Efficient numerical approaches for predicting supersonic flows with viscous-inviscid interaction

David Seitz Thompson

Iowa State University

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

Part of the Aerospace Engineering Commons

Recommended Citation

Thompson, David Seitz, "Efficient numerical approaches for predicting supersonic flows with viscous-inviscid interaction " (1987). Retrospective Theses and Dissertations. 11651.
https://lib.dr.iastate.edu/rtd/11651

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.
Thompson, David Seitz

EFFICIENT NUMERICAL APPROACHES FOR PREDICTING SUPersonic
FLOWS WITH VIScOUS/INVISCID INTERACTION

Iowa State University

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
Efficient numerical approaches for predicting supersonic flows with viscous/inviscid interaction

by

David Seitz Thompson

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

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>GOVERNING EQUATIONS</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Navier-Stokes Equations</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Transformation to Computational Space</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Thin-Layer Navier-Stokes Equations</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Preliminary Algorithm Considerations</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>SINGLE-PASS PHILOSOPHY</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Parabolized Navier-Stokes Equations</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Equivalent Flux Split Formulations</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>Minimum Marching Step Size</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>Numerical Algorithm</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td>Grid Generation</td>
<td>62</td>
</tr>
<tr>
<td></td>
<td>Results</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>MULTIPLE-PASS PHILOSOPHY</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>Global Pressure Iteration</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>Global Pressure Iteration Results</td>
<td>106</td>
</tr>
<tr>
<td></td>
<td>Treatment of Separated Flows</td>
<td>124</td>
</tr>
<tr>
<td></td>
<td>Separated Flow Results</td>
<td>137</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
<td>158</td>
</tr>
<tr>
<td>5</td>
<td>CONCLUDING REMARKS</td>
<td>161</td>
</tr>
</tbody>
</table>

BIBLIOGRAPHY ........................................ 164

ACKNOWLEDGEMENTS ................................ 167

APPENDIX A. JACOBIAN MATRICES ............... 168
CHAPTER 1. INTRODUCTION

Current interest in the transatmospheric flight vehicle has been the genesis of renewed research in the supersonic and hypersonic flight regimes. Due to the increased expense and difficulty associated with wind tunnel testing at the required conditions, it is clear that a major portion of the configuration design process must be performed using computational fluid dynamics (CFD) codes. As a result, a major thrust of research in the field of aerodynamics has been directed at improving the efficiency and reliability of CFD algorithms. Accurately predicting the complex phenomena that occur in a steady flow field surrounding a vehicle of this type requires solving the nonlinear compressible Navier-Stokes equations. Typically, the unsteady form of the equations of motion is solved by marching in time until a steady-state solution is obtained. In spite of improvements in algorithm efficiency and computer capabilities that have made the time-marching approach more tractable, considerable computational effort is required for computing even the simplest cases. The objective of this study is to identify and explore efficient and reliable numerical approaches that can be employed to predict complex supersonic/hypersonic viscous flows.

Rather than attempting to solve the complete three-dimensional problem, an algorithm is developed to predict the flow around a simpler two-dimensional configuration. The
specific geometry considered in this study is a flat plate followed by a ramp. This configuration, generally known as a compression corner, is illustrated schematically in Figure 1. The flow is turned by the presence of the ramp and the corresponding pressure rise is propagated upstream in the subsonic portion of the boundary layer. This upstream influence induces a strong interaction between the boundary layer and the outer inviscid flow. The overall effect of this interaction is the thickening of the boundary layer upstream of the corner so that the flow is gradually turned by a series of compression waves. For certain combinations of Mach number, Reynolds number, and ramp angle, the boundary layer has insufficient momentum to overcome the adverse pressure gradient and separation occurs. The separated boundary layer becomes a free shear layer external to a recirculating flow adjacent to the wall. If the separated region is large enough in extent, a region of relatively constant pressure, a pressure plateau, forms just downstream of the separation point. The pressure plateau is terminated when the shear layer impinges on the ramp and the flow reattaches. At the reattachment point, another compression occurs as the flow is turned once again. Downstream of the reattachment point, the boundary layer returns to a state of weak interaction with the outer inviscid flow. It is important to note that in the above discussion, the effects of the leading edge have been
Figure 1. Schematic diagram of compression corner flow field
ignored. The emphasis of this study is on predicting the effects associated with the strong viscous/inviscid interaction and the separated flow.

The remainder of this dissertation is divided into four chapters. Chapter 2 introduces the equations that govern the motion of a compressible viscous fluid, the Navier-Stokes (NS) equations. The unsteady form of the two-dimensional NS equations, as well as the additional equations needed for closure, are presented. A nonsingular transformation is used to map the simply-connected physical domain into a rectangular domain in computational space where the chain rule form of the governing equations is used. The thin-layer concept is discussed with respect to a heuristic argument and the thin-layer NS equations are given in computational space. Finally, some preliminary numerical considerations based on the characteristics of the governing equations are discussed.

Chapter 3 is primarily a discussion of the single-pass philosophy for solving the parabolized Navier-Stokes (PNS) equations. Relevant background material is cited that shows the stability, or lack thereof, of single-pass algorithms for the PNS equations is dependent upon the treatment of the streamwise pressure gradient, or equivalently, the streamwise convective flux vector. The different methods presented for suppressing the so-called departure solutions are interpreted in a flux splitting context and the cause of the undesirable
behavior is identified. A model problem is studied that shows that departure solutions are, in fact, the response of the numerical algorithm to an ill-posed problem. In addition, an alternative interpretation of an apparent minimum marching step size constraint is presented that indicates the reliability of such estimates is questionable. A consistent implicit first- or second-order single-pass algorithm is developed for the PNS equations including discussions about the numerical boundary condition procedure and the forms of artificial smoothing employed. By comparison with the predictions of other numerical methods, the PNS algorithm is shown to be inadequate for predicting the details of the compression corner flow field at moderate supersonic speeds. In fact, the results indicate that the single-pass philosophy is deficient since significant elliptic effects that are present in the flow field cannot be predicted. Chapter 3 concludes with a summary of these results.

Chapter 4 introduces the concept of a pseudo-time iteration for predicting compression corner flow field. Two different multiple-pass algorithms are developed for solving the TLNS equations for flows with significant streamwise elliptic behavior. A generalized differencing scheme for the streamwise pressure gradient is developed and used to show that two formally different GPI schemes are equivalent under certain conditions. The equivalence of these two algorithms
is employed along with a heuristic stability analysis of a model problem to explain the behavior of standard second-order GPI algorithms. Based on the heuristic stability analysis and examination of the modified equation, a conditionally stable well-behaved second-order GPI algorithm is developed. The GPI algorithm is discussed in terms of upwind relaxation schemes that appear in the literature. For flows with streamwise separation, the FLARE approximation is employed. Results predicted using the GPI algorithm show good agreement with results presented in the literature although the GPI results typically show less viscous/inviscid interaction. It was determined that the convergence rate of the GPI scheme was significantly improved through the addition of an unsteady term that enhanced the diagonal dominance of the algorithm. This algorithm proved to be unstable for flows with larger recirculating regions, apparently due to the manner in which the FLARE approximation was implemented.

In addition to the GPI algorithm, a second-order algorithm is developed using a hybrid approach. The standard conservative PNS algorithm is employed in the outer supersonic flow while in the subsonic region near the wall, a nonconservative SCM scheme is used to determine the differencing for the streamwise convective flux vector. Since there are no embedded shocks in the subsonic region, the SCM formulation provides an excellent approximation to the
streamwise convection of information. The second-order streamwise differencing scheme developed for the GPI algorithm is used in the SCM scheme while standard central differencing is employed for the transverse derivatives. Results predicted using the hybrid algorithm are very similar to results predicted using the GPI scheme with one exception: the hybrid algorithm shows more of a decrease in wall shear at the corner than the GPI results. The hybrid algorithm requires more iterations and more computer time per iteration than the GPI algorithm. However, the hybrid approach is capable of predicting flows with extensive streamwise separation.

Chapter 5 is a general summary of the entire work. This chapter is included primarily to restate the important results presented earlier in the text and suggest areas for future research.
CHAPTER 2. GOVERNING EQUATIONS

Navier-Stokes Equations

The equations governing the motion of a compressible viscous fluid are actually statements of three fundamental conservation laws: conservation of mass, conservation of momentum, and conservation of energy. For continuum flow of a Newtonian fluid, the Navier-Stokes (NS) equations are the appropriate forms of these conservation laws. Under the assumptions of zero body force and no external heat sources, the two-dimensional NS equations in nondimensional form are given in Cartesian (x,y) space by

\[
\frac{\partial U}{\partial t} + \frac{\partial E_i}{\partial x} + \frac{\partial F_i}{\partial y} = \frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y} \tag{1}
\]

with

\[
U = [\rho, \rho u, \rho v, E_t]^T
\]

\[
E_i = [\rho u, \rho u^2 + p, \rho uv, (E_t + p)u]^T
\]

\[
F_i = [\rho v, \rho uv, \rho v^2 + p, (E_t + p)v]^T
\]

\[
E_v = [0, \tau_{xx}, \tau_{xy}, u\tau_{xx} + v\tau_{xy} - q_x]^T
\]

\[
F_v = [0, \tau_{xy}, \tau_{yy}, u\tau_{xy} + v\tau_{yy} - q_y]^T
\]

where \( \rho \) is the density, \( u \) is the velocity component in the x
direction, \( v \) is the velocity component in the \( y \) direction, \( p \) is the pressure, \( \tau_{xx} \) and \( \tau_{yy} \) are the viscous normal stresses, \( \tau_{xy} \) is the viscous shear stress, and \( q_x \) and \( q_y \) are the rates of heat transfer due to conduction. In addition, \( E_t \) is the stagnation energy per unit volume and is given by

\[
E_t = \rho (e + 0.5(u^2 + v^2))
\]

where \( e \) is the internal energy per unit mass. Since no real gas effects are considered, the equations of state for a perfect gas are used and are given in nondimensional form by

\[
p = (\gamma - 1)\rho e
\]

\[
T = \gamma M_\infty^2 p/\rho
\]

where \( \gamma \) is the ratio of specific heats, taken to be 1.4, \( T \) is the nondimensional absolute temperature, and \( M_\infty \) is the freestream Mach number. The nondimensional viscous stress terms are given by

\[
\tau_{xx} = \frac{2}{3 Re_\infty} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)
\]

\[
\tau_{yy} = \frac{2}{3 Re_\infty} \left( \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} \right)
\]

\[
\tau_{xy} = \frac{\mu}{Re_\infty} \left( \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right)
\]
where $\mu$ is the coefficient of viscosity, $Re_\infty$ is the freestream Reynolds number, and Stokes' hypothesis has been utilized to eliminate the second coefficient of viscosity. Due to difficulties associated with numerically solving the NS equations, additional assumptions are required if turbulent flow is to be computed. The relative coarseness of the discrete mesh system employed in the finite-difference solution does not adequately resolve the turbulent mixing process. To avoid the inherent difficulties of turbulence modelling for separated flows, only laminar flow is considered in this study. The laminar viscosity coefficient is computed using Sutherland's equation

$$\mu = T^{1.5} \left( \frac{1 + K}{T + K} \right) , \quad K = \frac{110.4}{T_\infty}$$

where $T_\infty$ is the dimensional freestream temperature in degrees Kelvin. Finally, the gas is assumed to obey a Fourier conduction law so that the heat flux terms are given by

$$q_x = -\frac{\mu}{Re_\infty (\gamma - 1)M_\infty^2Pr} \frac{\partial T}{\partial x}$$

$$q_y = -\frac{\mu}{Re_\infty (\gamma - 1)M_\infty^2Pr} \frac{\partial T}{\partial y}$$

where the Prandtl number, $Pr$, is assumed to have a constant value of 0.72 to eliminate the thermal conductivity.
The above equations have been nondimensionalized in the following manner:

\[ t = \frac{\xi \nu}{L} , \quad x = \frac{x}{L} , \quad y = \frac{y}{L} \]

\[ u = \frac{\tilde{u}}{\nu} , \quad v = \frac{\tilde{v}}{\nu} , \quad e = \frac{\tilde{e}}{\nu^2} \]

\[ \rho = \frac{\tilde{\rho}}{\rho_\infty} , \quad T = \frac{T}{T_\infty} , \quad \mu = \frac{\tilde{\mu}}{\nu} \]

\[ p = \frac{\tilde{p}}{\rho_\infty^2 \nu^2} \]

where the tilde indicates a dimensional quantity, the \( \infty \) subscript indicates freestream values, and \( L \) is any arbitrarily defined reference length. The Reynolds number, \( \text{Re}_\infty \), appearing in the viscous stress and heat flux terms is given by

\[ \text{Re}_\infty = \frac{\tilde{\rho}_\infty \nu \xi L}{\tilde{\mu}_\infty} . \]

The above equations do not fully define the motion of a compressible viscous fluid. Boundary conditions and initial values must be specified to complete the problem statement. At solid boundaries, \( u \) and \( v \) are identically zero by the no-slip condition and either an isothermal or adiabatic wall can be assumed as appropriate. An additional boundary condition is required and usually consists of an assumption regarding the normal pressure gradient. In addition, boundary conditions are needed at the inflow, outflow, and infinity boundaries. If the unsteady equations are solved, initial
conditions are needed to begin the computation. Since both the boundary and initial conditions are closely linked to the numerical scheme utilized to solve the governing equations, further discussion will be deferred until specific numerical algorithms are presented.

Transformation to Computational Space

To facilitate finite-difference solution of the governing equations, the physical domain is mapped from Cartesian \((x,y)\) space to computational \((\xi,\eta)\) space by a nonsingular transformation of the form

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

The mapping is constructed such that the simply connected physical domain is transformed into a rectangular region in computational space. The physical boundaries are mapped to either constant \(\xi\) or constant \(\eta\) lines simplifying boundary condition application by eliminating interpolation at the boundaries. Lines of constant \(\xi\) and constant \(\eta\) that define the discrete network of grid points in the interior of the domain are mapped to a uniformly spaced computational grid. Therefore, relatively simple finite-difference approximations for equally spaced points can be employed in computational space.
Equation 1 is transformed to \((\xi, \eta)\) space using the chain rule and is given in chain rule conservation law form by

\[
\frac{\partial U}{\partial t} + \nabla \cdot (\mathbf{E} + \mathbf{F} + \mathbf{G} + \mathbf{T}) = 0
\]

where \(U, E, F, G,\) and \(T\) are defined as before. The viscous stresses and heat fluxes appearing in \(E\) and \(F\) are also transformed using the chain rule. The metrics appearing in equation 2 are given by

\[
\eta_{\xi} = \frac{\partial y}{\partial \xi}, \quad \eta_{\eta} = -\frac{\partial y}{\partial \eta}
\]

where \(J\) is the Jacobian of the transformation and is given by

\[
J = \frac{1}{(x_{\xi} y_{\eta} - x_{\eta} y_{\xi})}
\]

The chain rule form of the equations is employed rather than the strong conservation form to obviate any constraints that might be imposed by the geometric conservation law (GCL) of Thomas and Lombard (1). The GCL is the result of attempting to maintain a uniform flow on a time varying grid. An analogous expression exists for space-marching applications if the grid varies in the marching direction. Although time
invariant grids are employed for the multiple-pass algorithms presented here, grids that varied in the marching direction are utilized for the single-pass algorithm. The chain rule form automatically guarantees that a uniform flow is preserved since the metrics are not included in the flux terms. Although well documented, the shock capturing capabilities of the chain rule form are somewhat questionable. In the work of Lombard et al. (2), the chain rule form has been employed successfully for shock capturing applications. Hindman (3) has shown that if the metrics are evaluated at appropriate locations, the chain rule form is identical to a weakly conservative formulation with a consistently differenced source term. It is felt that the discrepancies introduced by using the chain rule form will only become readily apparent as the shock approaches a discontinuity. These effects should not be significant here since the presence of viscosity precludes the existence of true discontinuities in the flow.

Thin-Layer Navier-Stokes Equations

Baldwin and Lomax (4) noted that since most of the effort expended computing a high Reynolds number viscous flow is devoted to resolving gradients normal to no-slip boundaries, the gradients parallel to the boundaries are often inadequately resolved. It then follows that the viscous terms parallel to the wall can be neglected. This assumption, which
is the basis of the thin-layer approximation, is equivalent to assuming that normal diffusion dominates streamwise diffusion in a high Reynolds number flow. An identical result is obtained by considering an asymptotic expansion of the NS equations in powers of \( \varepsilon \), where \( \varepsilon^2 = 1/Re \), and retaining only the lowest-order terms in each equation.

Assuming that \( \xi \) is the streamwise direction, the thin-layer Navier-Stokes (TLNS) equations are obtained by neglecting all viscous terms containing \( \xi \) derivatives. In addition, the heat flux terms containing \( \xi \) derivatives are also neglected. The transformed TLNS equations are given by

\[
\frac{\partial U}{\partial t} + \xi \frac{\partial E_i}{\partial \xi} + \eta \frac{\partial F_i}{\partial \eta} + \frac{\eta}{\partial \eta} + \xi \frac{\partial E_i}{\partial \eta} + \eta \frac{\partial F_i}{\partial \eta} = \eta \frac{\partial E'}{\partial \eta} + \eta \frac{\partial F'}{\partial \eta}
\]

where

\[
E' = [0, \tau_{xx}', \tau_{xy}', \mu \tau_{xx} + \nu \tau_{xy} - g_x^i]_T
\]

\[
F' = [0, \tau_{xy}', \tau_{yy}', \mu \tau_{xy} + \nu \tau_{yy} - g_y^i]_T
\]

with

\[
\tau_{xx}' = \frac{2}{3} \frac{\mu}{Re_\infty} \left(2 \eta \frac{\partial U}{\partial \eta} - \eta \frac{\partial V}{\partial \eta}\right)
\]
\[ \tau_{yy} = \frac{2 \mu}{3 \text{Re}_\infty} \left( 2 \eta_y \frac{\partial v}{\partial \eta} - \eta_x \frac{\partial u}{\partial \eta} \right) \]

\[ \tau_{xy} = \frac{\mu}{\text{Re}_\infty} \left( \eta_y \frac{\partial u}{\partial \eta} + \eta_x \frac{\partial v}{\partial \eta} \right) \]

and

\[ q_x = - \frac{\mu}{\text{Re}_\infty (\gamma - 1) M_\infty^2 \text{Pr}} \frac{1}{\eta_x} \frac{\partial T}{\partial \eta} \]

\[ q_y = - \frac{\mu}{\text{Re}_\infty (\gamma - 1) M_\infty^2 \text{Pr}} \frac{1}{\eta_y} \frac{\partial T}{\partial \eta} . \]

**Preliminary Algorithm Considerations**

Steady flow fields are typically computed by solving the unsteady form of the governing equations, either the NS equations (equation 2) or the TLNS equations (equation 3), using a time-asymptotic approach. For a two-dimensional problem, the time-asymptotic approach is implemented by specifying a plane of initial data and using the chosen integration algorithm to advance the solution to the next time level. A sequence of new values is computed by repeated application of the numerical scheme that, hopefully, converges to the steady-state solution. It is possible to employ a time-marching scheme because both the NS and TLNS equations are systems of partial differential equations that are of
mixed hyperbolic/parabolic type in time. Equivalently, since the equations are of mixed hyperbolic/parabolic type in time, the time-marching approach is well posed.

Due to the computational expense associated with the time-marching approach, it is tempting to consider the possibility of eliminating the unsteady terms from the governing equations and use a space-marching algorithm. A space-marching approach is particularly attractive since one dimension is effectively removed from the problem under consideration. A two-dimensional space-marching scheme is initiated at the inflow boundary and, assuming that $\zeta$ is the marching direction, lines of data corresponding to lines of constant $\zeta$ are "marched" to the outflow boundary. Note that the solution process is completed in a single pass and that the computational effort required is roughly equivalent to the effort needed for a single step of a time-marching algorithm.

Unfortunately, the steady forms of both the NS and the TLNS equations are of mixed hyperbolic/elliptic type in the marching direction $\zeta$. Even a cursory examination reveals that the streamwise diffusion terms are at least partly responsible for the elliptic behavior of the steady NS equations. However, the presence of these terms does not fully account for the elliptic character of the equations since the same behavior is present in the steady form of the TLNS equations. The cause of this behavior is readily apparent if the
eigenvalue structure of the TLNS equations is examined. Equation 3 can be written in nonconservative form as

\[
\frac{\partial U}{\partial t} + A \frac{\partial U}{\partial \xi} + B \frac{\partial U}{\partial \eta} = \eta_x \frac{\partial E_v'}{\partial \eta} + \eta_y \frac{\partial F_v'}{\partial \eta} \tag{4}
\]

where

\[
A = \xi_x \frac{\partial E_i}{\partial U} + \xi_y \frac{\partial F_i}{\partial U}
\]

\[
B = \eta_x \frac{\partial E_i}{\partial U} + \eta_y \frac{\partial F_i}{\partial U}
\]

and the inviscid Jacobians \( \partial E_i / \partial U \) and \( \partial F_i / \partial U \) are given in Appendix A. The eigenvalues of \( A \) are the slopes of the characteristics in \((\xi, t)\) space. Positive eigenvalues denote wave propagation in time in the positive \( \xi \) direction while negative eigenvalues denote wave propagation in the negative \( \xi \) direction. The space-marching problem is well posed only if all of the eigenvalues of \( A \) are of the same sign and correspond to wave propagation in the marching direction. The eigenvalues of \( A \) are given by

\[
\lambda_{1,2} = k_n \hat{u}
\]

\[
\lambda_3 = k_n (\hat{u} + a)
\]

\[
\lambda_4 = k_n (\hat{u} - a) \tag{5}
\]
where \( \hat{u} \) is the velocity component normal to a line of constant \( \xi \) given by

\[
\hat{u} = \left( \xi_x u + \xi_y v \right) / k_n
\]

\( a \) is the nondimensional local sound speed given by

\[
a = \sqrt{\gamma p / \rho}
\]

and

\[
k_n = \sqrt{\xi_x^2 + \xi_y^2}
\]

Assuming that the direction of marching is positive \( \xi \), examination of the eigenvalues given in equation 5 yields the following two constraints: (1) reversed flow is not allowed, and (2) the velocity component in the marching direction must be supersonic. The first constraint eliminates computing flows with separation in the marching direction. The second constraint is far more restrictive since, regardless of freestream conditions, a region of subsonic flow always exists near a no-slip surface. Therefore, space-marching algorithms for the TLNS equations are not feasible unless the effects of the negative eigenvalues can be suppressed. The next section examines several methods of modifying the eigenvalue structure so that the space-marching problem is well posed. In addition, the effects of these modifications on solution accuracy are also discussed.
CHAPTER 3. SINGLE-PASS PHILOSOPHY

Parabolized Navier-Stokes Equations

The eigenvalue analysis presented in the previous chapter showed that the space-marching problem is not well posed for the TLNS equations. This has not, however, deterred development of space-marching algorithms for the so-called parabolized Navier-Stokes (PNS) equations. In two dimensions, the PNS equations are formally equivalent to the steady form of the TLNS equations. The three-dimensional PNS equations retain viscous terms in the crossflow plane that are not present in the TLNS equations. The crux of the PNS approximation is the suppression of the elliptic behavior present in the steady TLNS equations so that the resulting system of equations is of mixed hyperbolic/parabolic type in the streamwise direction and the space-marching problem is well posed. If the elliptic influence is not excluded and the steady form of the TLNS equations is solved using a conventional space-marching approach, exponentially growing solutions called departure solutions occur. It has long been understood that the streamwise pressure gradient is responsible for this behavior. In fact, Lighthill (5) reported a similar behavior for the boundary layer equations if the streamwise pressure gradient is not prescribed. The success of the PNS approach depends on introducing an
inconsistency in evaluating the streamwise pressure gradient that does not severely degrade the accuracy of the computation. In the following paragraphs that summarize several of the methods employed in PNS algorithms, x is assumed to be the marching direction.

Rudman and Rubin (6) considered hypersonic viscous flow near the sharp leading edge of a flat plate. A series expansion technique was employed to derive a system of equations that, to zeroth order, were roughly equivalent to the steady form of the TLNS equations with one major exception: the pressure gradient was neglected from the streamwise momentum equation. The resulting system of equations does not admit elliptic behavior in the marching direction so that the space-marching problem is well posed. The form given in reference 6 is often called the parabolic Navier-Stokes equations. Rudman and Rubin were quick to point out that neglecting the streamwise pressure gradient does not imply that it is small, only that to zeroth order, the primary balance in the streamwise momentum equation occurs between the inertia terms and the viscous terms. An explicit finite-difference scheme was used to integrate the governing partial differential equations. This paper is referenced primarily to illustrate that a stable space-marching scheme is obtained if the streamwise pressure gradient is omitted.

Lubard and Helliwell (7) employed an iterative, implicit
finite-difference scheme to solve the PNS equations for the case of hypersonic flow past a cone at high angle of attack. Of particular interest in this study are the results of a Fourier stability analysis for the space-marching scheme. Lubard and Helliwell found that there was no restriction on the marching step size if the streamwise pressure gradient was omitted. This is not surprising since, as in reference 6, the space-marching problem is well posed if the streamwise pressure gradient is neglected. A particularly unusual constraint is imposed on the marching step size if the streamwise pressure gradient is evaluated at the previous marching station. The space-marching scheme was shown to be unstable if the marching step size, $\Delta x$, was less than a certain minimum value, i.e., $\Delta x \geq (\Delta x)_{\text{min}}$ for stability. For a two-dimensional problem, the minimum step size is given by

$$(\Delta x)_{\text{min}} = \frac{(\rho u/\mu)((1/Mx^2) - 1)(\Delta y)^2}{4y\sin^2(k_m\Delta y/2)}$$

(6)

where $Mx$ is the local Mach number in the marching direction and $k_m$ is the wave number. According to Rubin (8), this minimum step size corresponds to the upstream extent of elliptic interaction present in the PNS equations. Stable space-marching solutions are obtained for $\Delta x \geq (\Delta x)_{\text{min}}$ since the region of elliptic interaction is overstepped and the elliptic behavior is suppressed. For $\Delta x < (\Delta x)_{\text{min}}$, the space-
marching scheme attempts to model the elliptic behavior inherent in the solution and a departure solution occurs. Rubin contends that the extent of upstream interaction is of the same order as the thickness of the subsonic layer. Lubard and Helliwell also determined that if no special treatment of the streamwise pressure gradient is employed, the minimum marching step size is doubled.

Vigneron et al. (9) successfully used the PNS equations to compute the supersonic flow over delta wing configurations. The PNS equations were solved using the noniterative, implicit approximate factorization algorithm of Beam and Warming (10). The most distinctive feature of this approach is the method employed to suppress the streamwise elliptic behavior in subsonic regions. The Vigneron splitting, given by

$$\frac{\partial p}{\partial x} = \omega \frac{\partial p}{\partial x} + (1 - \omega) \frac{\partial p}{\partial x}$$

essentially separates the streamwise pressure gradient into two parts: $\omega \frac{\partial p}{\partial x}$ which is the portion associated with hyperbolic behavior in the streamwise direction and the portion associated with the undesirable elliptic behavior, $(1 - \omega) \frac{\partial p}{\partial x}$. The "hyperbolic" portion of the pressure gradient requires no special treatment. The "elliptic" portion was treated as a source term and evaluated at the previous marching station or neglected completely. An
eigenvalue analysis was used to determine the range of $\omega$ that insured that the appropriate space-marching eigenvalues were real, i.e., the equations were hyperbolic. The PNS equations have real eigenvalues in regions where the Mach number in the marching direction is less than unity if

$$\omega \leq \frac{\gamma Mx^2}{1 + (\gamma - 1)Mx^2}. \quad (7)$$

It is important to note that at a no-slip boundary, $\omega$ is zero so that none of the "hyperbolic" portion of the streamwise pressure gradient may be included. Vigneron et al. also studied the stability of the numerical algorithm using a Fourier stability analysis and determined that unconditional stability was obtained only when the "elliptic" portion of the streamwise pressure gradient was neglected. Otherwise, stability constraints similar to those described by Lubard and Helliwell (7) were observed.

Schiff and Steger (11) computed the supersonic flow past several two-dimensional and three-dimensional bodies by solving the PNS equations using the implicit, noniterative algorithm of reference 10. In this study, departure solutions were suppressed by employing the so-called sublayer approximation. The crux of the sublayer approximation is the observation that for a thin subsonic layer, the pressure gradient normal to the body surface is negligible. Therefore,
the pressure in the subsonic region near a no-slip boundary is essentially impressed from the outer supersonic flow and does not depend on the local flow variables. The sublayer approximation was implemented by specifying the pressure in the subsonic region to be equal to the pressure in the adjacent supersonic flow at the previous marching station. Presumably, employing the sublayer approximation leads to a stable, well-posed space-marching algorithm since the dependence of the eigenvalues on the sound speed is eliminated. However, Schiff and Steger reported that departure solutions occurred as the marching step size was decreased. Barnette (12) studied the departure behavior of several different forms of the sublayer approximation. Significantly, Barnette found that departure solutions were a function of the downstream extent of the computational domain. That is, solutions that are apparently stable for a given marching step size are subject to departure behavior if the computational domain is extended far enough downstream. Unconditionally stable space-marching solutions were obtained only if the streamwise pressure gradient was neglected in the subsonic region.

Stephenson (13) also used the implicit scheme of reference 10 to compute the supersonic viscous flow past spherically blunted, biconic geometries at several angles of attack. The flux splitting ideas of Steger and Warming (14)
were employed to separate the streamwise convective flux vector into a part that was associated with wave propagation in the streamwise direction and a part associated with upstream propagation. Stephenson evaluated the part associated with upstream propagation at the previous marching station using backward differences. This approach is touted as being "stable" in reference 13 although no stability analysis of any type is presented.

Equivalent Flux Split Formulations

In light of the approach utilized by Stephenson, it is interesting to interpret the work of Vigneron et al. (9) and Schiff and Steger (11) in a flux splitting context since this could prove helpful in explaining the behavior of the different schemes in subsonic regions. First, a more detailed discussion of the flux splitting concepts and the specific application of Stephenson is presented. According to Steger and Warming (14), the streamwise convective flux vector $E_i$ satisfies the homogeneous property and has a complete set of linearly independent eigenvectors. Therefore, $E_i$ can be written using a similarity transformation as

$$E_i = AU = T A T^{-1} U$$

where $A$ is the Jacobian matrix $\partial E_i / \partial U$, $A$ is the diagonal matrix of the eigenvalues of $A$, and $T^{-1}$ is the matrix with
rows that are the left eigenvectors of $A$ taken in the same order as the elements of $\Lambda$. The eigenvalues of $A$ are given by equation 5 with $\xi_x = 1$ and $\xi_y = 0$,

$$\lambda_{1,2} = u$$
$$\lambda_3 = u + a$$
$$\lambda_4 = u - a .$$

Since the condition for marching is that the eigenvalues of $A$ must be of the same sign and correspond to wave propagation in the marching direction, a possible approach might be to split the matrix $A$ into two matrices, $A^+$ containing only positive elements and $A^-$ containing only negative elements. Using the similarity transformation, the streamwise convective flux vector $E_i$ can be written as

$$E_i = TA^+ T^{-1} U + TA^- T^{-1} U = E_i^+ + E_i^- .$$

In regions where the velocity component in the marching direction is supersonic, i.e., $u > a$, all of the eigenvalues are positive and $E_i^- = 0$. In subsonic regions, $\lambda_4$ is negative so that the eigenvalue splitting becomes

$$\lambda_{1,2}^+ = u$$
$$\lambda_3^+ = u + a$$
$$\lambda_4^+ = 0$$
$$\lambda_{1,2}^- = 0$$
$$\lambda_3^- = 0$$
$$\lambda_4^- = u - a .$$
It is important to note that although the eigenvalues of $\frac{\partial E_i^\pm}{\partial U}$ are not the same as those given above, they are, according to reference 14, nonnegative. Therefore, $E_i^+$ can be approximated using backward differences as part of the space-marching procedure. Although $E_i^-$ is the part of the flux vector associated with upstream influence, Stephenson evaluated $E_i^-$ at the previous marching station using a backward difference.

The Vigneron splitting of the streamwise pressure gradient is given by

$$E_i = E_i^* + \xi_i = T\alpha^* T^{-1} U + \xi T^{-1} U$$

where

$$E_i^* = [\rho u, \rho u^2 + \omega \rho, \rho uv, (E_t + p)u]^T$$

$$\xi_i = [0, (1 - \omega)p, 0, 0]^T.$$  

The equivalent eigenvalue splitting can be determined from the generalized flux vectors given in reference 14. Using these generalized flux vectors, the equivalent eigenvalue splitting for this case is given by

$$\lambda_{1,2}^* = \omega u$$

$$\lambda_3^* = \omega(u + a) + (1 - \omega)\gamma u$$

$$\lambda_4^* = \omega(u - a) + (1 - \omega)\gamma u$$
and

\[ \lambda_{1,2} = (1 - \omega)u \]
\[ \lambda_3 = (1 - \omega)(u + a) - (1 - \omega)\gamma u \]
\[ \lambda_4 = (1 - \omega)(u - a) - (1 - \omega)\gamma u \]

where \( \omega \) is defined by equation 7. It is interesting to note that if the requirement that \( \lambda^* \) must be nonnegative is enforced, then an expression for \( \omega \), given by

\[
\omega \leq \frac{\gamma Mx}{1 + (\gamma - 1)Mx}
\]

is obtained that is different from equation 7. This is a manifestation of the fact that the eigenvalues of \( \partial E_i^*/\partial U \) are different from the eigenvalues given above. In fact, although not shown here, an expression for \( \omega \) identical to equation 7 can be obtained by requiring that the eigenvalues of \( \partial E_i^*/\partial U \) be nonnegative. It is also interesting to note that although \( \lambda \) contains positive elements as well as negative elements, the eigenvalues of the Jacobian matrix \( \partial E_i/\partial U \) are all nonpositive. These eigenvalues are given by

\[ \lambda_{1,2,3} = 0 \]
\[ \lambda_4 = - (\gamma - 1)(1 - \omega)u \]

and \( \partial E_i/\partial U \) is given in Appendix A. As stated previously,
Vigneron et al. (9) reported unconditional stability only when $\dot{E}_i$ was neglected. The presence of the negative eigenvalue in $\partial\dot{E}_i/\partial U$ would seem to be the destabilizing factor when $\dot{E}_i$ is approximated using a backward difference since this eigenvalue represents upstream propagation of information. These observations are also applicable to the methods of Rudman and Rubin (6) ($\omega = 0$ and neglect $\dot{E}_i$) and Lubard and Helliwell (7) ($\omega = 1$ or $\omega = 0$ and evaluate $\dot{E}_i$ at the previous station).

According to Schiff and Steger (11), the equivalent eigenvalue splitting for the sublayer approximation is given by

$$\begin{align*}
\lambda_{1,2}^* &= u & \lambda_{1,2} &= 0 \\
\lambda_3^* &= u & \lambda_3 &= a \\
\lambda_4^* &= u & \lambda_4 &= -a .
\end{align*}$$

As in the equivalent eigenvalue splitting of the Vigneron approach, $\hat{\lambda}$ contains positive as well as negative elements. In this case, the eigenvalues of $\partial\dot{E}_i^*/\partial U$ and $\partial\dot{E}_i/\partial U$ are identical to those given above. In the sublayer approximation described in reference 11, $\dot{E}_i$ is evaluated at the previous marching station at a supersonic point outside the viscous sublayer. The presence of the negative eigenvalue would again seem to explain the behavior described in reference 12 where unconditionally stable solutions were obtained only if $\dot{E}_i$ was
neglected. Based on these observations, it would appear that since an analogous term is retained by Stephenson, the method described in reference 13 would be subject to departure solutions.

Each of the methods described above is inconsistent if the elliptic portion of the streamwise pressure gradient is retained since the streamwise grid spacing cannot be arbitrarily refined. The elliptic contribution to the streamwise convective flux vector causes the problem to be ill posed and must be eliminated if unconditionally stable space-marching algorithms are desired for the PNS equations. If, however, a rational method of determining the minimum marching step size is available, it may be possible to include the effects of the streamwise pressure gradient.

Minimum Marching Step Size

The constraint placed on the marching step size when the portion of the convective flux vector responsible for the elliptic behavior is included has been the subject of considerable interpretation. The generally accepted explanation is the one given by Rubin (8). As stated previously, Rubin asserts that the minimum marching step size appears to represent the upstream extent of the elliptic interaction and is the order of the thickness of the subsonic layer. Presumably, stable space-marching solutions are
obtained if this region is overstepped. Barnette (12) reported that a PNS algorithm employing the sublayer approximation was subject to departure solutions if the downstream extent of the computational domain was large enough. It is particularly unsettling that departure solutions can occur for apparently stable step sizes if the number of marching steps is increased. The purpose of this section is to investigate the departure solution with the idea of understanding the minimum marching step size using a simple model problem that still retains sufficient complexity to yield meaningful results.

Since the behavior of departure solutions is apparently closely coupled to convection rather than diffusion, the two-dimensional steady Euler equations are selected as a starting point for development of a model problem. The two-dimensional steady Euler equations in Cartesian \((x,y)\) space are given in nonconservative form by

\[
\frac{\partial U}{\partial x} + C \frac{\partial U}{\partial y} = 0
\]  

where

\[
C = A^{-1}B
\]

and the Jacobian matrices \(A = \partial E_i/\partial U\) and \(B = \partial F_i/\partial U\) are given in appendix A. In addition, periodic boundaries are assumed.
so that the problem under consideration becomes a pure initial value problem. Finally, the distribution of \( U \) must be specified on the initial data line.

At this point, meaningful analysis of equation 8 is a formidable, although not impossible task. Some of the concepts introduced in the discussion of flux split algorithms can be employed to simplify the analysis. Since the matrix \( C \) has a full complement of linearly independent eigenvectors, a similarity transformation exists that is of the form

\[
C = QAQ^{-1}
\]

where \( \Lambda \) is a diagonal matrix of the eigenvalues of \( C \), given by

\[
\lambda_{1,2} = \frac{v}{u}
\]

\[
\lambda_{3,4} = \frac{uv \pm a\sqrt{u^2 + v^2 - a^2}}{u^2 - a^2}
\]

and \( Q^{-1} \) is the matrix with rows that are the left eigenvectors of \( C \) taken in order. Employing the similarity transformation, equation 8 can be written as

\[
Q^{-1} \frac{\partial U}{\partial x} + \Lambda Q^{-1} \frac{\partial U}{\partial y} = 0.
\]

For the purpose of a linear stability analysis, \( Q^{-1} \) is assumed to be constant so that the system of equations given above can be transformed to the uncoupled system
\[
\frac{\partial z_q}{\partial x} + \lambda_q \frac{\partial z_q}{\partial y} = 0 \quad (q = 1, 2, 3, 4) \tag{9}
\]

where \( z_q \) are the elements of the vector \( Z \) given by

\[
Z = Q^{-1} U
\]

and \( \lambda_q \) are the eigenvalues of \( C \). Note that the elements of \( Z \) may be complex since the eigenvectors present in \( Q^{-1} \) have complex elements in subsonic regions.

Applying first-order Euler implicit differencing to equation 9 and employing central differencing to approximate the \( y \) derivative yields the finite-difference scheme

\[
z_{i+1,j} + \frac{\lambda \Delta x}{2 \Delta y} (z_{i+1,j+1} - z_{i+1,j-1}) = z_{i,j} \tag{10}
\]

where the \( q \) subscript is suppressed for simplicity, \( x = i \Delta x \), and \( y = j \Delta y \). The amplification factor \( G_q \) can be determined based on a Fourier stability analysis of the linearized equation. If a term of the form

\[
z_{i,j} = \Gamma \exp(ai \Delta x) \exp(\sqrt{-1} k_m j \Delta y),
\]

where \( k_m \) is the wave number and \( \Gamma \) may be complex, is substituted into equation 10, the amplification factor becomes

\[
G_q = (1 - \text{Im}(\lambda_q)) Cn \sin \beta + \sqrt{-1} \Re(\lambda_q) Cn \sin \beta)^{-1} \tag{11}
\]

where \( \lambda_q = \lambda_q / |\lambda_q| \), \( Cn = |\lambda_q| \Delta x / \Delta y \), \( \beta = k_m \Delta y \), and \( |\lambda_q| \) is the
magnitude of $\lambda_q$. For stability, the magnitude of $G_q$ must be less than or equal to unity. It is easy to show that if $\lambda_q$ is purely real, i.e., no regions of subsonic flow, the finite-difference algorithm is unconditionally stable. However, if $\text{Im}(\lambda_q) \neq 0$, the space-marching scheme is stable only if

$$C_n \geq \frac{2 \text{Im}(\lambda_q)}{\sin \beta}.$$  \hspace{1cm} (12)

The right-hand side of equation 12 can be interpreted as representing a minimum stable Courant number, $C_n$, for the numerical algorithm given by equation 10 and is analogous to equation 6. It is important to emphasize that this constraint arises solely because of the presence of an imaginary component in $\lambda_q$.

An alternative view of the minimum $C_n$ constraint, or equivalently, the restriction on the minimum marching step size, can be obtained by studying the functional relationship between the magnitude of the amplification factor $G_q$ and $\beta$ for fixed values of $C_n$ and $\lambda_q$. Figure 2 is a polar plot of the magnitude of the amplification factor versus $\beta$ for $C_n = 2.5, 5.0, 10.0$ and $\lambda_q = \sqrt{2}/2(1 + \sqrt{-1})$. This figure clearly shows that, for fixed $C_n$ and $\lambda_q$, there are certain ranges of $\beta$ where $|G_q| \geq 1$. That is, equation 12 defines the frequency components of the solution that are amplified by the numerical algorithm as the solution is advanced from one marching
Figure 2. Magnitude of amplification factor for Euler implicit scheme with no smoothing
station to the next. Significantly, Figure 2 is symmetric about $\pi/2$ so that there are unstable regions at both low and high frequencies. In light of this discussion, departure solutions appear to represent the response of the numerical algorithm for complex $\lambda_q$ when frequency components in the unstable range are present in the numerical solution.

From Figure 2 it is also apparent that increasing the Courant number only serves to decrease the range of $\beta$ where destabilizing effects are present and does not affect the maximum magnitude of the amplification factor. This behavior is readily explained by observing that in equation 11, the quantities $C_n$ and $\sin \beta$ always appear together as a product. For a fixed value of $\lambda_q$, $C_n$ essentially acts as a scale factor on $\beta$. Therefore, the maximum magnitude of $G_q$ is dependent only on $\lambda_q$. As stated previously, if $\text{Im}(\lambda_q) = 0$, the numerical algorithm is unconditionally stable, i.e., $|G_q| \leq 1$ for all values of $\beta$. However, if $\text{Re}(\lambda_q) = 0$, the magnitude of the amplification factor approaches infinity for two values of $\beta$ depending on the value of $C_n$. At a no-slip boundary, this is precisely the situation that occurs. The specific eigenvalue that causes the undesirable behavior is given by

$$
\lambda_4 = \frac{uv - a\sqrt{u^2 + v^2 - a^2}}{u^2 - a^2}
$$

and at a no-slip boundary $u = v = 0$ which yields $\lambda_4 = \sqrt{-1}$. 
For a typical boundary layer type flow, \( v << a \) in the region near the wall so that \( \text{Im}(\lambda_4) \gg \text{Re}(\lambda_4) \) and the maximum magnitude of \( G_4 \) is quite large. The important point to be made here is that by choosing \( C_n \) to be large, the range of \( \beta \) where the numerical scheme is unstable is significantly reduced. However, for \( \text{Im}(\lambda_q) \neq 0 \) and any finite \( C_n \), there are still frequency components of the solution that are amplified. For a very simple problem, it is possible to construct the grid and choose a value of \( C_n \) so that the numerical scheme is stable. Unfortunately, it is impossible to control the frequency content of the numerical solution for a realistic problem.

Typically, implicit algorithms for the PNS equations that utilize central differences employ some form of artificial dissipation to improve the numerical scheme by eliminating high frequency oscillations in the numerical solution. Fourth-difference explicit smoothing (see reference 10) can be added to equation 10 to obtain

\[
\begin{align*}
    z_{i+1,j} &= z_{i,j} + \frac{\lambda_{\Delta x}}{2\Delta y} (z_{i+1,j+1} - z_{i+1,j-1}) \\
    &= z_{i,j} + \frac{\alpha_{\Delta x}}{8} (z_{i,j+2} - 4z_{i,j+1} + 6z_{i,j} - 4z_{i,j-1} + z_{i,j-2})
\end{align*}
\]

where \( \alpha \) is the smoothing coefficient and the \( q \) subscript has again been suppressed for simplicity. The amplification
factor for the Euler implicit scheme with the explicit fourth-difference smoothing is given by

$$\mathcal{G}_q = f(\Delta x, \beta) \quad G_q$$

where

$$f(\Delta x, \beta) = 1 - \frac{\Delta x}{8} (2 \cos 2\beta - 8 \cos \beta + 6)$$

and $G_q$ is given by equation 11. Clearly, this form of smoothing is stabilizing only if the magnitude of $f(\Delta x, \beta)$ is less than unity which requires that $0 \leq \Delta x \leq 1$. Figure 3 is a polar plot of the magnitude of $G_q$ versus $\beta$ for $\Delta x = 0.0, 0.25, 0.5, \text{Cn} = 2.5$, and $\chi_q = \sqrt{2}/2(1 + \sqrt{-1})$. This figure illustrates that adding fourth-difference smoothing does indeed reduce the magnitude of the amplification factor at the higher frequencies. This does not imply, however, that adding smoothing will guarantee that no high frequencies are amplified. In fact, the purpose here is not to advocate using smoothing of this type but only to point out that, by adding smoothing, the "departure behavior" of the numerical scheme can be modified. This point is particularly significant since, according to reference 15, the AFWAL PNS code has no fewer than four different smoothing parameters. Figure 3 also shows that adding the fourth-difference smoothing does not appreciably affect the amplification factor at low
Figure 3. Magnitude of amplification factor for Euler implicit scheme with smoothing.
frequencies. If low frequency components of the solution in the unstable range are present, these components will be amplified and the solution will diverge. The amplification of low frequency components can be advanced as a plausible explanation of the behavior described by Barnette (12). Further experiments are required to verify this hypothesis.

While the analysis above is considerably different from that given by Rubin (8) for an incompressible viscous flow, the main results are, in fact, very similar except for one critical difference in interpretation. In the region near the wall, \( a \gg u \) and \( a \gg v \) so that the flow is essentially incompressible. Following the lead of Rubin, equation 12 is approximated for small values of \( \beta \), i.e., low frequency components of the solution, to obtain

\[
\frac{\Delta x}{\lambda} \geq \frac{1}{\pi} \tag{13}
\]

since \( \beta = k_m \Delta y \) and \( k_m = 2\pi/\lambda \) where \( \lambda \) is the wavelength of the solution component under consideration. The lowest frequency component of the solution has \( \lambda = 2y_m \) where \( y_m \) is the total height of the computational grid, or for an incompressible flow, the thickness of the subsonic layer, so that the equation 13 becomes

\[
\frac{\Delta x}{2y_m} \geq \frac{1}{\pi}
\]

which is precisely the result given in reference 8. Rubin
does not consider compressible supersonic flow in his theoretical development but only states that numerical results indicate that a similar relationship exits. The crucial difference here is that Rubin interprets \( \Lambda \) as a parameter related to the physics of the problem when, in fact, it is purely related to the geometry of the grid. The statement that the minimum marching step size is related to the thickness of the subsonic layer in a supersonic compressible flow is equivalent to stating that the components of the solution with wavelengths longer than the thickness of the subsonic layer are not affected by the numerical algorithm in the subsonic region. Based on these results, it is apparent that there is no reliable way of determining a stable marching step size or even if one exists.

At this point it should be noted that the analysis above assumed no special treatment of the streamwise pressure gradient and also neglected viscous effects. The resulting system in both cases is not diagonalizable and a more complex analysis is required. Although the results of this analysis are more qualitative in nature than quantitative, this does not affect the applicability of several of the more important conclusions to the full viscous problem. Therefore, the only consistent approach would appear to be neglecting the portion of the streamwise pressure gradient responsible for elliptic behavior. Unfortunately, as reported by Barnette (12) and
Rakich (16), neglecting this term can lead to significant errors in the predicted skin friction and heat transfer coefficients at no-slip surfaces. Part of the evaluation of the numerical scheme presented in the next section is to determine the severity of this error.

**Numerical Algorithm**

In two dimensions, the PNS equations are equivalent to the steady form of the TLNS equations. The PNS equations, repeated here for convenience, are given in chain rule form by

\[
\frac{\partial E_i}{\partial \xi} + \frac{\partial F_i}{\partial \eta} + \frac{\partial E_i}{\partial \eta} + \frac{\partial F_i}{\partial \eta} = 0
\]

where \(E_i, F_i, E_v',\) and \(F_v'\) are defined as before and \(\xi\) is the marching direction. The general strategy utilized here is to solve equation 14 using a space-marching approach employing an implicit finite-difference algorithm.

**Streamwise differencing**

The numerical scheme employed to solve equation 14 is a space-marching version of the implicit noniterative algorithm of Beam and Warming (10) specialized to two dimensions. This algorithm was first applied to the PNS equations, in three
dimensions, by Vigneron et al. (9). The generalized differencing of reference 10 is employed to advance the solution from the marching station \( j \) to the next marching station \( j+1 \). The generalized differencing of \( E_i \) specialized for an implicit scheme is given by

\[
\Delta E_{ij} = \left(1 - \frac{\theta}{3}\right) \Delta \xi \left(\frac{\partial E_i}{\partial \xi}\right)_{j+1} + \frac{\theta}{3} \Delta E_{ij-1} + O[(1-\theta)\Delta \xi^2 + \Delta \xi^3]
\]

where \( \Delta E_{ij} = E_{ij+1} - E_{ij} \), \( \theta = 0 \) is the first-order Euler implicit scheme, \( \theta = 1 \) is the second-order three-point backward scheme, and \( \xi = (j-1)\Delta \xi \). It is important to note that although the difference scheme is said to be first-order accurate for \( \theta = 0 \) and second-order accurate for \( \theta = 1 \), this refers to the formal accuracy on an equally spaced grid in computational space. If any form of grid stretching is utilized in physical space, there is no guarantee that this order of accuracy will be maintained. An analogous expression for \( \Delta F_{ij} \) can be obtained and is given by

\[
\Delta F_{ij} = \left(1 - \frac{\theta}{3}\right) \Delta \xi \left(\frac{\partial F_i}{\partial \xi}\right)_{j+1} + \frac{\theta}{3} \Delta F_{ij-1} + O[(1-\theta)\Delta \xi^2 + \Delta \xi^3]
\]

where \( \Delta F_{ij} = F_{ij+1} - F_{ij} \). By algebraic manipulation of the above two equations and equation 14, the implicit algorithm employed for the single-pass solutions can be obtained and is given in "delta" form by
\[ \xi_x \Delta E_{ij}^* + \xi_y \Delta F_{ij}^* \]

\[ + \left(1 - \frac{\theta}{3}\right) \Delta \xi \left\{ \eta \frac{\partial}{\partial \eta} \left( \Delta E_{ij} - \Delta E_{ij}' \right) + \eta \frac{\partial}{\partial \eta} \left( \Delta F_{ij} - \Delta F_{ij}' \right) \right\} \]

\[ = - \left(1 - \frac{\theta}{3}\right) \Delta \xi \left\{ \eta \frac{\partial}{\partial \xi} \left( E_{ij} - E_{ij}' \right) + \eta \frac{\partial}{\partial \xi} \left( F_{ij} - F_{ij}' \right) \right\} \]

\[ - \left(1 - \frac{\theta}{3}\right) \Delta \xi \left\{ \eta \frac{\partial}{\partial \eta} \left( E_{ij} - E_{ij}' \right) + \eta \frac{\partial}{\partial \eta} \left( F_{ij} - F_{ij}' \right) \right\} \]

\[ + \frac{\theta}{3} \left\{ \xi_x \Delta E_{ij}^* + \xi_y \Delta F_{ij}^* \right\} + O((1-\theta)\Delta \xi^2 + \Delta \xi^3) \]

where the "delta" terms are defined as

\[ \Delta E_{ij}^* = E_{ij}^* + \Delta E_{ij} \]

\[ \Delta E_{ij}' = E_{ij}' + \Delta E_{ij}' \]

\[ \Delta F_{ij}^* = F_{ij}^* + \Delta F_{ij} \]

\[ \Delta F_{ij}' = F_{ij}' + \Delta F_{ij}' \]

and the Vigneron technique has been utilized to split \( E_i \) and \( F_i \) as

\[ E_i = E_i^* + E_i + \Delta E_i \]

\[ F_i = F_i^* + F_i + \Delta F_i \]

where
\[ E_i^* = [\rho u, \rho u^2 + \omega p, \rho uv, (E_t + p)u]^T \]
\[ E_i = [0, (1 - \omega)p, 0, 0]^T \]
\[ F_i^* = [\rho v, \rho uv, \rho v^2 + \omega p, (E_t + p)v]^T \]
\[ F_i = [0, 0, (1 - \omega)p, 0]^T . \]

The parameter \( \omega \) has been discussed previously and is given in transformed \((\xi, \eta)\) space by

\[ \omega = \sigma f(M\xi) \quad \text{for} \quad \sigma f(M\xi) < 1 \]
\[ \omega = 1 \quad \text{for} \quad \sigma f(M\xi) \geq 1 \quad \text{(16)} \]

where

\[ f(M\xi) = \frac{\gamma M\xi^2}{1 + (\gamma - 1)M\xi^2} , \]

\( \sigma \) is a safety factor \( (\sigma = 0.8) \), and \( M\xi = \hat{u}/a \) where \( \hat{u} \) and \( a \) were defined previously. The terms involving \( E_i \) and \( F_i \) represent the part of the streamwise pressure gradient associated with elliptic influence and are included here only for completeness. In the actual single-pass algorithm, these terms are neglected based on the discussion in the previous section. In addition, the metric quantities \( \xi_x, \xi_y, \eta_x, \) and \( \eta_y \) are evaluated at \( j+1 \) as described in Appendix B.
Linearization

The "delta" terms in equation 15 are linearized using a Taylor series expansion about the known state, in this case marching station j. The inviscid flux vectors can be linearized as

\[ E_{i+1}^* = E_i^* + \left( \frac{\partial E_i^*}{\partial U} \right)_j \Delta U_j + O(\Delta^2) \]  

\[ F_{i+1}^* = F_i^* + \left( \frac{\partial F_i^*}{\partial U} \right)_j \Delta U_j + O(\Delta^2) \]  

\[ E_{i+1} = E_i + \left( \frac{\partial E_i}{\partial U} \right)_j \Delta U_j + O(\Delta) \]  

\[ F_{i+1} = F_i + \left( \frac{\partial F_i}{\partial U} \right)_j \Delta U_j + O(\Delta) \]  

where the Jacobian matrices \( \frac{\partial E_i^*}{\partial U}, \frac{\partial F_i^*}{\partial U}, \frac{\partial E_i}{\partial U}, \frac{\partial F_i}{\partial U} \) are given in Appendix A. Both \( \frac{\partial E_i}{\partial U} \) and \( \frac{\partial F_i}{\partial U} \) were derived assuming that \( \omega \) is independent of \( U \). Since \( \omega \) is actually a multiplier of the pressure gradient rather than the pressure, the streamwise location of the evaluation of \( \omega \) does not affect the order of accuracy of this linearization.

Ideally, \( \omega \) should be evaluated at the marching station \( j+1 \) since an implicit method is being used. The only detrimental effect of this approximation is that if \( \omega \) is a rapidly varying function of \( \xi \), it might be possible to include too much of the
streamwise pressure gradient resulting in a departure solution. This should not be a significant effect since \( \omega \) is a relatively smooth function of \( \xi \). In addition, the safety factor \( \sigma \) employed in the definition of \( \omega \) also helps to alleviate this effect. It is important to note that to achieve second-order accuracy, it is necessary for the evaluation of \( \partial E^* / \partial U \) and \( \partial F^* / \partial U \) to be more accurate than the evaluation of \( \partial E / \partial U \) and \( \partial F / \partial U \) since the latter terms are multiplied by \( \Delta \xi \). An obvious result of this is that it is consistent to neglect all of the terms in the second line of equation 15 if a first-order scheme is desired since they are multiplied by \( \Delta \xi \). However, the algorithm would no longer be implicit and stability considerations would restrict the maximum marching step size. The dependent variables are extrapolated from stations \( j \) and \( j-1 \) to evaluate the inviscid Jacobians at station \( j+1/2 \).

Linearizing the viscous "delta" terms is somewhat more difficult. The additional complexity is apparent if the generic functional relationships for \( E^v \) and \( F^v \), given by

\[
E^v = E^v(U, \mu(U), \eta_x, \eta_y) \\
F^v = F^v(U, \mu(U), \eta_x, \eta_y)
\]

are examined. Taylor series expansions about point \( j \) for \( E^v_{j+1} \) and \( F^v_{j+1} \) are employed to linearize the viscous "delta" terms as follows:
The Jacobian matrices $\frac{\partial E' \partial v'}{\partial U}$ and $\frac{\partial F' \partial v'}{\partial U}$, given in Appendix A, are derived assuming that $\mu$ is locally independent of $U$. This is equivalent to the linearization of Steger (17) except that an assumption regarding the cross derivative terms is not necessary since these terms were neglected in the thin-layer approximation. Of particular importance here is the fact that both of these Jacobian matrices contain $\eta$ derivative operators that act on $\Delta U$ as well as $\eta$ derivative operators that do not. This substantially modifies the manner in which these terms are approximated using finite differences. It is fully consistent to neglect the additional terms on the right-hand side of equation 18 for a first-order accurate scheme. The additional terms, also given in Appendix A, relate the changes in $E' \text{ and } F'$ in the streamwise direction to the changes in $\mu$ and the grid metrics.
By utilizing the linearizations given in equations 17a,b and 18a,b, equation 15 can be written in terms of $\Delta U$. The linearized algorithm is given by

$$\xi_x \left( \frac{\partial E_i^*}{\partial U} \right)_{j+\theta/2} + \xi_y \left( \frac{\partial F_i^*}{\partial U} \right)_{j+\theta/2} \Delta U_j$$

$$+ \left( 1 - \frac{\theta}{3} \right) \Delta \xi \eta_x \left\{ \frac{\partial}{\partial \eta} \left( \frac{\partial E_i}{\partial U} \right)_{j} - \left( \frac{\partial E_{i'}}{\partial U} \right)_{j} - \left( \frac{\partial E_{i''}}{\partial U} \right)_{j} \right\} \Delta U_j$$

$$+ \left( 1 - \frac{\theta}{3} \right) \Delta \xi \eta_y \left\{ \frac{\partial}{\partial \eta} \left( \frac{\partial F_i}{\partial U} \right)_{j} - \left( \frac{\partial F_{i'}}{\partial U} \right)_{j} - \left( \frac{\partial F_{i''}}{\partial U} \right)_{j} \right\} \Delta U_j$$

$$= - \left( 1 - \frac{\theta}{3} \right) \Delta \xi \left\{ \xi_x \left( \frac{\partial E_i^*}{\partial \xi} \right)_{j+1} + \xi_y \left( \frac{\partial F_i^*}{\partial \xi} \right)_{j+1} \right\}$$

$$- \left( 1 - \frac{\theta}{3} \right) \Delta \xi \left\{ \eta_x \left( \frac{\partial E_i}{\partial \eta} \right)_{j} - \left( \frac{\partial E_{i'}}{\partial \eta} \right)_{j} + \eta_y \left( \frac{\partial F_i}{\partial \eta} \right)_{j} - \left( \frac{\partial F_{i'}}{\partial \eta} \right)_{j} \right\}$$

$$+ \frac{\theta}{3} \left\{ \xi_x \Delta E_i^*_{j-1} + \xi_y \Delta F_i^*_{j-1} \right\}$$

$$+ \left( 1 - \frac{\theta}{3} \right) \Delta \xi \left\{ \eta_x \left( \frac{\partial E_{i'}}{\partial \eta_x} \right)_{j} \Delta \eta_x j + \eta_y \left( \frac{\partial E_{i'}}{\partial \eta_y} \right)_{j} \Delta \eta_y j \right\}$$

$$+ \left( 1 - \frac{\theta}{3} \right) \Delta \xi \left\{ \eta_x \left( \frac{\partial F_{i'}}{\partial \eta_x} \right)_{j} \Delta \eta_x j + \eta_y \left( \frac{\partial F_{i'}}{\partial \eta_y} \right)_{j} \Delta \eta_y j \right\}$$

$$+ O[(1-\theta)\Delta \xi^2 + \Delta \xi^3]$$
recalling that the terms containing $\hat{E}_i$ and $\hat{F}_i$ are neglected in subsonic regions so that departure solutions can be avoided.

$\eta$ difference approximations

The $\eta$ derivatives appearing on both the left- and right-hand sides of equation 19 are approximated using second-order central differences. For example, the term $\partial / \partial \eta ((\partial \hat{F}_i / \partial U) \hat{A}U_j)$ that appears on the left-hand side of equation 19 is approximated to order $\Delta \eta^2$ as

$$\frac{\partial}{\partial \eta} \left( \left( \frac{\partial \hat{F}_i}{\partial U} \right) \hat{A}U_j \right) = \frac{1}{2\Delta \eta} \left( \left( \frac{\partial \hat{F}_i}{\partial U} \right)_{j,k+1} \hat{A}U_j + \left( \frac{\partial \hat{F}_i}{\partial U} \right)_{j,k-1} \hat{A}U_j \right)$$

where $\eta = (k-1)\Delta \eta$. Once again, the order of accuracy estimate is based on an equally spaced grid in computational space. This differencing is applied to the $\eta$ derivatives of $\Delta E_i$ as well as the appropriate terms on the right-hand side of equation 19. The viscous terms in equation 19 are also differenced using second-order central differences but with a somewhat different approach due to the presence of $\eta$ derivatives in $E v'$ and $F v'$. The viscous terms containing $\eta$ derivatives that act on $\Delta U$ are differenced as

$$\frac{\partial}{\partial \eta} (\alpha \partial / \partial \eta (\beta \Delta U)) \frac{k}{\Delta \eta} = \frac{\alpha \partial / \partial \eta (\beta \Delta U) \frac{k+1/2}{\Delta \eta} - (\alpha \partial / \partial \eta (\beta \Delta U) \frac{k-1/2}{\Delta \eta}}$$

where
\[
\frac{\partial}{\partial \eta} (n \Delta u)_{k+1/2} = \frac{(n \Delta u)_{k+1} - (n \Delta u)_k}{\Delta \eta}
\]
\[
\frac{\partial}{\partial \eta} (n \Delta u)_{k-1/2} = \frac{(n \Delta u)_k - (n \Delta u)_{k-1}}{\Delta \eta}
\]

and
\[
\frac{\alpha_{k+1/2}}{a} = \left[ \frac{\alpha_{k+1} + \alpha_k}{2} \right]
\]
\[
\frac{\alpha_{k-1/2}}{a} = \left[ \frac{\alpha_k + \alpha_{k-1}}{2} \right].
\]

The other form of the viscous terms, due to the coordinate transformation and the second-order linearization, is differenced as
\[
\frac{\partial}{\partial \eta} (n \Delta u \frac{\partial}{\partial \eta} (\beta))_k = \frac{(n \Delta u \frac{\partial}{\partial \eta} (\beta))_{k+1/2} - (n \Delta u \frac{\partial}{\partial \eta} (\beta))_{k-1/2}}{\Delta \eta}
\]

where
\[
\frac{\partial}{\partial \eta} (\beta)_{k+1/2} = \frac{(\beta)_{k+1} - (\beta)_k}{\Delta \eta}
\]
\[
\frac{\partial}{\partial \eta} (\beta)_{k-1/2} = \frac{(\beta)_k - (\beta)_{k-1}}{\Delta \eta}
\]

and
\[(a\Delta U)_{k+1/2} = (a_{k+1} + a_k)(\Delta U_{k+1} + \Delta U_k)/4\]

\[(a\Delta U)_{k-1/2} = (a_k + a_{k-1})(\Delta U_k + \Delta U_{k-1})/4.\]

If the differencing described above is applied to equation 19, a block tridiagonal system, given by

\[
A_{j,k+1}\Delta U_{j,k+1} + D_{j,k}\Delta U_{j,k} + B_{j,k-1}\Delta U_{j,k-1} = F_{j,k} \tag{20}
\]

where \(A, D,\) and \(B\) are 4x4 matrices and \(F\) is a 4x1 vector, must be inverted at each marching station. This system of simultaneous equations is solved using the standard block tridiagonal inversion algorithm.

**Initial conditions**

To initiate the space-marching algorithm, the initial values of the dependent variables, or initial conditions, must be specified at the inflow boundary. In the cases where leading edge effects are not significant, the initial conditions are determined from a similarity solution for the compressible boundary layer equations. The Dorodnitsyn-Howarth transformation described by Thompson (18) is employed, along with the isothermal plate assumption, to convert the compressible boundary layer equations into a system of two ordinary differential equations. The momentum equation is solved first using a standard fourth-order Runge-Kutta technique. Using the solution of the momentum equation, the energy equation is then solved using a second-order finite-
difference algorithm. A natural cubic spline is used to interpolate the similarity solution to the grid employed for the PNS calculation.

In a flow where leading edge effects are important, the similarity solution does not provide adequate starting data since the effects of the leading edge compression are neglected. Therefore, a different procedure is employed to obtain the initial data for these cases. The initial data are computed by beginning the PNS solution at a station very close to the leading edge of the plate using approximate starting data. At the wall, the no-slip condition gives the values of \( u \) and \( v \) and the temperature is specified. The results of the strong-interaction analysis of Bertram and Blackstock (19) are utilized to compute the wall pressure at the inflow boundary. Since \( p \) and \( T \) are known, \( \rho \) is computed from the equation of state. Second-order polynomials are used to define the variation of \( \rho \), \( u \), \( p \), and \( T \) from the wall values to the freestream values. These polynomials satisfy zero-slope conditions at the "freestream" that is typically chosen five to ten grid points into the field. The value of \( v \) is assumed to be zero across the field. The solution is advanced using small marching step sizes until the oscillations due to the arbitrary initial data have diminished. These results are used as initial data for the actual PNS calculation using a larger step size.
Boundary conditions

If the numerical algorithm described by equation 20 is applied at \( k = 2 \), the resulting block tridiagonal system is given by

\[
\alpha_j,2 \Delta U_j,3 + \beta_j,2 \Delta U_j,2 + \gamma_j,2 \Delta U_j,1 = F_j,2 .
\]  

(21)

The quantity \( \Delta U_{j,1} \), the value of \( \Delta U_j \) at the wall is needed to compute the solution in the interior of the domain at the new marching station \( j+1 \). The actual value of \( \Delta U_{j,1} \) is specified by the boundary conditions of the problem. In the viscous flow problem considered here, the boundary conditions are of two types: 1) Dirichlet type, which specifies the boundary values, and 2) Neumann type, which specifies a relationship between the boundary values and the solution in the interior of the domain. In the discussion that follows, an implicit second-order numerical boundary condition procedure, similar to that discussed by Beam and Warming (10), is developed that is consistent with the PNS approximation.

For flow past a compression corner, the \( \eta = \text{constant} \) curve corresponding to \( k = 1 \) is a no-slip boundary. In this case, the vector \( \Delta U_{j,1} \) is given by

\[
\Delta U_{j,1} = [\Delta p_{j,1}, 0, \Delta E_{\perp j,1}]^T
\]

(22)

since \( u = v = 0 \) is the no-slip condition that is applied for all values of \( j \). Since the remaining two boundary conditions
consist of assumptions regarding the normal pressure gradient and the wall temperature, the quantities $\Delta p_{j,1}$ and $\Delta E_{tj,1}$ must be expressed in terms of the appropriate variables. As before, Taylor series expansions about station $j$ are employed to determine the values of the dependent variables at the next marching station $j+1$. The Taylor series expansion for $\rho_{j+1,1}$ about station $j$ is given by

$$\rho_{j+1,1} = \rho_{j,1} + \left(\frac{\partial \rho}{\partial \xi}\right)_{j,1} \Delta \xi + O(\Delta \xi^2).$$

By differentiating the perfect gas equation of state with respect to $\xi$ and approximating the resulting partial derivatives with first-order forward differences, an equation relating $\Delta p_{j,1}$ to $\Delta p_{j,1}$ and $\Delta T_{j,1}$ is obtained that is given by

$$\Delta p_{j,1} = \rho_{j,1} \left(\frac{\Delta p_{j,1}}{\rho_{j,1}} - \frac{\Delta T_{j,1}}{T_{j,1}}\right) + O(\Delta \xi^2).$$

In all of the cases considered here, the wall is assumed to have a constant specified temperature so that the above equation becomes

$$\Delta p_{j,1} = \rho_{j,1} \left(\frac{\Delta p_{j,1}}{\rho_{j,1}}\right) + O(\Delta \xi^2) \quad (23)$$

since $\Delta T_{j,1} = 0$. The truncation error of equation 23 is consistent with either the first- or second-order scheme since all of the terms in $B_{j,2}$ in equation 21 are multiplied by $\Delta \xi$. Similarly, an equation for $\Delta E_{tj,k}$ is obtained from the definition of $E_t$ and is given by
\[ \Delta E_{t,j,k} = \frac{1}{\gamma - 1} \Delta p_{t,j,k} + \frac{1}{2} (u^2 + v^2) j,k \Delta p_{t,j,k} + u_j k \Delta (\rho u)_{j,k} \]

\[ + v_j k \Delta (\rho v)_{j,k} + O(\Delta \xi^2) . \]  

(24)

The particular case of interest here is \( k = 1 \) which corresponds to a no-slip boundary. Applying equation 24 at the wall yields

\[ \Delta E_{t,j,1} = \frac{1}{\gamma - 1} \Delta p_{t,j,1} \]  

(25)

where the no-slip condition has been applied.

The pressure boundary condition employed here is the assumption that the pressure gradient normal to the wall at station \( j+1 \) is identically zero. This can be expressed mathematically as

\[ \frac{\partial p}{\partial n} = \frac{\partial \xi}{\partial n} \frac{\partial p}{\partial \xi} + \frac{\partial \eta}{\partial n} \frac{\partial p}{\partial \eta} = 0 \text{ at } j+1 \]  

(26)

where \( n \) is the normal to the boundary at \( k = 1 \). To be consistent with the equations applied at the field points, the Vigneron splitting of the streamwise pressure gradient is applied to equation 26 to obtain

\[ \frac{\partial \xi}{\partial n} (\omega \frac{\partial p}{\partial \xi} + (1 - \omega) \frac{\partial \bar{p}}{\partial \xi}) + \frac{\partial \eta}{\partial n} \frac{\partial p}{\partial \eta} = 0 \text{ at } j+1 \]
where \( \frac{\partial p}{\partial \xi} \) represents the "hyperbolic" portion of the streamwise pressure gradient and \( \frac{\partial \hat{p}}{\partial \xi} \) represents the "elliptic" portion. It is clear from the definition of \( \omega \), equation 16, that at the wall \( \omega = 0 \) since \( u = v = 0 \). Therefore, to be consistent with the field points, no portion of \( \frac{\partial p}{\partial \xi} \) is included in the pressure boundary condition since \( \frac{\partial \hat{p}}{\partial \xi} \) (actually \( \frac{\partial \hat{u}}{\partial \xi} \) and \( \frac{\partial \hat{v}}{\partial \xi} \)) is not included at the points adjacent to the boundary. After algebraic manipulation, equation 26 can be written as

\[
\frac{\partial}{\partial \eta} \left( \frac{\Delta p_{j,1}}{} \right) = - \left( \frac{\partial p}{\partial \eta} \right)_{j,1}.
\]

It should be noted that although the term on the right-hand side is analytically identical to zero, this may not be the case numerically. By constructing the boundary condition in this manner, the effects of round-off error are minimized.

The \( \eta \) derivatives in the equation above are approximated using second-order forward differences and the following algebraic equation for \( \Delta p_{j,1} \) is obtained:

\[
\Delta p_{j,1} = \frac{1}{3} \left( 4\Delta p_{j,2} - \Delta p_{j,3} \right) + \frac{1}{3} \left( -3\Delta p_{j,1} + 4\Delta p_{j,2} - \Delta p_{j,3} \right) + O(\Delta \eta^2).
\]  

(27)

Using equations 22-27, an expression for \( \Delta U_{j,1} \) in terms of \( \Delta U_{j,2}, \Delta U_{j,3} \), and values of the dependent variables at station
j is developed and is given by

\[ \Delta U_{j,1} = \frac{4}{3} M_{j,2} \Delta U_{j,2} - \frac{1}{3} M_{j,3} \Delta U_{j,3} + \frac{1}{3} S p_j \]  

(28)

where \( M_{j,k} \), a 4x4 matrix, and \( S p_j \), a 4x1 vector, are given in Appendix C. The resulting difference approximation at \( k = 2 \) is given by

\[ (A_{j,2} - \frac{1}{3} B_{j,2} M_{j,3}) \Delta U_{j,3} + (D_{j,2} + \frac{4}{3} B_{j,2} M_{j,2}) \Delta U_{j,2} = F_{j,2} - \frac{1}{3} B_{j,2} S p_j \]  

(29)

After the solution has been obtained at the interior points, equation 28 is employed to determine \( \Delta U_{j,1} \) so that the boundary conditions are explicitly enforced at the wall.

Depending on the particular test case, two different types of boundary conditions are applied on the upper boundary, the \( \eta = \) constant curve corresponding to \( k = k_{\text{max}} \). Since the interaction between this boundary and the region of interest is insignificant provided the upper boundary is located sufficiently far from the surface of the plate, the boundary conditions are chosen to be as simple as possible. One of the cases discussed in the next section is the flow past a flat plate. Since these results are compared with the similarity solution, the boundary conditions applied on the
upper boundary are chosen to match the similarity solution as closely as possible. Dirichlet boundary conditions are applied on the upper boundary by specifying the values of \( \rho, u, p, \) and \( T \) to be the appropriate freestream values while \( v \) is allowed to vary in the manner prescribed by the similarity solution. In the other cases discussed, the boundary condition applied at the upper boundary is a simple zero gradient condition given by \( \partial U/\partial \eta = 0 \). If an approach similar to the one outlined above is followed and second-order backward differences are used to approximate the \( \eta \) derivatives, the following equation for \( \Delta U_{j,kmax} \) is obtained:

\[
\Delta U_{j,kmax} = \frac{1}{3} (4\Delta U_{j,kmax-1} - \Delta U_{j,kmax-2})
\]

(30)

\[
- \frac{1}{3} (-3U_{j,kmax} + 4U_{j,kmax-1} - U_{j,kmax-2})
\]

The resulting difference approximation is given by

\[
(D_{j,kmax-1} + \frac{4}{3} A_{j,kmax-1})\Delta U_{j,kmax-1} + (B_{j,kmax-1} - \frac{1}{3} A_{j,kmax-1})\Delta U_{j,kmax-2} = F_{j,kmax-1} + \frac{1}{3} A_{j,kmax-1}(3U_{j,kmax} - 4U_{j,kmax-1} + U_{j,kmax-2})
\]
As before, the boundary condition at the upper boundary is explicitly enforced using equation 30 after the solution at the interior points is determined.

Artificial viscosity

Typically, when implicit finite-difference algorithms employing central differences are applied to convective processes, some sort of artificial smoothing is utilized to attempt to control the spurious high frequency oscillations that are often present in the numerical solution. One way of suppressing these oscillations is to add fourth-difference explicit smoothing of the form given by Beam and Warming (10). A consistent way of applying smoothing of this form to the PNS equations was given in reference 15. A similar approach was employed here by adding the following term to the right-hand side of equation 19:

\[- \left(1 - \frac{6}{3} \right) \frac{\alpha \Delta \xi}{8} \left\{ \xi_x \left( \frac{\partial E_i^*}{\partial U} \right)_{j+\theta/2} + \xi_y \left( \frac{\partial F_i^*}{\partial U} \right)_{j+\theta/2} \right\} D_4 U_{j,k} \]

where \( D_4 U_{j,k} \) is the usual fourth-difference stencil

\[ D_4 U_{j,k} = U_{j,k+2} - 4U_{j,k+1} - 6U_{j,k} - 4U_{j,k-1} + U_{j,k-2} \]

and \( \alpha \) is the smoothing coefficient. According to Beam and Warming, the smoothing described in reference 10 is stable for \( 0 \leq \alpha \leq 1+\theta \). By direct analogy, the smoothing given above is stable for \( 0 \leq \alpha \Delta \xi \leq 1+\theta \). This form of smoothing was applied
at points located more than one data point from a boundary. However, some sort of modification is needed at points adjacent to boundaries since this stencil would require data exterior to the computational domain. To eliminate this problem and still maintain fourth-order accuracy, a modified difference stencil was employed as suggested by Pulliam (20). The stencils used at $k = 2$ and $k = k_{\text{max}} - 1$ are given by

\[ \tilde{D}_4 U_{j,2} = -2U_{j,1} + 5U_{j,2} - 4U_{j,3} + U_{j,4} \]

and

\[ \tilde{D}_4 U_{j,k_{\text{max}}-1} = U_{j,k_{\text{max}}-3} - 4U_{j,k_{\text{max}}-2} + 5U_{j,k_{\text{max}}-1} - 2U_{j,k_{\text{max}}} \]

respectively. According to Pulliam, these difference stencils are fully dissipative and have proved to be reliable and robust for inviscid calculations.

Grid Generation

It is well established that the accuracy of viscous flow calculations is dependent on adequately resolving the large transverse gradients present in the boundary layer. The spacing between the coordinate lines parallel to the wall must be sufficiently small in order to reduce the truncation error and improve accuracy in this critical region. Outside of the
boundary layer, the transverse gradients are, in general, significantly smaller so that a fine mesh is not required. Therefore, the most efficient method of distributing the discrete mesh points is to employ some sort of grid stretching so that coordinate lines are clustered near the wall in physical (x,y) space. Similarly, since the most significant streamwise gradients in the flow occur near the shock that occurs at the intersection of the flat plate and the ramp, it is also more efficient to utilize some form of grid stretching in the streamwise direction.

Due to the simplicity of the geometry under consideration in the present study, an analytical grid generation scheme is employed to define the transformation between computational (ξ,η) space and physical (x,y) space. Unfortunately, a constraint on the choice of method utilized to generate the discrete mesh occurs because significant errors are introduced in regions of large gradients if a highly stretched grid is employed. Of the functions discussed by Thompson et al. (21), the hyperbolic sine function generates the smallest error due to stretching in the region of minimum spacing, i.e., where the largest gradients are expected to occur.

With this in mind, the functions used to define the transformation from the equally-spaced computational (ξ,η) grid with 1 ≤ ξ ≤ ξmax, 1 ≤ η ≤ ηmax, and Δξ = Δη = 1, to the physical (x,y) grid are chosen to be
\[ x(\xi) = x_0 + (x_c - x_0) \left\{ 1 + \frac{\sinh(\beta x (\xi - \xi_0))}{\sinh(\beta x \xi_0)} \right\} \]  \hspace{1cm} (31)

and

\[ y(\xi, \eta) = y_1(\xi) + (y_u(\xi) - y_1(\xi)) \frac{\sinh(\beta y \tilde{\eta})}{\sinh(\beta y)} \]  \hspace{1cm} (32)

where

\[ \xi = \frac{\xi - 1}{\xi_{\text{max}} - 1} \]  

\[ \xi_0 = \frac{1}{2\beta x} \ln \left\{ \frac{1 + (e^{\beta x} - 1)(x_c - x_0)/(x_f - x_0)}{1 + (e^{-\beta x} - 1)(x_c - x_0)/(x_f - x_0)} \right\} \]  

and

\[ \tilde{\eta} = \frac{\eta - 1}{\eta_{\text{max}} - 1} \]  

The nomenclature employed above is illustrated in Figure 4. The clustering parameters, \( \beta x \) and \( \beta y \), range from zero to infinity with small values giving very little clustering and large values giving tight clustering. The grid segments on the upper boundary are specified by defining the angle of the segment with respect to the plate. Figure 5 shows the grid generated using equations 31 and 32 for a compression corner with \( \delta = 10^\circ \), \( \xi_{\text{max}} = 201 \), \( \eta_{\text{max}} = 51 \), \( \beta x = 2.5 \), and \( \beta y = 6.5 \).
Figure 4. Schematic of computational grid
Figure 5. Typical grid for 10° compression corner
Every other line is deleted for clarity. The grid shown in Figure 5 is typical of the grids employed during this study for the single-pass solutions. It is important to note that for this transformation, the lines of constant $\xi$ are vertical lines in physical space, i.e., $\xi_y = 0$. The terms having $\xi_y$ as a multiplier are included in the development of the PNS algorithm only for completeness.

Results

The single-pass PNS algorithm described in the previous section was employed to compute the flows past a family of compression corner configurations, $\delta = 0^\circ$, $5^\circ$, $7.5^\circ$, and $10^\circ$, for the conditions $M_\infty = 3.0$, $Re_\infty = 1.68 \times 10^6$, $T_\infty = 217$ K, and $T_{wall} = 607.6$ K. The reference length $L$ was chosen to be the distance from the leading edge of the plate to the corner, $x_c$ in Figure 4. Carter (22) has studied these cases extensively using an explicit algorithm to solve the full NS equations. Comparisons were made only with numerical results to avoid the uncertainties that typically arise if numerical results disagree with experimental data. Lawrence et al. (23) have previously shown that the PNS approximation produces good results for hypersonic flows with minimal elliptic influence. The combination of Mach number and Reynolds number given above was chosen to produce conditions where it was possible for a significant elliptic influence to be present. The effect of
neglecting the elliptic portion of the streamwise pressure gradient was evaluated as well as the errors introduced by the basic single-pass philosophy.

The first results reported here are for the flow past a flat plate without leading edge effects for the conditions given above. This solution was computed to provide a method of code verification by comparing the PNS results with the readily available similarity solution. The grid employed for this calculation had \( \xi_{\text{max}} = 91, ~ \eta_{\text{max}} = 51 \), and was clustered in the transverse direction only using \( \beta_y = 4.5 \). The values of the dependent variables given by the similarity solution at \( x = 0.2 \) were used as starting data and Dirichlet boundary conditions were applied on the upper boundary as discussed previously. The second-order scheme, \( \theta = 1 \) in equation 19, was used to compute this solution. Figures 6, 7, and 8 show comparisons between the PNS results and the boundary layer similarity solution at \( x = 2.0 \). The PNS results show excellent agreement with the similarity solution in the region near the wall. Small differences between the two solutions occur near the edge of the boundary layer in all three cases. These discrepancies can be adequately explained by considering the methods used to compute each solution. In particular, the fact that the PNS algorithm is something less than second-order accurate in physical space whereas the methods employed to compute the similarity solution provide fourth-order
Figure 6. Comparison of streamwise velocity profiles at $x = 2.0$
Figure 7. Comparison of transverse velocity profiles at $x = 2.0$
Figure 8. Comparison of temperature profiles at $x = 2.0$
solutions for u and v and second-order solutions for T provides an acceptable explanation for this behavior. In addition, the step size used to compute the similarity solution was a constant value approximately equal to the minimum spacing near the wall in the highly stretched transverse grid employed for the PNS calculation.

Having verified that the results from the PNS code show good agreement with the similarity solution for the flat plate, a series of runs were made to show the effects of increasing the ramp angle \( \delta \) from 0° to 10°. The grid employed for each of these cases had \( \xi_{\text{max}} = 201, \eta_{\text{max}} = 51 \), and utilized clustering in both the transverse direction and the streamwise direction with \( \beta_y = 6.5 \) and \( \beta_x = 2.5 \). In this case, the leading edge compression alters the pressures in the flow field a considerable distance downstream. Therefore, the procedure described previously to obtain a starting solution including the leading edge effects was employed. The actual PNS solution discussed here was initiated at \( x = 0.2 \) and was terminated at \( x = 2.0 \). Neumann boundary conditions were applied on the upper boundary as discussed previously. The first-order scheme, \( \theta = 0 \) in equation 19, was used to compute these solutions.

Figure 9 shows a plot of the streamwise variation of the wall pressure predicted by the PNS code for \( \delta = 0^\circ, 5^\circ, 7.5^\circ, \) and \( 10^\circ \). As expected, the influence of the corner is not
Figure 9. Streamwise variation of wall pressure predicted by PNS code
propagated upstream since the space-marching approach eliminates the streamwise elliptic behavior from the solution. The magnitude of this error is made apparent by comparing the results computed by the PNS code, Figure 9, with the results of Carter (22), Figure 10. While the flat plate pressure distributions do agree quite well, the severity of the error introduced by the space-marching approach increases as the ramp angle $\delta$ increases, i.e., as the magnitude of the upstream influence increases. Although this behavior is not surprising, the magnitude of this error is unacceptable even for ramp angles of 5° and 7.5° where no separation is present.

Figure 11 shows a plot of the streamwise variation of the wall friction coefficient predicted by the PNS code for $\delta = 0°, 5°, 7.5°, \text{ and } 10°$. The wall friction coefficient was computed using

$$C_f = \frac{2\mu}{Re_\infty} \left| v_\eta \right| \frac{\partial}{\partial \eta} \left( \frac{\eta_y u - \eta_x v}{\left| v_\eta \right|} \right)$$  \hspace{1cm} (33)$$

where

$$\left| v_\eta \right| = \sqrt{\eta_x^2 + \eta_y^2}.$$ The $\eta$ derivatives in equation 33 were approximated using second-order forward differences at the wall. As expected, the wall friction coefficient decreases at the corner due to the shock-induced thickening of the boundary layer. Also, as
Figure 10. Streamwise variation of wall pressure predicted by Carter (22)
Figure 11. Streamwise variation of wall friction coefficient predicted by PNS code
the shock strength increases, i.e., increasing $\delta$, the decrease in friction coefficient becomes more significant. However, a comparison between Figure 11 and the results of Carter (22), shown in Figure 12, shows that, except for the $\delta = 0^\circ$ case, the skin friction distributions predicted by the PNS code are grossly in error. Significantly, this error increases as the shock strength, or alternatively, the streamwise pressure gradient, increases. This is the discrepancy alluded to by Barnette (12) and Rakich (16). The cause of this behavior can best be understood by evaluating the streamwise momentum equation at the wall. For simplicity, assume that $x$ is the streamwise direction so that the streamwise momentum equation evaluated at the no-slip boundary is given by

$$\frac{\partial \hat{p}}{\partial x} = \frac{\partial}{\partial y} \left( \mu \frac{\partial u}{\partial y} \right). \quad (34)$$

The term on the right-hand side of equation 34 is the transverse derivative of the wall shear. Significantly, the term on the left-hand side of equation 34 is the streamwise pressure gradient which was neglected at the wall so that a consistent PNS algorithm was obtained. The balance that should occur between the derivative of the wall shear and the gradient is therefore eliminated. These results indicate that although the subsonic flow in the boundary layer is coupled with the outer supersonic flow through the normal momentum
Figure 12. Streamwise variation of wall friction coefficient predicted by Carter (22)
equation, this coupling is insufficient to overcome the local error caused by neglecting the streamwise pressure gradient. In addition, since the upstream influence was not simulated by the space-marching algorithm, the cumulative effect of the increasing pressure gradient on the energy of the flow is also eliminated. As a final note, the flat plate results given by Carter (22) are actually weak-interaction results taken from Hayes and Probstein (24).

Some problems were encountered when the second-order algorithm, equation 19 with $\theta = 1$, was used to compute the flow for the larger ramp angles. A significantly smaller marching step size was required near the corner to obtain a solution. If grid clustering comparable to that employed above, $\beta_x = 2.5$, was used, undershoots in the post shock pressure resulted in negative values for the pressure. A clustering of $\beta_x = 10$ was required to eliminate the streamwise oscillations at the shock for the second-order algorithm. However, it should be noted that the same behavior was observed in first-order solutions if no grid clustering was employed. The results computed using the second-order algorithm did not differ appreciably from the results shown in Figures 9 and 11 and are not shown.
Summary

At this point, it is appropriate to summarize the most important conclusions of the previous sections concerning the single-pass philosophy. Of particular significance is the fact that the so-called departure solution was shown to be the response of the numerical algorithm to an ill-posed problem. Specifically, examination of the amplification factor for the Euler implicit scheme applied to a model problem showed that the presence of complex eigenvalues in subsonic regions is the destabilizing mechanism. If the portion of the convective flux vector responsible for the ellipticity was neglected, an unconditionally stable space-marching algorithm was obtained. In addition, it was shown that the minimum step size constraint associated with PNS algorithms retaining all of the streamwise pressure gradient can be interpreted as defining which frequency components of the solution are amplified. It was determined that the only rational approach to developing a consistent PNS algorithm was to neglect the portion of the streamwise flux vector associated with the elliptic behavior since it is impossible to control the frequency content of the numerical solution except in the simplest cases. An implicit PNS algorithm was developed using the Vigneron approach to determine the portion of the streamwise pressure gradient responsible for hyperbolic behavior. The portion of the pressure gradient responsible for the elliptic behavior was
neglected. Based on comparisons between the PNS algorithm developed here and full NS solutions, the single-pass space-marching technique was determined not to be a valid approach when significant elliptic behavior is present in the actual flow field. In fact, aside from the terms that must be neglected to obtain a consistent algorithm, the single-pass philosophy itself must be questioned since it eliminates the possibility of upstream influence. Therefore, some form of iterative algorithm that allows upstream propagation of information must be employed to solve problems of the type considered here. The remainder of this dissertation focuses on the multiple-pass philosophy for solving the TLNS equations.
CHAPTER 4. MULTIPLE-PASS PHILOSOPHY

Global Pressure Iteration

The results presented in the previous chapter show that solutions of the PNS equations computed using a numerical algorithm based on a consistent single-pass philosophy are grossly in error for flows with virtually any upstream elliptic interaction. Some of the deficiency is caused by the neglect of the portion of the streamwise pressure gradient responsible for elliptic behavior in subsonic regions. However, the major portion of the error is introduced because the single-pass philosophy precludes simulation of the streamwise elliptic behavior. Therefore, for flows of this type, an algorithm that does not suppress the streamwise ellipticity must be employed if accurate results are desired. As discussed previously, the "elliptic" portion of the streamwise pressure gradient actually corresponds to wave propagation in time in opposition to the marching direction. This suggests using a forward difference to approximate this part of the streamwise pressure gradient so that upstream influence is introduced in the numerical algorithm. Unfortunately, a single-pass algorithm can no longer be employed since the forward difference on the streamwise pressure gradient utilizes data downstream of the current marching station. An alternative approach is to use multiple
passes and evaluate the forward differences using pressures computed during the previous iteration. This process is repeated producing a sequence of solutions that, hopefully, converges to the correct steady-state solution. This is precisely the strategy employed in the so-called global pressure iteration (GPI). In the following paragraphs, where several methods of differencing the streamwise pressure gradient so that upstream influence is introduced are discussed, the marching direction is \( x, x = (j-1)\Delta x \), and \( j+1 \) is the marching station being computed.

Lin and Rubin (25) used a GPI algorithm to compute the supersonic viscous flow past a cone at incidence. An iterative implicit algorithm was developed that employed a coupled strongly implicit scheme to solve the momentum and continuity equations and a strongly implicit algorithm to solve the energy equation. The streamwise pressure gradient was approximated using a first-order forward difference as

\[
\frac{\Delta p}{\Delta x} = \frac{p^n_{j+2} - p^n_{j+1}}{\Delta x}
\]  

(35)

where \( n+1 \) denotes the iteration level being computed. Although the motivation for using the GPI technique is to introduce upstream influence in the subsonic portion of the boundary layer, equation 35 was employed throughout the entire
flow field, apparently without any detrimental effects. Because of the iterative approach employed in the GPI algorithm, the effects of the pressure are propagated upstream only one grid point per iteration. Lin and Rubin reported departure free solutions regardless of the streamwise marching step size when equation 35 was used to approximate the streamwise pressure gradient. However, due to the problem under consideration, the solutions obtained using the GPI algorithm did not differ appreciably from solutions obtained using a single-pass algorithm that employed backward differencing for the streamwise pressure gradient.

Rakich (16) developed a GPI scheme using the Vigneron splitting and applied it to computing the supersonic viscous flows past a 1° compression corner and a 1° expansion. The Vigneron splitting, as described previously, was used to determine the fraction of the streamwise pressure gradient to be treated in a hyperbolic manner. The remainder is the portion responsible for the streamwise elliptic behavior. Formally, this can be written as

$$\frac{\partial p}{\partial x} = \omega \frac{\partial p}{\partial x_{\text{hyp}}} + (1 - \omega) \frac{\partial p}{\partial x_{\text{imp}}}.$$  (36)

The "hyperbolic" portion was approximated using a first-order backward difference as in the single-pass algorithm discussed previously while the "elliptic" portion was approximated using
the "semi-implicit" differencing of equation 35, hence the "imp" subscript. Note that as before, the effects of the pressure are propagated upstream one grid point per iteration. Unlike the GPI algorithm of Lin and Rubin, the forward differencing was used only in subsonic regions where \( \omega < 1 \). The results predicted using the GPI algorithm showed good agreement with results predicted using a time-dependent NS code although the GPI solutions showed slightly less upstream influence than the NS solutions. According to Rakich, numerical testing indicated that the discrepancy was not due to the neglect of the streamwise viscous terms. Rakich also stated that the stability of the iteration was apparently decreased if a second-order accurate algorithm was employed.

Barnett and Davis (26) developed a GPI algorithm based on the Vigneron splitting and used it to predict the supersonic flows past the compression corner described previously and past a flat plate with a parabolic hump. In the terminology of Barnett and Davis, the algorithm presented in reference 26 is a two-step alternating-direction explicit (ADE) scheme. Although differences exist in the manner that the transverse derivatives are evaluated, the first step of the ADE scheme is similar to the method of Rakich (16). However, the treatment of the streamwise pressure gradient in each case is formally quite different. In reference 26, a fictitious unsteady term was appended to the portion of the streamwise pressure
gradient responsible for elliptic behavior. The spatial
derivative in the "elliptic" portion was evaluated using
pressures from the previous iteration. According to reference
26, the pseudo-unsteady term is included so that information
can be propagated in the subsonic portion of the boundary
layer in a time-like manner. The resulting expression for the
streamwise pressure gradient is given by

$$\frac{\partial p}{\partial x} = \omega \frac{\partial p}{\partial x_{\text{hyp}}} + (1 - \omega) \left( \frac{\partial p}{\partial x_{\text{exp}}} - \frac{\partial p}{\partial t} \right)$$

(37)

where $\frac{\partial p}{\partial t}$ is the fictitious unsteady derivative. As before,
the "hyperbolic" portion of the streamwise pressure gradient
was approximated using a first-order backward difference. A
first-order backward difference was used to approximate the
time derivative and a first-order forward difference was used
to approximate the spatial derivative in the "elliptic"
portion. The "exp" subscript is used to emphasize that the
pressures utilized to form the spatial difference are values
from the previous iteration. The resulting difference
expression for the "elliptic" portion of the streamwise
pressure gradient is given by

$$\frac{\partial p}{\partial x_{\text{exp}}} \frac{\partial p}{\partial t} = \frac{p_{n+1} - p_{j+1}}{\Delta t} - \frac{p_{j+1} - p_{j+2}}{\Delta x}$$

(38)
where $\Delta t$ is an arbitrarily chosen parameter that Barnett and Davis selected so that $\Delta t/\Delta x = 1$. It is now apparent why the negative sign multiplying the fictitious time derivative is necessary. The right-hand side of equation 38 is essentially an explicit upwind scheme for the linear convection equation that appears on the left-hand side of equation 38. The negative sign is necessary for stability since a forward difference is used to approximate $\partial p/\partial x$. After the first step of the ADE scheme was complete, the second step was implemented using a modified form of equation 37 to march from the downstream boundary of the computational domain to the inflow boundary. According to reference 26, the convergence rate of the algorithm was accelerated in comparison with the other GPI algorithms since the changes in the pressures were propagated upstream more rapidly by the second step of the ADE scheme. The results presented in reference 26 showed good agreement with the results predicted by Carter (22) for the $10^\circ$ compression corner. These results also did not show the same extent of upstream interaction as the NS solutions and the size of the region of separated flow was underpredicted.

Schiff and Steger (11) also discuss what is referred to as a "global pressure iteration". It should be noted that the GPI algorithm presented in reference 11 in no way resembles the GPI algorithms discussed above. In the GPI algorithm of Schiff and Steger, the streamwise pressure gradient is
evaluated using pressures from the previous iteration level and a backward difference in conjunction with the sublayer approximation discussed previously. The motivation for the development of this scheme is entirely different than the other GPI algorithms discussed. According to reference 11, this approach was developed to provide a method for refining the mesh to improve the truncation accuracy of the solution. Schiff and Steger never discuss this GPI algorithm in the context of elliptic interaction. Clearly, this approach does not address the fact that the space-marching approach is ill posed if the streamwise pressure gradient is evaluated using a backward difference. Not surprisingly, reference 11 and reference 12 reported that this method diverged in fewer than 5 iterations.

**Generalized GPI algorithm**

The different representations of the streamwise pressure gradient employed in the GPI algorithms discussed above are, in fact, three members of a family of algorithms. This point can be illustrated by combining equations 36 and 37 into a single generalized expression given by

\[
\frac{\partial p}{\partial x} = \omega \frac{\partial p}{\partial x_{\text{hyp}}} + (1 - \omega) \left( 1 - \beta \right) \left( \frac{\partial p}{\partial x_{\text{exp}}} - \alpha \frac{\partial p}{\partial x} \right) \\
+ (1 - \omega) \beta \frac{\partial p}{\partial x_{\text{imp}}} 
\]

(39)
where \( \omega = 0, \alpha = 0, \) and \( \beta = 1 \) gives the differencing of Lin and Rubin, \( \alpha = 0, \) and \( \beta = 1 \) gives the differencing of Rakich, and \( \alpha = 1, \) and \( \beta = 0 \) gives the first step of the ADE scheme of Barnett and Davis. If the difference approximations discussed above are employed to discretize equation 39, the following difference expression in "delta" form is obtained:

\[
\frac{\partial p}{\partial x} \frac{\Delta x}{\Delta x} = (\omega - (1 - \omega)(\bar{\alpha} + \beta)) \Delta p_j^n + (1 - \omega)(\Delta p_{j+1}^n + (\bar{\alpha} + \beta) (p_{j+1}^n - p_j^n))
\]

where \( \bar{\alpha} = \alpha \Delta x / \Delta t. \) It is apparent then that \( \bar{\alpha} \) represents an inverse Courant number and that \( \alpha \) is related in some way to an inverse wave speed. Implementation of the GPI algorithm is discussed in detail when the second-order scheme is developed.

It is important to note that since \( \bar{\alpha} \) and \( \beta \) always appear together as the sum \( \bar{\alpha} + \beta, \) any schemes having the same values for this sum are equivalent. The differencing given by Rakich corresponds to \( \bar{\alpha} + \beta = 1. \) The first step of the ADE scheme of Barnett and Davis has \( \Delta t / \Delta x = 1 \) which gives \( \bar{\alpha} + \beta = 1. \) Therefore, the GPI algorithm of Rakich and the first step of the ADE scheme of Barnett and Davis are identical, at least in terms of the treatment of the streamwise pressure gradient.

The fact that these two schemes are equivalent is significant and is employed in the discussions that follow.
Stability

One point not discussed above is the stability of the GPI algorithm. According to Rubin (8), a stability analysis indicates that any iterative scheme that evaluates the streamwise pressure gradient using only pressures from a previous sweep is unstable. Numerical experiments using the algorithm given above verify this conclusion. GPI solutions diverged in as few as three iterations when the "elliptic" portion of the streamwise pressure gradient was evaluated in a purely explicit manner, i.e., $\alpha + \beta = 0$. The key point is apparently the "semi-implicit" treatment of the spatial derivative, or equivalently, adding the fictitious time derivative. This also verifies the results of reference 8 since Rubin states that the GPI scheme is stable if the evaluation of the streamwise pressure gradient is to some degree implicit.

A better understanding of the behavior of the algorithm given by equation 40 can be gained from a heuristic analysis of the method of representing the "elliptic" portion of the streamwise pressure gradient. For now, assume that $\beta = 0$ and $\omega = 0$ so that the resulting difference equation is given by

$$
\Delta x \frac{\partial p}{\partial x} = - \alpha \Delta p_j + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_j) + \Delta p_{j+1} + \alpha (p_{j+1} - p_
inverse wave speed $\alpha$ multiplying the fictitious time
derivative. As stated previously, the differencing scheme
employed for the right-hand side of the above equation is
essentially a first-order upwind scheme for a linear
convection equation. A Fourier stability analysis shows that
the differencing scheme is stable when applied to a linear
convection equation if $\bar{\alpha} = \alpha \Delta x / \Delta t \geq 1$, which is the standard
Courant number restriction since $\bar{\alpha}$ is the inverse Courant
number. In cases where $\bar{\alpha} < 1$, the solution is amplified and
the iteration diverges. It is not unreasonable to presume
that since the portion of the algorithm responsible for
introducing the upstream influence is unstable for the
propagation of linear waves, the stability characteristics of
the full GPI algorithm would be somewhat deficient. Although
the heuristic analysis presented above does not take into
account the complexity of the GPI scheme, it apparently
provides a viable explanation of the behavior of the
algorithm. Numerical experiments indicate that the actual
value of $\bar{\alpha}$ where instabilities begin is problem dependent but
in general is in the range $0.5 < \bar{\alpha} < 0.9$. Based on the above
considerations and the equivalency of the differencing
schemes, stability is apparently guaranteed for $(\bar{\alpha} + \beta) \geq 1$.
It is recognized that the results given here are based on a
very approximate analysis. There seems to be no question,
however, that the GPI algorithm is stable when the Courant
number restriction is satisfied. The important points to be made here are that the behavior of the full GPI algorithm is dependent on the approach used to introduce the upstream influence and that if this method is stable when applied to a linear convection problem, the full GPI algorithm will also be stable. These results will be used to assist in the development of a second-order accurate GPI algorithm.

**Second-order GPI algorithm**

As mentioned previously, Rakich (16) stated that the second-order GPI algorithm developed using three-point differences to approximate the $x$ derivatives in equation 36 appeared to be less stable than the first-order scheme. This point is not elaborated upon further in reference 16. An expression for the streamwise pressure gradient, analogous to equation 40, can be written for a differencing scheme that is first- or second-order accurate in space and is given by

$$
\frac{\theta}{3} \frac{\partial p}{\partial x} \bigg|_{n+1} = \omega \left( \frac{\Delta p_{j}}{3} - \frac{\Delta p_{j-1}}{3} \right) \\
- (1 - \omega) \left( \frac{\theta}{3} \bar{a} + \beta \right) \Delta p_{j} \bigg|_{n+1} + \frac{\theta}{3} \Delta p_{j+2} \\
+ (1 - \omega) \left( \Delta p_{j+1} + \left( \frac{\theta}{3} \bar{a} + \beta \right) \left( p_{j+1} - p_{j} \right) \right)
$$

where, as before, $\theta = 0$ corresponds to the first-order scheme.
and $\theta = 1$ corresponds to the second-order scheme. Different algorithms generated using equation 42 are equivalent if the sum $((1 - \theta/3)\tilde{a} + \beta)$ is the same for each of the schemes. The case under consideration, $\theta = 1$, $\omega = 0$, $\tilde{a} = 1$, and $\beta = 0$, corresponds to an explicit upwind scheme for a linear convection equation using first-order temporal differencing and second-order upwind spatial differencing. After some algebraic manipulation, the resulting finite-difference expression is given by

$$\frac{\partial p}{\partial x} = -\tilde{a}\Delta p_j - \frac{1}{2}\Delta p_{j+2}$$

$$+ \left(\frac{-\Delta p_{j+1} + \tilde{a}(p_{j+1} - p_j)}{2}\right).$$

Note that the effects of the pressure are transmitted upstream two grid points per iteration. A Fourier stability analysis shows that this differencing scheme is unstable when applied to a linear convection equation. When the second-order algorithm was used to predict the compression corner flow, oscillations developed in the wall pressure just ahead of the shock after a few iterations. These spurious oscillations typically required many iterations to dissipate. In some instances, these oscillations became so severe that the iteration diverged. These results verify the comment made by Rakich concerning the stability of the second-order algorithm.
The cause of the instability described above is made readily apparent by examining what Warming and Hyett (27) call the modified equation of the finite-difference scheme. The modified equation is the partial differential equation that is actually solved when a finite-difference algorithm is employed to compute the solution to a partial differential equation. The modified equation for the differencing scheme given above applied to a linear convection equation is given by

\[
\text{F.D.E.} = - \alpha \frac{\partial p}{\partial t} + \frac{\partial p}{\partial x} \frac{\Delta \tau}{\Delta x} - \frac{\partial^2 p}{\partial x^2} + O[\Delta x^2, \Delta x \Delta \tau, \Delta \tau^2] \tag{44}
\]

where F.D.E. is the right-hand side of equation 43 divided by \( \Delta x \). Since the coefficient of the time derivative term on the right-hand side of equation 44 has the same sign as the coefficient of the diffusive term \( \partial^2 p/\partial x^2 \), the effect of the diffusive term is to destabilize the algorithm. In regions where \( \partial^2 p/\partial x^2 \) is significant, for example near a shock, the destabilizing influence of the diffusive term can cause the algorithm to become unstable, producing the oscillations described previously.

The purpose here, though, is not to use the heuristic analysis to attempt to prove that the second-order GPI scheme is unstable but to construct a method to introduce the elliptic influence with second-order spatial accuracy that satisfies the sufficient condition for stability of the GPI
algorithm obtained from the heuristic analysis, i.e., stability for a linear convection problem. The modified equation for the second-order scheme suggests a method of accomplishing just this. Equation 44 can be rewritten as

\[
\text{F.D.E.} + \frac{\Delta t}{2a} \frac{\partial^2 p}{\partial x^2} = -\alpha \frac{\partial p}{\partial x} + \frac{\partial p}{\partial \tau} + O[\Delta x^2, \Delta x \Delta \tau, \Delta \tau^2] \tag{45}
\]

so that the destabilizing term is eliminated without affecting the spatial accuracy of the algorithm. The diffusive term is approximated using a first-order forward difference which is consistent with the overall truncation error since this term is multiplied by \(\Delta \tau\). It is important to realize that including the diffusive term is not just adding artificial damping. The differencing scheme is actually modified so that the correct partial differential equation is solved. The resulting difference expression analogous to equation 43 is given by

\[
\frac{\partial p}{\partial x} \bigg|_{n+1} = -\alpha \Delta p_j + \frac{1}{2} \Delta p_{j+2} + \frac{3}{2} \Delta p_{j+1} + \frac{\alpha}{2} (p_{j+1} - p_j) + \frac{1}{2\alpha} (p_{j+3} - 2p_{j+2} + p_{j+1}) .
\tag{46}
\]

It is interesting that this differencing scheme is actually the single-step version of the upwind scheme of Warming and Beam (28). Therefore, the resulting second-order algorithm
has the same stability constraint when applied to a linear convection problem which, for this case, is given by $\tilde{a} \geq 1/2$. Based on the heuristic stability argument, a GPI algorithm using this differencing scheme is stable for $\tilde{a} \geq 1/2$.

The additional term added to equation 46 to stabilize the second-order algorithm can be included in the generalized algorithm by noting that two schemes are equivalent if the sum $((1 - \theta/3)\tilde{a} + \beta)$ is the same for each of the schemes. The resulting generalized differencing is given by

$$
(1 - \frac{\theta}{3})\Delta x \frac{\partial p}{\partial x} = \omega(\Delta p_j - \frac{\Delta p_{j-1}}{3}) + \frac{1}{3} (\tilde{a} + \beta)(\Delta p_{j+1} + \Delta p_{j+2}) + (1 - \omega)((1 - \frac{\theta}{3})\tilde{a} + \beta) p_{j+1} - p_{j})
$$

$$
\frac{2\theta(1 - \omega)}{3(2\tilde{a} + 3\beta)} (p_{j+3} - 2p_{j+2} + p_{j+1})
$$

where $(\tilde{a} + \beta/(1 - \theta/3)) \geq 1/(1 + \theta)$ for stability.

**Implementation of GPI algorithm**

The differencing scheme given above is now incorporated into the single-pass algorithm discussed previously. First, $x$ is formally replaced by $\xi$ in equation 47. The terms on the
right-hand side of equation 47 that are multiplied by \( \omega \) represent the hyperbolic contribution of the pressure gradient and are included in \( E_i^* \) and \( F_i^* \). The terms that are multiplied by \( (1 - \omega) \) form the elements of \( \frac{\partial E_i}{\partial \xi} \) and \( \frac{\partial F_i}{\partial \xi} \) on the right-hand side of equation 19. It should be noted, however, that the term

\[
- (1 - \omega)(1 - \frac{\theta}{3})\tilde{a} + \beta))\Delta p_j
\]

appearing in equation 47 is treated implicitly since \( \Delta p_j \) is evaluated at level \( n+1 \). By combining equation 47 with the generalized differencing of Beam and Warming (10) employed in the single-pass algorithm, the streamwise differencing for \( E_{ij} \) employed in the GPI scheme is obtained:

\[
\Delta E_{ij}^{n+1} - ((1 - \frac{\theta}{3})\tilde{a} + \beta)\Delta E_{ij} \quad (1 - \frac{\theta}{3})\Delta \xi \left( \frac{\partial E_i}{\partial \xi} \right)_{j+1}^{n+1}
\]

\[
\frac{\theta}{3} (\Delta E_{ij}^{n+1} + \Delta E_{ij}^{n+1})
\]

\[
- \Delta E_{ij+1} - ((1 - \frac{\theta}{3})\tilde{a} + \beta)(E_{ij+1} - E_{ij})
\]

\[
- \frac{2\theta}{3(2\tilde{a} + 3\beta)} (E_{ij+3} - 2E_{ij+2} + E_{ij+1})
\]

\[\text{(48)}\]
Likewise, a similar equation for $F_{ij}$ is obtained and is given by

\[
\Delta F_{ij}^{n+1} = ((1 - \frac{\theta}{3})\Delta V + \beta)\Delta F_{ij}^n = (1 - \frac{\theta}{3}) \Delta V \left( \frac{\Delta F_{ij}^n}{\Delta x} \right)_{j+1} \\
+ \frac{\theta}{3} (\Delta F_{ij-1}^{n+1} + \Delta F_{ij+2}^n)
\]

(49)

\[
\Delta F_{ij+1}^{n+1} = ((1 - \frac{\theta}{3})\Delta V + \beta)(F_{ij+1}^n - F_{ij}^n) \\
- \frac{2\theta}{3(2\alpha + 3\beta)} (F_{ij+3}^n - 2F_{ij+2}^n + F_{ij+1}^n).
\]

The equation corresponding to equation 15 for the single-pass algorithm is obtained by adding equation 48 multiplied by $\xi_x$ to equation 49 multiplied by $\xi_y$. The remainder of the development of the GPI algorithm follows the development of the single-pass algorithm outlined previously.

As before, the Taylor series approach is employed to linearize the implicit "delta" terms. The linearizations given by equations 17a,b and 18a,b are formally identical for both the single-pass and multiple-pass algorithms. The only difference is in the approach used to compute the values of the dependent variables at station $j+1/2$ used in equation 17a. In the GPI algorithm, these values are obtained by averaging
the values at station \( j \), iteration \( n+1 \) and station \( j+1 \), iteration \( n \) as shown in the following:

\[
u_{j+1/2}^{n+1} = \frac{1}{2} (u_j^{n+1} + u_{j+1}^n) + O(\Delta \tau, \Delta \xi^2) .
\]

This averaging procedure is used in an attempt to speed the convergence of the iterative scheme by introducing upstream influence via the Jacobians. In practice, this averaging provided minimal improvement in the rate of convergence versus the extrapolation employed in the single-pass algorithm. The linearizations of the implicit terms containing \( \bar{E_i} \) and \( \bar{F_i} \) were accomplished using a Taylor series expansion about station \( j \) and are given by

\[
\begin{align*}
\bar{E_i}_{j+1}^{n+1} &= \bar{E_i}_j^n + \left( \frac{\partial \bar{E_i}}{\partial U} \right)_{j+\theta/2}^{n+1} \Delta U_j^n + O(\Delta \tau, (1-\theta)\Delta \xi^2 + \Delta \xi^3) \\
\bar{F_i}_{j+1}^{n+1} &= \bar{F_i}_j^n + \left( \frac{\partial \bar{F_i}}{\partial U} \right)_{j+\theta/2}^{n+1} \Delta U_j^n + O(\Delta \tau, (1-\theta)\Delta \xi^2 + \Delta \xi^3)
\end{align*}
\]

(50)

where the Jacobians \( \partial \bar{E_i}/\partial U \) and \( \partial \bar{F_i}/\partial U \) are given in Appendix A. The averaging described above is also used to evaluate the Jacobians at \( j+1/2 \) in equation 50.
GPI boundary conditions

Since a forward difference is employed for the portion of the streamwise pressure gradient responsible for upstream influence, the multiple-pass GPI algorithm requires a boundary condition at the downstream boundary. This can be contrasted with the single-pass algorithm which did not allow a downstream boundary condition. The boundary condition used at the downstream boundary is \( \frac{\partial p}{\partial \xi} = 0 \), a simple zero-gradient condition. The boundary condition is implemented in the GPI algorithm by setting the pressure at \( j = j_{\text{max}}+1 \) and \( j = j_{\text{max}}+2 \) to be equal to the pressure at \( j = j_{\text{max}} \) computed during the previous iteration. It should be noted that this boundary condition procedure is roughly equivalent to employing the single-pass algorithm at the downstream boundary. Therefore, this boundary should be located in a region of very weak upstream influence. Boundary conditions on \( u, v, \) and \( \rho \) or \( T \) are not needed at the downstream boundary.

The numerical boundary condition procedure employed for the wall pressure is also modified for the GPI algorithm. As discussed previously, the wall pressure boundary condition employed here is the assumption that the pressure gradient normal to the wall is identically zero. When the boundary conditions were developed for the single-pass algorithm, the streamwise pressure gradient appearing in equation 26 was neglected so that the numerical boundary condition...
procedure would be consistent with the finite-difference algorithm applied at the field points. Similarly, in the multiple-pass algorithm, the portion of the streamwise pressure gradient responsible for upstream elliptic interaction should be included in a manner consistent with the GPI algorithm. The other boundary conditions at the wall are unchanged from the single-pass algorithm.

The wall pressure boundary condition employed in the second-order GPI algorithm is developed following basically the same procedure as that used for the single-pass algorithm. For an \( n = \text{constant} \) wall, the zero normal pressure gradient condition is given by equation 26 that is repeated here for convenience:

\[
\frac{\partial p}{\partial \hat{n}} = \frac{\partial \xi}{\partial \hat{n}} \frac{\partial p}{\partial \xi} + \frac{\partial \eta}{\partial \hat{n}} \frac{\partial p}{\partial \eta} = 0 \text{ at } j+1 \tag{51}
\]

where \( \hat{n} \) is the normal to the boundary at the wall and

\[
\frac{\partial \xi}{\partial \hat{n}} = \frac{\xi_x \eta_x + \xi_y \eta_y}{|\eta|}
\]

\[
\frac{\partial \eta}{\partial \hat{n}} = \frac{\xi_x^2 + \xi_y^2}{|\eta|}.
\]

To be consistent with the GPI algorithm applied at the field points, an expression for \( \partial p/\partial \xi \) is obtained from equation 47 and substituted into equation 51. After employing second-
order three-point forward differences to approximate the $\eta$ derivative and considerable algebraic manipulation, the following expression is obtained:

$$\Delta p_{j,1} = G_1 \xi_j + G_2 (4\Delta p_{j,2} - \Delta p_{j,3})$$  \hspace{1cm} (52)

where

$$G_1 = \frac{1}{((1 - \theta/3)\bar{a} + \beta)(\xi_x^2 + \xi_y^2) + 1.5(\eta_x^2 + \eta_y^2)}$$

$$G_2 = \frac{(\eta_x^2 + \eta_y^2)/2}{((1 - \theta/3)\bar{a} + \beta)(\xi_x^2 + \xi_y^2) + 1.5(\eta_x^2 + \eta_y^2)}$$

$$\xi_j = \frac{\xi_x^2 + \xi_y^2}{(1 - \theta/3)} \sigma_j + \frac{\eta_x^2 + \eta_y^2}{2} (-3p_{j,1} + 4p_{j,2} - p_{j,3})$$

and

$$\sigma_j = (\Delta p_{j+1} - \frac{\theta}{3} \Delta p_{j+2} + ((1 - \frac{\theta}{3})\bar{a} + \beta)(p_{j+1} - p_j))$$

$$+ \frac{2\theta}{3(2\bar{a} + 3\beta)} (p_{j+3} - 2p_{j+2} + p_{j+1})$$

Equation 52 is the multiple-pass equivalent to equation 27 for the single-pass algorithm. Finally, utilizing equations 22-25 and equation 52, the multiple-pass equation corresponding to
equation 28 for the single-pass scheme is obtained and is given by

\[ \Delta U_{j,1}^{n+1} = 4G_{2j,2}^{n+1} \Delta U_{j,2}^{n} - G_{2j,3}^{n+1} \Delta U_{j,3}^{n} + G_{1} \Sigma P_{j} \]  

(53)

where \( M_{j,k} \) is the same 4x4 matrix used for the single-pass algorithm, and \( \Sigma P_{j} \) is a 4x1 vector. The forms of each are given in Appendix C. The remainder of the implementation is identical to the single-pass algorithm except that equation 53 is used rather than equation 29.

The boundary conditions on the upper boundary are identical for both the GPI and the single-pass algorithms. The upstream boundary conditions correspond to the initial data in the single-pass approach and are held fixed during the computation. Since a time-marching algorithm is being used, the initial values of the dependent variables must also be specified on the interior of the computational domain. The initial data are obtained from the appropriate single-pass solution described previously.

**Relationship to upwind relaxation methods**

The multiple-pass GPI scheme described above can be viewed as a pseudo time-marching algorithm using a splitting technique for the streamwise convective flux vector. Rather than using a standard plus/minus eigenvalue splitting, the Vigneron splitting is employed. As shown previously, the net
result is that that streamwise convective flux vector is divided into two terms: a portion that corresponds to wave propagation in the streamwise direction and a portion that corresponds to wave propagation in opposition to the streamwise direction. Because of the construction of the algorithm, disturbances are propagated with infinite speed in the marching direction and only one (or two) grid points per iteration in the upstream direction. It should be emphasized that in the GPI scheme, characteristic differencing is not employed for the transverse derivatives.

The computational strategy employed in the GPI algorithm is similar to the approaches employed by Lombard et al. (2) and Stookesbury and Tannehill (34) for solving the TLNS equations and by Chakravarthy (29) for solving the Euler equations. The basic premise presented in references 2 and 29 is that efficient relaxation methods can be developed by utilizing upwind difference approximations. Chakravarthy contends that the effectiveness of these schemes is apparently related to the diagonal dominance that occurs naturally for iterative schemes based on characteristic differencing. He also states that second-order relaxation schemes developed using simple three-point differences like those employed in the GPI algorithm will not be successful since diagonal dominance is not preserved. It is not clear whether Chakravarthy is referring to rate of convergence or the
robustness of the algorithm. Lombard et al. also note that relaxation with the second-order method using three-point differences converges less rapidly and is less robust than the first-order method. It should be noted that the heuristic stability analysis for the GPI algorithm proposes a possible explanation for the apparent lack of robustness of the second-order schemes but does not address the rate of convergence.

In an attempt to enhance the diagonal dominance of the GPI algorithm, an unsteady term is appended to the implicit portion of the scheme and is approximated at level \( n+1 \) using a first-order backward difference. Since the implicit algorithm employed actually solves for \( \Delta U_j = U_{j+1} - U_j \), algebraic manipulation of the difference expression is necessary to obtain the appropriate formulation which is given by

\[
(1 - \frac{\Delta \xi}{3 \Delta t}) \frac{\Delta U_j}{\Delta t} + (\text{LHS}_{\text{GPI}}) \Delta U_j
\]

\[
= (1 - \frac{\Delta \xi}{3 \Delta t}) (U_{j+1} - U_j) + \text{RHS}_{\text{GPI}}
\]

(54)

where \( \Delta t \) in equation 54 has no relation to \( \Delta t \) discussed previously. Since the time step size always appears in the denominator in equation 54, it is clear that the basic GPI algorithm without the unsteady term is actually equation 54 with \( 1/\Delta t = 0 \) or equivalently, \( \Delta t = \infty \).
Global Pressure Iteration Results

The GPI algorithm described above was used to predict the flows past the 5° and 7.5° compression corner configurations with \( M_\infty = 3.0, \, Re_\infty = 1.68 \times 10^6, \, T_\infty = 217 \, \text{K}, \) and \( T_{\text{wall}} = 607.6 \, \text{K} \) that were computed previously using the single-pass algorithm. The results of Carter (22) indicate that incipient flow separation is occurring at the intersection of the flat plate and the ramp for the 7.5° case. Since GPI results typically show less upstream influence than full NS results, it was anticipated that the predicted results for the 7.5° ramp would not show any separation so that the algorithm discussed above could be used. The treatment of separated flows is detailed in the next section. The grid employed for each of the computations was generated using the analytical functions discussed previously with \( \xi_{\text{max}} = 61 \) and \( \eta_{\text{max}} = 51 \). Clustering was employed in both the streamwise direction and transverse direction with \( \beta_x = 2.5 \) and \( \beta_y = 6.5 \) respectively. Neumann boundary conditions were applied on the upper boundary and the initial conditions for each case were obtained from the corresponding first-order single-pass solution reported previously. In the discussion below, GPI predictions are compared with other results. In addition, the behavior of the numerical algorithm is also discussed. As a final note, the scales of the axes in the following figures were chosen to match those used in Figures 9-12.
Figure 13 shows a comparison between the streamwise variation of the wall pressure for the 5° compression corner predicted using the second-order GPI algorithm, the first-order single-pass results, and the second-order NS calculation of Carter (22). The agreement between the GPI results and the results of Carter is very good. As expected, the GPI results show somewhat less upstream influence than the full NS results. Two manifestations of the decreased upstream influence present in the GPI results are the somewhat steeper shock and the decreased viscous/interaction downstream of the corner. The slight pressure gradient in the GPI results just downstream of the inflow boundary is due to the inflow boundary conditions remaining fixed throughout the calculation. Figure 14 shows a contour plot of the pressures predicted for the 5° compression corner using the first-order single-pass algorithm. Each contour level represents a change in pressure of 0.0025 which corresponds to approximately 3 percent of the freestream pressure. Figure 14 clearly shows the effect of suppressing the streamwise elliptic behavior present in the TLNS equations. The compression that occurs because the flow is turned by the ramp is not initiated until the flow impinges on the ramp. This is in sharp contrast to the results predicted using the multiple-pass GPI scheme shown in Figure 15. In Figure 15, the influence of the corner is clearly propagated upstream in the subsonic portion of the
Figure 13. Streamwise distribution of wall pressure for the 5° compression corner
Figure 14. Pressure contours predicted using single-pass algorithm for the 5° compression corner
Figure 15. Pressure contours predicted using GPI algorithm for the 5° compression corner
boundary layer and a more gradual compression occurs. It is important to note the degree that the outer supersonic flow is also affected through the viscous/inviscid interaction. In both cases, the remnants of the leading-edge shock are visible upstream of the corner. Comparing the results predicted using the single-pass algorithm with the GPI results once again emphasizes the inadequacy of the single-pass approach for problems of this type.

Figure 16 shows a comparison between the streamwise variation of the wall friction coefficient for the 5° compression corner predicted using the second-order GPI algorithm, the first-order single-pass results, and the second-order NS calculation of reference 22. Once again, the agreement between the results predicted using the GPI algorithm and the full NS results is very good. As before, the GPI predictions show slightly less upstream influence than the NS results. The overall effect of the smaller extent of upstream influence is that the boundary layer ahead of the corner is not sufficiently thickened. Consequently, the decrease in wall friction coefficient is somewhat lessened. The single-pass results show only a very small decrease in friction coefficient. This once again emphasizes the fact that while the gross effects of the flow field are adequately predicted using a single-pass algorithm, the details of the flow, such as wall friction and heat transfer, may be severely
Figure 16. Streamwise distribution of wall friction coefficient for the 5° compression corner
in error. First-order solutions were also computed for the 5° compression corner but are not shown here. In general, the first-order results showed significantly less upstream influence than the second-order results and a correspondingly smaller decrease in the wall shear.

In all cases considered during this study, the parameter employed to measure the degree of convergence of the solution was the maximum change in the wall pressure divided by the old value of the wall pressure. The convergence tolerance was selected to be 0.00001 which corresponds to a 0.001 percent change in the wall pressure. Figure 17 shows a comparison of convergence histories for the GPI algorithm applied to the 5° compression corner with \( \alpha = 0 \) and \( \beta = 1 \) for different values of the time-step size \( \Delta t \) in equation 53. Figure 17 clearly indicates that including the unsteady term does indeed improve the rate of convergence of the GPI scheme. It should be noted that no attempt was made to determine an optimum value of \( \Delta t \) since the results obtained during this study indicate that it is problem dependent. Solutions obtained using the first-order scheme typically required on the order of 30 to 40 percent fewer iterations than the second-order solutions. This is not surprising since the first-order solution also showed significantly less upstream influence.

Figure 18 shows a comparison of the convergence histories for the second-order GPI scheme with \( \alpha = 0, \beta = 1, \)
Figure 17. GPI convergence histories for the 5° compression corner
Figure 18. GPI convergence histories for the 5° compression corner with and without the diffusive term.
and $1/\Delta t = 5$ applied to the $5^\circ$ compression corner with and without the added diffusive term. Transient oscillations occurred in the streamwise variation of the wall pressure when the diffusive term was removed. The relaxation scheme expended most of the computational effort smoothing these oscillations. Although not shown here, the steady-state solutions for both cases are virtually identical. These results indicate that the apparent lack of robustness of the basic second-order algorithm, given by equation 42, is due to two fundamentally different reasons: 1) the lack of diagonal dominance because of three-point differencing and 2) the basic instability of the method of introducing upstream influence. Adding the unsteady term obviously does enhance the diagonal dominance of the algorithm but does not, in and of itself, stabilize the algorithm. Modifying the differencing scheme employed to introduce the upstream influence is necessary to obtain a robust and reliable algorithm.

At this point, it is appropriate to briefly summarize some of attempts made to improve the convergence rate of the algorithm. One modification investigated was to include the unsteady term in equation 53 only in the subsonic region. This approach was not as effective as including the unsteady term everywhere. Using different values of $\bar{a}$ and $\beta$ in the GPI algorithm was also considered. Since the terms containing $\bar{a}$ and $\beta$ are included only in the subsonic region, this scheme is
actually a variation of the approach discussed above and did not prove profitable. In a different vein, several methods for propagating the effects of the pressures back upstream were also investigated. These approaches could all be described using the general classification auxiliary pressure equation (APE) schemes. The fundamental idea behind the APE approach is to derive an auxiliary equation for the pressure in the subsonic region and to use this relation to modify the pressures predicted during each iteration of the GPI scheme in a manner that accelerates the upstream wave propagation speed. The ADE scheme of Barnett and Davis (26) can be thought of as an APE scheme. The approach employed here was to solve a Poisson equation for pressure valid in the incompressible region of the flow so that the elliptic coupling in the subsonic region was enhanced. Numerous variations of this scheme were considered, but none improved the performance of the basic algorithm. The underlying problem associated with this approach was that the resulting pressure distribution was based on "old" velocity and density data that, in turn, were based on the "old" pressure distribution. Actually, the methods developed using this approach were nothing more than fairly sophisticated pressure "smoothers".

One point concerning the numerical aspects of the solution not discussed previously is the form of smoothing employed in the GPI algorithm. Several different types were
tested, including the standard fourth-difference smoothing of U given in reference 10. The most effective smoothing was found to be the fourth-difference smoothing given previously for the single-pass algorithm with one difference: the Jacobians of the terms responsible for the upstream influence that are treated implicitly, $\partial E_i/\partial U$ and $\partial F_i/\partial U$, were included in the matrix scaling $D_4 U$. The resulting form of the smoothing term is given by

$$-\left(1 - \frac{\theta}{3}\right) \frac{\alpha \Delta \xi}{\beta} \left\{ \xi_x \left( \frac{\partial E_i^*}{\partial U} \right)_{j+\theta/2} + \xi_y \left( \frac{\partial F_i^*}{\partial U} \right)_{j+\theta/2} \right\} D_4 U_{j,k}.$$ 

It is particularly interesting that this is the form of smoothing appropriate for a space-marching application. Although strictly speaking, the GPI algorithm is a time-marching algorithm, it still retains substantial space-marching behavior.

Figure 19 shows a comparison between the streamwise variation of the wall pressure for the 7.5° compression corner predicted using the second-order GPI algorithm, the first-order single-pass results, and the second-order NS calculation of reference 22. The GPI results show good agreement with the NS results with slightly less upstream influence for $x < 1.2$. Unfortunately, downstream of $x = 1.2$ the pressure predicted by
Figure 19. Streamwise distribution of wall pressure for the 7.5° compression corner
the GPI scheme is considerably higher than the full NS results. One possible cause for this anomalous behavior is the downstream boundary condition applied on the streamwise pressure gradient. However, this seems unlikely since the downstream pressure is adequately predicted for both the 5° compression corner discussed previously and the 10° compression corner to be discussed later. The cause of this discrepancy is as yet undetermined. Figures 20 and 21 show the pressure contours predicted using the single-pass algorithm and GPI algorithm respectively. As before, each contour level represents a change in the pressure of 0.0025. The single-pass results once again show a very steep shock occurring at the corner. The GPI results show a very gradual compression with the influence of the corner propagated a significant distance upstream.

Figure 22 shows a comparison of the streamwise variation of the wall friction coefficient predicted using the GPI scheme, the single-pass results, and the NS calculations of reference 22. There is very good agreement between the GPI results and the full NS results with the GPI results showing slightly less upstream influence causing the decrease in wall friction coefficient at the corner to be underpredicted. As expected, the GPI results do not indicate the onset of separation at the corner while the NS predictions show a small region of separated flow.
Figure 20. Pressure contours predicted using single-pass algorithm for the 7.5° compression corner
Figure 21. Pressure contours predicted using GPI algorithm for the 7.5° compression corner
Figure 22. Streamwise distribution of wall friction coefficient for the 7.5° compression corner
Figure 23 shows the convergence history for the 7.5° compression corner solution with $\alpha = 0$ and $\beta = 1$ for different values of $\Delta t$. Of particular interest here is the case corresponding to $\Delta t = \infty$. Very early in the iteration, a large region of separated flow developed at the corner so that the techniques described in the next section for the treatment of separated flow were needed. Much of the computational effort expended to obtain this solution was used to eliminate this anomalous region of separated flow. In the cases where the unsteady term was included, no flow separation occurred. The effects of including the unsteady term can be ascertained from these results. The numerical solution with $\Delta t = \infty$ "overshoots" the correct result and a region of separated flow develops. Adding the unsteady term essentially "stiffens" the system by enhancing the diagonal dominance so that the solution does not overshoot the correct result and the large region of separated flow does not develop.

Treatment of Separated Flows

The iterative approach discussed in the previous sections appears adequate for predicting supersonic flows with no streamwise separation. When streamwise separation is present, the component of the velocity in the marching direction is negative in the separated region. As seen from the eigenvalues of the streamwise convective flux vector given in
Figure 23. GPI convergence histories for the 7.5° compression corner
equation 5, the reversed flow velocities correspond to wave propagation in the direction opposite the marching direction. This is precisely the same situation that occurred in subsonic regions due to the presence of the streamwise pressure gradient. Therefore, it is anticipated that the solution procedure employed above would be unstable in separated regions since a backward difference was utilized to approximate the streamwise convective flux vector. The eigenvalue structure suggests using a forward difference to approximate the terms that are associated with the propagation of waves opposite to the marching direction. It is apparent then that the primary consideration in the numerical treatment of the streamwise separation problem is the method of determining which components of the convective flux vector are responsible for the "negative" time-like behavior. The two different strategies that were considered during this study for differencing the streamwise convective flux vector are discussed below.

The simplest approach for approximating the streamwise convective flux vector considered in this study, and the least satisfying from a theoretical standpoint, utilized the so-called FLARE approximation and is similar to the approach employed by Barnett and Davis (26). The FLARE approximation, developed for the boundary layer equations by Reyhner and Flügge-Lotz (30), is based on the observation that the
magnitude of the velocity in regions of reversed flow typically tends to be much smaller than the freestream velocity. In the boundary layer equations, the FLARE approximation is implemented by replacing the term $u_3u/3x$ with $C|u|\partial u/\partial x$ in separated regions where $C$ is an arbitrary constant between 0 and 1. The FLARE approximation is implemented in the GPI algorithm in regions of separated flow by replacing the streamwise convection terms of the form $u_3\partial E / \partial x$ with $|u|\partial E / \partial x$. This corresponds to replacing $u$ with $|u|$ in the diagonal elements of the Jacobian matrices $\partial E_i^*/\partial U$, $\partial F_i^*/\partial U$, $\partial E_i/\partial U$, and $\partial F_i/\partial U$. It is important to note that this substitution is used for each occurrence of $E_i^*$, $F_i^*$, $E_i$, or $F_i$ in a streamwise difference.

An additional complication is introduced by the streamwise pressure gradient. As discussed previously, the streamwise pressure gradient is responsible for introducing upstream influence, i.e., wave propagation in opposition to the local flow direction. In separated regions, the local streamwise direction is in the negative $x$ direction. The streamwise pressure gradient is responsible for wave propagation in the positive $x$ direction and should be approximated using a backward difference. If a backward difference is employed to approximate the streamwise pressure gradient in conjunction with the FLARE approximation, no mechanism exits for introducing wave propagation in the
negative x direction. In fact, the FLARE approximation should be unstable if a backward difference is employed for the streamwise pressure gradient. Based on these considerations, the streamwise pressure gradient was evaluated using a forward difference in separated regions.

The FLARE approximation is obviously very simple to implement. However, the overwhelming disadvantage associated with this approach is the fact that although the resulting algorithm is stable, it does not accurately model the physical convection processes that are occurring. As stated previously, an eigenvalue analysis suggests using a forward difference to approximate those terms responsible for the "negative" time-like behavior. Fortunately, because of the advent of numerical algorithms based on characteristic differencing schemes, a plethora of methods are available to determine the differencing of the streamwise convective flux vector. It should be emphasized that employing characteristic differencing for the streamwise convective flux vector is a necessity. The resulting relaxation scheme will be unstable unless some special treatment is employed for the streamwise convection terms.

The method employed here to determine the differencing of the streamwise convective flux vector is the split-coefficient matrix (SCM) method of Chakravarthy (31). The SCM approach is based on a nonconservative formulation of the governing
For illustrative purposes, the SCM approach is discussed in terms of the two-dimensional unsteady Euler equations, given by

\[
\frac{\partial U}{\partial t} + A \frac{\partial U}{\partial \xi} + B \frac{\partial U}{\partial \eta} = 0
\]

where

\[A = \xi_x \frac{\partial E_i}{\partial U} + \xi_y \frac{\partial F_i}{\partial U}\]

\[B = \eta_x \frac{\partial E_i}{\partial U} + \eta_y \frac{\partial F_i}{\partial U}\]

Since the Euler equations are hyperbolic in time, the following similarity transformation can be applied to the Jacobian matrices A and B:

\[A = T_A T^{-1}\]

\[B = S_B S^{-1}\]  \hspace{1cm} (55)

where \(\Lambda_A\) and \(\Lambda_B\) are diagonal matrices with elements that are the eigenvalues of A and B respectively and \(T^{-1}\) and \(S^{-1}\) are the matrices with rows that are the left eigenvectors of A and B respectively. In the SCM approach, the A and B matrices are split using equation 55 as
\[ A = A^+ + A^- = T_A^{\lambda^+}T^{-1} + T_A^{\lambda^-}T^{-1} \]
\[ B = B^+ + B^- = T_B^{\lambda^+}T^{-1} + T_B^{\lambda^-}T^{-1} \]

where \( \lambda_A^+ \) is the diagonal matrix with elements that are the positive eigenvalues of \( A \), \( \lambda_A^- \) is the diagonal matrix of the negative eigenvalues of \( A \), \( \lambda_B^+ \) is the diagonal matrix with elements that are the positive eigenvalues of \( B \), and \( \lambda_B^- \) is the diagonal matrix with elements that are the negative eigenvalues of \( B \). Using this splitting, the unsteady Euler equations can be written as

\[
\frac{\partial U}{\partial t} + A^+ \frac{\partial U}{\partial \xi} + A^- \frac{\partial U}{\partial \xi} + B^+ \frac{\partial U}{\partial \eta} + B^- \frac{\partial U}{\partial \eta} = 0. \tag{56}
\]

Backward differences are used to approximate the derivatives multiplied by \( A^+ \) and \( B^+ \) and forward differences are used to approximate those multiplied by \( A^- \) and \( B^- \).

In the hybrid algorithm described below, the SCM formulation is applied only to the streamwise convective flux vector in subsonic regions. The key consideration here is that no discontinuities are present in the subsonic region. Therefore, the nonconservative SCM scheme should yield excellent results. The SCM formulation was selected for two reasons: 1) the SCM formulation results in a consistent source-free algorithm and 2) the SCM algorithm is easily implemented in the existing code. The first point is
particularly significant in contrast to algorithms such as the flux splitting scheme of Steger and Warming (14). As explained in reference 32, the split flux formulation is subject to inconsistencies where a finite-difference approximation spans a change in eigenvalue sign, i.e., entering or exiting a region of reversed flow. Because of the nonconservative formulation, the SCM approach is free from behavior of this type. It is emphasized that standard central differences are used to approximate the transverse derivatives and that the standard conservative algorithm is used in the outer supersonic flow. Since central differences are used to approximate the transverse derivatives, it is necessary to include the fourth-order damping term in the subsonic region as well as in the supersonic region.

The SCM formulation is implemented using the differencing developed for the GPI algorithm, i.e., equation 47 with \( \tilde{\alpha} = 0 \) and \( \beta = 1 \), so that the resulting difference approximation for the streamwise convective flux vector is given by

\[
\left( 1 - \frac{\theta}{3} \right) \Delta \xi \left( \xi_x \frac{\partial E_i}{\partial \xi} + \xi_y \frac{\partial F_i}{\partial \eta} \right) \\
= (A^+ - A^-) \Delta U_j^{n+1} - \frac{\theta}{3} (A^+ \Delta U_{j-1}^{n+1} + A^- \Delta U_{j+2}^n) \\
+ A^- (U_{j+2}^n - U_j^n) + \frac{2\theta}{9} A^- (U_{j+3}^n - 2U_{j+2}^n + U_{j+1}^n)
\]
where
\[
A = \xi_x \frac{\partial E_i}{\partial U} + \xi_y \frac{\partial F_i}{\partial U}
\]
\[
A^+ = T_A^+ T^{-1}
\]
\[
A^- = T_A^- T^{-1}
\]

the Jacobians \( \partial E_i / \partial U \) and \( \partial F_i / \partial U \) are given in Appendix A, and \( T \) and \( T^{-1} \) are given in reference 31. The difference equations are linearized by evaluating the \( A \) matrix at the previous iteration level \( n \). As before, the unsteady term is appended to the algorithm as in equation 54.

Since the transverse derivatives are approximated using central differences in the hybrid algorithm, the numerical boundary condition procedures employed at the upper boundary and the wall are identical to those employed in the GPI algorithm. It should be noted that a slight inconsistency is introduced because of the manner in which the normal pressure gradient is evaluated. Recall that in the GPI algorithm, the Vigneron splitting is utilized to obtain an expression for the normal pressure gradient at the wall that is consistent with the difference approximation used at points adjacent to the wall. This inconsistency is not expected to introduce any significant errors.

As with the GPI algorithm, the hybrid algorithm requires
a boundary condition at the downstream boundary. Assuming no reversed flow at the downstream boundary, there is one negative eigenvalue in subsonic regions. This eigenvalue corresponds to wave propagation into the computational domain and must be replaced by a boundary condition. For simplicity, the approach employed here is to set the elements of the $A^{-1}$ matrix to zero at station $j = j_{\text{max}}$ for the first-order algorithm and also at station $j = j_{\text{max}} - 1$ for the second-order algorithm. This is equivalent to assuming that the upstream influence is negligible at the downstream boundary and is similar to the procedure employed in the GPI algorithm. The boundary conditions at the inflow boundary correspond to the initial data for the space-marching approach and are held fixed during the computation.

It should be noted that the hybrid algorithm is not restricted to computing flows with streamwise separation. In fact, this approach is applicable to unseparated flows as well. Before the separated results are discussed, second-order accurate results predicted using the hybrid algorithm are briefly compared with results predicted using the GPI algorithm for the 5° and 7.5° compression corner cases presented previously. Figure 24 shows the streamwise variation of the wall pressure predicted using both of these approaches and the full NS results of Carter (22) for the 5° and the 7.5° compression corner cases. The agreement between
Figure 24. Streamwise distribution of wall pressure for the 5° and 7.5° compression corners
the wall pressures predicted using the hybrid algorithm and
the GPI algorithm is excellent. Both cases show slightly less
inviscid/viscous interaction than the full NS results of
reference 22. The results predicted using the hybrid
algorithm also show a barely perceptible inflection point at
the corner. Figure 25 shows the streamwise variation of the
wall friction coefficient predicted using each of these
approaches for the 5° and the 7.5° compression corners. Of
particular interest here is the fact that the results
predicted using the hybrid algorithm show a more pronounced
decrease in wall shear at the corner than the results
predicted using the GPI algorithm. In fact, the hybrid
algorithm results show a small region of separated flow for
the 7.5° corner similar to the full NS results of Carter (22)
while the results predicted using the GPI algorithm do not.
The hybrid algorithm requires a larger number of iterations to
reach convergence than the GPI algorithm: 102 iterations for
the 5° corner and 142 iterations for the 7.5° corner,
approximately a 60 percent increase over the GPI algorithm.
The hybrid algorithm also requires more computational effort
per iteration than the GPI algorithm due to the computation of
the $A^+$ and $A^-$ matrices. As a final note, the optimum
convergence rate of the hybrid algorithm was obtained when
$1/\Delta t = 0$. The behavior of the algorithm will be discussed in
more detail in the next section.
Figure 25. Streamwise distribution of wall friction coefficient for the 5° and 7.5° compression corners
Separated Flow Results

The GPI algorithm with the FLARE approximation and the hybrid algorithm described above were used to compute the flow past the 10° compression corner configuration with \( M_\infty = 3.0 \), \( Re_\infty = 1.68 \times 10^4 \), \( T_\infty = 217 \text{ K} \), and \( T_{\text{wall}} = 607.6 \text{ K} \) that was computed previously using the single-pass algorithm. The grid employed for the calculation was generated using the analytical functions discussed previously with \( \xi_{\text{max}} = 61 \) and \( \eta_{\text{max}} = 51 \). As before, mesh points were clustered in physical space in both the streamwise direction and the transverse direction with \( \beta_x = 2.5 \) and \( \beta_y = 6.5 \) respectively. Neumann boundary conditions were applied on the upper boundary and the initial conditions were obtained from the first-order single-pass solution reported previously. The same criterion employed previously to determine convergence of the iteration, namely that the maximum change in the wall pressure for a single iteration should be less than 0.001 percent, was used. Although no comparison data are available, a solution for the flow past a 12.5° compression corner at the conditions given above was also obtained using the hybrid algorithm. It was not possible to obtain a solution for this case using the GPI algorithm with the FLARE approximation. In the paragraphs below, results predicted using the two algorithms are discussed. In addition, the behavior of each numerical scheme is discussed.
Figure 26 shows a comparison between the streamwise variation of the wall pressure for the 10° compression corner predicted using the second-order GPI algorithm with the FLARE approximation, the first-order single-pass solution, and the second-order NS calculation of Carter (22). The GPI results show fairly good agreement with the results of reference 22. However, the full NS results show somewhat more viscous/inviscid interaction as evidenced by the larger extent of upstream influence. In addition, the results of reference 22 show more of a tendency towards a pressure plateau as evidenced by the inflection point in the pressure distribution at the corner. Once again, the single-pass algorithm produces results that are unacceptable.

Figures 27 and 28 show contour plots of the pressures predicted using the first-order single-pass algorithm and the second-order GPI algorithm with the FLARE approximation respectively. As before, each pressure contour represents a change in pressure of 0.0025. The abrupt compression at the corner, characteristic of the single-pass algorithm, is clearly present in Figure 27. Figure 28 shows the effects of the inviscid/viscous interaction on the external supersonic flow. Of particular interest is the coalescence of the compression waves just upstream of the corner to form the primary shock in the outer flow. In addition, the compression waves just downstream of the corner coalesce to form a
Figure 26. Streamwise distribution of wall pressure for the 10° compression corner
Figure 27. Pressure contours predicted using the single-pass algorithm for the 10° compression corner.
Figure 28. Pressure contours predicted using the GPI algorithm for the 10° compression corner
secondary compression that merges with the primary shock to form a single resultant shock in the outer flow.

Although the GPI results shown in Figure 26 showed an acceptable level of error in the wall pressure distribution, the error present in the associated wall shear distribution is significant. Figure 29 shows a comparison between the streamwise variation of the wall friction coefficient for the 10° compression corner predicted using the second-order GPI algorithm with the FLARE approximation, the first-order single-pass solution, and the second-order NS calculation of reference 22. As shown in Figure 29, there is a significant discrepancy between the results predicted using the GPI algorithm and the full NS results of reference 22. The decrease in wall friction coefficient associated with the thickening boundary layer ahead of the corner is significantly underpredicted. Not surprisingly, the size of the separated region as well as the minimum value of the wall shear are also underpredicted. The single-pass results obviously do not provide an acceptable approximation to the physics of the flow.

In an attempt to resolve the discrepancies shown in Figures 26 and 29, the GPI results shown above were compared with the results of Hung and MacCormack (33) for the 10° compression corner. Figure 30 shows a comparison between the streamwise variation of the wall pressure for the 10°
Figure 29. Streamwise distribution of wall friction coefficient for the 10° compression corner
Figure 30. Streamwise distribution of wall pressure for the 10° compression corner
compression corner predicted using the second-order GPI algorithm with the FLARE approximation and the second-order NS calculation of reference 33. The GPI results show excellent agreement with the full NS results of reference 33. The full NS results show only slightly more viscous/inviscid interaction and more of a tendency toward development of a pressure plateau. Figure 31 shows a comparison between the streamwise variation of the wall friction coefficient for the 10° compression corner predicted using the second-order GPI algorithm with the FLARE approximation and the second-order NS calculation of reference 33. The GPI results show excellent agreement with the full NS calculations of Hung and MacCormack. Although not shown here, a comparison between the results presented in reference 22 and reference 33 gives a good indication of the numerical error band - the difference in solutions obtained using different numerical methods. The intent here is not to imply that the results of Carter (22) are incorrect but only to show that the results predicted using the GPI algorithm with the FLARE approximation are acceptable in light of the differences in the results shown in reference 22 and reference 33.

For this particular case, the FLARE approach obviously does not significantly degrade the numerical approximation of the physical flow. The primary reason for the success of the FLARE approximation is that the maximum magnitude of the
Figure 31. Streamwise distribution of wall friction coefficient for the 10° compression corner.
reversed flow velocity is less than 1 percent of the freestream velocity. The only difference of note is that the full NS solutions show a more pronounced secondary decrease in wall shear at the corner as shown in Figures 29 and 31. It is difficult to discern whether this behavior is due to the FLARE approximation or whether it is related to the decreased upstream influence characteristic of GPI solutions.

Figure 32 shows a plot of the convergence history for the second-order GPI algorithm with $\alpha = 0$ and $\beta = 1$ and the FLARE approximation for different values of $\Delta t$. Of interest here is the fact that a converged solution for $1/\Delta t = 0$ was not obtained. In this case, severe oscillations developed in the transverse pressure distribution at the corner and the iteration diverged. The addition of the unsteady term is necessary to assure diagonal dominance in this case.

Figure 33 shows a comparison between the streamwise variation of the wall pressure for the 10° compression corner predicted using the hybrid algorithm discussed above, the GPI algorithm with the FLARE approximation, and the NS calculation of Hung and MacCormack (33). The results predicted using the hybrid approach are in excellent agreement with the other results although the hybrid algorithm results show slightly less viscous/inviscid interaction than either the GPI results or the full NS results. In addition, the streamwise pressure distribution predicted using the hybrid algorithm shows an
Figure 32. Convergence history of GPI algorithm for the 10° compression corner
Figure 33. Streamwise distribution of wall pressure for the 10° compression corner
inflection point similar to that shown by the full NS results and shown only slightly by the results predicted using the GPI algorithm. This inflection point is an indication that a pressure plateau is beginning to form. No pressure contour plots are shown since they are qualitatively identical to those shown in Figure 28.

Figure 34 shows a comparison between the streamwise variation of the wall friction coefficient for the 10° compression corner predicted using the hybrid algorithm discussed above, the GPI algorithm with the FLARE approximation, and the NS calculation of reference 33. Once again, very good agreement is obtained between the hybrid approach and the other two methods. Of particular significance is the secondary decrease in wall shear that is shown in the full NS results and exhibited by the results predicted using the hybrid approach. This decrease is noticeably underpredicted in the GPI results. The presence of the pressure plateau and the secondary decrease in wall shear are indications that the details of the flow field in the separated region are being correctly predicted using the hybrid approach. These results suggest that the FLARE approximation was the mechanism that retarded the development of the pressure plateau and, consequently, resulted in underpredicting the secondary decrease in wall shear as discussed previously.
Figure 34. Streamwise distribution of wall friction coefficient for the 10° compression corner.
Figure 35 shows a plot of the convergence history for the hybrid algorithm for different values of $\Delta t$. Significantly, the hybrid approach required substantially more iterations for convergence than the GPI algorithm. In addition, the convergence rate was optimized for $1/\Delta t = 0$. Apparently, the algorithm is inherently diagonally dominant so that adding the unsteady term is unnecessary. Unfortunately, the resulting system is so "stiff" that the convergence rate is degraded when compared to the GPI algorithm.

To test the robustness of the two multiple-pass algorithms, the flow past the $12.5^\circ$ compression corner configuration with $M_\infty = 3.0$, $Re_\infty = 1.68 \times 10^4$, $T_\infty = 217$ K, and $T_{wall} = 607.6$ K was computed. It was not possible to obtain a converged solution using the GPI algorithm with FLARE approximation. Severe oscillations developed in the transverse pressure distribution near the corner and the iteration diverged. The actual number of iterations until divergence varied depending on the value of $\Delta t$. The point to be made is that the iteration did not diverge immediately but required development of a substantial region of reversed flow. Apparently, the implementation of the FLARE approximation employed here is not sufficient to guarantee the positivity of the appropriate eigenvalues. This behavior is not particularly disheartening since it is not anticipated that the FLARE approximation would have produced very good results.
Figure 35. Convergence history of hybrid algorithm for the 10° compression corner
in the immediate vicinity of the separated region since the physical convection processes are not being modelled. On the other hand, the hybrid algorithm was very robust and exhibited no erratic behavior. A converged solution was obtained in 367 iterations using $1/\Delta t = 0$, the optimum value for the other cases considered.

Figure 36 shows a plot of the streamwise variation of the wall pressure computed using the hybrid algorithm for the 12.5° compression corner. In this case, a well-defined pressure plateau is clearly visible at the corner indicating the presence of a significant region of recirculating flow. Figure 37 shows a contour plot of the pressures predicted using the hybrid algorithm. The increase in the spacing of the isobars at the wall near the corner also indicates the development of the pressure plateau. Also shown in Figure 37 is the shock structure described previously. The coalescence of the compressive waves in the supersonic outer flow into two distinct shocks that merge into a single resultant shock is clearly shown.

Figure 38 shows a plot of the streamwise variation of the wall friction coefficient. The most distinctive feature of the plot is the large secondary decrease in wall shear at the corner that is characteristic of flows of this type. The maximum reversed flow velocity occurring in the recirculation region is approximately 5 percent of the freestream velocity.
Figure 36. Streamwise distribution of wall pressure for the 12.5° compression corner
Figure 37. Pressure contours predicted using the hybrid algorithm for the 12.5° compression corner
Figure 38. Streamwise distribution of wall friction coefficient for the 12.5° compression corner
Although no data are available for comparison with this computed solution, the results predicted using the hybrid algorithm are intuitively correct. These results show that the hybrid algorithm is capable of predicting compression corner flow fields with significant streamwise elliptic behavior and regions of recirculating flow.

Summary

Two different multiple-pass algorithms were developed for solving the TLNS equations for flows with significant streamwise elliptic behavior. Both of these schemes employed some form of characteristic differencing for the streamwise convective flux vector. The splitting procedures were used to identify the portions of the streamwise convective flux vector that were responsible for "positive" time-like behavior and which portions were responsible for "negative" time-like behavior so that appropriate finite-difference approximations could be employed. In this way, a stable time-marching algorithm was developed. The significant conclusions reported in the previous sections concerning these multiple-pass relaxation schemes are now summarized.

A second-order GPI algorithm was developed using the Vigneron splitting for the streamwise pressure gradient. By combining two GPI schemes given in the literature, a generalized GPI algorithm was developed. Using this
generalized algorithm, the two formally different GPI schemes were shown to be equivalent under certain conditions. The equivalence of these two algorithms was employed along with a heuristic stability analysis of a model problem to explain the behavior of standard second-order GPI algorithms. Based on the heuristic stability analysis and examination of the modified equation, a conditionally stable well-behaved second-order GPI algorithm was developed. For flows with streamwise separation, the FLARE approximation was employed. Results predicted using the GPI algorithm showed good agreement with results presented in the literature although the GPI results typically showed less viscous/inviscid interaction. It was determined that the convergence rate of the GPI scheme was significantly improved through the addition of an unsteady term that enhanced the diagonal dominance of the algorithm. The size of the time step played a significant role in determining the convergence rate of the algorithm. This algorithm proved to unstable for flows with large recirculating regions, apparently due to the manner in which the FLARE approximation was implemented.

In addition to the GPI algorithm, a new second-order algorithm was developed using a hybrid approach. The standard conservative PNS algorithm was employed in the outer supersonic flow while in the subsonic region near the wall, a nonconservative SCM scheme was used to determine the
differencing for the streamwise convective flux vector. Since there were no embedded shocks in the subsonic region, the SCM formulation provided an excellent approximation to the streamwise convection of information. The second-order differencing scheme developed for the GPI algorithm was employed for the streamwise differencing. Standard central differencing was employed for the transverse derivatives. Results predicted using the hybrid algorithm were very similar to results predicted using the GPI scheme with one exception: the hybrid algorithm showed more of a decrease in wall shear at the corner than the GPI results. The hybrid algorithm required more iterations and more computer time per iteration than the GPI algorithm. However, unlike the GPI algorithm, the hybrid approach was capable of predicting flows with extensive streamwise separation.
CHAPTER 5. CONCLUDING REMARKS

As stated in the introduction, the purpose of this study was to identify and explore efficient and reliable numerical algorithms that can be employed to predict complex supersonic/hypersonic viscous flows. This objective was successfully accomplished although the cases considered here focused only on moderately supersonic flows. The first approach examined here was a highly efficient single-pass algorithm for solving the PNS equations. This approach was found to be inadequate purely because the single-pass philosophy precluded the simulation of the streamwise elliptic behavior present in the solution. The single-pass philosophy was abandoned and two relaxation schemes utilizing multiple passes were developed. In those instances when the GPI algorithm converged, it proved to be the more computationally efficient algorithm of the two multiple-pass algorithms considered. Results predicted using the GPI algorithm for cases with no streamwise flow reversal or only small separated regions were very good. Unfortunately, due to the divergence of the algorithm, cases with large regions of reversed flow could not be predicted. This divergence led to the development of the hybrid algorithm. The motivation for the development of the hybrid algorithm was that there was no unambiguous method of employing upwind differences for the streamwise convection terms in separated regions without resorting to some form of
splitting method. The results predicted for unseparated flows using the hybrid algorithm were of similar quality to those predicted using the GPI scheme. The hybrid algorithm had the added advantage that it converged for each case attempted during this study. In regions of recirculating flow, the hybrid algorithm produced results that showed the correct qualitative behavior.

This study has shown that efficient numerical techniques for flows of this type can be developed based on relaxation schemes for upwinded algorithms. One point not considered in this study is the upwinding of the transverse convective terms. Upwinding the transverse derivatives would eliminate the necessity of including the ad hoc artificial viscosity. Of course, in the outer supersonic flow, a conservative splitting must be used in order to accurately capture the embedded shock. It should be noted that in the algorithms discussed here, the streamwise convective terms are effectively upwinded in the outer supersonic flow. An additional topic for further research is an improvement in the convergence rate of the algorithm. The construction of the relaxation schemes considered here gives them a preferred direction. Recall that information is propagated back upstream one or at most two grid points per iteration. The convergence rate may be improved by including a sweep from the downstream boundary to the inflow boundary in regions where
the eigenvalues indicate the existence of "negative" time-like behavior. In this way information is propagated equally in the streamwise direction. In addition, iterative techniques such as the multigrid approach should be considered as possible methods of accelerating convergence of the algorithm.
BIBLIOGRAPHY


ACKNOWLEDGEMENTS

The author is greatly indebted to the many people who have made this work possible either through direct or indirect contributions. Thanks are especially due to Dr. Dale Anderson for his guidance, encouragement, and support. Without him, none of this would have been possible. The author wishes to express thanks to Dr. Len Wilson for his support and for agreeing to become the major professor of record. Thanks are also due to Dr. Rich Hindman with whom the author had many interesting and enlightening discussions. The author would also like to thank the many graduate students who contributed ideas and encouragement.

The author would like to thank his parents for instilling in him a curiosity about the world and a desire to learn. Finally, the author would like to express his extreme gratitude to his wife Gina for enduring many hardships without so much as a single complaint. Without her, none of this would have been worthwhile.

The partial support of the General Dynamics/Fort Worth Division under contract 1129566 is also gratefully acknowledged.
APPENDIX A. JACOBIAN MATRICES

The Jacobian matrices referred to in the discussions of the behavior of the governing equations and as well as those utilized in the development of the numerical algorithm are presented here. In all cases, the form in computational \((\xi, \eta)\) space is given.

The elements of the Jacobian matrix of the modified inviscid flux vector, \(\partial E^*/\partial U\), are given by

\[
\left(\frac{\partial E^*_i}{\partial U}\right)_{1,1} = 0, \quad \left(\frac{\partial E^*_i}{\partial U}\right)_{1,2} = 1
\]

\[
\left(\frac{\partial E^*_i}{\partial U}\right)_{1,3} = 0, \quad \left(\frac{\partial E^*_i}{\partial U}\right)_{1,4} = 0
\]

\[
\left(\frac{\partial E^*_i}{\partial U}\right)_{2,1} = \frac{\omega(\gamma - 1)}{2} (u^2 + v^2) - u^2
\]

\[
\left(\frac{\partial E^*_i}{\partial U}\right)_{2,2} = (2 - \omega(\gamma - 1))u
\]

\[
\left(\frac{\partial E^*_i}{\partial U}\right)_{2,3} = -\omega(\gamma - 1)v, \quad \left(\frac{\partial E^*_i}{\partial U}\right)_{2,4} = \omega(\gamma - 1)
\]

\[
\left(\frac{\partial E^*_i}{\partial U}\right)_{3,1} = -uv, \quad \left(\frac{\partial E^*_i}{\partial U}\right)_{3,2} = v
\]
\[
\begin{align*}
\frac{\partial E_i^*}{\partial U}_{3,3} &= u, \quad \frac{\partial E_i^*}{\partial U}_{3,4} = 0 \\
\frac{\partial E_i^*}{\partial U}_{4,1} &= (-\gamma E_t/\rho + (\gamma - 1)(u^2 + v^2))u \\
\frac{\partial E_i^*}{\partial U}_{4,2} &= \gamma E_t/\rho - \frac{(\gamma - 1)}{2} (3u^2 + v^2) \\
\frac{\partial E_i^*}{\partial U}_{4,3} &= -(\gamma - 1)uv, \quad \frac{\partial E_i^*}{\partial U}_{4,4} = \gamma u
\end{align*}
\]

where the symbols used above are identical to those defined in the text. The elements of the Jacobian matrix of the modified inviscid flux vector, \( \partial F_i^*/\partial U \), are given by

\[
\begin{align*}
\frac{\partial F_i^*}{\partial U}_{1,1} &= 0, \quad \frac{\partial F_i^*}{\partial U}_{1,2} = 0 \\
\frac{\partial F_i^*}{\partial U}_{1,3} &= 1, \quad \frac{\partial F_i^*}{\partial U}_{1,4} = 0 \\
\frac{\partial F_i^*}{\partial U}_{2,1} &= -uv, \quad \frac{\partial F_i^*}{\partial U}_{2,2} = v \\
\frac{\partial F_i^*}{\partial U}_{2,3} &= u, \quad \frac{\partial F_i^*}{\partial U}_{2,4} = 0 \\
\frac{\partial F_i^*}{\partial U}_{3,1} &= \frac{\omega(\gamma - 1)}{2} (u^2 + v^2) - v^2
\end{align*}
\]
\[
\begin{align*}
\left( \frac{\partial F_i}{\partial U} \right)_{3,2} &= -\omega (\gamma - 1)u \\
\left( \frac{\partial F_i}{\partial U} \right)_{3,3} &= (2 - \omega (\gamma - 1))v \\
\left( \frac{\partial F_i}{\partial U} \right)_{3,4} &= \omega (\gamma - 1) \\
\left( \frac{\partial F_i}{\partial U} \right)_{4,1} &= (-\gamma \frac{E_t}{\rho} + (\gamma - 1)(u^2 + v^2))v \\
\left( \frac{\partial F_i}{\partial U} \right)_{4,2} &= -(\gamma - 1)uv \\
\left( \frac{\partial F_i}{\partial U} \right)_{4,3} &= \gamma \frac{E_t}{\rho} - \frac{(\gamma - 1)}{2} (u^2 + 3v^2) \\
\left( \frac{\partial F_i}{\partial U} \right)_{4,4} &= \gamma v 
\end{align*}
\]

In the derivation of the Jacobians given above, \( \omega \) was treated as a constant as explained in the text. The Jacobians of the standard inviscid flux vectors, \( \partial E_i/\partial U \) and \( \partial F_i/\partial U \), are obtained using the definitions given above with \( \omega = 1 \). Note that the presence of \( \omega \) alters only the second row of \( \partial E_i/\partial U \) and only the third row of \( \partial F_i/\partial U \).

The Jacobian matrix of the portion of the streamwise inviscid flux vector responsible for the elliptic behavior,
\[
\frac{\partial \mathbf{E}_i}{\partial u}, \text{ is given by}
\]
\[
\begin{bmatrix}
0 & 0 & 0 & 0 \\
\frac{u^2 + v^2}{2} & -u & -v & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]
\[
(1 - \omega)(\gamma - 1)
\]

Similarly, the Jacobian matrix \( \frac{\partial \mathbf{E}_i}{\partial u} \) is given by
\[
\begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\frac{u^2 + v^2}{2} & -u & -v & 1 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

Note, of course, that if \( \omega = 1 \) all of the elements in both of the above matrices are identically zero.

During the discussion of the linearization of the viscous terms \( \Delta \mathbf{E} v' \) and \( \Delta \mathbf{F} v' \), it was noted that these terms are functions of \( U, \mu(U) \), and the metric quantities \( \eta_x \) and \( \eta_y \) and that \( \eta \) derivatives are contained in these terms. As such, the
viscous Jacobians are considerably more complex than the
inviscid Jacobians given above. The elements of the viscous
Jacobian $\frac{\partial E'}{\partial U}$ are derived assuming that $\mu$ and the metrics
are held constant and are given by

\[
\left( \frac{\partial E'}{\partial U} \right)_{1,1} = 0, \quad \left( \frac{\partial E'}{\partial U} \right)_{1,2} = 0
\]

\[
\left( \frac{\partial E'}{\partial U} \right)_{1,3} = 0, \quad \left( \frac{\partial E'}{\partial U} \right)_{1,4} = 0
\]

\[
\left( \frac{\partial E'}{\partial U} \right)_{2,1} = \frac{\mu}{Re_{\infty}} \left( - \frac{4}{3} \eta_x \frac{\partial}{\partial \eta} (u/\rho) + \frac{2}{3} \eta_y \frac{\partial}{\partial \eta} (v/\rho) \right)
\]

\[
\left( \frac{\partial E'}{\partial U} \right)_{2,2} = \frac{4\mu}{3Re_{\infty}} \eta_x \frac{\partial}{\partial \eta} (1/\rho)
\]

\[
\left( \frac{\partial E'}{\partial U} \right)_{2,3} = \frac{2\mu}{3Re_{\infty}} \eta_y \frac{\partial}{\partial \eta} (1/\rho), \quad \left( \frac{\partial E'}{\partial U} \right)_{2,4} = 0
\]

\[
\left( \frac{\partial E'}{\partial U} \right)_{3,1} = -\frac{\mu}{Re_{\infty}} \left( \eta_y \frac{\partial}{\partial \eta} (u/\rho) + \eta_x \frac{\partial}{\partial \eta} (v/\rho) \right)
\]

\[
\left( \frac{\partial E'}{\partial U} \right)_{3,2} = \frac{\mu}{Re_{\infty}} \eta_y \frac{\partial}{\partial \eta} (1/\rho)
\]

\[
\left( \frac{\partial E'}{\partial U} \right)_{3,3} = \frac{\mu}{Re_{\infty}} \eta_x \frac{\partial}{\partial \eta} (1/\rho), \quad \left( \frac{\partial E'}{\partial U} \right)_{3,4} = 0
\]
where it is understood that all partial derivatives except those marked with asterisks also operate on $\Delta U$. The $\eta$ derivatives designated by an asterisk operate only on those
terms contained within the following parentheses. The
elements of the Jacobian matrix $\partial Fv'/\partial U$ are derived in the
same manner as above and are given by

$$
\left( \frac{\partial Fv'}{\partial U} \right)_{1,1} = 0, \quad \left( \frac{\partial Fv'}{\partial U} \right)_{1,2} = 0
$$

$$
\left( \frac{\partial Fv'}{\partial U} \right)_{1,3} = 0, \quad \left( \frac{\partial Fv'}{\partial U} \right)_{1,4} = 0
$$

$$
\left( \frac{\partial Fv'}{\partial U} \right)_{2,1} = -\frac{\mu}{Re_\infty} \left( \eta_y \frac{\partial}{\partial \eta} (u/\rho) + \eta_x \frac{\partial}{\partial \eta} (v/\rho) \right)
$$

$$
\left( \frac{\partial Fv'}{\partial U} \right)_{2,2} = \frac{\mu}{Re_\infty} \eta_y \frac{\partial}{\partial \eta} (1/\rho)
$$

$$
\left( \frac{\partial Fv'}{\partial U} \right)_{2,3} = \frac{\mu}{Re_\infty} \eta_x \frac{\partial}{\partial \eta} (1/\rho), \quad \left( \frac{\partial Fv'}{\partial U} \right)_{2,4} = 0
$$

$$
\left( \frac{\partial Fv'}{\partial U} \right)_{3,1} = \frac{\mu}{Re_\infty} \left( -\frac{4}{3} \eta_y \frac{\partial}{\partial \eta} (v/\rho) + \frac{2}{3} \eta_x \frac{\partial}{\partial \eta} (u/\rho) \right)
$$

$$
\left( \frac{\partial Fv'}{\partial U} \right)_{3,2} = -\frac{2\mu}{3Re_\infty} \eta_x \frac{\partial}{\partial \eta} (1/\rho)
$$

$$
\left( \frac{\partial Fv'}{\partial U} \right)_{3,3} = \frac{4\mu}{3Re_\infty} \eta_y \frac{\partial}{\partial \eta} (1/\rho), \quad \left( \frac{\partial Fv'}{\partial U} \right)_{3,4} = 0
$$
\[
\left( \frac{\partial F_v'}{\partial U} \right)_{4,1} = \frac{\mu}{Re_{\infty}} \left( -\frac{4}{3} \eta y \frac{\partial}{\partial \eta} (v^2/\rho) - \eta y \frac{\partial}{\partial \eta} (u^2/\rho) \right) \\
+ \frac{2}{3} \eta x \frac{\partial}{\partial \eta} (u/\rho) - \eta x \frac{\partial}{\partial \eta} (v/\rho) \\
- \frac{\gamma}{Pr} \frac{\eta y}{\eta} \left( \frac{\rho}{\rho^2 (\gamma - 1)} - \frac{u^2 + v^2}{2\rho} \right) \\
+ \eta x \left( \frac{2}{3} (v/\rho) \frac{\partial}{\partial \eta} (u) - (u/\rho) \frac{\partial}{\partial \eta} (v) \right) \\
\left( \frac{\partial F_v'}{\partial U} \right)_{4,2} = \frac{\mu}{Re_{\infty}} \left( (1 - \frac{\gamma}{Pr}) \eta y \frac{\partial}{\partial \eta} (u/\rho) - \frac{2}{3} \eta x \frac{\partial}{\partial \eta} (1/\rho) \right) \\
+ \eta x \left( (1/\rho) \frac{\partial}{\partial \eta} (v) \right) \\
\left( \frac{\partial F_v'}{\partial U} \right)_{4,3} = \frac{\mu}{Re_{\infty}} \left( \left( \frac{4}{3} \frac{\gamma}{Pr} \right) \eta y \frac{\partial}{\partial \eta} (v/\rho) + \eta x \frac{\partial}{\partial \eta} (1/\rho) \right) \\
- \frac{2}{3} \eta x \left( (1/\rho) \frac{\partial}{\partial \eta} (u) \right) \\
\left( \frac{\partial F_v'}{\partial U} \right)_{4,4} = \frac{\gamma \mu}{Re \cdot Pr} \eta y \frac{\partial}{\partial \eta} (1/\rho)
\]

where the symbols have the same meanings as above.

Since \(E_v'\) and \(F_v'\) are also implicit functions of \(U\) through \(\mu(U)\), the Jacobians shown above do not give the
The complete relationship between the change in the viscous vectors and the changes in $U$ since $\mu$ was assumed locally constant. The Jacobian $(\partial \mathbf{E}' / \partial \mu)(\partial \mu / \partial U)$ that relates the changes in the viscous vector $\mathbf{E}'$ to the change in $\mu$ via changes in $U$ is given by

\[
(\gamma - 1) \frac{f(T)}{2p} \begin{bmatrix}
0 & 0 & 0 & 0 \\
\phi \mathbf{E}'_2 & -u \mathbf{E}'_2 & -v \mathbf{E}'_2 & \mathbf{E}'_2 \\
\phi \mathbf{E}'_3 & -u \mathbf{E}'_3 & -v \mathbf{E}'_3 & \mathbf{E}'_3 \\
\phi \mathbf{E}'_4 & -u \mathbf{E}'_4 & -v \mathbf{E}'_4 & \mathbf{E}'_4
\end{bmatrix}
\]

with

\[
f(T) = \frac{T + 3K}{T + K}
\]

\[
\phi = -\mathbf{E}'_c / \rho + u^2 + v^2
\]

where $K$ is the constant used in Sutherland's formula for the viscosity, $\mathbf{E}'_2$ is the second element of the $\mathbf{E}'$ vector, $\mathbf{E}'_3$ is the third element of the $\mathbf{E}'$ vector, and $\mathbf{E}'_4$ is the fourth element of the $\mathbf{E}'$ vector. Similarly, the Jacobian $(\partial \mathbf{F}' / \partial \mu)(\partial \mu / \partial U)$ is given by
where $Fv'_{2}$ is the second element of the $Fv'$ vector, etc.

Finally, there are changes in the viscous vectors due to changes in the metrics appearing in these terms. The Jacobian vector that relates the changes in $Ev'$ to changes in $\eta_{x}$, $\partial Ev'/\partial \eta_{x}$, is given by

\[
\mu \frac{\partial}{\partial Re_{\infty}}
\begin{bmatrix}
0 \\
\frac{4}{3} \frac{\partial}{\partial \eta} (u) \\
\frac{\partial}{\partial \eta} (v) \\
\frac{2}{3} \frac{\partial}{\partial \eta} (u^{2}) + \frac{1}{2} \frac{\partial}{\partial \eta} (v^{2}) \\
+ \frac{\gamma}{Pr} \frac{\partial}{\partial \eta} (E_{T}/\rho - \frac{1}{2} (u^{2} + v^{2}))
\end{bmatrix}
\]
and the Jacobian vector that relates changes in $E v'$ to changes in $\eta_y$, $\partial E v' / \partial \eta_y$, is given by

\[
\begin{bmatrix}
0 \\
- \frac{2}{3} \eta \frac{\partial}{\partial \eta} (v) \\
\frac{\mu}{Re_\infty} \eta \frac{\partial}{\partial \eta} (u) \\
- \frac{2}{3} u \eta \frac{\partial}{\partial \eta} (v) + v \eta \frac{\partial}{\partial \eta} (u)
\end{bmatrix}
\]

Similarly, the Jacobian vector $\partial E v' / \partial \eta_x$ is given by

\[
\begin{bmatrix}
0 \\
\eta \frac{\partial}{\partial \eta} (v) \\
- \frac{2}{3} \eta \frac{\partial}{\partial \eta} (v) \\
- \frac{2}{3} v \eta \frac{\partial}{\partial \eta} (u) + u \eta \frac{\partial}{\partial \eta} (v)
\end{bmatrix}
\]

and the Jacobian vector $\partial E v' / \partial \eta_y$ is given by
\[
\begin{align*}
((zA + zn) \frac{Z}{\tau} - \sigma/\sigma) \frac{ue}{e} & + \\
(\epsilon) \frac{ue}{e} \frac{Z}{\tau} + (\epsilon) \frac{ue}{e} \frac{Z}{\tau} \\
(\Lambda) \frac{ue}{e} \frac{e}{\eta} \\
(n) \frac{ue}{e} \\
0
\end{align*}
\]
APPENDIX B. METRIC DIFFERENCING

The metrics of the transformation that relate computational \((\xi, \eta)\) space to physical \((x,y)\) space must be defined so that the numerical solution can be obtained. Since the grid generation scheme described previously utilizes analytic functions to construct the grid, the metrics could be computed analytically. For generality, the metrics are computed numerically using finite-difference approximations. Thompson et al. (21) clearly show that the same finite-difference approximation that is used to compute the partial derivative under consideration should also be used to approximate the associated metric. Otherwise, the formal accuracy of the finite-difference approximation is degraded because an inconsistency occurs in the truncation error.

The metrics of the transformation, repeated here for convenience,

\[
\begin{align*}
\xi_x &= J_y \eta, & \xi_y &= -J_x \eta \\
\eta_x &= -J_y \xi, & \eta_y &= J_x \xi
\end{align*}
\]

where

\[
J = \frac{1}{(x_\xi y_\eta - x_\eta y_\xi)}
\]

are computed using the approach outlined above. In the finite-difference algorithm, second-order central differences
are employed for the $\eta$ derivatives. Therefore, $x_\eta$ and $y_\eta$ are approximated using

$$x_\eta = \frac{x_{k+1} - x_{k-1}}{2\Delta\eta}$$

and

$$y_\eta = \frac{y_{k+1} - y_{k-1}}{2\Delta\eta}$$

where it is assumed that $\Delta\eta$ is constant. Likewise, since one-sided differences are used in the solution algorithm to approximate the derivatives in the $\xi$ direction, the $\xi$ difference approximations in the metrics should also reflect this fact. It is noted that forward differences should be employed to approximate the metrics that multiply the forward-differenced terms in the global pressure iteration scheme. However, for simplicity, no special treatment was given these metrics. Assuming that backward differences are being employed, $x_\xi$ and $y_\xi$ are approximated using

$$x_\xi = \frac{x_j - x_{j-1} - \theta/3(x_{j-1} - x_{j-2})}{(1 - \theta/3)\Delta\xi}$$

and
\[ y_\xi = \frac{y_j - y_{j-1} - \theta/3(y_{j-1} - y_{j-2})}{(1 - \theta/3)\Delta \xi}. \]

where it is assumed that $\Delta \xi$ is constant. However, suppose this form of differencing is applied at a discontinuity in the mesh, i.e., at the intersection of the flat plate and the ramp. The first- or second-order differencing of $x_\xi$ and first-order differencing of $y_\xi$ present no difficulty. The second-order differencing of $y_\xi$ across the discontinuity produces an oscillation in the streamwise variation of the metric that, in turn, produces oscillations in the streamwise pressure distribution at the shock. To see how this occurs, assume that the corner is located at $j = j_c$ and that the grid is equally spaced in the $\xi$ direction so that the surface is defined as

\[ y(j) = 0 \text{ for } j \leq j_c \]

\[ y(j) = (j - j_c)\Delta y_{\text{wall}} \text{ for } j > j_c \]

where

\[ \Delta y_{\text{wall}} = (dy/dx)_{\text{wall}} \Delta x. \]

Applying the second-order backward difference for $y_\xi$ yields

\[ y_\xi(j_c) = 0 \]
\[ y_\xi(j\xi+1) = \frac{3}{2} \Delta y_{wall} \]

\[ y_\xi(j\xi+2) = \Delta y_{wall} \]

so that the flow actually "sees" a different surface than is present since the slope of the ramp is constant. Two alternative approaches were considered to remedy this problem: evaluating \( y_\xi \) using a first-order backward difference and using a second-order central difference. Although not shown here, an analysis similar to that of Thompson et al. (21) can be used to show that using the centrally-differenced metric introduces a smaller error. Therefore, a central difference should be employed at \( j = j\xi+1 \). In the computational mesh, no point was located precisely at the corner. The central difference was used at the points located on either side of the corner and eliminated the oscillations in the streamwise pressure distribution. Using the central difference effectively "rounds" the corner since some of the effect of the corner is transmitted one grid point upstream.
APPENDIX C. ADDITIONAL BOUNDARY CONDITION RELATIONS

This appendix contains the additional relations necessary to implement the numerical boundary condition procedure described in the text for both the single-pass and multiple-pass algorithms.

The elements of $M_{j,k}$, used in the single-pass and multiple-pass algorithms, are given by

$$
(M_{j,k})^{1,1} = (\gamma - 1)(\rho/p)_{j,1} \left( \frac{u^2 + v^2}{2} \right)_{j,k}
$$

$$
(M_{j,k})^{1,2} = - (\gamma - 1)(\rho/p)_{j,1} u_{j,k}
$$

$$
(M_{j,k})^{1,3} = - (\gamma - 1)(\rho/p)_{j,1} v_{j,k}
$$

$$
(M_{j,k})^{1,4} = (\gamma - 1)(\rho/p)_{j,1}
$$

$$
(M_{j,k})^{2,1} = 0, \quad (M_{j,k})^{2,2} = 0
$$

$$
(M_{j,k})^{2,3} = 0, \quad (M_{j,k})^{2,4} = 0
$$

$$
(M_{j,k})^{3,1} = 0, \quad (M_{j,k})^{3,2} = 0
$$

$$
(M_{j,k})^{3,3} = 0, \quad (M_{j,k})^{3,4} = 0
$$

$$
(M_{j,k})^{4,1} = \left( \frac{u^2 + v^2}{2} \right)_{j,k}, \quad (M_{j,k})^{4,2} = - u_{j,k}
$$

$$
(M_{j,k})^{4,3} = - v_{j,k}, \quad (M_{j,k})^{4,4} = 1.
$$
The source term for the single-pass algorithm is given by

$$
S_p_j = (-3p_{j,1} + 4p_{j,2} - p_{j,3})
$$

for $p_{j,1}$, $p_{j,2}$, and $p_{j,3}$, where $p_{j,1}$ is given by

$$
p_{j,1} = \frac{1}{(\gamma - 1)}
$$

The source term for the multiple-pass algorithm is given by

$$
\Sigma p_j = \Sigma
$$

where $\Sigma$ is given in equation 50.