1987

Application of an upwind algorithm to the parabolized Navier-Stokes equations

Scott Leroy Lawrence
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Part of the Aerospace Engineering Commons

Recommended Citation
Lawrence, Scott Leroy, "Application of an upwind algorithm to the parabolized Navier-Stokes equations " (1987). Retrospective Theses and Dissertations. 8555.
https://lib.dr.iastate.edu/rtd/8555

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
INFORMATION TO USERS

While the most advanced technology has been used to photograph and reproduce this manuscript, the quality of the reproduction is heavily dependent upon the quality of the material submitted. For example:

- Manuscript pages may have indistinct print. In such cases, the best available copy has been filmed.

- Manuscripts may not always be complete. In such cases, a note will indicate that it is not possible to obtain missing pages.

- Copyrighted material may have been removed from the manuscript. In such cases, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, and charts) are photographed by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is also filmed as one exposure and is available, for an additional charge, as a standard 35mm slide or as a 17”x 23” black and white photographic print.

Most photographs reproduce acceptably on positive microfilm or microfiche but lack the clarity on xerographic copies made from the microfilm. For an additional charge, 35mm slides of 6”x 9” black and white photographic prints are available for any photographs or illustrations that cannot be reproduced satisfactorily by xerography.
Lawrence, Scott Leroy

APPLICATION OF AN UPWIND ALGORITHM TO THE PARABOLIZED NAVIER-STOKES EQUATIONS

Iowa State University PH.D. 1987

University Microfilms International 300 N. Zeeb Road, Ann Arbor, MI 48106
PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark ✔.

1. Glossy photographs or pages
2. Colored illustrations, paper or print
3. Photographs with dark background
4. Illustrations are poor copy
5. Pages with black marks, not original copy
6. Print shows through as there is text on both sides of page
7. Indistinct, broken or small print on several pages✔
8. Print exceeds margin requirements
9. Tightly bound copy with print lost in spine
10. Computer printout pages with indistinct print
11. Page(s) lacking when material received, and not available from school or author.
12. Page(s) seem to be missing in numbering only as text follows.
13. Two pages numbered. Text follows.
14. Curling and wrinkled pages
15. Dissertation contains pages with print at a slant, filmed as received
16. Other

University Microfilms International
Application of an upwind algorithm to the parabolized Navier-Stokes equations

by

Scott Leroy Lawrence

A Dissertation Submitted to the Graduate Faculty in Partial Fulfillment of the Requirements for the Degree of DOCTOR OF PHILOSOPHY

Major: Aerospace Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa
1987
# TABLE OF CONTENTS

| I.  | INTRODUCTION                                      | 1 |
| II. | GOVERNING EQUATIONS                              | 4 |
|     | A. Navier-Stokes Equations                       | 4 |
|     | B. Coordinate Transformation                     | 5 |
|     | C. Parabolized Navier-Stokes Equations           | 6 |
|     | D. Integral Form of the Equations                | 8 |
| III.| NUMERICAL INTEGRATION APPROACH                   | 10 |
|     | A. Discretization of the Flow Field              | 10 |
|     | B. Discussion of Finite-Volume and Finite-Difference Formulations | 14 |
|     | C. Definition of Cell Geometry/Computational Metrics | 16 |
|     | D. Streamwise Numerical Flux Definition           | 17 |
|     | E. Crossflow Numerical Flux Definition, Inviscid | 19 |
|     | 1. First-Order Upwind Scheme                     | 19 |
|     | 2. Second-Order Upwind Scheme                    | 22 |
|     | 3. Treatment of Expansion Shocks                 | 24 |
|     | F. Crossflow Numerical Flux Definition, Viscous  | 25 |
|     | G. Implicit Algorithm                             | 26 |
|     | H. Boundary Conditions                           | 27 |
|     | I. Grid Generation                               | 30 |
| IV. | TWO-DIMENSIONAL NUMERICAL RESULTS                | 32 |
|     | A. Test Case I. Flat Plate Boundary-Layer         | 32 |
|     | B. Test Case II. Hypersonic Compression Corner    | 37 |
|     | C. Test Case III. Hypersonic Inlet               | 45 |
| V.  | THREE-DIMENSIONAL NUMERICAL RESULTS              | 52 |
|     | A. Test Case I. Circular Cone at Angle of Attack  | 52 |
|     | B. Test Case II. Hypersonic All-Body Vehicle     | 54 |
| VI. | CONCLUDING REMARKS                               | 72 |
| VII.| REFERENCES                                       | 74 |
| VIII.| ACKNOWLEDGEMENTS                                 | 78 |
| IX. | APPENDIX A: NAVIER-STOKES FLUX VECTORS           | 79 |
| X.  | APPENDIX B: NAVIER-STOKES FLUX-JACOBIANS         | 81 |
| XI. | APPENDIX C: DECOMPOSITION OF THE $D$ MATRIX      | 86 |
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Finite-volume geometry</td>
<td>11</td>
</tr>
</tbody>
</table>
| 2      | Boundary cells  
   a) Wall boundary; b) Symmetry boundary | 29   |
| 3      | Comparison of velocity profiles | 33   |
| 4      | Comparison of temperature profiles | 34   |
| 5      | Comparison of skin friction coefficients | 35   |
| 6      | Comparison of heat transfer coefficients | 35   |
| 7      | Compression corner test case | 38   |
| 8      | Computational grid | 38   |
| 9      | Comparison of wall pressure coefficients | 40   |
| 10     | Comparison of heat transfer coefficients | 40   |
| 11     | Comparison of wall pressure coefficients, $\alpha = 0.86^\circ$ | 42   |
| 12     | Comparison of heat transfer coefficients, $\alpha = 0.86^\circ$ | 42   |
| 13     | Comparison of computed pressure contours  
   a) Beam-Warming scheme; b) present method | 43   |
| 14     | Comparison of computed Mach number contours  
   a) Beam-Warming scheme; b) present method | 44   |
| 15     | Hypersonic inlet test case | 47   |
| 16     | Computational grid | 47   |
| 17     | Comparison of computed pressure contours  
   a) Beam-Warming scheme; b) present method | 49   |
| 18     | Comparison of computed Mach number contours  
   a) Beam-Warming scheme; b) present method | 50   |
| 19     | Comparison of wall pressure coefficients | 51   |
| 20     | Comparison of heat transfer coefficients | 51   |
iv

Page

53
55
56
57
58
59
60
62
63
65
66
67
68
69
70

Figure 21. Computational grid, $\alpha = 20^\circ$ .......................... 53
Figure 22. Comparison of flow-field geometries, $\alpha = 12^\circ$ ............... 55
Figure 23. Comparison of flow-field geometries, $\alpha = 20^\circ$ ............... 56
Figure 24. Comparison of flow-field geometries, $\alpha = 24^\circ$ ............... 57
Figure 25. Comparison of pressure coefficients, $\alpha = 12^\circ$ ............... 58
Figure 26. Comparison of pressure coefficients, $\alpha = 20^\circ$ ............... 59
Figure 27. Comparison of pressure coefficients, $\alpha = 24^\circ$ ............... 60
Figure 28. Schematic of the Ames All-Body vehicle .......................... 62
Figure 29. Computational grid, $\alpha = 0^\circ$, $x = 0.8m.$ ........................ 63
Figure 30. Symmetry plane Mach contours ........................................ 65
Figure 31. Crossflow plane Mach contours, $x = 0.8m.$ .......................... 66
Figure 32. Computational grid, $\alpha = 10^\circ$, $x = 0.8m.$ ........................ 67
Figure 33. Effect of the entropy fix on stagnation pressure
   a) without entropy fix; b) with entropy fix .............................. 68
Figure 34. Symmetry plane Mach contours, $\alpha = 10^\circ$ ...................... 69
Figure 35. Crossflow plane Mach contours, $x = 0.8m.$ .......................... 70
I. INTRODUCTION

The parabolized Navier-Stokes (PNS) equations have been used extensively to compute complex steady, supersonic, viscous flow fields (Refs. 1-3 are a few notable examples). Solutions to these equations are obtained by marching in space rather than time and are, therefore, obtained much more efficiently than are solutions to the unsteady Navier-Stokes equations. Unlike the boundary-layer equations, however, the PNS equations contain all of the terms of the Euler equations and, as a consequence, the interaction between the viscous and inviscid portions of the flow field are automatically taken into account.

The PNS equations have been integrated by using a variety of finite-difference schemes. Currently, refinements of the noniterative, implicit, approximate-factorization schemes developed by Vigneron et al. [1] and Schiff and Steger [2] are the state-of-the-art methods for solving the PNS equations. These schemes are based on a class of alternating-direction implicit (ADI) schemes developed by Lindemuth and Killeen [3], McDonald and Briley [4], and Beam and Warming [5] to solve time-dependent equations such as the unsteady Navier-Stokes equations. One of the major drawbacks of the Beam-Warming type of algorithm is that the central-differencing of fluxes across flow-field discontinuities tends to introduce errors into the solution in the form of local flow-property oscillations. In order to control these oscillations some type of artificial dissipation is required. The correct magnitude of this added "smoothing" is generally left for the user to specify through a trial-and-error process.

The design of future hypersonic flight vehicles will depend heavily on computational fluid dynamics for the prediction of aerodynamic and thermodynamic loads, as well as engine performance. One of the features that characterizes the
hypersonic flow regime is the presence of strong shock waves generated by the vehicle and by protuberances from the main body such as wings, canopies, and engine inlets. In the numerical calculation of these flows, the outer shock wave can easily be "fitted"; however, shocks generated within the main shock layer must be "captured" by the numerical algorithm. Thus, a need exists for an efficient computational tool that can easily and accurately resolve flow fields containing discontinuities. Upwind algorithms have received a great deal of attention in recent years for application to the unsteady Euler and Navier-Stokes equations owing to their exceptional shock-capturing capabilities. Until now, the application of upwind schemes has been confined to problems in which the unsteady Euler or Navier-Stokes equations are either marched in the time-direction (e.g., Refs. 6-9) or relaxed through a pseudotime variable (e.g., Refs. 10-13). In either case, the function of the upwinding is to locally model the temporal dispersion of flow-field discontinuities.

In the present study, which involves the integration of the PNS equations, the spatial propagation of flow-field information is locally modeled using a steady version of Roe's scheme [14]. The algorithm is implicit as well as second-order accurate in the crossflow directions. The resulting computer code has been used to compute laminar supersonic and hypersonic flow about three simple two-dimensional geometries and two three-dimensional geometries. The two-dimensional flow fields include flat plate boundary-layer flow, hypersonic flow over a 15° compression corner, and hypersonic flow into a converging inlet. Results are compared with those obtained using other numerical techniques as well as with experimental data. Validation of the method in three dimensions consists of the calculation of hypersonic flow over a 10° half-angle circular cone at three different angles of
attack and comparison of results with experimental measurements. In addition, hypersonic flows over an elliptic cone-based hypersonic vehicle configuration at two angles of attack have been computed and results are presented.
II. GOVERNING EQUATIONS

A. Navier-Stokes Equations

The equations which describe the flow of a Newtonian fluid in three dimensions, neglecting body forces and heat sources, can be written in differential form with respect to a Cartesian coordinate system as

\[
\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial (\mathbf{E}_i - \mathbf{E}_v)}{\partial x} + \frac{\partial (\mathbf{F}_i - \mathbf{F}_v)}{\partial y} + \frac{\partial (\mathbf{G}_i - \mathbf{G}_v)}{\partial z} = 0
\]

(1)

The dependent vector, \( \mathbf{U} \), the inviscid flux vectors, \( \mathbf{E}_i, \mathbf{F}_i, \) and \( \mathbf{G}_i \), and the viscous flux vectors, \( \mathbf{E}_v, \mathbf{F}_v, \) and \( \mathbf{G}_v \), are defined in Appendix A. In these equations, \( p \) is the nondimensional pressure; \( \rho \) is the density; \( u, v, \) and \( w \) are the velocity components in the \( x, y, \) and \( z \) directions, respectively; \( e \) is the internal energy; \( \tau \) is the viscous stress; and \( q \) is the heat conduction rate. The nondimensionalization has been performed (dimensional quantities are denoted by a tilde) in the following manner:

\[
\begin{align*}
  t &= \frac{\tilde{t}}{\tilde{L}/V_\infty} \\
  x &= \frac{\tilde{x}}{\tilde{L}} \\
  y &= \frac{\tilde{y}}{\tilde{L}} \\
  z &= \frac{\tilde{z}}{\tilde{L}} \\
  \rho &= \frac{\rho}{\rho_\infty} \\
  u &= \frac{\tilde{u}}{V_\infty} \\
  v &= \frac{\tilde{v}}{V_\infty} \\
  w &= \frac{\tilde{w}}{V_\infty} \\
  e &= \frac{\tilde{e}}{V_\infty^2} \\
  p &= \frac{\tilde{p}}{\rho_\infty V_\infty^2} \\
  T &= \frac{\tilde{T}}{T_\infty} \\
  \mu &= \frac{\tilde{\mu}}{\mu_\infty}
\end{align*}
\]

where \( \tilde{L} \) is the reference length of the flowfield (taken to be unity in the present computer code).

Through the use of the Prandtl number (assumed constant), the thermal conductivity is expressed in terms of the molecular viscosity and doesn't explicitly appear in the heat conduction terms of the energy equation. The molecular viscosity is computed according to Sutherland's law

\[
\mu = T^{3/2} \left( \frac{1 + 110.4/T_\infty}{T + 110.4/T_\infty} \right)
\]
where $T_\infty$ represents the temperature in degrees Kelvin.

The system is closed using the perfect gas equations of state which are written in nondimensional form as

\[ p = (\gamma - 1)pe \]
\[ T = \gamma M_\infty^2 \frac{p}{\rho} \]

B. Coordinate Transformation

The discretization of Eq. 1 over a body oriented system of grid points is generally considered impractical unless the equation is expressed in terms of a body oriented system of coordinates. In the present study, a coordinate transformation of the general form

\[ \xi = \xi(x, y, z) \]
\[ \eta = \eta(x, y, z) \]
\[ \zeta = \zeta(x, y, z) \]

is applied. Written with respect to this coordinate system and expressed in strong-conservation-law form, the governing equations take the form

\[ \frac{\partial U'}{\partial t} + \frac{\partial E'}{\partial \xi} + \frac{\partial F'}{\partial \eta} + \frac{\partial G'}{\partial \zeta} = 0 \]

where

\[ U' = \frac{U}{J} \]
\[ E' = \left( \frac{\xi}{J} \right) (E_i - E_v) + \left( \frac{\xi}{J} \right) (F_i - F_v) + \left( \frac{\xi}{J} \right) (G_i - G_v) \]
\[ F' = \left( \frac{\eta}{J} \right) (E_i - E_v) + \left( \frac{\eta}{J} \right) (F_i - F_v) + \left( \frac{\eta}{J} \right) (G_i - G_v) \]
\[ G' = \left( \frac{\zeta}{J} \right) (E_i - E_v) + \left( \frac{\zeta}{J} \right) (F_i - F_v) + \left( \frac{\zeta}{J} \right) (G_i - G_v) \]

and $J$ is the Jacobian of the transformation, given by

\[ J = \frac{\partial(\xi, \eta, \zeta)}{\partial(x, y, z)} \]
The metrics are given by the expressions

\[
\begin{align*}
\left( \frac{\xi_x}{J} \right) &= y_\eta z_\eta - y_\eta z_\eta \\
\left( \frac{\eta_y}{J} \right) &= -(y_\xi z_\xi - y_\eta z_\xi) \\
\left( \frac{\xi_y}{J} \right) &= -(x_\eta z_\eta - x_\eta z_\eta) \\
\left( \frac{\xi_z}{J} \right) &= x_\eta y_\xi - x_\eta y_\xi
\end{align*}
\]

and the Jacobian, \( J \), can be calculated using

\[
J^{-1} = x_\xi (y_\eta z_\eta - y_\eta z_\eta) - x_\eta (y_\xi z_\xi - y_\eta z_\xi) + x_\xi (y_\xi z_\eta - y_\eta z_\xi)
\]

The velocity and temperature derivatives of the Cartesian viscous stress and heat transfer terms are evaluated with respect to the new coordinates using the standard chain rule form of differentiation:

\[
\begin{align*}
\frac{\partial}{\partial x} &= \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} + \xi_z \frac{\partial}{\partial \zeta} \\
\frac{\partial}{\partial y} &= \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta} + \xi_z \frac{\partial}{\partial \zeta} \\
\frac{\partial}{\partial z} &= \xi_z \frac{\partial}{\partial \xi} + \eta_z \frac{\partial}{\partial \eta} + \xi_z \frac{\partial}{\partial \zeta}
\end{align*}
\]

C. Parabolized Navier-Stokes Equations

The Navier-Stokes equations are parabolized with respect to the streamwise coordinate direction by first making the following assumptions: 1) the flow is steady, and 2) the viscous derivatives in the streamwise direction are negligible in comparison with those in the crossflow directions. The latter assumption is generally considered valid for high Reynolds number flows. Flowfields for which the above assumptions are valid, satisfy the reduced system of partial differential equations

\[
\frac{\partial \mathbf{E}}{\partial \xi} + \frac{\partial \mathbf{F}}{\partial \eta} + \frac{\partial \mathbf{G}}{\partial \zeta} = 0
\] (3a)
where
\[ E = \left( \frac{\xi_x}{J} \right) E_i + \left( \frac{\xi_y}{J} \right) F_i + \left( \frac{\xi_z}{J} \right) G_i \]
\[ F = \left( \frac{\eta_x}{J} \right) (E_i - E_v) + \left( \frac{\eta_y}{J} \right) (F_i - F_v) + \left( \frac{\eta_z}{J} \right) (G_i - G_v) \] (3b)
\[ G = \left( \frac{\xi_x}{J} \right) (E_i - E_v) + \left( \frac{\xi_y}{J} \right) (F_i - F_v) + \left( \frac{\xi_z}{J} \right) (G_i - G_v) \]

The superscript asterisk on the viscous flux vectors indicates that derivatives with respect to \( \xi \) have been eliminated.

Equation 3a is a mixed set of hyperbolic-parabolic equations with respect to the \( \xi \) coordinate direction provided that: 1) the velocity outside of the boundary layer is supersonic, 2) the streamwise velocity component is everywhere greater than zero, and 3) the pressure gradient term in the streamwise momentum equation is either omitted or treated with some other technique suitable to suppress "departure behavior". The presence of the entire streamwise pressure gradient term permits information to be propagated upstream through the subsonic portion of the boundary layer. If a space-marching procedure is used to solve Eq. 3a, an ill-posed boundary value problem results since there is no mechanism for imposing a downstream boundary condition. The result is that in many cases exponentially growing solutions (departure solutions) are encountered.

A number of techniques have been proposed to avoid the difficulty described above. For this study, the approach developed by Vigneron et al. [1] is used. This approach involves separating the streamwise flux vector into two parts,

\[ \tilde{E}_i = \tilde{E}^* + \tilde{E}^p \] (4)

where
\[ \tilde{E}^* = \left[ \rho \hat{U}, \rho u \hat{U} + \left( \frac{\xi_x}{J} \right) \rho w \hat{U}, \rho \omega w \hat{U} + \left( \frac{\xi_y}{J} \right) \rho \omega w \hat{U}, \rho \omega w \hat{U} + \left( \frac{\xi_z}{J} \right) \omega p, (E_t + p) \hat{U} \right]^T \]
\[ \tilde{E}^p = (1 - \omega)p[0, 0, \left( \frac{\xi_x}{J} \right), \left( \frac{\xi_y}{J} \right), \left( \frac{\xi_z}{J} \right), 0]^T \] (5)
The equations that result from the substitution of Eq. 4 into Eq. 3a are referred to as the PNS equations, and they can be written as

\[
\dot{U} = \left( \frac{\xi_z}{J} \right) u + \left( \frac{\xi_y}{J} \right) v + \left( \frac{\xi_x}{J} \right) w
\]

If Eq. 6 is subjected to an eigenvalue analysis, it can be shown [1] that the system is hyperbolic-parabolic with respect to the new dependent vector \( \dot{\mathbf{E}} \), provided that \( \omega \) is specified according to the relation

\[
\omega = \min \left[ 1, \frac{\sigma \gamma M^2_{\xi}}{1 + (\gamma - 1) M^2_{\xi}} \right]
\]

where, \( M_{\xi} \) is the Mach number in the \( \xi \)-direction and \( \sigma \) is a safety factor included to provide for nonlinearities not accounted for in the analysis. The gradient of \( \dot{\mathbf{E}}^p \) represents that part of the pressure gradient that is responsible for introducing ellipticity into the Eq. 3a and, therefore, is usually neglected or treated as a source term so that Eq. 6 becomes hyperbolic-parabolic in nature.

D. Integral Form of the Equations

The laws of conservation of mass, momentum, and energy over a volume \( V \) bounded by a surface \( S \), can be expressed in integral form as

\[
\frac{\partial}{\partial t} \int_V U dV + \int_S (\mathbf{H} \cdot \mathbf{n}) dS = 0
\]

which for steady flow reduces to

\[
\int_S (\mathbf{H} \cdot \mathbf{n}) dS = 0
\]
The tensor $\overline{H}$ can be written in terms of the Cartesian fluxes by

$$\overline{H} = (E_i - E_v)i + (F_i - F_v)j + (G_i - G_v)k$$

This representation of the governing equations is useful in the development of a finite-volume-based numerical integration scheme. Finite-volume schemes have the advantage of being somewhat more flexible than finite-difference schemes which are generally restricted in application to regularly shaped, hexahedral (in three dimensions) grid cells. For this reason and others which will be discussed in the following chapter, the finite-volume approach has been adopted in the development of the present algorithm.
III. NUMERICAL INTEGRATION APPROACH

A. Discretization of the Flow Field

In the numerical integration of Eq. 6, the solution which is obtained is actually that for a large system of algebraic equations which, in character, represents the more compact but unsolvable system of partial differential equations. The solution then takes the form of mesh point values of the flow properties (finite-difference method) or local area averages of flow properties (finite-volume method). As stated in the previous chapter, the present work utilizes a finite-volume method in obtaining an approximate solution to Eq. 6. Consequently, the development of the algebraic system begins by discretizing the region of interest into small but finite hexahedrons such as the one illustrated schematically in Fig. 1. Hexahedral cells are used here for the simplicity of the resulting algorithm; however, as alluded to in the previous chapter, the finite-volume formulation is not restricted to the use of such a well-ordered mesh.

Since the numerical solution is to be acquired with a space-marching procedure, the region is discretized by successively adding discrete slabs of thickness $\Delta \xi$ as the solution proceeds. The $n$th slab ($n$ is the index of the $\xi$-coordinate) is bounded by the two two-dimensional ($\eta, \zeta$) systems of grid points at $n$ and $n + 1$. Using the terminology of Vinokur [15], these are referred to as the primary grids. The method with which the primary grids are generated in the present work is discussed in a following section. The vertices of each cell are located at mesh points of the primary grids ($j$ and $k$ are used as the indices of the $\eta$- and $\zeta$-coordinate directions, respectively) and are connected by straight-line segments. Given the two primary grids, secondary grids can be defined by averaging coordinates of the points that define the constant-$\xi$ cell faces. The numerical integration scheme
Figure 1. Finite-volume geometry
produces area-average flow properties, which are assigned to the locations of the secondary grid points. In order to describe the present finite-volume algorithm in the more familiar notation of finite differences, integer values of the coordinate indices are assigned to the secondary grid points, although in practice, the computer code only uses these points in the presentation of results.

The ability of the algorithm to correctly represent weak solutions is provided for by imposing local flux conservation through the application of Eq. 7 to the cell of Fig. 1. If it is assumed that the tensor $\mathbf{H}$ remains constant across each face of the cell, Eq. 7 takes the discretized form

$$
\mathbf{H}_{k,l} \cdot dS^n_{k,l} + \mathbf{H}^{n+\frac{1}{2}}_{k-\frac{1}{2},l} \cdot dS^n_{k-\frac{1}{2},l} + \mathbf{H}^{n+\frac{1}{2}}_{k,\frac{l-1}{2}} \cdot dS^n_{k,\frac{l-1}{2}} + \\
\mathbf{H}^{n+\frac{1}{2}}_{k,l+\frac{1}{2}} \cdot dS^n_{k,l+rac{1}{2}} + \mathbf{H}^{n+\frac{1}{2}}_{k+\frac{1}{2},l} \cdot dS^n_{k+\frac{1}{2},l} + \mathbf{H}^{n+\frac{1}{2}}_{k+l+rac{1}{2}} \cdot dS^n_{k+l+rac{1}{2}} = 0
$$

(9)

where the indexing of the cell faces is as follows:

- $dS_{ABC} \leftrightarrow dS^n_{k,l}$
- $dS_{DCH} \leftrightarrow dS^{n+\frac{1}{2}}_{k-\frac{1}{2},l}$
- $dS_{EFG} \leftrightarrow dS^{n+\frac{1}{2}}_{k,\frac{l-1}{2}}$
- $dS_{CG} \leftrightarrow dS^{n+\frac{1}{2}}_{k+\frac{1}{2},l}$
- $dS_{AEH} \leftrightarrow dS^{n+\frac{1}{2}}_{k+l+\frac{1}{2}}$
- $dS_{BC} \leftrightarrow dS^{n+\frac{1}{2}}_{k-\frac{1}{2},l+\frac{1}{2}}$
- $dS_{EF} \leftrightarrow dS^{n+\frac{1}{2}}_{k,\frac{l-1}{2}+\frac{1}{2}}$
- $dS_{AD} \leftrightarrow dS^{n+\frac{1}{2}}_{k+l+\frac{1}{2}}$

Expressing cell-face normals with respect to a Cartesian coordinate system as

- $dS^{n+1}_{k,l} = (l_1)^{n+1}_{k,l}i + (m_1)^{n+1}_{k,l}j + (n_1)^{n+1}_{k,l}k$
- $dS^{n+\frac{1}{2}}_{k+\frac{1}{2},l} = (l_2)^{n+\frac{1}{2}}_{k+\frac{1}{2},l}i + (m_2)^{n+\frac{1}{2}}_{k+\frac{1}{2},l}j + (n_2)^{n+\frac{1}{2}}_{k+\frac{1}{2},l}k$
- $dS^{n+\frac{1}{2}}_{k,\frac{l-1}{2}+\frac{1}{2}} = (l_3)^{n+\frac{1}{2}}_{k,\frac{l-1}{2}+\frac{1}{2}}i + (m_3)^{n+\frac{1}{2}}_{k,\frac{l-1}{2}+\frac{1}{2}}j + (n_3)^{n+\frac{1}{2}}_{k,\frac{l-1}{2}+\frac{1}{2}}k$
- $dS^n_{k+1,\frac{l}{2}} = -(l_4)^{n}_{k+1,\frac{l}{2}}i - (m_4)^{n}_{k+1,\frac{l}{2}}j - (n_4)^{n}_{k+1,\frac{l}{2}}k$
- $dS^{n+\frac{1}{2}}_{k-\frac{1}{2},l-\frac{1}{2}} = -(l_5)^{n+\frac{1}{2}}_{k-\frac{1}{2},l-\frac{1}{2}}i - (m_2)^{n+\frac{1}{2}}_{k-\frac{1}{2},l-\frac{1}{2}}j - (n_2)^{n+\frac{1}{2}}_{k-\frac{1}{2},l-\frac{1}{2}}k$
- $dS^{n+\frac{1}{2}}_{k,\frac{l}{2}-\frac{1}{2}} = -(l_6)^{n+\frac{1}{2}}_{k,\frac{l}{2}-\frac{1}{2}}i - (m_3)^{n+\frac{1}{2}}_{k,\frac{l}{2}-\frac{1}{2}}j - (n_3)^{n+\frac{1}{2}}_{k,\frac{l}{2}-\frac{1}{2}}k$
and remembering Eq. 8, Eq. 9 can be written in the form

\[
\begin{align*}
(\hat{\mathbf{E}}_i - \hat{\mathbf{E}}_v)^{n+1} &+ (\hat{\mathbf{F}}_i - \hat{\mathbf{F}}_v)^{n+\frac{1}{2}} + (\hat{\mathbf{G}}_i - \hat{\mathbf{G}}_v)^{n+\frac{1}{2}} - \\
(\hat{\mathbf{E}}_i - \hat{\mathbf{E}}_v)^{n} &- (\hat{\mathbf{F}}_i - \hat{\mathbf{F}}_v)^{n+\frac{1}{2}} - (\hat{\mathbf{G}}_i - \hat{\mathbf{G}}_v)^{n+\frac{1}{2}} = 0 \tag{10a}
\end{align*}
\]

where

\[
\begin{align*}
\hat{\mathbf{E}}_i &= l_1 \mathbf{E}_i + m_1 \mathbf{F}_i + n_1 \mathbf{G}_i \\
\hat{\mathbf{F}}_i &= l_2 \mathbf{E}_i + m_2 \mathbf{F}_i + n_2 \mathbf{G}_i \\
\hat{\mathbf{G}}_i &= l_3 \mathbf{E}_i + m_3 \mathbf{F}_i + n_3 \mathbf{G}_i \\
\hat{\mathbf{E}}_v &= l_1 \mathbf{E}_v + m_1 \mathbf{F}_v + n_1 \mathbf{G}_v \\
\hat{\mathbf{F}}_v &= l_2 \mathbf{E}_v + m_2 \mathbf{F}_v + n_2 \mathbf{G}_v \\
\hat{\mathbf{G}}_v &= l_3 \mathbf{E}_v + m_3 \mathbf{F}_v + n_3 \mathbf{G}_v \tag{10b}
\end{align*}
\]

Comparison of Eqs. 10a and 10b with Eqs. 3a and 3b of the previous chapter indicates that, if viscous derivatives in the streamwise direction are neglected, Eq. 10a may be considered a discretization of Eq. 3a. From this perspective, the metrics of Eq. 3b represent components of the surface normals as follows:

\[
\begin{align*}
\left( \frac{\xi_x}{J} \right) &= l_1 & \left( \frac{\xi_y}{J} \right) &= m_1 & \left( \frac{\xi_z}{J} \right) &= n_1 \\
\left( \frac{\eta_x}{J} \right) &= l_2 & \left( \frac{\eta_y}{J} \right) &= m_2 & \left( \frac{\eta_z}{J} \right) &= n_2 \\
\left( \frac{\xi_x}{J} \right) &= l_3 & \left( \frac{\xi_y}{J} \right) &= m_3 & \left( \frac{\xi_z}{J} \right) &= n_3
\end{align*}
\]

The above association is useful in that it provides a physical interpretation of the transformed equations. Also, a mechanism now exists for transferring knowledge concerning the partial differential equations to the integral form of the equations. Specifically, the Vigneron treatment of the streamwise pressure gradient can be incorporated into the definition of the streamwise numerical flux \(\hat{\mathbf{E}}\) for stable space marching solution of the equations. This will be discussed in Section C of this chapter.
B. Discussion of Finite-Volume and Finite-Difference Formulations

The association between Eqs. 3 and Eqs. 10 indicates that integration of the strong-conservation-law form of the equations is equivalent to enforcing flux conservation on a finite volume. For this reason, when the flowfield is discretized into hexahedral cells, the finite-volume approach and the finite-difference approach are sometimes difficult to distinguish from one another. However, algorithms in which the numerical fluxes of Eq. 10a are developed in a manner consistent with Eq. 9 and with reference to Fig. 1 will generally possess three desirable, “finite-volume” characteristics. First, local conservation is enforced over closed cells. The use of unclosed volumes creates sources and/or sinks in the flowfield which preclude the ability of the algorithm to preserve freestream flow. This condition is stated mathematically as

$$\int_S dS = 0$$

which results from the application of Eq. 7 under uniform flow conditions. Freestream preservation is considered a required characteristic since it represents a linear test of local flux conservation. The second characteristic of finite-volume schemes is that the sum of the local volumes equals the total volume of the flowfield. Finally, the form of Eq. 9 implies that the flow properties of $\overline{H}$ should be specified in a manner which is not strongly dependent on the shape of the neighboring mesh cells. This implication is lost in Eq. 10a.

An example of an algorithm which does not possess any of the above characteristics is the central-differencing of Eq. 6 as conventionally applied to the PNS equations. In the application of this method, the metric terms of Eq. 2 are generally calculated using standard central-differences and this method of evaluating the metrics is associated with volumes that are not closed. Freestream preservation
is generally imposed on the calculation by subtracting freestream conditions from the inviscid flux vectors; however, this only shifts errors away from the freestream rather than eliminating them. That is, this procedure attempts to compensate for flow lost through cracks in the mesh by introducing freestream flow into these openings. These errors cancel in the freestream but do not in the nonlinear regions of the flow.

A more satisfactory solution to the problem would be the use of metrics that are specially differenced to ensure freestream preservation such as those proposed by Pulliam and Steger [16]. Although this method does define closed volumes, the use of these metrics within the framework of Eq. 3a violates the third condition for a finite-volume scheme. That is, when the numerical flux, \((\hat{F}_i)_{k+\frac{1}{2},l}\), is defined in the conventional strong-conservation-law manner as

\[
(\hat{F}_i)_{k+\frac{1}{2},l} = \frac{1}{2} \left[(\hat{F}_i)_{k,l} + (\hat{F}_i)_{k+1,l}\right],
\]

the averaging of the flow properties is intertwined in a nonphysical manner with the definition of the cell-face geometry. In the extreme case of a grid singularity, this averaging can prevent the propagation of any flow-field information from the singularity.

Thus, the finite-volume approach was adopted in this study in order to ensure that flux conservation would be strictly maintained at the local level and to minimize the sensitivity of numerical solutions to any grid irregularities which may be present. The definition of the volume geometry is presented in the following section.
C. Definition of the Cell Geometry/Computational Metrics

As stated earlier, the primary grid points represent cell vertices and the cell edges are taken to be straight-line segments. In general, four points do not define a unique surface; however, each cell-face may be considered to consist of two planar triangles. In this case, the cell-face surface-area vector \( dS \) is provided by summing the triangular area vectors. The resultant vector is not dependent on which cell-face diagonal separates the triangular facets. In Ref. 15, Vinokur gives relatively efficient formulas for calculating these resultant vectors, which, for the three forward cell faces of Fig. 1, yield

\[
\begin{align*}
    dS_{k,i}^{n+1} &= \frac{1}{2} (r_{k+\frac{1}{2},i+\frac{1}{2}}^{n+1} - r_{k-\frac{1}{2},i-\frac{1}{2}}^{n+1}) \times (r_{k+\frac{1}{2},i-\frac{1}{2}}^{n+1} - r_{k-\frac{1}{2},i+\frac{1}{2}}^{n+1}) \\
    dS_{k+\frac{1}{2},i}^{n+1} &= \frac{1}{2} (r_{k+\frac{1}{2},i+\frac{1}{2}}^{n+1} - r_{k+\frac{1}{2},i-\frac{1}{2}}^{n+1}) \times (r_{k+\frac{1}{2},i+\frac{1}{2}}^{n+1} - r_{k+\frac{1}{2},i-\frac{1}{2}}^{n+1}) \\
    dS_{k,i+\frac{1}{2}}^{n+1} &= \frac{1}{2} (r_{k+\frac{1}{2},i+\frac{1}{2}}^{n+1} - r_{k-\frac{1}{2},i+\frac{1}{2}}^{n+1}) \times (r_{k-\frac{1}{2},i+\frac{1}{2}}^{n+1} - r_{k+\frac{1}{2},i+\frac{1}{2}}^{n+1})
\end{align*}
\]

where the vectors \( r \) here represent the position vectors of the primary grid points. The most important characteristic of these cell-face area vectors is that they are associated with a closed volume, that is, they satisfy the geometric conservation law, Eq. 11. Consequently, use of the cell geometry described above guarantees that the resulting algorithm will preserve freestream flow.

The Cartesian components of the surface normal vectors are given in terms of the metrics by

\[
\begin{align*}
    (\xi_x)_{k,i}^{n+1} &= \frac{1}{2} [(y_{i+1} - y_D)(z_H - z_{i+1}) - (y_H - y_{i+1})(z_{i+1} - z_D)] \\
    (\xi_y)_{k,i}^{n+1} &= -\frac{1}{2} [(x_{i+1} - x_D)(z_H - z_{i+1}) - (x_H - x_{i+1})(z_{i+1} - z_D)] \\
    (\xi_z)_{k,i}^{n+1} &= \frac{1}{2} [(x_{i+1} - x_D)(y_H - y_{i+1}) - (x_H - x_{i+1})(y_{i+1} - y_D)]
\end{align*}
\]
Unlike the surface areas, the cell volume is dependent on the manner in which the cell faces are broken up. In the present application, the cell volume is used only in the evaluation of metrics for the calculation of viscous terms; it is evaluated with the formula

\[ J = \frac{1}{3} \left( dS_{k,l}^n + dS_{k,l-\frac{1}{2}}^{n+\frac{1}{2}} + dS_{k+\frac{1}{2},l}^{n+\frac{1}{2}} \right) \cdot \left( x_{k+\frac{1}{2},l-\frac{1}{2}}^{n+1} - x_{k-\frac{1}{2},l+\frac{1}{2}}^{n+1} \right) \]

This formula is consistent in the sense that the geometry of each cell face is the same in the calculation of both the associated cell volumes. As a result, the sum of all of the cell volumes equals the total volume of the region.

D. Streamwise Numerical Flux Definition

In order to obtain a solution to Eq. 10a through a space-marching procedure, it is necessary to suppress the ellipticity that is inherent in the physics of the boundary-layer. This is done by neglecting the streamwise numerical viscous flux, \( \hat{E}_v \), and by introducing the Vigneron technique with the substitution

\[ (\hat{E}_i)^n_{k,l} = \hat{E}^v(dS_{k,l}^n, U_{k,l}^n) + \hat{E}^p(dS_{k,l}^n, U_{k,l}^{n-1}) \]
where the forms of $\hat{E}^-$ and $\hat{E}^p$ are given by Eq. 5, and $dS_{k,l}^n$ and $U_{k,l}^n$ indicate the location where the geometry and the physical variables (including $\omega$), respectively, are evaluated.

To avoid the difficulty of extracting the required flow properties from the flux vector $\hat{E}^-$, and to simplify the application of the implicit algorithm, a change is made in the dependent variable from $\hat{E}^-$ to $U$ through the following linearization

$$\hat{E}^-(dS^n, U^n) = \hat{A}^{n-1} U^n$$

where

$$\hat{A}^{n-1} = \frac{\partial \hat{E}^-(dS^n, U^{n-1})}{\partial U^{n-1}}$$

(see Appendix B for the form of this matrix). The discretized conservation law, Eq. 10a, then takes the form

$$\hat{A}_k^l \delta_{k,l}^{n+1} U_{k,l} = -(\hat{A}_k^l - \hat{A}_k^{n-1}) U_{k,l} - |(\hat{F}_i - \hat{F}_u)_{k+\frac{1}{2},l} - (\hat{F}_i - \hat{F}_u)_{k-\frac{1}{2},l}|$$

$$- |(\hat{G}_i - \hat{G}_o)_{k+\frac{1}{2},l} - (\hat{G}_i - \hat{G}_o)_{k-\frac{1}{2},l}| - |\hat{E}^p(dS_{k,l}^{n+1}, U_{k,l}^n) - \hat{E}^p(dS_{k,l}^n, U_{k,l}^{n-1})|$$

(12)

where

$$\delta_{k,l}^{n+1} U = U_{k,l}^{n+1} - U_{k,l}^n$$

At this level, the algorithm differs from the conventional PNS solver only in the fact that the metrics are evaluated at cell interfaces rather than at grid points. One obtains a central-differencing scheme by simply averaging the adjacent grid-point flow properties for use in the specification of cell-face numerical fluxes. This scheme will possess all of the undesirable shock-capturing characteristics of the conventional algorithm because its differencing stencil is insensitive to the sharply varying flow conditions associated with a shock. In contrast, upwind schemes
possess mechanisms for adapting the weighting within the differencing stencil to flow-field discontinuities. This characteristic is acquired through careful attention to the physics of flow-field information propagation in the development of the upwind numerical fluxes. The manner in which this physics is incorporated into the numerical fluxes of the present algorithm is described in the following sections.

E. Crossflow Numerical Flux Definition, Inviscid

1. First-Order Upwind Scheme

The algorithm developed in this study is based on Roe's scheme [14], but is modified in order to make it suitable for application to space-marching calculations. Roe's scheme belongs to the class of upwind schemes which defines numerical fluxes according to solutions of Riemann problems. With the present algorithm, the inviscid portions of the numerical fluxes are defined according to solutions of what will be referred to as steady, approximate Riemann problems (or StAR problems). The StAR problem is an initial-value problem that simply consists of a linearized version of the governing equations, with step-function initial conditions. The fluxes \( \hat{F}_i \) and \( \hat{G}_i \) are determined separately by splitting the two-dimensional StAR problem associated with the three-dimensional PNS equations into two one-dimensional StAR problems, each of which takes the form

\[
\frac{\partial \hat{E}^*}{\partial \xi} + D_{m+\frac{1}{2}} \frac{\partial \hat{E}^*}{\partial \kappa} = 0
\]  

with initial conditions

\[
\hat{E}^* n(\kappa) = \begin{cases} 
\hat{E}^* (dS_{m+\frac{1}{2}}, Q_m) & \text{where } \kappa < \kappa_{m+\frac{1}{2}}; \\
\hat{E}^* (dS_{m+\frac{1}{2}}, Q_{m+1}) & \text{where } \kappa > \kappa_{m+\frac{1}{2}}.
\end{cases}
\]
The coefficient matrix $D_{m+\frac{1}{2}}$ is defined by

$$D_{m+\frac{1}{2}} = \left( \frac{\kappa_x}{J} \right)_{m+\frac{1}{2}} \frac{\partial E_i}{\partial E^i} + \left( \frac{\kappa_y}{J} \right)_{m+\frac{1}{2}} \frac{\partial F_i}{\partial E^i} + \left( \frac{\kappa_z}{J} \right)_{m+\frac{1}{2}} \frac{\partial G_i}{\partial E^i}$$

In spite of the non-conservative form of Eq. 13, the local shock-capturing capabilities of the algorithm can be retained if the flow properties making up $D_{m+\frac{1}{2}}$ are averaged between the grid points $m$ and $m+1$, so that the relation

$$D_{m+\frac{1}{2}} \left[ \dot{E}^-(dS^n_{m+\frac{1}{2}}, U_{m+1}) - \dot{E}^+(dS^n_{m+\frac{1}{2}}, U_m) \right] =
\left( \frac{\kappa_x}{J} \right)_{m+\frac{1}{2}} \Delta E_i + \left( \frac{\kappa_y}{J} \right)_{m+\frac{1}{2}} \Delta F_i + \left( \frac{\kappa_z}{J} \right)_{m+\frac{1}{2}} \Delta G_i$$

is satisfied. When the flow is supersonic, Roe's averaging [14] of the variables $u$, $v$, $w$, and $h_t$, yields flow properties that satisfy Eq. 14. Presently, the upwind algorithm is applied only outside the sonic line of the flow-field, and the subsonic region of the boundary layer is treated with a central-differencing approach. This approach is taken because, in two dimensions, a degradation in stability was experienced when the upwinding was carried into the subsonic region. This is believed to be caused by a change in the nature of the flux vector introduced by the application of the Vigneron technique which makes strict Roe-averaging impossible. However, the lack of upwinding in the subsonic region does not appear to be a major drawback at this time since the nature of subsonic flow precludes the presence of discontinuities.

The solution to the above approximate Riemann problem consists of four constant-property regions separated by three surfaces of discontinuity emanating from the cell edge, $(\xi^n, \kappa_{m+\frac{1}{2}})$, and having slopes given by the eigenvalues of $D_{m+\frac{1}{2}}$. Of particular interest to the numerical algorithm is the resulting flux
across the $m + \frac{1}{2}$ cell interface. This first-order accurate inviscid flux consists of
an unbiased component plus a first-order upwind dissipation term and is given by

$$
H^i_{m+\frac{1}{2}} = \left( \frac{\kappa_x}{J} \right)_{m+\frac{1}{2}} \frac{1}{2} \left[ (E_i)_m + (E_i)_{m+1} \right] + \left( \frac{\kappa_y}{J} \right)_{m+\frac{1}{2}} \frac{1}{2} \left[ (F_i)_m + (F_i)_{m+1} \right]
$$

$$
+ \left( \frac{\kappa_z}{J} \right)_{m+\frac{1}{2}} \frac{1}{2} \left[ (G_i)_m + (G_i)_{m+1} \right]
$$

$$
- \frac{1}{2} (\text{sgn}D)_{m+\frac{1}{2}} \left[ \left( \frac{\kappa_x}{J} \right)_{m+\frac{1}{2}} \Delta E_i + \left( \frac{\kappa_y}{J} \right)_{m+\frac{1}{2}} \Delta F_i + \left( \frac{\kappa_z}{J} \right)_{m+\frac{1}{2}} \Delta G_i \right]
$$

In this equation, the matrix, $\text{sgn}D$, is defined as

$$
\text{sgn}D = R(\text{sgn}A)R^{-1}
$$

where $R$ is the matrix of right eigenvectors and $\text{sgn}A$ is the diagonal matrix which
has elements

$$
\text{sgn} \lambda^i = \frac{\lambda^i}{|\lambda^i|}
$$

The differencing operator $\Delta$ is the standard forward difference operator. The form
of these eigenvalues and eigenvectors is given in Appendix C. The differencing
operator $\Delta$ is the standard forward-difference operator.

First-order inviscid numerical fluxes in the $\eta$- and $\zeta$-directions are then given
by

$$
(\hat{F}^i)_k+\frac{1}{2},l = H^i_{k+\frac{1}{2},l}
$$

$$
(\hat{G}^i)_k,l+\frac{1}{2} = H^i_{k,l+\frac{1}{2}},
$$

respectively. In the definition of the flux in the $\eta$-direction, $H_{k+\frac{1}{2},l}$ is given by
inserting $\eta$ for $\kappa$ and $k + \frac{1}{2}, l$ for $m + \frac{1}{2}$. Likewise, for the flux in the $\zeta$-direction,
$H_{k,l+\frac{1}{2}}$ would be obtained by replacing $\kappa$ with $\zeta$ and $m + \frac{1}{2}$ with $k, l + \frac{1}{2}$. 

2. Second-Order Upwind Scheme

In extending the algorithm described above to second-order accuracy (in the crossflow directions), we wish to preserve the desirable shock-capturing characteristics while improving the accuracy in the more gradually varying regions of the flow. There are numerous examples of algorithms for solving the time-dependent equations that satisfy these requirements. The approach taken by Chakravarthy and Szema [17] was adapted here for application to the steady equations. This approach was chosen because of its relative simplicity, as well as its observed accuracy and reliability in application to the two-dimensional PNS equations. Also, because this method involves extrapolations only of flow properties and not of the metrics, it is thought to be more compatible with the finite-volume philosophy and, thus, expected to be less sensitive to grid irregularities.

The numerical fluxes in the \( \eta \)- and \( \zeta \)-coordinate directions are given in terms of the generic inviscid flux \( \mathbf{H}^{II} \) as

\[
\begin{align*}
(\tilde{\mathbf{F}}^{II})_{k+\frac{1}{2},l} & = \mathbf{H}_{k+\frac{1}{2},l}^{II}, \\
(\tilde{\mathbf{G}}^{II})_{k,l+\frac{1}{2}} & = \mathbf{H}_{k,l+\frac{1}{2}}^{II}
\end{align*}
\]

where \( \mathbf{H}^{II} \) is of the form

\[
\mathbf{H}_{m+\frac{1}{2}}^{II} = \mathbf{H}_{m+\frac{1}{2}}^{I} + \frac{1 - \phi}{4}(\tilde{\mathbf{F}}^{+}_{1})_{m+\frac{1}{2}} + \frac{1 + \phi}{4}(\tilde{\mathbf{F}}^{+}_{2})_{m+\frac{1}{2}} - \frac{1 + \phi}{4}(\tilde{\mathbf{F}}^{-}_{3})_{m+\frac{1}{2}}
\]  

(17)

The numerical flux changes, \( \tilde{\mathbf{F}} \) and \( \tilde{\mathbf{F}} \), are specified in the following manner.

First, intermediate variables \( (A)_{m+\frac{1}{2}} \) are defined by

\[
\begin{align*}
(A_1)_{m+\frac{1}{2}} & = R_{m+\frac{1}{2}}^{-1} [\mathbf{E}^{*}(dS_{m+\frac{1}{2}}^{n}, U_m) - \mathbf{E}^{*}(dS_{m+\frac{1}{2}}^{n}, U_{m-1})] \\
(A_2)_{m+\frac{1}{2}} & = R_{m+\frac{1}{2}}^{-1} [\mathbf{E}^{*}(dS_{m+\frac{1}{2}}^{n}, U_{m+1}) - \mathbf{E}^{*}(dS_{m+\frac{1}{2}}^{n}, U_m)] \\
(A_3)_{m+\frac{1}{2}} & = R_{m+\frac{1}{2}}^{-1} [\mathbf{E}^{*}(dS_{m+\frac{1}{2}}^{n}, U_{m+2}) - \mathbf{E}^{*}(dS_{m+\frac{1}{2}}^{n}, U_{m+1})]
\end{align*}
\]
where, as before, $R_{m+\frac{1}{2}}^{-1}$ is the matrix of left eigenvectors. These vectors are then limited relative to one another in order to achieve essentially nonoscillatory shock capturing. The elements of the new vectors, $(\tilde{A}_1)_{m+\frac{1}{2}}$, $(\tilde{A}_2)_{m+\frac{1}{2}}$, and $(\tilde{A}_3)_{m+\frac{1}{2}}$ are given by

$$
(\tilde{\alpha}_1)_{m+\frac{1}{2}} = \minmod[(\alpha_1^i)_{m+\frac{1}{2}}, b(\alpha_2^i)_{m+\frac{1}{2}}]
$$

$$
(\tilde{\alpha}_2)_{m+\frac{1}{2}} = \minmod[(\alpha_2^i)_{m+\frac{1}{2}}, b(\alpha_1^i)_{m+\frac{1}{2}}]
$$

$$
(\tilde{\alpha}_3)_{m+\frac{1}{2}} = \minmod[(\alpha_3^i)_{m+\frac{1}{2}}, b(\alpha_2^i)_{m+\frac{1}{2}}]
$$

respectively. The limiting operator is defined by

$$
\minmod[x, y] = \text{sgn}(x) \max[0, \min\{|x|, \frac{2}{\text{sgn}(x)}\}]
$$

and the parameter $b$ is a compression parameter which is usually determined by the accuracy parameter $\phi$ according to the function,

$$
b = \frac{3 - \phi}{1 - \phi}
$$

The flux limiting performs the function of reducing the accuracy of the scheme in the immediate vicinity of flow-field discontinuities so that the overshoots and undershoots characteristic of second-order methods are eliminated.

The numerical flux changes of Eq. 17 are then produced by multiplying the above limited vectors by the eigenvalues and right eigenvectors as follows:

$$
(d\vec{F}_1)_{m+\frac{1}{2}} = R_{m+\frac{1}{2}} (\Lambda_{m+\frac{1}{2}}^+) (\tilde{A}_1)_{m+\frac{1}{2}}
$$

$$
(d\vec{F}_2^+)_{m+\frac{1}{2}} = R_{m+\frac{1}{2}} (\Lambda_{m+\frac{1}{2}}^+) (\tilde{A}_2)_{m+\frac{1}{2}}
$$

$$
(d\vec{F}_2^-)_{m+\frac{1}{2}} = R_{m+\frac{1}{2}} (\Lambda_{m+\frac{1}{2}}^-) (\tilde{A}_2)_{m+\frac{1}{2}}
$$

$$
(d\vec{F}_3^-)_{m+\frac{1}{2}} = R_{m+\frac{1}{2}} (\Lambda_{m+\frac{1}{2}}^-) (\tilde{A}_3)_{m+\frac{1}{2}}
$$
where \((\Lambda_{m+\frac{1}{2}})^+\) and \((\Lambda_{m+\frac{1}{2}})^-\) are the diagonal matrices consisting of the elements

\[
\lambda_{m+\frac{1}{2}}^{i+} = \frac{1}{2} \left( \lambda_{m+\frac{1}{2}}^i + \lambda_{m+\frac{1}{2}}^i \right) \tag{18a}
\]

and

\[
\lambda_{m+\frac{1}{2}}^{i-} = \frac{1}{2} \left( \lambda_{m+\frac{1}{2}}^i - \lambda_{m+\frac{1}{2}}^i \right) \tag{18b}
\]

respectively.

It can be shown that when the nonlimited, inviscid, second-order numerical flux is substituted into the discretized conservation law, Eq. 12, the resulting algorithm has a leading truncation-error term associated with the inviscid terms of form

\[
T.E. = \frac{1}{4} \left( \phi - \frac{1}{3} \right) (\Delta \kappa)^2 \left[ \frac{\kappa_x \partial^3 F_i}{J} \frac{\partial^3 F_i}{\partial \kappa^3} + \frac{\kappa_y \partial^3 F_i}{J} \frac{\partial^3 F_i}{\partial \kappa^3} + \frac{\kappa_z \partial^3 G_i}{J} \frac{\partial^3 G_i}{\partial \kappa^3} \right]
\]

Thus, schemes of varying accuracy can be obtained simply by altering the value of the accuracy parameter, \(\phi\). Although experiments in two dimensions produced results that were relatively insensitive to \(\phi\) (for \(-\frac{1}{2} < \phi < \frac{1}{2}\)), the test calculations presented in this work were performed with the third-order inviscid numerical fluxes, that is, \(\phi = \frac{1}{3}\).

3. Treatment of Expansion Shocks

It is well known that the basic Roe's scheme does not satisfy the entropy condition necessary to prevent the occurrence of expansion shocks or glitches. A number of techniques have been proposed to treat this nonphysical behavior. The technique adopted here, due to Harten [18], consists of replacing the absolute-value operator of Eqs. 16 and 18 with the conditional operator

\[
\psi(z) = \begin{cases} \frac{|z|}{z^2 + \epsilon^2} & \text{where } z \geq \epsilon \\ \frac{\epsilon^2 + z^2}{2\epsilon} & \text{where } z < \epsilon \end{cases}
\]
where $\epsilon$ is a small positive parameter. In addition to eliminating expansion shocks, this technique has been useful in another respect. In some cases, if the eigenvalues are not treated, strong shocks that move slowly relative to the grid may tend to lurch across cell interfaces spawning small, nonphysical contact surfaces. This special treatment of the eigenvalues smooths the shock transition and, thus, eliminates this problem. The drawback is that the technique generally introduces another point into the shock-transition process and, thus, captured shocks may be thickened slightly.

F. Crossflow Numerical Flux Definition, Viscous

Viscous stress and heat transfer effects are currently incorporated in both crossflow directions by using a standard central-differencing approach. The velocity and temperature derivatives at interior cell faces (boundary point treatment will be described in the Boundary Conditions subsection) are approximated by differencing the information from the two adjacent grid points, and the cell-face viscosity is approximated using the average temperature. Chain-rule differentiation allows the $\eta$- and $\zeta$-differences to be transformed into $x$-, $y$-, and $z$-derivatives, and these rates of strain are then converted to stresses with the cell-face viscosity.

The required metric terms are approximated as follows

$$
(\eta_x)_{k+\frac{1}{2},l} = \left( \frac{J_{k,l} + J_{k+1,l}}{2} \right) \left( \frac{\eta_2}{J} \right)_{k+\frac{1}{2},l}, \quad (\zeta_x)_{k,l+\frac{1}{2}} = \left( \frac{J_{k,l} + J_{k,l+1}}{2} \right) \left( \frac{\zeta_2}{J} \right)_{k,l+\frac{1}{2}}
$$

$$
(\eta_y)_{k+\frac{1}{2},l} = \left( \frac{J_{k,l} + J_{k+1,l}}{2} \right) \left( \frac{\eta_2}{J} \right)_{k+\frac{1}{2},l}, \quad (\zeta_y)_{k,l+\frac{1}{2}} = \left( \frac{J_{k,l} + J_{k,l+1}}{2} \right) \left( \frac{\zeta_2}{J} \right)_{k,l+\frac{1}{2}}
$$

$$
(\eta_z)_{k+\frac{1}{2},l} = \left( \frac{J_{k,l} + J_{k+1,l}}{2} \right) \left( \frac{\eta_2}{J} \right)_{k+\frac{1}{2},l}, \quad (\zeta_z)_{k,l+\frac{1}{2}} = \left( \frac{J_{k,l} + J_{k,l+1}}{2} \right) \left( \frac{\zeta_2}{J} \right)_{k,l+\frac{1}{2}}
$$

The results presented in the Numerical Results section were computed neglecting circumferential ($\eta$) viscous derivatives; however, these terms exist in the computer code and can be included with a change in an input parameter.
G. Implicit Algorithm

From Eq. 17, one observes that the second-order numerical flux is made up of the first-order numerical flux plus correction terms. At present, the numerical algorithm evaluates the first-order flux of Eq. 15 at the \( n + 1 \) marching station and lags the correction terms at the \( n \) station. The first-order numerical flux is linearized as shown below.

\[
(H^1)^{n+1}_{m+\frac{1}{2}} = (H^1)^n_{m+\frac{1}{2}} + \left[ \frac{\partial (H^1)_{m+\frac{1}{2}}}{\partial U_{m+1}} \right]^n \delta^{n+1} U_{m+1} + \left[ \frac{\partial (H^1)_{m+\frac{1}{2}}}{\partial U_{m}} \right]^n \delta^{n+1} U_{m}
\]

The Roe-averaged \( \text{sgn}D \) matrix is assumed locally constant for the evaluation of the numerical flux-Jacobians of this equation.

Evaluation of the viscous terms at \( n + 1 \) is also done in a straightforward manner through the following linearizations:

\[
(\hat{F}_v)^{n+1}_{k+\frac{1}{2},l} = (\hat{F}_v)^n_{k+\frac{1}{2},l} + \left[ \frac{\partial (\hat{F}_v)_{k+\frac{1}{2},l}}{\partial U_{k+1,l}} \right]^n \delta^{n+1} U_{k+1,l} + \left[ \frac{\partial (\hat{F}_v)_{k+\frac{1}{2},l}}{\partial U_{k,l}} \right]^n \delta^{n+1} U_{k,l}
\]

\[
(\hat{G}_v)^{n+1}_{k,l+\frac{1}{2}} = (\hat{G}_v)^n_{k,l+\frac{1}{2}} + \left[ \frac{\partial (\hat{G}_v)_{k,l+\frac{1}{2}}}{\partial U_{k,l+1}} \right]^n \delta^{n+1} U_{k,l+1} + \left[ \frac{\partial (\hat{G}_v)_{k,l+\frac{1}{2}}}{\partial U_{k,l}} \right]^n \delta^{n+1} U_{k,l}
\]

The resulting block system of algebraic equations is approximately factored into two block-tridiagonal systems in the conventional manner and the algorithm is then written in the form,

\[
\left[ \tilde{A}_{k,l}+\frac{\partial \delta_{\eta}(\hat{F}_v^1-\hat{F}_v)}{\partial U_{k,l}} + \bar{\delta}_{\eta} \left( \frac{\partial (\hat{F}_v^1-\hat{F}_v)}{\partial U} \right) \right]^n \left[ (\tilde{A}_{k,l})^{-1} \right]^n
\times \left[ \tilde{A}_{k,l}+\frac{\partial \delta_{\zeta}(\hat{G}_v^1-\hat{G}_v)}{\partial U_{k,l}} + \bar{\delta}_{\zeta} \left( \frac{\partial (\hat{G}_v^1-\hat{G}_v)}{\partial U} \right) \right]^n \delta^{n+1} U_{k,l} = RHS^n
\]

(19a)
where
\[ R H S^n = -(\hat{A}^{n-1}_{k,1} - \hat{A}^n_{k,1})U^n_{k,1} - \delta_\gamma (\hat{F}^{n1}_{i} - \hat{F}^n_{v}) - \delta_\gamma (\hat{G}^{n1}_{i} - \hat{G}^n_{v}) - |\hat{E}^{n}(dS^{n+1}_{k,l}, U^n_{k,1}) - \hat{E}^{n}(dS^n_{k,l}, U^{n-1}_{k,1})| \] (19b)

The differencing operators in Eq. 19 are defined as follows:
\[ \delta_\kappa \Phi = \Phi_{m+\frac{1}{2}} - \Phi_{m-\frac{1}{2}} \]
\[ \delta_\kappa \left( \frac{\partial \Psi}{\partial U} \right) \cdot \Phi = \frac{\partial \Psi_{m+\frac{1}{2}}}{\partial U_{m+1}} \Phi_{m+1} - \frac{\partial \Psi_{m-\frac{1}{2}}}{\partial U_{m-1}} \Phi_{m-1} \]

For details on the flux-Jacobian matrices, see Appendix B.

H. Boundary Conditions

With the finite-volume approach used in this work, the region boundaries are located at cell interfaces instead of at the locations where the flow properties are stored (see Fig. 2). Thus, boundary conditions are imposed through the specification of fluxes at the boundary cell faces. Presently, no-slip conditions are applied at the wall (a constant-\( \zeta \) surface) by allowing no flux through the boundary-cell interfaces. Viscous stresses at the wall are evaluated with three-point one-sided differences using the fact that the velocity is zero at the wall. In the case of an isothermal wall, the wall viscosity is computed from the specified wall temperature, and the heat conduction is obtained using a three-point, one-sided temperature difference. If the wall is adiabatic, the temperature at the wall is extrapolated for the calculation of the viscosity, and the wall heat conduction is set to zero. Finally, the wall pressure required for the evaluation of the inviscid fluxes is extrapolated using the zero-gradient extrapolation

\[ P_{k,w} = P_{k,1} \]
The implicit imposition of these boundary conditions requires that we account for the one-sided differences in the linearization. Thus, the viscous flux \( \left( \hat{G}_v \right)_{k,w}^{n+1} \) is linearized as follows:

\[
(\hat{G}_v)_{k,w}^{n+1} = (\hat{G}_v)_{k,w}^n + \left[ \frac{\partial (\hat{G}_v)_{k,w}}{\partial U_{k,1}} \right]^n \delta^{n+1} U_{k,1} + \left[ \frac{\partial (\hat{G}_v)_{k,w}}{\partial U_{k,2}} \right]^n \delta^{n+1} U_{k,2}
\]

The left-hand-side of Eq. 19a must then be altered by replacing the \( \delta \) operator with the boundary operator

\[
\delta_t \left( \frac{\partial \Psi}{\partial U} \right) \Phi = \frac{\partial (\Psi_{m+\frac{1}{2}} - \Psi_{m-\frac{1}{2}})}{\partial U_{m+1}} \Phi_{m+1}
\]

A similar procedure would be required for the inviscid flux if a higher-order extrapolation was applied in approximating the wall pressure.

The flow fields computed thus far have all contained a pitch plane of symmetry at \( z = 0 \). For the evaluation of fluxes at this plane, two auxiliary grid points were added across each symmetry plane, as shown schematically in Fig. 2b, so that the boundary cell interfaces could be treated as interior points. Conditions at the auxiliary points were specified by

\[
U_{KMAX+1,l} = \bar{S} U_{KMAX,l}
\]

\[
U_{KMAX+2,l} = \bar{S} U_{KMAX-1,l}
\]

where

\[
\bar{S} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

This boundary condition was implemented implicitly by imposing the first condition above at the \( n + 1 \) marching station so that

\[
\delta^{n+1} U_{KMAX+1,l} = \bar{S} \delta^{n+1} U_{KMAX,l}
\]
Figure 2. Boundary cells

a) Wall boundary; b) Symmetry boundary
Finally, since in the present calculations all shock waves are captured within the mesh, conditions at the far-field boundary were specified to be those of the freestream.

I. Grid Generation

The primary grids used in the test calculations of this study were generated using a simple algebraic method; this method was used because of its relative speed and simplicity in comparison with methods that involve the solution of partial differential equations. For the simple bodies considered thus far, the method seems to furnish grids that are sufficiently clustered in both the normal and circumferential directions to resolve the important physics of the flow fields.

The grid generation begins by distributing points along the body (both geometries studied were analytically defined) and clustering in the circumferential direction according to the local curvature of the body. Constant- lines are then defined to be straight-line segments oriented along analytically determined body-normal rays and extending to an elliptically shaped outer grid line. Finally, grid points are clustered along the constant- lines according to the Roberts exponential stretching function (see Ref. 19):

\[ z(\xi, \eta, \zeta) = \frac{\beta(\xi, \eta) + 1 - [\beta(\xi, \eta) - 1] \left[ \frac{\beta(\xi, \eta) + 1}{\beta(\xi, \eta) - 1} \right]^{1-\eta}}{\left[ \frac{\beta(\xi, \eta) + 1}{\beta(\xi, \eta) - 1} \right]^{1-\eta} + 1} \]

The position vectors of the grid points are then given by

\[ r(\xi, \eta, \zeta) = r(\xi, \eta, 0) + z(\xi, \eta, \zeta)[r(\xi, \eta, \zeta_{\text{max}}) - r(\xi, \eta, 0)] \]
where $\zeta = 0$ at the body surface and $\zeta = \zeta_{\text{max}}$ at the freestream edge of the grid. The stretching parameter varies linearly with streamwise distance and cubically with circumferential angle in order to allow control of the boundary-layer resolution throughout the flow-field.
IV. TWO-DIMENSIONAL NUMERICAL RESULTS

The two-dimensional version of the upwind algorithm for the PNS equations was validated by applying the new code to three two-dimensional test cases.

A. Test Case I. Flat Plate Boundary-Layer

The first test case, intended to determine the algorithm's ability to compute flowfields dominated by viscous effects, was supersonic, laminar flow over a flat plate. The freestream flow conditions for this case are

\[
\begin{align*}
M_\infty &= 2.0 \\
Re_\infty / L &= 1.65 \times 10^5/m. \\
\hat{T}_\infty = \hat{T}_w &= 221.6K \\
Pr &= 0.72 \\
\gamma &= 1.4
\end{align*}
\]

For the calculation of this test case, two PNS codes were employed: the first used a two-dimensional version of the algorithm described here and the second used a two-dimensional version of the central-differencing algorithm described in Ref. 1. Results of both codes were compared with those obtained from the compressible boundary-layer code of Pletcher [20].

Initial conditions were provided by the boundary-layer code at the streamwise location, \( \bar{z} = 0.305m \). Marching of the PNS codes proceeded from this point with a marching stepsize of \( \Delta \bar{z} = 0.1 \times 10^{-2}m \). The grid normal to the wall for both PNS calculations was equally spaced with \( \Delta \bar{y} = 0.1524 \times 10^{-3} \) m and the top of the grid was kept at a constant height of \( 0.61 \times 10^{-2} \) m.

Profiles of tangential velocity and temperature at the marching station, \( \bar{z} = 0.9144m \), are compared in Figs. 3 and 4, respectively. Figs. 5 and 6 display the
Figure 3. Comparison of velocity profiles
Figure 4. Comparison of temperature profiles
Figure 5. Comparison of skin friction coefficients

Figure 6. Comparison of heat transfer coefficients
streamwise variation of the derivative quantities of skin friction and heat transfer coefficients. The formulas used to compute these quantities are

\[ C_f = \frac{\mu_{wall}}{Re_\infty} \frac{\partial u}{\partial n} \]

\[ C_h = \frac{\mu_{wall}}{PrRe_\infty} \frac{1}{\frac{1}{2}(\gamma - 1)M_\infty^2 + 1 - T_w} \frac{\partial T}{\partial n} \]

where \( n \) represents the distance normal to the wall. In all four of these figures, excellent agreement is observed between the PNS results and those of the boundary-layer code. The upwind code gives slightly better results in this case than does the Beam-Warming code, but this is believed to be due to a difference in treatment of the viscous terms and not the upwinding.

The stepsize used to produce these results corresponds to a maximum Courant number of approximately 50. As the stepsize is increased, the results of both codes gradually deteriorate, but this effect is slightly more pronounced with the upwind algorithm. This seems to indicate that, for flowfields with relatively gradual variations where no added smoothing is required (none was used in the Beam-Warming calculation of this case), central-differencing may be the most appropriate approach. Nevertheless, in this test case, the new algorithm has exhibited a satisfactory ability to compute viscous regions without the upwind dissipation overwhelming the physical viscosity.

An indication of the relative computer effort required by the two PNS codes is given by a comparison of the CPU times and storage involved. The time spent in CPU by the Beam-Warming code is \( 0.92 \times 10^{-4} \) sec./step/grid point on a Cray-XMP computer. As would be expected, the upwind algorithm is slower, requiring \( 0.25 \times 10^{-3} \) sec./step/grid point. The additional time is spent computing and multiplying the eigenvalues and eigenvectors for the evaluation of the upwind
dissipation terms. It should be noted that vectorization was not a high priority in the development of either of these codes. The storage requirements of both codes are very mild since storage is required in only one dimension.

Also of some interest is the programming effort involved in developing the upwind code. The new 2-D code contains approximately 30% more Fortran statements than does the 2-D Beam-Warming code. These additional statements are fairly evenly distributed between the calculation of eigenvalues and eigenvectors, the evaluation of the $dF$'s, and the upwinding of the left-hand-side.

B. Test Case II. Hypersonic Compression Corner

The second test case computed was that of hypersonic laminar flow over a 15 deg wedge. The flow conditions, chosen to correspond with one of the cases studied experimentally by Holden and Moselle [21], are given below

\[
\begin{align*}
M_\infty &= 14.1 \\
\tilde{T}_\infty &= 72.2K \\
Re_\tilde{t} &= 1.04 \times 10^5 \\
\gamma &= 1.4
\end{align*}
\]

where $Re_\tilde{t}$ is the freestream Reynolds number based on the distance from the leading edge to the beginning of the ramp. This flow is supersonic in the inviscid region and exhibits no separation of the boundary-layer. Thus, the space marching procedure is stable. Also, since the flowfield contains an extremely strong shock wave, this case provides a good test of the shock capturing capabilities of the new algorithm. The flow is illustrated schematically in Fig. 7.

The initial conditions for this case were specified using the second-order Roe's scheme code marched with $\Delta \tilde{x} = 0.5 \times 10^{-3} \text{ m}$ from freestream conditions
Figure 7. Compression corner test case

Figure 8. Computational grid
to the downstream station at $\tilde{x} = 0.1 \, m$. Both PNS codes were then restarted from these results and marched further downstream. The grid shown in Fig. 8 is representative of the grid used in the calculations; however, only every third grid line in each direction is printed. Forty-five grid points were distributed in the normal direction with a stretching parameter of 1.08. The marching stepsize downstream of $\tilde{x} = 0.1 m$ was kept constant at $0.2 \times 10^{-2} \, m$. and the calculations were terminated at $\tilde{x} = 0.9 m$.

The streamwise distribution of wall pressure coefficient, defined by

$$C_p = \frac{\tilde{p}_w}{\tilde{\rho}_\infty \tilde{V}_\infty}$$

is shown in Fig. 9 and the heat transfer coefficient distribution is given in Fig. 10. Results of the first- and second-order Roe's schemes and the Beam-Warming scheme are compared with the experimental results of Holden and Moselle. Smoothing terms of the form suggested by Hung and MacCormack [22] were added to the right-hand-side of the Beam-Warming algorithm to control the nonlinear instabilities associated with the strong shock wave of this test case. As anticipated, smoothing was not required for any of the calculations using the new algorithm.

The results indicate that the computed wall pressures are relatively insensitive to changes in algorithm; however, the derivative quantity of heat transfer is noticeably improved by the change from first- to second-order accuracy. The slight qualitative disagreement near the corner between the numerical results and the experiment is due to the single-sweep space marching procedure, that is, the flow upstream is not "warned" of the oncoming compression. The consistent overprediction of both pressure and heat transfer coefficients is not so easily explained. The same trend can be seen in the results of Hung and MacCormack [22] for the
Figure 9. Comparison of wall pressure coefficients

Figure 10. Comparison of heat transfer coefficients
numerical integration of the full unsteady Navier-Stokes equations. One possibility is that, in the experiment, the shock-induced flow approached the model at a small angle of attack. This hypothesis was briefly investigated numerically and the results of calculations performed at an angle of attack of $0.86^\circ (v_\infty = 0.015)$ are given in Figs. 11 and 12. An improvement is realized in both the pressure and heat transfer coefficient distributions, though the latter are still slightly overestimated. Another possible explanation is that the leading edge flow is not being adequately resolved by the present grid and that increasing the number of grid points in that region would produce significantly different results throughout the shock layer. This possibility is currently under investigation. Calculations were also performed using the Beam-Warming scheme with an equation of state given by equilibrium air curve fits [23]. However, because of the very low freestream static temperature, virtually no real gas effects were observed and the results are not included here.

The details of the shock intersection region are illustrated in the contour maps of pressure and Mach number presented in Figs. 13 and 14. These figures reveal an interesting aspect of this test case in that the inviscid flowfield along the line, $\tilde{z} = 0.73m$, represents a large scale version of the steady Riemann problem considered in the development of the new upwind algorithm. The resulting shock wave and expansion fan are evident in the pressure contours and the contact surface appears in the Mach contours. The most noteworthy feature of these figures, however, is that the flowfield resolution by the upwind scheme is, in general, markedly superior to that achieved using the conventional algorithm. Using the new algorithm, the shock transition takes place over one or two grid points with no associated oscillations, whereas the Beam-Warming scheme generates oscillations
Figure 11. Comparison of wall pressure coefficients, $\alpha = 0.86^\circ$

Figure 12. Comparison of heat transfer coefficients, $\alpha = 0.86^\circ$
Figure 13. Comparison of computed pressure contours
   a) Beam-Warming scheme; b) present method
Figure 14. Comparison of computed Mach number contours
a) Beam-Warming scheme; b) present method
which extend several grid points into the shock layer. The oscillatory behavior exhibited near the wall in Fig. 13a is believed to be generated by the central-differencing used in the subsonic region and accentuated by the sharp addition of upwind dissipation experienced at the sonic line. Development of a more satisfactory method of "algorithm transition" is one of the projects currently being investigated.

An effort was undertaken in this case, also, to determine the upper bound on marching stepsize. This study proved the new algorithm to be significantly more robust than the conventional scheme. At large Courant numbers, the smoothing required by the central-differencing scheme to maintain positive pressures at the shock undershoots tended to initiate departure behavior near the wall. This comparison is somewhat suspect due to the fact that smoothing was added in the Beam-Warming code throughout the flowfield. Applying the smoothing only to the supersonic portion of the flow would probably yield an approach more consistent with that taken in the upwind code.

C. Test Case III. Hypersonic Inlet

The final test case is basically an extension of the previous case. The rear of the ramp is turned back to horizontal and a reflection condition is applied to the top of the grid resulting in the case of hypersonic flow into a two-dimensional converging inlet. The ramp angle is, again, 15 deg and the horizontal distance from the compression corner to the top of the ramp is 0.4 m. A schematic illustration of this flowfield is shown in Fig. 15. The freestream conditions that were applied
are given below

\[
\begin{align*}
M_\infty &= 15.0 \\
T_\infty &= 100.0K \\
\widetilde{T}_w &= 1000K \\
Re_l &= 8.0 \times 10^4 \\
Pr &= 0.72
\end{align*}
\]

\(\gamma = 1.4\)

Again, \(\widetilde{Re}_l\) is the Reynolds number based on \(\tilde{l}\), the distance from the leading edge to the beginning of the ramp. This test case is challenging in that it involves many interactions between shock waves and expansion fans and also because it requires the capturing of shock waves that are oblique to the grid, i.e., shock waves that appear to move as the computation proceeds.

Conditions for initializing the PNS calculations for this case were provided in a manner similar to that used in the previous test case. Fig. 16 shows a representation of the grid with the reflected portion included. Here, only every fourth grid line in the \(\eta\) direction and every eighth in the \(\xi\) direction have been printed. Forty-five grid points were again distributed in the \(\eta\) direction with a stretching parameter of 1.08 and the line of symmetry was located at \(\tilde{y} = 0.15m\). The calculations proceeded from initialization at \(\tilde{x} = 0.1m\) to the final marching station at \(\tilde{x} = 1.5m\) using the constant stepsize, \(\Delta \tilde{x} = 0.2 \times 10^{-2}m\).

Computed pressure and Mach number contours in the narrow region of the inlet are presented in Figs. 17 and 18. The shock waves are sharply and smoothly captured with the upwind scheme whereas oscillations persist in calculations performed with the Beam-Warming scheme despite the addition of artificial smoothing. Also, the results shown in Figs. 17a and 18a were obtained in virtually the
Figure 15. Hypersonic inlet test case

Figure 16. Computational grid
first attempt. On the other hand, several runs with the conventional code using different smoothing coefficients were necessary before satisfactory results were obtained.

Wall distributions of pressure and heat transfer coefficient are plotted in Figs. 19 and 20. Note that in this test case, the resolution of the inviscid phenomena directly influences the near-wall flow properties resulting in discrepancies observed in Figs. 19 and 20. Auxiliary calculations indicate that some of the disagreement apparent in Fig. 20 is due to the difference in the viscous treatment.
Figure 17. Comparison of computed pressure contours
a) Beam-Warming scheme; b) present method
Figure 18. Comparison of computed Mach number contours
a) Beam-Warming scheme; b) present method
Figure 19. Comparison of wall pressure coefficients

Figure 20. Comparison of heat transfer coefficients
V. THREE-DIMENSIONAL NUMERICAL RESULTS

In order to evaluate the performance of the new algorithm in solving the three-dimensional PNS equations, flows past two simple test geometries were computed.

A. Test Case I. Circular Cone at Angle of Attack

The first flow-field computed with the new three-dimensional code was that of hypersonic laminar flow past a 10° half-angle circular cone. The flow conditions, chosen to correspond to those investigated experimentally by Tracy [24], were

\[ M_{\infty} = 7.95 \]
\[ \tilde{T}_{\infty} = 309.8 \text{ K} \]
\[ \tilde{T}_{cc} = 55.39 \text{ K} \]
\[ Pr = 0.72 \]
\[ Re_{\infty} = 4.101 \times 10^6 / m \quad \gamma = 1.4 \]

Calculations were performed for these freestream conditions applied to the 10° cone oriented at angles of attack of 12°, 20°, and 24°. The grids used in these calculations contained 60 points equally spaced circumferentially and 45 points exponentially stretched from the body to an elliptically shaped outer boundary. The grid used for the 20° case is shown in Fig. 21. Initial conditions for these cases were provided by specifying freestream conditions at the apex of the cone and marching initially with a very small step-size until the shock and viscous layers were well developed. The step-size was then gradually increased as the calculation proceeded.

Figures 22-24 show computed Mach number contours in comparison with flow-field geometries deduced by Tracy from pitot pressure surveys. Both computed and experimental results are plotted in spherical coordinates for \( \tilde{x} = 0.1016 \text{ m} \). At an angle of attack of 12° the viscous layer is shown to have separated from the lee side of the cone, leaving a small crossflow recirculation zone between the lee side
Figure 21. Computational grid, $\alpha = 20^\circ$
viscous layer and the lee side cone generator. The computed outer shock position is in close agreement with the experiment as is the location of the boundary-layer edge throughout the region of attached flow. When the flow separates, some disagreement is observed in the width of the shear layer, which may be attributable to the neglect of circumferential viscous terms in the calculation and to the relatively large mesh spacing in this region. When the angle of attack is increased to 20°, the recirculation region becomes more pronounced and, because of the supersonic crossflow, is accompanied by a crossflow shock wave. This is actually a fairly mild shock and, thus, appears in the computed results only as a deflection in the Mach contours. With further increase in the angle of attack, this crossflow shock increases in strength and develops into a lambda-shock pattern. This is reflected in Fig. 24 where, despite the relatively coarse mesh, the computed results for \( \alpha = 24° \) display a crossflow shock structure similar to that of the experiment.

Figures 25-27 show, for the three different angles of attack, comparisons of computed pressure coefficients with those measured by Tracy. Very good agreement is observed in all three cases although the computed pressures on the windward side are slightly lower than those measured experimentally. The same tendency has been observed in previous calculations of these cases \cite{25,26} and is generally attributed to experimental pressure taps that were large in relation to the windward boundary layer thickness.

B. Test Case II. Hypersonic All-Body Vehicle

The second test geometry consists of a simple all-body hypersonic aircraft configuration shown schematically in Fig. 28. These calculations are being performed in conjunction with an experimental investigation of this geometry currently in the
$M_\infty = 7.95, \quad Re_\infty = 4.101 \times 10^6/m, \quad \tilde{x} = 0.1016 \, m$

Figure 22. Comparison of flow-field geometries, $\alpha = 12^\circ$
$M_\infty = 7.95, \quad Re_\infty = 4.101 \times 10^6/m, \quad \tilde{x} = 0.1016 \, m$

Figure 23. Comparison of flow-field geometries, $\alpha = 20^\circ$
Figure 24. Comparison of flow-field geometries, $\alpha = 24^\circ$
$M_\infty = 7.95, \ Re_\infty = 4.101 \times 10^6/m, \ \widetilde{x} = 0.1016 \ m$

Figure 25. Comparison of pressure coefficients, $\alpha = 12^\circ$
$M_\infty = 7.95$, $Re_\infty = 4.101 \times 10^8/m$, $\tilde{x} = 0.1016$ m

Figure 26. Comparison of pressure coefficients, $\alpha = 20^\circ$
$M_\infty = 7.95$, $Re_\infty = 4.101 \times 10^6/m$, $\bar{x} = 0.1016 \text{ m}$

![Graph showing comparison of pressure coefficients, $\alpha = 24^\circ$](image-url)

**Figure 27.** Comparison of pressure coefficients, $\alpha = 24^\circ$
3.5-ft hypersonic wind tunnel at Ames Research Center. The freestream conditions used in the present calculations are those currently being used experimentally. They are

\[
\begin{align*}
M_\infty &= 7.4 & \bar{T}_w &= 300 \text{ K} \\
\bar{T}_\infty &= 176.3 \text{ K} & Pr &= 0.72 \\
Re_\infty &= 16.4 \times 10^6 / m & \gamma &= 1.4
\end{align*}
\]

Numerical results have obtained for angles of attack of 0° and 10°. Unfortunately, because the data are not yet available, experimental validation in this case is not possible at this time. At the present stage of development, the code does not include any turbulence modeling; thus, in spite of the high Reynolds number, all calculations of this case were performed assuming laminar flow.

The grid used for the zero-incidence case contained 60 cells circumferentially, clustered toward the tip according to curvature, and 43 cells stretched from the wall to an elliptic outer boundary. The crossflow grid at \( \bar{x} = 0.8 \text{ m} \) is shown in Fig. 29. For this case, the high Reynolds number made the generation of a starting solution by freestream start-up at the apex nearly impossible. Thus, an iterative step-back procedure was implemented which generates a conically similar starting solution at a prescribed \( \bar{x} \)-location. The results presented here were produced using initial conditions generated at \( \bar{x} = 0.1 \text{ m} \).

Computed Mach contours at the symmetry plane are shown in Fig. 30, where the region between the apex and the starting solution is included by way of the conical flow assumption. Noteworthy features here are the oblique shock wave, captured within two grid cells by the upwind algorithm, and the strong expansion fan that emanates from the "break point" of the body. Crossflow plane Mach
Figure 28. Schematic of the Ames All-Body vehicle
Figure 29. Computational grid, $\alpha = 0^\circ$, $x = 0.8m$. 
contours showing the cross-sectional shape of the bow shock and the break-point expansion at \( \tilde{x} = 0.8 \) m are shown in Fig. 31.

For the angle-of-attack case, because of the presence of larger circumferential gradients, the mesh employed with the new code consisted of 90 cells circumferentially and, again, 43 cells in the outward direction. A representative crossflow grid for this case is shown in Fig. 32. Initial conditions were again provided at \( \tilde{x} = 0.1 \) m using the conical step-back procedure. The generation of the starting solution for this case included an opportunity to see the effect of the entropy fix described in the previous chapter. Fig. 33a illustrates the nonconverged flow-field (represented by total pressure contours) in the absence of any special treatment of the eigenvalues, whereas Fig. 33b shows the same flow-field after 100 steps incorporating the augmented eigenvalues. The procedure has eliminated the nonphysical behavior near the "stagnation" region while leaving the remainder of the flow-field essentially unaltered.

The computed shock shape on the plane of symmetry for this case is indicated by the Mach contours of Fig. 34. A weak expansion fan does exist on the lee side of the vehicle but, due to the strength of the wind side shock wave, it doesn't appear in these contour plots. Figure 35 illustrates the cross section of the flow-field at \( \tilde{x} = 0.8 \) m. The oblique shock wave and the break-point expansion fan are clearly visible on the windward side, whereas the dominant feature on the lee side is the shock-induced crossflow separation bubble embedded within the viscous layer.

These calculations were stopped at the streamwise station, \( \tilde{x} = 0.85 \) m, because limitations of the present grid generator prevent a good resolution of extremely thin cross sections. The calculation of the 0° and 10° angle-of-attack cases required approximately 1 and 1.5 hr, respectively, of CPU time on the Cray
$M_\infty = 7.4$
$Re_\infty = 16.4 \times 10^6 / \text{m.}$
$\alpha = 0^\circ$

Figure 30. Symmetry plane Mach contours
$M_\infty = 7.4$
$Re_\infty = 16.4 \times 10^6$/m.
$\alpha = 0^\circ$

Figure 31. Crossflow plane Mach contours, $x = 0.8$m.
Figure 32. Computational grid, $\alpha = 10^\circ$, $x = 0.8m$. 
Figure 33. Effect of the entropy fix on stagnation pressure
a) without entropy fix; b) with entropy fix
\[M_\infty = 7.4\]
\[Re_\infty = 16.4 \times 10^6/m.\]
\[\alpha = 10^\circ\]

Figure 34. Symmetry plane Mach contours, \(\alpha = 10^\circ\)
$M_\infty = 7.4$
$Re_\infty = 16.4 \times 10^6/m.$
$\alpha = 10^\circ$

Figure 35. Crossflow plane Mach contours, $x = 0.8m.$
X-MP computer to march from the initial conditions to 0.85 m. The generation of the starting solutions required about 40 and 60 min, respectively, beginning with free-stream conditions. Efforts toward increasing the vectorization and improving the overall efficiency of the code are scheduled for the near future.
VI. CONCLUDING REMARKS

A new algorithm for the solution of the three-dimensional parabolized Navier-Stokes equations has been developed. It incorporates upwind dissipation terms in the crossflow directions to facilitate the capture of strong shock waves without user-specified smoothing coefficients. The algorithm, based on Roe's approximate Riemann solver, is implicit and second-order accurate in the crossflow directions. In addition the finite-volume approach has been adopted in the development of the new algorithm and, as a result, the method is flux conservative in nonlinear flow regions as well as in the freestream.

The method has been validated in both two and three dimensions. Validation in two dimensions consisted of the computation of supersonic laminar flow past a flat plate, hypersonic laminar flow past a 15° compression corner, and hypersonic laminar flow into a converging inlet. The accuracy of the new scheme was then evaluated by comparing results with experimental data as well as with results obtained using established numerical techniques. The comparison of computed wall coefficient quantities between results of the present code and results of conventional methods shows close agreement; however, as expected, the new algorithm appears to be clearly superior in the resolution of the flow-field. Both numerical approaches compare well with experimental data, although a slight over-prediction of wall coefficient quantities, possibly due to inadequate grid spacing, was observed.

The new algorithm was also applied to the three-dimensional hypersonic flow past two simple test geometries including a 10° half-angle circular cone and an
elliptic cone-based hypersonic vehicle. Results show good qualitative and quantitative agreement with experimental data for the cone flow cases. Although experimental data for the generic hypersonic vehicle is not yet available, the computed flow fields for this geometry indicate the presence of expected physical features, and these are shown to be well resolved by the new algorithm. As expected, the new code is more expensive per grid point than the conventional method; however, the increased accuracy and reliability of the new algorithm in shock capturing situations is expected to offset this disadvantage. Plans for the immediate future include the incorporation into the code of a simple turbulence model, as well as equilibrium-air real gas effects.
VII. REFERENCES


VIII. ACKNOWLEDGEMENTS

The author would like to express his most sincere appreciation to Professor John Tannehill who suggested the problem and who provided essential guidance and motivation throughout the completion of this work.

The author also wishes to acknowledge financial support from the Applied Computational Fluids branch of NASA Ames Research center under the leadership of former branch chief Dr. Paul Kutler, and present branch chief Dr. Terry Holst. The members of the ACF branch have also been generous in providing the benefits of their technical expertise. In particular, the author is grateful to Dr. Denny Chaussee and to Dr. Man Mohan Rai for the patient sharing of their insight and experience and for their continued encouragement and humor.

In addition to support from NASA Ames Research Center under Grant NCA2-IR340-501, this work was partially supported by the Computational Fluid Dynamics Center, Iowa State University, Ames, Iowa.
IX. APPENDIX A: NAVIER-STOKES FLUX VECTORS
The inviscid flux vectors of the Navier-Stokes equations, Eq. 1, are given by

\[ E_i = [\rho u, \rho u^2 + p, \rho uv, \rho uw, (E_t + p)u]^T \]
\[ F_i = [\rho v, \rho uv, \rho v^2 + p, \rho vw, (E_t + p)v]^T \]
\[ G_i = [\rho w, \rho uw, \rho vw, \rho w^2 + p, (E_t + p)w]^T \]

where the total energy is

\[ E_t = \rho e + \frac{1}{2}(u^2 + v^2 + w^2) \]

The viscous flux vectors are of the form

\[ E_v = [0, \tau_{zx}, \tau_{zy}, \tau_{zz}, u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z]^T \]
\[ F_v = [0, \tau_{zy}, \tau_{yy}, \tau_{yz}, u\tau_{zy} + v\tau_{yy} + w\tau_{yz} - q_y]^T \]
\[ G_v = [0, \tau_{zz}, \tau_{yz}, \tau_{zz}, u\tau_{zz} + v\tau_{yz} + w\tau_{zz} - q_z]^T \]

where the viscous stress and heat transfer terms are given by

\[
\tau_{zx} = \frac{2}{3} \frac{\mu}{Re_L} \left( \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z} \right) \\
\tau_{zy} = \frac{2}{3} \frac{\mu}{Re_L} \left( \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z} \right) \\
\tau_{zz} = \frac{2}{3} \frac{\mu}{Re_L} \left( \frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) \\
\tau_{xy} = \frac{\mu}{Re_L} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\
\tau_{yz} = \frac{\mu}{Re_L} \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \\
\tau_{xz} = \frac{\mu}{Re_L} \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\
q_x = \frac{\mu}{(\gamma - 1)M_{\infty}^8Re_LPr} \frac{\partial T}{\partial x} \\
q_y = \frac{\mu}{(\gamma - 1)M_{\infty}^8Re_LPr} \frac{\partial T}{\partial y} \\
q_z = \frac{\mu}{(\gamma - 1)M_{\infty}^8Re_LPr} \frac{\partial T}{\partial z}
\]
APPENDIX B: NAVIER-STOKES FLUX-JACOBIANS
The inviscid flux-Jacobians required for the left-hand-side of the implicit algorithm are given below:

\[
\lambda^s(dS^{m1}, U^{m2}) = \begin{bmatrix}
0 & \xi_{f} & \xi_{\phi} & \xi_{u} & \xi_{w} & 0 \\
\omega \xi_{f} - u \tilde{U} & - \xi_{f} [\omega (\gamma - 1) - 1]u & - \xi_{f} [\omega (\gamma - 1) v] & - \xi_{f} [\omega (\gamma - 1) w] & \xi_{f} [\omega (\gamma - 1)] \\
\omega \xi_{f} - w \tilde{U} & \xi_{f} [\omega (\gamma - 1)]u & - \xi_{f} [\omega (\gamma - 1) - 1]v & - \xi_{f} [\omega (\gamma - 1) w] & \xi_{f} [\omega (\gamma - 1)] \\
(\theta^2 - h^2) \tilde{U} & - \xi_{f} h_t - \tilde{U} (\gamma - 1)u & - \xi_{f} h_t - \tilde{U} (\gamma - 1)v & - \xi_{f} h_t - \tilde{U} (\gamma - 1)w & \gamma \tilde{U}
\end{bmatrix}
\]

\[
\beta_t (dS_{m1}, U_{m2}) = \begin{bmatrix}
0 & \xi_{f} & \xi_{\phi} & \xi_{u} & \xi_{w} & 0 \\
\xi_{f} \phi^2 - u \theta & \theta - \xi_{f} (\gamma - 2)u & - \xi_{f} (\gamma - 1) v & - \xi_{f} (\gamma - 1) w & \xi_{f} (\gamma - 1) \\
\xi_{f} \phi^2 - w \theta & \xi_{f} [\omega (\gamma - 1)]u & \theta - \xi_{f} (\gamma - 2) v & - \xi_{f} (\gamma - 1) w & \xi_{f} (\gamma - 1) \\
(\phi^2 - h_t) \theta & - \xi_{f} h_t - \theta (\gamma - 1)u & - \xi_{f} h_t - \theta (\gamma - 1)v & - \xi_{f} h_t - \theta (\gamma - 1)w & \gamma \theta
\end{bmatrix}
\]

where the metrics of these matrices are evaluated at the m1 cell face and the physical properties are evaluated at the m2 secondary grid point. The contravariant velocities are defined by

\[
\tilde{U} = \frac{\xi_{x} u}{f} + \frac{\xi_{y} v}{f} + \frac{\xi_{z} w}{f}
\]

\[
\theta = \frac{\kappa_{x} u}{f} + \frac{\kappa_{y} v}{f} + \frac{\kappa_{z} w}{f}
\]
and

\[ \phi^2 = \frac{1}{2}(\gamma - 1)(u^2 + v^2 + w^2) \]

Using the above matrix \( \hat{B}_i \), the Jacobians of the first-order upwind numerical flux are approximated as follows

\[ \frac{\partial \mathbf{H}^{i+\frac{1}{2}}}{\partial \mathbf{U}_m} = -\left(1 + \text{sgn}D\right)B_i(dS_m^+\frac{1}{2}, \mathbf{U}_m) \]

\[ \frac{\partial \mathbf{H}^{i+\frac{1}{2}}}{\partial \mathbf{U}_{m+1}} = -\left(1 - \text{sgn}D\right)B_i(dS_{m+\frac{1}{2}}, \mathbf{U}_{m+1}) \]

Neglecting cross-derivative terms, the viscous flux vectors in the \( \eta \) and \( \zeta \) directions are given by

\[
\mathbf{\tilde{P}}_\eta = \frac{\mu J}{\text{Re}} \left( \begin{array}{c}
0 \\
11u_\eta + 15v_\eta + 15w_\eta \\
15u_\eta + 23v_\eta + 25w_\eta \\
15u_\eta + 25v_\eta + 35w_\eta \\
\frac{1}{2}u^2_\eta + 15w_\eta + f_{35}(\frac{1}{2}u^2)\eta + f_{41}T_\eta \\
\end{array} \right)
\]

\[
\mathbf{\tilde{G}}_\eta = \frac{\mu J}{\text{Re}} \left( \begin{array}{c}
0 \\
g_{11}u_\eta + g_{13}v_\eta + g_{15}w_\eta \\
g_{13}u_\eta + g_{23}v_\eta + g_{25}w_\eta \\
g_{15}u_\eta + g_{25}v_\eta + g_{35}w_\eta \\
g_{11}(\frac{1}{2}u^2)\eta + g_{13}(uv)\eta + g_{15}(uw)\eta + g_{23}(\frac{1}{2}v^2)\eta + g_{25}(uw)\eta + g_{35}(\frac{1}{2}w^2)\eta + g_{41}T_\eta \\
\end{array} \right)
\]

respectively, where

\[
f_{00} = \left( \frac{\eta \xi}{J} \right)^2 + \left( \frac{\eta \eta}{J} \right)^2 + \left( \frac{\eta \xi}{J} \right)^2
\]

\[
f_{11} = f_{11} + \frac{1}{3} \left( \frac{\eta \xi}{J} \right)^2
\]

\[
f_{13} = \frac{1}{3} \left( \frac{\eta \xi}{J} \right) \left( \frac{\eta \eta}{J} \right)
\]

\[
f_{15} = \frac{1}{3} \left( \frac{\eta \xi}{J} \right) \left( \frac{\eta \xi}{J} \right)
\]

\[
f_{23} = f_{00} + \frac{1}{3} \left( \frac{\eta \eta}{J} \right)^2
\]

\[
f_{25} = \frac{1}{3} \left( \frac{\eta \eta}{J} \right) \left( \frac{\eta \xi}{J} \right)
\]

\[
f_{35} = f_{00} + \frac{1}{3} \left( \frac{\eta \eta}{J} \right)^2
\]

\[
f_{41} = \frac{1}{(\gamma - 1)M_{\infty}^2 Pr} f_{00}
\]
The viscous flux-Jacobians are given by defining a generic viscous flux-Jacobian as follows:

\[
\hat{\mathbf{b}}_v(\mathbf{dS}_{m1}, \mathbf{U}_{m2}) = \frac{\mu_{1 \text{Re}}}{\rho} \frac{\mathbf{1}_3}{\rho} \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & h_{11} \left( -\frac{u}{\rho} \right) + h_{12} \left( -\frac{v}{\rho} \right) + h_{15} \left( \frac{w}{\rho} \right) & h_{11} \left( \frac{1}{\rho} \right) & h_{13} \left( \frac{1}{\rho} \right) & h_{15} \left( \frac{1}{\rho} \right) & 0 \\
0 & h_{12} \left( -\frac{u}{\rho} \right) + h_{23} \left( -\frac{v}{\rho} \right) + h_{25} \left( \frac{w}{\rho} \right) & h_{13} \left( \frac{1}{\rho} \right) & h_{23} \left( \frac{1}{\rho} \right) & h_{25} \left( \frac{1}{\rho} \right) & 0 \\
0 & h_{15} \left( -\frac{u}{\rho} \right) + h_{25} \left( -\frac{v}{\rho} \right) + h_{35} \left( \frac{w}{\rho} \right) & h_{15} \left( \frac{1}{\rho} \right) & h_{25} \left( \frac{1}{\rho} \right) & h_{35} \left( \frac{1}{\rho} \right) & 0 \\
0 & h_{11} \left( -\frac{u^2}{\rho} \right) + h_{13} \left( -\frac{2uv}{\rho} \right) & h_{11} \left( \frac{u}{\rho} \right) + h_{13} \left( \frac{u}{\rho} \right) + h_{15} \left( -\frac{u}{\rho} \right) + h_{15} \left( \frac{u}{\rho} \right) & 0 \\
0 & h_{13} \left( -\frac{2uw}{\rho} \right) + h_{23} \left( -\frac{v^2}{\rho} \right) & h_{13} \left( \frac{v}{\rho} \right) + h_{23} \left( \frac{v}{\rho} \right) + h_{25} \left( -\frac{v}{\rho} \right) + h_{25} \left( \frac{v}{\rho} \right) & 0 \\
0 & h_{25} \left( -\frac{2uw}{\rho} \right) + h_{35} \left( -\frac{w^2}{\rho} \right) & h_{25} \left( \frac{w}{\rho} \right) + h_{35} \left( \frac{w}{\rho} \right) + h_{35} \left( -\frac{w}{\rho} \right) + h_{35} \left( \frac{w}{\rho} \right) & 0 \\
M_{\infty}^2 h_{41} \left( \frac{\gamma \Phi^2 - c^2}{\rho} \right) & M_{\infty}^2 h_{41} \left( \frac{-\Phi}{\rho} \right) & M_{\infty}^2 h_{41} \left( \frac{-\Phi}{\rho} \right) & M_{\infty}^2 h_{41} \left( \frac{-\Phi}{\rho} \right) & M_{\infty}^2 h_{41} \left( \frac{-\Phi}{\rho} \right)
\end{bmatrix}
\]
where, again, the geometry \((h \text{ terms})\) is evaluated at \(m1\) and the flow properties are given at \(m2\). For the fluxes in the \(\eta\)-direction, \(h_{ij} = f_{ij}\) and

\[
\frac{\partial (F_v)_{k+\frac{1}{2},l}}{\partial U_{k+1,l}} = \tilde{B}_v(dS_{k+\frac{1}{2},l, \ U_{k+1,l}})
\]

\[
\frac{\partial (F_v)_{k-\frac{1}{2},l}}{\partial U_{k,l}} = -\tilde{B}_v(dS_{k-\frac{1}{2},l, \ U_{k,l}})
\]

For the \(\zeta\)-direction, \(h_{ij} = g_{ij}\) and

\[
\frac{\partial (G_v)_{k,l+\frac{1}{2}}}{\partial U_{k,l+1}} = B_v(dS_{k,l+\frac{1}{2}, \ U_{k,l+1}})
\]

\[
\frac{\partial (G_v)_{k,l-\frac{1}{2}}}{\partial U_{k,l}} = -B_v(dS_{k,l-\frac{1}{2}, \ U_{k,l}})
\]
XI. APPENDIX C: DECOMPOSITION OF THE $D$ MATRIX
In the solution of the steady approximate Riemann problem, Eq. 13, knowledge of the eigenvalues and eigenvectors of the coefficient matrix $D_{m+\frac{1}{2}}$ is required. In this work, these quantities were derived with respect to a Cartesian coordinate system and then rotated into the required local orientation. The rotation is defined by the transformation matrix

$$T = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & \xi_x' & \xi_y' & \xi_z' & 0 \\
0 & \eta_x' & \eta_y' & \eta_z' & 0 \\
0 & \zeta_x' & \zeta_y' & \zeta_z' & 0 \\
0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

This matrix, when multiplied into the governing equations, leaves the continuity equation and energy equation unchanged. The momentum equations, however, are altered such that conservation of momentum in the rotated coordinate directions is enforced. If the transformation is orthogonal and the $\xi'$ axis is oriented such that

$$\xi_x' = l'(\frac{\xi_x}{J}), \quad \xi_y' = l'(\frac{\xi_y}{J}), \quad \xi_z' = l'(\frac{\xi_z}{J})$$

where

$$l' = \frac{1}{\sqrt{\left(\frac{\xi_x}{J}\right)^2 + \left(\frac{\xi_y}{J}\right)^2 + \left(\frac{\xi_z}{J}\right)^2}}$$

then the dependent vector of this system of equations will be of Cartesian form with the Cartesian velocity components replaced by the velocity components in the rotated coordinate directions. That is,

$$TE_i = \frac{1}{l'}E'$$
where
\[ \mathbf{E}' = [\rho u', \rho(u')^2 + p, \rho u'v', \rho u'w', (E_t + p)u']^T \]

and
\[
\begin{pmatrix}
  u' \\
  v' \\
  w'
\end{pmatrix} =
\begin{pmatrix}
  \xi_x' & \xi_y' & \xi_z' \\
  \eta_x' & \eta_y' & \eta_z' \\
  \zeta_x' & \zeta_y' & \zeta_z'
\end{pmatrix}
\begin{pmatrix}
  u \\
  v \\
  w
\end{pmatrix}
\]

The factor \( l' \) is the inverse of the constant-\( \xi \) cell-face area and therefore is needed to scale between the flux, \( \mathbf{E}_i \), and the vector \( \mathbf{E}' \), which is a flux per unit area.

Multiplication of \( \mathbf{F}_i \) or \( \mathbf{G}_i \) by the transformation matrix yields a vector of form
\[
T(\mathbf{F}_i, \mathbf{G}_i) = [\rho \theta, \rho u' \theta + \kappa_{\xi'} p, \rho v' \theta + \kappa_{\eta'} p, \rho w' \theta + \kappa_{\zeta'} p, (E_t + p) \theta]^T
\]

where
\[
\begin{align*}
\kappa_{\xi'} &= \xi_x' \left( \frac{\kappa_x}{J} \right) + \xi_y' \left( \frac{\kappa_y}{J} \right) + \xi_z' \left( \frac{\kappa_z}{J} \right) \\
\kappa_{\eta'} &= \eta_x' \left( \frac{\kappa_x}{J} \right) + \eta_y' \left( \frac{\kappa_y}{J} \right) + \eta_z' \left( \frac{\kappa_z}{J} \right) \\
\kappa_{\zeta'} &= \zeta_x' \left( \frac{\kappa_x}{J} \right) + \zeta_y' \left( \frac{\kappa_y}{J} \right) + \zeta_z' \left( \frac{\kappa_z}{J} \right)
\end{align*}
\]

This notation is chosen for the left-hand-side terms because, due to the orthogonal nature of the the transformation from \((x, y, z)\) to \((\xi', \eta', \zeta')\), they satisfy the relations
\[
\begin{align*}
\left( \frac{\kappa_x}{J} \right) &= \kappa_{\xi'} \xi_x' + \kappa_{\eta'} \eta_x' + \kappa_{\zeta'} \zeta_x' \\
\left( \frac{\kappa_y}{J} \right) &= \kappa_{\xi'} \xi_y' + \kappa_{\eta'} \eta_y' + \kappa_{\zeta'} \zeta_y' \\
\left( \frac{\kappa_z}{J} \right) &= \kappa_{\xi'} \xi_z' + \kappa_{\eta'} \eta_z' + \kappa_{\zeta'} \zeta_z'
\end{align*}
\]
Substitution of these relations into the definition of $\theta$ yields

$$\theta = \kappa_{\zeta'} u' + \kappa_{\eta'} v' + \kappa_{\zeta'} w'$$

and, as a result,

$$T(\bar{F}_i, \bar{G}_i) = \kappa_{\zeta'} F' + \kappa_{\eta'} F' + \kappa_{\zeta'} G'$$

where

$$F' = [\rho u', \rho u' v', \rho (v')^2 + p, \rho u' w', (E_t + p) v']^T$$

$$G' = [\rho w', \rho u' w', \rho v' w', \rho (w')^2 + p, (E_t + p) w']^T$$

Differentiating this vector with respect to $T\bar{E}_i$ yields the flux-Jacobian matrix $D'$ which is of form

$$D' = l' \left( \kappa_{\zeta'} I + \kappa_{\eta'} \frac{\partial F'}{\partial E'} + \kappa_{\zeta'} \frac{\partial G'}{\partial E'} \right)$$

This matrix is a similarity transformation of $D$ as shown below:

$$D' = R' \Lambda' R'^{-1} = T R A R^{-1} T^{-1} = T D T^{-1}$$

so that eigenvalues of $D'$ are eigenvalues of $D$ and the right eigenvector matrices are related by $R' = TR$.

For reasons which should become clear in the following discussion, the $\xi'$-coordinate direction of the rotated system is chosen such that the direction cosines are given by

$$\xi' = m^l \xi''$$

$$\eta' = m^l \eta''$$

$$\zeta' = m^l \zeta''$$
where
\[ \xi''_x = \xi_x \kappa_y - \xi_y \kappa_x \quad \xi''_y = \xi_x \kappa_z - \xi_z \kappa_x \quad \xi''_z = \xi_y \kappa_z - \xi_z \kappa_y \]
and
\[ m' = \frac{1}{\sqrt{(\xi''_x)^2 + (\xi''_y)^2 + (\xi''_z)^2}} \]
The final coordinate axis is defined to be orthogonal to the others according to the right hand rule:
\[ \eta'_x = \xi'_x \xi'_y - \xi'_y \xi'_x \quad \eta'_y = \xi'_x \xi'_z - \xi'_z \xi'_x \quad \eta'_z = \xi'_y \xi'_z - \xi'_z \xi'_y \]
For the above choice of \( \xi' \), \( \kappa' \) vanishes and \( D' \) can be written as
\[ D' = R_c(\kappa' \xi' I + \eta' \kappa' \Lambda_c) R^{-1} \]
where \( R_c \) is the matrix containing the right eigenvectors of the flux-Jacobian \( \frac{\partial \mathbf{F}'}{\partial \mathbf{E}'} \) and \( \Lambda_c \) contains its eigenvalues.
The forms of the vectors \( \mathbf{E}' \) and \( \mathbf{F}' \) indicate that the eigenvectors will be of Cartesian form which considerably simplifies their derivation. Thus, the eigenvectors of the matrix \( D \) are obtained by
\[ R = T^{-1} R_c \quad \text{and} \quad R^{-1} = R_c^{-1} T \]
and the eigenvalues are computed from
\[ \lambda^i = \lambda'(\kappa_{\xi', i} + \kappa_{\eta', i} \lambda^i_c) \]
The Cartesian eigenvalues, with the Cartesian velocities replaced with the velocity components in the rotated coordinate directions, are

\[
\lambda_{1,2,3}^c = \frac{u'}{u'}
\]

\[
\lambda_4^c = \frac{u'v' + c\sqrt{d}}{(u')^2 - c^2}
\]

\[
\lambda_5^c = \frac{u'v' - c\sqrt{d}}{(u')^2 - c^2}
\]

where \(c\) represents the speed of sound and

\[
\sqrt{d} = \sqrt{(u')^2 + (v')^2 - c^2}
\]

The matrix of right eigenvectors, \(R\), is shown below:

\[
R = \begin{bmatrix}
\frac{1}{c^2} & 0 & -\frac{1}{(u')^2 + (v')^2} & -\frac{2c^2\mu^4}{2c^2\mu^5} \\
\cdot & \cdot & \cdot & \cdot \\
u & \frac{\zeta_x}{c^2} & -2u + \zeta_x w' & -u + c\omega_x/\sqrt{d} \\
\cdot & \cdot & \cdot & \cdot \\
v & \frac{\zeta_y}{c^2} & -2v + \zeta_y w' & -v + c\omega_y/\sqrt{d} \\
\cdot & \cdot & \cdot & \cdot \\
w & \frac{\zeta_z}{c^2} & -2w + \zeta_z w' & -w + c\omega_z/\sqrt{d} \\
\cdot & \cdot & \cdot & \cdot \\
\phi^2 & \frac{w'}{(u')^2 + (v')^2} & -\frac{(u')^2 + (v')^2 - h_t}{(u')^2 + (v')^2} & -h_t \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\frac{1}{(\gamma - 1)c^2} & \frac{u'}{(u')^2 + (v')^2} & \frac{2c^2\mu^4}{2c^2\mu^5} & -h_t \end{bmatrix}
\]
where

\[ \mu_c^4 = v' - u' \lambda_c^4 \]
\[ \mu_c^5 = v' - u' \lambda_c^5 \]
\[ \omega_x = \xi_x' v - \xi_y' w \]
\[ \omega_y = \xi_z' w - \xi_x' u \]
\[ \omega_z = \xi_y' u - \xi_z' v \]

The matrix of left eigenvectors, \( R^{-1} \), is

\[
R^{-1} = \begin{bmatrix}
\xi^2 - \phi^2 & \cdots & (\gamma - 1)u & \cdots & (\gamma - 1)v & \cdots & (\gamma - 1)w & \cdots & -(\gamma - 1)
\end{bmatrix}
\]

These eigenvectors can be shown to be independent for any transformation provided that the velocity in the \( \xi' \)-direction is nonzero.