NASA Technical Memorandum 83123

Two-boundary Grid Generation for the Solution of the Three-dimensional Compressible Navier-Stokes Equations

R. E. Smith

May 1981

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

TABLE OF CONTENTS

N91-22951 #

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

B	parameter governing grid concentration for spike-nosed body grids
C_1, C_2	intermediate variables used in computing the pressure on a cylinder surface for wedge-cylinder corner meshes
C_V	specific heat at constant volume
C_P	specific heat at constant pressure
c_∞	free stream speed of sound
D_1, D_2, D_3	intermediate variables used in computing pressure on a wedge surface for wedge-cylinder corner meshes
e	internal energy
F, G, H	vector fluxes for coordinate directions
$\bar{F}$	symbol for flux vectors in a compact definition of the equations of motion
f_1, f_2, f_3, f_4	basis functions for cubic connecting function
g'', h''	second derivatives for tension spline approximation
$\vec{i}, \vec{j}, \vec{k}$	unit vectors in the physical coordinate system
J	Jacobian matrix
J^{-1}	inverse Jacobian matrix
$\lvert J^{-1} \rvert$	determinant of inverse Jacobian matrix
K	magnitude of normal vector on bounding surfaces
$\bar{K}$	coefficient of heat conduction
k_1, k_2	parameters governing the grid concentration for wedge-cylinder grids
$\bar{k}, \bar{k}_1$	parameters governing the grid concentration for spike-nosed body grids
$\hat{k}$	parameter governing the grid concentration for planar intersecting corner grids

$\tilde{k}$	parameter governing grid concentration for planar intersecting corner grids
L	characteristic length
M	number of points describing the inside boundary for spike-nosed body grids
M_∞	free stream Mach number
m,n	number of points describing boundaries for tension spline approximation
N	number of points in tension spline approximation
$\overline{N}$	normal direction
P	pressure
$\bar{\dot{q}}$	heat conduction vector
q_x,q_y,q_z	components of heat conduct vector
r	radial direction
r_o,r_1,r_2	cylinder radii for wedge-cylinder grid description
R	radius of circle describing the outside boundary for spike-nosed body meshes
$R_{e_\infty}=Re$	free stream Reynolds number
s	parametric variable for an airfoil grid
$\overline{s},\overline{t}$	parametric variables
s_o	Sutherland viscosity law constant
S	four-dimensional array containing state variables and transformation data
T	temperature
T_r	reference temperature for Sutherland viscosity law
t	time

$\hat{t}$	parametric variable for inner boundary of spike-nosed body meshes
U	vector of state variables
u,v,w	velocity components in the physical domain
$\overline{u}$	velocity vector
$\overline{v}$	velocity used in computing time step for the finite difference technique
$X(),Y(),Z(),$ $x(),y(),z()$	functions relating the computational domain to the physical domain
$\underline{X}(),\underline{Y}(),\underline{Z}()$	functions relating the computational domain to the physical domain with boundary parameterization and third independent variable "connecting function"
x,y,z	coordinates for the physical domain
$\overline{x},\overline{y}$	symbols for coordinates in the compact definition of the equations of motion
$\hat{x},\hat{y}$	coordinates for the inside boundary of spike-nosed body grids
x_o	physical coordinate position in windward direction for the initial solid surface in wedge-cylinder corner grids
x_L	physical coordinate in the windward direction for the final grid plane in the wedge-cylinder corner grids
$\overline{\alpha}$	coefficient for pressure damping
γ	ratio of specific heats
$\gamma_x,\gamma_y,\gamma_z$	directional cosines of the normal vector at a solid wall
$\Delta\xi,\Delta\eta,\Delta\zeta$	constant increments in the computational coordinates
Δt	increment in time
θ	angle defining a parametric variable for the outside boundary of spike-nosed grids
$\theta_1,\theta_2,\theta_3$	angles defining boundaries of wedge cylinder corner grids

θ_y, θ_z	angles defining boundaries of planar intersecting corner grids
$\hat{\theta}$	intermediate angle for computing pressure boundary condition for wedge-cylinder corner grids
μ	molecular viscosity
μ_r	reference viscosity in Sutherland viscosity law
μ_β	bulk viscosity
ξ, η, ζ	coordinates in the computational domain
$\bar{\xi}, \bar{\eta}, \bar{\zeta}$	redistributed coordinates relative to the computational domain
ρ	density
σ	tension parameter
τ	stress tensor
$\left.\begin{matrix}\tau_{xx}, \tau_{xy}, \tau_{xz}, \\ \tau_{yy}, \tau_{yz}, \tau_{zz}\end{matrix}\right\}$	elements of the stress tensor
ϕ	wedge angle for wedge-cylinder corner grids
$\bar{\phi}$	angle of rotation for three-dimensional spike-nosed grids
ϕ_x, ϕ_y, ϕ_z	components of viscous dissipation function
ψ_1, ψ_2	intermediate angles for wedge-cylinder grids

Subscripts:

B	boundary value
W	solid wall value
∞	free stream value

Superscripts:

I	inside boundary for spike-nosed body grids
o	outside boundary for spike-nosed body grids

Indices:

I,J,K,L	indices used in four-dimensional S array
i,j,k	point indices
ℓ	boundary indicator

Operators:

∇	gradient operator
$\circ$	inner product operator
$\left.\begin{matrix} L_\xi \ L_\eta \ L_\zeta \\ L_x, L_y, L_z \end{matrix}\right\}$	finite difference operators
Lin	linear interpolation operator
∂	partial differentiation

SUMMARY

TWO-BOUNDARY GRID GENERATION FOR THE SOLUTION OF THE THREE-DIMENSIONAL COMPRESSIBLE NAVIER-STOKES EQUATIONS

Robert Edward Smith

A grid generation technique called the "two-boundary technique" is developed and applied for the solution of the three-dimensional compressible Navier-Stokes equations describing laminar flow. The Navier-Stokes equations are presented relative to a xyz cartesian coordinate system and are transformed to a $\xi\eta\zeta$ computational coordinate system. The grid generation technique provides the Jacobian matrix describing the transformation.

The "two-boundary technique" is based on algebraically defining two distinct boundaries of a flow domain and joining these boundaries with a "connecting function" which is proposed to be linear or cubic polynomials. The algebraic boundary representation can be analytical functions or numerical interpolation functions. Control of the distribution of the grid in the physical domain is achieved by embedding "control functions" which redistribute the uniform grid of the computational domain and concentrate or disperse the grid in the physical domain. The computer program to solve the Navier-Stokes equations is based on a

MacCormack time-split technique and is specifically designed for the vector architecture and virtual memory of the CYBER 203 computer. The program "Navier-Stokes solver" is written in the SL/1 language which allows 32-bit word arithmetic operations and storage. The program can run with 5 × 10^4 grid points using only primary memory, and the computational speed is 4 × 10^{-5} seconds per grid point per time step.

Using the "two-boundary technique," grids are developed for two distinctly different flow field problems, and compressible supersonic laminar flow solutions are obtained using the Navier-Stokes solver. Grids and solutions are obtained for a family of three-dimensional corners at Mach number 3.64 and Reynolds numbers 2.92 × 10^5/m and 3.9 × 10^6/m. Also, grids are derived for spike-nosed bodies, and solutions are obtained at Mach number 3 and Reynolds number 7.87 × 10^6/m.

Coupled with the Navier-Stokes solver, the "two-boundary technique" is demonstrated to be viable for grid generation associated with computing supersonic laminar flow. The technique is easy to apply and is applicable to a wide class of geometries. The "two-boundary technique" can serve as the foundation for generating grids with highly complex boundaries and yield grid point distributions that can capture rapidly changing variables in a flow field.

1. INTRODUCTION

In recent years, the availability of large scale scientific computer systems has resulted in rapid progress in the field of Computational Fluid Dynamics. There is now the capability to calculate many complex unsteady two-dimensional and steady three-dimensional flows. MacCormack and Lomax [1]* summarize the "state of the art" for the computation of compressible viscous fluid flow. For a heat conducting compressible fluid acting near body surfaces with large separation regions or inviscid-viscid interactions, the numerical solution of the Navier-Stokes equations is the preferred approach [1]. An emerging problem, however, is the generation of grid systems on which solutions can be obtained when there are complex boundary geometries. This problem is compounded in three dimensions. This study addresses the solution of the three-dimensional compressible Navier-Stokes equations, the generation of grids, and the solution algorithm-computer relationship. The emphasis is placed on grid generation.

An algebraic grid generation technique applicable to the Navier-Stokes equations is developed, and a three-dimensional Navier-Stokes solver (compressible laminar flow) based on a proven numerical technique (MacCormack time-split algorithm [1-4]) is developed for the CDC CYBER 203 vector computer [5]. Also, flow visualization techniques have been developed in conjunction with this research but will not be discussed in detail. In order to evaluate the overall system for computing viscous compressible flow, and in particular the grid generation

*The numbers in brackets indicate references.

technique, grids are determined for a family of three-dimensional corners and two spike-nosed bodies.

The grid generation technique is called the "two-boundary technique." It is applicable in two and three dimensions and is a methodology for direct computation of the physical grid as a function of a uniform rectangular computational grid. The Jacobian matrix of the transformation can be obtained by direct analytic differentiation. This is in contrast to the indirect approach where an elliptic partial differential equation system is solved for the coordinates of the physical grid relative to the computational grid, and in which the Jacobian matrix must be obtained by numerical differentiation. The indirect approach is popularly known as the "TTM method" [1, 6-10]. In the "two-boundary technique," two separate non-intersecting boundaries are defined by means of algebraic functions or numerical interpolation functions. These functions have as independent variables, coordinates which are normalized to unity. Another function with an independent variable defined on the unit interval connects the boundaries.

The "two-boundary technique" is based upon concepts found in the theory of surface definition [11,12]. Gordon and Hall [13] postulate the essentials of the technique and emphasize finite element grids. Also, Eiseman [14-16] uses a form of the technique in generating grids for multiconnected two-dimensional domains. In this investigation the "two-boundary technique" is developed and is analyzed for finite difference solutions for fluid flow applications. Low order polynomials (linear and cubic) are used for connecting functions. For the cubic connecting function, orthogonality can be enforced at the boundaries through knowledge of the normal derivatives there. Control of the grid (grid spacing in the physical domain) is achieved by the superposition onto the independent variables algebraic or transcendental functions with desirable characteristics. Splines under tension [17-19] are proposed for approximate boundary definition. The "two-boundary technique" is used to algebraically generate grids for a family of three-dimensional corners and to generate a combined algebraic-numeric grid for spike-nosed bodies. The derivatives composing the Jacobian matrix for the three-dimensional corners and spike-nosed bodies are presented for obtaining numerical solutions of the Navier Stokes equations.

The CDC CYBER 203 is a large scale computer with vector processing architecture and virtual memory. Generally efficiency using a vector computer increases with increasing vector length, however, considerable attention must be given to the algorithm-machine architecture relation and balancing the vector length with practical limits of primary memory. A MacCormack time-split solution algorithm is programmed for the CYBER 203 computer and is called the "Navier-Stokes solver." The MacCormack technique is used because of its robustness and adaptability to vector processing. Another primary consideration when developing a "Navier-Stokes solver" on a large complex computer is the capability to solve a wide class of problems with a minimum of programming changes. This has been accomplished by programming the complete transformed equations of motion and storing all nine elements of the Jacobian matrix of the transformation at each grid point (transformation data).

Supplying the transformation data from a grid generation technique and programming the boundary conditions for a given problem (separate subroutine) allows the program be applied to virtually any laminar fluid flow problem. Since the split MacCormack technique is used, two-dimensional solutions can be obtained without unnecessary computations. The operator for the third dimension is bypassed. A final important point relative to the Navier-Stokes solver is that the MacCormack technique is written in the SL/1 language [20] and uses the 32-bit arithmetic option of the CYBER 203. By using 32-bit words, twice the in-core storage is available and approximately twice the computational speed is achieved compared to the use of normal 64-bit words. There are approximately two million words of primary memory and the computational speed is 4×10^{-5} seconds per grid point per time step for the 32-bit word length. For the explicit technique, no significant degeneration in accuracy is observed using the smaller word size. The Navier-Stokes solver is independent of the grid generation technique, and the transformation data from any technique can be used by the code.

Using the "two-boundary technique" grids are developed for two distinctly different flow field problems, and compressible supersonic laminar flow solutions are obtained using the computer program based on the MacCormack technique. A set of algebraic grid generation equations are developed using the "two-boundary technique" for a family of three-dimensional corners consisting of wedge-cylinder, plate-cylinder, approximate wedge-plate, and approximate rectangular corners. It is also shown that exact grids for planar intersecting corners can be derived with the "two-boundary technique." Corner flow solutions are obtained on a 20 × 36 × 36 grid and a 12 × 64 × 64 grid. The solutions obtained on the 12 × 64 × 64 grid are compared with physical experiments and other numerical experiments. The Mach number used is 3.64 and the Reynolds number is 2.92×10^5/m and 3.9×10^6/m.

Also, algebraic grids are derived using the "two-boundary technique" for spike-nosed bodies. In particular, grids for a one-half inch spike-nosed body and a one and one-half inch spike-nosed body are obtained. Supersonic flow solutions at Mach number 3 and Reynolds number 7.87×10^6/m are obtained about these configurations. Unlike the flows about the three-dimensional corners, the flow about the spike-nosed bodies is unsteady. The amplitude of the oscillations about the one-half inch nose body is quite small, however, the one and one-half inch spike-nosed body flow field oscillates with a large amplitude. The high amplitude solutions are compared with physical experiments. The flow fields are two-dimensional axisymmetric, but are solved with a three-dimensional Navier-Stokes solver resulting in considerable savings of development time for a specialized axisymmetric code.

For flow visualization, a relatively novel approach has been developed where a color spectrum is used to display a scalar variable such as density, Mach number, etc., on a two-dimensional slice of a flow field. Sequences of pictures can show the history of a developing flow or a scan of the flow field in a three-dimensional domain. The Diccomed Digital Display/Film Writer system which is normally used for environmental image processing is used for the flow visualization.

In summary, the main objectives of this study are the development of an algebraic grid generation procedure, the development of software to solve the compressible three-dimensional Navier-Stokes equations on a vector computer using the results of the grid generation technique, and the application of the grid generation technique and software to solve specific supersonic flow problems. The organization is as follows. In Chapter 2 the three dimensional compressible Navier-Stokes equations are presented relative to a Cartesian coordinate system and are transformed to a uniform grid computational coordinate system. This introduces the information that must be determined by the grid generation technique. The "two-boundary technique" is developed and applied to generate grids and Jacobian derivatives for a family of three-dimensional corners, spike-nosed bodies, and an airfoil configuration. In Chapter 3, the MacCormack technique is presented, and its compatibility with the CYBER 203 is described. In Chapter 4, supersonic flow solutions about three-dimensional corners and spike-nosed bodies obtained with the "two-boundary technique" and Navier-Stokes solver are described.

2. ANALYSIS

This chapter develops the equations of motion and the "two-boundary technique" for grid generation. Grids and boundary conditions are developed for a family of three-dimensional corners and for spike-nosed bodies. Also, grids are developed for airfoil boundaries using splines under tension.

2.1 Navier-Stokes Equations of Motion

The governing equations which describe the motion of a viscous compressible heat conducting fluid are the continuity equation, momentum equations, and energy equation. These equations are derived from the concept of continuum mechanics. The continuum concept and derivation of the Navier-Stokes equations of motion are found in several references, of which Schlichting [21] is the most notable.

Expressed in symboic form the Navier-Stokes equations of motion are:

$$\text{Continuity:} \qquad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{u}) = 0, \qquad (2.1a)$$

$$\text{Momentum:} \qquad \frac{\partial (\rho \bar{u})}{\partial t} + \nabla \cdot (\rho \bar{u}\bar{u} - \tau) = 0, \qquad (2.1b)$$

$$\text{Energy:} \qquad \frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho e \bar{u} + \bar{\dot{q}} - \bar{u} \cdot \tau) = 0. \qquad (2.1c)$$

The stress tensor, dissipation function, and heat conduction for a rectangular cartesian coordinate system are:

$$\tau = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \tau_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \end{bmatrix} \equiv \text{stress tensor}$$

where

$$\tau_{xx} = -p + 2\mu\frac{\partial u}{\partial x} + (\mu_\beta - \frac{2}{3}\mu)\ (\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}),$$

$$\tau_{yy} = -P + 2\mu\frac{\partial v}{\partial y} + (\mu_\beta - \frac{2}{3}\mu)\ (\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z})$$

$$\tau_{zz} = -P + 2\mu\frac{\partial w}{\partial z} + (\mu_\beta - \frac{2}{3}\mu)\ (\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}),$$

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

$$\tau_{xz} = \mu(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}),$$

$$\tau_{yz} = \mu(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}),$$

$$\bar{u} \cdot \tau = \left\{\begin{array}{l} \phi_x = u\tau_{xx} + v\tau_{xy} + w\tau_{xz} \\ \phi_y = u\tau_{xy} + v\tau_{yy} + w\tau_{yz} \\ \phi_z = u\tau_{xz} + v\tau_{yz} + w\tau_{zz} \end{array}\right\} \equiv \text{dissipation function},$$

and

$$\bar{\dot{q}} = \left\{\begin{array}{l} q_x = -\bar{K}\frac{\partial T}{\partial x} \\ q_y = -\bar{K}\frac{\partial T}{\partial y} \\ q_z = -\bar{K}\frac{\partial T}{\partial z} \end{array}\right\} \equiv \text{Heat flux vector},$$

The viscosity coefficient μ is a function of temperature and is adequately approximated by Sutherland's semiempirical equation:

$$\frac{\mu}{\mu_r} = \frac{T_r + S_o}{T + S_o} \left(\frac{T}{T_r}\right)^{\frac{3}{2}}$$

with

$$S_o = 198.6°\text{R for air.}$$

The bulk viscosity coefficient μ_β is set equal to zero. This is a reasonable assumption for a monatomic gas where the molecules has no internal degree of freedom. For a polyatomic gas the bulk viscosity is not always zero and can be the same order of magnitude as the molecular viscosity in sound propagation and shock structure. A detailed discussion of bulk viscosity is given by Vincenti and Kruger [22].

At this point there are five coupled partial differential equations and one algebraic equation with eight unknowns: ρ, u, v, w, P, e, T, and μ. In order to have a complete system, there must be two additional equations relating the unknowns. The equation of state is

$$P = P\ (\rho, T),$$

and for a perfect gas $P = \rho RT$ and $e = C_V T$ where C_V is the specific heat at constant volumn, and R is the gas constant. For compatible boundary conditions this system of equations is solvable.

2.2 Transformed Equations of Motion

The equations of motion in Section 2.1 are expressed in terms of a Cartesian coordinate system. If an object is defined in this coordinate system and a flow is to take place about the object, it is desirable to perform the computation in a coordinate system which conforms to the boundaries of the object. There are two primary reasons for wanting the coordinate system to be boundary-fitted. Boundary-fitted coordinates afford the ability to apply boundary conditions exactly avoiding interpolation error, and they minimize the logic that is necessary to apply boundary conditions. The penalty for these advantages is added complexity of the equations of motion. Another consideration is that when the domain of the flow field is discretized, it is desirable to have grid points concentrated in certain regions where high rates of change are likely to occur. For instance, in the boundary layer region more grid points are necessary to resolve the rapid change in the state variables. If the cartesian coordinate system where the object is defined is called the physical domain, the coordinate system relative to the boundaries of the object is called the computational domain. The relationship between the physical domain and the computational domain is a unique single-valued transformation with continuous derivatives such that if the coordinates in the computational domain are ξ, η, ζ: then

$$\xi = \xi(x,y,z), \quad \eta = \eta(x,y,z), \quad \text{and} \quad \zeta = \zeta(x,y,z).$$

Conversely,

$$x = x(\xi,\eta,\zeta), \quad y = y(\xi,\eta,\zeta), \quad \text{and} \quad z = z(\xi,\eta,\zeta)$$

where x, y, and z are coordinates in the physical domain. Since the equations of motion in terms of the Cartesian coordinate system of the physical domain are advantageously solved in terms of the coordinate system of the computational domain, the equations must be transformed. This is accomplished by expressing the derivatives of the state variables with respect to the xyz components of the physical domain in terms of the $\xi\eta\zeta$ components of the computational domain as follows:

$$\begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial \xi} & \frac{\partial u}{\partial \eta} & \frac{\partial u}{\partial \zeta} \\ \frac{\partial v}{\partial \xi} & \frac{\partial v}{\partial \eta} & \frac{\partial v}{\partial \zeta} \\ \frac{\partial w}{\partial \xi} & \frac{\partial w}{\partial \eta} & \frac{\partial w}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} & \frac{\partial \xi}{\partial z} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} & \frac{\partial \eta}{\partial z} \\ \frac{\partial \zeta}{\partial x} & \frac{\partial \zeta}{\partial y} & \frac{\partial \zeta}{\partial z} \end{bmatrix} .$$

Notice that u, v, and w are the velocities along the x, y, and z axes in the physical domain.

In terms of the transformed variables the stress tensor becomes:

$$\tau_{xx} = -P + 2\mu\left(\frac{\partial\xi}{\partial x}\ \frac{\partial u}{\partial\xi} + \frac{\partial\eta}{\partial x}\ \frac{\partial u}{\partial\eta} + \frac{\partial\zeta}{\partial x}\ \frac{\partial u}{\partial\zeta}\right)$$

$$- 2/3\ \mu\ \left(\frac{\partial\xi}{\partial x}\ \frac{\partial u}{\partial\xi} + \frac{\partial\eta}{\partial x}\ \frac{\partial u}{\partial\eta} + \frac{\partial\zeta}{\partial x}\ \frac{\partial u}{\partial\zeta} + \frac{\partial\xi}{\partial y}\ \frac{\partial v}{\partial\xi} + \frac{\partial\eta}{\partial y}\ \frac{\partial v}{\partial\eta} + \frac{\partial\zeta}{\partial y}\ \frac{\partial v}{\partial\zeta}\right.$$

$$\left. + \frac{\partial\xi}{\partial z}\ \frac{\partial w}{\partial z} + \frac{\partial\eta}{\partial z}\ \frac{\partial w}{\partial\eta} + \frac{\partial\zeta}{\partial z}\ \frac{\partial w}{\partial\zeta}\right)\ ,$$

$$\tau_{yy} = -P + 2\mu\left(\frac{\partial\xi}{\partial y}\ \frac{\partial v}{\partial\xi} + \frac{\partial\eta}{\partial y}\ \frac{\partial v}{\partial\eta} + \frac{\partial\zeta}{\partial y}\ \frac{\partial v}{\partial\zeta}\right)$$

$$- 2/3\ \mu\ \left(\frac{\partial\xi}{\partial x}\ \frac{\partial u}{\partial\xi} + \frac{\partial\eta}{\partial x}\ \frac{\partial u}{\partial\eta} + \frac{\partial\zeta}{\partial x}\ \frac{\partial u}{\partial\zeta} + \frac{\partial\xi}{\partial y}\ \frac{\partial v}{\partial\xi} + \frac{\partial\eta}{\partial y}\ \frac{\partial v}{\partial\eta} + \frac{\partial\zeta}{\partial y}\ \frac{\partial v}{\partial\zeta}\right.$$

$$\left. + \frac{\partial\xi}{\partial z}\ \frac{\partial w}{\partial\xi} + \frac{\partial\eta}{\partial z}\ \frac{\partial w}{\partial\eta} + \frac{\partial\zeta}{\partial z}\ \frac{\partial w}{\partial\zeta}\right),$$

$$\tau_{zz} = -P + 2\mu\left(\frac{\partial\xi}{\partial z}\ \frac{\partial w}{\partial\xi} + \frac{\partial\eta}{\partial z}\ \frac{\partial w}{\partial\eta} + \frac{\partial\zeta}{\partial z}\ \frac{\partial w}{\partial\zeta}\right)$$

$$- 2/3\ \mu\ \left(\frac{\partial\xi}{\partial x}\ \frac{\partial u}{\partial\xi} + \frac{\partial\eta}{\partial x}\ \frac{\partial u}{\partial\eta} + \frac{\partial\zeta}{\partial x}\ \frac{\partial u}{\partial\zeta} + \frac{\partial\xi}{\partial y}\ \frac{\partial v}{\partial\xi} + \frac{\partial\eta}{\partial y}\ \frac{\partial v}{\partial\eta} + \frac{\partial\zeta}{\partial y}\ \frac{\partial v}{\partial\zeta}\right.$$

$$\left. + \frac{\partial\xi}{\partial z}\ \frac{\partial w}{\partial\xi} + \frac{\partial\eta}{\partial z}\ \frac{\partial w}{\partial\eta} + \frac{\partial\zeta}{\partial z}\ \frac{\partial w}{\partial\zeta}\right),$$

$$\tau_{xy} = \mu\left(\frac{\partial\xi}{\partial y}\ \frac{\partial u}{\partial\xi} + \frac{\partial\eta}{\partial y}\ \frac{\partial u}{\partial\eta} + \frac{\partial\zeta}{\partial y}\ \frac{\partial u}{\partial\zeta} + \frac{\partial\xi}{\partial x}\ \frac{\partial v}{\partial\xi} + \frac{\partial\eta}{\partial x}\ \frac{\partial v}{\partial\eta} + \frac{\partial\zeta}{\partial x}\ \frac{\partial v}{\partial\zeta}\right),$$

$$\tau_{xz} = \mu \left(\frac{\partial \xi}{\partial x} \frac{\partial w}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial w}{\partial \eta} + \frac{\partial \zeta}{\partial x} \frac{\partial w}{\partial \zeta} + \frac{\partial \xi}{\partial z} \frac{\partial u}{\partial \xi} + \frac{\partial \eta}{\partial z} \frac{\partial u}{\partial \eta} + \frac{\partial \zeta}{\partial z} \frac{\partial u}{\partial \zeta} \right),$$

and

$$\tau_{yz} = \mu \left(\frac{\partial \xi}{\partial z} \frac{\partial v}{\partial \xi} + \frac{\partial \eta}{\partial z} \frac{\partial v}{\partial \eta} + \frac{\partial \zeta}{\partial z} \frac{\partial v}{\partial \zeta} + \frac{\partial \xi}{\partial y} \frac{\partial w}{\partial \xi} + \frac{\partial \eta}{\partial y} \frac{\partial w}{\partial \eta} + \frac{\partial \zeta}{\partial y} \frac{\partial w}{\partial \zeta} \right).$$

The heat conduction becomes:

$$\dot{q}_x = -\overline{K} \left(\frac{\partial \xi}{\partial x} \frac{\partial T}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial T}{\partial \eta} + \frac{\partial \zeta}{\partial x} \frac{\partial T}{\partial \zeta} \right),$$

$$\dot{q}_y = \overline{K} \left(\frac{\partial \xi}{\partial y} \frac{\partial T}{\partial \xi} + \frac{\partial \eta}{\partial y} \frac{\partial T}{\partial \eta} + \frac{\partial \zeta}{\partial y} \frac{\partial T}{\partial \zeta} \right),$$

and

$$\dot{q}_z = \overline{K} \left(\frac{\partial \xi}{\partial z} \frac{\partial T}{\partial \xi} + \frac{\partial \eta}{\partial z} \frac{\partial T}{\partial \eta} + \frac{\partial \zeta}{\partial z} \frac{\partial T}{\partial \zeta} \right).$$

$$\text{The second order tensor} \equiv \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} & \frac{\partial \xi}{\partial z} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} & \frac{\partial \eta}{\partial z} \\ \frac{\partial \zeta}{\partial x} & \frac{\partial \zeta}{\partial y} & \frac{\partial \zeta}{\partial z} \end{bmatrix} \equiv J$$

is called the Jacobian matrix of the transformation whose elements are the nine derivatives specifying the rate of change of the computational coordinates with respect to the physical coordinates. The equations of motion (Eq. (2.1)) are in conservation form [3] and can be expressed as

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = 0, \tag{2.2}$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{bmatrix}, \qquad F = \begin{bmatrix} \rho u \\ \rho uu - \tau_{xx} \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \\ \rho eu + \dot{q}_x - \phi_x \end{bmatrix},$$

$$G = \begin{bmatrix} \rho v \\ \rho uv - \tau_{xy} \\ \rho vv - \tau_{yy} \\ \rho vw - \tau_{xz} \\ \rho ev + \dot{q}_y - \phi_y \end{bmatrix}, \qquad H = \begin{bmatrix} \rho w \\ \rho uw - \tau_{xz} \\ \rho vw - \tau_{yz} \\ \rho ww - \tau_{zz} \\ \rho ew + \dot{q}_z - \phi_z \end{bmatrix}.$$

Equation (2.2) written with respect to the transformed variables of the computational domain becomes:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial F}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial F}{\partial \zeta}\frac{\partial \zeta}{\partial x} + \frac{\partial G}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial G}{\partial \eta}\frac{\partial \eta}{\partial y} + \frac{\partial G}{\partial \zeta}\frac{\partial \zeta}{\partial y}$$

$$+ \frac{\partial H}{\partial \xi}\frac{\partial \xi}{\partial z} + \frac{\partial H}{\partial \eta}\frac{\partial \eta}{\partial z} + \frac{\partial H}{\partial \zeta}\frac{\partial \zeta}{\partial z} = 0. \qquad (2.3a)$$

This can be written compactly as:

$$\frac{\partial U}{\partial t} + \sum_{j=1}^{3}\sum_{i=1}^{3} \frac{\partial \bar{F}_j}{\partial \bar{x}_i} \frac{\partial \bar{x}_i}{\partial \bar{y}_j} = 0, \qquad (2.3b)$$

where

$$\begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{bmatrix} = \begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix}, \quad \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \bar{y}_3 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \begin{bmatrix} \bar{F}_1 \\ \bar{F}_2 \\ \bar{F}_3 \end{bmatrix} = \begin{bmatrix} F \\ G \\ H \end{bmatrix} .$$

Given the Jacobian matrix at each grid point and initial and boundary conditions, the transformed governing equations of motion are in a form to be numerically solved. It is noted at this point that the equations of motion are in weak conservation form relative to the Jacobian derivatives.

2.3 Definition of a Computational Domain and Transformation Data

In the previous section the Navier-Stokes equations are transformed from a Cartesian coordinate system to a computational domain. In so doing nine additional unknowns, which are the elements of the Jacobian matrix, are added to the problem. When the finite difference technique described herein is applied to Equation (2.3), the Jacobian matrix must be known at each grid point. The objective of a grid generation technique is to provide the Jacobian matrix which is henceforth called transformation data. The computational domain is defined in this section along with the formulas necessary for computing the transformation data based on known functional relations between the computational domain and the physical domain. The next three sections concentrate on determining functional relations between the computational and physical domains. The computational domain is defined to be a rectangular parallelepiped and a uniform grid is superimposed onto the domain (Fig. 1) such that:

$$\begin{aligned} &0 \le \xi \le 1, \\ &0 \le \eta \le 1, \\ &0 \le \zeta \le 1, \\ &\Delta\xi = \text{constant}_1, \\ &\Delta\eta = \text{constant}_2, \\ &\Delta\zeta = \text{constant}_3. \end{aligned} \tag{2.4}$$

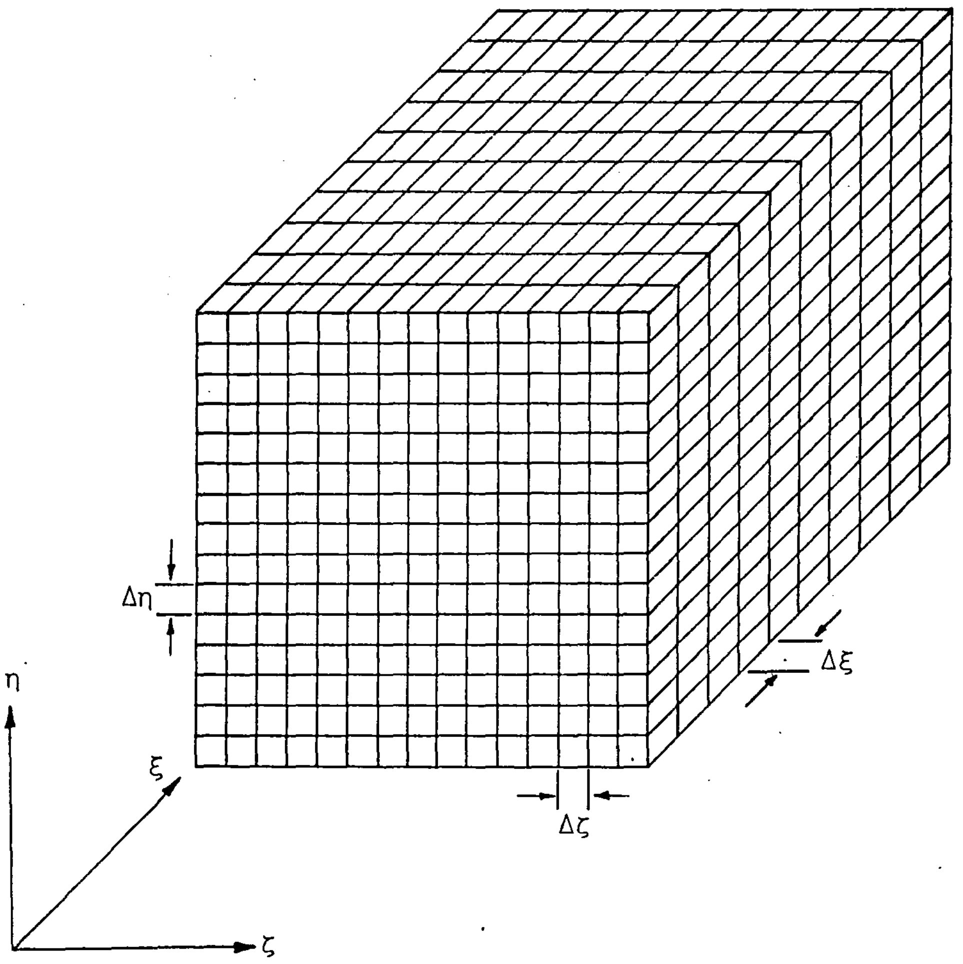


Fig. 1 Computational domain.

A functional relation between the computational domain and the physical domain can be expressed as

$$x = x(\xi,\eta,\zeta), \quad y = y(\xi,\eta,\zeta), \quad \text{and} \quad z = z(\xi,\eta,\zeta). \qquad (2.5)$$

Further, these functions must map boundaries in the computational domain onto boundaries in the physical domain such that

$$x_B = x(\xi_B,\eta_B,\zeta_B), \quad y_B = y(\xi_B,\eta_B,\zeta_B), \quad \text{and} \quad z_B = z(\xi_B,\eta_B,\zeta_B),$$

where x_B, y_B and z_B define the boundaries of the physical domain and ξ_B, η_B, and ζ_B define the boundaries of the computational domain. The transformation data is composed of the rates of change of the computational coordinates with respect to the physical coordinates. If the inverse functional relations

$$\xi = \xi(x,y,z), \quad \eta = \eta(x,y,z), \quad \text{and} \quad \zeta = \zeta(x,y,z) \qquad (2.6)$$

are known, the transformation data can be directly found by differentiation. It is not necessary, however, to know the inverse functional relations to determine the transformation data. The Jacobian matrix can be evaluated by differentiating the functional relation (Eq. (2.5)). That is

$$J^{-1} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}$$

and then

$$J = \frac{\text{Transposed of Cofactor } (J^{-1})}{|J^{-1}|} \;,$$

where $|J^{-1}|$ is the Jacobian determinate and J^{-1} is the inverse Jacobian matrix.

$$|J^{-1}| = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{vmatrix}$$

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

$$+ \frac{\partial x}{\partial \zeta} \left(\frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \eta} \frac{\partial z}{\partial \xi}\right)$$

and

$$\begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} & \frac{\partial \xi}{\partial z} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} & \frac{\partial \eta}{\partial z} \\ \frac{\partial \zeta}{\partial x} & \frac{\partial \zeta}{\partial y} & \frac{\partial \zeta}{\partial z} \end{bmatrix} = J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}^{-1} .$$

$$J = \frac{1}{|J^{-1}|}\begin{bmatrix} (\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \eta}) & -(\frac{\partial x}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial z}{\partial \eta}) & (\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \eta}) \\ -(\frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \xi}) & (\frac{\partial x}{\partial \xi}\frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial z}{\partial \xi}) & -(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \xi}) \\ (\frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \xi}) & -(\frac{\partial x}{\partial \xi}\frac{\partial z}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial z}{\partial \xi}) & (\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}) \end{bmatrix} \tag{2.7}$$

provided $|J^{-1}| \neq 0$.

The transformation data can be pre-evaluated and stored or it can be computed as needed. The trade off is the additional computation cost versus the storage cost. For the Navier-Stokes solver discussed in this study, the transformation data is precomputed and stored for later use.

2.4 Two-Boundary Grid Generation

A computational domain is postulated by Equation (2.4). It is a rectangular parallelpiped in three dimensions and a square in two dimensions. The physical domain is a subdomain of a Cartesian coordinate system. A transformation between the physical domain and the computational domain is a mathematical relationship mapping one domain onto the other. Similarly a grid in one domain is mapped onto a grid in the other domain. When the transformation maps boundaries in the physical domain onto boundaries in the computational domain the term "boundary-fitted coordinate system" is used to describe the transformation.

An indirect (differential) approach for finding the relationship between the computational and physical grids described by Thompson et al. [6-10] has been highly successful. In this approach the elliptic system of partial differential equations which must be satisfied by the mapping between the two domains is numerically solved by an iterative technique such as Successive-Over-Relaxation (SOR). The numerical solution is the grid in the physical domain corresponding to the grid in the computational domain. The transformation data is obtained by numerical differentiation, and a grid change requires a new solution of the elliptic system.

A direct (algebraic) approach, where an explicit functional relationship between the computational domain and the physical domain is known, has the advantages that changes to the grid are direct, rapidly obtained, and transformation data is analytically available.

A direct algebraic approach called the "two-boundary technique" is described in the present paper. The technique has a wide variety of applications in both two and three dimensions. A preliminary description of the technique is presented in [23]. Symbolically, the relation between the computational domain and physical domain can be written as

$$x = X(\xi,\eta,\zeta), \tag{2.8a}$$

$$y = Y(\xi,\eta,\zeta), \tag{2.8b}$$

$$z = Z(\xi,\eta,\zeta), \tag{2.8c}$$

$$0 \leq \xi \leq 1,$$
$$0 \leq \eta \leq 1,$$
$$0 \leq \zeta \leq 1.$$

Equation (2.8) is equivalent to Equation (2.5). For a boundary-fitted relationship between the two domains, boundaries in the computational domain should map onto boundaries in the physical domain as shown in Figure 2. For instance, for the boundaries $\eta = 0$ and $\eta = 1$ in the computation domain, Equation (2.8) becomes

$$x_1 = X(\xi,0,\zeta) = X_1(\xi,\zeta),$$
$$y_1 = Y(\xi,0,\zeta) = Y_1(\xi,\zeta),$$
$$z_1 = Z(\xi,0,\zeta) = Z_1(\xi,\zeta),$$
$$x_2 = X(\xi,1,\zeta) = X_2(\xi,\zeta),$$
$$y_2 = Y(\xi,1,\zeta) = Y_2(\xi,\zeta),$$
$$z_2 = Z(\xi,1,\zeta) = Z_2(\xi,\zeta).$$

Here, $X_1(\xi,\zeta)$, $X_2(\xi,\zeta)$, etc. are boundaries in the physical domain and, as such, are functions defined only at the boundaries. An appropriate explicit expression for Equation (2.8) would separate one variable (η) to be independently varied with parameters dependent on position and derivatives on the boundaries. Since the boundaries are

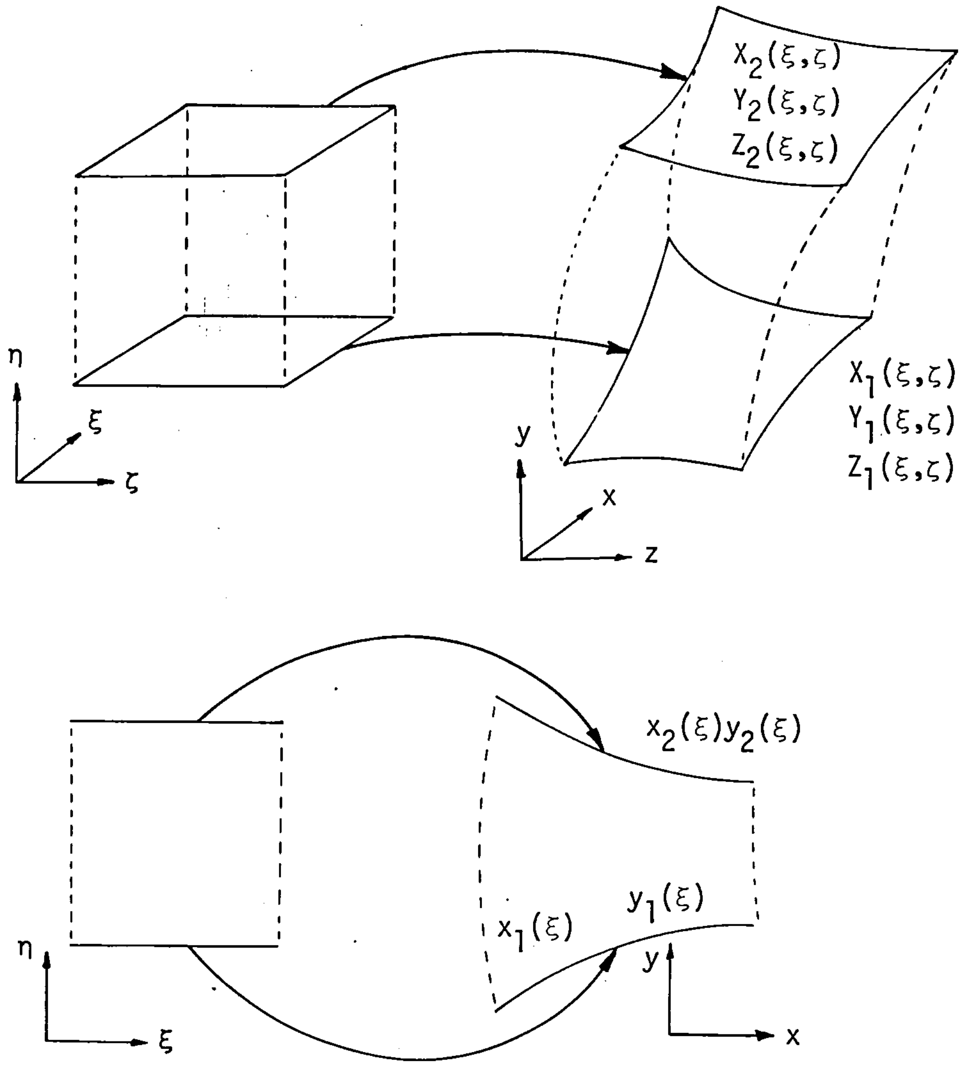


Fig. 2 Boundary mapping from the computational domain to the physical domain.

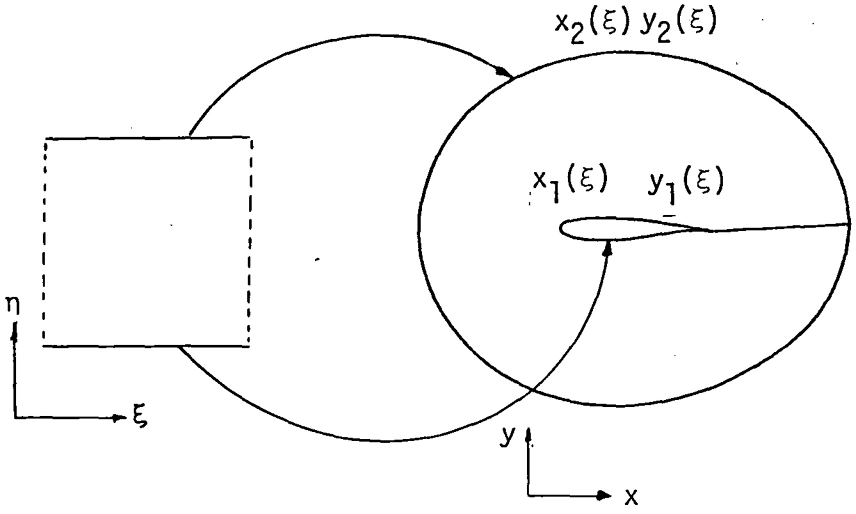


Fig. 2 (Concluded)

themselves functions and can be determined independently, Equation (2.8) can be rewritten as

$$x = X(\xi,\eta,\zeta) = \underline{X}(X_1(\xi,\zeta),\ X_2(\xi,\zeta),\ \eta), \tag{2.9a}$$

$$y = Y(\xi,\eta,\zeta) = \underline{Y}(Y_1(\xi,\zeta),\ Y_2(\xi,\zeta),\ \eta), \tag{2.9b}$$

$$z = Z(\xi,\eta,\zeta) = \underline{Z}(Z_1(\xi,\zeta),\ Z_2(\xi,\zeta),\ \eta). \tag{2.9c}$$

The explicit forms of Equation (2.9) proposed herein are simple parametric linear and cubic polynomials.

Linear

$$x = X_2(\xi,\zeta)\eta + X_1(\xi,\zeta)(1 - \eta), \tag{2.10a}$$

$$y = Y_2(\xi,\zeta)\eta + Y_1(\xi,\zeta)(1 - \eta), \tag{2.10b}$$

$$z = Z_2(\xi,\zeta)\eta + Z_1(\xi,\zeta)(1 - \eta), \tag{2.10c}$$

$$0 \leq \eta \leq 1,$$

Cubic

$$x = X_1(\xi,\zeta)f_1(\eta) + X_2(\xi,\zeta)f_2(\eta) + \frac{dX_1}{d\eta}(\xi,\zeta)f_3(\eta)$$

$$+ \frac{dX_2}{d\eta}(\xi,\zeta)f_4(\eta), \tag{2.11a}$$

$$y = Y_1(\xi,\zeta)f_1(\eta) + Y_2(\xi,\zeta)f_2(\eta) + \frac{dY_1}{d\eta}(\xi,\zeta)f_3(\eta)$$

$$+ \frac{dY_2}{d\eta}(\xi,\zeta)f_4(\eta), \tag{2.11b}$$

$$z = Z_1(\xi,\zeta)f_1(\eta) + Z_2(\xi,\zeta)f_2(\eta) + \frac{dZ_1}{d\eta}(\xi,\zeta)f_3(\eta)$$

$$+ \frac{dZ_2}{d\eta}(\xi,\zeta)f_4(\eta), \tag{2.11c}$$

where:

$$f_1(\eta) = 2\eta^3 - 3\eta^2 + 1,$$

$$f_2(\eta) = -2\eta^3 + 3\eta^2,$$

$$f_3(\eta) = \eta^3 - 2\eta^2 + \eta,$$

$$f_4(\eta) = \eta^3 - \eta^2,$$

$$0 \leq \eta \leq 1.$$

A function such as Equation (2.10) or Equation (2.11) is topologically referred to as a homotopy [24]. Blending-function [11] is another name that has been given to such equations for problems in surface design. Herein, because of the context in which they are used, they are defined as "connecting functions."

Applying a cubic connecting function implies that the physical grid can be forced to be orthogonal at the boundaries since the derivatives $\frac{dX_1}{d\eta}(\xi,\zeta)$, $\frac{dX_2}{d\eta}(\xi,\zeta)$, etc. can be computed from the cross product of the tangential derivatives $\frac{dX_1}{d\xi}(\xi,\zeta)$, $\frac{dX_1}{d\zeta}(\xi,\zeta)$, $\frac{dY_1}{d\xi}(\xi,\zeta)$, $\frac{dY_1}{d\zeta}(\xi,\zeta)$, etc. That is,

$$\frac{dX_\ell}{d\eta}(\xi,\zeta)\,\vec{i} + \frac{dY_\ell}{d\eta}(\xi,\zeta)\,\vec{j} + \frac{dZ_\ell}{d\eta}(\xi,\zeta)\vec{k} =$$

$$K \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \frac{dX_\ell}{d\xi}(\xi,\zeta) & \frac{dY_\ell}{d\xi}(\xi,\zeta) & \frac{dZ_\ell}{d\xi}(\xi,\zeta) \\ \frac{dX_\ell}{d\zeta}(\xi,\zeta) & \frac{dY_\ell}{d\zeta}(\xi,\zeta) & \frac{dZ_\ell}{d\zeta}(\xi,\zeta) \end{vmatrix} \quad \ell = 1,2$$

where $\vec{i}$, $\vec{j}$, and $\vec{k}$ are unit vectors and K is the magnitude of the normal vector. Applying this procedure will force the grid to be orthogonal at the boundaries but not necessarily anywhere else. For the linear connecting function, the physical grid will seldom be orthogonal.

Given the connecting function and parametric boundary functions, a uniform computational grid can be mapped onto the physical domain forming a physical grid. Concentration of grid points in the η direction is accomplished by choosing a function $\bar{\eta} = \bar{\eta}(\eta)$ such that

$0 \leq \eta \leq 1$, $0 \leq \bar{\eta} \leq 1$, and $\frac{d\bar{\eta}}{d\eta} > 0$ (Fig. 3). For example, contracting the physical grid towards one boundary or the other can be accomplished by

$$\bar{\eta} = \frac{e^{\hat{k}\eta} - 1}{e^{\hat{k}} - 1}; \quad 0 \leq \eta \leq 1. \tag{2.12}$$

where $\hat{k}$ is a free parameter whose magnitude dictates the degree of contraction. Embedding this exponential function in the linear connecting function, Equation (2.10) becomes:

$$x = X_2(\xi,\zeta)\bar{\eta} + X_1(\xi,\zeta)(1 - \bar{\eta}),$$

$$y = Y_2(\xi,\zeta)\bar{\eta} + Y_1(\xi,\zeta)(1 - \bar{\eta}),$$

$$z = Z_2(\xi,\zeta)\bar{\eta} + Z_1(\xi,\zeta)(1 - \bar{\eta}),$$

$$0 \leq \bar{\eta} \leq 1.$$

Once the connecting function has been chosen, the remaining problem is the determination of the boundary functions which are independent of η. For the "two-boundary technique" the approach is to choose parametric variables $\bar{s}$ and $\bar{t}$ associated with the boundaries such that

$$x_1(\xi,\zeta) \rightarrow X_1(\bar{s},\bar{t}),$$

$$y_1(\xi,\zeta) \rightarrow Y_1(\bar{s},\bar{t}),$$

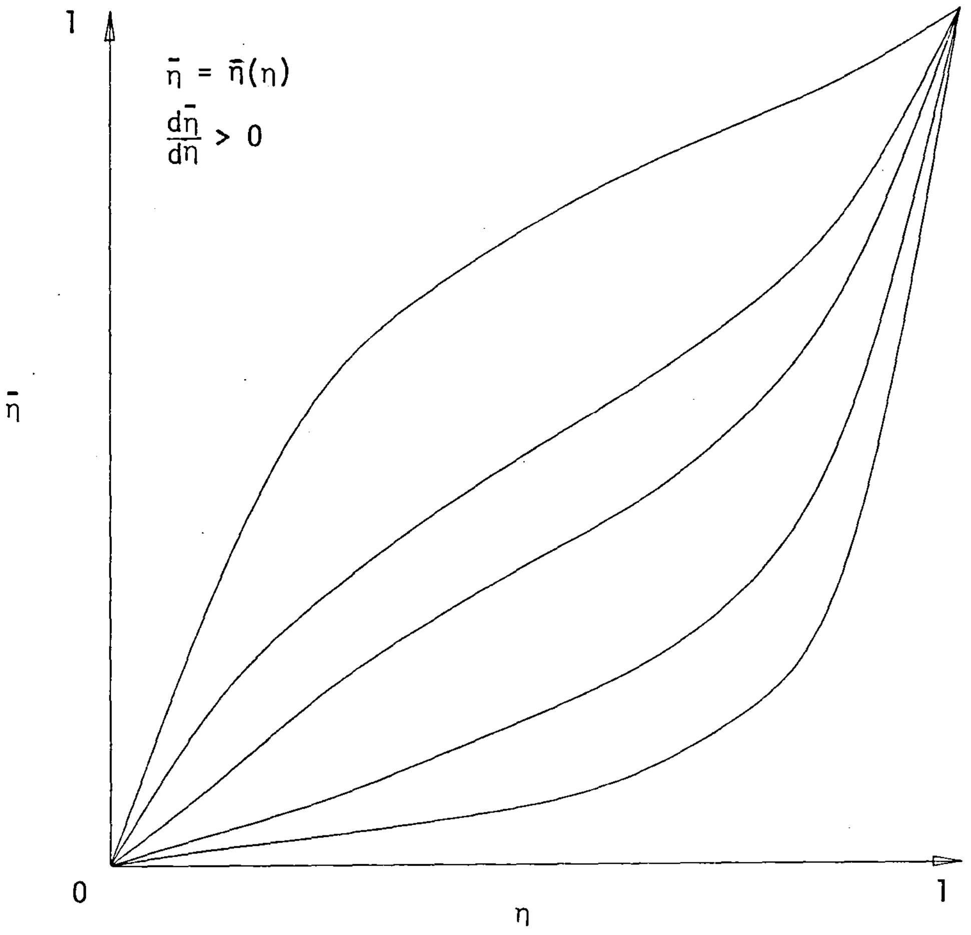


Fig. 3 Grid control function.

$$z_1(\xi,\zeta) \rightarrow Z_1(\bar{s},\bar{t}),$$

$$x_2(\xi,\zeta) \rightarrow X_2(\bar{s},\bar{t}),$$

$$y_2(\xi,\zeta) \rightarrow Y_2(\bar{s},\bar{t}),$$

$$z_2(\xi,\tau) \rightarrow Z_2(\bar{s},\bar{t})$$

$$0 \leq \xi \leq 1, \quad \bar{s}_{min} \leq \bar{s} \leq \bar{s}_{max},$$

$$0 \leq \zeta \leq 1, \quad \bar{t}_{min} \leq \bar{t} \leq \bar{t}_{max}.$$

The choice of parametric variables can vary from problem to problem. A relationship between (ξ,ζ) and $(\bar{s},\bar{t})$ is

$$\bar{s} = \xi(\bar{s}_{max} - \bar{s}_{min}) + \bar{s}_{min},$$

$$\bar{t} = \zeta(\bar{t}_{max} - \bar{t}_{min}) + \bar{t}_{min}.$$

This is a linear relation which maps the unit interval onto the parametric variables. Control of the physical grid at the boundaries is accomplished in the same manner as for the connecting function. That is,

$$\bar{\xi} = \bar{\xi}(\xi), \quad \frac{d\bar{\xi}}{d\xi} > 0,$$

$$\bar{\zeta} = \bar{\zeta}(\zeta), \quad \frac{d\bar{\zeta}}{d\zeta} > 0,$$

$$0 \leq \bar{\xi} \leq 1, \quad 0 \leq \xi \leq 1,$$

$$0 \leq \bar{\zeta} \leq 1, \quad 0 \leq \zeta \leq 1.$$

Since the connecting function is dependent on the boundary position, control of the entire grid is accomplished.

2.4.1 Approximate Boundary-Fitted Coordinate Systems Using Tension Spline Functions

It is often the case that boundaries in a physical domain are described by discrete sets of points. The boundaries may be open or closed (Fig. 2). An approximate boundary-fitted coordinate system can be obtained using the "two-boundary technique" and a tension spline function interpolation to the discrete data defining the boundaries. Tension splines [17-19] are chosen because standard cubic splines [25] and other higher ordered interpolation techniques often result in wiggles in the approximation. Wiggles on a boundary using the "two-boundary technique" propagate into the interior grid. The tension parameter embedded in the tension spline interpolation allows control of the "curvedness" of the approximation. A very large magnitude of the tension parameter corresponds to a linear interpolation whereas a very small value corresponds to cubic splines. Tension splines can be applied in two and three dimensions. A two-dimensional example is presented.

Using the tension spline technique, a point set on boundary one is defined by $\{x_i, y_i\}_{i=1}^{i=n}$ and on boundary two by $\{x_j, y_j\}_{j=1}^{j=m}$.

Approximate arc length is used as a parametric independent variable. The approximate arc length is:

$$s_i = [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{1/2} + s_{i-1},$$

$$s_j = [(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2]^{1/2} + s_{j-1},$$

$$i = 1...n$$

$$j = 1...m$$

$$s_1 = 0$$

$$0 \leq s_i \leq s_n$$

$$0 \leq s_j \leq s_m.$$

From the computational coordinate system the unit interval ($0 \leq \xi \leq 1$) must be mapped onto each boundary; that is:

$$s = s(\xi),$$

$$0 \leq \xi \leq 1.$$

This is accomplished by letting

$s = \xi s_n$ on boundary one and

$s = \xi s_m$ on boundary two.

The tension spline function is a piecewise continuous set of transcendental functions where x and y between the ℓ and $\ell + 1$ points are defined by

$$x = g''(s_\ell)\,\frac{\sinh[\sigma(s_{\ell+1} - s)]}{\sigma^2\sinh[\sigma(s_{\ell+1} - s_\ell)]}$$

$$+ g''(s_{\ell+1})\,\frac{\sinh[\sigma(s - s_\ell)]}{\sigma^2\sinh[\sigma(s_{\ell+1} - s_\ell)]}$$

$$+ \left[x_\ell - \frac{g''(s_\ell)}{\sigma^2}\right]\left(\frac{s_{\ell+1} - s}{s_{\ell+1} - s_\ell}\right)$$

$$+ \left[x_{\ell+1} - \frac{g''(s_{\ell+1})}{\sigma^2}\right]\left(\frac{s - s_\ell}{s_{\ell+1} - s_\ell}\right) , \qquad (2.13)$$

$$y = h''(s_\ell)\,\frac{\sinh[\sigma(s_{\ell+1} - s)]}{\sigma^2\sinh[\ (s_{\ell+1} - s_\ell)]}$$

$$+ h''(s_{\ell+1})\,\frac{\sinh[\sigma(s - s_\ell)]}{\sigma^2\sinh[\sigma(s_{\ell+1} - s_\ell)]}$$

$$+ \left[y_\ell - \frac{h''(s_\ell)}{\sigma^2}\right]\left(\frac{s_{\ell+1} - s}{s_{\ell+1} - s_\ell}\right)$$

$$+ \left[y_{\ell+1} - \frac{h''(s_{\ell+1})}{\sigma^2}\right]\left(\frac{s - s_\ell}{s_{\ell+1} - s_\ell}\right) \tag{2.14}$$

$s = s(\xi) = \xi, s_{max}$,

$\ell = i$ on boundary one,

$\ell = j$ on boundary two,

$\sigma \equiv$ tension parameter.

The unknowns in these equations are $g''(s_\ell)$ and $h''(s_\ell)$ which are second derivatives at the data points $\{x_\ell, y_\ell\}_{\ell=1}^{\ell=N}$ where $N = n$ for boundary one, $N = m$ for boundary two, and are obtained through enforcement of the continuity of the first derivatives at the data points and the specification of two end conditions. A tridiagonal system of linear equations results for each set of unknowns. The solution of the tridiagonal systems yield $g''(s_\ell)$ and $h''(s_\ell)$.

The cubic connecting function (Eq. 2.11), and the exponential function (Eq. 2.12) provide the relationship between the computational domain and the physical domain. The derivatives $\frac{dX_\ell}{d\eta}$ and $\frac{dY_\ell}{d\eta}$ are:

$$\frac{dX_\ell}{d\eta} = K \frac{dY_\ell}{ds} ,$$

$$\frac{dY_\ell}{d\eta} = -K \frac{dX_\ell}{ds} .$$

By defining a grid with constants $\Delta\xi$ and $\Delta\eta$ in the computational domain a corresponding grid is explicitly defined in the physical domain.

An example of a grid about a Kármán-Trefftz airfoil is presented using the spline under tension approximation to the boundaries and a cubic connecting function. Table 1 contains the data describing the airfoil boundary and outer boundary. Figure 4 shows the approximation to the airfoil boundary and Figure 5 shows the grid. A tension parameter value of 2 is used. Transformation data have not been computed for this example.

2.4.2 Transformation for a Wedge-Cylinder Corner

An application of the two-boundary technique using analytical surface functions is a family of three-dimensional corner geometries which occur in many aerodynamic situations (Fig. 6). Supersonic flow about these geometries is characterized by strong visid-inviscid interactions which are adequately analyzed only through the numerical solution of the Navier-Stokes equations. When solving this system of equations with a finite difference technique, a grid must be designed to capture the interactions and allow for accurate application of the boundary conditions.

Table 1. Data description for an airfoil grid

Inside boundary	
x ft.	y ft.
.49950	-.000031
.49860	-.001400
.49600	-.002760
.48620	-.005550
.47060	-.008510
.39010	-.028590
.26960	-.029970
.12270	-.040790
-.03480	-.048450
-.18750	-.050520
-.32110	-.045590
-.42390	-.0377390
-.48650	-.016530
-.50270	.003820
-.47110	.026640
-.39690	.048640
-.28790	.066100
-.15380	.075290
-.00560	.074530
.14400	.064350
.28230	.04750
.39580	.028260
.47200	.011300
.48710	.006810
.49510	.003040
.49880	.001430
.49950	-.000031

Outside boundary	
x ft.	y ft.
0.0	3.0
2.12	2.12
3.0	0.0
-2.12	-2.12
0.0	-3.0
-2.12	-2.12
-3.0	0.0
-2.12	2.12
0.0	3.0

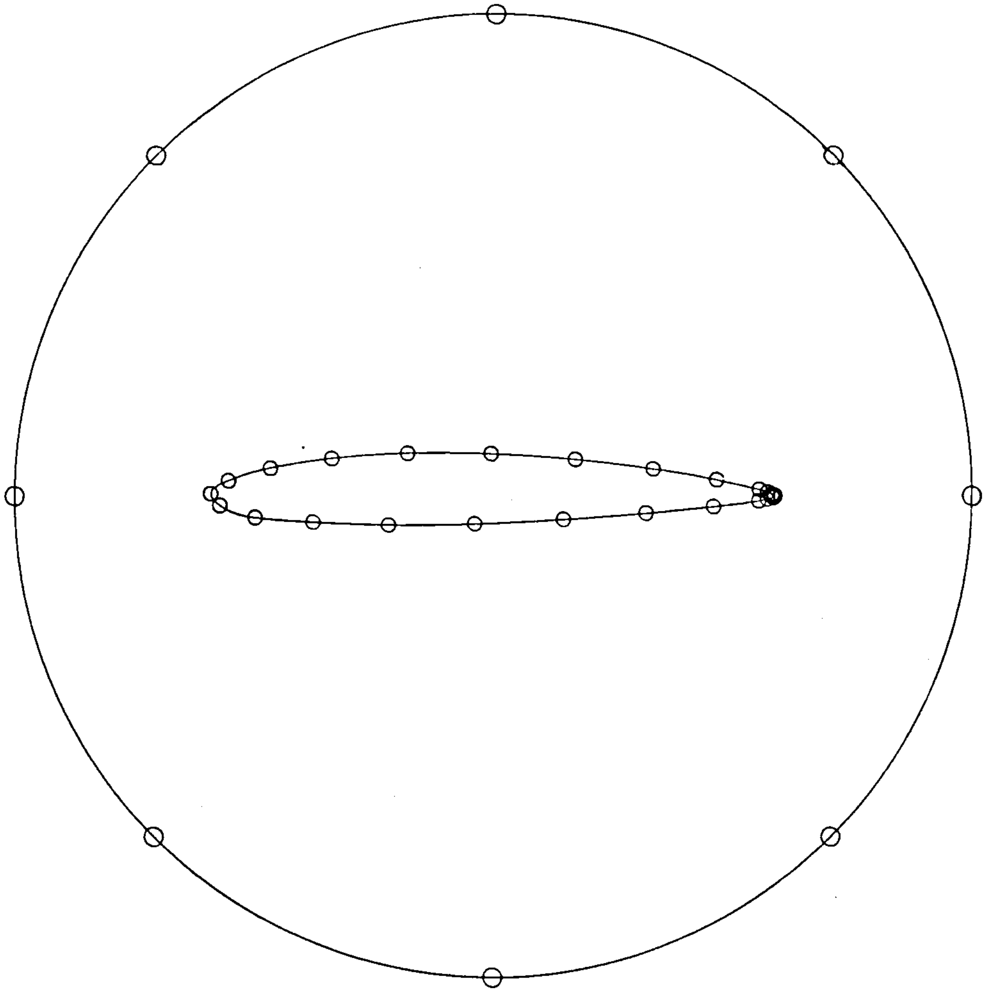


Fig. 4 Boundary definition for Kármán-Trefftz airfoil.

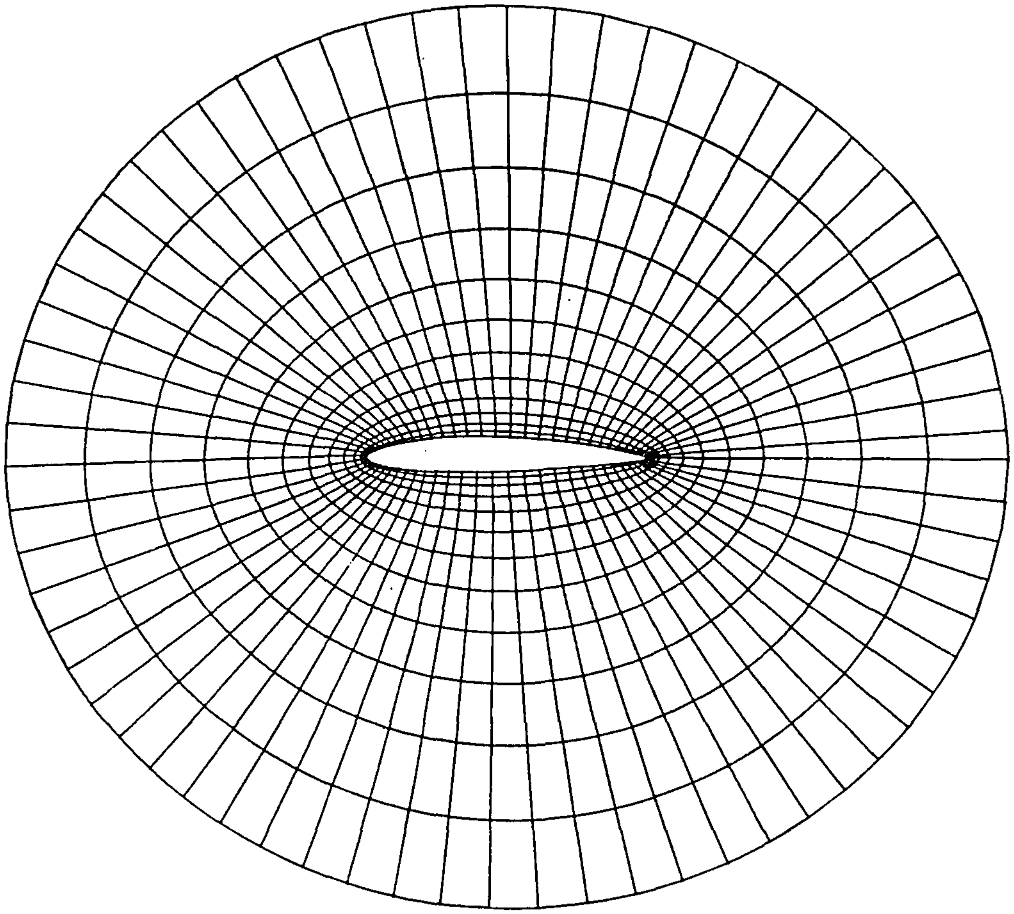
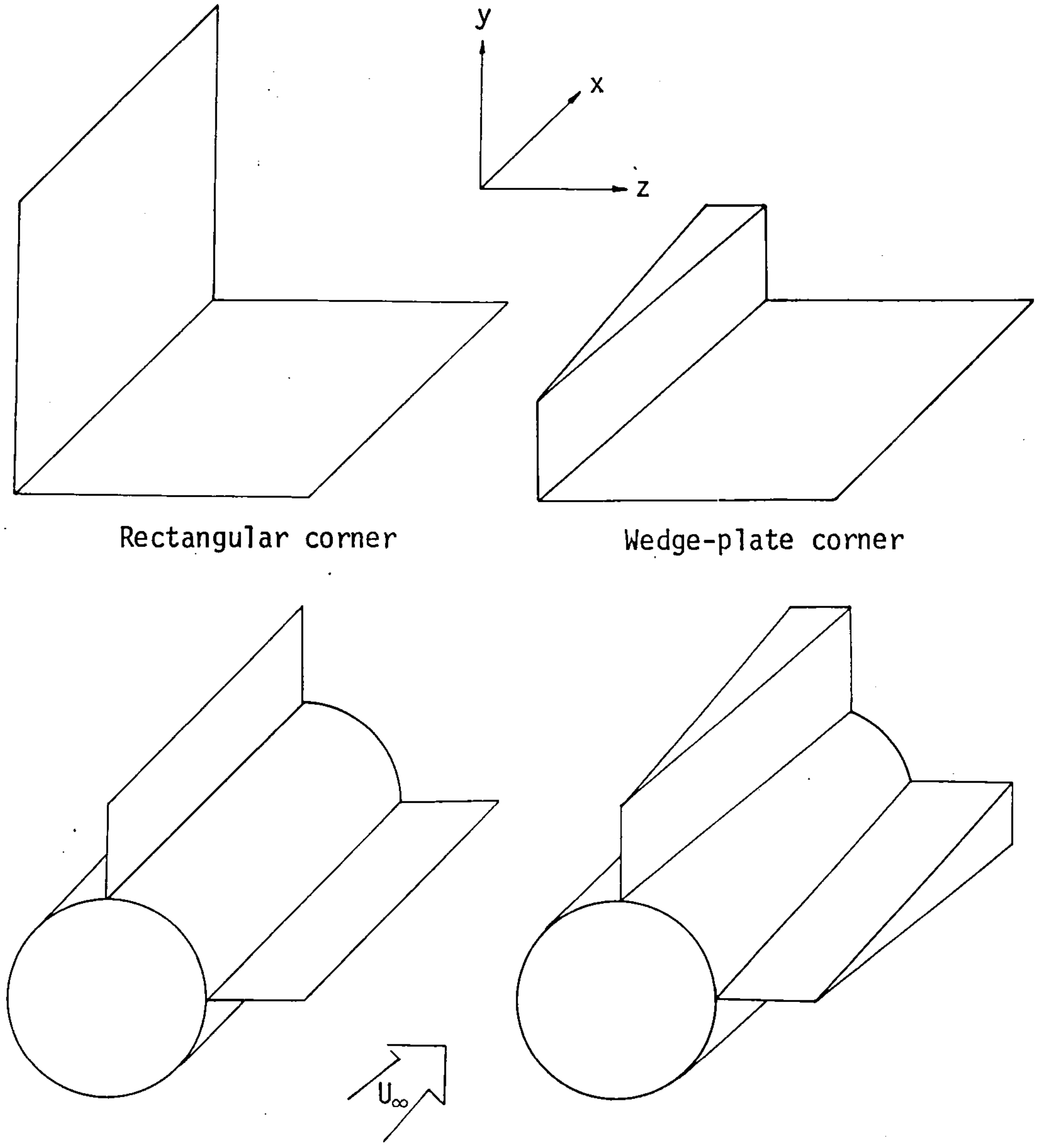


Fig. 5 Grid for Kármán-Trefftz airfoil obtained with the "two-boundary technique".



Rectangular corner

Wedge-plate corner

Plate-cylinder corner

Wedge-cylinder corner

Fig. 6 Three-dimensional corner geometries.

The "two-boundary technique" is applied to the wedge-cylinder corner with the aid of Figure 7. The other corner geometries are derived from the wedge-cylinder definition. The physical domain is the region enclosed by the circular cylinder with radius r_0, an outer surface defined by the wedge angle and a second cylinder radius, and two planes. The left plane (wedge surface) is oriented at an angle ϕ (wedge angle) with the longitudinal axis of the cylinder but parallel to the vertical axis. The right plane (symmetry plane) is oriented with angle θ_2 relative to the vertical axis of the cylinder and includes the longitudinal axis. The upstream and downstream boundaries are cross sections of the region defined by $x = x_0$ and $x = x_L$ and are perpendicular to the longitudinal axis. The "two-boundary technique" is applied to this geometry by considering the inside cylinder surface as boundary one and the outside surface as boundary two. It is desired that ξ, η, and ζ map into the region described above and that

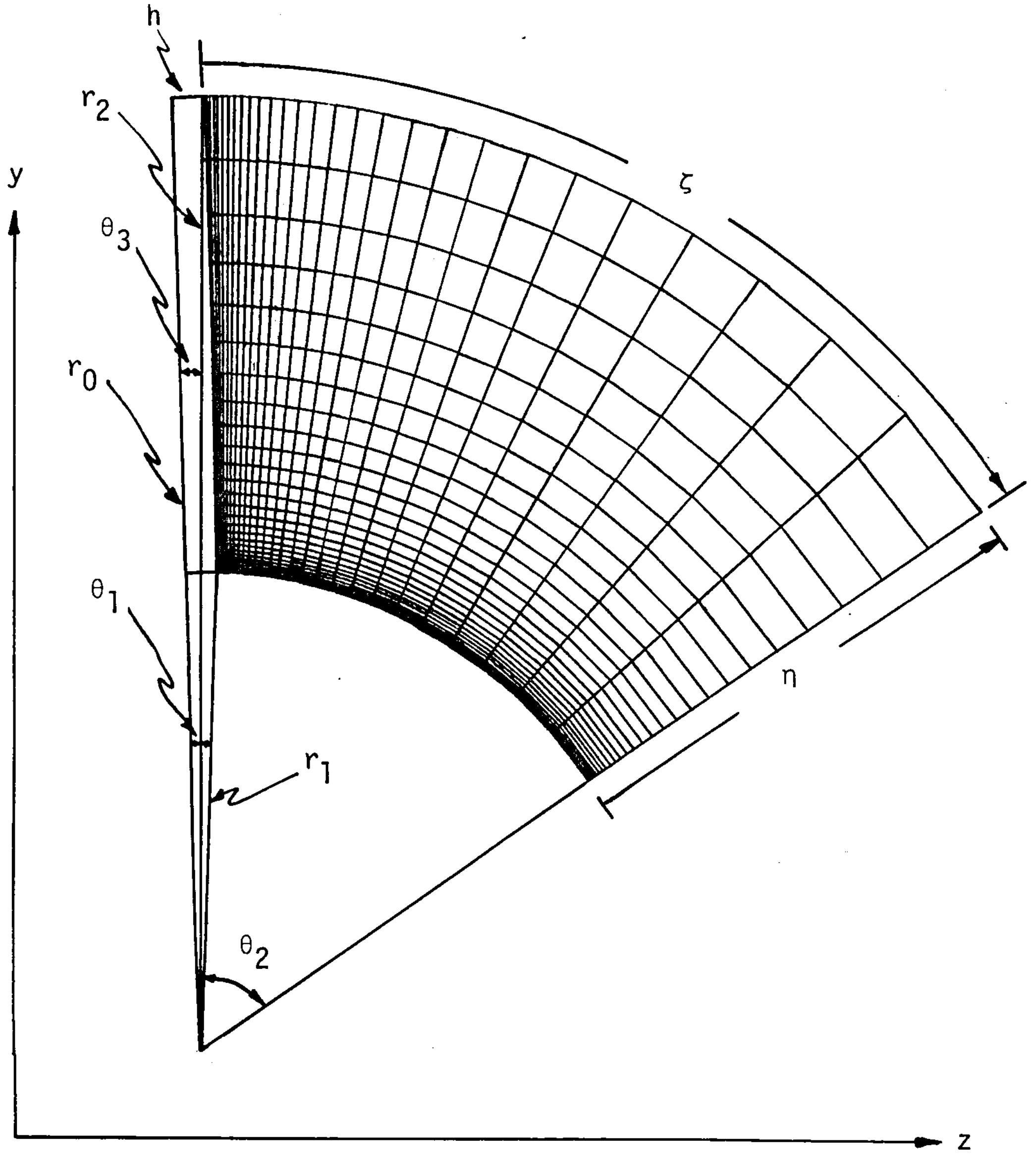
$$0 \leq \xi \leq 1,$$

$$0 \leq \eta \leq 1,$$

$$0 \leq \zeta \leq 1.$$

The boundaries are defined by

$$r_0, r_1, x_0, x_L, \theta_2, \phi, k_1, k_2$$



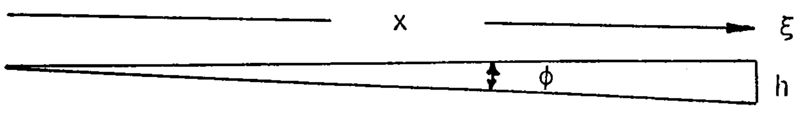


Fig. 7 Projection of wedge onto the x-y plane and cross section of grid in the y-z plane.

Boundary one:

$$X(\xi,0,\zeta) = X(\xi) = (x_L - x_0)\xi + x_0, \tag{2.15a}$$

$$Y(\xi,0,\zeta) = Y_1(\xi,\zeta) = r_0 \cos \psi_1(\xi,\zeta), \tag{2.15b}$$

$$Z(\xi,0,\zeta) = Z_1(\xi,\zeta) = r_0 \sin \psi_1(\xi,\zeta). \tag{2.15c}$$

Boundary two:

$$X(\xi,1,\zeta) = X(\xi) = (x_L - x_0)\xi + x_0, \tag{2.16a}$$

$$Y(\xi,1,\zeta) = Y_2(\xi,\zeta) = r_2(\xi) \cos \psi_2(\xi,\zeta), \tag{2.16b}$$

$$Z(\xi,1,\zeta) = Z_2(\xi,\zeta) = r_2(\xi) \sin \psi_2(\xi,\zeta). \tag{2.16c}$$

Where:

$$\psi_1(\xi,\zeta) = \overline{\xi}\theta_2 + (1 - \overline{\zeta})\theta_1,$$

$$\psi_2(\xi,\zeta) = \overline{\xi}\theta_2 + (1 - \overline{\xi})\theta_3,$$

$$\theta_1 = \sin^{-1}\left(\frac{x(\xi) \tan \phi}{r_0}\right),$$

$$\theta_3 = \sin^{-1}\left(\frac{x(\xi) \tan \phi}{r_1}\right),$$

$$r_2 = [(x(\xi) \tan \phi)^2 + r_1^2]^{1/2},$$

$$\bar{\zeta} = \frac{e^{k_1\zeta} - 1}{e^{k_1} - 1}$$

$$0 \leq \xi \leq 1,$$

$$0 \leq \zeta \leq 1.$$

A linear connecting function is used to generate the internal grid. The function is

$$x = X_2(\xi,\zeta)\bar{\eta} + X_1(\xi,\zeta)(1 - \bar{\eta}) = X(\xi) = (x_L - x_0)\xi + x_0, \tag{2.17a}$$

$$y = Y_2(\xi,\zeta)\bar{\eta} + Y_1(\xi,\zeta)(1 - \bar{\eta}), \tag{2.17b}$$

$$z = Z_2(\xi,\zeta)\bar{\eta} + Z_1(\xi,\zeta)(1 - \bar{\eta}), \tag{2.17c}$$

$$\bar{\eta} = \frac{e^{k_2\eta} - 1}{e^{k_2} - 1}, \tag{2.17d}$$

$$0 \leq \eta \leq 1.$$

An exponential function is used on both η and ζ to concentrate the grid in the corner. Figure 8 shows the grid at $x = x_L$ for corresponding corner surfaces shown in Figure 6. The planar corners are closely approximated by letting the radii be very large.

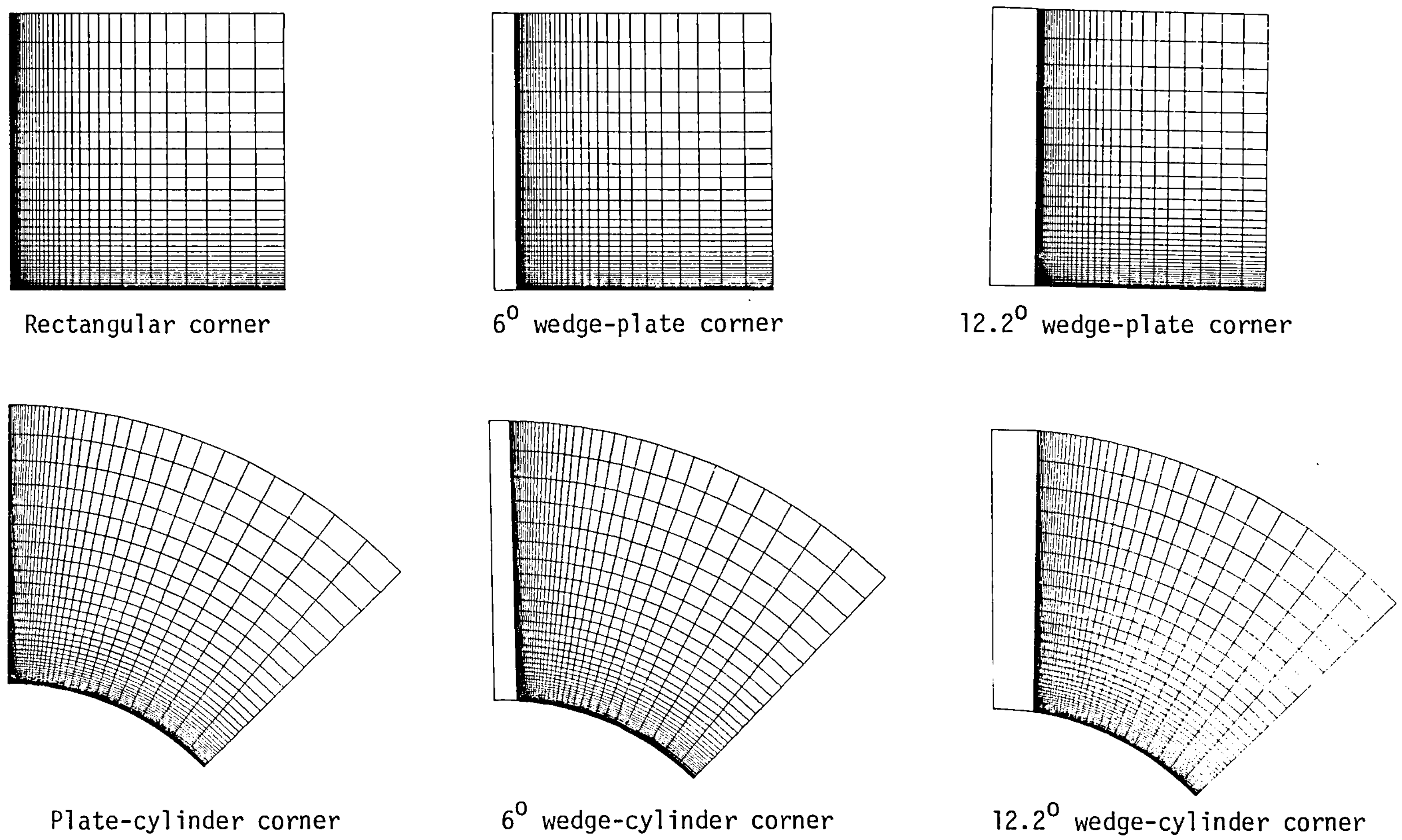


Fig. 8 Grids for wedge-plate and wedge-cylinder corners at $x/x_L = 1$.

Information needed for the equations of motion (Eq. (2.3)) is the transformation data which is obtained from Equation (2.7). The derivatives in Equation (2.7) are obtained by analytic differentiation of x, y, and z (Eq. (2.17)) with respect to ξ, η, and ζ. These derivatives are:

$$\frac{\partial x}{\partial \xi} = x_L - x_0$$

$$\frac{\partial x}{\partial \eta} = 0$$

$$\frac{\partial x}{\partial \zeta} = 0$$

$$\frac{\partial y}{\partial \xi} = \bar{\eta} \frac{\partial y}{\partial \xi} (\xi,1,\zeta) + (1 - \bar{\eta}) \frac{\partial y}{\partial \xi} (\xi,0,\zeta)$$

$$\frac{\partial y}{\partial \eta} = \frac{\partial \bar{\eta}}{\partial \eta} y(\xi,1,\zeta) - \frac{\partial \bar{\eta}}{\partial \eta} y(\xi,0,\zeta)$$

$$\frac{\partial y}{\partial \zeta} = \bar{\eta} \frac{\partial y}{\partial \zeta} (\xi,1,\zeta) + (1 - \bar{\eta}) \frac{\partial y}{\partial \zeta} (\xi,0,\zeta)$$

$$\frac{\partial z}{\partial \xi} = \bar{\eta} \frac{\partial z}{\partial \xi} (\xi,1,\zeta) + (1 - \bar{\eta}) \frac{\partial z}{\partial \xi} (\xi,0,\zeta)$$

$$\frac{\partial z}{\partial \eta} = \frac{\partial \bar{\eta}}{\partial \eta} z(\xi,1,\zeta) - \frac{\partial \bar{\eta}}{\partial \eta} z(\xi,1,\zeta)$$

$$\frac{\partial z}{\partial \zeta} = \bar{\eta} \frac{\partial z}{\partial \zeta} (\xi,1,\zeta) + (1 - \bar{\eta}) \frac{\partial z}{\partial \zeta} (\xi,0,\zeta)$$

where

$$\frac{\partial y}{\partial \xi}\,(\xi,0,\zeta) = -r_1\ \sin[\bar{\zeta}\theta_2 + (1 - \bar{\zeta})\theta_1](1 - \bar{\zeta})\ \frac{\partial \theta_1}{\partial \xi}$$

$$\frac{\partial y}{\partial \xi}\,(\xi,1,\zeta) = \frac{\partial r_2}{\partial \xi}\cos[\bar{\zeta}\theta_2 + (1 - \bar{\zeta})\theta_3] - r_2\ \sin(\bar{\zeta}\theta_2$$

$$- (1 - \bar{\zeta})\theta_3)(1 - \bar{\zeta})\ \frac{\partial \theta_3}{\partial \xi}$$

$$\frac{\partial y}{\partial \zeta}\,(\xi,0,\zeta) = -r_1\ \sin[\bar{\zeta}\theta_2 + (1 - \bar{\zeta})\theta_1][(\theta_2 - \theta_1)]\ \frac{\partial \bar{\zeta}}{\partial \zeta}$$

$$\frac{\partial y}{\partial \zeta}\,(\xi,1,\zeta) = -r_2\ \sin[\bar{\zeta}\ {}_2 + (1 - \bar{\zeta})\theta_3][(\theta_2 - \theta_3)]\ \frac{\partial \bar{\zeta}}{\partial \zeta}$$

$$\frac{\partial z}{\partial \xi}\,(\xi,0,\zeta) = r_1\ \cos[\bar{\zeta}\theta_2 + (1 - \bar{\zeta})\theta_1][1 - \bar{\zeta}]\ \frac{\partial \theta_1}{\partial \xi}$$

$$\frac{\partial z}{\partial \xi}\,(\xi,1,\zeta) = \frac{\partial r_2}{\partial \xi}\sin[\bar{\zeta}\theta_2 + (1 - \bar{\zeta})\theta_3] + r_2\ \cos[\bar{\zeta}\theta_2$$

$$+ (1 - \bar{\zeta})\theta_3](1 - \bar{\zeta})\ \frac{\partial \theta_3}{\partial \xi}$$

$$\frac{\partial z}{\partial \zeta}\,(\xi,0,\zeta) = r_1\ \cos[\bar{\xi}\theta_2 + (1 - \bar{\zeta})\theta_1][\theta_2 - \theta_1]\ \frac{\partial \bar{\zeta}}{\partial \zeta}$$

$$\frac{\partial z}{\partial \zeta}\,(\xi,1,\zeta) = r_2\ \sin[\bar{\zeta}\theta_2 + (1 - \bar{\zeta})\theta_3][\theta_2 - \theta_3]\ \frac{\partial \bar{\zeta}}{\partial \zeta}$$

$$\frac{\partial\theta_1}{\partial\xi} = \frac{1}{\left[1 + \left(\frac{x \tan\phi}{r_1}\right)^2\right]^{1/2}} \cdot \frac{\partial x}{\partial\xi}\,\frac{\tan\phi}{r_1}$$

$$\frac{\partial\theta_3}{\partial\xi} = \frac{1}{\left[1 + \left[\frac{x \tan\phi}{r_2}\right]^2\right]^{1/2}} \cdot \frac{\partial}{\partial\xi}\left(\frac{x \tan\phi}{r_2}\right)$$

$$\frac{\partial}{\partial\xi}\left(\frac{x \tan\phi}{r_2}\right) = \left[r_2 \tan\phi \,\frac{\partial x}{\partial\xi} - x \tan\phi \,\frac{\partial r_2}{\partial\xi}\right]/r_2^{\,2}$$

$$\frac{\partial r_2}{\partial\xi} = \frac{1}{r_2}\, x \tan\phi \,\frac{\partial x}{\partial\xi}$$

The application of this technique in the Navier-Stokes solver is found in Chapter 4.

2.4.3 Transformation for a Spike-Nosed Body

The "two-boundary technique" is applied to generate grids about spike-nosed bodies (Fig. 9 and Fig. 10). Supersonic flow about these bodies is unsteady and separation occurs in the nose-shoulder region. Consequently, grids must be concentrated in the nose-shoulder region and be adequately spaced to define the shock and the boundary layer on top of the shoulder. A linear approximation to the inner boundary and a circular arc outer boundary are used in the two-boundary technique. Concentration of the grid is accomplished by superimposing an exponential function onto the connecting function and a combined exponential and parabolic algebraic function is superimposed onto the parametric variable along the boundaries. The grid is cast in three-dimensions by rotating the two-dimensional description about the axis of symmetry.

The inside boundary is defined by the set of points

$$\left\{\hat{x}^I_\ell,\ \hat{y}^I_\ell\right\}_{\ell=1}^{\ell=M}$$

The parametric variable associated with this boundary is accumulated cord length where

$$\hat{t}_1 = 0, \quad \text{and}$$

$$\hat{t}_\ell = [(\hat{x}^I_\ell - \hat{x}^I_\ell)^2 + (\hat{y}^I_\ell - \hat{y}^I_{\ell-1})^2]^{1/2} + \hat{t}_{\ell-1}.$$

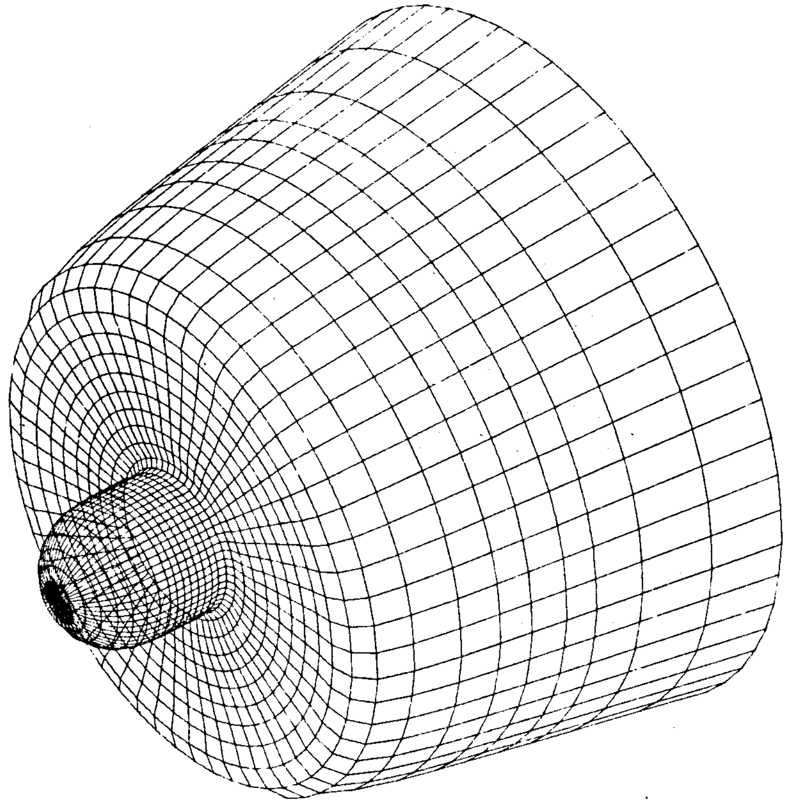


Fig. 9 One-half inch spike-nosed body.

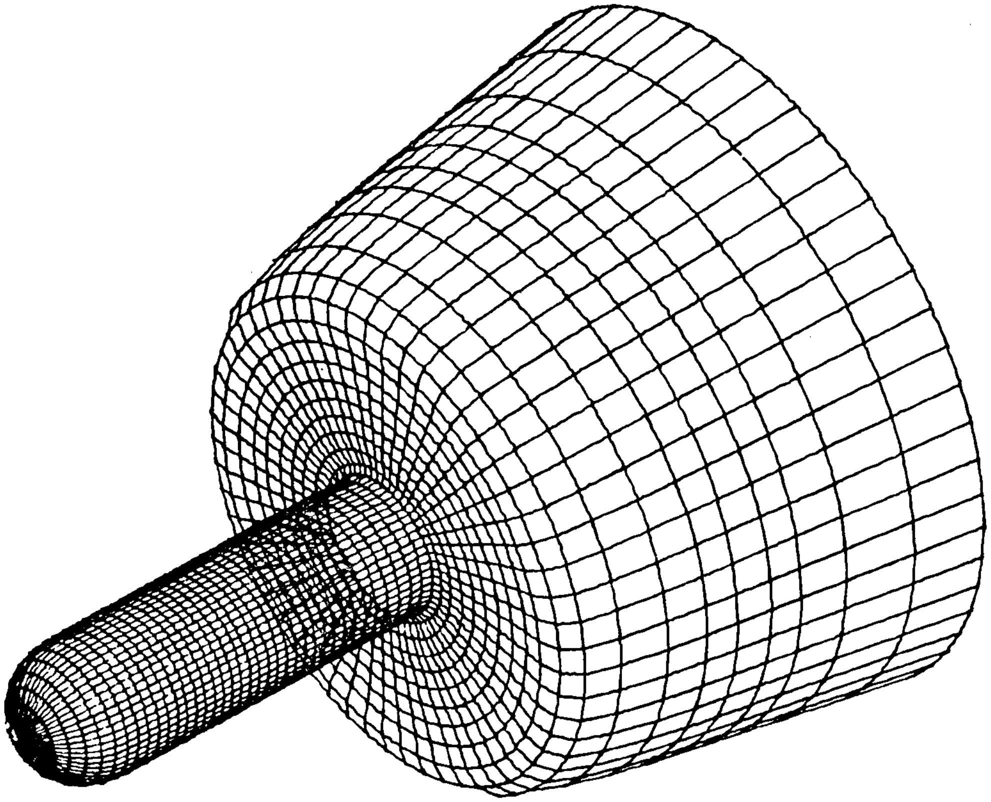


Fig. 10 One and one-half inch spike-nosed body.

With the parametric variable defined, two data sets are formed. They are:

$$\left\{\hat{t}_{\ell},\ \hat{x}^{I}_{\ell}\right\}_{\ell=1}^{\ell=M} \qquad \left\{\hat{t}_{\ell},\ \hat{y}^{I}_{\ell}\right\}_{\ell=1}^{\ell=M} .$$

A linear approximation to the inside boundary for a spike-nosed body is accomplished by linear interpolation of the above data sets. The transformation data requires the derivatives of the boundary definition with respect to the parametric variable and the derivative sets are formed and saved for later use. The derivative sets are:

$$\left\{\hat{t}_{\alpha},\ \frac{d\hat{x}^{I}}{dt_{\alpha}}\right\}_{\alpha=1}^{M-1} , \qquad \left\{\hat{t}_{\alpha},\ \frac{d\hat{y}^{I}}{dt_{\alpha}}\right\}_{\alpha=1}^{\alpha=M-1} ,$$

where

$$\frac{d\hat{x}^{I}}{d\hat{t}_{\alpha}} = \frac{\hat{x}_{\alpha+1} - \hat{x}_{\alpha-1}}{\hat{t}_{\alpha+1} - \hat{t}_{\alpha-1}} , \quad \frac{d\hat{y}^{I}}{d\hat{t}_{\alpha}} = \frac{\hat{y}^{I}_{\alpha+1} - \hat{y}^{I}_{\alpha-1}}{\hat{t}_{\alpha+1} - \hat{t}_{\alpha-1}}$$

$$\alpha = 2 \ . \ . \ . \ . \ M-1 .$$

The linear interpolation to the point sets can be symbolically written

$$\hat{x}^I = \mathrm{Lin}_{\hat{x}}\left(\left\{\hat{t}_\ell,\ \hat{x}^I_\ell\right\}_{\ell=1}^{\ell=M}\right), \quad \frac{d\hat{x}^I}{dt} = \mathrm{Lin}_{d\hat{x}}\left(\left\{\hat{t}_\alpha,\ \frac{d\hat{x}^I}{d\hat{t}}\right\}_{\alpha=2}^{\alpha=M-1}\right) \quad ,$$

$$\hat{y}^I = \mathrm{Lin}_{\hat{y}}\left(\left\{\hat{t}_\ell,\ y^I_\ell\right\}_{\ell=1}^{\ell=M}\right), \quad \frac{d\hat{y}^I}{dt} = \mathrm{Lin}_{d\hat{y}}\left(\left\{\hat{t}_\alpha,\ \frac{d\hat{y}^I}{d\hat{t}}\right\}_{\alpha=2}^{\alpha=M-1}\right),$$

where Lin denotes a linear interpolation.

The outside (top) boundary is a circular arc defined by

$$\hat{x}^o = -R\cos\theta + \hat{x}_o$$

$$\hat{y}^o = R\sin\theta + \hat{y}_o.$$

The physical domain in two dimensions is the region between the inside and outside boundaries. A rectangular region can be mapped onto the physical domain based on information from the boundaries and a linear connecting function. The transformation is

$$\hat{x} = \hat{x}^o\bar{\eta} + \hat{x}^I(1 - \bar{\eta})$$

$$\hat{y} = \hat{y}^o\bar{\eta} + \hat{y}^I(1 - \bar{\eta}).$$

Given the computational grid $\{\zeta_{jk}, \eta_{jk}\}^{\substack{j=m\\k=m}}_{\substack{j=1\\k=1}}$ the physical grid is

$$\{\hat{x}_{jk}, \hat{y}_{jk}\}^{\substack{j=n\\j=m}}_{\substack{j=1\\k=1}}$$

where

$$\hat{x}_{jk} = \hat{x}^{o}_{k}\bar{\eta}_j + \hat{x}^{I}_{k}(1 - \bar{\eta}_j)$$

$$\hat{y}_{jk} = \hat{y}^{o}_{k}\bar{\eta}_j + \hat{y}^{I}_{k}(1 - \bar{\eta}_j).$$

$$\theta_k = (\theta_{max} - \theta_{min})\bar{\zeta}_k + \theta_{min},$$

$$\hat{t}_k = \hat{t}_M\bar{\zeta}_k \qquad 0 \leq \bar{\zeta} \leq 1.$$

Concentration of the grid near the inside boundary and in the nose-shoulder region is accomplished by intermediate transformations

$$\bar{\eta} = \frac{e^{\bar{k}\eta} - 1}{e^{\bar{k}} - 1}, \qquad 0 \leq \bar{\eta} \leq 1,$$

and

$$\bar{\zeta} = \frac{e^{\bar{k}_1\zeta} - 1}{(e^{\bar{k}_1} - 1)(1 + B\zeta - B\zeta^2)}, \qquad 0 \leq \bar{\zeta} \leq 1.$$

The parameters $\bar{k}$, $\bar{k}_1$, and B govern the concentration of the grid pcints. The above derivation for the transformation between a computation domain and a physical domain is for a two-dimensional slice of a flow field. A three-dimensional representation is

$$x_{ijk} = \hat{y}_{jk}(\sin[\xi_i(\bar{\phi}_{max} - \bar{\phi}_{min}) + \bar{\phi}_{min}]) \tag{2.18a}$$

$$y_{ijk} = \hat{y}_{jk}(\cos[\xi_i(\bar{\phi}_{max} - \bar{\phi}_{min}) + \bar{\phi}_{min}]) \tag{2.18b}$$

$$z_{ijk} = \hat{x}_{jk} \tag{2.18c}$$

$$0 \leq \xi \leq 1.$$

The elements of the Jacobian matrix are:

$$\frac{\partial x}{\partial \xi} = (\bar{\phi}_{max} - \bar{\phi}_{min})\hat{y}(\cos[\xi(\bar{\phi}_{max} - \bar{\phi}_{min}) + \bar{\phi}_{min}]),$$

$$\frac{\partial x}{\partial \eta} = [\sin(\xi[\bar{\phi}_{max} - \bar{\phi}_{min}] + \bar{\phi}_{min})] \frac{\partial \hat{y}}{\partial \eta} ,$$

$$\frac{\partial x}{\partial \zeta} = [\sin(\xi[\bar{\phi}_{max} - \bar{\phi}_{min}] + \bar{\phi}_{min})] \frac{\partial \hat{y}}{\partial \zeta} ,$$

$$\frac{\partial y}{\partial \xi} = -(\bar{\phi}_{max} - \bar{\phi}_{min})\hat{y}(\sin[\xi(\bar{\phi}_{max} - \bar{\phi}_{min}) + \bar{\phi}_{min}]),$$

$$\frac{\partial y}{\partial \eta} = [\cos(\xi[\bar{\phi}_{max} - \bar{\phi}_{min}] + \bar{\phi}_{min})] \frac{\partial \hat{y}}{\partial \eta} ,$$

$$\frac{\partial y}{\partial \zeta} = [\cos(\xi[\bar{\phi}_{max} - \bar{\phi}_{min}] + \bar{\phi}_{min})] \frac{\partial \hat{y}}{\partial \zeta} ,$$

$$\frac{\partial z}{\partial \xi} = 0,$$

$$\frac{\partial z}{\partial \eta} = \frac{\partial \hat{x}}{\partial \eta} ,$$

$$\frac{\partial z}{\partial \zeta} = \frac{\partial \hat{x}}{\partial \zeta} ,$$

$$\frac{\partial \hat{x}}{\partial \eta} = \hat{x}^{o} \frac{\partial \bar{\eta}}{\partial \eta} + \hat{x}^{I} \frac{\partial \bar{\eta}}{\partial \eta} ,$$

$$\frac{\partial \hat{x}}{\partial \zeta} = \bar{\eta} \frac{\partial \hat{x}^{o}}{\partial \theta} \frac{\partial \theta}{\partial \bar{\zeta}} \frac{\partial \zeta}{\partial \zeta} + (1 - \bar{\eta}) \frac{\partial \hat{x}^{I}}{\partial \hat{t}} \frac{\partial \hat{t}}{\partial \bar{\zeta}} \frac{\partial \bar{\zeta}}{\partial \zeta} ,$$

$$\frac{\partial \hat{y}}{\partial \eta} = \hat{y}^{o} \frac{\partial \bar{\eta}}{\partial \eta} - \hat{y}^{I} \frac{\partial \bar{\eta}}{\partial \eta} ,$$

$$\frac{\partial \hat{y}}{\partial \zeta} = \bar{\eta} \frac{\partial \hat{y}^{o}}{\partial \theta} \frac{\partial \theta}{\partial \bar{\zeta}} \frac{\partial \bar{\zeta}}{\partial \zeta} + (1 - \eta) \frac{\partial \hat{y}^{I}}{\partial \hat{t}} \frac{\partial \hat{t}}{\partial \bar{\zeta}} \frac{\partial \bar{\zeta}}{\partial \zeta} ,$$

$$\frac{\partial \bar{\eta}}{\partial \eta} = \bar{k} \frac{e^{\bar{k}\eta}}{e^{\bar{k}} - 1} ,$$

$$\frac{\partial \bar{\zeta}}{\partial \zeta} = \frac{1}{e^{\bar{k}_1} - 1} \left[\frac{(1 + B\zeta - B\zeta^2)\bar{k}_1 (e^{\bar{k}_1 \zeta} - 1)(B - 2B\zeta)}{(1 + B\zeta - B\zeta^2)^2} \right]$$

$$\frac{\partial \hat{x}^{o}}{\partial \theta} = R \sin \theta,$$

$$\frac{\partial \hat{y}^{o}}{\partial \theta} = R \cos \theta,$$

$$\frac{\partial \theta}{\partial \bar{\zeta}} = (\theta_{max} - \theta_{min}),$$

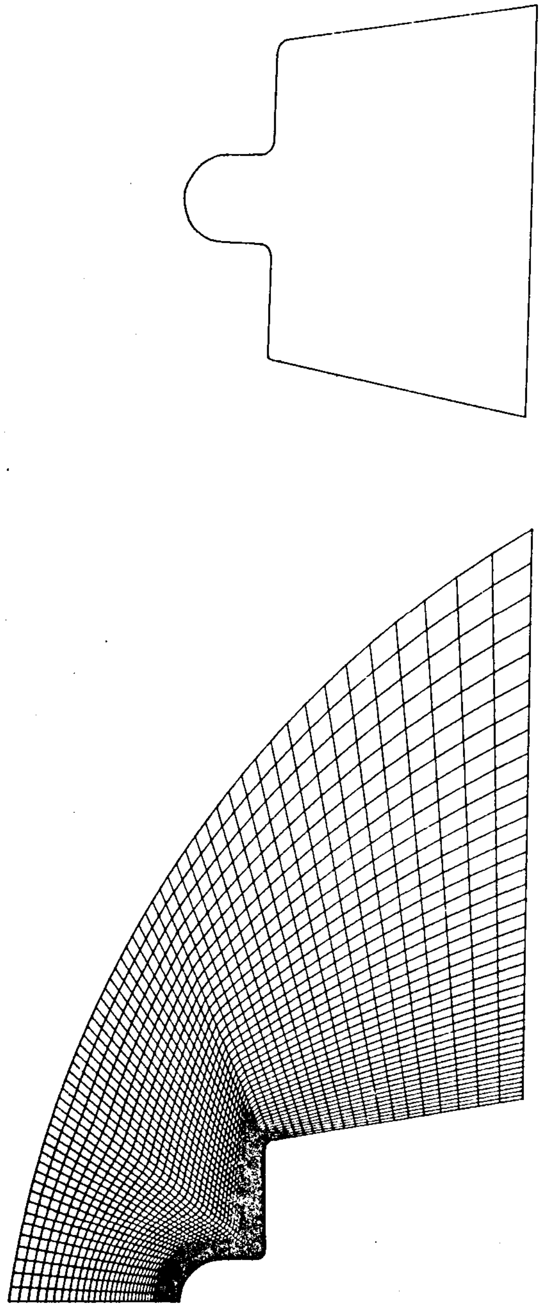


Fig. 11 Grid generated with the "two-boundary technique" for a one-half inch spike-nosed body.

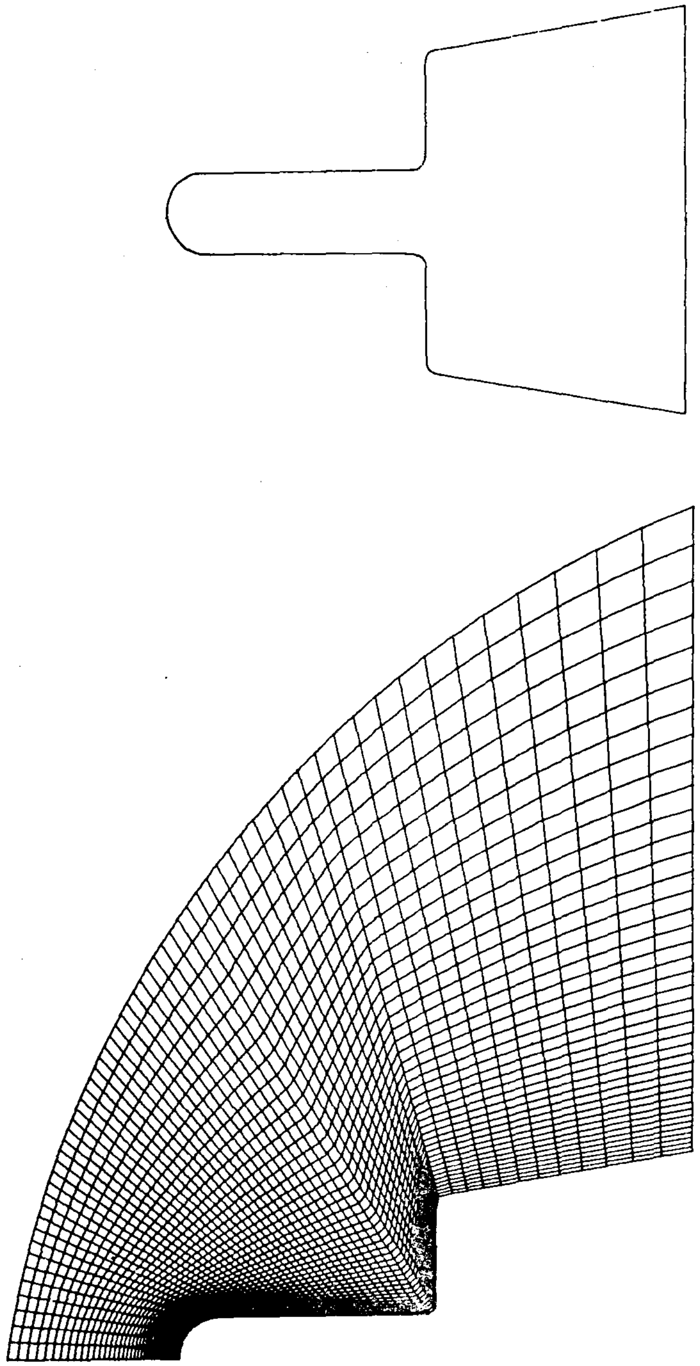


Fig. 12 Grid generated with the "two-boundary technique" for a one and one-half inch spike-nosed body.

Table 3. Data description for a one and one-half inch spike-nosed body

Inside boundary								
Pt	x	y	Pt	x	y	Pt	x	y
1	0.0	0.0	21	.01042	.01804	41	.12475	.02357
2	.00003	.00109	22	.01138	.01856	42	.12494	.02428
3	.00011	.00218	23	.01236	.01903	43	.12500	.02500
4	.00026	.00326	24	.01337	.01945	44	.12500	.02667
5	.00046	.00433	25	.01440	.01981	45	.12500	.07417
6	.00071	.00539	26	.01544	.02012	46	.12500	.07533
7	.00102	.00644	27	.01650	.02038	47	.12506	.07606
8	.00138	.00747	28	.01757	.02058	48	.12525	.07676
9	.00180	.00847	29	.01866	.02072	49	.12556	.07742
10	.00227	.00946	30	.01974	.02080	50	.12597	.07801
11	.00279	.01042	31	.02083	.02083	51	.12649	.07853
12	.00336	.01135	32	.02163	.02083	52	.12708	.07894
13	.00398	.01225	33	.12000	.02083	53	.12774	.07925
14	.00464	.01311	34	.12083	.02083	54	.12844	.07944
15	.00535	.01394	35	.12156	.02090	55	.12873	.07948
16	.00610	.01473	36	.12226	.02108	56	.12883	.07949
17	.00689	.01548	37	.12292	.02139	57	.25000	.09929
18	.00772	.01619	38	.12351	.02181			
19	.00859	.01685	39	.12403	.02232			
20	.00949	.01747	40	.12444	.02292			

Outside Boundary

x_0 = .43333 ft.	R = .52208 ft.	θ_f = 69.5°
k_2 = .0001	y_0 = -.07500 ft.	θ_0 = 8.295°
k_1 = 2.2	B = 2	

Table 2. Data description for a one-half inch spike-nosed body

Inside Boundary								
Pt	x	y	Pt	x	y	Pt	x	y
1	0.0	0.0	21	.01042	.01804	41	.04142	.02397
2	.00003	.00109	22	.01138	.01856	42	.04161	.02428
3	.00011	.00218	23	.02136	.01903	43	.04167	.02500
4	.00026	.00326	24	.01337	.01945	44	.04167	.02667
5	.00046	.00433	25	.01440	.07981	45	.04167	.07417
6	.00071	.00539	26	.01544	.02012	46	.04167	.07533
7	.00102	.00644	27	.01650	.02038	47	.04173	.07606
8	.00138	.00747	28	.01757	.02058	48	.04192	.07676
9	.00180	.00847	29	.01866	.02072	49	.04223	.07742
10	.00227	.00946	30	.01974	.02080	50	.04264	.07801
11	.00279	.01042	31	.02083	.02083	51	.04316	.07853
12	.00336	.01135	32	.02163	.02083	52	.04375	.07894
13	.00398	.01225	33	.03666	.02083	53	.04441	.07925
14	.00464	.01311	34	.03749	.02083	54	.04511	.07944
15	.00535	.01394	35	.03822	.02090	55	.04550	.07948
16	.00610	.01473	36	.03892	.02108	56	.04549	.07949
17	.00689	.01548	37	.03958	.02139	57	.16687	.09929
18	.00772	.01619	38	.04018	.02181			
19	.00859	.01685	39	.04070	.02232			
20	.00949	.01147	40	.04111	.02292			

Outside Boundary		
x_0 = .4333 ft.	Radius = .52208 ft.	θ_f = 51.55°
$\bar{k}_2$ = .0001	y_0 = .07500 ft.	θ_0 = 8.295°
$\bar{k}_1$ = 2.2	B = L	

$$\frac{\partial\hat{x}^I}{\partial\hat{t}} = \mathrm{Lin}_{dx}\left(\left\{\hat{t}_\alpha, \frac{d\hat{x}^I}{d\hat{t}_\alpha}\right\}_{\alpha=2}^{M=1}\right) ,$$

$$\frac{\partial\hat{y}^I}{\partial\hat{t}} = \mathrm{Lin}_{dy}\left(\left\{\hat{t}_\alpha, \frac{d\hat{y}^I}{d\hat{t}_\alpha}\right\}_{\alpha=2}^{M=1}\right) ,$$

$$\frac{\partial\hat{t}}{\partial\bar{\zeta}} = \hat{t}_m .$$

Grids generated with this application of the "two-boundary technique" to the bodies shown in Figures 9 and 10 are shown in Figures 11 and 12. Tables 2 and 3 give the data used to generate the grids. The use of the grid in the Navier-Stokes solver is discussed in Chapter 4.

2.5 Initial and Boundary Conditions

Initial conditions are free stream conditions except at solid boundaries where no slip is imposed on the velocity. Free stream conditions are established from the Mach number $\equiv M_\infty$, Reynolds number $\equiv R_{e_\infty} = \mathrm{Re}$, characteristic length $\equiv L$, and free stream temperature $\equiv T_\infty$. The speed of sound is

$$c_\infty = [\gamma(\gamma - 1)C_V T_\infty]^{1/2}$$

where $\gamma = 1.4$ and $C_V = 4290 \, (\frac{ft}{sec})^2 \frac{1}{deg}$ for air which is considered a perfect gas. The free stream velocity is

$$u_\infty = M_\infty c_\infty ,$$

$$v_\infty = 0 ,$$

and

$$w_\infty = 0 .$$

The free stream viscosity is

$$\mu_\infty = \frac{2.27 \times 10^{-8} T_\infty^{3/2}}{T_\infty + 198.6} .$$

The free stream density, pressure, and energy are

$$\rho_\infty = \mu_\infty u_\infty L / R_{e_\infty} ,$$

$$P_\infty = \rho R T_\infty ,$$

and

$$e_\infty = C_V T_\infty .$$

No slip boundary conditions are imposed on the velocity at solid walls. A solid wall is considered to be isothermal and the temperature T_W is fixed. The boundary condition on energy is $e_W = C_V \cdot T_W$. The solid wall boundary condition for density $\equiv \rho_W$ is obtained through a condition on pressure at the wall and the relation of density to pressure in the equation of state. The wall pressure boundary condition is obtained by approximately satisfying the momentum equations at the wall. Assuming that the gradient of the shear stress is zero at a solid wall implies that $\frac{\partial P}{\partial \bar{N}} = 0$ where $\bar{N}$ indicates the normal direction.

In general, the zero pressure gradient boundary condition at a solid surface can be enforced given the direction cosines $(\gamma_x, \gamma_y, \gamma_z)$ of the normal vector on the surface. Then

$$\left.\frac{\partial P}{\partial \bar{N}}\right|_W = (\gamma_x \gamma_y \gamma_z) \begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} & \frac{\partial \zeta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} & \frac{\partial \zeta}{\partial y} \\ \frac{\partial \xi}{\partial z} & \frac{\partial \eta}{\partial z} & \frac{\partial \zeta}{\partial z} \end{pmatrix} \begin{pmatrix} \frac{\partial P}{\partial \xi} \\ \frac{\partial P}{\partial \eta} \\ \frac{\partial P}{\partial \zeta} \end{pmatrix}_W = 0. \qquad (2.19)$$

2.5.1 Boundary Conditions for Supersonic Flow About Wedge-Cylinder Corners

For wedge-cylinder geometries the transformation between the computational domain and the physical domain is given by Equation (2.17). The upstream boundary conditions at $\xi = 0$ are the free stream conditions. Solid walls occur at $\eta = 0$ and $\zeta = 0$ for $\xi > 0$. For $\eta = 0$ the condition $\frac{\partial P}{\partial N} = 0 \rightarrow \frac{\partial P}{\partial r} = 0$ where r is the radial direction from the center line of the cylinder.

$$\left.\frac{\partial P}{\partial r}\right|_{\eta=0} = \frac{\partial P}{\partial y}\frac{\partial y}{\partial r} + \frac{\partial P}{\partial z}\frac{\partial z}{\partial r} + \frac{\partial P}{\partial x}\frac{\partial x}{\partial r} = 0.$$

From Equation (2.17)

$$\frac{\partial y}{\partial r} = \cos(\overline{\zeta}\theta_2 + (1 - \overline{\zeta})\theta_1) = \cos\hat{\theta}$$

$$\frac{\partial z}{\partial r} = \sin(\overline{\zeta}\theta_2 + (1 - \overline{\zeta})\theta_1) = \sin\hat{\theta}$$

$$\frac{\partial x}{\partial r} = 0.$$

Therefore,

$$\left.\frac{\partial P}{\partial r}\right|_{\eta=0} = \frac{\partial P}{\partial y}\cos\hat{\theta} + \frac{\partial P}{\partial z}\sin\hat{\theta}$$

and

$$\left.\frac{\partial P}{\partial r}\right|_{\eta=0} = \left(\frac{\partial P}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial P}{\partial \eta}\frac{\partial \eta}{\partial y} + \frac{\partial P}{\partial \zeta}\frac{\partial \zeta}{\partial y}\right)\cos\hat{\theta}$$

$$+ \left(\frac{\partial P}{\partial \xi}\frac{\partial \xi}{\partial z} + \frac{\partial P}{\partial \eta}\frac{\partial \eta}{\partial z} + \frac{\partial P}{\partial \zeta}\frac{\partial \zeta}{\partial z}\right)\sin\hat{\theta}$$

where

$$\begin{bmatrix} \frac{\partial\xi}{\partial y} & \frac{\partial\xi}{\partial z} \\ \frac{\partial\eta}{\partial y} & \frac{\partial\eta}{\partial z} \\ \frac{\partial\zeta}{\partial y} & \frac{\partial\zeta}{\partial z} \end{bmatrix} \equiv \text{elements of the Jacobian matrix at } \eta = 0.$$

Since $\frac{\partial\xi}{\partial y} = \frac{\partial\xi}{\partial z} = 0$,

$$\left.\frac{\partial P}{\partial r}\right|_{\eta=0} = \left(\frac{\partial P}{\partial\eta}\frac{\partial\eta}{\partial y} + \frac{\partial P}{\partial\zeta}\frac{\partial\zeta}{\partial y}\right)\cos\hat{\theta} + \left(\frac{\partial P}{\partial\eta}\frac{\partial\eta}{\partial z} + \frac{\partial P}{\partial\zeta}\frac{\partial\zeta}{\partial z}\right)\sin\hat{\theta} = 0.$$

$\frac{\partial P}{\partial\eta}$ is approximated by the one-sided difference

$$\frac{\partial P}{\partial\eta} = \frac{3P_{1,k} - 4P_{2,k} + P_{3,k}}{2\Delta\eta}$$

and, $\frac{\partial P}{\partial\zeta}$ approximated by the central difference

$$\frac{\partial P}{\partial\zeta} = \frac{P_{2,k+1} - P_{2,k-1}}{2\Delta\zeta} .$$

Then

$$\left.\frac{\partial P}{\partial r}\right|_{\eta=0} = \left(\frac{3P_{1,k} - 4P_{2,k} + P_{3,k}}{2\Delta\eta}\right) C_1 + \left(\frac{P_{2,k+1} - P_{2,k-1}}{2\Delta\zeta}\right) C_2 = 0,$$

where

$$C_1 = \frac{\partial\eta}{\partial y}\cos\hat{\theta} + \frac{\partial\eta}{\partial z}\sin\hat{\theta}$$

$$C_2 = \frac{\partial\zeta}{\partial y}\cos\hat{\theta} + \frac{\partial\zeta}{\partial z}\sin\hat{\theta}.$$

Consequently, the pressure on the boundary $\eta = 0$ is approximated by

$$P_{1,k} = \frac{4P_{2,k} - P_{3,k}}{3} - \frac{\Delta\eta}{3\Delta\zeta} \frac{C_2}{C_1} (P_{2,k+1} - P_{2,k-1}).$$

The normal pressure gradient boundary condition $\left.\frac{\partial P}{\partial \bar{N}}\right|_{\zeta=0} = 0$ for the wedge surface of a wedge cylinder corner is

$$\frac{\partial P}{\partial N} = \frac{\partial P}{\partial \xi} (-\sin\phi \frac{\partial \xi}{\partial x}) + \frac{\partial P}{\partial \eta} (\frac{\partial \eta}{\partial x} (-\sin\phi) + \frac{\partial \eta}{\partial z} \cos\phi)$$

$$+ \frac{\partial P}{\partial \zeta} (\frac{\partial \zeta}{\partial x} (-\sin\phi) + \frac{\partial \zeta}{\partial z} \cos\phi) = 0$$

where the directional cosines are: $\gamma_x = \sin\phi$, $\gamma_y = 0$, $\gamma_z = \cos\phi$. The finite difference approximation for P at $\zeta = 0$ is

$$P_{i,j,1} = \frac{4P_{i,j,2} - P_{i,j,3}}{3} - \frac{\Delta\zeta}{3\Delta\eta} \frac{D_2}{D_1} (P_{i,j+1,2} - P_{i,j-1,2})$$

$$- \frac{2\Delta\zeta}{3\Delta\xi} \frac{D_3}{D_1} (P_{i,j,2} - P_{i-1,j,2})$$

where,

$$D_1 = \frac{\partial \zeta}{\partial x} (-\sin\phi) + \frac{\partial \zeta}{\partial z} \cos\phi,$$

$$D_2 = \frac{\partial \eta}{\partial x} (-\sin\phi) + \frac{\partial \eta}{\partial z} \cos\phi,$$

and

$$D_3 = -\sin\phi \frac{\partial \xi}{\partial x} .$$

The far field boundary conditions $(\xi = 1,\ \eta = 1,\ \zeta = 1)$ are

$\left.\frac{\partial U}{\partial \xi}\right|_{\xi=1} = 0$, $\left.\frac{\partial U}{\partial \eta}\right|_{\eta=1} = 0$, and $\left.\frac{\partial U}{\partial \zeta}\right|_{\zeta=1} = 0$. The condition $\frac{\partial U}{\partial \xi} = 0$ implies that there is no change in the state variables with a change in ξ. The approximation is $U_i = U_{i-1}$. The condition $\frac{\partial U}{\partial y} = 0$ implies that there is no change in the state variables with a change in y. For the wedge-cylinder corners, this implies two-dimensional flow on a flat plate and/or inclined plate. That is:

$$\frac{\partial U}{\partial y} = \frac{\partial U}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial U}{\partial \eta}\frac{\partial \eta}{\partial y} + \frac{\partial U}{\partial \zeta}\frac{\partial \zeta}{\partial y} = 0.$$

Noting $\frac{\partial \xi}{\partial y} = 0$ and applying the finite difference approximation

$$\frac{\partial U}{\partial y} = \left(\frac{U_{j,k} - U_{j-1,k}}{\Delta\eta}\right)\frac{\partial \eta}{\partial y} + \left(\frac{U_{j-1,k+1} - U_{j-1,k-1}}{2\Delta\zeta}\right)\frac{\partial \zeta}{\partial y} = 0$$

implies

$$U_{j,k} = U_{j-1,k} + \frac{\Delta\eta}{2\Delta\zeta}\,\frac{\frac{\partial \zeta}{\partial y}}{\frac{\partial \eta}{\partial y}}\,(U_{j-1,k+1} - U_{j-1,k-1}).$$

The condition $\frac{\partial U}{\partial \zeta} = 0$ implies that there is no change in the state variables at $\zeta = 1$ or that symmetry is imposed. In either case $\zeta_k = \zeta_{k-1}$ for this study.

2.5.2 Boundary Conditions for Supersonic Flow About Spike-Nosed Bodies

The transformation between a computational domain and the physical domain for spike-nosed bodies is given by Equation (2.18). In this case the solid surface is at $\eta = 0$, $0 \leq \zeta \leq 1$, and $0 \leq \xi \leq 1$. No slip conditions are imposed on the velocity and the temperature is fixed. The pressure is found in a similar manner to that used for the wedge-cylinder geometries except a less accurate approximation to the zero pressure gradient is used. In this case a first order approximation is used and the pressure on the solid boundary is set equal to the pressure at the grid point next to the boundary. That is:

$$P_{i,1,k} = P_{i,2,k}.$$

The density at the boundary is found by the application of the equation of state using the boundary temperature and computed pressure.

At $\zeta = 0$, $0 \leq \eta \leq 1$, and $0 \leq \xi \leq 1$ a symmetry boundary condition is imposed. The three-dimensional grid is obtained by rotating a two-dimensional slice of the grid about the axis of symmetry. The line $\zeta = 0$, $0 \leq \eta \leq 1$, and $0 \leq \xi \leq 1$ is coincident with the line of symmetry. The Jacobian matrix (Eq. (2.7)) is singular along this line. This does not create a problem, however, since the condition is imposed without using the transformation data from this line. The symmetry condition is

$$\left.\frac{\partial U}{\partial \zeta}\right|_{\zeta=0} = 0 \quad \text{or} \quad U_{i,j,1} = U_{i,j,2}.$$

Free stream boundary conditions are imposed at $\eta = 1$, $0 \leq \zeta \leq 1$, and $0 \leq \xi \leq 1$. At $\zeta = 1$, $0 \leq \eta \leq 1$, and $0 \leq \xi \leq 1$ a no-change boundary condition is imposed. That is:

$$\left.\frac{\partial U}{\partial \zeta}\right|_{\zeta=1} = 0 \quad \text{or} \quad U_{i,j,N} = U_{i,j,N-1}.$$

3. COMPUTATIONAL ASPECTS

The computational requirements for computing a grid and the Jacobian matrix associated with a grid using the "two-boundary technique" are relatively minimal. It is however, necessary to plot the grid to visually assure that the desired constraints are satisfied. Ultimately, this phase of problem solving should be in an interactive mode with high bandwidth communications between the computer and a graphics terminal. The transformation data for an acceptable grid can be stored on a permanent file for later use. An alternate approach is to program the equations for the Jacobian matrix within a program for the solution of a flow-field. In this manner only parameters for the grid generation must be supplied.

The computational requirements for the solution of the three-dimensional Navier-Stokes equations are extreme and tax the capability of any presently existing computer [1]. The approach taken in this study is to adapt a viable numerical technique to the available large scale computer. The computer is the STAR-100 or its successor the CYBER 203. The computer architecture is based on vector processing with virtual memory storage. In addition to the vector architecture there are two aspects relative to the computer that have been very important in this study: (1) the capability of halfword arithmetic and storage; and (2) the effect of data transfer between secondary memory and primary memory. Halfword arithmetic has been used almost exclusively in the computations discussed later allowing for much larger grids than would be otherwise possible. Frequent transfers of data from primary memory to secondary memory and back have been minimized or avoided by constraining the grid size on which a solution is attempted. The transfer of data to and from secondary memory is relatively inefficient and is discouraged by high cost to the user.

Another computational aspect is that the "Navier-Stokes solver" is relatively general. The application of initial and boundary conditions are performed in separate subroutines from the general solution procedure. Defining a new problem by initial and boundary condition does not require major programing modifications.

In this chapter the MacCormack time-split algorithm is examined and vectorized for the CYBER 203 computer. Also, the program organization and how it relates to the virtual memory is presented.

3.1 Computational Technique

The computational technique used in this investigation is the MacCormack time-split predictor-corrector algorithm [2-4] which was proposed about 1970 and is a derivative of the MacCormack unsplit predictor corrector algorithm [26]. Both techniques are explicit which implies that they are time step stability limited [27]. Also, both techniques are second order accurate, and many investigators have been highly successful in applying them to a variety of fluid flow simulations [1,3,4,28]. An advantage of the MacCormack techniques is that they are relatively easy to apply to the transformed equations of motion (Eq. (2.3)). The split operator algorithm has the added advantage that different time step magnitudes can be used in each operator. A third hybrid scheme [29] can be applied by subdividing the operators into implicit and explicit portions. This approach, however, is more complex and the success of its use is somewhat case dependent [30-31]. The explicit time-split technique has been chosen for this investigation because of its simplicity and vectorization characteristics for application on the CYBER 203 computer, however, both the unsplit and split algorithms are presented herein for contrast and clarity.

3.1.1 MacCormack Technique

The unsplit algorithm has the following two steps applied to the transformed equations of motion (Eq. (2.3)).

Predictor step:

$$\overline{U}^{n+1}_{i,j,k} = U^{n}_{i,j,k}$$

$$- \frac{\Delta t}{\Delta\xi} \left[(F_i - F_{i-1}) \frac{\partial\xi}{\partial x} i + (G_i - G_{i-1}) \frac{\partial\xi}{\partial y} i + (H_i - H_{i-1}) \frac{\partial\xi}{\partial z} i \right]_{j,k}$$

$$- \frac{\Delta t}{\Delta\eta} \left[(F_j - F_{j-1}) \frac{\partial\eta}{\partial x} j + (G_j - G_{j-1}) \frac{\partial\eta}{\partial y} j + (H_h - H_{j-1}) \frac{\partial\eta}{\partial z} j \right]_{i,k}$$

$$- \frac{\Delta t}{\Delta\zeta} \left[(F_k - F_{k-1}) \frac{\partial\zeta}{\partial x} k + (G_k - G_{k-1}) \frac{\partial\zeta}{\partial y} k + (H_k - H_{k-1}) \frac{\partial\zeta}{\partial z} k \right]_{i,j} ;$$

Corrector step:

$$U^{n+1}_{i,j,k} = \frac{1}{2}\left(U^{n}_{i,j,k} + \overline{U}^{n+1}_{i,j,k}\right.$$

$$- \frac{\Delta t}{\Delta \xi}\left[(F_{i+1} - F_{i})\,\frac{\partial \xi}{\partial x}\,i + (G_{i+1} - G_{i})\,\frac{\partial \xi}{\partial y}\,i + (H_{i+1} - H_{i})\,\frac{\partial \xi}{\partial z}\,i\right]_{j,k}$$

$$- \frac{\Delta t}{\Delta \eta}\left[(F_{j+1} - F_{j})\,\frac{\partial \eta}{\partial x}\,j + (G_{j+1} - G_{j})\,\frac{\partial \eta}{\partial y}\,j + (H_{j+1} - H_{j})\,\frac{\partial \eta}{\partial z}\,j\right]_{i,k}$$

$$\left.- \frac{\Delta t}{\Delta \zeta}\left[(F_{k+1} - F_{k})\,\frac{\partial \zeta}{\partial x}\,k + (G_{k+1} - G_{k})\,\frac{\partial \zeta}{\partial y}\,k + (H_{k+1} - H_{k})\,\frac{\partial \zeta}{\partial z}\,k\right]_{i,j}\right) .$$

This algorithm is applied for a time step by passing through a data base consisting of the state variables (ρ, ρu, ρv, ρw, ρe) and the transformation data and applying the predictor step. The corrector step is applied with the output of the predictor step, and the old state variables in the data base are replaced with the new values. The algorithm is repeated until a steady state solution is reached or an otherwise chosen stopping point is reached.

3.1.2 MacCormack Time-Split Technique

The split algorithm consists of a predictor and corrector step for each coordinate direction. Consequently, a predictor and corrector step for a coordinate direction is called an operator for that direction (i.e., $L_{\text{direction}}$ (time step)). A time step is completed in this algorithm with the application of each operator applied symmetrically about the operator for the coordinate direction of primary flow. That is, for the corner flows studied herein

$$U^{n+1}_{i,j,k} = \left[L_\eta(\Delta t_\eta)\right] \left[L_\zeta(\Delta t_\zeta)\right] \left[L_\xi(\Delta t_\xi)\right] \left[L_\zeta(\Delta t_\zeta)\right] \left[L_\eta(\Delta t_\eta)\right] U^n_{i,j,k}$$

where

$$\Delta t_\eta = \Delta t_\zeta = \frac{1}{2}\Delta t_\xi .$$

For the spike-nosed body flow

$$U^{n+1}_{i,j,k} = \left[L_\xi(\Delta t_\xi)\right] \left[L_\eta(\Delta t_\eta)\right] \left[L_\zeta(\Delta t_\zeta\right] \left[L_\eta(\Delta t_\zeta)\right] \left[L_\xi(\Delta t_\xi)\right] U^n_{i,j,k}$$

where

$$\Delta t_\xi = \Delta t_\eta = \frac{1}{2}\,\Delta t_\zeta .$$

Each operator is defined by an output state solution U^{out} for a given input state solution U^{in}. Therefore,

$$L_\xi(\Delta t_\xi) = U^{out}_{i,j,k}$$

where

Predictor step:

$$\bar{U}_{i,j,k} = U^{in}_{i,j,k} - \frac{\Delta t_\xi}{\Delta\xi}\left[(F_i - F_{i-1})\,\frac{\Delta\xi}{\partial x}\,i + (G_i - G_{i-1})\,\frac{\partial\xi}{\partial y}\,i\right.$$

$$\left. + (H_i - H_{i-1})\,\frac{\partial\xi}{\partial z}\,i\right]_{j,k};$$

Corrector step:

$$U^{out}_{i,j,k} = \frac{1}{2}\left(U^{in}_{i,j,k} + \overline{U}_{i,j,k} - \frac{\Delta t_\xi}{\Delta\xi}\left[(F_{i+1} - F_i)\,\frac{\partial\xi}{\partial x}\,i\right.\right.$$

$$\left.\left. + (G_{i+1} - G_i)\,\frac{\partial\xi}{\partial y}\,i + (H_i - H_{i-1})\,\frac{\partial\xi}{\partial z}\,i\right]_{j,k}\right) \quad ,$$

$$L_\eta(\Delta t_\eta) = U^{out}_{i,j,k}$$

where

Predictor step:

$$\overline{U}_{i,j,k} = U^{in}_{i,j,k} - \frac{\Delta t_\eta}{\Delta\eta}\left[(F_j - F_{j-1})\,\frac{\partial\eta}{\partial x}\,j + (G_j - G_{j-1})\,\frac{\partial\eta}{\partial y}\,j\right.$$

$$\left. + (H_j - H_{j-1})\,\frac{\partial\eta}{\partial z}\,j\right]_{i,k} \quad ;$$

Corrector step:

$$U^{out}_{i,j,k} = \frac{1}{2} \left(U^{in}_{i,j,k} + \overline{U}_{i,j,k} - \frac{\Delta t_\eta}{\Delta \eta} \left[(F_{j+1} - F_j) \frac{\partial \eta}{\partial x} j \right.\right.$$

$$\left.\left. + (G_{j+1} - G_j) \frac{\partial \eta}{\partial y} j + (H_{j+1} - H_j) \frac{\partial \eta}{\partial z} j \right]_{i,k} \right) ,$$

$$L_\zeta(\Delta t_\zeta) = U^{out}_{i,j,k}$$

where

Predictor step:

$$\overline{U}_{i,j,k} = U^{in}_{i,j,k} - \frac{\Delta t_\zeta}{\Delta \zeta} \left[(F_k - F_{k-1}) \frac{\partial \zeta}{\partial x} k \right.$$

$$\left. + (G_k - G_{k-1}) \frac{\partial \zeta}{\partial y} k \; (H_k - H_{k-1}) \frac{\partial \zeta}{\partial z} k \right]_{i,j} ;$$

<u>Corrector step</u>:

$$U^{out}_{i,j,k} = \frac{1}{2}\left(U^{in}_{i,j,k} + \bar{U}_{i,j,k} - \frac{\Delta t_\zeta}{\Delta\zeta}\left[(F_{k+1} - F_k)\frac{\partial\zeta}{\partial x}k\right.\right.$$

$$\left.\left. + (G_{k+1} - G_k)\frac{\partial\zeta}{\partial y}k + (H_{k+1} - H_k)\frac{\partial\zeta}{\partial z}k\right]_{i,j}\right).$$

The unsplit algorithm requires only one pass through the data base per time step while the split algorithm requires several passes through the data base per time step. When the data base exceeds the primary storage capacity of the CYBER 203 computer a time penalty is imposed when data is called from secondary memory. However, a data management procedure has been implemented to minimize the penalties associated with the use of secondary memory.

It has been noted that the MacCormack algorithms are second order accurate. Forward and backward differences are applied such that after the predictor and corrector steps are completed an effective central difference approximation is obtained [3]. This is demonstrated with derivatives of velocity components required in the viscous stress terms. Consider

$$\frac{\partial u}{\partial\xi},\ \frac{\partial u}{\partial\eta}\ \text{and}\ \frac{\partial u}{\partial\zeta}$$

For a predictor step:

L_ξ

$$\frac{\partial u}{\partial \xi} = \left(\frac{u_{i+1} - u_i}{\Delta\xi}\right)_{j,k}, \quad \frac{\partial u}{\partial \eta} = \left(\frac{u_{j+1} - u_{j-1}}{\partial\Delta\eta}\right)_{i,j}, \quad \frac{\partial u}{\partial \zeta} = \left(\frac{u_{k+1} - u_{k-1}}{\partial\Delta\zeta}\right)_{i,j};$$

L_η

$$\frac{\partial u}{\partial \eta} = \left(\frac{u_{j+1} - u_j}{\Delta\eta}\right)_{i,k}, \quad \frac{\partial u}{\partial \xi} = \left(\frac{u_{i+1} - u_{i-1}}{2\Delta\xi}\right)_{j,k}, \quad \frac{\partial u}{\partial \xi} = \left(\frac{u_{k+1} - u_{k-1}}{2\Delta\zeta}\right)_{i,j};$$

L_ζ

$$\frac{\partial u}{\partial \zeta} = \left(\frac{u_{k+1} - u_k}{\Delta\zeta}\right)_{i,j}, \quad \frac{\partial u}{\partial \xi} = \left(\frac{u_{i+1} - u_{i-1}}{2\Delta\xi}\right)_{j,k}, \quad \frac{\partial u}{\partial \eta} = \left(\frac{u_{j+1} - u_{j-1}}{2\Delta\eta}\right)_{i,j}.$$

For a corrector step:

L_ξ

$$\frac{\partial u}{\partial \xi} = \left(\frac{u_i - u_{i-1}}{\Delta\xi}\right)_{j,k}, \quad \frac{\partial u}{\partial \eta} = \left(\frac{u_{j+1} - u_{j-1}}{2\Delta\eta}\right)_{i,k}, \quad \frac{\partial u}{\partial \zeta} = \left(\frac{u_{k+1} - u_{k-1}}{2\Delta\zeta}\right)_{i,j};$$

L_η

$$\frac{\partial u}{\partial \eta} = \left(\frac{u_j - u_{j-1}}{\Delta\eta}\right)_{i,k}, \quad \frac{\partial u}{\partial \xi} = \left(\frac{u_{i+1} - u_{i-1}}{2\Delta\xi}\right)_{j,k}, \quad \frac{\partial u}{\partial \zeta} = \left(\frac{u_{k+1} - u_{k-1}}{2\Delta\zeta}\right)_{i,j};$$

L_ζ

$$\frac{\partial u}{\partial \zeta} = \left(\frac{u_k - u_{k-1}}{\Delta\zeta}\right)_{i,j}, \quad \frac{\partial u}{\partial \xi} = \left(\frac{u_{i+1} - u_{i-1}}{2\Delta\xi}\right)_{j,k}, \quad \frac{\partial u}{\partial \eta} = \left(\frac{u_{j+1} - u_{j-1}}{2\Delta\eta}\right)_{i,k}.$$

The unsplit and split MacCormack algorithms are time step stability limited, and there is no complete stability analysis to indicate the maximum allowable time step. A conservative time step employed by Shang and Hankey [3-4] has been used. This time step is

$$\Delta t \leq \min \left[\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z} + c\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}} \right]^{-1}$$

where

$c \equiv$ local speed of sound,

A point that must be considered in the application of the MacCormack algorithms to viscous compressible flow with strong shock waves is the inclusion of terms to dampen oscillations in the region of a shock. A pressure dampening term suggested by MacCormack [31] is included in the finite difference approximation to the equations of motion. This term is

$$-\bar{\alpha}_\ell \Delta t_\ell \delta_\ell^3 \; \frac{\partial}{\partial \delta_\ell} \left[\frac{|v_\ell| + c}{4P} \; \frac{\partial^2 P}{\partial \delta_\ell^2} \right] \frac{\partial U}{\partial \delta_\ell} \qquad \ell = 1,2,3$$

where

$\delta_1 = \xi, \qquad \delta_2 = \eta, \;$ and $\; \delta_3 = \zeta.$

3.2 Application of Vector Processing to the Computational Technique

The MacCormack time-split algorithm has been programmed to run on the CDC STAR-100 and CYBER 203 computers. The program called the Navier-Stokes solver was first written in STAR FORTRAN and is described by Smith and Pitts [32]. The program has since been written in the SL/1 language [20] where 32-bit arithmetic is used to increase the computational speed and incore storage.

3.2.1 Vector Processing Using the CYBER 203 Computer

The CYBER 203 is a vector processing computer capable of achieving high result rates when a high degree of parallelism is present in the computation. When an identical operation is to be performed on consecutive elements in memory, a vector instruction is issued to perform the operation. Each vector instruction involves a time penalty, called vector startup, regardless of the length of the vector. As the length of the vector increases, the operation becomes more efficient since the penalty becomes relatively less important.

The CYBER 203 has about one million words of primary memory with virtual memory architecture. Memory is referred to as pages. The two page sizes on the CYBER 203 are "small" pages which are 512 64-bit words and "large" pages which are 65536 words or 128 small pages. A user can have access to about 15 large pages in primary memory at any one time. The movement of data from secondary memory into primary memory involves moving pages of data in and out of primary memory. This is called a "page fault" and involves a startup time and transmission time just as vector operations do. It then becomes important to make the most efficient use of the data when it is in primary memory in order to avoid a situation where the machine is spending more time moving pages of data in and out of primary memory than it is spending on actual computations. This is often referred to as "thrashing." Storing the data for a large data base program, such as a three-dimensional Navier-Stokes solver, in a conventional manner could very possibly lead to this situation. If, however, you design an interleaved data base [33] where the variables that are currently being used are stored together then it could result in less movement of "pages" of data.

A capability of the CYBER 203 architecture is half-word arithmetic. This means that computation can be performed with 32-bit words and approximately two million 32-bit words can be stored in primary memory. The speed of computation is approximately twice that achieved with 64-bit operations. A high level programming language SL/1 [20] is used to access the half-word capability. It is shown in the next chapter that for the MacCormack time-split algorithm 32-bit arithmetic is quite adequate and the computational speed and primary storage are approximately doubled as compared to FORTRAN version which uses 64-bit words.

3.2.2 Program Organization and Data Management

The Navier-Stokes solver is a derivative of a serial FORTRAN code which operates on the CDC-CYBER 175 and 7600 computers [2-3].

Seven of the nine elements of the Jacobian matrix are programmed into the serial code and some redundant computation is performed to coexist with available memory. The Navier-Stokes solver is written in SL/1 and maintains the MAIN program logic and time step calculations found in the serial code (Fig. 13). The operator calculations are redesigned around the vector architecture of the CYBER 203 computer. Also, all nine elements of the Jacobian matrix are included in the new code.

Normally it may be thought that vector lengths should be equal to the total number of grid points and vector operations sweep through the entire grid with each variable dimensioned to the number of grid points. However, for the number of variables involved and the large number of grid points this leads very quickly to the "thrashing" situation described earlier. Instead, vectors are computed in planes in the ξ directions (Fig. 14) with vector lengths approximately equal to the number of grid points in a plane. Temporary reusable vectors are maintained for three local planes and a four-dimensional array S(I,L,J,K) contains the five state variables and nine elements of the metric coefficient for each grid point.

Forming S(I,L,J,K) is the essence of interleaving the data base. The index L refers to the 14 variables. The index I refers to the plane in the ξ direction and the indices J and K refer to the grid points in a plane. In this manner all variables for each plane are stored in contiguous locations. Computation proceeds from the first plane to the last plane for each operator.

In order to minimize the sweeps through the data base S(I,L,J,K), the corrector step for a plane is performed as soon as enough planes of the predictor step are available (Fig. 15). Consequently for the application of each operator, there is one sweep through the data base and five sweeps for a time step.

Within a ξ plane, a vector sweep is from the lower left hand corner to the upper right hand corner. The exact length, starting point, and end point of the vector is dependent on the operator and the direction of the finite differencing. Vector operations include boundary points where erroneous values are computed during a vector computation. The boundary condition subroutine is called to compute the boundary conditions and overwrite the erroneous values.

The transformation data which consist of the nine derivatives of the computational coordinates with respect to the physical coordinates at each grid point are computed in a separate program and stored on a disk file. Once a geometry is established, the transformation data remains constant. It is read from the disk file by the Navier-Stokes solver and stored in the S array (L = 6...14).

Externally the Navier-Stokes solver operates like the serial program starting with some initial conditions and integrating with the finite-difference algorithm until "steady state" is reached, or the program is stopped. A restart capability is included so that a solution can be obtained in several runs with intermediate observation of the solution.

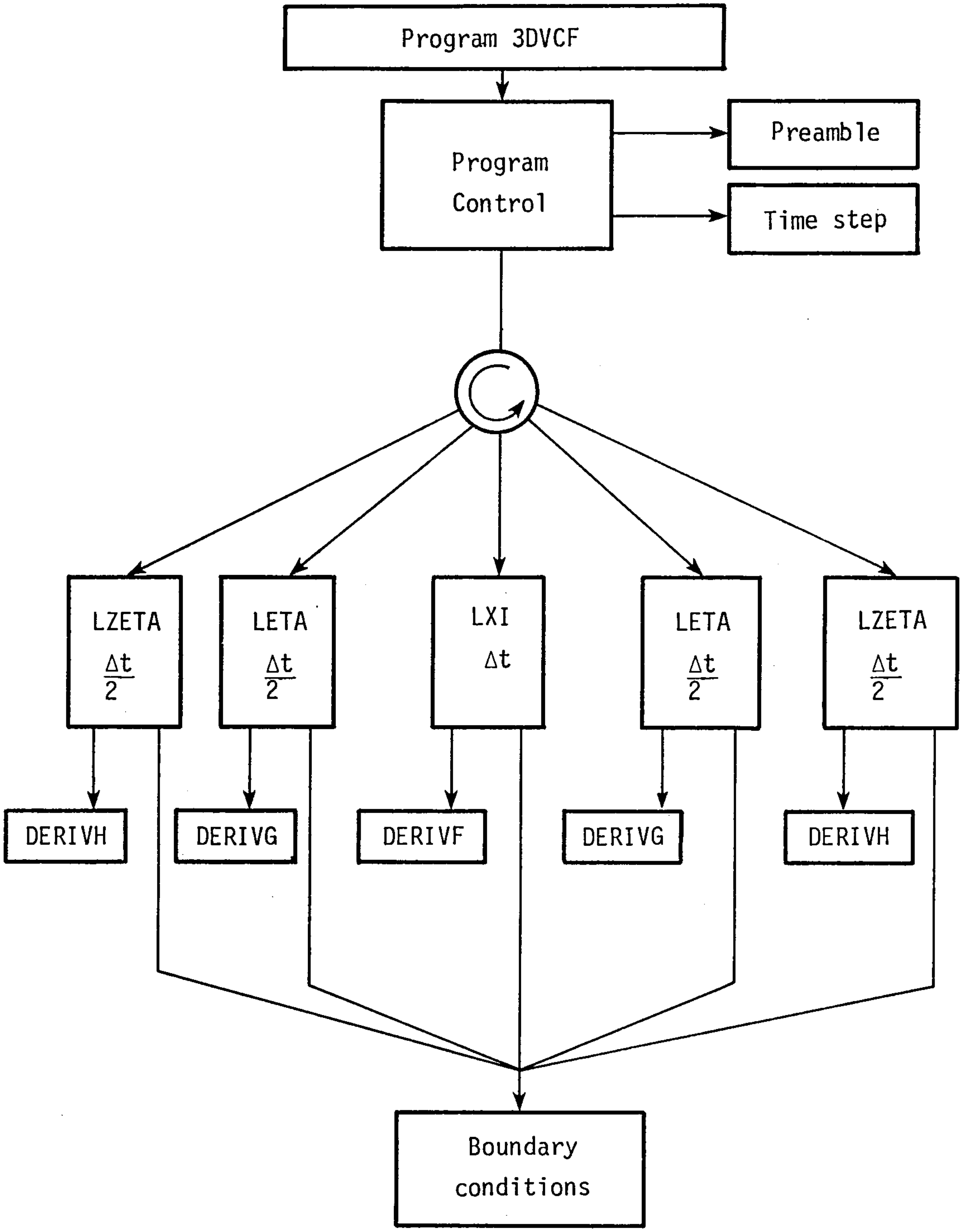


Fig. 13 Flowchart for Navier-Stokes solver.

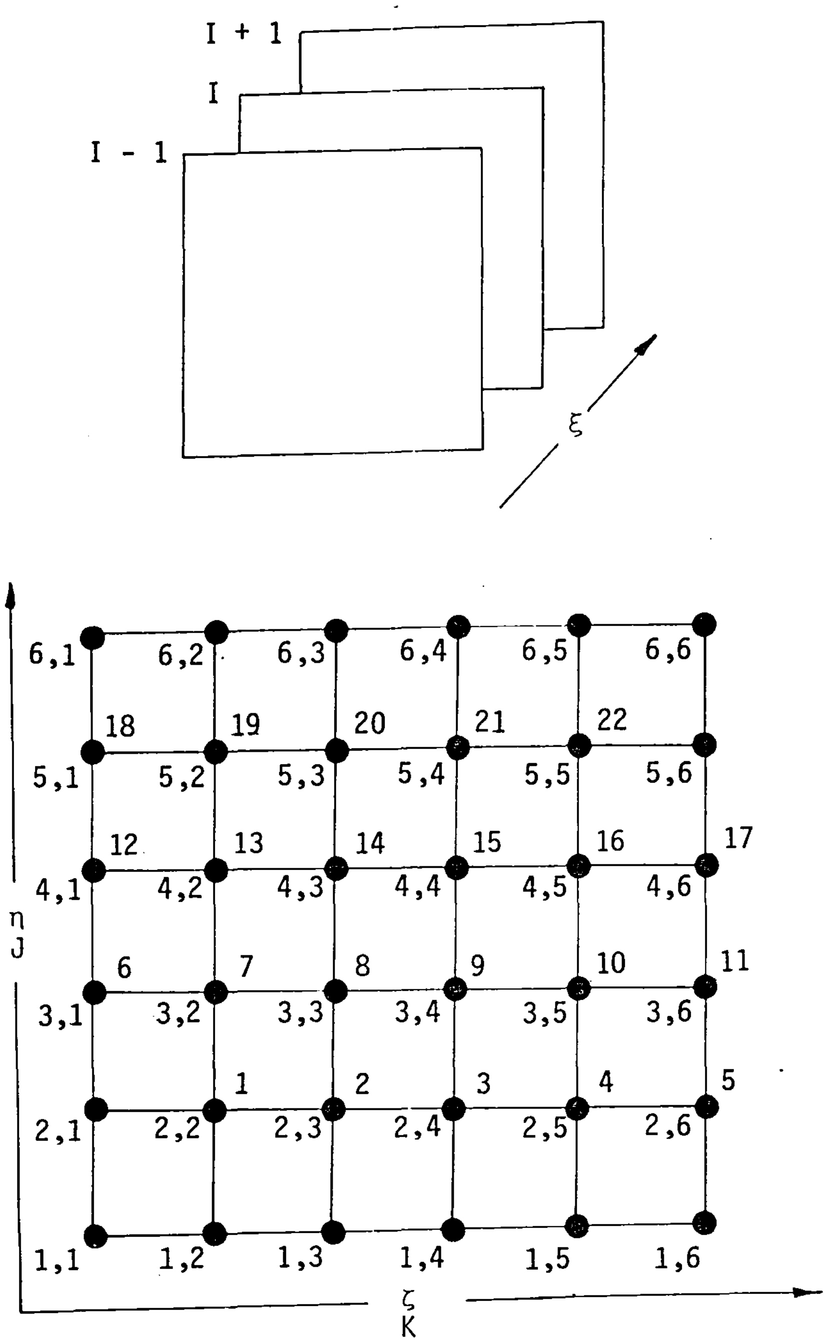


Fig. 14 Vector arrangement in planes.

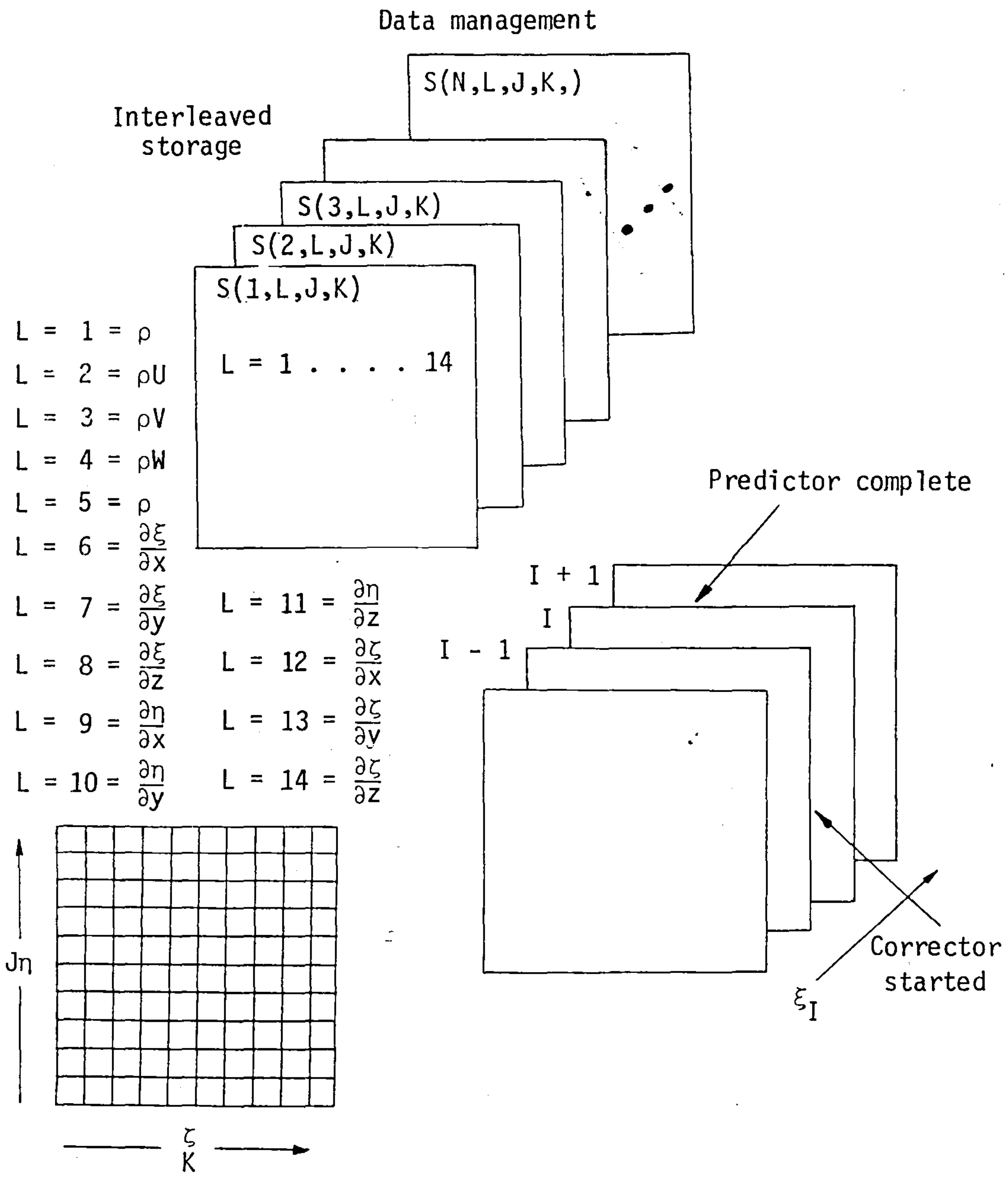

Data management
S(N,L,J,K,)
Interleaved storage
S(3,L,J,K)
S(2,L,J,K)
S(1,L,J,K)
L = 1 14
L = 1 = ρ
L = 2 = ρU
L = 3 = ρV
L = 4 = ρW
L = 5 = ρ
L = 6 = $\frac{\partial \xi}{\partial x}$
L = 7 = $\frac{\partial \xi}{\partial y}$
L = 8 = $\frac{\partial \xi}{\partial z}$
L = 9 = $\frac{\partial \eta}{\partial x}$
L = 10 = $\frac{\partial \eta}{\partial y}$
L = 11 = $\frac{\partial \eta}{\partial z}$
L = 12 = $\frac{\partial \zeta}{\partial x}$
L = 13 = $\frac{\partial \zeta}{\partial y}$
L = 14 = $\frac{\partial \zeta}{\partial z}$
Predictor complete
I + 1
I
I - 1
Corrector started
ξ_I
Jη
ζ
K

Fig. 15 Date management for Navier-Stokes solver.

4. RESULTS AND DISCUSSION

Assuming that the equations of motion (Eq. (2.3)) are valid and that a finite difference technique (MacCormack time-split algorithm) is used to numerically solve them, the physical grid and its relation to the computational grid form the foundation for a solution. This chapter is aimed at establishing the applicability of the "two-boundary technique" for grid generation by obtaining solutions to complex flow fields using transformation data derived from the technique. Also, the robustness of the MacCormack technique and the computational capabilities of the CYBER 203 computer are demonstrated.

The "two-boundary technique" is applied to two distinctly different supersonic flow problems. In Chapter 2, relationships between the computational grid and the physical grids for a family of three-dimensional corners and spike-nosed bodies are derived. The derivatives for the transformation data are also presented. In this chapter the transformation data and boundary conditions for the flows are applied in the Navier-Stokes solver. The three-dimensional corner flow fields are extremely complex with shock-boundary layer interactions and three-dimensional separation. The grids are concentrated near the solid boundaries and in the corner. The flow about the spike-nosed bodies is characterized by a strong bow shock and a highly separated region between the nose and shoulder. The grid is concentrated in this region, and the solutions are unsteady. The two flow problems are considered separately, and the grid characteristics are pointed out.

4.1 Supersonic Corner Flow Using Two-Boundary Grid Generation

Supersonic flow about three-dimensional corners occurs in many high speed aerodynamic situations. Over the years there have been many experiments to study this flow phenomenon [34-38]. More recently there have been numerical experiments to compute supersonic flow about three-dimensional corners. Inviscid compressible solutions about planar three-dimensional corners have been obtained by Kutler, Shanker et al. [39-40] and Marconi [41]. Inviscid solutions, however, do not account for the strong inviscid-viscid interactions which occur. Asymptotic viscous solutions have been obtained by Weinberg and Rubin [42] and Ghia and Davis [43]. These solutions require extensive assumptions about the flow and also do not adequately describe the inviscid-viscid interactions.

The solution of the compressible Navier-Stokes equations is the most conclusive way to compute supersonic flow about three-dimensional corners, and there are several published solutions. Shang and Hankey [3] using a time split MacCormack technique for the solution of the Navier-Stokes equations simulated the flow about an asymmetric 15^o wedge-plate corner at Mach number 12.5. This numerical simulation corresponds to the physical experiment of Cooper and Hankey [38]. For this simulation the largest computational mesh was 8 × 32 × 36. In a later numerical study Shang and Hankey [4] computed a turbulent flow about a symmetric 9.48^o planar corner at Mach number 3.

Hung and MacCormack [44] computed supersonic laminar flow about a 10^o asymmetric planar corner preceded by a rectangular corner. They used a time-split MacCormack technique where the conventional two-step scheme is used in the L_x operator and in the L_y and L_z operators in the regions far from the solid boundaries. In the inner regions the L_y and L_z operators are further split into hyperbolic and parabolic operators for the inviscid and viscous terms. Later Hung and MacCormack [45] extended the code for the accelerated technique to compute turbulent flow, and Horstman and Hung [46] studied several planar turbulent corner flows.

Most published discussion on supersonic flow about three-dimensional corners including that mentioned above has dealt with planar intersecting boundaries. This means that rectangular coordinates or a simple transformation have been used. Herein, both planar intersecting and planar-cylinder intersecting corners are discussed. In Chapter 2, Equation (2.17), derived using the "two-boundary technique," relates a physical domain with wedge-cylinder boundaries to a computational domain. A plane-cylinder corner is formed by letting the wedge angle be zero and planar intersecting corners are approximated by letting the radii be very large. The transformation data obtained by differentiating Equation (2.17) is used in the Navier-Stokes solver along with the boundary conditions discussed in Chapter 2.

During this investigation many solutions with varying Mach number, Reynolds number, and grid size have been obtained. Portions of the following discussion have been presented by the author in [47]. The initial numerical experiment is the solution of the flow about a rectangular corner (Fig. 5a) with a 31 × 31 × 31 grid. The velocity solution u/u_∞ at $x/x_L = .554$ where the flow is two-dimensional is shown in Figure 16 and is compared with that presented by Hung and MacCormack [44]. The Mach number is 3 and the Reynolds number is 2.78×10^5/m (7×10^3/in.). The concentration of the grid based on Equation (2.17a) has a value of $k_1 = k_2 = 3.8$. The agreement of the velocity with that of the reference is good. This initial solution is obtained on the STAR-100 computer with the FORTRAN version of the Navier-Stokes solver. The computational rate is 1.5×10^{-4} seconds per grid point per time step. The remaining corner flow solutions to be discussed have been obtained on the CYBER 203 computer and the SL/1 version of the Navier-Stokes solver. The computational rate for the largest grid used is 4×10^{-5} seconds per grid point per time step. No significant degradation of the solutions using the small 32-bit word size observed.

The next step in this experiment is to obtain the solution of a family of wedge-cylinder and wedge-plate corner flows using a $20 \times 36 \times 36$ grid. There are three planar corners with 0, 6, and 12.2^o wedge angles and three wedge-cylinder corners with the same wedge angles. Equation (2.17) has been used to define the grids and transformation data. The physical dimensions of the domain are shown in Figure 17, and the grid at $x/x_L = 1$ are shown in Figure 7. The concentration parameter in Equation (2.17a) is 3.8. Two additional solutions for a planar 18^o corner have been obtained--one with the

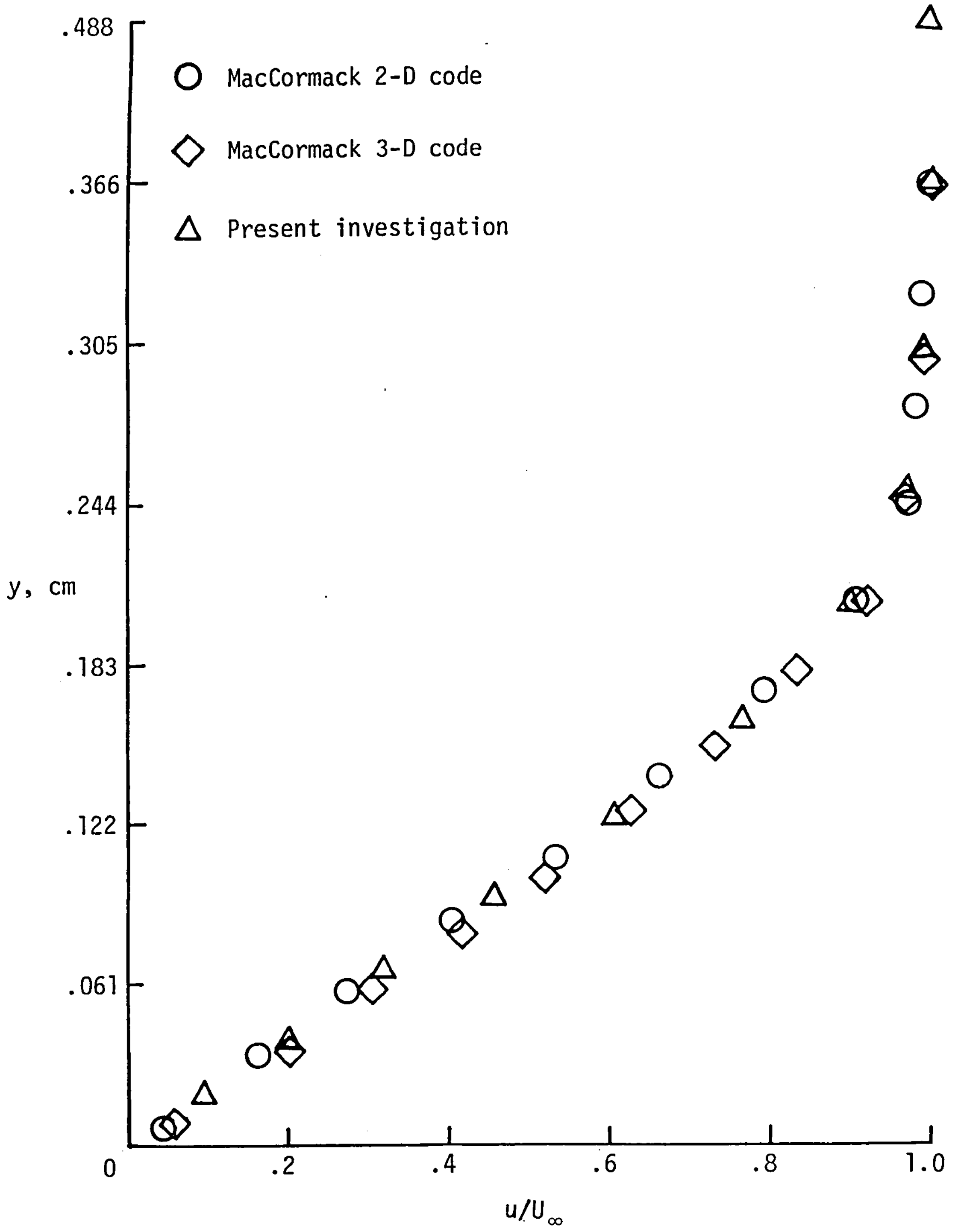


Fig. 16 Two-dimensional solutions.

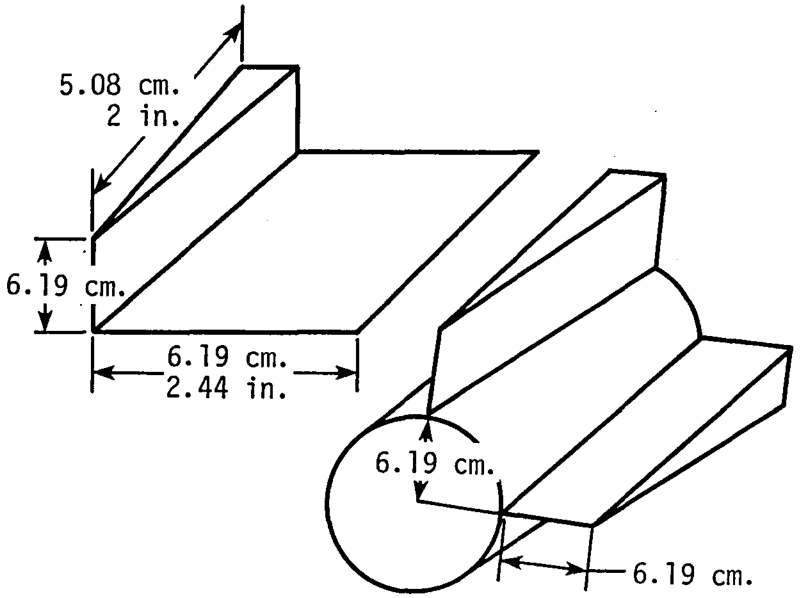


Fig. 17 Physical dimensions of corners.

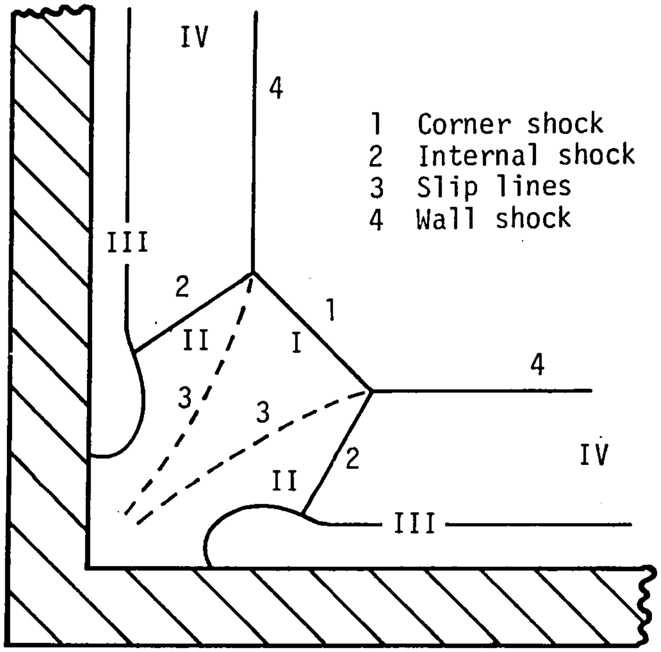


Fig. 18 Three-dimensional corner flow characteristics.

Reynolds number used in all other solutions, and the other with a Reynolds number equal to 3.9×10^6/m (1×10^5/in.). The physical position of the right boundary is moved further to the right because the effects of the shock are too close to the original position. The Mach number is 3.64 and the Reynolds number except for the one 18^o planar wedge corner is 2.92×10^5/m (7.42×10^3/in.). The free stream and body temperatures are respectively 217 K (390 R) and 607 K (1092 R). The fluid properties are for air. Solutions are started impulsively except for the one high Reynolds number solution and are run for approximately eight characteristic times for a steady state solution. For the high Reynolds number case, the initial state is that of the lower Reynolds number solution. A steady state is obtained through several increments until the higher Reynolds number is reached.

Flow visualization is presented in the form of continuous tone density distributions and a combination of continuous tones and contours for the temperature at $x/x_L = 0.83$. These distributions are made for this document from color distributions obtained on the Dicomed Graphics System mentioned in the introduction. Velocity vectors (uxw) at the first grid point above the flat plate and cylinder surface are shown to indicate the direction of flow near the base surface. Velocity vectors (vxw) at $x/x_L = .83$ are shown to indicate the cross-flow velocity field.

The primary observation from this experiment and others is that the flow is basically conical. Also, the wedge-cylinder flow fields for the cylinder radius used are very similar to the planar corners for the same wedge angle. For the 0 and 6^o wedge angle solutions, the zones of flow (Fig. 18) described by Charwat and Redekopp [34] are readily seen (Figs. 19, 20, 21, 22, and 23). That is, for a traverse plane of a symmetric planar corner, there is Zone I (a region of conical flow bounded by slip lines and the corner shock), Zone II (a region of complex flow bounded by slip lines and a strong internal shock), Zone III (an outer interaction region characterized by a compressive fan centered at the triple shock intersection point), and Zone IV (undisturbed wedge flow). Figure 18 depicts the flow situation in a symmetric traverse plane. Charwat and Redekopp [34] also concluded that the flow structure remains qualitatively similar with change in Mach number for symmetric planar corners, and is distorted without losing its identity for asymmetric planar corners. The rectangular corner solution (Fig. 19) shows the zones of flow in a symmetric pattern. The plate-cylinder corner solution (Fig. 22) is very similar to the rectangular corner flow. The base surface curvature modifies the symmetry to some extent, but close to the corner the flow is almost identical. The temperature distributions for the rectangular corner and plate-cylinder are very orderly and follow the zones of flow.

As predicted by Charwat and Redokopp [34] the 6^o wedge corners show the zones of flow somewhat distorted (Figs. 20 and 23). Again the 6^o wedge-cylinder corner solution is very similar to the 6^o planar corner solution. Also, the 6^o corner solutions show two interesting characteristics that are observed in the larger wedge angle solutions but not observed for the 0^o solutions. The crossflow velocity separates on the wedge surface about the internal shock.

The crossflow velocity separation is observed in all the non-zero wedge angle cases computed at Reynolds number equal to 2.92×10^5/m (Figs. 10, 21, 23, 24, and 25). The second characteristic is the high crossflow velocity near the flat plate or cylinder surface under the internal shock. This phenomenon is observed on all the non-zero wedge angle solutions.

Cooper and Hankey [38] performed experiments on a 15^o asymmetric planar corner at Mach 12.5. They observed one triple point instead of the two observed by Charwat and Redekopp [34]. The numberical experiment by Shang and Hankey [3] basically confirmed the one triple point observation. The 12.2 and 18^o wedge angle solutions obtained at the low Reynolds number and described herein appear to show two triple points although highly distorted. The corner shocks are almost vertical. The high Reynolds number solution for the 18^o wedge angle, however, indicates that there is only one triple point. In fact, the high Reynolds number solution is qualitatively very similar to that of Shang and Hankey [3]. The relevance of Reynolds number is discussed further at a later point.

Korkegi [48] has described supersonic flow in three-dimensional corners and associates separation with the shock strength which is a function of the wedge angle. Both laminar and turbulent corner flow display the separation phenomenon, however, a larger shock strength is required to cause separation in a turbulent flow. Korkegi [48] describes three-dimensional separation as a line along which the flow lifts off a continuous solid surface and three-dimensional reattachment as a line of flow impingement on a continuous solid surface. Further, secondary separation can occur when the shock strength is sifficient for the reverse flow region to separate. The convergence and divergence of the uxw velocity vectors close to the surface indicate separation and reattachment. The 0 and 6° wedge angle corner solutions (Figs. 19, 20, 21, and 23) show no strong indication of separation on the plate or cylinder surfaces. The uxw velocity vectors align themselves with the direction of the wedge inside the shock and align themselves somewhat with shock in the shock region. The 12.2° wedge corner solution (Figs. 21 and 24) show evidence of convergence of the uxw velocity vectors or separation, but divergence is not clear.

The 18° wedge corner solution at the lower Reynolds number (Fig. 25) displays both the convergence and divergence of the uxw velocity vectors. Figure 26 shows the solution of the 18° wedge corner at a Reynolds number of 3.9×10^6/m (1×10^5/in.). The plate surface pressure (Fig. 27) shows the trough-like pressure variation described by Korkegi [48] as an indication of secondary separation. The interesting point about the 18° wedge corner is the effect of increasing the Reynolds number. Starting with the steady state solution for the low Reynolds number 18° wedge corner solution, the Reynolds number is gradually increased and a new steady state achieved. The velocity vectors are rescaled since the velocity is much greater in magnitude at the same relative position as the low Reynolds number

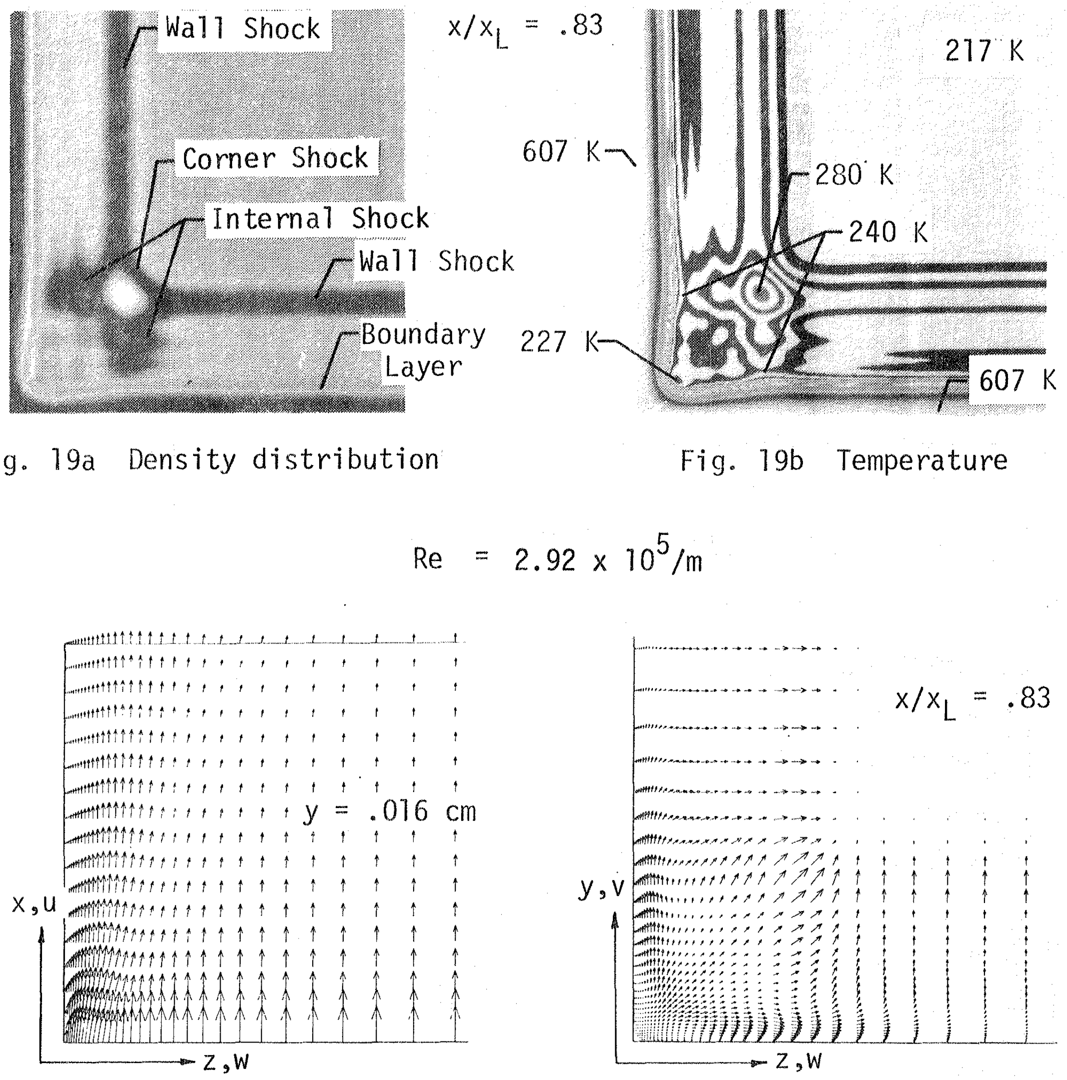


Fig. 19a Density distribution

Fig. 19b Temperature

Fig. 19c u x w Velocity

Fig. 19d v x w Crossflow velocity

Fig. 19 Flow field description for a rectangular corner.

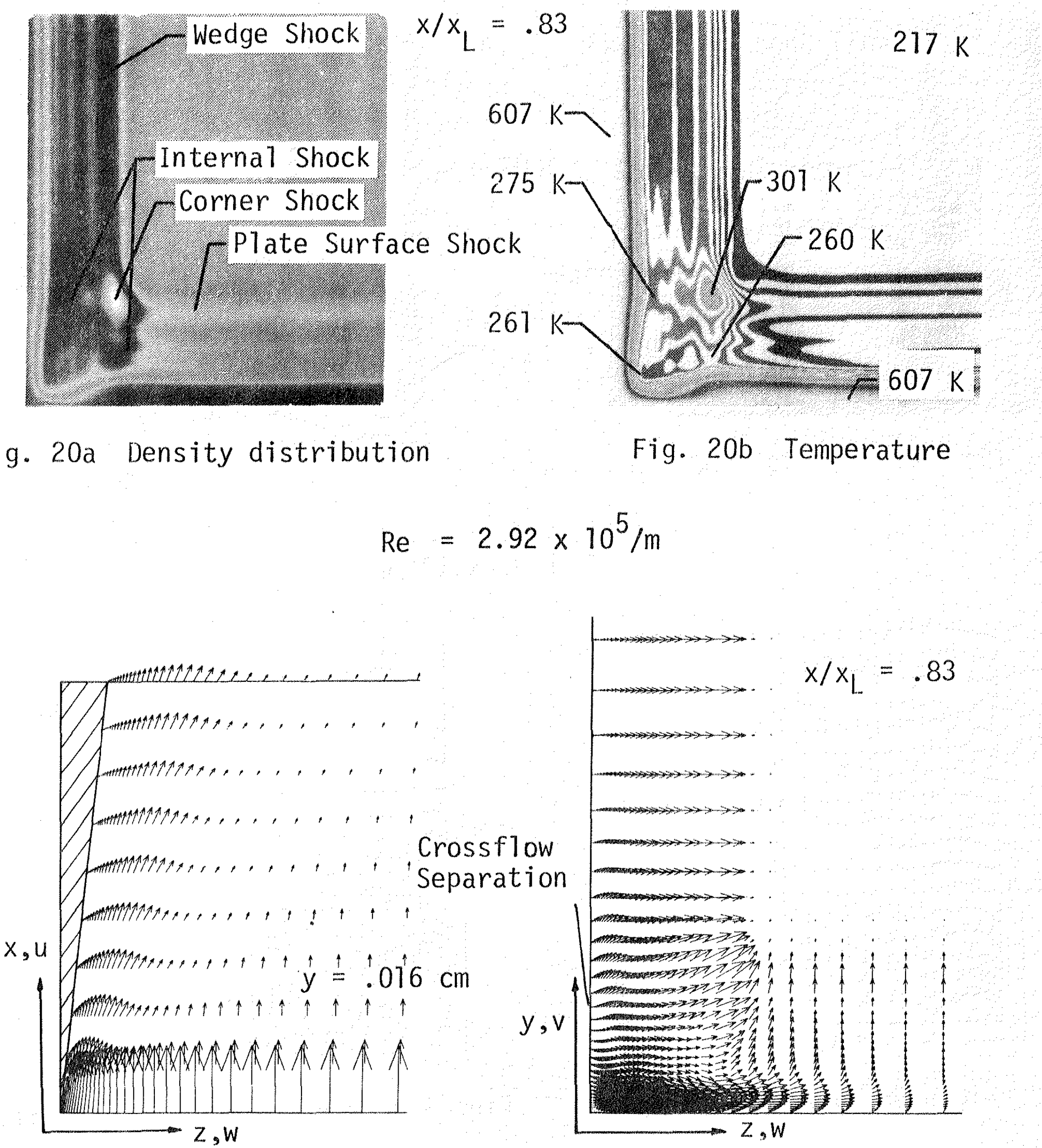


Fig. 20a Density distribution

Fig. 20b Temperature

Fig. 20c u x w Velocity

Fig. 20d v x w Crossflow velocity

Fig. 20 Flow field description for a 6^o wedge-plate corner.

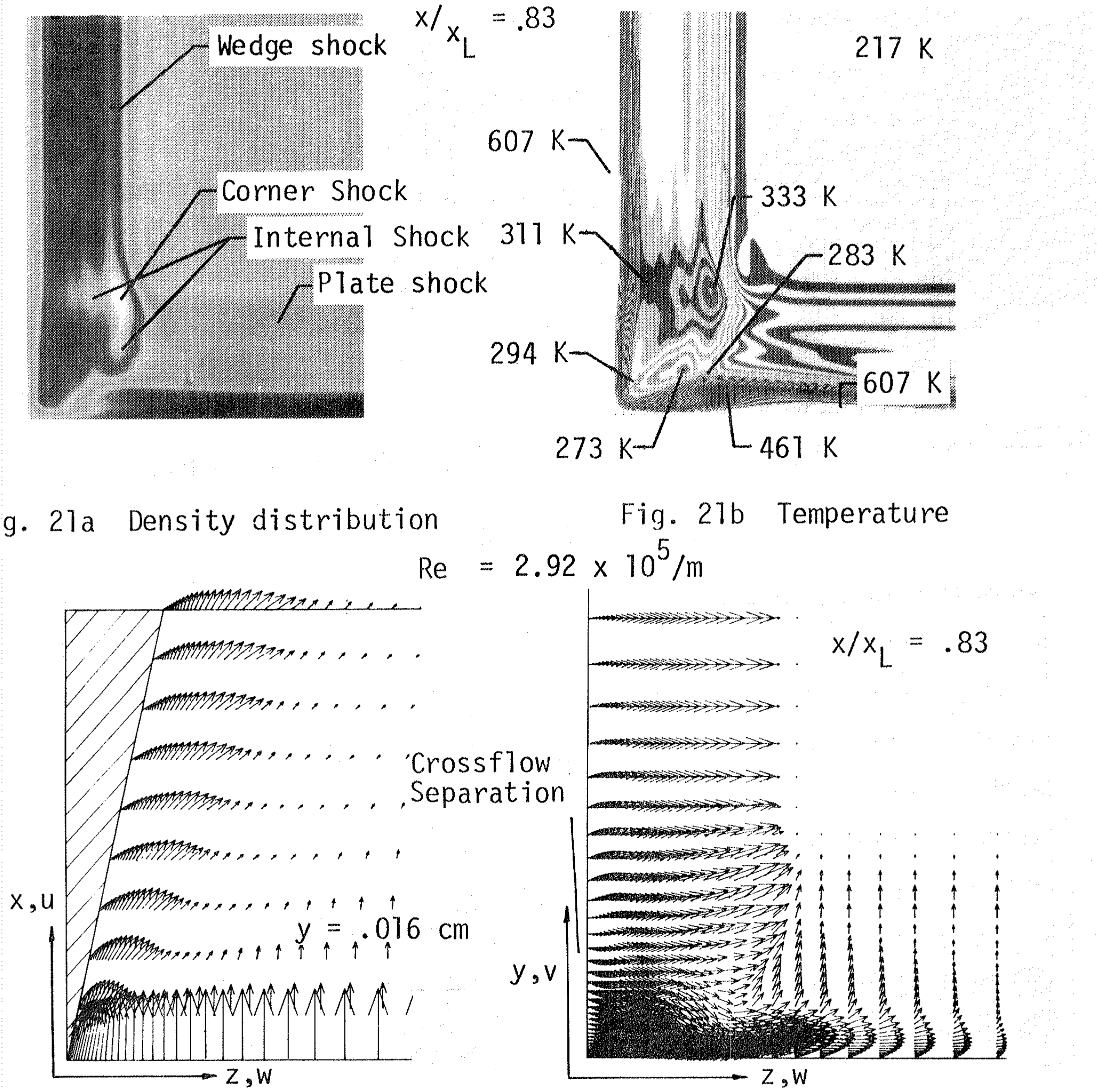


Fig. 21a Density distribution

Fig. 21b Temperature

Fig. 21c u x w Velocity

Fig. 21d v x w Crossflow velocity

Fig. 21 Flow field description for a 12.2^{o} wedge-plate corner.

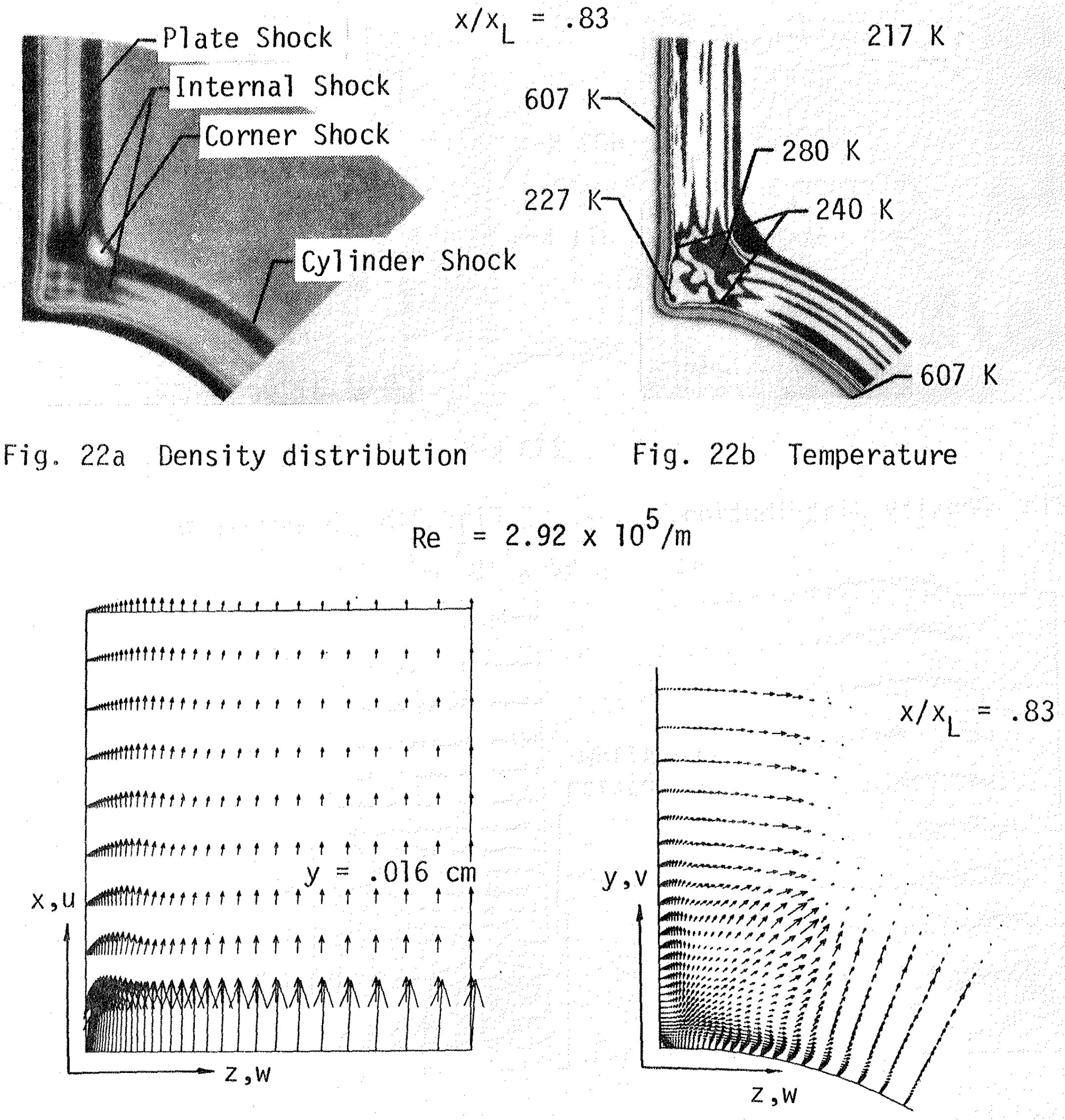


Fig. 22a Density distribution

Fig. 22b Temperature

Fig. 22c u x w Velocity

Fig. 22d v x w Crossflow velocity

Fig. 22 Flow field description for a plate-cylinder corner.

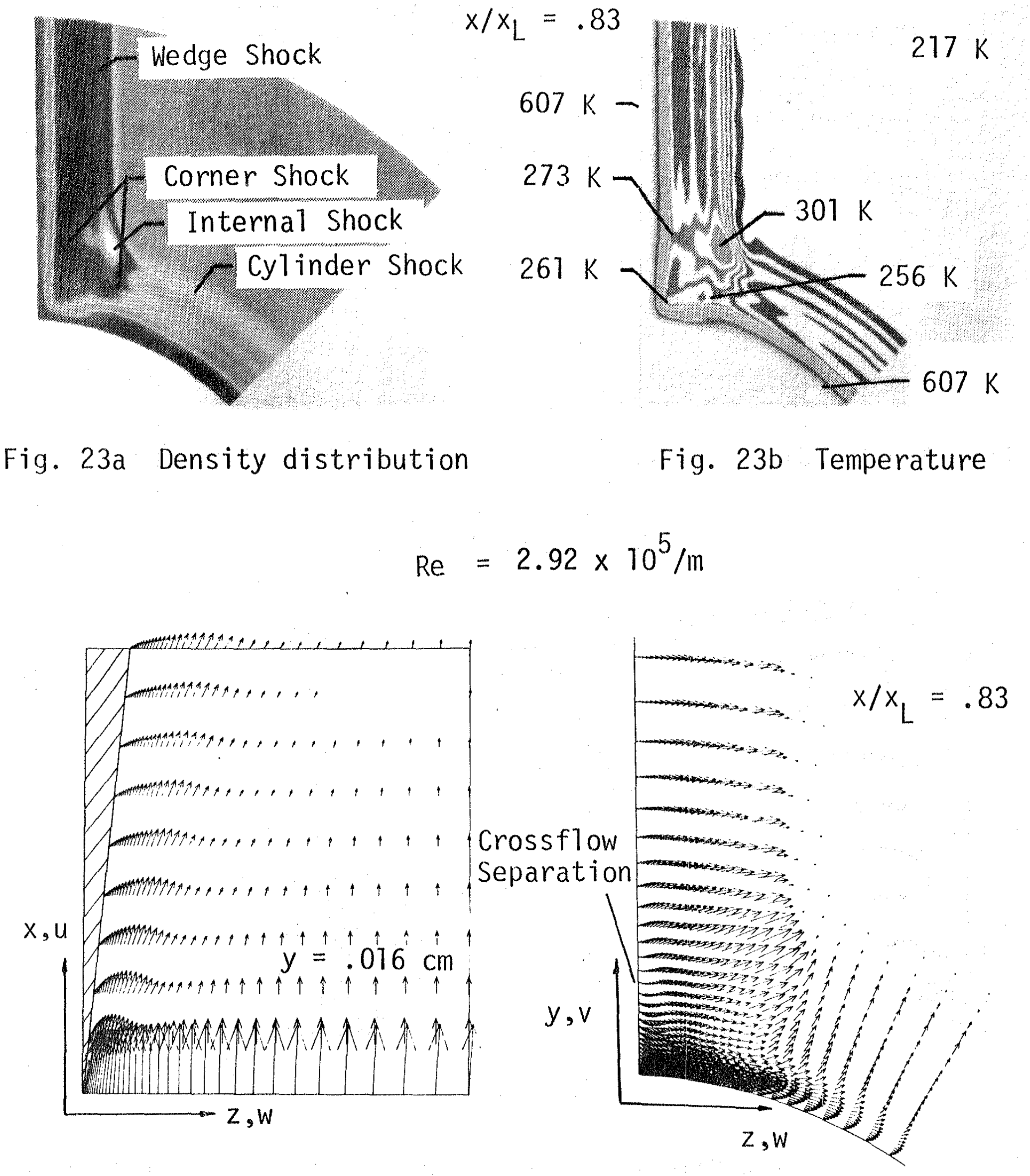


Fig. 23a Density distribution

Fig. 23b Temperature

Fig. 23c u x w Velocity

Fig. 23d v x w velocity

Fig. 23 Flow field description for a 6^0 wedge-cylinder corner.

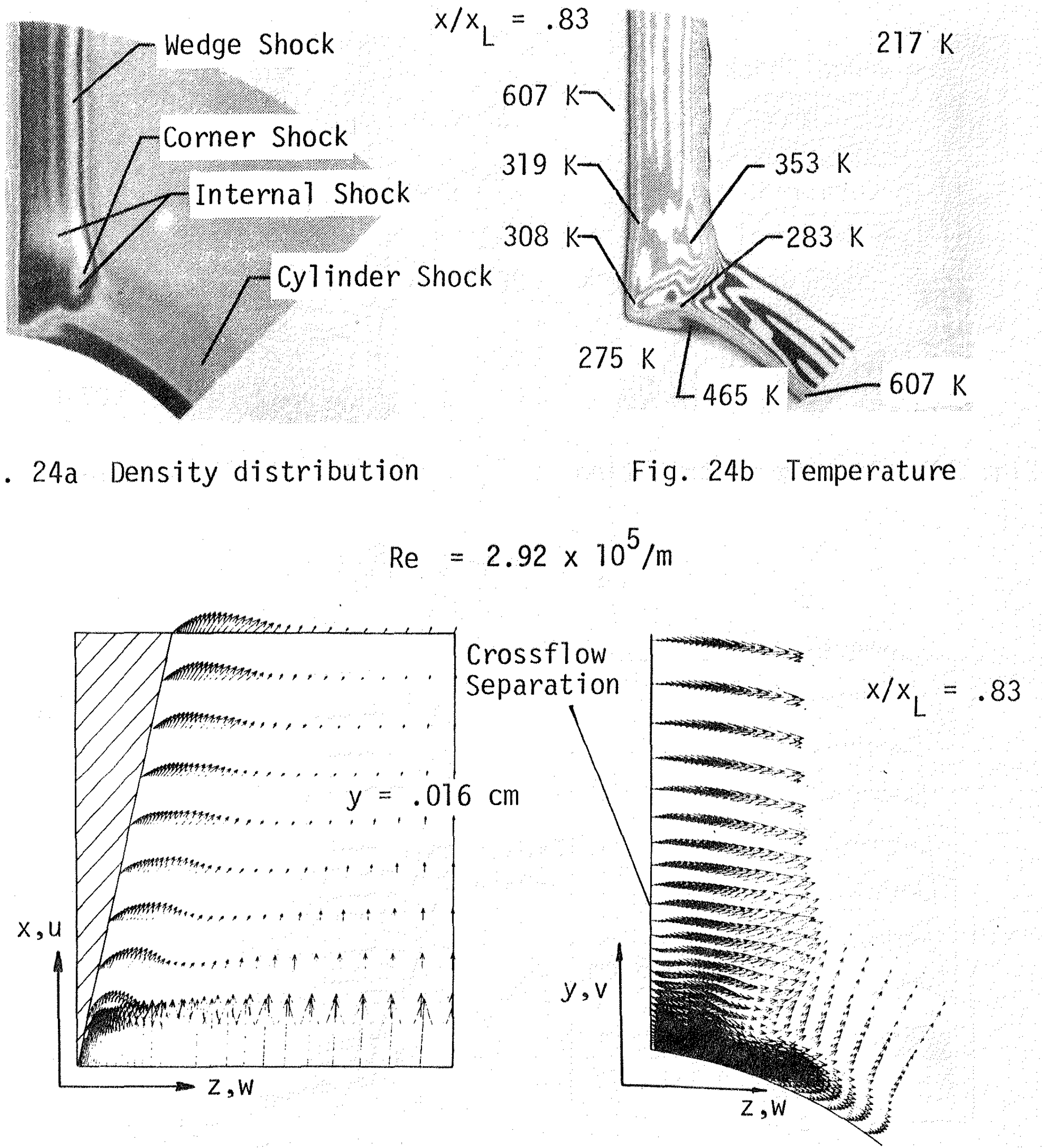


Fig. 24a Density distribution

Fig. 24b Temperature

Fig. 24c u x w Velocity

Fig. 24d v x w Crossflow velocity

Fig. 24 Flow field description for a 12.2^o wedge-cylinder corner.

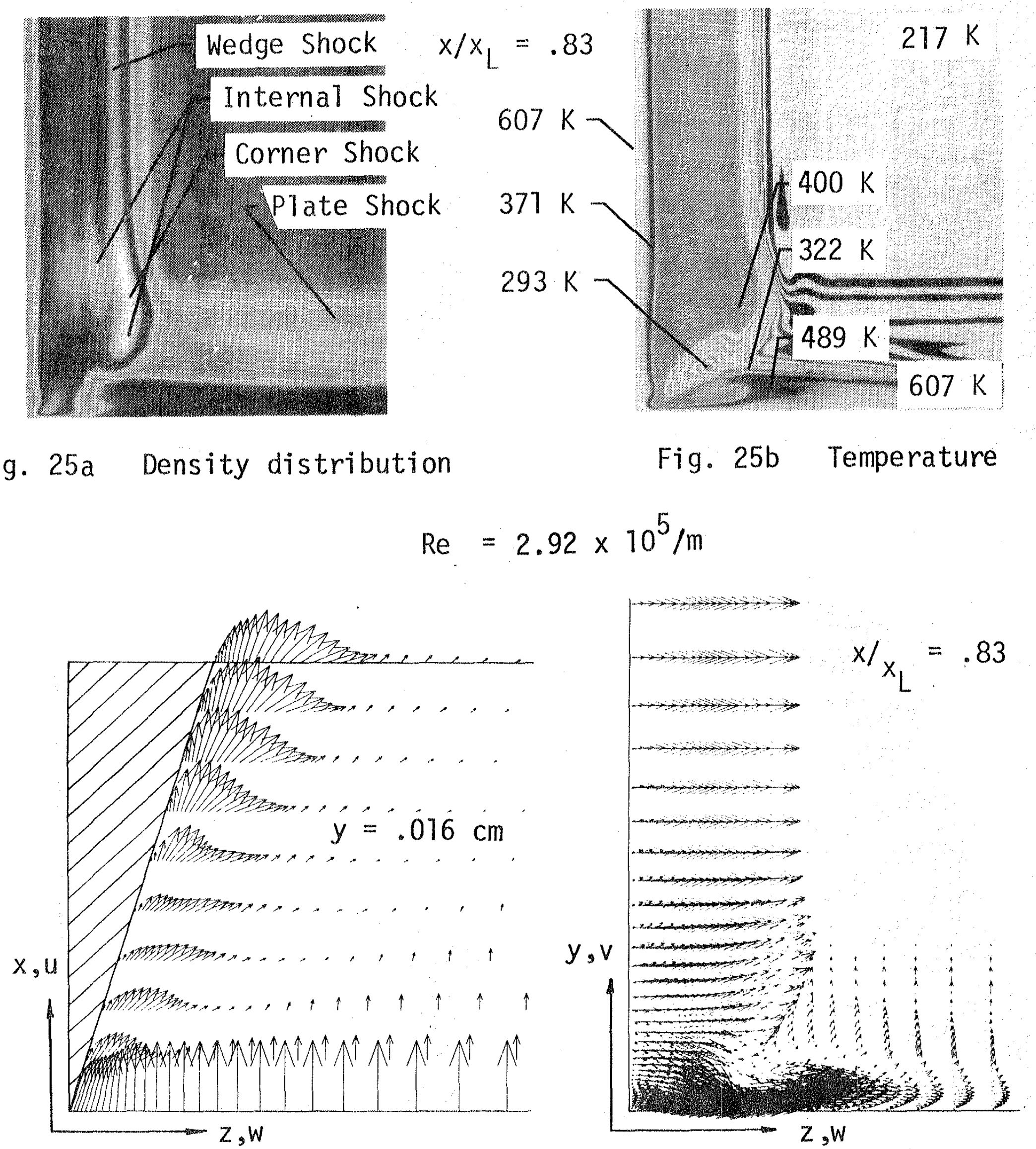


Fig. 25a Density distribution

Fig. 25b Temperature

Re = 2.92 x 10^5/m

Fig. 25c u x w Velocity

Fig. 25d v x w Crossflow velocity

Fig. 25 Flow field description for an 18^0 wedge-plate corner.

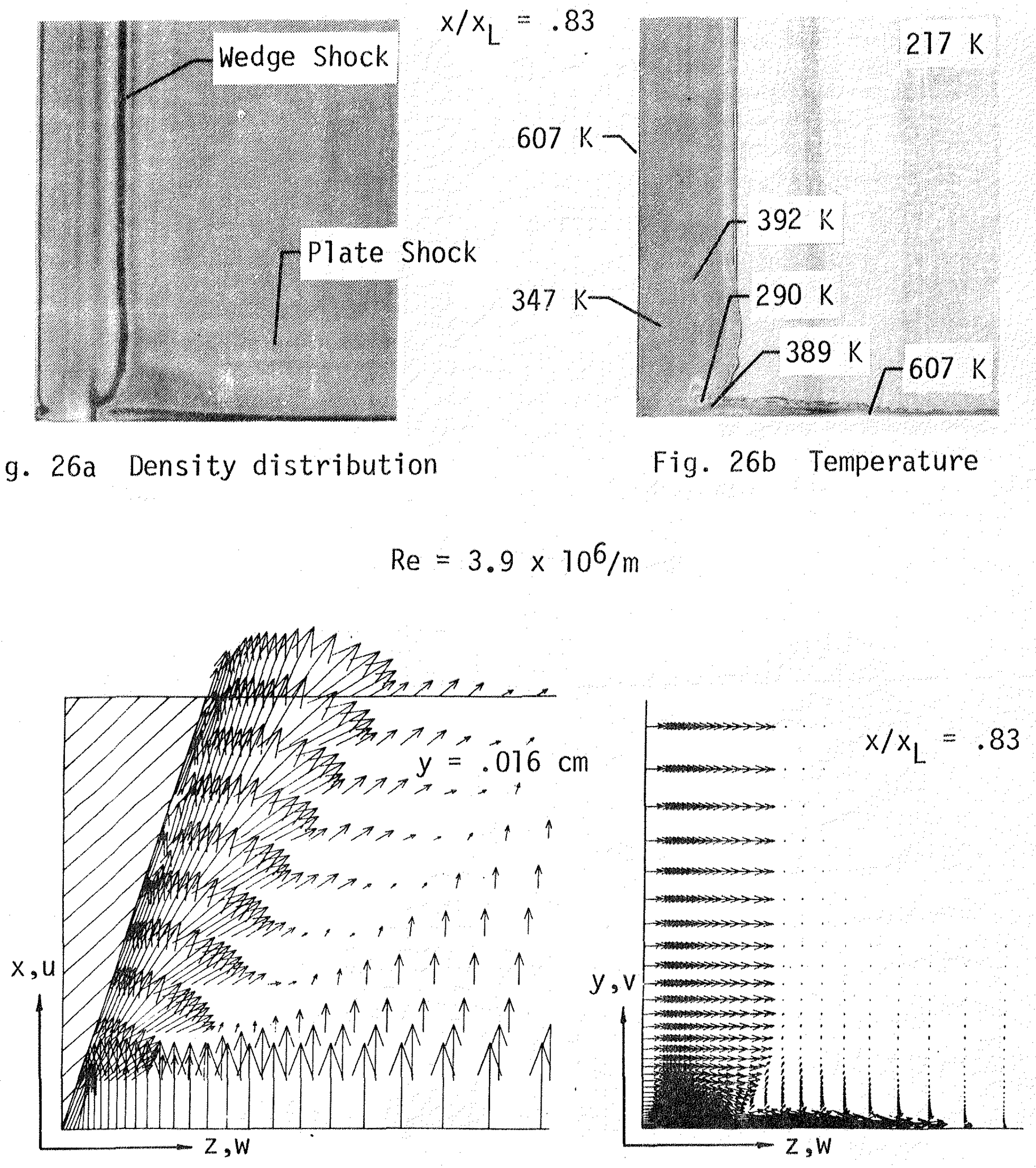


Fig. 26a Density distribution

Fig. 26b Temperature

Fig. 26c u x w Velocity

Fig. 26d v x w Crossflow velocity

Fig. 26 Flow field description for an 18^{o} wedge-plate corner (high Reynolds number).

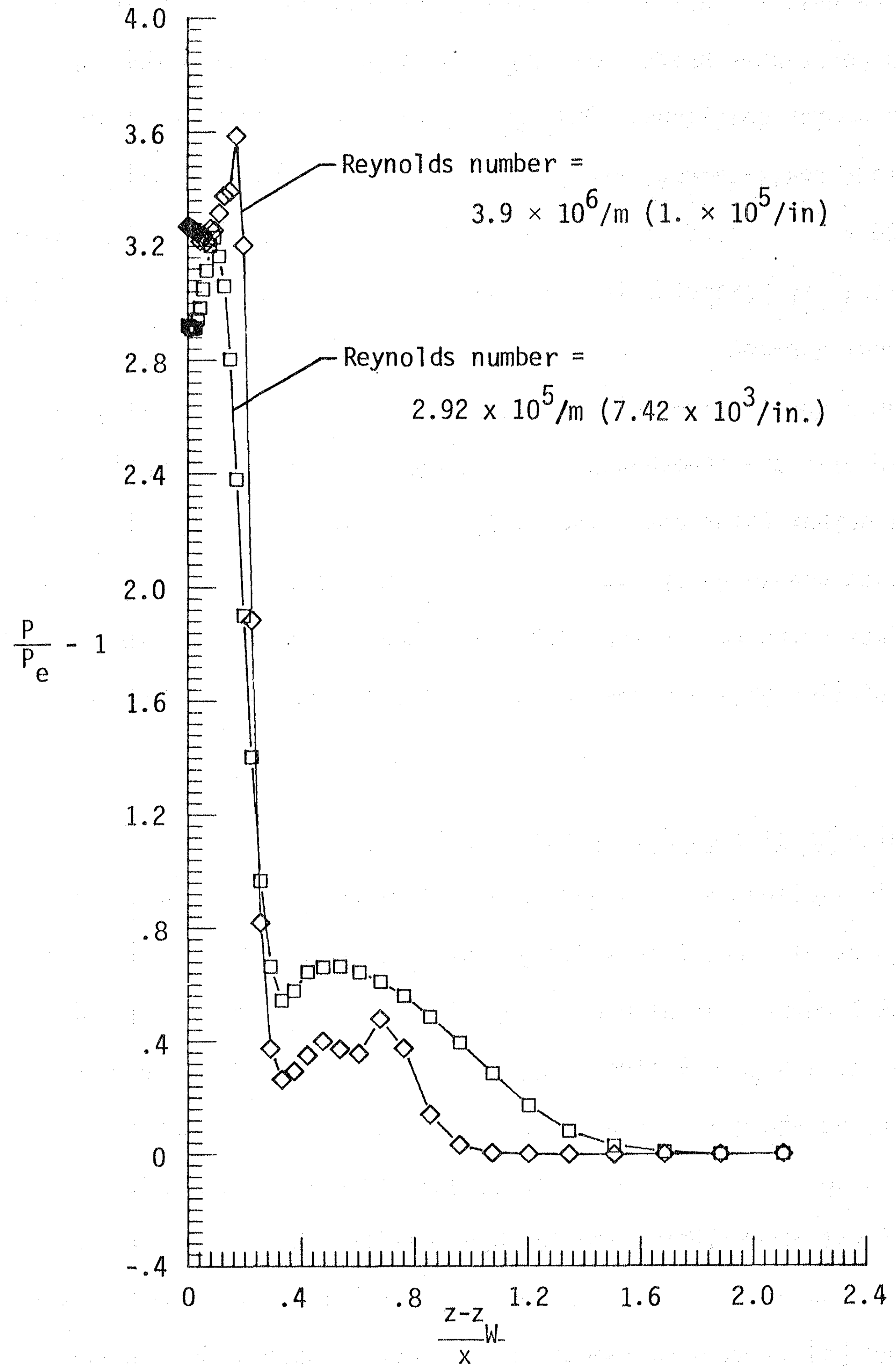


Fig. 27 Surface pressure for an 18^o wedge-plate corner.

case. The grid is coarse for defining the boundary layer, but the solution correlates better with physical experiments than the low Reynolds number solutions. Only one triple point is observed and separation, reattachment, and secondary separation are readily seen in Figure 26 and the surface pressure in Figure 27. Obviously the Reynolds number is a key parameter in the inviscid-viscid interaction in a three-dimensional corner.

From a qualitative point of view it is apparent that the grids generated with the "two-boundary technique" perform well for the low Reynolds number solutions. The surface pressure for the 0, 6, and 12.2^{o} wedge-plate and wedge-cylinder solutions are shown in Figures 28-29. The surface pressure for the 12.2^{o} wedge-plate corner is later compared with a solution obtained for the same boundary geometry with a finer grid.

4.1.1 High Resolution Grid Solutions

It is implied in the above section that more grid points are needed to resolve the fine structure of supersonic flow about three-dimensional corners at high Reynolds numbers. In this section three solutions at a high Reynolds number and one at a low Reynolds number are described where a 12 × 64 × 64 grid is used. The objective of the section is to present the most highly defined solutions within existing computational capabilities and further validate the use of the "two-boundary technique." The coordinate transformation presented by Shang and Hankey [4] is used to compute the solution about a 12.2^{o} symmetric

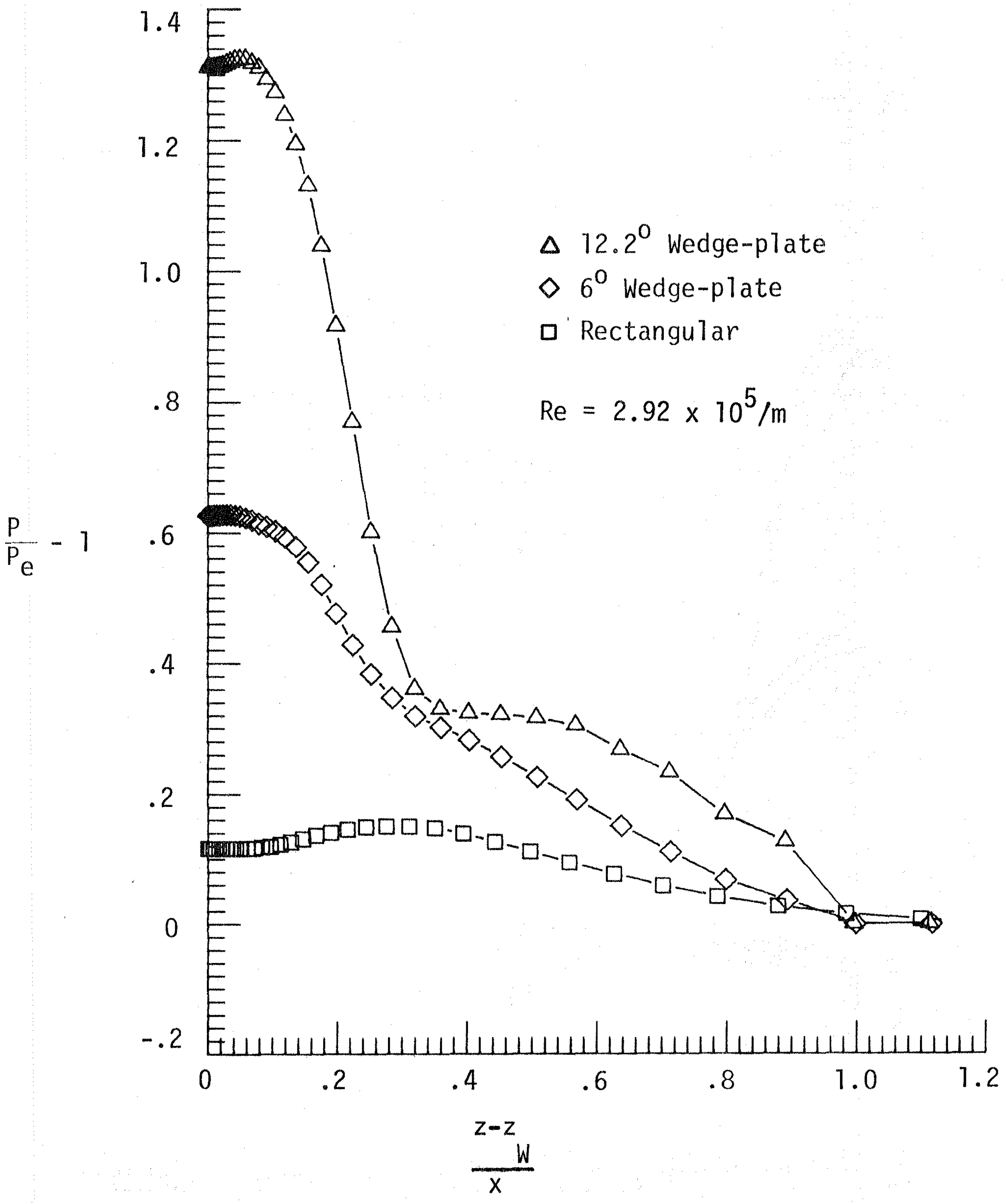


Fig. 28 Surface pressure for wedge-plate corners.

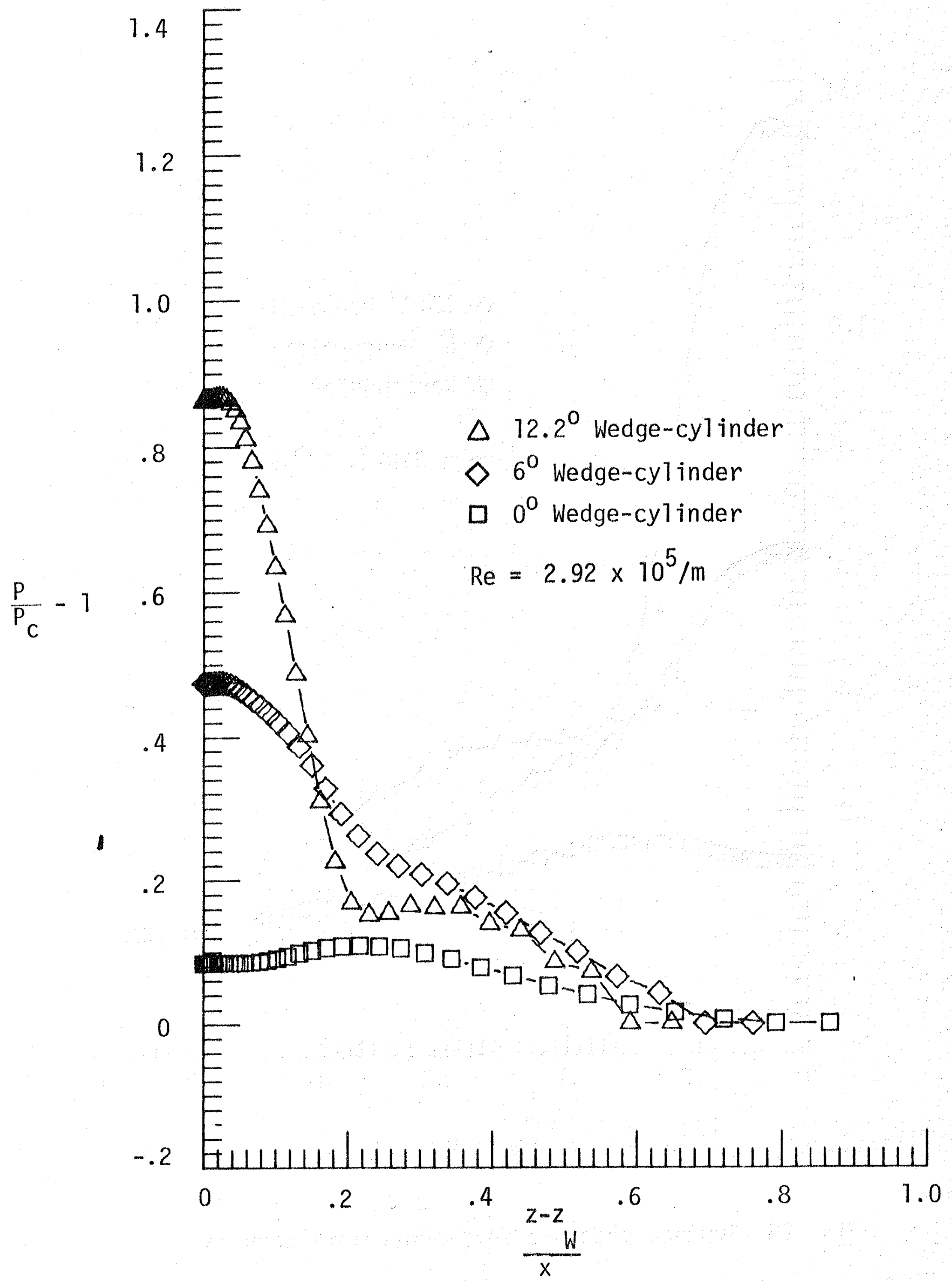


Fig. 29 Surface pressure for wedge-cylinder corners.

wedge-wedge corner (Fig. 30) at Mach 3.64 and Reynolds number 3.9×10^6/m (1×10^5/in.). The solution is compared with the experimental data obtained by Charwat and Redekopp [34]. Using the Shang-Hankey transformation which can also be derived with the "two-boundary technique" and letting one of the wedge angles be zero, the solution of the flow about a 12.2° asymmetric wedge corner is obtained with the $12 \times 64 \times 64$ grid. A $12 \times 64 \times 64$ grid is generated using Equation (2.17) and a solution is again obtained for the flow about the 12.2° asymmetric corner. The procedure for obtaining solutions at the high Reynolds number is to first start the solution at a low Reynolds number and increment to the larger value. The flow conditions are changed on the upstream plane and are integrated downstream.

The transformation used in [4] is

$$\xi = x/x_L \tag{3.1a}$$

$$\eta = \frac{1}{\tilde{k}} \ell n[1 + (e^{\tilde{k}} - 1)(\frac{y}{x} - \tan \delta_W) \frac{1}{Y_L}] \tag{3.1b}$$

$$\zeta = \frac{1}{\tilde{k}} \ell n[1 + (e^{\tilde{k}} - 1)(\frac{z}{x} - \tan \theta) \frac{1}{Z_L}] \ . \tag{3.1c}$$

This transformation can be derived with the "two-boundary technique" where

$$x_1(\xi,\zeta) = x(\xi) = \xi \ X_L$$

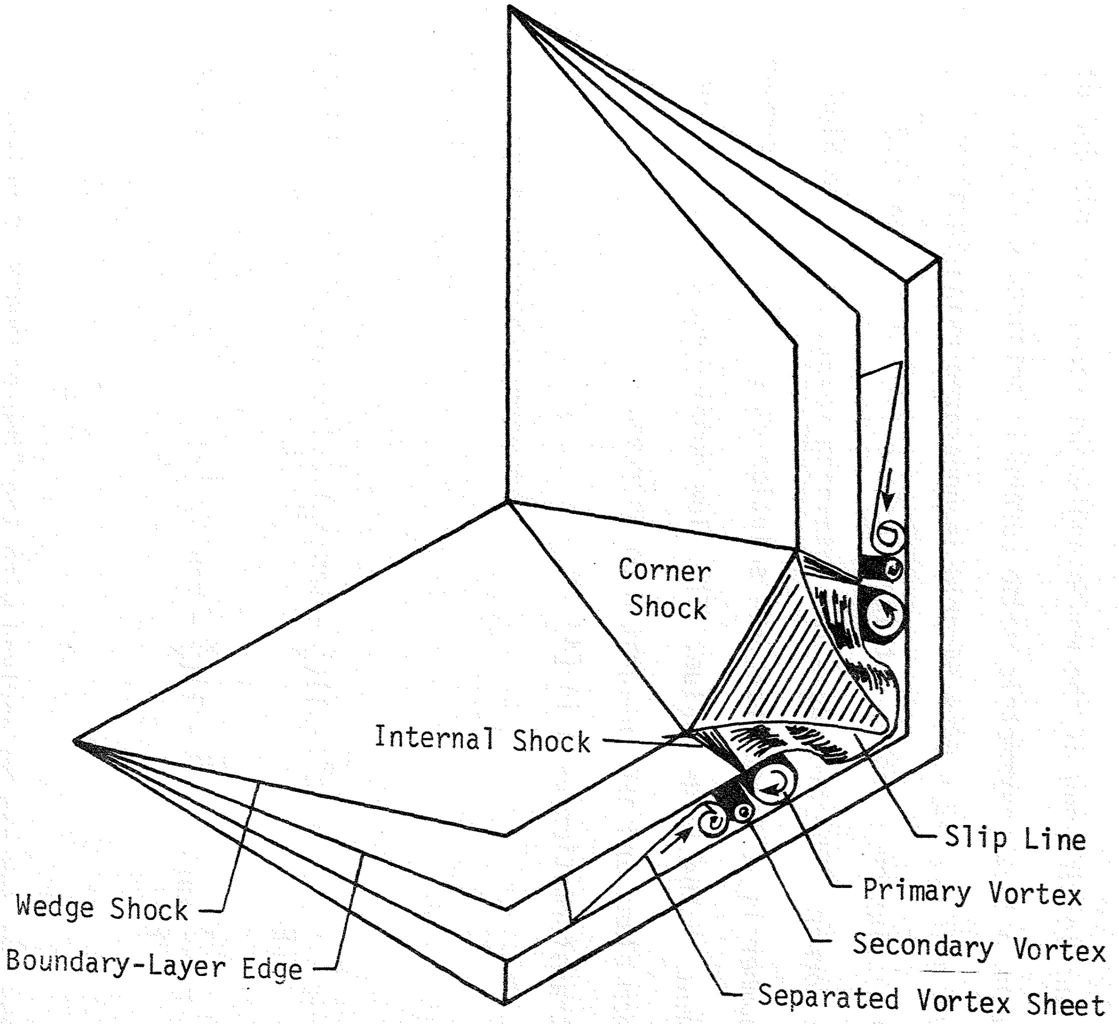


Fig. 30 Hypothetical symmetric corner flow from Ref. 37.

$$y_1(\xi,\zeta) = \xi\ X_L \tan \delta_W$$

$$z_1(\xi,\zeta) = \xi\ X_L(\tan \theta_W + Z_L)\bar{\zeta} + \xi\ X_L(\tan \theta_W)(1 - \bar{\zeta})$$

$$x_2(\xi,\zeta) = x(\xi) = \xi\ X_L$$

$$y_2(\xi,\zeta) = \xi\ X_L(\tan \delta_W + Y_L)$$

$$z_2(\xi,\zeta) = z_1(\xi,\zeta)$$

$$\bar{\zeta} = \frac{e^{\tilde{k}\zeta} - 1}{e^{\tilde{k}} - 1}$$

$$0 \leq \zeta \leq 1$$

$$0 \leq \bar{\zeta} \leq 1$$

Using a linear connecting function and rearranging terms yields

$$x = \xi\ X_L \tag{3.2a}$$

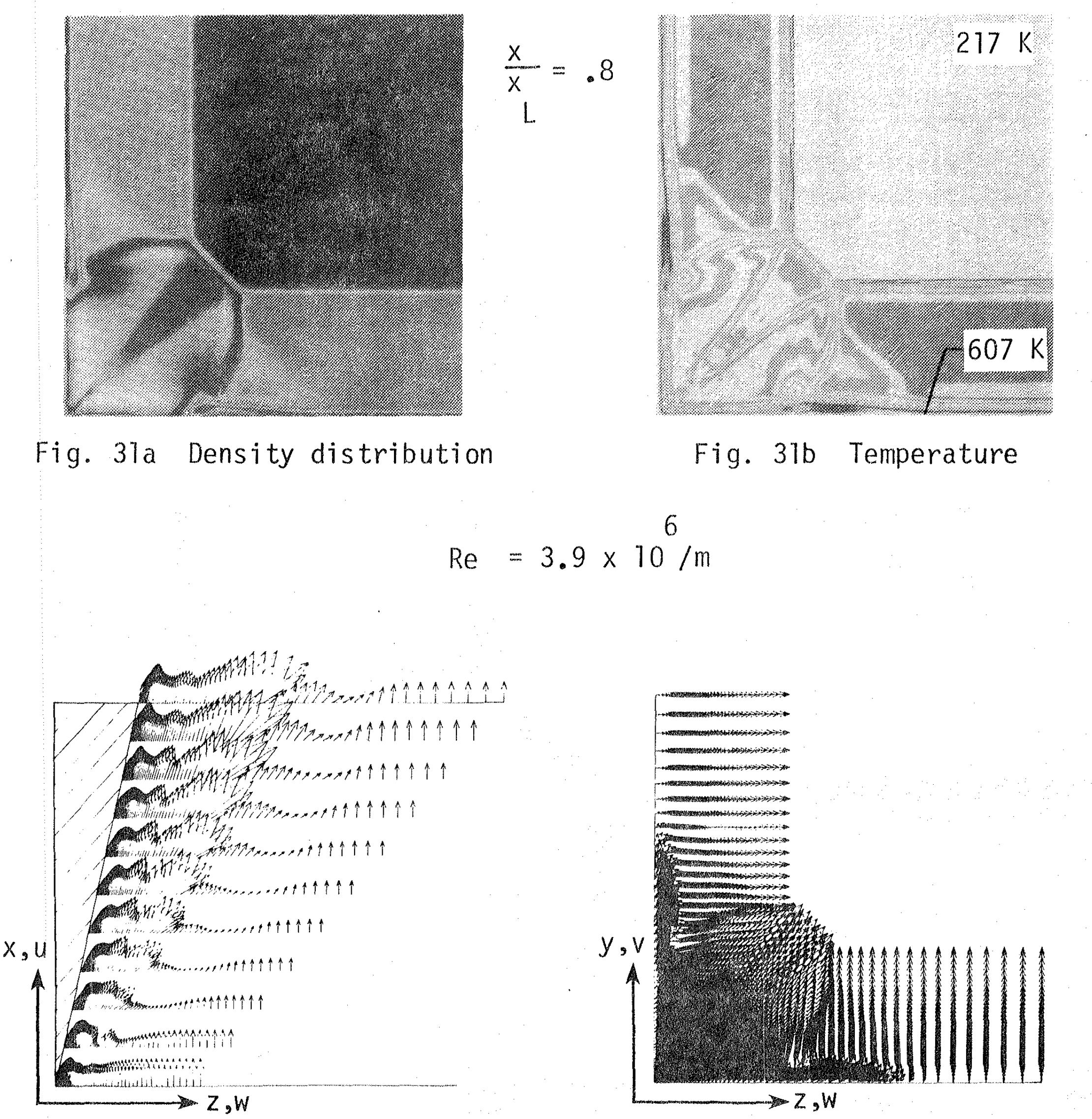
$$y = \xi\ X_L\{\tan \delta_W + Y_L(\frac{e^{\tilde{k}\eta} - 1}{e^{\tilde{k}} - 1})\} \tag{3.2b}$$

$$z = \xi\ X_L\{\tan \theta_W + Z_L(\frac{e^{\tilde{k}\zeta} - 1}{e^{\tilde{k}} - 1})\} \tag{3.2c}$$

$$0 \leq \eta \leq 1 \quad .$$

Equation (3.1) is the inverse of Equation (3.2). This transformation applies only to corners formed from planar intersecting boundaries, and the transformation data is derived directly by differentiating ξ, η, and ζ with respect of x, y, and z.

The solution obtained with a 12 × 64 × 64 grid and Equation (3.1) for a 12.2^0 symmetric wedge corner is shown in Figure 31-34. The density and temperature distributions show the zones of flow described by Charwat and Redekopp [34]. In this solution, the slip lines are very well defined. The line contour plot (Fig. 32) and the surface distribution of the density (Fig. 33) also shows the zones of flow and the rapid change in density associated with the slip lines. The slip line definition is not as apparent for the rectangular corner solution shown in the previous section. No crossflow separation is observed, which is the case with the rectangular corner. The position of the shock structure from [34] is superimposed on the density contour plot (Fig. 32) and the agreement is good. In [34] the shock structure is not perfectly symmetric, and only the upper half is used for comparison. The surface pressure in Figure 34 shows the comparison of the Navier-Stokes solution with Charwat's experiment [34]. The plateau, dip, and overshoot described by Korkegi [48] occur at nearly the same position. There is, however, some disagreement in the pressure magnitude. The Navier-Stokes solution is on the low side. There is pressure dispersion in Charwat's data [34] but the pressure magnitude is still larger than that obtained in the Navier-Stokes solution. Two possible sources for this difference are (1) the Reynolds number may not be exactly the same



$\frac{x}{x_L} = .8$

Fig. 31a Density distribution

Fig. 31b Temperature

$Re = 3.9 \times 10^6 /m$

Fig. 31c u x w Velocity

Fig. 31d v x w Crossflow Velocity

Fig. 31 Flow field solution for a symmetric 12.2° wedge-wedge corner.

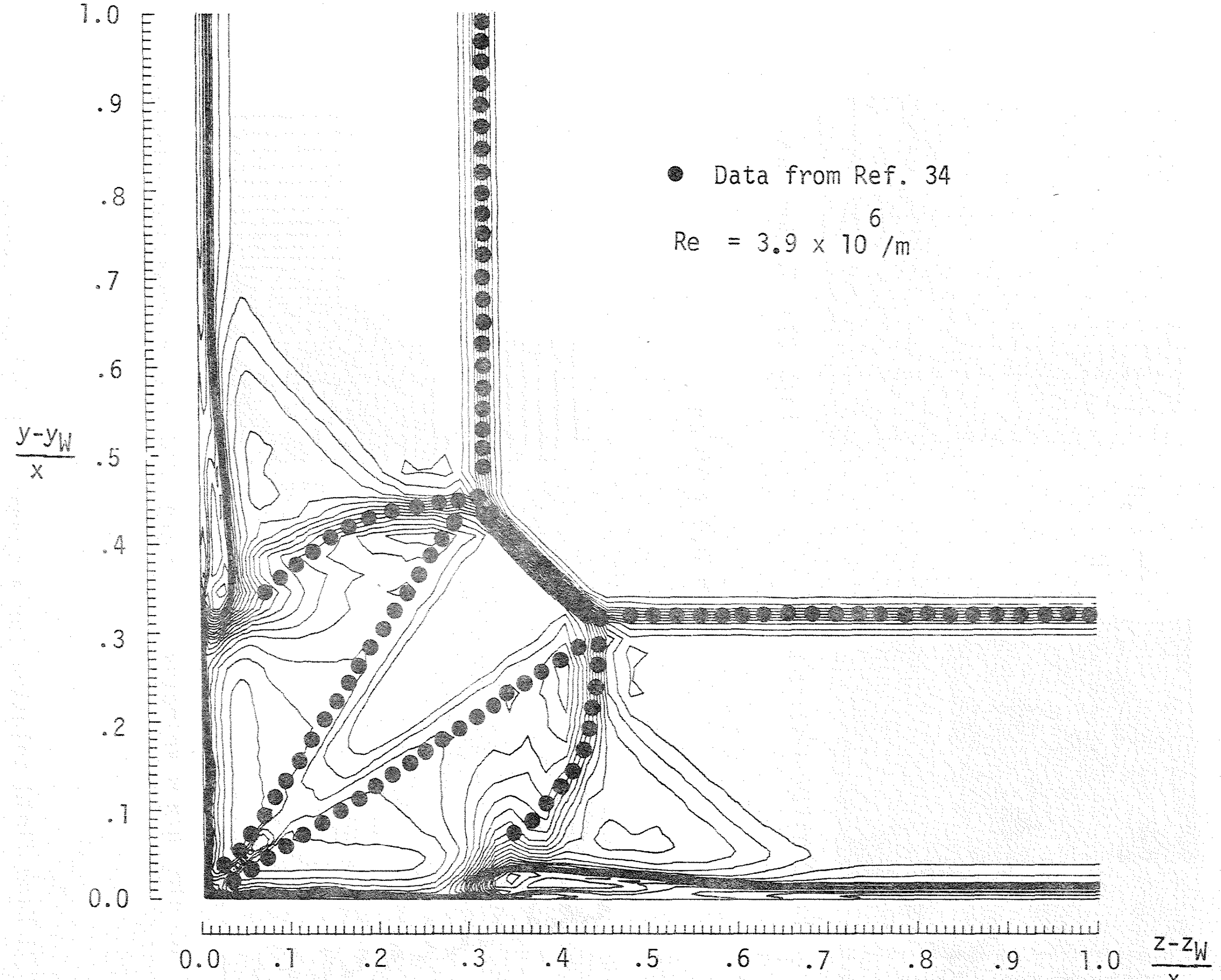


Fig. 32 Line contour plot of density for a 12.2^o symmetric corner.

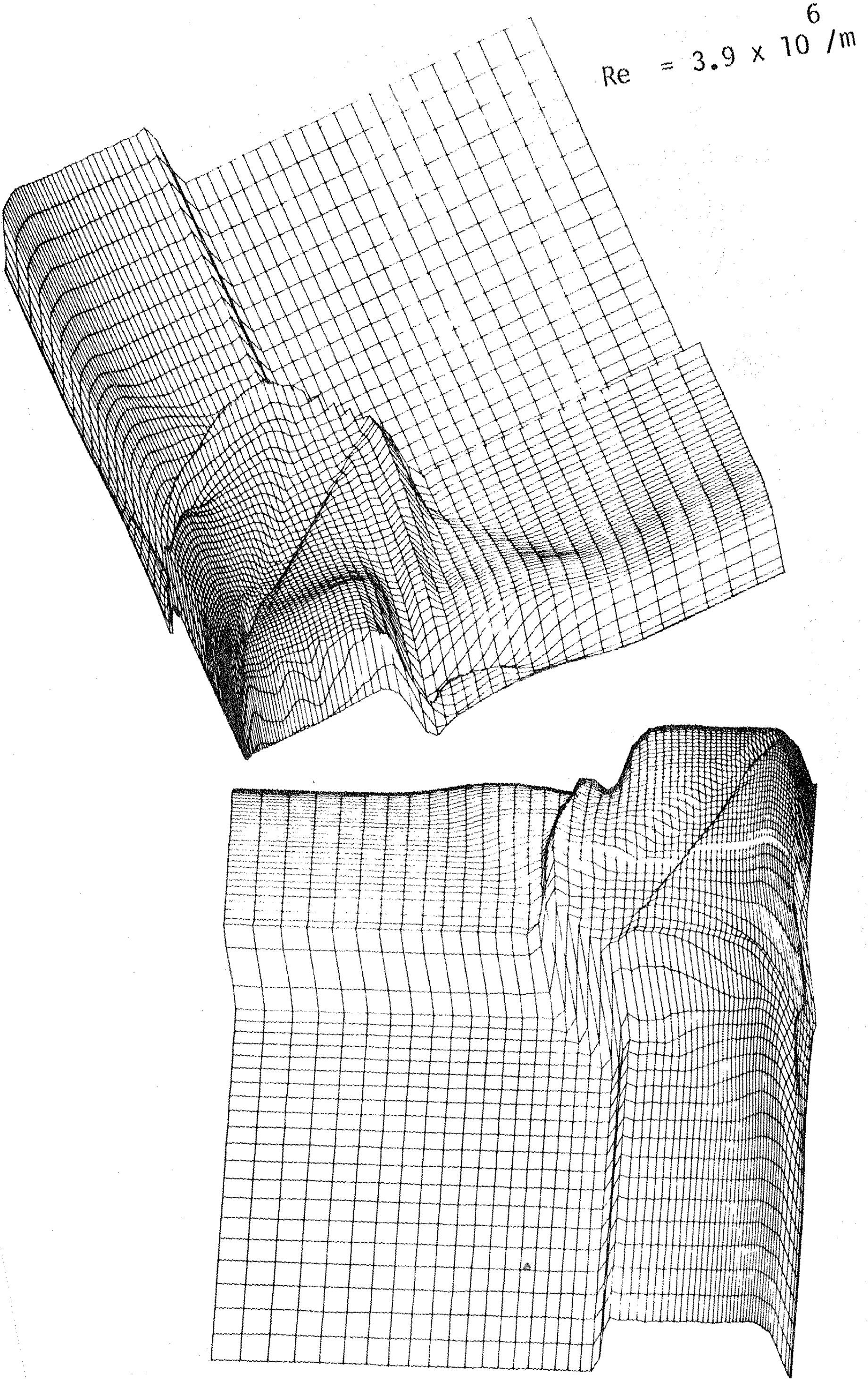


Fig. 33 Surface distribution of density - two views - 12.2° symmetric corner.

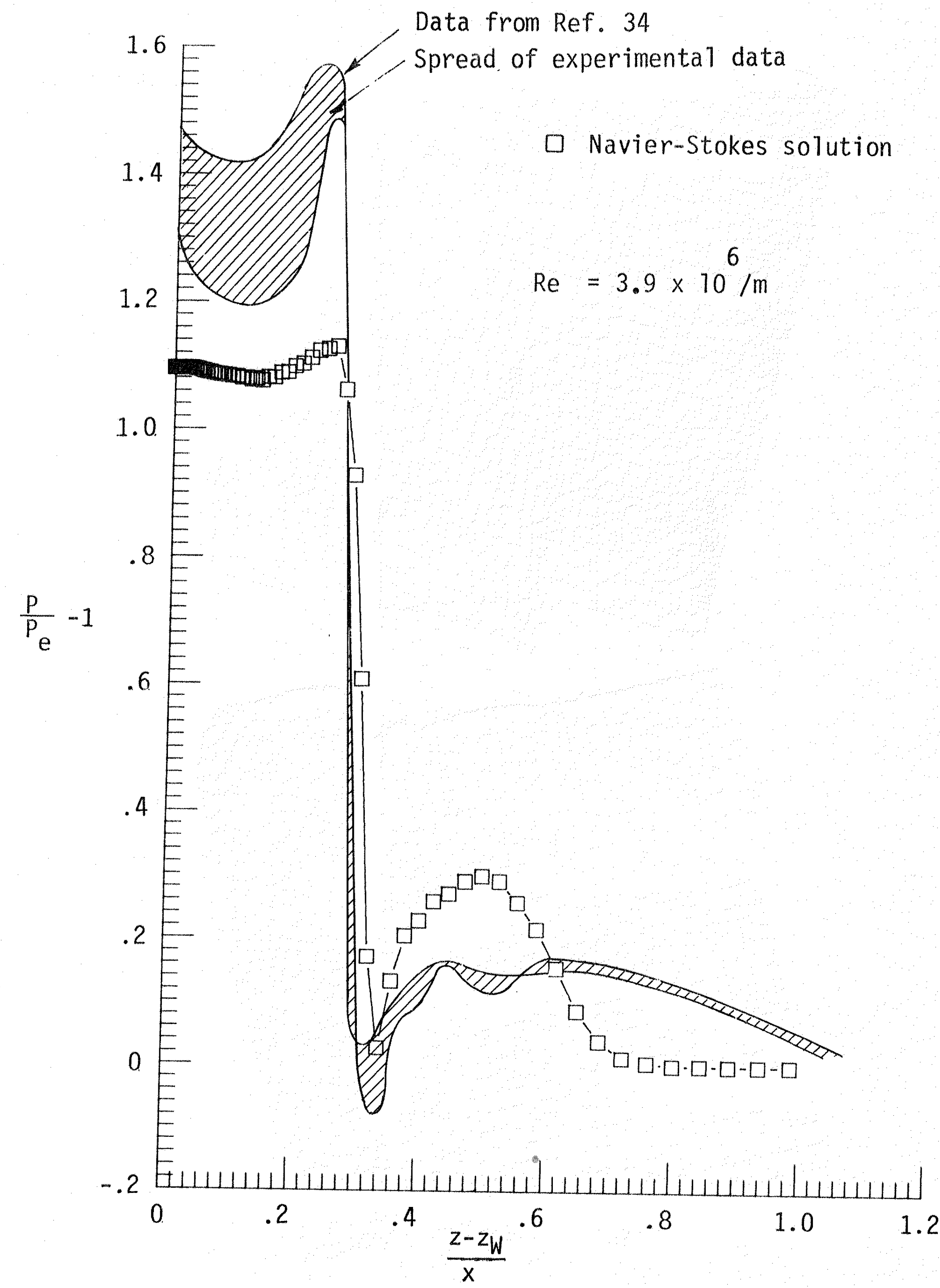


Fig. 34 Surface pressure for a 12.2° symmetric corner.

in the numerical and physical experiments, or (2) the pressure damping in the numerical technique may be retarding the pressure magnitude. The plateau region (Fig. 34) is associated with the relatively low uxw velocity vectors in Figure 31 and the separated region. The separation line is under the plateau near the undisturbed flow and reattachment occurs at the high pressure rise and where there is corresponding large velocity vectors.

The dip in the pressure curve is associated with secondary separation as described by Korkegi [48]. This secondary separation region is shown in the density contour plot (Fig. 32) and there are closed contour lines which indicate vortical motion as described by Watson [37]. However, the author has not observed vortical motion in the velocity data.

It is evident that the 12 × 64 × 64 grid derived from Equation (3.1) allows for the definition of the fine structure of the flow. The solution obtained with the Navier-Stokes solver is in good comparison with the corresponding experiment. The possible concentration of more grid points in the secondary separation region may be desirable but would not work well with Equation (3.1) because of its exponential characteristics.

Following the solution for a symmetric planar corner flow, supersonic flow about an asymmetric corner is obtained by letting the wedge angle ζ_w in Equation (3.1) on the bottom surface be zero. The wedge angle θ_w remains 12.2^o and the 12 × 64 × 64 grid for this corner is generated with Equation (3.1). The Navier-Stokes solver is applied with the transformation data starting impulsively with a low Reynolds number and incrementing to 3.9 × 10^6/m. The solution is shown in Figure 35-38, but is not compared to a physical experiment. The solution obtained with a grid generated using Equation (2.17) is compared to this solution.

For the asymmetric corner and the 12 × 64 × 64 grid only one triple point is observed. However, the internal shock from the wedge generates a very similar flow pattern as in the symmetric corner near the bottom surface. The pattern is, however, closer to the wedge surface since there is no corner shock. The crossflow separation is observed at the high Reynolds number but is relative weak compared to the lower Reynolds number solutions.

Equation (2.17) is used to generate a 12 × 64 × 64 grid with the previously used physical dimensions and contraction parameters $k_1 = k_2 = 2.9$. The transformation data for the grid is applied in the Navier-Stokes solver in the same manner as before. The surface pressure for the solution at Reynolds number 2.92 × 10^5/m is shown in Figure 39 and compared with that obtained with the 20 × 36 × 36 grid. The agreement is good where the grid is dense and the solution at Reynolds number 3.9 × 10^6/m is obtained through a series of increments. This solution is presented in Figures 40-43. The surface pressure in Figure 43 is compared to that obtained with the grid defined with Equation (3.1). Overall the agreement is good. One point that is noticed, however, is that the shock is more smeared in the solution obtained with the transformation data from Equation (2.17). This is

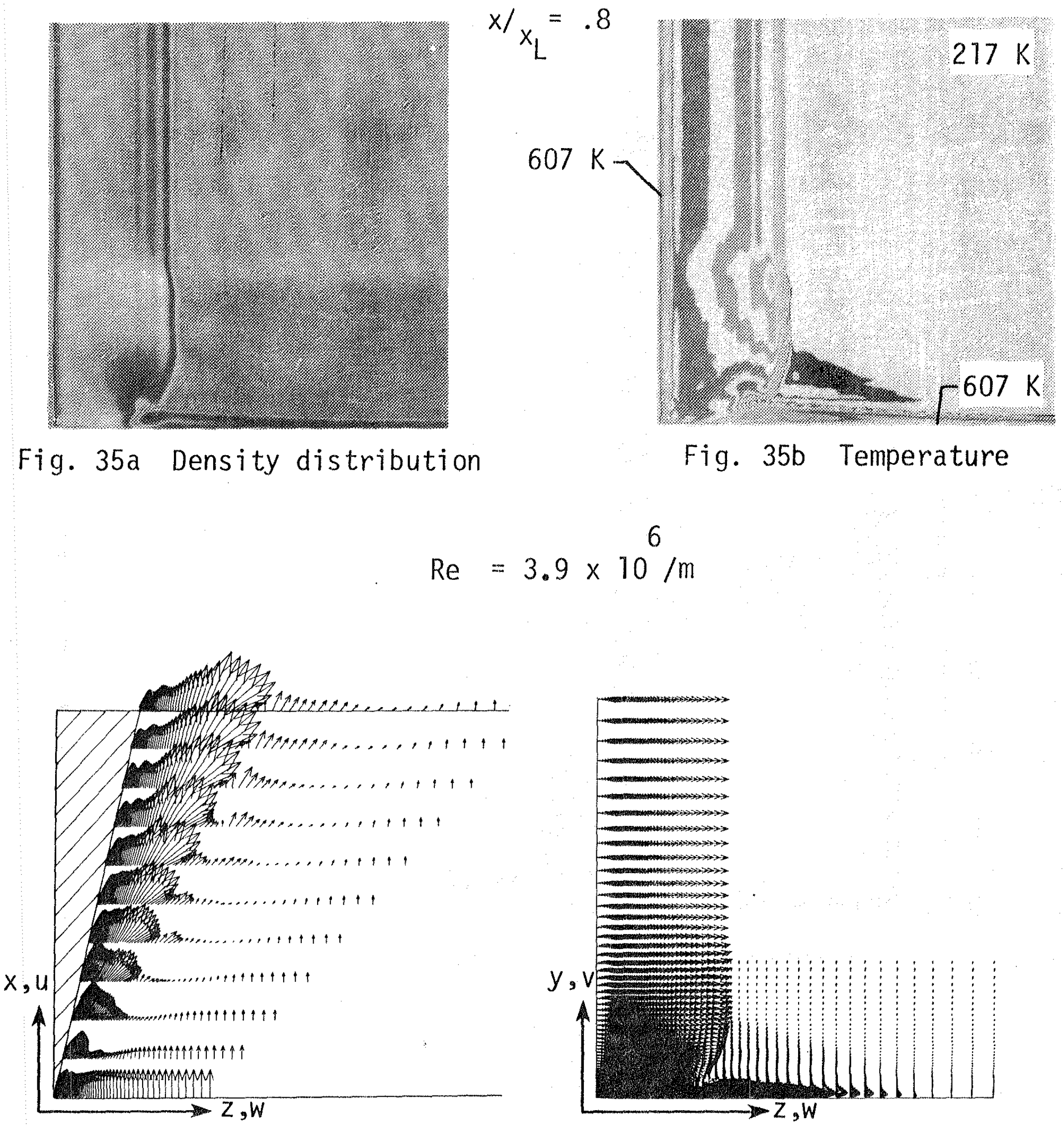


Fig. 35a Density distribution

Fig. 35b Temperature

Fig. 35c u x w Velocity

Fig. 35d v x w Crossflow velocity

Fig. 35 Flow field description for a 12.2° asymmetric corner (12 × 64 × 64 grid-exact boundaries)

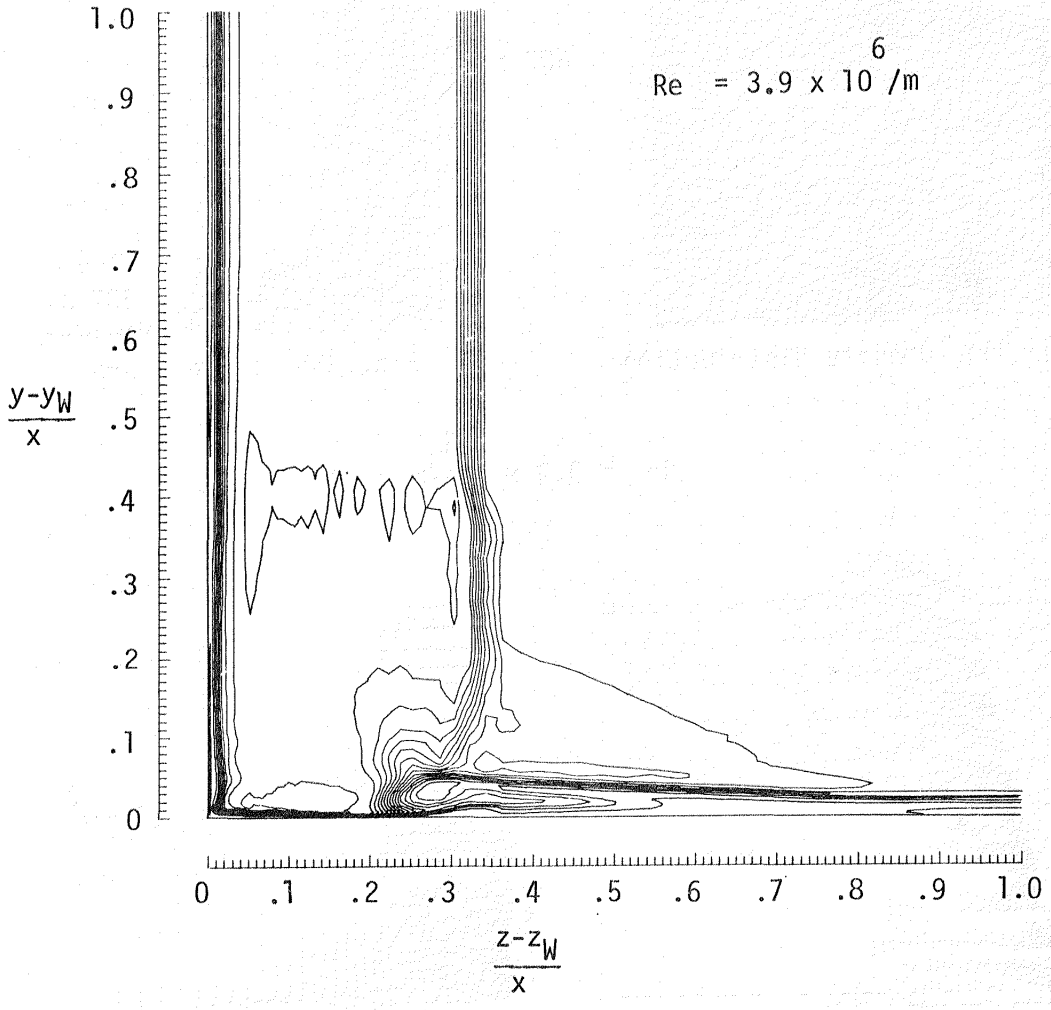
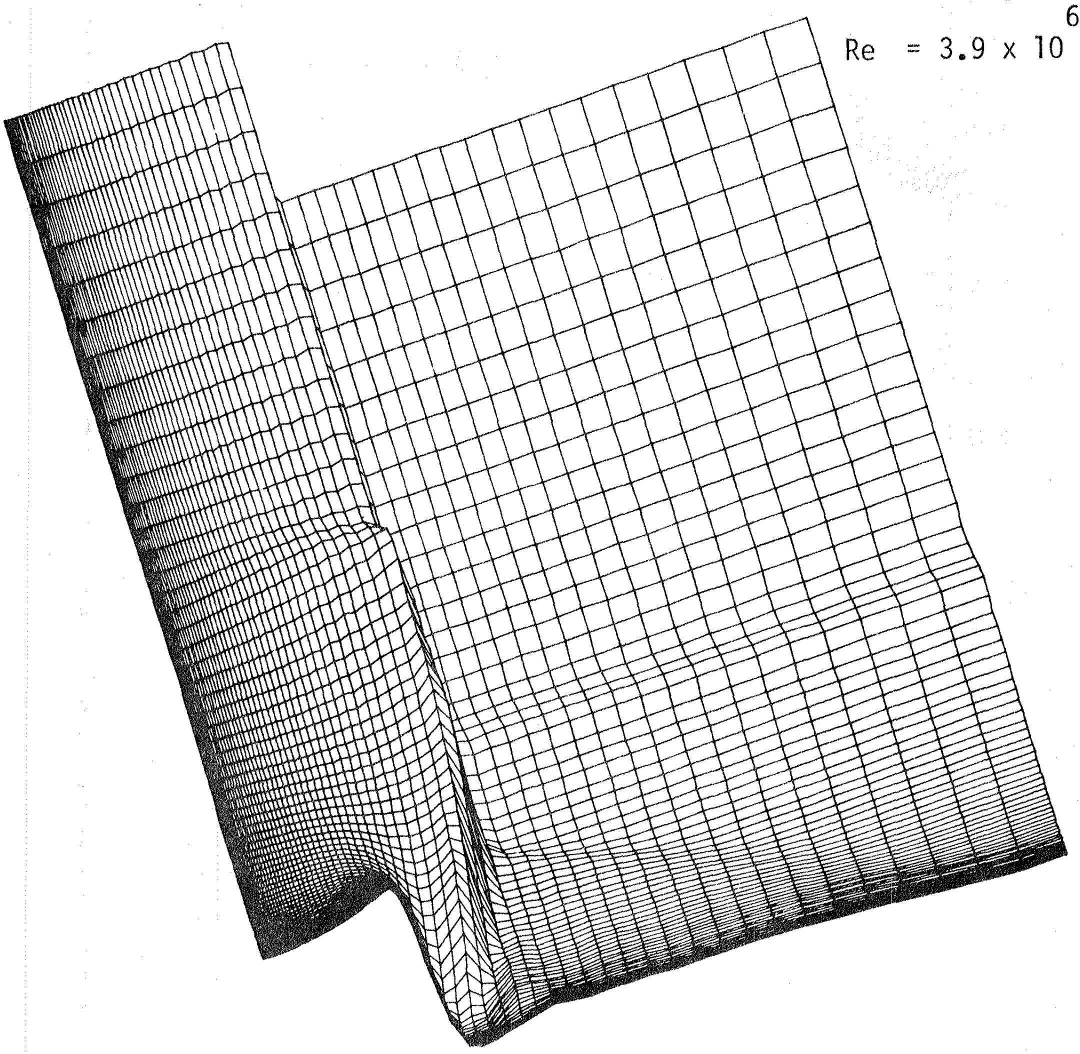


Fig. 36 Line contour plot of 12.2^o asymmetric corner (12 × 64 × 64 grid-exact boundaries).



$Re = 3.9 \times 10^6/m$

Fig. 37 Perspective view of distribution of density for 12.2° asymmetric corner $x/x_L = .8$ (12 × 64 × 64 grid - exact boundaries).

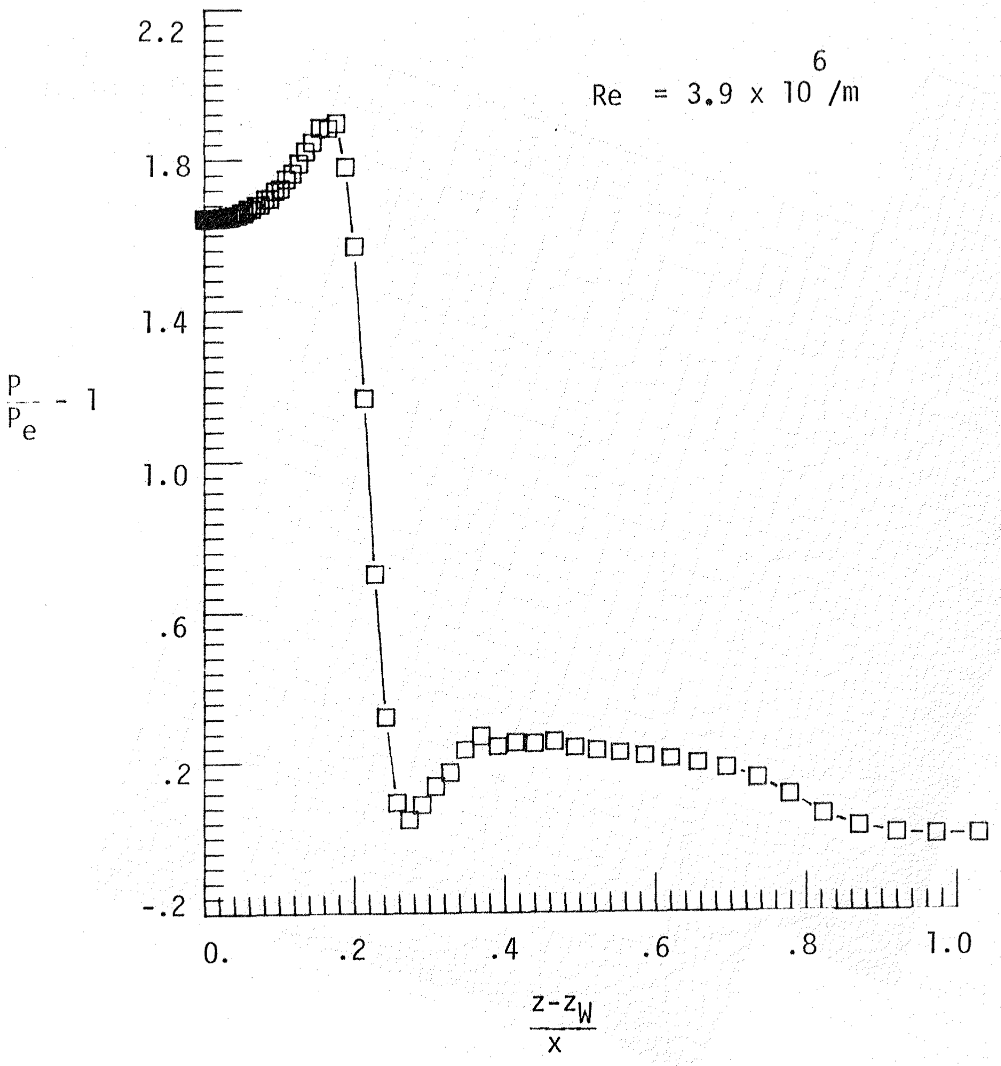


Fig. 38 Surface pressure for 12.2° asymmetric corner (12 × 64 × 64 grid - exact boundaries).

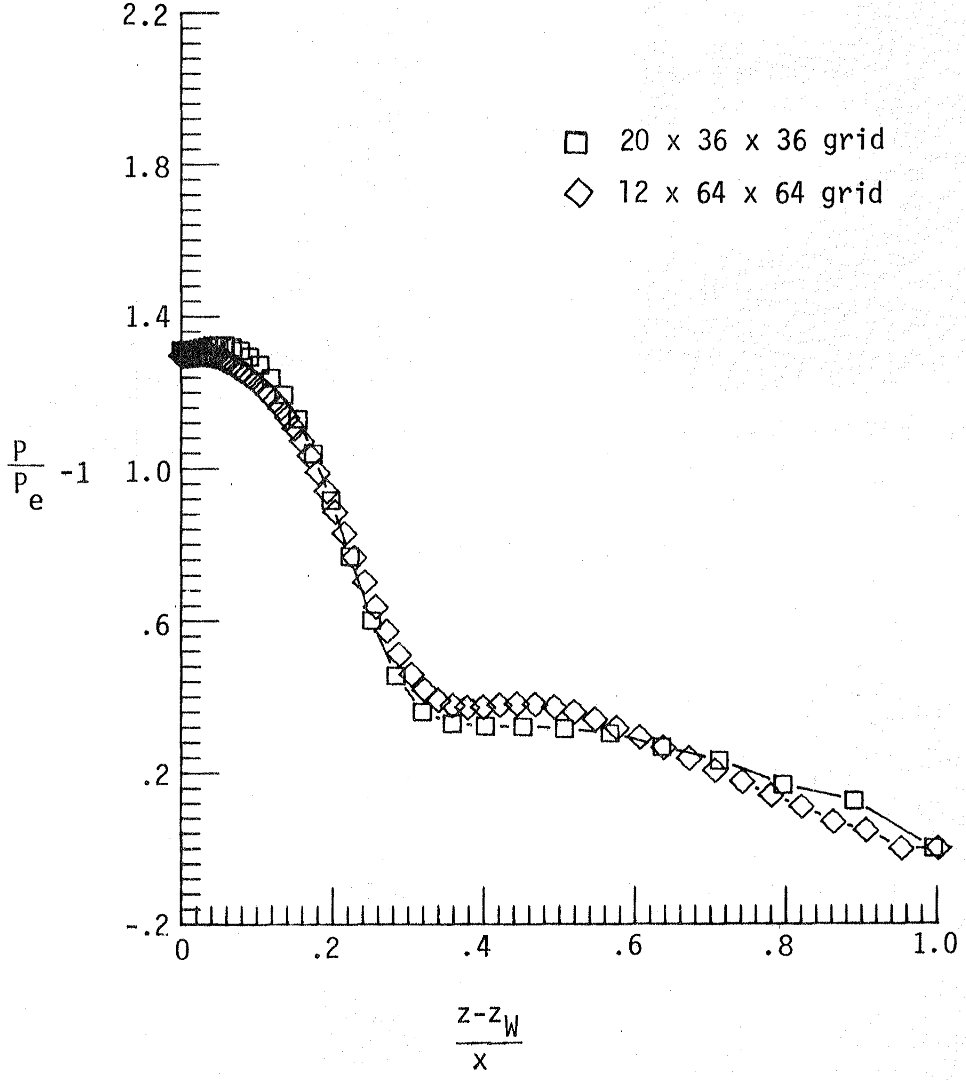


Fig. 39 Surface pressure comparison for grid refinement at Re = 2.92 × 10^5/m.

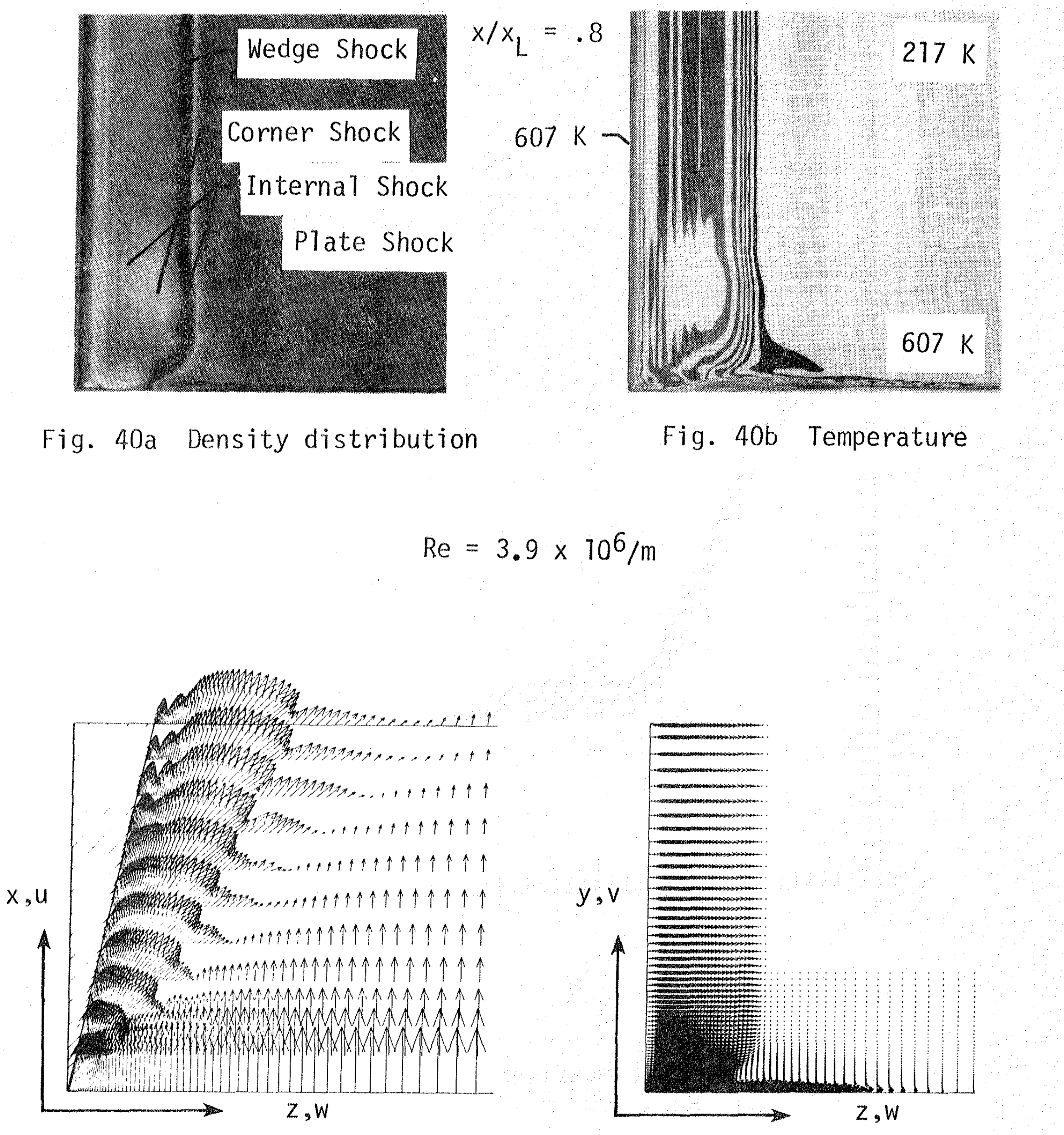


Fig. 40a Density distribution

Fig. 40b Temperature

Fig. 40c u x w Velocity

Fig. 40d v x w Crossflow velocity

Fig. 40 Flow field description for a 12.2^0 asymmetric corner (12 × 64 × 64 grid - approximate boundaries).

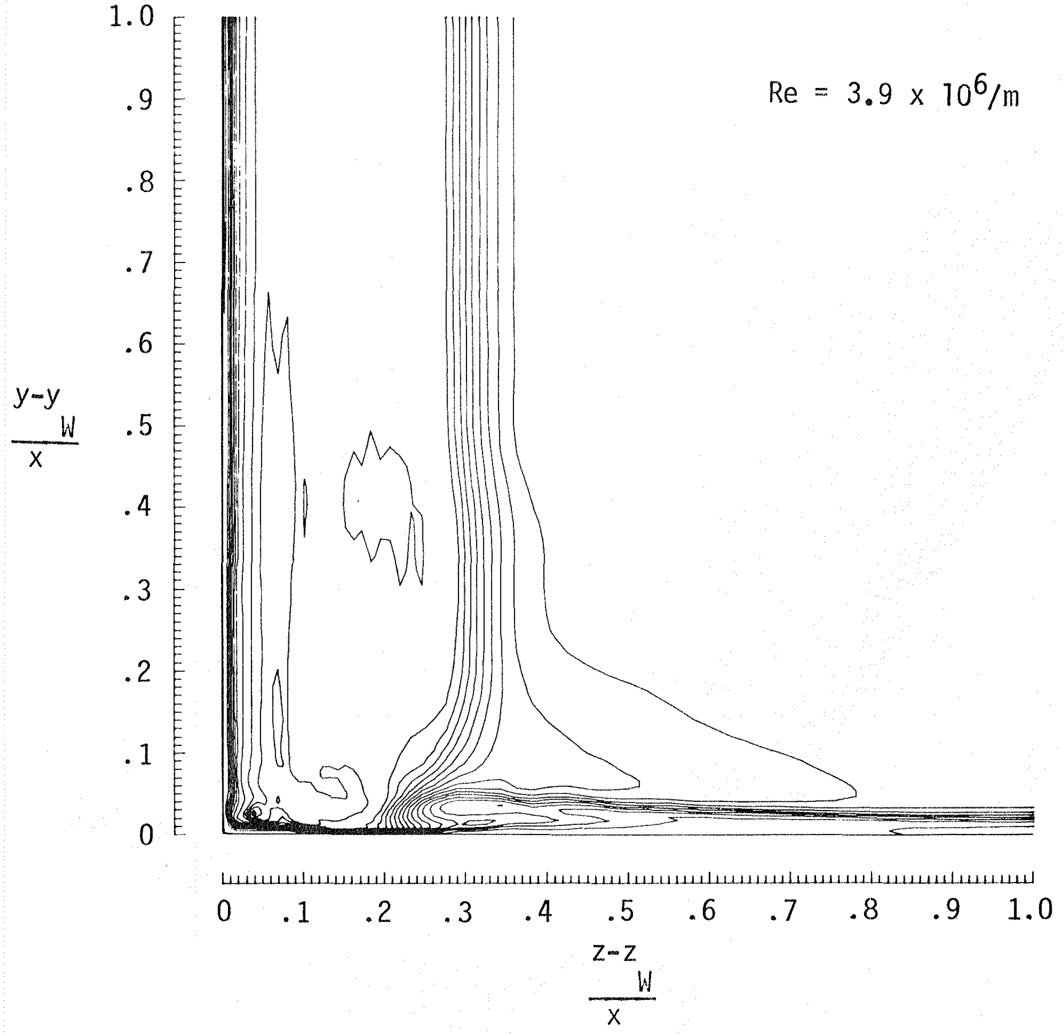


Fig. 41 Line contour plot for 12.2^0 asymmetric corner (12 × 64 × 64 grid - approximate boundaries).

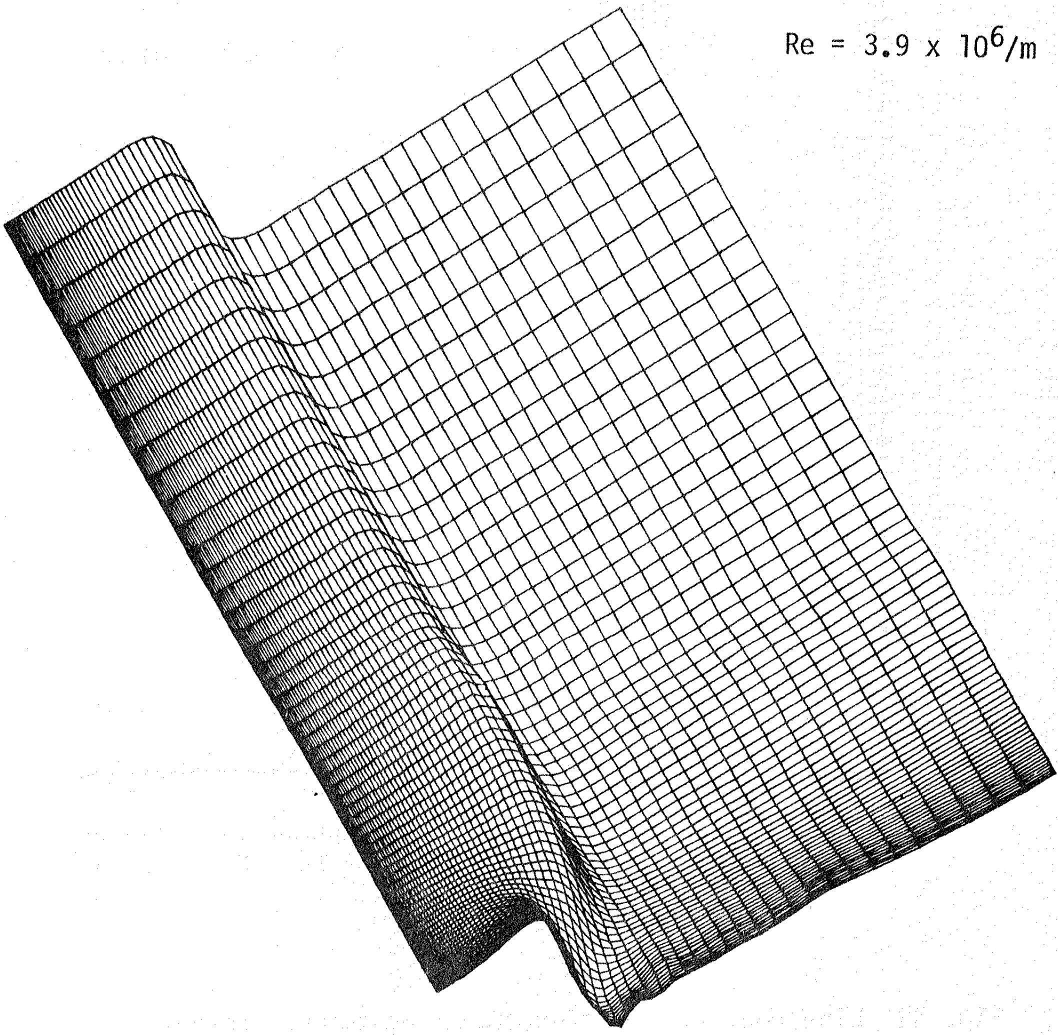


Fig. 42 Perspective view of distribution of density for 12.2^o assymmetric corner x/x_L = .8 (12 × 64 × 64 grid - approximate boundaries).

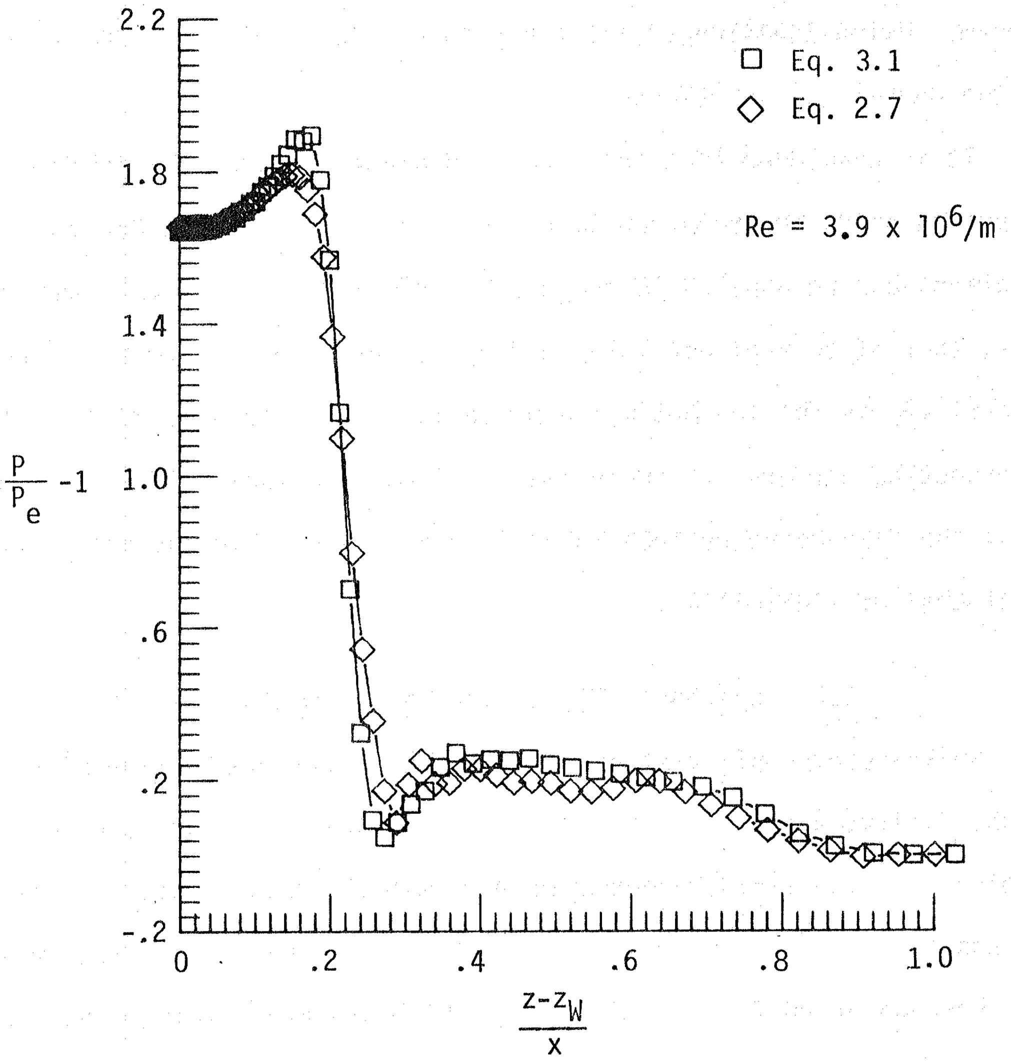


Fig. 43 Comparison of surface pressure for two different grids.

attributed to the growth of the transverse mesh distribution in the windward direction. Using Equation (3.1) there is a very fine grid upstream and the grid becomes coarser with the expanding flow downstream. Using Equation (2.17) the grid is coarse upstream and is finer at the downstream positions.

It is concluded that the "two-boundary technique" in viable for computing grids about three-dimensional corners where the Navier-Stokes equations can be applied to compute supersonic flow. It is noted, however, that if only planar intersecting corners are of interest, Equation (2.17) is not the optimal application of the technique. Planar intersecting corners are approximated using Equation (2.17). Nevertheless, the "two-boundary technique" is highly versatile and affords a great deal of flexibility.

4.2 Supersonic Flow About Spike-Nosed Bodies

Spike-nosed configuration occurs in many supersonic flow situations. Unlike the three-dimensional corner flows that are studied herein, the flow fields about spike-nosed bodies can be highly unsteady. The unsteadiness manifests itself as self-sustained oscillations which have been observed for a wide variety of shear-layer impingement configurations [49]. Two spike-nosed bodies are examined in this section: (1) a body with a small nose length to should height ratio (0.71); and (2) a body with a large nose length to shoulder height ratio (2.14). These two configurations have been examined by Shang, Hankey, and Smith [50]. The first two authors posed the problem and this author applied the "two-boundary technique" to generate the grids and obtained the initial solutions with the Navier-Stokes solver. The emphasis herein is on the application of the "two-boundary technique" for grid generation about the spike-nosed bodies. The "two-boundary technique" has been used to generate grids using a linear approximation to the body surface, a circular arc outer boundary, and a linear connecting function. A parabolic algebraic function and an exponential function are used to concentrate the grid in the nose shoulder region. The details of the application of the "two-boundary technique" to these geometries are developed in Chapter 2 along with the derivatives for the transformation data. The three-dimensional Navier-Stokes solver is used to obtain the numerical solutions to the flow fields. For these solutions the z-coordinate direction is the windward direction. An axisymmetric solution is obtained by the rotation of the grid about the z-axis and solving the three-dimensional equations of motion. The argument for this approach rather than developing an axisymmetric solver is the considerable time savings compared to the programming a specific code for this problem. These solutions demonstrate the versatility of the Navier-Stokes solver.

4.2.1 One-Half-Inch Spike-Nosed Body

The spike-nosed body with the small ratio of nose length to shoulder height has a nose length of 12.5 mm (0.5 in.) and a shoulder height of 19.05 mm (0.75 in.). The boundary surface for this body is shown in Figure 9 and the grid generated with the "two-boundary technique" is shown in Figure 11. The data used to generate the grid is found in Table 2. Figure 44 shows the density distribution of the developing flow. A relatively low density region compared to the surroundings develops at the top of the nose and later sheds off the shoulder. After this point only low amplitude oscillations occur in the nose-shoulder region. Plots of pressure along three lines of the grid (Fig. 45) are shown after 4000 time steps in Figures 46-48.

Two modifications to the data for generating the grid for the one-half-inch spike-nosed body are performed and the solutions recomputed. The objectives of the modifications are to assure the validity of the grid generation technique.

The first modification is a change in the concentration of grid points. The new grid is obtained by making the constant $k = 3$ instead of 2.2 in Table 2. This means that there are fewer grid points to define the shock in the radial direction and more points near the inner boundary. Figures 49-51 show the pressure comparison with the original case at the 4000th time step for the three lines. The second modification moves the outer boundary closer to the inner boundary concentrating more points in the shock region. The radius of the circular arc defining the outer is changed from 13.26 mm (0.522 ft) to 12.55 mm (0.494 ft) and the constant k is set equal to 1.75 so that the distance of the nearest grid point to the inner boundary is approximately the same as that in the original case. Figures 52-54 show the density comparison with the original case and the first modification. It is seen from the plots that there is overall good agreement. The differences that exist

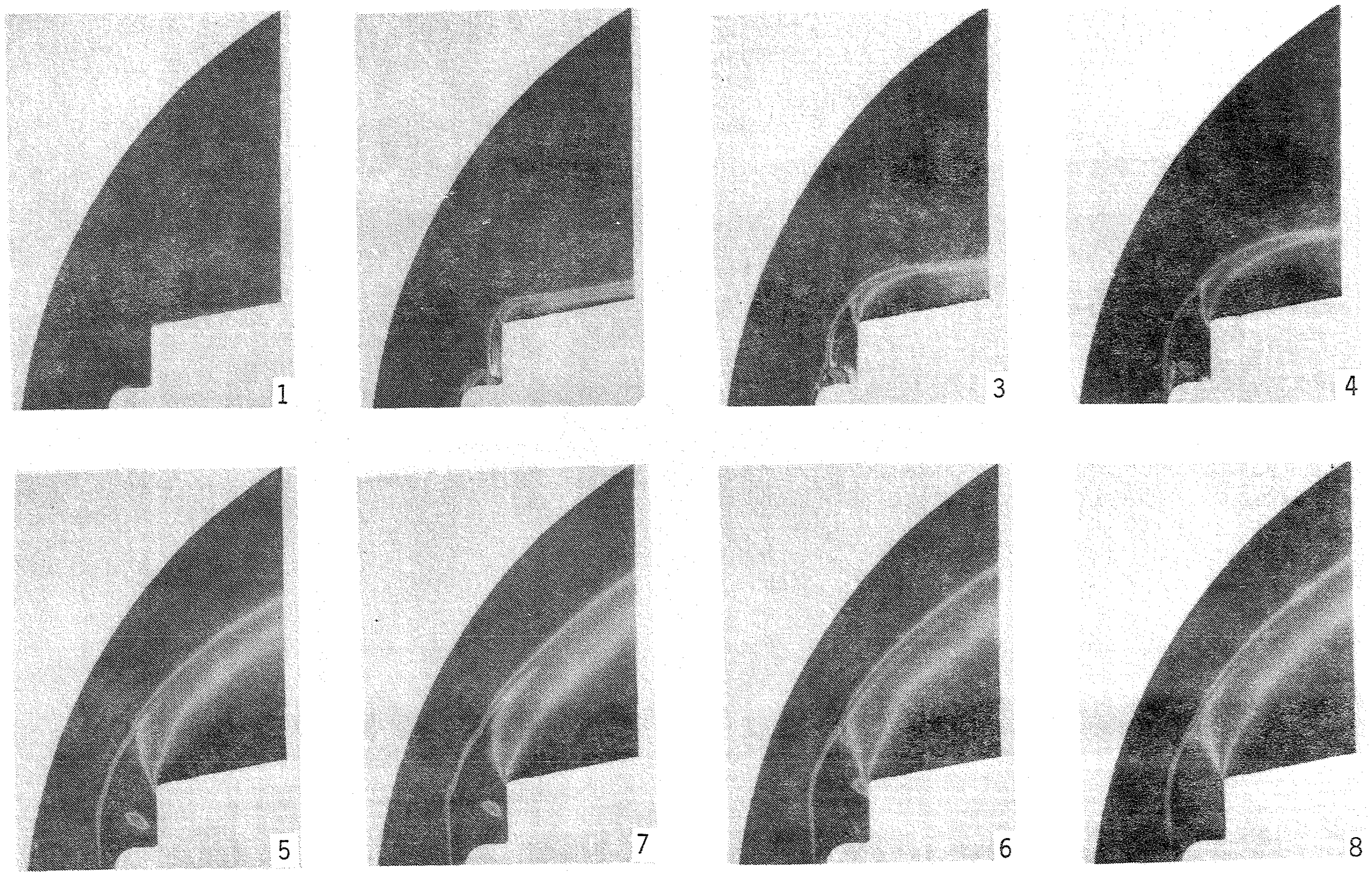


Fig. 44 Development of density solution for one-half inch spike-nosed body.

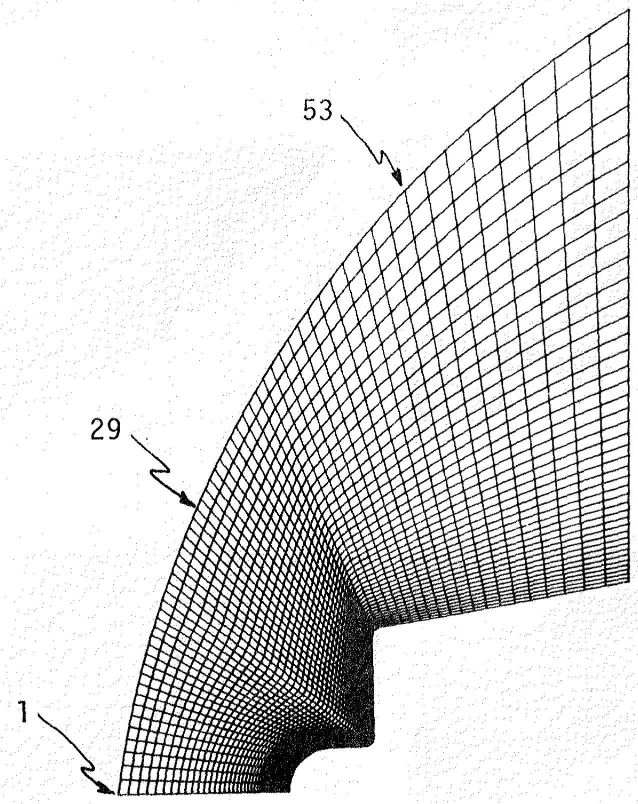


Fig. 45 Grid lines for quantative comparison one-half inch spike-nosed body.

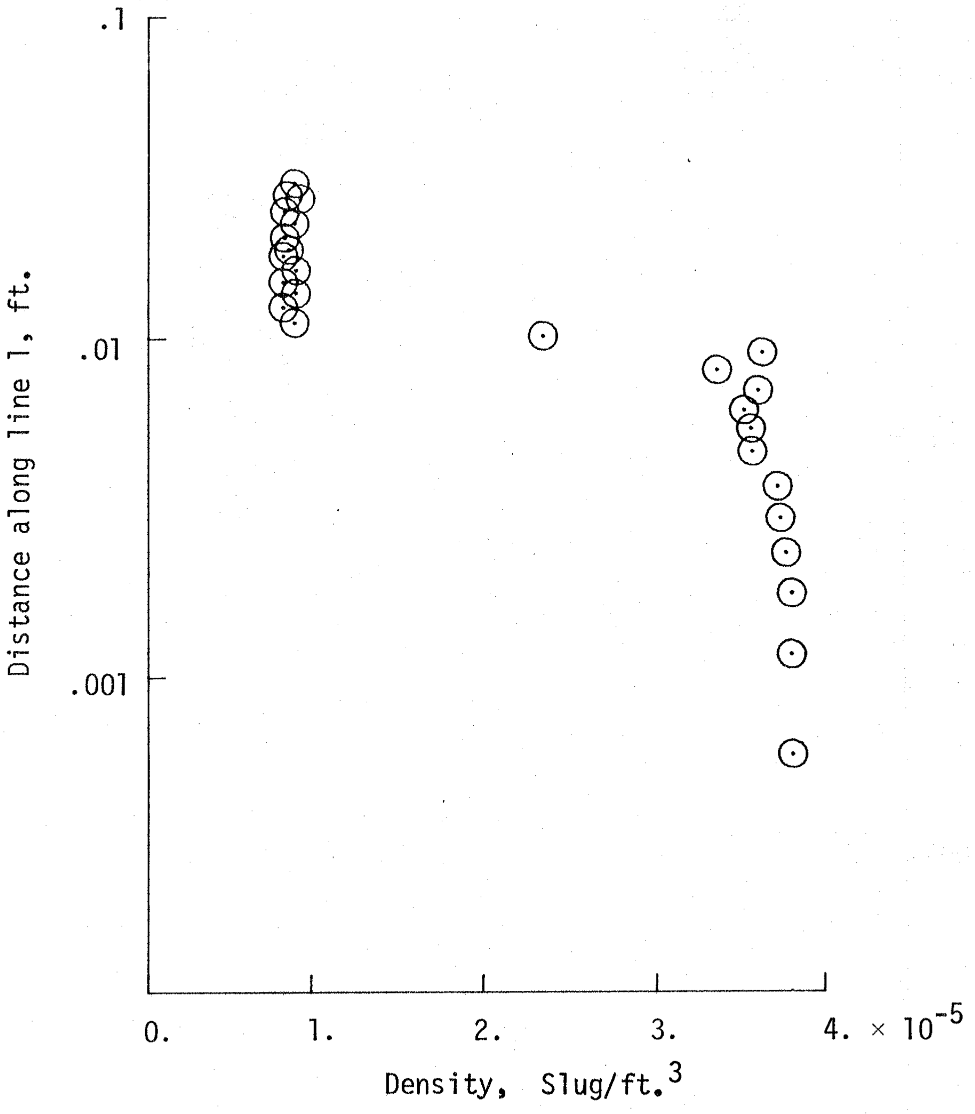


Fig. 46 Density along line one for base case.

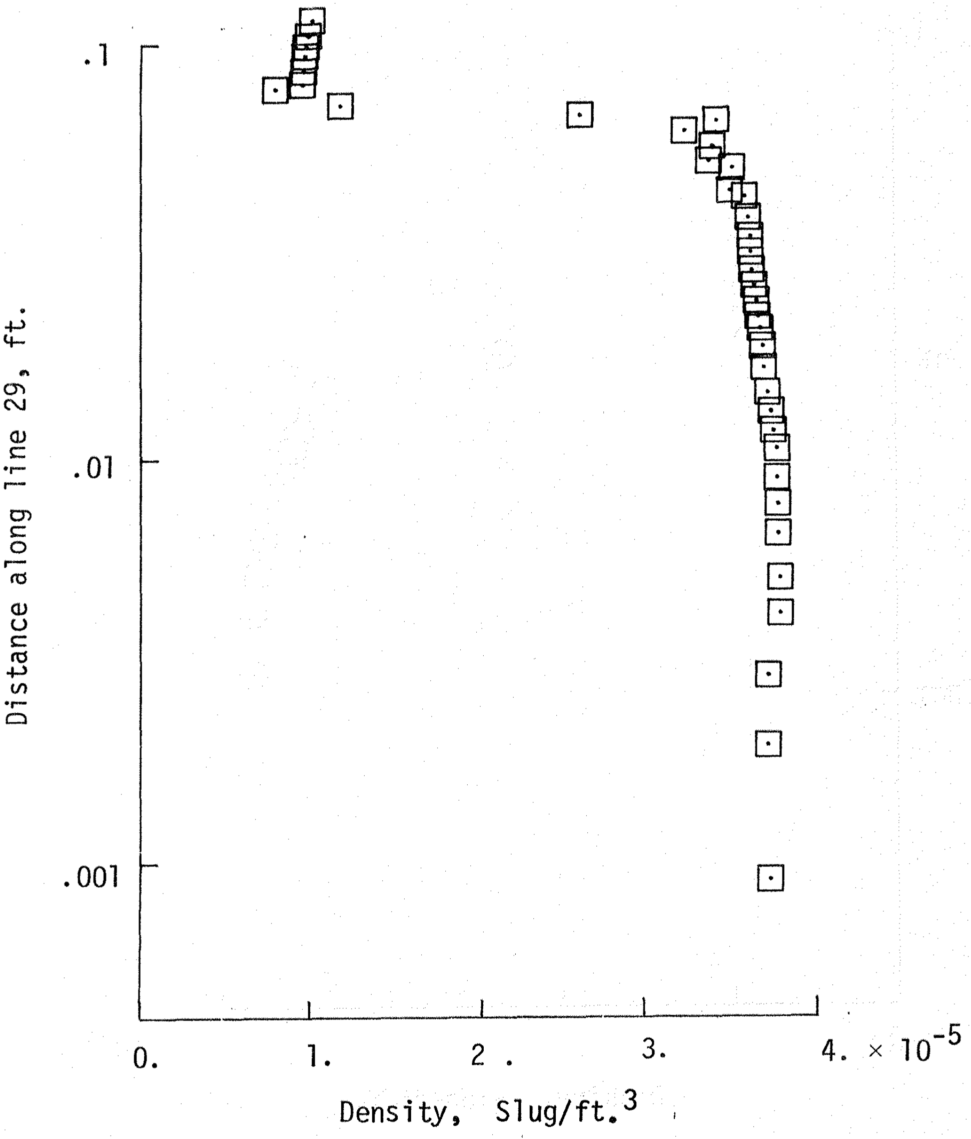


Fig. 47 Density along line twenty-nine for base case.

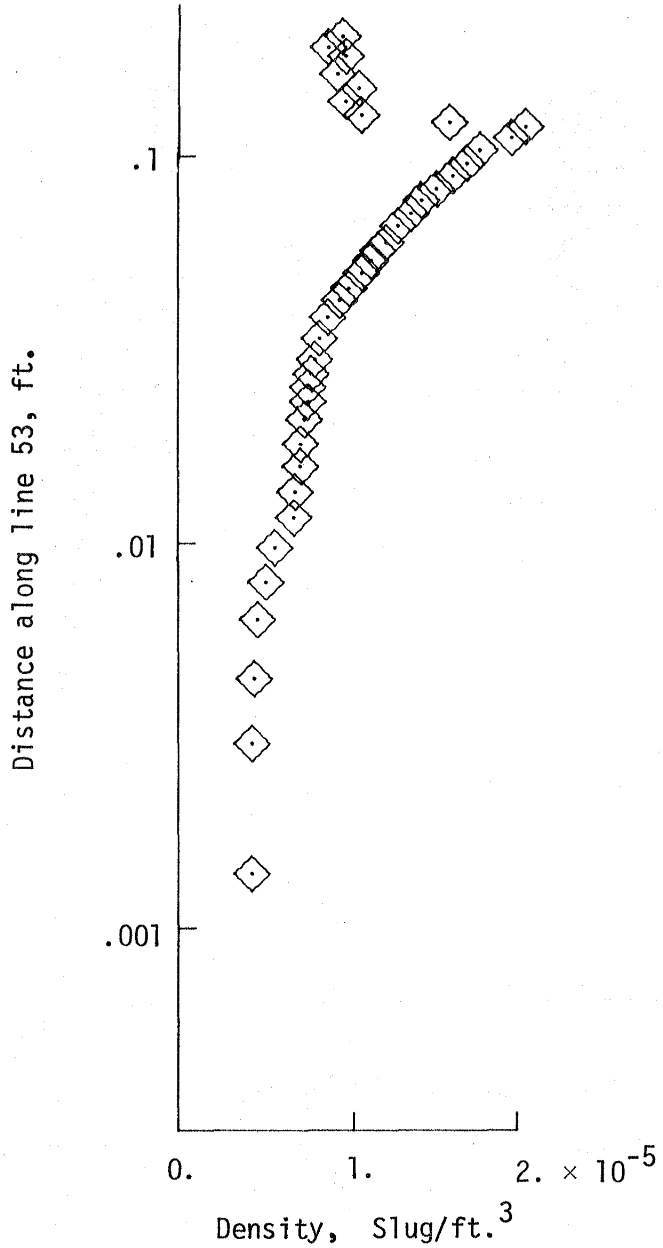


Fig. 48 Density along line fifty-three for base case.

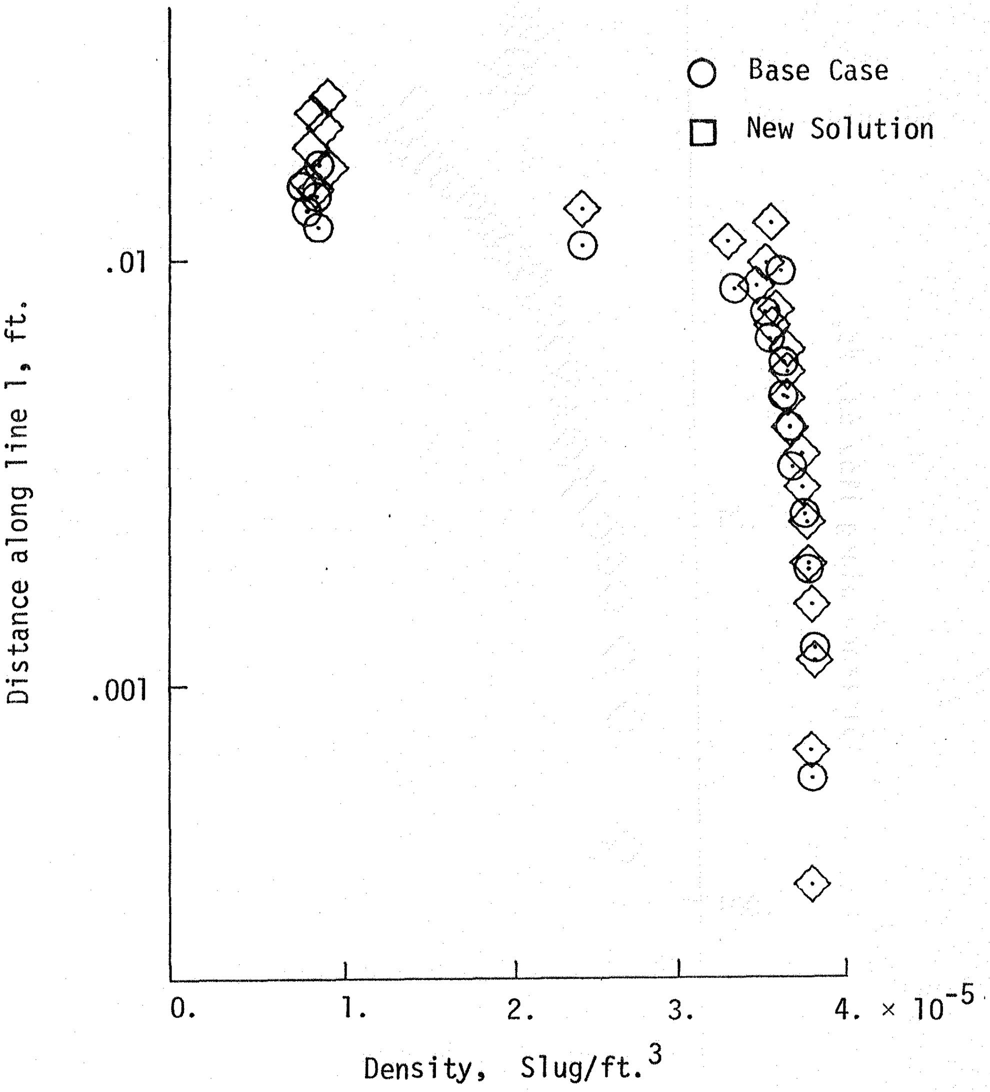


Fig. 49 Comparison of density solution for grid concentration change line = 1.

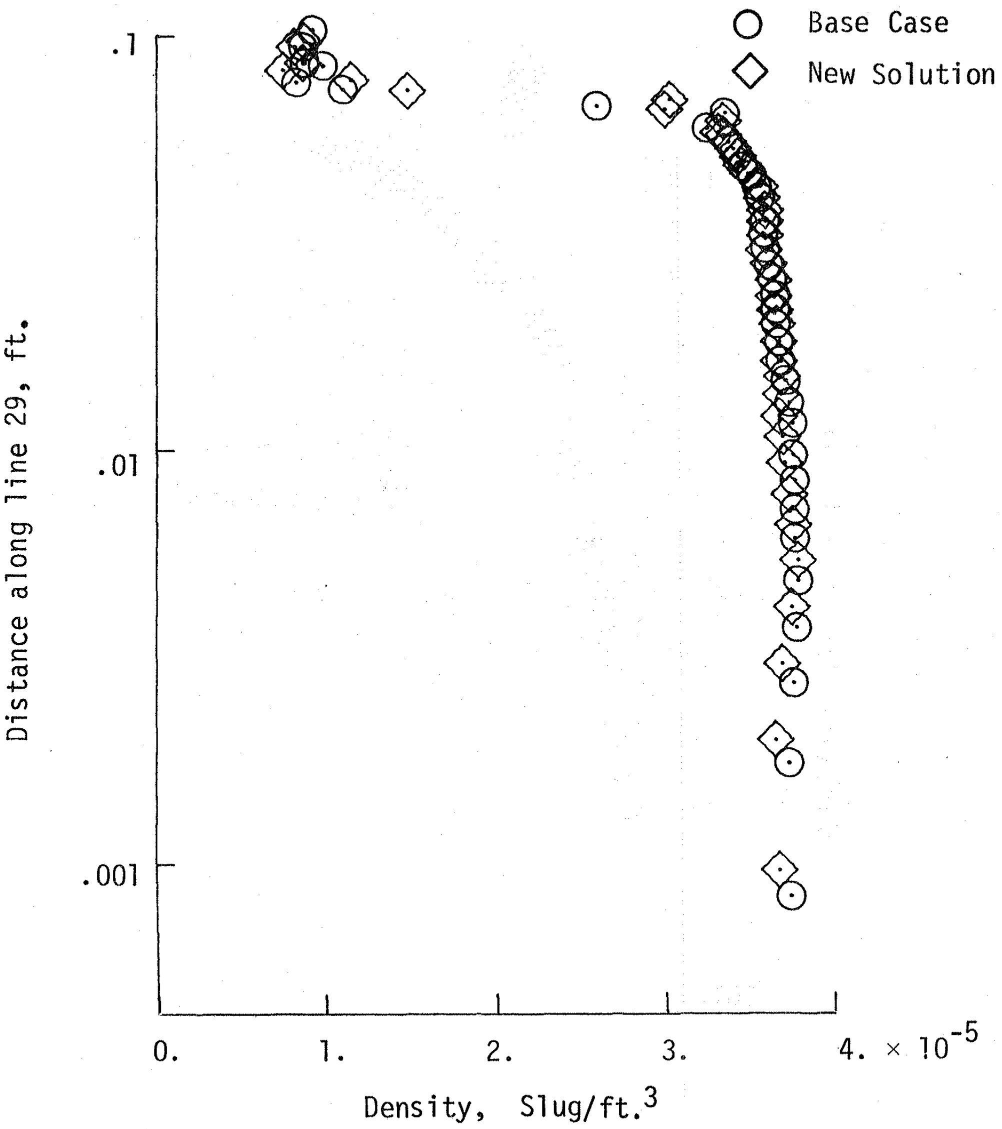


Fig. 50 Comparison of density solution for grid concentration change line = 29.

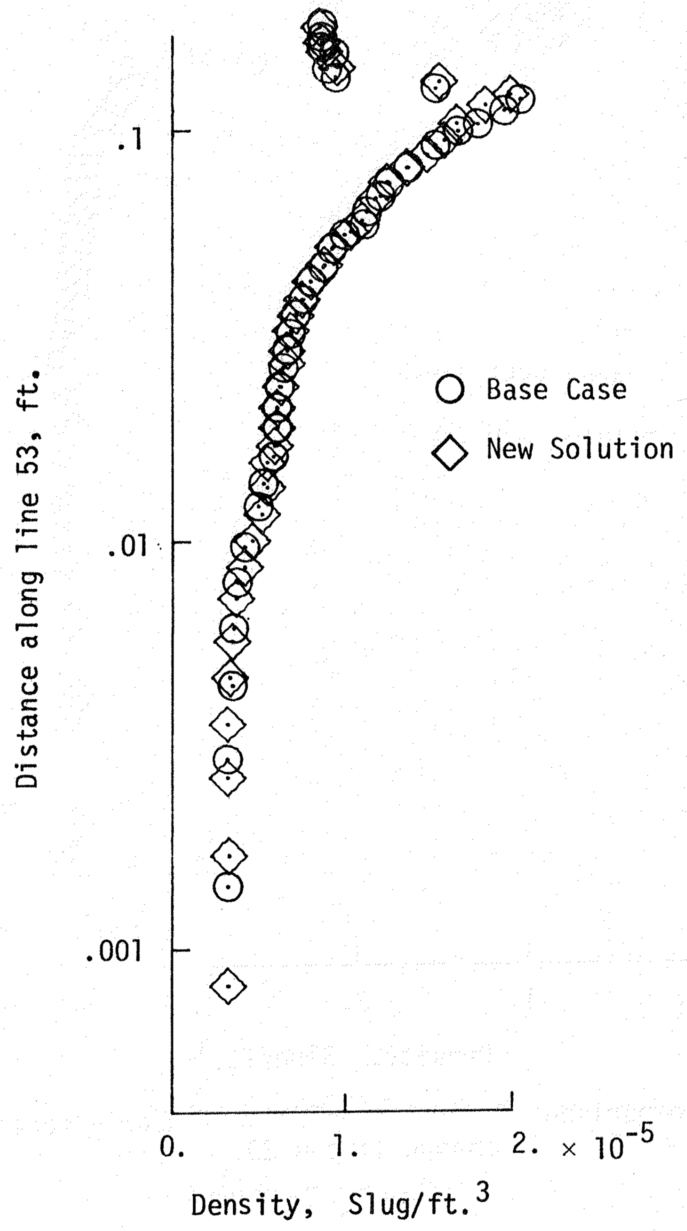


Fig. 51 Comparison of density solution for grid concentration change line = 53.

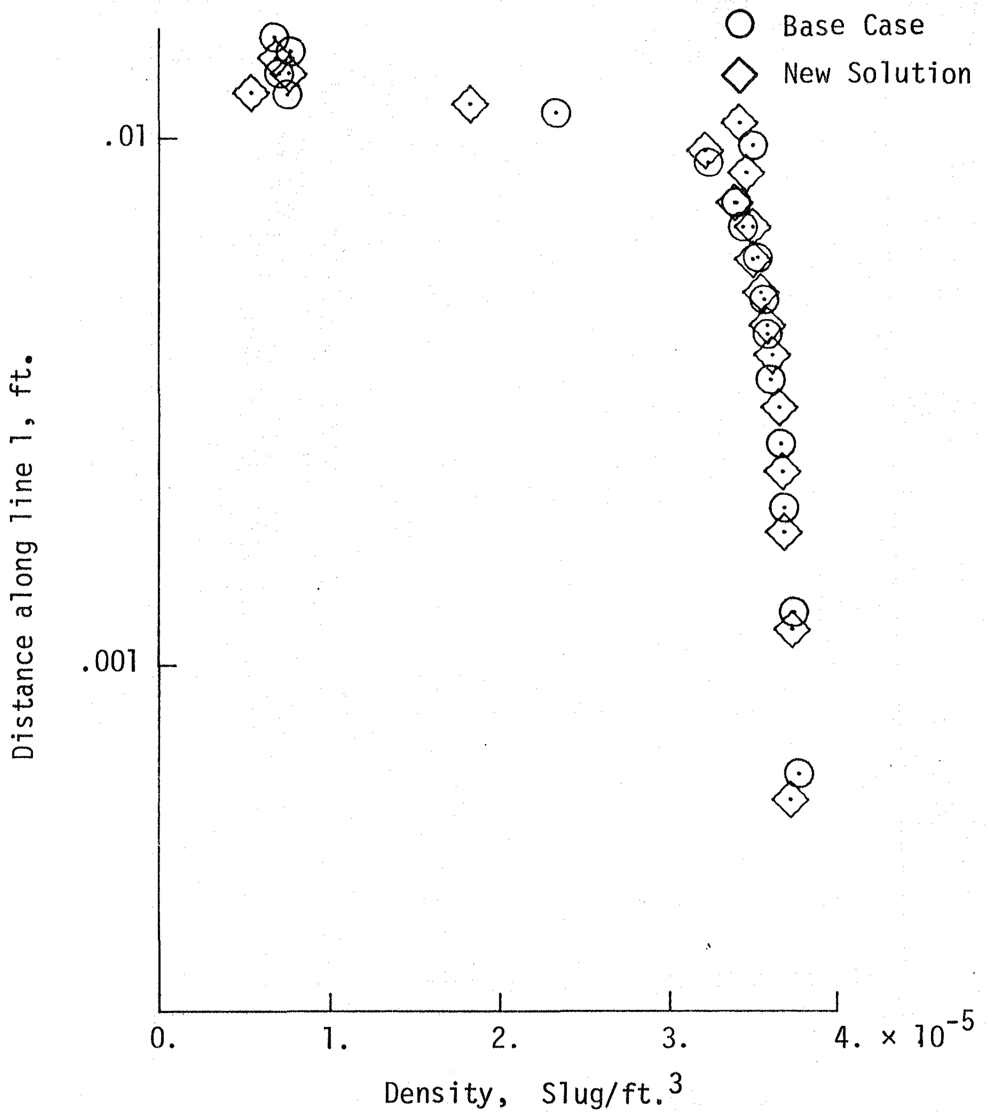


Fig. 52 Comparison of density solution for outer boundary change 1.

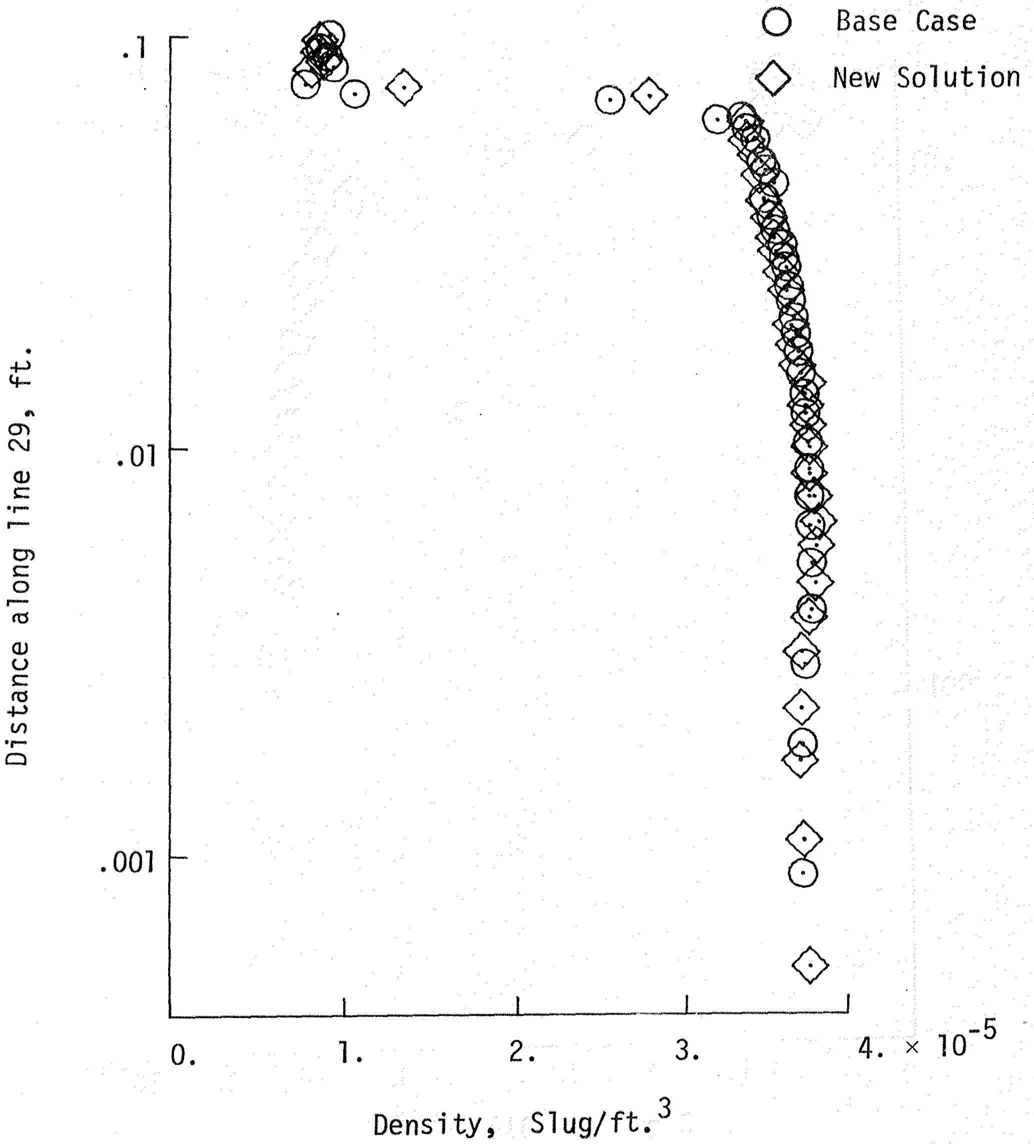


Fig. 53 Comparison of density solution for outer boundary change line = 29.

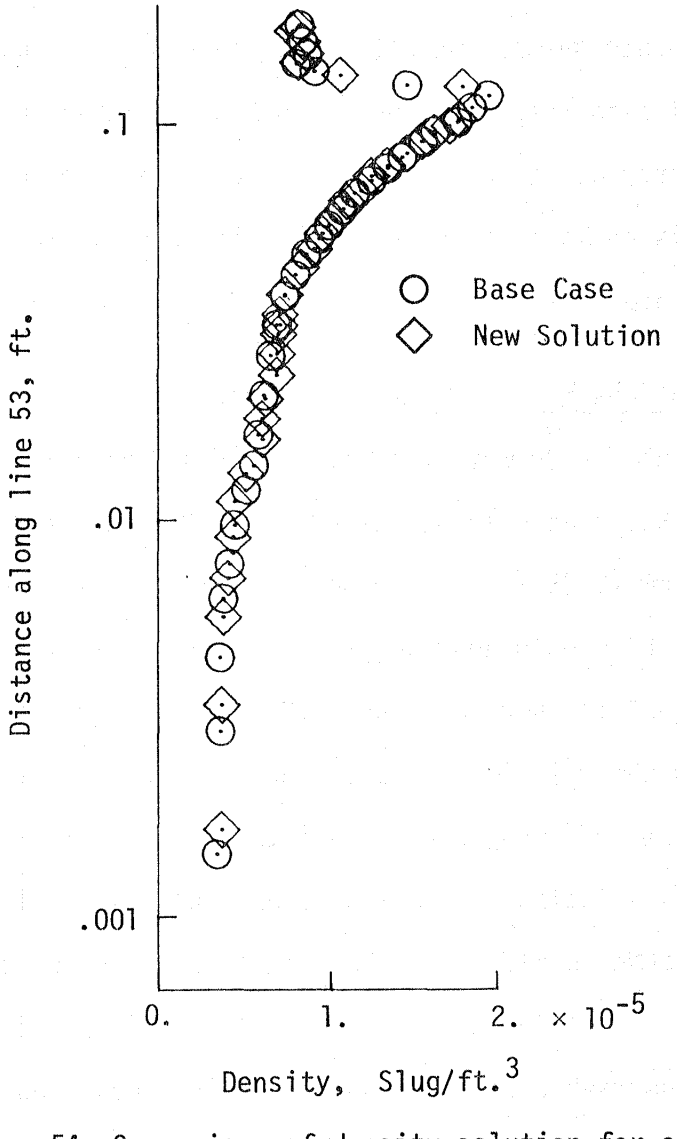


Fig. 54 Comparison of density solution for outer boundary change line = 53.

are due to the grid spacing in the shock region and inner boundary region. The shock is more smeared when there are fewer grid points in that region and the placing of grids points closer to the inner boundary affects the accuracy there. No attempt is made here to analyze the low amplitude oscillation for the one-half-inch spike-nosed body. It is seen that the perturbed grids produce good agreement with the original case and, this is an indication that the grid generation technique is viable for this application.

4.2.2 One and One-Half-Inch Spike-Nosed Body

The spike-nosed body with the large ratio of nose length to shoulder height has a nose length of 38.1 mm (1.5 in.) and a shoulder height of 19.05 mm (0.75 in.). The body surface for this configuration is shown in Figure 10 and the grid generated with the "two-boundary technique" is shown in Figure 12. Table 3 contains the data used to generate the grid. Table 3 is the same as Table 2 except the nose length is one inch longer and the outside circular arc is longer. The Navier-Stokes solution for this grid and initial conditions produces high amplitude oscillation. The characteristics of the oscillation can be seen from experimental observation. Figure 55 shows a sequence of shadowgraph pictures [51] of a spiked-nosed body flow field in one cycle of oscillation. After the initial transient, the bow shock interact with the shock from the shoulder near the top of the shoulder. A strong reverse flow occurs between the shock and the boundary forcing the shock to bulge out. With the shock bulged out,

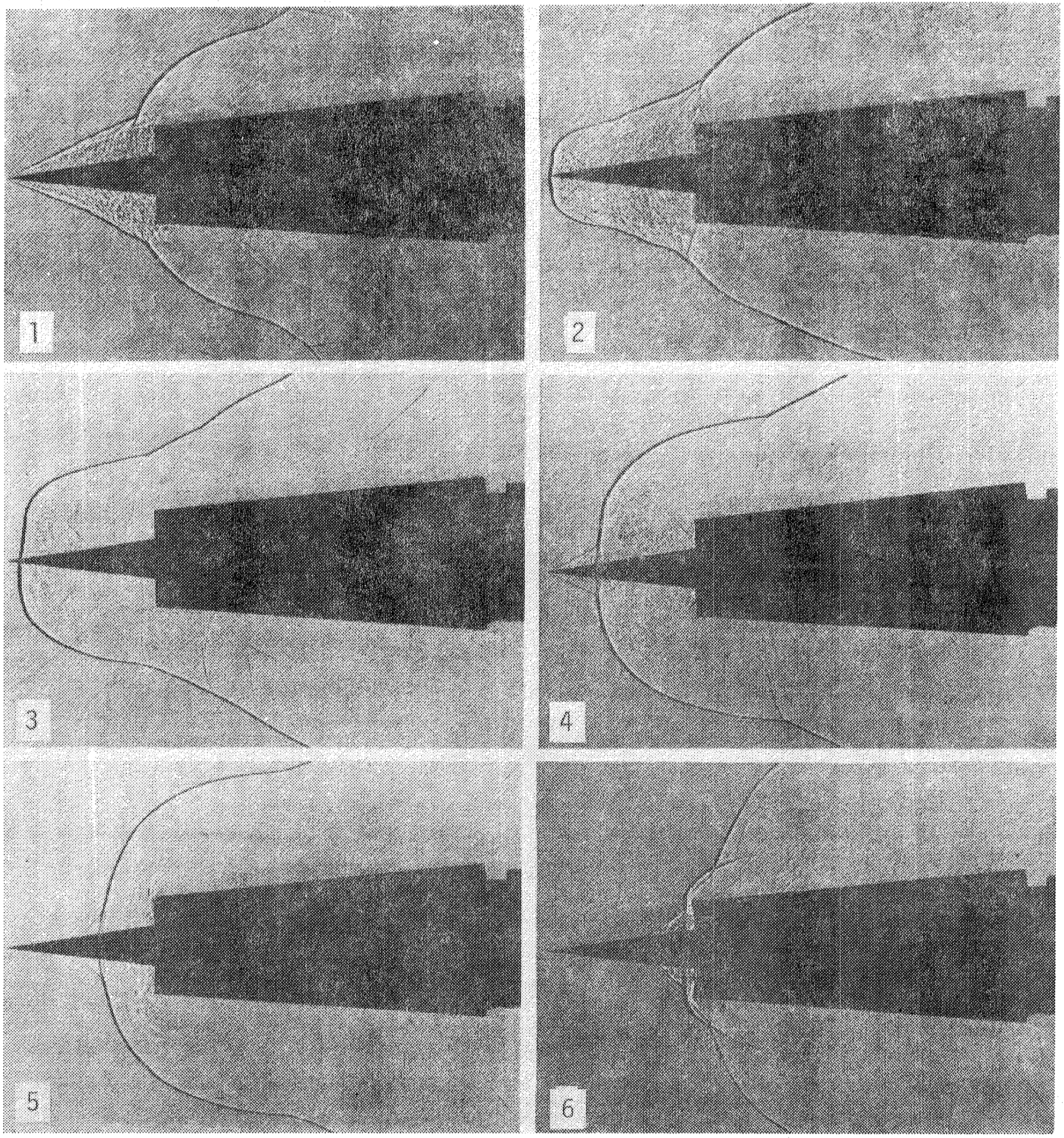


Fig. 55 Shadowgraphs of oscillating flow field.

the high energy fluid behind the shock along with shock is forced downstream by the outer flow until the bow shock again interacts with shoulder shock. The cycle repeats itself. This phenomena is also shown by Harney [52] for blunt nosed bodies which is approximated by the one and one-half-inch spike-nosed body. Widhopf [53] has shown the oscillating phenomena over indented nose tips while numerically solving the Navier-Stokes equations. Figure 56 shows a sequence of density distributions for one cycle taken from the numerical solution. The phenomenon described above is observed in the numerical solution of the flow about the one and one-half-inch spike-nose body. The cycle repeats itself and maintains its form. The oscillation phenomenon is independent of Reynolds number [50] but is dependent on the speed of sound which in turn depends on the free stream temperature. The primary frequency is 3100 cycles/sec where the free temperature is 217 K (290 R). In [51] using a Reynolds number of 7.78 × 10^6/m and free stream temperature of 111 K (200 R) a lower primary frequency of 2665 cycles/sec is observed. The pressure at approximately point 29 (Fig. 48) and the observed pressure from an experiment conducted at the Wright-Patterson Flight Dynamics Laboratory are shown in Figure 57. It is seen in the figure that the wave form and frequency from the experiment and the numerical solution are in good agreement.

The "two-boundary technique" for grid generation has been successfully applied for two spike-nosed bodies. The Navier-Stokes solver has produced unsteady numerical solutions for several initial conditions

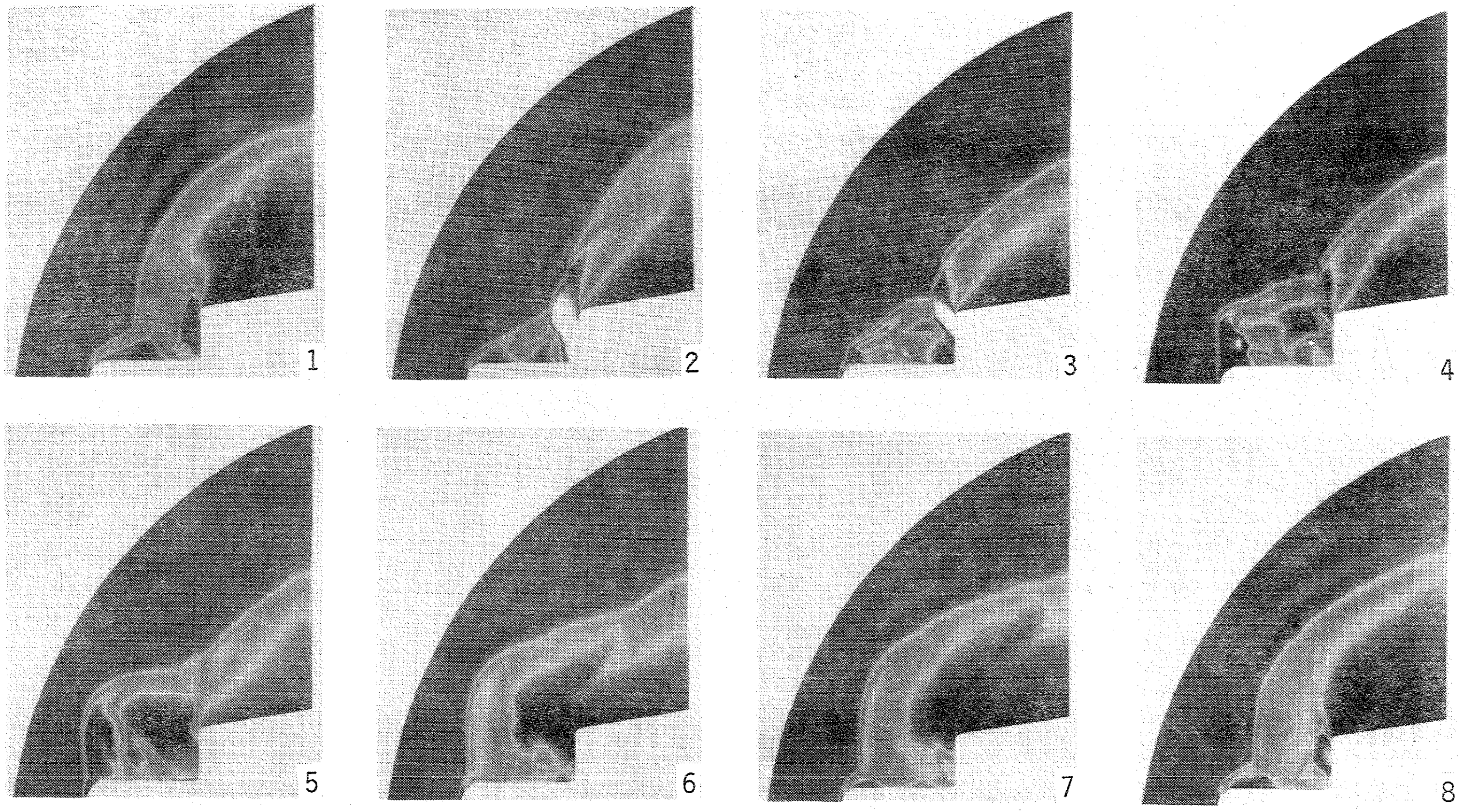


Fig. 56 Density distribution during one cycle of oscillation for one and one-half inch spike-nosed body.

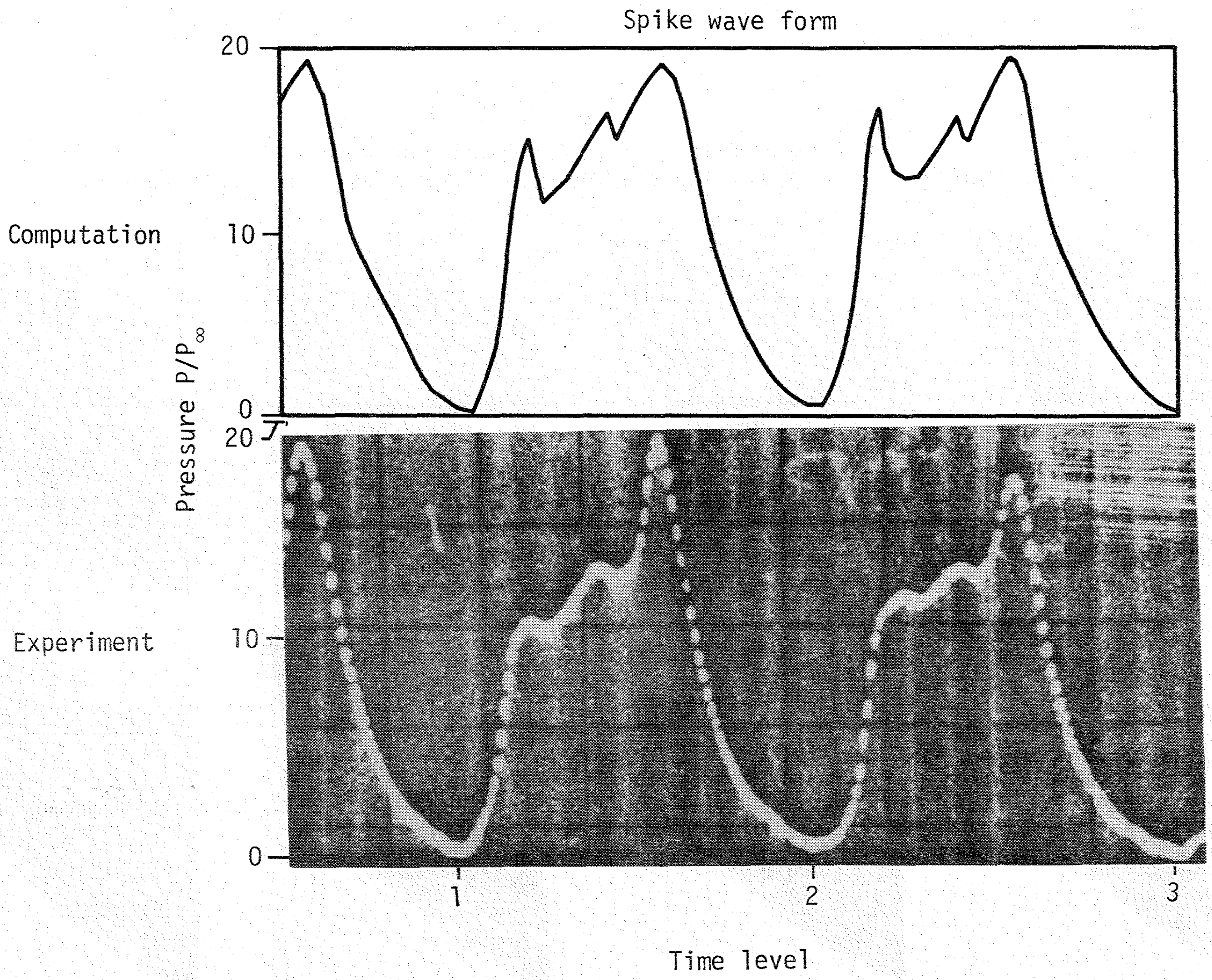


Figure 57.- Surface pressure on one and one-half inch spike-nosed body.

with Reynolds numbers up to 7.78 × 10^6/m. Overall, the solutions that have been obtained simulate the observed phenomena very well.

5. CONCLUSIONS

An algebraic grid generation technique has been developed and explored in conjunction with the solution of the compressible three-dimensional Navier-Stokes equations. The technique called the "two-boundary technique" is simple to understand, easy to apply, and has a high degree of generality for the finite difference solution of complex flow field problems. The "two-boundary technique" allows direct control of a grid and direct computation of the Jacobian derivatives.

The viability of the grid generation technique is demonstrated through the development and application of a Navier-Stokes solver which operates on the CDC CYBER-203 vector computer. The computer program is based on a MacCormack time-split technique which is chosen because of its compatibility with vector computer architecture. The finite difference algorithm is written in the SL/1 programming language, and the 32-bit word length arithmetic and storage option is used. This option doubles the number of grid points that can be used for a given amount of memory and approximately doubles the computational rate as compared to the normal 64-bit words. Using SL/1 and the halfword option the computational rate is 4×10^{-5} seconds per grid point per time step, and solutions with 5×10^4 grid points can be obtained without using secondary memory. It is concluded from the numerical experiments presented in the present study that the 32-bit word length is adequate when solving the Navier-Stokes equations for supersonic laminar flow using an explicit MacCormack technique.

Complex supersonic flow field solutions are obtained for two distinctly different geometries using the "two-boundary technique" for grid generation and the Navier-Stokes solver. First, supersonic flow solutions about a family of three-dimensional corners are obtained. These flow fields reach a steady state but are characterized by strong shocks and three-dimensional separation. The Mach number is 3.64, Reynolds numbers are 2.72×10^5/m and 3.9×10^6/m, and the fluid properties are for air. It is shown that the solutions obtained agree well with physical experiments and other numerical experiments. Also, corner flow solutions with 5×10^4 grid points are among the most refined Navier-Stokes solutions obtained to date. The second flow situation is supersonic flow about spike-nosed bodies. In this case, the flow is axisymmetric, unsteady, and characterized by a strong bow shock and massive separation. The Mach number is 3 and the Renolds number is 7.78×10^6/m. The numerical solutions show dramatically the oscillating flow generated by the interaction of the bow shock and shoulder wall of the body. The surface pressure and oscillation frequency compare very well with corresponding wind tunnel experiments. The successful numerical solution of the flow fields support the primary conclusion that the "two-boundary technique" is viable for generating grids for complex flow field solutions. Also, for the spike-nosed bodies, considerable development time for a specialized axisymmetric code is saved.

Plans for the use of the "two-boundary technique" include development of grids with wing-fuselage boundaries, analysis of non-orthogonal grids, development of additional spike-nosed body grids, and the development of numerical grid control functions.

REFERENCES

1. MacCormack, R. W.; and Lomax, H.: "Numerical Solution of Compressible Viscous Flows." Annual Review Fluid Mechanics, 1979, Vol. 11, pp. 238-316, Annual Reviews, Inc.

2. MacCormack, R. W.; and Paullay, A. J.: "Computational Efficiency Achieved by Time Splitting of Finite Difference Operators." AIAA paper 72-154, Jan. 1972.

3. Shang, J. S.; and Hankey, W. L.: "Numerical Solution of the Navier-Stokes Equations for a Three-Dimensional Corner." AIAA paper 77-169, Los Angeles, CA, also AIAA Journal, Vol. 15, Nov. 1977, pp. 1575-82.

4. Shang, J. S.; Hankey, W. L.; and Petty, J. S.: "Three-Dimensional Supersonic Interacting Turbulent Flow Along a Corner." AIAA 78-1210, Seattle, WA, July 1978.

5. Control Data Corporation: "Control Data CYBER-200 Model 203 Computer Hardware Reference Manual." Publication Number 60256010, May 1979.

6. Thompson, J. F.; Thames, F. C.; and Mastin, C. W.: "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate Systems for Fields Containing Any Number of Arbitrary Two-Dimensional Bodies." Journal of Computational Physics, Vol. 15, July 1974, pp. 299-319.

7. Thames, F. C.; Thompson, J. F.; Mastin, C. W.; and Walker, R. L.: "Numerical Solutions for Viscous and Potential Flow About Arbitrary Two-Dimensional Bodies Using Body-Fitted Coordinate Systems." Journal of Computational Physics, Vol. 24, 245, July 1977, pp. 245-273.

8. Thompson, J. F.; Thames, F. C.; and Shanks, S. P.: "Use of Numerically Generated Body-Fitted Coordinate Systems for Solutions of the Navier-Stokes Equations." Proceeding of AIAA 2nd Computer Fluid Dynamics Conference, Hartford, CT, July 1975.

9. Thompson, J. F.; Thames, F. C.; and Mastin, C. W.: "Boundary-Fitted Curvilinear Coordinate Systems for Solution of Partial Differential Equations on Field Containing Any Number of Arbitrary Two-Dimensional Bodies." NASA CR-2739, July 1977.

10. Mastin, C. W.; and Thompson, J. F.: "Elliptic Systems and Numerical Transformations." Journal of Mathematical Analysis and Applications, Vol. 62, Jan. 1978, pp. 52-62.

11. Coon, S. A.: "Surfaces for Computer-Aided Design of Space Forms." MAC TR-41 (Contract No. AF-33(6000-42859) MIT, June 1967. Available from DDC as AD 663504.

12. Gordon, W. J.: "Free-Form Surface Interpolation Through Curved Networks." General Motors Research Report (GMR 921), Sept. 1969.

13. Gordon, W. J.; and Hall, C.: "Construction of Curvilinear Coordinate Systems and Applications to Mesh Generation." International Journal for Numerical Methods In Engineering, Vol. 7, July 1973, pp. 461-477.

14. Eiseman, P. R.: "Three-Dimensional Coordinates About Wings." Proceeding of the AIAA 4th Computer Fluid Dynamic Conference, Williamsburg, VA, July 1979.

15. Eiseman, P. R.: "A Multi-Surface Method of Coordinate Generation." Journal of Computational Physics, Vol. 33, Oct. 1979, pp. 118-150.

16. Eiseman, P. R.: "Geometric Methods in Computational Fluid Dynamics." ICASE Report 80-11, April 1980.

17. Schwiekert, D.: "An Interpolation Curve Using Splines in Tension." Journal of Mathematics and Physics, Vol. 45, Sept. 1966, pp. 312-317.

18. Cline, A. K.: "Scaler- and Planar-Valued Curve Fitting Using Splines Under Tension." Communications of the ACM, Vol. 17, No. 4, April 1974, pp. 218-233.

19. Pruess, S.: "Properties of Splines in Tension." Journal of Approximation Theory, Vol. 17, Aug. 1970, pp. 86-96.

20. SL/1 Reference Manual, Analysis and Computation Division, NASA Langley Research Center, Hampton, VA.

21. Schlichting, H.: Boundary Layer Theory, McGraw-Hill, 1968.

22. Vincenti, W. G.; Kruger, C. H.: Introduction to Physical Gas Dynamics, John Wiley & Son, Inc., 1965.

23. Smith, R. E.; and Weigel, B. L.: "Analytic and Approximate Boundary Fitted Coordinate Systems for Fluid Flow Similation." AIAA Paper 80-0192, Pasadena, CA, Jan. 1980.

24. Dagundji, J.: Topology, Allyn and Bacon, 1968.

25. Ahlberg, J. H.; Nilson, E. N.; and Walsh, S. L.: The Theory of Splines and Their Applications, Academic Press, Inc., 1967.

26. MacCormack, R. W.: "The Effect of Viscosity in Hypervelocity Impact Cratering." AIAA Paper 69-354, May 1969.

27. Roach, P. J.: "Computational Fluid Dynamics." Hermosa Publishers, 1972.

28. Holst, T. L.: "Numerical Solution of Axisymmetric Boattail Flow Fields with Plume Simulators." AIAA Paper 77-224, Jan. 1977.

29. MacCormack, R. W.: "An Efficient Numerical Method for Solving the Time-Dependent Compressible Navier-Stokes Equations at High Reynolds Number." Computing in Applied Mechanics, ADM, Vol. 18, New York Society of Mechanical Engineering, June 1973.

30. Shang, J. S.: "Implicit-Explicit Method for Solving the Navier-Stokes Equations." AIAA Journal, Vol. 16, No. 5, May 1978, pp. 495-502.

31. MacCormack, R. W.; and Baldiwn, B. S.: "A Numerical Method for Solving the Navier-Stokes Equations with Application to Shock Boundary Layer Interactions." AIAA Paper 75-1, Jan. 1975.

32. Smith, R. E.; and Pitts, J. I.: "The Solution of the Three-Dimensional Compressible Navier-Stokes Equations on a Vector Computer." Third IMAC International Symposium on Computer Methods for Partial Differential Equations, Lehigh University, PA, June 1979.

33. Lambiotte, J. J.: "Effects of Virtual Memory on Efficient Solution of Two Model Problems." NASA TM X-3512, July 1977.

34. Charwat, A. F.; and Redekopp, L. G.: "Supersonic Interference Flow Along the Corner of Intersecting Wedges." Men. RM-4863-PR (Contract No. AF49(638)-1700), RAND Corp., July 1966.

35. Stainback, P. C.: "An Experimental Investigation at a Mach Number of 4.95 of Flow in the Vicinity of a 90° Interior Corner Alined with the Free-Stream Velocity." NASA TN D-184, Feb. 1960.

36. Stainback, P. C.: "Heat-Transfer Measurements at a Mach Number of 8 in the Vicinity of a 90° Interior Corner Alined with the Free-Stream Velocity." NASA TN D-2417, Aug. 1964.

37. Watson, R. D.: "Experimental Study of Sharp- and Blunt-Nose Streamwise Corners at Mach 20." NASA TN D-7398, April 1974.

38. Cooper, J. R.; and Hankey, W. L.: "Flow Field Measurements in an Asymmetric Axial Corner at $M = 12.5$." AIAA Journal, Vol. 12, Oct. 1974, pp. 1353-1357.

39. Kulter, P.: "Numerical Solution for the Inviscid Supersonic Flow in the Corner Formed by Two Intersecting Wedges." AIAA Paper 73-675, Palm Springs, CA, July 1973.

40. Shankar, V.; Anderson, D.; and Kulter, P.: "Numerical Solutions for Supersonic Corner Flow." Journal of Computational Physics, Vol. 17, Oct. 1975, pp. 160-180.

41. Marconi, F.: "Supersonic, Inviscid, Conical Corner Flowfields." AIAA Journal, Vol. 18, Jan. 1980, pp. 78-84.

42. Weinberg, B.; and Rubin, S.: "Compression Corner Flow." Journal of Fluid Mechanics, Vol. 56, Part 4, May 1975, pp. 753-774.

43. Ghia, K.; and Davis, R.: "A Study Compressible Potential and Asymptotic Viscous Flows for Corner Regions." AIAA Journal, Vol. 12, March 1974, pp. 355-359.

44. Hung, C.; and MacCormack, R.: "Numerical Solution of Supersonic Laminar Flow Over a Three-Dimensional Compression Corner." AIAA Paper 77-694, June 1977.

45. Hung, C.; and MacCormack, R.: "Numerical Solution of Three-Dimensional Shock Wave and Turbulent Boundary Layer Interactions." AIAA Journal, Vol. 16, Oct. 1978, pp. 1090-1096.

46. Horstman, C.; and Hung, C.: "Computation of Three-Dimensional Turbulent Separated Flows at Supersonic Speeds." AIAA Paper 79-0002, Jan. 1979.

47. Smith, R. E.: "Numerical Solutions of the Navier-Stokes Equations for a Family of Three-Dimensional Corner Geometries." AIAA Paper 80-1349, July 1980.

48. Korkegi, R.: "On the Structure of Three-Dimensional Shock-Induced Separated Flow Regions." AIAA Journal, Vol. 14, No. 5, May 1976, pp. 597-600.

49. Rockwell, D.; and Nandascher, E.: "Self-Sustained Oscillations of Impinging Free Shear Layers." Annual Review of Fluid Mechanics, Vol. II, 1979, pp. 67-94, Annual Review, Inc., Palo Alto, CA.

50. Shang, J. S.; Hankey, W. L.; and Smith, R. E.: "Flow Oscillations of Spike-Tipped Bodies." AIAA Paper 80-0068, Jan. 1980.

51. Butz, J. S.: "Hypersonic Aircraft Will Face Technical Cost Problems." Aviation Week Including Space Technology, Vol. 70, No. 25, June 1959, pp. 156-169.

52. Harney, D. J.: "Oscillating Shocks on Spike Nose Tips at Mach 3." AFFDL-TM-79-9-FX, Air Force Flight Dynamics Laboratory, WRAFB, Ohio, 1979.

53. Widhopf, G. F.; and Voctoria, K. J.: "Numerical Solution of the Unsteady Navier-Stokes Equations for the Oscillatory Flow Over a Concave Body." Lecture Notes in Physics, No. 35, June 1974, Springler-Verlag, pp. 431-444.

1. Report No. NASA TM-83123	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle
Two-Boundary Grid Generation for the Solution of the Three-Dimensional Compressible Navier-Stokes Equations

5. Report Date
May 1981

6. Performing Organization Code
505-31-83-02

7. Author(s)
R. E. Smith

8. Performing Organization Report No.

9. Performing Organization Name and Address

NASA Langley Research Center
Hampton, VA 23665

10. Work Unit No.

11. Contract or Grant No.

13. Type of Report and Period Covered
Technical Memorandum

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, DC 20546

14. Sponsoring Agency Code

15. Supplementary Notes
This report is a dissertation submitted to Old Dominion University for partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Mechanical Engineering.

16. Abstract A grid generation technique called the "two-boundary technique" is developed and applied for the solution of the three-dimensional Navier-Stokes equations. The Navier-Stokes equations are transformed from a cartesian coordinate system to a computational coordinate system, and the grid generation technique provided the Jacobian matrix describing the transformation.

The "two-boundary technique" is based on algebraically defining two distinct boundaries of a flow domain and joining these boundaries with either a linear or cubic polynomial. Control of the distribution of the grid is achieved by applying functions to the uniform computational grid which redistribute the computational independent variables and consequently concentrate or disperse the grid points in the physical domain.

The Navier-Stokes equations are solved using a MacCormack time-split technique. The technique is programed for the CYBER-203 computer in the SL/1 language and uses 32-bit word arithmetic. Two distinct flow field problems are solved using the grid generation technique and the Navier-Stokes solver (computer program). Grids and supersonic laminar flow solutions are obtained for a family of three-dimensional corners and two spike-nosed bodies. The "two-boundary technique" is demonstrated to be viable for grid generation associate with supersonic flow. The technique is easy to apply and is applicable to a wide class of geometries.

17. Key Words (Suggested by Author(s))
computational fluids, grid generation, supersonic flow

18. Distribution Statement

Unclassified - Unlimited

Subject Category 64

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 172	22. Price* A08

*For sale by the National Technical Information Service, Springfield, Virginia 22151

End of Document