Summer 1987

Topology and Grid Adaption for High-Speed Flow Computations

Jamshid Samareh Abolhassani
Old Dominion University

TOPOLOGY AND GRID ADAPTION FOR HIGH-SPEED FLOW COMPUTATIONS

by

Jamshid Samareh Abolhassani
B.S. May 1981, West Virginia Institute of Technology
M.E. May 1983, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MECHANICAL ENGINEERING

OLD DOMINION UNIVERSITY
August 1987

Approved by:

Dr. Surendra N. Tiwari (Director)

Dr. Robert E. Smith (Co-Director)

Dr. Ernst von Lavante

Dr. Robert L. Ash

Dr. John J. Swetits

DEDICATION

To my parents, Mr. Mashalla Samareh Abolhassani and Mrs. Khanomagh Yazdani.

ACKNOWLEDGEMENTS

I would like to express my appreciation to my technical monitor at NASA Langley Research Center, Dr. Robert E. Smith, for his invaluable guidance during my entire graduate work. This work would have been impossible without his patience, supervision, suggestions and encouragement. Also, I would like to thank my academic advisor, Dr. Surendra N. Tiwari, for his help and encouragement. I wish to thank my dissertation committee members, Drs. Ernest von Lavante, Robert L. Ash and John J. Swetits for providing me with helpful suggestions. I would like to thank the faculty of Mechanical Engineering and Mechanics of Old Dominion University for giving me a good education. I wish to thank the members of the Computer Applications Branch of NASA Langley Research Center for their help and providing me with a friendly environment.

I would like to express my appreciation to Ms. Joan I. Pitts of NASA Langley Research Center for her numerous contributions. I wish to thank Dr. Lars-Erik Eriksson of Old Dominion University for his helpful discussion on the implementation of smoothing and Dr. R. W. Newsome of Air Force Wright Aeronautical Laboratories for his discussion on the conical solutions of Navier-Stokes equations. I wish to express my deepest appreciation to my parents in Iran, Mr. Mashalla Samareh Abolhassani and Mrs. Khanomagh Yazdani, for their moral support. This work is dedicated to them. This work was supported partially by NASA Langley Research Center through Grant NCC1-68.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

$A_{i,j}$	matrix in Eq. (6.20)
b_i	coefficients for weight function f_i
C_1, C_2, C_3	constants in Eqs. (5.5b), (5.10)
C_v	specific heat at constant volume
c_j	right-hand side of Eq. (6.20)
e	internal energy per unit volume
e_i	covariant base vector
e^i	contravariant base vector
E	total energy per unit volume
F_i, f_i	weight functions
F, G, H	vector fluxes for coordinate directions
g_{ij}	covariant fundamental metric coefficient
g^{ij}	contravariant fundamental metric coefficient
I	weighted integral
I_o	weighted integral for orthogonality
I_s	weighted integral for smoothness
I_w	weighted integral for volume variations
I_1, I_2,I_3	weighted integrals for one-dimensional adaption
J	Jacobian matrix
K	coefficient of thermal conductivity
L	Length of the wing
m	coefficient in Eq. (6.12)
M	Mach number

n	magnitude of a normal vector on a surface
P	static pressure
P_r	Prandtl number
$\dot{q}$	heat flux
q_x, q_y, q_z	components of heat conduction vectors
$\bar{r}$	position vector
R_i	percentage of grid points allocated to f_i
R	universal gas constant
R_e	Reynolds number
S, s	arc length
T	static temperature
T	free stream temperature
T_w	wall temperature
T_o	reference temperature for Sutherland's viscosity law
t	time
S_o	constant for Sutherland's viscosity law
$\bar{U}$	vector of state variables
u, v, w	velocity components in the physical coordinate system
W	weight function
V_ℓ	contravariant velocity in the computational coordinate system
x, y, z	coordinates for the physical domain

Greek Symbols

$\bar{\alpha}$	coefficient for pressure damping
α, β	constant in Eqs. (6.12), (6.16a)
γ	ratio of specific heats
δ_{ij}	Kronecker delta function

δ	interval in computational coordinate system
λ_o, λ_w	coefficient of orthogonality, volume variation
κ	bulk viscosity
μ	molecular viscosity
μ_o	reference viscosity in Sutherland's viscosity law
μ'	second coefficient of viscosity
ξ, η, ζ	coordinates for the computational domain
θ	angular distance
ρ	density
$\bar{\tau}$	stress tensor
τ_{xx}, τ_{xy}, τ_{xz} τ_{yy}, τ_{yz}, τ_{zz}	elements of the stress tensor
ϕ_x, ϕ_y, ϕ_z	components of the viscous dissipation functions
Γ	area

Subscript

w	solid walls
∞	free-stream value
max	maximum (for arc length)
min	minimum (for time step)

Superscript

n	time level
T	transpose
-1	inverse

Indices

i, j, k	point indices

Operators

∇	gradient
$\cdot$	inner product
L_ξ, L_η, L_ζ	finite-difference operators
∂	partial differentiation
d	total derivative
Σ	summation
x	cross product

ABSTRACT

TOPOLOGY AND GRID ADAPTION FOR HIGH-SPEED FLOW COMPUTATIONS

Jamshid Samareh Abolhassani
Old Dominion University, 1987
Directors: Dr. Surendra N. Tiwari
Dr. Robert E. Smith

This study investigates the effects of grid topology and grid adaption on numerical solutions of the Navier-Stokes equations. In the first part of this study, a general procedure is presented for computation of high-speed flow over complex three-dimensional configurations. This includes the grid generation and solution algorithm for Navier-Stokes equations in a general three-dimensional curvilinear coordinate system. The flow field is simulated on the surface of a Butler wing in a uniform stream. Results are presented for Mach number 3.5 and a Reynolds number of 2,000,000. The O-type and H-type grids have been used for this study, and the results are compared together and with other theoretical and experimental results. The results demonstrate that while the H-type grid is suitable for the leading and trailing edges, a more accurate solution can be obtained for the middle part of the wing with an O-type grid. In spite of some discrepancies, the present numerical results compare favorably with the experimental results. In the second part of this study, methods of grid adaption are reviewed and a method is developed with the capability of adapting to several variables. This method is based on a variational

approach and is an algebraic method. Also, the method has been formulated in such a way that there is no need for any matrix inversion. This method is used in conjunction with the calculation of hypersonic flow over a blunt-nose body. A movie has been produced which shows simultaneously the transient behavior of the solution and the grid adaption.

For both cases, the simulations are done by integrating the viscous Navier-Stokes equations. These equations govern the unsteady, viscous, compressible and heat-conducting flow of an ideal gas, and all viscous terms are retained. The equations are written in curvilinear coordinates so that the body surface is represented accurately. The computer codes are written in FORTRAN, is vectorized and currently run on the CDC Vector Processing System (VPS-32, CYBER 205) computer. The results indicate the viability and validity of the proposed methods.

Chapter 1

INTRODUCTION

Continuum problems in engineering are quite often modeled by systems of nonlinear partial differential equations. These equations are usually complex and, in most instances, must be solved by numerical means. The numerical solution of the governing partial differential equations has two steps: (1) grid generation and (2) numerical integration. Grid generation is the division of the solution domain into discrete interconnected points called grid points. The accumulation of grid points is called a grid. The location of grid points is primarily a function of boundary geometry and the physics of the problem. The grid should conform to the boundaries and be concentrated in regions where there are large gradients. In the second step, the derivatives in the partial differential equations are approximated with algebraic expressions (usually by Taylor series expansions where higher order terms are truncated). The most commonly used techniques for integrating the governing equations are classified as finite difference techniques, finite volume techniques or finite element techniques. The accuracy of a numerical solution depends on both the solution technique and the grid. Grid points must be appropriately defined to apply boundary conditions and must be sufficiently close together to resolve the physics to the desired level of accuracy. They must also be oriented relative to each other in such a manner that errors are not

introduced into the solution. On the other hand, computer speed and memory limit the number of grid points that can be used in the solution of a given problem. It is, therefore, necessary to distribute grid points to maximize overall accuracy while covering the entire region of interest.

If the grid points are ordered in such a way that the relationship between any grid point and its neighbors is the same for all grid points, then this grid is called a structured grid. A structural grid can be generated numerically by determining the values of the physical coordinates in the physical domain from the values on the boundaries. This can be accomplished in two basic ways: (1) by algebraic interpolation from the boundary values, and (2) by solving a set of partial differential equations with the boundary geometry as a boundary condition. The resulting boundary-fitted coordinate system is a curvilinear coordinate system having some coordinate lines (surfaces in three-dimensions) conforming to the shape of each boundary. When the governing equations are transformed onto such a coordinate system, a finite approximation can be made using neighboring points at coordinate line intersections, without the need for interpolation, regardless of boundary shape and boundary movement. Thus, quite general code can be written for the numerical solutions of the governing partial differential equations on arbitrary regions. Although many of the accomplishments in grid generation have occurred within the field of computational fluid dynamics, the techniques are equally applicable in electromagnetics, solid mechanics, and other areas involving solutions of partial differential equations in an arbitrary region. The

literature on grid generation is extensive and this is critically reviewed in [1, 2]*.

An examination of the Taylor series expansion of a function about a point in the solution domain, reveals that the truncation error depends on derivatives of the solution and characteristics of the grid, such as distribution of spacing, orthogonality and aspect ratio. The selection of a grid topology has direct effect on the solution of a given problem through the introduction of singularities and constraints on the orthogonality of the grid. Eriksson [3] has studied the effects of grid singularities on the solution of the Euler equations. It was pointed out that the nonconservative centered scheme is likely to be unstable at mesh singularities, whereas, the conservative centered finite volume scheme is stable in a local sense. Also, it was concluded that the local time-stepping gives rise to exponentially growing modes for nonconservative schemes. Mastin and Thompson [1, 4] examined two sources of truncation error in the numerical solutions of partial differential equations on a curvilinear coordinate system. The error sources are derived from grid spacing and the degree of nonorthogonality (grid skewness). It is possible for a poor distribution or orientation of grid points to introduce errors into a numerical solution. For example, sudden changes in the line spacing and excessively skewed lines can introduce negative numerical diffusion into a solution. Although precise orthogonality is not essential, some error terms vanish for an orthogonal system. Also, Raithby [5] has observed that an excessive

*The numbers in brackets indicate references.

grid skewness will exaggerate the truncation error. The literature survey indicates a lack of information on the effects of grid topology on the solution, especially in three-dimensions. In general, the coordinate system should have lines concentrated in regions of an expected high variation of the physical solutions. The coordinate system should be coupled with the physical solution so that the coordinate lines continually adapt to resolve the evolving gradients in the physical solution. A significant amount of work on the grid adaption is available in the literature and this is discussed, in detail, in Chap. 6.

In regard to the grid generation and solution of continuum problems, this study has two distinct objectives. These are: (1) the qualitative assessment of errors resulting from the relative orientation of grid points and overall grid topology, and (2) the proper redistribution of the grid points over a region to minimize the truncation error through grid adaption. In order to study the effects of grid topology, flow over a Butler wing [6] has been simulated using two different grid topologies. Solutions obtained with the two topologies are compared with each other and with experimental results obtained by Squire [7-9]. In the second part of this study, a method of grid adaption with the capability of adapting the grid points to several variables is proposed. This method is formulated in such a way that it is not necessary to solve a system of equations and is, therefore, very efficient computationally.

The physical models used in this study are discussed in Chap. 2. The formulations of the governing equations are presented in Chap. 3. The method of solution and the initial and boundary conditions are

explained briefly in Chaps. 4 and 5, respectively. Chapter 6 contains a review of the grid-adaption method and discussion of the proposed new method. Finally, some critical results are presented in Chap. 7.

Chapter 2

PHYSICAL MODELS

The two basic physical models considered for this study are discussed briefly in this chapter. These are a Butler wing and a blunt leading edge of a panel holder.

The Butler wing is a good test case for investigating the effects of grid topology on the numerical solutions of Navier-Stokes equations. This is due to a unique feature of its geometry. The Butler wing is a delta wing which was proposed by D. S. Butler [6]. The plan form of the wing is an isosceles triangle, and the leading edges of the wing lay along the Mach lines of the unperturbed stream. The first twenty percent of the wing is conical and the last eighty percent of the wing has elliptical cross sections with increasing eccentricity along the x-axis (Fig. 2.1). At the trailing edge, the elliptical cross section has infinite eccentricity and is a straight line. The Butler wing is symmetric about (x-z) and (x-y) planes. This permits the use of one quarter of the entire physical domain with zero angle of attack (Fig. 2.1). However, if the angle of attack is greater than zero, then half of the physical domain should be considered. The semi-major and minor axes are given by

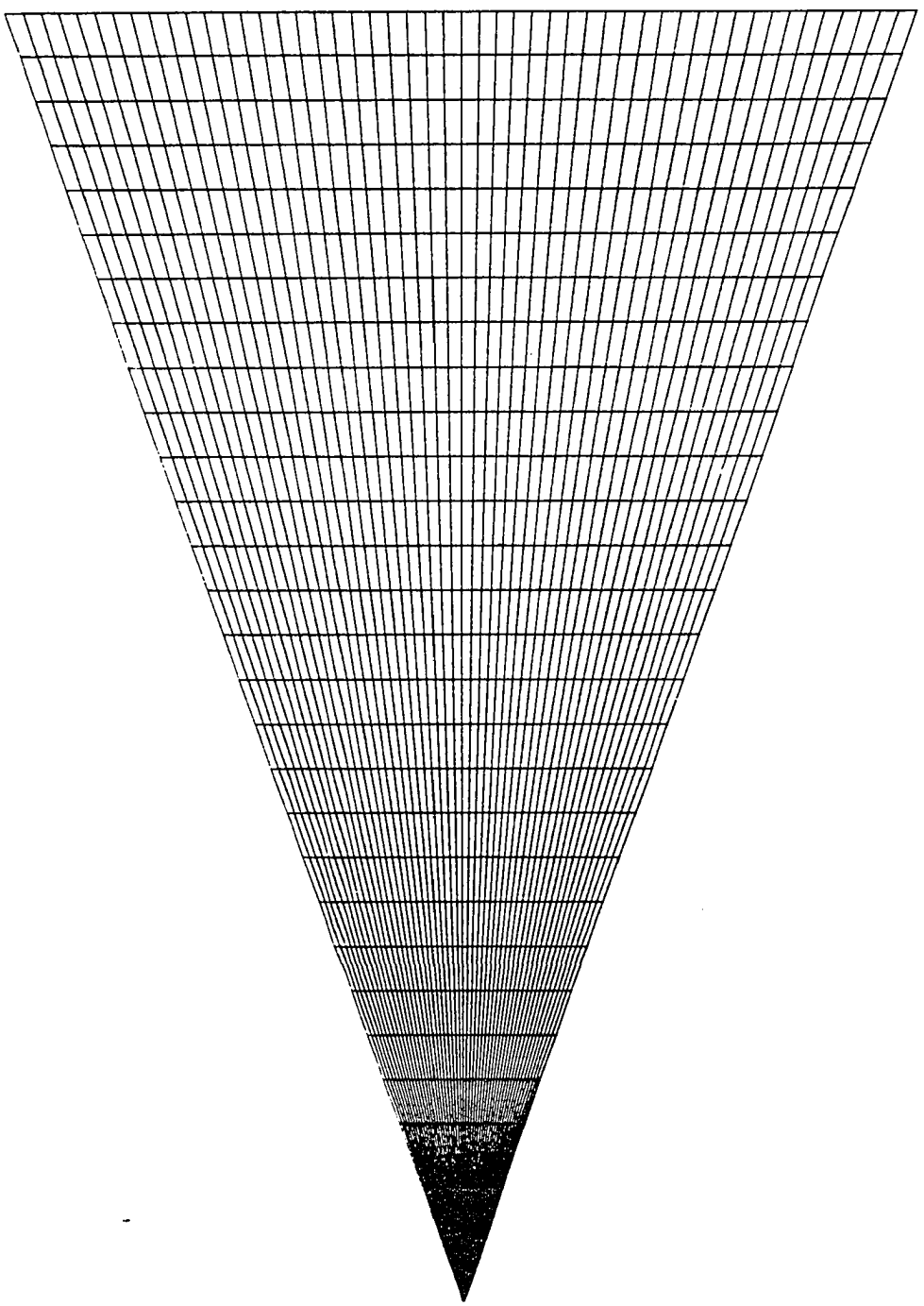


Fig. 2.1a Physical model of a Butler Wing (top view)

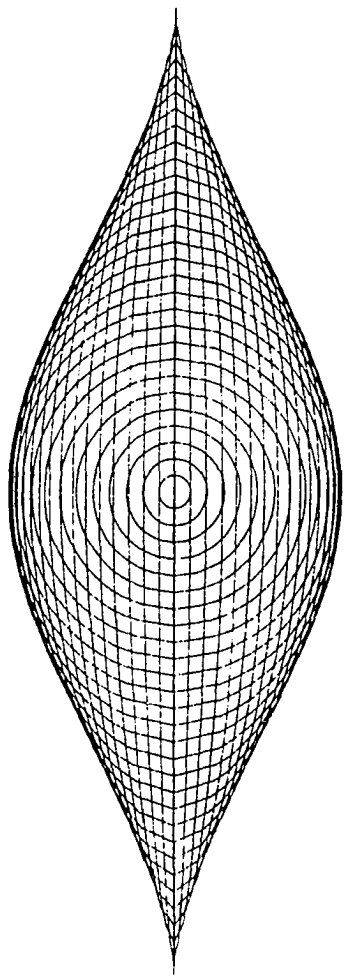


Fig. 2.1b Physical model of a Butler wing (front view)

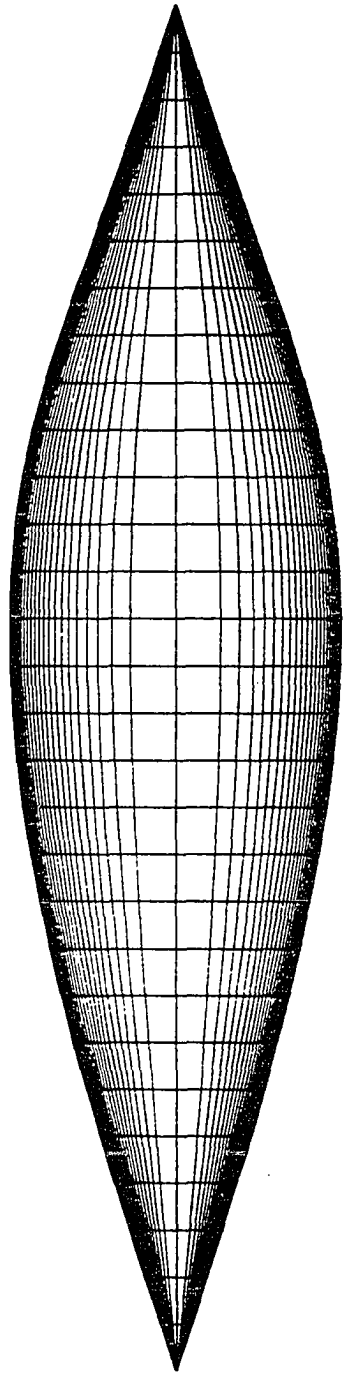


Fig. 2.1c Physical model of a Butler wing (side view)

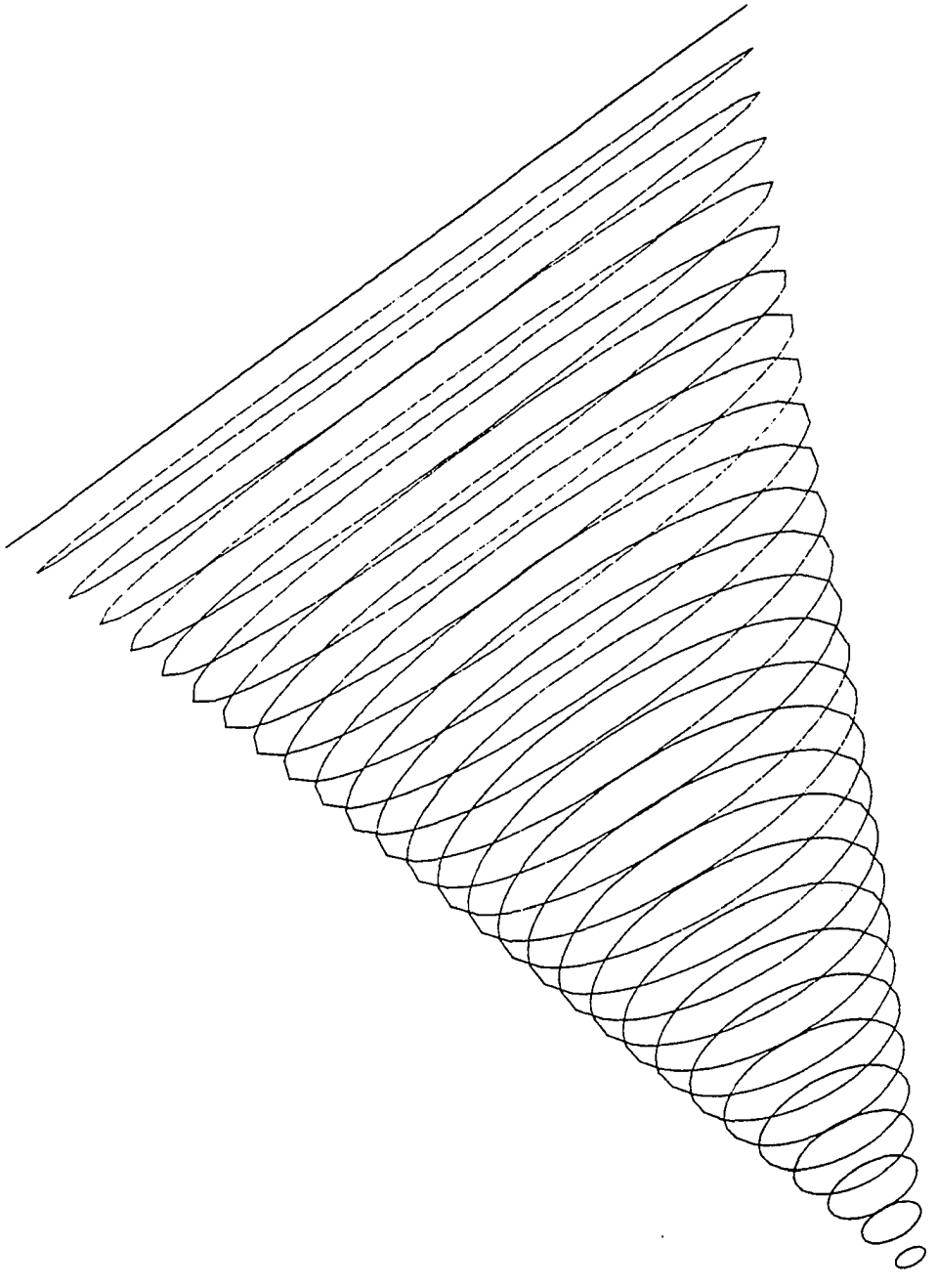


Fig. 2.1d Physical model of a Butler wing (oblique View)

$$\text{Major axis (semi-span)} = \frac{x}{\Lambda} \qquad 0 \leq x \leq L \qquad (2.1a)$$

$$\text{Minor axis (thickness on centerline)} = \frac{x}{\Lambda} \qquad 0 \leq x \leq 0.2L \qquad (2.1b)$$

$$= \frac{x}{\Lambda}\left[1 - \left[\frac{x - 0.2L}{0.8x}\right]^4\right] \qquad 0.2L < x \leq L \qquad (2.1c)$$

where

$$\Lambda^2 = M_\infty^2 - 1 \ .$$

The model is 0.8 ft. (0.2438 m) long, and the geometry has been generated for a Mach number of 3.5. That is the semi-apex angle of the plan form and the initial conical nose is $\sin^{-1}(1./3.5) = 16.602^0$.

Butler has compared the experimental results for surface pressure with the theoretical results [6]. These theoretical results are obtained from inviscid equations of motion which are simplified by using the slender-body approximations. Walkden and Caine estimated the pressure on the surface of a Butler wing at zero incident in a steady uniform stream. They numerically integrated the two semi-characteristic forms of the equations governing inviscid supersonic flow of an ideal gas with constant specific heats [10]. Squire has obtained experimental results for a Butler wing with varying Mach number and angle of attack [9]. In all previous analytical and numerical investigations, the inviscid form of the equations of motion has been used.

Grid generation is the first step which should be considered in obtaining flow field solutions over any configuration. Due to the data base management of the present program, it is necessary to map an entire physical domain into a rectangular parallelepiped. Among the grid types, selection of an O-type grid for cross sections in the stream-wise direction produces a point singularity at the nose tip and a line

singularity at the trailing edge (Fig. 2.2). Nevertheless, an O-type grid maps the entire solid boundary onto an entire face of the parallelepiped. It is also possible to generate orthogonal grid in the regions where there is relative high curvature. However, an H-type grid does not map the solid boundary onto an entire face of the computational box (Fig. 2.2). This creates a potential problem in updating the boundary conditions near the leading edges of the wing and also the grid in some regions could be highly skewed. But, there are no singularities in the grid. Figure 2.2 shows a topological comparison between H-type and O-type grids for the Butler wing. Both types of grids have been used in this study, and the results are compared with other numerical, analytical and experimental results.

Another aspect of this study is the grid adaption for high-speed flow computation. In the hypersonic flow about blunt bodies (Fig. 2.3), the temperature, pressure and density of the flow increase almost explosively across a shock wave. At the same time, the curved shock wave is close to the body. Numerical simulation of this phenomena has been a great challenge to the computational fluid dynamics researchers. Presently, there is a great deal of interest in improving the quality of numerical simulation techniques, and grid adaption is one way to achieve this goal.

The accuracy of finite-difference solutions depends on the fineness of the grid. Therefore, the finer the grid, the more accurate the numerical solution will be. Also, the accuracy of solutions depends on the resolution of the solution gradient. The presence of large gradients causes the error to be large in the difference approximation of derivatives. In the presence of shock waves, more artificial

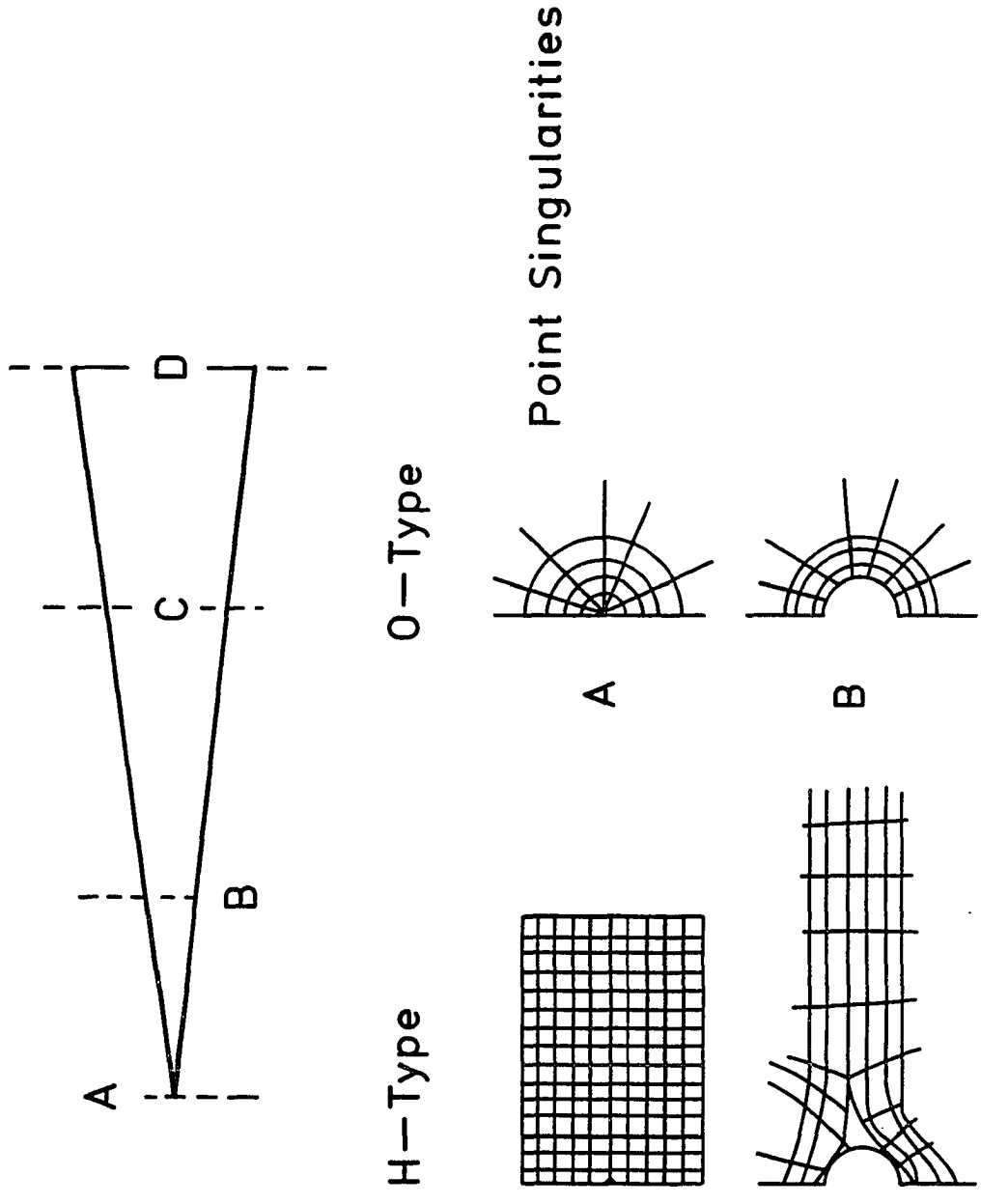


Fig. 2.2 Topology of the Butler wing

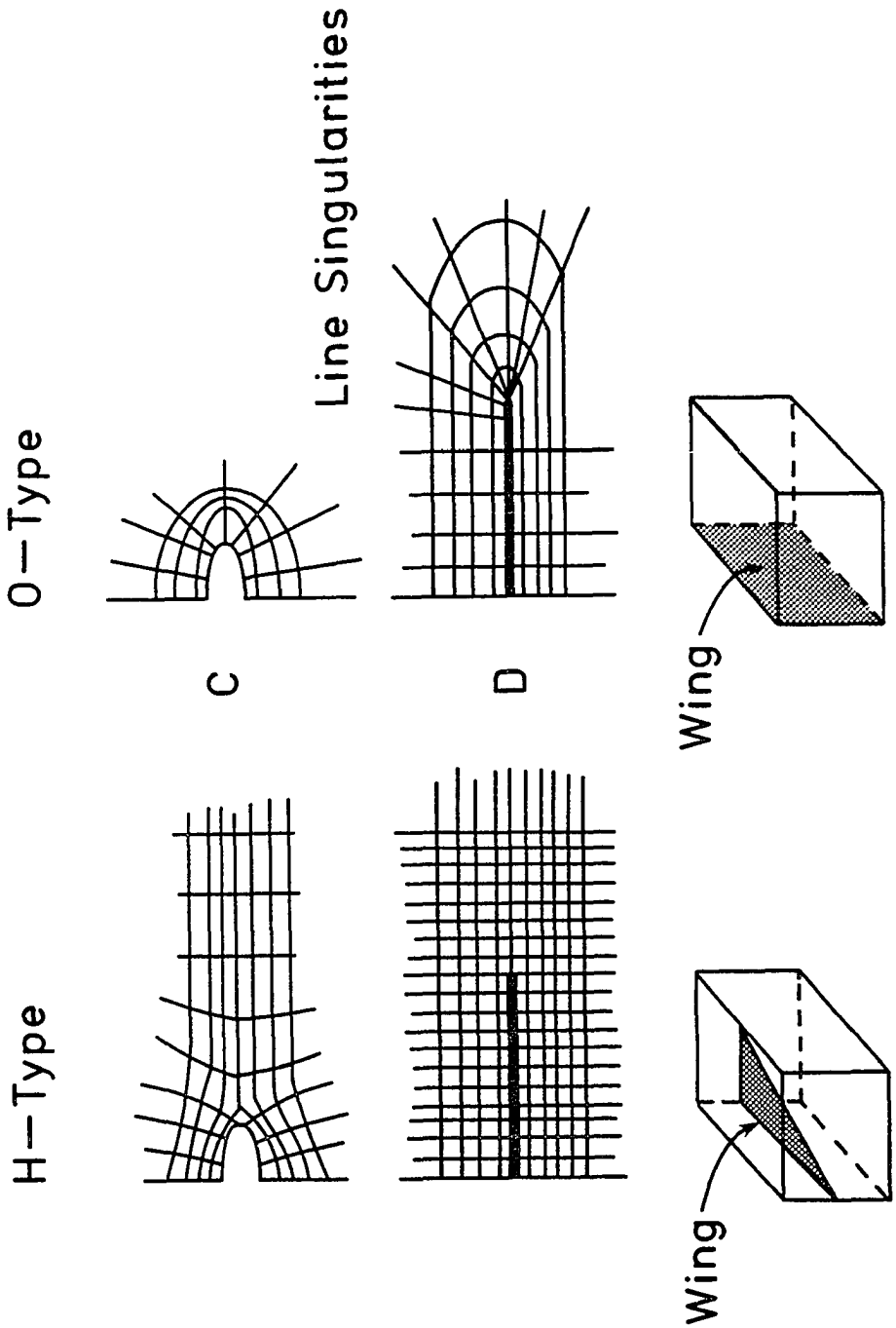


Fig. 2.2 Continued

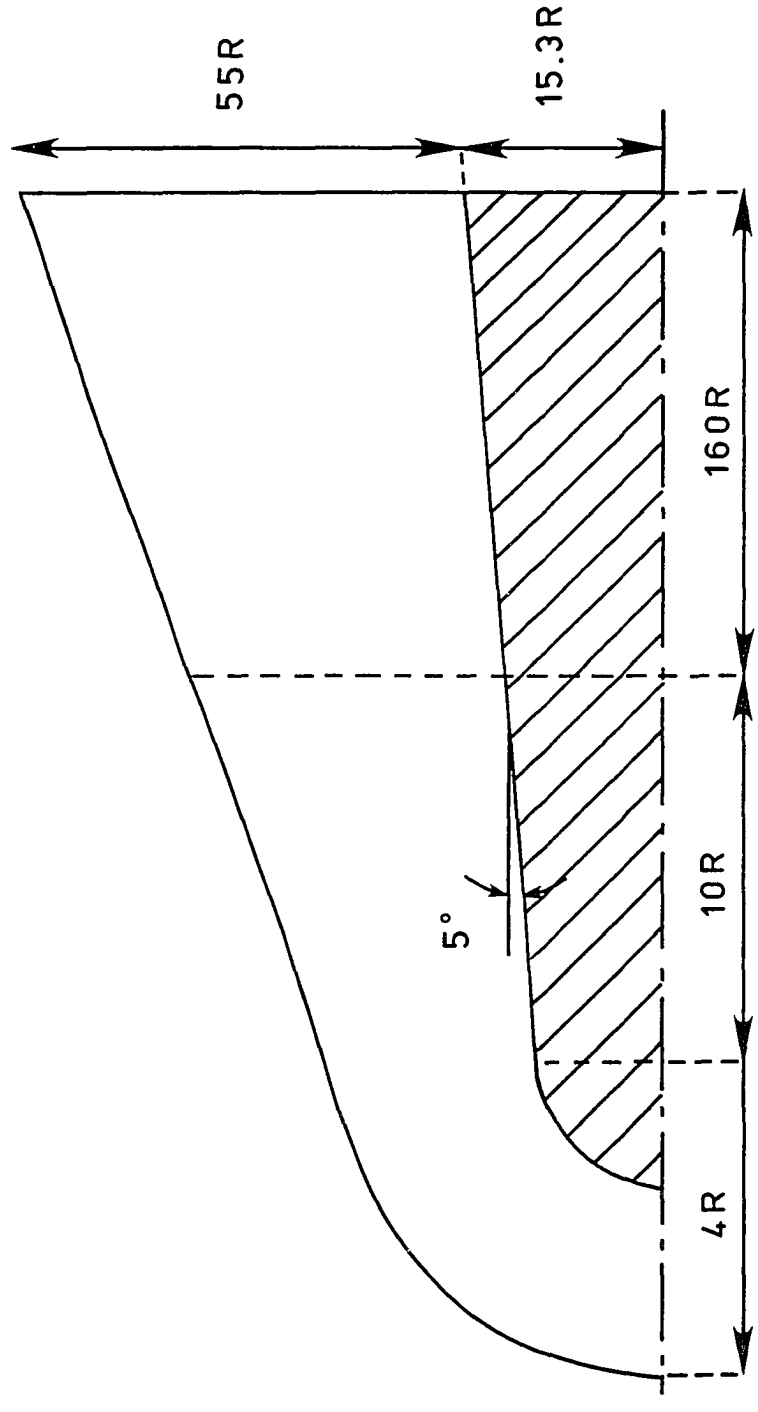


Fig. 2.3 Physical model of the blunt leading-edge of a panel holder

diffusion must be added to retain adequate smoothness of the solutions. Therefore, there is a need for schemes that can resolve large gradients without adding additional grid points. An adaptive scheme moves the grid points to regions of high gradients, when the locations of these gradients are not known a priori. Also, an adaptive method reduces the total number of grid points required to achieve a given accuracy, but it requires more computer time. In some instances, the computer time makes grid adaption impractical. The ideas used in the construction of adaptive grid techniques are limited only by one's imagination; and any scheme that works in the sense of providing a better solution is a good one. The ultimate answer to numerical solutions of partial differential equations may well be to dynamically adapt grids, rather than to devise more elaborate difference representations and solution methods [11].

In both cases, the flow is simulated by solving the Navier-Stokes equations numerically. The equations are unsteady, compressible, viscous and three- or two-dimensional. The time dependency of the governing equations allows the solution to progress naturally from an arbitrary initial guess to an asymptotic steady state, if one exists. The equations are transformed from physical coordinates to computational coordinates, allowing the solutions to be computed in a rectangular domain. The equations are solved by the MacCormack time-split technique [12, 13] which is vectorized and programmed to run on the CDC VPS-32 (CYBER 205) computer. The code is written in 32-bit (half-word) FORTRAN. The details of the formulation and solution procedure are presented in the subsequent chapters.

Chapter 3

GOVERNING EQUATIONS

3.1 Navier-Stokes Equations

The governing equations for a thermal fluid system are the conservation of mass, momentum and energy. These equations are developed for an arbitrary region under the assumption that the system is a continuum. Equations of motion for a viscous, compressible, unsteady and heat-conducting fluid can be written as [14]

$$\text{Continuity:} \qquad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{u}) = 0 \ , \tag{3.1a}$$

$$\text{Momentum:} \qquad \frac{\partial (\rho \bar{u})}{\partial t} + \nabla \cdot (\rho \bar{\bar{u}}\bar{u} - \bar{\tau}) = 0 \ , \tag{3.1b}$$

$$\text{Energy:} \qquad \frac{\partial (E)}{\partial t} + \nabla \cdot (E\bar{u} + \bar{\dot{q}} - \bar{u} \cdot \bar{\tau}) = 0 \ , \tag{3.1c}$$

where E is the total energy per unit volume given by $E = \rho(e + v \cdot v/2)$ and e is the internal energy per unit volume. Equations (3.1b-3.1c) can be simplified by assuming that the stress at a point is linearly dependent on the rate of strain (deformation) of the fluid (Newtonian fluid)

$$\tau_{ij} = - P \, \delta_{ij} + \mu \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \delta_{ij} \, \mu' \, \frac{\partial u_k}{\partial x_k} \right] . \tag{3.2a}$$

The Kronecker delta function is denoted by δ_{ij}, and μ' is the second coefficient of viscosity which is related to the coefficient of bulk

viscosity (κ) by the expression $\kappa = 2\mu/3+\mu'$. The contribution of κ can be neglected if the pressure in a fluid is not changed abruptly during its expansion or contraction, in other words, the hydrostatic pressure is assumed to be equal to the average of the normal stresses. Under this assumption, the stress tensor can be related to the pressure and velocity components as

$$\tau_{ij} = - P \delta_{ij} + \mu \left[\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k}\right] . \qquad (3.2b)$$

For an isotropic system, the heat flux in Eq. (3.1c) can be expressed in terms of temperature gradient (Fourier law of heat conduction) as

$$\dot{q} = - K \nabla T \qquad (3.3)$$

where K is the coefficient of thermal conduction. A common approximation used for viscosity is based on the kinetic theory of gases using an idealized intermolecular-force potential. The relation is

$$\frac{\mu}{\mu_o} = \left(\frac{T}{T_o}\right)^{3/2} \frac{T_o + S_o}{T + S_o} \qquad (3.4)$$

where

$$S_o = 198.6 \ ^0R$$

$$T_o = 492 \ ^0R$$

$$\mu_o = 0.35x10^6 \ (lbf\text{-}sec)/ft^2$$

the coefficients of thermal conduction K can be determined from the Prandtl number as

$$K = \frac{\gamma \mu C_v}{P_r} \tag{3.5}$$

where C_v is the specific heat at a constant volume and γ is the ratio of specific heats.

It is essential to have a supplementary relation to close the system of equations, Eqs. (3.1a-3.1c). By neglecting the intermolecular forces (thermally perfect system), the thermodynamic properties can be described by the equation of state

$$P = \rho RT \tag{3.6}$$

where R is the universal gas constant. The assumption of thermally perfect gas permits the internal energy to be expressed as a function of temperature only i.e., e=e(T). In addition, the assumption of a calorically perfect gas [e(0)=0] allows the following relation

$$e = C_v T \ . \tag{3.7}$$

A substitution of Eq. (3.6) into Eq. (3.7) results in

$$P = \rho e \ (\gamma - 1) \ . \tag{3.8}$$

The equations of motion are in conservative form. For simplicity, these equations can be written in a compact vector form as

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = 0 \ , \tag{3.9}$$

where

$$U = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{Bmatrix} , \qquad F = \begin{Bmatrix} \rho u \\ \rho uu - \tau_{xx} + P \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \\ Eu + \dot{q}_x - \phi_x + Pu \end{Bmatrix} ,$$

$$G = \begin{Bmatrix} \rho v \\ \rho uv - \tau_{yx} \\ \rho vv - \tau_{yy} + P \\ \rho vw - \tau_{yz} \\ Ev + \dot{q}_y - \phi_y + Pv \end{Bmatrix}, \qquad H = \begin{Bmatrix} \rho w \\ \rho uw - \tau_{zx} \\ \rho vw - \tau_{zy} \\ \rho ww - \tau_{zz} + P \\ Ew + \dot{q}_z - \phi_z + Pw \end{Bmatrix},$$

$$\tau_{xx} = -P + 2\mu \frac{\partial u}{\partial x} - \frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right),$$

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

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

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

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

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

$$\phi_x = u\tau_{xx} + v\tau_{xy} + w\tau_{xz},$$

$$\phi_y = u\tau_{xy} + v\tau_{yy} + w\tau_{yz},$$

$$\phi_z = u\tau_{xz} + v\tau_{yz} + w\tau_{zz},$$

$$q_x = -K \frac{\partial T}{\partial x},$$

$$q_y = -K \frac{\partial T}{\partial y},$$

$$q_z = -K \frac{\partial T}{\partial z}.$$

3.2 Navier-Stokes Equations in Computational Coordinates

For the sake of generality, the governing equations are transformed from a physical domain into a computational domain as

$$\frac{\partial U}{\partial t} + \begin{Bmatrix} \xi_x \\ \xi_y \\ \xi_z \end{Bmatrix} \left\{ \frac{\partial F}{\partial \xi} \; \frac{\partial G}{\partial \xi} \; \frac{\partial H}{\partial \xi} \right\} + \begin{Bmatrix} \eta_x \\ \eta_y \\ \eta_z \end{Bmatrix} \left\{ \frac{\partial F}{\partial \eta} \; \frac{\partial G}{\partial \eta} \; \frac{\partial H}{\partial \eta} \right\}$$

$$+ \begin{Bmatrix} \zeta_x \\ \zeta_y \\ \zeta_z \end{Bmatrix} \left\{ \frac{\partial F}{\partial \zeta} \; \frac{\partial G}{\partial \zeta} \; \frac{\partial H}{\partial \zeta} \right\} = 0 \; . \qquad (3.10)$$

The stress tensor, dissipation function and heat conduction must also be transformed from a physical domain into a computational domain. Using Eqs. (A.23c) and (A.24a), they can be expressed as

$$\tau_{xx} = -P + 2\mu \left(\frac{\partial \xi}{\partial x} \frac{\partial u}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial u}{\partial \eta} + \frac{\partial \zeta}{\partial x} \frac{\partial u}{\partial \zeta} \right) - 2/3 \; \mu \left(\frac{\partial \xi}{\partial x} \frac{\partial u}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial u}{\partial \eta} + \frac{\partial \zeta}{\partial x} \frac{\partial u}{\partial \zeta} \right.$$

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

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

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

$$\tau_{yy} = -P + 2\mu \left(\frac{\partial \xi}{\partial y} \frac{\partial v}{\partial \xi} + \frac{\partial \eta}{\partial y} \frac{\partial v}{\partial \eta} + \frac{\partial \zeta}{\partial y} \frac{\partial v}{\partial \zeta} \right) - 2/3 \; \mu \left(\frac{\partial \xi}{\partial x} \frac{\partial u}{\partial \xi} + \frac{\partial \eta}{\partial x} \frac{\partial u}{\partial \eta} + \frac{\partial \zeta}{\partial x} \frac{\partial u}{\partial \zeta} \right.$$

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

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

$$\tau_{zz} = -P + 2\mu \left(\frac{\partial \xi}{\partial z}\frac{\partial w}{\partial \xi} + \frac{\partial \eta}{\partial z}\frac{\partial w}{\partial \eta} + \frac{\partial \zeta}{\partial z}\frac{\partial w}{\partial \zeta}\right) - 2/3\ \mu \left(\frac{\partial \xi}{\partial x}\frac{\partial u}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial u}{\partial \eta} + \frac{\partial \zeta}{\partial x}\frac{\partial u}{\partial \zeta}\right.$$

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

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

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

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

The transformation is based on the chain rule. The transformation coefficients can be computed from a functional relation between the computational coordinates and the physical coordinates as

$$\xi = \xi\ (x, y, z) ,$$

$$\eta = \eta\ (x, y, z) , \qquad (3.11)$$

$$\zeta = \zeta\ (x, y, z) .$$

If the relation in Eq. (3.11) were known, the transformation coefficient could have been computed by direct differentiation. If not, after some algebraic manipulations (Eq. A.14), the transformation coefficients can be computed by

$$\begin{Bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} & \frac{\partial \xi}{\partial z} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} & \frac{\partial \eta}{\partial z} \\ \frac{\partial \zeta}{\partial x} & \frac{\partial \zeta}{\partial y} & \frac{\partial \zeta}{\partial z} \end{Bmatrix} = [J] = \begin{Bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{Bmatrix}^{-1} \tag{3.12}$$

where $[J^{-1}]$ is defined as

$$[J] = \frac{1}{|J^{-1}|} \begin{Bmatrix} \left(\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \eta}\right) & -\left(\frac{\partial x}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial z}{\partial \eta}\right) & \left(\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \eta}\right) \\ -\left(\frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \xi}\right) & \left(\frac{\partial x}{\partial \xi}\frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial z}{\partial \xi}\right) & -\left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \xi}\right) \\ \left(\frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \xi}\right) & -\left(\frac{\partial x}{\partial \xi}\frac{\partial z}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial z}{\partial \xi}\right) & -\left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}\right) \end{Bmatrix} \tag{3.13}$$

$$|J^{-1}| = \begin{Bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{Bmatrix} = \frac{\partial x}{\partial \xi}\left(\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \eta}\right) - \frac{\partial x}{\partial \eta}\left(\frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \xi}\right) + \frac{\partial x}{\partial \zeta}\left(\frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \xi}\right)$$

In the case of a Butler wing, the grid planes are perpendicular to the x-coordinate. Consequently, physical coordinates can be written as

$$\begin{aligned} x &= x(\xi) , \\ y &= y(\xi, \eta, \zeta) , \\ z &= z(\xi\, \eta, \zeta) . \end{aligned} \tag{3.14}$$

This reduces the transformation coefficients from nine to five non-zero elements and, therefore, reduces the memory requirements.

In the case of grid adaption, the physical geometry is two-dimensional. Therefore, all derivatives with respect to the x and ξ coordinates are set to zero. This simplifies the governing equations to the following form

$$\frac{\partial U}{\partial t} + \begin{bmatrix} \eta_y \\ \eta_z \end{bmatrix} \left(\frac{\partial G}{\partial \eta} \frac{\partial H}{\partial \eta} \right) + \begin{bmatrix} \zeta_y \\ \zeta_z \end{bmatrix} \left(\frac{\partial G}{\partial \zeta} \frac{\partial H}{\partial \zeta} \right) = 0 \tag{3.15}$$

where

$$U = \begin{Bmatrix} \rho \\ \rho v \\ \rho w \\ E \end{Bmatrix} , \qquad G = \begin{Bmatrix} \rho v \\ \rho vv - \tau_{yy} + P \\ \rho vw - \tau_{yz} \\ Ev + \dot{q}_y - \phi_y + Pv \end{Bmatrix} ,$$

$$H = \begin{Bmatrix} \rho w \\ \rho vw - \tau_{zy} \\ \rho ww - \tau_{zz} + P \\ Ew + \dot{q}_z - \phi_z + Pw \end{Bmatrix} .$$

There are four non-zero transformation coefficients.

In both cases, the transformed governing equations are called Chain Ruled Conservation Law Form (CRCLF) [15]. However, the governing equations can be written with metric coefficients inside the differentiations; this form is called the Strong Conservation Law Form (SCLW). It has been shown [15] that CRCLF requires no special considerations on how to compute the metric coefficients or their derivatives and it also requires that fewer arithmetic operations be performed compared to the other forms. Also, it has been shown that it has the ability to capture weak shocks.

Chapter 4

METHOD OF SOLUTION

A time-marching method is used to compute the solution so that the possible transient features can be readily captured. This explicit method is a time-split predictor-corrector algorithm which is second-order accurate in time and space [12]. The governing equations are split into three groups of operators, each aligned with transformed coordinates. Then, these equations, Eq. (3.10), are discretized in the computational directions. In a compact form, they can be expressed as

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

where

$$\Delta t_\eta = \Delta t_\zeta = \frac{1}{2}\,\Delta t_\xi \tag{4.1}$$

and L_ξ, L_η, and L_ζ are the operators in the ξ, η, and ζ directions, respectively. A time step is completed in this algorithm with the application of each operator applied symmetrically about the middle operator. Operator L_ξ can be defined as

$$L_\xi(\Delta t_\xi) = U^{out}_{i,j,k}\ , \tag{4.2a}$$

for the predictor step:

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

$$\left. + (H_i - H_{i-1})\,\frac{\partial\xi}{\partial z}\,i\right]_{j,k}\ , \tag{4.2b}$$

for the corrector step:

$$U^{out}_{i,j,k} = \frac{1}{2}\left(U^{in}_{i,j,k} + \bar{U}_{i,j,k} - \frac{\Delta t_\xi}{\Delta\xi}\left[(F_{i+1} - F_i)\frac{\partial\xi}{\partial x}\,i + (G_{i+1} - G_i)\frac{\partial\xi}{\partial y}\,i + (H_{i+1} - H_i)\frac{\partial\xi}{\partial z}\,i\right]_{j,k}\right). \qquad (4.2c)$$

Operator L_η can be defined as

$$L_\eta(\Delta t_\eta) = U^{out}_{i,j,k}, \qquad (4.3a)$$

for the predictor step:

$$\bar{U}_{i,j,k} = U^{in}_{i,j,k} - \frac{\Delta t_\eta}{\Delta\eta}\left[(F_j - F_{j-1})\frac{\partial\eta}{\partial x}\,j + (G_j - G_{j-1})\frac{\partial\eta}{\partial y}\,j + (H_j - H_{j-1})\frac{\partial\eta}{\partial z}\,j\right]_{i,k}, \qquad (4.3b)$$

for the corrector step:

$$U^{out}_{i,j,k} = \frac{1}{2}\left(U^{in}_{i,j,k} + \bar{U}_{i,j,k} - \frac{\Delta t_\eta}{\Delta\eta}\left[(F_{j+1} - F_j)\frac{\partial\eta}{\partial x}\,j + (G_{j+1} - G_j)\frac{\partial\eta}{\partial y}\,j + (H_{j+1} - H_j)\frac{\partial\eta}{\partial z}\,j\right]_{i,k}\right). \qquad (4.3c)$$

Operator L_ζ can be defined as

$$L_\zeta(\Delta t_\zeta) = U^{out}_{i,j,k}, \qquad (4.4a)$$

for the predictor step:

$$\bar{U}_{i,j,k} = U^{in}_{i,j,k} - \frac{\Delta t_\zeta}{\Delta\zeta}\left[(F_k - F_{k-1})\frac{\partial\zeta}{\partial x}\,k + (G_k - G_{k-1})\frac{\partial\zeta}{\partial y}\,k + (H_k - H_{k-1})\frac{\partial\zeta}{\partial z}\,k\right]_{i,j}, \qquad (4.4b)$$

for the corrector step:

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

$$\left.\left. + (G_{k+1} - G_k)\frac{\partial\zeta}{\partial y}k + (H_{k+1} - H_k)\frac{\partial\zeta}{\partial z}k\right]_{ij}\right) . \qquad (4.4c)$$

Fluxes are computed by a forward-difference approximation for the predictor and a backward-difference approximation for the corrector. Therefore, the algorithm is second-order accurate in space. More details can be found in [12-14].

The solution is stable if the time step of each operator does not exceed the allowable step size for that operator. The finite-difference scheme is consistent if the sums of the time steps for each operator are equal. The solution is second-order accurate if the operators are applied symmetrically [12-14].

This method has a time-step stability limit, but there is no rigorous stability analysis available. A commonly used conservative time-step is

$$\Delta t \leq \min\left[\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z} + c\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right)\right]^{-1} \qquad (4.5)$$

where c is the local speed of sound. This is valid for cartesian coordinates. Using Fourier series, a similar equation is derived for general curvilinear coordinates in Appendix B

$$\Delta t_\xi \leq \min\left[\frac{\Delta\xi}{|u\xi_x + v\xi_y + w\xi_z| + c(\xi_x^2 + \xi_y^2 + \xi_z^2)^{1/2}}\right] , \qquad (4.6a)$$

$$\Delta t_\eta \leq \min\left[\frac{\Delta\eta}{|u\eta_x + v\eta_y + w\eta_z| + c(\eta_x^2 + \eta_y^2 + \eta_z^2)^{1/2}}\right] , \qquad (4.6b)$$

$$\Delta t_\zeta \leq \min \left[\frac{\Delta\zeta}{|u\zeta_x + v\zeta_y + w\zeta_z| + c(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)^{1/2}}\right] . \qquad (4.6c)$$

In the supersonic and hypersonic regions, there exists large gradients which require a very fine grid to resolve them. Most central-difference methods admit a solution which has sawtooth or plus-minus waves with the shortest wave length that the grid can support. In the case of a nonlinear problem, these short waves interact, vanish, and reappear again as distorted long waves, or oscillations. These oscillations eventually cause the solution to blow up if they are not resolved. These numerical oscillations are caused by truncation error and can be reduced by grid refinements. The oscillations of "low frequency" can be suppressed by adding a fourth-order damping term. A common damping used is pressure damping. This is usually expressed in cartesian coordinates as

$$-\bar{\alpha}_\ell \, \Delta t_\ell \, \delta_\ell^3 \frac{\partial}{\partial \delta_\ell} \left[\frac{|v_\ell| + c}{4P} \frac{\partial^2 P}{\partial \delta_\ell^2} \frac{\partial \bar{U}}{\partial \delta_\ell}\right] \quad \ell = 1,2,3 \ , \qquad (4.7a)$$

where

$$\frac{1}{4P} \frac{\partial^2 P}{\partial \delta_1^2} = \frac{\partial^2 P}{\partial \xi^2} = \frac{|P_{i+1,j,k} - 2P_{i,j,k} + P_{i-1,j,k}|}{|P_{i+1,j,k} + 2P_{i,j,k} + P_{i-1,j,k}|} ,$$

$$\frac{1}{4P} \frac{\partial^2 P}{\partial \delta_2^2} = \frac{\partial^2 P}{\partial \eta^2} = \frac{|P_{i,j+1,k} - 2P_{i,j,k} + P_{i,j-1,k}|}{|P_{i,j+1,k} + 2P_{i,j,k} + P_{i,j-1,k}|} ,$$

$$\frac{1}{4P} \frac{\partial^2 P}{\partial \delta_3^2} = \frac{\partial^2 P}{\partial \zeta^2} = \frac{|P_{i,j,k+1} - 2P_{i,j,k} + P_{i,j,k-1}|}{|P_{i,j,k+1} + 2P_{i,j,k} + P_{i,j,k-1}|} .$$

The last term in Eq. (4.7a) is computed by a forward-difference approximation for the predictor step and a backward-difference approximation for the corrector step. It is common to use only an approximation of the second derivatives of pressure in the transformed coordinate system, instead of using the actual rigorous transformations. The use of covariant velocities may raise problems for complex grid topologies. One remedy for general curvilinear coordinates is to use contravariant velocities defined by

$$\begin{aligned} U &= (\xi_x u + \xi_y v + \xi_z w)\ (x_\xi^2 + y_\xi^2 + z_\xi^2)^{1/2} \\ V &= (\eta_x u + \eta_y v + \eta_z w)\ (x_\eta^2 + y_\eta^2 + z_\eta^2)^{1/2} \\ W &= (\zeta_x u + \zeta_y v + \zeta_z w)\ (x_\zeta^2 + y_\zeta^2 + z_\zeta^2)^{1/2} \end{aligned} \tag{4.7b}$$

To implement Eq. (4.7a), flux F in Eqs. (4.2a)-(4.2c) can be replaced by

$$F - \frac{\bar{\alpha}_\xi}{4P} \frac{\partial^2 P}{\partial\, \delta_\xi^2} \cdot \frac{|U| + c}{(g^{11})^{1/2}} \frac{\partial \bar{U}}{\partial \xi} \cdot \tag{4.8a}$$

Flux G in Eqs. (4.3a)-(4.3c) can be replaced by

$$G - \frac{\bar{\alpha}_\eta}{4P} \frac{\partial^2 P}{\partial\, \delta_\eta^2} \cdot \frac{|V| + c}{(g^{22})^{1/2}} \frac{\partial \bar{U}}{\partial \eta} \cdot \tag{4.8b}$$

Flux H in Eqs. (4.4a)-(4.4c) can be replaced by

$$H - \frac{\bar{\alpha}_\zeta}{4P} \frac{\partial^2 P}{\partial\, \delta_\zeta^2} \cdot \frac{|W| + c}{(g^{33})^{1/2}} \frac{\partial \bar{U}}{\partial \zeta} \cdot \tag{4.8c}$$

The equations for g^{11}, g^{22} and g^{33} are defined in Eq. (A.10), and U, V and W are defined in Eq. (4.7b)

Chapter 5

INITIAL AND BOUNDARY CONDITIONS

In computational fluid dynamics the initial conditions usually correspond to a real situation for a transient problem, or a rough guess for a steady state problem. In practice, initial conditions are obtained from experiments, empirical relations, approximate theories or previous computational results. An improper initial guess may result in generating unrealistic strong transient waves which propagate throughout the computational region, dominating the flow field and eventually leading to a solution failure. An important requirement for the initial conditions is that they should be physically as close as possible to the actual nature of the flow field in the region under study. This will minimize the number of iterations required for convergence. An attractive approach is to initialize the entire flow field with a crude and simple guess (e.g., free stream condition). During the course of the computation, both body and upstream boundary conditions are changed in a gradual manner to their final values over a prescribed number of iterations. In the present study, this technique is applied in only one step which is equivalent to impulsive initial conditions.

It is equally important to implement a realistic, accurate and stable method to determine boundary conditions. The application of certain conditions may cause numerical instability even though the flow is physically stable. Most of the boundary conditions currently

implemented are drawn mainly from intuition, simple analytical expressions, wind tunnel experiments and computational experimentation. In the selection of boundary conditions, consideration should be given to the following criteria: convergence, stability, computer time and above all the physical justification. For the Butler wing case, there are five different boundary conditions. They are upstream, downstream, outer, solid boundaries, and symmetry.

5.1 Upstream Boundary Condition for a Butler Wing

For the case of an H-grid, the upstream boundary is located at six grid point spacings ahead of the nose of the wing. The following undisturbed free stream conditions are assumed for this boundary

$$\bar{U}_{1,j,k} = \bar{U}_{\infty} \, . \tag{5.1}$$

For the case of an O-type grid, the upstream boundary is set at five percent of the chord from the tip of the wing to avoid the point singularity. The conical assumption has been made for this boundary [16]. Flow is said to be conical if the physical conditions such as pressure and velocity do not vary with position along any ray through a point, referred to as the vertex. For this case, the viscous-conical solutions are obtained for a cone at the proper angle of attack. This is done by creating a conical grid which has straight lines (rays) from the vertex (Fig. 5.1). Then, the conical Navier-Stokes equations are integrated for the middle plane (plane B), then planes A and C are set equal to plane B. This procedure is repeated until convergence is reached. This solution is valid provided the wing is sufficiently slender.

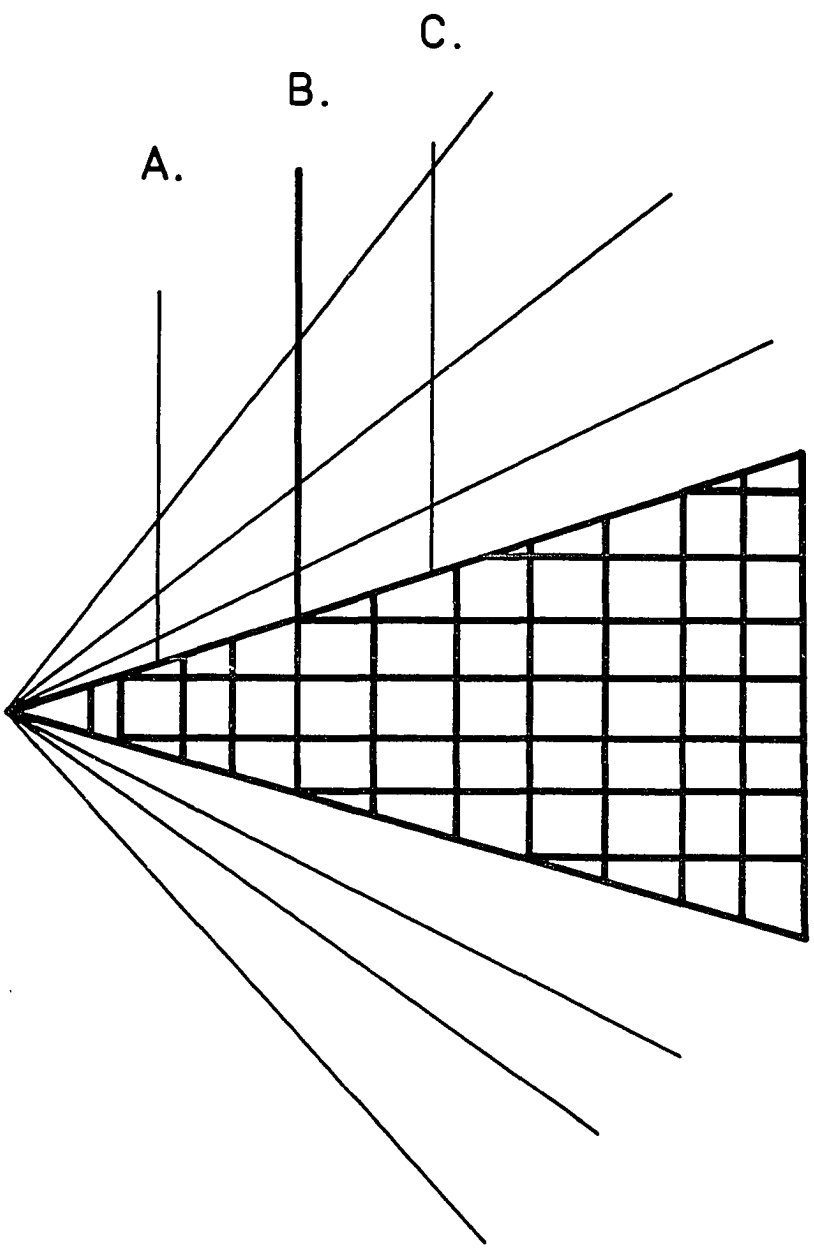


Fig. 5.1 Conical grid

5.2 Downstream Boundary Conditions for a Butler Wing

A zero gradient in the ξ-direction (parallel to the primary direction of flow) is assumed for the downstream boundary, i.e.,

$$\left.\frac{\partial \bar{U}}{\partial \xi}\right|_{IL,j,k} = 0 \ . \tag{5.2a}$$

A backward-difference is used to approximate the Eq. (4.2a) which results in

$$\bar{U}_{IL,j,k} = \bar{U}_{IL-1,j,k} \ . \tag{5.2b}$$

5.3 Far-field Boundary Conditions for a Butler Wing

The outer boundary is located far away from the body to avoid any influence on the interaction region. Presently, a zero normal gradient of fluxes is assumed for this boundary, i.e.,

$$\left.\frac{\partial \bar{U}}{\partial \eta}\right|_{i,JL,k} = 0 \ . \tag{5.3a}$$

Similar to the down stream boundary condition, a backward difference is used to approximate Eq. (5.3a) which results in

$$\bar{U}_{i,JL,k} = \bar{U}_{i,JL-1,k} \ . \tag{5.3b}$$

5.4 Solid-Wall Boundary Conditions for a Butler Wing

The walls are assumed to be impermeable and no-slip boundary conditions are applied, therefore, all velocity components are assumed to be zero. Similarly, the wall is assumed to have a constant temperature T_w. A zero normal pressure gradient is assumed for the solid surface, i.e.

$$\left.\frac{\partial P}{\partial n}\right|_{i,1,k} = 0 \quad . \tag{5.4}$$

This appears to be a boundary-layer approximation (i.e., a zero normal pressure gradient). It is, however, a much milder approximation, since constant pressure is not applied through the boundary layer but over one grid line in the boundary layer. This approximation has yielded stable computation for both the non-separated and separated boundary layers [17]. For general curvilinear coordinates, Eq. (A.25b) can be used to express Eq. (5.4) as

$$\frac{\partial P}{\partial n} = \frac{\xi_x \eta_x + \xi_y \eta_y + \xi_z \eta_z}{(\eta_x^2 + \eta_y^2 + \eta_z^2)^{1/2}} P_\xi + \frac{\eta_x^2 + \eta_y^2 + \eta_z^2}{(\eta_x^2 + \eta_y^2 + \eta_z^2)^{1/2}} P_\eta$$

$$+ \frac{\zeta_x \eta_x + \zeta_y \eta_y + \zeta_z \eta_z}{(\eta_x^2 + \eta_y^2 + \eta_z^2)^{1/2}} P_\zeta = 0 \quad . \tag{5.5a}$$

This equation is approximated using a central-difference approximation on the wall surface, and a second-order backward-difference approximation normal to the wall.

$$P_{i,1,k} = [4 P_{i,2,k} - P_{i,3,k} + 2 (C_2 (P_{i+1,2,k} - P_{i-1,2,k})$$

$$+ C_1 (P_{i,2,k+1} - P_{i,2,k-1}))]/3 \ , \tag{5.5b}$$

where

$$C_1 = \frac{g^{23}}{g^{22}} ,$$

$$C_2 = \frac{g^{12}}{g^{22}} . \tag{5.5c}$$

g^{12}, g^{23} and g^{22} are defined by Eqs. (A.9)-(A.10). Then, density is computed based on pressure and wall temperature. At the wall, all boundary conditions are second order accurate and are satisfactory even for a skewed grid. Leading-edges of the wing have high curvatures near the back of the wing, therefore the normal pressure may not be equal to zero. Consequently, the above boundary condition may not be physically viable. However, the results indicate they are accurate enough for this problem.

In the case of H-grid, a zero gradient in the η-direction is assumed for the symmetry boundary, i.e.

$$\left.\frac{\partial U}{\partial \eta}\right|_{i,1,k} = 0 . \tag{5.6a}$$

A backward-difference approximation is used to approximate the Eq. (5.6a) which results in

$$\bar{U}_{i,1} = \bar{U}_{i,2,k} . \tag{5.6b}$$

Also, the velocity components normal to this boundary is set equal to zero

$$w = 0 . \tag{5.6c}$$

5.5 Boundary Conditions for Blunt Leading-Edge

There are four boundaries in the computational domain with five different boundary conditions. They are upstream, downstream, outer, solid and symmetry boundary conditions. The top boundary (j=JL) contains the upstream and the outer boundaries. The upstream boundary condition is assumed to be the same as the freestream condition which can be expressed as

$$\bar{U}_{JL,k} = \bar{U}_{\infty} \cdot \tag{5.7}$$

Similar to the previous case, the outer boundary is located far away from the body to avoid any influence on the interaction region. Using a backward difference approximation, boundary condition for the top boundary can be expressed as

$$\bar{U}_{JL,k} = \bar{U}_{JL-1,k} \cdot \tag{5.8}$$

A zero gradient in the ζ-direction is assumed for the downstream boundary. Using a backward-difference approximation, the following can be written

$$\bar{U}_{j,KL} = \bar{U}_{j,KL-1} \cdot \tag{5.9}$$

At the wall, all velocity components are assumed to be zero. A zero normal pressure gradient is assumed for the solid surface, which can be expressed as

$$P_{1,k} = [4\, P_{2,k} - P_{3,k} + 2\, C_3\, (P_{2,k+1} - P_{2,k-1})]/3 \,, \tag{5.10}$$

where

$$C_3 = \frac{\eta_y \zeta_y + \eta_z \zeta_z}{\eta_z^2 + \eta_y^2} .$$

A zero gradient in the y-direction is assumed for the symmetry boundary, application of a backward-difference approximation yields the following

$$\bar{U}_{j,1} = \bar{U}_{j,2} . \qquad (5.11a)$$

Also, the velocity component normal to this boundary is set equal to zero

$$v = 0 . \qquad (5.11b)$$

Chapter 6

ADAPTIVE GRID GENERATION

For hypersonic flow about blunt bodies (Fig. 2.3), the temperature, pressure and density of the flow increases almost explosively across shock wave. At the same time, curved shock waves are close to the body. Numerical simulations of this phenomena have been a great challenge to computational fluid dynamics researchers. Presently, there is a great deal of interest in improving the quality of numerical simulation techniques and grid adaption is one way to achieve this goal.

As stated earlier, grid generation is the first step in the numerical solutions of partial differential equations for complex geometric domains. Basically, grid generation is the creation of boundary-fitted curvilinear coordinates. The second step is the construction of difference equations for the partial differential equations. It is apparent that the accuracy of finite difference solutions depends on the fineness of the grid. Therefore, the finer the grid, the more accurate the numerical solution will be. Also, the accuracy of solutions depends on the resolution of the solution gradient. The presence of large gradients causes the error to be large in the difference approximation of derivatives. In the presence of a shock wave, more artificial diffusion must be added to retain adequate smoothness of the solutions. Therefore, there is a need for schemes that can resolve large gradients without adding additional grid points. An

adaptive scheme moves the grid points to regions of high gradients, when the locations of these gradients are not known a priori. Also, an adaptive method reduces the total number of grid points required to achieve a given accuracy, but it requires more computer time. In some instances, the computer time makes this method impractical. The ideas used in the construction of adaptive grid techniques are limited only by one's imagination; and any scheme that works in the sense of providing a better solution is a good one. The ultimate answer to numerical solutions of partial differential equations may well be to dynamically adapt grids, rather than to devise more elaborate difference representations and solution methods [11, 18].

6.1 Literature Survey

Adaptive methods have been used in the solutions of ordinary differential equations. In order to control the local truncation error [19], variable-step initial-value problems are solved by adjusting the step size as the integration advances. Adaptive methods have also been implemented for solving equations of motion in conjunction with the method of lines [20]. In this case, the time step is automatically adjusted to control local error. Similarly, adaptive methods have been used to solve boundary value problems [21-26]. An optimal grid for a two-point boundary value problem can be determined either implicitly or explicitly. In the implicit approach, the weight function depends upon the solution. As a result the original boundary value problem is converted into an augmented system in which the dependent variables and the grid are computed simultaneously. In the explicit approach, the weight function does not depend on the solution. Instead, it depends

upon a previously calculated solution. In the implicit approach, a nonlinear two-point boundary value problem should be solved even for a linear problem. Implicit techniques do not preserve the linear/nonlinear character of the original problem. Moreover, even for the nonlinear problem, the augmented system is usually more difficult to solve than the original problem. On the other hand, the explicit technique preserves the linear/nonlinear character of the original two-point boundary value problem.

Adaptive schemes are divided into two basic categories: differential and algebraic. Differential methods are based on a variational approach. Brackbill and Saltzman [27-30] have developed a technique for constructing adaptive grids using a variational approach. In their scheme, a function which contains a measure of grid smoothness, orthogonality and volume variation is minimized by using a variational principle. The smoothest grid can be generated by solutions of Laplace equations which are better known as elliptic systems [31-32]. This approach ignores the effects of orthogonality, and it is very slow. This method has been modified for better efficiency by dropping the second derivative terms in one coordinate direction [33]. This makes the equations parabolic, therefore they can be solved by a marching technique. A method which considers the orthogonality and volume variation has been developed by Steger and Sorenson [34]. This method is widely known as the hyperbolic method, and it can be solved by a non-iterative marching technique. The variational approach provides a solid mathematical basis for the adaptive methods, but the Euler-Lagrange equations must be solved in addition to the original governing differential equations. On the other hand, an algebraic method requires much less computational effort, but the grid may not be smooth.

Rai and Anderson [35-40] have developed an algebraic technique where the grid movement is governed by estimates of the local error in the numerical solution. This is achieved by requiring the points in the large error regions to attract other points and points in the low error region to repel other points. Nakahashi and Deiwert [41] have formulated an algebraic method which is based on the variational principle. To reduce the overall solution error, a spring analogy is used to redistribute the grid points. In this case, operator-splitting and one-sided controls for the orthogonality and smoothness are used to make the method practical, robust and efficient. Dwyer [42-45] has used an adaptive method in which the points are moved along one set of the original coordinate lines in response to the evolving gradients in the physical solution. The analysis shows that the percentage change in a dependent variable can be determined a priori. Improvement in speed by an order of magnitude is obtained, but some problems with excessive skewness are encountered.

Generally, dynamic adaption can be performed in two ways. One is to keep the computational space fixed and include the grid speed in the flow field equations. This is an ideal method to use for unsteady flow. The second way is to set the grid speed equal to zero and interpolate the solution onto the new grid after each adaption. The first way is an ideal method to use in unsteady flow while the second way is equivalent to solving a sequence of boundary-value problems and is an economical way to treat steady flows where solutions are approached asymptotically. In the second approach it is generally sufficient to adapt just a few times during the course of the computation. In this approach, the grid distribution at time N+1 is determined from time N. Dwyer adapts the

grid points after each integration step or after a selected number of steps [42]. However, the grid speed can be obtained by postulating a law governing it which is based on some solution properties. These equations can be integrated with the governing partial differential equations to yield the new grid distribution [46]. The advantage of this technique is that the location of grid points and grid speed are time accurate.

The literature survey indicates that most techniques adapt to just one variable. This means that the weight function is based on the solution of one variable only. However, the solutions of equations of motion produce several dependent variables. Viscous-hypersonic flow over a blunt-body has large gradients in pressure, velocity, etc. in different parts of the flow field. For instance, there is a large gradient in pressure near the shock region, and at the same time there is a large gradient in velocity near the solid body. Therefore, there is a need for the development of an efficient grid adaption method which utilizes several variables simultaneously.

6.2 Methods of Grid Adaption

One reason to use grid adaption is to minimize the error over some domain by rearranging the grid. Calculus of variations can be used to perform this minimization. In general, a weighted integral, which is a measure of some grid or solution property over some domain, can be expressed as

$$I = \int_{\forall} W \, d\forall, \tag{6.1}$$

where W is the weight function to be minimized. The selection of W may vary from problem to problem. There is a collection of definitions for W in [1]. The weight function can be based on grid properties such as cell-volume, the average of the square of diagonal lengths, the cell area/volume ratio and cell skewness [33]. There exists a differential equation which minimizes the integral I in Eq. (6.1). This differential equation is called the Euler-Lagrange equation [47]. The Euler-Lagrange equation can be found in [27-30].

6.3 Multidimensional Grid Adaption

Brackbill and Saltzman [27-30] have developed a technique based on a variational approach. In their scheme, a function which contains measures of grid smoothness, orthogonality, and volume variation is minimized. To maximize the smoothness of the grid, the following integral must be minimized

$$I_s = \int_{\forall} \sum_{i=1}^{3} \nabla \xi^i \cdot \nabla \xi^i \, d\forall \, . \tag{6.2}$$

This is simply the sum of the squares of cell-edge lengths. Similarly, orthogonality can be acquired by minimizing the integral I_o

$$I_o = \int_{\forall} (\nabla \xi^i \cdot \nabla \xi^j)^2 \, d\forall \, , \tag{6.3}$$

with (i,j,k) cyclic. This integral vanishes for an orthogonal grid. The concentration or cell-volume variation can be obtained by minimizing the integral

$$I_w = \int_{\forall} W \, J \, d\forall \, , \tag{6.4}$$

where W is a specified weight function. This causes the cells to be small where the weight function is large. The grid generation system which provides smoothness, orthogonality and concentration is obtained by minimizing the total integral I which is a linear combination of I_s, I_o and I_w

$$I = I_s + \lambda_o I_o + \lambda_w I_w . \tag{6.5}$$

The competing features such as smoothness, orthogonality, and cell volume variation can be stressed by the proper choice of the coefficients λ_o and λ_w. For example, a large λ_o will result in a nearly orthogonal grid at the cost of the smoothness and the concentration. The Euler-Lagrange equations for the sums of those individual integrals form the system of partial differential equations from which the coordinate system is generated. The equations are quasi-linear, second-order partial differential equations with coefficients which are quadratic functions of the first derivatives [29]. This variational formulation is equivalent to Winslow's method [31] where λ_o and λ_w are set equal to zero. The Euler equations are those given by Winslow, and their solution maximizes the smoothness. This is also used by Thompson et al. [32]. The additional terms alter other characteristics of the mapping in a similar way. The cell-size variation and skewness can be controlled by proper selection of I_s, I_o and I_w. The use of a variational approach provides a solid mathematical basis for grid adaption. But, the Euler-Lagrange equations must be solved in addition to the governing equations of fluid motion. For further information, readers are refered to excellent articles by Thompson [11, 18].

6.4 One-Dimensional Grid Adaption

The Euler-Lagrange equation Eq. (6.5), is general and capable of adapting grids simultaneously in multiple dimensions. When the solution varies predominately in a single direction, one-dimensional adaption can be applied with the grid points constrained to move along one family of fixed curvilinear coordinate lines. The fixed family of lines is established by generating a full multidimensional grid using any standard grid generation technique. The points generated for the initial grid together with some interpolation procedure, e.g., cubic or linear interpolation, serve to define the fixed lines along which the grid points will move during the adaption. This is done explicitly, therefore there is no need to solve any differential equations.

A technique called equidistribution is developed to improve the solutions of boundary value problems [21-26]. This technique has proven to be effective and efficient. This technique is used to minimize the error by redistributing grid points such that a weight function is constant over each interval. The Euler-Lagrange equation is

$$x_\xi \, W = \text{constant} \, . \tag{6.6}$$

This minimizes the following integral

$$I_1 = \int_0^1 W(\xi) \, x_\xi^2 \, d\xi \, . \tag{6.7}$$

Equation (6.7) represents the energy of a system of springs with the spring constant $W(\xi)$, spanning each grid interval. The weight function is associated with the grid points themselves and not with

their locations. An alternative viewpoint results from integrating over x, instead of ξ, i.e., summing over the grid intervals rather than over the grid points. This can be expressed as

$$I_2 = \int_0^1 \frac{\xi_x^2}{W(x)}\, dx \ . \tag{6.8}$$

The Euler-Lagrange equation for this formulation is given by Eq. (6.6). ξ_x is considered to represent the point density. This variational problem represents a minimization over the density of the grid points subjected to a weight function. This can produce smooth grid distributions. Here the weight function W(x) is associated with the location of grid points. If the weight function is associated with the grid points themselves rather than their locations, $W = W(\xi)$. Equation (6.6) is the Euler equation for the following integral

$$I_3 = \int_0^1 \left[\frac{\xi_x}{W(\xi)}\right]^2 d\xi \tag{6.9}$$

where ξ is a measure of the smoothness of the grid distribution, with the emphasis placed on smoothness in certain regions. This is inversely proportional to the weight function $W(\xi)$. Equation (6.6) is the Euler-Lagrange equation for the integrals in Eqs. (6.7-6.9) which can be written as

$$\xi(x) = \int_0^x W(t)\, dt \ . \tag{6.10}$$

Equation (6.10) can be written in terms of arc length as

$$\xi(s) = \int_0^s W(t)\, dt \ . \tag{6.11}$$

The weight function is used to reduce the grid point spacing where W is large, and the weight function should be some measure of the error. White [26] has suggested the following form of the weight function

$$W = [\alpha + |U^{(m)}|^2]^{1/2m} , \tag{6.12}$$

where α is constant. With m=1 and α=0, this becomes

$$W = |U_x| . \tag{6.13}$$

A combination of this equation and Eq. (6.6) yields,

$$U_\xi = \text{Constant} . \tag{6.14}$$

This choice replaces the grid points so that the same change in the solution occurs at each grid interval. This is simply the solution gradient. Taking n=1 and α=1 yields

$$W = \sqrt{1 + |U_x|^2} . \tag{6.15a}$$

Combination of Eq.(6.15a) and Eq. (6.6) results in

$$\sqrt{x_\xi^2 + U_\xi^2} = S_\xi = \text{Constant} . \tag{6.15b}$$

This produces a uniform distribution of arc lengths on the solution curve. White's results [25] indicated that the arc length form is favored. The disadvantage of this method is that the weight function near the extreme solution, i.e. U_x=0 locally, is treated as a flat region. Concentration near the solution extreme can be achieved by incorporating some effect of the second derivative (U_{xx}) into the weight function [42] such as

$$W = 1 + \alpha\, f(U_x) + \beta\, g(U_{xx}) \,, \tag{6.16a}$$

where α and β are positive parameters. Equation (6.10) can be rewritten in normalized form as

$$\xi(x) = \frac{\int_0^x W(t)\, dt}{\int_0^1 W(t)\, dt} \,. \tag{6.16b}$$

With the second derivative terms included, the value of α must be continually updated to keep the same relational emphasis or concentration. Therefore, a system of two equations and two unknowns must be solved for each fixed grid line [43-45]. It will be shown later that through the reformulation of Eq. (6.16b), the parameter can be found directly or the matrix inversion can be entirely avoided.

6.5 One-Dimensional Grid Adaption With Several Variables

Flow equation solutions consist of several variables. Therefore, the weight function should also be a function of more than one variable. It is desirable to devise a scheme in which grid points can adapt to several variables with control of the magnitude of adaption for each variable. In the case of high-speed flow, velocity has large gradients in some regions where pressure is constant or vice versa. In general, the weight function can be expressed as

$$W = 1 + \sum_{i=1}^{N} b_i\, f_i \tag{6.17}$$

where N is the number of variables, b_i are constants, f_i are variables or their derivatives, and 1 is for uniformity. A substitution of Eq. (6.17) into Eq. (6.16b) results in

$$\xi(S) = \frac{S + \sum_{i=1}^{N} b_i F_i(S)}{S_{max} + \sum_{i=1}^{N} b_i F_i(S_{max})} , \qquad (6.18)$$

where

$$F_i(S) = \int_0^S f_i(t)\, dt .$$

It should be noted that Eq. (6.18) is for adapting along a fixed grid line. To ensure that $\xi(S)$ increases monotonically, b_i and f_i should be positive. In order to keep the same relative emphasis of the concentration along each grid line, b_i should be computed based on some percentage of the grid points being allocated to each variable. The percentage of grid points assigned to a particular function f_i can be expressed as

$$R_j = \frac{b_j F_j(S_{max})}{S_{max} + \sum_{i=1}^{N} b_i F_i(S_{max})} , \quad j=1,2,\ldots,N . \qquad (6.19)$$

Rearranging this equation results in

$$[A_{ji}]\{b_j\} = \{c_j\} , \qquad (6.20)$$

where

$$A_{ji} = R_j \frac{F_i(S_{max})}{F_j(S_{max})} \qquad i \neq j$$

$$= R_j - 1 \qquad i = j, \quad c_j = -R_j S_{max}/F_j(S_{max}) .$$

Therefore, a system of N equations and N unknowns must be solved for each fixed grid line (N is the number of variables). This can be avoided by the reformulation of Eq. (6.18). The crucial steps are outlined here. Equation (6.19) can be rewritten as

$$b_j = \frac{R_j}{F_j(S_{max})} \left[S_{max} + \sum_{i=1}^{N} b_i F_i (S_{max}) \right] . \qquad (6.21)$$

A substitution of Eq. (6.21) into Eq. (6.18) results in

$$\xi(S) = \frac{S + W_{max} \sum_{i=1}^{N} \frac{R_i F_i(S)}{F_i(S_{max})}}{S_{max} + W_{max} \sum_{i=1}^{N} R_i} , \qquad (6.22)$$

where

$$W_{max} = \left[S_{max} + \sum_{i=1}^{N} b_i F_i (S_{max}) \right] .$$

A summation of Eq. (6.19) over all j values yields

$$\sum_{j=1}^{N} R_j = \frac{\sum_{j=1}^{N} b_j F_j (S_{max})}{S_{max} + \sum_{i=1}^{N} b_i F_i (S_{max})} . \qquad (6.23)$$

Rearrangement of Eq. (6.23) results in

$$\sum_{i=1}^{N} b_i F_i (S_{max}) = \frac{S_{max} \sum_{i=1}^{N} R_i}{1 - \sum_{i=1}^{N} R_i} . \qquad (6.24)$$

A substitution of Eq. (6.24) into Eq. (6.22) yields

$$\xi(S) = \frac{S}{S_{max}} \left[1 - \sum_{i=1}^{N} R_i \right] + \sum_{i=1}^{N} R_i \frac{F_i(S)}{F_i(S_{max})} . \qquad (6.25)$$

This reformulation avoids the need for continuous updating the b_i's to keep the same relative emphasis on concentration. Applying this equation, grid points can be adapted with more than one variable without any need for matrix inversions. Eiseman [48] proposed a very similar approach but did not elaborate on this.

Presently, Eq. (6.18) is approximated by a trapezoidal rule. For unequally-spaced data, this can be written as

$$F_i(S) = \int_0^S f_i(t)\, dt = \frac{1}{2} \sum_{j=1}^{JL} [(t_{j-1} + t_j)\, (f_i(t_j) - f_i(t_j - 1))] ,$$

where

$$F_i(0) = 0 .$$

In the initial stages of the solutions, there exist large oscillations in the flow and fluid properties. Consequently, the adapted grid will have these oscillations as well. In order to have a smooth grid, these oscillations can be smoothed out with the following filter

$$\bar{r}^{n+1}_{jk} = (\bar{r}^{n}_{jk} + \bar{r}^{n}_{j+1,k} + \bar{r}^{n}_{j-1,k} + \bar{r}^{n}_{j,k-1} + \bar{r}^{n}_{j,k+1})/5 , \qquad (6.25a)$$

where

$$\bar{r} = (y,z)^T .$$

This is equivalent to the Lapalace filter which can be expressed as

$$\frac{\partial^2 r}{\partial \eta^2} + \frac{\partial^2 r}{\partial \zeta^2} = 0 . \qquad (6.25b)$$

This slightly reduces the effect of adaption, but it filters out low frequency oscillations.

In some physical phenomena like shocks, the length of discontinuity is of the order of molecule size. If the weight function contains the derivatives with respect to physical coordinates, the adapted grid tends to have a very small spacing. This can be corrected by reducing the effect of the weight functions near the discontinuities. Therefore, the weight functions (f_i) can be multiplied by the following function

$$(1 - e^{-\frac{\Delta S}{\Delta S_{min}}})$$

where ΔS_{min} is some allowable minimum spacing. This function varies from zero to one and is proportional to the spacing. Also, this can be corrected by replacing the derivatives of physical coordinates with the derivatives of the computational coordinates.

After the new adapted grid has been created, the solutions are interpolated into the new grid distribution. Presently, a piece-wise linear spline has been used. However, this may create some problems in the case of unsteady flow. Either higher order interoplation should be used or the time derivatives of the grid point should be incorporated into transformed governing equations.

Chapter 7

RESULTS AND DISCUSSION

In previous experiments [9], the surface oil-flow patterns over the Butler wing at various angles of attack and at Mach number 3.5 show no signs of transition and the nature of the oil streak lines is typical of a laminar flow. Therefore, results are obtained only for laminar flow over a Butler wing at a Mach number of 3.5, Reynolds number of 2×10^6 (based on chord length), free stream static temperature of 390°R (216.67°K), wall temperature of 1092°R (606.67°K), length of 0.80 ft (0.2438 m), specific heat ratio --- of 1.4, specific heat at a constant volume of 4290 $(ft/sec)^2/R$, and at zero and ten degrees angle of attack. In this study, a two boundary grid generation (TBGG) technique [13] is used. This method is essentially an algebraic method. The application of the TBGG method requires that the entire body be sliced into different cross sections. These cross sections are obtained in the stream-wise direction by analytical descriptions of the wing surface, (Eqs. 2.1a-2.1c). Then, two types of grid are generated for this wing; the H-type and O-type. Then, results of both cases are compared and discussed.

7.1 H-Type Grid

In this case, the entire flow field is sliced into fifty-five stations in the stream-wise direction, and each station has 64x36 grid points (Fig. 7.1). There is a total of 126,720 grid points which take

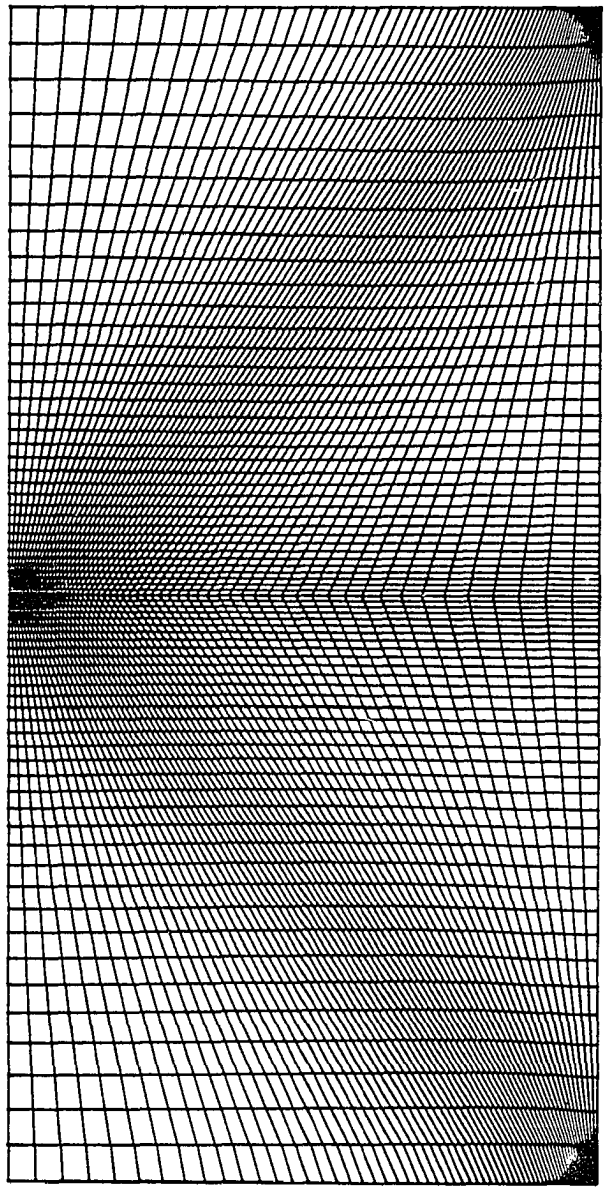


Fig. 7.1 H-type grid for a Butler wing

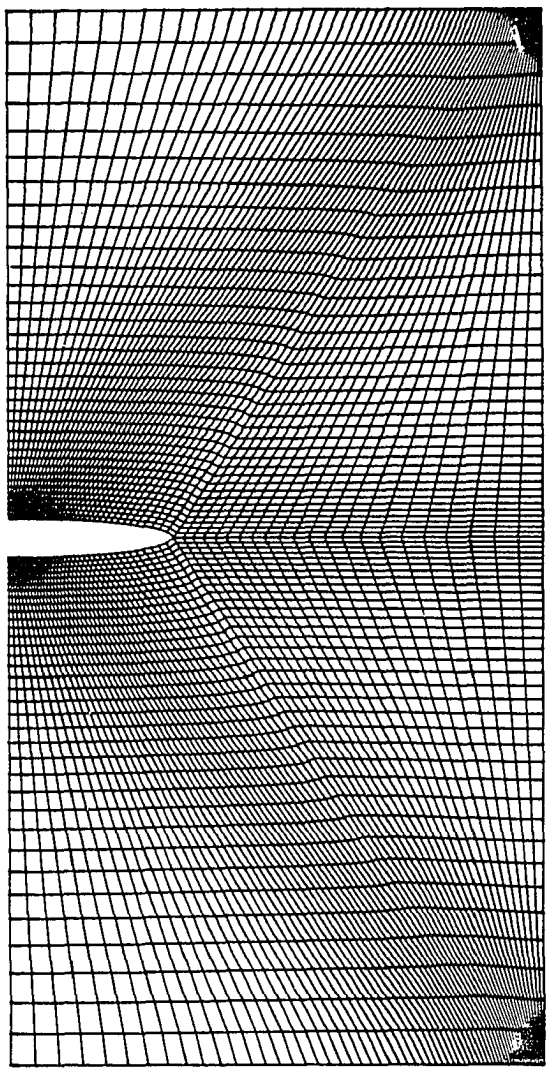


Fig. 7.1 Continued

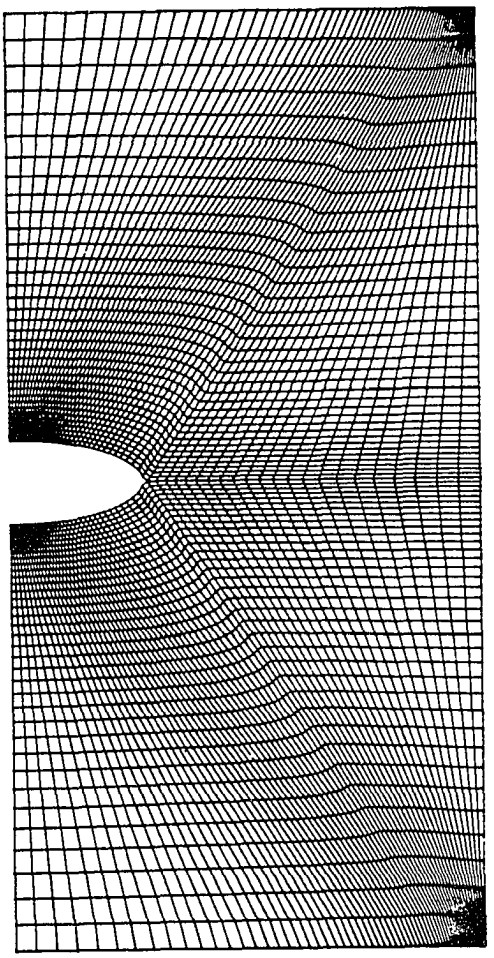
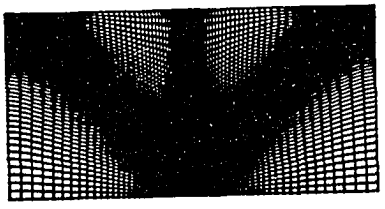
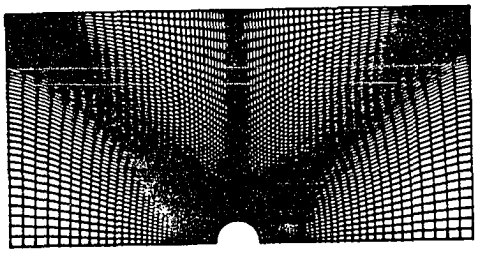


Fig. 7.1 Continued





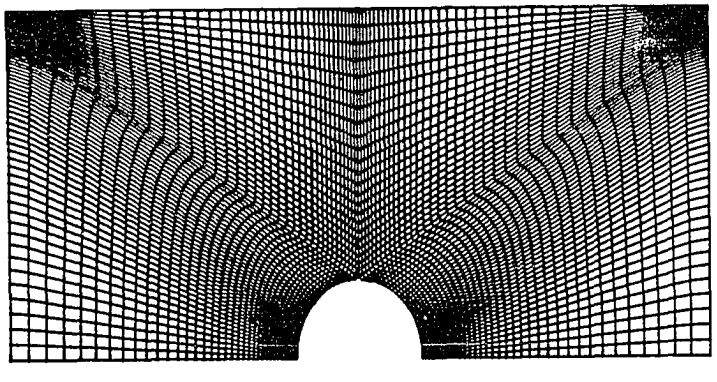


Fig. 7.1 Continued

2.8 million 32-bit words of primary memory (16 variables). The required computational time is 1.9×10^{-5} sec/grid-point/iteration (2.5 sec/iteration) which is typical for a CYBER 205 with two pipes. Results are obtained for zero angle of attack. The computed pressures are plotted in (Figs. 7.2a-7.2c). The pressure coefficient along the center line is shown in Fig. 7.2a. The results are compared with available experimental and numerical results [6-9]. The results on the center line are in excellent agreement with the experimental and previously obtained numerical results (Refs. 6-9). At 41.66% and 68.33% chordwise position, the pressure ratios are plotted against the conical span-wise coordinates $y/x\tan(\theta)$, (Figs. 7.2b-7.2c). They are in good agreement with experimental and numerical results. However, there are some discrepancies in the results between 30° and 60°. This is probably due to the fact that grid lines are not orthogonal near those regions and this may be a direct consequence of the H-type grid. Also in this case, the wing's leading edges are represented by a jagged-line, in other words grid lines are not along the leading edges of the wing. This does not allow us to compute the boundary conditions using a second-order accurate formula. Therefore solutions are not second-order accurate near solid boundaries.

7.2 O-Type Grid

For this case, the physical domain is limited to 5% to 95% of the wing. This is done to avoid any singularities. The conical Navier-Stokes solutions are enforced for the upstream boundary which is located at 5% of the wing. This solution is obtained by the integration of the Navier-Stokes equations [16] for a conical grid with proper boundary

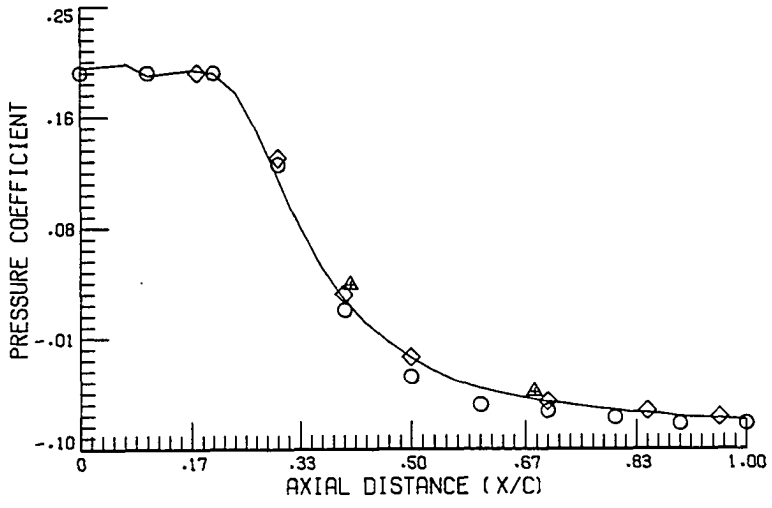


Fig. 7.2a Pressure coefficient along the center line

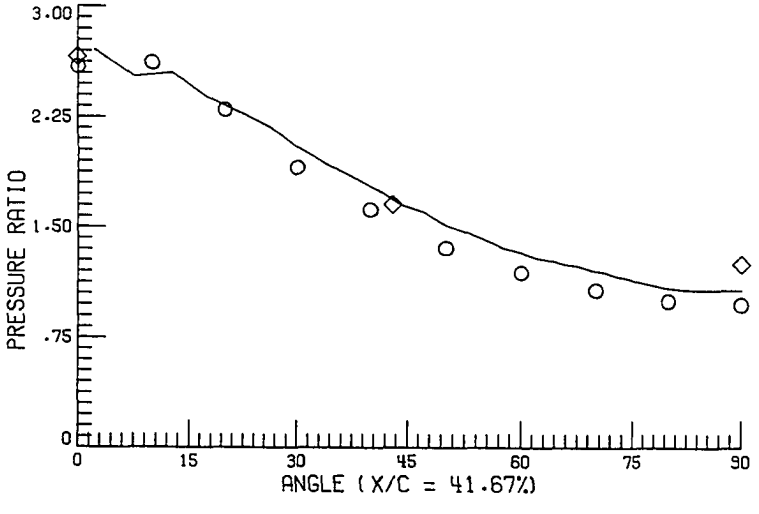


Fig. 7.2b Pressure ratio at 41.67% of chord

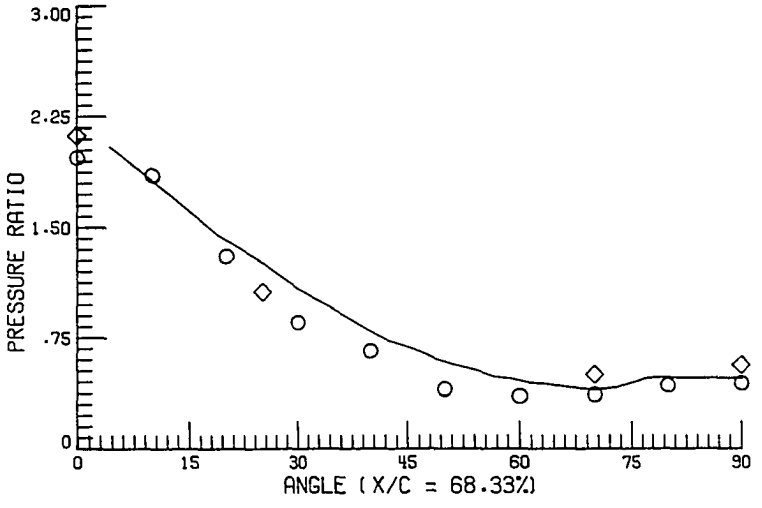


Fig. 7.2c Pressure ratio at 68.33% of chord

conditions. The wing is sliced into forty-one stations in the stream-wise direction, and each station has 41x127 grid points (Fig. 7.3). There is a total of 213,487 grid points which take 4.7 million 32-bit words of primary memory. Results are obtained for zero and ten degrees angle of attack.

Results for zero angle of attack are compared with results obtained with the O-type grid, with results from the experiments and other numerical results [6-9]. The computed pressure is plotted in Figs. 7.4a-7.4c. The pressure coefficient along the center line (Fig. 7.4a) is in good agreement with other numerical and experimental results. Nevertheless, there is some discrepancy near the nose region. This may be due to the fact that the upstream solutions are based on the conical solutions. But, these solutions match exactly with results from H-type grids. This is because the grid topology near the center line is the same for both grid types. At 41.67% and 68.33% chordwise positions, the pressure ratio is plotted against the conical span-wise coordinates $y/x\tan(\theta)$. They are in excellent agreement with experimental and numerical results (Figs. 7.4b, 7.4c). In addition they are much closer to the experimental results compared to the results obtained from the H-type grid. This may be due to good grid orthogonality in the case of an O-type grid, and the wing's leading edges are represented by a straight line. On the thick sections near the nose the pressure is highest on the centerline and falls toward the leading edge. Figure 7.4d shows the cross-flow velocity at 5%, 23%, 41%, 59%, 77% and 95%.

The results for ten degrees angle of attack are compared with experimental results. The computed pressure is plotted in Figs. 7.5a-7.5d. At 17%, 30%, 50% and 70% chordwise positions, the pressure ratio

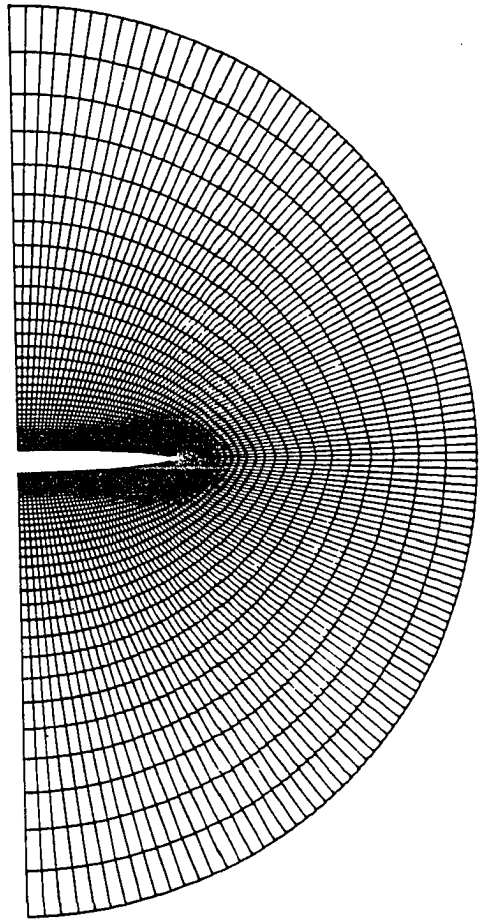


Fig. 7.3 O-type grid for a Butler wing

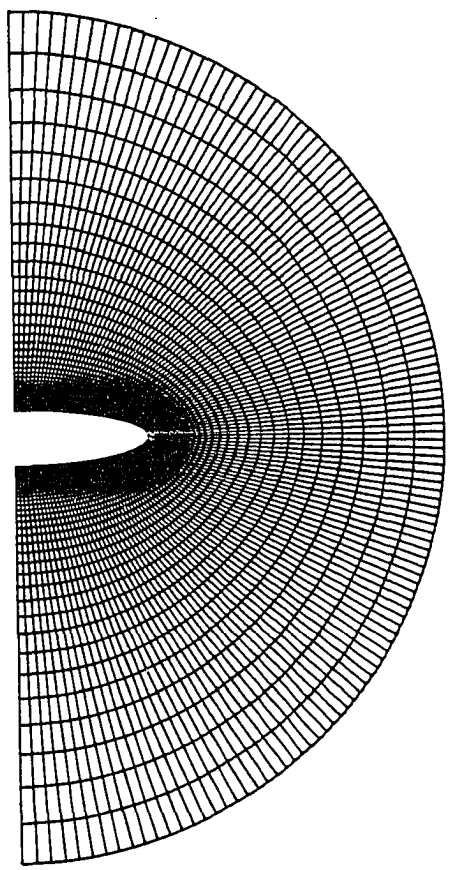


Fig. 7.3 Continued

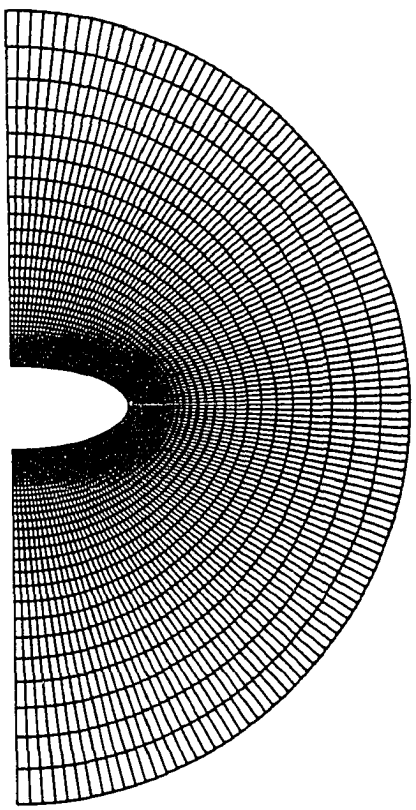
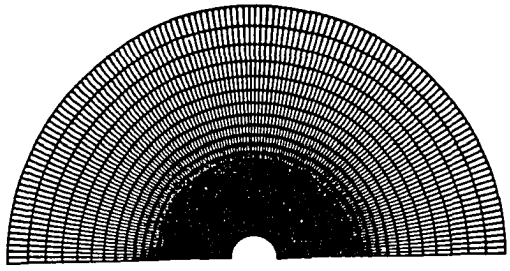
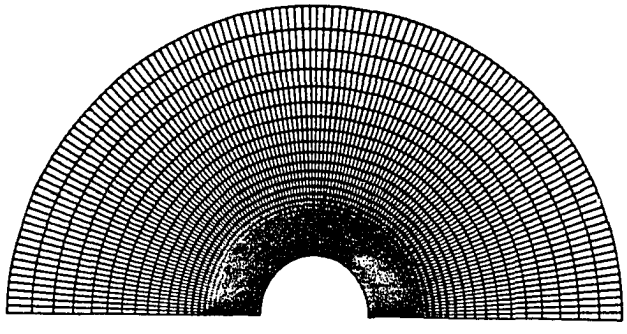


Fig. 7.3 Continued





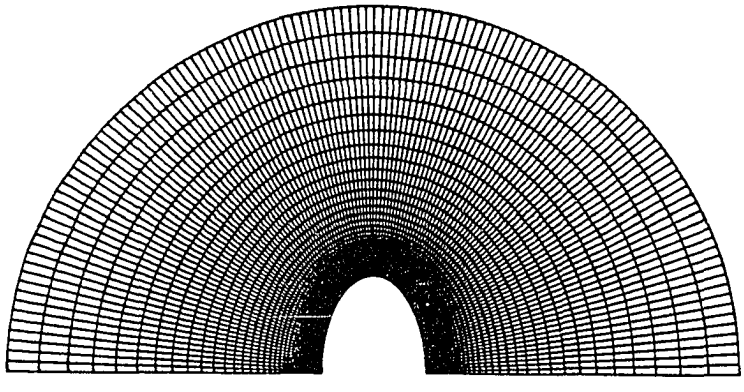


Fig. 7.3 Continued

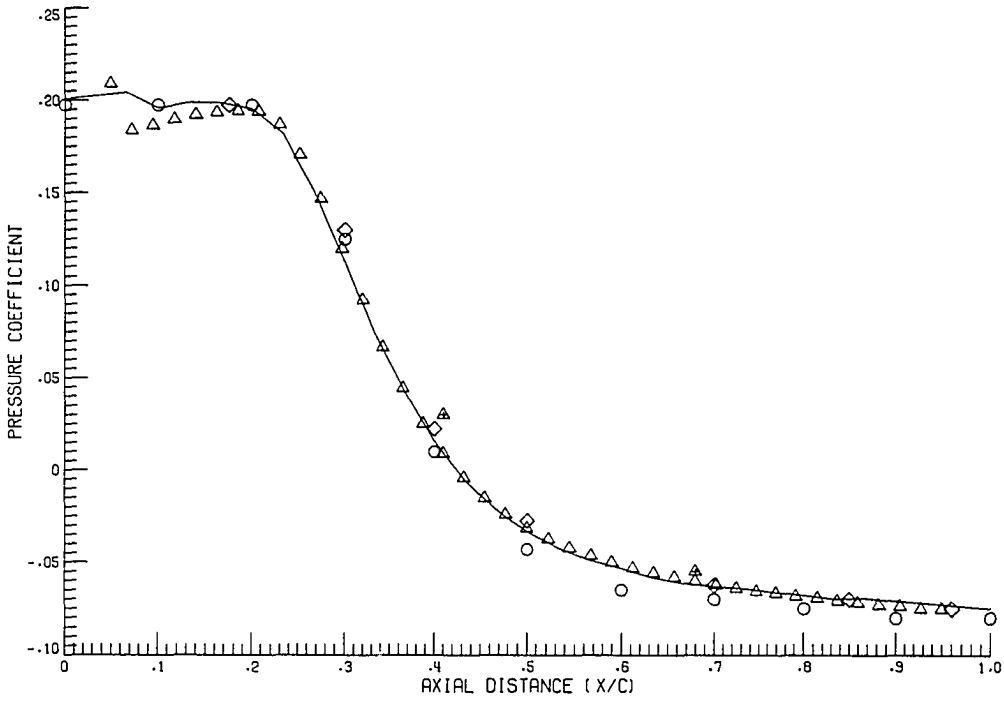


Fig. 7.4a Pressure coefficient along the center line

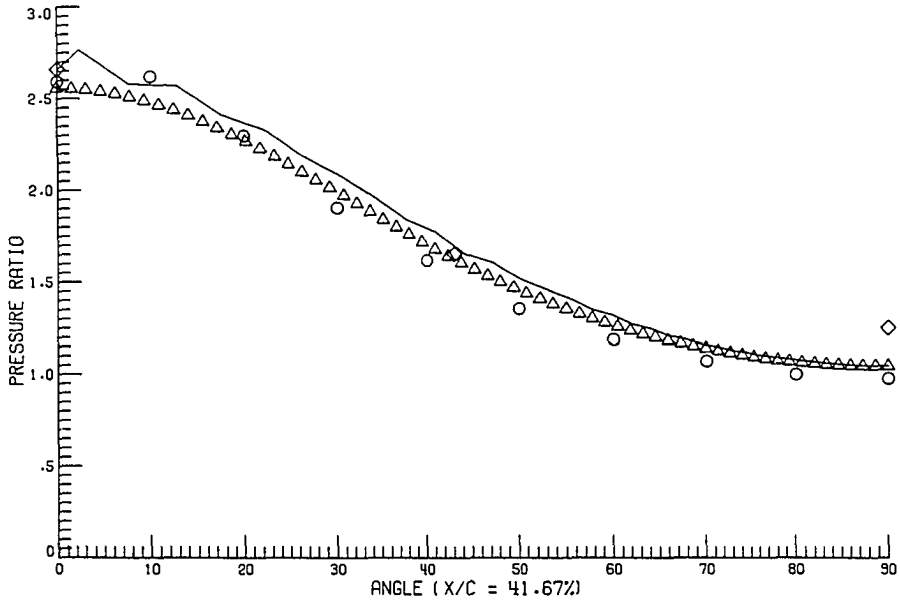


Fig. 7.4b Pressure ratio at 41.67% of chord

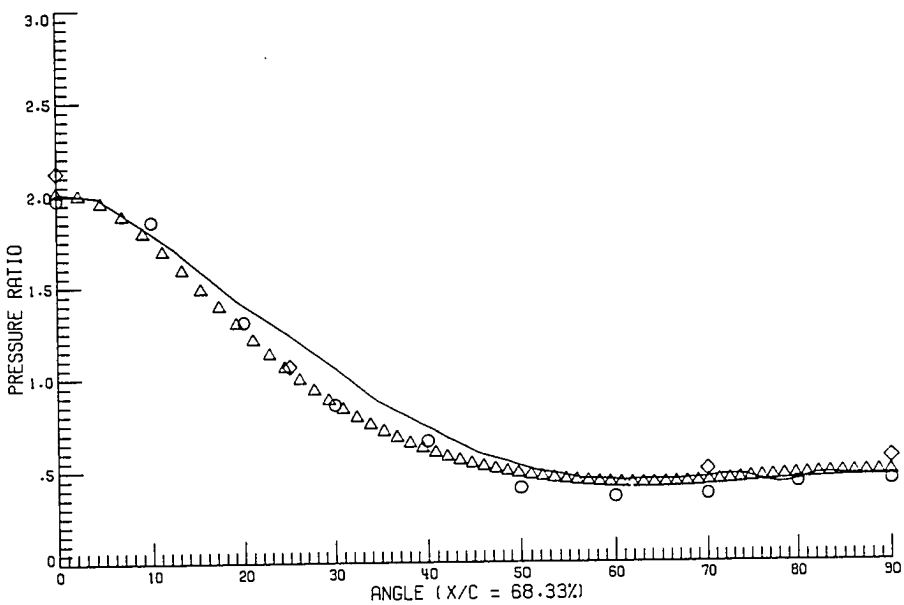


Fig. 7.4c Pressure ratio at 68.33% of chord

Fig. 7.4d Cross-flow velocity vectors
(zero angle of attack)

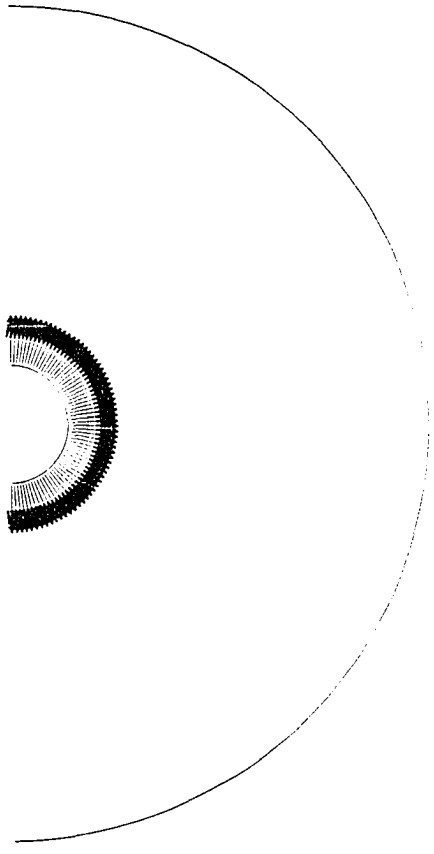


Fig. 7.4d Continued

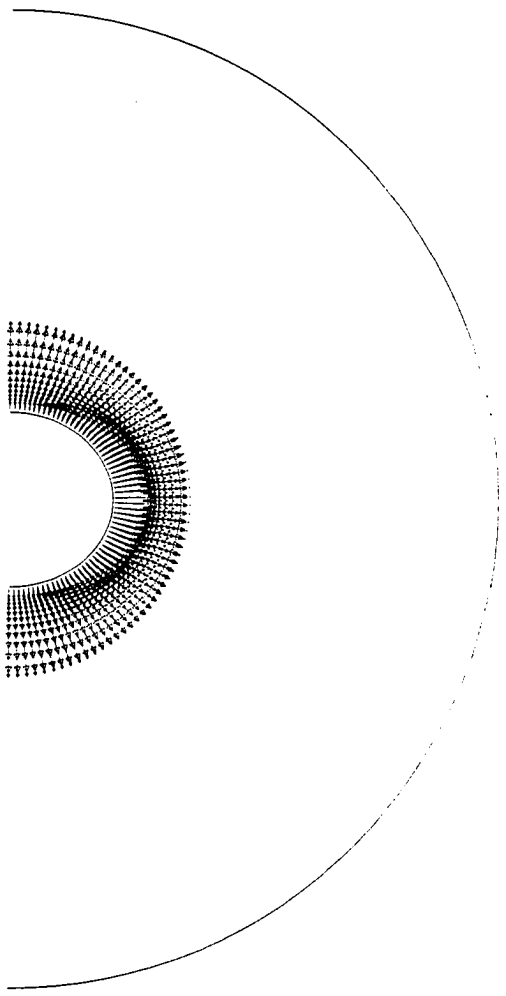


Fig. 7.4d Continued

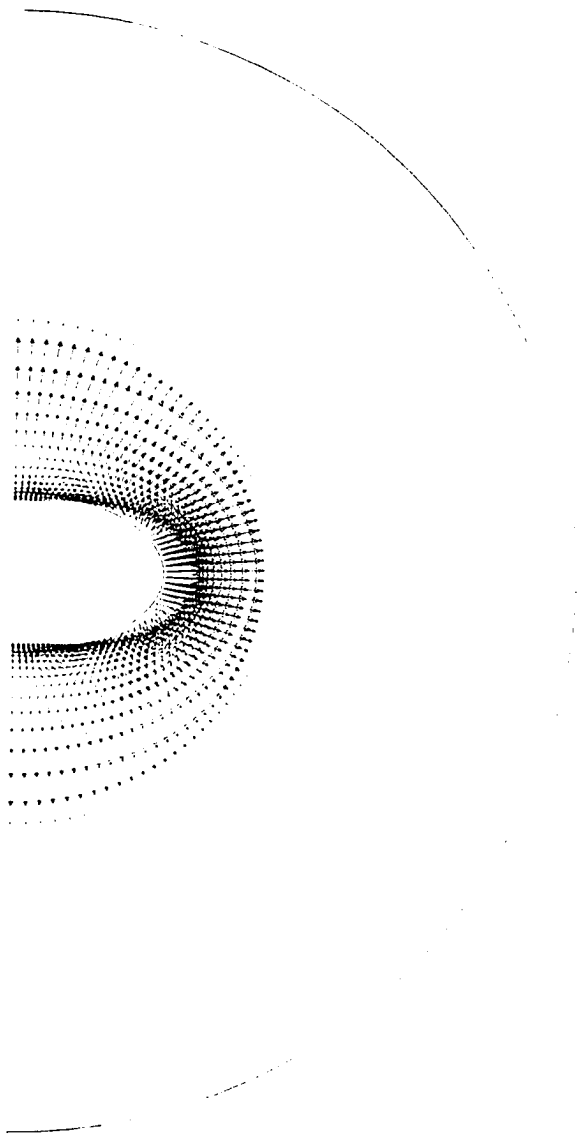


Fig. 7.4d Continued

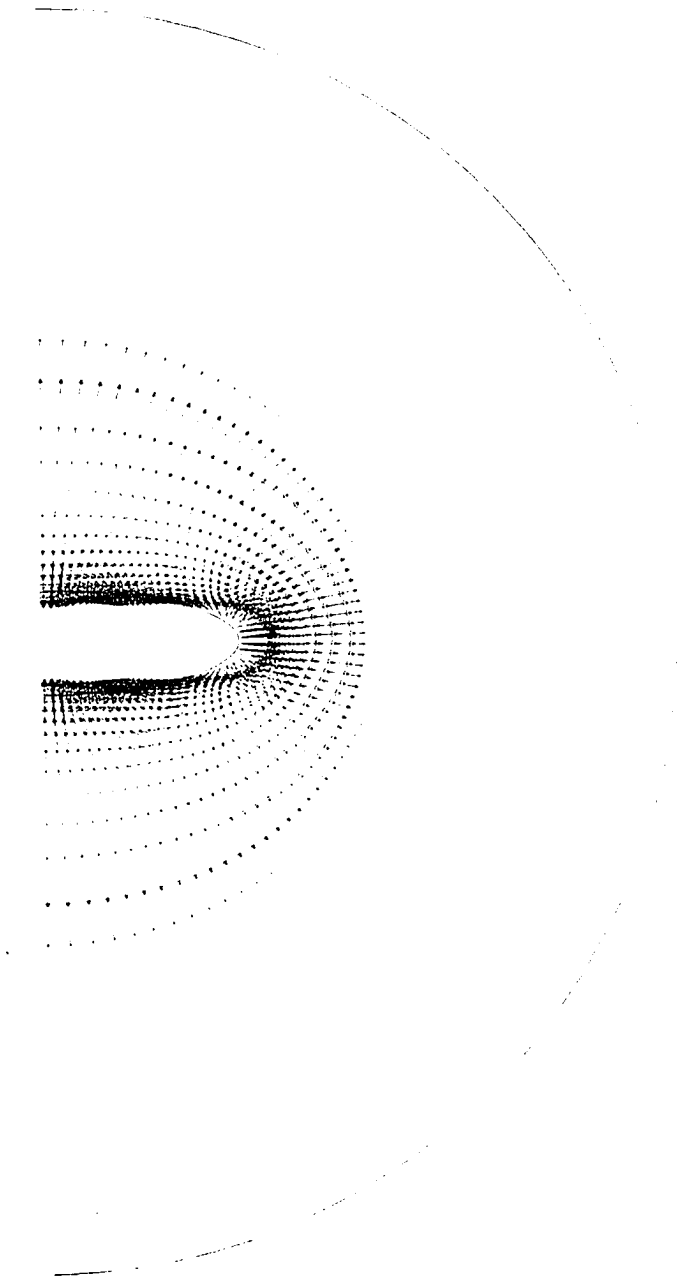


Fig. 7.4d Continued

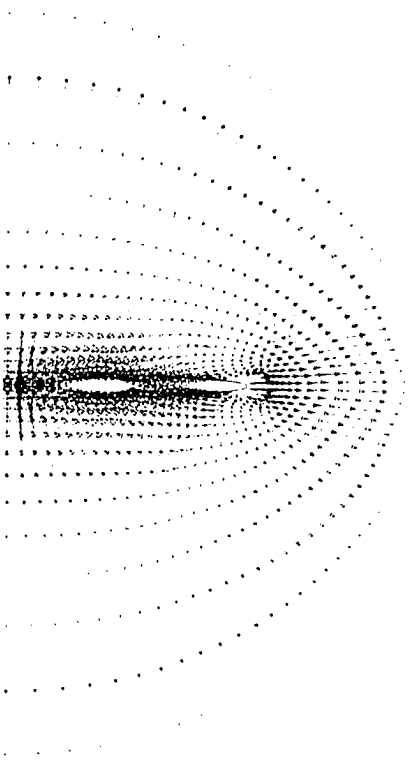


Fig. 7.4d Continued

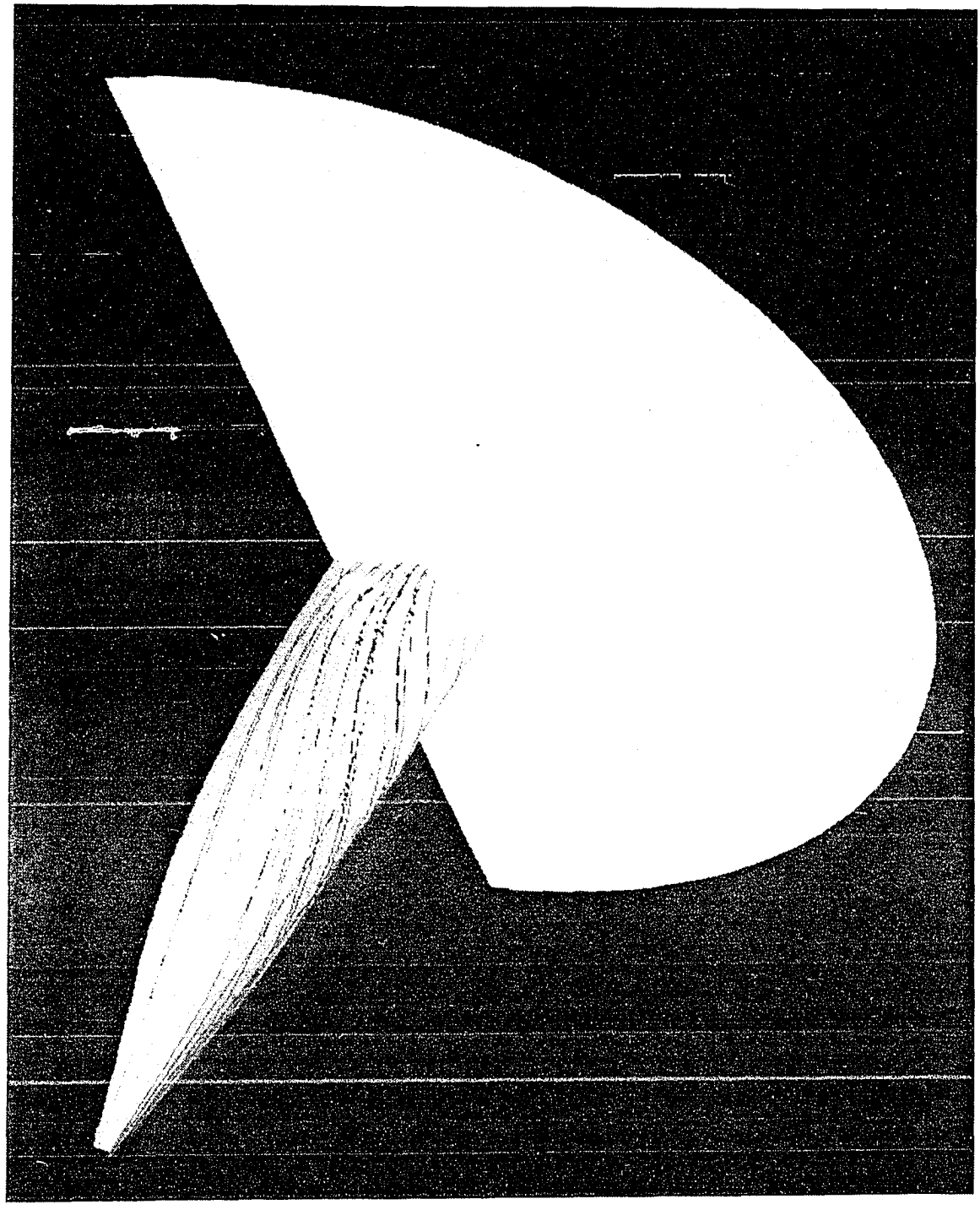


Fig. 7.4e Streak lines (zero angle of attack)

is plotted against the conical spanwise coordinate y/xtan(θ). They are in good agreement with experimental and numerical results (Figs. 7.5a-7.5d). As in the previous case, on the thick sections near the nose the pressure is highest on the centerline and falls toward the leading edge, whereas near the trailing edge the spanwise distribution is more "wing like" with the maximum pressure at the leading edge. The change-over is shown by the pressure peaks in the pressure distributions at x/c=0.5 and 0.7 at 10 degrees angle of attack. There are some discrepancies near x/c=30%-50%; this may be due to the fact that Squire [9] has not used the exact model of the Butler wing. In order to mount the model in the wind tunnel, the lower surface is distorted to include a sting support. Figure 7.5e shows the cross-flow velocity at 5%, 23%, 41%, 59%, 77% and 95%. These figures show a weak cross-wise separation on the suction side which is confined to the body. At 59%, the cross flow has separated but a well-defined vortex is not visible. Squire [7] has performed a series of tests to investigate the effects of thickness on the longitudinal characteristics of a delta wing with different aspect ratios. The tests on the thick symmetrical delta wings have confirmed that the lift curve slope decreases as the thickness increases. This loss of lift is associated with a weaker vortex system giving less nonlinear lift. Squire [9] also observed a pair of vortices at the trailing edges, but there was no sign of any span-wise flow outboard of these vortices. The Butler wing has a round leading edge for most of its length, and previous numerical experiments have indicated that flow-field solutions are inconsistent using both the Euler and the Navier-Stokes equations for this type of geometry [49]. The problem appears to be the determination of the initial location of separation over the

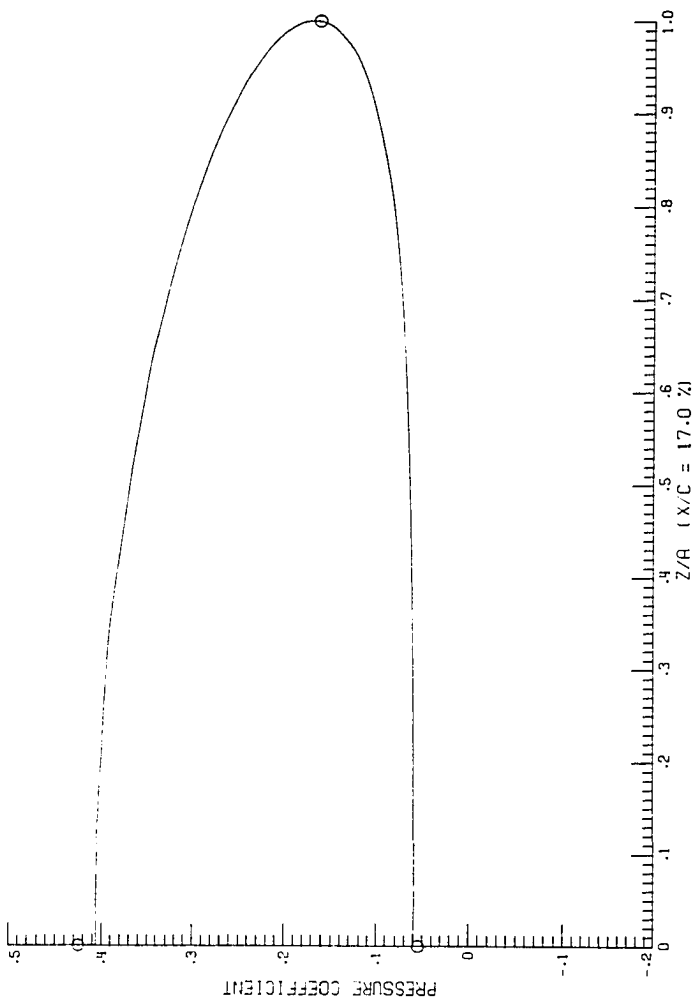


Fig. 7.5a Pressure coefficient (x/c=17%, ten degree angle of attack)

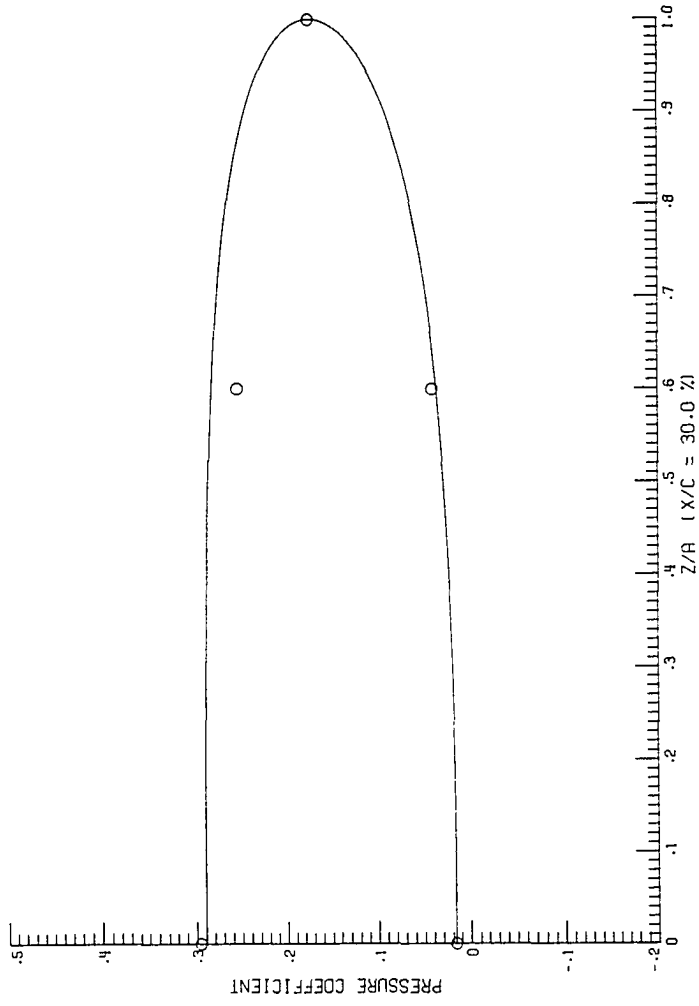


Fig. 7.5b Pressure coefficient (x/c=30%, ten degree angle of attack)

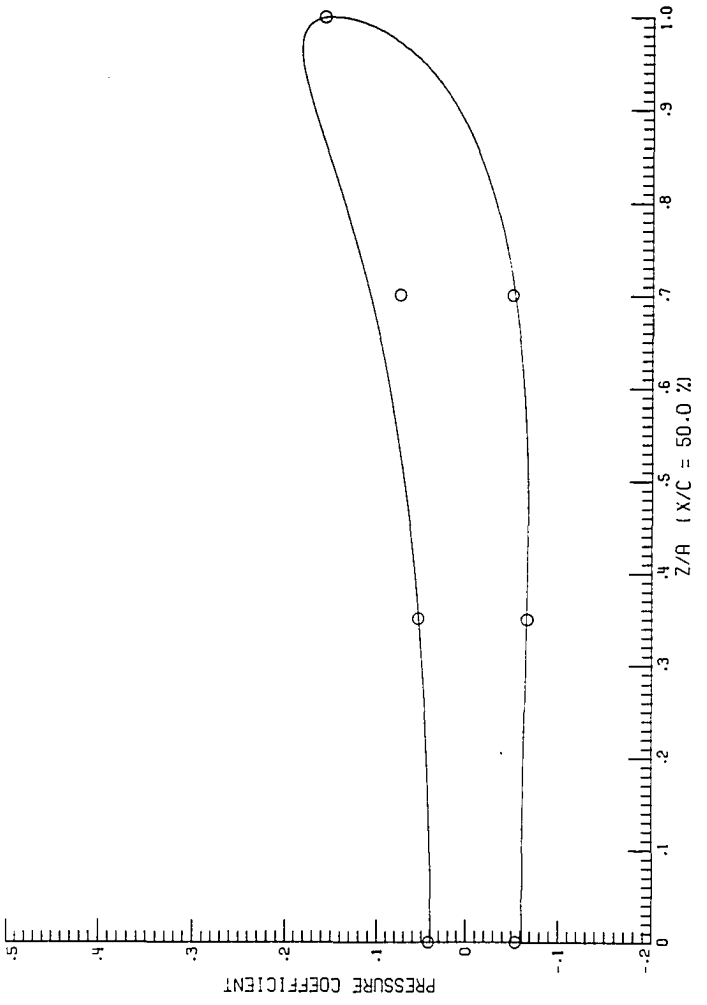


Fig. 7.5c Pressure coefficient (x/c=50%, ten degree angle of attack)

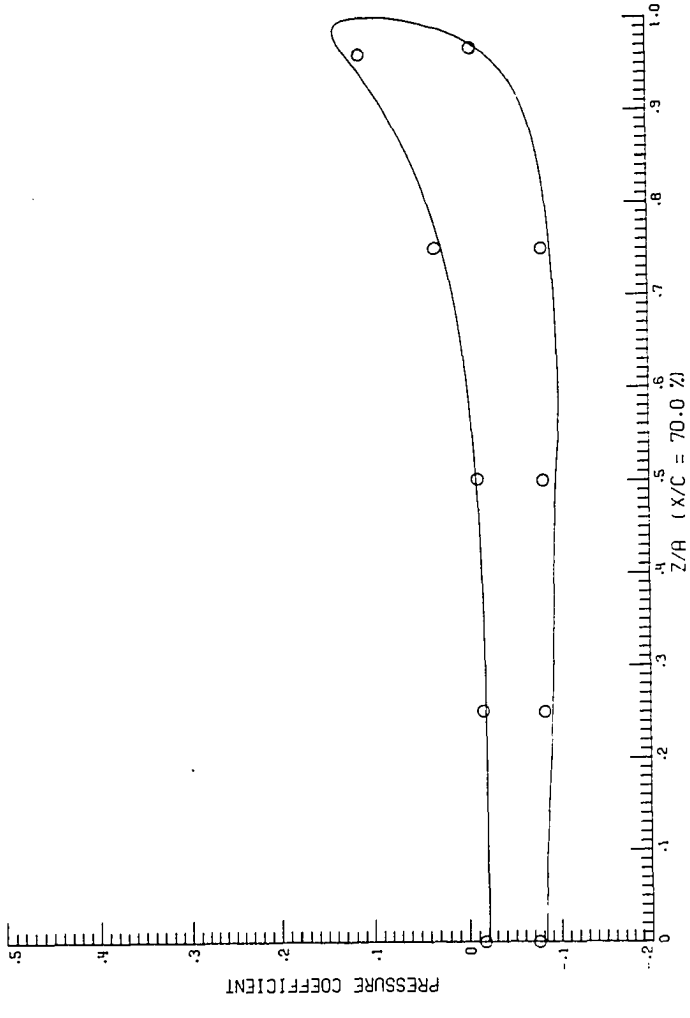


Fig. 7.5d Pressure coefficient (x/c=70%, ten degree angle of attack)

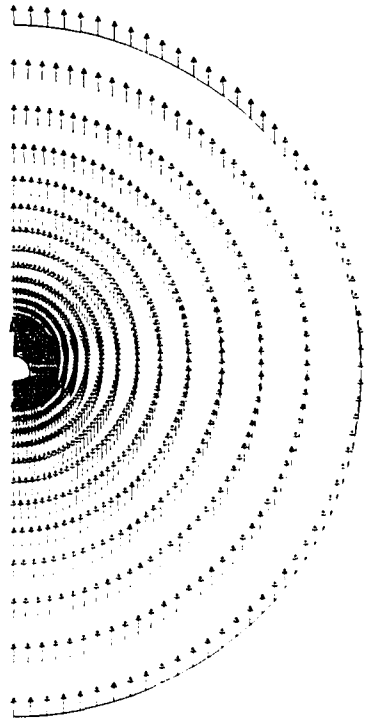


Fig. 7.5e Cross-flow velocity vectors
(ten degree angle of attack)

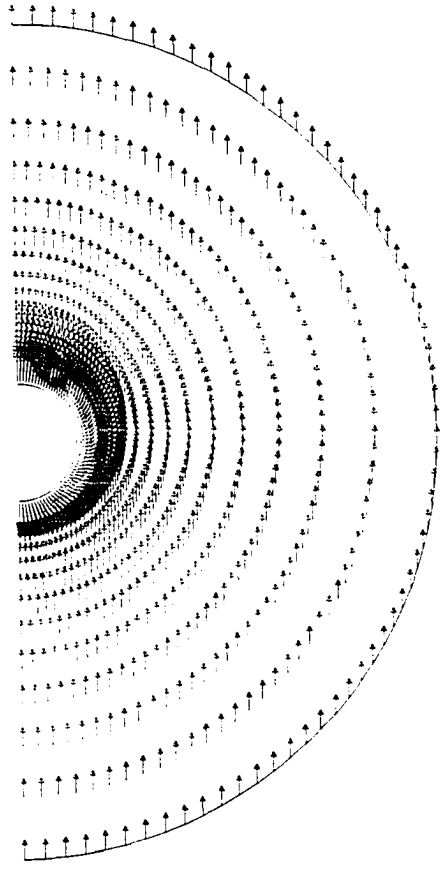


Fig. 7.5e Continued

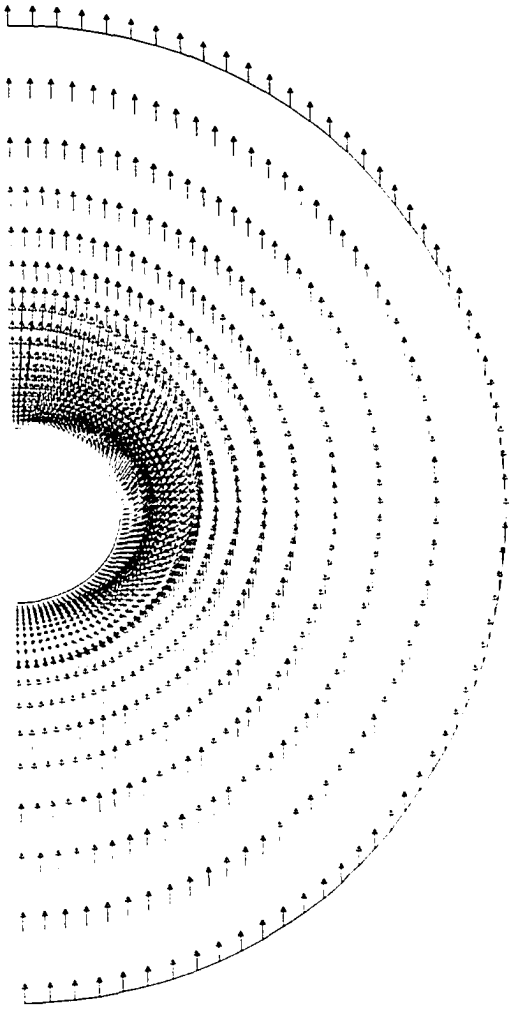


Fig. 7.5e Continued

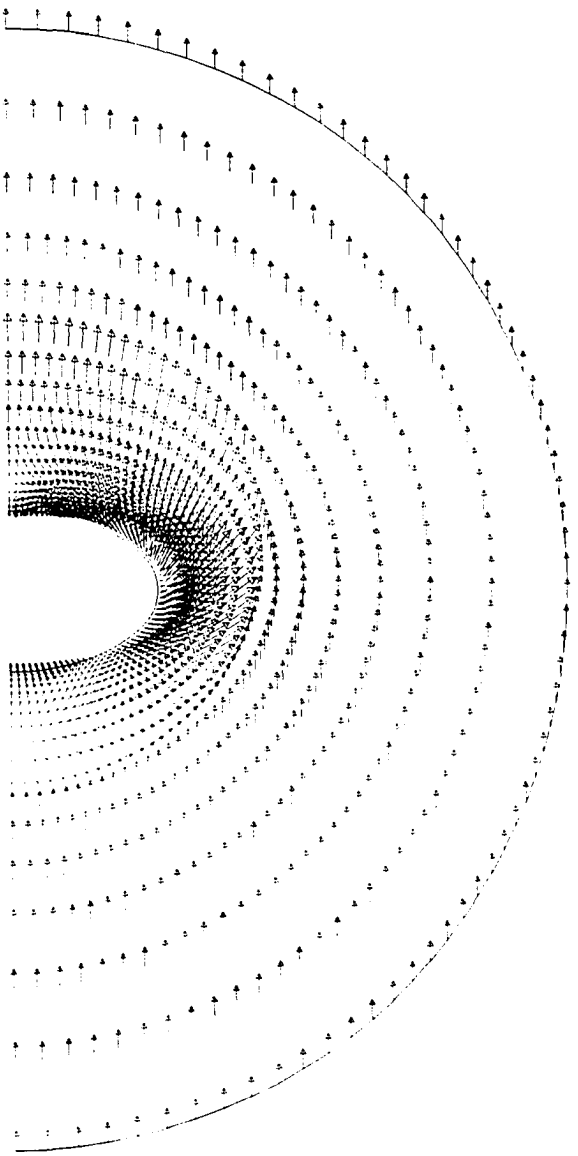


Fig. 7.5e Continued

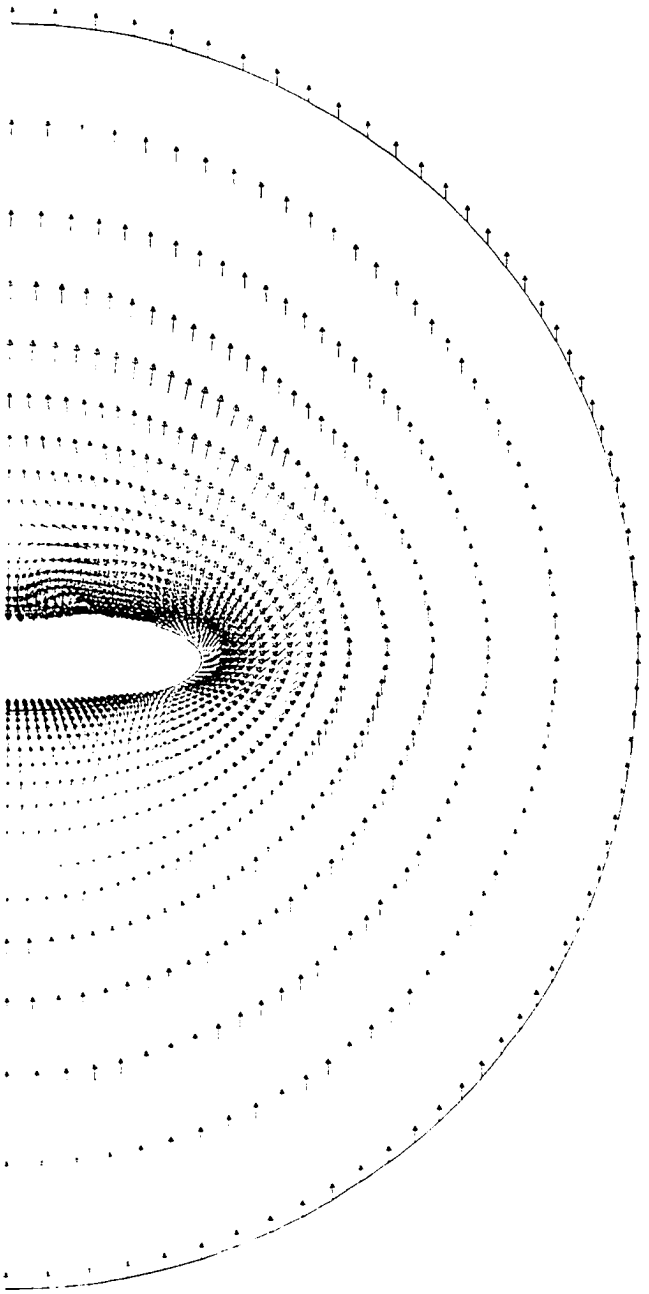


Fig. 7.5e Continued

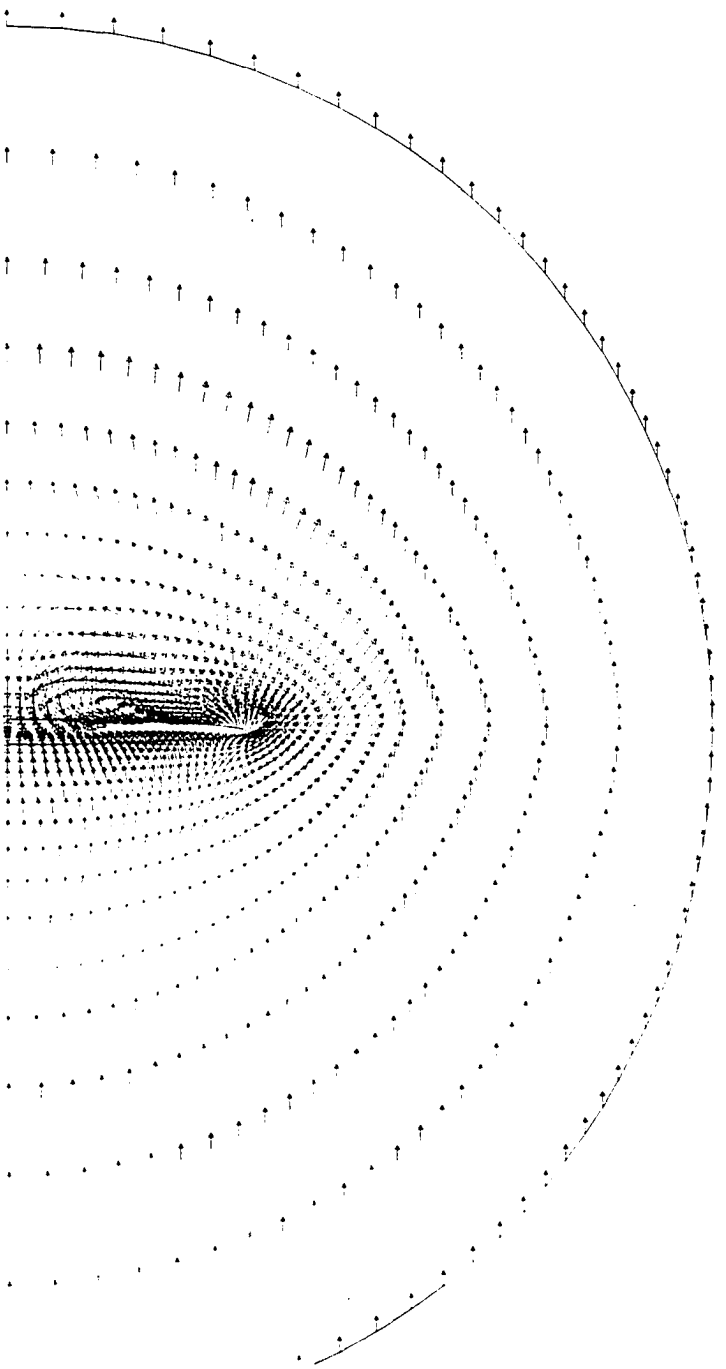


Fig. 7.5e Continued

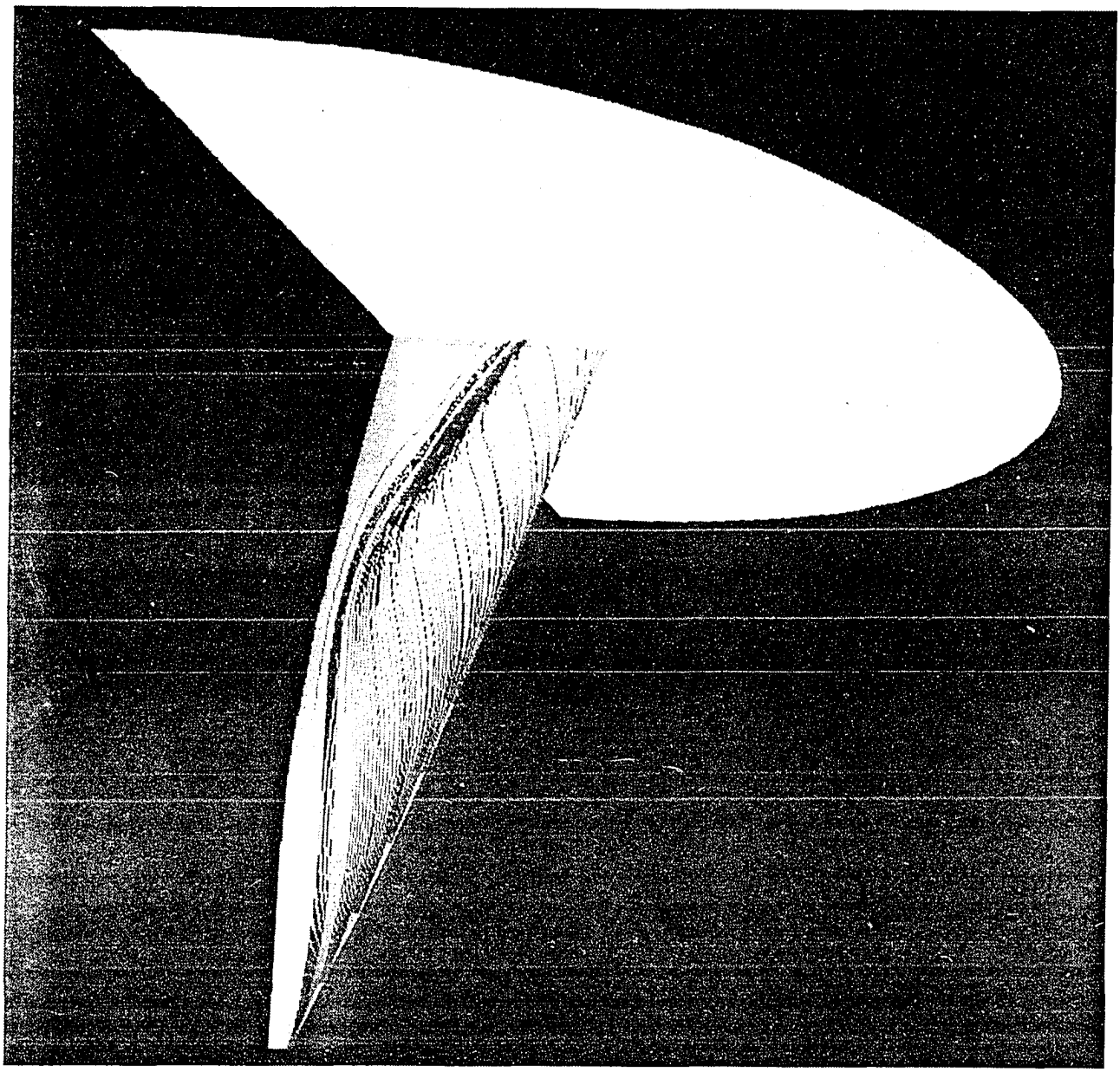


Fig. 7.5f Streak lines (ten degree angle of attack)

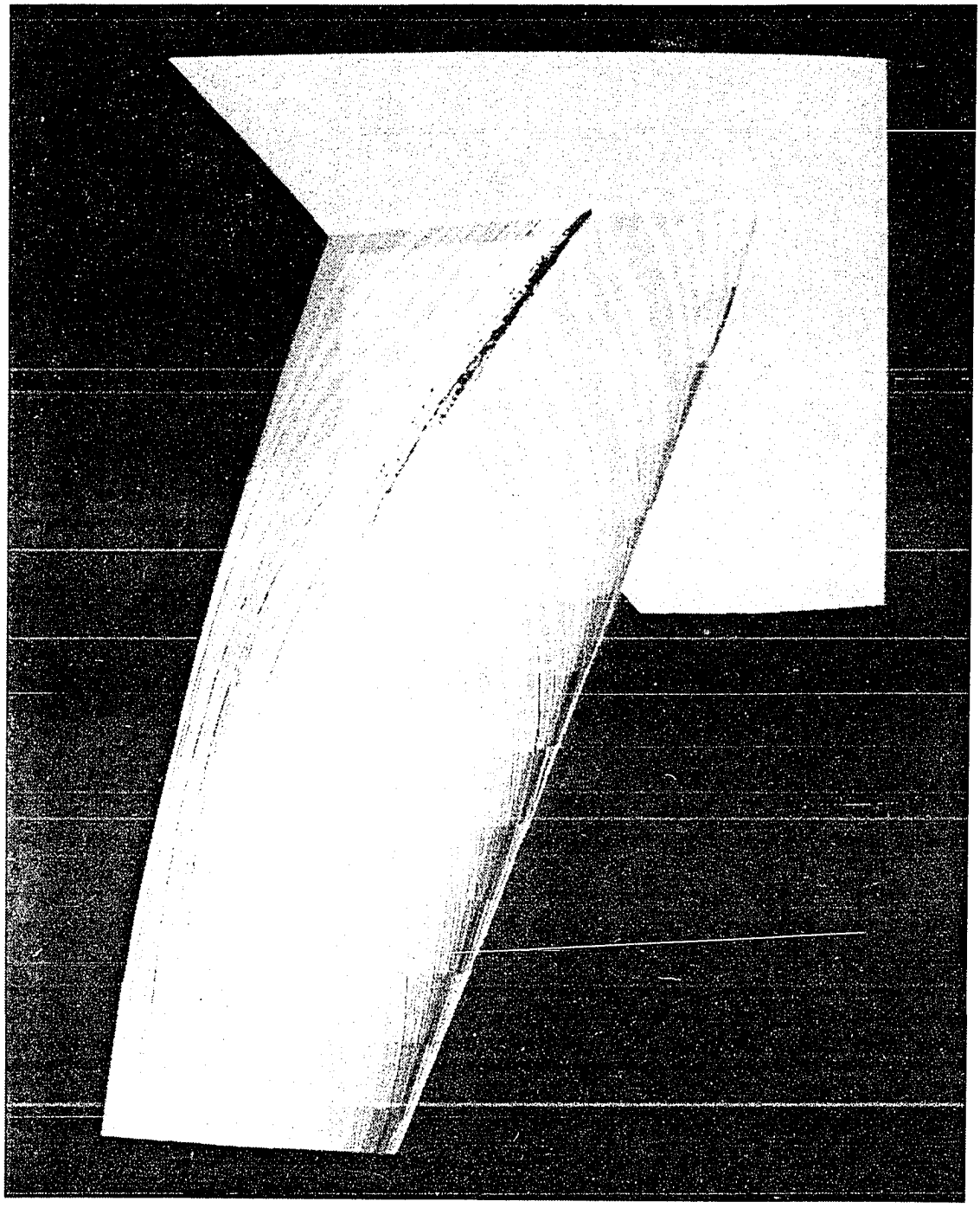


Fig. 7.5f Streak lines (ten degree angle of attack)

suction side. If the leading edge is sharp, there is no problem since the separation line is along the leading edge. For a rounded leading-edge, the separation point is not defined uniquely because the numerical dissipation influences where the separation point occurs [49]. The leading-edge separation for the rounded-edge may be a numerical phenomenon. It is possible that this problem of uniqueness occurs in this study as it does with all other numerical experiments, having rounded leading edge and artificial dissipation applied in the solution technique.

The question has been raised concerning the application of the Navier-Stokes equations in a flow field study when only the pressure solution and streak lines are compared with experimental data. It is argued that the solution of the Euler equations would produce the same information with a high degree of accuracy. This is a legitimate argument if the sole purpose of the study is to collect inviscid information. However, we believe that the solution of the Navier-Stokes equations provides considerably more information if there are enough grid points in the proper locations. Also, the solution of the Navier-Stokes equations provides experience for the future when computer speed will be fast enough to handle the large number of grid points.

7.3 Static Grid Adaption

The second part of this study concentrates on finite-difference methods in which the grid points adapt to the solution dynamically to obtain an accurate solution for hypersonic flow. A computer program has been written to utilize Eq. (6.25) for grid adaption. Presently, this code is being run on the Network Operating System (NOS) and the Vector

Processing System (VPS-32) at NASA Langley Research Center. Hypersonic flow over a blunt nose is a typical test case in computational fluid dynamics. This problem has a detached shock which should be resolved accurately, and the location and the magnitude of the shock are not known a priori. The grid should adapt as the solution progresses. This problem is used to analyze and verify the adaptive method. The equations of motion are solved by the MacCormack method [12-13] for hypersonic flow over a small-radius blunt-body with the inclined-plate afterbody (Fig. 2.3). The blunt leading edge is a part of the panel holder which has been tested at the Langley Research Center [50]. The results are obtained at the following conditions: free stream Mach number of 6.8, a pressure of 9.26 lb/ft^2, velocity of 6510 ft/sec, temperature of 375°R (static temperature), Reynolds number of 220,000, specific heat ratio of 1.38, universal gas constant of 1771 $ft^2/sec^2/R$ and a wall temperature of 540° Rankine.

Two tests have been performed: static adaption and dynamic adaption. For static adaption, the solution has been obtained with the fixed grid points shown in Fig. (7.6). Then, the grid points are adapted to two variables, the first and the second derivatives of pressure. Results are shown in Fig. (7.7). In this case, twenty percent of the grid points are allocated to first and second derivatives of pressure ($R_1=R_2=20\%$). For the same case, Fig. (7.8) shows adaption with $R_1=R_2=50\%$. In this case, all the grid points are allocated to the first and the second derivatives of the pressure. This explains the large voids in the constant pressure regions. Figures 7.7-7.8 lack grid resolution in the vicinity of the solid boundaries. This is due to the constant pressure near the solid boundaries. But Eq. (6.25) can adapt

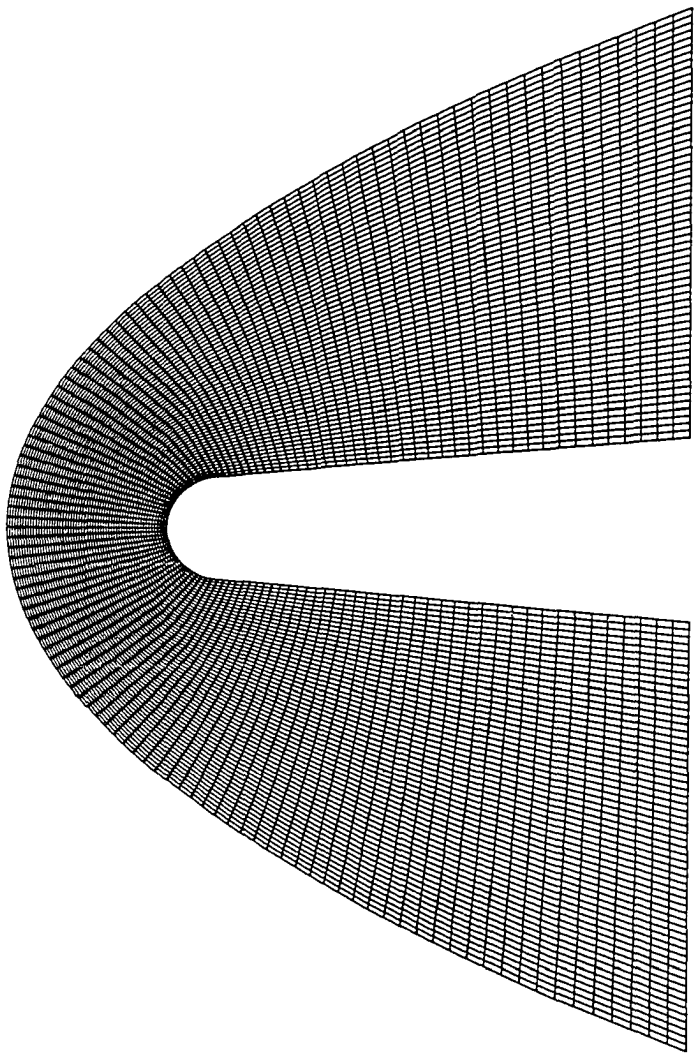


Fig. 7.6 Initial grid distribution

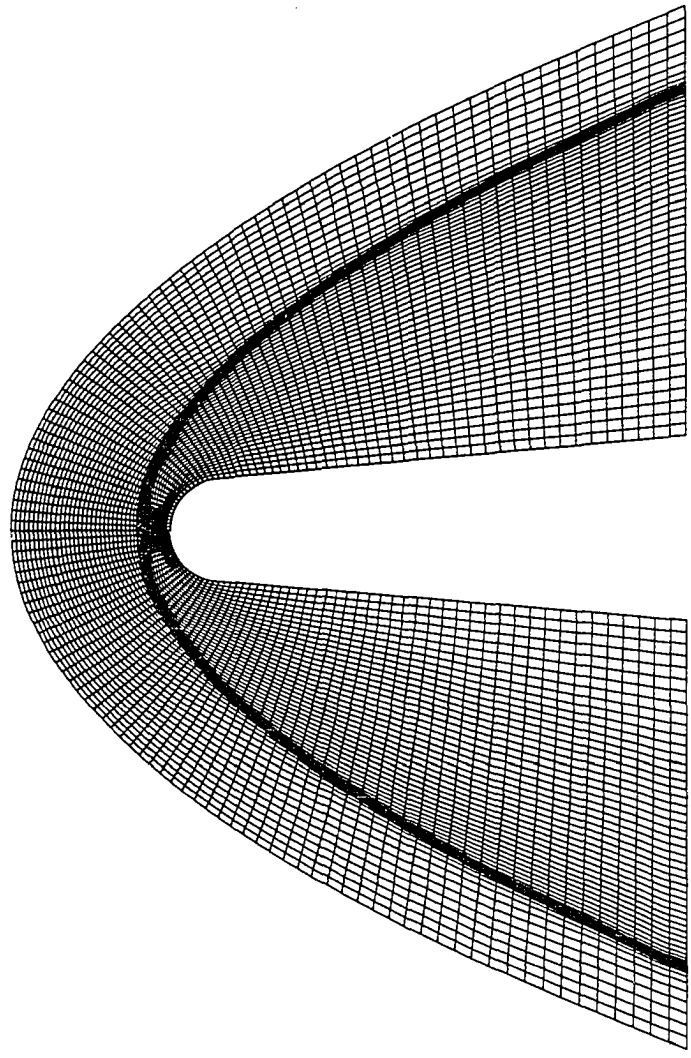


Fig. 7.7 Adapted grid (R_1=R_2=20%)

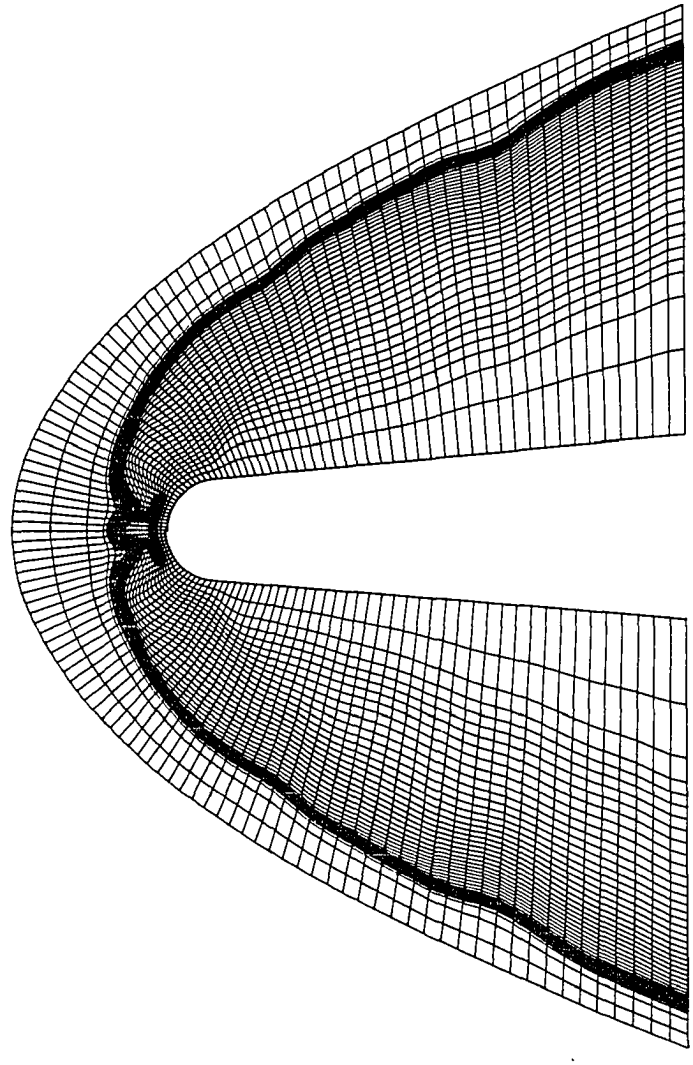


Fig. 7.8 Adapted grid (R_1=R_2=50%)

to several variables. Figure 7.9 shows the grid points which are adapted to two variables --- pressure and velocity. The weight function consists of the first derivatives of pressure, the velocity and the second derivative of pressure. Twenty percent of the grid points are allocated to each function, and forty percent of the grid points are allocated for uniformity. This avoids the creation of any void. It is noted that grid points are clustered near the shock and the solid body. Therefore, it is possible to resolve the pressure as well as the velocity gradients in the boundary layer region. In Fig. (7.10), thirty percent of grid points are allocated to the first derivative of Mach number. If there was a chemical reaction in process, some of the grid points could have been allocated for resolving the gradients of the chemical species.

7.4 Dynamic Grid Adaption

The above procedure has been applied dynamically to the same problem. Figure 15a shows the initial grid distribution for this problem. Figure 7.11 shows sequences of grid distributions at a different time. In this case, grid points are adapted to six variables: the first and second derivatives of pressure, Mach number and velocity. Ten percent of the grid points are allocated equally to first derivatives, and five percent of the grid points are allocated to their second derivatives. Fifty-five percent of the grid points are also allocated to the uniformity. A movie has been produced of this work which shows the dynamic adaption. A few frames are shown in Fig. (7.12). They demonstrate how grid points are attracted toward high gradient regions and repelled from low gradient regions.

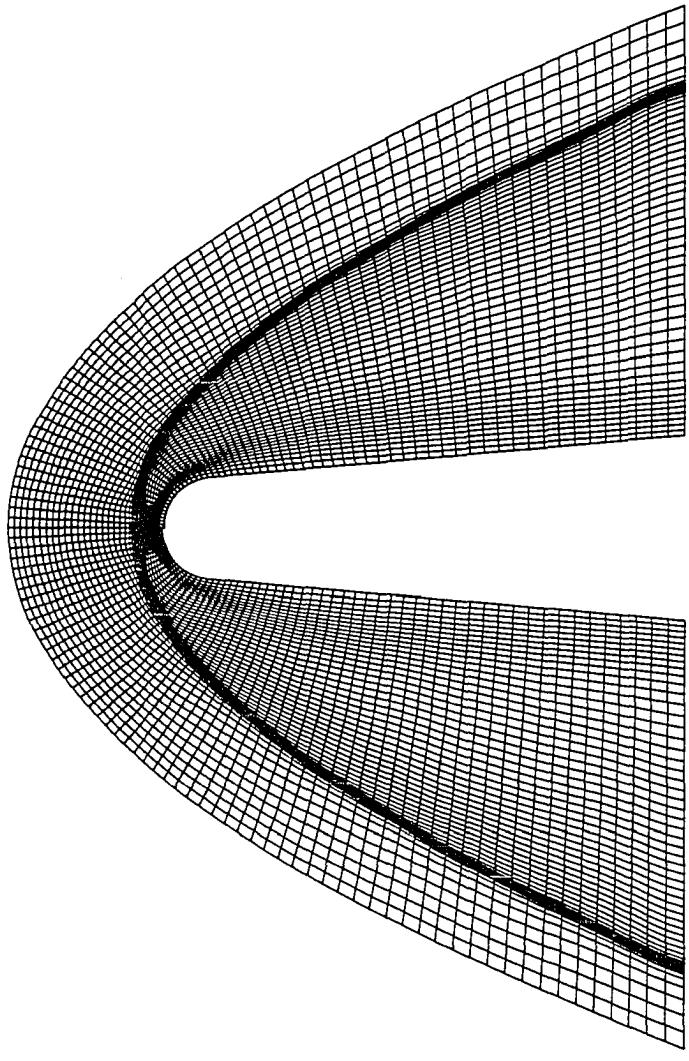


Fig. 7.9 Adapted grid (R_1=R_2=R_3=20%)

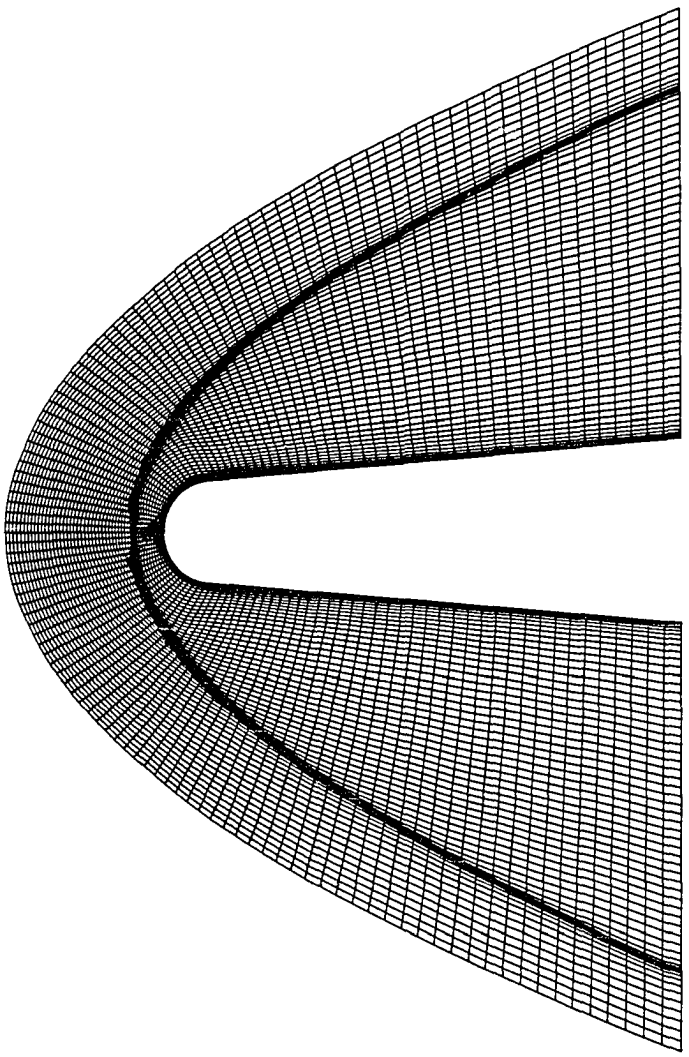


Fig. 7.10 Adapted grid (R_5=30%)

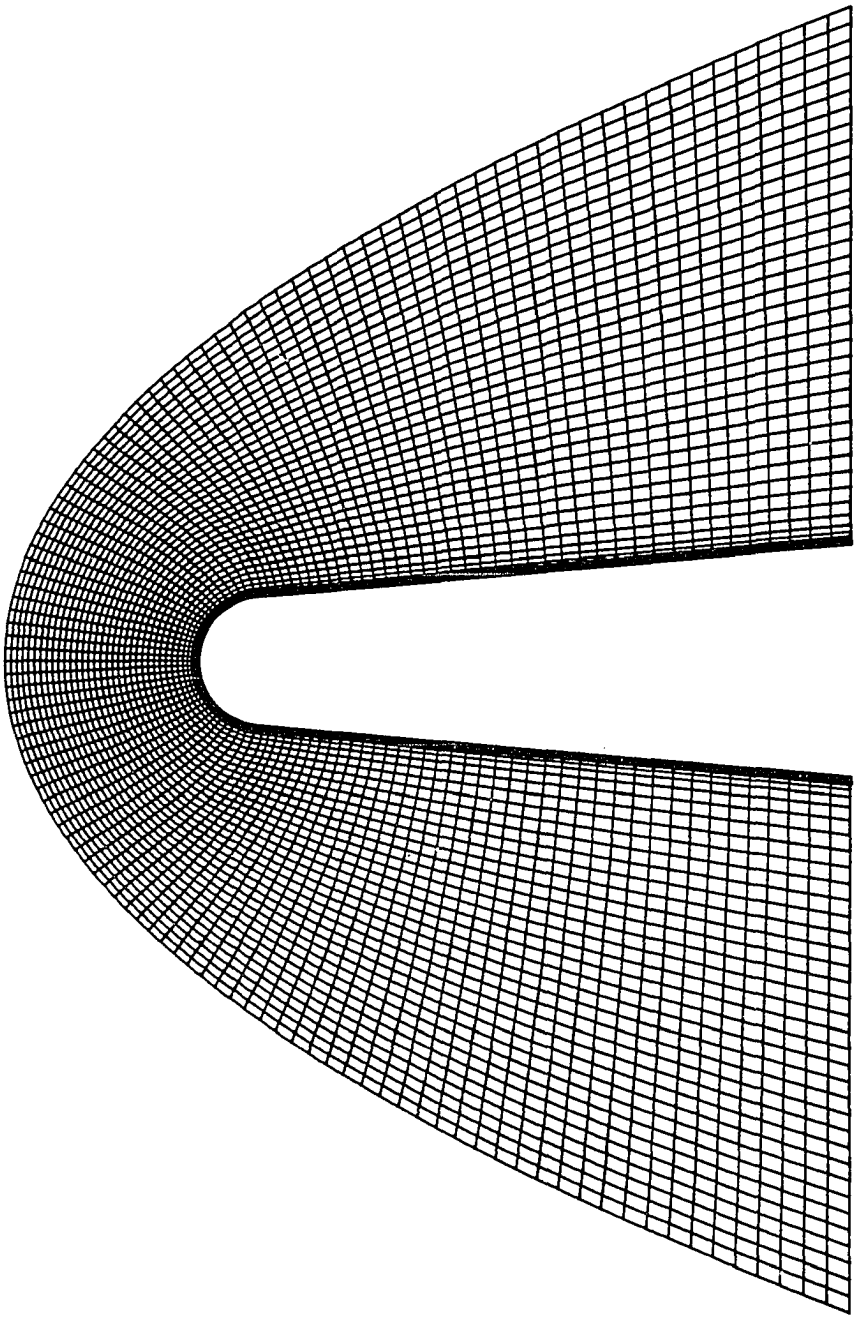


Fig. 7.11 Adapted grid (Dynamic)

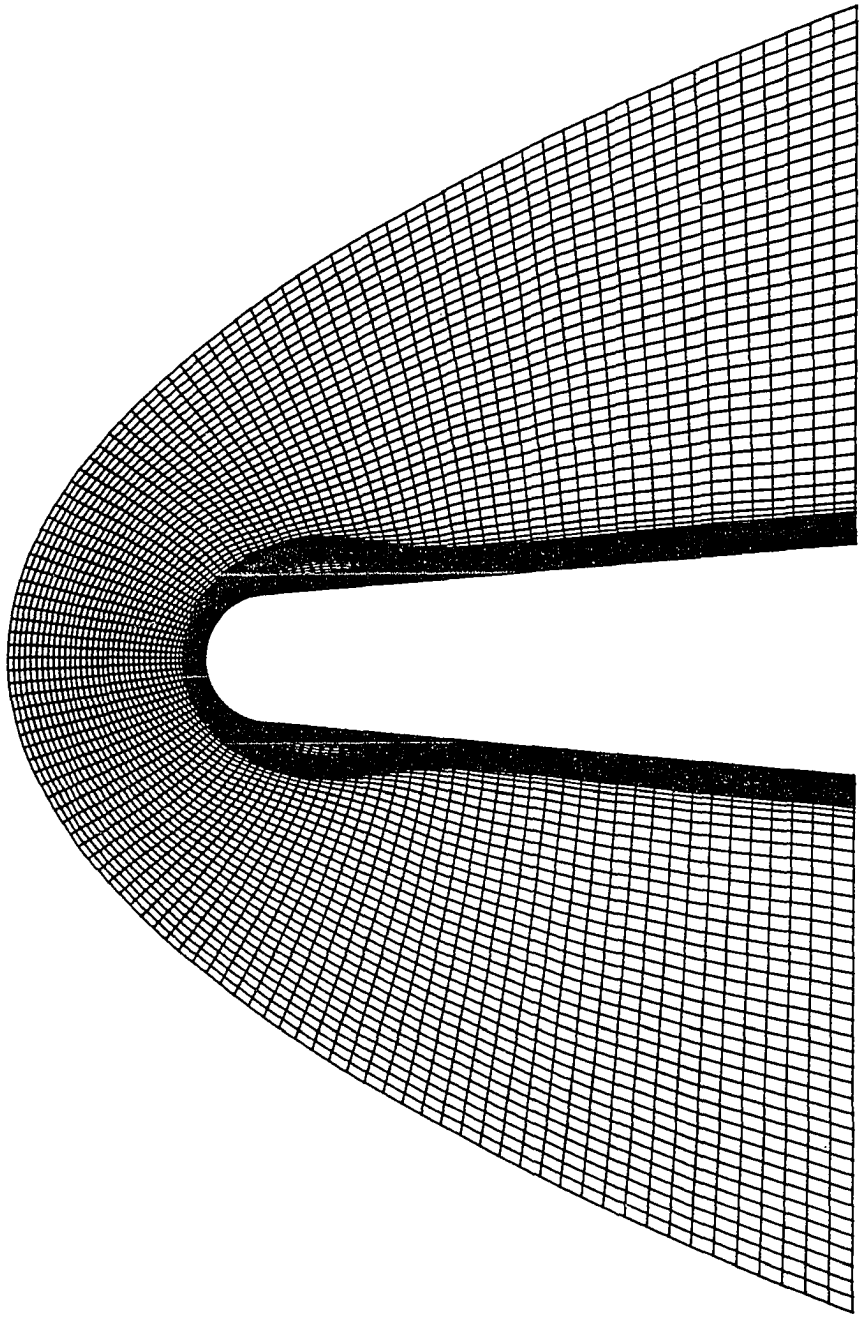


Fig. 7.11 Continued

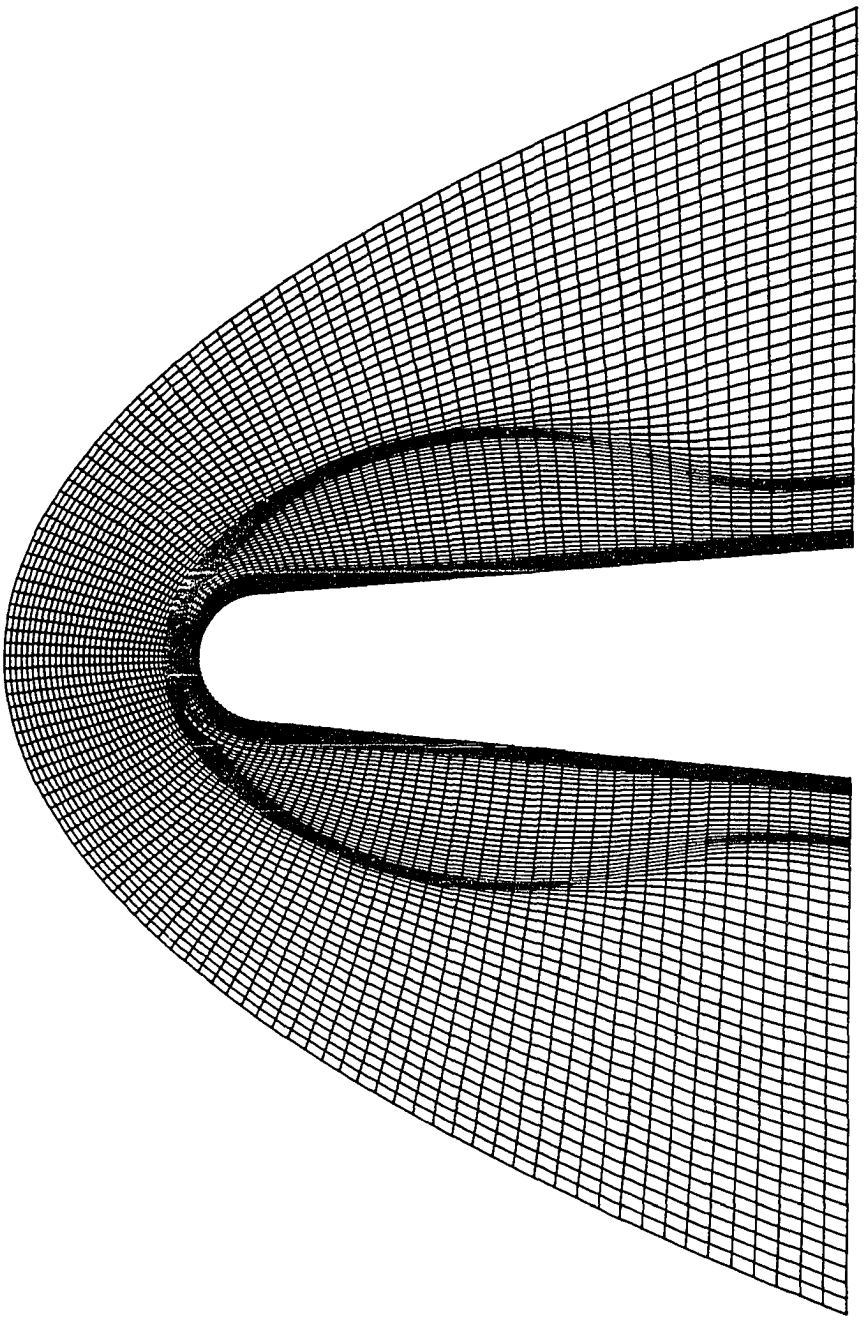


Fig. 7.11 Continued

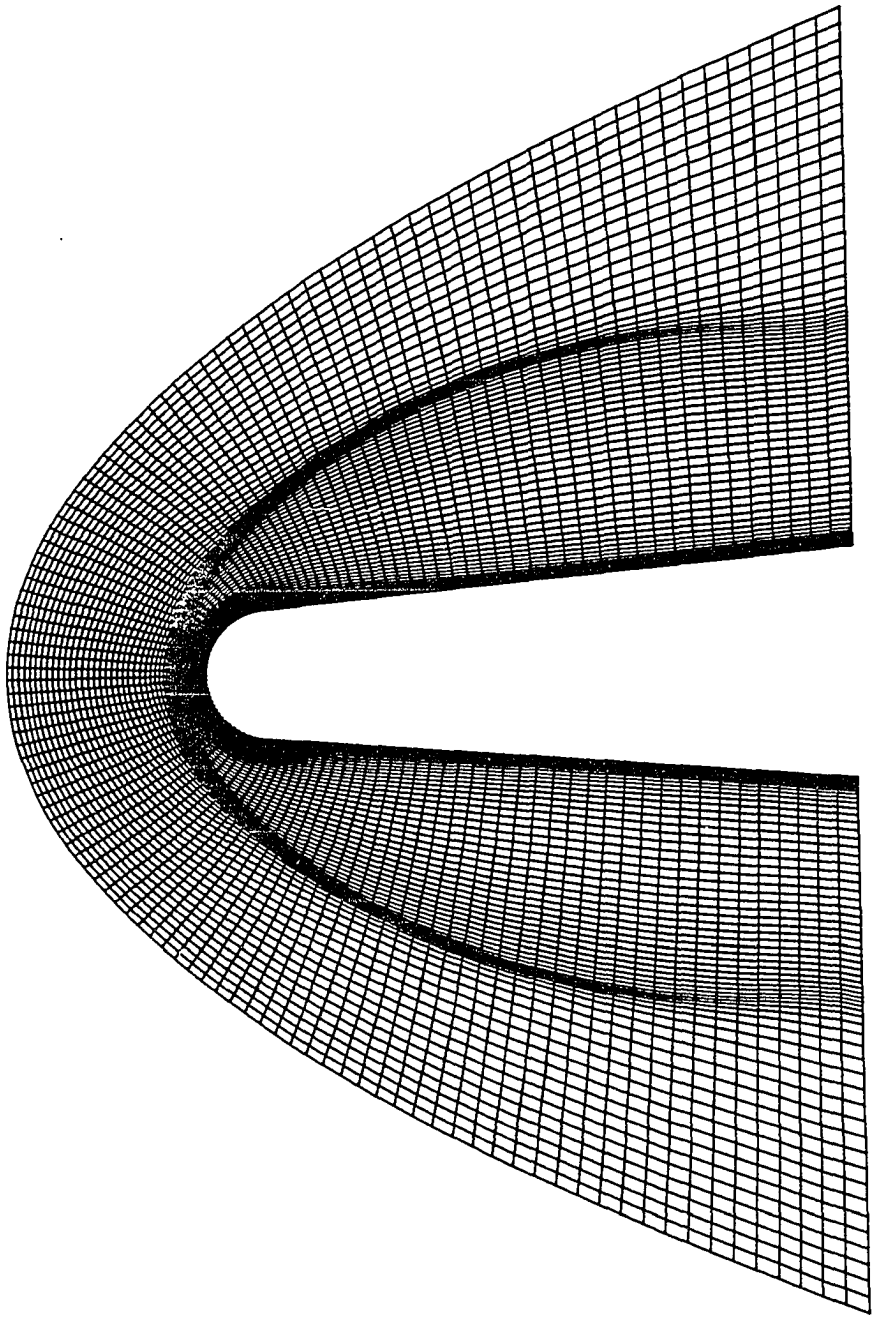


Fig. 7.11 Continued

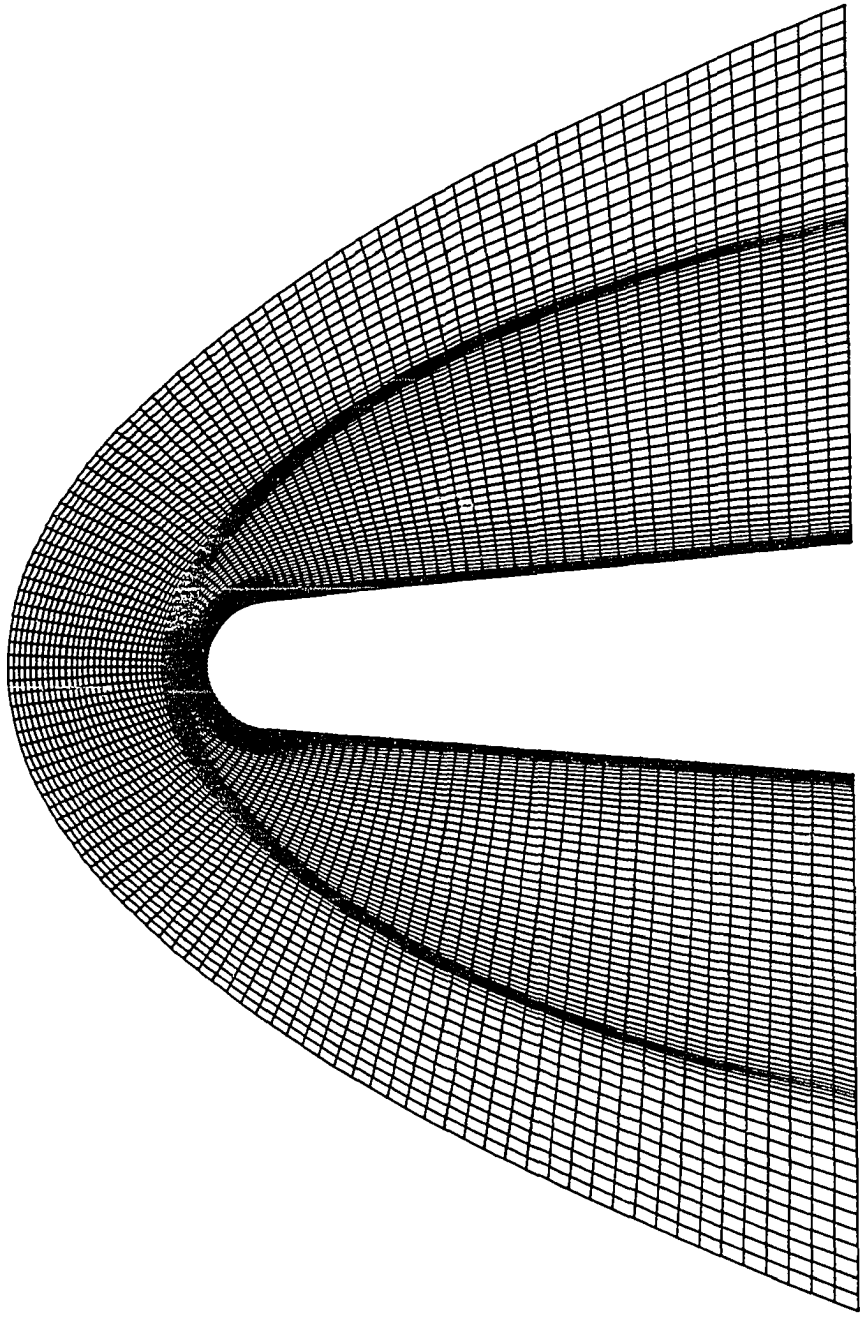


Fig. 7.11 Continued

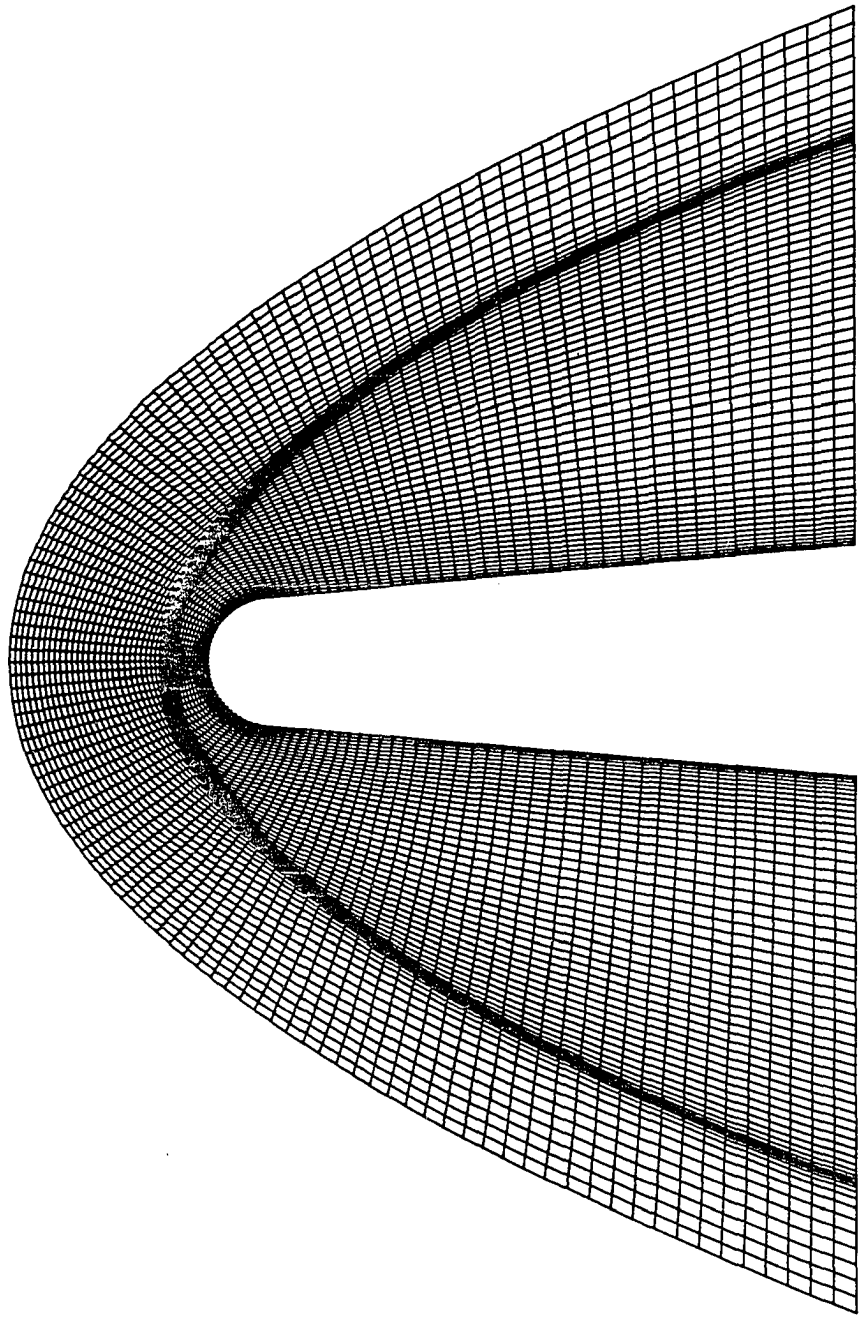


Fig. 7.11 Continued

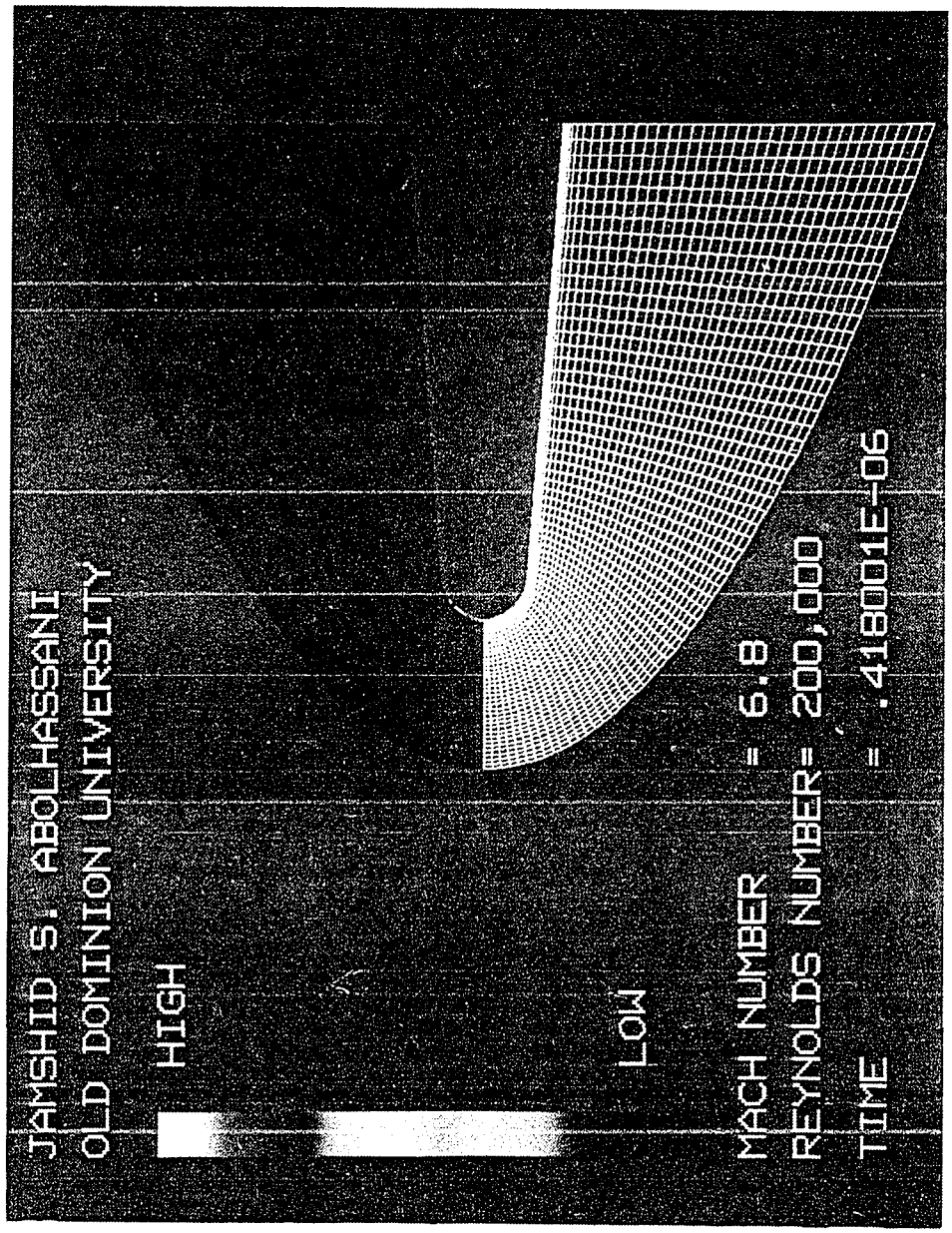


Fig. 7.12 Adapted grid (dynamic)

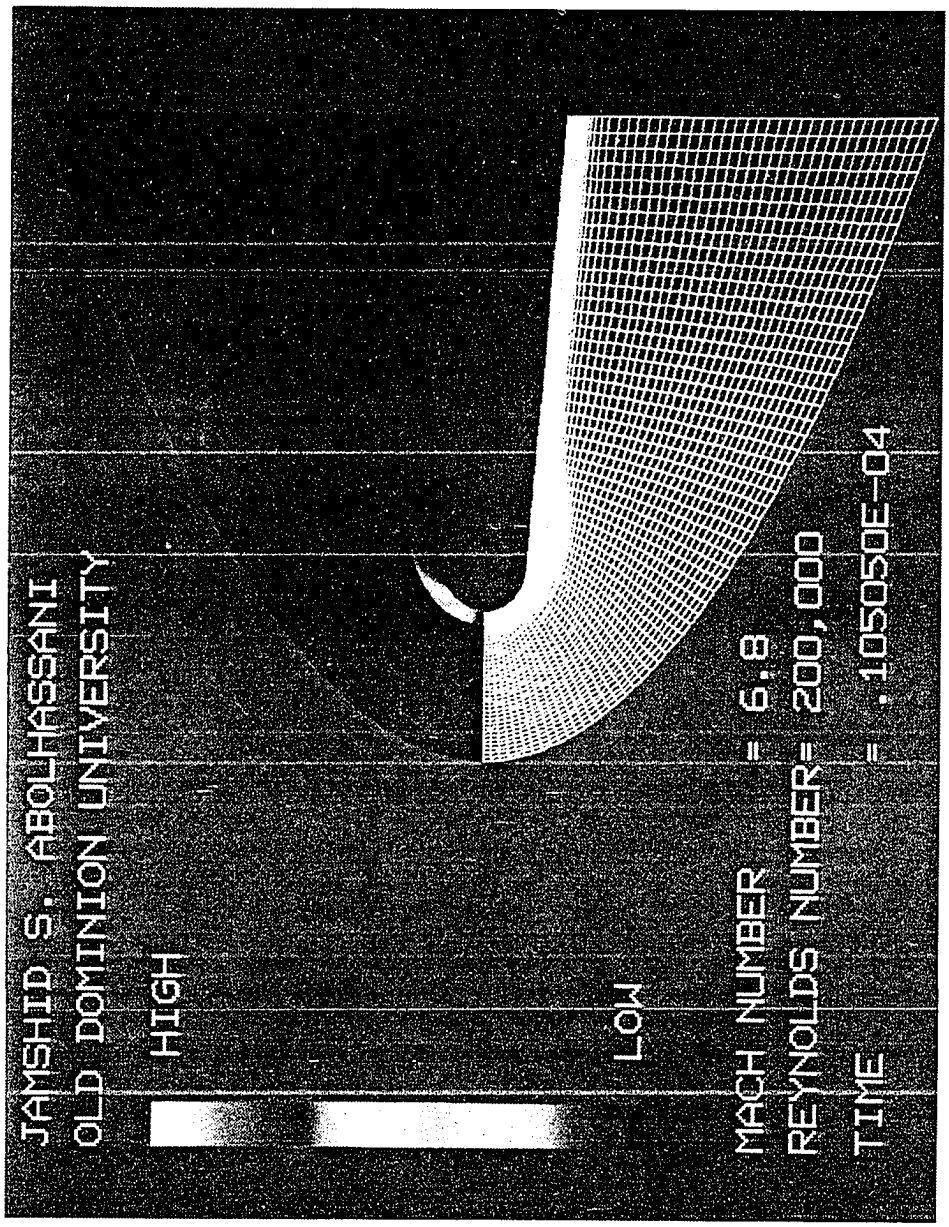


Fig. 7.12 Continued

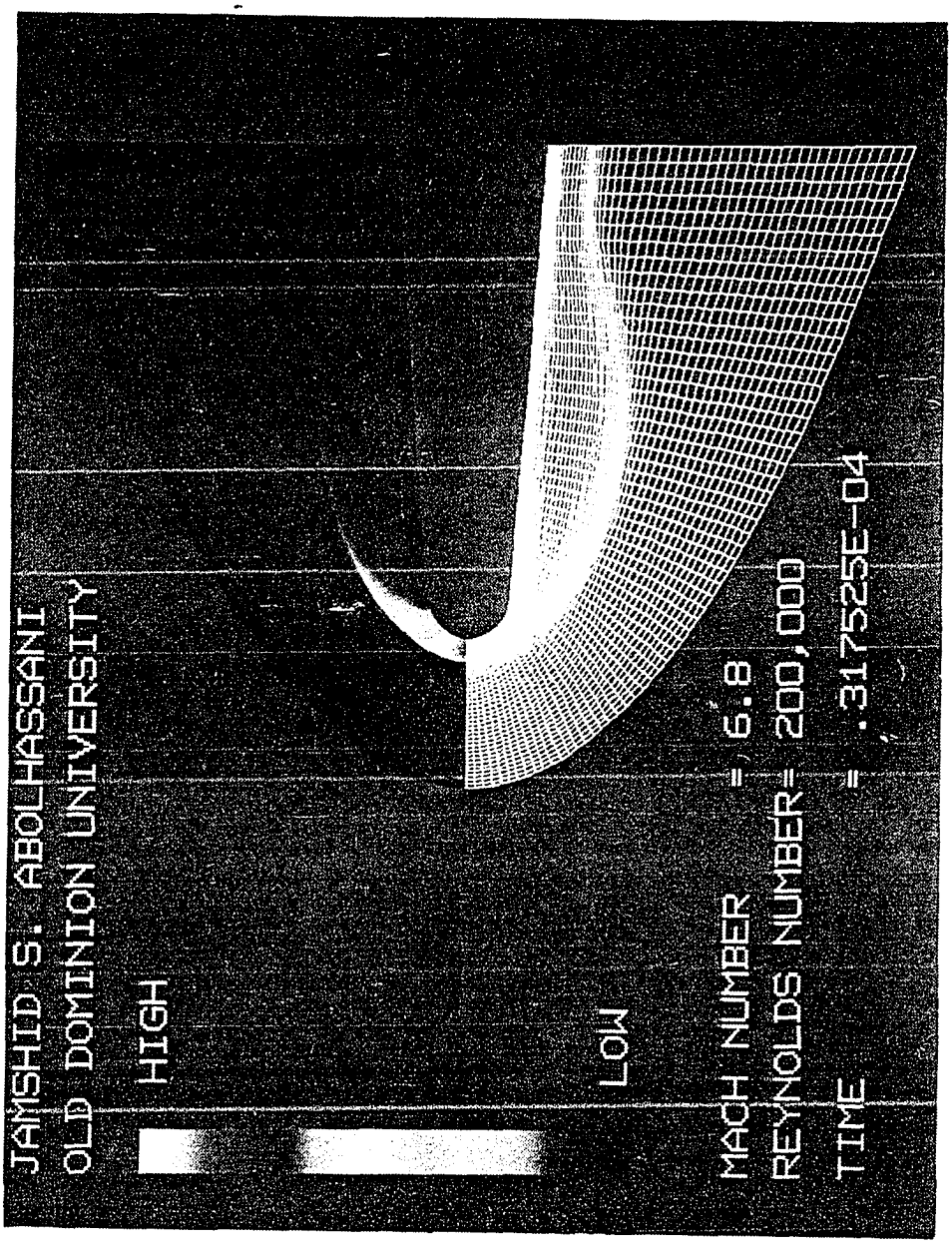


Fig. 7.12 Continued

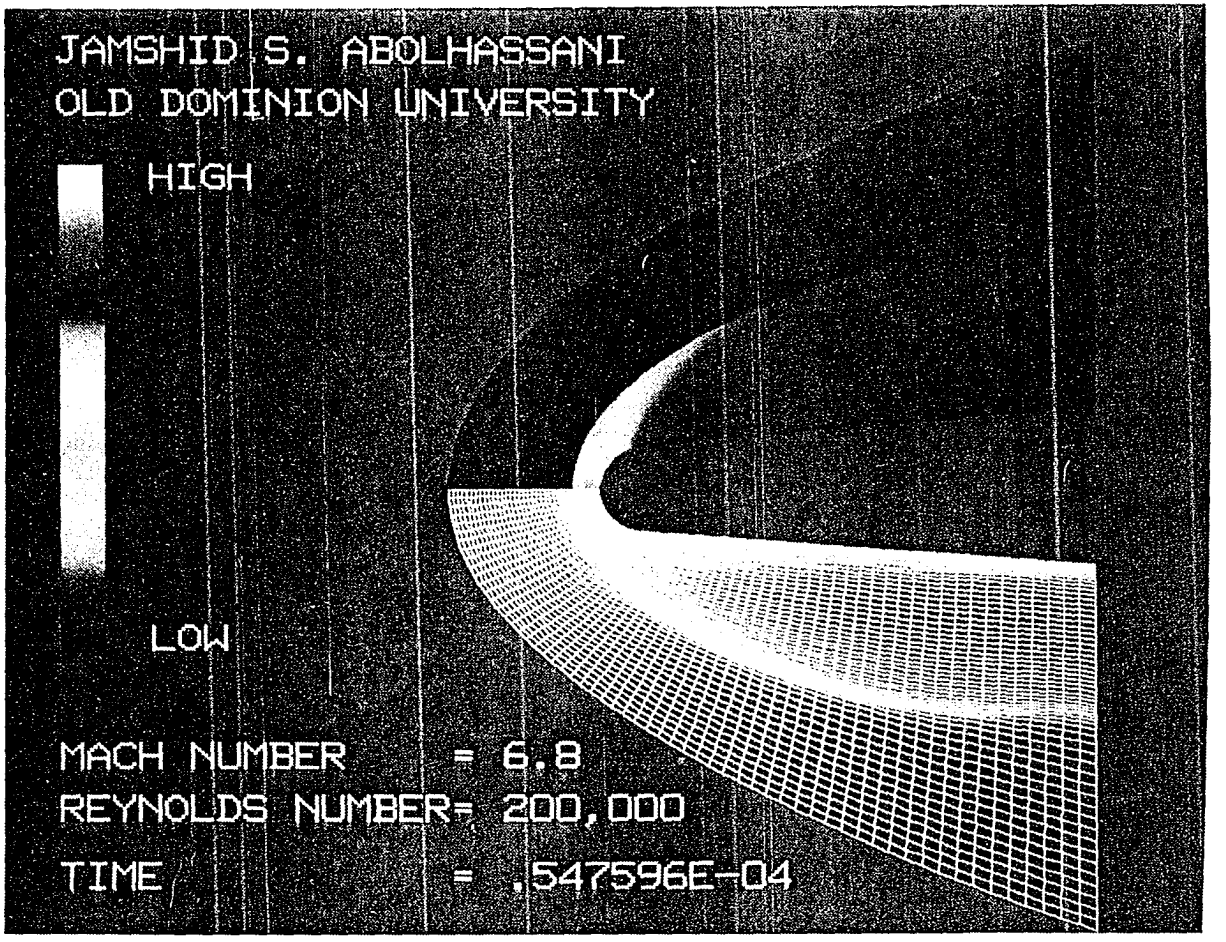


Fig. 7.12 Continued

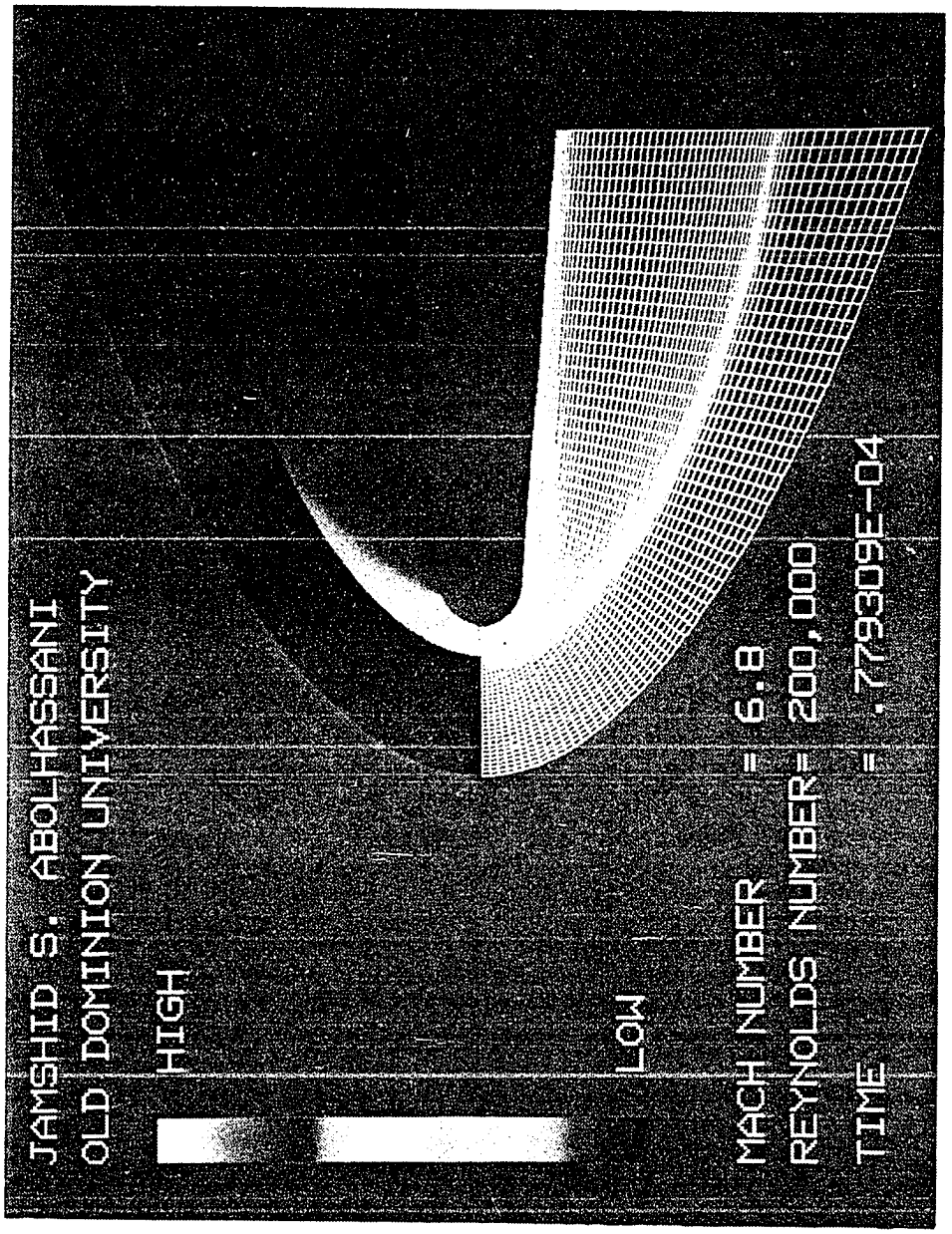


Fig. 7.12 Continued

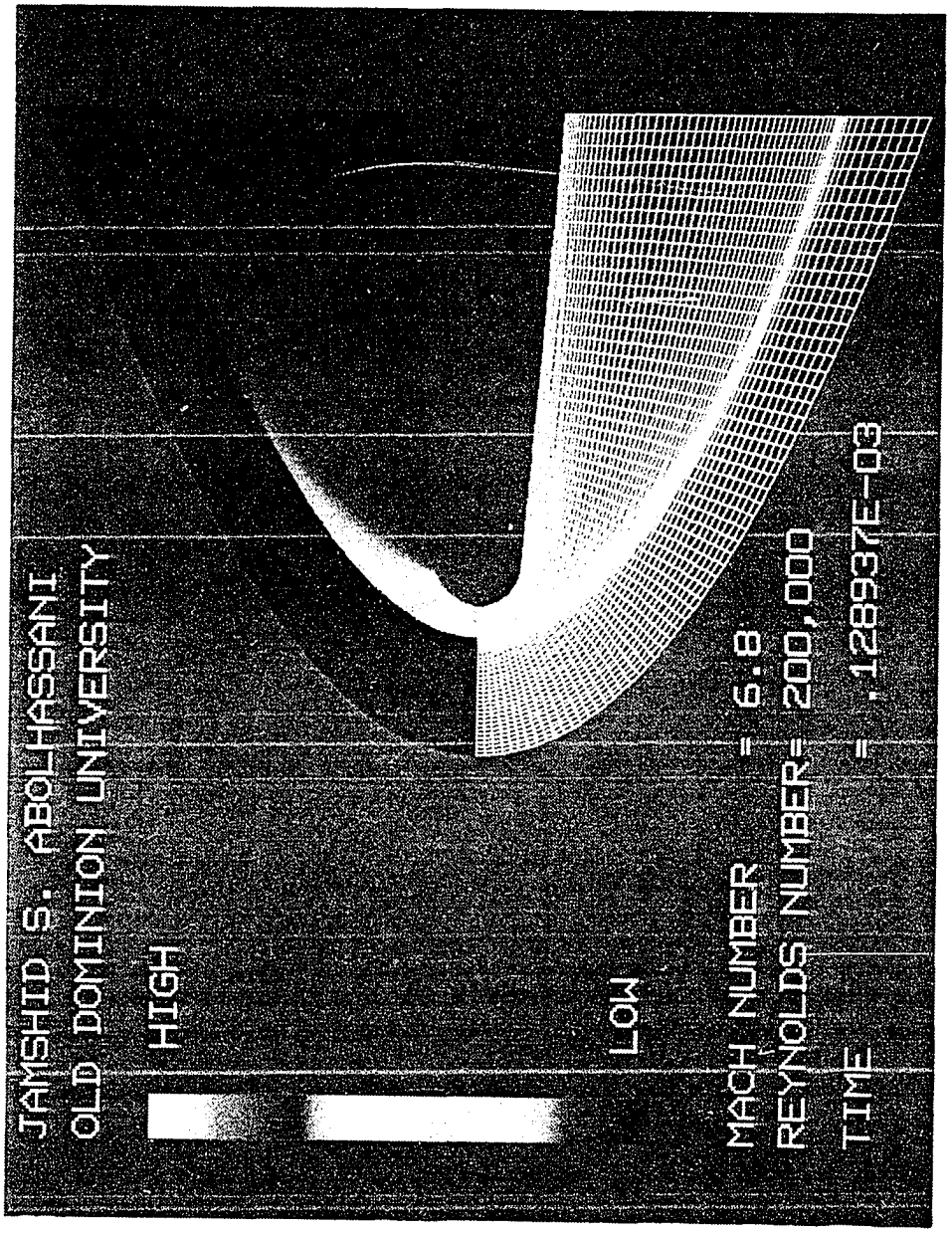


Fig. 7.12 Continued

Chapter 8

CONCLUDING REMARKS

General formulations are presented to investigate the flow field over complex configurations for high-speed free stream conditions. An advanced algebraic method is used to generate grids around these configurations. The computational procedure developed is applied to investigate the flow field over a Butler wing. Illustrative results obtained for specified free stream conditions compare very well with available experimental and numerical results. Results are obtained for laminar flow over a Butler wing at a Mach number of 3.5, Reynolds number of 2×10^6/ft (6.56×10^6/m), free static temperature of 390°R (216.67°K), wall temperature of 1092°R (606.67°K), length of 0.80 ft (0.2438 m) when the wing is at zero and ten degrees angle of attack. Two types of grids have been generated for this wing; H-type and O-type. Results of both cases are compared and discussed. The future plans to extend this study are to include the sting, use multiple grid, different Mach numbers, and to include turbulence.

A grid adaption method has been developed with the capability of adapting grid points to several variables. This method is an algebraic method, and has been formulated in such a way that there is no need for any matrix inversion. The method is used in conjunction with the calculation of hypersonic flow over a blunt-nose body. A movie has been produced which shows simultaneously the transient behavior of the

solution and the grid adaption. The results indicate the viability and validity of the proposed method. The future plans for this technique are to use this problem with a true unsteady problem, to study the effect of interpolation, to consider more complex geometries and finally to adapt the problem in three-dimensions.

REFERENCES

1. Thompson, J. F., Warsi, Z. U. A. and Mastin, C. W., Numerical Grid Generation: Foundations and Applications, North-Holland Publishing, New York, 1985.

2. Thompson, J. F., "General Curvilinear Coordinate Systems," Numerical Grid Generation, Joe F. Thompson, Editor, Elsevier Science Publishing Co., Inc., New York, 1982, pp. 1-28.

3. Eriksson, L.E., "Transfinite Mesh Generation and Computer-Aided Analysis of Mesh Effects," Ph.D. Dissertation, Department of Computer Sciences, Uppsala University, Uppsala, Sweden, 1984.

4. Mastin, C. W., "Error Induced by Coordinate System," Numerical Grid Generation, Joe F. Thompson, Editor, Elsevier Science Publishing Co., Inc., New York, 1982, pp. 31-37.

5. Raithby, G. D., "Skew Upstream Differencing Schemes for Problems Involving Fluid Flow," Mechanical Engineering, Vol. 1976, pp. 153-164.

6. Butler, D. S., "The Numerical Solution of Hyperbolic System of Partial Differential Equations in Three Independent Variables," Proceedings of the Royal Society, Series A, Mathematical and Physical Sciences, No. 1281, Vol. 255, April 1960, pp. 232-252.

7. Squire, L. C., "Some Effects of Thickness on the Longitudinal Characteristic of the Sharp Edges of Delta Wings," The Aeronautical Journal of the Royal Aeronautical Society, Vol. 72, February 1968, pp. 151-155.

8. Squire, L. C., "Measured Pressure Distribution and Shock Shapes on a Simple Delta Wing," Cambridge University, Department of Engineering, CUED/Aero/Tr9, 1979.

9. Squire, L. C., "Measured Pressure Distributions and Shock Shapes on a Simple Delta Wing," The Aeronautical Quarterly, Vol. 32, Part 3, August 1981, pp. 188-198.

10. Walkden, F. and Caine, P., "Surface Pressure on a Wing Moving with Supersonic Speed," Proceedings of Royal Society of London, Series A, Vol. 341, 1974, pp. 177-193.

11. Thompson, J. F., "A Survey of Dynamically-Adaptive Grids in the Numerical Solutions of Partial Differential Equations," AIAA Paper 84-1606, AIAA 17th Fluid and Plasma Dynamics, and Laser Conference, Snowmass, Colorado, June 25-27, 1984.

12. MacCormack, R. W. and Baldwin, B. S., "A Numerical Method for Solving the Navier-Stokes Equations with Application to Shock-Boundary Layer Intersections," AIAA Paper 75-1, AIAA 13th Aerospace Sciences Meeting, Pasadena, CA, January 20-22, 1975.

13. Smith, R. E., "Two-Boundary Grid Generation for the Solution of the Three-Dimensional Compressible Navier-Stokes Equations," NASA Technical Memorandum 83123, May 1981.

14. Anderson, D. A., Tannehill, J. C. and Pletcher, R. H., Computational Fluid Mechanics and Heat Transfer, McGraw-Hill Book Company, New York, 1984.

15. Hindman, R., "Geometrically Induced Errors and their Relationship to the Form of Governing Equations and the Treatment of Generalized Mapping," AIAA Paper 81-1008, June 1986.

16. Newsome, R. W. and Adams, M. S., "Numerical Simulation of Vortical-Flow Over an Elliptical-Body Missile at High Angle of Attack," AIAA Paper 86-0559, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, January 6-9, 1986.

17. Roache, P. J., Computational Fluid Dynamics, Hermosa Publisher, 1972.

18. Eiseman, P. R., "Adaptive Grid Generation," The First Congress on Computational Mechanics, Austin, Texas, September, 1986.

19. Gear, C. W., Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

20. Abolhassani, J. S., "Applications of the Method of Lines for Solutions of Navier-Stokes Equations Using a Nonuniform Grid Distribution," Master Thesis, Mechanical Engineering and Mechanics Department, College of Engineering and Technology, Old Dominion University, Norfolk, Virginia 23508, May 1983.

21. Denny, V. E. and Landis, R. B., "A New Method for Solving Two-Point Boundary Problems Using Optimal Node Distribution," Journal of Computational Physics, Vol. 9, February 1972, pp. 120-137.

22. De Rivas, E. K., "On the Use of Nonuniform Grids in Finite Difference Equations," Journal Computational Physics, Vol. 10, October, 1972, pp. 202-210.

23. Pereyra, V. and Sewell, E. G., "Mesh Selection for Discrete Solution of Boundary-Value Problems in Ordinary Differential Equations," Numerische Mathematik, Vol. 23, 1975, pp. 261-268.

24. Russell, R. D. and Christiansen, J., "An Adaptive Mesh Selection Strategies for Solving Boundary Layer Problems," SIAM Journal of Numerical Analysis, Vol. 15, No. 1, February 1978, pp. 59-80.

25. White, A. B., Jr., "On Selection of Equidistribution Meshes for Two-Point Boundary-Value Problems," SIAM Journal of Numerical Analysis, Vol. 16, No. 3, June 1979, pp. 472-502.

26. White, A. B., Jr., "On the Numerical Solution of Initial/Boundary Value Problems in One Space Dimension," SIAM Journal of Numerical Analysis, Vol. 19, 1982, pp. 683-697.

27. Brackbill, J. U. and Saltzman, J., "An Adaptive Computation Mesh for the Solution of Singular Perturbation Problems," Numerical Grid Generation Techniques, Robert E. Smith, Editor, NASA Conference Publication 2166, 1980, pp. 193-196.

28. Brackbill, J. U., "Coordinate System Control: Adaptive Mesh," Numerical Grid Generation, Joe F. Thompson, Editor, Elsevier Science Publishing Co., Inc., 1982, pp. 277-294.

29. Brackbill, J. U. and Saltzman, J. S., "Adaptive Zoning for Singular Problems in Two Dimensions," Journal of Computational Physics, Vol. 46, No. 3, June 1982, pp. 342-368.

30. Saltzman, J. S. and Brackbill, J. U., "Application and Generalization of Variational Methods for Generating Adaptive Meshes," Numerical Grid Generation, Edited By J. F. Thompson, Elsevier Science Publishing Company, 1985.

31. Winslow, A. M., "Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh," Journal of Computational Physics, Vol. 1, No. 2, November 1966, pp. 149-172.

32. Thompson, J. F., Thames, F. C. and Mastin, C. W., "Automatic Numerical Generating of Body-Fitted Curvilinear System for Field-Containing Any Number of Arbitrary Two-Dimensional Bodies," Journal of Computational Physics, Vol. 15, July 1974, pp. 299-319.

33. Nakamura, S., "Marching Grid Generation Using Parabolic Partial Differential Equations," Numerical Grid Generation, J. F. Thompson, Editor, North-Holland, 1982, pp. 775-786.

34. Steger, J. L. and Sorenson, R. L., "Use of Hyperbolic Partial Differential Equations to Generate Body Fitted Coordinates," Numerical Grid Generations Techniques, Robert E. Smith, Editor, NASA Conference Publishing 2166, 1980, pp. 463-478.

35. Rai, M. M. and Anderson, D. A., "Grid Evolution in Time Asymptotic Problems," Numerical Grid Generation Techniques, Robert E. Smith, Editor, NASA Conference Publishing 2166, 1980, pp. 409-430.

36. Rai, M. M. and Anderson, D. A., "The Use of Adaptive Grid in Conjunction with Shock Capturing Methods," AIAA paper 81-1012, St. Louis, Missouri, June 1981.

37. Rai, M. M. and Anderson D. A., "Application of Adaptive Grids to Fluid Flow Problems with Asymptotic Solutions," AIAA Journal, Vol. 20, 1982, pp 496-502.

38. Anderson, D. A. and Rai, M. M., "The Use of Solution Adaptive Grids in Solving Partial Differential Equations," Numerical Grid Generation, By J. F. Thompson, Editor, Elsevier Science Publishing Company, 1982, pp. 317-338.

39. Anderson, D. A., "Adaptive Grid Method for Partial Differential Equations," ASME Applied Mechanics, Bioengineering and Fluids Engineering Conference, Houston, Texas, June 20-22, 1983.

40. Anderson, D. A. and Steinbrenner, J., "Generating Adaptive Grids with a Conventional Grid Scheme," AIAA Paper 86-0427, Presented at the AIAA 24th Aerospace Science meeting, Reno, Nevada, January 6-9, 1986.

41. Nakahashi, Kazuhiro and Deiwert, G. S., "A Self Adaptive-Grid Method with Application to Airfoil Flow," AIAA Seventh Computational Fluid Dynamic Conference, Cincinnati, Ohio, July 15-17, 1985 (Also AIAA paper 85-1525).

42. Dwyer, H. A., Kee, R. J. and Sanders, B. R. , "An Adaptive Grid for Problem in Fluid Mechanics and Heat Transfer," AIAA Paper 79-1464, Williamsburg, Virginia, July 1979.

43. Dwyer, H. A., Smooke, M. D. and Kee, R. J., "Adaptive Gridding for Finite Difference Solutions to Heat and Mass Transfer Problems," Numerical Grid Generation, J. F. Thompson, Editor, Elsevier Science Publishing Company, 1982, pp. 339-356.

44. Dwyer, H. A., "A Discussion of Some Criteria for the Use of Adaptive Gridding," Adaptive Computational Method for Partial Differential Equations, Edited by Ivo Babuska, Jagdish Chandra, Joseph E. Flaherty, SIAM Publishing Company, 1983, Also AIAA paper 83-1706, AIAA 16th Fluid and Plasma Dynamics Conference, Danvers, Mass., February 1983.

45. Dwyer, H. A., "Grid Adaption for Problems in Fluid Dynamics," AIAA Journal, Vol. 22, No. 12, December 1984, pp. 1705-1712.

46. Hindman, R. G. and Spencer, J., "A New Approach to Truly Adaptive Grid Generation," AIAA Paper 83-0450, Reno, Nevada, January 1983.

47. Weinstock, R., Calculus of Variations, McGraw-Hill Book Company, New York, 1952.

48. Eiseman, P. R., "Grid Generation for Fluid Mechanics Computations," Annual Review of Fluid Mechanics, Vol. 17, 1985, pp. 487-522.

49. Kandil, O. A. and Chuang, A., "Influence of Numerical Dissipation in Computing Supersonic Vortex-Dominated Flows," AIAA Paper 86-1073, AIAA/ASME 4th Fluid Mechanics, Plasma Dynamics and Laser Conference, Atlanta, GA, May 12-14, 1986.

50. Deveikis, W. D. and Hunt, L. R., "Loading and Heating of a Large Flat Plate at Mach 7 in the Langley 8-Foot High Temperature Structure Tunnel," NASA TM-D-7275, 1975.

APPENDICES

APPENDIX A

MATHEMATICAL DETAILS FOR THE TRANSFORMED EQUATIONS

A.1 Curvilinear Coordinates

In covariant coordinates system (x_i), the position vector of a point from the origin is expressed as

$$\bar{r} = e_i\, x_i = e_1\, x_1 + e_2\, x_2 + e_3\, x_3 \qquad (A.1)$$

where e_i is the covariant base vector.

In the present study, covariant coordinates are labeled as x, y, and z ($\bar{i}$, $\bar{J}$, $\bar{k}$) and contravariant coordinates are labeled as ξ, η and ζ (i, j, k). The covariant base vectors are defined as

$$\underline{e}_i = \frac{\partial r}{\partial x^{\bar{i}}},$$

or

$$\begin{Bmatrix} e_1 \\ e_2 \\ e_3 \end{Bmatrix} = \begin{pmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & z_\zeta \end{pmatrix} \begin{Bmatrix} i \\ j \\ k \end{Bmatrix} = [J^{-1}]^T \begin{Bmatrix} i \\ j \\ k \end{Bmatrix} \qquad (A.2)$$

where J is the Jacobian of transformation. Magnitude of Jacobian ($|J|$) is the local value of the ratio of an elemental volume in the mapped (usually cube) cell to the corresponding elemental volume in the physical (usually distorted) cell.

The contravariant base vectors are defined as

$$\underline{e}^i = \frac{\partial x^i}{\partial x_i}$$

or

$$\begin{Bmatrix} e^1 \\ e^2 \\ e^3 \end{Bmatrix} = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} \begin{Bmatrix} \bar{i} \\ \bar{j} \\ \bar{k} \end{Bmatrix} = [J] \begin{Bmatrix} \bar{i} \\ \bar{j} \\ \bar{k} \end{Bmatrix} . \qquad (A.3)$$

Position vector can be expressed explicitly in terms of contravariant vector (x^i); however, the infinitesimal vector $\underline{dr}$ can be expressed as

$$dr = \frac{\partial r}{\partial x^i} d x^i = e_i \, dx^i . \qquad (A.4)$$

Also the magnitude of arclength (ds) can be expressed as

$$(ds)^2 = dr \cdot dr \qquad (A.5)$$

Substitution of Eq. A.4 into Eq. A.5 will result in

$$ds^2 = (e_i \cdot e_j) \, dx_i \, dx_j = g_{ij} \, dx_i \, dx_j \qquad (A.6)$$

where g_{ij} is called covariant fundamental metric coefficients.

These coefficients can be defined as

$$g_{ij} = [J^{-1}]^T [J^{-1}] = \frac{\partial x_k}{\partial x^i} \cdot \frac{\partial x_k}{\partial x^j} . \qquad (A.7)$$

They are defined as

$$g_{11} = x_\xi^2 + y_\xi^2 + z_\xi^2 , \qquad (A.8a)$$

$$g_{12} = g_{21} = x_\xi \, x_\eta + y_\xi \, y_\eta + z_\xi \, z_\eta , \qquad (A.8b)$$

$$g_{13} = g_{31} = x_\xi\, x_\zeta + y_\xi\, y_\zeta + z_\xi\, z_\zeta \,, \tag{A.8c}$$

$$g_{22} = x_\eta^2 + y_\eta^2 + z_\eta^2 \,, \tag{A.8d}$$

$$g_{23} = g_{32} = x_\eta\, x_\zeta + y_\eta\, y_\zeta + z_\eta\, z_\zeta \,, \tag{A.8e}$$

$$g_{33} = x_\zeta^2 + y_\zeta^2 + z_\zeta^2 \,. \tag{A.8f}$$

Similarly, contravariant fundamental metric coefficients are defined as

$$g^{ij} = e^i \cdot e^j = [J]\,[J]^T = \frac{\partial x^i}{\partial x_k} \cdot \frac{\partial x^j}{\partial x_k} \,, \tag{A.9}$$

or

$$g^{11} = \xi_x^2 + \xi_y^2 + \xi_z^2 \,, \tag{A.10a}$$

$$g^{21} = g^{12} = \xi_x\, \eta_x + \xi_y\, \eta_y + \xi_z\, \eta_z \,, \tag{A.10b}$$

$$g^{13} = g^{31} = \xi_x\, \zeta_x + \xi_y\, \zeta_y + \xi_z\, \zeta_z \,, \tag{A.10c}$$

$$g^{22} = \eta_x^2 + \eta_y^2 + \eta_z^2 \,, \tag{A.10d}$$

$$g^{23} = g^{32} = \eta_x\, \zeta_x + \eta_y\, \zeta_y + \eta_z\, \zeta_z \,, \tag{A.10e}$$

$$g^{33} = \zeta_x^2 + \zeta_y^2 + \zeta_z^2 \,. \tag{A.10f}$$

Furthermore, there exists a unique relationship between contravariant and covariant fundamental metric coefficients,

$$g^{ij} = \frac{g_{rs}\, g_{\ell s} - g_{rt}\, g_{\ell s}}{|g_{ij}|} = \frac{G_{ij}}{|g_{ij}|} = \frac{\text{Cofactor of } g_{ij}}{|g_{ij}|} \,, \tag{A.11}$$

where

$$G_{11} = g_{22}\, g_{33} - g_{23}^2 \,, \tag{A.12a}$$

$$G_{12} = G_{21} = g_{13}\, g_{23} - g_{12}\, g_{33} \,, \tag{A.12b}$$

$$G_{13} = G_{31} = g_{12}\, g_{23} - g_{13}\, g_{22} \,, \tag{A.12c}$$

$$G_{22} = g_{11}\, g_{33} - g_{13}^2 \,, \tag{A.12d}$$

$$G_{23} = G_{32} = g_{12}\, g_{13} - g_{23}\, g_{11} \,, \tag{A.12e}$$

$$G_{33} = g_{11}\, g_{22} - g_{12}^2 \,, \tag{A.12f}$$

where

$$g_{ij} = \frac{G^{ij}}{|g^{ij}|} = \frac{\text{Cofactor of } g^{ij}}{|g^{ij}|} \,.$$

There is also a relationship between covariant and contravariant base vector

$$\underline{e}^i = \frac{\underline{e}_j \times \underline{e}_k}{|g_{ij}|^{1/2}} \,, \tag{A.13}$$

where

$$|g_{ij}| = |J^{-1}|^2 \,,$$

i.e.

$$e^1 = \frac{(y_\eta z_\zeta - y_\zeta z_\eta)\, i - (x_\eta z_\zeta - x_\zeta z_\eta)\, j + (x_\eta y_\zeta - x_\zeta y_\eta)\, k}{|J^{-1}|} \,,$$

or

$$\begin{Bmatrix} e^1 \\ e^2 \\ e^3 \end{Bmatrix} = \frac{1}{|J^{-1}|} \begin{Bmatrix} (y_\eta z_\zeta - y_\zeta z_\eta) & -(x_\eta z_\zeta - x_\zeta z_\eta) & (x_\eta y_\zeta - x_\zeta y_\eta) \\ -(y_\xi z_\zeta - y_\zeta z_\xi) & (x_\xi z_\zeta - x_\zeta z_\xi) & -(x_\xi y_\zeta - x_\zeta y_\xi) \\ (y_\xi z_\eta - y_\eta z_\xi) & -(x_\xi z_\eta - x_\eta z_\zeta) & (x_\xi y_\eta - x_\eta y_\xi) \end{Bmatrix} , \tag{A.14}$$

where

$$J = \underline{e}^i \,.$$

There is also a relationship between contravariant and covariant base vector

$$\underline{e}_i = \frac{\underline{e}^j \times \underline{e}^k}{|g^{ij}|} ,$$

where

$$|g^{ij}| = |J|^2 , \tag{A.15}$$

i.e.

$$\begin{Bmatrix} \underline{e}_1 \\ \underline{e}_2 \\ \underline{e}_3 \end{Bmatrix} = \frac{1}{|J|^{-1}} \begin{Bmatrix} (\eta_y\zeta_z - \eta_z\zeta_y) & -(\eta_x\zeta_z - \eta_z\zeta_x) & (\eta_x\zeta_y - \eta_y\zeta_x) \\ -(\xi_y\zeta_z - \xi_z\zeta_y) & (\xi_x\zeta_z - \xi_z\zeta_x) & -(\xi_x\zeta_y - \xi_y\zeta_x) \\ (\xi_y\eta_z - \xi_z\eta_y) & -(\xi_x\eta_z - \xi_z\eta_x) & (\xi_x\eta_y - \xi_y\eta_x) \end{Bmatrix}, \tag{A.16a}$$

or from Eq. (A.2)

$$[J^{-1}] = [e_1 \quad e_2 \quad e_3]^T = \begin{Bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{Bmatrix},$$

$$= \begin{Bmatrix} (\eta_y\zeta_z - \eta_z\zeta_y) & -(\xi_y\zeta_z - \xi_z\zeta_y) & (\xi_y\eta_z - \xi_z\eta_y) \\ -(\eta_x\zeta_z - \eta_z\zeta_x) & (\xi_x\zeta_z - \xi_z\zeta_x) & -(\xi_x\eta_z - \xi_z\eta_x) \\ (\eta_x\zeta_y - \eta_y\zeta_x) & -(\xi_x\zeta_y - \xi_y\zeta_x) & (\xi_x\eta_y - \xi_y\eta_x) \end{Bmatrix} / |J| . \tag{A.16b}$$

The relationship between vector bases can be obtained also by matrix algebra. From basic matrix identity, Jacobian can be written as

$$[J] = [J^{-1}] = \frac{\text{Transpose of cofactor } [J^{-1}]}{|J^{-1}|} = \frac{[[J^{-1}]^*]^T}{|J^{-1}|} . \tag{A.17}$$

Equation (A.17) is the same as Eq. (A.14). There also exists an inverse relation for Eq. (A.17).

A.2 Vector Representation in Curvilinear Coordinates

A vector $F(F_x i, F_y j, F_z k)$ can be expressed in contravarient coordinates as

$$F_i = (g_{ii})^{1/2} F^i \quad . \tag{A.18}$$

This is a projection of F_i on x^i coordinate

$$\bar{F}^i = \frac{F_i \, e^i}{|e^i|} \quad ,$$

where

$$|e^i| \, F^i = \frac{\partial x^i}{\partial x} F_x + \frac{\partial x^i}{\partial y} F_y + \frac{\partial x^i}{\partial z} F_z \quad . \tag{A.19}$$

Equation (A.19) can be expressed as

$$\begin{Bmatrix} \bar{F}_\xi/(g_{11})^{1/2} \\ \bar{F}_\eta/(g_{22})^{1/2} \\ \bar{F}_\zeta/(g_{33})^{1/2} \end{Bmatrix} = \begin{pmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{pmatrix} \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = [J] \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} \tag{A.20}$$

where g_{ij} is defined in Eq. (A.8).

The inverse relation to Eq. (A.20) is

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = [J^{-1}] \begin{Bmatrix} \bar{F}_\xi/(g_{11}^{1/2}) \\ \bar{F}_\eta/(g_{22}^{1/2}) \\ \bar{F}_\zeta/(g_{33}^{1/2}) \end{Bmatrix} \quad . \tag{A.21}$$

For example, the velocity vector in covariant coordinates can be written as

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} U/(g_{11}^{1/2}) \\ V/(g_{22}^{1/2}) \\ W/(g_{33}^{1/2}) \end{Bmatrix} . \qquad \text{(A.22a)}$$

Also, the velocity vector in contravariant coordinates can be expressed as

$$\begin{Bmatrix} U/(g_{11})^{1/2} \\ V/(g_{22})^{1/2} \\ W/(g_{33})^{1/2} \end{Bmatrix} = [J] \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \qquad \text{(A.22b)}$$

where u and U are velocities in the covariant and the contravariant coordinate systems, respectively.

A.3 Normal Derivative in Curvilinear Coordinates

A normal derivative of a scalar variable (A) can be computed as

$$\left(\frac{\partial A}{\partial n}\right)^{i} = \bar{n} \cdot \nabla A \qquad \text{(A.23a)}$$

where

$$\bar{n} = \frac{\nabla \bar{S}}{|\nabla S|} = \frac{\bar{e}_i}{|e_i|} = \frac{\bar{e}_i}{(g^{ii})^{1/2}} , \qquad \text{(A.23b)}$$

$$\bar{\nabla} A = A_x \bar{i} + A_y \bar{j} + A_z \bar{k} \qquad \text{(A.23c)}$$

or

$$\begin{Bmatrix} A_x\ i \\ A_y\ j \\ A_z\ k \end{Bmatrix} = [J]^T \begin{Bmatrix} A_\xi \\ A_\eta \\ A_\zeta \end{Bmatrix} . \qquad \text{(A.24a)}$$

Equation (A.23a) can be written as

$$\left.\frac{\partial A}{\partial n}\right|_{\substack{=\\ \text{constant } \phi}} = \frac{[\phi_x\ \phi_y\ \phi_z]\ [J]^T}{(\phi_x^2 + \phi_y^2 + \phi_z^2)^{1/2}} \begin{Bmatrix} A_\xi \\ A_\eta \\ A_\zeta \end{Bmatrix} . \qquad \text{(A.24b)}$$

For a constant ξ, Eq. (A.24b) can be written as

$$\left.\frac{\partial A}{\partial n}\right|_\xi = \frac{g^{11} A_\xi + g^{12} A_\eta + g^{13} A_\zeta}{(g^{11})^{1/2}} . \qquad \text{(A.25a)}$$

For a constant η, Eq. (A.24b) can be written as

$$\left.\frac{\partial A}{\partial n}\right|_\eta = \frac{g^{21} A_\xi + g^{22} A_\eta + g^{23} A_\zeta}{(g^{22})^{1/2}} . \qquad \text{(A.25b)}$$

For a constant ζ, Eq. (A.24b) can be written as

$$\left.\frac{\partial A}{\partial n}\right|_\zeta = \frac{g^{31} A_\xi + g^{32} A_\eta + g^{33} A_\zeta}{(g^{33})^{1/2}} , \qquad \text{(A.25c)}$$

also

$$(A)^i_n = \frac{e^i}{|e^i|} \cdot \nabla A = \frac{1}{|e^i|} \sum_{j=1}^{3} (e^i \cdot e^j) A^j_i$$

$$= \frac{1}{(g_{ii})^{1/2}} \sum_{j=1}^{3} g^{ij} \frac{\partial A}{\partial x^i} , \qquad (A.26)$$

e.g.

$$(A)^\xi_n = \frac{1}{(g^{11})^{1/2}} [g^{11} A_\xi + g^{12} A_\eta + g^{13} A_\zeta] . \qquad (A.27)$$

This equation is similar to Eq. (A.25a).

A.4 Miscellaneous Relations

The angle between two grid lines is given by

$$\cos \theta_{ij} = \frac{e_i \cdot e_j}{|e_i| \, |e_j|} ,$$

$$\cos \theta_{ij} = \frac{g_{ij}}{[|g_{ii}| \, |g_{jj}|]^{1/2}} . \qquad (A.28)$$

Therefore, for the orthogonal grid, the following should be true,

$$g_{ij} = 0 \quad \text{for} \quad i \neq j . \qquad (A.29)$$

Arclength is defined as

$$(ds)^2 = \sum_{i=1}^{3} \sum_{j=1}^{3} g_{ij} \, dx^i \, dx^j . \qquad (A.30a)$$

An arclength along x^i coordinate is defined as

$$(ds)^i = (g_{ii})^{1/2} \, dx^i . \qquad (A.30b)$$

The area of an element on which the x^i constant is defined as

$$(d\Gamma^i = |\underline{e}_j \, d\, x^j \cdot \underline{e}_k \, d\, x^k| = (g_{ij})\, dx^i\, dx^j \,, \qquad (A.31)$$

e.g.

$$d\Gamma^\xi = |e_2\, d\eta \times e_3\, d\zeta| = \sqrt{G_{11}}\, d\eta\, d\zeta = \frac{\sqrt{g_{11}}}{J}\, d\eta\, d\zeta \,, \qquad (A.32a)$$

$$d\Gamma^\eta = G_{22}\, d\xi\, d\zeta = \frac{\sqrt{g_{22}}}{J}\, d\xi\, d\zeta \,, \qquad (A.32b)$$

$$d\Gamma^\zeta = \sqrt{G_{33}}\, d\xi\, d\eta = \frac{\sqrt{g_{33}}}{J}\, d\xi\, d\eta \,.$$

The volume of an element is defined as

$$dV = \frac{\partial(x,y,z)}{d(\xi,\eta,\zeta)} = |J^{-1}|\, d\xi\, d\eta\, d\zeta$$

$$= |e_1 \cdot (e_2 \times e_3)|\, d\xi\, d\eta\, d\zeta = (|g_{ij}|)^{1/2}\, d\xi\, d\eta\, d\zeta \,. \qquad (A.33)$$

APPENDIX B

TIME-STEP ANALYSIS

In order to analyze the governing equations, they can be divided into two parts: inviscid and viscous. This makes the task of analyzing much simpler. However, this is another assumption which may impair the results.

B.1 Inviscid Part

For the inviscid part, the governing equations can be written as

$$\rho_t + \nabla \cdot (\rho u) = 0 \,, \tag{B.1a}$$

$$(\rho u)_t + \nabla \cdot (\rho u u) + (\nabla \cdot P)\, \delta_{ij} = 0 \,, \tag{B.1b}$$

$$e_t + \nabla \cdot (e v) + [\nabla \cdot (\rho u)]\, \delta_{ij} = 0 \,, \tag{B.1c}$$

$$e = \frac{P}{\gamma - 1} + \frac{\rho(u^2 + v^2 + w^2)}{2} \,, \tag{B.1d}$$

$$\gamma P = \rho c^2 \,. \tag{B.1e}$$

After some algebraic manipulations, Eqs. (B.1a)-(B.1d) can be written as

$$q_t + A\, q_x + B\, q_y + C\, q_z = 0 \tag{B.2}$$

where

$$q = (\rho, u, v, w, P)^T \,,$$

$$A = \begin{Vmatrix} u & \rho & 0 & 0 & 0 \\ 0 & u & 0 & 0 & 1/\rho \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & \rho c^2 & 0 & 0 & u \end{Vmatrix}, \quad B = \begin{Vmatrix} v & 0 & \rho & 0 & 0 \\ 0 & v & 0 & 0 & 0 \\ 0 & 0 & v & 0 & 1/\rho \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & \rho c^2 & 0 & v \end{Vmatrix},$$

$$C = \begin{Vmatrix} w & 0 & 0 & \rho & 0 \\ 0 & w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ 0 & 0 & 0 & w & 1/\rho \\ 0 & 0 & 0 & \rho c^2 & w \end{Vmatrix}.$$

Equation (B.2) can be transformed from physical coordinates to computational coordinates as

$$q_t + \bar{A}\, q_\xi + \bar{B}\, q_\eta + \bar{C}\, q_\zeta = 0 ,$$

where

$$\begin{aligned} \bar{A} &= A\,\xi_x + B\,\xi_y + C\,\xi_z , \\ \bar{B} &= A\,\eta_x + B\,\eta_y + C\,\eta_z , \\ \bar{C} &= A\,\zeta_x + B\,\zeta_y + C\,\zeta_z . \end{aligned} \tag{B.3}$$

This equation can be split as follows

$$\begin{aligned} q^{n+1} - q^n + \bar{A}\,\Delta t_\xi\, q^n_\xi &= 0 , \\ q^{n+2} - q^{n+1} + \bar{B}\,\Delta t_\eta\, q^{n+1}_\eta &= 0 , \\ q^{n+3} - q^{n+2} + \bar{C}\,\Delta t_\zeta\, q^{n+2}_\zeta &= 0 . \end{aligned} \tag{B.4}$$

This splitting is valid and stable if the time-step of each operator does not exceed the allowable step-size for each operator. It is consistent if the sum of the time steps for each operator be equal. This method would be second order accurate if the sequences of the

operator are symmetric. In order to analyze this equation, finite-Fourier series can be introduced to monitor the growth and decay of the error as

$$q = \bar{q}\, e^{st}\, e^{ik\xi} \ . \tag{B.5}$$

Substituting this equation in Eq. (B.4) results in

$$\frac{e^{s\Delta t} - 1}{\Delta\, t_\xi} + \bar{A}\, \frac{e^{ik\Delta\eta} - \bar{e}^{ik\Delta\eta}}{2\,\Delta\,\eta} = 0 \ . \tag{B.6}$$

Rearranging and collecting terms, yields the following

$$e^{s\Delta t} = [G] = \left[I - i\,\bar{A}\left(\frac{\Delta t_\xi}{\Delta\xi}\sin\theta\right)\right] \ , \ \theta = k\,\Delta\,\xi \ . \tag{B.7}$$

The system is stable if the largest eigenvalue is less than unity. This condition insures that the error always decays.

$$|G| < 0 \ . \tag{B.8}$$

Therefore, the determinant of [G] can be set equal to zero

$$\det\ \left[\bar{A}\,\frac{\Delta t_\xi}{\Delta\xi}\sin\theta - \lambda I\right] = 0 \ ,$$

or

$$\begin{Bmatrix} E\bar{u}-\lambda & \rho\xi_x E & \rho\xi_y E & \rho\xi_z E & 0 \\ 0 & E\bar{u}-\lambda & 0 & 0 & E\xi_{x/\rho} \\ 0 & 0 & E\bar{u}-\lambda & 0 & E\xi_{y/\rho} \\ 0 & 0 & 0 & E\bar{u}-\lambda & E\xi_{z/\rho} \\ 0 & \rho c\xi_x E & \rho c\xi_y E & \rho c\xi_z E & E\bar{u}-\lambda \end{Bmatrix} = 0 \ ,$$

where

$$E = \frac{\Delta t_\xi}{\Delta \xi} \sin\theta \, (A\xi_x + B\,\xi_y + C\,\xi_z) \, ,$$

$$\bar{u} = u\,\xi_x + v\,\xi_y + w\,\xi_z \, .$$

The determinant can be written as

$$(\bar{u}\,E - \lambda)^3 \left[(\bar{u}\,E - \lambda)^2 - E^2\,c^2\,(\xi_x^2 + \xi_y^2 + \xi_z^2)\right] = 0 \, . \qquad (B.10)$$

Equation (B.10) can be solved for λ as

$$\lambda_{1,2,3} = \frac{\Delta t_\xi}{\Delta \xi} \sin\theta \, [|u\,\xi_x + v\,\xi_y + w\,\xi_z|] \, , \qquad (B.11)$$

$$\lambda_{4,5} = \frac{\Delta t_\xi}{\Delta \xi} \sin\theta [\,|u\,\xi_x + v\,\xi_y + w\,\xi_z| + c(\xi_x^2 + \xi_y^2 + \xi_z^2)^{1/2}] \, .$$

Consequently, there are five eigenvalues, and the time step is based on the maximum of them,

$$\Delta t_\xi \leq \frac{\Delta \xi}{|u\,\xi_x + v\,\xi_y + w\,\xi_z| + c(\xi_x^2 + \xi_y^2 + \xi_z^2)^{1/2}} \, . \qquad (B.12a)$$

Similar expressions can be found for Δt_η and Δt_ζ

$$\Delta t_\eta \leq \frac{\Delta \eta}{|u\,\eta_x + v\,\eta_y + w\,\eta_z| + c(\eta_x^2 + \eta_y^2 + \eta_z^2)^{1/2}} \, , \qquad (B.12b)$$

$$\Delta t_\zeta \leq \frac{\Delta \zeta}{|u\,\zeta_x + v\,\zeta_y + w\,\zeta_z| + c(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)^{1/2}} \, . \qquad (B.12c)$$

B.2 Viscous Part

For the viscous part, the governing equations can be written as

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = 0 \quad , \tag{B.13}$$

where

$$u = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{Bmatrix} \quad , \quad F = \begin{Bmatrix} 0 \\ -\tau_{xx} \\ -\tau_{xy} \\ -\tau_{xz} \\ q_x - \phi_x \end{Bmatrix} \quad , \quad G = \begin{Bmatrix} 0 \\ -\tau_{xy} \\ -\tau_{yy} \\ -\tau_{yz} \\ q_y - \phi_y \end{Bmatrix} \quad , \quad H = \begin{Bmatrix} 0 \\ -\tau_{xz} \\ -\tau_{yz} \\ -\tau_{zz} \\ q_z - \phi_z \end{Bmatrix} \quad .$$

Equation (B.13) can be transformed from physical coordinates to computational coordinates as

$$\frac{\partial \bar{U}}{\partial t} + [J] \begin{Bmatrix} F_\xi & F_\eta & F_\zeta \\ G_\xi & G_\eta & G_\zeta \\ H_\xi & H_\eta & H_\zeta \end{Bmatrix} = 0 \quad . \tag{B.14}$$

Similar to the previous case, Eq. (B.4) is split into three components. For example, the ξ component can be written as

$$\frac{\partial U}{\partial t} + \xi_x F_\xi + \xi_y G_\xi + \xi_z H_\xi = 0 \quad . \tag{B.15}$$

After some algebraic manipulation, Eq. (B.15) can be written as

$$\frac{\partial q}{\partial t} + [A] q_{\xi\xi} + [B] q_{\xi\eta} + [C] q_{\xi\xi} = 0 \quad , \tag{B.16}$$

where

$$q = \begin{Bmatrix} \rho \\ u \\ v \\ w \\ p \end{Bmatrix} \quad ,$$

$$[A] = \begin{Bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & (\lambda-\mu)\xi_x^2+\mu g^{11} & (\phi+\mu)\xi_y\xi_x & (\phi+\mu)\xi_x\xi_z & 0 \\ 0 & (\phi+\mu)\xi_y\xi_x & (\lambda-\mu)\xi_y^2+\mu g^{11} & (\phi+\mu)\xi_z\xi_y & 0 \\ 0 & (\phi+\mu)\xi_z\xi_x & (\phi+\mu)\xi_z\xi_y & (\lambda-\mu)\xi_z^2+\mu g^{11} & 0 \\ \alpha g^{11} & 0 & 0 & 0 & \beta g^{11} \end{Bmatrix},$$

$$[B] = \begin{Bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & (\lambda-\mu)\xi_x\eta_x+\mu g^{12} & \phi\xi_x\eta_y+\mu\xi_y\eta_x & \phi\xi_x\eta_z+\mu\xi_z\eta_x & 0 \\ 0 & \phi\xi_y\eta_x+\mu\xi_x\eta_y & (\lambda-\mu)\xi_y\eta_y+\mu g^{12} & \phi\xi_y\eta_z+\mu\xi_z\eta_y & 0 \\ 0 & \phi\xi_z\eta_x+\mu\xi_x\eta_z & \phi\xi_z\eta_y+\mu\xi_y\eta_z & (\lambda-\mu)\xi_z\eta_x+\mu g^{12} & 0 \\ \alpha g^{12} & 0 & 0 & 0 & \beta g^{12} \end{Bmatrix},$$

$$[C] = \begin{Bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & (\lambda-\mu)\xi_x\zeta_x+\mu g^{13} & \phi\xi_x\zeta_y+\mu\xi_y\zeta_y & \phi\xi_x\zeta_z+\mu\xi_z\zeta_x & 0 \\ 0 & \phi\xi_y\zeta_x+\mu\xi_x\zeta_y & (\lambda-\mu)\xi_y\zeta_y+\mu g^{13} & \phi\xi_y\zeta_z+\mu\xi_y\zeta_y & 0 \\ 0 & \phi\xi_z\zeta_x+\mu\xi_x\zeta_y & \phi\xi_z\zeta_y+\mu\xi_y\zeta_z & (\lambda-\mu)\xi_z\zeta_z+\mu g^{13} & 0 \\ \alpha g^{13} & 0 & 0 & 0 & \beta g^{13} \end{Bmatrix},$$

where

$$\lambda = \frac{4}{3}\mu + \mu_B$$

$$\phi = \mu_B - \frac{2}{3}\mu$$

$$\alpha = -\frac{\gamma\mu P}{Pr\ \rho^2\ (\gamma-1)}$$

$$\beta = \frac{\gamma\mu}{Pr\ \rho(\gamma-1)}$$

Similar to the inviscid case, the following finite-Fourier series can be introduced as

$$q = \bar{q} \cdot e^{st} \cdot e^{ik_1\xi} \cdot e^{ik_2\eta} \cdot e^{ik_3\zeta} . \qquad (B.17)$$

Substitution of Eq. (B.17) into (B.16) results in the following

$$[G] = [I - \bar{A} - \bar{B} - \bar{C}] \qquad (B.18)$$

where

$$\bar{A} = \frac{4\Delta t_\xi \sin^2 \frac{\theta_1}{2}}{\Delta \xi^2} A, \quad \bar{B} = \frac{- \sin \theta_1 \sin \theta_2 \Delta t_\xi}{\Delta\xi \Delta\eta} B ,$$

$$\bar{C} = \frac{- \sin \theta_1 \sin \theta_2 \Delta t_\xi}{\Delta\xi \Delta\zeta} C ,$$

$$\theta_1 = k_1 \Delta\xi , \theta_2 = k_2 \Delta\eta , \theta_3 = k_3 \Delta\zeta .$$

This system is stable if the largest eigenvalue is less than unity, this condition insures the decay of error. Therefore, the determinant of [G] can be set equal to zero as

$$\begin{pmatrix} -\lambda & 0 & 0 & 0 & 0 \\ 0 & G_{22}-\lambda & G_{23}-\lambda & G_{24}-\lambda & 0 \\ 0 & G_{32}-\lambda & G_{33}-\lambda & G_{32}-\lambda & 0 \\ 0 & G_{42}-\lambda & G_{43}-\lambda & G_{44}-\lambda & 0 \\ G_{51}-\lambda & 0 & 0 & 0 & G_{55}-\lambda \end{pmatrix} = 0 , \qquad (B.19a)$$

where

$$G_{ij} = \bar{A}_{ij} + \bar{B}_{ij} + \bar{C}_{ij}$$

or

$$\lambda(G_{55}-\lambda)\,[(G_{22}-\lambda)\,[(G_{33}-\lambda)\,(G_{44}-\lambda) - (G_{32}-\lambda)\,(G_{43}-\lambda)]$$

$$- (G_{23}-\lambda)\,[(G_{32}-\lambda)\,(G_{44}-\lambda) - (G_{32}-\lambda)\,(G_{42}-\lambda)]$$

$$+ (G_{24}-\lambda)\,[(G_{32}-\lambda)\,(G_{43}-\lambda) - (G_{33}-\lambda)\,(G_{42}-\lambda)]] = 0 \ . \quad \text{(B.19b)}$$

Equation (B.19b) can be solved for λ as

$$\lambda_1 = 0 \ , \ \lambda_2 = G_{55} \ . \quad \text{(B.20)}$$

or

$$\lambda_2 = \beta\left[\frac{4 \sin^2 \frac{\theta_1}{2}}{\Delta\xi} g^{11} - \frac{\sin\theta_1 \sin\theta_2}{\Delta\eta} g^{12} + \frac{\sin\theta_1 \sin\theta_3}{\Delta\zeta} g^{13}\right] \frac{\Delta t_\xi}{\Delta\xi} \ .$$

After all algebraic manipulations, the following are found

$$\Delta t_\xi \leq \frac{\gamma\mu}{\rho P_r(\gamma-1)} \left[2 \frac{\xi_x^2 + \xi_y^2 + \xi_z^2}{\Delta\xi^2} + \left|\frac{\xi_x \eta_x + \xi_y \eta_y + \xi_z \eta_z}{\Delta\xi \ \Delta\eta}\right| + \left|\frac{\xi_x \zeta_x + \xi_y \zeta_y + \xi_z \zeta_z}{\Delta\xi \ \Delta\zeta}\right|\right]^{-1} . \quad \text{(B.21a)}$$

Similar expressions can be found for Δt_η and Δt_ζ

$$\Delta t_\eta \leq \frac{\gamma\mu}{\rho P_r(\gamma-1)} \left[2 \frac{\eta_x^2 + \eta_y^2 + \eta_z^2}{\Delta\eta^2} + \left|\frac{\xi_x \eta_x + \xi_y \eta_y + \xi_z \eta_z}{\Delta\xi \ \Delta\eta}\right| + \left|\frac{\eta_x \zeta_x + \eta_y \zeta_y + \eta_z \zeta_z}{\Delta\eta \ \Delta\zeta}\right|\right]^{-1}, \quad \text{(B.21b)}$$

$$\Delta t_\zeta < \frac{\gamma\mu}{\rho P_r(\gamma-1)} \left[2 \frac{\zeta_x^2 + \zeta_y^2 + \zeta_z^2}{\Delta\zeta^2} + \left| \frac{\zeta_x \eta_x + \zeta_y \eta_y + \zeta_z \eta_z}{\Delta\zeta \Delta\eta} \right| + \left| \frac{\zeta_x \xi_x + \zeta_y \xi_y + \zeta_z \xi_z}{\Delta\zeta \Delta\xi} \right| \right]^1 . \quad \text{(B.21c)}$$

Three more eigenvalues need to be computed from Eq. (B.21b) which requires solution of a nonlinear equation for each point in the computational space.

BIOGRAPHY

Jamshid Samareh Abolhassani was born on March 25, 1959, in Tehran, Iran. He was raised in the southern part of Tehran where he obtained his early education. He graduated from Rahnama High School in May 1977. In the summer of 1978, he came to the United States to pursue his education. After one month of studying English in Washington, D.C., he moved to Montgomery, West Virginia. There, he enrolled in the mechanical engineering program at West Virginia Institute of Technology. He received his Bachelor of Science degree in mechanical engineering in May 1981. Then, he moved to Norfolk, Virginia, where he attended the Old Dominion University and received his Master of Engineering in May 1983. During his graduate work at Old Dominion, he was very closely associated with the Computer Applications Branch of the Analysis and Computation Division of NASA Langley Research Center. His fascination for mathematics and science shaped the foundation for his interest in digital computation which was the basis of this research program.