TIME ACCURATE INTERNAL FLOW SOLUTIONS OF THE THIN SHEAR LAYER EQUATIONS

by

Robert Hull Bush

GT&PDL Report No. 156 February 1981

GAS TURBINE & PLASMA DYNAMICS LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS

TIME ACCURATE INTERNAL FLOW SOLUTIONS OF THE THIN SHEAR LAYER EQUATIONS

by

Robert Hull Bush

GT&PDL Report No. 156 February 1981

This research, carried out in the Gas Turbine and Plasma Dynamics Laboratory, MIT, was supported by the NASA Lewis Research Center under Grant No. NAG3-9.

ABSTRACT

An implicit factored algorithm for the solution of the thin shear layer approximation of the Navier-Stokes equations is described and explicit boundary conditions are developed for internal flow problems. This scheme is compared to theoretical predictions and experimental data, as well as to other more thoroughly tested numerical schemes. The examples presented demonstrate the ability of the factored algorithm to accurately predict internal flow fields and provide insight into the difficulties associated with the numerical simulation of internal flow fields.

I am afraid that I rather give myself away when I explain. Results without causes are much more impressive.

Sherlock Holmes
"The Stock-Broker's Clerk"
Sir Arthur Conan Doyle

ACKNOWLEDGEMENTS

The author wishes to acknowledge with gratitude the assistance given to him by Prof. W. T. Thompkins, Jr. through the course of this research. Thanks are also due to his many friends and colleagues at M.I.T. for their endless discussions and the insight they provided. Special thanks to Beth Sakey who did the typing with skill and patience. Finally, sincere appreciation is given to his family and friends who gave constant encouragement.

This research was supported by NASA Lewis Research Center under grant No. NAG3-9. The support is especially appreciated.

TABLE OF CONTENTS

FIGURES

SYMBOLS

a	Speed of sound
C_f	Coefficient of friction
e	Internal energy
E_t	total energy
E	Flux vector in X(or ξ) direction
F	Flux vector in Y(or η) direction
J	Jacobian determinate
j	Node counter in ξ direction
k	Node counter in η direction
k	Thermal diffusivity
l	Length
n	Time level
p	Pressure
q	Vector of flow variables
R	Viscous vector containing viscous terms due to X(or ξ) variations
Re	Reynolds Number
S	Viscous vector containing viscous terms due to Y(or η) variations
t	Time
T	Temperature
u	Velocity component in X(or ξ) direction
U	Total velocity
v	Velocity component in Y(or η) direction

x	Physical space coordinate
y	Physical space coordinate
ρ	Density
μ	Viscosity
ξ	Computational coordinate along body
η	Computational coordinate normal to body
τ_{ij}	Shear stress on i face in j direction
$( )_o$	Reference value of ( ), usually the value at upstream stagnation conditions
$( )'$	Non dimensionalized quantity
$(\hat{})$	Vectors and matrices which have been re-defined for a non-orthogonal grid system by multiplying by the appropriate metrics

1.0 Introduction

During the past several years there has been rapid improvement in the ability to numerically simulate flow fields of aerodynamic interest. Most new algorithms developed have been applied to external flows because of the relative complexity of internal flows in which viscous layers, shocks and complex geometries often interact. However, there are now algorithms which appear to accurately model these phenomena. The purpose of this thesis is to apply one of them, an implicit factored algorithm described by Warming and Beam (1), to internal flow problems.

The flow fields studied were the flow through supersonic inlet diffusers and the flow through transonic compressor blade rows. This study was not overly concerned with the computational speed of the algorithm, but concentrated on studying the properties of internal flow calculations and the effect of boundary condition models. The computer codes and boundary condition models developed give results which agree well with theory and experiments and will serve as a base line calculation to which new, faster schemes may be compared.

The thin shear layer equations were solved because they incorporate the viscous and compressibility effects which occur often in internal flow fields. The strong conservation form of these equations was used to ensure conservation of mass, momentum and energy through the computational domain.

This form is retained through a general transformation of the equations to computational space. This transformation does not require that the grid mesh in physical space be orthogonal.

The implicit factored algorithm was used to solve the equations because of its generality, ease of conversion to three dimensions and proven performance in computing external flows. Another advantage of the factored scheme is that it breaks the calculation into several small sections making it possible to program and run the scheme on inexpensive 'mini' computers. Implicit schemes generally require more operations per node point than explicit schemes to advance a calculation one time step. However, they do not suffer the severe time step restriction of explicit schemes and with large time steps can often reach a converged solution in fewer total operations. The stability limit of explicit schemes is particularly severe when viscous flow fields are studied because of the close grid spacing required to resolve large shear stress gradients. Because of their stability properties, implicit schemes are gaining popularity for viscous calculations.

Explicit boundary conditions were developed which were shown to provide a good representation of the boundaries for the examples considered.

A series of example cases were calculated to demonstrate the accuracy of the solution scheme. The first exam-

ple, an inviscid diffuser calculation, was compared to a theoretical prediction and an independent calculation by Tong (2) using an explicit, time marching algorithm. The second example was a comparison of a shock boundary layer interaction calculation with experimental results obtained by Hakkinen, et al. (3). The third example, a viscous diffuser using the same geometry as the inviscid diffuser, demonstrated the effect that boundary layer blockage can have on diffuser performance. Finally, a more complex example was the calculation of the inviscid flow through a compressor blade row. This flow was also compared to an explicit, time marching calculation by Tong (2).

2.0 Factored Schemes for the Unsteady, Compressible Navier-Stokes Equations

2.1 System of Equations

The Navier-Stokes equations govern the flow of an unsteady, compressible fluid. Throughout this thesis it is assumed that the fluid is isotropic and that stress is linear with rate of strain. These assumptions are valid for most common fluids through the range of flow conditions of interest. It is assumed also that Stokes' hypothesis ($\lambda = -\frac{2}{3}\mu$) is valid. This is equivalent to assuming that the bulk viscosity is zero [see Schlicting (4)].

Following Steger (5), the Navier-Stokes equations may be written in vector form for two-dimensional flow as:

$$\frac{\partial}{\partial t} q + \frac{\partial}{\partial x} E + \frac{\partial}{\partial y} F = \frac{\partial}{\partial x} R + \frac{\partial}{\partial y} S \qquad (2.1)$$

where

$$q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{bmatrix} \qquad E = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E_t + p) \end{bmatrix} \qquad F = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E_t + p) \end{bmatrix}$$

$$R = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ R_4 \end{bmatrix} \qquad S = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ S_4 \end{bmatrix}$$

and

$$E_t = \rho(e + \tfrac{1}{2} U^2)$$

$$\tau_{xx} = (\lambda + 2\mu) \frac{\partial}{\partial x} u + \lambda \frac{\partial}{\partial y} v$$

$$\tau_{xy} = \mu(\frac{\partial}{\partial y} u + \frac{\partial}{\partial x} v)$$

$$\tau_{yy} = (\lambda + 2\mu) \frac{\partial}{\partial y} v + \lambda \frac{\partial}{\partial x} u$$

$$R_4 = u\tau_{xx} + v\tau_{xy} + k P_R^{-1}(\gamma - 1)^{-1} \frac{\partial}{\partial x} a^2$$

$$S_4 = u\tau_{xy} + v\tau_{yy} + k P_R^{-1}(\gamma - 1)^{-1} \frac{\partial}{\partial y} a^2$$

Note that each line of the vector equation corresponds to one of the conservation equations. The first line corresponds to the continuity equation, while the second, third, and fourth lines correspond to the conservation equation for X-momentum, Y-momentum, and energy, respectively. This form of the Navier-Stokes equations is termed the "strong conservation" form; when coupled with appropriate difference forms local and global conservation of mass, momentum, and energy is assured.

2.2 Non-Dimensionalization for the Navier-Stokes Equations

To non-dimensionalize the equation system of Section 2.1, the following non-dimensional quantities are defined:

$$\rho' = \frac{\rho}{\rho_o} \qquad u' = \frac{u}{a_o} \qquad v' = \frac{v}{a_o}$$

$$l' = \frac{l}{l_o} \qquad t' = t\,\frac{a_o}{l_o} \qquad T' = \frac{T}{T_o}$$

$$p' = \frac{p}{\rho_o a_o^2} \qquad e' = \frac{e}{a_o^2} \qquad E_t' = \frac{E_t}{\rho_o a_o^2}$$

$$\mu' = \frac{\mu}{\mu_o} \qquad k' = \frac{k}{k_o} \tag{2.2}$$

where ρ_o = stagnation density at reference conditions.

a_o = stagnation speed of sound at reference conditions.

l_o = reference length.

T_o = stagnation temperature at reference conditions.

μ_o = reference viscosity.

k_o = reference thermal diffusivity.

Using these non-dimensional quantities, the strong conservation form becomes:

$$\frac{\partial}{\partial t} q' + \frac{\partial}{\partial x} E' + \frac{\partial}{\partial y} F' = \frac{1}{Re_o} \left[\frac{\partial}{\partial x} R' + \frac{\partial}{\partial y} S' \right] \tag{2.3}$$

where $Re_o = \frac{\rho_o u_o l_o}{\mu_o}$, and q' , E' , F' , R' , and S' have the same form as q , E , F , R , S , except that the non-dimensional variables are used in their formation.

The non-dimensional variables can be combined with the equations of state and energy and certain definitions to obtain several useful relations. For example, when non-dimensional variables are used,

$$p = \rho RT \quad \text{becomes} \quad p' = \frac{\rho' T'}{\gamma} , \tag{2.4}$$

$$e = C_v T \quad \text{becomes} \quad e' = \frac{T'}{\gamma(\gamma - 1)} , \tag{2.5}$$

and $a^2 = \gamma RT$ becomes $a^2 = T'$.

Other useful relations are:

$$p' = (\gamma - 1)\rho' e' = (\gamma - 1)(E_t' - \frac{1}{2}\rho' U'^2) \tag{2.6}$$

$$M = M' = \frac{u'}{a'} \tag{2.7}$$

$$R_e = Re_o Re' \tag{2.8}$$

and $$C_f = \frac{1}{Re_o} C_f' \tag{2.9}$$

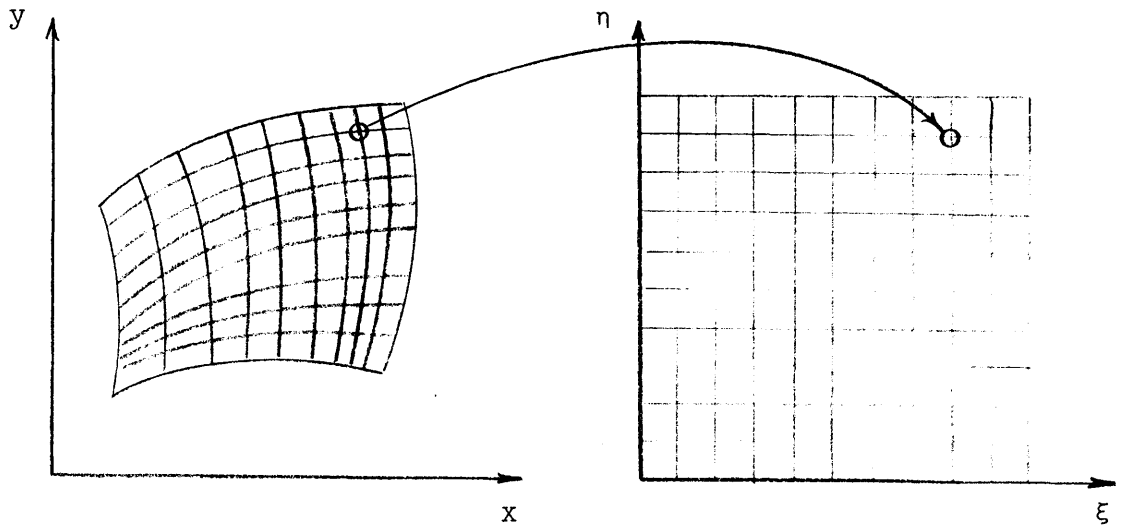
Because the reference conditions chosen correspond to upstream stagnation conditions, it can be shown that the non-dimensional upstream total pressure is $\frac{1}{\gamma}$ for all cases, or

$$p'_t = \frac{1}{\gamma} \tag{2.10}$$

It should be noted that if the Prandtl number (P_R) and the specific heat (C_p) both are considered constant, then there is an identity between the non-dimensional viscosity and thermal diffusivity.

2.3 Grid Systems

The strong conservation form of the Navier-Stokes equations written in vector form for two spacial dimensions (x,y) is written as equation (2.1). If new independent variables are defined which transform the physical coordinates (x,y) into a set of "computational" coordinates (ξ,η), a strong conservation form of the equation may be maintained [see Steger (5) or Appendix A1].



Subject to the general transformation

$$\xi = \xi(x,y) \qquad \eta = \eta(x,y) \tag{2.11}$$

the Navier Stokes equations can be written as

$$\frac{\partial}{\partial t}\hat{q} + \frac{\partial}{\partial \xi}\hat{E} + \frac{\partial}{\partial \eta}\hat{F} = \frac{1}{Re_o}\left[\frac{\partial}{\partial \xi}\hat{R} + \frac{\partial}{\partial \xi}\hat{S}\right] \tag{2.12}$$

where

$$J = \left(\frac{\partial}{\partial x}\xi\right)\left(\frac{\partial}{\partial y}\eta\right) - \left(\frac{\partial}{\partial y}\xi\right)\left(\frac{\partial}{\partial x}\eta\right)$$

$$\hat{q} = \frac{q'}{J}$$

$$\hat{E} = \frac{1}{J}\left[\left(\frac{\partial}{\partial x}\xi\right) E' + \left(\frac{\partial}{\partial y}\xi\right) F'\right]$$

$$\hat{F} = \frac{1}{J}\left[\left(\frac{\partial}{\partial x}\eta\right) E' \quad \left(\frac{\partial}{\partial y}\eta\right) F'\right]$$

$$\hat{R} = \frac{1}{J}\left[\left(\frac{\partial}{\partial x}\xi\right) R' \quad \left(\frac{\partial}{\partial y}\xi\right) S'\right]$$

$$\hat{S} = \frac{1}{J}\left[\left(\frac{\partial}{\partial x}\eta\right) R' \quad \left(\frac{\partial}{\partial y}\eta\right) S'\right]$$

A proof that this form is equivalent to equation (2.1), can be found in Appendix A-1. The computational coordinates may be defined such that they form a rectangular grid in which $\Delta\xi = \Delta\eta = 1$. A solution scheme using this set of equations does not require analytic functions defining the transformation functions $\xi(x,y)$ and $\eta(x,y)$. The solution of this equation set requires only that the metrics $\left(\frac{\partial}{\partial x}\xi \ , \ \frac{\partial}{\partial y}\xi \ , \ \frac{\partial}{\partial x}\eta \ , \ \frac{\partial}{\partial y}\eta\right)$ be determined. These are easily calculated from the derivatives $\left(\frac{\partial}{\partial \xi}x \ , \ \frac{\partial}{\partial \eta}x \ , \ \frac{\partial}{\partial \xi}y \ , \ \frac{\partial}{\partial \eta}y\right)$ which are themselves calculated by finite difference approximations using only the physical location of the mesh points, for example:

$$\frac{\partial}{\partial \eta}y = \frac{1}{2}\left(Y_{J,k+1} - Y_{J,k-1}\right)/\Delta\eta \quad . \tag{2.13}$$

The following relations are used to compute the metrics from the derivatives

$$\left(\frac{\partial}{\partial \xi}x \ , \ \frac{\partial}{\partial \eta}x \ , \ \frac{\partial}{\partial \xi}y \ , \ \frac{\partial}{\partial \eta}y\right) \quad .$$

$$\frac{\partial}{\partial x}\xi = J\frac{\partial}{\partial \eta}y \qquad \frac{\partial}{\partial y}\xi = -J\frac{\partial}{\partial \eta}x \tag{2.14 A-D}$$

$$\frac{\partial}{\partial x}\eta = -J\frac{\partial}{\partial \xi}y \qquad \frac{\partial}{\partial y}\eta = J\frac{\partial}{\partial \xi}x$$

The form (2.12) of equation (2.1) allows very general solution schemes to be written because the equations are independent of the geometry chosen. Changing geometry requires only that the metrics $\left(\frac{\partial}{\partial x}\xi\ ,\ \frac{\partial}{\partial y}\xi\ ,\ \frac{\partial}{\partial x}\eta\ ,\ \frac{\partial}{\partial y}\eta\right)$ be changed.

2.4 Thin Shear Layer Approximation

High Reynolds number flows have a tendency to restrict areas of significant viscous stresses to regions of small spacial extent, e.g., boundary layers. Generally, it is difficult and expensive to resolve all viscous stress terms in these areas, and some assumption about the flow must be introduced to make calculations of practical value. For these "thin shear layer" flows, the important stress terms are those normal to the flow direction. In most cases, only these stress terms need be accurately resolved. To take advantage of this behavior, we space enough grid points normal to the viscous regions to resolve the stear stress terms, while spacing points in the streamwise direction approximately one layer thickness apart. Because such a grid cannot resolve the streamwise stress terms and they are assumed small, all viscous derivatives in the streamwise direction are neglected. The terms normal to the viscous region are retained. This is known as the "thin shear layer" approximation.

For flows with solid boundaries, the viscous regions are aligned with the boundaries. We may define the transformation to computational space such that it maps solid surfaces

to η = constant surfaces, making ξ the streamwise direction. Neglecting the (ξ) viscous terms, equation (2.12) simplifies

$$\text{to} \quad \frac{\partial}{\partial t}\hat{q} + \frac{\partial}{\partial \xi}\hat{E} + \frac{\partial}{\partial \eta}\hat{F} = \frac{1}{Re_o}\frac{\partial}{\partial \eta}\hat{S} \tag{2.15}$$

where $\hat{q}$, $\hat{E}$, and $\hat{F}$ are defined in Section 2.3, and $\hat{S}$ is re-defined as:

$$\hat{S} = \begin{bmatrix} 0 \\ \mu\left[\left(\frac{\partial}{\partial x}\eta\right)^2 + \left(\frac{\partial}{\partial y}\eta\right)^2\right]\frac{\partial}{\partial \eta}u + \frac{\mu}{3}\left(\frac{\partial}{\partial x}\eta\right)\left[\left(\frac{\partial}{\partial x}\eta\right)\left(\frac{\partial}{\partial \eta}u\right)+\left(\frac{\partial}{\partial y}\eta\right)\left(\frac{\partial}{\partial \eta}v\right)\right] \\ \mu\left[\left(\frac{\partial}{\partial x}\eta\right)^2 + \left(\frac{\partial}{\partial y}\eta\right)^2\right]\frac{\partial}{\partial}u + \frac{\mu}{3}\left(\frac{\partial}{\partial y}\right)\left[\left(\frac{\partial}{\partial x}\right)\left(\frac{\partial}{\partial \eta}u\right)+\left(\frac{\partial}{\partial y}\right)\left(\frac{\partial}{\partial \eta}v\right)\right] \\ kP_R^{-1}(\gamma-1)^{-1}\left[\left(\frac{\partial}{\partial x}\eta\right)^2+\left(\frac{\partial}{\partial y}\eta\right)^2\right]\frac{\partial}{\partial \eta}a^2 \quad \frac{1}{2}\mu\left[\left(\frac{\partial}{\partial x}\eta\right)^2\left(\frac{\partial}{\partial y}\eta\right)^2\right]\frac{\partial}{\partial \eta}(u^2+v^2) \\ + \frac{1}{6}\mu\left\{\left(\frac{\partial}{\partial x}\eta\right)^2\left(\frac{\partial}{\partial \eta}u^2\right)+\left(\frac{\partial}{\partial y}\eta\right)^2\frac{\partial}{\partial \eta}v^2+\left(2\frac{\partial}{\partial x}\eta\right)\left(\frac{\partial}{\partial y}\eta\right)\frac{\partial}{\partial \eta}(uv)\right\} \end{bmatrix}$$

For flows with solid boundaries, the thin shear layer approximation drops the same terms which are dropped in boundary layer theory. However, the thin shear layer approximation retains the η-momentum equation, and a constant pressure is not imposed through the viscous region. Unlike boundary layer theory, there is no need to match the inviscid and viscous regions, and there is no singularity at separation points.

2.5 Approximate Factorization

The thin shear layer equations derived in the previous sections may be descretized in time to obtain a large matrix equation involving derivatives in both ξ and η. Approximate factorization separates this matrix equation into a series of equations, each of which contains ξ derivatives or η derivatives, but not both. The time descretization and factorization are described in the next sections.

2.5.1 Time Descretization

A generalized time differencing formula may be defined as:

$$\Delta\hat{q}^{n} = \frac{\theta\Delta t}{1+\varepsilon}\frac{\partial}{\partial t}\Delta\hat{q}^{n} + \frac{\Delta t}{1+\varepsilon}\frac{\partial}{\partial t}\hat{q}^{n} + \frac{\varepsilon}{1+\varepsilon}\Delta\hat{q}^{n-1} + O[(\theta-\varepsilon-\tfrac{1}{2})\Delta t^{2} + \Delta t^{3}] \tag{2.16}$$

where $\Delta\hat{q}^{n} = \hat{q}^{n+1} - \hat{q}^{n}$ [see Beam and Warming (6)]. This difference equation is equivalent to many common time differencing equations. The parameters θ and ε determine the type of scheme and its accuracy. For example, $\theta = 1$, $\varepsilon = 0$ reduces equation (2.16) to the common Euler implicit difference formula:

$$\hat{q}^{n+1} - \hat{q}^{n} = \Delta t\left(\frac{\partial}{\partial t}\hat{q}^{n+1}\right) + O(\Delta t^{2}) \tag{2.17}$$

Other choices of θ and ε are listed in table 2.5.1* , along with the differencing scheme they represent.

*See Beam and Warming (6).

Table 2.5.1 - Partial list of schemes contained in Eq.(2.16)

Θ	ε	SCHEME	ERROR
0	0	Euler explicit	$O(\Delta t^2)$
0	$-\frac{1}{2}$	Leapfrog explicit	$O(\Delta t^3)$
$\frac{1}{2}$	0	Trapezoidal implicit	$O(\Delta t^3)$
1	0	Euler implicit	$O(\Delta t^2)$
1	$\frac{1}{2}$	3-point-backward implicit	$O(\Delta t^3)$

The thin shear layer equation may be written as:

$$\frac{\partial}{\partial t}\hat{q} + \frac{\partial}{\partial \xi}\hat{E} + \frac{\partial}{\partial \eta}\hat{F} = \frac{1}{Re_o}\frac{\partial}{\partial \eta}\hat{S} \qquad (2.18)$$

as described in Section 2.4. This equation can be solved simply for $\frac{\partial}{\partial t}\hat{q}$ to obtain:

$$\frac{\partial}{\partial t}\hat{q} = -\frac{\partial}{\partial \xi}\hat{E} + \frac{\partial}{\partial \eta}\left(-\hat{F} + \frac{1}{Re_o}\frac{\partial}{\partial \eta}\hat{S}\right) \quad . \qquad (2.19)$$

This relation is substituted into the time descretization equation (2.16) to obtain:

$$\Delta\hat{q}^{n} = \frac{\theta\Delta t}{1+\varepsilon}\left[-\frac{\partial}{\partial\xi}\Delta\hat{E}^{n} + \frac{\partial}{\partial\eta}\left(-\Delta\hat{F}^{n} + \frac{1}{Re_o}\Delta\hat{S}^{n}\right)\right]$$

$$+ \frac{\Delta t}{1+\varepsilon}\left[-\frac{\partial}{\partial\xi}\hat{E}^{n} + \frac{\partial}{\partial\eta}\left(-\hat{F}^{n} + \frac{1}{Re_o}\hat{S}^{n}\right)\right]$$

$$+ \frac{\varepsilon}{1+\varepsilon}\Delta\hat{q}^{n-1} + O\left[\left(\theta - \varepsilon - \frac{1}{2}\right)\Delta t^{2} + \Delta t^{3}\right] \qquad (2.20)$$

At any location (ξ,η), $\hat{E}$ and $\hat{F}$ are functions of $\hat{q}$ only, as can be seen from their definitions (Sections 2.1 and 2.3). Thus, they are differentiable in $\hat{q}$ and the matrices $\hat{A}$ and $\hat{B}$ may be defined as:

$$\hat{A} \equiv \frac{\partial}{\partial\hat{q}}\hat{E} \qquad (2.21.A)$$

$$\hat{B} \equiv \frac{\partial}{\partial\hat{q}}\hat{F} \qquad (2.22.B)$$

These definitions imply the matrix relations:

$$\Delta\hat{E} = \hat{A}\,\Delta\hat{q} \qquad (2.21.B)$$

$$\Delta\hat{F} = \hat{B}\,\Delta\hat{q} \qquad (2.22.B)$$

We note that $\hat{S}$ is more complex than $\hat{E}$ or $\hat{F}$ in that it is a function not only of $\hat{q}$ but also $\frac{\partial}{\partial \eta}\hat{q}$ (see Section 2.4). However, the form of $\hat{S}$ is still such that $\hat{M}$ may be defined where:

$$\hat{M} \equiv \frac{\partial}{\partial \hat{q}} \hat{S} \tag{2.23.A}$$

as explained by Steger (5). This implies the relation:

$$\Delta \hat{S} = \hat{M} \, \Delta \hat{q} \tag{2.23.B}$$

These relations may be used in (2.20) to obtain:

$$\left[I + \frac{\theta \Delta t}{1+\varepsilon} \frac{\partial}{\partial \xi} \hat{A}^{n} + \frac{\theta \Delta t}{1+\varepsilon} \frac{\partial}{\partial \eta} \left(\hat{B}^{n} - \frac{1}{Re_o} \hat{M}^{n} \right) \right] \Delta \hat{q}^{n}$$

$$= \frac{\Delta t}{1+\varepsilon} \left[-\frac{\partial}{\partial \xi} \hat{E}^{n} + \frac{\partial}{\partial \eta} \left(-\hat{F}^{n} + \frac{1}{Re_o} \hat{S}^{n} \right) \right]$$

$$+ \frac{\varepsilon}{1+\varepsilon} \Delta \hat{q}^{n-1} + O[(\theta - \varepsilon - \tfrac{1}{2})\Delta t^{2} + \Delta t^{3}] \tag{2.24}$$

This is the time descretized form of the thin shear layer equations. The time accuracy is variable depending on the choice of the parameters which determine the differencing scheme, θ and ε. This form is known as the "delta formulation" because the dependent variable is $\Delta \hat{q}$. The procedure used to march from time level (n) to (n + 1) is to solve for $\Delta \hat{q}$, and then add it to $\hat{q}$.

2.5.2 Factoring the Time Descretized Equation

Given the solution at time level (n), all the quantities on the right-hand side of equation (2.24) may be computed to any desired accuracy, if given appropriate difference expressions. Thus, the right-hand side can be considered a "known" quantity and will henceforth be abbreviated "R.H.S."

The operator acting on $\Delta\hat{q}$ in equation (2.24) may be split into two operators by adding a term of order $(\Delta t)^2$ in $\Delta\hat{q}$ (Δt^3 in $\hat{q}$). The split equation is:

$$\left[I + \frac{\theta\Delta t}{1+\varepsilon}\frac{\partial}{\partial\xi}\hat{A}^n\right]\left[I + \frac{\partial\Delta t}{1+\varepsilon}\frac{\partial}{\partial\eta}\left(\hat{B}^n - \frac{1}{Re_o}\hat{M}^n\right)\right]\Delta\hat{q}^n = \text{R.H.S.} \quad (2.25)$$

Note that this factored equation differs from the original equation by

$\left(\frac{\theta\Delta t}{1+\varepsilon}\right)^2\left(\frac{\partial}{\partial\xi}\hat{A}^n\right)\left(\frac{\partial}{\partial\eta}\left[\hat{B}^n - \frac{1}{Re_o}\hat{M}^n\right]\right)\Delta\hat{q}^n$. This term is the same order as the terms dropped in the original time descretization and so does not affect the formal accuracy of the scheme.

This factored equation may be solved in two parts. If we define

$$\Delta q^* = \left[I + \frac{\theta\Delta t}{1+\varepsilon}\frac{\partial}{\partial\eta}\left(\hat{B} - \frac{1}{Re_o}\hat{M}^n\right)\right]\Delta\hat{q}^n \quad (2.26)$$

then solving for $\Delta\hat{q}^n$ becomes a two-step process:

A. Solve for $\Delta\hat{q}^n$ from:

$$\left[I + \frac{\theta\Delta t}{1+\epsilon}\frac{\partial}{\partial\xi}\hat{A}^n\right]\Delta q^* = \text{R.H.S.} \qquad (2.27.A)$$

B. Solve for $\Delta\hat{q}^n$ from the definition of Δq^*:

$$\left[I + \frac{\theta\Delta t}{1+\epsilon}\frac{\partial}{\partial\eta}\left(\hat{B}^n - \frac{1}{Re_o}\hat{M}^n\right)\right]\Delta\hat{q}^n = \Delta q^* \qquad (2.27.B)$$

These equations are block tridiagonal matrices when the difference operators are expanded. These equations have the form:

$$\begin{bmatrix} B_1 & C_1 & & & & \\ A_2 & B_2 & C_2 & & & \\ & A_3 & B_3 & C_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & A_{J-1} & B_{J-1} & C_{J-1} \\ & & & & A_J & B_J \end{bmatrix} \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \\ \vdots \\ \Delta q_{J-1} \\ \Delta q_J \end{bmatrix} = \begin{bmatrix} rhs_1 \\ rhs_2 \\ rhs_3 \\ \vdots \\ rhs_{J-1} \\ rhs_J \end{bmatrix} \qquad (2.28)$$

where A_j , B_j , C_j are 4x4 matrices ('blocks') and Δq_j and rhs_j are 4 element vectors.

Notice that each equation now contains ξ or η derivatives, but not both. This uncouples the two directions and reduces the dynamic memory required and the operation count required

for a solution to the equations, as described in the next section.

2.5.3 Advantages of the Factored Scheme

Equation (2.24) is a matrix equation for the vector $\Delta\hat{q}^n$. Note that $\hat{A}$, $\hat{B}$, and $\hat{M}$ are (4x4) matrices and that $\Delta\hat{q}$ is a large vector made up of (1x4) matrices. If the computational domain has J points in ξ and K points in η, the size of the vector $\Delta\hat{q}$ is (4JK). The size of the matrix multiplying $\Delta\hat{q}$ is (4JK) x (4JK). Inverting a matrix of this size requires at least $(4JK)^2$ storage locations and $(4JK)^3$ operations. A typical problem with 100 grid points in each direction would require 1.6×10^9 storage locations to store the matrix.

An alternative to solving this large matrix equation is found in the approximate factorization technique described in Section 2.5.2 . The factorization separates the matrix equation into two equations, one with ξ derivatives and one with η derivatives. Using central difference schemes for the spacial operators reduces each equation to a block tridiagonal matrix equation. The storage requirement for each tridiagonal matrix is 3 x (4JK). The storage may be reduced further by noting that in the equation containing only ξ derivatives, each line of constant η (K = const) is independent (similarly for the η equation, lines of constant J are independent). Thus, each line may be solved independently, and the storage requirement for each matrix equation is only 3 x (4J). For

J = K = 100, the dynamic memory requirement has been reduced from 1.6×10^9 to 1200. The factorization has reduced a single (4JK) x (4JK) matrix equation to (J + K) equations using 3 x (4J) matrices.

The factored scheme also greatly reduces the operation count. Inverting the large (4JK) x (4JK) matrix would require on the order of $(4JK)^3$ operations. Each of the block tri-diagonal matrices of the factored scheme requires only $J(4)^3$ operations [see Issacson and Keller (7)], or $(J + K)\ J(4)^3$ operations for the entire equation set. For the example of J = K = 100, the operation count is reduced from 6.4×10^{13} to 1.3×10^6 operations.

Thus, the factored schemes have reduced both the operation count and the dynamic memory requirement. Besides the obvious advantage of reducing run time, the factored schemes have broken the equation into smaller sections which enable large programs to be programmed and run on inexpensive "mini" computers. Implementation of the scheme on minicomputers is described in Section 2.7 .

2.6 Stability and Smoothing

A stability analysis of the factored scheme derived for the thin shear layer equations in Section 2.5, would be quite intractable. However, we can gain insight to the stability of the scheme by studying the model linear one-dimensional wave equation

$$\frac{\partial}{\partial t}u + a\frac{\partial}{\partial x}u = 0 \qquad (2.28)$$

When written in a delta formulation, it has a form similar to each operator of the factored scheme of Section 2.5. A Von-Newmann stability analysis of this equation shows it to be neutrally stable for all Courant numbers ($u\,\Delta t/\Delta x$) (see Appendix A2).

The operators of equation (2.27) actually have non-linear terms, and experience shows that it is necessary to add dissipative terms to damp the growth of high frequency (short wavelength) oscillations. The smoothing terms presently employed are fourth derivatives acting on $\hat{q}$, multiplied by a constant and Δt. These terms, explained by Pulliam and Steger in reference (8), are higher order and appended explicitly to the right-hand side of the equation. The smoothing terms are multiplied by Δt to assure consistency with $\Delta\hat{q}$. Fourth derivative terms are used because they are small except where $\hat{q}$ varies rapidly, that is, near the short wavelength disturbances which are unstable.

These explicit smoothing terms introduce a stability limit on the scheme which can be alleviated by adding similar implicit smoothing terms to the operators of (2.27). The implicit smoothing terms are second-derivative terms acting on $\Delta\hat{q}$, again multiplied by Δt to assure consistency. The addition of these implicit terms regains the unlimited stability of the original scheme [see Pulliam and Stegar (8)].

The added terms are higher order ($0\ (\Delta t\ \Delta x^2\ \frac{\partial^3}{\partial x^2 \partial t}\ \hat{q})$ and so do not formally affect the accuracy of the scheme. However, if the constant multiplying the terms is large $\left(0\ \frac{1}{\Delta t \Delta x^2}\right)$, the time accuracy has been found to be disturbed. This time error is such that the solution approaches the steady state at a faster rate. Thus, it can be beneficial if steady state solutions are sought. However, if time accurate solutions are sought, this error is not acceptable, and care must be taken that the smoothing terms are not so large as to affect the time accuracy. Unfortunately, reducing the smoothing terms may lead to spacial oscillations which do not die out in time. The trade-off between spacial and time accuracy should be studied before time accurate solutions are pursued.

2.7 Operation on Mini-Computers

Many of the inexpensive "mini" computers in use today have a limited amount of storage accessible to a program. For fluid dynamic calculations which often require many grid points, and therefore a large amount of storage, this restriction can limit the effectiveness of mini-computers and force the use of large, relatively expensive, super computers. The factored scheme described in Section 2.5 overcome the size restriction imposed by mini-computers by breaking the calculation into smaller sections. Large regions can be computed by

storing the large amount of data on an external device and accessing this data one computational line at a time.

The present code runs on a Digital Equipment Corp. PDP 11-70, which has a maximum program size of 64K bytes. The "external" storage device used is the four million bytes of extended memory available on this machine. This memory has a rapid data transfer rate and is therefore preferable to relatively slow disk or tape drives, but must be accessed through the memory management hardware by subroutines which "map" a portion of the memory region into the program address area. The mapping subroutines essentially point 4K words of memory to an area in the program.

The present version of the code allows grid lines of 90 nodes, that is, computational domains which have 90 points in each coordinate direction. This is more than sufficient for the examples we will use to test the algorithm.

2.8 Execution Time of the Code

The current implementation of this algorithm requires 115 seconds to advance the calculation one time step, on a 41 x 61 grid. Approximately 80% of the run time is used to invert the block matrices defined in section 2.5.2. For a 41 x 61 grid there are 41 inversions of a 3 x 244 matrix and 61 inversions of a 3 x 164 matrix (see section 2.5.3). This corresponds to approximately .091 seconds/node used during the matrix inversion, on the PDP 11-70.

3.0 Boundary Conditions

The thin shear layer equations usually require flow conditions to be specified on all boundaries. The mathematical conditions required generally are in terms of the flow properties or their derivatives. The numerical approximations require information about all four variables which constitute $\hat{q}$, (ρ , ρu , ρv , E_t). However, the boundary conditions used need not directly set ρ , ρu , ρv , and E_t , or their derivatives, as any four independent conditions can be used to determine $\hat{q}$. Often physical considerations, such as the domain of dependence, limit the number of physical boundary conditions to less than the four required numerically. For such cases, the remaining variables required numerically, must depend on conditions inside the computational domain and are calculated from interior points.

In external flow problems, typically, the flow tends asymptotically to freestream conditions far from the body, making the specification of the physically proper boundary conditions relatively simple. However, in internal flow problems where the upstream and downstream flow conditions generally are not known until the solution is known, prescribing boundary conditions becomes more difficult. Choosing the conditions to prescribe can be done by carefully modeling the physics of the flow and experimenting with reasonable boundary conditions.

The variables which are to be determined from points in-

side the calculation domain may be set implicitly or explicitly. Setting the boundary conditions implicitly imples that they are set during the finite difference equation solution for the interior points. Thus, implicit boundary conditions alter the basic solution scheme and make a change of boundary conditions difficult to implement. Setting the boundary conditions explicitly uses the solution of the previous time step to compute the boundary conditions for the next iteration. This type of boundary conditions becomes a modular element in a program, which is changed easily. The explicit boundary conditions reduce the time accuracy at the boundaries to first order, but their versatility make them preferable to implicit schemes for purposes of testing and validating the factored schemes.

3.1 Upstream Boundary Conditions

The inviscid diffuser examples to be considered have uniform supersonic flow upstream. Supersonic flow allows no information to travel upstream and all flow quantities may be specified. The upstream boundary is fixed at freestream conditions at all times.

The viscous diffuser problems to be considered have a supersonic core flow between two parallel flat walls at the inlet to the calculation domain. If the walls are far apart, the effect of the opposite wall is considered negligible and viscous regions near the walls should behave as standard

boundary layers. The equations governing boundary layer flow are parabolic and may be marched forward in space. Because we expect boundary layer-like flow, we specify all four flow quantities through the upstream boundary layers, and since supersonic points may be marched forward in space also, we specify all four flow quantities across the entire inlet region. It should be noted that, in general, subsonic regions near the upstream boundary have some upstream influence. This suggests that some flow variable should not be specified but determined from inside the computational domain. It is only because we expect parabolic behavior similar to the boundary layer equations that we do not need to "float" one of the variables in the subsonic region of the upsteam boundary layers. The inlet boundary layers were approximated by laminar flat plate boundary layers, with a zero freestream pressure gradient. The profiles were generated on a computational grid by Usab (9).

The cascade problems to be considered have uniform supersonic flow upstream. However, the upstream boundary (figure 1) is not perpendicular to the flow direction and the component of velocity normal to this boundary may not be supersonic. In this case, information may travel upstream and one flow variable should be determined from inside the computational domain. If the boundary is far from the blade sections, we expect any disturbances reaching the boundary to be small, if the compressor is operating close to design. Thus for the calculations done, the boundary is held fixed

throughout the calculation.

3.2 Wall Boundary Conditions

In inviscid flow, the flow is tangent to a wall. The magnitude of the velocity is taken to be the magnitude of the total velocity one node off the wall. For viscous flows, the condition of no-slip requires that the velocities at the wall be zero. The energy at the wall can be determined by requiring that the walls be adiabatic, that is, the change in total internal energy is zero normal to a wall.

A fourth flow variable may be determined from a condition on the pressure at the wall. One such condition is to set the derivative of the pressure normal to the wall to zero. For viscous calculations with grids not normal to the wall, this can be done by noting that the velocity at the wall is zero, and combining the ξ and η momentum equations into a single equation for the pressure. Following the work of Steger (5), we take $\frac{\partial}{\partial x}\eta$(ξ -momentum equation) plus $\frac{\partial}{\partial y}\eta$ (η - momentum equation) and solve the resulting tri-diagonal matrix equation for the pressure. After setting the wall velocity to zero, this equation becomes:

$$\left[\left(\frac{\partial}{\partial x}\eta\right)\left(\frac{\partial}{\partial x}\xi\right) + \left(\frac{\partial}{\partial y}\eta\right)\left(\frac{\partial}{\partial y}\xi\right)\right]\frac{\partial}{\partial \xi}p + \left[\left(\frac{\partial}{\partial x}\eta\right)^2 + \left(\frac{\partial}{\partial y}\eta\right)^2\right]\frac{\partial}{\partial \eta}p = 0 \qquad (3.1)$$

In finite difference form this equation is:

$$\left[\left(\frac{\partial}{\partial x}\eta\right)\left(\frac{\partial}{\partial x}\xi\right) + \left(\frac{\partial}{\partial y}\eta\right)\left(\frac{\partial}{\partial y}\xi\right)\right] \frac{1}{2\Delta\xi}\left(P_{j+1,k} - P_{j-1,k}\right)$$

$$-\left[\left(\frac{\partial}{\partial x}\eta\right)^2 + \left(\frac{\partial}{\partial y}\eta\right)^2\right]\frac{1}{\Delta\eta} P_{j,k} = -\left[\left(\frac{\partial}{\partial x}\eta\right)^2 + \left(\frac{\partial}{\partial y}\eta\right)^2\right]\frac{1}{\Delta\eta} P_{j,k+1}$$

(3.2)

The equations for each j along a wall can be combined into a single matrix equation which can be solved by Gaussian Elimination.

This set of boundary conditions determines the four flow quantities (u , v , e , p) at the walls, which uniquely determines $\hat{q} = (\rho , \rho u , \rho v , E_t)$.

3.3 Downstream Boundary Conditions

Supersonic downstream boundaries have all information coming from inside the computational domain and no flow properties are to be prescribed. Subsonic outflow boundaries require that one, and only one, boundary condition be prescribed. This is usually a condition on the pressure.

The easiest and most obvious way to determine the flow quantities which are to be set by the interior points is to extrapolate them along grid lines. This type condition has been used with the algorithm and often gives reasonable results. However, they may lead to errors in the steady state

solution because extrapolation techniques set the first or second derivative of the extrapolated variable to zero, which may not be a correct condition.

An alternative to the extrapolation techniques can be found by examining the governing equation (2.24). Written for a trapezoidal formulation ($\theta = \frac{1}{2}$, $\varepsilon = 0$) the difference equation (2.24) becomes:

$$\left\{ I + \frac{1}{2}\Delta t \left[\frac{\partial}{\partial \xi}\hat{A}^{n} + \frac{\partial}{\partial \eta}\left(\hat{B}^{n} - \frac{1}{Re_o}\hat{M}^{n} \right) \right] \right\} \Delta \hat{q}^{n}$$

$$= \Delta t \left\{ - \frac{\partial}{\partial \xi}\hat{E}^{n} + \frac{\partial}{\partial \eta}\left(-\hat{F}^{n} + \frac{1}{Re_o}\hat{S}^{n} \right) \right\} + O(\Delta t^{3}) \tag{3.3}$$

where $\hat{E}$, $\hat{F}$, etc. are defined in Chapter 2. This matrix equation could be solved directly on the outflow boundary, except that an expression for $\frac{\partial}{\partial \xi}\hat{A}^{n}$ is needed. If we lag this term [setting $\left(\frac{\partial}{\partial \xi}\hat{A}^{n} \right) \Delta \hat{q}^{n} = \left(\frac{\partial}{\partial \xi}\hat{A}^{n} \right) \Delta \hat{q}^{n-1}$] and move it to the right-hand side, this implicit operator becomes identical to the implicit η operator used in the basic solution scheme, equation (2.27). The only modification is done on the "known" right-hand side. Lagging this term does not affect the formal time accuracy because

$$\Delta \hat{q}^{n} = \Delta \hat{q}^{n-1} + O(\Delta t^{2}) \tag{3.4}$$

However, lagging the ξ derivative does affect the stability of the scheme. For example, if a VonNeumann stability analysis is done for a 2-D model wave equation

$$\frac{\partial}{\partial t} u + a \frac{\partial}{\partial x} u + b \frac{\partial}{\partial y} u = 0 \qquad (3.5)$$

solved using this scheme, the scheme is shown to be only conditionally stable (see Appendix A3).

3.4 Periodic Boundary Conditions

Cascade geometries require a periodic condition up and downstream of the blade section. This condition simulates an infinite array of blades by assuring that the flow repeats itself, that is, the conditions at the bottom of a blade passage are repeated at the top. An implicit periodicity condition is easily encorporated into this solution scheme by using a periodic block inverter in equation (2.27.B). Since these "periodic" solvers require twice as many numerical operations and twice as much storage as the simple tridiagonal solvers, the present code sets the periodic conditions explicitly, lagging them one time step.

To update the boundaries for the next iteration, the flow conditions are extrapolated across the boundary, setting them to the average of the conditions above and below the boundary. This condition should be sufficiently accurate provided that the grid is closely packed near the periodic bound-

aries and there are not large gradients in the flow properties along the boundaries.

4.0 Example Calculations

4.1 Inviscid Diffuser Calculation

The factored scheme at Chapter 2 solves the thin shear layer equations in the form:

$$\frac{\partial}{\partial t}\hat{q} + \frac{\partial}{\partial x}\hat{E} + \frac{\partial}{\partial y}\hat{F} = \frac{\partial}{\partial y}\hat{S}. \qquad (4.1)$$

The viscous terms are contained in the vector $\hat{S}$ and when $\hat{S}$ is set to zero, the equations reduce to the inviscid Euler equations. Because of the relative abundance of theoretical and computational predictions of inviscid flow fields, it is of interest to compare the results from the present solution scheme operating on the Euler equations, to other more thoroughly tester Euler equation solution schemes. These comparisons give an indication of the accuracy of the factored scheme and information about its other properties.

The internal flow through a sharp cornered diffuser is used as a test case because the theoretical solution is known and because it contains examples of nearly all of the internal flow boundary conditions. The Mach number at the inlet to the diffuser is chosen to be 2.0 and the geometry chosen to generate a pressure rise of 1.4 across a reflected oblique shock. The lower wall of the diffuser is flat and the upper wall has a compression wedge of 3.09° (.054 radians). The compression wedge generates an oblique shock at approximately 31° (.541 radians). The upper wall turns back to parallel

near where the reflected shock is expected to intersect the wall. The geometry and theoretical shock structure are shown in figure 2 .

The same computational grid was used for this example and the viscous diffuser example in the following section. The grid lines are clustered near the walls to capture the large gradients in the boundary layer expected in the viscous examples. There are 25 points spaced geometrically across the expected boundary layer thickness and 11 points spaced evenly across the rest of the channel. This grid is shown in figure 3 . If only inviscid calculations are to be done, the number of grid points across the channel can be reduced from the present 61 to 15 with minimal affect on the computed solutions.

The boundary conditions for this case assume uniform flow through the upstream boundary. The wall conditions assume an adiabatic wall $\left(\frac{\partial T}{\partial n} = 0\right)$ and a pressure gradient of zero normal to the wall $\left(\frac{\partial P}{\partial n} = 0\right)$. The velocity at the wall is tangent to the wall and equal to the total velocity one node off the wall. The downstream boundary conditions are determined as described in section 3.3, except that the viscous terms are set to zero. This boundary condition imposes a Courant number restriction of .5 on the calculation, but gives an accurate representation of the boundary.

The criteria used to determine convergence is the root

mean square change in the density $\left(\left[\sum_{j,k}\left(\frac{\Delta\rho}{\rho}\right)^2\right]^{1/2}\right)$. This value is normalized by the time step and convergence is assumed when the value is less than 10^{-2}, that is:

$$\frac{1}{\Delta t}\left[\sum_{j,k}\left(\frac{\Delta\rho}{\rho}\right)^2\right]^{1/2} < 10^{-2} \quad . \tag{4.2}$$

This case converged after 175 iterations. At convergence, the largest fractional change in density at any point was $6. \times 10^{-5}$. The convergence history is shown in figure 4 . The run time on the PDP 11-70 is approximately 5.6 hours.

Figure 5 shows the shock structure for the converged solution in the form of a pressure contour plot. The theoretical shock locations (figure 2) are overlaid as heavy lines. The reflected shock is not completely cancelled, partly because the expansion corner is not positioned exactly where the shock intersects the upper wall, and partly because of the finite width of shocks in shock capturing type solutions.

The flow field predicted by the factored scheme can also be compared to a calculation by Tong (2) which uses an explicit, time marching scheme due to MacCormack. Figure (6) shows the pressure contours produced by the calculations by Tong. Comparison of this figure and figure (5) demonstrates the similarity of the shock structures computed by the two solution schemes. Figures (7) and (8) show the wall pres-

sure distributions computed by the two schemes. The two solutions have similar shock resolution but the factored scheme does not have low frequency waves across the diffuser which are apparent in the explicit scheme. The pressure variations on the upper wall after location .6 are due to the expansion corner and the reflected shock interacting with the wall. The pressure drop on the lower surface after location .8 is due to the upper wall expansion reaching the lower wall. The similarity of the two computed flow fields, including similar behavior at the boundaries, indicates that the factored scheme, and the present set of boundary conditions, can predict inviscid flow fields which agree with the predictions of a well tested explicit, time marching MacCormack scheme.

The present solution scheme compares well with both theory and the MacCormack solution, indicating that the factored scheme can compute solutions of the Euler equations with good accuracy. This scheme is presently slower than the MacCormack algorithm because of the time step restriction imposed by the boundary conditions. The advantage of the scheme is its ability to compute viscous flows with little additional effort.

4.2 Oblique Shock Boundary Layer Interation

Flows of general interest, which we hope to simulate, have both viscous layers and complex shock structures. In order to assure that these phenomena can be accurately predicted, computed solutions must be compared to some'known' solution. Unfortunatley, there are no theoretical predictions and few reliable experimental measurements of these phenomena. One notable exception is a series of oblique shock boundary layer experiments done by Hakkinen, et al (3). The phenomena observed and measured through the interaction region include boundary layer velocity profiles, wall pressure and skin friction distributions. Thus, these experiments incorporate many of the phenomena which are to be modeled and serve as a standard test case for viscous interaction computational models.

The shock boundary layer interaction is reproduced within a supersonic diffuser because our goal is to produce internal flow calculations and develop reliable internal flow boundary conditions. The shocks are easily generated within a diffuser with a compression wedge, as in the inviscid diffuser example. The drawback to simulating the interaction within a diffuser is that there is a finite area and mass flow through the diffuser, and there may be blockage effects due to boundary layer growth. However, this effect can be minimized and developing an accurate free-stream boundary condition is of little value for internal flow calculations.

The blockage due to boundary layer growth can be reduced by eliminating the boundary layer on the upper wall. This can be done by simply not enforcing the no-slip condition on the upper wall. The blockage due to only the lower wall is small until reverse flows are generated, substantially increasing the boundary layer thickness. The blockage tends to decrease the velocity (for supersonic flow) and increases the static pressure. As will be shown, the pressure rise due to blockage does not greatly influence the flow until downstream of the region of interest.

The experiment chosen for comparison has an inlet Mach number of 2.0 and a pressure rise of 1.4 across the reflected shock. The geometry of the inviscid diffuser example produces the correct pressure rise across the reflection and the same geometry and grid are used for this example. The Reynolds number is chosen such that the shock reflects from the lower wall at the Reynold's number corresponding to experiment ($Re_{SHOCK} = 5.96 \times 10^5$). To avoid modeling the leading edge boundary layer growth, the leading edge of the diffuser is not within the computational domain. A flat plate boundary layer profile at a Reynolds number of $.25 \times 10^5$ is input on the upstream boundary. The profile was produced by Usab (9).

This example retains all the terms in the thin shear layer equations (section 2.4). The upstream boundary condition assumes uniform flow at the test conditions except along the lower wall where the boundary layer profile is input (see

section 3.1). The walls are assumed to be adiabatic and the pressure gradient normal to the wall is zero $\left(\frac{\partial T}{\partial n} = 0, \frac{\partial P}{\partial n} = 0\right)$. The condition of no-slip is not enforced at the upper wall; there is no flow through the wall and the velocity along the wall is equal to the total velocity one node off the wall. The lower wall does enforce the no-slip condition on the velocities. The downstream boundary is the one described in section 3.3. This imposes a stability limit which requires the Courant number $\left(\frac{u\Delta t}{\Delta x}\right)$ to be less than .5 .

The convergence criteria is the same as the one used for the inviscid diffuser example, that is:

$$\frac{1}{\Delta t}\left[\sum_{j,k}\left(\frac{\Delta\rho}{\rho}\right)^2\right]^{1/2} \leq 10^{-2} \quad . \tag{4.3}$$

This case took 492 iterations to converge and the largest fraction change in density at convergence was approximately 10^{-6} . The run time on the PDP 11-70 is approximately 15.7 hours.

Figure 4 shows the convergence rate for this case and the inviscid example. The initial convergence rate is similar for the two cases, indicating that the initial shock structures are formed in a similar number of iterations. Once these shock structures are formed, the inviscid case converges rapidly. The viscous case has much slower convergence rate because of the interation of the shocks and the boundary layer. As the

boundary layer adjusts to the pressure distribution of the shocks, it disrupts the flow, causing a change in the shock structure. This interaction slows convergence.

The pressure contours computed for this test case are shown in figure 9 . These contours show a continuous pressure rise between the incident and reflected shock which is not evident in the inviscid example (figure 5). This pressure rise is caused by the blockage due to the displacement thickness of the boundary layer. The velocity profiles computed have a maximum displacement thickness which is 2.5% of the channel width. An area change of 2.5% in a Mach 2.0 flow will cause a pressure rise of roughly 10%. This is the approximate rise seen in the contours.

The blockage has a significant effect away from the wall where constant pressure is expected. However, near the wall, where the shock generates a rapid pressure rise, the blockage has a smaller effect and the contours are as one would expect for a reflected shock. The pressure rise is spread in the upstream direction as can be seen by comparing the pressure distribution of figure 10 to the inviscid pressure distribution in figure 7 . Figure 10 shows that the computed wall pressure distribution is in good agreement with experiment through the boundary layer interaction region. Near the end of the reverse flow region ($R_{ex} \approx 3.2 \times 10^5$) the blockage begins to raise the pressure. This rise is small compared to the pressure rise due to the shock reflection and does not greatly affect the skin friction coefficient until

after the reverse flow region.

Figure 10 also shows the computed and experimental skin friction coefficients. A boundary layer type calculation was also used to predict the skin friction up to separation. The boundary layer calculation used the computed free stream conditions as the edge conditions. The agreement of the boundary-layer calculation and the experiment indicate that the shock structure is well modeled outside the boundary layer. The good agreement of the computed skin friction with both the experimental results and the boundary layer calculation indicates that the viscous layers can also be predicted accurately. The experimental apparatus could not measure negative skin frictions but the edges of the reverse flow region are well defined. These points are well predicted by the calculation giving an indication that the reverse flow region is accurately modeled. The flow downstream of the reverse flow region differs from the experiment because the upper wall expansion and shock re-reflection begin to interact with the lower wall boundary layer.

The computed boundary layer profiles are also in general agreement with the experimentally determined profiles (see figure 11). The discrepancies are most likely due to insufficient boundary layer resolution and uncertainties in experimentally determining velocity profiles.

The excellent agreement between this calculation and the experimental data indicates that we can accurately pre-

dict shock structures, laminar boundary layers, and their interaction. We thus anticipate the ability to accurately compute viscous internal flow problems of more general interest.

4.3 Viscous Diffuser Calculation

The geometry of the previous examples is a typical transonic diffuser geometry. If the transonic, viscous flow through this geometry can be successfully computed, we can expect to be able to accurately predict the flow through arbitrary diffuser geometries.

The boundary conditions employed to compute this flow field are the same as those used in the oblique shock boundary layer interaction example, except that the condition of no-slip is enforced on both walls. The convergence criteria are the same used in the previous examples. That is:

$$\frac{1}{\Delta t} \left[\sum_{j,k} \left(\frac{\Delta \rho}{\rho} \right)^2 \right]^{1/2} \leq 10^{-2} \quad . \tag{4.4}$$

This example converged after 532 iterations, which translates to approximately 17 hours on the PDP 11-70. At convergence, the largest fraction change in density at any node is $2. \times 10^{-5}$. As can be seen in figure 4 , the convergence history is similar to the oblique shock boundary layer interaction example. The convergence rate is the same for both cases but the magnitude of the error is slightly higher for

this case because shocks interact with boundary layers on both walls.

Figure 12 shows the pressure contours computed for the fully viscous diffuser. This example produces a substantial pressure gradient in the channel between the incident and reflected shocks. This gradient is larger than computed in either the inviscid or oblique shock boundary layer interaction example (figures 5 and 9). This is due to the added contraction caused by the upper wall boundary layer displacement thickness.

This example demonstrates that even relatively thin boundary layers (maximum displacement thickness of each boundary layer is less than 2.5% of the channel width) can greatly influence the flow within a supersonic diffuser.

4.4 Inviscid Cascade Calculation

An internal flow problem which has received much attention in recent years is the flow through a two dimensional transonic compressor cascade. Understanding flow of this type is an important step in designing high speed axial compressor stages which have improved performance and efficiency.

The factored algorithm described in this thesis was used to predict the inviscid flow through a compressor cascade as a first step towards a viscous cascade calculation. The solution computed by this scheme can be compared to other well tested solution techniques to determine if the numerical

scheme is correct. Agreement of the schemes does not necessarily imply the correct physical solution because it is characteristic of internal flow problems that the far field conditions must be determined as part of the solution and the model used to predict the boundary conditions can affect the solution. The solution can also be affected by the way the Kutta condition is modeled. If the same numerical solution is obtained by different solution schemes, using the same boundary and Kutta condition models, then it can be inferred that the numerical algorithms are correct. However, the accuracy of the models must be proven before we assume that a computed solution corresponds to the physical solution.

The cascade calculation was done with a typical transonic compressor blade geometry. The upstream flow angle was 64.14° and the non-dimensional exit static pressure was 0.4835. The computational domain extends one chord length upstream and downstream. The grid mesh was generated by von Lavante (10) and is shown in figure 1 . This body fitted grid has lines of constant η in the streamwise direction and lines of constant ξ in the blade to blade direction. The ξ lines are normal to the blade at the blade surface and periodic upstream and downstream. There are 21 points on the blade and 10 both upstream and downstream of the blade. The grid lines are clustered near the blade with 20 points spaced geometrically from each surface and 20 evenly spaced points across the rest of the channel (see figure 1)

The solution presented was obtained after 1500 iterations. The convergence criterion used for the previous examples:

$$\frac{1}{\Delta t}\left[\sum_{jk}\left(\frac{\Delta\rho}{\rho}\right)^2\right]^{1/2}$$

was approximately 2.5 and the largest fractional change in density at any node was 10^{-4} . The solution was plotted every one hundred iterations and the blade surface pressures were found to change less than five percent for five hundred iterations (see figures 13 and 14).

The downstream boundary condition model is described in section 3.3 . The flow through the boundary is assumed subsonic and the static pressure is held constant across the boundary. The Mach number contours of figure 15 demonstrate the ability of the downstream boundary condition model to allow flow disturbances to be convected downstream.

The inlet Mach number for this calculation was greater than one, but the component of the Mach number normal to the upstream boundary was less than one and information should be able to pass through the boundary. Fortunately, for any given back pressure there is a unique set of upstream conditions. If the upstream boundary condition model fixes conditions which are close to the correct conditions, and the upstream boundary is far from the blade, we expect the flow conditions

to adjust to the proper conditions before entering the blade passage. For this calculation the upstream boundary conditions were fixed and the back pressure adjusted to be compatible with them. The small adjustment of the flow at the upstream boundary seen in figure 15 indicates that the flow can make slight adjustments before entering the blade passage. This boundary condition should be modified to allow upstream influence so that the choice of upstream conditions does not depend on the back pressure.

Figure 15 shows the Mach number contours computed for the compressor cascade by the factored algorithm. The figure shows many of the features we expect of cascade flows. There is a small compression region at the leading edge due to the turning of the flow, a large expansion region on the suction surface, a shock wave between the pressure and suction surface, and a disturbance at the trailing edge which is convected downstream.

The most interesting of these phenomena is the disturbance downstream of the blade section. In viscous flow we expect the pressure and suction surface boundary layers to combine and form a wake which is convected downstream. Inviscid flow theory allows the stagnation streamline to leave the body at any location, depending on the circulation around the blade. The Kutta condition is introduced to produce a unique solution, generally the one for which the streamline leaves the body at the trailing edge. This eliminates an infinite acceleration around a

sharp trailing edge and models the physical flow as closely as an inviscid approximation will allow. The Kutta condition model used for this calculation assures that the pressure is continuous across the trailing edge. The Mach number contours indicate that the stagnation streamline does not leave the blade at the trailing edge but rather on the suction surface. This indicates that the Kutta condition model used for the calculation did not sufficiently simulate the expected solution, despite the fine grid spacing provided at the trailing edge (see figure 1).

Figure 16 shows the Mach number contours generated by Tong (2) using an explicit, time marching scheme with the same geometry and back pressure. This solution was computed on a much coarser grid than the grid used with the factored algorithm. This grid has 18 points on the blade and 10 evenly spaced points between the blades. The characteristics of the flow for the two calculations are very similiar. The expansion region is in the same location although the highest Mach number reached is only 1.8 for this calculation compared to 2.4 for the previous solution. This is probably due to the fine grid resolution used with the factored algorithm. The shock strength and location were also computed to be the same in both calculations, although the shock is not spread as much in the factored algorithm calculation. It is interesting to note that the trailing edge flow is very similar in both calculations despite the use of different algorithms and grid

meshes. The explicit scheme also forces constant pressure at the trailing edge to model the Kutta condition. This reinforces the notion that the low velocity region is a result of the Kutta condition model used in the calculations, not a result of grid spacing or a property of the solution algorithm.

The excellent agreement of the two schemes in predicting the flow field characteristics implies that the factored scheme is solving the Euler equations correctly. However, neither solution accurately models the physical flow because the modeling of the Kutta condition does not force the velocity discontinuity to the trailing edge.

5.0 Conclusions

The implicit factored algorithm presented solves the two dimensional thin shear layer equations in internal flow geometries. Boundary conditions have been developed which correctly model the internal flow boundaries. This algorithm and set of boundary conditions has been programmed and used to predict the flow through supersonic diffusers and through an axial compressor cascade. The algorithm has been shown to predict flow characteristics which agree well with theory, experiments and other computational schemes. This computer code may be used as a baseline calculation to which new solution schemes and new boundary condition models may be compared.

Improvements in the computational efficiency of the code can be made by developing a new downstream boundary condition which does not impose a stability limit on the calculation. The upstream boundary condition should also be improved to allow information to travel upstream in subsonic regions.

In order to more accurately simulate inviscid cascade conditions, a Kutta condition model must be developed which moves the velocity discontinuity to the trailing edge. This development may be bypassed by going to viscous calculations where there is no need for a Kutta condition model.

REFERENCES

1. Warming, R.E. and Beam, R.M., "On the Construction and Application of Implicit Factored Schemes for Conservation Laws," Symposium on Computational Fluid Dynamics, New York, April 16-17, 1977; SIAM-AMS Proceedings, Vol. 11, 1977.

2. Tong, S.S., private communication, see also: Thompkins, W.T. and Tong, S.S., "Inverse or Design Calculations for Non-Potential Flow in Turbomachinery Blade Passages," ASME Paper No. 81-GT-78, March 1981.

3. Hakkinen, R.J., Greber, I., Trilling, L., and Abarbanel, S.S., "The Interaction of an Oblique Shock Wave with a Laminar Boundary Layer," NASA Memo 2-18-59W, 1959.

4. Schlichting, H., "Boundary Layer Theory," McGraw Hill Book Company, New York, Sixth Edition, 1968.

5. Steger, J.L., "Implicit Finite-Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries," AIAA Journal, Vol. 16, No. 7, July 1978, pp 679-686.

6. Beam, R.M. and Warming, R.F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," AIAA Journal, Vol. 16, No. 4, April 1978, pp. 393-402.

7. Issacson, E. and Keller, H.B., "Analysis of Numerical Methods," John Wiley and sons, New York, 1966.

8. Pulliam, T.H. and Steger, J.L., "On Implicit Finite Difference Simulations of Three Dimensional Flows," AIAA 16th Aerospace Sciences Meeting, Huntsville, Alabama, Paper 78-10, January 16-18, 1978.

9. Usab, W.J., Jr., "Prediction of Three-Dimensional Compressible Turbulent Boundary Layers on Transonic Compressor Blades," Gas Turbine & Plasma Dynamics Laboratory, M.I.T., Report No. 152, October 1980.

10. von Lavante, E., private communication.

APPENDIX A1

-STRONG CONSERVATION FORM OF THE NAVIER-STOKES EQUATIONS WRITTEN IN NON-ORTHOGONAL COORDINATES-

The strong conservation form of the Navier-Stokes equations written in vector form is:

$$\frac{\partial q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = \frac{1}{Re_o} \left[\frac{\partial R}{\partial x} + \frac{\partial S}{\partial y} \right] \tag{A1.1}$$

If a rectangular computational grid (ξ,η, t) is defined, a general transformation to physical coordinates (x,y,t), which are not necessarily orthogonal, may be defined as:

$$\xi = \xi(x,y) \qquad \eta = \eta(x,y) \qquad t = t \ . \tag{A1.2}$$

It will be shown that an equation of the form:

$$\frac{\partial}{\partial t}\hat{q} + \frac{\partial}{\partial \xi}\hat{E} + \frac{\partial}{\partial \eta}\hat{F} = \frac{1}{Re_o} \left[\frac{\partial}{\partial \xi}\hat{R} + \frac{\partial}{\partial \eta}\hat{S} \right] \tag{A1.3}$$

where

$$\hat{q} = q/J$$

$$\hat{E} = \left(E\frac{\partial \xi}{\partial x} + F\frac{\partial \xi}{\partial y} \right)/J$$

$$\hat{F} = \left(E\frac{\partial \eta}{\partial x} + F\frac{\partial \eta}{\partial y}\right)/J$$

$$\hat{R} = \left(R\frac{\partial \xi}{\partial x} + S\frac{\partial \xi}{\partial y}\right)/J$$

$$\hat{S} = \left(R\frac{\partial \eta}{\partial x} + S\frac{\partial \eta}{\partial y}\right)/J$$

$$J = \frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial y} - \frac{\partial \xi}{\partial y}\frac{\partial \eta}{\partial x} = \frac{1}{\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}} \quad ,$$

is equivalent to the strong conservation form of the Navier-Stokes equations.

To show that (A1.3) is equivalent to (A1.1), the definitions of $\hat{E}$, $\hat{F}$ and $\hat{S}$ are plugged into equation (A1.3) and the equation is rearranged as follows.

$$\frac{\partial}{\partial t}\left(q/J\right) + \frac{\partial}{\partial \xi}\left(E\frac{1}{J}\frac{\partial \xi}{\partial x} + F\frac{1}{J}\frac{\partial \xi}{\partial y}\right) + \frac{\partial}{\partial \eta}\left(E\frac{1}{J}\frac{\partial \eta}{\partial x} + E\frac{1}{J}\frac{\partial \eta}{\partial y}\right)$$

$$= \frac{\partial}{\partial \xi}\left(R\frac{1}{J}\frac{\partial \xi}{\partial x} + S\frac{1}{J}\frac{\partial \xi}{\partial y}\right) + \frac{\partial}{\partial \eta}\left(R\frac{1}{J}\frac{\partial \eta}{\partial x} + S\frac{1}{J}\frac{\partial \eta}{\partial y}\right) \qquad (A1.4)$$

$$\underline{\frac{\partial}{\partial t}\left(q/J\right)} + \underline{\frac{1}{J}\frac{\partial \xi}{\partial x}\left(\frac{\partial E}{\partial \xi}\right)} + E\frac{\partial}{\partial \xi}\left(\frac{1}{J}\frac{\partial \xi}{\partial x}\right) + \underline{\frac{1}{J}\frac{\partial \xi}{\partial y}\left(\frac{\partial F}{\partial \xi}\right)} + F\frac{\partial}{\partial \xi}\left(\frac{1}{J}\frac{\partial \xi}{\partial y}\right)$$

$$+ \underline{\frac{1}{J}\frac{\partial \eta}{\partial x}\left(\frac{\partial E}{\partial \eta}\right)} + E\frac{\partial}{\partial \eta}\left(\frac{1}{J}\frac{\partial \eta}{\partial x}\right) + \underline{\frac{1}{J}\frac{\partial \eta}{\partial y}\left(\frac{\partial F}{\partial \eta}\right)} + F\frac{\partial}{\partial \eta}\left(\frac{1}{J}\frac{\partial \eta}{\partial y}\right)$$

$$= \underline{\frac{1}{J}\frac{\partial \xi}{\partial x}\left(\frac{\partial R}{\partial \xi}\right)} + R\frac{\partial}{\partial \xi}\left(\frac{1}{J}\frac{\partial \xi}{\partial x}\right) + \underline{\frac{1}{J}\frac{\partial \xi}{\partial y}\left(\frac{\partial S}{\partial \xi}\right)} + S\frac{\partial}{\partial \xi}\left(\frac{1}{J}\frac{\partial \xi}{\partial y}\right)$$

$$+ \underline{\frac{1}{J}\frac{\partial \eta}{\partial x}\left(\frac{\partial R}{\partial \eta}\right)} + R\frac{\partial}{\partial \eta}\left(\frac{1}{J}\frac{\partial \eta}{\partial x}\right) + \underline{\frac{1}{J}\frac{\partial \eta}{\partial y}\left(\frac{\partial S}{\partial \eta}\right)} + S\frac{\partial}{\partial \eta}\left(\frac{1}{J}\frac{\partial \eta}{\partial y}\right) \qquad \text{(A1.5)}$$

The underlined terms of (A1.5) are equation (A1.1) divided by the Jacobian determinant and so it only remains to show that the other terms sum to zero. This requires some simple identities between the metrics. Two such identities may be derived from the expressions for the total derivatives $d\xi$ and $d\eta$.

$$d\xi = \frac{\partial \xi}{\partial x}\,dx + \frac{\partial \xi}{\partial y}\,dy \qquad \text{(A1.6A)}$$

$$d\eta = \frac{\partial \eta}{\partial x}\,dx + \frac{\partial \eta}{\partial y}\,dy \qquad \text{(A1.6B)}$$

Multiplying $d\xi$ by $\frac{\partial \eta}{\partial x}$ and $d\eta$ by $\frac{\partial \xi}{\partial x}$ and re-arranging gives:

$$\frac{\partial \eta}{\partial x} d\xi - \frac{\partial \xi}{\partial x} d\eta = \frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial x} dx + \frac{\partial \eta}{\partial x}\frac{d\xi}{\partial y} dy - \frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial x} dx - \frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial y} dy$$

$$= \left(\frac{\partial \eta}{\partial x}\frac{\partial \xi}{\partial y} - \frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial y}\right) dy$$

$$= -Jdy$$

$$= -J\left(\frac{\partial y}{\partial \xi} d\xi + \frac{\partial y}{\partial \eta} d\eta\right) \qquad \text{(A1.7)}$$

This equation implies the identities:

$$\frac{\partial \eta}{\partial x} = -J\frac{\partial y}{\partial \xi} \; ; \; \frac{\partial \xi}{\partial x} = J\frac{\partial y}{\partial \eta} \qquad \text{(A1.8A,B)}$$

Similiar relations may be derived by evaluating

$\left(\frac{\partial \eta}{\partial y} d\xi - \frac{\partial \xi}{\partial y} d\eta\right)$ to obtain:

$$\frac{\partial \xi}{\partial y} = -J\frac{\partial x}{\partial \eta} \; ; \; \frac{\partial \eta}{\partial y} = J\frac{\partial x}{\partial \xi} \qquad \text{(A1.8C,D)}$$

The two terms of equation (A1.5) containing E cancel as we can see by inserting the relations (A1.8A,B) .

$$E\frac{\partial}{\partial\xi}\left(\frac{1}{J}\frac{\partial\xi}{\partial x}\right) + E\frac{\partial}{\partial\eta}\left(\frac{1}{J}\frac{\partial\eta}{\partial x}\right) = E\left[\frac{\partial}{\partial\xi}\left(\frac{1}{J}\left(J\frac{\partial y}{\partial\eta}\right)\right) + \frac{\partial}{\partial\eta}\left(\frac{1}{J}\left(-J\frac{\partial y}{\partial\xi}\right)\right)\right]$$

$$= E\left(\frac{\partial^2 y}{\partial\xi\partial\eta} - \frac{\partial^2 y}{\partial\xi\partial\eta}\right)$$

$$= 0 \qquad \text{(A1.9)}$$

Similarly, the terms containing F, R or S may be combined to show that they also vanish in pairs.

Thus, under this transformation, (A1.3) is identical to (A1.1) . The only difference is the slightly more complicated form of the flux vectors $\hat{E}$, $\hat{F}$, $\hat{R}$ and $\hat{S}$. Thus the computer time required to solve the equations is only minimally changed while adding great flexibility to the geometries which can be calculated. Changing the geometries only requires recalculating the metrics.

APPENDIX A2

-STABILITY ANALYSIS OF MODEL ONE-DIMENSIONAL WAVE EQUATION-

The one dimensional scalar wave equation:

$$\frac{\partial}{\partial t} u + a \frac{\partial}{\partial x} u = 0 \qquad \text{(A2.1)}$$

may be descretized in time using a trapezoidal delta formulation (see Section 2.5.1) to obtain:

$$\Delta u^{n} = -\frac{\Delta t}{2} a \frac{\partial}{\partial x} \Delta u^{n} - a \Delta t \frac{\partial}{\partial x} u^{n} + O(\Delta t^2) \quad . \qquad \text{(A2.2)}$$

If central difference expressions are used to approximate the ξ and η derivatives, the equation may be written:

$$u_j^{n+1} - u_j^{n} = -\frac{a\Delta t}{4\Delta x}\left[\left(U_{j+1}^{n+1} - U_{j-1}^{n+1}\right) - \left(U_{j+1}^{n} - U_{j-1}^{n}\right)\right]$$

$$-\frac{a\Delta t}{2\Delta x}\left(U_{j+1}^{n} - U_{j-1}^{n}\right) + O\left(\Delta t^2, \Delta x^2\right) \qquad \text{(A2.3)}$$

If the solution is represented as:

$$U_j^{n} = A^{n} e^{i(pj\Delta x)}, \qquad \text{(A2.4)}$$

a VonNeumann stability analysis may be done using equation (A2.3) . After this substitution and the use of Euler's identity, $e^{i\theta} = \cos\theta + i \sin\theta$, equation (A2.3) becomes:

$$(A - 1) = -\frac{a\Delta t}{4\Delta x}(2i \sin p\,\Delta x)(A - 1) - \frac{a\Delta t}{2\Delta x}(2i \sin p\,\Delta x).$$

Solving for A yields:

$$A = \frac{1 - \frac{a\Delta t}{4\Delta x}(2i \sin p\,\Delta x)}{1 + \frac{a\Delta t}{4\Delta x}(2i \sin p\,\Delta x)}.$$

Thus the magnitude of the growth factor A is one for all time steps (Δt), grid spacing (Δx) and wave numbers (p). The scheme is unconditionally stable.

APPENDIX A3

-STABILITY ANALYSIS OF MODEL TWO-DIMENSIONAL WAVE EQUATION

The two dimensional scalar wave equation

$$\frac{\partial}{\partial t} u + a \frac{\partial}{\partial x} u + b \frac{\partial}{\partial y} u = 0 \qquad (3.1)$$

may be descretized in time using a trapezoidal delta formulation (see Section 2.5.1) to obtain:

$$\Delta u^n = -\frac{\Delta t}{2} \left[a \frac{\partial}{\partial x} \left(\Delta u^n \right) + b \frac{\partial}{\partial y} \left(\Delta u^n \right) + 2a \frac{\partial}{\partial x} u^n + 2b \frac{\partial}{\partial y} u^n \right] + O(\Delta t^2) \qquad (A3.2)$$

where $\Delta u^n = u^{n+1} - u^n$. We can obtain the differencing scheme used on the downstream boundary by noting that $\Delta u^n = \Delta u^{n-1} + O(\Delta t^2)$. Then, using central difference approximations for the y-derivatives, and first order backwards difference approximations for the x-derivatives, the finite difference form of equations (A3.3) becomes:

$$\left(U^{n+1}_{j,k} - U^{n}_{j,k}\right) = -\frac{a\Delta t}{2\Delta x}\left[\left(U^{n}_{j,k} - U^{n}_{j-1,k}\right) - \left(U^{n-1}_{j,k} - U^{n-1}_{j-1,k}\right)\right]$$

$$-\frac{b\Delta t}{4\Delta y}\left[\left(U^{n+1}_{j,k+1} - U^{n+1}_{j,k-1}\right) - \left(U^{n}_{j,k+1} - U^{n}_{j,k-1}\right)\right]$$

$$-\frac{a\Delta t}{\Delta x}\left[\left(U^{n}_{j,k} - U^{n}_{j-1,k}\right) - \frac{b\Delta t}{2\Delta y}\left(U^{n}_{j,k+1} - U^{n}_{j,k-1}\right)\right]$$

$$+ O(\Delta t^2, \Delta x) \qquad \text{(A3.4)}$$

To perform a VonNeumann stability analysis, we set

$$C_x = \frac{a\Delta t}{\Delta x}\ , \quad C_y = b\frac{\Delta t}{\Delta y} \quad \text{and} \quad U^{n}_{j,k} = A^{n}\, e^{i(pj\Delta x + qk\Delta y)}\ .$$

Substituting these expressions into equation (A3.4) and dividing by a common factor of $A^{n-1}\, e^{i(pj\Delta x + qk\Delta y)}$ we obtain:

$$A^2\left[1 + i\left(\frac{1}{2} C_y \sin q\Delta y\right)\right]$$

$$+ A\left[-1 + \frac{3}{2} C_x\left(1 - e^{-ip x}\right) + i\left(\frac{1}{2} C_y \sin q\Delta y\right)\right]$$

$$- \frac{1}{2} C_x\left(1 - e^{-ip x}\right) = 0 \qquad \text{A3.5}$$

This is a quadratic equation with complex coefficients for A. The roots of the equation can be found for a range of courant numbers (C_x , C_y) and wave numbers ($p\Delta x$). The scheme is stable if the growth factor (A) is less than one, and unstable for A greater than one.

The growth factor has been calculated for a range of conditions and the results summarized in figure A3.1 . Thus the model wave equation is conditionally stable under the scheme proposed.

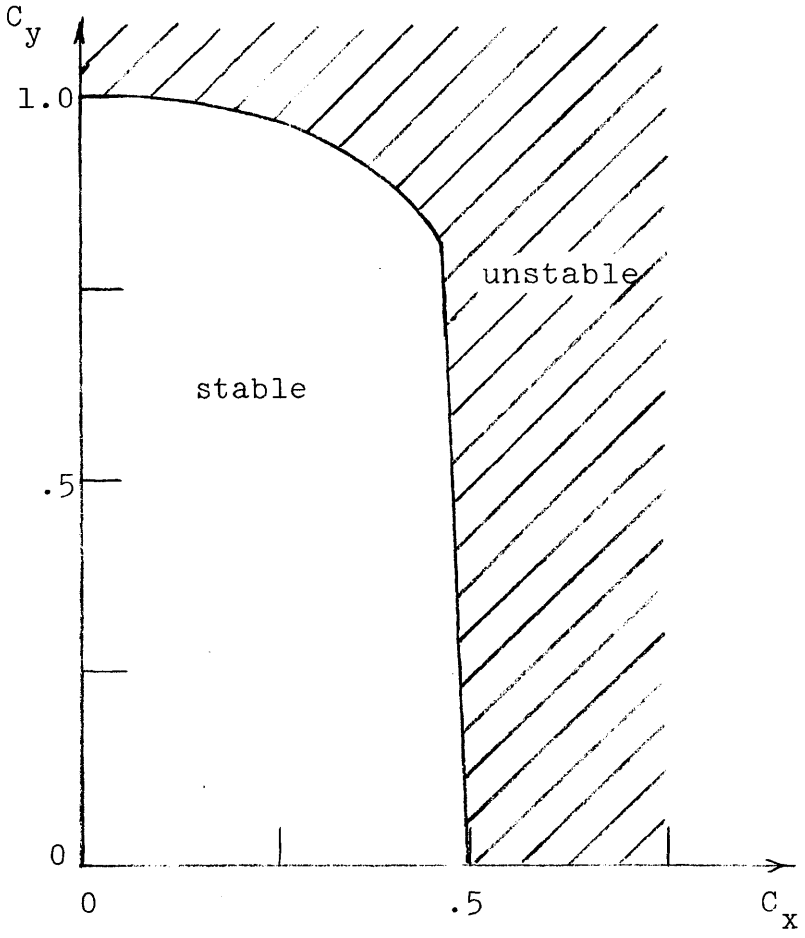


Figure A3.1: Stable region.

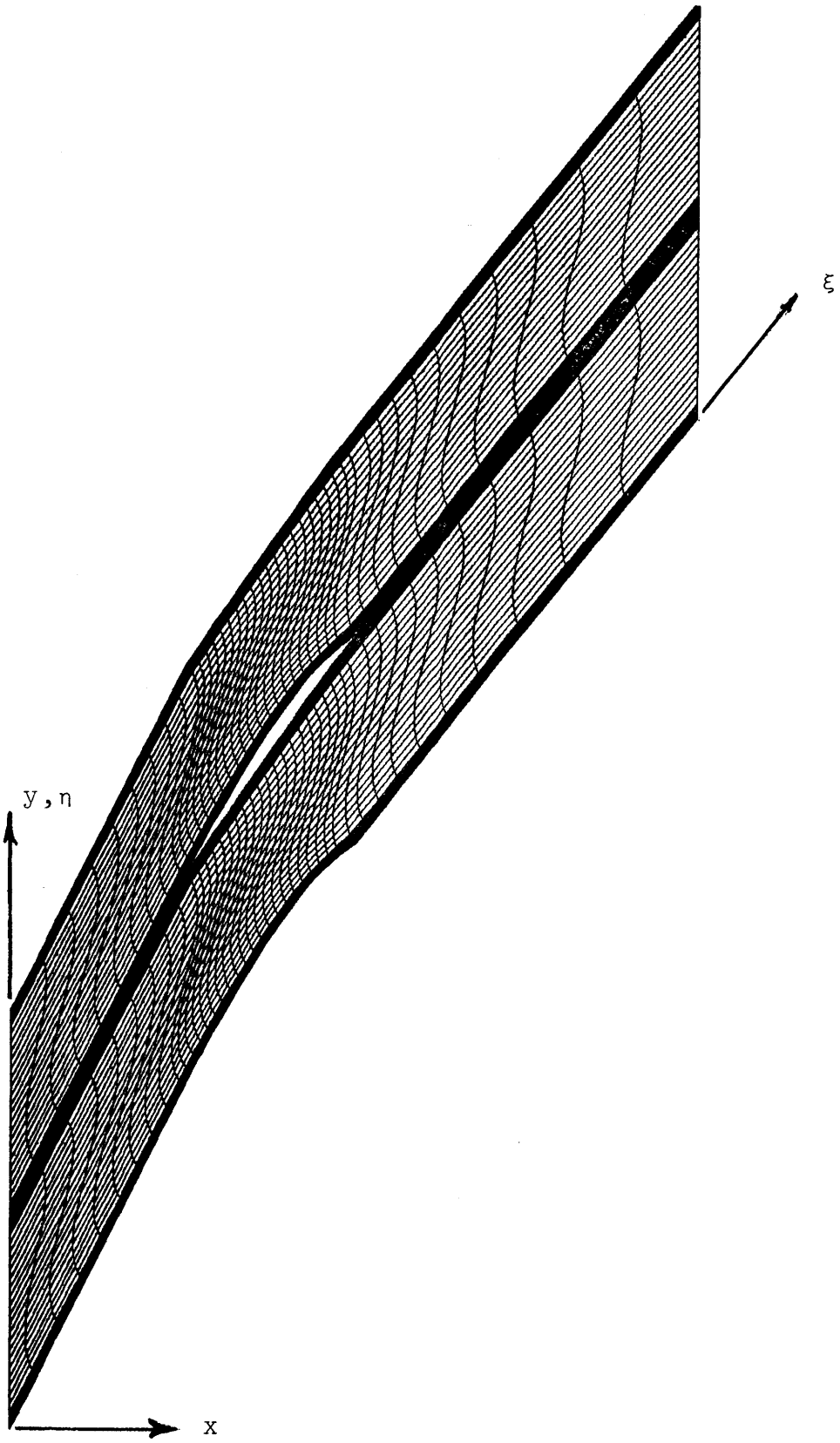


Figure 1: Computational grid for cascade calculations.

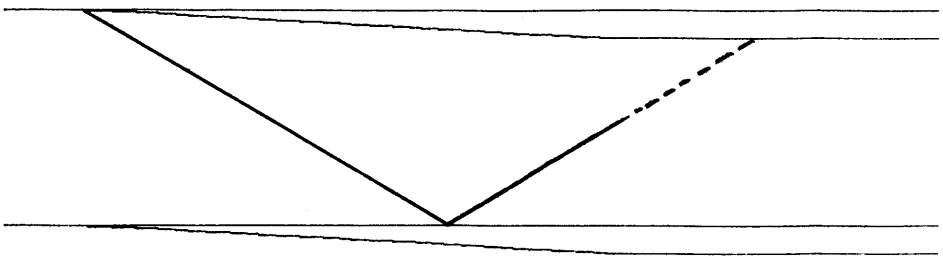


Figure 2: Geometry and theoretical incident and reflected shock locations for inviscid diffuser.

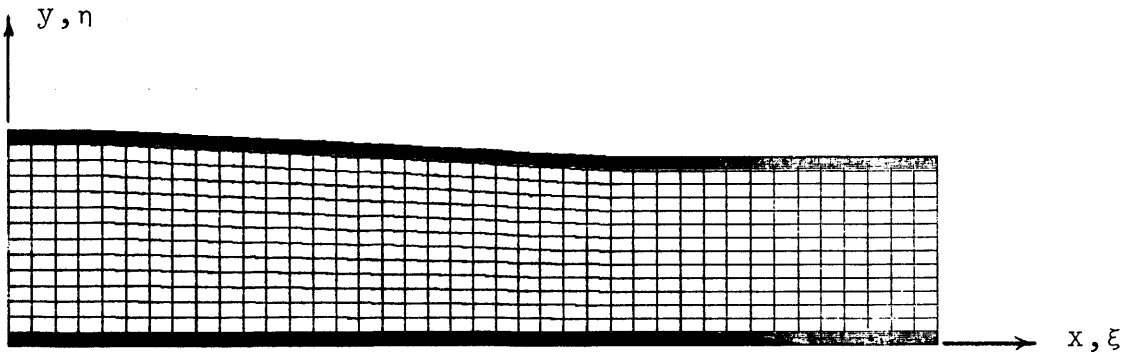


Figure 3: Computational grid for diffuser calculations.

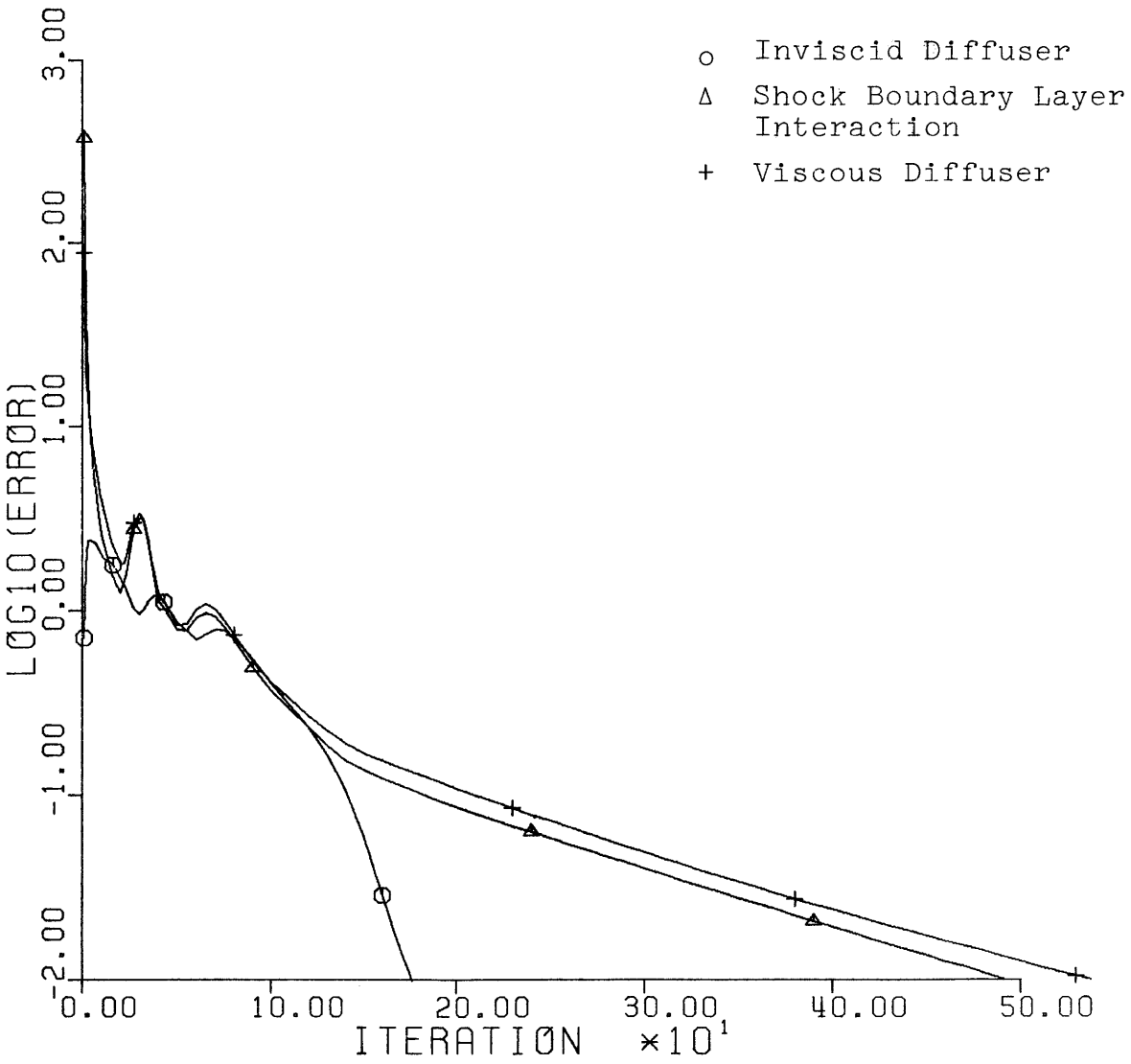


Figure 4: Convergence history for diffuser calculations.

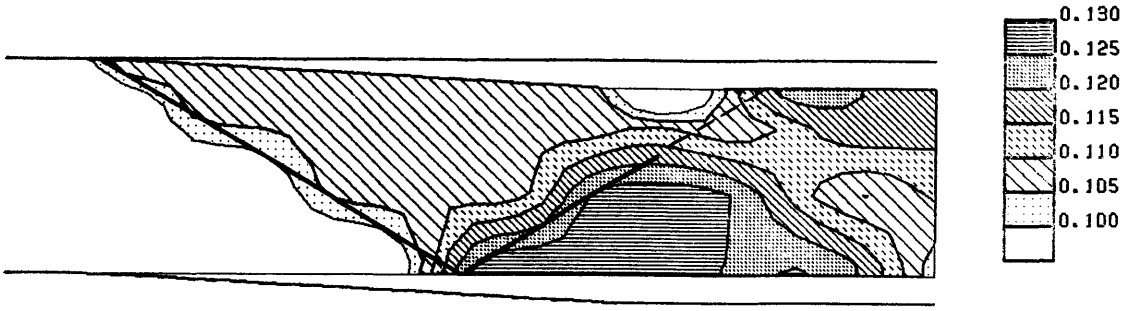


Figure 5: Pressure contours for the inviscid diffuser as computed by the present scheme.

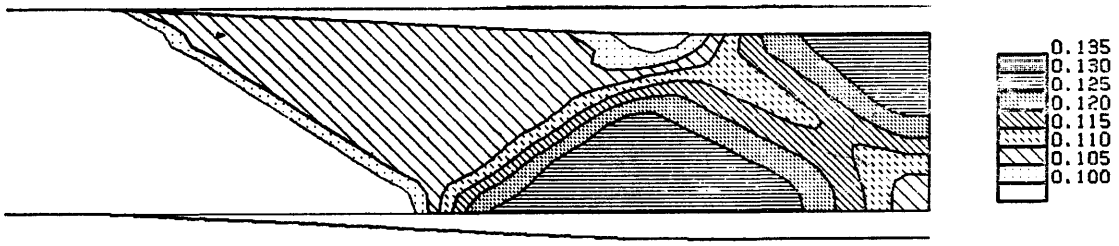


Figure 6: Pressure contours for the inviscid diffuser as computed by Tong (2) using a MacCormack scheme.

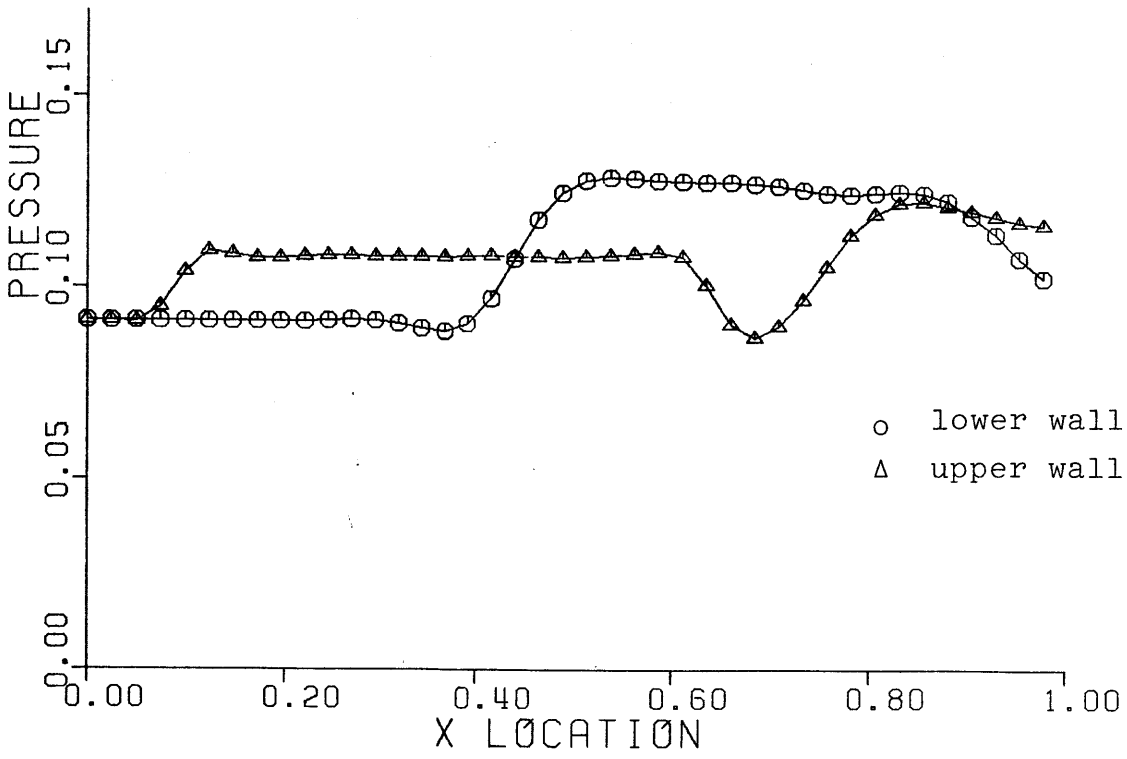


Figure 7: Wall pressure distribution for the inviscid diffuser as computed by the present scheme.

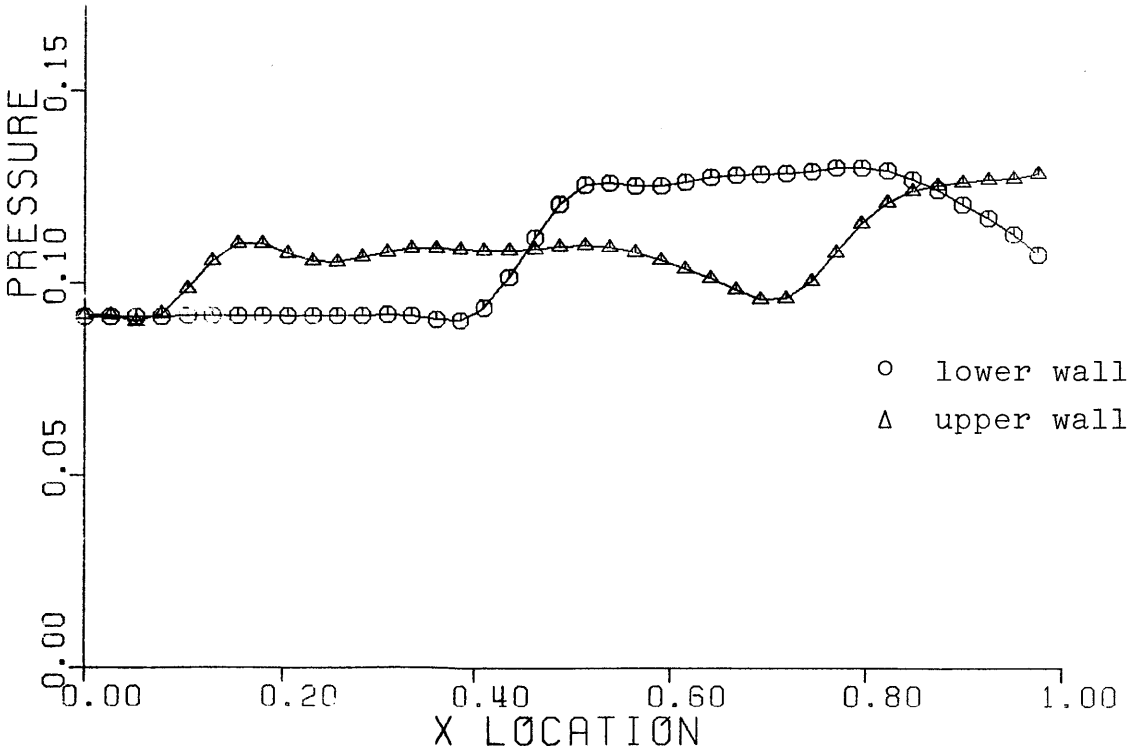


Figure 8: Wall pressure distribution for the inviscid diffuser as computed by Tong (2) using a MacCormack scheme.

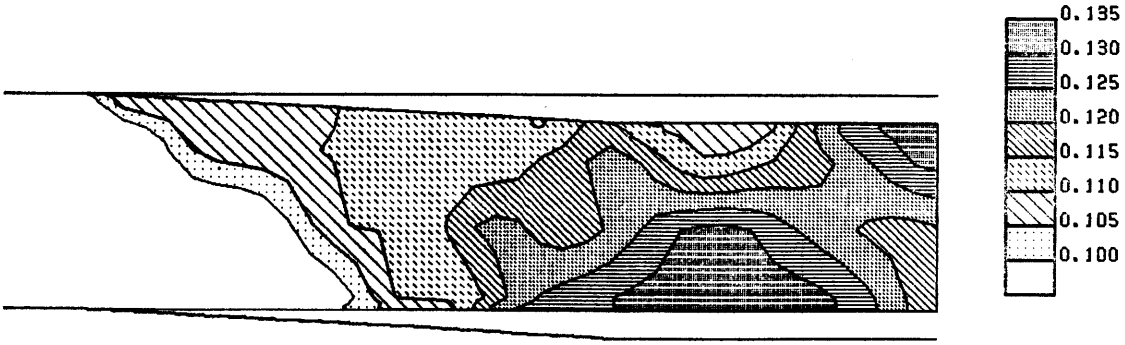


Figure 9: Pressure contours for the shock boundary layer interaction example.

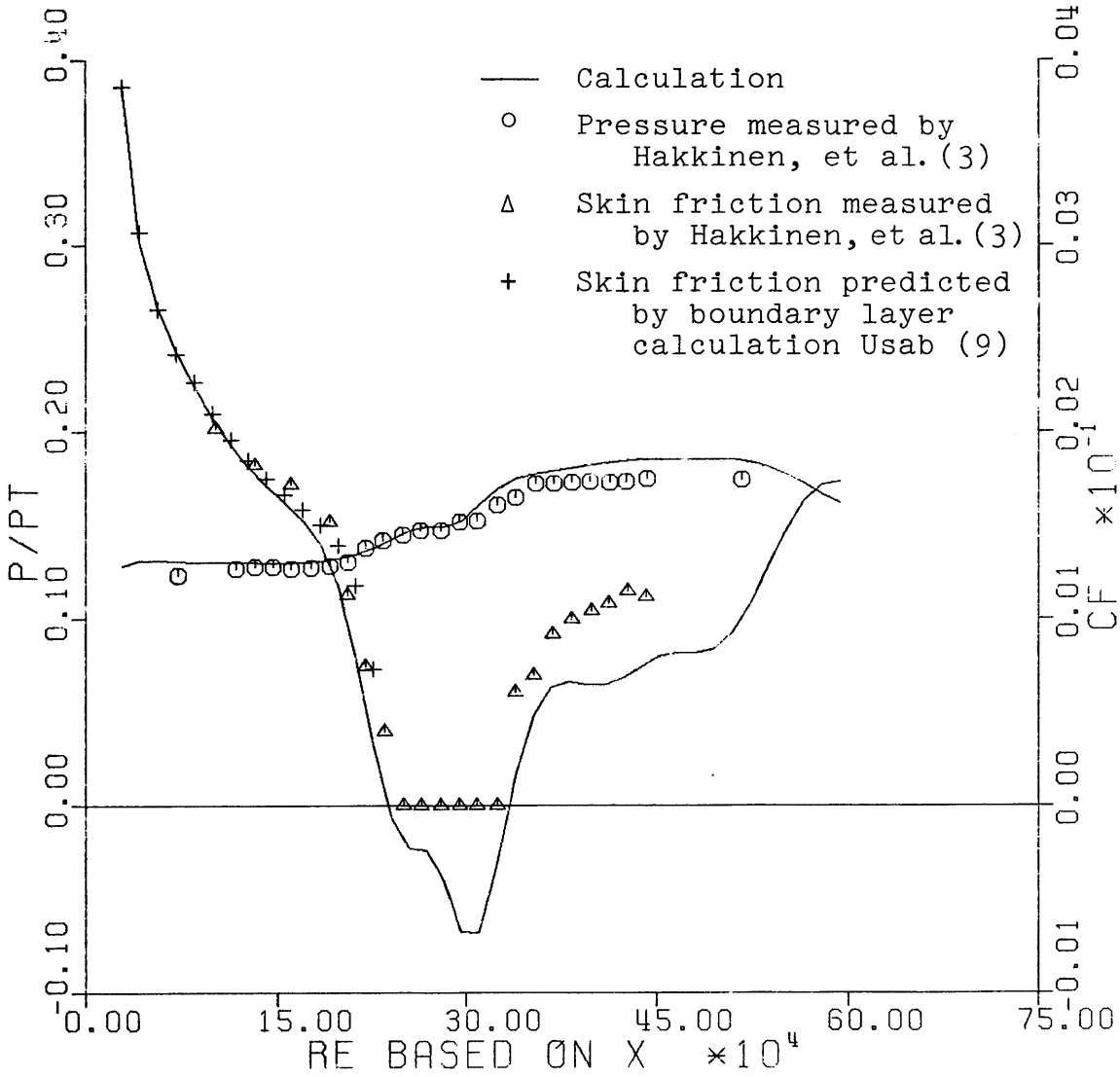


Figure 10: Wall pressure and skin friction distribution for the shock boundary layer interaction example.

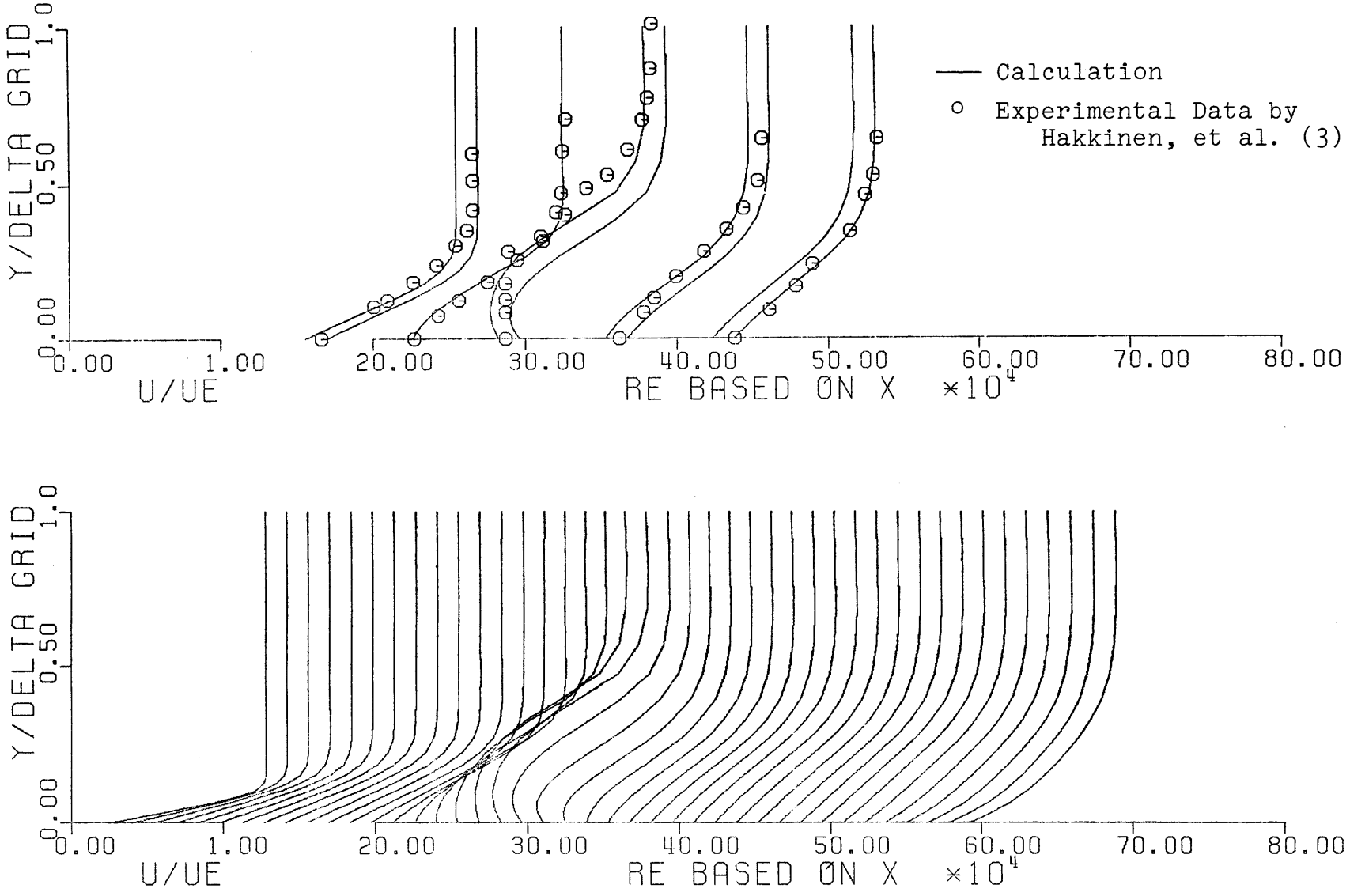


Figure 11: Boundary layer velocity profiles computed for shock boundary layer interaction.

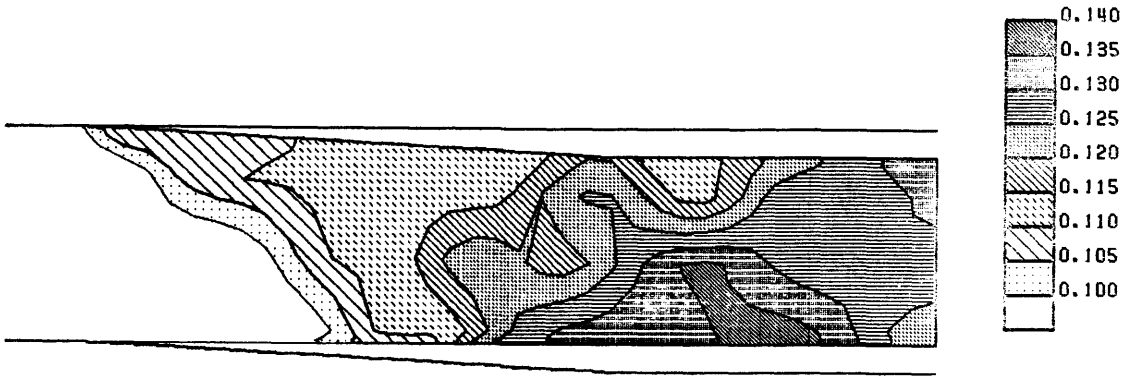


Figure 12: Pressure contours computed for the viscous diffuser by the present scheme.

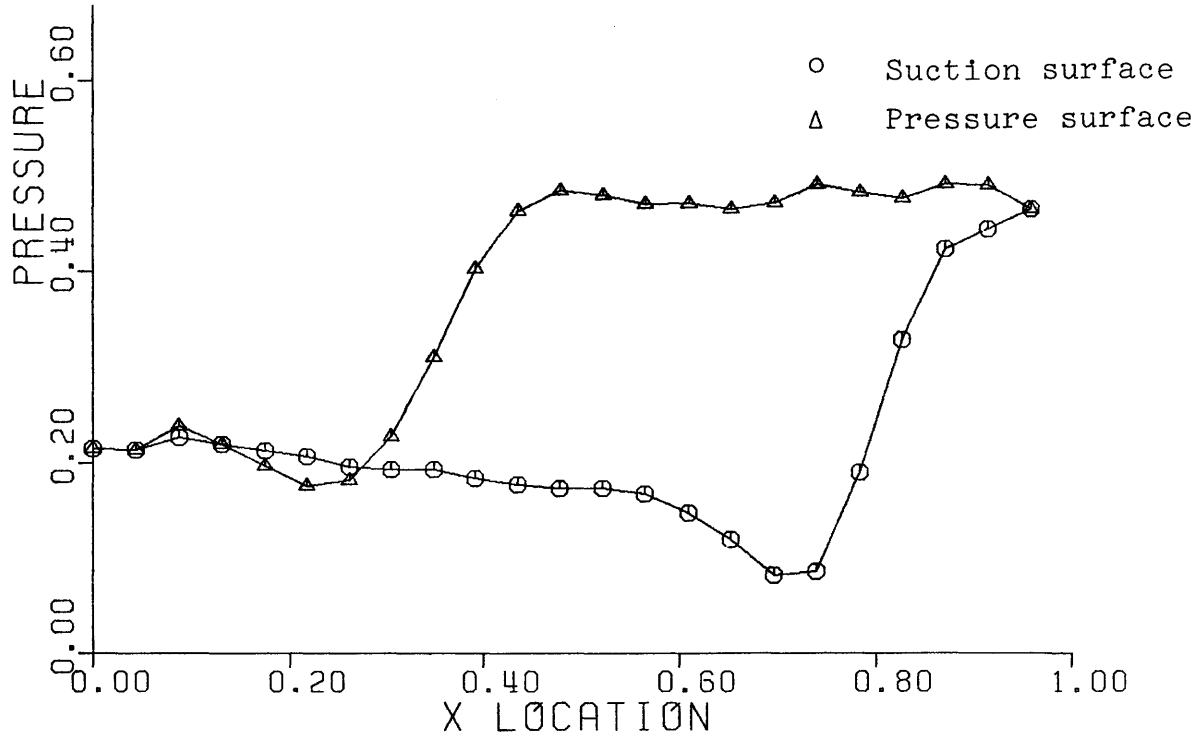


Figure 13: Cascade blade pressure distribution at iteration 1000 .

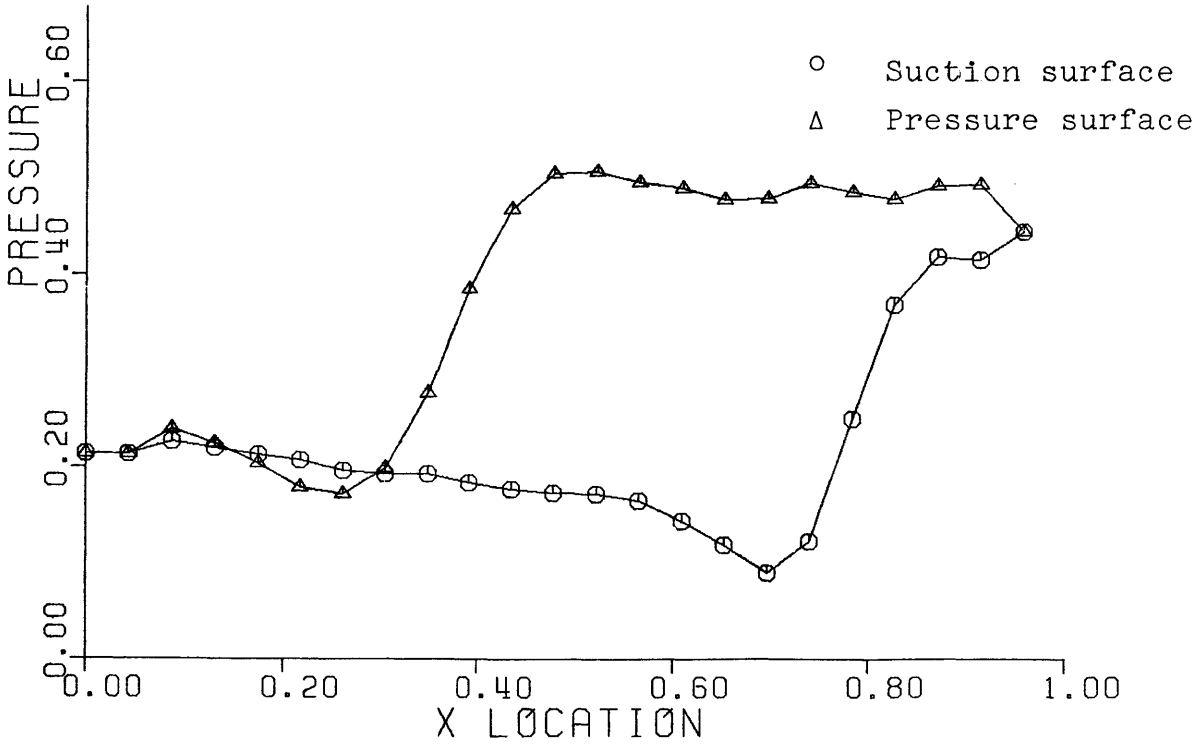


Figure 14: Cascade blade pressure distribution at iteration 1500 .

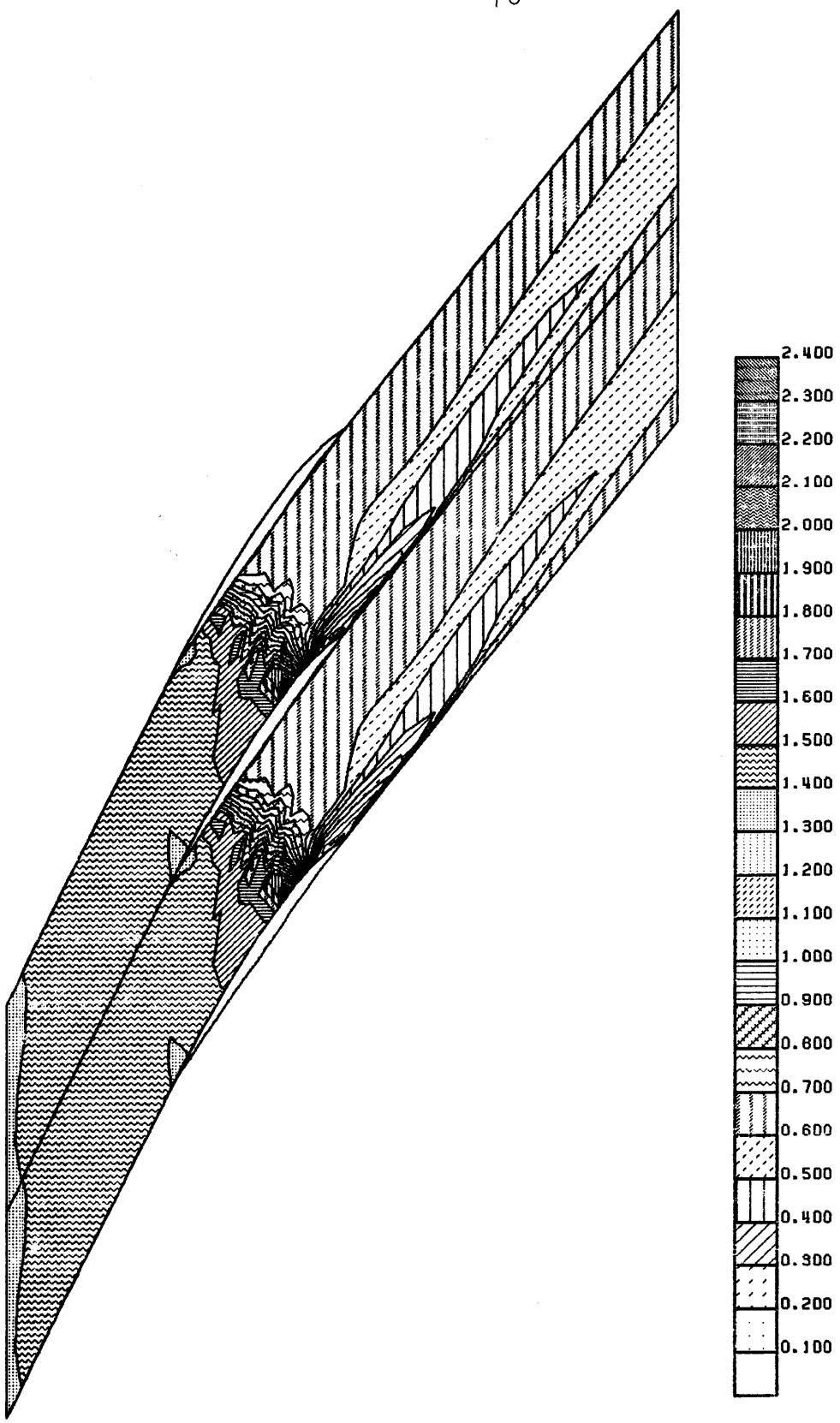


Figure 15: Mach number contours computed for the inviscid cascade by the present scheme.

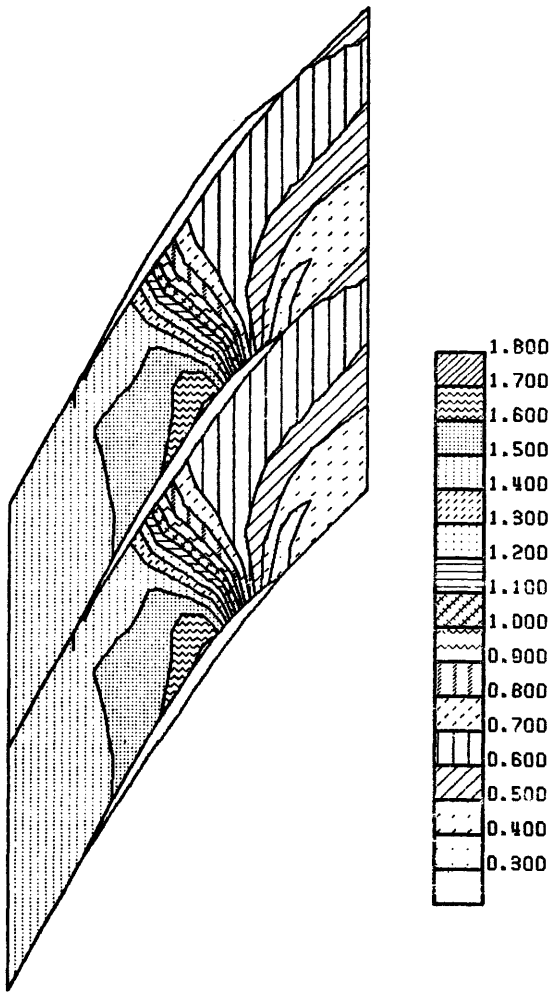


Figure 16: Mach number contours computed for the inviscid cascade by Tong (2) using a MacCormack scheme.