLID BODY EXPRESSED IN TERMS OF GRID POINT INDEX ON EACH Y-COORDINATE LINE AT UP TO 5 STREAMWISE LOCATIONS.</td>
</tr>
<tr>
<td>IC</td>
<td>CCOM</td>
<td>INDEX REFERRING TO SUPERDIAGONAL MATRIX ELEMENTS</td>
</tr>
<tr>
<td>ICDC(NN+1,2)</td>
<td>CDC</td>
<td>RECORD INDEX FOR READMS AND WRITEMS MASS STORAGE DEVICES-CDC COMPUTER ONLY</td>
</tr>
<tr>
<td>ICOMP</td>
<td>REST</td>
<td>FLAG FOR COMPUTER OPTIONS</td>
</tr>
<tr>
<td>ICONS(3,NEOS)</td>
<td>OPER</td>
<td>FLAG FOR CONVECTIVE FORMULATION BASED ON MACH NUMBER AND EQUATION</td>
</tr>
</tbody>
</table>

68
<table>
<thead>
<tr>
<th>FORTRAN SYMBOL</th>
<th>COMMON BLOCK</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICORD</td>
<td>GEOM</td>
<td>FLAG FOR COORDINATE TRANSFORMATION OPTIONS</td>
</tr>
<tr>
<td>ICPLD(NEQS,2)</td>
<td>EQN</td>
<td>COUPLED EQUATION SENTINEL</td>
</tr>
<tr>
<td>ID</td>
<td>CCOM</td>
<td>INDEX REFERRING TO VECTOR MATRIX ELEMENTS</td>
</tr>
<tr>
<td>IDIF(40)</td>
<td>OPER</td>
<td>INDEX FOR TYPE OF DIFFERENCING OF BOUNDARY CONDITIONS</td>
</tr>
<tr>
<td>IDMPY</td>
<td>DMP</td>
<td>DUMP LINE NO. IN Y DIRECTION</td>
</tr>
<tr>
<td>IDMPZ</td>
<td>DMP</td>
<td>DUMP LINE NO. IN Z DIRECTION</td>
</tr>
<tr>
<td>IDUM2(NN)</td>
<td>ADDRG</td>
<td>INDICATOR OF THE POINT IN DIFFERENCE MOLECULE WHERE Y DERIVATIVE IS TAKEN</td>
</tr>
<tr>
<td>IDUM3(NN)</td>
<td>ADDRG</td>
<td>INDICATOR OF THE POINT IN DIFFERENCE MOLECULE WHERE Z DERIVATIVE IS TAKEN</td>
</tr>
<tr>
<td>IEQ</td>
<td>EQN</td>
<td>EQUATION NUMBER INDEX</td>
</tr>
<tr>
<td>IEQBC(20,2,NEQS)</td>
<td>BOUND</td>
<td>SPECIFIED BOUNDARY CONDITION FOR WALL</td>
</tr>
<tr>
<td>IEQNUM(NEQS)</td>
<td>EQN</td>
<td>IDENTIFICATION NUMBER OF EQUATIONS TO BE SOLVED</td>
</tr>
<tr>
<td>IFBW(NN,NN)</td>
<td>GRID</td>
<td>INDICATOR WHICH TYPES GRID POINTS IN THE COMPUTATIONAL CROSS-SECTION PLANE.</td>
</tr>
<tr>
<td>IFLARE</td>
<td>OPER</td>
<td>SENTINEL WHICH TELLS IF FLARE OPTION IS USED</td>
</tr>
<tr>
<td>IGDMP</td>
<td>DMP</td>
<td>FLAG FOR DUMP OPTIONS</td>
</tr>
<tr>
<td>IGEOM</td>
<td>GEOM</td>
<td>FLAG FOR COORDINATE SYSTEM OPTIONS</td>
</tr>
<tr>
<td>IH</td>
<td>GRID</td>
<td>UPPER BOUNDARY POINT ON THE LINE WHERE IMPLICIT SOLUTION IS OBTAINED</td>
</tr>
<tr>
<td>IHSTAG</td>
<td>EQN</td>
<td>FLAG FOR ENERGY EQUATION OPTIONS</td>
</tr>
<tr>
<td>IJ</td>
<td>DIECOM</td>
<td>COLUMN OR ROW NUMBER ON WHICH CALCULATION IS BEING MADE</td>
</tr>
<tr>
<td>IL</td>
<td>GRID</td>
<td>LOWER BOUNDARY POINT ON THE LINE WHERE IMPLICIT SOLUTION IS OBTAINED</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IMIXL</td>
<td>LAWW</td>
<td>FLAG FOR MIXING LENGTH OPTIONS</td>
</tr>
<tr>
<td>IND</td>
<td>VAR</td>
<td>FLAG WHICH TELLS IF MESH POINT BELONGS TO SUPersonic OR SUBsonic REGION</td>
</tr>
<tr>
<td>INDC(NN)</td>
<td>DIFCOM</td>
<td>MACH NUMBER INDICATOR</td>
</tr>
<tr>
<td>INDL</td>
<td>VAR</td>
<td>STORAGE OF MACH NUMBER INDICATOR IN GENERAL PURPOSE STORAGE</td>
</tr>
<tr>
<td>INH12</td>
<td>ADD</td>
<td>SENTINEL USED TO DETERMINE FORMULATION USED IN CALCULATION OF TRANSVERSE DERIVATIVE OF H1</td>
</tr>
<tr>
<td>INH21</td>
<td>ADD</td>
<td>SENTINEL USED TO DETERMINE FORMULATION USED IN CALCULATION OF STREAMWISE DERIVATIVE OF H2</td>
</tr>
<tr>
<td>INH31</td>
<td>ADD</td>
<td>SENTINEL USED TO DETERMINE FORMULATION USED IN CALCULATION OF STREAMWISE DERIVATIVE OF H3</td>
</tr>
<tr>
<td>INH32</td>
<td>ADD</td>
<td>SENTINEL USED TO DETERMINE FORMULATION USED IN CALCULATION OF TRANSVERSE DERIVATIVE OF H3</td>
</tr>
<tr>
<td>IOPTWF</td>
<td>LAWW</td>
<td>SENTINEL WHICH DETERMINES WALL FUNCTION FORMULATION</td>
</tr>
<tr>
<td>IOPTYZ(3,NEQS,2)</td>
<td>OPER</td>
<td>FLAG FOR DIFFERENCING FORMULATION BASED ON MACH NUMBER, EQUATION, AND ADI DIRECTION</td>
</tr>
<tr>
<td>IPLOI</td>
<td>CIO</td>
<td>MARCHING STATION INTERVAL FOR STORAGE OF PLOTTING INFORMATION</td>
</tr>
<tr>
<td>IPRINT</td>
<td>CIO</td>
<td>PRIMARY MARCHING STATION INTERVAL FOR PRINTING</td>
</tr>
<tr>
<td>IPROF</td>
<td>PFILE</td>
<td>FLAG FOR INITIAL PROFILE OPTIONS</td>
</tr>
<tr>
<td>IPRTE</td>
<td>GASL</td>
<td>FLAG WHICH DETERMINES EQ. OF STATE FORMULATION</td>
</tr>
<tr>
<td>IRSTIN</td>
<td>REST</td>
<td>STREAMWISE STATION NUMBER FOR RESTART</td>
</tr>
<tr>
<td>IRSTOT</td>
<td>REST</td>
<td>STREAMWISE INTERVAL FOR SAVING RESTART INFORMATION</td>
</tr>
<tr>
<td>ISONIC</td>
<td>SUPER</td>
<td>FLAG FOR SONIC LINE INTERPOLATION LOGIC</td>
</tr>
</tbody>
</table>

70
<table>
<thead>
<tr>
<th>FORTRAN SYMBOL</th>
<th>COMMON BLOCK</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISS(NN,5,4)</td>
<td>SUPER</td>
<td>GRID POINT LOCATION OF SONIC LINE</td>
</tr>
<tr>
<td>ISSHFT</td>
<td>SUPER</td>
<td>INDEX USED TO DETERMINE IF SONIC LINE IS LAST SUBSONIC POINT OR FIRST SUPERS</td>
</tr>
<tr>
<td>ISTART</td>
<td>METRIC</td>
<td>INITIAL CONDITION INDEX</td>
</tr>
<tr>
<td>ISURF(4)</td>
<td>BOUND</td>
<td>LOCATION OF FOUR BOUNDARIES AS IDENTIFIED BY NSURF IN TERMS OF GRID POINT INDEX. ISURF(1) AND ISURF(2) ARE Y-INDICES OF NSURF(1) AND NSURF(2) RESPECTIVELY, WHEREAS ISURF(3) AND ISURF(4) ARE Z-INDICES OF NSURF(3) AND NSURF(4) RESPECTIVELY.</td>
</tr>
<tr>
<td>ISW</td>
<td>UNIVAC</td>
<td>SENTINEL FOR WORD ADDRESSABLE OR SECTOR-ORIENTED MASS STORAGE DEVICE - UNIVAC ONLY</td>
</tr>
<tr>
<td>ISYM</td>
<td>ZPLOT</td>
<td>SYMMETRY OPTION FOR PLOTS</td>
</tr>
<tr>
<td>ITRANS</td>
<td>LAWW</td>
<td>FLAG WHICH TELLS WHETHER TRANSITION TURBULENCE MODEL LOGIC IS USED</td>
</tr>
<tr>
<td>IUNITS</td>
<td>UNITS</td>
<td>FLAG USED TO DETERMINE SET OF DIMENSION UNITS USED</td>
</tr>
<tr>
<td>IVARNO(NEQS)</td>
<td>EQN</td>
<td>IDENTIFICATION NUMBER OF DEPENDENT VARIABLES</td>
</tr>
<tr>
<td>IVARPR(25)</td>
<td>PRNT</td>
<td>INDEX OF VARIABLES TO BE PRINTED</td>
</tr>
<tr>
<td>IVISC</td>
<td>VISC</td>
<td>FLAG FOR VISCOSITY OPTIONS</td>
</tr>
<tr>
<td>IWALF</td>
<td>VISC</td>
<td>SENTINEL WHICH DETERMINES IF WALL FUNCTION LOGIC IS NEEDED IN THE CALCULATION OF WALL VISCOSITY</td>
</tr>
<tr>
<td>IWR</td>
<td>0</td>
<td>SENTINEL FOR NAMELIST REST PRINT</td>
</tr>
<tr>
<td>IYGD(ND1FM)</td>
<td>ADDR</td>
<td>GEOMETRY ADDRESSES</td>
</tr>
<tr>
<td>IZCT(3)</td>
<td>ZEX1</td>
<td>RELATIVE UNIT NO. FOR VIRTUAL MEMORY STORAGE</td>
</tr>
<tr>
<td>I11G(8,3)</td>
<td>FGCOM</td>
<td>INDEX NEEDED IN THE CALCULATION OF METRIC</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IZIG(8,3)</td>
<td>FGCOM</td>
<td>INDEX NEEDED IN THE CALCULATION OF METRIC INFORMATION</td>
</tr>
<tr>
<td>JA(5)</td>
<td>ADDR</td>
<td>SHIFT LOGIC INDEX</td>
</tr>
<tr>
<td>JADDO(NDIFM)</td>
<td>ADDR</td>
<td>ADDRESSES IN THE OPPOSITE DIRECTION</td>
</tr>
<tr>
<td>JADDP(NDIFM)</td>
<td>ADDR</td>
<td>ADDRESSES IN THE PRIMARY DIRECTION</td>
</tr>
<tr>
<td>JBOUND(NN,5,4)</td>
<td>BOUND</td>
<td>BOUNDARY TYPE INDICATOR AT EACH POINT ON BOUNDARY</td>
</tr>
<tr>
<td>JDMAX</td>
<td>EQN</td>
<td>Y GRID POINT LOCATION OF MAXIMUM DELTA</td>
</tr>
<tr>
<td>JDRUM</td>
<td>CIO</td>
<td>ADD CODE DEVICE</td>
</tr>
<tr>
<td>JDUM</td>
<td>DIFCOM</td>
<td>INDEX DENOTING RELATIVE POINT ABOUT WHICH DERIVATIVE IS LOCATED</td>
</tr>
<tr>
<td>JEQBC(20,2 NEQS)</td>
<td>BOUND</td>
<td>SPECIFIED BOUNDARY CONDITION FOR NON-WALL</td>
</tr>
<tr>
<td>JEON(NEQS,2)</td>
<td>EQN</td>
<td>EXTERNAL EQUATION NUMBER</td>
</tr>
<tr>
<td>JGSTOK</td>
<td>LIN</td>
<td>VALUE OF JG NEEDED BY SUBROUTINE EOS</td>
</tr>
<tr>
<td>JMIN(LP,NN)</td>
<td>BOUND</td>
<td>LOWER GRID POINT INDEX LIMIT OF EACH COMPUTATIONAL LOOP(LP) ON EACH OF NN Y-COORDINATE LINES</td>
</tr>
<tr>
<td>JPLOI</td>
<td>CIO</td>
<td>DEVICE FOR PLOTTING</td>
</tr>
<tr>
<td>JPRINT</td>
<td>CIO</td>
<td>SECONDARY MARCHING STATION INTERVAL FOR PRINTING</td>
</tr>
<tr>
<td>JPROF(4)</td>
<td>PROFILE</td>
<td>SENTINEL FOR BOUNDARY VALUES DURING INITIAL PROFILE GENERATION</td>
</tr>
<tr>
<td>JRSTIN</td>
<td>REST</td>
<td>LOGICAL FILE FROM WHICH RESTART INFORMATION IS READ</td>
</tr>
<tr>
<td>JRSTOT</td>
<td>REST</td>
<td>LOGICAL FILE ON WHICH RESTART INFORMATION IS WRITTEN</td>
</tr>
<tr>
<td>JVAR(NEQS,2)</td>
<td>EQN</td>
<td>VARIABLE NUMBER ASSOCIATED WITH EACH EQUATION DURING AN ADI SWEEP</td>
</tr>
</tbody>
</table>

72
<table>
<thead>
<tr>
<th>FORTRAN SYMBOL</th>
<th>COMMON BLOCK</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>JWR(5)</td>
<td>CIO</td>
<td>SENITMEL FOR NAMELIST PRINT OPTION</td>
</tr>
<tr>
<td>JX</td>
<td>OPER</td>
<td>RELATIVE MARCHING STATION COUNTER</td>
</tr>
<tr>
<td>JXDUM</td>
<td>OPER</td>
<td>ABSOLUTE MARCHING STATION COUNTER</td>
</tr>
<tr>
<td>JXDUMP</td>
<td>DMP</td>
<td>MARCHING STATION WHEN DUMP OUTPUT IS REQUESTED</td>
</tr>
<tr>
<td>KA(5)</td>
<td>ADDR</td>
<td>INDICIES NECESSARY TO CALCULATE GEOMETRIC GROUPINGS</td>
</tr>
<tr>
<td>KMAX</td>
<td>EQN</td>
<td>Z GRID POINT LOCATION OF MAXIMUM DELTA</td>
</tr>
<tr>
<td>KDRUM</td>
<td>CIO</td>
<td>DEVICE FOR TEMPORARY STORAGE - USED IN GEOMETRY GENERATION</td>
</tr>
<tr>
<td>KMAX(LP,NN)</td>
<td>BOUND</td>
<td>UPPER GRID POINT INDEX LIMIT OF EACH COMPUTATIONAL LOOP(LP) ON EACH OF NN Z-COORDINATE LINES</td>
</tr>
<tr>
<td>KMIN(LP,NN)</td>
<td>BOUND</td>
<td>LOWER GRID POINT INDEX LIMIT OF EACH COMPUTATIONAL LOOP(LP) ON EACH OF NN Z-COORDINATE LINES</td>
</tr>
<tr>
<td>JMAX(LP,NN)</td>
<td>BOUND</td>
<td>UPPER GRID POINT INDEX LIMIT OF EACH COMPUTATIONAL LOOP(LP) ON EACH OF NN Y-COORDINATE LINES</td>
</tr>
<tr>
<td>LADD(3)</td>
<td>BOUND</td>
<td>ADDRESSES FOR BOUNDARY CONDITIONS</td>
</tr>
<tr>
<td>LBRKY(10)</td>
<td>BOUND</td>
<td>BREAK-OFF POINTS ON THE SPANWISE(Z-DIRECTION) COMPUTATIONAL BOUNDARY TO BE USED TO DETERMINE THE SURFACE NUMBER OF Y-PERSPECTIVE SEGMENTED BOUNDARY IN SUBROUTINE WHERE.</td>
</tr>
<tr>
<td>LBRKZ(10)</td>
<td>BOUND</td>
<td>BREAK-OFF POINTS ON THE TRANSVERSE(Y-DIRECTION) COMPUTATIONAL BOUNDARY TO BE USED TO DETERMINE THE SURFACE NUMBER OF Z-PERSPECTIVE SEGMENTED BOUNDARY IN SUBROUTINE WHERE.</td>
</tr>
<tr>
<td>LDRUM</td>
<td>CIO</td>
<td>DEVICE FOR FINAL METRIC INFORMATION</td>
</tr>
<tr>
<td>LEQ1</td>
<td>EQN</td>
<td>LOWEST INDEX OF EQUATIONS SOLVED EITHER BY COUPLED OR UNCOUPLED ADI SWEEP</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LEO2</td>
<td>EQN</td>
<td>HIGHEST INDEX OF EQUATIONS SOLVED EITHER BY COUPLED OR UNCOUPLED ADI SWEEP</td>
</tr>
<tr>
<td>LEV(3)</td>
<td>ADDR</td>
<td>GEOMETRY LEVEL</td>
</tr>
<tr>
<td>LEVEL</td>
<td>ADDRG</td>
<td>GEOMETRY LEVEL</td>
</tr>
<tr>
<td>LEXTY(2,NN)</td>
<td>BOUND</td>
<td>EXTENTS OF COMPUTATIONAL DOMAIN OF EACH Y-COORDINATE LINE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LEXTY(1,NN) = LOWER LIMIT; LEXTY(2,NN) = UPPER LIMIT</td>
</tr>
<tr>
<td>LEXTZ(2,NN)</td>
<td>BOUND</td>
<td>EXTENTS OF COMPUTATIONAL DOMAIN OF EACH Z-COORDINATE LINE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LEXTZ(1,NN) = LOWER LIMIT; LEXTZ(2,NN) = UPPER LIMIT</td>
</tr>
<tr>
<td>LGA1(NN)</td>
<td>ADDRG</td>
<td>ADDRESS FOR POINT LOGIC OF GEOMETRIC VARIABLES</td>
</tr>
<tr>
<td>LGA2(NDIFM,NN)</td>
<td>ADDRG</td>
<td>ADDRESS FOR Y - DERIVATIVE OF GEOMETRIC VARIABLES</td>
</tr>
<tr>
<td>LGA3(NDIFM,NN)</td>
<td>ADDRG</td>
<td>ADDRESS FOR Z - DERIVATIVE LOGIC OF GEOMETRIC VARIABLES</td>
</tr>
<tr>
<td>LGA4(NDIFMAA2,NN)</td>
<td>ADDRG</td>
<td>ADDRESS FOR CROSS DERIVATIVE(Y-Z) LOGIC OF GEOMETRIC VARIABLES</td>
</tr>
<tr>
<td>LGA5(NDIFMAA2,NN)</td>
<td>ADDRG</td>
<td>ADDRESS FOR CROSS DERIVATIVE(Z-Y) LOGIC OF GEOMETRIC VARIABLES</td>
</tr>
<tr>
<td>LP</td>
<td>SECT</td>
<td>INDEX OF COMPUTATIONAL LOOP IN EITHER Y OR Z DIRECTION.</td>
</tr>
<tr>
<td>LREF</td>
<td>REF</td>
<td>REFERENCE LENGTH</td>
</tr>
<tr>
<td>LREFI</td>
<td>REF</td>
<td>INVERSE OF REFERENCE LENGTH</td>
</tr>
<tr>
<td>LSHIFT</td>
<td>GEOM</td>
<td>SHIFT INDEX FOR COORDINATE TRANSFORMATION</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LVG(8)</td>
<td>BOUND</td>
<td>BOUNDARY POINT INDICATOR FOR BOUNDARY CONDITION</td>
</tr>
<tr>
<td>MASS1</td>
<td>CIO</td>
<td>GENERAL PURPOSE MASS STORAGE DEVICE</td>
</tr>
<tr>
<td>MASS2</td>
<td>CIO</td>
<td>GENERAL PURPOSE MASS STORAGE DEVICE</td>
</tr>
<tr>
<td>MCPLD</td>
<td>EQN</td>
<td>NUMBER OF COUPLED EQUATIONS TO BE SOLVED</td>
</tr>
<tr>
<td>MEFF</td>
<td>VAR</td>
<td>INDEX FOR EFFECTIVE VISCOSITY</td>
</tr>
<tr>
<td>MEFFL</td>
<td>VAR</td>
<td>STORAGE LEVEL OF EFFECTIVE VISCOSITY IN GENERAL PURPOSE STORAGE</td>
</tr>
<tr>
<td>MEOK</td>
<td>EQN</td>
<td>LEQ1 - 1</td>
</tr>
<tr>
<td>MEQS</td>
<td>EQN</td>
<td>TOTAL NUMBER OF EQUATIONS TO BE SOLVED</td>
</tr>
<tr>
<td>MEQS1</td>
<td>EQN</td>
<td>INDEX OF FIRST EQUATION TO BE SOLVED</td>
</tr>
<tr>
<td>MEQS2</td>
<td>EQN</td>
<td>INDEX OF LAST EQUATION TO BE SOLVED</td>
</tr>
<tr>
<td>MGDMP</td>
<td>DMP</td>
<td>FLAG FOR DUMP OPTIONS</td>
</tr>
<tr>
<td>MGD1</td>
<td>GRID</td>
<td>IL + 1</td>
</tr>
<tr>
<td>MGD2</td>
<td>GRID</td>
<td>IH - 1</td>
</tr>
<tr>
<td>MIN</td>
<td>CIO</td>
<td>INPUT DEVICE</td>
</tr>
<tr>
<td>MINF</td>
<td>FREE</td>
<td>FREE STREAM MACH NUMBER</td>
</tr>
<tr>
<td>ML</td>
<td>VAR</td>
<td>INDEX FOR MIXING LENGTH</td>
</tr>
<tr>
<td>MLEVEL</td>
<td>PARAM</td>
<td>MAXIMUM NO. OF STORAGE LEVELS</td>
</tr>
<tr>
<td>MLL</td>
<td>VAR</td>
<td>STORAGE LEVEL OF MIXING LENGTH IN GENERAL PURPOSE STORAGE</td>
</tr>
<tr>
<td>MN</td>
<td>VAR</td>
<td>INDEX FOR MACH NUMBER</td>
</tr>
<tr>
<td>MNL</td>
<td>VAR</td>
<td>STORAGE LEVEL OF MACH NO IN GENERAL PURPOSE STORAGE</td>
</tr>
<tr>
<td>MOUT</td>
<td>CIO</td>
<td>OUTPUT DEVICE</td>
</tr>
<tr>
<td>MREF</td>
<td>REF</td>
<td>REFERENCE MACH NUMBER</td>
</tr>
<tr>
<td>MREFI</td>
<td>REF</td>
<td>INVERSE OF REFERENCE MACH NUMBER</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MSDD</td>
<td>CIO</td>
<td>TEMPORARY MASS STORAGE DEVICE</td>
</tr>
<tr>
<td>MSD1</td>
<td>CIOD</td>
<td>GENERAL PURPOSE MASS STORAGE DEVICE</td>
</tr>
<tr>
<td>MSD2</td>
<td>CIOD</td>
<td>GENERAL PURPOSE MASS STORAGE DEVICE</td>
</tr>
<tr>
<td>MSECY</td>
<td>SECT</td>
<td>MAXIMUM NO. OF Y-PERSPECTIVE SUBSECTIONS (USED TO SET UP THE SUBSECTIONS FOR CROSEC)</td>
</tr>
<tr>
<td>MSECZ</td>
<td>SECT</td>
<td>MAXIMUM NUMBER OF Z-PERSPECTIVE SUBSECTIONS (USED TO SET UP THE SUBSECTIONS FOR CROSEC)</td>
</tr>
<tr>
<td>MSGVAR(25)</td>
<td>PRNT</td>
<td>TITLE OF VARIABLES TO BE PRINTED</td>
</tr>
<tr>
<td>MU</td>
<td>VAR</td>
<td>INDEX FOR LAMINAR VISCOSITY</td>
</tr>
<tr>
<td>MUINF</td>
<td>FREE</td>
<td>FREE STREAM LAMINAR VISCOSITY</td>
</tr>
<tr>
<td>MUL</td>
<td>VAR</td>
<td>STORAGE LEVEL OF LAMINAR VISCOSITY IN GENERAL PURPOSE STORAGE</td>
</tr>
<tr>
<td>MUREF</td>
<td>REF</td>
<td>REFERENCE VISCOSITY</td>
</tr>
<tr>
<td>MUREFI</td>
<td>REF</td>
<td>INVERSE OF REFERENCE VISCOSITY</td>
</tr>
<tr>
<td>MUI</td>
<td>VAR</td>
<td>INDEX FOR TURBULENT VISCOSITY</td>
</tr>
<tr>
<td>MUILL</td>
<td>VAR</td>
<td>STORAGE LEVEL OF TURBULENT VISCOSITY IN GENERAL PURPOSE STORAGE</td>
</tr>
<tr>
<td>MWINF</td>
<td>FREE</td>
<td>FREE STREAM MOLECULAR WEIGHT</td>
</tr>
<tr>
<td>MWREF</td>
<td>REF</td>
<td>REFERENCE MOLECULAR WEIGHT</td>
</tr>
<tr>
<td>MWREFI</td>
<td>REF</td>
<td>INVERSE OF REFERENCE MOLECULAR WEIGHT</td>
</tr>
<tr>
<td>MZVAR</td>
<td>PARAM</td>
<td>MAXIMUM NUMBER OF STORAGE VARIABLES</td>
</tr>
<tr>
<td>NABC</td>
<td>CCOM</td>
<td>NABC = ID</td>
</tr>
<tr>
<td>NANG</td>
<td>METRIC</td>
<td>ANGLE OF COORDINATE LINES RELATIVE TO HORIZONTAL</td>
</tr>
<tr>
<td>NRBRKX</td>
<td>BOUND</td>
<td>NUMBER OF STREAMWISE LOCATIONS WHERE SOLID BODIES EMBEDDED IN THE COMPUTATIONAL DOMAIN START OR TERMINATE.</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>NCMAX(5)</td>
<td>SECT</td>
<td>UPPER LIMIT OF BOUNDARY LINE WHERE ENDCAP SOLUTION IS OBTAINED.</td>
</tr>
<tr>
<td>NCMIN(5)</td>
<td>SECT</td>
<td>LOWER LIMIT OF BOUNDARY LINE WHERE ENDCAP SOLUTION IS OBTAINED.</td>
</tr>
<tr>
<td>NCPLD</td>
<td>PARAM</td>
<td>MAXIMUM NUMBER OF COUPLED EQUATIONS</td>
</tr>
<tr>
<td>NCPLD2</td>
<td>CCOM</td>
<td>NCPLD+2</td>
</tr>
<tr>
<td>NCRF</td>
<td>SECT</td>
<td>EQUAL TO EITHER NCREF(ISEC) OR NZCRF(ISEC) WHERE ISEC INDICATES A SUBSECTION IN THE CROSS-SECTION.</td>
</tr>
<tr>
<td>NCRL</td>
<td>SECT</td>
<td>EQUAL TO EITHER NCRL(ISEC) OR NZCRL(ISEC) WHERE ISEC INDICATES A SUBSECTION IN THE CROSS-SECTION.</td>
</tr>
<tr>
<td>NCIR</td>
<td>DIFCOM</td>
<td>CENTER OF DIFFERENCE MOLECULE</td>
</tr>
<tr>
<td>NCUP</td>
<td>EQN</td>
<td>NUMBER OF COUPLED EQUATIONS</td>
</tr>
<tr>
<td>NDIFM</td>
<td>PARAM</td>
<td>MAXIMUM NUMBER OF GRID POINTS IN A DIFFERENCE MOLECULE</td>
</tr>
<tr>
<td>NDIFM1</td>
<td>DIFCOM</td>
<td>2*NDI FM</td>
</tr>
<tr>
<td>NDI FM1</td>
<td>DIFCOM</td>
<td>NDI FM - 1</td>
</tr>
<tr>
<td>N DI FP1</td>
<td>DIFCOM</td>
<td>NDI FM + 1</td>
</tr>
<tr>
<td>NE(2)</td>
<td>GEOM</td>
<td>NUMBER OF GRID POINTS IN Y AND Z DIRECTIONS</td>
</tr>
<tr>
<td>NEQN(NEQS,2)</td>
<td>EQN</td>
<td>NUMBER OF COUPLED EQUATIONS TO BE SOLVED IN EACH ADI SWEEP</td>
</tr>
<tr>
<td>NEQS</td>
<td>PARAM</td>
<td>MAXIMUM NUMBER OF EQUATIONS IN CODE</td>
</tr>
<tr>
<td>NEY</td>
<td>GRID</td>
<td>NUMBER OF GRID POINTS IN Y DIRECTION</td>
</tr>
<tr>
<td>NEYM1</td>
<td>GRID</td>
<td>NEY - 1</td>
</tr>
<tr>
<td>NEZ</td>
<td>GRID</td>
<td>NUMBER OF GRID POINTS IN Z DIRECTION</td>
</tr>
<tr>
<td>NEZM1</td>
<td>GRID</td>
<td>NEZ - 1</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>NE2S</td>
<td>CIO</td>
<td>SENTINEL FOR SPREADING OF 2-D PROFILE TO 3-D</td>
</tr>
<tr>
<td>NFILE</td>
<td>REST</td>
<td>SEQUENCE NUMBER OF RESTART INFORMATION</td>
</tr>
<tr>
<td>NGEOMV</td>
<td>METRIC</td>
<td>NUMBER OF METRIC COEFFICIENTS AND Derivatives</td>
</tr>
<tr>
<td>NH1</td>
<td>METRIC</td>
<td>INDEX FOR METRIC COEFFICIENT IN X-DIRECTION</td>
</tr>
<tr>
<td>NH12</td>
<td>METRIC</td>
<td>INDEX FOR Derivative OF X METRIC IN Y DIRECTION</td>
</tr>
<tr>
<td>NH2</td>
<td>METRIC</td>
<td>INDEX FOR METRIC COEFFICIENT IN Y-DIRECTION</td>
</tr>
<tr>
<td>NH21</td>
<td>METRIC</td>
<td>INDEX FOR Derivative OF Y METRIC IN X DIRECTION</td>
</tr>
<tr>
<td>NH3</td>
<td>METRIC</td>
<td>INDEX FOR METRIC COEFFICIENT IN Z-DIRECTION</td>
</tr>
<tr>
<td>NH31</td>
<td>METRIC</td>
<td>INDEX FOR Derivative OF Z METRIC IN X DIRECTION</td>
</tr>
<tr>
<td>NH32</td>
<td>METRIC</td>
<td>INDEX FOR Derivative OF Z METRIC IN Y DIRECTION</td>
</tr>
<tr>
<td>NIII</td>
<td>CIO</td>
<td>NO. OF FALSE MARCHING STEPS USED TO GENERATE THE INITIAL PROFILE</td>
</tr>
<tr>
<td>NIN</td>
<td>DIFCOM</td>
<td>MAXIMUM LINES OF STORAGE IN CORE AT ONE TIME.</td>
</tr>
<tr>
<td>NINC</td>
<td>DIFCOM</td>
<td>NIN/2 + 1</td>
</tr>
<tr>
<td>NJD</td>
<td>DIFCOM</td>
<td>GRID POINT LOCATION FOR START OF SECOND SWEEP</td>
</tr>
<tr>
<td>NJF(10)</td>
<td>SECT</td>
<td>LEFT BOUNDARY INDEX OF EACH Y-PERSPECTIVE SUBSECTION USED FOR Y-SWEEP OF ADI.</td>
</tr>
<tr>
<td>NJL(10)</td>
<td>SECT</td>
<td>RIGHT BOUNDARY INDEX OF EACH Y-PERSPECTIVE SUBSECTION USED FOR Y-SWEEP OF ADI.</td>
</tr>
<tr>
<td>NJSEG</td>
<td>SECT</td>
<td>NUMBER OF SUBSECTIONS IN EACH CROSS SECTION USED FOR Y-SWEEP OF ADI</td>
</tr>
<tr>
<td>NKF(10)</td>
<td>SECT</td>
<td>LOWER BOUNDARY INDEX OF EACH Z-PERSPECTIVE SUBSECTION USED FOR Z-SWEEP OF ADI.</td>
</tr>
<tr>
<td>NKL(10)</td>
<td>SECT</td>
<td>UPPER BOUNDARY INDEX OF EACH Z-PERSPECTIVE SUBSECTION USED FOR Z-SWEEP OF ADI.</td>
</tr>
<tr>
<td>NKSEG</td>
<td>SECT</td>
<td>NUMBER OF SUBSECTIONS IN EACH CROSS SECTION</td>
</tr>
</tbody>
</table>

78
<table>
<thead>
<tr>
<th>FORTRAN SYMBOL</th>
<th>COMMON BLOCK</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLOOPY(NN)</td>
<td>SECT</td>
<td>NUMBER OF LOOPS ON Y-COORDINATE LINE</td>
</tr>
<tr>
<td>NLOOPOZ(NN)</td>
<td>SECT</td>
<td>NUMBER OF LOOPS ON Z-COORDINATE LINE.</td>
</tr>
<tr>
<td>NHAXWF</td>
<td>LAWW</td>
<td>MAXIMUM NUMBER OF ITERATIONS ALLOWABLE IN CALCULATION OF WALL SHEAR VELOCITY</td>
</tr>
<tr>
<td>NN</td>
<td>PARAM</td>
<td>MAXIMUM NUMBER OF GRID POINTS IN Y OR Z DIRECTION</td>
</tr>
<tr>
<td>NPADI</td>
<td>EQN</td>
<td>NUMBER OF COUPLED AND UNCOUPLED EQUATIONS TO BE SOLVED DURING EACH ADI SWEEP</td>
</tr>
<tr>
<td>NPROF(10)</td>
<td>PROFILE</td>
<td>NUMBER OF SURFACES WHERE INITIAL BOUNDARY LAYER PROFILES ARE GENERATED IN EACH LOOP EITHER IN Y OR Z DIRECTION. NPROF IS EITHER 1 OR 2.</td>
</tr>
<tr>
<td>NPTSX(4)</td>
<td>INTBC</td>
<td>NUMBER OF STREAMWISE LOCATIONS WHERE TRANSPARATION DATA IS INPUT</td>
</tr>
<tr>
<td>NPTSYZ(15,4)</td>
<td>INTBC</td>
<td>NUMBER OF CROSS-PLANE LOCATIONS WHERE TRANSPARATION IS INPUT</td>
</tr>
<tr>
<td>NPUNCH</td>
<td>CIO</td>
<td>PUNCH DEVICE</td>
</tr>
<tr>
<td>NRGT(2)</td>
<td>DIFCOM</td>
<td>MCTR POINTS FROM RIGHT OR TOP BOUNDARY</td>
</tr>
<tr>
<td>NS</td>
<td>GEOM</td>
<td>LAST MARCHING STATION</td>
</tr>
<tr>
<td>NSAVED</td>
<td>REST</td>
<td>SEQUENCE NUMBER OF RESTART STATIONS SAVED</td>
</tr>
<tr>
<td>NSECRY</td>
<td>SECT</td>
<td>NUMBER OF Y-PERSPECTIVE SUBSECTIONS IN THE CROSS-SECTION FOR SUBROUTINE CROSEC.</td>
</tr>
<tr>
<td>NSECRZ</td>
<td>SECT</td>
<td>NUMBER OF Z-PERSPECTIVE SUBSECTIONS IN THE CROSS-SECTION FOR SUBROUTINE CROSEC.</td>
</tr>
<tr>
<td>NSMAX</td>
<td>PARAM</td>
<td>2 GREATER THAN NS</td>
</tr>
<tr>
<td>NSURF(4)</td>
<td>BOUND</td>
<td>SURFACE NUMBER OF FOUR BOUNDARIES WITH MINIMUM DISTANCE AWAY FROM A GRID POINT CONSIDERED.</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>NUNERR</td>
<td>CIO</td>
<td>DEVICE FOR MGAUSS ERROR CHECK</td>
</tr>
<tr>
<td>NVSOLV</td>
<td>EQN</td>
<td>NUMBER OF DEPENDENT VARIABLES</td>
</tr>
<tr>
<td>NWORD2(50)</td>
<td>STRAGE</td>
<td>SIZE OF EACH COMMON BLOCK</td>
</tr>
<tr>
<td>NYCRF(10)</td>
<td>SECT</td>
<td>LOWER BOUNDARY INDEX OF EACH Z-PERSPECTIVE SUBSECTION FOR USE IN CROSEC.</td>
</tr>
<tr>
<td>NYCRL(10)</td>
<td>SECT</td>
<td>UPPER BOUNDARY INDEX OF EACH Z-PERSPECTIVE SUBSECTION FOR USE IN CROSEC.</td>
</tr>
<tr>
<td>NZCRF(10)</td>
<td>SECT</td>
<td>LEFT BOUNDARY INDEX OF EACH Y-PERSPECTIVE SUBSECTION FOR USE IN CROSEC.</td>
</tr>
<tr>
<td>NZCRL(10)</td>
<td>SECT</td>
<td>RIGHT BOUNDARY INDEX OF EACH Y-PERSPECTIVE SUBSECTION FOR USE IN CROSEC.</td>
</tr>
<tr>
<td>OMBWD</td>
<td>LIN</td>
<td>1.0 - BWD</td>
</tr>
<tr>
<td>OMEGWGF</td>
<td>LAWW</td>
<td>UNDER-RELAXATION FACTOR FOR WALL FUNCTION FORMULATION</td>
</tr>
<tr>
<td>P</td>
<td>VAR</td>
<td>INDEX FOR STATIC PRESSURE</td>
</tr>
<tr>
<td>PCON1</td>
<td>REF</td>
<td>(GAMMA-1.0)/GAMMA</td>
</tr>
<tr>
<td>PCON2</td>
<td>REF</td>
<td>0.5 * PCON1</td>
</tr>
<tr>
<td>PINF</td>
<td>FREE</td>
<td>FREE STREAM STATIC PRESSURE</td>
</tr>
<tr>
<td>PL</td>
<td>VAR</td>
<td>STORAGE LEVEL OF STATIC PRESSURE IN GENERAL PURPOSE STORAGE</td>
</tr>
<tr>
<td>PLTFLD(NN,NN,8)</td>
<td>ZPLOT</td>
<td>GENERAL PURPOSE STORAGE FOR PLOT INFORMATION</td>
</tr>
<tr>
<td>PR</td>
<td>REF</td>
<td>PRANDIL NUMBER</td>
</tr>
<tr>
<td>PREF</td>
<td>REF</td>
<td>REFERENCE PRESSURE</td>
</tr>
</tbody>
</table>

80
<table>
<thead>
<tr>
<th>FORTRAN SYMBOL</th>
<th>COMMON BLOCK</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFI</td>
<td>REF</td>
<td>INVERSE OF REFERENCE PRESSURE</td>
</tr>
<tr>
<td>PREPS</td>
<td>VISC</td>
<td>PRANDTL NO. IN TURBULENT ENERGY DISSIPATION EQUATION</td>
</tr>
<tr>
<td>PRESS(20,2)</td>
<td>BOUND</td>
<td>SPECIFIED PRESSURE AT BOUNDARY</td>
</tr>
<tr>
<td>PRT</td>
<td>REF</td>
<td>TURBULENT PRANDTL NO.</td>
</tr>
<tr>
<td>PRTKE</td>
<td>VISC</td>
<td>PRANDTL NO. IN TURBULENT KINETIC ENERGY EQUATION</td>
</tr>
<tr>
<td>PZERO</td>
<td>FREE</td>
<td>STAGNATION PRESSURE</td>
</tr>
<tr>
<td>Q1D(8,NN)</td>
<td>METRIC</td>
<td>INTERMEDIATE STORAGE ARRAY FOR METRIC INFORMATION</td>
</tr>
<tr>
<td>Q2D(8,NN)</td>
<td>METRIC</td>
<td>INTERMEDIATE STORAGE ARRAY FOR METRIC INFORMATION</td>
</tr>
<tr>
<td>R</td>
<td>VAR</td>
<td>INDEX FOR DENSITY</td>
</tr>
<tr>
<td>RAILD</td>
<td>LAWW</td>
<td>EMPIRICAL NUMERICAL CONSTANT IN MIXING LENGTH COMPUTATION</td>
</tr>
<tr>
<td>RE</td>
<td>REF</td>
<td>REYNOLDS NUMBER</td>
</tr>
<tr>
<td>REI</td>
<td>REF</td>
<td>INVERSE OF REYNOLDS NUMBER</td>
</tr>
<tr>
<td>REI2</td>
<td>REF</td>
<td>2.0 * REI</td>
</tr>
<tr>
<td>REPL</td>
<td>FREE</td>
<td>REYNOLDS NUMBER PER UNIT LENGTH</td>
</tr>
<tr>
<td>RGAS</td>
<td>REF</td>
<td>GAS CONSTANT</td>
</tr>
<tr>
<td>RHO(NDIFM)</td>
<td>ADDR</td>
<td>STORAGE OF DENSITY FOR FIRST DERIVATIVES</td>
</tr>
<tr>
<td>RHOINF</td>
<td>FREE</td>
<td>FREE STREAM DENSITY</td>
</tr>
<tr>
<td>RHOREF</td>
<td>REF</td>
<td>REFERENCE DENSITY</td>
</tr>
<tr>
<td>RHOWL(NN,5,4)</td>
<td>BOUND</td>
<td>STORAGE FOR DENSITY ON THE BOUNDARY</td>
</tr>
<tr>
<td>RHREFI</td>
<td>REF</td>
<td>INVERSE OF REFERENCE DENSITY</td>
</tr>
<tr>
<td>RUNIV</td>
<td>UNITS</td>
<td>UNIVERSAL GAS CONSTANT</td>
</tr>
<tr>
<td>SAVE(NEOS,NN)</td>
<td>EQN</td>
<td>STORAGE FOR CHANGES DURING FIRST ADI SWEEP</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SN(NN)</td>
<td>LIN</td>
<td>STORAGE FOR N TH LEVEL TERMS</td>
</tr>
<tr>
<td>SQ1</td>
<td>ADD</td>
<td>ADD CODE STREAMWISE LOCATION</td>
</tr>
<tr>
<td>SQ2</td>
<td>ADD</td>
<td>ADD CODE STREAMWISE LOCATION</td>
</tr>
<tr>
<td>SYSTEM</td>
<td>ZPLOT</td>
<td>SENTINEL FOR COORDINATE SYSTEM - PLOTS ONLY</td>
</tr>
<tr>
<td>T</td>
<td>VAR</td>
<td>INDEX FOR STATIC TEMPERATURE</td>
</tr>
<tr>
<td>TEMPS(NN,5,4)</td>
<td>BOUND</td>
<td>STORAGE ARRAY FOR TEMPERATURE ON BOUNDARY</td>
</tr>
<tr>
<td>TEMPSN(NN,5,4)</td>
<td>BOUND</td>
<td>STORAGE ARRAY FOR TEMPERATURE AT NTH STREAMWISE STATION - BOUNDARIES ONLY</td>
</tr>
<tr>
<td>TINF</td>
<td>FREE</td>
<td>FREE STREAM STATIC TEMPERATURE</td>
</tr>
<tr>
<td>TITLE(6)</td>
<td>ZPLOT</td>
<td>TITLE FOR PLOT FILE</td>
</tr>
<tr>
<td>TKE</td>
<td>VAR</td>
<td>INDEX FOR TURBULENT KINETIC ENERGY</td>
</tr>
<tr>
<td>TKEINF</td>
<td>LAWW</td>
<td>FREE STREAM TURBULENT KINETIC ENERGY</td>
</tr>
<tr>
<td>TL</td>
<td>VAR</td>
<td>STORAGE LEVEL OF STATIC TEMPERATURE IN GENERAL PURPOSE STORAGE</td>
</tr>
<tr>
<td>TREF</td>
<td>REF</td>
<td>REFERENCE TEMPERATURE</td>
</tr>
<tr>
<td>TREFI</td>
<td>REF</td>
<td>INVERSE OF REFERENCE TEMPERATURE</td>
</tr>
<tr>
<td>TT1(2)</td>
<td>GEOM</td>
<td>MESH DISTRIBUTION FACTOR</td>
</tr>
<tr>
<td>TT2(2)</td>
<td>GEOM</td>
<td>MESH DISTRIBUTION FACTOR</td>
</tr>
<tr>
<td>TWALL(20,2)</td>
<td>BOUND</td>
<td>SPECIFIED TEMPERATURE AT BOUNDARY</td>
</tr>
<tr>
<td>TWOD</td>
<td>GEOM</td>
<td>FLAG FOR TWO DIMENSIONAL LOGIC</td>
</tr>
<tr>
<td>TZERO</td>
<td>FREE</td>
<td>STAGNATION TEMPERATURE</td>
</tr>
<tr>
<td>T2(2)</td>
<td>GEOM</td>
<td>MESH DISTRIBUTION FACTOR</td>
</tr>
<tr>
<td>U</td>
<td>VAR</td>
<td>INDEX FOR VELOCITY IN X-DIRECTION</td>
</tr>
<tr>
<td>UDUE(20,2)</td>
<td>LAWW</td>
<td>BOUNDARY LAYER THICKNESS SAMPLING CRITERIA</td>
</tr>
<tr>
<td>UDUM(NDIFM)</td>
<td>ADDR</td>
<td>STORAGE OF VELOCITY FOR FIRST DERIVATIVES</td>
</tr>
<tr>
<td>UINF</td>
<td>FREE</td>
<td>FREE STREAM VELOCITY</td>
</tr>
</tbody>
</table>

82
<table>
<thead>
<tr>
<th>FORTRAN SYMBOL</th>
<th>COMMON BLOCK</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>UREF</td>
<td>REF</td>
<td>REFERENCE VELOCITY</td>
</tr>
<tr>
<td>UREFI</td>
<td>REF</td>
<td>INVERSE OF REFERENCE VELOCITY</td>
</tr>
<tr>
<td>USCALE</td>
<td>ADD</td>
<td>METRIC SCALE FACTOR</td>
</tr>
<tr>
<td>USTAR(NN,5,4)</td>
<td>LAWW</td>
<td>FRICTION VELOCITY ON SOLID WALL BOUNDARY</td>
</tr>
<tr>
<td>UTIL(NDIFM)</td>
<td>ADDR</td>
<td>STORAGE OF DEPENDENT VARIABLE FOR FIRST DERIVATIVES</td>
</tr>
<tr>
<td>V</td>
<td>VAR</td>
<td>INDEX FOR VELOCITY IN Y-DIRECTION</td>
</tr>
<tr>
<td>VELSQ(NN,NN)</td>
<td></td>
<td>STORAGE FOR UAA2 + VAA2 + WAA2 - EQUIVALENCED TO PLTFLD(1,1,4)</td>
</tr>
<tr>
<td>VKB</td>
<td>LAWW</td>
<td>SECOND CONSTANT IN LOGARITHMIC LAW OF THE WALL</td>
</tr>
<tr>
<td>VKC</td>
<td>LAWW</td>
<td>VON KARMAN CONSTANT</td>
</tr>
<tr>
<td>VNO(15,15,4)</td>
<td>INIBC</td>
<td>INPUT TRANSPERSION RATES</td>
</tr>
<tr>
<td>W</td>
<td>VAR</td>
<td>INDEX FOR VELOCITY IN Z-DIRECTION</td>
</tr>
<tr>
<td>X(502)</td>
<td>GEOM</td>
<td>STREAMWISE LOCATION</td>
</tr>
<tr>
<td>XBRKX(5)</td>
<td>BOUND</td>
<td>STREAMWISE LOCATION WHERE AN EMBEDDED SOLID BODY STARTS OR TERMINATES</td>
</tr>
<tr>
<td>XENIR</td>
<td>GEOM</td>
<td>STARTING STREAMWISE LOCATION</td>
</tr>
<tr>
<td>XG1(NN,2)</td>
<td>OPER</td>
<td>FIRST DERIVATIVES OF COMPUTATIONAL COORDINATES WITH RESPECT TO PHYSICAL COORDINATES</td>
</tr>
<tr>
<td>XG2(NN,2)</td>
<td>OPER</td>
<td>SECOND DERIVATIVES OF COMPUTATIONAL COORDINATES WITH RESPECT TO PHYSICAL COORDINATES</td>
</tr>
<tr>
<td>XOB(4)</td>
<td>BOUND</td>
<td>INITIAL LOCATION OF SOLID OBSTACLE IN X DIRECTION</td>
</tr>
<tr>
<td>XVNO(15,15,4)</td>
<td>INIBC</td>
<td>STREAMWISE LOCATIONS WHERE TRANSPERSION DATA IS INPUT</td>
</tr>
<tr>
<td>X0(2)</td>
<td>GEOM</td>
<td>MESH DISTRIBUTION FACTOR</td>
</tr>
<tr>
<td>Y(NN,NN,2)</td>
<td></td>
<td>PHYSICAL DISTANCES FROM BOUNDARIES - EQUIVALENCED TO PLTFLD(1,1,2)</td>
</tr>
<tr>
<td>FORTRAN SYMBOL</td>
<td>COMMON BLOCK</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>YAW</td>
<td>REF</td>
<td>YAW ANGLE</td>
</tr>
<tr>
<td>YPLUSL(NN,NN)</td>
<td>GEOM</td>
<td>STORAGE FOR RHO &amp; UIAU / VISLAM - EQUIVALENCED TO PFIELD(1,1,1)</td>
</tr>
<tr>
<td>YS(2,2)</td>
<td>GEOM</td>
<td>NONDIMENSIONAL EXTENTS OF COMPUTATIONAL DOMAIN</td>
</tr>
<tr>
<td>YSAVE(NN,2)</td>
<td>GEOM</td>
<td>COMPUTATIONAL COORDINATES</td>
</tr>
<tr>
<td>YZPROF(NN)</td>
<td>PROFILE</td>
<td>TEMPORARY STORAGE ARRAY FOR PHYSICAL COORDINATES</td>
</tr>
<tr>
<td>YZUNO(15,15,4)</td>
<td>INTBC</td>
<td>CROSS-PLANE LOCATIONS WHERE TRANSPIRATION DATA IS INPUT</td>
</tr>
<tr>
<td>ZNIRN(NDIM,3)</td>
<td>ZEX1</td>
<td>ARRAY FOR VIRTUAL MEMORY STORAGE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NDim = NN x NN x NZVAR x MLEVEL</td>
</tr>
<tr>
<td>ZZ(M,L,K)</td>
<td>VARZZ</td>
<td>GENERAL PURPOSE STORAGE FOR DEPENDENT AND DERIVED VARIABLES - M = NZVAR, L = MLEVEL, K = NDIEM x NN</td>
</tr>
</tbody>
</table>
Tables I and II present sample input and output for one of the test cases previously discussed. The sample input was used to calculate the flow field of rectangular supersonic jet (aspect ratio = 1) interacting with the ambient stream. The input data is for an initial run (IRSTIN = 0) with a restart to be written every 10 marching steps (IRSTOT=10). The case is to be run on CRAY-1 computer. The streamwise and two transverse momentum as well as continuity and stagnation enthalpy (IHSTAG = 1) version of the energy equation are to be solved. The reference length is 2.0m (IUNITs = 2), the Reynolds's number per m is $1.6685 \times 10^6$, the free stream Mach number 2.0 and the free stream pressure is 2864.0 Nt/m$^2$. 50 x 50 grid points are utilized in both directions of the computational cross-section (NE = 50,50) and a Cartesian coordinate system is to be utilized (IGEM = 1). The initial run is to be marched 10 steps (NS = 10) starting at a streamwise location of 0.0 (XENTR = 0.0). The streamwise marching step size is 0.01 (DELX = 0.01). Then grid point characteristics is specified (IFBW(1,1) = ... ...) to define a computational domain in the cross-section. For a detailed explanation of the indicators, computer program list should be consulted. During this initial run grid point characteristics will change at one streamwise location (NBRKX = 1). At the streamwise location 0.015 (XBRKX(1) = 0.015) an embedded solid body will begin (IEDGE = 0). The solid body is a splitter bounded by two surface-fitted coordinate lines (18 and 19 in Y-direction and 18 and 19 in Z-direction). In the Namelist LISTR, IFBW is respecified as shown in Table I. An information on IBOUND and JEQBC corresponding to the new IFBW is also provided. Grid points are clustered in the vicinity of the embedded solid body, i.e., about 0.2 in Y-direction and 0.2 in Z-direction (XO(1) = 0.2,0.2). The packing is to be moderately tight (T2(1) = 3.0, 3.0). An initial profile is generated within the program to supply a uniform stream needed for this case (IPROF = 1) and modified to give two different magnitudes of velocity. Printout is given every 5 steps (IPRINT = 5) and plot information is written every 5 steps (IPLOT = 5).

The output for the three-dimensional case consists of NAMELIST information and difference weight information and flow field information at each 5th streamwise station. The NAMELIST information is provided as a means
for the user to check the input data. Three-dimensional flow field output is
controlled by the variable IVARPR. For three dimensions, however, the output
is in the form of a cross-sectional plane of output. All variables (except
the pressure) are in a non-dimensional form with respect to the reference
conditions which are displayed in NAMELIST LIST 2. The pressure terms are
nondimensionalized with respect to the freestream pressure. The integer
variables IY and IZ represent the transverse and spanwise grid point
locations respectively while Z and Y are the corresponding computational
positions. Table II is a portion of the output for the three-dimensional
rectangular jet case i.e., the cross plane distribution of the streamwise
velocity (UVEL) and pressure (PRES). Other variables can (and were) printed
out, but for reasons of economy of space are not presented here. Following
the flow field information is the subsonic-supersonic grid point position
indicator (ISS) with respect to the lower and upper surface (right and left
surface for IADI = 3), and the boundary indicator (JBOUND). The ISS values
tell the grid point where the flow transitions from subsonic to supersonic
flow while the variable JBOUND tells the type of surface (JBOUND = 1
corresponding to a wall and JBOUND = 2 corresponding to a nonwall). Finally,
plot file information is displayed.
REFERENCES


REFERENCES (Continued)


REFERENCES (Continued)


FIGURE 1. - SHOCK STRUCTURE FOR A TYPICAL OVEREXPANDED AXISYMMETRIC NOZZLE
FIGURE 2 - SCHEMATIC OF JET NEAR FIELD STRUCTURE FOR
SUPersonic AND SUBSONIC EXTERNAL FLOWS
(REF. 18)
FIGURE 3 - SCHEMATIC OF AXISYMMETRIC JET FLOW
FIGURE 4(a) - STREAMWISE STATIC PRESSURE DISTRIBUTION
FIGURE 5 - SCHEMATIC OF 3-DIMENSIONAL JET FLOW
THE FIGURE IS NOT SCALED

FIGURE 6 - SCHEMATIC OF COMPUTATIONAL DOMAIN (SIDE VIEW)
FIGURE 7 - CROSS-SECTION OF COMPUTATIONAL DOMAIN
(ASPECT RATIO = 1)
Figure 8 - Aspect Ratio = 1 - Streamwise Development of Stagnation Pressure Isobars
FIGURE 9(a) - SUPersonic jet flow (aspect ratio = 1) - Streamwise development of the secondary velocity in the cross-section
FIGURE 9(b) - SUPersonic JET FLOW (ASPECT RATIO = 1) - STREAMWISE DEVELOPMENT OF THE SECONDARY VELOCITY IN THE CROSS-SECTION
Figure 9(d) - Supersonic Jet Flow (Aspect Ratio = 1) - Streamwise Development of the Secondary Velocity in the Cross-Section
FIGURE 10 - ASPECT RATIO = 2 — STREAMWISE DEVELOPMENT OF STAGNATION PRESSURE ISOBARS
FIGURE 12 - A SAMPLE CROSS-SECTION WITH EMBEDDED SOLID BODIES.
<table>
<thead>
<tr>
<th>TOP RECORD</th>
<th>NOZZLE(AR=1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
</tr>
<tr>
<td>&amp;REST</td>
<td></td>
</tr>
<tr>
<td>ICOMP = 3,</td>
<td></td>
</tr>
<tr>
<td>IRSTIN = 0, NFILE = 0, NSAVED = 0,</td>
<td></td>
</tr>
<tr>
<td>JRSTOT = 10,</td>
<td></td>
</tr>
<tr>
<td>JRSTOT = 10,</td>
<td></td>
</tr>
<tr>
<td>&amp;END</td>
<td></td>
</tr>
<tr>
<td>&amp;LIST1</td>
<td></td>
</tr>
<tr>
<td>IHSTAG = 1,</td>
<td></td>
</tr>
<tr>
<td>IBOUND = 40A2,</td>
<td></td>
</tr>
<tr>
<td>IEQBC(1,1,1) = 40A2,</td>
<td></td>
</tr>
<tr>
<td>IEQBC(1,1,2) = 40A2,</td>
<td></td>
</tr>
<tr>
<td>IEQBC(1,1,3) = 40A2,</td>
<td></td>
</tr>
<tr>
<td>IEQBC(1,1,4) = 40A16,</td>
<td></td>
</tr>
<tr>
<td>IEQBC(1,1,5) = 40A8,</td>
<td></td>
</tr>
<tr>
<td>JEQBC(1,1,1) = 11,12,38A11,</td>
<td></td>
</tr>
<tr>
<td>JEQBC(1,1,2) = 2,12,18A2,20A11,</td>
<td></td>
</tr>
<tr>
<td>JEQBC(1,1,3) = 11,12,18A11,20A2,</td>
<td></td>
</tr>
<tr>
<td>JEQBC(1,1,4) = 14,12,38A14,</td>
<td></td>
</tr>
<tr>
<td>JEQBC(1,1,5) = 17,12,38A17,</td>
<td></td>
</tr>
<tr>
<td>&amp;END</td>
<td></td>
</tr>
<tr>
<td>&amp;LIST2</td>
<td></td>
</tr>
<tr>
<td>LREF = 1.0,</td>
<td></td>
</tr>
<tr>
<td>REPL = 1.6685E+06,</td>
<td></td>
</tr>
<tr>
<td>MINE = 2.0,</td>
<td></td>
</tr>
<tr>
<td>PINE = 2864.0,</td>
<td></td>
</tr>
<tr>
<td>PR = 0.71,</td>
<td></td>
</tr>
<tr>
<td>IUNITS = 2,</td>
<td></td>
</tr>
<tr>
<td>&amp;END</td>
<td></td>
</tr>
<tr>
<td>&amp;LIST3</td>
<td></td>
</tr>
<tr>
<td>TWOD = .FALSE.,</td>
<td></td>
</tr>
<tr>
<td>XO(1) = 0.2,0.2,</td>
<td></td>
</tr>
<tr>
<td>T2(1) = 3.0,3.0,</td>
<td></td>
</tr>
<tr>
<td>NE = 50,50,</td>
<td></td>
</tr>
<tr>
<td>IGEOM = 1,</td>
<td></td>
</tr>
<tr>
<td>DELX = 0.01,</td>
<td></td>
</tr>
<tr>
<td>IAP = 1,</td>
<td></td>
</tr>
<tr>
<td>DXMIN = 0.01,</td>
<td></td>
</tr>
<tr>
<td>DXMAX = 0.01,</td>
<td></td>
</tr>
<tr>
<td>AP = 1.0,</td>
<td></td>
</tr>
<tr>
<td>NS = 10,</td>
<td></td>
</tr>
<tr>
<td>XENTR = 0.0,</td>
<td></td>
</tr>
<tr>
<td>IFBW(1,1) = 5,48A3,5,50A2,</td>
<td></td>
</tr>
<tr>
<td>3,48A1,3,50A2,</td>
<td></td>
</tr>
<tr>
<td>3,48A1,3,50A2,</td>
<td></td>
</tr>
<tr>
<td>3,48A1,3,50A2,</td>
<td></td>
</tr>
</tbody>
</table>
3, 48A1, 3, 50A2,
3, 48A1, 3, 50A2,
3, 48A1, 3, 50A2,
3, 48A1, 3, 50A2,
3, 48A1, 3, 50A2,
3, 48A1, 3, 50A2,
3, 48A1, 3, 50A2,
3, 48A1, 3, 50A2,
NBRKX = 1,
XBRKX(1) = 0.015,
IBRMN = 17*18,2*1,481*0,
IBRMX = 19*19,481*0,
EDGE = 0,
&END
&LIST4
IVISC = 3,
IPROF = 1,
IMIXL = 1,
&END
&LIST5
PRINT = 5,
IPLCT = 5,
&END
&LISTR
IFBW(1,1) = 5,16*3,2*5,30*3,5,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
3,16*1,2*3,30*1,3,50*2,
5,16*3,5,3,30*1,3,50*2,
5,17*3,4,30*1,3,50*2,
IBOUND = 2,2*1,2,3*1,3*2,10*2,2,2*1,2,3*1,3*2,10*2,
JEQBC(1,1,1) = 3*11,12,3*11,12,11,12,30*11,
JEQBC(1,1,2) = 3*2,12,3*2,12,2,12,10*2,20*11,
JEQBC(1,1,3) = 3*11,12,3*11,12,11,12,10*11,20*2,
JEQBC(1,1,4) = 3*14,12,3*14,12,14,12,30*14,
JEQBC(1,1,5) = 3*17,12,3*17,12,17,12,30*17,

&END
EOF
A new method has been developed for two and three-dimensional computations of viscous supersonic flows with embedded subsonic regions adjacent to solid boundaries. The approach employs a reduced form of the Navier-Stokes equations which allows solution as an initial-boundary value problem in space, using an efficient noniterative forward marching algorithm. Numerical instability associated with forward marching algorithms for flows with embedded subsonic regions is avoided by approximation of the reduced form of the Navier-Stokes equations in the subsonic regions of the boundary layers. Supersonic and subsonic portions of the flow field are simultaneously calculated by a consistently split linearized block implicit computational algorithm. The results of computations for a series of test cases relevant to internal supersonic flow is presented and compared with data. Comparison between data and computation are in general excellent thus indicating that the computational technique has great promise as a tool for calculating supersonic flow with embedded subsonic regions. Finally, a User's Manual is presented for the computer code used to perform the calculations.