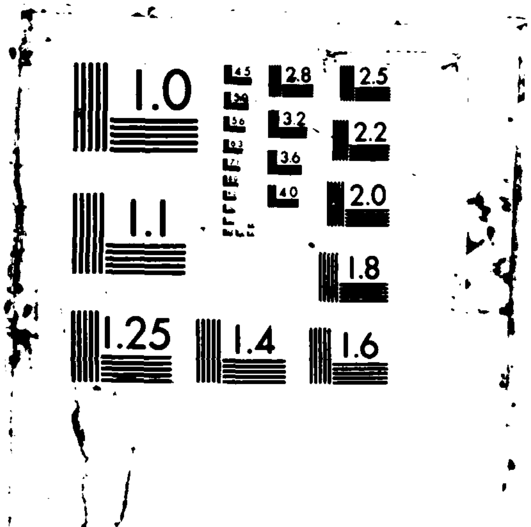
AD-A179 113 NUMERICAL STUDY OF SUPERSONIC FLOWS USING DIFFERENT 1/1
TECHNIQUES(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON
AFB OH SCHOOL OF ENGINEERING G W HUBAND DEC 86
UNCLASSIFIED AFIT/GA/AA/86D-8 F/G 20/4 NL



CROCOPY RESOLUTION TEST CHART

(1

NUMERICAL STUDY OF SUPERSONIC FLOWS

USING DIFFERENT TECHNIQUES

THESIS

Gary W. Huband
Captain, USAF

AFIT/GA/AA/86D-8

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GA/AA/86D-8

1

NUMERICAL STUDY OF SUPERSONIC FLOWS USING DIFFERENT TECHNIQUES

THESIS

Gary W. Huband
Captain, USAF

AFIT/GA/AA/86D-8

AFIT/GA/AA/86D-8

NUMERICAL STUDY OF SUPERSONIC FLOWS USING DIFFERENT TECHNIQUES

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Astronautical Engineering

Gary W. Huband, B.S.
Captain, USAF

December 1986

Approved for public release; distribution unlimited

PREFACE

The purpose of this work was to obtain the solution to three supersonic flow problems using three different numerical techniques.

First, a shock/boundary layer problem is solved using MacCormack's explicit technique. Then, using the same technique a shrouded rocket nozzle problem is solved. These two problems showed that the explicit scheme required many minutes of computer time to solve.

In order to explore more efficient codes to solve the shroud problem, space marching algorithms were studied. A space marching algorithm using flux-splitting in the streamwise direction was applied to an approximate form of the Navier-Stokes equation. Flux-splitting combined with a global iteration approach should allow the shroud problem to be solved with a space marching algorithm. The flux-splitting code was applied to two supersonic flow problems and very good results were obtained.

This work, of course, is not the result of my labor alone. Dr Elrod helped me a great deal on the shroud problem. Major Hodge showed me how to use the explcit code and familiarized me with the Cray XMP. Dr Shang, of the Flight Dynamics Lab, has helped me understand the nuiances of MacCormack's explicit scheme. Finally, I have to thank my wife, Jackie, for having the patience to endure my absence from her for these past few months.

TABLE OF CONTENTS

List of Figures

List of Symbols

Symbol	Definition
A	area
a	speed of sound
B	body force vector
E	streamwise flux vector
E_t	total energy per unit volume
e	internal energy per unit mass
F	normal flux vector
F_x	total force in x-direction
f	force on shroud due to shear stresses
I	identity matrix
J	jacobian of coordinate transformation
k	coefficient of thermal conductivity
L	reference length
LHS	implicit terms of Beam-Warming scheme
M	Mach number
m	mass flow rate
P	pressure vector
Pr	Prandtl number
p	pressure
[Q]	jacobian of inviscid streamwise flux vector
q	heat flux vector
R	real gas constant
[R]	jacobian of inviscid normal flux vector
Re	Reynold's number

Symbol	Definition
RHS	explicit terms of Beam-Warming scheme
r	common ratio, defined in equation 68
T	thrust or temperature
U	vector of conserved variables
x	streamwise direction coordinate
y	normal direction coordinate
u	streamwise velocity
v	normal velocity
$[W]$	Jacobian of viscous normal flux vector
$[w]$	elements of [W] without derivatives
ρ	density
τ	shear stress
μ	viscosity coefficient
γ	ratio of specific heats
Φ	dissipation function
σ	safety factor
ω	explicit-implicit weighting factor

Superscripts

I	streamwise marching station
$+$	positive flux
$-$	negative flux
T	transpose of a matrix
$*$	non-dimensional
n	time step index

Subscripts	Definition
a	ambient
b	shroud base
i	inviscid
n	nozzle exit
j	spatial step index
o	stagnation or reference condition
s	shroud exit
v	viscous
x	partial derivative with respect to x
ξ	partial derivative with respect to ξ
η	partial derivative with respect to η
y	partial derivative with respect to y
f	freestream
L	based on reference length
CFL	stability condition
Δ	based on mesh size

Abstract

In the present study, solutions to supersonic fluid dynamic problems using different numerical techniques are given.

First, the full Navier-Stokes equations are implemented in an explicit scheme to solve a shock/boundary layer interaction problem and a shrouded rocket nozzle problem. In the shroud problem, Mach contours, pressure contours, velocity vectors, shroud wall pressure and base wall pressure were plotted. Comparison of the base wall pressure with experimental values were inconclusive. Shroud thrust increment calculations showed that the shroud only produces thrust gain at ambient pressures very close to zero. It was also found that a great deal of computer time was needed to solve these problems with the explicit method.

To arrive at a more efficient scheme, an approximation was made to the full equations. The approximation is valid for supersonic flows with thin shear layers and also enables a more efficient scheme to be applied to the equations.

These schemes are called space marching schemes and allow a solution of a flow without separation in a single sweep of the flowfield. Problems with separation regions can also be solved, but a global iteration process which requires multiple sweeps is needed.

Two space marching techniques were explored. The first, credited to Vigneron, splits the pressure from the streamwise flux vector. This code was written for single sweep solution of a flowfield and it was used to solve a subsonic Couette flow and

a supersonic boundary layer problem. The second code, which is original to this work, uses flux-splitting on the streamwise flux vector and implements the global iteration technique. The same two problems solved using Vigneron's technique are solved using the flux splitting technique and the shock/boundary layer interaction problem is also solved.

It was found that Vigneron's technique is more efficient for solving flows without separation. The flux-splitting code, however, was found to be about fourteen times faster than the explicit code when applied to the shock/boundary layer interaction problem.

An efficiency parameter in the form of CPU time divided by the number of grid points divided by the number of iterations was calculated for each code. For Vigneron's method the value was 4.7×10^{-4} , for the explicit method the value was 2.2×10^{-5} and for the current method the value was 4.6×10^{-4} .

I. INTRODUCTION

Engineers are continually looking for ways to increase the efficiency of rocket engines since for every additional pound of thrust gained another pound of payload can be added or a higher orbit can be achieved. One way of increasing the thrust is to increase the effectiveness of the nozzle. Nozzles are designed for a specific altitude and at any other altitude the nozzle thrust is not optimum. Deployable nozzles are one method of increasing the thrust above the design altitude and shrouding the nozzle is possibly another method.

Figures 1 and 2 show the basic features of shrouded rocket nozzles. When the flow from the nozzle(s) is overexpanded the shroud is ineffective since the exhaust plume will not attach to the shroud. But, when the flow from the nozzle(s) is underexpanded the exhaust plume will expand out and attach to the shroud wall essentially providing a larger exhaust exit area. There is also the possibility that the shroud will be short enough so that the expanded flow will not attach to the shroud, but that case will not be studied here. Allowing the exhaust plume to further expand so that the exit pressure is closer to ambient pressure means that the exhaust flow is nearly optimum and hopefully the thrust is a maximum. However, as will be seen in Chapter II, many factors affect the thrust of the shroud and

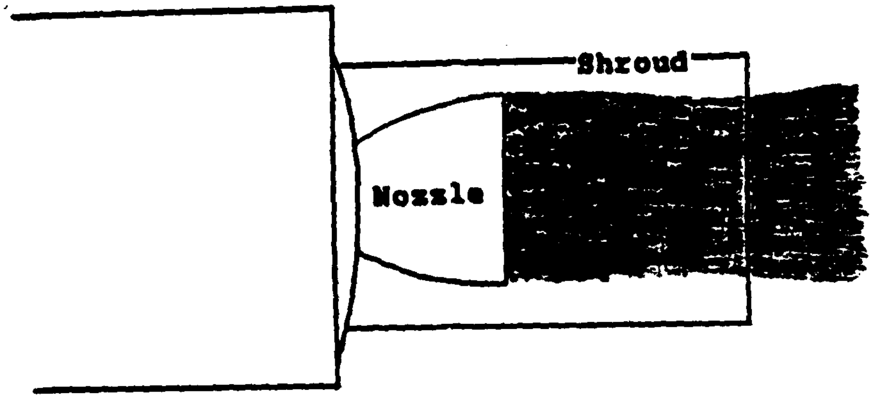


Figure 1 Shrouded Nozzle: Overexpanded

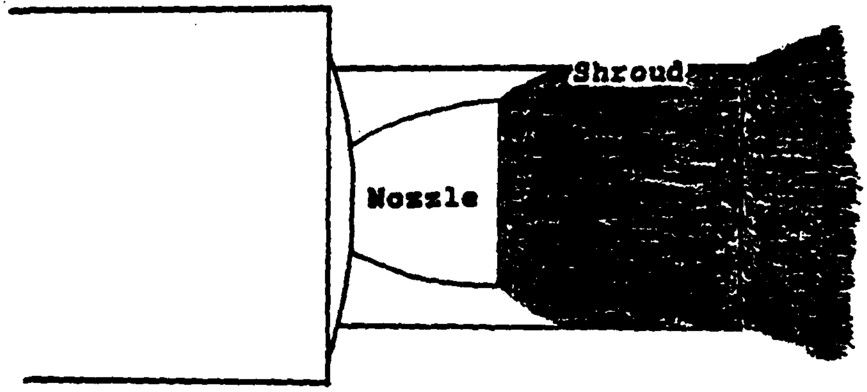


Figure 2 Shrouded Nozzle: Underexpanded

the addition of the shroud may cause a net thrust gain or a net thrust loss compared to the unshrouded nozzle.

Goethert (7) was the first to study the effects of shrouded rocket nozzles. Actually, Goethert used the shroud to eliminate the problem of nozzle burn through which occured on the clustered nozzles of the early Polaris missles. After Goethert, Holmes and Matz (8) studied the effects of shroud shape, shroud size and nozzle spacing on the shroud flow field. More recently, Moran (11) has completed experimental studies with various shroud shapes and nozzle configurations on both two and three dimensional nozzles.

On the numerical side, Roache and Mueller (13) studied incompressible and compressible flow over backsteps, which is similar to a single shrouded nozzle. Recently, Bardina and Lombard (3) have applied a three-dimensional split coefficient implicit code to shrouded multi-nozzles and successfully predicted the characteristic features of the flow.

The primary objective of the present work is to develop numerical algorithms capable of efficiently simulating predominately supersonic flowfields, such as shrouded rocket nozzles. Numerical simulation of the flow inside the shroud will normally give more detailed information than data obtained from experiment. This information may lead to an accurate understanding of the flow phenomena and consequently the optimum performance of the shroud can be predicted.

The flow problem inside the shroud of the rocket nozzle is formulated using the complete unsteady Navier-Stokes (NS) equations in terms of the conserved variables. The numerical scheme used to solve this problem is the explicit MacCormack method (2:479-489).

Results are first obtained for one special case for which published results are available; this consists of the shock/boundary layer interaction problem (4,18). Comparison of the present results with the available data provides a test of the validity and correctness of the present analysis and solution procedure.

Solutions are then obtained for the case of primary interest in the present study, namely the flow for a single shrouded nozzle. This model problem corresponds to experiment 3B from reference 11. The geometric details of the problem are given in Appendix A.

Upon completion of the shroud problem it was found that the CPU time required to obtain the solution was excessive. This results from the nature of the explicit scheme which has a limitation on the time step in order to insure a stable scheme. Therefore, many iterations and large amounts of computer time were required to reach a steady-state solution. To remove the time step restriction, fully implicit methods have been investigated. The implicit methods, however, still require many iterations to reach the steady state and consequently, still require large computational costs.

In an effort to decrease the computational costs associated with the implicit algorithms, space marching procedures have been studied. Partial differential equations are classified according to three types -- elliptic, parabolic and hyperbolic. In elliptic type problems the solution at one point depends on the solution at all other points in the domain. The problem must then be solved within a closed domain so that the influence of all points in the domain is taken into account. In parabolic and hyperbolic type problems, however, the solution at one point only depends on the points in a region of the domain. These types of problems are called marching problems since the solution advances outward from known initial conditions.

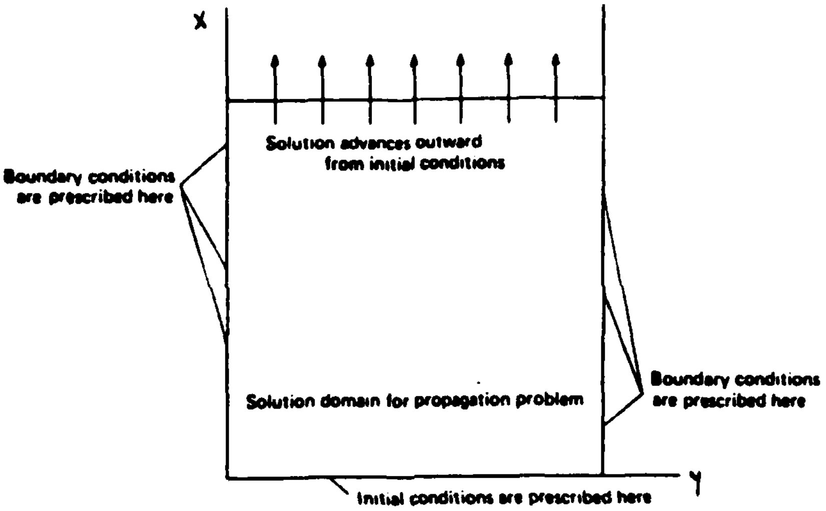

X
Solution advances outward from initial conditions
Boundary conditions are prescribed here
Solution domain for propagation problem
Boundary conditions are prescribed here
Initial conditions are prescribed here
Y

Figure 3 Solution domain for a marching problem

In the present efforts, two types of marching algorithms are investigated to solve the steady, approximate Navier-Stokes equations (ANS). The ANS equations differ from the NS equations in neglecting the streamwise diffusion terms and the time dependent terms. Now, when the flow is supersonic the ANS equations are parabolic/hyberbolic in nature, which allows the solution to be marched in the streamwise direction. Unlike the parabolic boundary layer equations, the ANS equations retain the interaction between the viscous and inviscid portions of the flow field which allows the ANS equations to model elliptic flows.

Since the ANS equations are elliptic when the flow is subsonic, then some method must be found to suppress the ellipticity so that the equations may be marched. Rubin and Lin (15), Schiff and Steger (16) and Vigneron, Rakich and Tannehill (19) have successfully applied such methods. In all of these methods, the streamwise flux vector was split in a way that treats the pressure term separately. For example, in the Schiff and Steger method, the pressure was assumed to be constant in the viscous subsonic layer and equal to the value from the adjacent supersonic flow, while Rubin and Lin indicate that setting the streamwise derivative of the pressure to zero can be a valid approximation for flows with cold walls. In any case, the error is confined to only a thin sublayer adjacent to the wall. Vigneron, Rakich and Tannehill approximated the streamwise derivative term with a weighting between implicit and explicit differencing that depends on the local Mach number.

The method of Vigneron, Rakich and Tannehill, known as Vigneron's method, was used to solve two problems. The first problem is the Couette flow inside a channel and the second problem is the supersonic laminar flow over a flat plate. Because the Couette problem is a fully developed flow, the streamwise derivatives are zero and the ANS equations are the exact equations for the problem. Also, the Couette flow has an exact analytical solution. The supersonic boundary layer flow has been solved by Lawerence and Tannehill (10) and provides a more severe check for the code.

A new approach to spatially marching the ANS equations is implemented in the present work. In this method, referred to as the current method, the Van Leer flux-splitting technique (1) is used to decompose the streamwise flux vector into two flux vectors which separately model the information traveling upstream and the information traveling downstream. Such splitting would appear to be better than the pressure splitting method since it is more representative of the physics of the flow field and would assure that the upstream traveling information does not destabilize the marching algorithm. Solutions are obtained for the same two problems solved using Vigneron's technique and both are compared. Also, the solution to the shock/boundary layer interaction problem is obtained and compared to the solution from the explicit code and the results obtained by Thomas (18).

Finally, a qualitative comparison between all the methods is given.

II. THEORY

In this chapter the equations governing the flow problems are examined. For the shroud problem a simple control volume analysis will reveal the important properties that affect the thrust gain of the shroud/nozzle configuration. Next, the equations to be used in the numerical schemes are presented; starting with the general form, advancing to the vector form of the unsteady Navier-Stokes equations and finally arriving at the approximate Navier-Stokes equations. Then, the flux-splitting technique used in the current method is discussed. Finally, the chapter will conclude by presenting the analytical solution to the Couette flow problem.

SHROUD ANALYSIS

Many important features of this problem can be analyzed by a simple control volume analysis. Equations for the thrust of the shroud and the nozzle are derived by using the x-momentum equation

$$F_x = \int u_x \, dm \tag{1}$$

Assuming that the flow properties and ambient conditions are a constant, then Eq. 1 can be applied to the control volume (CV) shown in figure 4 to give

$$T_e = m u_e + (p_e - p_a) A_e \tag{2}$$

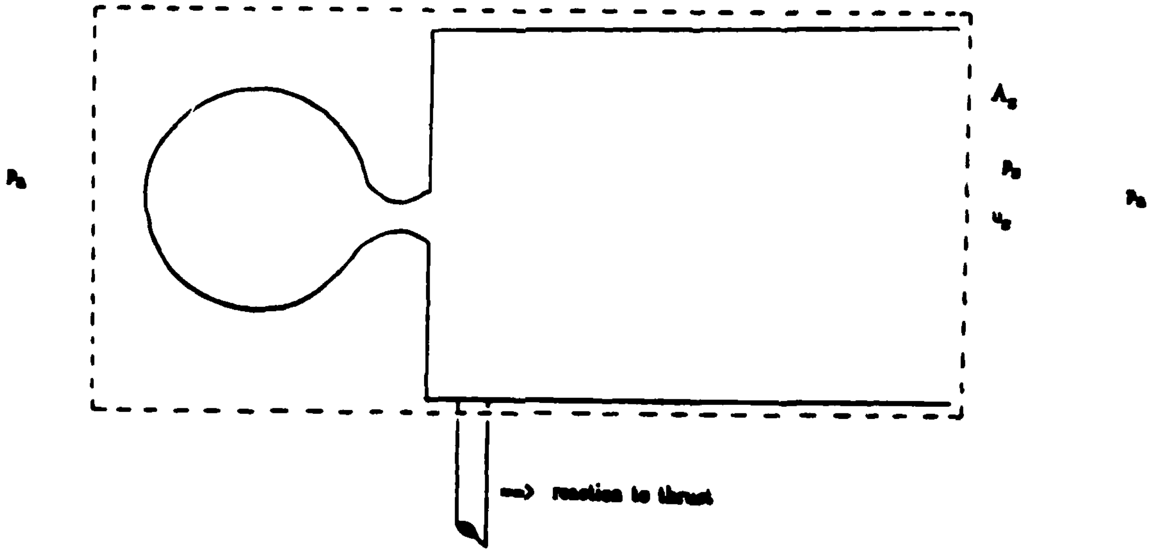


Figure 4 First shroud control volume

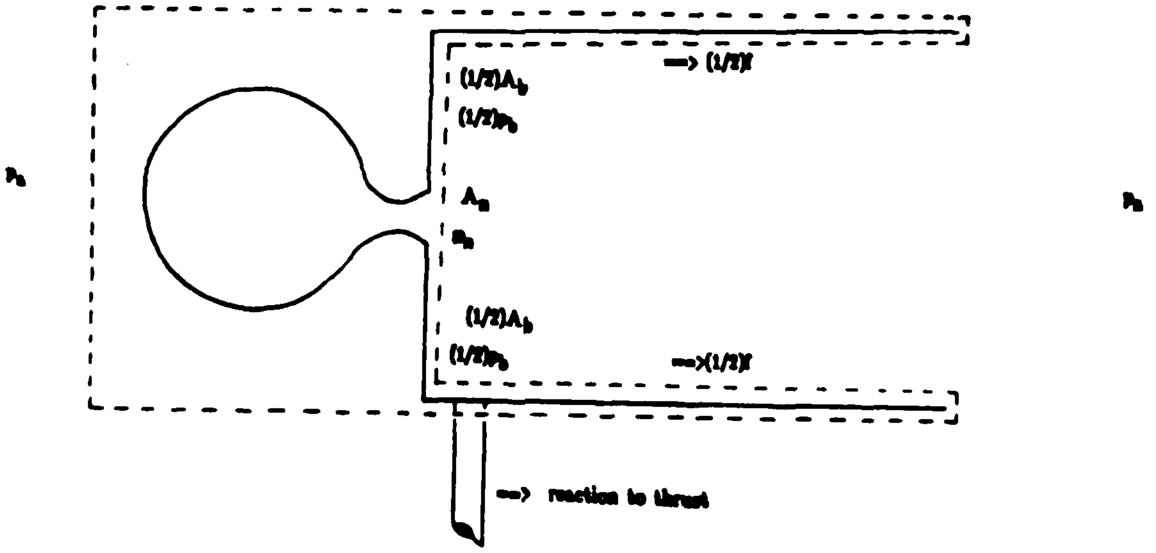


Figure 5 Second shroud control volume

This is the familiar rocket thrust equation. Now, by applying Eq. 1 to the CV shown in figure 5, another equation for the thrust of the shroud can be obtained

$$T_s = mu_n + (p_b - p_a)A_b + (p_n - p_a)A_n - f \qquad (3)$$

Recognizing that the thrust of the nozzle alone is given by

$$T_n = mu_n + (p_n - p_a)\, A_n \qquad (4)$$

then Eq 3 can be rewritten as

$$\Delta T = T_s - T_n = (p_b - p_a)\, A_b - f \qquad (5)$$

Eq 5 shows the importance of the shroud base pressure in increasing the thrust of the nozzle. If the shroud base pressure force is higher than the ambient pressure force plus the shear force, then the thrust increment is positive; otherwise the thrust increment is zero or negative.

GOVERNING EQUATIONS

The integral analysis given above can only give information on the gross properties of the flow, such as the thrust. For a more detailed analysis, however, the equations of motion expressing the conservation of mass, momentum and energy are required in the differential form. In generalized coordinates the conservation of mass equation is

$$\partial\rho/\partial t + \nabla \cdot (\rho V) = 0 \tag{6}$$

The conservation of linear momentum equation is

$$\rho(DV/Dt) = -\nabla p + \nabla \cdot \tau + \rho B \tag{7}$$

The conservation of energy equation is

$$\rho(De/Dt) = -p(\nabla \cdot V) + \Phi - \nabla \cdot q \tag{8}$$

where Φ is the dissipation function and is given by

$$\Phi = \tau : (\nabla \quad V) \tag{9}$$

In order to complete this set of equations the pressure, the stress tensor and the heat conduction vector must be related to the other variables. Only perfect gases will be considered here and thus the following equation of state will be used

$$p = \rho RT \tag{10}$$

where R is the universal gas constant. Since only Newtonian fluids will be studied, Stoke's law of friction can be used to relate the shear stress tensor to the gradient of the velocity vector as

$$\tau = \lambda \nabla \bullet V \, I + \mu[\nabla V + (\nabla V)^T] \qquad (11)$$

Finally, the heat flux vector is related to the gradient of the temperature through Fourier's law of heat conduction

$$q = - k\nabla T \qquad (12)$$

where the conductivity, k, is a function of the temperature.

For numerical solution these equations can be expressed in vector form in cartesian coordinates. Also, to simplify the finite difference algorithms for problems requiring complex grids, the equations are written in terms of a computational space. Eq 6 through Eq 12 are then written as

$$\begin{aligned} &\partial/\partial t(U/J) + \partial/\partial\xi\{J^{-1}[\xi_x(E_i - E_v) + \xi_y(F_i - F_v)]\} \\ &\quad + \partial/\partial\eta\{J^{-1}[\eta_x(E_i - E_v) + \eta_y(F_i - F_v)]\} = 0 \end{aligned} \qquad (13)$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{bmatrix} \qquad (14)$$

$$E_i = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E_t + p)u \end{bmatrix} \tag{15}$$

$$E_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} - v\tau_{xy} - q_x \end{bmatrix} \tag{16}$$

$$F_i = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E_t + p)v \end{bmatrix} \tag{17}$$

$$F_v = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - q_y \end{bmatrix} \tag{18}$$

$$E_t = \rho \, [e + 1/2 \, (u^2 + v^2)] \tag{19}$$

$$\begin{aligned}\tau_{xx} &= (2/3)\mu[2(\xi_x u_\xi+\eta_x u_\eta)-(\xi_y v_\xi+\eta_y v_\eta)]\\ \tau_{yy} &= (2/3)\mu[2(\xi_y v_\xi+\eta_y v_\eta)-(\xi_x u_\xi+\eta_x u_\eta)]\\ \tau_{xy} &= \mu(\xi_y u_\xi+\eta_y u_\eta+\xi_x v_\xi+\eta_x v_\eta)\end{aligned} \tag{20}$$

$$\begin{aligned}q_x &= -k\,(\xi_x T_\xi+\eta_x T_\eta)\\ q_y &= -k\,(\xi_y T_\xi+\eta_y T_\eta)\end{aligned} \tag{21}$$

For most high Reynold's number flows solved by finite differences, an approximate form of the Navier-Stokes equations (ANS) can be solved. The ANS equations to be used are sometimes incorrectly called the "parabolized" Navier-Stokes equations although they are actually parabolic/hyperbolic. The ANS equations are derived from the NS equations with the following reasoning. The grid densities allowed by computer memory and cost limitations usually only allow the viscous terms to be accurately resolved in one coordinate direction. The viscous terms in the direction normal to a surface are needed for such quantities as shear forces and energy transfer rates. The streamwise viscous terms, then, are not accurately resolved even when the NS equations are solved. The ANS equations are arrived at by simply neglecting the streamwise viscous terms and the time dependent terms of the full Navier-Stokes equations written in Cartesian coordinates and can be expressed as

$$\partial E_i^*/\partial x^* + \partial F_i^*/\partial y^* - \partial F_v^*/\partial y^* = 0 \tag{22}$$

$$E_i^* = \begin{bmatrix} \rho^* u^* \\ \rho^* u^{*2} + p^* \\ \rho^* u^* v^* \\ (E_t^* + p^*) u^* \end{bmatrix} \tag{23}$$

$$F_i^* = \begin{bmatrix} \rho^* v^* \\ \rho^* u^* v^* \\ \rho^* v^{*2} + p^* \\ (E_t^* + p^*) v^* \end{bmatrix} \tag{24}$$

$$F_v^* = \mu^*/\mathrm{Re}_L \begin{bmatrix} 0 \\ \partial u^*/\partial y^* \\ 4/3\ \partial v^*/\partial y^* \\ u^* \partial u^*/\partial y^* + (4/3) v^* \partial v^*/\partial y^* + [(\gamma-1) M_f^2 \mathrm{Pr}]^{-1}\ \partial T^*/\partial y^* \end{bmatrix} \tag{25}$$

Where the Reynold's number is defined by

$$\mathrm{Re}_L = \rho_f u_f L/\mu_f \tag{26}$$

and the equation of state is now

$$p^* = \rho^* T^* / \gamma M_1^2 \tag{27}$$

The variables are non-dimensionalized as follows

$$\begin{array}{lll} u^* = u/u_1 & v^* = v/u_1 & x^* = x/L \\ y^* = y/L & \rho^* = \rho/\rho_1 & e^* = e/u_1^2 \\ p^* = p/\rho_1 u_1^2 & \mu^* = \mu/\mu_1 & T^* = T/T_1 \\ & a^* = a/u_1 & \end{array} \tag{28}$$

In many problems the viscous/inviscid regions interact and to capture this interaction with the fewest number of grid points, stretching must be applied in the normal direction. To facilitate this stretching, the equations are written in computational coordinates with only the normal coordinate transformed. The ANS equations can be easily modified for these coordinates as follows. The normal direction in computational coordinates will be designated as eta. The normal direction derivative in physical space can then be written in terms of the normal direction derivative in computational space

$$\partial/\partial y^* = \partial\eta/\partial y^* \; \partial/\partial\eta$$

substitiuting into Eq 22 yields

$$\partial E_i^*/\partial x^* + \eta_{y^*}\partial F_i^*/\partial\eta - \eta_{y^*}\partial F_v^*/\partial\eta = 0$$

dividing by η_{y^*} gives

$$(1/\eta_{y^*})\,\partial E_i^*/\partial x^* + \partial F_i^*/\partial\eta - \partial F_v^*/\partial\eta = 0 \qquad (29)$$

where

$$F_v^* = \mu^*\eta_{y^*}/Re_L \begin{bmatrix} 0 \\ \partial u^*/\partial\eta \\ (4/3)\partial v^*/\partial\eta \\ u^*\partial u^*/\partial\eta + (4/3)v^*\partial v^*/\partial\eta + [(\gamma-1)M_l^2 Pr]^{-1}\,\partial T^*/\partial\eta \end{bmatrix} \qquad (30)$$

These equations are used in the current method.

Flux Splitting

Since the streamwise flux vector of the ANS equations only contains inviscid terms, then this vector may be "split" into two flux vectors. One flux vector, called the plus flux vector, models information propagation downstream. The negative flux vector models information propagation upstream.

Starting with the derivative of the streamwise flux vector

$$\partial E^*_i/\partial x^*$$

This can be written as

$$[Q]\,\partial U^*/\partial x^*$$

where

$$[Q] = \partial E_i^*/\partial U^* \tag{31}$$

Now E_i^* has the property (2:282)

$$E_i^* = [Q]U^* \tag{32}$$

A similarity transformation also exits so that (2:281)

$$[L]\,[Q]\,[R] = [\lambda] \tag{33}$$

where

[L] = matrix of left eigenvectors,

[R] = matrix of right eigenvectors,

$[\lambda]$ = matrix of eigenvalues

The eigenvalue matrix can be split into two parts. One part contains the positive eigenvalues and the other part contains the negative eigenvalues. This is written as

$$[\lambda] = [\lambda]^+ + [\lambda]^- \tag{34}$$

Substituting Eq 34 into Eq 33

$$[L]\,[Q]\,[R] = [\lambda]^{+} + [\lambda]^{-}$$

Solving for [Q]

$$\begin{aligned}[Q] &= [L]^{-1}\,[\lambda]^{+}\,[R]^{-1} + [L]^{-1}\,[\lambda]^{-}\,[R]^{-1} \\ &= [Q]^{+} + [Q]^{-} \qquad (35)\end{aligned}$$

The streamwise flux vector can then be written

$$\begin{aligned}E_i^{*} &= [Q]^{+}\,U^{*} + [Q]^{-}\,U^{*} \\ &= E_i^{*+} + E_i^{*-} \qquad (36)\end{aligned}$$

The exact elements of E_i^{+*} and E_i^{-*} may differ. The form used in this work was derived by Van Leer and is given by

$$E_i^{*+-} = \begin{bmatrix} E^{+-}{}_{mass} \\ E^{+-}{}_{mass}[(\gamma-1)u^{*} +- 2a^{*}]/\gamma \\ E^{+-}{}_{mass}v^{*} \\ E^{+-}{}_{mass}\{[[(\gamma-1)u^{*} +- 2a^{*}]^{2}/[2(\gamma^{2}-1)] + v^{*2}/2\} \end{bmatrix} \qquad (37)$$

where

$$E^{+-}{}_{mass} = +- \rho^{*}a^{*}[(1/2)\,(M_x +- 1)]^{2} \qquad (38)$$

When the streamwise Mach number is greater than or equal to one then

$$E_i^{+*} = E_i^*$$

and when the streamwise Mach number is less than or equal to negative one then

$$E_i^{-*} = E_i^*$$

When combined with the proper finite difference approximations, flux splitting provides a realistic model of the physics of a flow. E_i^{+*} models the information propagation in the downstream direction and E_i^{-*} models the information propagation in the upstream direction. Appropriate finite differences, then will take the correct information propagation into account when approximating the derivatives of E_i^{+*} and E_i^{-*}. For the E_i^{+*} derivative a backward difference would be used and for the E_i^{-*} derivative a forward difference would be used. An example of a backward difference is

$$(\partial u/\partial x)_{i,j} \sim [u_{i,j} - u_{i-1,j}]/\Delta x$$

and an example of a forward difference is

$$(\partial u/\partial x)_{i,j} \sim [u_{i+1,j} - u_{i,j}]/\Delta x$$

This splitting method will be applied to the current method described in Chapter III.

Couette Flow Solution

A Couette flow is an incompressible flow which consists of a stationary lower plate and a moving upper plate with a fluid between the plates. For the purposes of this work the fluid is air and the flow is considered two-dimensional and fully developed. With these assumptions the x-momentum equation of the NS or ANS equations reduces to

$$d^2u^*/dy^{*2} = 0$$

With the boundary conditions

$$u^*(y^*=0) = 0 \quad ; \quad u^*(y^*=1) = 1$$

the solution is

$$u^*(y^*) = y^* \qquad (39)$$

The energy equation is

$$(du^*/dy^*)^2 + [(\gamma-1)M_1{}^2Pr]^{-1}\,(d^2T^*/dy^{*2}) = 0$$

and the boundary condtions are

$$T^*(y^*=0) = 1 \quad ; \quad T^*(y^*=1) = 1$$

Substituting Eq 39 into the energy equation and applying the boundary conditions gives the solution

$$T^*(y^*) = 1.0 - \{[(\gamma-1)M_l^2 Pr]/2\} (y^{*2} - y^*) \tag{40}$$

III. Numerical Solution Methods

The important concepts of solving the governing equations using finite difference methods are discussed in this chapter. First, the governing equations are examined carefully with the aim of selecting an efficient numerical scheme for their solution. Thereafter, the finite difference form of the equations is presented, together with the initialization used for all variables. The procedure for solving the resulting algebraic simultaneous equations is described and some details about the implementation of the boundary conditions are discussed.

Inherently, the governing equations are an extremely stiff, non-linear system of equations and therefore their numerical solution deserves special care. The most widespread method of solution is the use of finite difference discretization. Simply put, the finite difference method converts partial differential equations (PDE) into a system of algebraic equations. These equations are then written at the discretized grid points used to approximate the flow domain. The system of equations can then be solved explicitly or implicitly.

In an explicit scheme the finite difference form of the differential equation is written so that only one unknown appears in the equation. For example, the heat equation

$$\partial u/\partial t = \alpha \, \partial^2 u/\partial x^2 \tag{41}$$

An explicit formulation is

$$(u_j^{n+1} - u_j^n)/\Delta t = (\alpha/(\Delta x)^2)\,[u_{j+1}^n - 2u_j^n + u_{j-1}^n] \qquad (42)$$

where n is the time index and j is the space index. Because an initial condition must be specified, then the values at n will always be known. Only one unknown, u_j^{n+1}, appears in the equation. Explicit methods are generally easy to code, but require very small time steps to satisfy stability restrictions. Therefore, many iterations and large computer times are required to reach a steady-state solution. This can be a severe problem when very fine grids must be used to capture flow details.

An implicit finite difference equation for the heat equation could be written by evaluating the terms on the right hand side of Eq 42 at the n+1 time level. The implicit schemes, then require the simultaneous solution of several equations. For a linear system of equations implicit methods are unconditionally stable. However, to be able to solve the discretized form of the governing equations implicitly, they must first be linearized. This leads to a matrix system of equations which can be solved to obtain the variables at each grid point, but the method is no longer unconditionally stable. The maximum time step is much larger than allowed for the explicit methods so that fewer iterations and lower computer times are required. Also, the explicit methods may fail for problems having very steep gradients so the only choice is to use an implicit method.

MacCormack's Explicit Method

The explicit method used in this work is a two-dimensional version of MacCormack's predictor-corrector scheme(2:479-489). This scheme was chosen because it was readily available and had been successfully used in earlier AFIT studies(14). The empirical stability constraint on this scheme is given in cartesian coordinates by (2:484)

$$\Delta t \leq \sigma(\Delta t)_{CFL}/(1 + 2/Re_{\Delta}) \qquad (43)$$

where sigma is approximately 0.9 and $(\Delta t)_{CFL}$ is the inviscid CFL condition given by (2:484)

$$(\Delta t)_{CFL} \leq [|u|/x + |v|/y + a(1/(\Delta x)^2 + 1/(\Delta y)^2)^{1/2}]^{-1} \qquad (44)$$

The particular code used in this study was obtained from the Computational Aerodynamics Group, Flight Dynamics Lab, Wright-Patterson Air Force Base. The code has been vectorized for the Cray XMP supercomputer and contains a Baldwin-Lomax arithmetic turbulence model, although all the problems considered here will be laminar. The explicit code discretizes the dimensional form of the uns' ady NS equations in computational coordinates with all variables in English units. For more details on MacCormack's explicit scheme see reference 2 and for the details on the code see reference 14.

Space Marching Algorithms

The first method follows the method used by Vigneron to solve the ANS equations using a space marching scheme. This allows the solution of a predominately supersonic flow to be obtained with a single sweep of the problem domain. In the subsonic portions of the domain information traveling upstream must be suppressed, and Vigneron achieved this by "splitting" the streamwise pressure gradient from the streamwise flux vector. Eq 22 would then be

$$\partial E_i^{*\omega}/\partial x^* + \partial P^*/\partial x^* + \partial F_i^*/\partial y^* - \partial F_v^*/\partial y^* = 0 \qquad (45)$$

where

$$E_i^{*\omega} = \begin{bmatrix} \rho^* u^* \\ \rho^* u^{*2} + \omega p^* \\ \rho^* u^* v^* \\ (E_t^* + p^*)u^* \end{bmatrix} \qquad (46)$$

and

$$P^* = \begin{bmatrix} 0 \\ (1-\omega)p^* \\ 0 \\ 0 \end{bmatrix} \qquad (47)$$

Vigneron's stablility analysis showed that for the solution to be marched in the streamwise direction omega must satisfy

$$\omega < \gamma M_x^2/[1 + (\gamma-1)M_x^2] \qquad (48)$$

For computational purposes a safety factor is applied to Eq 48

$$\omega = \sigma\gamma M_x^2/[1 + (\gamma-1)M_x^2] \qquad (49)$$

where sigma is usually 0.9. The Beam-Warming finite difference equation is given by

$$\Delta^i E = \Delta x \partial/\partial x\ (\Delta^i E) + \Delta x \partial/\partial x\ (E^i) \qquad (50)$$

where

$$\Delta^i E = E^{i+1} - E^i \qquad (51)$$

This is the Euler implicit form which is first order accurate in the marching (streamwise) direction and second order accurate in the normal direction. When Eq 50 is applied to Eq 45 the streamwise pressure gradient term is treated as an explicit term and taken to the right hand side (RHS). This term is then backward differenced to retain the space marching technique. However, when this is done the streamwise step size must meet the stability requirement

$$(\Delta x)_{min} > \{((1/4)(\rho u/\mu)[(1/M_x^2) - 1](\Delta y)^2\}/\{\gamma \sin^2(\beta/2)\} \qquad (52)$$

where β is the wave number. This requirement can be a severe restriction for low-speed flows. This restriction can be removed using two methods -- neglecting the streamwise pressure gradient term and specifying the pressure field. Neglecting the streamwise pressure gradient is an obvious way to remove the restriction, but the method must then be used only for flows with small or no streamwise pressure gradient. Specifying the pressure field allows the streamwise pressure gradient term to be forward differenced and again removes the stability restriction. However, if the pressure field is not exactly known, then the method must be modified for global iterations where the field is swept until the correct solution is reached. This diminishes the attractiveness of the single-sweep marching technique, but only a few sweeps are usually required. The code developed for the present work using Vigneron's method will be referred to as PANS. Vigneron's method closely follows the current method except that the PANS code is a single sweep code while the current method is written for global iterations.

The goal here is to develop a method which can successfully be applied to large separation regions. Because the flux-splitting correctly models the characteristic propagation of information in both subsonic and supersonic flows it would be a better candidate for a successful method. However, to correctly apply the flux-splitting a forward difference must be applied to the negative flux term. This means that the marching scheme must be modified for global iterations since the forward difference requires data that has not been calculated. To solve this problem the negative flux term will be calculated using "old" data, that is, data obtained from the previous solution sweep.

The derivation of the current algorithm follows that given for Vigneron's method in reference 2. Putting Eq 22 into delta form and substituting into Eq 50 gives

$$\Delta^I E_i^* = -\Delta x^*[\partial/\partial y^* \, (\Delta^I F_i^* - \Delta^I F_v^*)] - \Delta x^*[\partial/\partial y^* \, (F_i^{*I} - F_v^{*I})] \qquad (53)$$

From Eq 36 the following equation can be written

$$\Delta^I E_i^* = \Delta^I E_i^{+*} + \Delta^I E_i^{-*}$$

Substituting Eq 36 into Eq 53 results in

$$\Delta^I E_i^{*+} + \Delta^I E_i^{*-} = -\Delta x^*[\partial/\partial y^* \, (\Delta^I F_i^* - \Delta^I F_v^*)] - \Delta^*[\partial/\partial y^* \, (F_i^* - F_i^*)] \qquad (54)$$

This equation is linearized according to

$$\Delta^I F_i^* \sim [R]^I \Delta^I U^* \qquad (55)$$

$$\Delta^I F_v^* \sim [W]^I \Delta^I U^* \qquad (56)$$

where

$$[R]^I = (\partial F_i^*/\partial U^*)^I \qquad (57)$$

$$[W]^{I} = (\partial F_{v}^{*}/\partial U^{*})^{I} \tag{58}$$

According to the flux-splitting method the positive streamwise flux vector for subsonic streamwise Mach number is written

$$\Delta^{I}E^{*+} \sim [Q]^{+I}\Delta^{I}U^{*} \tag{59}$$

and when the streamwise mach number is supersonic

$$\Delta^{I}E^{*+}=\Delta^{I}E^{*} \sim [Q]^{I}\Delta^{I}U^{*} \tag{60}$$

The elements of [Q], $[Q]^{+}$, [R] and [V] are given in Appendix B. Now Eq 54 becomes

$$\begin{aligned} [Q]^{+I}\Delta^{I}U^{*} + \Delta^{I}E_{i}^{*-} = -\Delta x^{*}[\partial/\partial y^{*}([R]^{I}\Delta^{I}U^{*} - [W]^{I}\Delta^{I}U^{*})] \\ -\Delta x^{*}[\partial/\partial y^{*} (F_{i}^{*} - F_{v}^{*})] \end{aligned} \tag{61}$$

Taking the implicit terms to the left-hand-side (LHS) and the explicit terms to the right-hand-side (RHS) results in

$$\begin{aligned} [Q]^{+I}\Delta^{I}U^{*} + \Delta x^{*}[\partial/\partial y^{*} ([R]^{I}\Delta^{I}U^{*} - [W]^{I}\Delta^{I}U^{*})] \\ = -\Delta x^{*}[\partial/\partial y^{*} (F_{i}^{*} - F_{v}^{*})] - \Delta^{I}E_{i}^{-*} \end{aligned} \tag{62}$$

The first partial derivatives are approximated by second order central differences, for example

$$\partial/\partial y^* ([R]^j \Delta^l U^*) \sim \{([R]^j \Delta^l U^*)_{j+1} - ([R]^j \Delta^l U^*)_{j-1}\}/(2\Delta y^*) \qquad (63)$$

and for the second derivative terms

$$\begin{aligned}\partial/\partial y^* [\mu^* \partial/\partial y^* (w\Delta^l U_l^*)] \sim {} & (\mu_j^* + \mu_{j+1}^*)[(w\Delta^l U_l^*)_{j+1} - (w\Delta^l U_l^*)_j]/[2(\Delta y^*)^2] \\ & - (\mu_j^* + \mu^*_{j-1})[(w\Delta^l U_l^*)_j - (w\Delta^l U_l^*)_{j-1}]/[2(\Delta y^*)^2]\end{aligned} \qquad (64)$$

This gives the LHS a block tri-diagonal structure whose elements are given in Appendix C. The negative split-flux vector on the RHS is evaluated using a first order forward difference

$$\Delta^l E_j^{*-} \sim \Delta^{l+1} E_j^{*-} \qquad (65)$$

The viscosity coefficients were evaluated using Sutherland's law

$$\mu^* = \mu_0^* T_r^{1/2} \{T^{*\,3/2}/(T^* + T_0^*)\} \qquad (66)$$

The current code, referred to as FANS, is carried out as follows:

Step 1: Specify the initial conditions for the whole flowfield. For the PANS code only the conditions at I=1 need to be specified.

Step 2: Calculate E_i^{-*} for the whole flowfield. Skip this step for the PANS code.

Step 3: Calculate the streamwise Mach number and the viscosity coefficient for each J (normal) point at the I station. For the PANS code calculate omega instead of the streamwise Mach number.

Step 4: Calculate the F_i^* vector for each J at I

Step 5: Calculate the derivative of F_i^* with respect to y for each J at I

Step 6: Calculate the derivative of F_v^* with respect to y for each J at I

Step 7: Calculate the RHS vectors for each J

Step 8: Calculate the linearization Jacobians. For the FANS code if the local streamwise Mach number is less than one then calculate $[Q]^+$ instead of [Q]

Step 9: Calculate the LHS matrices

Step 10: Solve for delta-U at I+1 using the block tri-diagonal matrix solver

Step 11: Update the solution using

$$U^{*l+1} = U^{*l} + \Delta^l U^*$$

Step 12: Calculate the explicit boundary conditions

Step 13: Repeat Steps 3 through 12 until the solution for the last streamwise station is obtained. The first sweep of the flowfield is finished and the PANS code stops here

Step 14: Go to Step 2 and repeat the process until the solution has converged.

Please note that the PANS code is not written in computational coordinates. Therefore, a grid does not need to be generated; simply specify the streamwise step size, the number of points in the streamwise direction, the normal step size and the number of points in the normal direction. The FANS code does, however, require a grid to be generated for the normal coordinate.

Although both codes were run on the Cray XMP, no effort was made to vectorize the codes. In fact, vectorization was not implemented during the compilation of the codes.

IV Results and Discussion

In this chapter each problem that was solved will be briefly discussed. The problems are the following:

- Shock/Boundary Layer Interaction (explicit)
- Single Nozzle in a Short Shroud (explicit).
- Couette Flow (PANS, FANS)
- Supersonic Boundary Layer (PANS, FANS)
- Shock/Boundary Layer Interaction (FANS)

A brief description of each problem will be followed by the results and finally a discussion of the signifigance of the results.

Shock/Boundary Layer Interaction (explicit)

This problem consists of a shock wave impinging on the boundary layer over a flat plate as shown in figure 6. The results obtained here will be compared to the results of Thomas and Walters(18) and also the results of Beam and Warming(4).

The computational grid, shown in figure 7, is the same as used by Beam and Warming but the upper boundary was increased to allow the leading edge shock to exit through the downstream boundary resulting in a 32x62 cartesian grid. The boundary conditions for the supersonic inflow were freestream up to the point where the shock entered the domain and then the inviscid shock jump conditions from that point to the top of the domain. The conditions along the top of the domain were also set to the shock jump conditions and along the outflow boundaries the

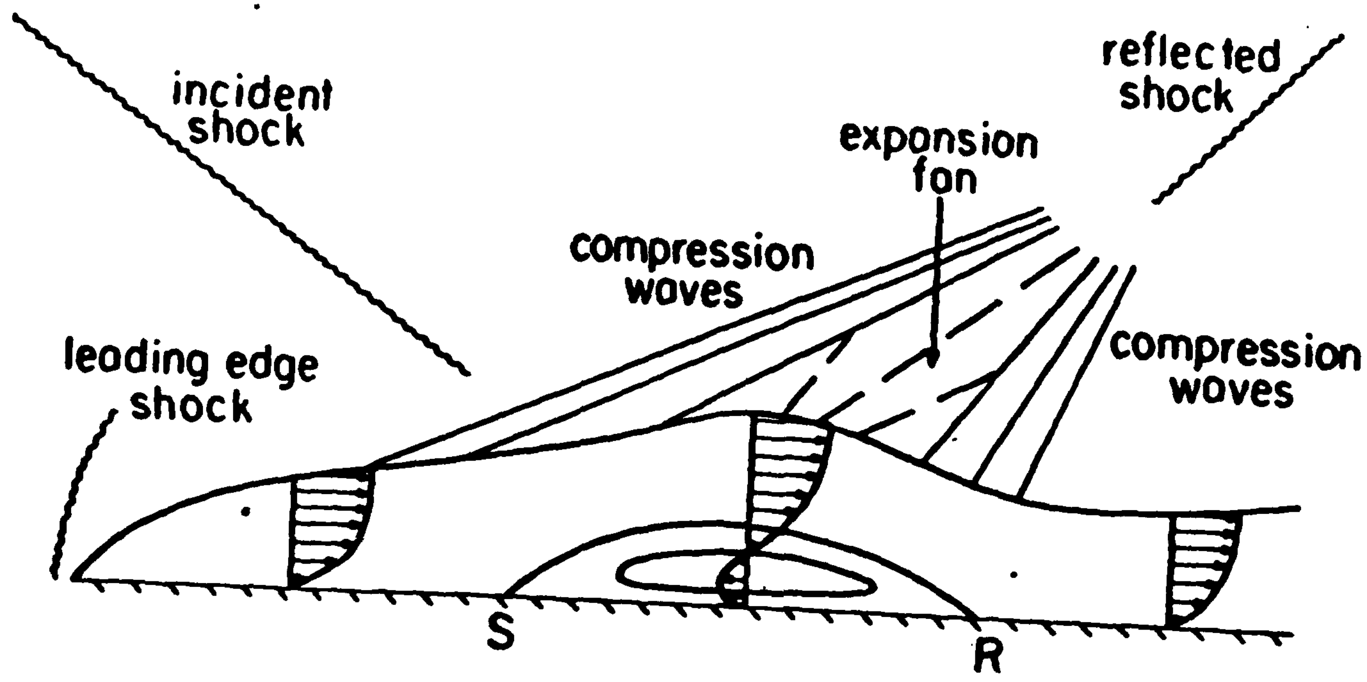


Figure 6 Shock/Boundary Layer Interaction

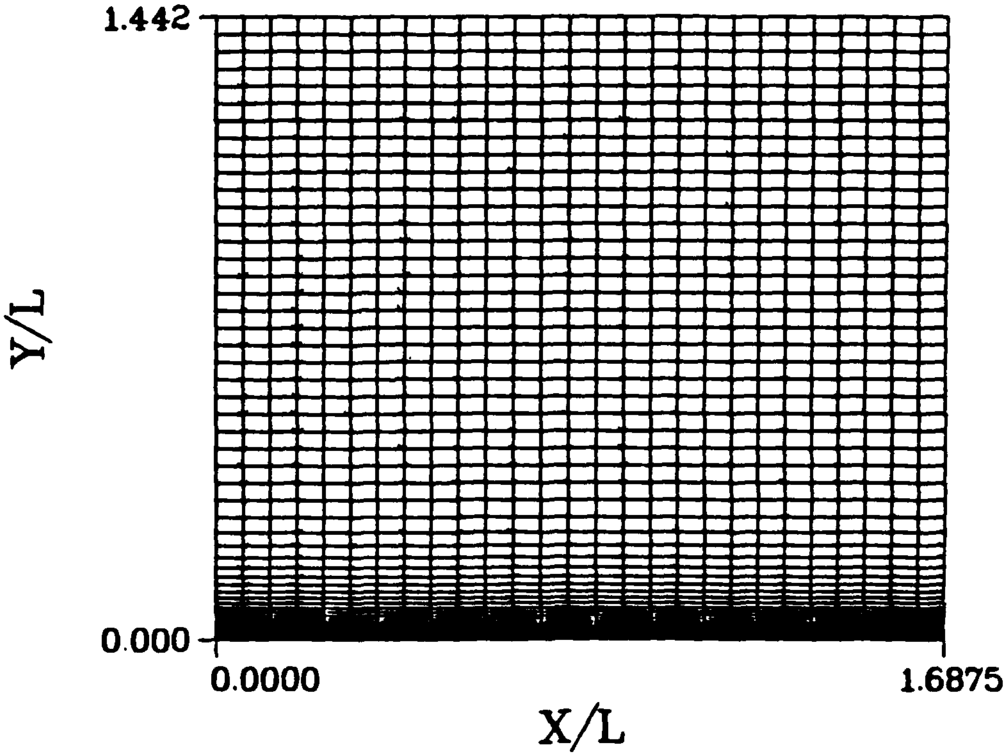


Figure 7 Flat plate physical grid

conditions were extrapolated from values of interior points. Along the bottom of the domain symmetry conditions were used before the leading edge of the plate and on the plate no-slip and constant wall temperature conditions were applied. The wall temperature was set to the free-stream temperature of 460 Rankine.

Figure 8 shows the pressure contours, Figure 9 shows the pressure along the bottom of the domain and Figure 10 shows the friction coefficient along the plate. The results agree well with the solution obtained by Beam and Warming (4). However, a few problems should be pointed out. First, a large amount of damping was necessary to achieve a smooth solution. Next, as shown by Thomas and Walters (18), a grid density of 60x90 points is necessary to accurately resolve the separation region. The grid used by the explicit code was 32x62 points and required 22,000 iterations with a run time of 968 seconds on the Cray XMP. The efficiency parameter, CPU time in seconds divided by the number of grid points divided by the number of iterations, was 2.2×10^{-6} . Only one streamwise station exhibited the negative streamwise velocity characteristic of the separation region. Finally, the separation region of the flow was very slow in developing which reveals another weakness of the explicit code -- slow convergence of the solution in subsonic regions.

Single Nozzle in a Short Shroud

The problem to be solved now is the two-dimensional single nozzle in a short shroud, experiment 3B from reference 11.

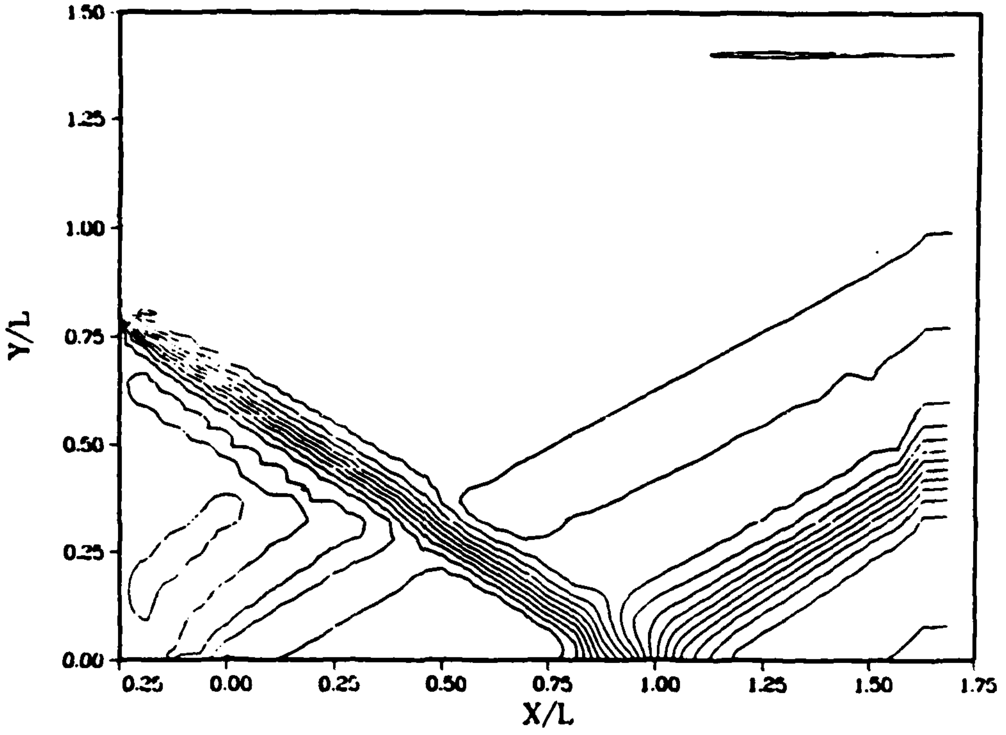


Figure 8 Shock/Boundary Layer: Pressure contours using the explicit code

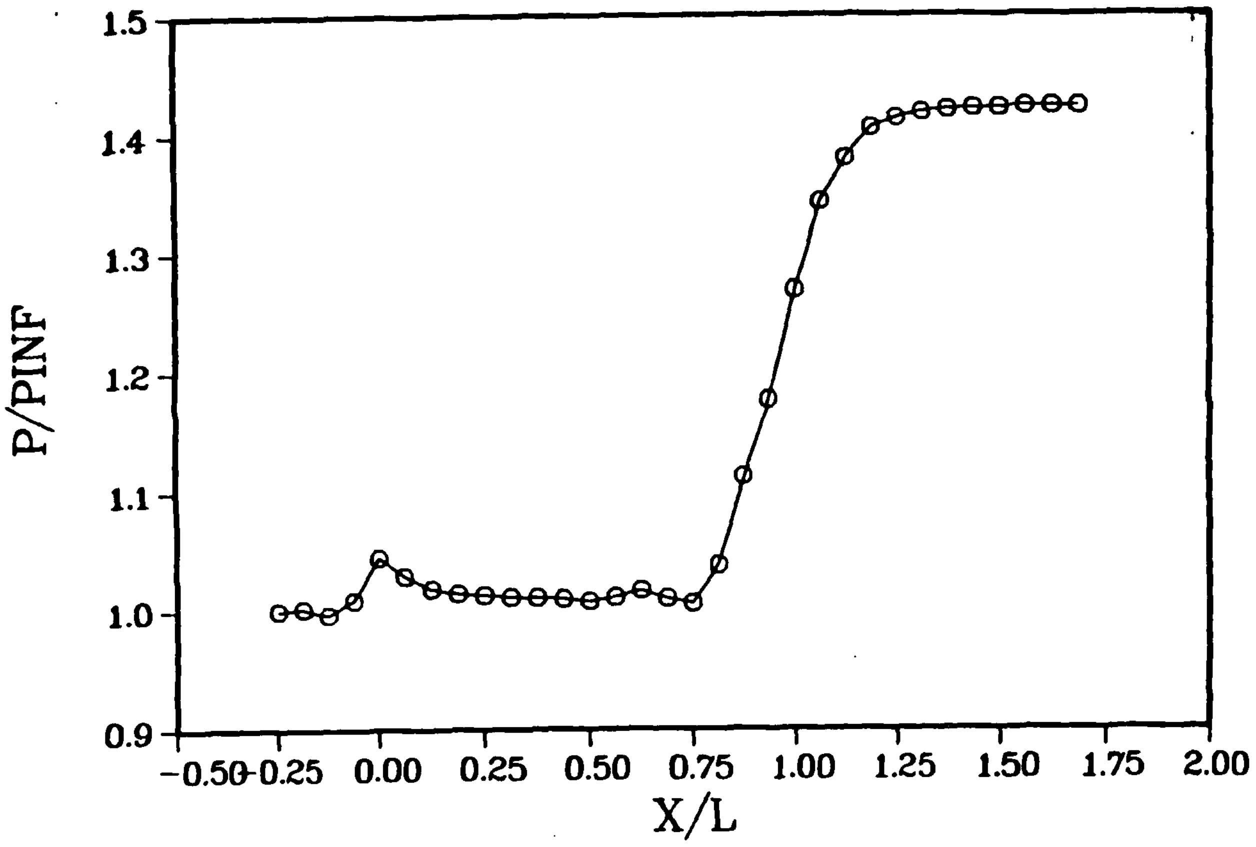


Figure 9 Shock/Boundary Layer: Wall pressure using the explicit code

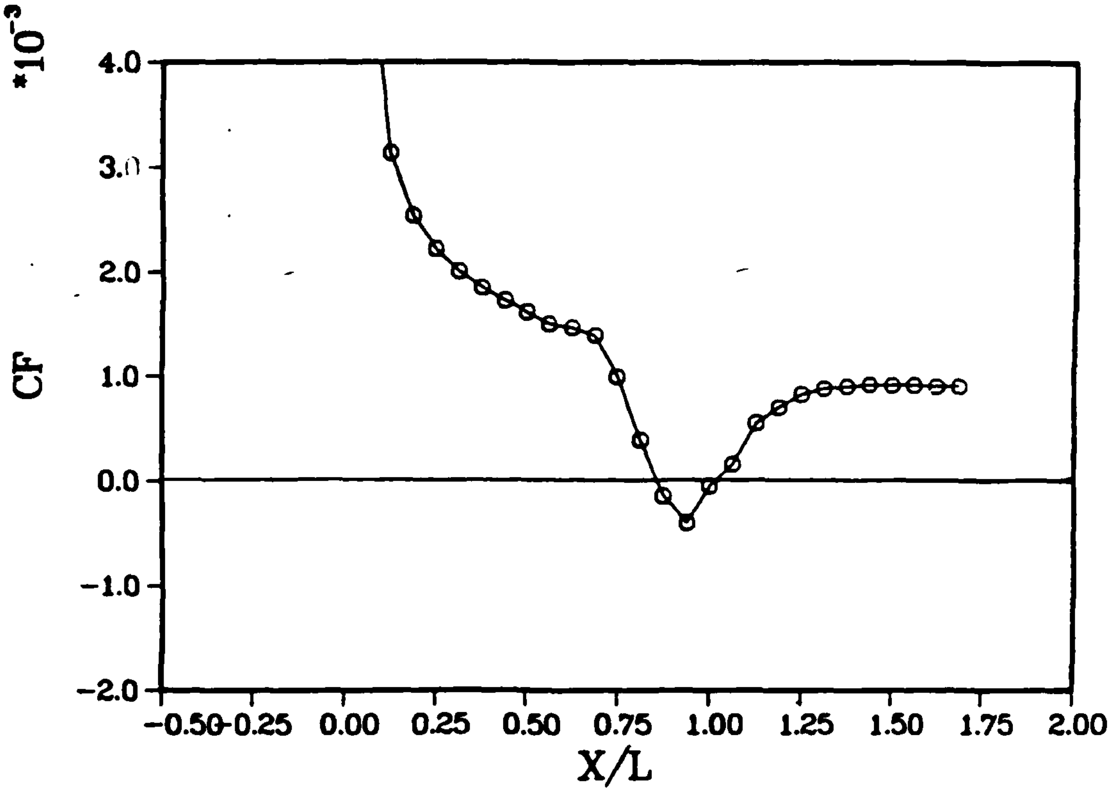


Figure 10 Shock/Boundary Layer: Friction coefficient using the explicit code

This is a cold air flow where a chamber was pressurized to 13680 pounds per square foot (psf) and the exhaust chamber was evacuated to about 36 psf. The pressure chamber exhausted into the test section where the nozzle block and shroud block were located. The exhaust chamber pressure would gradually increase during the experiment resulting in an unsteady flow where the nozzle exhaust plume would eventually separate from the shroud wall. The nozzle had a 4:1 exit area to throat area with an isentropic exit Mach number of 2.94. More detatils on the dimensions are given in Appendix A. For this numerical solution the exhaust chamber pressure will be assumed to be a steady 36 psf so that a steady solution with the exhaust plume attached to the shroud wall can be computed.

In order to keep the cost and computer time to a minimum the number of grid points and the location of the grid points in the domain must be carefully selected. The flow domain was chosen to be a two-dimensional cartesian coordinate system with x representing the streamwise direction and y the normal direction. The reference length was chosen to be the length of the shroud -- two inches. Because of the large separation region in this flow a 45x45 grid was chosen as the best trade-off between run time and accuracy. To improve the accuracy of the results the grid points were clustered around high gradient regions and regions of interest. In this case, the high gradient region was near the exit of the nozzle and the base wall was the region of interest since this is where the pressures from the experiment were taken. After about five short computer runs the grid in Figure 11 was

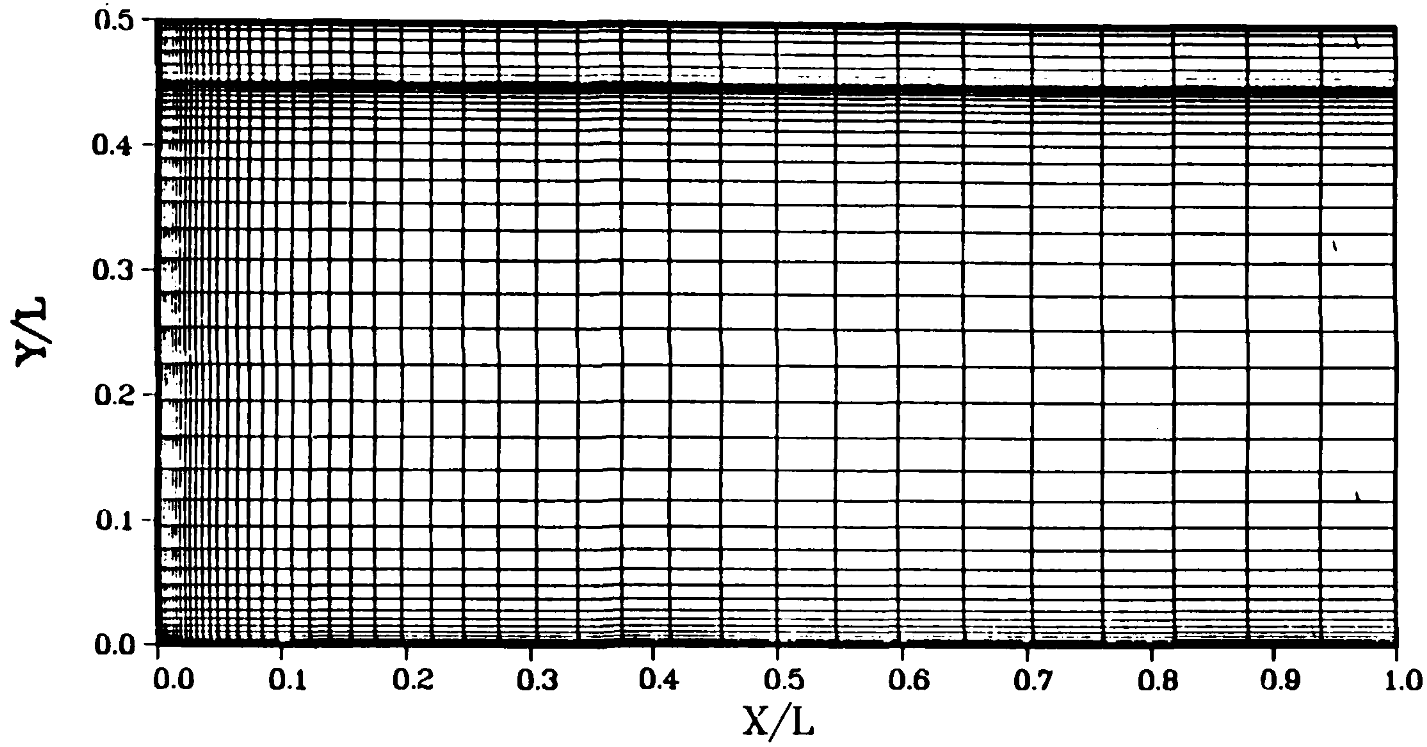


Figure 11 Shroud Grid

chosen as the best for the problem.

Next, the boundary conditions will be discussed. There are five important regions where boundary conditions must be applied in this problem -- the shroud wall, the base wall, the nozzle exit, the symmetry line and the shroud exit. For both the shroud wall and the base wall no-slip and constant wall temperature conditions were used; to calculate the density at the wall the normal gradient of the pressure was assumed to be zero. The nozzle exit (shroud inlet) conditions were taken to be the exit conditions for an isentropic quasi-one-dimensional nozzle with stagnation conditions equal to the conditions in the pressure chamber of the experiment. In order to increase the grid density the flow was assumed to be symmetrical about the centerline of the shroud so that the upper boundary of the domain is a symmetry line. Symmety line conditions were set to be zero gradient except for the normal velocity component which was set to zero. Finally, since the shroud exit is a predominately supersonic flow then the shroud exit conditions can be calculated from interior points by simple extrapolation.

To arrive at realistic initial conditions the problem must be thought of as an unsteady problem which reaches a steady-state solution. The initial conditions would then be that of the exhaust chamber before the pressure chamber valve was released, which were taken to be the following:

$$p_0 = 36 \text{ psf}$$

$$u_0 = 0.0 \text{ ft/sec}$$

$$v_0 = 0.0 \text{ ft/sec}$$

T_0 = 460 Rankine

ρ_0 = 0.000045553 slugs/cu ft

except at the nozzle exit.

The results of the shroud computations are presented in Figures 12 through 16. The Mach contour plot (Figure 12), pressure contour plot (Figure 13) and velocity vector plot (Figure 14) reveal the important features of the flow. The Mach contour plot shows the expansion of the nozzle plume from a nozzle exit Mach number of approximately three to a shroud exit Mach number of approximately five and a half. Somewhat harder to see is the compression shock formed when the exhaust plume strikes the shroud wall. This feature is located in the lower right hand corner of the plot and shows a drop in Mach number from 5.5 to 2.5 across the shock.

Details of the seperation region are also revealed in the Mach contour plot. This region has a very complex flow pattern containing many eddies and a surprisingly wide range of Mach contours. For instance, embedded in the predominately subsonic separation region is a supersonic region. This region stretches from the plume/wall interaction area towards the shroud base. This phenomenon also shows up in the velocity vector plot. Here, the arrows close to the plume/shroud interaction region are seen to point towards the shroud base. Moving closer to the base, the arrows gradually lengthen, indicating a fluid velocity increase, until the flow turns parrallel to wall and the velocity decreases.

Both the velocity vector plot and the pressure contour plot

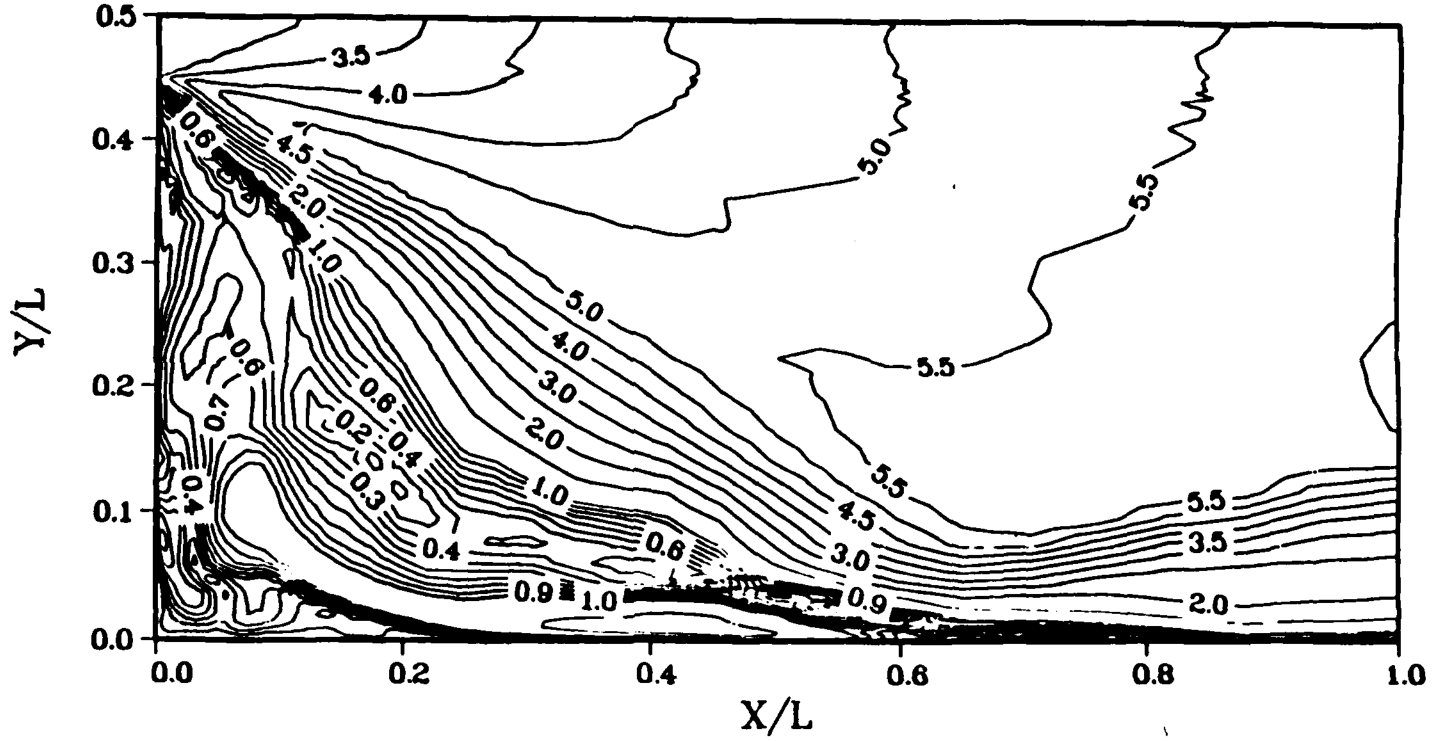


Figure 12 Shroud: Mach contours

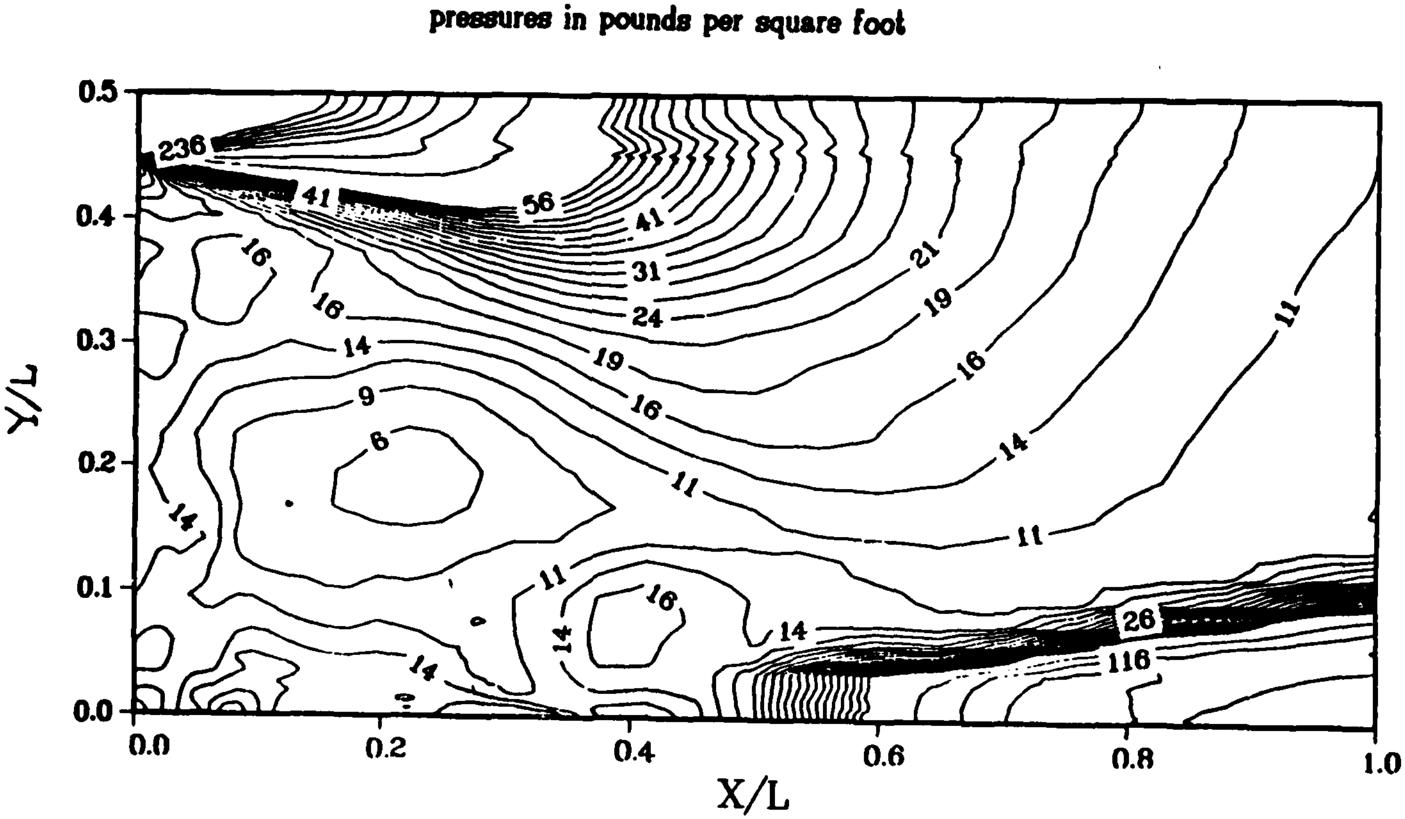


Figure 13 Shroud: Pressure contours

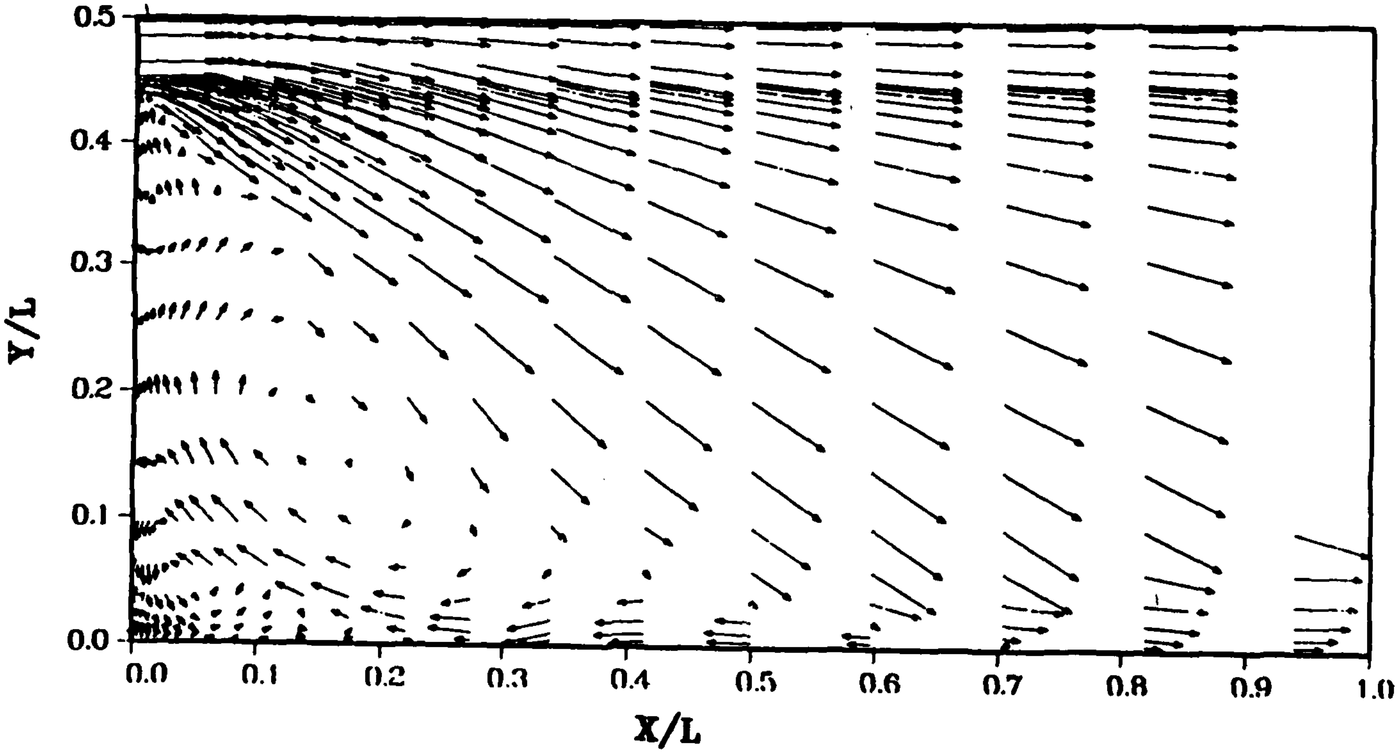


Figure 14 Shroud: Velocity vectors

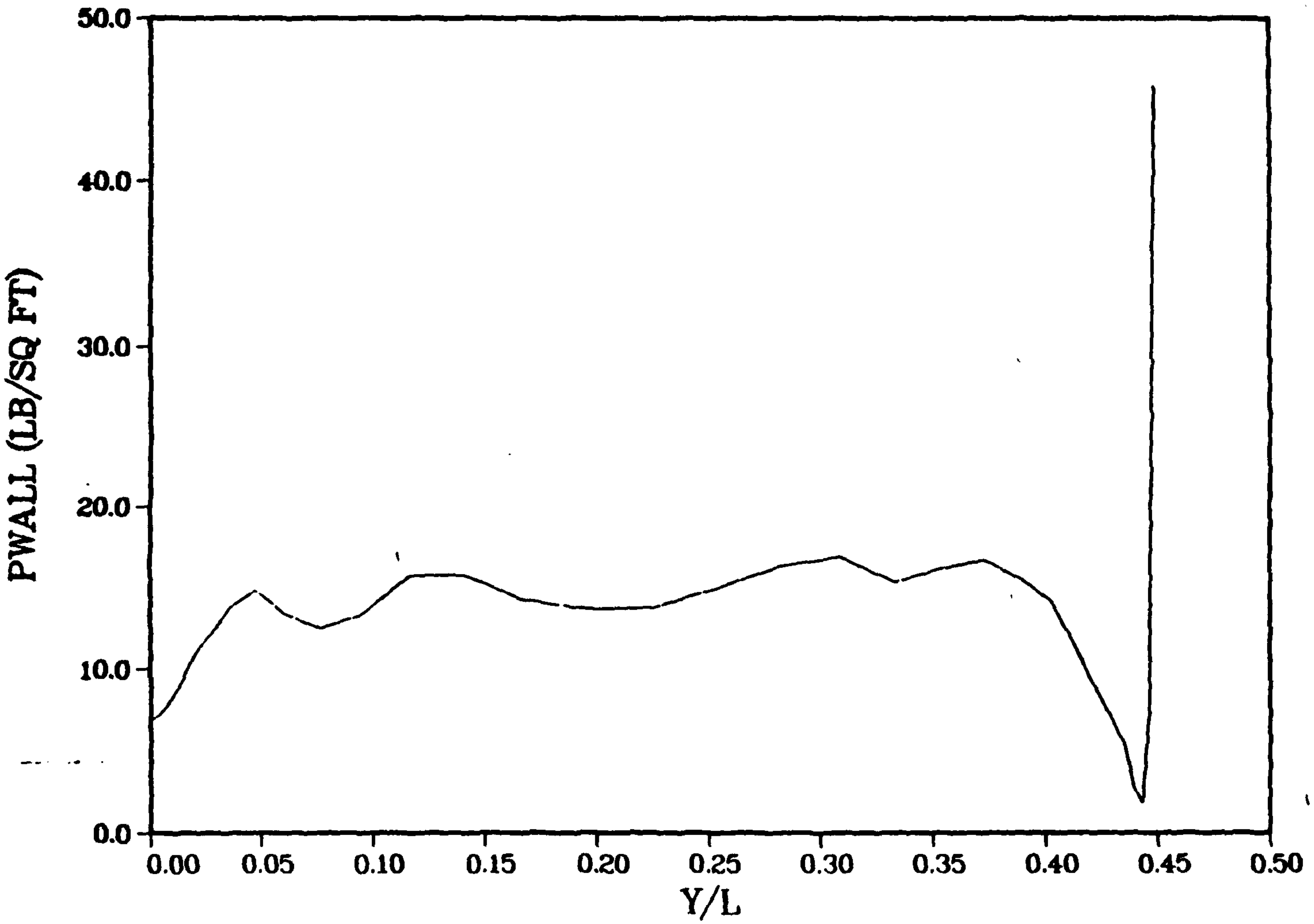


Figure 15 Shroud: Base pressure

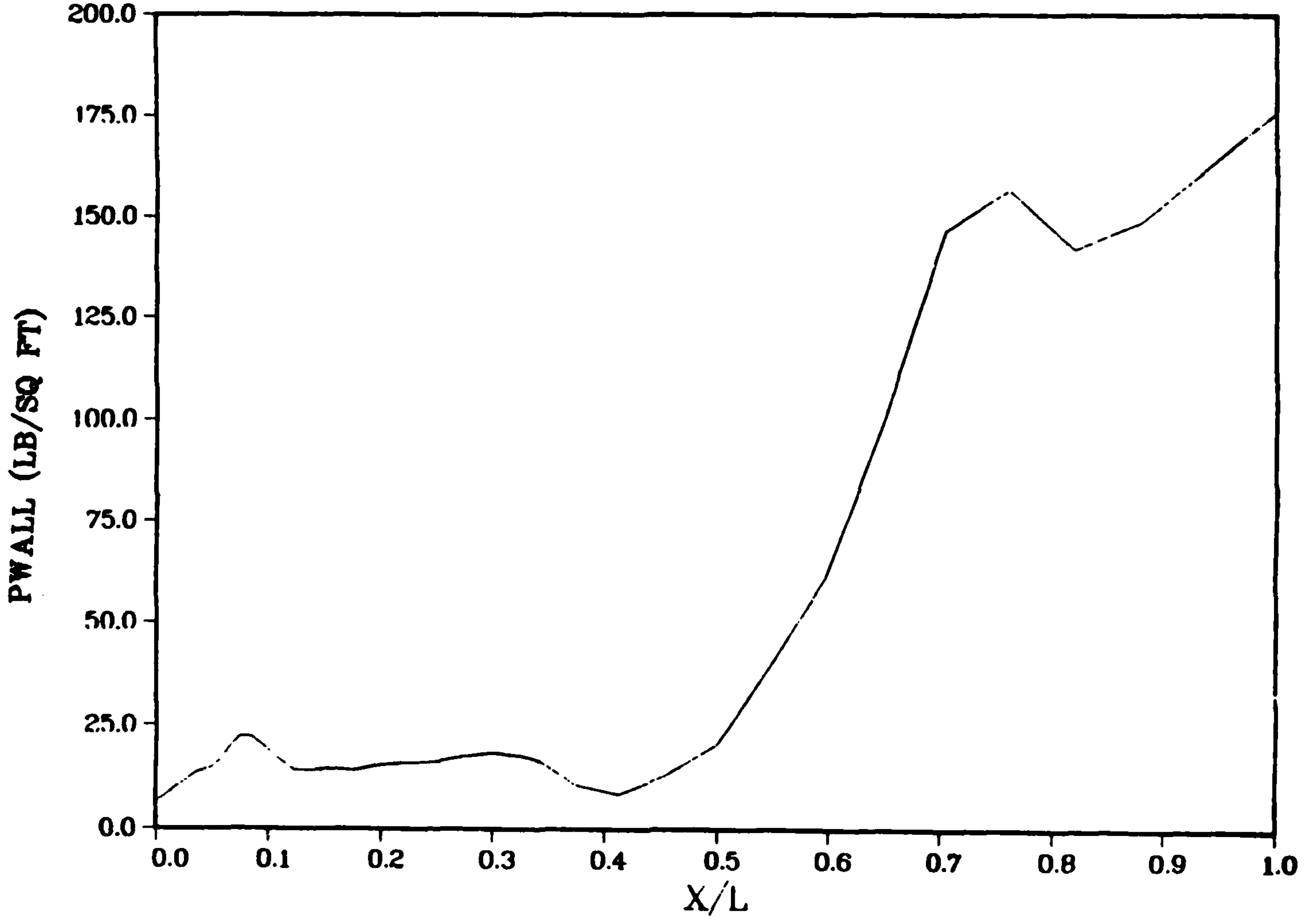


Figure 16 Shroud: Wall pressure

where w is the width of the shroud which was two inches. The ambient pressure is a constant and is taken to be thirty six pounds per square foot. The change in thrust is found to be a loss of 0.5 pounds force from a nozzle thrust of 14.7 pounds force. This is a 3.7 percent loss. If the ambient pressure is chosen to be zero, however, a thrust gain of 0.4 pounds force is calculated, which is a 2.2 percent gain. Thus, only under certain ambient conditions does the shroud increase the thrust over the nozzle alone.

The run time on the Cray XMP for eighty thousand time steps was 2373 seconds. The efficiency parameter is then 1.4×10^{-5} .

Couette Flow

The Couette flow problem has been used previously as a model problem to test codes (4,18). A Couette flow consists of a stationary lower plate and a moving upper plate with a fluid between the plates. For the purposes of this work the fluid is air and the flow is considered two-dimensional and steady state. The characteristic length is the distance between the plates. The exact solution to this problem for an incompressible flow is, for the velocity distribution given by Eq 39 and for the temperature distribution is given by Eq 40. The pressure and the coefficient of viscosity were assumed constant and the density is calculated from the equation of state.

Figure 17 shows the results from the PANS and FANS codes for the Couette flow problem. Both codes are in very good agreement with the exact solution. Besides providing a dry run for the

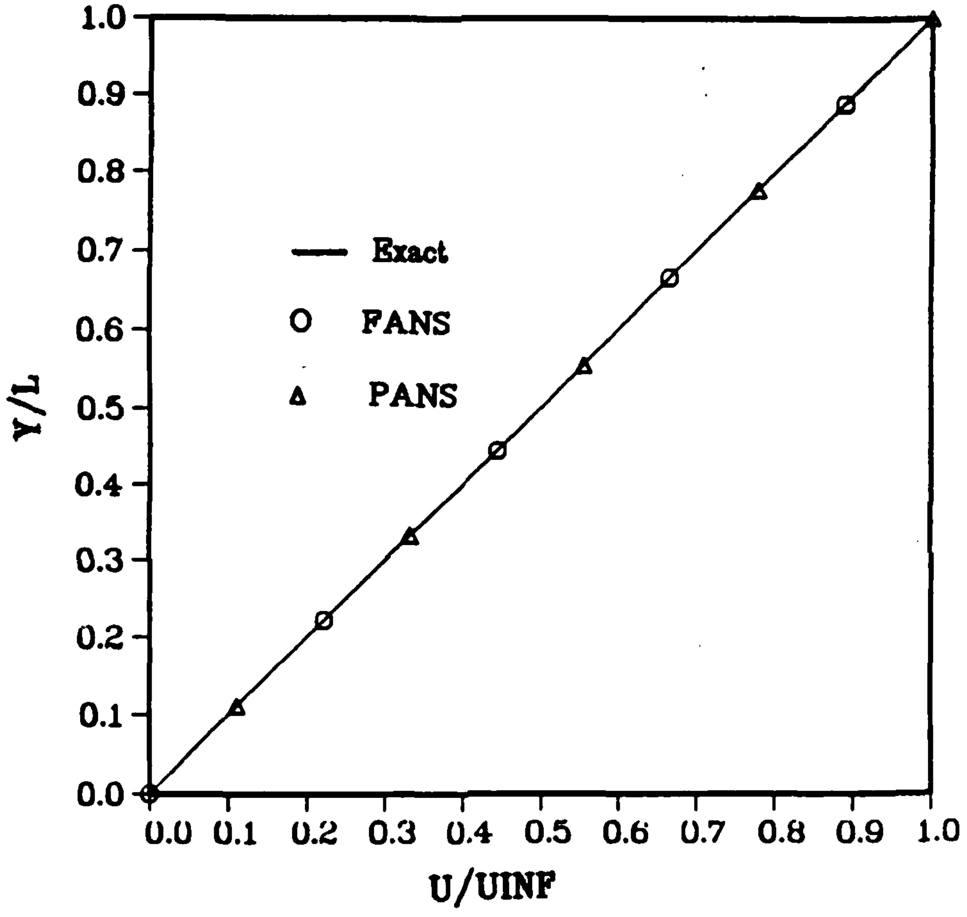


Figure 17 Couette flow: velocity distribution

codes, this exercise also demonstrated one very important point. In both codes the flow fields were initialized so that forward differences could be used on the "split" terms and, as mentioned earlier, this should relax the step size restriction. Eq 49 applied to this problem yields

$$(\Delta x^*)_{min} > 1.6$$

The step size used, however, was

$$\Delta x^* = 0.1$$

In short, initializing the flow field and using forward differences on the split terms significantly reduces the size of the streamwise step size.

Supersonic Boundary Layer

As a second, more severe test for the PANS and FANS codes the supersonic, laminar boundary layer flow previously worked by Lawrence and Tannehill (10) will be solved.

This is a flat plate, zero pressure gradient flow. The freestream conditions were the following:

$$M_f = 2.0$$

$$T_f = 399.0 \text{ Rankine}$$

$$Re_L = 1.65 \times 10^6$$

The characteristic length was taken to be one meter measured from the leading edge of the plate.

The flow was initialized at x^* = 0.305 using data from the boundary layer code of Cebeci (6). The initial data was then marched to x^* = 0.915 and compared to data at the same location from Cebeci's code. Step sizes used were Δx^* = 0.001 and Δy^* = 0.1524×10^{-3} with six hundred and twelve grid points in the x-direction and forty grid points in the y-direction.

Figure 18 presents the tangential velocity profile computed with a single sweep of the PANS and FANS codes compared to the boundary layer code. Figure 19 presents the temperature profile. Clearly, the PANS code provides a more accurate solution than the FANS code for a single sweep. This is expected, since for a single sweep the pressure vector was set to zero in the PANS code and the negative flux vector was set to zero in the FANS code. Thus, the PANS code only neglects a portion of the upstream traveling information, but the FANS code completely eliminates this information.

Next, the FANS code was applied to the same problem, but using global iterations. Figures 20 and 21 compare the FANS code run for four sweeps to the PANS code run for one sweep and the boundary layer code. Now that the minus flux term has been included in the FANS code, the solution is just as accurate as the PANS solution.

An indication of the relative computer effort required by the two codes is given by a comparison of CPU times. The figure of merit here is the CPU time in seconds divided by the number of sweeps divided by the number of grid points. For the FANS code

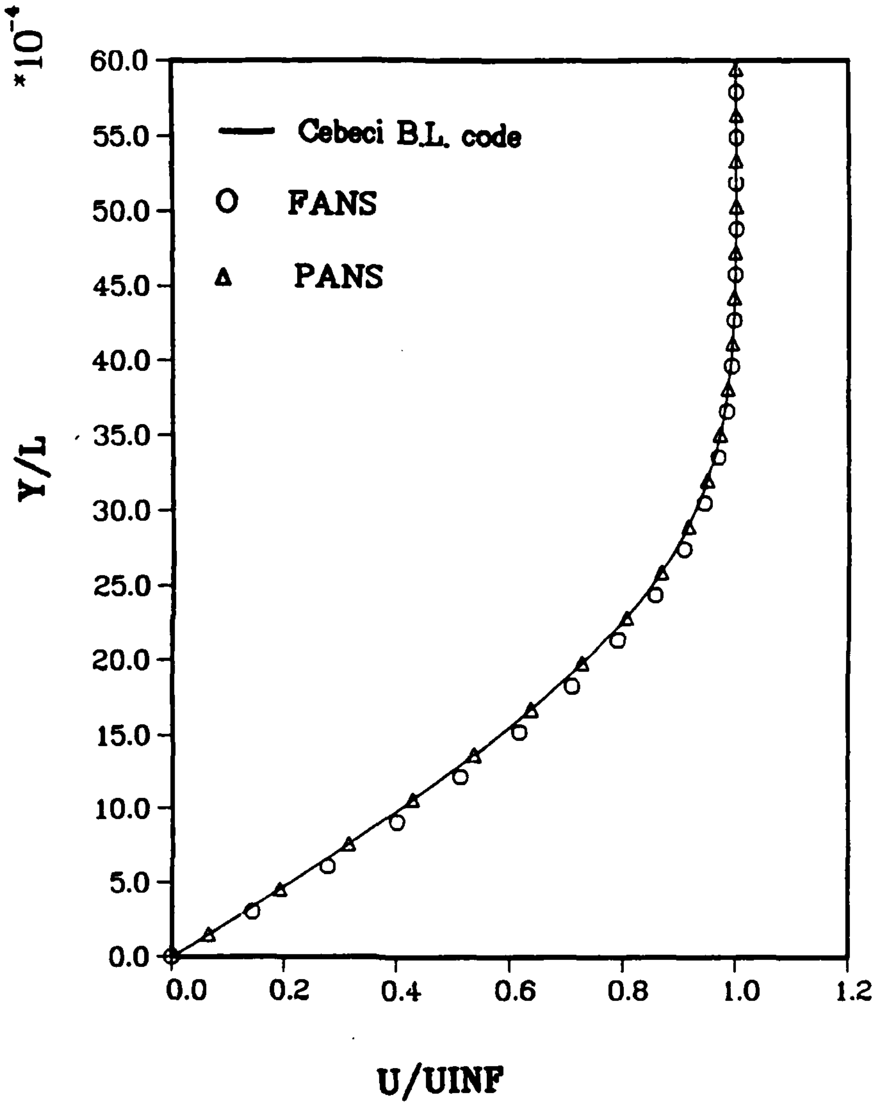


Figure 18 Boundary Layer: Velocity profile (one sweep)

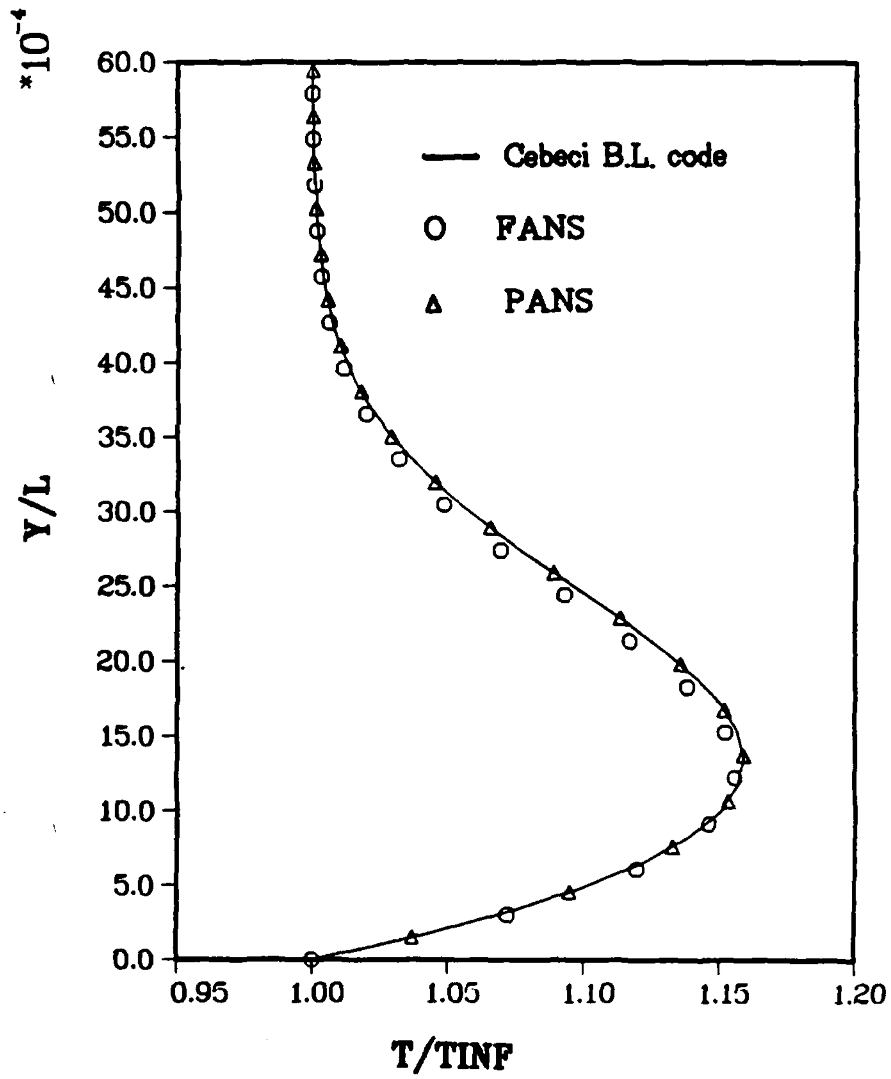


Figure 19 Boundary Layer: Temperature profile (one sweep)

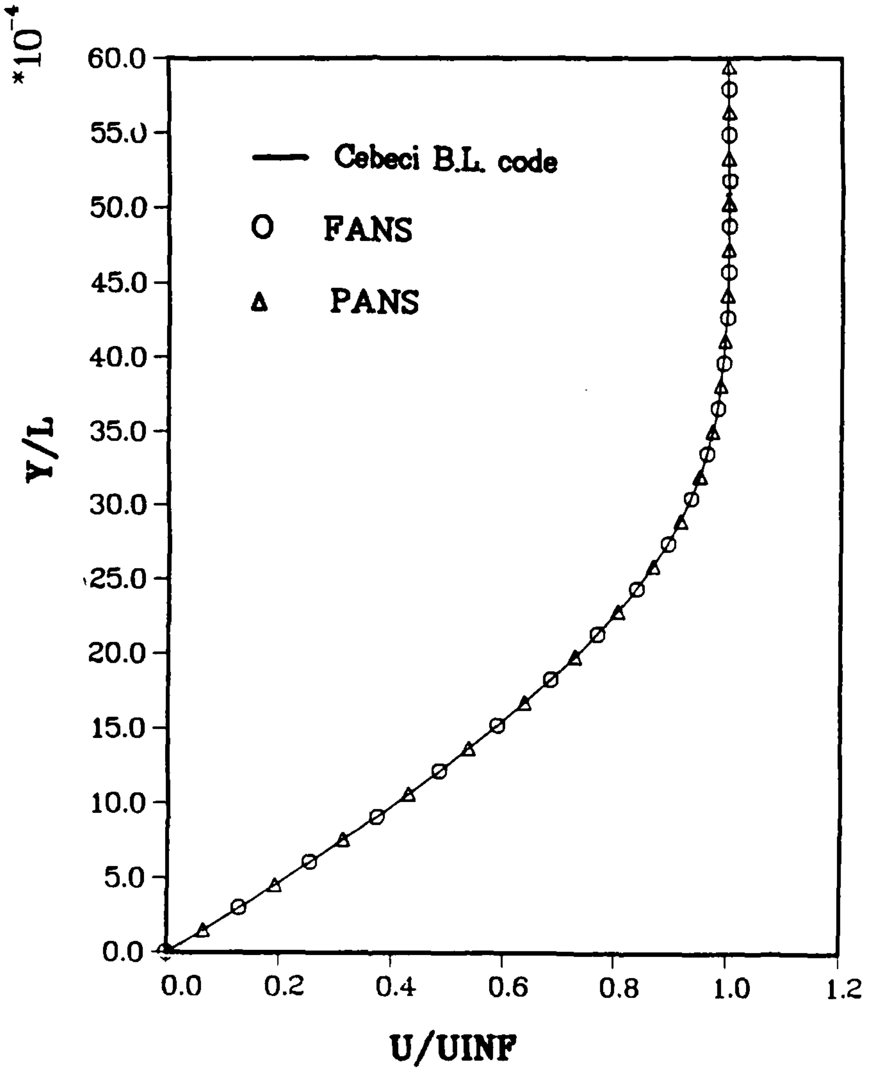


Figure 20 Boundary Layer: Velocity profile (four sweeps)

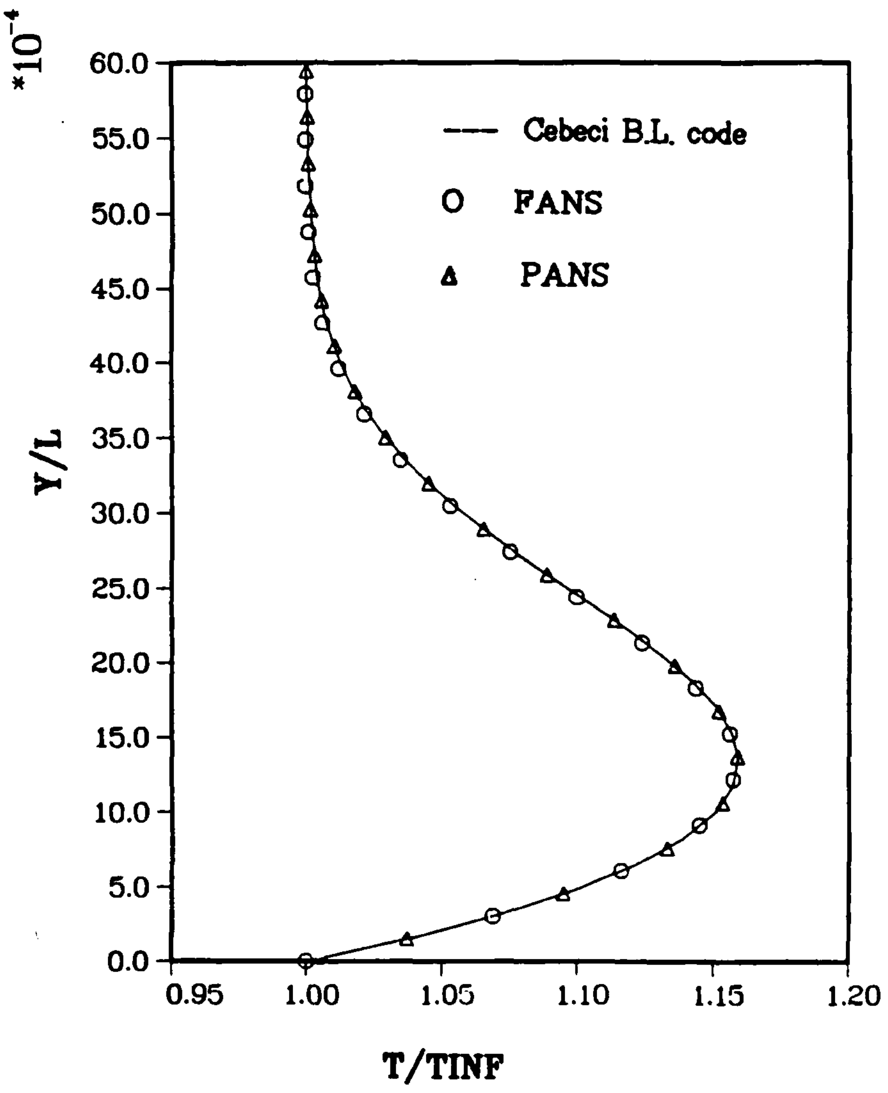


Figure 21 Boundary Layer: Temperature profile (four sweeps)

this was 4.6×10^{-4} and for the PANS code 4.7×10^{-4} .

Shock/Boundary Layer Interaction (FANS)

As another test the FANS code was applied to the shock/boundary layer interaction problem previously solved using the explicit code.

To keep the number of grid points to a minumum stretching in the normal direction was used. The stretching was determined using a geometric progression (5:8)

$$\Delta y_n^* = \Delta y^*_{min}\, r^{n-1} \qquad (68)$$

where r is called the common ratio. It was found that the algorithm could not handle a large amount of stretching. The maximum value of the common ratio was found to be approximately 1.05. By specifying the location of the outer boundary and the number of points in the normal direction, the minimum y step size could be calculated from

$$\Delta y^*_{min} = y_n^*(1-r)/(1-r^n) \qquad (69)$$

Using the same domain size used by Beam and Warming gives

$$\Delta y^*_{min} = 2.9097 \times 10^{-4}$$

With a grid using four hundred and eighty five points in the streamwise direction and one hundred and one points in the normal

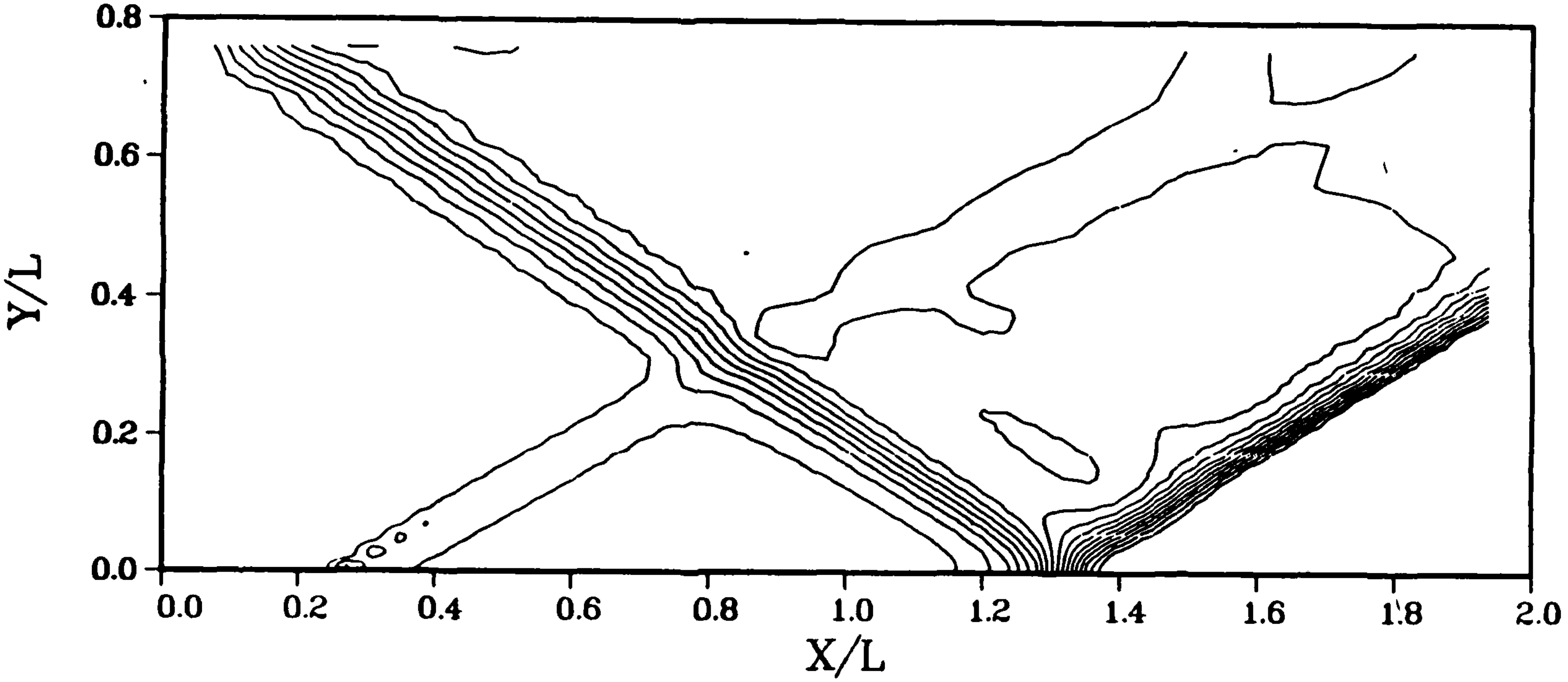


Figure 22 Shock/Boundary Layer: Pressure contours using the flux splitting code

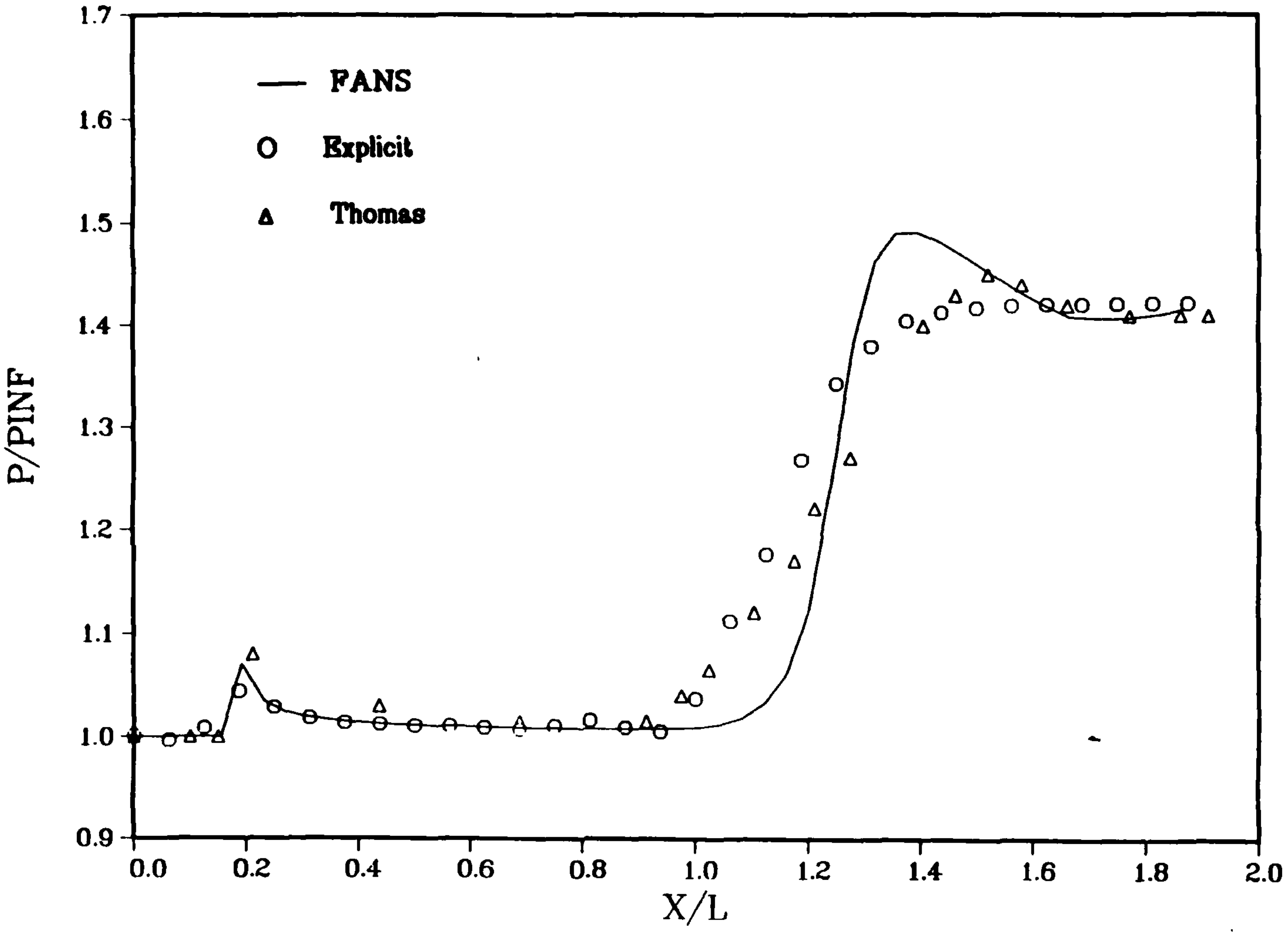


Figure 23 Shock/Boundary Layer: Wall pressure comparison

direction, solution accuracy would be of the same order as Beam and Warming's solution.

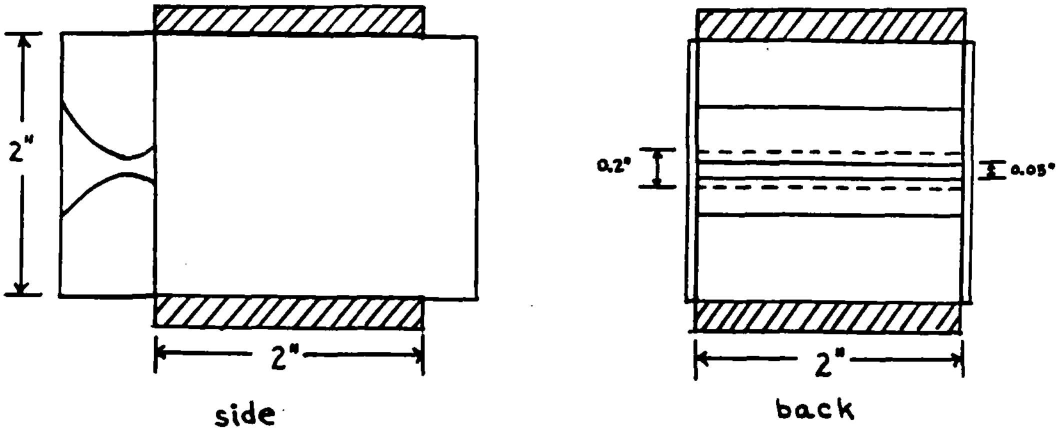
Figures 22 and 23 present the results of the computation. As can be seen in figure 22, the pressure contour plot, the code did an excellent job of capturing the leading edge shock, the impinging shock and the reflected shock.

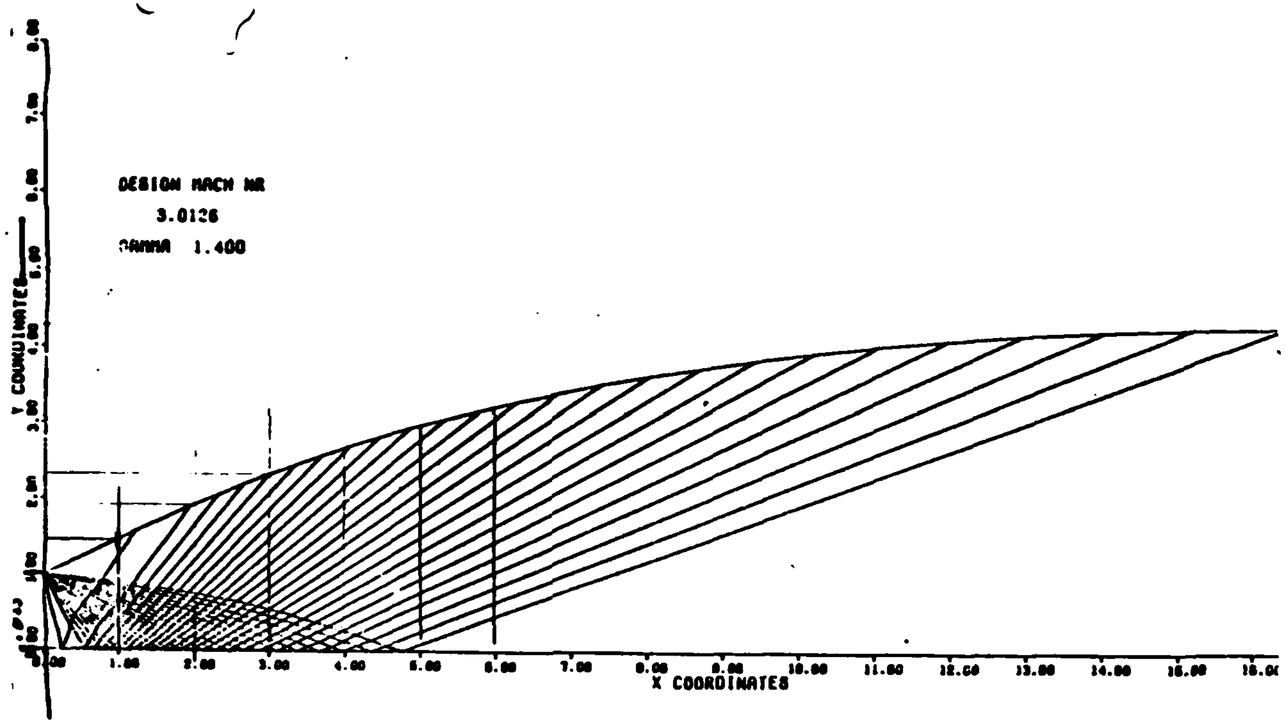
For comparison the results of the explicit code and Thomas' results have been plotted on figure 23. Here, there is quite a difference. The region over which the pressure rises from before the shock interaction to after the shock interaction is smaller in the FANS results. There is also a "hump" in the FANS pressure and a smaller hump in Thomas' data, but the explcit pressure curve levels out. Both of these differences are attributable to the damping in each code. The explicit code contains artificial damping terms and Thomas' code contains damping which results from the upwind differences, but the FANS code does not contain added damping. This means that the FANS code will capture the shocks with less smearing, but overshoots, as seen in the pressure hump, may occur.

The code took sixty eight seconds and three global sweeps to solve the problem on the Cray XMP. This gives an efficiency parameter of 4.6×10^{-4} . Compared to the results for the same problem for the explicit code, the FANS code is 14 times faster. The efficiency parameter, however, is about an order of magnitude larger which is to be expected since the FANS code was not vectorized.

Appendix A: Dimensions of the Shroud

The shroud used in this work is a model of nozzle block 3B used in the experiments conducted by Moran (11). The block consisted of a single converging-diverging, two-dimensional nozzle cut out of aluminum. The sides of the shroud were formed from plexiglass and the top and bottom wall of the shroud was made from removable wood blocks. The figure on this page shows the shroud dimensions and the drawing on the next page the original nozzle design.




DESIGN MACH NR
3.0126
GAMMA 1.400
Y COORDINATES
0.00
1.00
2.00
3.00
4.00
5.00
6.00
7.00
8.00
X COORDINATES
0.00
1.00
2.00
3.00
4.00
5.00
6.00
7.00
8.00
9.00
10.00
11.00
12.00
13.00
14.00
15.00
16.00

Appendix B: Jacobians

In this appendix the elements of the Jacobians defined by Eqs 55, 56, 59 and 60 are presented. These Jacobians are the partial derivative of a vector with respect to a vector and are therefore matrices. Each element will be denoted by the symbol given to the corresponding Jacobian. Subscripts on the symbols denote the row and column of the element.

For the [Q] matrix:

$Q_{s11} = 0$

$Q_{s12} = 1$

$Q_{s13} = 0$

$Q_{s14} = 0$

$Q_{s21} = (1/2)[(\gamma-3)u^{*2}+(\gamma-1)v^{*2}]$

$Q_{s22} = [2-(\gamma-1)]u^{*}$

$Q_{s23} = -(\gamma-1)v^{*}$

$Q_{s24} = (\gamma-1)$

$Q_{s31} = -u^{*}v^{*}$

$Q_{s32} = v^{*}$

$Q_{s33} = u^{*}$

$Q_{s34} = 0$

$$Q_{41} = -[(\gamma E_t^*/\rho) + (\gamma-1)(u^{*2}+v^{*2})]u^*$$

$$Q_{42} = (\gamma E_t^*/\rho) - (\gamma-1)(3u^{*2}+v^{*2})/2$$

$$Q_{43} = -(\gamma-1)u^* v^*$$

$$Q_{44} = \gamma u^*$$

For the $[Q]^+$ matrix the following are defined:

$$c_1 = (\partial a^*/\partial U_1^*) = [\gamma(\gamma-1)/2a^*]\,[-(E_t^*/\rho^{*2}) + (u^{*2}+v^{*2})/\rho^*]$$

$$c_2 = (\partial a^*/\partial U_2^*) = [\gamma(\gamma-1)/2a^*]\,[-u^*/\rho^*]$$

$$c_3 = (\partial a^*/\partial U_3^*) = [\gamma(\gamma-1)/2a^*]\,[-v^*/\rho^*]$$

$$c_4 = (\partial a^*/\partial U_4^*) = [\gamma(\gamma-1)/2a^*]\,[1/\rho^*]$$

$$B = [(\gamma-1)/\gamma]u^* + 2a^*/\gamma$$

$$E_1^+ = \rho^* a^*[(1/2)(M_x+1)]^2$$

$$D = \{[[(\gamma-1)u^*+2a^*]^2/2(\gamma^2-1)\} + v^{*2}/2$$

the elements are

$$Q_{11}^+ = (1/4)\,[(u^{*2}/a^*)-(\rho^* u^{*2}/a^{*2})c_1+a^*+\rho^* c_1]$$

$$Q_{12}^+ = (1/4)\,[(2u^*/a^*)-(\rho^* u^{*2}/a^{*2})c_2+2+\rho^* c_2]$$

$$Q_{13}^{+} = (1/4)\ [\rho^{*}-(\rho^{*}u^{*2}/a^{*2})]c_{2}$$

$$Q_{14}^{+} = (1/4)\ [\rho^{*}-(\rho^{*}u^{*2}/a^{*2})]c_{4}$$

$$Q_{21}^{+} = BQ_{11}^{+} + (E_{1}^{+}/\rho)\ [-(\gamma-1)u^{*}/\rho^{*} + 2c_{1}]$$

$$Q_{22}^{+} = BQ_{21}^{+} + (E_{1}^{+}/\gamma)\ [(\gamma-1)(1/\rho^{*}) + 2c_{2}]$$

$$Q_{23}^{+} = BQ_{13}^{+} + (2E_{1}^{+}/\gamma)c_{3}$$

$$Q_{24}^{+} = BQ_{14}^{+} + (2E_{1}^{+}/\gamma)c_{4}$$

$$Q_{31}^{+} = Q_{11}^{+}v^{*} - E_{1}^{+}(v^{*}/\rho^{*})$$

$$Q_{32}^{+} = Q_{12}^{+}v^{*}$$

$$Q_{33}^{+} = Q_{13}^{+}v^{*} + E_{1}^{+}(1/\rho^{*})$$

$$Q_{34}^{+} = Q_{14}^{+}v^{*}$$

$$Q_{41}^{+} = Q_{11}^{+}D + E_{1}^{+}\{[-(\gamma-1)^{2}(u^{*2}/\rho^{*2})+(4a^{*}+2(\gamma-1)u^{*})c_{1}-(2(\gamma-1)a^{*}u^{*}/\rho^{*})]/(\gamma^{2}-1) - v^{*2}/\rho^{*}\}$$

$$Q_{42}^{+} = Q_{12}^{+}D + E_{1}^{+}\{[(\gamma-1)^{2}u^{*}/\rho^{*} + 4a^{*}c_{2} + 2(\gamma-1)u^{*}c_{2} + 2(\gamma-1)a^{*}/\rho^{*}]/(\gamma^{2}-1)\}$$

$$Q_{43}^{+} = Q_{13}^{+}D + E_{1}^{+}\{[4a^{*}c_{3} + 2(\gamma-1)u^{*}c_{3}]/(\gamma^{2}-1) + v^{*}/\rho^{*}\}$$

$$Q_{44}^{+} = Q_{14}^{+}D + E_{1}^{+}\{[4a^{*}+2(\gamma-1)u^{*}]c_{4}/(\gamma^{2}-1)\}$$

For the elements of the [R] matrix

$R_{11} = 0$

$R_{12} = 0$

$R_{13} = 1$

$R_{14} = 0$

$R_{21} = -u^* v^*$

$R_{22} = v^*$

$R_{23} = u^*$

$R_{24} = 0$

$R_{31} = [(\gamma-1)u^{*2}+(\gamma-3)v^{*2}]/2$

$R_{32} = -(\gamma-1)u^*$

$R_{33} = (3-\gamma)v^*$

$R_{34} = \gamma-1$

$R_{41} = [-(\gamma E_t^*/\rho^*)+(\gamma-1)(u^{*2}+v^{*2})]v^*$

$R_{42} = -(\gamma-1)u^* v^*$

$R_{43} = (\gamma E_t^*/\rho^*) - (\gamma-1)(u^{*2}+3v^{*2})/2$

$R_{44} = \gamma v^*$

For the elements of the [W] matrix

$W_{11} = 0$

$W_{12} = 0$

$W_{13} = 0$

$W_{14} = 0$

$$W_{21} = -\mu/\mathrm{Re}\ [\partial_y(u^*/\rho^*)]$$

$$W_{22} = \mu/\mathrm{Re}\ [\partial_y(1/\rho^*)]$$

$$W_{23} = 0$$

$$W_{24} = 0$$

$$W_{31} = -(4/3)(\mu/\mathrm{Re})\ \partial_y(v^*/\rho^*)$$

$$W_{32} = 0$$

$$W_{33} = (4/3)(\mu/\mathrm{Re})\ \partial_y(1/\rho^*)$$

$$W_{34} = 0$$

$$W_{41} = -\mu/\mathrm{Re}\ [\partial_y(u^{*2}/\rho^*) - (4/3)\partial_y(v^{*2}/\rho^*) - (\gamma/\mathrm{Pr})\partial_y\{[p^*/(\gamma-1)\rho^{*2}] - [u^{*2}+v^{*2}/2\rho^*]\}]$$

$$W_{42} = (\mu/\mathrm{Re})(1-\gamma/\mathrm{Pr})\ \partial_y(u^*/\rho^*)$$

$$W_{43} = (\mu/\mathrm{Re})(4/3 - \gamma/\mathrm{Pr})\ \partial_y(v^*/\rho^*)$$

$$W_{44} = (\mu/\mathrm{Re})(\gamma/\mathrm{Pr})\ \partial_y(1/\rho^*)$$

Appendix C: Block Tri-diagonal Matrix Elements

When the derivative approximations of Eqs 63 and 64 are applied to the left-hand-side of Eq 62, a block tri-diagonal matrix results. A block tri-diagonal matrix is a matrix which has only three bands of non-zero block elements. These bands consist of the diagonal, sub-diagonal and supra-diagonal block elements. Each block element is a matrix which, in this case, are four row by four column matrices. Now, let [A] be a diagonal block element, [B] a sub-diagonal block element and [C] a supra-diagonal block element. The elements are then given by

$$[A]_j = [Q]_j^I + [(\Delta x^*/2Re(\Delta y^*)^2) \{(\mu_j^* + \mu_{j+1}^*) + (\mu_j^* + \mu_{j-1}^*)\}] [w]_j$$

$$[B]_j = -[(\Delta x^*/2\Delta y^*)[R]_{j-1} + (\Delta x^*/2Re(\Delta y^*)^2)(\mu_j^* + \mu_{j-1}^*)[w]_{j-1}]$$

$$[C]_j = (\Delta x^*/2\Delta y^*)[R]_{j+1} - (\Delta x^*/2Re(\Delta y^*)^2)(\mu_j^* + \mu_{j+1}^*)[w]_{j+1}$$

Bibliography

1. Anderson, W.K. and Thomas, J.L. "A Comparison of Finite Volume Flux Vector Splittings for the Euler Equations," AIAA 23rd Aerospace Sciences Meeting, January,1985. AIAA-85-0122.

2. Anderson, Dale A. and others. Computational Fluid Mechanics and Heat Transfer. New York: Hemisphere Publishing Corporation, 1984.

3. Bardina, Jorge and Lombard, C.K. "Three Dimensional CSCM Method for the Compressible Navier-Stokes Equations with Application to a Multi-Nozzle Exhaust Flowfield," AIAA/SAE/ASME/ASEE 21st Joint Propulsion Conference. AIAA-85-1193. AIAA, New York, 1985.

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

5. Beyer, W.H. CRC Standard Mathematical Tables, 27th Edition. Boca Ration: CRC Press Inc., 1984.

6. Cebeci, Tuncer and Bradshaw, Peter. Physical and Computational Aspects of Convective Heat Transfer. New York: Springer-Verlag Inc, 1984.

7. Goethert, Bernard H. Studies of the Flow Characteristics and Performance of Multi-Nozzle Rocket Exhausts. AEDC-TR-59-16. Arnold AFS, Tennessee: ARO Inc., October 1959. (AD 313-155)

8. Holmes, D.K. and Matz, R.J. An Investigation of the Multi-nozzle Rocket Shroud Concept. AEDC-TR-65-249. Arnold AFS, Tennessee: ARO Inc., February 1966.

9. James, M.L. Applied Numerical Methods for Digital Computation. New York: Harper & Row, Publishers, 1977.

10. Lawrence, S.L. and Tannehill, J.C. "An Upwind Algorithm for the Parabolized Navier-Stokes Equations," AIAA/ASME 4th Fluid Mechanics, Plasma Dynamics and Lasers Conference, May 1986, AIAA-86-1117.

11. Moran, J. Clustered Nozzles with Variable Shrouds. MS Thesis, Wright-Patterson AFB, Ohio: AFIT, Dec 1985.

12. Moretti, Gino. "Importance of Boundary Conditions in the Numerical Treatment of Hyperbolic Equations," The Physics of Fluids Supplement II,1969, pp II-13 to II-20.

13. Roache, P.J. and Mueller, T.J. "Numerical Solutions of Compressible and Incompressible Laminar Separated Flows," *AIAA Fluid and Plasma Dynamics Conference*, June 1969, AIAA-68-741.

14. Roberts, T.K. *A Numerical Solution of a Nonisothermal Wall using the Two Dimensional Navier-Stokes Equations*. MS Thesis, Wright-Patterson AFB, Ohio: AFIT, March 1985.

15. Rubin, S.G. and Lin, T.C. "Numerical Methods for Two- and Three-Dimensional Viscous Flow Problems: Application to Hypersonic Leading Edge Equations," PIBAL Rept No. 71-8, Polytechnic Institute of Brooklyn, Farmingdale N.Y. April, 1971.

16. Schiff, L.B. and Steger, J.L. "Numerical Solution of Steady Supersonic Viscous Flow," *17th Aerospace Sciences Meeting*, January, 1979. AIAA-79-0130.

17. Shapiro, Ascher H. *The Dynamiacs and Thermodynamics of Compressible Fluid Flow, Volume I*. New York: John Wiley & Sons, 1953.

18. Thomas, James L. and Walters, Robert W. "Upwind Relaxation Algorithms for the Navier Stokes Equations," AIAA-85-1501.

19. Vigneron, Y.C. and others. "Calculation of Supersonic Viscous Flow over Delta Wings With Sharp Subsonic Leading Edges," *AIAA 11th Fluid and Plasma Dynamics Conference*, July 1978, AIAA-78-1137.

20. Walters, R.W. and Dwoyer, D.L. "An Efficient Iteration Strategy For The Solution Of The Euler Equations," AIAA-85-1529.

VITA

Captain Gary W. Huband was born on 3 November 1959 in Richmond, Virginia. He graduated from high school in Chester, Virginia in 1978 and attended the University of Virginia from which he received the degree of Bachelor of Science in Aerospace Engineering in May 1982. Upon graduation, he received a commission in the USAF through the ROTC program. After graduating he was assigned to the 6585th Test Group at Holloman AFB, New Mexico. There he participated in the flight testing of inertial navigation and weapon systems and also received his non-rated officer aircrew wings. In June 1985 he entered the School of Engineering, Air Force Institute of Technology.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GA/~~ENG~~/86D-8

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL *(If applicable)*
AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Air Force Institute of Technology
Wright-Patterson AFB, Ohio 45433

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11. TITLE *(Include Security Classification)*
See box 19

12. PERSONAL AUTHOR(S) Huband, Gary Wayne, B.S., Capt, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ______

14 DATE OF REPORT *(Year, Month, Day)*
1986 December

15 PAGE COUNT
85

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
20	04	

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Supersonic flows, Parabolized Navier-Stokes equations
Shrouded rocket nozzles

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Title: NUMERICAL STUDY OF SUPERSONIC FLOWS USING DIFFERENT TECHNIQUES

Thesis Chairman: Ahmad Halim
Associate Professor of Aerospace Engineering

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT □ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Ahmad Halim

22b TELEPHONE *(Include Area Code)*
513-255-2040

22c OFFICE SYMBOL
AFIT/ENY

Previous editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

Block 19

The purpose of this work was to obtain the solution to three supersonic flow problems using three different numerical techniques.

First, a shock/boundary layer problem is solved using MacCormack's explicit technique. Then, using the same technique a shrouded rocket nozzle problem is solved. These two problems showed that the explicit scheme required many minutes of computer time to solve.

In order to explore more efficient codes to solve the shroud problem, space marching algorithms were studied. A space marching algorithm using flux-splitting in the streamwise direction was applied to an approximate form of the Navier-Stokes equation. Flux-splitting combined with a global iteration approach should allow the shroud problem to be solved with a space marching algorithm. The flux-splitting code was applied to two supersonic flow problems and very good results were obtained.