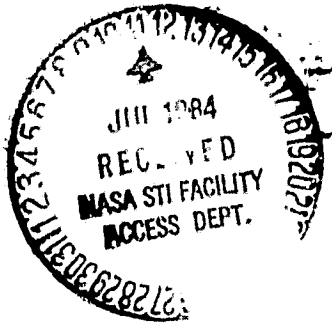


General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.



Solution of Three-Dimensional Time-Dependent Viscous Flows
Part 2: Development of the Computer Code

(NASA-CR-166565-Pt-2) SOLUTION OF
3-DIMENSIONAL TIME-DEPENDENT VISCOUS FLOWS.
PART 2: DEVELOPMENT OF THE COMPUTER CODE
(Scientific Research Associates, Inc.) 82 p
HC A05/MF A01

N84-27002

Unclas
13669

CSCL 20D G3/34

Bernard C. Weinberg
Henry McDonald

CONTRACT NAS2- 10016
December 1980



Solution of Time-Dependent Viscous Flows
Part 2: Development of the Computer Code

Bernard C. Weinberg
Henry McDonald
Scientific Research Associates, Inc.
Glastonbury, Connecticut

Prepared for
Ames Research Center
under Contract NAS2-10016



National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

TABLE OF CONTENTS

	PAGE
SUMMARY	1
INTRODUCTION	2
LIST OF SYMBOLS	4
ANALYSIS	6
Background	6
Coordinate System	10
Governing Equations	12
Continuity Equation	12
Momentum Equations	14
Energy Equation	19
Equation of State	19
Linearizations	20
The Turbulence Model	21
Spatial Difference Approximations	21
QR Operator	21
Linearized Block Implicit Scheme	26
Implementation of the LBI Scheme Employing the QR Operator Technique	32
Boundary Conditions and Initial Conditions	36
The Computer Code	37
NUMERICAL RESULTS	39
Oscillating Flat Plate	39
Three-Dimensional Flat Plate Boundary Layer	41
Two-Dimensional Howarth Flow	42
CONCLUSIONS	44
APPENDICES	45
Appendix A - Linearization Technique	45
Appendix B - Tensor Notation and General Coordinates	50
REFERENCES	64
TABLES	67
FIGURES	72

SUMMARY

The flow over a helicopter rotor blade in forward flight is an important example of three-dimensional time-dependent flow. The boundary layers on the rotor blade set loss levels and control retreating blade stall. As a consequence there is considerable interest in developing a numerical scheme for solving the time-dependent viscous compressible three-dimensional flow equations to aid in the design of helicopter rotors.

In the present report, the development of a computer code to solve a three-dimensional unsteady approximate form of the Navier-Stokes equations employing a Linearized Block Implicit technique in conjunction with a QR operator scheme is described. Results of calculations of several Cartesian test cases are presented. These results indicate that the computer code can be applied to more complex flow fields such as these encountered on rotating airfoils.

INTRODUCTION

The behavior of boundary layers on wings and bodies has long been of interest to aerodynamicists. In both steady and unsteady flows the boundary layers are known to govern a major portion of the losses and to significantly influence the vehicle lift and moment coefficients. When the flow is steady, boundary layer prediction schemes based on numerical solution to the governing partial differential equations of motion have reached a high level of sophistication and predictive accuracy, even in three space dimensions. In unsteady flows, such as are commonly encountered in rotary winged aircraft, some progress has been made in two space dimensions but little to date has appeared on unsteady three-dimensional boundary layers.

Two particular problems arise with time-dependent three-dimensional boundary layers relative to the steady case. The first of these is the rather obvious one of time integration with its added requirements of transient accuracy coupled with an increase in the computational labor. The second of these is the so-called negative cross flow problem, which to some extent has troubled the steady boundary layer prediction schemes. Kendall, et al (Ref. 1) discuss the negative cross flow problem for steady three-dimensional boundary layers in a very illuminating fashion. This particular problem arises when the spanwise component of velocity changes sign and will be discussed in detail subsequently. Because of the interest by external aerodynamicists in swept wing boundary layers where the negative cross flow problem (in this case flow from tip to root) is not usually encountered, the negative cross flow problem has not received a great deal of attention to date. However in transient flows, particularly those encountered on rotor blades in forward flight, negative cross flows are frequently encountered. For instance, the advancing rotor blade has cross flows of one sign during the first ninety degrees of rotation and these can change sign over part of the blade during the second ninety degrees.

Thus to be of practical value, time-dependent three-dimensional boundary layer prediction schemes require high computational efficiency and transient accuracy coupled to the ability to treat arbitrary cross flow profiles.

In this report we describe the development of a computer code for the efficient solution of three-dimensional time-dependent viscous flows on fixed and rotary aircraft. The Linearized Block Implicit (LBI) technique of Briley and McDonald (Ref. 2) in coordination with a tridiagonal QR operator scheme (Ref. 3) is employed to solve the reduced turbulent Navier-Stokes equations which are derived for nonorthogonal coordinates in generalized tensor form. The rationale for the choice of this approach is discussed in detail in Ref. 3.

The basic assumptions made in the derivation of these equations are that the pressure does not vary normal to the shear layer and that in the energy equation the square of the normal velocity is neglected with respect to the other velocity components ($T_0 = \text{constant}$). The latter assumption is included only for computational simplification purposes and is not essential in the analysis. A novel method is employed for solving the continuity equation in conjunction with the reduced Navier-Stokes equations. The continuity equation is split by employing the Douglas-Gunn procedure to obtain a consistent approximation to the full equation which is then solved as an integral. Results of computations on model problems in Cartesian coordinates are presented.

LIST OF SYMBOLS

a, b, c, d	coefficients of differential equation
a_{ij}^n	coefficient of differential operator
\mathcal{D}	nonlinear three-dimensional differential operator
ds	differential arc length
\vec{e}_i	covariant basis vector
$f_j, f(x)$	right-hand side of differential equation
g_{ij}, g^{ij}	components of metric tensor
h	step size
h_1, h_2	scale factors
J	Jacobian
$L(u)$	linear differential operator
\mathcal{L}_i	linear three-dimensional differential operator
L_i	product operator $\vec{Q}_i \mathcal{L}_i$
M^n	difference between \mathcal{D}^n and \mathcal{L}^n
P	pressure
Q	tridiagonal difference operator
q^-, q^c, q^+	components of Q
\vec{q}	velocity vector
R	tridiagonal difference operator
r^-, r^c, r^+	components of R
R_e	Reynolds number
R_{c_j}	cell Reynolds number
S	source term
t	time
T	static temperature
T_o	stagnation temperature

u^i contravariant velocity component
 u, v, w physical velocity component
 x^i coordinate direction in i direction

Greek Symbols

β factor for centering time step
 γ ratio of specific heats
 Γ_{ij}^k Christoffel symbol
 δ_j^i Kronecker delta function
 Δ velocity divergence
 Δt time step
 Δx_i step size in i direction
 e^{jkl} strain tensor
 η boundary layer coordinate
 λ viscous stability parameter
 μ molecular viscosity
 ρ density
 σ^{ik} stress tensor
 ϕ vector of unknowns in LBI scheme
 ψ source term in LBI scheme
 ω frequency of oscillation
 ω CFL stability parameter

Subscripts

e edge
 p physical
 ∞ free stream

Superscripts

n time level
 $*, **, ***$ intermediate time level in ADI procedure

ANALYSIS

Background

In this section the requirements of a three-dimensional unsteady viscous flow computer code for flow over airfoils are discussed.

Three-dimensional boundary layers occur on the wings and fuselages of both fixed and rotary wing aircraft. In both types of vehicles, the boundary layers are important in setting loss levels and determining useful operating ranges. As is well known, boundary layers are sensitive to pressure gradients. In time-dependent flow the temporal acceleration terms appear in the momentum equation in a form very similar to the conventional imposed pressure gradient and so for qualitative evaluation purposes can be regarded as "pseudo" or "auxiliary" pressure gradients. Viewed in this manner the temporal acceleration terms can be seen to influence quantities of practical importance such as skin friction, displacement thickness and the onset of separation. At the range of frequencies typically encountered in rotary wing aircraft aerodynamic problems, it is clear, for instance, from the extensive review of McCroskey (Ref. 4), that very significant transient boundary layer effects can be observed.

In examining the flow problems of practical interest such as loss levels or the onset of separation it is evident that all three space dimensions must be considered. In conventional aircraft the sweep effect is of interest and inherently three-dimensional. In rotary wing aircraft in forward flight clearly very substantial transient changes occur in what might be termed the local sweep angle. However, generally speaking, the boundary layers remain thin unless catastrophic flow separation occurs or the flow at the wing or rotor tip is considered. As a consequence it might be supposed that the usual three-dimensional thin boundary sheet approximations (Nash and Patel, Ref. 5) could be used to produce a valid set of governing equations. Fortunately some improvements in thin boundary sheet approximations are possible as a result of having to eliminate the negative cross flow problem mentioned earlier.

The negative cross flow problem is best explained in a somewhat intuitive manner, and a good physical description of the problem is given by Kendall, et al (Ref. 1). Looking at the suction surface of a conventional swept back wing the boundary layer cross flow, w , is usually outward in the z positive direction along the span from root to tip. Thus conventional boundary layer integration schemes have developed by forward marching the streamwise velocity u in the streamwise x direction and simultaneously marching out along the span in the z positive direction. In view of the physics of the problem, the spanwise marching scheme does not normally encounter

negative w , i.e., spanwise inflow. This is very fortunate because it is difficult, indeed it could be argued impossible, to structure a physically satisfactory unconditionally stable noniterative scheme which permits forward marching in the spanwise direction with a negative w cross flow. At least intuitively the problem of negative cross flow implies information being transferred upstream against the spanwise marching direction. Conventional stability analyses confirm the inability to forward march into regions of significant negative w . From experience with attempts to march the two-dimensional boundary layer equations into a region of separated flow and its obvious relationship to the negative cross flow problem, it is not surprising that spanwise marching into a negative cross flow region is not accomplished without special treatment. Recently conventional boundary layer developers have been turning to performing an implicit spanwise construction to remove the restriction of only positive cross flows (Kendall, et al, Ref. 1). Lin and Rubin (Ref. 6) in their predictor-corrector boundary region solutions for flow over a yawed cone at moderate incidence also show that allowing diffusion in the spanwise direction not only eliminates the problems associated with negative cross flow, but improves upon the solutions obtained by three-dimensional boundary layer techniques.

Boundary conditions applied at the tip can influence the flow inboard, if required by the physics of the flow. For these reasons the implicit spanwise construction has been a feature of the three-dimensional duct flow analysis of Briley (Ref. 7) and McDonald and Briley (Ref. 8). As a consequence of these observations and the need to remove the negative cross flow restriction, a spanwise implicit formulation seems mandatory for the rotary wing applications and at least desirable for fixed wing applications, especially as it can be had for a very modest increase in code computational labor. Based on the experience in Refs. 7 and 8, the spanwise implicit sweep would only result in about a 20% increase relative to the explicit spanwise marching approach. The extension of the conventional three-dimensional boundary layer equations to allow spanwise diffusion is easily accomplished, and in view of the improved physical representation which thus follows, it is recommended and has been implemented in this effort.

As a matter of course it has been assumed that normal to the wall an implicit formulation would be structured. In recent years for boundary layer type problems there has been little dispute as to the efficiency gains to be had from an implicit formulation normal to the wall (Ref. 9). However in the streamwise direction for steady 2-D flow, the equations are normally forward marched and the implicit stability obtained entirely from being implicit in the normal to the wall direction. In time-dependent flows a similar structure is to be had so that at each time level one

streamwise (explicit) forward marching sweep could be made with two implicit sweeps in the spanwise and normal directions to give the desired unconditional stability. As mentioned earlier the explicit sweep would probably require less computational effort by about 20% than an implicit streamwise sweep and of course less storage. However, since the solution is being time marched the opportunity to take a streamwise implicit sweep at roughly the same cost as the explicit sweep does arise. If one does perform a streamwise explicit sweep, then the linearization of nonlinear terms is performed about the known spatial marching level. If an implicit streamwise structure is adopted, then full time linearization can be utilized. That is the linearization of the nonlinear terms is performed about the known time level. As is pointed out in Ref. 8, it is easier to obtain a consistent spatial-temporal order accurate linearization by marching in time than in space (in time the nonlinear marching derivatives have the form $u_{,t}$ whereas in space marching they have the form $u_{,i}u_{,j}$). Further by structuring implicitly in the space marching direction, (small) regions of axial reverse flow would be permitted. As a result of these combined benefits of linearization and separation, a streamwise implicit structure is advocated and has been implemented in this effort.

Transient calculations mean that, in essence, a full 3-D spatial integration is carried out at each time step. Thus, spatial accuracy is very important to minimize the spatial grid point density for efficiency since many time steps are contemplated in a given cycle. In order to get the most out of a given spatial difference formula, the errors from representing nonlinear terms by linear combinations of terms should be less than or equal to the spatial discretization errors. If the linearization introduces a greater error than the spatial differencing, then either a coarser spatial mesh could be used, or iteration, or some form of linearization improvement is called for. Iteration across a time step is not recommended since this only reduces the linearization error and computationally costs as much as a complete time step. Cutting back the time step would be preferable to iterating to preserve the linearization error at some acceptable level, since cutting back on the time step would improve both the transient error and the linearization error. This point is clearly demonstrated in Ref. 3. To obtain a linearization, which introduces errors of at most the same as the spatial difference formulae, a Taylor series expansion about the known time level can be performed. This process clearly demands a formal block, i.e., coupled, treatment of the system of equations. For instance in the streamwise momentum equation a typical term is linearized:

$$(uw)^{n+1} = u^{n+1}w^n + u^n w^{n+1} - u^n w^n + O(\Delta t^2)$$

and clearly one cannot lag w^{n+1} at the old time level n without introducing a first order time error in order to get an uncoupled system, i.e., w^{n+1} not appearing in the streamwise momentum equation. Thus formal linearization and consideration of the resulting errors indicate the coupled system ought to be treated from the accuracy point of view. This is further reinforced when it is realized that block, i.e., coupled, systems are not computationally expensive (in a relative sense).

Additionally a second type of approximation arises unconnected with linearization but arising from basic coupling terms in the original equations and if indeed some terms in an equation are time lagged in order to uncouple the equation system and these terms are of equal importance to the terms retained, then again an iterative updating is called for in order to achieve stability, accuracy and consistency. (This could be termed ad hoc equation uncoupling). Blottner (Ref. 9) has shown that many iterations around the ad hoc uncoupled set (>10) are sometimes required in order to achieve an overall solution accuracy commensurate with the local difference molecule accuracy. The linearization technique is described in Ref. 8, together with its application to block coupled splitting schemes. Schemes of this general type are here termed "split linearized block implicit" or split LBI schemes, and are reviewed in detail by Briley and McDonald (Ref. 2).

As a general observation, care is required to obtain acceptable transient accuracy for long time integration with conventional finite difference schemes. A Crank-Nicolson centered time implicit scheme for instance, although second order in time, shows quite a dispersion problem (relative to other schemes) on the simple pure convection problem. However, the problem of transient accuracy is significantly reduced in the typical boundary layer problem since the time dependency is continuously input through initial and boundary conditions and relatively the concern is with "short" time integrations. The computational problem is more of what the phase lag of the wall shear is, relative to the prescribed free stream disturbance, than concern over the convection velocity of a wave in a shear after a long propagation time. The interest is in forced oscillations with a minimum scale of the boundary layer thickness over a few cycles of the motion, just enough to obtain repetition cyclically. It is, therefore, expected that a significant dispersion problem will not arise with a conventional implicit scheme.

The governing equations that are considered here are the Navier-Stokes equations, continuity, energy and the equation of state which are written in generalized tensor form for a body oriented coordinate system (boundary layer coordinates). In accordance with the boundary layer assumptions, the normal momentum equation is eliminated and the pressure is specified throughout the viscous layer in its stead. For the energy equation constant stagnation temperature T_0 is assumed. This assumption is a good approximation for the flow fields considered, and is thus included here only for purposes of simplification. In the analysis that follows, the full energy equation could equally well have been used. Employing the equation of state which relates the pressure p to the velocity components u and w by an algebraic equation, the problem can be reduced to one involving only the three velocity components, u , w and v and three equations, the streamwise and spanwise momentum equations and the continuity equation. Hence, we consider a block-three system rather than a block-four system which leads to a significant reduction in computer time. If the full energy equation were to be considered, a block-four system would result due to the inclusion of the temperature as an additional unknown.

Coordinate System

Since the goal of this effort is to solve for the flow over airfoils an understanding of the type of geometries to be considered is essential to guide the choice of the coordinate system and the structure of the computer code.

Consider a typical finite span swept wing airfoil as shown in Fig. 1. The coordinate system is not only dependent upon the geometry of the airfoil but also upon the approximations that are made to the governing Navier-Stokes equations. As in boundary layer theory we also assume that in the approximate form of the Navier-Stokes equations the pressure is constant normal to the shear layer. Inherent in the assumptions is that the shear layer is thin. As pointed out by Howarth (Ref. 10) the boundary layer assumptions lead to the conditions that one coordinate direction must be normal to the body surface while the other coordinate directions must lie on the body surface. Furthermore, the coordinate lines normal to the surface are straight. These conditions uncouple the metric data on the surface from that in the normal direction. Hence the metric data for the surface coordinates are functions of the surface coordinates alone, while the metric data for the normal coordinate direction are functions of that coordinate alone.

The choice of the surface coordinates is rather arbitrary and is based on considerations such as the ease of construction or the grid distribution on the wing surface.

In the numerical solution of the flow over an airfoil there are many advantages to be gained by the judicious choice of coordinates. The most obvious advantage is that the physical boundaries of a flow region can be represented by coordinate surfaces. This removes the need for fractional cells in general; hence, the complications and loss of accuracy associated with a boundary interpolation are removed. Another advantage is that a uniform numerical method can be used. The solution can then be performed with a fixed number of cells in any given direction and with a uniform mesh spacing.

In Fig. 2 we can see the advantages of a nonorthogonal grid which conforms with the boundaries of the swept wing and covers the entire airfoil over a Cartesian grid which does not. In addition the coordinate transformation can be constructed to contain distributions for physical space mesh points. In this context, the uniform mesh of computational space is simply mapped into a suitably distributed mesh in physical space. The resolution of large solution gradients is the major objective in the selection of a coordinate mesh distribution, as in the resolution of an attached boundary layer. Another more subtle example is the resolution of large gradients in computational coordinates due to regions of high curvature on the bounding surfaces. When the transformation contains the mesh point distribution there is no need to construct the apparatus for the discrete approximation of derivatives on a nonuniform mesh. This results in a savings in both computer logic and storage.

Therefore, in this work a coordinate system is chosen that conforms with the boundaries of the physical domain i.e., the wing surface which in general will be nonorthogonal. In addition, in order to suitably distribute grid points in regions of large gradients, provisions are made for analytical grid transformations (Ref. 11) in each coordinate direction.

In view of the type of geometries to be considered and the assumptions made to obtain the approximate form of the Navier-Stokes equations a specialized nonorthogonal coordinate system is advocated where the metric tensor which has four independent components is given by

$$g_{ij} = \begin{bmatrix} g_{11} & g_{12} & 0 \\ g_{12} & g_{22} & 0 \\ 0 & 0 & g_{33} \end{bmatrix}$$

The subscripts 1 and 2 refer to the directions on the surface of the body while subscript 3 refers to the direction normal to the body. Furthermore, the metric data in the coordinate directions on the airfoil surface do not vary with the normal direction, i.e. the metric data in a 1 - 2 surface above the body are evaluated on the body surface (Ref. 10). Since we will be dealing with nonorthogonal coordinates it is advantageous to derive the equations in general nonorthogonal coordinates employing generalized tensors. In Appendix B a brief description of tensor notation is given. Further details can be found in Refs. 12 and 13.

An important feature of the analysis to follow is that the governing equations which are derived, under the prescribed assumptions, are invariant for any coordinate system or any grid transformation (although, of course, the physical approximations are coordinate dependent). The grid transformations are absorbed into the geometrical coefficients, leaving the equations unaltered in form. This point has a considerable effect on the development of the computer code. Since the only geometric information that must be input is the definition of the metric data and their derivatives, it can be contained in one subroutine without modifying the remainder of the code.

Governing Equations

In view of the ultimate goal of this program, to solve an approximate form of the unsteady three-dimensional Navier-Stokes equations on airfoil shapes, the governing equations are derived in general nonorthogonal coordinates and are given in generalized tensor notation. We will show that this notation aids in the ordering of the various terms in the equations and in many respects simplifies the construction of the computer code.

In the following derivation the governing equations are nondimensionalized as follows, x^i with respect to the characteristic length L , the velocity with respect to U_∞ , density, pressure and temperature with respect to ρ_∞ , $\rho_\infty U_\infty^2$ and U_∞^2/c_p respectively and time with respect to L/U_∞ . Viscosity is nondimensionalized with respect to μ_∞ .

Continuity Equation

Consider the continuity equation written in vector form so that it is independent of coordinate system i.e.,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \vec{q} = 0 \quad (1)$$

where ρ is the density and \vec{q} is the velocity vector. Expressing the velocity vector in a covariant basis

$$\vec{q} = u^i \vec{e}_i \quad (2)$$

where u^i is the i -th contravariant velocity component and \vec{e}_i is the covariant basis vector in the x^i direction. The velocity vector may be expressed in a number of different forms, each with certain attributes. Here for the moment the velocity vector is expressed in a covariant basis, for simplicity. Later the velocity components will be transformed into physical components for numerical solution. The contravariant basis exhibits variation in its components for instance in slug flow, if the coordinates are such that the metric varies. For boundary layer flows the physical velocity components are roughly aligned with the coordinates and exhibit no variation with the metric per se. As such, it is felt that the actual computations are better performed on the physical components.

The divergence of a vector (cf Appendix B) is given by

$$\nabla \cdot \rho \vec{q} = \rho u^k |_{,k} = (\rho u^k)_{,k} + \rho u^i \Gamma_{ik}^k = \frac{1}{J} (J \rho u^k)_{,k} \quad (3)$$

where $\rho u^k |_{,k}$ is the covariant derivative, $\rho u^k_{,k}$ is the partial derivative in the x^k direction, J is the Jacobian and Γ_{ik}^k is the Christoffel symbol (cf Appendix B).

In Equation 3 two forms of the divergence are presented, one involving the Christoffel symbol or curvature term directly and the other the Jacobian. The former is perhaps more restrictive since it requires additional smoothness of the geometrical quantities. However, herein we use either form solely from the point of convenience. For the continuity equation we use the form involving the Jacobian while in the momentum equations the form involving the Christoffel symbols is employed for the evaluation of the explicit (lagged) diffusion terms. Thus the form of the continuity equation which is used can be expressed as

$$\frac{\partial \rho}{\partial t} + \frac{1}{J} (J \rho u^k)_{,k} = 0 \quad (4)$$

Momentum Equations

The momentum equations in vector form can be written as

$$\rho \frac{\partial \bar{q}}{\partial t} = \rho \left[\frac{\partial \bar{q}}{\partial t} + (\bar{q} \cdot \nabla) \bar{q} \right] = \nabla \cdot \bar{\sigma} \quad (5)$$

where $\bar{\sigma}$ is the stress tensor.

In generalized tensor notation Equation 5 becomes

$$\rho \left[\frac{\partial u^i}{\partial t} + u^k u^i |_{|k} \right] \bar{e}_i = \sigma^{ik} |_{|k} \bar{e}_i \quad (6)$$

The stress tensor is defined as

$$\sigma^{ik} = - \left[p + \frac{2}{3} \frac{\mu}{Re} \Delta \right] \delta^{ik} + \frac{\mu}{Re} \epsilon^{ik} \quad (7)$$

where μ is the viscosity, p is the pressure, Δ is the velocity divergence and ϵ^{ik} is the strain tensor. The Reynolds number, Re , is defined as $\rho_{\infty} U_{\infty} L / \mu_{\infty}$. The strain tensor is defined as

$$\epsilon^{ik} = u^i |_{|m} g^{mk} + u^k |_{|m} g^{im} \quad (8)$$

where g^{mk} and g^{mi} are the components of the metric tensor. Employing the fact that $\delta^{kj} = g^{kj}$ and substituting the definition of the strain into the stress tensor we obtain

$$\sigma^{ik} = - \left(p + \frac{2}{3} \frac{\mu}{Re} \Delta \right) g^{ik} + \frac{\mu}{Re} u^i |_{|m} g^{mk} + \frac{\mu}{Re} u^k |_{|m} g^{mi} \quad (9)$$

Substituting Equation 9 into the momentum equation and employing the relationship

$$g^{kj} |_{|k} = 0 \quad (10)$$

we obtain for the i -th momentum equation in the \vec{e}_i direction

$$\rho \left[\frac{\partial u^i}{\partial t} + u^k u^i \Big|_k \right] = -g^{ik} \left(p + \frac{2}{3} \frac{\mu}{Re} \Delta \right) \Big|_k$$

$$+ g^{mk} \left[\frac{\mu}{Re} u^i \Big|_m \right] \Big|_k + g^{ml} \left[\frac{\mu}{Re} u^k \Big|_m \right] \Big|_k \quad (11)$$

In Ref. 3 it was pointed out that the QR Operator scheme requires that derivatives in any direction operate on only one variable. In the momentum equation this requirement prevents the implicit treatment of certain diffusion terms that arise due to the curvature of the body. Although these terms are often treated explicitly anyway the use of standard finite difference techniques instead of the QR Operators would give one the opportunity to treat these terms implicitly, if so desired. However, the use of the QR Operator scheme requires these terms be treated explicitly. This together with the quasi-linear form of the governing equations are the major limitations that arises in the treatment of the approximate form of the Navier-Stokes equations considered from the use of the QR Operator scheme. In the usual boundary layer approximations these explicitly treated terms would not appear in the equations since they are of order $O(Re^{-1/2})$ or smaller, and should, therefore, be of little consequence. In principle, the quasi-linear and, for instance, the full conservative form of the differential equations, are equivalent. In discrete form, various formulations of the governing equations exhibit different properties (Ref.29). In the present problem, no distinct disadvantage appears to arise from the required use of a quasi-linear form of the governing equations.

The requirement that derivatives in any direction operate only on one variable would be more restrictive in the treatment of the pressure gradient term in the full Navier-Stokes equations. The linearization of this term introduces derivatives of all the velocity components in a given direction. According to the limitations of the QR Operator scheme described above, some of these terms must be treated explicitly. Since an explicit treatment of these terms could reduce the stability bound of the calculation scheme, an alternate procedure should be considered. This would involve the addition of an auxiliary equation relating the pressure gradient term to the derivatives of the velocity components and would increase the block size of the system. An assessment to the efficiency of such a procedure has not been carried out and further work in this area would appear to be warranted.

In the discussion that follows, we will first split the governing equations into an explicit part and an implicit part in accordance with the QR Operator requirements. Thereafter, we will cast the resulting equations into "standard form", so that the equations can be appropriately linearized and treated with the LBI technique.

**ORIGINAL PAGE 19
OF POOR QUALITY**

Since mixed partial derivatives are commonly treated explicitly in orthogonal coordinate systems, we will do likewise in generalized nonorthogonal coordinates and extend this concept to include mixed second covariant derivatives. All other second covariant derivatives are retained as implicit. Although such a procedure would automatically treat more terms explicitly than one does for orthogonal coordinates, it simplifies the bookkeeping requirements in the construction of the computer code, and is thus adopted here. Furthermore, by not splitting up the covariant derivative for the purpose of making it implicit, but rather retaining it as a unit could prevent instabilities that may arise due to time splitting. This occurs when two portions of one term should cancel identically but cannot due to their being split between two sweeps.

Diffusion Terms

Consider the term

$$(\mu u^m |_{,j}) |_{,k} \quad (12)$$

If $j = k$ the term is retained as implicit, and if $j \neq k$ then it is lagged. We will consider the case $j \neq k$ first. Upon expanding the explicit part of the diffusion term it becomes

$$\begin{aligned} (\mu u^i |_{,j}) |_{,k} = & [\mu (u^i |_{,j} + u^n \Gamma^i_{jn})]_{,k} + \mu [u^m |_{,j} + u^n \Gamma^m_{jn}] \Gamma^i_{mn} \\ & - \mu [u^i |_{,m} + u^n \Gamma^i_{mn}] \Gamma^m_{jk} \end{aligned} \quad (13)$$

Note that the first term on the right-hand side of the equation is in conservation form. Although the implicit equations are treated in quasi-linear form, for the purpose of evaluating the explicit terms the most convenient representation is used. The implicit terms, with $j = k$ become (note there is no sum on j)

$$(\mu u^i |_{,j})_{,j} = \mu (u^i |_{,j})_{,j} + [(\mu u^n)_{,j} \Gamma^i_{jn} + \mu u^n |_{,j} \Gamma^i_{jn} - \mu u^i |_{,n} \Gamma^n_{jj}] + (\mu S^i_{jn}) u^n + T^i_{jj} \quad (14)$$

where

$$S^i_{jn} = \Gamma^m_{jn} \Gamma^i_{mj} - \Gamma^i_{mn} \Gamma^m_{jj} + \Gamma^i_{jn,j} \quad (15)$$

**ORIGINAL PAGE IS
OF POOR QUALITY**

and

$$T_{jj}^i = 2\mu [u^m_{,j} \Gamma_{mj}^i + u^m_{,j} \Gamma_{mj}^i] + \mu_{,j} [u^m \Gamma_{mj}^i + u^n \Gamma_{jn}^i] \quad (16)$$

$m=i+1$, $n=5-2i$ and no sum on m and n

Since T_{jj}^i involves velocity components and derivatives in directions other than the i -th direction, the term is also treated explicitly.

Hence the total diffusion terms for the i -th momentum equation is given in quasi-linear form as

$$\begin{aligned} & \sum_{j=1}^3 \left\{ [c_i^j g^{jj} \mu] u^i_{,jj} + [c_i^j g^{jj} (\mu_{,j} + 2\mu \Gamma_{jj}^i)] - \sum_{k=1}^3 c_i^k g^{kk} \mu \Gamma_{kk}^i \right\} u^i_{,j} \\ & + \left[\sum_{k=1}^3 c_i^k g^{kk} \mu s_{kj}^i \right] u^j + \delta_i^j \left[\sum_{k=1}^3 c_i^k g^{kk} \mu_{,k} \Gamma_{ki}^i \right] u^i \} \\ & + \sum_{j=1}^3 T_{jj}^i c_i^j g^{jj} + \sum_{k=1}^3 \left\{ \sum_{\substack{n=1 \\ k \neq n}}^3 g^{nk} [\mu u^i]_{,n} \right\} |_k + \sum_{\substack{n=1 \\ n \neq i}}^3 g^{ni} [\mu u^k]_{,n} |_k \} \end{aligned} \quad (17)$$

where

$$\delta_i^j = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (18)$$

and

$$c_i^j = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad (19)$$

and repeated indicies do not indicate summation. The last two summations can be combined into one explicit term, so that the diffusion term for the i -th momentum equation becomes

ORIGINAL PAGE IS
OF POOR QUALITY

$$\begin{aligned}
 & \sum_{j=1}^3 \left\{ [c_i^j g^{jj} \mu] u_{,jj}^j + [c_i^j g^{jj} (\mu_{,j} + 2\mu \Gamma_{jj}^j) - \sum_{k=1}^3 c_i^k g^{kk} \mu \Gamma_{kk}^j] u_{,j}^j \right. \\
 & \left. + \delta_i^j \left[\sum_{k=1}^3 c_i^k g^{kk} \mu_{,k} \Gamma_{ki}^j \right] u^j + \left[\sum_{k=1}^3 c_i^k g^{kk} \mu s_{kj}^j \right] u^j \right\} \\
 & + \sum_{k=1}^3 \left\{ c_i^k g^{kk} T_{kk}^j + \sum_{\substack{n=1 \\ k \neq n}}^3 g^{nk} [\mu u^j]_{,n} \right\} \Big|_k + \sum_{\substack{n=1 \\ n \neq i}}^3 g^{ni} [\mu u^k]_{,n} \Big|_k
 \end{aligned} \tag{20}$$

and there is no summation for repeated indices. Note that the diffusion operator terms for a given direction have been cast into standard quasi-linear form i.e.,

$$a u_{xx} + b u_x + c u + d$$

Convective Terms

The convective term for the i-th equation can be written as

$$\rho u^j u^i \Big|_j = \rho u^j [u_{,j}^i + u^m \Gamma_{mj}^i] \tag{21}$$

which equals when expanded

$$\rho u^j u^i \Big|_j = \sum_{j=1}^3 \left\{ \rho u^j u_{,j}^i + \sum_{m=1}^2 \rho u^j u^m \Gamma_{mj}^i + \delta_3^j \sum_{n=1}^3 \rho u^n u^j \Gamma_{nj}^i \right\} \tag{22}$$

and the last term is nonzero only when $j = 3$. The full momentum equation is obtained by substituting Equations (20) and (22) into Equation (11) and treating the pressure gradient and velocity divergence as explicit terms. Since the pressure is specified and impressed upon the viscous layer, its specification replaces the normal momentum equation. Thus, the streamwise and spanwise momentum equations are the only two retained.

ORIGINAL PAGE 19
OF POOR QUALITY

Energy Equation

The energy equation employed here states that the stagnation temperature is constant throughout

$$T_0 = T + \frac{1}{2} q^2 \quad (23)$$

The generalized tensor notation q^2 is given by

$$q^2 = u^i u^j g_{ij}$$

where u^i and u^j are the contravariant velocity components. Incorporating the assumptions made concerning the coordinate system we employ, i.e.

$$g_{13} = g_{23} = 0$$

we obtain

$$q^2 = (u^1)^2 g_{11} + 2u^1 u^2 g_{12} + (u^2)^2 g_{22} + (u^3)^2 g_{33}$$

Neglecting the term involving $(u^3)^2$ with respect to the other terms, and defining physical velocity components, i.e.

$$u_p = u^1 h_1$$

$$w_p = u^2 h_2$$

we obtain

$$T_0 = T + \frac{1}{2} (u_p^2 + w_p^2) + \frac{g_{12}}{h_1 h_2} u_p w_p \quad (24)$$

This is the form of the energy equation used.

Equation of State

The equation of state assumes a perfect gas and is given by

$$P = \frac{\gamma - 1}{\gamma} \rho T \quad (25)$$

Linearizations

The following analyses assume a set of linear partial differential equations. However, the convective part of the momentum equation and the continuity equation are nonlinear, containing terms that involve the product of density and velocity components. In order to overcome this difficulty we employ the linearization procedure (described in Ref. 8 and reviewed in Appendix A) to linearize the aforementioned terms by Taylor series expansion about the known time level solution.

The density is first eliminated by employing the equations of state and energy, and thereafter the resulting terms are linearized. These terms are of the following form

$$\begin{aligned}
 (\rho\psi\theta)^{n+\beta} &= (\rho^n\psi^n)\theta^{n+\beta} + (\rho^n\theta^n)\psi^{n+\beta} \\
 &+ \frac{\rho^n\psi^n\theta^n}{T^n} \left[(u^1)^n + \frac{g_{12}}{h_1 h_2} (u^2)^n \right] (u^1)^{n+\beta} \\
 &+ \frac{\rho^n\psi^n\theta^n}{T^n} \left[(u^2)^n + \frac{g_{12}}{h_1 h_2} (u^1)^n \right] (u^2)^{n+\beta} \\
 &+ \frac{\rho^n\psi^n\theta^n}{T^n} \left[2(T^n - T_0) + \left(\frac{\gamma}{\gamma-1} \right) \frac{p^{n+\beta}}{\rho^n} \right] - 2\rho^n\psi^n\theta^n
 \end{aligned} \tag{26}$$

where all velocity components are the contravariant ones, and θ is always a velocity component, (u^1, u^2, u^3) while ψ can be either a velocity component or a derivative of a velocity component. In the case of a term containing ρu^i we set $\psi^n = \psi^{n+\beta} = 1$.

It is important to note that in the preceding equations the contravariant velocity components are used. However, as noted in Ref. 14 it appears advantageous to solve for the physical velocity components. Therefore, when the governing equations are subsequently cast into a form amenable to the application of the LBI scheme, they are transformed so that the physical velocity components appear.

The Turbulence Model

We treat the set of three-dimensional ensemble averaged turbulent reduced Navier-Stokes equations. Ensemble averaging permits the appearance of low frequency (relative to the turbulence) time dependent "mean" flow. It is, therefore, necessary to specify a turbulence model suitable for this problem.

The approach taken in the present effort assumes an isotropic turbulent viscosity, μ_T , relating the Reynolds' stress tensor to mean flow gradients.

$$\text{Reynolds stress} = \bar{\sigma}_{\text{Rey}} = \frac{\mu_T}{\text{Re}} \left[\epsilon^i k - \frac{2}{3} \delta_j^i \Delta \right] \quad (27)$$

Using Favre averaging (Ref. 15) the governing equations then are identical to the laminar equations with velocity and density being taken as mean variables and viscosity being taken as the sum of the molecular viscosity, μ , and the turbulent viscosity, μ_T .

Spatial Difference Approximations

QR Operator Notation

In this section implicit tridiagonal finite difference approximations to the first and second derivatives and to the spatial differential operator will be considered. The very versatile QR Operator notation will be introduced, which allows as special cases a variety of schemes such as standard second order finite differences, first order upwind differences, fourth order operator compact implicit (OCI), fourth order generalized OCI and exponential type methods. Since all these schemes are of the same form, a single subroutine which defines the difference weights is all that is required to identify the method, while leaving the basic structure of the program unaltered. Subsequently, the results of numerical experiments employing some of these schemes will be presented. The rationale for the use of the QR approach in the present problem is discussed in detail in Ref. 3.

The QR formulation allows for ADI methods and permits the treatment of systems of coupled equations, i.e., LBI methods. Although variable mesh schemes can be employed within the QR framework, it is believed preferable to use analytic transformations to obtain a uniform computational mesh, hence attention is restricted to uniform mesh formulations.

The general concepts and notation for two point boundary value problems will be introduced and then the methodology extended to more general linear and nonlinear parabolic partial differential equations in one dimension. The extension to multi-dimensional problems will also be indicated.

Consider the two point boundary value problem

$$\tilde{L}(u) = \tilde{a}(x)u_{xx} + \tilde{b}(x)u_x + \tilde{c}(x)u = \tilde{f}(x) \quad (28)$$

with $u(0)$ and $u(1)$ prescribed. Derivative boundary conditions, although not described here, can easily be incorporated into the framework of the Q-R operation notation.

Let the domain be discretized so that $x_j = (j-1)h$, $j = 1, 2, \dots, J+1$, and $U_j \sim u(x_j)$, $F_j \sim u_x(x_j)$, $S_j \sim u_{xx}(x_j)$ and $h = 1/J$ is the mesh width. The numbering convention was chosen here to be compatible with FORTRAN coding.

Without loss in generality for $a(x) \neq 0$, Eq. (29) can be divided by $a(x)$ so that we may treat instead the following equation

$$\bar{L}(u) = u_{xx} + b(x)u_x + c(x)u = f(x) \quad (29)$$

where

$$b(x) = \tilde{b}(x)/\tilde{a}(x), \quad c(x) = \tilde{c}(x)/\tilde{a}(x) \quad \text{and} \quad f(x) = \tilde{f}(x)/\tilde{a}(x)$$

The spatial differential operator is identified as

$$L(u) = u_{xx} + b(x)u_x + c(x)u \quad (30)$$

Substituting the finite difference approximations to the first and second derivatives

$$\frac{D_0}{2h} U_j = \frac{U_{j+1} - U_{j-1}}{2h} = F_j = u_x(x_j) + O(h^2) \quad (31)$$

$$\frac{D_+ D_-}{h^2} U_j = \frac{U_{j-1} - 2U_j + U_{j+1}}{h^2} = S_j = u_{xx}(x_j) + O(h^2) \quad (32)$$

into Eq. (29) and rearranging, we obtain

$$L(u) \sim S_j + b_j F_j = \left[\frac{1}{h^2} - \frac{b_j}{2h} \right] U_{j-1} + \left[c_j - \frac{2}{h^2} \right] U_j + \left[\frac{1}{h^2} + \frac{b_j}{2h} \right] U_{j+1} = f_j$$

or

$$\left[1 - \frac{Rc_j}{2} \right] U_{j-1} + [h^2 c_j - 2] U_j + \left[1 + \frac{Rc_j}{2} \right] U_{j+1} = h^2 f_j \quad (33)$$

where $Rc_j = hb_j$ is the cell Reynolds number.

Equation (33) can be generalized by introducing operator format, i.e.,

$$r_j^- U_{j-1} + r_j^c U_j + r_j^+ U_{j+1} = h^2 (q_j^- f_{j-1} + q_j^c f_j + q_j^+ f_{j+1}) \quad (34)$$

where the superscripts (-) minus, (c) center, and (+) plus indicate the difference weight that multiplies the variable evaluated at the (j-1), (j) and (j+1) grid points, respectively, and where the r_j 's and q_j 's for grid point j are functions of h, b_{j-1} , b_j , b_{j+1} , c_{j-1} , c_j and c_{j+1} . Comparing Eqs. (33) and (34) we can identify the r_j 's and q_j 's, viz.,

$$\begin{aligned} r_j^- &= 1 - Rc_j / 2 & q_j^- &= 0 \\ r_j^c &= h^2 c_j - 2 & q_j^c &= 1 \\ r_j^+ &= 1 + Rc_j / 2 & q_j^+ &= 0 \end{aligned} \quad (35)$$

We now define the tridiagonal difference operators Q and R

$$\begin{aligned} R[U_j] &= r_j^- U_{j-1} + r_j^c U_j + r_j^+ U_{j+1} \\ Q[f_j] &= q_j^- f_{j-1} + q_j^c f_j + q_j^+ f_{j+1} \end{aligned} \quad (36)$$

Noting that $L(u) = f$ and substituting Eq. (36) into Eq. (15) we obtain

$$R[U_j] = h^2 Q[L(u)_j] = h^2 Q[f_j] \quad (37)$$

Alternatively by employing the inverse operator Q^{-1} an expression for $L(u)_j$ can be obtained

$$L(u)_j = \frac{1}{h^2} Q^{-1} R U_j \quad (38)$$

For standard central finite differences $Q = Q^{-1} = I$, the identity matrix, (the spatial operator is given explicitly in terms of U_{j-1} , U_j and U_{j+1}) so that nothing was gained in obtaining Eq. (38). However, in general, for higher order methods Q is tridiagonal and Q^{-1} is a full matrix. Hence Eq. (38) gives us a means of expressing the spatial operator for a wider class of difference approximations. The formalism in Eq. (38) is also applicable for first and second derivatives appearing alone (cf. Ref. 28). It should be pointed out, however, that Eq. (38) is not the most general formulation since the compact implicit formulas cannot be combined to yield a single scalar equation relating the spatial operator to the function values (Ref. 28).

In Refs. 3 and 16 a technique due to Berger, et al is described for constructing fourth order tridiagonal methods which possess a monotonicity property as the cell Reynolds number is increased, $R_c \rightarrow \infty$. We will not repeat it here. However, the resulting Q and R coefficients are given in Table II.

Another family of schemes that can be expressed in Q - R operator notation are the so-called exponential methods. The idea, originally due to Allen (Ref. 17) (independently derived by Il'in (Ref. 18) and McDonald (Ref. 19)) and employed by Dennis (Ref. 20), is to set the difference weights so that the numerical solution is equated to the analytic solution for the locally frozen constant coefficient equation. The Q and R coefficients of this exponential scheme is given in Table III. This method is second order accurate for $R_c=O(1)$ and becomes first order accurate as $R_c \rightarrow \infty$ where the scheme reverts to first order upwind differencing.

Another exponential scheme which is uniformly second order accurate was developed by El-Mistikawy and Werle (Refs. 24 and 25). The "exponential box scheme" which is incorporated in their solution of the boundary layer equations with strong blowing, is based on a spatial operator of the form given in Eq. (29). Berger, et al (Ref. 23) derived the counterpart for an operator of the form given in Eq. (30), but with $c = 0$. The Q and R coefficients are presented in Table IV. Although this scheme reverts to

second order upwind differences as $Rc \rightarrow \infty$, it does not possess a maximum principle analogous to the ordinary differential equation it is approximating as does the exponential scheme of Allen (Ref. 17). In Table V a second order central difference scheme is presented which adds artificial viscosity to the spatial operator when $|Rc| > 2$ so that $|Rc|$ never exceeds 2. This scheme is employed in the solution of several model problems and will be discussed in greater detail in the section on numerical results.

Application to Coupled Nonlinear Parabolic Equations

Before considering the LBI technique, we discuss some of the limitations placed on the Q-R operator scheme in solving a system of nonlinear parabolic equations.

Given a system of m nonlinear parabolic equations in m unknowns,

$$\sum_{i=1}^m \left\{ \frac{1}{a_{ij}^{n+\beta}} \frac{(u_{ij}^{n+1} - u_{ij}^n)}{\Delta t} - N_i^{n+\beta}(u_1, u_2, \dots, u_m, x_1, x_2, x_3, t) \right\} = 0$$

$$j = 1, 2, \dots, J+1$$

where $N_i^{n+\beta}$ is a quasilinear spatial operator, the Q-R formalism carries directly over provided that for any equation only one independent variable is operated upon by the differential operator. For example,

$$\frac{1}{a(u,w,v)} u_t = u_{xx} + b(u,v,w)u_x + c(u,v,w)$$

is allowed since x derivatives of u only appear, while

$$\frac{1}{a(u,w,v)} u_t = u_{xx} + b(u,v,w)u_x + c(u,v,w) + d(u,v,w)w_x$$

is not allowed since x derivatives of both u and w appear. The approximate form of unsteady Navier-Stokes equations used here, when written in quasi-linear form, falls within the class of allowable differential operators. Thus, for the problem being addressed in the present study the OCI schemes are applicable. Note that within the

splitting approach, non allowable terms in the OCI scheme such as dw_x above, may be split off and treated by a special implicit sweep. Provided care is taken and for instance the Douglas-Gunn formalism is achieved too, no particular problem arises other than the cost of an additional implicit sweep is incurred.

Thus multidimensional problems and/or more general equation forms can usually be accommodated by a splitting procedure, which reduces the differential operator to a sequence of one-dimensional problems which have the appropriate allowable form. However, as with standard finite differences, to avoid the cost of additional implicit sweeps special procedures must be applied to cross derivative terms, e.g., extrapolation or explicit treatment.

Linearized Block Implicit Scheme

Consider a system of nonlinear partial differential equations

$$A \bar{\Phi}_t = \mathcal{D} \bar{\Phi} + \bar{\Psi} \quad (39)$$

where $\bar{\Phi}$ is a vector of unknowns and $\bar{\Psi}$ is a source term vector which is a function of x^1, x^2, x^3 and t . Extension to source terms which are functions of $\bar{\Phi}$ are discussed in Ref. (8). \mathcal{D} is a three-dimensional nonlinear differential operator and of the matrix A acting on the momentum equations is equal to ρI where ρ is the density and I the unity matrix.

Equation (39) may be entered about $n+\beta$ time level, i.e. $t^{n+\beta} = (n+\beta)\Delta t = n\Delta t + \beta\Delta t = t^n + \beta\Delta t$, and written

$$A^{n+\beta} \left[\bar{\Phi}^{n+1} - \bar{\Phi}^n \right] / \Delta t = \mathcal{D}^{n+\beta} \bar{\Phi}^{n+\beta} + \bar{\Psi}^{n+\beta} \quad (40)$$

$0 \leq \beta \leq 1$ is a parameter allowing one to center the time step, i.e., $\beta = 0$ corresponds to a forward difference, $\beta = 1/2$ to Crank-Nicolson and $\beta = 1$ to a backward difference.

Equation (40) can be linearized by Taylor series expansion in time about the n^{th} time level by the procedure described in Ref. 8 to give a second order linearization

$$A^n \left[\bar{\Phi}^{n+1} - \bar{\Phi}^n \right] / \Delta t = \mathcal{L}^n \left[\bar{\Phi}^{n+\beta} - \bar{\Phi}^n \right] - \mathcal{D}^n \bar{\Phi}^n + \bar{\Psi}^{n+\beta} \quad (41)$$

where \mathcal{L} is the linearized differential operator obtained from \mathcal{D} by Taylor Series expansion in time.

ORIGINAL PAGE IS
OF POOR QUALITY

The difference between the nonlinear operator \mathcal{D} and the linear operator \mathcal{L} is defined as $M^n = \mathcal{D}^n - \mathcal{L}^n$. The intermediate level $n + \beta$ is defined as

$$\bar{\Phi}^{n+\beta} = \beta \bar{\Phi}^{n+1} + (1-\beta) \bar{\Phi}^n \quad (42)$$

Using these relationships and dropping the vector superbar for convenience a two-level hybrid implicit explicit scheme is obtained

$$A^n (\Phi^{n+1} - \Phi^n) / \Delta t = \beta \mathcal{L}^n (\Phi^{n+1} - \Phi^n) + \mathcal{L}^n \Phi^n + M^n \Phi^n + \Psi^{n+\beta} \quad (43)$$

The vector $\psi^{n+\beta}$ represents all of the terms in the system of equations which are treated explicitly. More about this will be said later, but for the moment note that $\psi^{n+\beta}$ may be approximated to the requisite order of accuracy by some multilevel linear explicit relationship, or approximated by ψ^n with a consequent order reduction in temporal accuracy.

The operator \mathcal{L} is now expressed as sum of convenient, easily invertible sub-operators $\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 + \dots + \mathcal{L}_m$. In the usual ADI framework these suboperators are associated with a specific coordinate direction. Further it is supposed that these suboperators are expressed in the QR notation introduced earlier. Writing $\psi^{n+\beta}$ and $M^n \Phi^n$ as a single source term $S^{n+\beta}$ the algorithm is written

$$A^n [\Phi^{n+1} - \Phi^n] / \Delta t = \beta [\mathcal{L}_1^n + \mathcal{L}_2^n + \mathcal{L}_3^n] [\Phi^{n+1} - \Phi^n] + [\mathcal{L}_1^n + \mathcal{L}_2^n + \mathcal{L}_3^n] \Phi^n + S^{n+\beta} \quad (44)$$

To solve this system efficiently it is split into a sequence of easily invertible operations following a generalization of the procedure of Douglas and Gunn (Ref. 28) in its natural extension to systems of partial differential equations. The Douglas-Gunn splitting of Eq. (44) can be written as the following three-step procedure

$$A^n [\Phi^* - \Phi^n] / \Delta t = \beta \mathcal{L}_1^n (\Phi^* - \Phi^n) + [\mathcal{L}_1^n + \mathcal{L}_2^n + \mathcal{L}_3^n] \Phi^n + S^{n+\beta} \quad (45)$$

$$A^n [\Phi^{**} - \Phi^n] / \Delta t = \beta \mathcal{L}_1^n [\Phi^* - \Phi^n] + \beta \mathcal{L}_2^n [\Phi^{**} - \Phi^n] + [\mathcal{L}_1^n + \mathcal{L}_2^n + \mathcal{L}_3^n] \Phi^n + S^{n+\beta}$$

$$A^n [\Phi^{***} - \Phi^n] / \Delta t = \beta \mathcal{L}_1^n [\Phi^* - \Phi^n] + \beta \mathcal{L}_2^n [\Phi^{**} - \Phi^n] + \beta \mathcal{L}_3^n [\Phi^{***} - \Phi^n] \\ + [\mathcal{L}_1^n + \mathcal{L}_2^n + \mathcal{L}_3^n] \Phi^n + S^{n+\beta}$$

which can be written in the alternative form

$$\begin{aligned}
 [A^n - \Delta t \beta \mathcal{L}_1^n] [\Phi^* - \Phi^n] &= \Delta t [\mathcal{L}_1^n + \mathcal{L}_2^n + \mathcal{L}_3^n] \Phi^n + \Delta t S^{n+\beta} \\
 [A^n - \Delta t \beta \mathcal{L}_2^n] [\Phi^{**} - \Phi^n] &= A^n [\Phi^* - \Phi^n] \\
 [A^n - \Delta t \beta \mathcal{L}_3^n] [\Phi^{***} - \Phi^n] &= A^n [\Phi^{**} - \Phi^n]
 \end{aligned} \tag{46}$$

if the intermediate levels are eliminated, the scheme can be written in the so-called factored form

$$\begin{aligned}
 (A^n - \beta \Delta t \mathcal{L}_1^n) A^{n-1} (A^n - \beta \Delta t \mathcal{L}_2^n) A^{n-1} (A^n - \beta \Delta t \mathcal{L}_3^n) (\Phi^{n+1} - \Phi^n) = \\
 \Delta t (\mathcal{L}_1^n + \mathcal{L}_2^n + \mathcal{L}_3^n) \Phi^n + \Delta t S^{n+\beta}
 \end{aligned} \tag{47}$$

At this point it becomes necessary to consider the structure of the operators \mathcal{L}_1 , \mathcal{L}_2 and \mathcal{L}_3 . It will be recalled from the one-dimensional scalar problem that use of the QR format greatly facilitated the introduction of a wide variety of spatial difference formulae. It follows that in the extension to multidimensions undertaken here it follows that use of the QR formulation results in the appearance of the inverse operator Q^{-1} with the sub-blocks of the \mathcal{L}_1 , \mathcal{L}_2 , \mathcal{L}_3 operators. In order to implement the scheme the inverse operator Q^{-1} must be cleared. Accordingly, the scalar operator Q is generalized to the vector operator \vec{Q}_i with (diagonal) sub-blocks Q_{ji} . In this generalization $j = 1, 2$ apply on the momentum equations and $j = 3$ applies on the continuity equation. The i subscript is associated with the coordinate directions of the \mathcal{L}_i operators. The discretization results in one diagonal sub-block for each grid point for each of the three \vec{Q}_i . Each intermediate step of the algorithm is now premultiplied by the \vec{Q}_i associated with the \mathcal{L}_i implicit operator. Writing the product operator $\vec{Q}_i \mathcal{L}_i$ as L_i , the inverse operators are thus removed and the scheme written, once again dropping the vector superscript for convenience

$$\begin{aligned}
 [Q_1 A^n - \Delta t \beta L_1^n] [\Phi^* - \Phi^n] &= \Delta t L_1 \Phi^n + \Delta t Q_1 [\mathcal{L}_2^n + \mathcal{L}_3^n] \Phi^n + \Delta t Q_1 S^{n+\beta} \\
 [Q_2 A^n - \Delta t \beta L_2^n] [\Phi^{**} - \Phi^n] &= Q_2 A^n [\Phi^* - \Phi^n] \\
 [Q_3 A^n - \Delta t \beta L_3^n] [\Phi^{***} - \Phi^n] &= Q_3 A^n [\Phi^{**} - \Phi^n] \\
 \mathcal{L}_2^n \Phi^n &= Q_2^{-1} R_2 \Phi^n & \mathcal{L}_3^n \Phi^n &= Q_3^{-1} R_3 \Phi^n \\
 \Phi^{n+1} &= \Phi^{***} + O(\Delta t^3)
 \end{aligned} \tag{48}$$

With the removal of the inverse operator Q^{-1} , the question of the proper intermediate level solution boundary conditions can be addressed. As is pointed out by Briley and McDonald (Ref. 2), the proper intermediate level boundary conditions may be derived by running through the intermediate steps in reverse order. Defining a boundary condition operator B_1^n after linearizing the appropriate physical boundary condition by Taylor series expansion in time as

$$B_1^n(\Phi^{n+1}) = g(t, \Phi^n)$$

and applying this operator to the algorithm defines the boundary conditions as

$$\begin{aligned} B_3^n Q_3 A^n [\Phi^{***} - \Phi^n] &= [B_3^n Q_3 A^n - \Delta t \beta B_3^n L_3^n] [\Phi^{***} - \Phi^n] \\ B_2^n Q_2 A^n [\Phi^* - \Phi^n] &= [B_2^n Q_2 A^n - \Delta t \beta B_2^n L_2^n] [\Phi^{***} - \Phi^n] \end{aligned} \quad (49)$$

and note that unless $B_1^n L_1^n$ commute (an unlikely event except with Dirichlet boundary conditions, where $B_1^n = I$) the exact boundary conditions cannot be derived. A number of possible strategies are possible at this point aimed at various levels of approximation to $B_1^n L_1^n$. For the present the term $\Delta t \beta B_1^n L_1^n [\phi - \phi^n]$ is neglected. This introduces an error of order $O[\Delta t(\phi - \phi^n)]$ into the solution but note that this error disappears at steady state where $\phi^{***} = \phi^{**} = \phi^*$. Neglect of the $\Delta t \beta B_1^n L_1^n [\phi - \phi^n]$ term is of course equivalent to applying the physical boundary conditions on the intermediate level variables.

This completes the general derivation of the algorithm and attention is now given to the specific forms of the L_1^n operators including the rather special form of the component operator for the continuity equation.

It is worth noting that the operator \mathcal{D} or \mathcal{L} can be split into any number of components which need not be associated with a particular coordinate direction. As pointed out by Douglas and Gunn (Ref. 28), the criterion for identifying sub-operators is that the associated matrices be "easily solved" (i.e., narrow-banded). Thus, mixed derivatives and the complicating terms which might inhibit the use of OCI can be treated implicitly within such a framework, although this would increase the number of intermediate steps and thereby complicate the solution procedure.

An inspection of Eq. (48) reveals that only the linearized operators L_1^n , L_2^n and L_3^n appear. Indeed, the computer code employs this feature by evaluating these three operators before the first sweep, storing them and accessing them as needed in the

subsequent three sweeps. In addition, the terms arising from the nonlinear terms are immediately absorbed into $S^{n+\beta}$ as they appear, allowing for an efficient evaluation of the terms in the differential equation

The operators $\mathcal{L}_1^n, \mathcal{L}_2^n, \mathcal{L}_3^n$ can be represented in standard form at each grid point, i.e.

$$\mathcal{L}_1^n \Phi_1^n = a_{11}^n \Phi_{1,1} + a_{12}^n \Phi_{1,1} + a_{13}^n \Phi_1 + a_{14}^n \Phi_2 + a_{15}^n \Phi_3 \quad (50)$$

In Eq. (50) the subscripts of Φ indicate the velocity component (associated with the corresponding direction and " , " indicates a derivative. The subscripts of the a_{ij}^n refer to the direction (i) and the term in the equation (j) respectively. Note that the equation is in quasi-linear form, a necessity of the QR operator technique employed here since the coefficients of the derivative operators need to be identified. Alternate schemes have been proposed by Leventhal (Ref. 33) for equations in conservation form but are not considered here. In the following section we will describe how this entire operator is discretized by employing the QR operator format, and show how the discretization is incorporated into the LBI framework in order to solve the system of equations (48).

We consider first, however, the continuity equation. Since it is a first order partial differential equation it does not have the standard form of Eq. (49). Furthermore, ρ has been linearized and eliminated in favor of the u^i velocity components so that the continuity equation has become an equation for the three velocity components, and not density.

As pointed out by McDonald and Briley (8) skillful partitioning of the resulting matrix can lead to significant decreases in computation time. An inspection of the system of equations under consideration reveals that substantial savings can be realized if the equations are partitioned appropriately. Due to the use of a boundary layer coordinate system, in the two momentum equations the normal velocity appears only in conjunction with terms associated with the normal "3" direction. Hence, in the first two sweeps, for the streamwise and spanwise momentum equation one is required to solve only for the two corresponding velocity components without the need of considering the continuity equation. However, on the third sweep where all 3 velocity components appear, one must solve all 3 equations. This strategy reduces the solution procedure to the inversion of two 2×2 block matrices and one 3×3 block matrix rather than three 3×3 block matrices which accounts for the reduction in computation time.

If we were to consider the full Navier-Stokes equations which include a normal momentum equation the aforementioned partitioning could not be applied since the normal velocity would appear in all three sweeps.

**ORIGINAL PAGE IS
OF POOR QUALITY**

The question that arises is how to appropriately split the continuity equation, since it need only be solved on the third sweep. Here again the Douglas-Gunn formulation gives us the answer. The continuity equation written in conservation form becomes,

$$\frac{\partial \rho}{\partial t} + \frac{1}{J} \frac{\partial}{\partial x^i} [J \rho u^i] = 0 \quad (51)$$

linearizing ρ we obtain in increment form

$$\begin{aligned} & A^n \Delta u^{n+1} + B^n \Delta w^{n+1} + \frac{\Delta t \beta}{J} \frac{\partial}{\partial x^3} [v^n A^n \Delta u^{n+1} + v^n B^n \Delta w^{n+1} + \rho^n \Delta v^{n+1}] \\ &= - \frac{\Delta t}{J} \frac{\partial}{\partial x^i} [J \rho u^i]^n + \frac{\Delta t \beta}{J} \frac{\partial}{\partial x^i} [(\rho^n + u^n A^n) \Delta u^{n+1} + (u^n B^n) \Delta w^{n+1}] \\ & \quad + \frac{\Delta t \beta}{J} \frac{\partial}{\partial x^2} [(\rho^n + w^n B^n) \Delta w^{n+1} + (w^n A^n) \Delta u^{n+1}] \end{aligned} \quad (52)$$

where all the velocity components are the contravariant components $u = u^1$, $w = u^2$ and $v = u^3$. J is the Jacobian and

$$A^n = \frac{\rho^n}{\tau^n} [g_{11} u^n + g_{12} w^n]$$

$$B^n = \frac{\rho^n}{\tau^n} [g_{22} w^n + g_{12} u^n]$$

Approximating equation (52) by employing the Douglas-Gunn procedure as a third sweep equation, we obtain a consistent approximation to the continuity equation, i.e., the x^1 derivative term is evaluated at the * level and the x^2 derivative term is evaluated at the ** level. The values of the intermediate derivative terms are obtained after the solution of the first two sweeps of the two momentum equations. Note that these terms do not contain the normal velocity. We can thus write the equation in symbolic form

$$\begin{aligned}
 & A^n \Delta u^{n+1} + B^n \Delta w^{n+1} + \frac{\Delta t \beta}{J} \frac{\partial}{\partial x^3} \left[J \left\{ A^n v^n \Delta u^{n+1} + v^n B^n \Delta w^{n+1} + \rho^n \Delta v^{n+1} \right\} \right] \\
 & = s^n - \beta \frac{\Delta t}{J} \frac{\partial}{\partial x^1} \left[J \{ \} \right]^* - \beta \frac{\Delta t}{J} \frac{\partial}{\partial x^2} \left[J \{ \} \right]** \quad (53)
 \end{aligned}$$

Since the only term involving v is in the x^3 derivative term, we can directly integrate the equation with respect to x^3 , i.e.

$$\begin{aligned}
 & \int_{x^3} \left[A^n \Delta u^{n+1} + B^n \Delta w^{n+1} \right] dx^3 + \Delta t \frac{\beta \Delta t}{J} \left[v^n A^n \Delta u^{n+1} + v^n B^n \Delta w^{n+1} + \rho^n \Delta v^{n+1} \right] \\
 & = \int_{x^3} \left\{ s^n - \frac{\beta \Delta t}{J} \left[\right]^* - \frac{\beta \Delta t}{J} \left[\right]** \right\} dx^3 \quad (54)
 \end{aligned}$$

In the next section we describe how this is done very easily via the Q-R operator scheme. The concept of integrating directly the continuity equation is not new. Davis (Ref. 34) in his coupled procedure for the solution of two-dimensional steady boundary layer equation used a trapezoidal rule to integrate the continuity equation. Weinberg (Ref. 35) (Ref. 36) also used a fourth order Simpson integration scheme to solve the compressible boundary layer equations. Such procedures are stable and offer a viable alternative to approximating the derivatives by finite differences. Note that conceptually the continuity equation in integrated form is treated on each sweep of the Douglas-Gunn splitting, although in actuality this can be viewed as having the same form as each sweep and the integration operator can be incorporated into the \mathcal{L} and \mathcal{D} difference operators, and as a result the stability and consistency of the original splitting is retained.

Implementation of the LBI Scheme Employing the QR Operator Technique

Consider the third sweep of Eq. (47) in which we solve both momentum equations and the continuity equation. The momentum equations are in the form

$$\left[A^n - \beta \Delta t \mathcal{L}_3^n \right] \Delta \Phi^{***} = A^n \Delta \Phi^{**} \quad (55)$$

where $\Delta\phi^{***}$ is the column vector of unknowns, u , v , w . Here we have implicitly assumed that the equations have been appropriately normalized and that the contravariant velocity components have been suitably transformed into their physical components. Employing physical components, (cf. Ref. 14) leads to a better behaved solution since these components are not unduly influenced by geometrical variations.

For the streamwise momentum equation we obtain

$$\mathcal{L}_3^n \Delta\Phi^{***} = \Delta u_{,33} + a_{23} \Delta u_{,3} + a_{33} \Delta u + a_{43} \Delta w + a_{53} \Delta v \quad (55a)$$

while for the spanwise momentum equation we obtain

$$\mathcal{L}_3^n \Delta\Phi^{***} = \Delta w_{,33} + b_{23} \Delta w_{,3} + b_{33} \Delta w + b_{43} \Delta v + b_{53} \Delta u \quad (55b)$$

where we have omitted superscript *** from Δu , Δv and Δw . Now in equation 55a, we can approximate

$$\Delta u_{,33} + a_{23} \Delta u_{,3} + a_{33} \Delta u \quad \text{by} \quad \frac{Q_1^{-1} R_1}{\Delta x_3^2} \Delta u \quad (56)$$

the operator equivalent, so that

$$\mathcal{L}_3^n \Delta\Phi^{***} = \frac{Q_1^{-1} R_1 \Delta u}{\Delta x_3^2} + a_{43} \Delta w + a_{53} \Delta v \quad (57)$$

Similar approximations are made for Equation (55b). After substituting Equation (56) into Equation (54) and multiplying thru by Q we obtain for the streamwise momentum equation

$$[Q_1 \rho^n - \beta \lambda R_1] \Delta u - \beta \Delta t Q_1 a_{43} \Delta w - \beta \Delta t Q_1 a_{53} \Delta v = Q_1 \rho^n \Delta u^{**} \quad (58)$$

where $\lambda = \Delta t / \Delta x_3^2$

Similarly for the spanwise momentum equation we obtain

$$\left[Q_2 \rho^n - \beta \lambda R_2 \right] \Delta w - \beta \Delta t Q_2 b_{243} \Delta v - \beta \Delta t Q_2 b_{253} \Delta u = Q_2 \rho^n \Delta w^{**} \quad (59)$$

We employ the same type of procedure for the continuity equation. Since the continuity equation involves only first derivatives, so that they can be represented as

$$\frac{\partial}{\partial x^3} = \frac{Q_c^{-1} R_c}{\Delta x_3} \quad (60)$$

The operators Q_c and R_c are constructed to approximate the weights associated with either a second order trapezoidal rule or a fourth order Simpson's rule, i.e.

Trapezoidal rule

$$q_c^- = 0, \quad q_c^c = \frac{1}{2}, \quad q_c^+ = \frac{1}{2}$$

$$r_c^- = 0, \quad r_c^c = -1, \quad r_c^+ = 1$$

Simpson's rule

$$q_c^- = \frac{1}{3}, \quad q_c^c = \frac{4}{3}, \quad q_c^+ = \frac{1}{3}$$

$$r_c^- = -1, \quad r_c^c = 0, \quad r_c^+ = 1$$

The continuity equation thus becomes

$$JA^n \Delta u + JB^n \Delta w + \frac{\Delta t \beta}{\Delta x_3} Q_c^{-1} R_c \left[JA^n v^n \Delta u + JB^n v^n \Delta w + J\rho^n \Delta v \right] = \text{RHS} \quad (61)$$

ORIGINAL PAGE IS
OF POOR QUALITY

where RHS contains all the terms due to the linearization procedure and the terms evaluated at the * and ** levels. Multiplying thru by Q_c and setting $\omega = \Delta t / \Delta x_3$ we obtain

$$\left[Q_c J A^n + \beta \omega R_c J A^n v^n \right] \Delta u + \left[Q_c J B^n + \beta \omega R_c J B^n v^n \right] \Delta w + \left[\beta \omega R_c J \rho^n \right] \Delta v = Q_c (RHS) \quad (62)$$

The resulting matrix derived from Equations 57, 58 and 61 becomes a block 3 tridiagonal matrix (Q and R are tridiagonal operators) with each sub block taking on the form

$$\begin{bmatrix} \left[\begin{array}{ccc} Q_1 \rho^n - \beta \lambda R_1 & & \\ & -\beta \Delta t Q_1 a_{43} & \\ & & -\beta \Delta t Q_1 a_{53} \end{array} \right] & \begin{bmatrix} \Delta u \\ \\ \end{bmatrix} & \begin{bmatrix} Q_1(\Delta u^{**}) \\ \\ \end{bmatrix} \\ \left[\begin{array}{ccc} -\beta \Delta t Q_2 b_{53} & & \\ & Q_2 \rho^n - \beta \lambda R_2 & \\ & & -\beta \Delta t Q_2 b_{43} \end{array} \right] & \begin{bmatrix} \\ \Delta w \\ \end{bmatrix} & = \begin{bmatrix} \\ Q_2(\Delta w^{**}) \\ \end{bmatrix} \\ \left[Q_c J A^n + \beta \omega R_c J A^n v^n \right] & \left[Q_c J B^n + \beta \omega R_c J B^n v^n \right] & \left[\beta \omega R_c J \rho^n \right] & \begin{bmatrix} \\ \\ \Delta v \end{bmatrix} & \begin{bmatrix} \\ \\ Q_c (RHS) \end{bmatrix} \end{bmatrix}$$

This matrix is inverted by standard LU decomposition.

Boundary Conditions and Initial Conditions

The type of boundary conditions employed in the solution of the approximate form of the Navier-Stokes equations are described in this section. On the body surface no slip is prescribed for all the velocity components. At the outer edge of the viscous layer the values of streamwise and spanwise velocity components are also prescribed. However, the value of the normal velocity component is not set, but rather computed as part of the numerical solution as is the practice in standard boundary layer procedures.

At the inflow boundaries, (upstream) velocity profiles are fixed, while extrapolation conditions are employed at the outflow boundaries (downstream). Further discussion of this matter is given in the section of numerical results.

The intermediate boundary conditions employed on the first two sweeps are the physical ones. Since all the multidimensional problems considered here to date have been steady, there has not yet been any need for a high temporal accuracy solution, and the imposition of physical intermediate boundary conditions did not impair the quality of the solutions obtained. These results are in keeping with the analysis of McDonald and Briley (Ref. 2) for second order spatial schemes. (The type of schemes used to date in the present study). Fourth order methods have not yet been applied to any of the test problems in this report.

The question of proper intermediate boundary conditions for fourth order methods until recently has not been resolved. Fairweather and Mitchell (Ref. 37) developed nonphysical intermediate boundary conditions for a fourth order solution of Laplace's equation, and showed that, in general, the use of noncorrected i.e. physical boundary conditions leads to a loss in steady state accuracy for their method. As pointed out by Fairweather and Mitchell (Ref. 37) their scheme is inconsistent. It is this inconsistency that requires one to use appropriately derived intermediate boundary conditions in order to recover a steady state solution independent of time. However, if a consistent scheme were to be used, e.g. Douglas-Gunn, then physical boundary conditions can be applied without any loss in steady state accuracy. A more complete discussion of the implementation of ADI schemes with the appropriate intermediate boundary conditions for QR operator schemes (including the fourth-order generalized OCI scheme) will appear in a forthcoming report. These conclusions generalize the results obtained by Briley and McDonald (Ref. 2) for second order finite difference methods to higher order schemes and to those schemes that can be cast into a QR operator framework.

The Computer Code

The type of numerical algorithm employed as well as its formulation has a marked impact on the structure of the computer code. We need to consider both the number of CPU operations as well as the memory requirements. Usually the number of operations can be reduced at the expense of increasing the amount of storage. However, for three-dimensional problems the accessible fast (small core) memory becomes a severe limitation even without attempting to optimize the operation count.

The storage requirements for the solution of the approximate form of the three-dimensional Navier-Stokes equations for even modest size grids (e.g. $30 \times 30 \times 30$) exceed the available small core memory of a machine like the CDC 7600. One must then resort to either mass storage devices such as disks or slow access memory (large core).

In using such devices both access time and transfer rates must be considered. When small amounts of data are being transferred frequently (what we are considering) then access time becomes a significant factor. Therefore, a combination of strategies must be employed in order to optimize both access time and transfer rate. As a result of these necessary manipulations, the resulting code was far more complex than the one that would have been developed specifically for a machine with "unlimited" storage resources.

An investigation of the operation count of the LBI scheme in conjunction with the QR Operator technique leads to the conclusion that the most significant fraction of time is spent in computing the matrix coefficients, i.e. the linearization coefficients and difference weights. This amount far exceeds the time required for the matrix inversion. Hence it is worthwhile to optimize the calculation of these coefficients and if possible store their values. This procedure was accomplished by storing the operator coefficients on L_2^n and L_3^n as they were computed in the first sweep on the right-hand side of the differential equation. On the second and third sweeps L_2^n and L_3^n were accessed respectively and were not recomputed. It was for this reason that the formulation of the LBI scheme referred to linearized operators L_i^n 's instead of D_i^n 's on the right-hand side of the equation.

The general structure of the computer code will not be described. After the input section and the initialization of data e.g. geometry, grid transformations, flowfield, etc. the actual construction of the difference operators is begun. The first derivatives of the velocity components and viscosity are obtained for the entire

flow field and stored for ready access when needed for the computation of the appropriate terms in the governing equations. Thereafter the terms that are to be lagged (treated explicitly) are evaluated and absorbed into the function S^n .

The operators L_1 , L_2 and L_3 are then computed. These are used to evaluate the appropriate Q and R coefficients which are then stored for easy retrieval during each of the ADI sweeps.

In the first sweep the matrix resulting from the application of the L_1 operators for the streamwise and spanwise momentum equation is solved as a 2 x 2 coupled system. The solution of this system, the * level quantities, are then used to construct the right-hand side of the second sweep equations and to evaluate the appropriate * level term in the continuity equation. At this point the L_2 operator is accessed and again a 2 x 2 system of equations for the streamwise and spanwise momentum equation is solved. The ** level quantities are then used to construct the right-hand side of the third sweep equations as well as the appropriate terms in the continuity equation. The third sweep equations consist of the two momentum equations and the continuity equation, with the L_3 operator being accessed from memory. The resulting 3 x 3 system of equations are solved for the three velocity components.

After the ADI procedure is completed, the thermodynamic quantities, density, temperature and viscosity are computed. The procedure is then repeated at the following time step.

It is noteworthy that the scheme just described operates on vectors, i.e. lines of data. Therefore, it could show promise for vectorized machines.

NUMERICAL RESULTS

In this section we describe the numerical results obtained by using the computer code described in the previous sections on several Cartesian test problems. The goal of these calculations is to validate the computer code. No specific timing experiments have yet been conducted to evaluate the code's efficiency. As mentioned in the previous section, the code was structured for the calculation of three-dimensional problems so that the disk writing and large core data allocation are necessary to run the code efficiently in that mode. For two-dimensional problems these artifices, although not necessary except for very large mesh problems, are still employed. Hence as an observation, it is noted that the code is not as efficient in the two-dimensional mode as it could be.

Since the primary goal in this portion of the effort was to obtain a working computer code, the following test cases are considered:

1. One-dimensional unsteady oscillating flat plate;
2. Three-dimensional boundary layer without pressure gradient; and
3. Two-dimensional steady boundary layer with and without pressure gradient.

These cases check out three basic features of the code, viz., time dependent behavior, pressure gradient effects and three-dimensional effects.

All the calculations employ second order finite difference techniques. Although a fourth order generalized OCI option is incorporated into the code (and the code is structured particularly to encompass that scheme), to date none of the cases considered were run employing this scheme. In subsequent work fourth order calculations will be compared to second order finite difference as they relate to efficiency gains and convergence rates.

Oscillating Flat Plate

The first case considered is that of an oscillating flat plate in a uniform stream. It is a one-dimensional unsteady problem and is thus only a function of the normal distance to the wall and time.

Schlichting (Ref. 38) gives the exact solution to this problem, which he terms 'Stokes second problem', as

$$u(\eta, t) = U_{\infty} e^{-\eta} \cos(\omega t - \eta)$$

where $\eta = y \sqrt{\frac{\omega}{2\nu}}$ and $\omega =$ the frequency of oscillation.

The boundary conditions are
at the wall

$$\eta = 0 \quad u(0,t) = U_{\infty} \cos \omega t$$
$$v(0,t) = 0$$

at the outer edge

$$\eta \rightarrow \infty \quad u(0,t) = 0$$

Since we must specify the velocity at a finite value of η we set u at $\eta = \eta_e$ to the exact value of $u(\eta_e, t)$. As initial conditions, the velocity field is set to the exact value at $t = 0$. Since the solution does not take into account any initial conditions of the fluid, the relationship for $u(\eta, t)$ represents a "steady state solution".

The computer is run in the two-dimensional mode, with 5 grid points in the streamwise direction, so that additional boundary conditions are required at the inflow and exit planes. This is accomplished by setting the first derivative of the velocity components to zero there, which eliminates any variation in the streamwise direction.

For the problem considered, the following parameters are employed.

$$\omega = 100 \text{ rad/sec}$$

$$Re = 100$$

$$\eta_e = 5.0$$

A uniformly distributed grid of 11 points in the normal direction is used as well as a fixed time step of $\omega \Delta t = 5^\circ$. Second order central differences are employed for the calculation which includes all the viscous terms appearing in the governing equation. Since there is no variation in the streamwise direction and the normal velocity is zero throughout, the governing equations reduce computationally to their one-dimensional counterpart. In Fig. 3 the computed skin friction coefficient is compared to the analytical result over one and one-half time cycles. The analytical value is

$$C_f = -\frac{2}{Re} \sqrt{\frac{\omega}{\nu}} \sin\left(\frac{\pi}{4} - \omega t\right)$$

Even with a relatively coarse mesh there is good agreement between theory and the numerical results. Although the Crank-Nicolson results ($\beta = 1/2$) are more accurate than the fully implicit calculation ($\beta = 1$) as can be seen in Fig. 4 where the L_2 error of each calculation is compared, the skin friction results appear much closer. We can account for the discrepancy by looking at the evaluation of the skin friction coefficient.

The error in the skin friction coefficient is made up in part from the error in the solution of the differential equation and in part from the approximation of the first derivative in the computation of the shear. The error in the evaluation of the first derivative is given as

$$\text{error} \left(u_\eta \Big|_{\eta=0} \right) = \alpha_1 u_{\eta\eta\eta} = \alpha_2 \left(\sqrt{\frac{\omega}{2\nu}} \right)^3$$

where α_1 and α_2 are bounded constants

For our case $\frac{\omega}{2\nu} = 50$ so that the total truncation error is dominated by this term.

The results of this calculation indicate that accurate time dependent calculations can be obtained.

Three-Dimensional Flat Plate Boundary Layer

The second problem considered is the three-dimensional viscous flow over a flat plate skewed at an angle of 45° to the flow direction. In Fig. 5 the computational domain is shown in which the solution is obtained. An equally spaced mesh of 11 points in each of the three directions is employed.

Two cases are considered. In the first case boundary layer assumptions are made, and the only viscous terms retained in the momentum equation are those that appear in the standard three-dimensional boundary layer equations. Diffusion terms in the streamwise and spanwise directions are omitted. In addition, first order approximations to the streamwise and spanwise first derivatives (in the marching directions)

are made. The second case considered retains all the viscous terms. The derivative approximation to the spatial operators employed entered differences with artificial viscosity (cf Table V).

The boundary conditions applied are no slip at the surface and velocity specified at the outer edge of the viscous layer. There the velocity components are set to $1/\sqrt{2}$. Note that the vector sum of the two components is 1. At the inflow boundaries 1 and 2 (cf Fig. 5) the velocity profiles are set while at the outflow boundaries 3 and 4 second derivative extrapolation conditions are employed.

Both cases compared well with one another and to the Blasius solution. In Fig. 6 the displacement thickness δ^* and the momentum thickness θ along the diagonal of the computational domain are compared to the theoretical result. Although symmetry was retained across the diagonal, at a constant ξ value, there was some variation of the integrated properties as a function of ξ from their values on the diagonal.

Two-Dimensional Howarth Flow

The third case considered is a two-dimensional steady flow problem; the laminar boundary layer on flat plate in the presence of an adverse pressure gradient, the Howarth flow. Here again we consider two cases. The first employs the boundary layer approximations while the second retains all the terms in the streamwise momentum equations.

A 21 x 21 mesh is used in both cases. The boundary conditions employed are no slip at the wall, streamwise velocity specified at the outer edge, velocity profile specified at the inflow plane and second derivative extrapolation at the downstream plane.

The external velocity field is linearly retarded, i.e. $u_e = 1 - Kx$, so that there is an adverse pressure gradient. Due to the adverse pressure gradient separation will occur at some point downstream. Howarth (Ref. 39) computed the separation point at a value of $x^* = Kx = .1200$. In the calculations considered here we choose our domain to span $.05 \leq x^* \leq .12$ which terminates at a location corresponding to Howarth's predicted separation point.

The initial conditions for the problem were constructed from the local Falkner Skan similarity solution with the appropriate pressure gradient parameter imposed at the corresponding streamwise location.

In Fig. 7 the computed skin friction coefficients for both cases are compared to the results of Howarth. The case considered corresponds to Briley's cases (Ref. 40), i.e. $Re = 62,500$ and $k = 3$. There is good agreement over most of the domain, with an overprediction of the shear as the separation point is approached. This discrepancy is not surprising due to the coarse mesh employed. It is interesting that the "boundary layer" case gives somewhat better predictions near the separation point than the "full equation" case. This is probably due to the lack of upstream influence in the boundary layer solution which did not allow the flow to adjust upstream of separation. Both calculations converged within 30 - 40 iterations, with a maximum change between two (large) time steps being less than $5 \cdot 10^{-4}$.

CONCLUSIONS

In this report a computer code is described that can be applied to the solution of an approximate form of the time-dependent Navier-Stokes equations over airfoil sections. The governing equations are more general than the conventional boundary layer equations, notably in the inclusion of streamwise and spanwise diffusion terms, although the pressure is still imposed by the external flow, as in conventional boundary layer theory. The computer code which treats the governing equations incorporates the split LBI scheme in conjunction with QR operator scheme that permits a variety of spatial difference schemes, including standard second order finite differences, exponential type methods and fourth order OCI techniques. In the split LBI scheme, an implicit sweep is performed in each spatial coordinate direction. A careful ordering of these sweeps permits an uncoupling of the continuity equation from the system in the first two implicit sweeps. Thus on the first two sweeps the (tridiagonal) system block size is reduced from 3×3 to 2×2 with a resulting cost savings. On the last sweep of each time step all the equations in the system are linearly coupled and 3×3 blocks must be eliminated. Results of several Cartesian problems employing second order finite differences indicated that the proposed method is viable and has application to more complex flows. Future efforts will aim at exercising the fourth order OCI schemes option, treating actual airfoil sections and considering turbulent flow. Comparisons will also be made among the various types of spatial schemes to assess the overall efficiency gains made by employing higher order methods. The treatment of inflow and outflow boundary conditions will also be of primary interest.

APPENDIX A

Linearization Technique

A number of techniques have been used for implicit solution of the following first-order nonlinear scalar equation in one dependent variable $\phi(x,t)$:

$$\partial\phi/\partial t = F(\phi) \partial G(\phi)/\partial x \quad (A1)$$

Special cases of Eq. (A1) include the conservation form if $F(\phi) = 1$, and quasi-linear flow if $G(\phi) = \phi$. Previous implicit methods for Eq. (A1) which employ nonlinear difference equations and also methods based on two-step predictor-corrector schemes are discussed by Ames (Ref. 41, p. 82) and von Rosenberg (Ref. 42), p. 56). One such method is to difference nonlinear terms directly at the implicit time level to obtain nonlinear implicit difference equations; these are then solved iteratively by a procedure such as Newton's method. Although otherwise attractive, there may be difficulty with convergence in the iterative solution of the nonlinear difference equations, and some efficiency is sacrificed by the need for iteration. An implicit predictor-corrector technique has been devised by Douglas and Jones (Ref. 43) which is applicable to the quasilinear case ($G = \phi$) of Eq. (A1). The first step of their procedure is to linearize the equation by evaluating the nonlinear coefficient as $F(\phi^n)$ and to predict values of $\phi^{n+1/2}$ using either the backward difference or the Crank-Nicolson scheme. Values for ϕ^{n+1} are then computed in a similar manner using $F(\phi^{n+1/2})$ and the Crank-Nicolson scheme. Gourlay and Morris (Ref. 44) have also proposed implicit predictor-corrector techniques which can be applied to Eq. (A1). In the conservative case ($F = 1$), their technique is to define $\hat{G}(\phi)$ by the relation $G(\phi) = \phi\hat{G}(\phi)$ when such a definition exists, and to evaluate $\hat{G}(\phi^{n+1})$ using values for ϕ^{n+1} computed by an explicit predictor scheme. With \hat{G} thereby known at the implicit time level, the equation can be treated as linear and corrected values of ϕ^{n+1} are computed by the Crank-Nicolson scheme.

A technique is described here for deriving linear implicit difference approximations for nonlinear differential equations. The technique is based

on an expansion of nonlinear implicit terms about the solution at the known time level, t^n , and leads to a one-step, two-level scheme which, being linear in unknown (implicit) quantities, can be solved efficiently without iteration. This idea was applied by Richtmyer and Morton (Ref. 29, p. 203) to a scalar nonlinear diffusion equation. Here, the technique is developed for problems governed by ℓ nonlinear equations in ℓ dependent variables which are functions of time and space coordinates. The technique will be described for the three-dimensional, unsteady equations.

The solution domain is discretized by grid points having equal spacings in the computational coordinates, Δy^1 , Δy^2 and Δy^3 in the y^1 , y^2 and y^3 directions, respectively, and an arbitrary time step, Δt . The subscripts i , j , k and superscript n are grid point indices associated with y^1 , y^2 , y^3 and t , respectively, and thus $\phi_{i,j,k}^n$ denotes $\phi(y_i^1, y_j^2, y_k^3, t^n)$. It is assumed that the solution is known at the n level, t^n , and is desired at the $(n+1)$ level, t^{n+1} . At the risk of an occasional ambiguity, one or more of the subscripts is frequently omitted, so that ϕ^n is equivalent to $\phi_{i,j,k}^n$.

The numerical method employed is quite general and is formally derived for systems of governing equations which have the following form:

$$\partial H(\phi)/\partial t = \mathcal{D}(\phi) + S(\phi) \tag{A2}$$

where ϕ is a column vector containing ℓ dependent variables, H and S are column vector functions of ϕ , and \mathcal{D} is a column vector whose elements are spatial differential operators which may be multidimensional. The generality of Eq. (A2) allows the method to be developed concisely and permits various extensions and modifications (e.g., noncartesian coordinate systems, turbulence models) to be made more or less routinely. It should be emphasized, however, that the Jacobian $\partial H/\partial \phi$ must usually be nonsingular if the ADI techniques as applied to Eq. (A2) are to be valid. A necessary condition is that each dependent variable appear in one or more of the governing equations as a time derivative. An exception would occur if for instance, a variable having no time derivative also appeared in only one equation, so that this equation could be decoupled from the remaining equations and solved a posteriori by an alternate method.

The linearized difference approximation is derived from the following implicit time-difference replacement of Eq. (A2):

$$(H^{n+1} - H^n)/\Delta t = \beta [\mathcal{D}(\phi^{n+1}) + S^{n+1}] + (1 - \beta) [\mathcal{D}(\phi^n) + S^n] \quad (A3)$$

where, for example, $H^{n+1} \equiv H(\phi^{n+1})$. The form of \mathcal{D} and the spatial differencing are as yet unspecified. A parameter β ($0 \leq \beta \leq 1$) has been introduced so as to permit a variable centering of the scheme in time. Equation (A3) produces a backward difference formulation for $\beta = 1$ and a Crank-Nicolson formulation for $\beta = 1/2$.

The linearization is performed by a two-step process of expansion about the known time level t^n and subsequent approximation of the quantity $(\partial\phi/\partial t)^n \Delta t$, which arises from chain rule differentiation, by $(\phi^{n+1} - \phi^n)$. The result is

$$H^{n+1} = H^n + (\partial H/\partial \phi)^n (\phi^{n+1} - \phi^n) + O(\Delta t)^2 \quad (A4a)$$

$$S^{n+1} = S^n + (\partial S/\partial \phi)^n (\phi^{n+1} - \phi^n) + O(\Delta t)^2 \quad (A4b)$$

$$\mathcal{D}(\phi^{n+1}) = \mathcal{D}(\phi^n) + (\partial \mathcal{D}/\partial \phi)^n (\phi^{n+1} - \phi^n) + O(\Delta t)^2 \quad (A4c)$$

The matrices $\partial H/\partial \phi$ and $\partial S/\partial \phi$ are standard Jacobians whose elements are defined, for example, by $(\partial H/\partial \phi)_{qr} \equiv \partial H_q/\partial \phi_r$. The operator elements of the matrix $\partial \mathcal{D}/\partial \phi$ are similarly ordered, i.e., $(\partial \mathcal{D}/\partial \phi)_{qr} \equiv \partial \mathcal{D}_q/\partial \phi_r$; however, the intended meaning of the operator elements requires some clarification. For the q^{th} row, the operation $(\partial \mathcal{D}_q/\partial \phi)^n (\phi^{n+1} - \phi^n)$ is understood to mean that $\{\partial/\partial t \mathcal{D}_q[\phi(x, y, z, t)]\}^n \Delta t$ is computed and that all occurrences of $(\partial \phi_r/\partial t)^n$ arising from chain rule differentiation are replaced by $(\phi_r^{n+1} - \phi_r^n)/\Delta t$.

After linearization as in Eqs. (A4), Eq. (A3) becomes the following linear implicit time-differenced scheme:

$$(\partial H^n / \partial \phi)(\phi^{n+1} - \phi^n) / \Delta t = \mathcal{D}(\phi^n) + S^n + \beta (\partial \mathcal{D} / \partial \phi + \partial S^n / \partial \phi)(\phi^{n+1} - \phi^n) \quad (A5)$$

Although H^{n+1} is linearized to second order in Eq. (A4), the division by Δt in Eq. (A3) introduces an error term of order Δt . A technique for maintaining formal second-order accuracy in the presence of nonlinear time derivatives is discussed by McDonald and Briley (Ref. 8), however, a three-level scheme results. Second-order temporal accuracy can also be obtained (for $\beta = 1/2$) by a change in dependent variable to $\hat{\phi} \equiv H(\phi)$, provided this is convenient, since the nonlinear time derivative is then eliminated. The temporal accuracy is independent of the spatial accuracy.

On examination, it can be seen that Eq. (A5) is linear in the quantity $(\phi^{n+1} - \phi^n)$ and that all other quantities are either known or evaluated at the n level. Computationally, it is convenient to solve Eq. (A5) for $(\phi^{n+1} - \phi^n)$ rather than ϕ^{n+1} . This both simplifies Eq. (A5) and reduces roundoff errors, since it is presumably better to compute a small $O(\Delta t)$ change in an $O(1)$ quantity than the quantity itself. To simplify the notation, a new dependent variable ψ defined by

$$\psi \equiv \phi - \phi^n \quad (A6)$$

is introduced, and thus $\psi^{n+1} = \phi^{n+1} - \phi^n$, and $\psi^n = 0$. It is also convenient to rewrite Eq. (A5) in the following simplified form:

$$(A + \Delta t \mathcal{L}) \psi^{n+1} = \Delta t [\mathcal{D}(\phi^n) + S^n] \quad (A7a)$$

where the following symbols have been introduced to simplify the notation:

$$A \equiv \partial H^n / \partial \phi - \beta \Delta t (\partial S^n / \partial \phi) \quad (A7b)$$

$$\mathcal{L} \equiv -\beta (\partial \mathcal{D} / \partial \phi) \quad (A7c)$$

It is noted that $\mathcal{L}(\psi)$ is a linear transformation and thus $\mathcal{L}(0) = 0$. Furthermore if $\mathcal{L}(\phi)$ is linear, then $\mathcal{L}(\psi) = -\beta \mathcal{D}(\psi)$.

Spatial differencing of Eq. (A7a) is accomplished simply by replacing derivative operators such as $\partial/\partial y^1$, $\partial^2/\partial y^1 \partial y^1$ by corresponding finite difference operators, D_1 , D_1^2 . Henceforth, it is assumed that \mathcal{D} and \mathcal{L} have been discretized in this manner, unless otherwise noted.

Before proceeding, some general observations seem appropriate. The foregoing linearization technique assumes only Taylor expandability, an assumption already implicit in the use of a finite difference method. The governing equations and boundary conditions are addressed directly as a system of coupled nonlinear equations which collectively determine the solution. The approach thus seems more natural than that of making ad hoc linearization and decoupling approximations, as is often done in applying implicit schemes to coupled and/or nonlinear partial differential equations. With the present approach, it is not necessary to associate each governing equation and boundary condition with a particular dependent variable and then to identify various "nonlinear coefficients" and "coupling terms" which must then be treated by lagging, predictor-corrector techniques, or iteration. The Taylor expansion procedure is analogous to that used in the generalized Newton-Raphson or quasi-linearization methods for iterative solution of nonlinear systems by expansion about a known current guess at the solution (e.g., Bellman & Kalaba, Ref. 45). However, the concept of expanding about the previous time level apparently had not been employed to produce a noniterative implicit time-dependent scheme for coupled equations, wherein nonlinear terms are approximated to a level of accuracy commensurate with that of the time differencing. The linearization technique also permits the implicit treatment of coupled nonlinear boundary conditions, such as stagnation pressure and enthalpy at subsonic inlet boundaries, and in practice, this latter feature was found to be crucial to the stability of the overall method (Ref. 30).

APPENDIX B

Tensor Notation and General Coordinates

In the space R^n with coordinates $(x^1, x^2, x^3, \dots, x^n)$ let the position vector be denoted by \vec{r} . A basis can be formed by constructing the covariant vectors \vec{e}_i

$$\vec{e}_i = \frac{\partial \vec{r}}{\partial x^i} \tag{B1}$$

The inner (dot) product of these vectors forms the function g_{ij} , which we will refer to as the metric tensor

$$g_{ij} = \vec{e}_i \cdot \vec{e}_j \tag{B2}$$

The reason for this terminology will be described subsequently. In three-dimension g_{ij} has the form

$$g_{ij} = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix}$$

The metric tensor g_{ij} is symmetric in the indices i and j so that there are 6 independent components. For an orthogonal coordinate system the off diagonal terms g_{ij} , $i \neq j$ are zero.

The differential arc length ds is defined as

$$\begin{aligned} (ds)^2 &= d\vec{r} \cdot d\vec{r} \\ &= \left(\frac{\partial \vec{r}}{\partial x^i} dx^i \right) \cdot \left(\frac{\partial \vec{r}}{\partial x^j} dx^j \right) \\ &= dx^i dx^j \vec{e}_i \cdot \vec{e}_j \\ ds^2 &= g_{ij} dx^i dx^j \end{aligned} \tag{B3}$$

Expanding we obtain

$$ds^2 = g_{11} dx^1 dx^1 + g_{22} dx^2 dx^2 + g_{33} dx^3 dx^3 + 2g_{12} dx^1 dx^2 + 2g_{13} dx^1 dx^3 + 2g_{23} dx^2 dx^3 \quad (B4)$$

For orthogonal curvilinear coordinates the arc length reduces to

$$ds^2 = g_{11} dx^1 dx^1 + g_{22} dx^2 dx^2 + g_{33} dx^3 dx^3 \quad (B5)$$

so that g_{11} , g_{22} , g_{33} can be identified with the scale factors as h_1^2 , h_2^2 and h_3^2 respectively.

The Kronecker delta function is defined as

$$\bar{e}^i \cdot \bar{e}_j = \delta_j^i = \begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases} \quad (B6)$$

Notes on raising and lowering indices

(a) multiplying by δ_i^j changes the index but keeps it in same position

$$u_j \delta_i^j = u_i$$

$$u^i \delta_i^j = u^j$$

(b) multiplying by g_{ij} and g^{ij}

$$u^i g_{ij} = u_j \quad \text{lowers index}$$

$$u_i g^{ij} = u^j \quad \text{raises index}$$

(c) multiplying δ_i^j by the metric tensor g^{jk}

$$\delta_j^i g^{jk} = \delta^{ki} \quad \text{raises index}$$

$$\delta_j^i g^{jk} = g^{ik} \quad \text{substitutes index}$$

Hence we obtain the following relationship

$$\delta^{ik} = g^{ik}$$

Note that δ^{ik} is not the Kronecker delta function in general. Only in the case of Cartesian coordinates where $g^{ik} = 1$ and $i = k$ is δ^{ik} the Kronecker delta function.

In order to obtain the contravariant basis \vec{e}^i we write

$$\vec{e}_i \cdot \vec{e}_j = g_{ij} = g_{ik} \delta_j^k$$

Employing equation (B-6)

$$\vec{e}_i \cdot \vec{e}_j = g_{ik} [\vec{e}^k \cdot \vec{e}_j]$$

we obtain

$$\vec{e}_i = g_{ik} \vec{e}^k \tag{B7}$$

Hence the metric tensor g_{ik} relates the covariant to the contravariant base vectors.

We may now define g^{jk} as

$$g^{jk} = \vec{e}^j \cdot \vec{e}^k \tag{B8}$$

We wish to relate g^{jk} to g_{jk} . Dotting equation (B-7) by \vec{e}^j we obtain

$$\vec{e}_i \cdot \vec{e}^j = g_{ik} \vec{e}^k \cdot \vec{e}^j$$

$$\delta_i^j = g_{ik} g^{kj} \tag{B9}$$

Now from the definition of g^{ij} (Equation B-8) we obtain the desired result

$$\vec{e}^j \cdot \vec{e}^i = g^{ij} = g^{ik} \delta_k^j = g^{ik} \vec{e}^j \cdot \vec{e}_k$$

**ORIGINAL PAGE IS
OF POOR QUALITY**

Therefore, g^{ik} also relates the covariant and contravariant basis vectors, i.e.

$$\bar{e}^i = g^{ik} \bar{e}_k \quad (\text{B10})$$

The metric tensors g^{ij} and g_{ij} can also be used to raise and lower indices. For example, consider the product of the tensors C_{jk} , C^{jk} , C^j_k , and C_k^j with g^{ij} or g_{ij}

$$C_{jk} g^{ij} = C^i_k$$

$$C^{jk} g_{ij} = C_i^k$$

$$C_j^k g^{ij} = C^{ki}$$

$$C_k^j g_{ij} = C_{ik}$$

Derivatives

The derivative of a basis vector \bar{e}_i is a vector in R^n so that it can be represented as a linear combination of other basis vectors, \bar{e}_k 's. Let \bar{e}_i be a vector, then its derivative with respect to x^j is

$$\frac{\partial \bar{e}_i}{\partial x^j} = D_j \bar{e}_i = \Gamma_{ji}^k \bar{e}_k$$

Dotting this expression with \bar{e}^l we obtain

$$D_j \bar{e}_i \cdot \bar{e}^l = \Gamma_{ji}^k \bar{e}_k \cdot \bar{e}^l = \Gamma_{ji}^k \delta_k^l = \Gamma_{ji}^l$$

where Γ_{ji}^l is the Christoffel symbol.

$$\Gamma_{ji}^l = D_j \bar{e}_i \cdot \bar{e}^l$$

Note that the Christoffel symbol $\overset{\circ}{\Gamma}_{ji}^k$ is not a third order tensor.
Representing $\overset{\circ}{\Gamma}_{ji}^k$ in terms of the g_{ij} 's we obtain

$$g_{mj} = \bar{e}_m \cdot \bar{e}_j$$

$$\begin{aligned} D_i g_{mj} &= D_i \bar{e}_m \cdot \bar{e}_j = D_i \bar{e}_m \cdot \bar{e}_j + \bar{e}_m \cdot D_i \bar{e}_j \\ &= \Gamma_{im}^k \bar{e}_k \cdot \bar{e}_j + \bar{e}_m \cdot \Gamma_{ij}^k \bar{e}_k \end{aligned}$$

$$D_i g_{mj} = \Gamma_{im}^k g_{kj} + g_{mk} \Gamma_{ij}^k$$

Similarly we can show that

$$D_j g_{mi} = g_{ki} \Gamma_{jm}^k + g_{mk} \Gamma_{ji}^k$$

and

$$-D_m g_{ij} = -g_{kj} \Gamma_{mi}^k - g_{ik} \Gamma_{mj}^k$$

Employing the symmetry property of the Γ 's in their lower index, i.e.

$$\Gamma_{ij}^k = \Gamma_{ji}^k$$

and adding the three derivatives we obtain

$$\Gamma_{ij}^k = \frac{g^{km}}{2} [D_i g_{mj} + D_j g_{mi} - D_m g_{ij}]$$

which relates the Christoffel symbol to the derivatives of the components of the metric tensor.

The Kronecker delta function is defined as

$$\delta_j^k = \vec{e}^k \cdot \vec{e}_j$$

Taking its derivative, we obtain

$$\begin{aligned} D_l \delta_j^k &= 0 = D_l \vec{e}^k \cdot \vec{e}_j \\ 0 &= D_l \vec{e}^k \cdot \vec{e}_j + \vec{e}^k \cdot D_l \vec{e}_j \end{aligned}$$

Substituting in the expression for the derivative of a basis vector

$$D_l \vec{e}_j = \Gamma_{lj}^m \vec{e}_m$$

and rearranging,

$$D_l \vec{e}^k \cdot \vec{e}_j = -\vec{e}^k \cdot \Gamma_{lj}^m \vec{e}_m = -\Gamma_{lj}^m \delta_m^k = -\Gamma_{lj}^k$$

we obtain

$$\Gamma_{ij}^k = -D_l \vec{e}^k \cdot \vec{e}_j$$

which relates the Christoffel symbol to the derivative of a contravariant basis vector.

Other properties of Christoffel symbols are

$$(a) \quad \Gamma_{ijk} g^{kl} = \Gamma_{ij}^l = \Gamma_{lj}^k \delta_k^l$$

$$(b) \quad \Gamma_{ij}^k g_{kl} = \Gamma_{ijl}$$

There is also symmetry of lower indices, i.e.

$$\Gamma_{\underline{ij}k} = \Gamma_{\underline{ji}k}$$

Covariant Derivative

Consider the derivative of a vector \vec{u} , expressed in a covariant basis

$$\vec{u} = u^i \vec{e}_i$$

$$\frac{\partial \vec{u}}{\partial x^j} = D_j u^i \vec{e}_i = D_j u^i \vec{e}_i + u^i D_j \vec{e}_i$$

Now the derivative of a basis vector is

$$D_j \vec{e}_i = \Gamma_{ij}^k \vec{e}_k$$

so that the last term in the expression becomes

$$u^i D_j \vec{e}_i = u^i \Gamma_{ij}^k \vec{e}_k$$

Since k and i are dummy indices, we may interchange indices, i.e.

$$u^i D_j \vec{e}_i = u^i \Gamma_{ij}^k \vec{e}_k = u^k \Gamma_{kj}^i \vec{e}_i$$

Combining, we obtain the desired expression

$$\frac{\partial \vec{u}}{\partial x^j} = (D_j u^i + u^k \Gamma_{kj}^i) \vec{e}_i$$

The term in the parentheses is called the covariant derivative and is denoted as follows

$$\frac{\partial \vec{u}}{\partial x^j} = u^i |_{,j} \vec{e}_i$$

In a Cartesian coordinate system the covariant derivative reduces to the partial derivative, i.e.

$$u^i |_{,j} = u^i_{,j}$$

Consider the derivative of a vector in a contravariant basis which is given by

$$D_i u_j \vec{e}^j = D_i u_j \vec{e}^j + u_j D_i \vec{e}^j$$

The last term can be expressed in terms of Christoffel symbols, i.e.

$$u_j D_i \bar{e}^j = u_j \Gamma_{ik}^j \bar{e}^k = u_k \Gamma_{ij}^k \bar{e}^j$$

Thus we obtain the desired result

$$D_i u_j \bar{e}^j = (\dot{D}_i u_j - u_k \Gamma_{ij}^k) \bar{e}^j = (u_{j,i} - u_k \Gamma_{ij}^k) \bar{e}^j$$

$$D_i u_j \bar{e}^j = u_j |_{,i} \bar{e}^j$$

Now in Cartesian coordinates the covariant derivative of the metric tensor is zero,

$$g_{ij,k} = \delta_{ij,k} = 0$$

Since this is a tensor equation it is valid not only in a Cartesian system but in every coordinate system, i.e.

$$g_{ij} |_{,k} = 0$$

Consider now the vector u_ℓ .

$$u_\ell = u^k g_{\ell k}$$

Taking the covariant derivative of u_ℓ and using the previous relationship, we obtain the desired result

$$u_\ell |_{,i} = (u^k g_{\ell k}) |_{,i} = u^k |_{,i} g_{\ell k} + u^k g_{\ell k} |_{,i}$$

$$u_\ell |_{,i} = u^k |_{,i} g_{\ell k}$$

An alternate expression for the derivative of a covariant vector is given by

$$\begin{aligned} D_i u_j \bar{e}^j &= u_j |_{,i} \bar{e}^j = u^j |_{,i} g_{lj} \bar{e}^l \\ &= u^j |_{,i} \bar{e}_j \end{aligned}$$

Note that the dummy indices are interchanged.

Vector Operators

In this section we consider the vector operators, gradient and divergence operating on scalars, vectors and tensors.

Gradient

Scalar (ϕ)

The gradient of a scalar is given as

$$\nabla u = \bar{e}^k \otimes D_k \phi = g^{km} \bar{e}_m \otimes D_k \phi$$

where \otimes denotes a tensor product.

Vector ($\bar{u} = u^j \bar{e}_j$)

The gradient of a vector becomes

$$\begin{aligned} \nabla \bar{u} &= \bar{e}^k \otimes D_k u^j \bar{e}_j = \bar{e}^k \otimes \{ D_k u^j \bar{e}_j + u^j D_k \bar{e}_j \} \\ &= u^j |_{,k} \bar{e}^k \otimes \bar{e}_j \\ &= g^{mk} u^j |_{,k} \bar{e}_m \otimes \bar{e}_j \end{aligned}$$

which is a second order tensor.

Tensor

Consider a tensor of the form $\bar{u} = a^{ij} \bar{e}_i \otimes \bar{e}_j$. The gradient of this tensor can be obtained as follows

$$\begin{aligned} \nabla \bar{u} &= \bar{e}^k \otimes D_k a^{ij} \bar{e}_i \otimes \bar{e}_j \\ &= \bar{e}^k \otimes \{ D_k a^{ij} (\bar{e}_i \otimes \bar{e}_j) + a^{ij} (D_k \bar{e}_i) \otimes \bar{e}_j + a^{ij} \bar{e}_i \otimes D_k \bar{e}_j \} \\ &= \bar{e}^k \otimes \{ D_k a^{ij} (\bar{e}_i \otimes \bar{e}_j) + a^{ij} \Gamma_{ki}^l \bar{e}_l \otimes \bar{e}_j + a^{ij} \bar{e}_i \otimes \Gamma_{kj}^l \bar{e}_l \} \end{aligned}$$

which yields a third order tensor

$$\begin{aligned} \nabla \bar{u} &= \{ D_k a^{ij} + a^{lj} \Gamma_{kl}^i + a^{il} \Gamma_{kl}^j \} \bar{e}^k \otimes \bar{e}_i \otimes \bar{e}_j \\ &= \{ D_k a^{ij} + a^{lj} \Gamma_{kl}^i + a^{il} \Gamma_{kl}^j \} g^{mk} \bar{e}_m \otimes \bar{e}_i \otimes \bar{e}_j \\ &= (a^{ij} |_{k}) g^{mk} \bar{e}_m \otimes \bar{e}_i \otimes \bar{e}_j \end{aligned}$$

ORIGINAL PAGE IS
OF POOR QUALITY

Divergence

The divergence of a vector of the form ($\vec{U} = u^i \vec{e}_i$) can be obtained as follows

$$\begin{aligned}\nabla \cdot \vec{U} &= \vec{e}^k \cdot D_k u^i \vec{e}_i \\ &= \vec{e}^k \cdot u^i \Big|_k \vec{e}_i = u^i \Big|_k \vec{e}^k \cdot \vec{e}_i \\ &= u^i \Big|_k \delta_i^k \\ \nabla \cdot \vec{U} &= u^k \Big|_k\end{aligned}$$

From the definition of the covariant derivative we obtain

$$u^k \Big|_k = D_k u^k + u^i \Gamma_{ik}^k$$

It can be shown (Ref. 13) that

$$\Gamma_{ik}^k = \frac{1}{J} D_i J$$

Substituting this expression into the definition of the covariant derivative we obtain an alternate form of the divergence involving the Jacobian

$$u^k \Big|_k = D_k u^k + u^i \frac{1}{J} D_i J = D_k u^k + u^k \frac{1}{J} D_k J$$

$$u^k \Big|_k = \frac{1}{J} D_k (J u^k)$$

Note that this relationship is in conservation form.

Consider now the tensor of the form $(\bar{u} = a^{ij} \bar{e}_i \otimes \bar{e}_j)$
Again we obtain its divergence as follows

$$\begin{aligned} \nabla \cdot \bar{u} &= \nabla \cdot a^{ij} \bar{e}_i \otimes \bar{e}_j \\ &= \bar{e}^k \cdot D_k (a^{ij} \bar{e}_i \otimes \bar{e}_j) \\ &= \bar{e}^k [D_k a^{ij} \bar{e}_i \otimes \bar{e}_j + a^{ij} D_k \bar{e}_i \otimes \bar{e}_j + a^{ij} \bar{e}_i \otimes D_k \bar{e}_j] \\ &= D_k a^{ij} \bar{e}^k \cdot \bar{e}_i \otimes \bar{e}_j + a^{ij} \Gamma_{ki}^{\ell} \bar{e}^k \cdot \bar{e}_\ell \otimes \bar{e}_j + a^{ij} \bar{e}^k \cdot \bar{e}_i \otimes \Gamma_{kj}^{\ell} \bar{e}_\ell \\ &= D_k a^{ij} \delta_i^k \bar{e}_j + a^{ij} \Gamma_{ki}^{\ell} \delta_\ell^k \bar{e}_j + a^{ij} \Gamma_{kj}^{\ell} \delta_i^k \bar{e}_\ell \end{aligned}$$

by interchanging indices j and l we obtain

$$\nabla \cdot \bar{u} = [D_i a^{ij} + a^{ij} \Gamma_{\ell i}^{\ell} + a^{i\ell} \Gamma_{i\ell}^j] \bar{e}_j$$

which reduces to

$$\nabla \cdot \bar{u} = a^{ij} |_{,i} \bar{e}_j$$

or

$$\nabla \cdot \bar{u} = \left\{ \frac{1}{J} D_i (a^i J) + a^{i\ell} \Gamma_{i\ell}^j \right\} \bar{e}_j$$

We will consider now the tensor of the following form ($\bar{u} = a_j^i \bar{e}_i \otimes \bar{e}^j$)
The divergence of \bar{u} becomes

$$\begin{aligned} \nabla \cdot \bar{u} &= \bar{e}^k \left[D_k a_j^i \bar{e}_i \otimes \bar{e}^j + a_j^i D_k \bar{e}_i \otimes \bar{e}^j + a_j^i \bar{e}_i \otimes D_k \bar{e}^j \right] \\ &= \bar{e}^k \cdot \left[D_k a_j^i \bar{e}_i \otimes \bar{e}^j + a_j^i \Gamma_{ki}^\ell \bar{e}_\ell \otimes \bar{e}^j - a_j^i \bar{e}_i \otimes \Gamma_{k\ell}^j \bar{e}^\ell \right] \end{aligned}$$

Rearranging and interchanging dummy indices we obtain

$$\begin{aligned} \nabla \cdot u &= \left[D_k a_j^i + a_j^\ell \Gamma_{k\ell}^i - a_\ell^i \Gamma_{kj}^\ell \right] (\bar{e}^k \cdot \bar{e}_i) \otimes \bar{e}^j \\ &= \left[D_i a_j^i + a_j^\ell \Gamma_{i\ell}^i - a_\ell^i \Gamma_{ij}^\ell \right] \bar{e}^j \end{aligned}$$

Now the contravariant basis vector is defined as

$$\bar{e}^j = \bar{e}_k g^{jk}$$

Substituting into the expression for $\nabla \cdot \bar{u}$ and with some manipulation we obtain the divergence of the second order tensor as

$$\nabla \cdot \bar{u} = \left[D_i a_j^i + a_j^\ell \Gamma_{i\ell}^i - a_\ell^i \Gamma_{ij}^\ell \right] g^{jk} \bar{e}_k$$

$$\nabla \cdot \bar{u} = \left[\frac{1}{J} D_i (a_j^i J) - a_\ell^i \Gamma_{ij}^\ell \right] g^{jk} \bar{e}_k$$

In order to obtain $\nabla \cdot \bar{u}$ in the \vec{e}_i coordinate system we employ the property of the metric tensor to raise and lower indices

$$\nabla \cdot \bar{u} = [D_k a_j^i + a_j^\ell \Gamma_{k\ell}^i - a_\ell^i \Gamma_{kj}^\ell] \bar{e}^k \cdot \bar{e}^j \otimes \bar{e}_i$$

$$\bar{e}^k \cdot \bar{e}^j = g^{kj}$$

and obtain the desired result,

$$\nabla \cdot \bar{u} = \{ [D_k a_j^i + a_j^\ell \Gamma_{k\ell}^i - a_\ell^i \Gamma_{kj}^\ell] g^{kj} \} \bar{e}_i$$

REFERENCES

1. Kendall, R.M., Bonnett, W.S., Nardo, C.T., and Abbett, M.J.: Computational Procedure for Three-Dimensional Boundary Layers on Aircraft and Aerospace Vehicles. Proceedings ATAA 2nd Fluid Dynamics Conference, Hartford, 1975, pp. 113-121.
2. Briley, W.R., and McDonald, H.: On the Structure and Use of Linearized Block ADI and Related Schemes. J. Comp. Physics, Vol. 34, No. 1, 1980.
3. Weinberg, B.C., and McDonald H.: Solution of Three-Dimensional Time Dependent Viscous Flows, Part 1: Investigation of Candidate Algorithms. SRA Report R79-90004, Final Contractor's Report (NAS2-10016), 1979.
4. McCroskey, W.J.: Some Current Research in Unsteady Fluid Dynamics. J. Fluids Engineering, Vol. 99, March 1977, pp. 8-38.
5. Nash, J.F., and Patel, V.C.: Three-Dimensional Turbulent Boundary Layers. Scientific and Business Consultants Inc., 1972.
6. Lin, T.C., and Rubin, S.G.: Viscous Flow over a Cone at Moderate Incidence. Part 2. Supersonic Boundary Layer., J. Fluid Mech., Vol. 59, Part 3, 1973, pp. 593-620.
7. Briley, W.R.: Numerical Method for Predicting Three-Dimensional Steady Viscous Flow in Ducts. J. Comp. Physics, Vol. 14, No. 1, 1974.
8. McDonald, H., and Briley, W.R.: Three-Dimensional Supersonic Flow of a Viscous or Inviscid Gas. J. of Comp. Physics, Vol. 19, No. 2, 1975.
9. Blottner, F.G.: Investigation of Some Finite Difference Techniques for Solving the Boundary Layer Equations. Computer Methods in Applied Mechanics and Engineering. Vol. 6, 1975, pp. 1-30.
10. Howarth, L.: The Boundary Layer in Three-Dimensions-Part I. Derivation of the Equations for Flow Along a General Curved Surface. Philosophical Magazine, Vol. 7, 1951, pp. 239.
11. Roberts, G.O.: Computational Meshes for Boundary Layer Problems. Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics, Springer-Verlag, New York, 1971, p. 171.
12. Flugge, W.: Tensor Analysis and Continuum Mechanics, Springer-Verlag, 1972.
13. Eiseman, P.R.: The Numerical Solution of the Fluid Dynamical Equations in Curvilinear Coordinates. Air Force Weapons Laboratory Report AFWL-TR-73-172, 1973.
14. Gibelg, H.J., Shamroth, S.J., and Eiseman, P.R.: Analysis of Strong-Interaction Dynamic Stall for Laminar Flow on Airfoils. NASA CR-2969, April 1978.

15. Favre, A.: Equations des Gaz Turbulents Compressibles. J. de Mecanique, Vol. 4, pp. 361-392, 1965.
16. Berger, A.E., Solomon, J.M., Ciment, M., Leventhal, S.H., and Weinberg, B.C.: Generalized OCI Schemes for Boundary Layer Problems, Math Comp., Vol. 35, No. 151, 1980.
17. Allen, D.N., and Southwell, R.V.: Relaxation Methods Applied to Determine the Motion, in Two Dimensions, of a Viscous Fluid Past a Fixed Cylinder, Quart. J. Mech. Appl. Math., 8, 1955, pp. 129-145.
18. Il'in, A.M.: Differencing Scheme for a Differential Equation with a Small Parameter Affecting the Highest Derivative. Mat. Zametki, 6, 1969, pp. 237-248 - Math. Notes, 6, 1969, pp. 596-602.
19. McDonald, H.: A Novel Finite Difference Scheme Particularly Suited for Fluid Flow Problems Containing Discontinuities. Scientific Research Associates Report P76-1, August 1976.
20. Dennis, S.C.R.: Finite Differences Associated with Second-Order Differential Equations, Quart. J. Mech. Appl. Math., Vol. 23, 1960, pp. 487-507.
21. El-Mistikawy, T.M., and Werle, M.J.: Numerical Methods for Inviscid, Viscous Fluid Flows, Report No. AFL 77-9-34, University of Cincinnati, March 1977.
22. El-Mistikawy, T.M., and Werle, M.J.: Numerical Method for Boundary Layers with Blowing - the Exponential Box Scheme. AIAA J., 16, 1978, pp. 749-751.
23. Berger, A.E., Solomon, J.M., and Ciment, M.: An Analysis of a Uniformly Accurate Finite Difference Method for a Singular Perturbation Problem. Paper presented at Second International Conference on Computational Methods in Nonlinear Mechanics, March 26-30, Austin, Texas.
24. Peaceman, D.W., and Rachford, H.H.: The Numerical Solution of Parabolic and Elliptic Differential Equations. Soc. for Indust. Appl. Math., Vol. 3, 1955, pp. 28-41.
25. Douglas, J.: On the Numerical Integration of $u_{xx} + u_{yy} = u_t$ by Implicit Methods. Soc. for Indust. Appl. Math., Vol. 3, 1955, pp. 42-65.
26. Mitchell, A.R.: Computational Methods in Partial Differential Equations. Wiley, New York, 1969.
27. Yanenko, N.N.: The Method of Fractional Steps. Springer-Verlag, 1971.
28. Douglas, J., and Gunn, J.E.: A General Formulation of Alternating Direction Methods. Numerische Math., Vol. 6, 1964, p. 428.
29. Richtmyer, R.D., and Morton, K.W.: Difference Methods for Initial Value Problems. Second Edition. Interscience Publishers, New York, New York, 1967.
30. Briley, W.R., and McDonald, H.: Solution of the Multidimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method. J. Comp. Physics, Vol. 24, No. 4, August, 1977, p. 372.

31. Gibeling, H.J., McDonald, H., and Briley, W.R.: Development of a Three-Dimensional Combustor Flow Analysis, Volume I: Theoretical Studies. Technical Report AFAPL-TR-75-59, Vol. I, July 1975.
32. Gibeling, H.J., McDonald, H., and Briley, W.R.: Development of a Three-Dimensional Combustor Flow Analysis, Volume II: Theoretical Studies. Technical Report AFAPL-TR-75-59, Vol. II, October, 1976.
33. Leventhal, S.H.: The Operator Compact Implicit Method for Reservoir Simulation, Proceedings of Fifth SPE Symposium on Numerical Reservoir Simulation, 1979.
34. Blottner, F.G.: Computational Techniques for Boundary Layers in "Computational Methods for Inviscid and Viscous Two and Three-Dimensional Flow Fields". Fluid Dynamics Institute, Hanover, New Hampshire, 1975.
35. Weinberg, B.C., Leventhal, S.H., and Ciment, M.: The Operator Compact Implicit Scheme for Viscous Flow Problems. AIAA Paper No. 77-638. Presented at the 3rd AIAA Comp. Fluid Dynamics Conf., Albuquerque, N.M., 1977.
36. Weinberg, B.C.: Viscous Flow Calculations Employing a Fourth Order Generalized Operator Compact Implicit Scheme, Paper 79-1468, Presented at 4th AIAA Comp. Fluid Dynamics Conf., Williamsburg, Va., 1979.
37. Fairweather, G., and Mitchell, A.K.: A New Computational Procedure for ADI Methods, SIAM J. Numer. Anal. Vol. 4, No. 2, 1967.
38. Schlichting, H.: Boundary Layer Theory, Fourth Edition, McGraw-Hill Book Co., Inc., New York, 1960.
39. Howarth, L.: On the Solution of the Laminar Boundary Layer Equations, Proc. Roy. Soc. A. 164, 1938.
40. Briley, W.R.: A Numerical Study of Laminar Separation Bubbles Using the Navier-Stokes Equations, J. Fluid Mech., Vol. 47, Part 4, pp. 713-736, 1971.
41. Ames, W.F.: Numerical Methods for Partial Differential Equations. Barnes & Noble, Inc., New York, New York, 1969.
42. von Rosenberg, D.A.: Methods for the Numerical Solution of Partial Differential Equations. American Elsevier Publishing Co., Inc., New York, New York, 1969.
43. Douglas, J., and Jones, B.F.: On Predictor-Corrector Methods for Nonlinear Parabolic Differential Equations. Soc. for Indust. Appl. Math., Vol. 11, 1963, pp. 195-204.
44. Gourlay, A.R., and Morris, J.L.: Finite-Difference Methods for Nonlinear Hyperbolic Systems. Math. Comp., Vol. 22, 1968, pp. 28-39.
45. Bellman, R.E., and Kalaba, R.E.: Quasilinearization and Nonlinear Boundary-Value Problems. American Elsevier Publ. Co., Inc., New York, 1965.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE I. - OPERATOR COEFFICIENTS FOR STANDARD
OPERATOR COMPACT IMPLICIT SCHEME

$$\begin{aligned}
 q_j^- &= 6 - 5\rho_j + 2\rho_{j+1} - \rho_j\rho_{j+1} \\
 q_j^c &= 60 + 16\rho_{j+1} - 16\rho_{j-1} - 4\rho_{j-1}\rho_{j+1} \\
 q_j^+ &= 6 + 5\rho_j - 2\rho_{j-1} - \rho_j\rho_{j-1} \\
 r_j^- &= q_j^- \left(1 - \frac{3}{2}\rho_{j-1}\right) + q_j^c \left(1 - \frac{1}{2}\rho_j\right) + q_j^+ \left(1 + \frac{1}{2}\rho_{j+1}\right) + h^2 q_j^- c_{j-1} \\
 r_j^c &= -(r_j^+ + r_j^-) + h^2 (q_j^- c_{j-1} + q_j^c c_j + q_j^+ c_{j+1}) \\
 r_j^+ &= q_j^- \left(1 - \frac{1}{2}\rho_{j-1}\right) + q_j^c \left(1 + \frac{1}{2}\rho_j\right) + q_j^+ \left(1 + \frac{3}{2}\rho_{j+1}\right) + h^2 q_j^+ c_{j+1}
 \end{aligned}$$

where

$$\rho_j = hb_j$$

**ORIGINAL PAGE IS
OF POOR QUALITY**

**TABLE II. - OPERATOR COEFFICIENTS FOR GENERALIZED
OPERATOR COMPACT IMPLICIT SCHEME**

$$q_j^- = 6 + [p_1 - 3]Rc_j + [p_2]Rc_j^2$$

$$q_j^c = 60 + [10p_1]Rc_j + [p_3]Rc_j^2 + [\tau_{j+1}p_4]Rc_j^3$$

$$q_j^+ = 6 + [p_1 + 3]Rc_j + [p_1 + p_2]Rc_j^2 + [p_4]Rc_j^3$$

where

$$p_1 = 3, \quad p_2 = 0, \quad p_3 = \max[\pi_1, \pi_2]$$

$$\pi_1 = (\tau_{j+1} + \tau_{j-1})p_2 + \tau_{j+1}p_1 + \bar{\pi}_1 \quad \pi_2 = 15 - 2p_2 + (\sigma_2 - 1)p_1 - 3(\tau_{j+1} + \sigma_2) + \bar{\pi}_2$$

$$\bar{\pi}_1 = \begin{cases} 0 & \sigma_1 \geq 0 \\ \frac{3}{8}\sigma_1^2(10 - \tau_{j+1} - \tau_{j-1}) & \sigma_1 \leq 0 \end{cases} \quad \bar{\pi}_2 = \begin{cases} 0 & 2p_1 - \sigma_2 \geq 0 \\ (2p_1 - \sigma_2)^2/8 & 2p_1 - \sigma_2 < 0 \end{cases}$$

$$\sigma_1 = p_1/3 + \frac{(\tau_{j-1} - \tau_{j+1})}{10 - \tau_{j+1} - \tau_{j-1}}$$

$$2\sigma_2 = 3\tau_{j+1} - \tau_{j-1} + 10 + 2h\tau_{j-1}\left(\frac{c_{j-1}}{b_{j-1}}\right)$$

$$p_4 = \frac{1}{2} [1 + \tau_{j+1}]^{-1} \pi_3 \quad \pi_3 = p_3 - \pi_1 + \bar{\pi}_1 + 2\tau_{j-1}\left(2 + h\frac{c_{j-1}}{b_{j-1}}\right)p_2$$

with h sufficiently small so that

$$10b_j - b_{j-1} - b_{j+1} > 0 \quad \text{and} \quad 2 + hc_{j+1}/b_{j+1} > 0 \quad \text{for } j=2, \dots, J \quad \text{and } c_j \leq 0$$

where

$$\tau_{j-1} = b_{j-1}/b_j, \quad \tau_{j+1} = b_{j+1}/b_j \quad \text{and} \quad Rc_j = hb_j$$

r_j^-, r_j^c, r_j^+ given in TABLE I

TABLE III.- OPERATOR COEFFICIENTS FOR
ALLEN SOUTHWELL EXPONENTIAL SCHEME

$$r_j^- = Rc_j e^{-Rc_j} / (1 - e^{-Rc_j})$$

$$r_j^+ = Rc_j / (1 - e^{-Rc_j})$$

$$r_j^c = -Rc_j + c_j$$

$$q_j^- = 0$$

$$q_j^c = 1$$

$$q_j^+ = 0$$

where $Rc_j = hb_j$

TABLE IV. - OPERATOR COEFFICIENTS FOR EL-MISTIKAWY WERLE
EXPONENTIAL BOX SCHEME

$$r_j^- = \rho^- \exp(-\rho^-) / [1 - \exp(-\rho^-)]$$

$$r_j^+ = \rho^+ / [1 - \exp(-\rho^+)]$$

$$r_j^c = -(r_j^+ + r_j^-)$$

$$q_j^- = (1 - r_j^-) / (2\rho^-)$$

$$q_j^+ = (r_j^+ - 1) / (2\rho^+)$$

$$q_j^c = q_j^- + q_j^+$$

where

$$\rho^- = \frac{1}{2}(\rho_{j-1} + \rho_j) , \quad \rho^+ = \frac{1}{2}(\rho_j + \rho_{j+1})$$

and

$$\rho_j = hb_j$$

TABLE V. - OPERATOR COEFFICIENTS FOR SECOND ORDER
FINITE DIFFERENCES WITH ARTIFICIAL VISCOSITY

$$\begin{aligned}
 r_j^- &= 1 - S \\
 r_j^c &= -2 + ct \\
 r_j^+ &= 1 + S \\
 q_j^- &= 0 \\
 q_j^c &= t \\
 q_j^+ &= 0
 \end{aligned}$$

where $S = 1$ for $Rc_j > 2$

$S = -1$ for $Rc_j < -2$

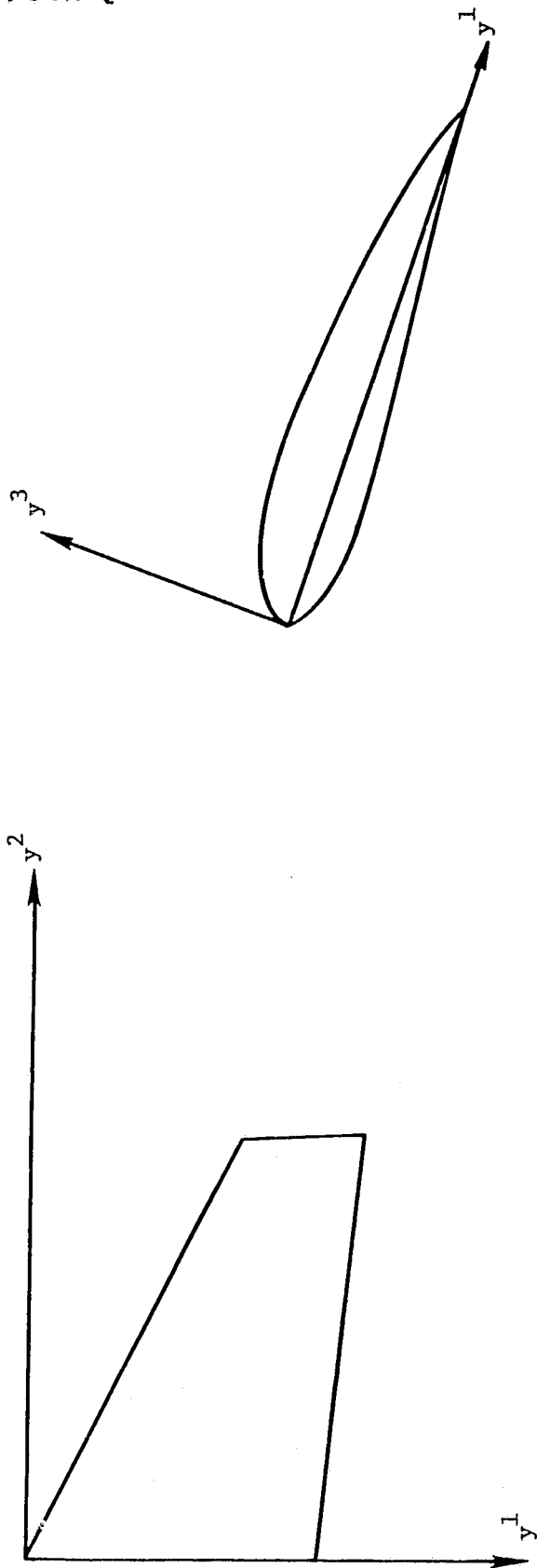
$$t = \frac{2}{|Rc_j|}$$

and $Rc_j = hb_j$

for $|Rc_j| < 2$, r_j^- , r_j^c , r_j^+ , q_j^- , q_j^c , q_j^+

reduce to standard finite differences.

ORIGINAL PAGE IS
OF POOR QUALITY

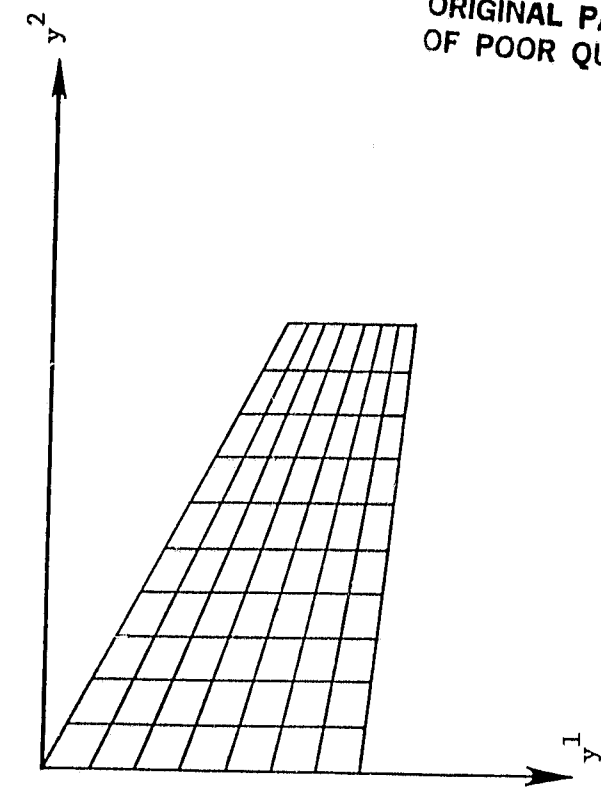


(a) Plan Form

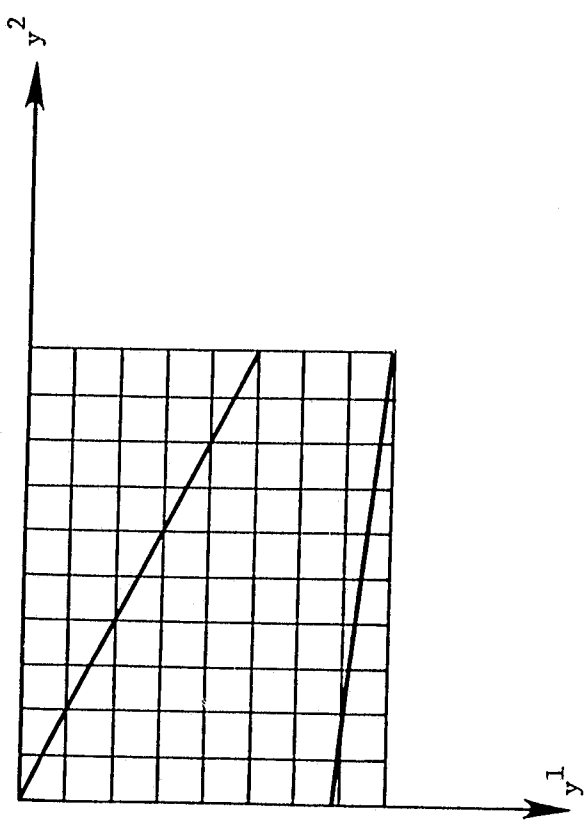
(b) Cross Section

Fig. 1 - Typical Airfoil Section.

ORIGINAL PAGE IS
OF POOR QUALITY



(a) Cartesian Coordinates



(b) Nonorthogonal Coordinates

Fig. 2 - Effect of Coordinate System on Grid Distribution.

ORIGINAL PAGE IS
OF POOR QUALITY

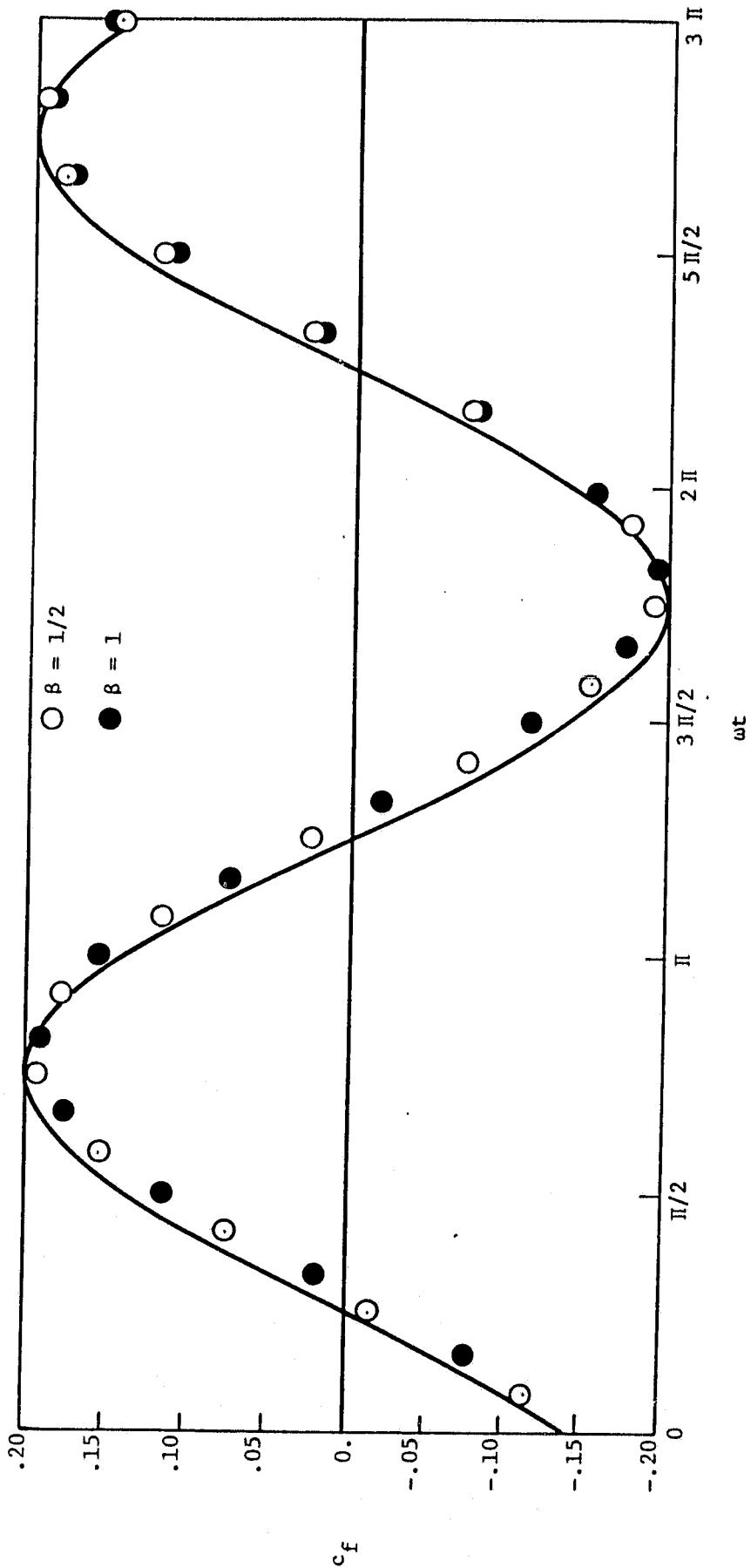


Fig. 3 - Skin Friction Coefficient for Oscillating Flat Plate.

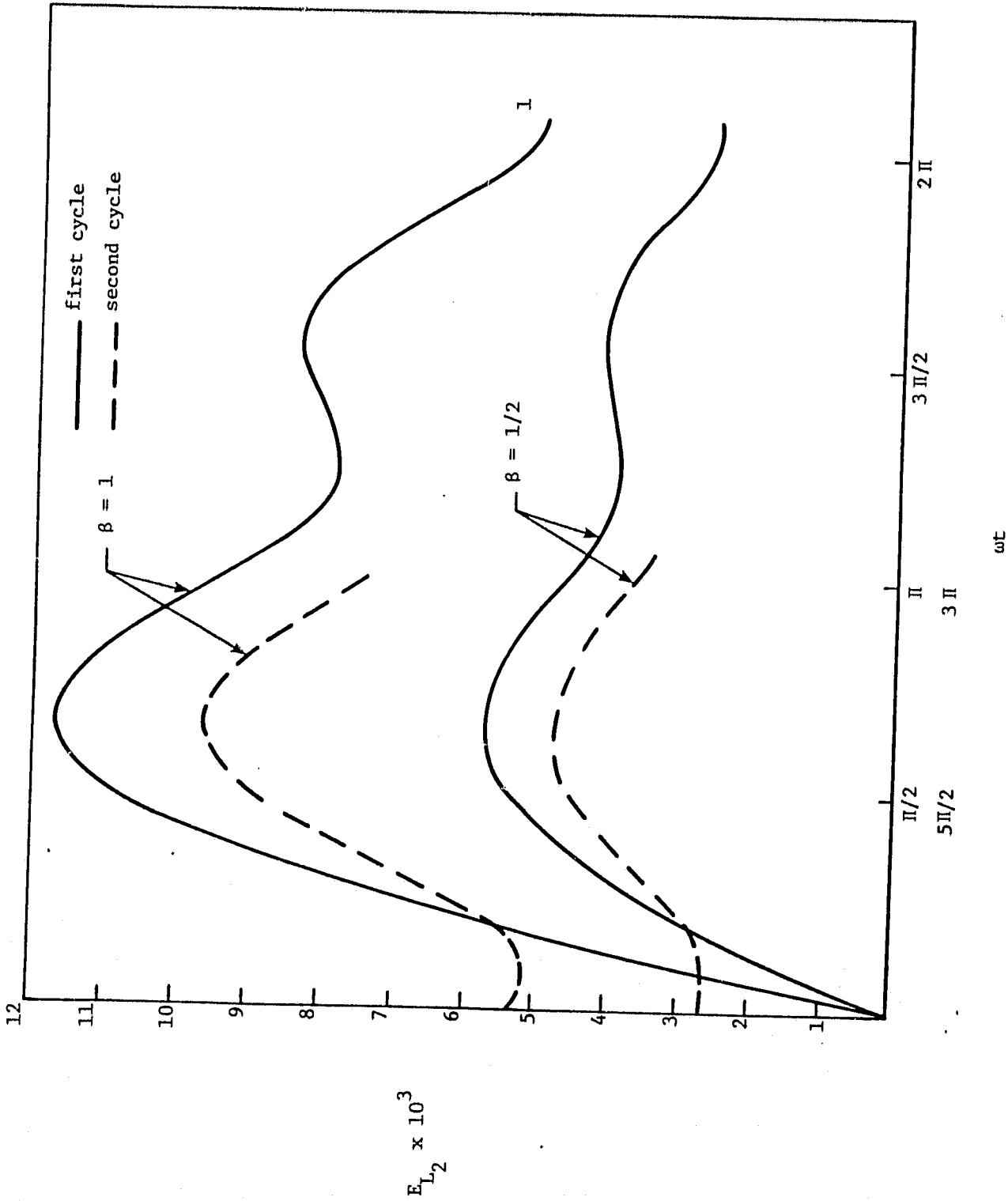


Fig. 4 - L_2 Error in Computing Flow Field for Oscillating Flat Plate.

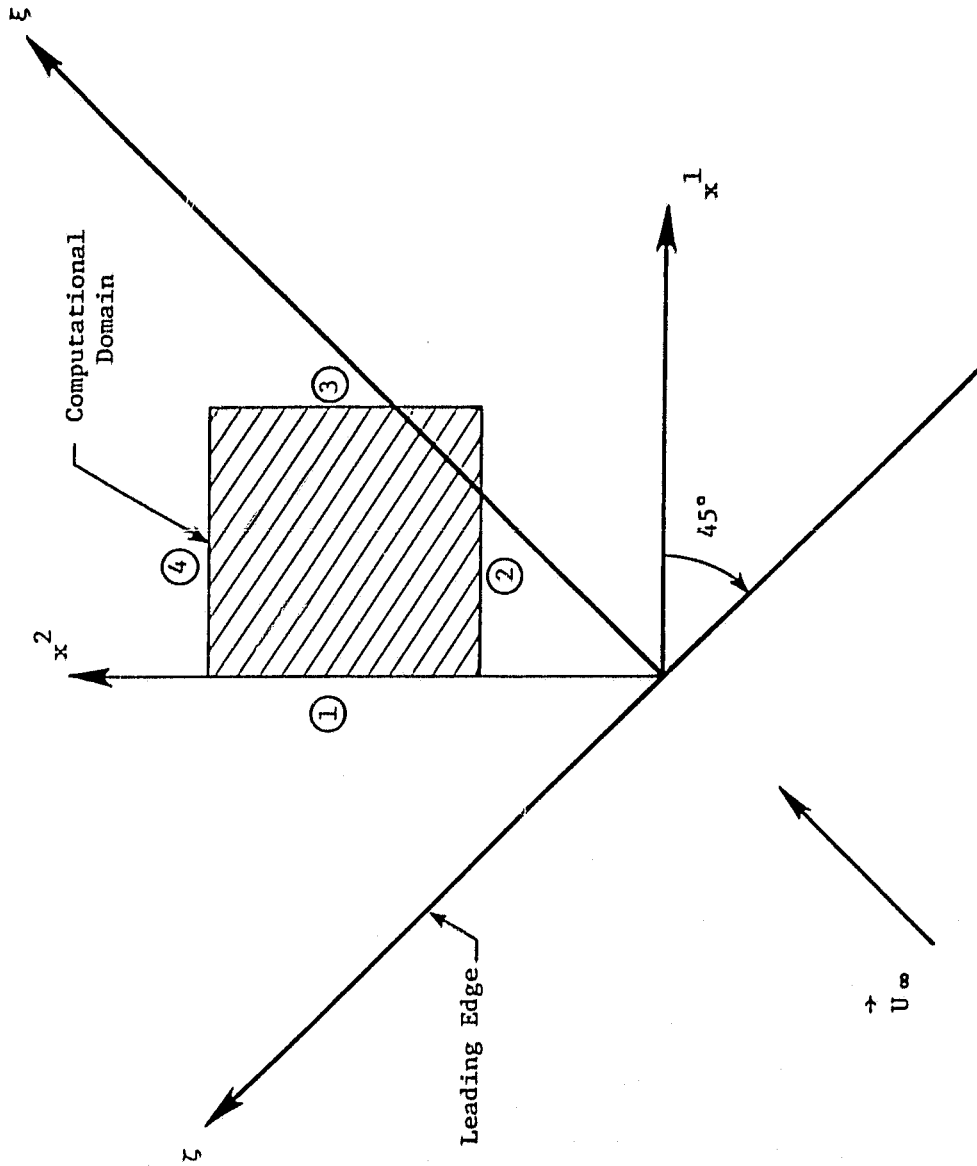
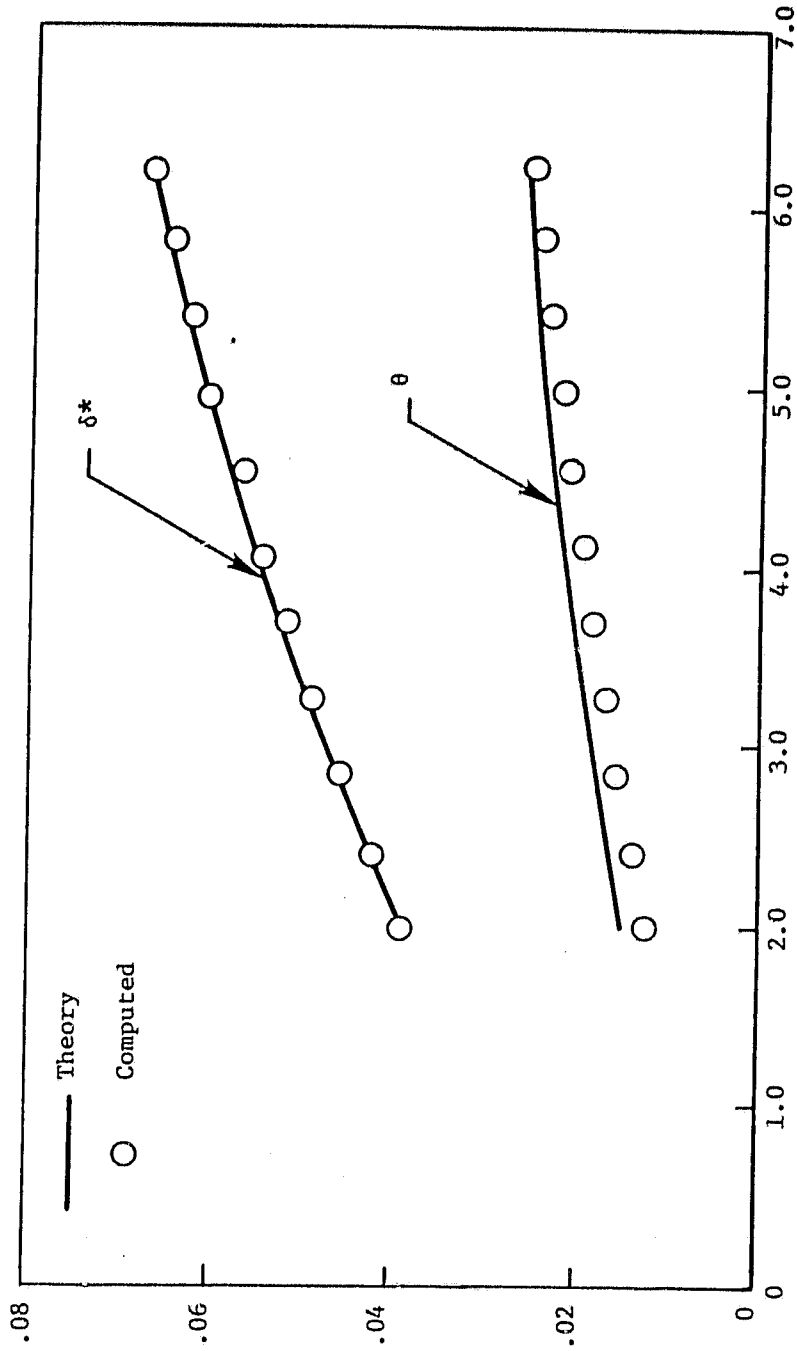


Fig. 5 - Computational Domain for Three-Dimensional Blasius Flow Problem.



ξ

Fig. 6 - Displacement Thickness and Momentum Thickness Distribution Along Diagonal.

δ^*, θ

ORIGINAL PAGE IS
OF POOR QUALITY

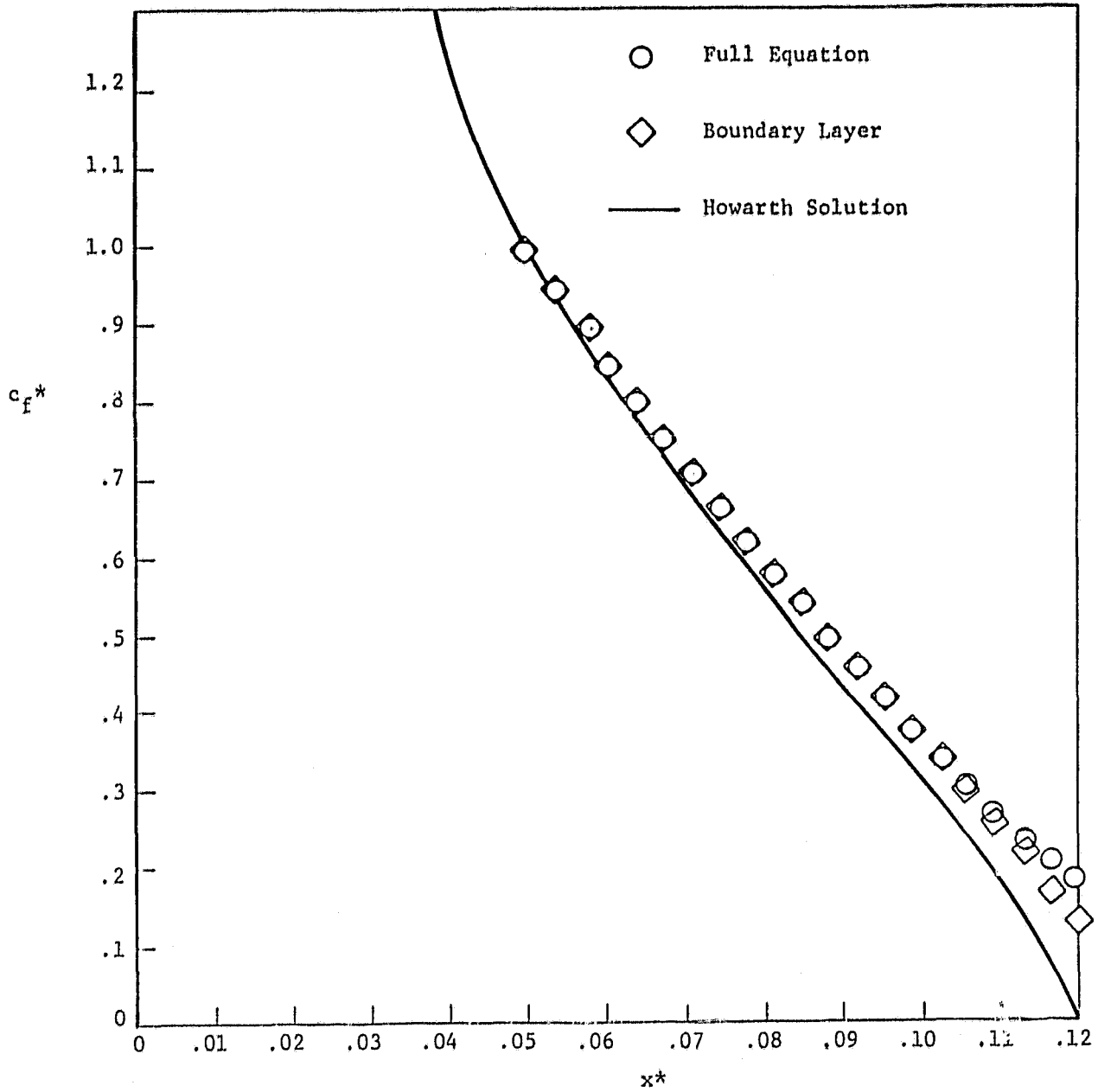


Fig. 7 - Skin Friction Distribution for Two-Dimensional Howarth Flow.