NASA Technical Memorandum 106337

# Proteus Three-Dimensional Navier-Stokes Computer Code–Version 1.0

## Volume 1–Analysis Description

Charles E. Towne, John R. Schwab, and Trong T. Bui
*Lewis Research Center*
*Cleveland, Ohio*

October 1993



NASA

## CONTENTS

# PRINCIPAL NOTATION

## SYMBOLS

Unless specified otherwise, all variables are nondimensional.

| Symbol | Definition |
|---|---|
| $a$ | Speed of sound. |
| $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ | Coefficient submatrices in block tridiagonal system of equations. |
| $\mathbf{A}'$, $\mathbf{B}'$, $\mathbf{C}'$ | Coefficient submatrices for boundary conditions. |
| $c_p$, $c_v$ | Specific heats at constant pressure and volume. |
| $\mathbf{E}$, $\mathbf{F}$, $\mathbf{G}$ | Inviscid flux vectors in the Cartesian coordinate form of the governing equations. |
| $\hat{\mathbf{E}}$, $\hat{\mathbf{F}}$, $\hat{\mathbf{G}}$ | Inviscid flux vectors in the computational coordinate form of the governing equations. |
| $E_T$ | Total energy per unit volume. |
| $\mathbf{E}_V$, $\mathbf{F}_V$, $\mathbf{G}_V$ | Viscous flux vectors in the Cartesian coordinate form of the governing equations. |
| $\hat{\mathbf{E}}_V$, $\hat{\mathbf{F}}_V$, $\hat{\mathbf{G}}_V$ | Viscous flux vectors in the computational coordinate form of the governing equations. |
| $\hat{\mathbf{E}}_{V_1}$, $\hat{\mathbf{F}}_{V_1}$, $\hat{\mathbf{G}}_{V_1}$ | Non-cross derivative viscous flux vectors in the computational coordinate form of the governing equations. |
| $\hat{\mathbf{E}}_{V_2}$, $\hat{\mathbf{F}}_{V_2}$, $\hat{\mathbf{G}}_{V_2}$ | Cross derivative viscous flux vectors in the computational coordinate form of the governing equations. |
| $\mathbf{F}$, $\mathbf{G}$, $\mathbf{H}$ | Flux vectors in the Cartesian coordinate form of the $k$-$\varepsilon$ turbulence model equations. |
| $\hat{\mathbf{F}}$, $\hat{\mathbf{G}}$, $\hat{\mathbf{H}}$ | Flux vectors in the computational coordinate form of the $k$-$\varepsilon$ turbulence model equations. |
| $h_T$ | Stagnation enthalpy per unit mass. |
| $i, j, k$ | Grid indices in the $\xi$, $\eta$, and $\zeta$ directions. |
| $J$ | Jacobian matrix of the generalized grid transformation. |
| $k$ | Effective thermal conductivity coefficient. |
| $k$ | Turbulent kinetic energy. |
| $k_l$, $k_t$ | Laminar and turbulent thermal conductivity coefficient. |
| $L_r$ | Dimensional reference length. |
| $N_{eq}$ | Number of governing equations being solved. |
| $N_1$, $N_2$, $N_3$ | Number of grid points in the $\xi$, $\eta$, and $\zeta$ directions. |
| $p$ | Static pressure. |
| $Pr_r$ | Reference Prandtl number. |
| $Pr_l$, $Pr_t$ | Laminar and turbulent Prandtl number. |
| $q_x$, $q_y$, $q_z$ | Heat fluxes in the Cartesian $x$, $y$, and $z$ directions. |

| Symbol | Definition |
|---|---|
| $\mathbf{Q}$ | Vector of dependent variables in the Cartesian coordinate form of the governing equations. |
| $\hat{\mathbf{Q}}$ | Vector of dependent variables in the computational coordinate form of the governing equations. |
| $R$ | Gas constant. |
| $Re_r$ | Reference Reynolds number. |
| $\mathbf{S}$ | Source term subvector in block tridiagonal system of equations. |
| $\mathbf{S}'$ | Source term subvector for boundary conditions. |
| $\mathbf{S}$, $\mathbf{T}$ | Non-derivative terms in the Cartesian coordinate form of the $k$-$\varepsilon$ turbulence model equations. |
| $\hat{\mathbf{S}}$, $\hat{\mathbf{T}}$ | Non-derivative terms in the computational coordinate form of the $k$-$\varepsilon$ turbulence model equations. |
| $t$ | Physical time. |
| $T$ | Static temperature. |
| $u$, $v$, $w$ | Velocities in the Cartesian $x$, $y$, and $z$ directions. |
| $\mathbf{W}$ | Vector of dependent variables in the Cartesian coordinate form of the $k$-$\varepsilon$ turbulence model equations. |
| $\hat{\mathbf{W}}$ | Vector of dependent variables in the computational coordinate form of the $k$-$\varepsilon$ turbulence model equations. |
| $x$, $y$, $z$ | Cartesian coordinates. |
| $\gamma$ | Ratio of specific heats, $c_p/c_v$. |
| $\delta$ | Difference operator. |
| $\Delta$, $\nabla$ | First-order forward and backward difference operators. |
| | Turbulent dissipation rate. |
| $\varepsilon_\xi^{(2)}$, $\varepsilon_\xi^{(4)}$ | Second- and fourth-order explicit artificial viscosity coefficients in constant coefficient model. |
| $\varepsilon_I$ | Implicit artificial viscosity coefficient. |
| $\varepsilon_\xi^{(2)}$, $\varepsilon_\xi^{(4)}$, etc. | Second- and fourth-order artificial viscosity coefficients in nonlinear coefficient model. |
| $\theta_1$, $\theta_2$, $\theta_3$ | Parameters determining type of time differencing used. |
| $\kappa_2$, $\kappa_4$ | Constants in nonlinear coefficient artificial viscosity model. |
| $\lambda$ | Effective second coefficient of viscosity. |
| $\lambda_l$, $\lambda_t$ | Laminar and turbulent second coefficient of viscosity. |
| $\mu$ | Effective viscosity coefficient. |
| $\mu_l$, $\mu_t$ | Laminar and turbulent viscosity coefficient. |
| $\nu$ | Laminar kinematic viscosity. |
| $\xi$, $\eta$, $\zeta$ | Computational coordinate directions. |
| $\rho$ | Static density. |
| $\sigma$ | Pressure gradient scaling parameter in nonlinear coefficient artificial viscosity model. |

| Symbol | Definition |
|---|---|
| $\tau$ | Computational time. |
| $\tau_{xx}$, $\tau_{xy}$, etc. | Elements of shear stress tensor. |
| $\psi$ | Spectral radius in nonlinear coefficient artificial viscosity model. |

## SUBSCRIPTS

| Subscript | Definition |
|---|---|
| $i, j, k$ | Denotes grid location in $\xi$, $\eta$, and $\zeta$ directions. |
| $r$ | Denotes dimensional reference condition. |
| $t$ | Denotes differentiation with respect to physical time. |
| $x, y, z$ | Denotes differentiation with respect to Cartesian coordinate directions. |
| $\xi, \eta, \zeta$ | Denotes differentiation with respect to computational coordinate directions. |
| $\tau$ | Denotes differentiation with respect to computational time. |

## SUPERSCRIPTS

| Superscript | Definition |
|---|---|
| $n$ | Denotes time level. |
| $*$, $**$ | Denotes solution after first and second ADI sweep. |

PAGE _____ INTENTIONALLY BLANK

# PROTEUS THREE-DIMENSIONAL NAVIER-STOKES COMPUTER CODE - VERSION 1.0

## Volume 1 - Analysis Description

Charles E. Towne, John R. Schwab, Trong T. Bui

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio

## SUMMARY

A computer code called *Proteus* has been developed to solve the three-dimensional, Reynolds-averaged, unsteady compressible Navier-Stokes equations in strong conservation law form. The objective in this effort has been to develop a code for aerospace propulsion applications that is easy to use and easy to modify. Code readability, modularity, and documentation have been emphasized.

The governing equations are written in Cartesian coordinates and transformed into generalized nonorthogonal body-fitted coordinates. They are solved by marching in time using a fully-coupled alternating-direction-implicit solution procedure with generalized first- or second-order time differencing. The boundary conditions are also treated implicitly, and may be steady or unsteady. Spatially periodic boundary conditions are also available. All terms, including the diffusion terms, are linearized using second-order Taylor series expansions. Turbulence is modeled using either an algebraic or two-equation eddy viscosity model.

The program contains many operating options. The thin-layer or Euler equations may be solved as subsets of the Navier-Stokes equations. The energy equation may be eliminated by the assumption of constant total enthalpy. Explicit and implicit artificial viscosity may be used to damp pre- and post-shock oscillations in supersonic flow and to minimize odd-even decoupling caused by central spatial differencing of the convective terms in high Reynolds number flow. Several time step options are available for convergence acceleration, including a locally variable time step and global time step cycling. Simple Cartesian or cylindrical grids may be generated internally by the program. More complex geometries require an externally generated computational coordinate system.

The documentation is divided into three volumes. Volume 1, the current volume, is the Analysis Description, and presents the equations and solution procedure used in *Proteus*. It describes in detail the governing equations, the turbulence model, the linearization of the equations and boundary conditions, the time and space differencing formulas, the ADI solution procedure, and the artificial viscosity models. Volume 2 is the User's Guide, and contains information needed to run the program. It describes the program's general features, the input and output, the procedure for setting up initial conditions, the computer resource requirements, the diagnostic messages that may be generated, the job control language used to run the program, and several test cases. Volume 3 is the Programmer's Reference, and contains detailed information useful when modifying the program. It describes the program structure, the Fortran variables stored in common blocks, and the details of each subprogram.

A two-dimensional/axisymmetric version of *Proteus* code also exists, and was originally released in late 1989.

PAGE _____ INTENTIONALLY BLANK

# 1.0 INTRODUCTION

Much of the effort in applied computational fluid dynamics consists of modifying an existing program for whatever geometries and flow regimes are of current interest to the researcher. Unfortunately, nearly all of the available non-proprietary programs were started as research projects with the emphasis on demonstrating the numerical algorithm rather than ease of use or ease of modification. The developers usually intend to clean up and formally document the program, but the immediate need to extend it to new geometries and flow regimes takes precedence.

The result is often a haphazard collection of poorly written code without any consistent structure. An extensively modified program may not even perform as expected under certain combinations of operating options. Each new user must invest considerable time and effort in attempting to understand the underlying structure of the program if intending to do anything more than run standard test cases with it. The user's subsequent modifications further obscure the program structure and therefore make it even more difficult for others to understand.

The *Proteus* three-dimensional Navier-Stokes computer program is a user-oriented and easily-modifiable flow analysis program for aerospace propulsion applications. Readability, modularity, and documentation were primary objectives during its development. The entire program was specified, designed, and implemented in a controlled, systematic manner. Strict programming standards were enforced by immediate peer review of code modules; Kernighan and Plauger (1978) provided many useful ideas about consistent programming style. Every subroutine contains an extensive comment section describing the purpose, input variables, output variables, and calling sequence of the subroutine. With just three clearly-defined exceptions, the entire program is written in ANSI standard Fortran 77 to enhance portability. A master version of the program is maintained and periodically updated with corrections, as well as extensions of general interest (e.g., turbulence models.)

The *Proteus* program solves the unsteady, compressible, Reynolds-averaged Navier-Stokes equations in strong conservation law form. The governing equations are written in Cartesian coordinates and transformed into generalized nonorthogonal body-fitted coordinates. They are solved by marching in time using a fully-coupled alternating-direction-implicit (ADI) scheme with generalized time and space differencing (Briley and McDonald, 1977; Beam and Warming, 1978). Turbulence is modeled using either the Baldwin and Lomax (1978) algebraic eddy-viscosity model or the Chien (1982) two-equation model. All terms, including the diffusion terms, are linearized using second-order Taylor series expansions. The boundary conditions are treated implicitly, and may be steady or unsteady. Spatially periodic boundary conditions are also available.

The program contains many operating options. The thin-layer or Euler equations may be solved as subsets of the Navier-Stokes equations. The energy equation may be eliminated by the assumption of constant total enthalpy. Explicit and implicit artificial viscosity may be used to damp pre- and post-shock oscillations in supersonic flow and to minimize odd-even decoupling caused by central spatial differencing of the convective terms in high Reynolds number flow. Several time step options are available for convergence acceleration, including a locally variable time step and global time step cycling. Simple grids may be generated internally by the program; more complex geometries require external grid generation, such as that developed by Chen and Schwab (1988).

The documentation is divided into three volumes. Volume 1, the current volume, is the Analysis Description, and presents the equations and solution procedure used in *Proteus*. It describes in detail the governing equations, the turbulence model, the linearization of the equations and boundary conditions, the time and space differencing formulas, the ADI solution procedure, and the artificial viscosity models. Volume 2 is the User's Guide, and contains information needed to run the program. It describes the program's general features, the input and output, the procedure for setting up initial conditions, the computer resource requirements, the diagnostic messages that may be generated, the job control language used to run the

program, and several test cases. Volume 3 is the Programmer's Reference, and contains detailed information useful when modifying the program. It describes the program structure, the Fortran variables stored in common blocks, and the details of each subprogram.

A two-dimensional/axisymmetric version of *Proteus* code also exists, and was originally released in late 1989 (Towne, Schwab, Benson, and Suresh, 1990).

The authors would like to acknowledge the significant contributions made by their co-workers. Tom Benson provided part of the original impetus for the development of *Proteus*, and did the original coding of the block tri-diagonal inversion routines. Simon Chen did the original coding of the Baldwin-Lomax turbulence model, and consulted in the implementation of the nonlinear coefficient artificial viscosity model. William Kunik developed the original code for computing the metrics of the generalized nonorthogonal grid transformation. Frank Molls has created a separate diagonalized version of the code. Ambady Suresh did the original coding for the second-order time differencing and for the nonlinear coefficient artificial viscosity model. These people, along with Dick Cavicchi, Julie Conley, Jason Solbeck, and Pat Zeman, have also run many debugging and verification cases.

# 2.0 GOVERNING EQUATIONS

## 2.1 GOVERNING EQUATIONS IN CARTESIAN COORDINATES

The basic governing equations are the three-dimensional compressible Navier-Stokes equations. These equations may be found in several standard references (e.g., Hughes and Gaylord, 1964; Schlichting, 1968; White, 1974; Anderson, Tannehill, and Pletcher, 1984.) In Cartesian coordinates, the three-dimensional equations can be written in strong conservation law form using vector notation as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} + \frac{\partial \mathbf{G}}{\partial z} = \frac{\partial \mathbf{E}_V}{\partial x} + \frac{\partial \mathbf{F}_V}{\partial y} + \frac{\partial \mathbf{G}_V}{\partial z} \tag{2.1}$$

where

$$\mathbf{Q} = [\rho \quad \rho u \quad \rho v \quad \rho w \quad E_T]^T \tag{2.2a}$$

$$\mathbf{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (E_T + p)u \end{bmatrix} \tag{2.2b}$$

$$\mathbf{F} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (E_T + p)v \end{bmatrix} \tag{2.2c}$$

$$\mathbf{G} = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (E_T + p)w \end{bmatrix} \tag{2.2d}$$

$$\mathbf{E}_V = \frac{1}{Re_r} \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - \frac{1}{Pr_r} q_x \end{bmatrix} \tag{2.2e}$$

$$\mathbf{F}_V = \frac{1}{Re_r}\begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{xy} + v\tau_{yy} + w\tau_{yz} - \dfrac{1}{Pr_r} q_y \end{bmatrix} \tag{2.2f}$$

$$\mathbf{G}_V = \frac{1}{Re_r}\begin{bmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{xz} + v\tau_{yz} + w\tau_{zz} - \dfrac{1}{Pr_r} q_z \end{bmatrix} \tag{2.2g}$$

Equation (2.1) thus represents, in order, the continuity, $x$-momentum, $y$-momentum, $z$-momentum, and energy equations, with dependent variables $\rho$, $\rho u$, $\rho v$, $\rho w$, and $E_T$.

The shear stresses and heat fluxes are given by

$$\begin{aligned}
\tau_{xx} &= 2\mu \frac{\partial u}{\partial x} + \lambda\left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \\
\tau_{yy} &= 2\mu \frac{\partial v}{\partial y} + \lambda\left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \\
\tau_{zz} &= 2\mu \frac{\partial w}{\partial z} + \lambda\left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \\
\tau_{xy} &= \mu\left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\
\tau_{xz} &= \mu\left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \\
\tau_{yz} &= \mu\left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \\
q_x &= -k \frac{\partial T}{\partial x} \\
q_y &= -k \frac{\partial T}{\partial y} \\
q_z &= -k \frac{\partial T}{\partial z}
\end{aligned} \tag{2.3}$$

In these equations, $t$ represents time; $x$, $y$, and $z$ represent the Cartesian coordinate directions; $u$, $v$, and $w$ are the velocities in the $x$, $y$, and $z$ directions; $\rho$, $p$, and $T$ are the static density, pressure, and temperature; $E_T$ is the total energy per unit volume; and $\mu$, $\lambda$, and $k$ are the coefficient of viscosity, second coefficient of viscosity, and coefficient of thermal conductivity.

All of the above equations have been nondimensionalized using appropriate normalizing conditions. Lengths have been nondimensionalized by $L_r$, velocities by $u_r$, density by $\rho_r$, temperature by $T_r$, viscosity

by $\mu_r$, thermal conductivity by $k_r$, pressure and total energy by $\rho_r u_r^2$, and time by $L_r/u_r$. The reference Reynolds and Prandtl numbers are thus defined as $Re_r = \rho_r u_r L_r/\mu_r$ and $Pr_r = \mu_r u_r^2/k_r T_r$.[1]

Turbulence is modeled using the Boussinesq approach (Schlichting, 1968). The equations presented in this section are thus used for both laminar and turbulent flow. For turbulent flow they represent the Reynolds time-averaged form of the Navier-Stokes equations, with density fluctuations neglected. They may also be interpreted as the Favre or mass-weighted time-averaged form of the equations. With Favre time averaging, however, the velocities and thermal variables represent mass-averaged quantities defined by $\tilde{u} = \overline{\rho u}/\overline{\rho}$, etc., where the overbar represents a conventional Reynolds time-averaged quantity. Details on Reynolds and Favre time-averaging procedures may be found in Cebeci and Smith (1974), and in Anderson, Tannehill, and Pletcher (1984). In either case, $\mu$, $\lambda$, and $k$ represent *effective* coefficients. For example, in turbulent flow $\mu = \mu_l + \mu_t$, where $\mu_l$ and $\mu_t$ are the laminar and turbulent viscosity coefficients, and $\mu_t$ comes from some appropriate turbulence model. The models currently available in the *Proteus* code are the algebraic eddy viscosity model of Baldwin and Lomax (1978) and the two-equation model of Chien (1982), implemented as described in Section 9.0.

## 2.2 EQUATION OF STATE

In addition to the equations presented above, an equation of state is required to relate pressure to the dependent variables. Any appropriate equation, or even table, could be used. The equation currently built into the *Proteus* code is the equation of state for thermally perfect gases, $p = \rho RT$, where $R$ is the gas constant. For calorically perfect gases, this can be rewritten as

$$p = (\gamma - 1)\left[E_T - \frac{1}{2}\rho(u^2 + v^2 + w^2)\right] \tag{2.4}$$

where $\gamma$ is the ratio of specific heats, $c_p/c_v$. Here the gas constant and specific heats have been nondimensionalized by $u_r^2/T_r$.

If the flow is such that we can assume a perfect gas with constant stagnation enthalpy, the energy equation may be eliminated. This assumption is reasonable, for example, in inviscid regions, and in adiabatic wall boundary layers if the Prandtl number is near 1 (Briley and McDonald, 1977). The stagnation enthalpy is defined as

$$h_T = c_p T + \frac{1}{2}(u^2 + v^2 + w^2) \tag{2.5}$$

Here the stagnation enthalpy is nondimensionalized by $u_r^2$. The temperature is thus

$$T = \frac{1}{c_p}\left[h_T - \frac{1}{2}(u^2 + v^2 + w^2)\right] \tag{2.6}$$

and the equation of state becomes

$$p = \frac{\gamma - 1}{\gamma}\rho\left[h_T - \frac{1}{2}(u^2 + v^2 + w^2)\right] \tag{2.7}$$

This equation of state does not require the total energy $E_T$, and the energy equation need not be solved. The total energy may be computed from

$$E_T = \rho h_T - p \tag{2.8}$$

## 2.3 GENERALIZED GRID TRANSFORMATION

Because the governing equations in the previous section are written in Cartesian coordinates, they are not well suited for general geometric configurations. For most applications a body-fitted coordinate system

---

[1] Note that this Prandtl number does not have a physically meaningful value, but is merely defined by a combination of the normalizing conditions for $c_p$, $\mu$, and $k$ that appear when the equations are nondimensionalized.

is desired. This greatly simplifies the application of boundary conditions and the bookkeeping in the numerical method used to solve the equations. The following generalized grid transformation, which can be orthogonal or nonorthogonal, is therefore used to transform the governing equations from physical $(x, y, z, t)$ coordinates to computational $(\xi, \eta, \zeta, \tau)$ coordinates.

$$\begin{aligned} \xi &= \xi(x, y, z, t) \\ \eta &= \eta(x, y, z, t) \\ \zeta &= \zeta(x, y, z, t) \\ \tau &= t \end{aligned} \tag{2.9}$$

In *Proteus*, the spatial computational domain is a cube, with $\xi$, $\eta$, and $\zeta$ each running from 0 to 1. Using the chain rule for partial differentiation, the derivatives in the Cartesian form of the governing equations can be replaced using the following expressions.

$$\begin{aligned} \frac{\partial}{\partial x} &= \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} + \zeta_x \frac{\partial}{\partial \zeta} \\ \frac{\partial}{\partial y} &= \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta} + \zeta_y \frac{\partial}{\partial \zeta} \\ \frac{\partial}{\partial z} &= \xi_z \frac{\partial}{\partial \xi} + \eta_z \frac{\partial}{\partial \eta} + \zeta_z \frac{\partial}{\partial \zeta} \\ \frac{\partial}{\partial t} &= \xi_t \frac{\partial}{\partial \xi} + \eta_t \frac{\partial}{\partial \eta} + \zeta_t \frac{\partial}{\partial \zeta} + \frac{\partial}{\partial \tau} \end{aligned} \tag{2.10}$$

In the above equations, and in those to follow, subscripts $x$, $y$, and $z$, or $\xi$, $\eta$, and $\zeta$, denote partial differentiation in that coordinate direction. The only task remaining, then, is to develop expressions for the metric coefficients $\xi_x$, $\eta_x$, etc. In differential form we can write

$$\begin{aligned} d\xi &= \xi_x dx + \xi_y dy + \xi_z dz + \xi_t dt \\ d\eta &= \eta_x dx + \eta_y dy + \eta_z dz + \eta_t dt \\ d\zeta &= \zeta_x dx + \zeta_y dy + \zeta_z dz + \zeta_t dt \\ d\tau &= dt \end{aligned}$$

In matrix form this becomes

$$\begin{bmatrix} d\xi \\ d\eta \\ d\zeta \\ d\tau \end{bmatrix} = \begin{bmatrix} \xi_x & \xi_y & \xi_z & \xi_t \\ \eta_x & \eta_y & \eta_z & \eta_t \\ \zeta_x & \zeta_y & \zeta_z & \zeta_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \\ dt \end{bmatrix}$$

Similarly,

$$\begin{bmatrix} dx \\ dy \\ dz \\ dt \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta & x_\zeta & x_\tau \\ y_\xi & y_\eta & y_\zeta & y_\tau \\ z_\xi & z_\eta & z_\zeta & z_\tau \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \\ d\zeta \\ d\tau \end{bmatrix}$$

Therefore,

$$\begin{bmatrix} \xi_x & \xi_y & \xi_z & \xi_t \\ \eta_x & \eta_y & \eta_z & \eta_t \\ \zeta_x & \zeta_y & \zeta_z & \zeta_t \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta & x_\zeta & x_\tau \\ y_\xi & y_\eta & y_\zeta & y_\tau \\ z_\xi & z_\eta & z_\zeta & z_\tau \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1}$$

After taking the inverse,

$$\begin{bmatrix} \xi_x & \xi_y & \xi_z & \xi_t \\ \eta_x & \eta_y & \eta_z & \eta_t \\ \zeta_x & \zeta_y & \zeta_z & \zeta_t \\ 0 & 0 & 0 & 1 \end{bmatrix} = J \begin{bmatrix} y_\eta z_\zeta - y_\zeta z_\eta & x_\zeta z_\eta - x_\eta z_\zeta & x_\eta y_\zeta - x_\zeta y_\eta & F_{14} \\ y_\zeta z_\xi - y_\xi z_\zeta & x_\xi z_\zeta - x_\zeta z_\xi & x_\zeta y_\xi - x_\xi y_\zeta & F_{24} \\ y_\xi z_\eta - y_\eta z_\xi & x_\eta z_\xi - x_\xi z_\eta & x_\xi y_\eta - x_\eta y_\xi & F_{34} \\ 0 & 0 & 0 & 1/J \end{bmatrix}$$

where

$$\begin{aligned} F_{14} &= x_\tau(y_\zeta z_\eta - y_\eta z_\zeta) + y_\tau(x_\eta z_\zeta - x_\zeta z_\eta) + z_\tau(x_\zeta y_\eta - x_\eta y_\zeta) \\ F_{24} &= x_\tau(y_\xi z_\zeta - y_\zeta z_\xi) + y_\tau(x_\zeta z_\xi - x_\xi z_\zeta) + z_\tau(x_\xi y_\zeta - x_\zeta y_\xi) \\ F_{34} &= x_\tau(y_\eta z_\xi - y_\xi z_\eta) + y_\tau(x_\xi z_\eta - x_\eta z_\xi) + z_\tau(x_\eta y_\xi - x_\xi y_\eta) \end{aligned}$$

and $J$ is the Jacobian of the transformation,

$$J = \frac{\partial(\xi, \eta, \zeta)}{\partial(x,y,z)} = \begin{vmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{vmatrix}$$

$$J = \xi_x(\eta_y\zeta_z - \eta_z\zeta_y) + \xi_y(\eta_z\zeta_x - \eta_x\zeta_z) + \xi_z(\eta_x\zeta_y - \eta_y\zeta_x) \tag{2.11}$$

This can be evaluated from the known physical $(x, y, z)$ coordinates by noting $J = 1/J^{-1}$ and

$$J^{-1} = \frac{\partial(x,y,z)}{\partial(\xi, \eta, \zeta)} = \begin{vmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{vmatrix}$$

$$J^{-1} = x_\xi(y_\eta z_\zeta - y_\zeta z_\eta) + x_\eta(y_\zeta z_\xi - y_\xi z_\zeta) + x_\zeta(y_\xi z_\eta - y_\eta z_\xi) \tag{2.12}$$

The metric coefficients themselves are

$$\begin{aligned} \xi_x &= J(y_\eta z_\zeta - y_\zeta z_\eta) \\ \xi_y &= J(x_\zeta z_\eta - x_\eta z_\zeta) \\ \xi_z &= J(x_\eta y_\zeta - x_\zeta y_\eta) \\ \eta_x &= J(y_\zeta z_\xi - y_\xi z_\zeta) \\ \eta_y &= J(x_\xi z_\zeta - x_\zeta z_\xi) \\ \eta_z &= J(x_\zeta y_\xi - x_\xi y_\zeta) \\ \zeta_x &= J(y_\xi z_\eta - y_\eta z_\xi) \end{aligned} \tag{2.13}$$

$$\zeta_y = J(x_\eta z_\xi - x_\xi z_\eta)$$

$$\zeta_z = J(x_\xi y_\eta - x_\eta y_\xi)$$

$$\xi_t = -x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z$$

$$\eta_t = -x_\tau \eta_x - y_\tau \eta_y - z_\tau \eta_z$$

$$\zeta_t = -x_\tau \zeta_x - y_\tau \zeta_y - z_\tau \zeta_z$$

Unless the physical coordinates ($x$, $y$, $z$) are defined analytically as functions of the computational coordinates ($\xi$, $\eta$, $\zeta$), the metric coefficients must be computed numerically. The method used to do this is important, and is discussed in Section 2.5.

## 2.4 GOVERNING EQUATIONS IN COMPUTATIONAL COORDINATES

Applying the generalized grid transformation of the previous section to equation (2.1) yields

$$\mathbf{Q}_\tau + \mathbf{Q}_\xi \xi_t + \mathbf{Q}_\eta \eta_t + \mathbf{Q}_\zeta \zeta_t + \mathbf{E}_\xi \xi_x + \mathbf{E}_\eta \eta_x + \mathbf{E}_\zeta \zeta_x + \mathbf{F}_\xi \xi_y + \mathbf{F}_\eta \eta_y + \mathbf{F}_\zeta \zeta_y + \mathbf{G}_\xi \xi_z + \mathbf{G}_\eta \eta_z + \mathbf{G}_\zeta \zeta_z$$
$$- \mathbf{E}_{V_\xi} \xi_x - \mathbf{E}_{V_\eta} \eta_x - \mathbf{E}_{V_\zeta} \zeta_x - \mathbf{F}_{V_\xi} \xi_y - \mathbf{F}_{V_\eta} \eta_y - \mathbf{F}_{V_\zeta} \zeta_y - \mathbf{G}_{V_\xi} \xi_z - \mathbf{G}_{V_\eta} \eta_z - \mathbf{G}_{V_\zeta} \zeta_z = 0 \qquad (2.14)$$

This equation is in chain-rule, or weakly conservative form. That is, the conservation flow variables are used, but the metrics appear as coefficients of the derivatives instead of inside the derivatives. Following Vinokur (1974), the strong conservation law form can be recovered by first dividing by the Jacobian then adding and subtracting like terms. For example, the

$$\mathbf{E}_\xi \xi_x$$

term becomes

$$\frac{\mathbf{E}_\xi \xi_x}{J} = \left[ \frac{\mathbf{E} \xi_x}{J} \right]_\xi - \mathbf{E} \left( \frac{\xi_x}{J} \right)_\xi$$

Doing this for all the terms, and rearranging, results in

$$\left(\frac{\mathbf{Q}}{J}\right)_\tau + \left[ \frac{\mathbf{E}\xi_x + \mathbf{F}\xi_y + \mathbf{G}\xi_z + \mathbf{Q}\xi_t}{J} \right]_\xi + \left[ \frac{\mathbf{E}\eta_x + \mathbf{F}\eta_y + \mathbf{G}\eta_z + \mathbf{Q}\eta_t}{J} \right]_\eta + \left[ \frac{\mathbf{E}\zeta_x + \mathbf{F}\zeta_y + \mathbf{G}\zeta_z + \mathbf{Q}\zeta_t}{J} \right]_\zeta$$
$$- \left[ \frac{\mathbf{E}_V \xi_x + \mathbf{F}_V \xi_y + \mathbf{G}_V \xi_z}{J} \right]_\xi - \left[ \frac{\mathbf{E}_V \eta_x + \mathbf{F}_V \eta_y + \mathbf{G}_V \eta_z}{J} \right]_\eta - \left[ \frac{\mathbf{E}_V \zeta_x + \mathbf{F}_V \zeta_y + \mathbf{G}_V \zeta_z}{J} \right]_\zeta$$
$$- \mathbf{Q} \left\{ \left(\frac{1}{J}\right)_\tau + \left(\frac{\xi_t}{J}\right)_\xi + \left(\frac{\eta_t}{J}\right)_\eta + \left(\frac{\zeta_t}{J}\right)_\zeta \right\} - (\mathbf{E} - \mathbf{E}_V) \left\{ \left(\frac{\xi_x}{J}\right)_\xi + \left(\frac{\eta_x}{J}\right)_\eta + \left(\frac{\zeta_x}{J}\right)_\zeta \right\}$$
$$- (\mathbf{F} - \mathbf{F}_V) \left\{ \left(\frac{\xi_y}{J}\right)_\xi + \left(\frac{\eta_y}{J}\right)_\eta + \left(\frac{\zeta_y}{J}\right)_\zeta \right\} - (\mathbf{G} - \mathbf{G}_V) \left\{ \left(\frac{\xi_z}{J}\right)_\xi + \left(\frac{\eta_z}{J}\right)_\eta + \left(\frac{\zeta_z}{J}\right)_\zeta \right\} = 0 \qquad (2.15)$$

The last four terms, in braces, are called the metric invariant terms. By using the expressions for the metric coefficients, given by equations (2.13), one can show that the metric invariants are identically zero. This is not necessarily true when derivatives are approximated by finite differences, however. This point is explored further in Section 2.5. With the metric invariant terms eliminated, no metrics or flow variables appear as coefficients, and the strong conservation law form of the governing equations has been recovered.

Equation (2.15) can be rewritten as

$$\frac{\partial \hat{\mathbf{Q}}}{\partial \tau} + \frac{\partial \hat{\mathbf{E}}}{\partial \xi} + \frac{\partial \hat{\mathbf{F}}}{\partial \eta} + \frac{\partial \hat{\mathbf{G}}}{\partial \zeta} = \frac{\partial \hat{\mathbf{E}}_V}{\partial \xi} + \frac{\partial \hat{\mathbf{F}}_V}{\partial \eta} + \frac{\partial \hat{\mathbf{G}}_V}{\partial \zeta} \qquad (2.16)$$

where

$$\hat{\mathbf{Q}} = \frac{\mathbf{Q}}{J}$$

$$\hat{\mathbf{E}} = \frac{1}{J}(\mathbf{E}\xi_x + \mathbf{F}\xi_y + \mathbf{G}\xi_z + \mathbf{Q}\xi_t)$$

$$\hat{\mathbf{F}} = \frac{1}{J}(\mathbf{E}\eta_x + \mathbf{F}\eta_y + \mathbf{G}\eta_z + \mathbf{Q}\eta_t)$$

$$\hat{\mathbf{G}} = \frac{1}{J}(\mathbf{E}\zeta_x + \mathbf{F}\zeta_y + \mathbf{G}\zeta_z + \mathbf{Q}\zeta_t)$$

$$\hat{\mathbf{E}}_V = \frac{1}{J}(\mathbf{E}_V\xi_x + \mathbf{F}_V\xi_y + \mathbf{G}_V\xi_z)$$

$$\hat{\mathbf{F}}_V = \frac{1}{J}(\mathbf{E}_V\eta_x + \mathbf{F}_V\eta_y + \mathbf{G}_V\eta_z)$$

$$\hat{\mathbf{G}}_V = \frac{1}{J}(\mathbf{E}_V\zeta_x + \mathbf{F}_V\zeta_y + \mathbf{G}_V\zeta_z)$$

Using equations (2.2a) through (2.2g) these can be expanded as

$$\hat{\mathbf{Q}} = \frac{1}{J}[\rho \quad \rho u \quad \rho v \quad \rho w \quad E_T]^T \tag{2.17a}$$

$$\hat{\mathbf{E}} = \frac{1}{J}\begin{bmatrix} \rho u\xi_x + \rho v\xi_y + \rho w\xi_z + \rho\xi_t \\ (\rho u^2 + p)\xi_x + \rho uv\xi_y + \rho uw\xi_z + \rho u\xi_t \\ \rho uv\xi_x + (\rho v^2 + p)\xi_y + \rho vw\xi_z + \rho v\xi_t \\ \rho uw\xi_x + \rho vw\xi_y + (\rho w^2 + p)\xi_z + \rho w\xi_t \\ (E_T + p)u\xi_x + (E_T + p)v\xi_y + (E_T + p)w\xi_z + E_T\xi_t \end{bmatrix} \tag{2.17b}$$

$$\hat{\mathbf{F}} = \frac{1}{J}\begin{bmatrix} \rho u\eta_x + \rho v\eta_y + \rho w\eta_z + \rho\eta_t \\ (\rho u^2 + p)\eta_x + \rho uv\eta_y + \rho uw\eta_z + \rho u\eta_t \\ \rho uv\eta_x + (\rho v^2 + p)\eta_y + \rho vw\eta_z + \rho v\eta_t \\ \rho uw\eta_x + \rho vw\eta_y + (\rho w^2 + p)\eta_z + \rho w\eta_t \\ (E_T + p)u\eta_x + (E_T + p)v\eta_y + (E_T + p)w\eta_z + E_T\eta_t \end{bmatrix} \tag{2.17c}$$

$$\hat{\mathbf{G}} = \frac{1}{J}\begin{bmatrix} \rho u\zeta_x + \rho v\zeta_y + \rho w\zeta_z + \rho\zeta_t \\ (\rho u^2 + p)\zeta_x + \rho uv\zeta_y + \rho uw\zeta_z + \rho u\zeta_t \\ \rho uv\zeta_x + (\rho v^2 + p)\zeta_y + \rho vw\zeta_z + \rho v\zeta_t \\ \rho uw\zeta_x + \rho vw\zeta_y + (\rho w^2 + p)\zeta_z + \rho w\zeta_t \\ (E_T + p)u\zeta_x + (E_T + p)v\zeta_y + (E_T + p)w\zeta_z + E_T\zeta_t \end{bmatrix} \tag{2.17d}$$

$$\hat{\mathbf{E}}_V = \frac{1}{J}\frac{1}{Re_r}\begin{bmatrix} 0 \\ \tau_{xx}\xi_x + \tau_{xy}\xi_y + \tau_{xz}\xi_z \\ \tau_{xy}\xi_x + \tau_{yy}\xi_y + \tau_{yz}\xi_z \\ \tau_{xz}\xi_x + \tau_{yz}\xi_y + \tau_{zz}\xi_z \\ \beta_x\xi_x + \beta_y\xi_y + \beta_z\xi_z \end{bmatrix} \tag{2.17e}$$

$$\hat{\mathbf{F}}_V = \frac{1}{J}\frac{1}{Re_r}\begin{bmatrix} 0 \\ \tau_{xx}\eta_x + \tau_{xy}\eta_y + \tau_{xz}\eta_z \\ \tau_{xy}\eta_x + \tau_{yy}\eta_y + \tau_{yz}\eta_z \\ \tau_{xz}\eta_x + \tau_{yz}\eta_y + \tau_{zz}\eta_z \\ \beta_x\eta_x + \beta_y\eta_y + \beta_z\eta_z \end{bmatrix} \tag{2.17f}$$

$$\hat{\mathbf{G}}_V = \frac{1}{J}\frac{1}{Re_r}\begin{bmatrix} 0 \\ \tau_{xx}\zeta_x + \tau_{xy}\zeta_y + \tau_{xz}\zeta_z \\ \tau_{xy}\zeta_x + \tau_{yy}\zeta_y + \tau_{yz}\zeta_z \\ \tau_{xz}\zeta_x + \tau_{yz}\zeta_y + \tau_{zz}\zeta_z \\ \beta_x\zeta_x + \beta_y\zeta_y + \beta_z\zeta_z \end{bmatrix} \tag{2.17g}$$

where

$$\beta_x = u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - \frac{1}{Pr_r}q_x$$

$$\beta_y = u\tau_{xy} + v\tau_{yy} + w\tau_{yz} - \frac{1}{Pr_r}q_y$$

$$\beta_z = u\tau_{xz} + v\tau_{yz} + w\tau_{zz} - \frac{1}{Pr_r}q_z$$

In the viscous terms, the shear stresses and heat fluxes are defined exactly as in equations (2.3), except the derivatives in the Cartesian coordinate directions must be evaluated using the chain rule. For example,

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial \xi}\xi_x + \frac{\partial u}{\partial \eta}\eta_x + \frac{\partial u}{\partial \zeta}\zeta_x$$

Note that $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$ have exactly the same form as $\hat{\mathbf{E}}$, but with $\xi$ replaced by $\eta$ and $\zeta$, respectively. Similarly, $\hat{\mathbf{F}}_V$ and $\hat{\mathbf{G}}_V$ have exactly the same form as $\hat{\mathbf{E}}_V$, but with $\xi$ replaced by $\eta$ and $\zeta$, respectively.

## 2.5 METRIC INVARIANTS

The governing differential equation in computational coordinates, equation (2.16), can be rewritten as

$$\left(\frac{\mathbf{Q}}{J}\right)_\tau + \left[\frac{\mathbf{E}\xi_x + \mathbf{F}\xi_y + \mathbf{G}\xi_z + \mathbf{Q}\xi_t}{J}\right]_\xi + \left[\frac{\mathbf{E}\eta_x + \mathbf{F}\eta_y + \mathbf{G}\eta_z + \mathbf{Q}\eta_t}{J}\right]_\eta + \left[\frac{\mathbf{E}\zeta_x + \mathbf{F}\zeta_y + \mathbf{G}\zeta_z + \mathbf{Q}\zeta_t}{J}\right]_\zeta$$
$$- \left[\frac{\mathbf{E}_V\xi_x + \mathbf{F}_V\xi_y + \mathbf{G}_V\xi_z}{J}\right]_\xi - \left[\frac{\mathbf{E}_V\eta_x + \mathbf{F}_V\eta_y + \mathbf{G}_V\eta_z}{J}\right]_\eta - \left[\frac{\mathbf{E}_V\zeta_x + \mathbf{F}_V\zeta_y + \mathbf{G}_V\zeta_z}{J}\right]_\zeta = 0$$

When this equation is applied to uniform flow, **E**, **F**, **G**, etc., are all constant, resulting in

$$\frac{1}{J}\mathbf{Q}_\tau + \left(\frac{1}{J}\right)_\tau \mathbf{Q} + (\mathbf{E} - \mathbf{E}_V)\left(\frac{\xi_x}{J}\right)_\xi + (\mathbf{F} - \mathbf{F}_V)\left(\frac{\xi_y}{J}\right)_\xi + (\mathbf{G} - \mathbf{G}_V)\left(\frac{\xi_z}{J}\right)_\xi + \mathbf{Q}\left(\frac{\xi_t}{J}\right)_\xi$$
$$+ (\mathbf{E} - \mathbf{E}_V)\left(\frac{\eta_x}{J}\right)_\eta + (\mathbf{F} - \mathbf{F}_V)\left(\frac{\eta_y}{J}\right)_\eta + (\mathbf{G} - \mathbf{G}_V)\left(\frac{\eta_z}{J}\right)_\eta + \mathbf{Q}\left(\frac{\eta_t}{J}\right)_\eta$$
$$+ (\mathbf{E} - \mathbf{E}_V)\left(\frac{\zeta_x}{J}\right)_\zeta + (\mathbf{F} - \mathbf{F}_V)\left(\frac{\zeta_y}{J}\right)_\zeta + (\mathbf{G} - \mathbf{G}_V)\left(\frac{\zeta_z}{J}\right)_\zeta + \mathbf{Q}\left(\frac{\zeta_t}{J}\right)_\zeta = 0$$

Collecting terms,

$$\frac{1}{J}\mathbf{Q}_\tau + \mathbf{Q}\left[\left(\frac{1}{J}\right)_\tau + \left(\frac{\xi_t}{J}\right)_\xi + \left(\frac{\eta_t}{J}\right)_\eta + \left(\frac{\zeta_t}{J}\right)_\zeta\right]$$
$$+ (\mathbf{E} - \mathbf{E}_V)\left[\left(\frac{\xi_x}{J}\right)_\xi + \left(\frac{\eta_x}{J}\right)_\eta + \left(\frac{\zeta_x}{J}\right)_\zeta\right]$$
$$+ (\mathbf{F} - \mathbf{F}_V)\left[\left(\frac{\xi_y}{J}\right)_\xi + \left(\frac{\eta_y}{J}\right)_\eta + \left(\frac{\zeta_y}{J}\right)_\zeta\right]$$
$$+ (\mathbf{G} - \mathbf{G}_V)\left[\left(\frac{\xi_z}{J}\right)_\xi + \left(\frac{\eta_z}{J}\right)_\eta + \left(\frac{\zeta_z}{J}\right)_\zeta\right] = 0 \qquad (2.18)$$

For $\mathbf{Q}_\tau$ to be zero, which it should be for uniform flow, the terms in brackets must vanish. These terms are the metric invariants discussed briefly in Section 2.4. By using the expressions for the metric coefficients given by equations (2.13), one can show that in differential form these metric invariants are indeed identically zero. When finite differences are used to approximate derivatives, however, this is not necessarily true. In two dimensions, when the centered difference formula of equation (5.1) is used to approximate derivatives, the metric invariants do turn out to be identically zero. But in three dimensions, when the metric coefficients are computed numerically using equations (2.13), the metric invariants are not identically zero.

To show this, let $M_2$ denote the second metric invariant term (the second bracketed term in equation (2.18).) Then, using equations (2.13) for the metric coefficients and applying difference operators,

$$M_2 \simeq \delta_\xi(\delta_\eta y\, \delta_\zeta z - \delta_\zeta y\, \delta_\eta z) + \delta_\eta(\delta_\zeta y\, \delta_\xi z - \delta_\xi y\, \delta_\zeta z) + \delta_\zeta(\delta_\xi y\, \delta_\eta z - \delta_\eta y\, \delta_\xi z)$$

Without loss of generality we can let $\Delta\xi = \Delta\eta = \Delta\zeta = 1/2$. Then, using central differences,

$$\begin{aligned} M_2 \simeq\ & \delta_\xi[(y_{j+1} - y_{j-1})(z_{k+1} - z_{k-1}) - (y_{k+1} - y_{k-1})(z_{j+1} - z_{j-1})] \\ & + \delta_\eta[(y_{k+1} - y_{k-1})(z_{i+1} - z_{i-1}) - (y_{i+1} - y_{i-1})(z_{k+1} - z_{k-1})] \\ & + \delta_\zeta[(y_{i+1} - y_{i-1})(z_{j+1} - z_{j-1}) - (y_{j+1} - y_{j-1})(z_{i+1} - z_{i-1})] \end{aligned}$$

The subscripts $i$, $j$, and $k$ represent grid point indices in the $\xi$, $\eta$, and $\zeta$ directions. For notational convenience, terms without an explicitly written $i$, $j$, or $k$ subscript are understood to be at $i$, $j$, or $k$. Expanding,

$$\begin{aligned} M_2 \simeq\ & \delta_\xi[(y_{j+1}z_{k+1} - y_{j-1}z_{k+1} - y_{j+1}z_{k-1} + y_{j-1}z_{k-1}) - (y_{k+1}z_{j+1} - y_{k-1}z_{j+1} - y_{k+1}z_{j-1} + y_{k-1}z_{j-1})] \\ & + \delta_\eta[(y_{k+1}z_{i+1} - y_{k-1}z_{i+1} - y_{k+1}z_{i-1} + y_{k-1}z_{i-1}) - (y_{i+1}z_{k+1} - y_{i-1}z_{k+1} - y_{i+1}z_{k-1} + y_{i-1}z_{k-1})] \\ & + \delta_\zeta[(y_{i+1}z_{j+1} - y_{i-1}z_{j+1} - y_{i+1}z_{j-1} + y_{i-1}z_{j-1}) - (y_{j+1}z_{i+1} - y_{j-1}z_{i+1} - y_{j+1}z_{i-1} + y_{j-1}z_{i-1})] \end{aligned}$$

Performing the final difference operation,

$$\begin{aligned} M_2 \simeq\ & (y_{j+1}z_{k+1} - y_{j-1}z_{k+1} - y_{j+1}z_{k-1} + y_{j-1}z_{k-1})_{i+1} - (y_{j+1}z_{k+1} - y_{j-1}z_{k+1} - y_{j+1}z_{k-1} + y_{j-1}z_{k-1})_{i-1} \\ & - (y_{k+1}z_{j+1} - y_{k-1}z_{j+1} - y_{k+1}z_{j-1} + y_{k-1}z_{j-1})_{i+1} + (y_{k+1}z_{j+1} - y_{k-1}z_{j+1} - y_{k+1}z_{j-1} + y_{k-1}z_{j-1})_{i-1} \\ & + (y_{k+1}z_{i+1} - y_{k-1}z_{i+1} - y_{k+1}z_{i-1} + y_{k-1}z_{i-1})_{j+1} - (y_{k+1}z_{i+1} - y_{k-1}z_{i+1} - y_{k+1}z_{i-1} + y_{k-1}z_{i-1})_{j-1} \\ & - (y_{i+1}z_{k+1} - y_{i-1}z_{k+1} - y_{i+1}z_{k-1} + y_{i-1}z_{k-1})_{j+1} + (y_{i+1}z_{k+1} - y_{i-1}z_{k+1} - y_{i+1}z_{k-1} + y_{i-1}z_{k-1})_{j-1} \\ & + (y_{i+1}z_{j+1} - y_{i-1}z_{j+1} - y_{i+1}z_{j-1} + y_{i-1}z_{j-1})_{k+1} - (y_{i+1}z_{j+1} - y_{i-1}z_{j+1} - y_{i+1}z_{j-1} + y_{i-1}z_{j-1})_{k-1} \\ & - (y_{j+1}z_{i+1} - y_{j-1}z_{i+1} - y_{j+1}z_{i-1} + y_{j-1}z_{i-1})_{k+1} + (y_{j+1}z_{i+1} - y_{j-1}z_{i+1} - y_{j+1}z_{i-1} + y_{j-1}z_{i-1})_{k-1} \end{aligned}$$

Finally, collecting terms,

$$M_2 \simeq y_{i+1,j+1,k}(z_{i+1,j,k+1} - z_{i+1,j,k-1} - z_{i,j+1,k+1} + z_{i,j+1,k-1})$$
$$+ y_{i+1,j-1,k}(-z_{i+1,j,k+1} + z_{i+1,j,k-1} + z_{i,j-1,k+1} - z_{i,j-1,k-1})$$
$$+ \cdots$$

For $M_2$ to be identically zero, the terms in parentheses in the above equation would have to vanish identically. This is clearly not the case for a general three-dimensional coordinate system.

There are fixes that have been developed to ensure that the finite difference equations do satisfy uniform flow, including the use of averaging formulas for the metric coefficients or simply subtracting the error from the equations (Pulliam and Steger, 1978). These methods are somewhat inelegant, however, and can be expensive to use. A cleaner and completely rigorous method is to rewrite the formulas defining the metric coefficients, equations (2.13), in conservation form (Thomas and Lombard, 1979). Using $\xi_x$ as an example,

$$\xi_x = J(y_\eta z_\zeta - y_\zeta z_\eta)$$
$$= J[(y_\eta z)_\zeta - (y_\zeta z)_\eta]$$

In *Proteus*, therefore, the metric coefficients are actually computed using the following equations.

$$\begin{aligned}
\xi_x &= J[(y_\eta z)_\zeta - (y_\zeta z)_\eta] \\
\xi_y &= J[(x_\zeta z)_\eta - (x_\eta z)_\zeta] \\
\xi_z &= J[(x_\eta y)_\zeta - (x_\zeta y)_\eta] \\
\eta_x &= J[(y_\zeta z)_\xi - (y_\xi z)_\zeta] \\
\eta_y &= J[(x_\xi z)_\zeta - (x_\zeta z)_\xi] \\
\eta_z &= J[(x_\zeta y)_\xi - (x_\xi y)_\zeta] \\
\zeta_x &= J[(y_\xi z)_\eta - (y_\eta z)_\xi] \\
\zeta_y &= J[(x_\eta z)_\xi - (x_\xi z)_\eta] \\
\zeta_z &= J[(x_\xi y)_\eta - (x_\eta y)_\xi] \\
\xi_t &= -x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z \\
\eta_t &= -x_\tau \eta_x - y_\tau \eta_y - z_\tau \eta_z \\
\zeta_t &= -x_\tau \zeta_x - y_\tau \zeta_y - z_\tau \zeta_z
\end{aligned} \tag{2.19}$$

To verify that computing the metrics in this way does lead to metric invariants that are identically zero, we now reevaluate $M_2$, the second bracketed term of equation (2.18). Applying difference operators,

$$M_2 \simeq \delta_\xi[\delta'_\zeta (y_\eta z) - \delta'_\eta (y_\zeta z)] + \delta_\eta[\delta'_\xi (y_\zeta z) - \delta'_\zeta (y_\xi z)] + \delta_\zeta[\delta'_\eta (y_\xi z) - \delta'_\xi (y_\eta z)]$$

Note that a distinction is made, for now, between the difference operators outside the brackets, $\delta_\xi$, $\delta_\eta$, and $\delta_\zeta$, and the difference operators inside the brackets, $\delta'_\xi$, $\delta'_\eta$, and $\delta'_\zeta$. The operators outside the brackets represent derivatives of the metric coefficients in $M_2$. These terms originated as part of the flux terms in equation (2.16). The operators $\delta_\xi$, $\delta_\eta$, and $\delta_\zeta$ are therefore the same as the operators used to represent derivatives in the governing differential equation. Now, note that the finite difference approximation of $M_2$ will vanish identically if, for example,

$$\delta_\xi[\delta'_\zeta (y_\eta z)] = \delta_\zeta[\delta'_\xi (y_\eta z)]$$

This will be true if $\delta'_\xi = \delta_\xi$, $\delta'_\eta = \delta_\eta$, and $\delta'_\zeta = \delta_\zeta$. This can be verified by expanding $\delta_\xi(\delta_\zeta f)$ and $\delta_\zeta(\delta_\xi f)$ using the centered difference formula of equation (5-1), and comparing. When the metric coefficients are

computed using equations (2.19), therefore, the derivatives of the parenthetical terms must be approximated using the same difference operators as those used to represent derivatives in equation (2.16). It does not matter how the $x_\xi$, $x_\eta$, etc., *inside* the parentheses are computed.

This procedure for computing the metrics ensures that the last three metric invariant terms in equation (2.18) are identically zero when differenced. The first metric invariant term must be handled somewhat differently. Setting it equal to zero gives

$$\left(\frac{1}{J}\right)_\tau + \left(\frac{\xi_t}{J}\right)_\xi + \left(\frac{\eta_t}{J}\right)_\eta + \left(\frac{\zeta_t}{J}\right)_\zeta = 0 \tag{2.20}$$

This is a statement of the geometric conservation law described by Thomas and Lombard (1979). For grids that do not change with time, this equation is, of course, automatically satisfied when differenced. However, for time-dependent grids it is not. In that case, the grid transformation Jacobian $J$ should be found by solving equation (2.20) at the new time level, using the same differencing scheme as in the governing flow equations, and not computed algebraically from equation (2.12). The current version of *Proteus* does not solve this equation, and thus strictly applies only to time-independent grids.

PAGE _____ INTENTIONALLY BLANK

## 3.0 TIME DIFFERENCING

The governing equations are solved by marching in time from some known set of initial conditions using a finite difference technique. The time differencing scheme currently used in *Proteus* is the generalized scheme of Beam and Warming (1978). The time derivative term in equation (2.16) is written as

$$\frac{\partial \hat{Q}}{\partial \tau} \simeq \frac{\Delta \hat{Q}^n}{\Delta \tau} = \frac{\theta_1}{1+\theta_2}\frac{\partial(\Delta \hat{Q}^n)}{\partial \tau} + \frac{1}{1+\theta_2}\frac{\partial \hat{Q}^n}{\partial \tau} + \frac{\theta_2}{1+\theta_2}\frac{\Delta \hat{Q}^{n-1}}{\Delta \tau} + O\left[\left(\theta_1 - \frac{1}{2} - \theta_2\right)\Delta\tau, (\Delta\tau)^2\right]$$

or,

$$\Delta \hat{Q}^n = \frac{\theta_1 \Delta\tau}{1+\theta_2}\frac{\partial(\Delta \hat{Q}^n)}{\partial \tau} + \frac{\Delta\tau}{1+\theta_2}\frac{\partial \hat{Q}^n}{\partial \tau} + \frac{\theta_2}{1+\theta_2}\Delta \hat{Q}^{n-1} + O\left[\left(\theta_1 - \frac{1}{2} - \theta_2\right)(\Delta\tau)^2, (\Delta\tau)^3\right] \tag{3.1}$$

where $\Delta \hat{Q}^n = \hat{Q}^{n+1} - \hat{Q}^n$. The superscripts $n$ and $n+1$ denote the known and unknown time levels, respectively.

The parameters $\theta_1$ and $\theta_2$ determine the type of time differencing scheme used. Some of the methods available with the above formula are given in the following table.

| $\theta_1$ | $\theta_2$ | Method | Truncation Error |
|---|---|---|---|
| 0 | 0 | Euler explicit | $O(\Delta\tau)^2$ |
| 0 | − 1/2 | Leapfrog explicit | $O(\Delta\tau)^3$ |
| 1 | 0 | Euler implicit | $O(\Delta\tau)^2$ |
| 1/2 | 0 | Trapezoidal implicit | $O(\Delta\tau)^3$ |
| 1 | 1/2 | 3-point backward implicit | $O(\Delta\tau)^3$ |

Note that even though the generalized time differencing formula includes explicit methods, the *Proteus* code assumes an implicit method is being used. Note also that the truncation error listed in the table is the error in the expression for $\Delta \hat{Q}^n$. The overall numerical method used in modelling the differential equations requires $\Delta \hat{Q}^n/\Delta\tau$, so the order of the overall method is this truncation error divided by $\Delta\tau$.

Solving equation (2.16) for $\partial \hat{Q}/\partial\tau$ and substituting the result into equation (3.1) for $\partial(\Delta \hat{Q}^n)/\partial\tau$ and $\partial \hat{Q}^n/\partial\tau$ yields

$$\begin{aligned}\Delta \hat{Q}^n = &-\frac{\theta_1 \Delta\tau}{1+\theta_2}\left(\frac{\partial(\Delta \hat{E}^n)}{\partial\xi} + \frac{\partial(\Delta \hat{F}^n)}{\partial\eta} + \frac{\partial(\Delta \hat{G}^n)}{\partial\zeta}\right) - \frac{\Delta\tau}{1+\theta_2}\left(\frac{\partial \hat{E}^n}{\partial\xi} + \frac{\partial \hat{F}^n}{\partial\eta} + \frac{\partial \hat{G}^n}{\partial\zeta}\right) \\ &+ \frac{\theta_1 \Delta\tau}{1+\theta_2}\left(\frac{\partial(\Delta \hat{E}_V{}^n)}{\partial\xi} + \frac{\partial(\Delta \hat{F}_V{}^n)}{\partial\eta} + \frac{\partial(\Delta \hat{G}_V{}^n)}{\partial\zeta}\right) + \frac{\Delta\tau}{1+\theta_2}\left(\frac{\partial \hat{E}_V{}^n}{\partial\xi} + \frac{\partial \hat{F}_V{}^n}{\partial\eta} + \frac{\partial \hat{G}_V{}^n}{\partial\zeta}\right) \\ &+ \frac{\theta_2}{1+\theta_2}\Delta \hat{Q}^{n-1} + O\left[\left(\theta_1 - \frac{1}{2} - \theta_2\right)(\Delta\tau)^2, (\Delta\tau)^3\right]\end{aligned} \tag{3.2}$$

PAGE _____ INTENTIONALLY BLANK

# 4.0 LINEARIZATION PROCEDURE

## 4.1 INVISCID TERMS

Equation (3.2) is nonlinear, since, for example, $\Delta\hat{\mathbf{E}}^n = \hat{\mathbf{E}}^{n+1} - \hat{\mathbf{E}}^n$ and the unknown $\hat{\mathbf{E}}^{n+1}$ is a nonlinear function of the dependent variables and of the metric coefficients resulting from the generalized grid transformation. The equations must therefore be linearized to be solved by the finite difference procedure used in *Proteus*. This is done by expanding each nonlinear expression in a Taylor series in time about the known time level $n$. Letting $G$ represent any nonlinear expression,

$$G^{n+1} = G^n + \left(\frac{\partial G}{\partial \tau}\right)^n \Delta\tau + O(\Delta\tau)^2 \tag{4.1}$$

where

$$\frac{\partial G}{\partial \tau} = \frac{\partial G}{\partial \rho}\frac{\partial \rho}{\partial \tau} + \frac{\partial G}{\partial (\rho u)}\frac{\partial (\rho u)}{\partial \tau} + \frac{\partial G}{\partial (\rho v)}\frac{\partial (\rho v)}{\partial \tau} + \frac{\partial G}{\partial (\rho w)}\frac{\partial (\rho w)}{\partial \tau} + \frac{\partial G}{\partial E_T}\frac{\partial E_T}{\partial \tau}$$

Note that *for linearization purposes only* the metric scale coefficients have been assumed to be locally independent of time. Note also that for this linearization procedure to be second order accurate, $\partial G/\partial\tau$ (and therefore $\partial\rho/\partial\tau$, $\partial(\rho u)/\partial\tau$, etc.) need only be first order accurate. Using forward differences, then, so that

$$\left(\frac{\partial \rho}{\partial \tau}\right)^n = \frac{\rho^{n+1} - \rho^n}{\Delta\tau} + O(\Delta\tau)$$

$$= \frac{\Delta\rho^n}{\Delta\tau} + O(\Delta\tau)$$

etc., equation (4.1) becomes

$$G^{n+1} = G^n + \left(\frac{\partial G}{\partial \rho}\right)^n \Delta\rho^n + \left(\frac{\partial G}{\partial (\rho u)}\right)^n \Delta(\rho u)^n + \left(\frac{\partial G}{\partial (\rho v)}\right)^n \Delta(\rho v)^n + \left(\frac{\partial G}{\partial (\rho w)}\right)^n \Delta(\rho w)^n + \left(\frac{\partial G}{\partial E_T}\right)^n \Delta E_T{}^n + O(\Delta\tau)^2 \tag{4.2}$$

As an example the $\partial(\rho u v \xi_y)/\partial\xi$ term from the $x$-momentum equation (part of the second element of $\partial\hat{\mathbf{E}}/\partial\xi$) will be used. The nonlinear part of this term is $(\rho uv)^{n+1}$. Rewriting this in terms of the dependent variables,

$$(\rho uv)^{n+1} = \left[\frac{(\rho u)(\rho v)}{\rho}\right]^{n+1}$$

Using equation (4.2), this is linearized as

$$(\rho uv)^{n+1} = (\rho uv)^n - (uv)^n(\rho^{n+1} - \rho^n) + v^n[(\rho u)^{n+1} - (\rho u)^n] + u^n[(\rho v)^{n+1} - (\rho v)^n] + O(\Delta\tau)^2$$

which can be rewritten as

$$\Delta(\rho uv)^n = -(uv)^n\Delta\rho^n + v^n\Delta(\rho u)^n + u^n\Delta(\rho v)^n + O(\Delta\tau)^2$$

This linearization procedure, when applied to the entire $\Delta\hat{\mathbf{E}}^n$ term in the vector equation (3.2), can be written as

$$\Delta\hat{\mathbf{E}}^n = \left(\frac{\partial\hat{\mathbf{E}}}{\partial\hat{\mathbf{Q}}}\right)^n \Delta\hat{\mathbf{Q}}^n + O(\Delta\tau)^2 \tag{4.3}$$

where $(\partial\hat{\mathbf{E}}/\partial\hat{\mathbf{Q}})^n$ is a Jacobian coefficient matrix (not to be confused with the Jacobian $J$ of the generalized grid transformation.) Similar equations can be written for $\Delta\hat{\mathbf{F}}^n$ and $\Delta\hat{\mathbf{G}}^n$.

Each term in each element of $\hat{\mathbf{E}}$, $\hat{\mathbf{F}}$, and $\hat{\mathbf{G}}$, given by equations (2.17b) through (2.17d), is linearized using the above procedure to generate the elements of the Jacobian coefficient matrices $\partial\hat{\mathbf{E}}/\partial\hat{\mathbf{Q}}$, $\partial\hat{\mathbf{F}}/\partial\hat{\mathbf{Q}}$, and $\partial\hat{\mathbf{G}}/\partial\hat{\mathbf{Q}}$. (Note that $\partial\hat{\mathbf{E}}/\partial\hat{\mathbf{Q}} = J\partial\hat{\mathbf{E}}/\partial\mathbf{Q}$.) When this is done $\partial\hat{\mathbf{E}}/\partial\hat{\mathbf{Q}}$ can be written as

$$\frac{\partial\hat{\mathbf{E}}}{\partial\hat{\mathbf{Q}}} = \begin{bmatrix} \xi_t & \xi_x & \xi_y & \xi_z & 0 \\ \frac{\partial p}{\partial\rho}\xi_x - uf_1 & \xi_t + f_1 + u\xi_x + \frac{\partial p}{\partial(\rho u)}\xi_x & u\xi_y + \frac{\partial p}{\partial(\rho v)}\xi_x & u\xi_z + \frac{\partial p}{\partial(\rho w)}\xi_x & \frac{\partial p}{\partial E_T}\xi_x \\ \frac{\partial p}{\partial\rho}\xi_y - vf_1 & v\xi_x + \frac{\partial p}{\partial(\rho u)}\xi_y & \xi_t + f_1 + v\xi_y + \frac{\partial p}{\partial(\rho v)}\xi_y & v\xi_z + \frac{\partial p}{\partial(\rho w)}\xi_y & \frac{\partial p}{\partial E_T}\xi_y \\ \frac{\partial p}{\partial\rho}\xi_z - wf_1 & w\xi_x + \frac{\partial p}{\partial(\rho u)}\xi_z & w\xi_y + \frac{\partial p}{\partial(\rho v)}\xi_z & \xi_t + f_1 + w\xi_z + \frac{\partial p}{\partial(\rho w)}\xi_z & \frac{\partial p}{\partial E_T}\xi_z \\ -f_1\left(f_2 - \frac{\partial p}{\partial\rho}\right) & f_2\xi_x + f_1\frac{\partial p}{\partial(\rho u)} & f_2\xi_y + f_1\frac{\partial p}{\partial(\rho v)} & f_2\xi_z + f_1\frac{\partial p}{\partial(\rho w)} & \xi_t + f_1\left(1 + \frac{\partial p}{\partial E_T}\right) \end{bmatrix} \tag{4.4}$$

where $f_1 = u\xi_x + v\xi_y + w\xi_z$ and $f_2 = (E_T + p)/\rho$. The Jacobian matrices $\partial\hat{\mathbf{F}}/\partial\hat{\mathbf{Q}}$ and $\partial\hat{\mathbf{G}}/\partial\hat{\mathbf{Q}}$ have the same form as $\partial\hat{\mathbf{E}}/\partial\hat{\mathbf{Q}}$, but with $\xi$ replaced by $\eta$ and $\zeta$, respectively.

The linearized pressure terms have deliberately been left in terms of $\partial p/\partial\rho$, $\partial p/\partial(\rho u)$, etc. The expressions to be used for these derivatives depend on the equation of state. Those currently built into the *Proteus* code, for a perfect gas, are presented in Section 4.3.

## 4.2 VISCOUS TERMS

The nonlinear viscous terms in equation (3.2), involving $\Delta\hat{\mathbf{E}}_V^n$, $\Delta\hat{\mathbf{F}}_V^n$, and $\Delta\hat{\mathbf{G}}_V^n$, must also be linearized. To do this, the elements of $\hat{\mathbf{E}}_V$, $\hat{\mathbf{F}}_V$, and $\hat{\mathbf{G}}_V$, given in equations (2.17e) through (2.17g), must first be rewritten in terms of the dependent variables, and with derivatives in the Cartesian directions transformed to derivatives in the computational directions using the chain rule. When the resulting expressions are substituted into equation (3.2), mixed second derivatives appear as well as second derivatives in a single coordinate direction. The mixed, or cross, derivative terms would lead to considerable complications in the implicit numerical solution algorithm if they were linearized using the procedure presented in Section 4.1. The two types of second derivatives are thus treated differently, and $\hat{\mathbf{E}}_V$, $\hat{\mathbf{F}}_V$, and $\hat{\mathbf{G}}_V$ are written as

$$\begin{aligned} \hat{\mathbf{E}}_V &= \hat{\mathbf{E}}_{V_1} + \hat{\mathbf{E}}_{V_2} \\ \hat{\mathbf{F}}_V &= \hat{\mathbf{F}}_{V_1} + \hat{\mathbf{F}}_{V_2} \\ \hat{\mathbf{G}}_V &= \hat{\mathbf{G}}_{V_1} + \hat{\mathbf{G}}_{V_2} \end{aligned} \tag{4.5}$$

where $\hat{\mathbf{E}}_{V_1}$, $\hat{\mathbf{F}}_{V_1}$, and $\hat{\mathbf{G}}_{V_1}$ only contain derivatives in the $\xi$, $\eta$, and $\zeta$ directions, respectively, and $\hat{\mathbf{E}}_{V_2}$, $\hat{\mathbf{F}}_{V_2}$, and $\hat{\mathbf{G}}_{V_2}$ contain derivatives in the remaining directions. The fully expanded expressions for $\hat{\mathbf{E}}_{V_1}$, $\hat{\mathbf{E}}_{V_2}$, etc., are fairly long, and therefore are presented in Appendix A.

### 4.2.1 Non-Cross Derivatives

Examination of the elements of $\hat{\mathbf{E}}_{V_1}$ in equations (A.2a) through (A.2d), and (A.2f), shows that every term has the form $fg_\xi$, where $g$ is a function of the dependent variables, and $f$ is a function of $\mu$, $\lambda$, $k$, and/or the metric coefficients. Expanding in a Taylor series about time level $n$ gives

$$(fg_\xi)^{n+1} = (fg_\xi)^n + \left[ \frac{\partial (fg_\xi)}{\partial \tau} \right]^n \Delta\tau + O(\Delta\tau)^2$$

*For linearization purposes only,* we will assume $f$ is locally independent of time. We can thus write

$$(fg_\xi)^{n+1} = (fg_\xi)^n + f^n \frac{\partial}{\partial \xi} \left[ \frac{\partial g}{\partial \tau} \right]^n \Delta\tau + O(\Delta\tau)^2$$

where

$$\frac{\partial g}{\partial \tau} = \frac{\partial g}{\partial \rho} \frac{\partial \rho}{\partial \tau} + \frac{\partial g}{\partial (\rho u)} \frac{\partial (\rho u)}{\partial \tau} + \cdots$$

Therefore

$$(fg_\xi)^{n+1} = (fg_\xi)^n + f^n \frac{\partial}{\partial \xi} \left[ \frac{\partial g}{\partial \rho} \Delta\rho + \frac{\partial g}{\partial (\rho u)} \Delta(\rho u) + \cdots \right]^n + O(\Delta\tau)^2$$

As with the inviscid terms, the linearization procedure for the entire $\Delta\hat{\mathbf{E}}_{V_1}{}^n$ viscous term in equation (3.2) can be written as

$$\Delta\hat{\mathbf{E}}_{V_1}^n = \left( \frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}} \right)^n \Delta\hat{\mathbf{Q}}^n + O(\Delta\tau)^2 \tag{4.6}$$

Similar equations may be written for $\Delta\hat{\mathbf{F}}_{V_1}^n$ and $\Delta\hat{\mathbf{G}}_{V_1}^n$. The Jacobian coefficient matrix $\partial\hat{\mathbf{E}}_{V_1}/\partial\hat{\mathbf{Q}}$ is

$$\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}} = \frac{1}{Re_r}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{21} & \alpha_{xx}\frac{\partial}{\partial \xi}\left(\frac{1}{\rho}\right) & \alpha_{xy}\frac{\partial}{\partial \xi}\left(\frac{1}{\rho}\right) & \alpha_{xz}\frac{\partial}{\partial \xi}\left(\frac{1}{\rho}\right) & 0 \\ \left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{31} & \alpha_{xy}\frac{\partial}{\partial \xi}\left(\frac{1}{\rho}\right) & \alpha_{yy}\frac{\partial}{\partial \xi}\left(\frac{1}{\rho}\right) & \alpha_{yz}\frac{\partial}{\partial \xi}\left(\frac{1}{\rho}\right) & 0 \\ \left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{41} & \alpha_{xz}\frac{\partial}{\partial \xi}\left(\frac{1}{\rho}\right) & \alpha_{yz}\frac{\partial}{\partial \xi}\left(\frac{1}{\rho}\right) & \alpha_{zz}\frac{\partial}{\partial \xi}\left(\frac{1}{\rho}\right) & 0 \\ \left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{51} & \left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{52} & \left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{53} & \left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{54} & \alpha_0\frac{\partial}{\partial \xi}\left(\frac{\partial T}{\partial E_T}\right) \end{bmatrix} \tag{4.7}$$

where

$$\alpha_{xx} = (2\mu + \lambda)\xi_x{}^2 + \mu\xi_y{}^2 + \mu\xi_z{}^2$$

$$\alpha_{yy} = \mu\xi_x{}^2 + (2\mu + \lambda)\xi_y{}^2 + \mu\xi_z{}^2$$

$$\alpha_{zz} = \mu\xi_x{}^2 + \mu\xi_y{}^2 + (2\mu + \lambda)\xi_z{}^2$$

$$\alpha_{xy} = (\mu + \lambda)\xi_x\xi_y$$

$$\alpha_{xz} = (\mu + \lambda)\xi_x\xi_z$$

$$\alpha_{yz} = (\mu + \lambda)\xi_y\xi_z$$

$$\alpha_0 = k(\xi_x{}^2 + \xi_y{}^2 + \xi_z{}^2)$$

$$\left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{21} = -\alpha_{xx}\frac{\partial}{\partial \xi}\left(\frac{u}{\rho}\right) - \alpha_{xy}\frac{\partial}{\partial \xi}\left(\frac{v}{\rho}\right) - \alpha_{xz}\frac{\partial}{\partial \xi}\left(\frac{w}{\rho}\right)$$

$$\left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{31} = -\alpha_{xy}\frac{\partial}{\partial \xi}\left(\frac{u}{\rho}\right) - \alpha_{yy}\frac{\partial}{\partial \xi}\left(\frac{v}{\rho}\right) - \alpha_{yz}\frac{\partial}{\partial \xi}\left(\frac{w}{\rho}\right)$$

$$\left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{41} = -\alpha_{xz}\frac{\partial}{\partial \xi}\left(\frac{u}{\rho}\right) - \alpha_{yz}\frac{\partial}{\partial \xi}\left(\frac{v}{\rho}\right) - \alpha_{zz}\frac{\partial}{\partial \xi}\left(\frac{w}{\rho}\right)$$

$$\left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{51} = -\alpha_{xx}\frac{\partial}{\partial \xi}\left(\frac{u^2}{\rho}\right) - \alpha_{yy}\frac{\partial}{\partial \xi}\left(\frac{v^2}{\rho}\right) - \alpha_{zz}\frac{\partial}{\partial \xi}\left(\frac{w^2}{\rho}\right)$$
$$- 2\alpha_{xy}\frac{\partial}{\partial \xi}\left(\frac{uv}{\rho}\right) - 2\alpha_{xz}\frac{\partial}{\partial \xi}\left(\frac{uw}{\rho}\right) - 2\alpha_{yz}\frac{\partial}{\partial \xi}\left(\frac{vw}{\rho}\right) + \alpha_0\frac{\partial}{\partial \xi}\left(\frac{\partial T}{\partial \rho}\right)$$

$$\left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{52} = -\left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{21} + \alpha_0\frac{\partial}{\partial \xi}\left(\frac{\partial T}{\partial (\rho u)}\right)$$

$$\left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{53} = -\left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{31} + \alpha_0\frac{\partial}{\partial \xi}\left(\frac{\partial T}{\partial (\rho v)}\right)$$

$$\left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{54} = -\left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{41} + \alpha_0\frac{\partial}{\partial \xi}\left(\frac{\partial T}{\partial (\rho w)}\right)$$

Like the pressure terms discussed earlier, the form of the temperature terms will depend on the equation of state being used. Those currently built into the *Proteus* code, for a perfect gas, are presented in Section 4.3.

Note that in equation (4.6) the derivatives appearing in the Jacobian coefficient matrix $\partial \hat{\mathbf{E}}_{V_1}/\partial \hat{\mathbf{Q}}$ are also to be applied to the $\Delta \hat{\mathbf{Q}}^n$ appearing outside the parentheses. For example, the element in the second row and second column of $\partial \hat{\mathbf{E}}_{V_1}/\partial \hat{\mathbf{Q}}$, which corresponds to the $\Delta(\rho u)$ term in the $x$-momentum equation, is $\alpha_{xx}\partial(1/\rho)/\partial \xi$. For this term, the notation used in equation (4.6) means

$$\left(\frac{\partial \hat{\mathbf{E}}_{V_1}}{\partial \hat{\mathbf{Q}}}\right)_{22}^{n} \Delta \hat{\mathbf{Q}}_2^n = \alpha_{xx}{}^n \frac{\partial}{\partial \xi}\left(\frac{1}{\rho}\right)^n \Delta(\rho u/J)^n$$
$$= \alpha_{xx}{}^n \frac{\partial}{\partial \xi}\left(\frac{\Delta(\rho u/J)^n}{\rho^n}\right)$$

The Jacobian coefficient matrices for the remaining non-cross derivative viscous terms, $\partial \hat{\mathbf{F}}_{V_1}/\partial \hat{\mathbf{Q}}$ and $\partial \hat{\mathbf{G}}_{V_1}/\partial \hat{\mathbf{Q}}$, have the same form as $\partial \hat{\mathbf{E}}_{V_1}/\partial \hat{\mathbf{Q}}$, but with $\xi$ replaced by $\eta$ and $\zeta$, respectively.

### 4.2.2 Cross Derivatives

As stated earlier, linearizing the cross derivative viscous terms in the same way as the remaining terms is very complicated within the framework of the implicit numerical solution algorithm used in *Proteus*. They are therefore simply lagged (i.e., evaluated at the known time level $n$ and treated as source terms.) As noted by Beam and Warming (1978), this does not lead to a formal accuracy loss since

$$\begin{aligned}
\Delta \hat{\mathbf{E}}_{V_2}^n &= \Delta \hat{\mathbf{E}}_{V_2}^{n-1} + O(\Delta\tau)^2 \\
\Delta \hat{\mathbf{F}}_{V_2}^n &= \Delta \hat{\mathbf{F}}_{V_2}^{n-1} + O(\Delta\tau)^2 \\
\Delta \hat{\mathbf{G}}_{V_2}^n &= \Delta \hat{\mathbf{G}}_{V_2}^{n-1} + O(\Delta\tau)^2
\end{aligned} \tag{4.8}$$

## 4.3 EQUATION OF STATE

The expressions to be used for $\partial p/\partial \rho$, $\partial T/\partial \rho$, etc., which arise from the linearization procedure, depend on the equation of state. The equation currently built into *Proteus* is for perfect gases, and can be written as

$$p = (\gamma - 1)\left[E_T - \frac{1}{2}\rho(u^2 + v^2 + w^2)\right] \tag{4.9}$$

or, in terms of temperature, as

$$T = \frac{1}{c_v}\left[\frac{E_T}{\rho} - \frac{1}{2}(u^2 + v^2 + w^2)\right] \tag{4.10}$$

With this equation of state, then, the appropriate derivatives are

$$\frac{\partial p}{\partial \rho} = \frac{\gamma - 1}{2}(u^2 + v^2 + w^2) \tag{4.11a}$$

$$\frac{\partial p}{\partial (\rho u)} = -(\gamma - 1)u \tag{4.11b}$$

$$\frac{\partial p}{\partial (\rho v)} = -(\gamma - 1)v \tag{4.11c}$$

$$\frac{\partial p}{\partial (\rho w)} = -(\gamma - 1)w \tag{4.11d}$$

$$\frac{\partial p}{\partial E_T} = \gamma - 1 \tag{4.11e}$$

$$\frac{\partial T}{\partial \rho} = -\frac{1}{c_v}\left[\frac{E_T}{\rho^2} - \frac{1}{\rho}(u^2 + v^2 + w^2)\right] \tag{4.12a}$$

$$\frac{\partial T}{\partial (\rho u)} = -\frac{u}{c_v \rho} \tag{4.12b}$$

$$\frac{\partial T}{\partial (\rho v)} = -\frac{v}{c_v \rho} \tag{4.12c}$$

$$\frac{\partial T}{\partial (\rho w)} = -\frac{w}{c_v \rho} \tag{4.12d}$$

$$\frac{\partial T}{\partial E_T} = \frac{1}{c_v \rho} \tag{4.12e}$$

If constant stagnation enthalpy is assumed, as discussed in Section 2.2, the appropriate equation of state is

$$p = \frac{\gamma - 1}{\gamma}\rho\left[h_T - \frac{1}{2}(u^2 + v^2 + w^2)\right] \tag{4.13}$$

and the temperature becomes

$$T = \frac{1}{c_p}\left[h_T - \frac{1}{2}(u^2 + v^2 + w^2)\right] \tag{4.14}$$

With these equations, the derivatives of $p$ and $T$ with respect to the dependent variables are

$$\frac{\partial p}{\partial \rho} = \frac{\gamma - 1}{\gamma}\left[h_T + \frac{1}{2}(u^2 + v^2 + w^2)\right] \tag{4.15a}$$

$$\frac{\partial p}{\partial (\rho u)} = -\frac{\gamma - 1}{\gamma} u \tag{4.15b}$$

$$\frac{\partial p}{\partial (\rho v)} = -\frac{\gamma - 1}{\gamma} v \tag{4.15c}$$

$$\frac{\partial p}{\partial (\rho w)} = -\frac{\gamma - 1}{\gamma} w \tag{4.15d}$$

$$\frac{\partial T}{\partial \rho} = \frac{1}{c_p \rho}(u^2 + v^2 + w^2) \tag{4.16a}$$

$$\frac{\partial T}{\partial (\rho u)} = -\frac{u}{c_p \rho} \tag{4.16b}$$

$$\frac{\partial T}{\partial (\rho v)} = -\frac{v}{c_p \rho} \tag{4.16c}$$

$$\frac{\partial T}{\partial (\rho w)} = -\frac{w}{c_p \rho} \tag{4.16d}$$

## 4.4 LINEARIZED GOVERNING EQUATION

The linearized form of equation (3.2) can now be written as

$$\begin{aligned}
&\Delta\hat{\mathbf{Q}}^n + \frac{\theta_1 \Delta\tau}{1+\theta_2}\left\{\frac{\partial}{\partial\xi}\left[\left(\frac{\partial\hat{\mathbf{E}}}{\partial\hat{\mathbf{Q}}}\right)^n \Delta\hat{\mathbf{Q}}^n\right] + \frac{\partial}{\partial\eta}\left[\left(\frac{\partial\hat{\mathbf{F}}}{\partial\hat{\mathbf{Q}}}\right)^n \Delta\hat{\mathbf{Q}}^n\right] + \frac{\partial}{\partial\zeta}\left[\left(\frac{\partial\hat{\mathbf{G}}}{\partial\hat{\mathbf{Q}}}\right)^n \Delta\hat{\mathbf{Q}}^n\right]\right\} \\
&- \frac{\theta_1 \Delta\tau}{1+\theta_2}\left\{\frac{\partial}{\partial\xi}\left[\left(\frac{\partial\hat{\mathbf{E}}_{V_1}}{\partial\hat{\mathbf{Q}}}\right)^n \Delta\hat{\mathbf{Q}}^n\right] + \frac{\partial}{\partial\eta}\left[\left(\frac{\partial\hat{\mathbf{F}}_{V_1}}{\partial\hat{\mathbf{Q}}}\right)^n \Delta\hat{\mathbf{Q}}^n\right] + \frac{\partial}{\partial\zeta}\left[\left(\frac{\partial\hat{\mathbf{G}}_{V_1}}{\partial\hat{\mathbf{Q}}}\right)^n \Delta\hat{\mathbf{Q}}^n\right]\right\} = \\
&- \frac{\Delta\tau}{1+\theta_2}\left(\frac{\partial\hat{\mathbf{E}}}{\partial\xi} + \frac{\partial\hat{\mathbf{F}}}{\partial\eta} + \frac{\partial\hat{\mathbf{G}}}{\partial\zeta}\right)^n + \frac{\Delta\tau}{1+\theta_2}\left(\frac{\partial\hat{\mathbf{E}}_{V_1}}{\partial\xi} + \frac{\partial\hat{\mathbf{F}}_{V_1}}{\partial\eta} + \frac{\partial\hat{\mathbf{G}}_{V_1}}{\partial\zeta}\right)^n \\
&+ \frac{(1+\theta_3)\Delta\tau}{1+\theta_2}\left(\frac{\partial\hat{\mathbf{E}}_{V_2}}{\partial\xi} + \frac{\partial\hat{\mathbf{F}}_{V_2}}{\partial\eta} + \frac{\partial\hat{\mathbf{G}}_{V_2}}{\partial\zeta}\right)^n - \frac{\theta_3\Delta\tau}{1+\theta_2}\left(\frac{\partial\hat{\mathbf{E}}_{V_2}}{\partial\xi} + \frac{\partial\hat{\mathbf{F}}_{V_2}}{\partial\eta} + \frac{\partial\hat{\mathbf{G}}_{V_2}}{\partial\zeta}\right)^{n-1} \\
&+ \frac{\theta_2}{1+\theta_2}\Delta\hat{\mathbf{Q}}^{n-1} + O\left[\left(\theta_1 - \frac{1}{2} - \theta_2\right)(\Delta\tau)^2, (\theta_3 - \theta_1)(\Delta\tau)^2, (\Delta\tau)^3\right]
\end{aligned} \tag{4.17}$$

There are a couple of things that should be mentioned about this equation. First, this equation is in so-called "delta" form. We will actually be solving this equation for $\Delta\hat{\mathbf{Q}}^n$ and recovering $\hat{\mathbf{Q}}^{n+1}$ from $\hat{\mathbf{Q}}^{n+1} = \Delta\hat{\mathbf{Q}}^n + \hat{\mathbf{Q}}^n$. And second, in the coefficients of the cross derivative viscous terms the time differencing parameter $\theta_1$ has been replaced by $\theta_3$. For second order time differencing (i.e., if $\theta_1 = \theta_2 + 1/2$), $\theta_3$ should be set equal to $\theta_1$. For first order time differencing, however, $\theta_3$ can be set equal to zero without losing accuracy.

PAGE _____ INTENTIONALLY BLANK

## 5.0 SPACE DIFFERENCING

To solve equation (4.17) an evenly spaced grid is defined in the computational $(\xi, \eta, \zeta)$ coordinate system. Spatial derivatives are then approximated by finite difference formulas. First derivatives in the $\xi$ direction are approximated using the following second-order central difference formula.

$$\left(\frac{\partial f}{\partial \xi}\right)_{i,j,k} \simeq \delta_\xi f_{i,j,k} = \frac{f_{i+1,j,k} - f_{i-1,j,k}}{2\Delta\xi} \tag{5.1}$$

The subscripts $i, j$, and $k$ represent grid point indices in the $\xi$, $\eta$, and $\zeta$ directions. The computational grid spacing $\Delta\xi$ is constant, and equal to $1/(N_1 - 1)$, where $N_1$ is the number of grid points in the $\xi$ direction. Similar formulas are used for first derivatives in the $\eta$ and $\zeta$ directions.

The non-cross derivative viscous terms in the $\xi$ direction in equation (4.17) all have the form

$$\frac{\partial}{\partial \xi}\left[f \frac{\partial}{\partial \xi}(g\Delta Q)\right]$$

where $Q$ represents one of the elements of $\hat{\mathbf{Q}}$. Using central differences this is approximated by

$$\begin{aligned}
\frac{\partial}{\partial \xi}\left[f \frac{\partial}{\partial \xi}(g\Delta Q)\right]_{i,j,k} &\simeq \delta_\xi[f \delta_\xi(g\Delta Q)]_{i,j,k} \\
&= \frac{1}{\Delta\xi}\{f_{i+1/2,j,k}\delta_\xi(g\Delta Q)_{i+1/2,j,k} - f_{i-1/2,j,k}\delta_\xi(g\Delta Q)_{i-1/2,j,k}\} \\
&= \frac{1}{(\Delta\xi)^2}\{f_{i+1/2,j,k}[(g\Delta Q)_{i+1,j,k} - (g\Delta Q)_{i,j,k}] \\
&\qquad - f_{i-1/2,j,k}[(g\Delta Q)_{i,j,k} - (g\Delta Q)_{i-1,j,k}]\} \\
&= \frac{1}{2(\Delta\xi)^2}\{(f_{i,j,k} + f_{i+1,j,k})[(g\Delta Q)_{i+1,j,k} - (g\Delta Q)_{i,j,k}] \\
&\qquad - (f_{i,j,k} + f_{i-1,j,k})[(g\Delta Q)_{i,j,k} - (g\Delta Q)_{i-1,j,k}]\} \\
&= \frac{1}{2(\Delta\xi)^2}\{(f_{i-1,j,k} + f_{i,j,k})(g\Delta Q)_{i-1,j,k} \\
&\qquad - (f_{i-1,j,k} + 2f_{i,j,k} + f_{i-1,j,k})(g\Delta Q)_{i,j,k} \\
&\qquad + (f_{i,j,k} + f_{i+1,j,k})(g\Delta Q)_{i+1,j,k}\}
\end{aligned} \tag{5.2}$$

Similar formulas are used for second derivatives in the $\eta$ and $\zeta$ directions.

Cross derivative viscous terms in the $\xi$-$\eta$ direction are evaluated using the following central difference formula.

$$\frac{\partial}{\partial \xi}\left(f\frac{\partial g}{\partial \eta}\right)_{i,j,k} \simeq \delta_\xi (f \delta_\eta g)_{i,j,k}$$

$$= \frac{1}{2\Delta\xi}\left[f_{i+1,j,k}(\delta_\eta g)_{i+1,j,k} - f_{i-1,j,k}(\delta_\eta g)_{i-1,j,k}\right]$$

$$= \frac{1}{4\Delta\xi\Delta\eta}\left[f_{i+1,j,k}(g_{i+1,j+1,k} - g_{i+1,j-1,k}) - f_{i-1,j,k}(g_{i-1,j+1,k} - g_{i-1,j-1,k})\right] \tag{5.3}$$

Similar formulas are used for the remaining cross derivatives. Note that this formula is only needed for the source terms, since the viscous cross derivative terms are lagged one time level.

When first derivatives are needed normal to a computational boundary, such as for Neumann boundary conditions, either first- or second-order one-sided differencing is used. The first-order formula at the $\xi = 0$ boundary is

$$\left(\frac{\partial f}{\partial \xi}\right)_{1,j,k} \simeq \frac{1}{\Delta\xi}(f_{2,j,k} - f_{1,j,k}) \tag{5.4}$$

and at the $\xi = 1$ boundary,

$$\left(\frac{\partial f}{\partial \xi}\right)_{N_1,j,k} \simeq \frac{1}{\Delta\xi}(f_{N_1,j,k} - f_{N_1-1,j,k}) \tag{5.5}$$

The second-order formula at the $\xi = 0$ boundary is

$$\left(\frac{\partial f}{\partial \xi}\right)_{1,j,k} \simeq \frac{1}{2\Delta\xi}(-3f_{1,j,k} + 4f_{2,j,k} - f_{3,j,k}) \tag{5.6}$$

and at the $\xi = 1$ boundary,

$$\left(\frac{\partial f}{\partial \xi}\right)_{N_1,j,k} \simeq \frac{1}{2\Delta\xi}(f_{N_1-2,j,k} - 4f_{N_1-1,j,k} + 3f_{N_1,j,k}) \tag{5.7}$$

Similar formulas are used at the computational boundaries in the $\eta$ and $\zeta$ direction.

# 6.0 BOUNDARY CONDITIONS

Choosing boundary conditions is perhaps the most important step in solving a flow problem with *Proteus*. Since the equations being solved at interior points are the same for every problem, the boundary conditions are what determines the final flow field for steady flows.

With the difference formulas presented in Section 5.0, $N_{eq}$ boundary conditions are required at each computational boundary, where $N_{eq}$ is the number of equations being solved. Note, however, that this is a numerical requirement, not a mathematical one. For example, for one-dimensional Euler flow $N_{eq} = 3$. However, characteristic theory shows that, mathematically, only two conditions may be specified at a subsonic inflow boundary, and only one at a subsonic outflow boundary (Pulliam, 1986a). Some sort of extrapolation is typically used for the additional numerical boundary conditions.

A variety of boundary conditions are built into the *Proteus* code, including: (1) specified values and/or gradients of Cartesian velocities $u$, $v$, and $w$, normal velocity $V_n$, coordinate direction velocities $V_\xi$, $V_\eta$, and $V_\zeta$, pressure $p$, temperature $T$, and density $\rho$; (2) specified values of total pressure $p_T$, total temperature $T_T$, and flow angles; (3) linear extrapolation; and (4) spatial periodicity. Another useful boundary condition is a "no change from initial condition" option for $u$, $v$, $w$, $p$, $T$, $\rho$, $p_T$, and/or $T_T$. Provision is also made for user-written boundary conditions. The boundary conditions may be steady, unsteady, or time-periodic. The exact combination of boundary conditions to be used will depend on the problem being run.

The boundary conditions in *Proteus* are treated implicitly. They may be viewed simply as additional equations to be solved by the ADI solution algorithm. And, in general, they involve nonlinear functions of the dependent variables. They must therefore be linearized using the procedure described in Section 4.0. The following sections describe this linearization for the general types of boundary conditions currently built into *Proteus*.

## 6.1 NO CHANGE FROM INITIAL CONDITIONS, $\Delta g = 0$

This boundary condition simply sets the boundary value of the function equal to its initial condition value. It can be written as

$$\Delta g^n = g^{n+1} - g^n = 0 \tag{6.1}$$

In general, $g$ can be a nonlinear combination of the dependent variables $\hat{\mathbf{Q}}$. Linearizing $g$ using the procedure described in Section 4.0, we get

$$g^{n+1} = g^n + \left( \frac{\partial g}{\partial \hat{\mathbf{Q}}} \right)^n \Delta \hat{\mathbf{Q}}^n + O(\Delta\tau)^2 \tag{6.2}$$

Neglecting the $O(\Delta\tau)^2$ linearization error, the linearized form of equation (6.1) can thus be written as

$$\left( \frac{\partial g}{\partial \hat{\mathbf{Q}}} \right)^n \Delta \hat{\mathbf{Q}}^n = 0 \tag{6.3}$$

## 6.2 SPECIFIED FUNCTION, $g = f$

A specified function at a boundary can be written simply as

$$g^{n+1} = f \tag{6.4}$$

where $g$ is the function being specified and $f$ is the value being specified. Note that $f$ can vary along the boundary, and can be time-dependent. Using equation (6.2) and neglecting the linearization error, the linearized boundary condition becomes

$$\left(\frac{\partial g}{\partial \hat{\mathbf{Q}}}\right)^n \Delta\hat{\mathbf{Q}}^n = f - g^n \tag{6.5}$$

## 6.3 SPECIFIED COORDINATE DIRECTION GRADIENT, $\partial g/\partial\phi = f$

A specified gradient of a function in a coordinate direction can be written as

$$\left(\frac{\partial g}{\partial \phi}\right)^{n+1} = f \tag{6.6}$$

where $g$ is the function whose gradient is being specified, $f$ is the specified value, and $\phi$ is the coordinate direction $\xi$, $\eta$, or $\zeta$. Note that $f$ can vary along the boundary, and can be time-dependent.

The linearized form of $g$ is given by equation (6.2). The linearized form of equation (6.6) can thus be written as

$$\left(\frac{\partial g}{\partial \phi}\right)^n + \frac{\partial}{\partial \phi}\left[\left(\frac{\partial g}{\partial \hat{\mathbf{Q}}}\right)^n \Delta\hat{\mathbf{Q}}^n\right] = f + O(\Delta\tau)^2 \tag{6.7}$$

Replacing differential operators with difference operators and neglecting the linearization error, the linearized boundary condition can be written as

$$\delta_\phi\left[\left(\frac{\partial g}{\partial \hat{\mathbf{Q}}}\right)^n \Delta\hat{\mathbf{Q}}^n\right] = f - \delta_\phi g^n \tag{6.8}$$

..here $\delta_\phi$ represents the one-sided difference operator to be used at the boundary. Options are available in *Proteus* to use either first-order two-point or second-order three-point differencing.

Note that this boundary condition is a specified value of the derivative with respect to the computational coordinate, not with respect to the physical distance in the direction of the computational coordinate. Following Korn and Korn (1968), and using the properties of the generalized coordinate transformation, it can be shown that for the $\xi$ direction the two derivatives are related by

$$\frac{\partial g}{\partial s_\xi} = \frac{1}{\sqrt{G_{\xi\xi}}}\frac{\partial g}{\partial \xi}$$

where

$$G_{\xi\xi} = \frac{1}{J^2}\left[(\eta_y\zeta_z - \eta_z\zeta_y)^2 + (\eta_z\zeta_x - \eta_x\zeta_z)^2 + (\eta_x\zeta_y - \eta_y\zeta_x)^2\right]$$

Similarly, for the $\eta$ direction,

$$\frac{\partial g}{\partial s_\eta} = \frac{1}{\sqrt{G_{\eta\eta}}}\frac{\partial g}{\partial \eta}$$

where

$$G_{\eta\eta} = \frac{1}{J^2}\left[(\xi_y\zeta_z - \xi_z\zeta_y)^2 + (\xi_z\zeta_x - \xi_x\zeta_z)^2 + (\xi_x\zeta_y - \xi_y\zeta_x)^2\right]$$

And, for the $\zeta$ direction,

$$\frac{\partial g}{\partial s_\zeta} = \frac{1}{\sqrt{G_{\zeta\zeta}}}\frac{\partial g}{\partial \zeta}$$

where

$$G_{\zeta\zeta} = \frac{1}{J^2}\left[(\xi_y\eta_z - \xi_z\eta_y)^2 + (\xi_z\eta_x - \xi_x\eta_z)^2 + (\xi_x\eta_y - \xi_y\eta_x)^2\right]$$

If the value $f = 0$, of course, the two derivatives are equivalent.

## 6.4 SPECIFIED NORMAL DIRECTION GRADIENT, $\nabla g \cdot \vec{n} = f$

A specified gradient of a function normal to the boundary can be written as

$$\nabla g^{n+1} \cdot \vec{n} = f \tag{6.9}$$

where $g$ is the function whose gradient is being specified, $f$ is the specified value, and $\vec{n}$ represents the unit vector normal to the boundary. Note that $f$ can vary along the boundary, and can be time-dependent.

For illustrative purposes, assume we are specifying a gradient normal to a constant $\xi$ boundary. Then

$$\vec{n} = \frac{\nabla\xi}{|\nabla\xi|} = \frac{1}{m}\xi_x\vec{i} + \frac{1}{m}\xi_y\vec{j} + \frac{1}{m}\xi_z\vec{i}$$

where

$$m = \sqrt{\xi_x^{\,2} + \xi_y^{\,2} + \xi_z^{\,2}}$$

Equation (6.9) can then be written as

$$\frac{1}{m}(g_x^{n+1}\xi_x + g_y^{n+1}\xi_y + g_z^{n+1}\xi_z) = f \tag{6.10}$$

Using the chain rule to expand $g_x^{n+1}$, $g_y^{n+1}$, and $g_z^{n+1}$,

$$\begin{aligned} g_x^{n+1} &= g_\xi^{n+1}\xi_x + g_\eta^{n+1}\eta_x + g_\zeta^{n+1}\zeta_x \\ g_y^{n+1} &= g_\xi^{n+1}\xi_y + g_\eta^{n+1}\eta_y + g_\zeta^{n+1}\zeta_y \\ g_z^{n+1} &= g_\xi^{n+1}\xi_z + g_\eta^{n+1}\eta_z + g_\zeta^{n+1}\zeta_z \end{aligned}$$

Substituting into equation (6.10) and rearranging,

$$g_\xi^{n+1}(\xi_x^{\,2} + \xi_y^{\,2} + \xi_z^{\,2}) + g_\eta^{n+1}(\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z) + g_\zeta^{n+1}(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z) = mf$$

Solving for $g_\xi^{n+1}$,

$$\left(\frac{\partial g}{\partial \xi}\right)^{n+1} = \frac{f}{m} - \frac{1}{m^2}(\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z)\left(\frac{\partial g}{\partial \eta}\right)^{n+1} - \frac{1}{m^2}(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z)\left(\frac{\partial g}{\partial \zeta}\right)^{n+1} \tag{6.11}$$

Now, in order to incorporate this equation into the ADI solution procedure used in *Proteus*, the $\partial g/\partial\eta$ and $\partial g/\partial\zeta$ terms in equation (6.11) are lagged one level, and evaluated at time level $n$ instead of $n+1$. Strictly speaking, this introduces an $O(\Delta\tau)$ error into the solution. In practice, however, the actual error will depend

on the degree of nonorthogonality of the coordinates near the boundary. For orthogonal coordinates no error is introduced.

Using equation (6.2), and introducing difference operators and neglecting the linearization error, we can now write the linearized boundary condition as

$$\delta_\xi\left[\left(\frac{\partial g}{\partial \hat{Q}}\right)^n \Delta\hat{Q}^n\right] = \frac{f}{m} - \frac{1}{m^2}(\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z)\delta_\eta g^n - \frac{1}{m^2}(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z)\delta_\zeta g^n - \delta_\xi g^n \quad (6.12a)$$

where $\delta_\xi$ represents the one-sided difference operator to be used at the boundary. Options are available in *Proteus* to use either first-order two-point or second-order three-point differencing.

Note that the unit vector $\vec{n}$ in equation (6.9) is in the direction of increasing $\xi$. Therefore, a positive value for $f$ in equation (6.12a) indicates a flux in the direction of increasing $\xi$. Thus, a positive $f$ at $\xi = 0$ implies a flux into the computational domain, and a positive $f$ at $\xi = 1$ implies a flux out of the computational domain.

Specifying a gradient normal to a constant $\eta$ or $\zeta$ boundary is done in an exactly analogous manner. The resulting equation for an $\eta$ boundary is

$$\delta_\eta\left[\left(\frac{\partial g}{\partial \hat{Q}}\right)^n \Delta\hat{Q}^n\right] = \frac{f}{m} - \frac{1}{m^2}(\eta_x\xi_x + \eta_y\xi_y + \eta_z\xi_z)\delta_\xi g^n - \frac{1}{m^2}(\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z)\delta_\zeta g^n - \delta_\eta g^n \quad (6.12b)$$

where

$$m = \sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2}$$

For a $\zeta$ boundary the equation is

$$\delta_\zeta\left[\left(\frac{\partial g}{\partial \hat{Q}}\right)^n \Delta\hat{Q}^n\right] = \frac{f}{m} - \frac{1}{m^2}(\zeta_x\xi_x + \zeta_y\xi_y + \zeta_z\xi_z)\delta_\xi g^n - \frac{1}{m^2}(\zeta_x\eta_x + \zeta_y\eta_y + \zeta_z\eta_z)\delta_\eta g^n - \delta_\zeta g^n \quad (6.12c)$$

where

$$m = \sqrt{\zeta_x^2 + \zeta_y^2 + \zeta_z^2}$$

A positive value for $f$ in equation (6.12b) indicates a flux in the direction of increasing $\eta$. Thus, a positive $f$ at $\eta = 0$ implies a flux into the computational domain, and a positive $f$ at $\eta = 1$ implies a flux out of the computational domain.

## 6.5 LINEAR EXTRAPOLATION

Linear extrapolation from the two adjacent interior points is also available as a boundary condition. At the $\xi = 0$ boundary, where $i = 1$, this can be written as

$$g_i^{n+1} - 2g_{i+1}^{n+1} + g_{i+2}^{n+1} = 0 \quad (6.13)$$

Note that this is equivalent to setting $(\partial^2 g/\partial\xi^2)_{i+1} = 0$. Using equation (6.2), we can write the linearized boundary condition as

$$\left(\frac{\partial g}{\partial \hat{Q}}\right)_i^n \Delta\hat{Q}_i^n - 2\left(\frac{\partial g}{\partial \hat{Q}}\right)_{i+1}^n \Delta\hat{Q}_{i+1}^n + \left(\frac{\partial g}{\partial \hat{Q}}\right)_{i+2}^n \Delta\hat{Q}_{i+2}^n = -g_i^n + 2g_{i+1}^n - g_{i+2}^n \quad (6.14)$$

Analogous extrapolation boundary conditions can easily be written for the remaining boundaries.

PAGE _____ INTENTIONALLY BLANK

# 7.0 SOLUTION PROCEDURE

## 7.1 ADI ALGORITHM

The governing equations, presented in linearized matrix form as equation (4.17), are solved by an alternating direction implicit (ADI) method. The form of the ADI splitting is the same as used by Briley and McDonald (1977), and by Beam and Warming (1978). Although the split equations can be developed in more than one way, in this discussion the approximate factorization approach is used.

Letting LHS(4.17) represent the left hand side of equation (4.17), we can write

$$\text{LHS(4.17)} = \left\{ I + \frac{\theta_1 \Delta\tau}{1+\theta_2} \left[ \frac{\partial}{\partial\xi}\left( \frac{\partial \hat{E}}{\partial \hat{Q}} - \frac{\partial \hat{E}_{V_1}}{\partial \hat{Q}} \right) + \frac{\partial}{\partial\eta}\left( \frac{\partial \hat{F}}{\partial \hat{Q}} - \frac{\partial \hat{F}_{V_1}}{\partial \hat{Q}} \right) + \frac{\partial}{\partial\zeta}\left( \frac{\partial \hat{G}}{\partial \hat{Q}} - \frac{\partial \hat{G}_{V_1}}{\partial \hat{Q}} \right) \right] \right\}^n \Delta\hat{Q}^n \qquad (7.1)$$

where **I** represents the identity matrix. Note that in this equation, using the $\partial/\partial\xi$ term as an example, the notation used is meant to imply

$$\left[ \frac{\partial}{\partial\xi}\left( \frac{\partial \hat{E}}{\partial \hat{Q}} - \frac{\partial \hat{E}_{V_1}}{\partial \hat{Q}} \right) \right] \Delta\hat{Q} = \frac{\partial}{\partial\xi}\left( \frac{\partial \hat{E}}{\partial \hat{Q}} \Delta\hat{Q} - \frac{\partial \hat{E}_{V_1}}{\partial \hat{Q}} \Delta\hat{Q} \right)$$

The term in braces in equation (7.1) can be factored to give

$$\begin{aligned}
\text{LHS(4.17)} = & \left[ I + \frac{\theta_1 \Delta\tau}{1+\theta_2} \frac{\partial}{\partial\xi}\left( \frac{\partial \hat{E}}{\partial \hat{Q}} - \frac{\partial \hat{E}_{V_1}}{\partial \hat{Q}} \right) \right]^n \left[ I + \frac{\theta_1 \Delta\tau}{1+\theta_2} \frac{\partial}{\partial\eta}\left( \frac{\partial \hat{F}}{\partial \hat{Q}} - \frac{\partial \hat{F}_{V_1}}{\partial \hat{Q}} \right) \right]^n \cdot \\
& \left[ I + \frac{\theta_1 \Delta\tau}{1+\theta_2} \frac{\partial}{\partial\zeta}\left( \frac{\partial \hat{G}}{\partial \hat{Q}} - \frac{\partial \hat{G}_{V_1}}{\partial \hat{Q}} \right) \right]^n \Delta\hat{Q}^n - \left( \frac{\theta_1 \Delta\tau}{1+\theta_2} \right)^2 \left[ \frac{\partial}{\partial\xi}\left( \frac{\partial \hat{E}}{\partial \hat{Q}} - \frac{\partial \hat{E}_{V_1}}{\partial \hat{Q}} \right) \frac{\partial}{\partial\eta}\left( \frac{\partial \hat{F}}{\partial \hat{Q}} - \frac{\partial \hat{F}_{V_1}}{\partial \hat{Q}} \right) \right. \\
& \left. + \frac{\partial}{\partial\xi}\left( \frac{\partial \hat{E}}{\partial \hat{Q}} - \frac{\partial \hat{E}_{V_1}}{\partial \hat{Q}} \right) \frac{\partial}{\partial\zeta}\left( \frac{\partial \hat{G}}{\partial \hat{Q}} - \frac{\partial \hat{G}_{V_1}}{\partial \hat{Q}} \right) + \frac{\partial}{\partial\eta}\left( \frac{\partial \hat{F}}{\partial \hat{Q}} - \frac{\partial \hat{F}_{V_1}}{\partial \hat{Q}} \right) \frac{\partial}{\partial\zeta}\left( \frac{\partial \hat{G}}{\partial \hat{Q}} - \frac{\partial \hat{G}_{V_1}}{\partial \hat{Q}} \right) \right]^n \Delta\hat{Q}^n \\
& - \left( \frac{\theta_1 \Delta\tau}{1+\theta_2} \right)^3 \left[ \frac{\partial}{\partial\xi}\left( \frac{\partial \hat{E}}{\partial \hat{Q}} - \frac{\partial \hat{E}_{V_1}}{\partial \hat{Q}} \right) \frac{\partial}{\partial\eta}\left( \frac{\partial \hat{F}}{\partial \hat{Q}} - \frac{\partial \hat{F}_{V_1}}{\partial \hat{Q}} \right) \frac{\partial}{\partial\zeta}\left( \frac{\partial \hat{G}}{\partial \hat{Q}} - \frac{\partial \hat{G}_{V_1}}{\partial \hat{Q}} \right) \right]^n \Delta\hat{Q}^n \qquad (7.2)
\end{aligned}$$

The last two terms represent the splitting error. Note that, since $\Delta\hat{Q}^n = O(\Delta\tau)$, these terms can be neglected without affecting the overall time accuracy of the algorithm, even when second order time differencing is used.

Equation (4.17) can thus be rewritten in spatially factored form, and, neglecting the temporal truncation and splitting error terms, becomes

$$\left[I+\frac{\theta_1\Delta\tau}{1+\theta_2}\frac{\partial}{\partial\xi}\left(\frac{\partial\hat{\mathbf{E}}}{\partial\hat{\mathbf{Q}}}-\frac{\partial\hat{\mathbf{E}}_{V_1}}{\partial\hat{\mathbf{Q}}}\right)\right]^n\left[I+\frac{\theta_1\Delta\tau}{1+\theta_2}\frac{\partial}{\partial\eta}\left(\frac{\partial\hat{\mathbf{F}}}{\partial\hat{\mathbf{Q}}}-\frac{\partial\hat{\mathbf{F}}_{V_1}}{\partial\hat{\mathbf{Q}}}\right)\right]^n\left[I+\frac{\theta_1\Delta\tau}{1+\theta_2}\frac{\partial}{\partial\zeta}\left(\frac{\partial\hat{\mathbf{G}}}{\partial\hat{\mathbf{Q}}}-\frac{\partial\hat{\mathbf{G}}_{V_1}}{\partial\hat{\mathbf{Q}}}\right)\right]^n\Delta\hat{\mathbf{Q}}^n=$$
$$-\frac{\Delta\tau}{1+\theta_2}\left(\frac{\partial\hat{\mathbf{E}}}{\partial\xi}+\frac{\partial\hat{\mathbf{F}}}{\partial\eta}+\frac{\partial\hat{\mathbf{G}}}{\partial\zeta}\right)^n+\frac{\Delta\tau}{1+\theta_2}\left(\frac{\partial\hat{\mathbf{E}}_{V_1}}{\partial\xi}+\frac{\partial\hat{\mathbf{F}}_{V_1}}{\partial\eta}+\frac{\partial\hat{\mathbf{G}}_{V_1}}{\partial\zeta}\right)^n+\frac{(1+\theta_3)\Delta\tau}{1+\theta_2}\left(\frac{\partial\hat{\mathbf{E}}_{V_2}}{\partial\xi}+\frac{\partial\hat{\mathbf{F}}_{V_2}}{\partial\eta}+\frac{\partial\hat{\mathbf{G}}_{V_2}}{\partial\zeta}\right)^n$$
$$-\frac{\theta_3\Delta\tau}{1+\theta_2}\left(\frac{\partial\hat{\mathbf{E}}_{V_2}}{\partial\xi}+\frac{\partial\hat{\mathbf{F}}_{V_2}}{\partial\eta}+\frac{\partial\hat{\mathbf{G}}_{V_2}}{\partial\zeta}\right)^{n-1}+\frac{\theta_2}{1+\theta_2}\Delta\hat{\mathbf{Q}}^{n-1} \tag{7.3}$$

Equation (7.3) can be split into the following three-sweep sequence.

Sweep 1 ($\xi$ direction)

$$\Delta\hat{\mathbf{Q}}^*+\frac{\theta_1\Delta\tau}{1+\theta_2}\frac{\partial}{\partial\xi}\left[\left(\frac{\partial\hat{\mathbf{E}}}{\partial\hat{\mathbf{Q}}}\right)^n\Delta\hat{\mathbf{Q}}^*\right]-\frac{\theta_1\Delta\tau}{1+\theta_2}\frac{\partial}{\partial\xi}\left[\left(\frac{\partial\hat{\mathbf{E}}_{V_1}}{\partial\hat{\mathbf{Q}}}\right)^n\Delta\hat{\mathbf{Q}}^*\right]=$$
$$-\frac{\Delta\tau}{1+\theta_2}\left(\frac{\partial\hat{\mathbf{E}}}{\partial\xi}+\frac{\partial\hat{\mathbf{F}}}{\partial\eta}+\frac{\partial\hat{\mathbf{G}}}{\partial\zeta}\right)^n+\frac{\Delta\tau}{1+\theta_2}\left(\frac{\partial\hat{\mathbf{E}}_{V_1}}{\partial\xi}+\frac{\partial\hat{\mathbf{F}}_{V_1}}{\partial\eta}+\frac{\partial\hat{\mathbf{G}}_{V_1}}{\partial\zeta}\right)^n$$
$$+\frac{(1+\theta_3)\Delta\tau}{1+\theta_2}\left(\frac{\partial\hat{\mathbf{E}}_{V_2}}{\partial\xi}+\frac{\partial\hat{\mathbf{F}}_{V_2}}{\partial\eta}+\frac{\partial\hat{\mathbf{G}}_{V_2}}{\partial\zeta}\right)^n-\frac{\theta_3\Delta\tau}{1+\theta_2}\left(\frac{\partial\hat{\mathbf{E}}_{V_2}}{\partial\xi}+\frac{\partial\hat{\mathbf{F}}_{V_2}}{\partial\eta}+\frac{\partial\hat{\mathbf{G}}_{V_2}}{\partial\zeta}\right)^{n-1}$$
$$+\frac{\theta_2}{1+\theta_2}\Delta\hat{\mathbf{Q}}^{n-1} \tag{7.4a}$$

Sweep 2 ($\eta$ direction)

$$\Delta\hat{\mathbf{Q}}^{**}+\frac{\theta_1\Delta\tau}{1+\theta_2}\frac{\partial}{\partial\eta}\left[\left(\frac{\partial\hat{\mathbf{F}}}{\partial\hat{\mathbf{Q}}}\right)^n\Delta\hat{\mathbf{Q}}^{**}\right]-\frac{\theta_1\Delta\tau}{1+\theta_2}\frac{\partial}{\partial\eta}\left[\left(\frac{\partial\hat{\mathbf{F}}_{V_1}}{\partial\hat{\mathbf{Q}}}\right)^n\Delta\hat{\mathbf{Q}}^{**}\right]=\Delta\hat{\mathbf{Q}}^* \tag{7.4b}$$

Sweep 3 ($\zeta$ direction)

$$\Delta\hat{\mathbf{Q}}^n+\frac{\theta_1\Delta\tau}{1+\theta_2}\frac{\partial}{\partial\zeta}\left[\left(\frac{\partial\hat{\mathbf{G}}}{\partial\hat{\mathbf{Q}}}\right)^n\Delta\hat{\mathbf{Q}}^n\right]-\frac{\theta_1\Delta\tau}{1+\theta_2}\frac{\partial}{\partial\zeta}\left[\left(\frac{\partial\hat{\mathbf{G}}_{V_1}}{\partial\hat{\mathbf{Q}}}\right)^n\Delta\hat{\mathbf{Q}}^n\right]=\Delta\hat{\mathbf{Q}}^{**} \tag{7.4c}$$

In the above equations, $\hat{\mathbf{Q}}^*$ and $\hat{\mathbf{Q}}^{**}$ represent intermediate solutions to the governing equations.[2] It should be noted that in *Proteus*, physical (i.e, $n+1$ level) boundary conditions are used during the first two ADI

---

[2] The notation here is somewhat inconsistent. The quantity $\Delta\hat{\mathbf{Q}}^n=\hat{\mathbf{Q}}^{n+1}-\hat{\mathbf{Q}}^n$, but $\Delta\hat{\mathbf{Q}}^*=\hat{\mathbf{Q}}^*-\hat{\mathbf{Q}}^n$, not $\hat{\mathbf{Q}}^{n+1}-\hat{\mathbf{Q}}^*$. Similarly, $\Delta\hat{\mathbf{Q}}^{**}=\hat{\mathbf{Q}}^{**}-\hat{\mathbf{Q}}^n$, not $\hat{\mathbf{Q}}^{n+1}-\hat{\mathbf{Q}}^{**}$.

sweeps. This introduces an $O(\Delta\tau)$ error in $\partial\mathbf{Q}/\partial\tau$ on the boundary for unsteady flows, but no error for steady flows. This point is discussed in detail by Briley and McDonald (1980).

Applying the spatial differencing formulas of Section 5.0 results in

Sweep 1 ($\xi$ direction)

$$\Delta\hat{\mathbf{Q}}_i^* + \frac{\theta_1\Delta\tau}{(1+\theta_2)\Delta\xi}\left[ -\alpha\left(\frac{\partial\hat{\mathbf{E}}}{\partial\hat{\mathbf{Q}}}\right)_{i-1}^{n}\Delta\hat{\mathbf{Q}}_{i-1}^* + (2\alpha-1)\left(\frac{\partial\hat{\mathbf{E}}}{\partial\hat{\mathbf{Q}}}\right)_{i}^{n}\Delta\hat{\mathbf{Q}}_i^* + (1-\alpha)\left(\frac{\partial\hat{\mathbf{E}}}{\partial\hat{\mathbf{Q}}}\right)_{i+1}^{n}\Delta\hat{\mathbf{Q}}_{i+1}^*\right]$$
$$-\frac{\theta_1\Delta\tau}{(1+\theta_2)(\Delta\xi)^2}\left[(f_{i-1}+f_i)^n g_{i-1}^n\Delta\hat{\mathbf{Q}}_{i-1}^* - (f_{i-1}+2f_i+f_{i+1})^n g_i^n\Delta\hat{\mathbf{Q}}_i^* + (f_i+f_{i+1})^n g_{i+1}^n\Delta\hat{\mathbf{Q}}_{i+1}^*\right] =$$
$$-\frac{\Delta\tau}{1+\theta_2}\left(\delta_\xi\hat{\mathbf{E}}+\delta_\eta\hat{\mathbf{F}}+\delta_\zeta\hat{\mathbf{G}}\right)^n + \frac{\Delta\tau}{1+\theta_2}\left(\delta_\xi\hat{\mathbf{E}}_{V_1}+\delta_\eta\hat{\mathbf{F}}_{V_1}+\delta_\zeta\hat{\mathbf{G}}_{V_1}\right)^n$$
$$+\frac{(1+\theta_3)\Delta\tau}{1+\theta_2}\left(\delta_\xi\hat{\mathbf{E}}_{V_2}+\delta_\eta\hat{\mathbf{F}}_{V_2}+\delta_\zeta\hat{\mathbf{G}}_{V_2}\right)^n - \frac{\theta_3\Delta\tau}{1+\theta_2}\left(\delta_\xi\hat{\mathbf{E}}_{V_2}+\delta_\eta\hat{\mathbf{F}}_{V_2}+\delta_\zeta\hat{\mathbf{G}}_{V_2}\right)^{n-1}$$
$$+\frac{\theta_2}{1+\theta_2}\Delta\hat{\mathbf{Q}}^{n-1} \tag{7.5a}$$

Sweep 2 ($\eta$ direction)

$$\Delta\hat{\mathbf{Q}}_j^{**} + \frac{\theta_1\Delta\tau}{(1+\theta_2)\Delta\xi}\left[ -\alpha\left(\frac{\partial\hat{\mathbf{F}}}{\partial\hat{\mathbf{Q}}}\right)_{j-1}^{n}\Delta\hat{\mathbf{Q}}_{j-1}^{**} + (2\alpha-1)\left(\frac{\partial\hat{\mathbf{F}}}{\partial\hat{\mathbf{Q}}}\right)_{j}^{n}\Delta\hat{\mathbf{Q}}_j^{**} + (1-\alpha)\left(\frac{\partial\hat{\mathbf{F}}}{\partial\hat{\mathbf{Q}}}\right)_{j+1}^{n}\Delta\hat{\mathbf{Q}}_{j+1}^{**}\right]$$
$$-\frac{\theta_1\Delta\tau}{(1+\theta_2)(\Delta\xi)^2}\left[(f_{j-1}+f_j)^n g_{j-1}^n\Delta\hat{\mathbf{Q}}_{j-1}^{**} - (f_{j-1}+2f_j+f_{j+1})^n g_j^n\Delta\hat{\mathbf{Q}}_j^{**} + (f_j+f_{j+1})^n g_{j+1}^n\Delta\hat{\mathbf{Q}}_{j+1}^{**}\right] =$$
$$\Delta\hat{\mathbf{Q}}^* \tag{7.5b}$$

Sweep 3 ($\zeta$ direction)

$$\Delta\hat{\mathbf{Q}}_k^{n} + \frac{\theta_1\Delta\tau}{(1+\theta_2)\Delta\xi}\left[ -\alpha\left(\frac{\partial\hat{\mathbf{G}}}{\partial\hat{\mathbf{Q}}}\right)_{k-1}^{n}\Delta\hat{\mathbf{Q}}_{k-1}^{n} + (2\alpha-1)\left(\frac{\partial\hat{\mathbf{G}}}{\partial\hat{\mathbf{Q}}}\right)_{k}^{n}\Delta\hat{\mathbf{Q}}_k^{n} + (1-\alpha)\left(\frac{\partial\hat{\mathbf{G}}}{\partial\hat{\mathbf{Q}}}\right)_{k+1}^{n}\Delta\hat{\mathbf{Q}}_{k+1}^{n}\right]$$
$$-\frac{\theta_1\Delta\tau}{(1+\theta_2)(\Delta\xi)^2}\left[(f_{k-1}+f_k)^n g_{k-1}^n\Delta\hat{\mathbf{Q}}_{k-1}^{n} - (f_{k-1}+2f_k+f_{k+1})^n g_k^n\Delta\hat{\mathbf{Q}}_k^{n} + (f_k+f_{k+1})^n g_{k+1}^n\Delta\hat{\mathbf{Q}}_{k+1}^{n}\right] =$$
$$\Delta\hat{\mathbf{Q}}^{**} \tag{7.5c}$$

The subscripts $i$, $j$, and $k$ represent grid point indices in the $\xi$, $\eta$, and $\zeta$ directions. For notational convenience, terms without an explicitly written $i$, $j$, or $k$ subscript are understood to be at $i$, $j$, or $k$. In the viscous terms on the left hand side, $f$ is the coefficient of $\partial/\partial\xi$ (or $\partial/\partial\eta$ or $\partial/\partial\zeta$, depending on the sweep) in the $\partial\hat{\mathbf{E}}_{V_1}/\partial\hat{\mathbf{Q}}$ (or $\partial\hat{\mathbf{F}}_{V_1}/\partial\hat{\mathbf{Q}}$ or $\partial\hat{\mathbf{G}}_{V_1}/\partial\hat{\mathbf{Q}}$) Jacobian coefficient matrix. Similarly, $g$ is the term in the parentheses following $\partial/\partial\xi$ (or $\partial/\partial\eta$ or $\partial/\partial\zeta$) in the $\partial\hat{\mathbf{E}}_{V_1}/\partial\hat{\mathbf{Q}}$ (or $\partial\hat{\mathbf{F}}_{V_1}/\partial\hat{\mathbf{Q}}$ or $\partial\hat{\mathbf{G}}_{V_1}/\partial\hat{\mathbf{Q}}$) Jacobian coefficient matrix. Equations (7.5a) through (7.5c) represent the three-sweep alternating direction implicit (ADI) algorithm used to advance the solution from time level $n$ to $n+1$.

## 7.2 MATRIX INVERSION PROCEDURE

### 7.2.1 Non-Periodic Boundary Conditions

The complete set of algebraic equations for the first ADI sweep with non-periodic boundary conditions can be written in the following block matrix form.[3]

$$\begin{bmatrix} \mathbf{B}'_1 & \mathbf{C}'_1 & \mathbf{A}'_1 & & & & & & & \\ \mathbf{A}_2 & \mathbf{B}_2 & \mathbf{C}_2 & & & & & & & \\ & \mathbf{A}_3 & \mathbf{B}_3 & \mathbf{C}_3 & & & & & & \\ & & & \cdot & & & & & & \\ & & & & \cdot & & & & & \\ & & & & & \cdot & & & & \\ & & & & & \mathbf{A}_{N_1-2} & \mathbf{B}_{N_1-2} & \mathbf{C}_{N_1-2} & & \\ & & & & & & \mathbf{A}_{N_1-1} & \mathbf{B}_{N_1-1} & \mathbf{C}_{N_1-1} & \\ & & & & & & \mathbf{C}'_{N_1} & \mathbf{A}'_{N_1} & \mathbf{B}'_{N_1} & \end{bmatrix} \begin{bmatrix} \Delta\hat{\mathbf{Q}}^*_1 \\ \Delta\hat{\mathbf{Q}}^*_2 \\ \Delta\hat{\mathbf{Q}}^*_3 \\ \cdot \\ \cdot \\ \cdot \\ \Delta\hat{\mathbf{Q}}^*_{N_1-2} \\ \Delta\hat{\mathbf{Q}}^*_{N_1-1} \\ \Delta\hat{\mathbf{Q}}^*_{N_1} \end{bmatrix} = \begin{bmatrix} \mathbf{S}'_1 \\ \mathbf{S}_2 \\ \mathbf{S}_3 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{S}_{N_1-2} \\ \mathbf{S}_{N_1-1} \\ \mathbf{S}'_{N_1} \end{bmatrix} \tag{7.6}$$

These equations result from the application of equation (7.5a) for $i = 2$ to $N_1 - 1$, with boundary conditions added at $i = 1$ and $i = N_1$. The parameter $\Delta\hat{\mathbf{Q}}^*$ is the $N_{eq}$-element vector containing the unknown dependent variables; $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are the $N_{eq} \times N_{eq}$ coefficient submatrices at $i-1$, $i$, and $i+1$, respectively; and $\mathbf{S}$ is the $N_{eq}$-element subvector containing the explicit source terms. Also, $\mathbf{A}'$, $\mathbf{B}'$, and $\mathbf{C}'$ are the coefficient submatrices and $\mathbf{S}'$ the source term subvector for the boundary conditions. A variety of boundary conditions may be used. They are described briefly in Section 6.0, and in greater detail in Volumes 2 and 3.

Note that the equations at the boundaries may contain coefficients at the boundary point and the two adjacent interior points. This occurs, for example, when extrapolation or second order gradient boundary conditions are specified. As written, therefore, the coefficient matrix in equation (7.6) is not block tridiagonal. However, $\mathbf{A}'_1$ can be eliminated by multiplying the second row of the matrix by $\mathbf{A}'_1\,\mathbf{C}_2^{-1}$ and subtracting from the first row. $\mathbf{C}'_{N_1}$ can be eliminated in a similar manner. Doing this, we define

$$\begin{aligned} \mathbf{B}_1 &= \mathbf{B}'_1 - \mathbf{A}'_1\,\mathbf{C}_2^{-1}\mathbf{A}_2 \\ \mathbf{C}_1 &= \mathbf{C}'_1 - \mathbf{A}'_1\,\mathbf{C}_2^{-1}\mathbf{B}_2 \\ \mathbf{S}_1 &= \mathbf{S}'_1 - \mathbf{A}'_1\,\mathbf{C}_2^{-1}\mathbf{S}_2 \end{aligned} \tag{7.7}$$

and

$$\begin{aligned} \mathbf{A}_{N_1} &= \mathbf{A}'_{N_1} - \mathbf{C}'_{N_1}\,\mathbf{A}_{N_1-1}^{-1}\mathbf{B}_{N_1-1} \\ \mathbf{B}_{N_1} &= \mathbf{B}'_{N_1} - \mathbf{C}'_{N_1}\,\mathbf{A}_{N_1-1}^{-1}\mathbf{C}_{N_1-1} \\ \mathbf{S}_{N_1} &= \mathbf{S}'_{N_1} - \mathbf{C}'_{N_1}\,\mathbf{A}_{N_1-1}^{-1}\mathbf{S}_{N_1-1} \end{aligned} \tag{7.8}$$

---

[3] Although this discussion is written for the first ADI sweep, an exactly analogous procedure is followed for the second and third sweeps.

The set of algebraic equations solved during the first ADI sweep can now be written as

$$\begin{bmatrix} \mathbf{B}_1 & \mathbf{C}_1 & & & & & & & \\ \mathbf{A}_2 & \mathbf{B}_2 & \mathbf{C}_2 & & & & & & \\ & \mathbf{A}_3 & \mathbf{B}_3 & \mathbf{C}_3 & & & & & \\ & & \cdot & & & & & & \\ & & & \cdot & & & & & \\ & & & & \cdot & & & & \\ & & & & & \mathbf{A}_{N_1-2} & \mathbf{B}_{N_1-2} & \mathbf{C}_{N_1-2} & \\ & & & & & & \mathbf{A}_{N_1-1} & \mathbf{B}_{N_1-1} & \mathbf{C}_{N_1-1} \\ & & & & & & & \mathbf{A}_{N_1} & \mathbf{B}_{N_1} \end{bmatrix} \begin{bmatrix} \Delta\hat{\mathbf{Q}}_1^* \\ \Delta\hat{\mathbf{Q}}_2^* \\ \Delta\hat{\mathbf{Q}}_3^* \\ \cdot \\ \cdot \\ \cdot \\ \Delta\hat{\mathbf{Q}}_{N_1-2}^* \\ \Delta\hat{\mathbf{Q}}_{N_1-1}^* \\ \Delta\hat{\mathbf{Q}}_{N_1}^* \end{bmatrix} = \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \\ \mathbf{S}_3 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{S}_{N_1-2} \\ \mathbf{S}_{N_1-1} \\ \mathbf{S}_{N_1} \end{bmatrix} \tag{7.9}$$

Since the coefficient matrix is now block tridiagonal, the equations can be solved using the block matrix version of the Thomas algorithm (e.g., see Anderson, Tannehill, and Pletcher, 1984). The procedure can be summarized as follows:

1. Define $\mathbf{D}_1 = \mathbf{B}_1$.
2. Compute $\mathbf{E}_1 = \mathbf{D}_1^{-1}\mathbf{C}_1$ and $\Delta\hat{\mathbf{Q}}_1' = \mathbf{D}_1^{-1}\mathbf{S}_1$.
3. For $i = 2$ to $N_1$, compute
   $\mathbf{D}_i = \mathbf{B}_i - \mathbf{A}_i\mathbf{E}_{i-1}$
   $\mathbf{E}_i = \mathbf{D}_i^{-1}\mathbf{C}_i$
   $\Delta\hat{\mathbf{Q}}_i' = \mathbf{D}_i^{-1}(\mathbf{S}_i - \mathbf{A}_i\Delta\hat{\mathbf{Q}}_{i-1}')$
   (Actually, $\mathbf{E}_i$ is only needed for $i = 2$ to $N_1 - 1$).
4. Then, set $\Delta\hat{\mathbf{Q}}_{N_1} = \Delta\hat{\mathbf{Q}}'_{N_1}$.
5. Finally, for $i = N_1 - 1$ to 1, compute $\Delta\hat{\mathbf{Q}}_i = \Delta\hat{\mathbf{Q}}_i' - \mathbf{E}_i\Delta\hat{\mathbf{Q}}_{i+1}$.

In the *Proteus* code, in step 2 $\mathbf{E}_1$ and $\Delta\hat{\mathbf{Q}}_1'$ are actually obtained by solving $\mathbf{D}_1\mathbf{E}_1 = \mathbf{C}_1$ and $\mathbf{D}_1\Delta\hat{\mathbf{Q}}_1' = \mathbf{S}_1$ using LU decomposition of $\mathbf{D}$. A similar procedure is used to compute $\mathbf{E}_i$ and $\Delta\hat{\mathbf{Q}}_i'$ in step 3.

### 7.2.2 Spatially Periodic Boundary Conditions

In computational coordinates a spatially periodic boundary condition in the $\xi$ direction may be represented as shown in Figure 7.1.[4]

[4] As in Section 7.2.1, this discussion is written for the first ADI sweep, but an exactly analogous procedure is followed for spatially periodic boundary conditions in the second and third sweeps.

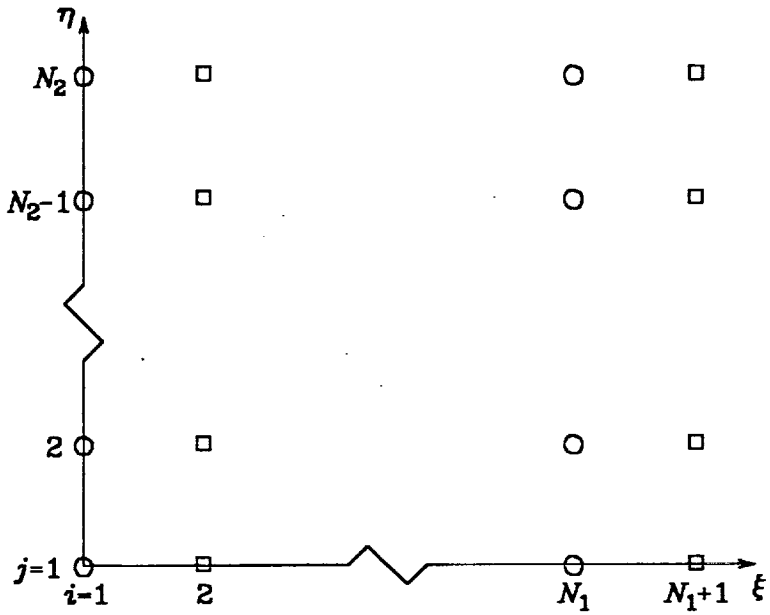


**Figure 7.1 - Spatially periodic boundary condition.**

The grid points along the $i = 1$ and $i = N_1$ lines are "similar" in the geometric sense, and have the same flow solution. Therefore, for a spatially periodic boundary condition in the $\xi$ direction, $\hat{Q}_1 = \hat{Q}_{N_1}$.

To implement this boundary condition, an additional set of points is added at $i = N_1 + 1$, setting $\hat{Q}_{N_1+1} = \hat{Q}_2$. This allows us to use central differencing in the $\xi$ direction at $i = N_1$, computing the coefficients in the same way as at the interior points.

The resulting set of algebraic equations will consist of $N_1 - 1$ equations (for $i = 2$ to $N_1$), with $N_1 + 1$ unknowns. The block coefficient matrix thus has $N_1 - 1$ rows and $N_1 + 1$ columns, as follows:

$$\begin{bmatrix} \mathbf{A}_2 & \mathbf{B}_2 & \mathbf{C}_2 & & & & & & & & & \\ & \mathbf{A}_3 & \mathbf{B}_3 & \mathbf{C}_3 & & & & & & & & \\ & & \mathbf{A}_4 & \mathbf{B}_4 & \mathbf{C}_4 & & & & & & & \\ & & & \cdot & & & & & & & & \\ & & & & \cdot & & & & & & & \\ & & & & & \cdot & & & & & & \\ & & & & & & \mathbf{A}_{N_1-2} & \mathbf{B}_{N_1-2} & \mathbf{C}_{N_1-2} & & \\ & & & & & & & \mathbf{A}_{N_1-1} & \mathbf{B}_{N_1-1} & \mathbf{C}_{N_1-1} & \\ & & & & & & & & \mathbf{A}_{N_1} & \mathbf{B}_{N_1} & \mathbf{C}_{N_1} \end{bmatrix} \begin{bmatrix} \Delta\hat{\mathbf{Q}}_1^* \\ \Delta\hat{\mathbf{Q}}_2^* \\ \Delta\hat{\mathbf{Q}}_3^* \\ \Delta\hat{\mathbf{Q}}_4^* \\ \cdot \\ \cdot \\ \cdot \\ \Delta\hat{\mathbf{Q}}_{N_1-2}^* \\ \Delta\hat{\mathbf{Q}}_{N_1-1}^* \\ \Delta\hat{\mathbf{Q}}_{N_1}^* \\ \Delta\hat{\mathbf{Q}}_{N_1+1}^* \end{bmatrix} = \begin{bmatrix} \mathbf{S}_2 \\ \mathbf{S}_3 \\ \mathbf{S}_4 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{S}_{N_1-2} \\ \mathbf{S}_{N_1-1} \\ \mathbf{S}_{N_1} \end{bmatrix} \tag{7.10}$$

These equations result from the application of equation (7.5a) for $i = 2$ to $N_1$. As in the previous section, parameter $\Delta\hat{\mathbf{Q}}^*$ is the $N_{eq}$-element vector containing the unknown dependent variables; **A**, **B**, and **C** are the $N_{eq} \times N_{eq}$ coefficient submatrices at $i - 1$, $i$, and $i + 1$, respectively; and **S** is the $N_{eq}$-element subvector containing the explicit source terms.

Since $\hat{\mathbf{Q}}_1 = \hat{\mathbf{Q}}_{N_1}$ and $\hat{\mathbf{Q}}_2 = \hat{\mathbf{Q}}_{N_1+1}$, equation (7.10) can be rewritten with $N_1 - 1$ unknowns as:

$$\begin{bmatrix} \mathbf{B}_2 & \mathbf{C}_2 & & & & & & & & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{B}_3 & \mathbf{C}_3 & & & & & & & \\ & \mathbf{A}_4 & \mathbf{B}_4 & \mathbf{C}_4 & & & & & & \\ & & & \cdot & & & & & & \\ & & & & \cdot & & & & & \\ & & & & & \cdot & & & & \\ & & & & & & \mathbf{A}_{N_1-2} & \mathbf{B}_{N_1-2} & \mathbf{C}_{N_1-2} & \\ & & & & & & & \mathbf{A}_{N_1-1} & \mathbf{B}_{N_1-1} & \mathbf{C}_{N_1-1} \\ \mathbf{C}_{N_1} & & & & & & & & \mathbf{A}_{N_1} & \mathbf{B}_{N_1} \end{bmatrix} \begin{bmatrix} \Delta\hat{\mathbf{Q}}_2^* \\ \Delta\hat{\mathbf{Q}}_3^* \\ \Delta\hat{\mathbf{Q}}_4^* \\ \cdot \\ \cdot \\ \cdot \\ \Delta\hat{\mathbf{Q}}_{N_1-2}^* \\ \Delta\hat{\mathbf{Q}}_{N_1-1}^* \\ \Delta\hat{\mathbf{Q}}_{N_1}^* \end{bmatrix} = \begin{bmatrix} \mathbf{S}_2 \\ \mathbf{S}_3 \\ \mathbf{S}_4 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{S}_{N_1-2} \\ \mathbf{S}_{N_1-1} \\ \mathbf{S}_{N_1} \end{bmatrix} \tag{7.11}$$

An efficient algorithm to solve this system can be derived that is similar to the Thomas algorithm for block tridiagonal systems. The procedure can be summarized as follows:

1. Define $\mathbf{D}_2 = \mathbf{B}_2$ and $\mathbf{F}_2 = \mathbf{C}_{N_1}$.

2. Compute $\mathbf{E}_2 = \mathbf{D}_2^{-1}\mathbf{C}_2$, $\mathbf{G}_2 = \mathbf{D}_2^{-1}\mathbf{A}_2$, and $\Delta\hat{\mathbf{Q}}_2' = \mathbf{D}_2^{-1}\mathbf{S}_2$.

3. For $i = 3$ to $N_1 - 1$, compute

$$\mathbf{D}_i = \mathbf{B}_i - \mathbf{A}_i\mathbf{E}_{i-1}$$

$$\mathbf{E}_i = \mathbf{D}_i^{-1}\mathbf{C}_i$$

$$\mathbf{F}_i = -\mathbf{F}_{i-1}\mathbf{E}_{i-1}$$

$$\mathbf{G}_i = -\mathbf{D}_i^{-1}\mathbf{A}_i\mathbf{G}_{i-1}$$

$$\Delta\hat{\mathbf{Q}}'_i = \mathbf{D}_i^{-1}(\mathbf{S}_i - \mathbf{A}_i\Delta\hat{\mathbf{Q}}'_{i-1})$$

4. Compute

$$\mathbf{G}_{N_1-1} = \mathbf{D}_{N_1-1}^{-1}(\mathbf{C}_{N_1-1} - \mathbf{A}_{N_1-1}\mathbf{G}_{N_1-2})$$

$$\mathbf{F}_{N_1-1} = \mathbf{A}_{N_1} - \mathbf{F}_{N_1-2}\mathbf{E}_{N_1-2}$$

$$\mathbf{D}_{N_1} = \mathbf{B}_{N_1} - \sum_{i=2}^{N_1-1}\mathbf{F}_i\mathbf{G}_i$$

$$\Delta\hat{\mathbf{Q}}'_{N_1} = \mathbf{D}_{N_1}^{-1}\left(\mathbf{S}_{N_1} - \sum_{i=2}^{N_1-1}\mathbf{F}_i\Delta\hat{\mathbf{Q}}'_i\right)$$

5. Then, set $\Delta\hat{\mathbf{Q}}_{N_1} = \Delta\hat{\mathbf{Q}}'_{N_1}$.

6. Compute $\Delta\hat{\mathbf{Q}}_{N_1-1} = \Delta\hat{\mathbf{Q}}'_{N_1-1} - \mathbf{G}_{N_1-1}\Delta\hat{\mathbf{Q}}_{N_1}$.

7. Finally, for $i = N_1 - 2$ to 2, compute $\Delta\hat{\mathbf{Q}}_i = \Delta\hat{\mathbf{Q}}'_i - \mathbf{E}_i\Delta\hat{\mathbf{Q}}_{i+1} - \mathbf{G}_i\Delta\hat{\mathbf{Q}}_{N_1}$.

In the *Proteus* code, in step 2 $\mathbf{E}_2$, $\mathbf{G}_2$, and $\Delta\hat{\mathbf{Q}}'_2$ are actually obtained by solving $\mathbf{D}_2\mathbf{E}_2 = \mathbf{C}_2$, $\mathbf{D}_2\mathbf{G}_2 = \mathbf{A}_2$, and $\mathbf{D}_2\Delta\hat{\mathbf{Q}}'_2 = \mathbf{S}_2$ using LU decomposition of $\mathbf{D}$. A similar procedure is used to compute $\mathbf{E}_i$, $\mathbf{G}_i$, and $\Delta\hat{\mathbf{Q}}'_i$ in step 3, and $\mathbf{G}_{N_1-1}$ and $\Delta\hat{\mathbf{Q}}'_{N_1}$ in step 4.

## 7.3 UPDATING BOUNDARY VALUES

With the ADI algorithm described in Section 7.1, if gradient or extrapolation boundary conditions are used for the first or second sweep, the boundary values from the first two sweeps must be updated after the third sweep. This point is easiest to illustrate by looking at a figure.

In Figure 7.2, a $4 \times 4 \times 4$ grid is shown in computational space for a three-dimensional problem. This example assumes that no spatially periodic boundary conditions are being used. The circles represent grid points at which the intermediate values $\hat{\mathbf{Q}}^*$ are computed during the first ADI sweep. These include the boundary points at $\xi = 0$ and $\xi = 1$. The squares represent grid points at which the intermediate values $\hat{\mathbf{Q}}^{**}$ are computed during the second ADI sweep, including the boundary points at $\eta = 0$ and $\eta = 1$. The triangles represent grid points at which the final values $\hat{\mathbf{Q}}^{n+1}$ are computed during the third ADI sweep, including the boundary points at $\zeta = 0$ and $\zeta = 1$. If gradient or extrapolation boundary conditions are used during the first and/or second sweep, so that the boundary values depend on the interior values, then the intermediate values at the $\xi$ and/or $\eta$ boundaries must be updated after the third sweep to be consistent with the final values at the interior points.

To do this, after the last sweep the difference equations are rewritten and solved at the $\xi$ and $\eta$ boundaries. At the $\xi = 0$ boundary,

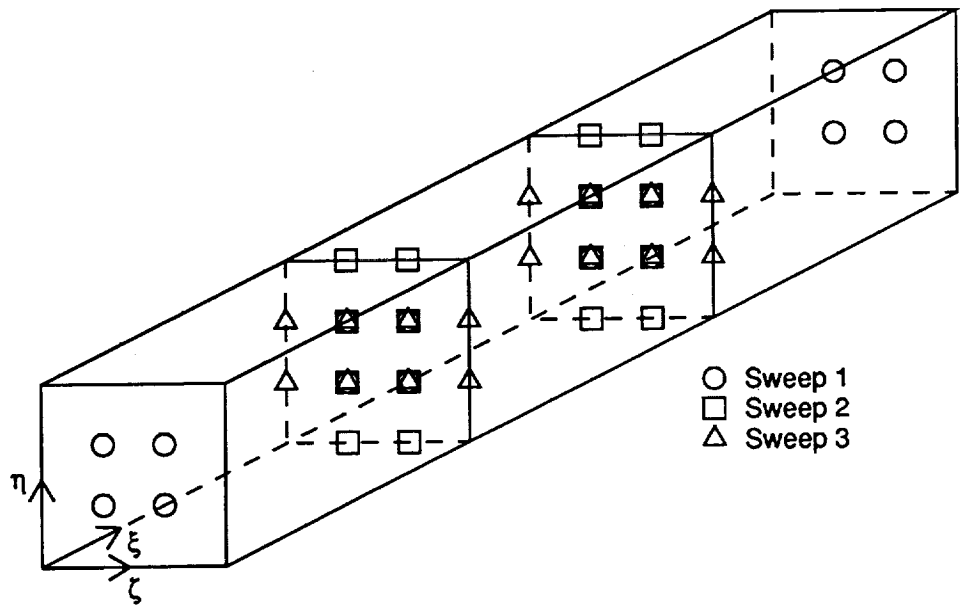


**Figure 7.2 - Updating boundary values.**

$$\mathbf{B}'_1{}^n\Delta\hat{\mathbf{Q}}_1^n + \mathbf{C}'_1{}^n\Delta\hat{\mathbf{Q}}_2^n + \mathbf{A}'_1{}^n\Delta\hat{\mathbf{Q}}_3^n = \mathbf{S}'_1{}^n \tag{7.12}$$

The subscripts refer to the value of $i$, the index in the $\xi$ direction. This equation is applied for $j = 2$ to $N_2 - 1$ in the $\eta$ direction, and for $k = 2$ to $N_3 - 1$ in the $\zeta$ direction. For notational convenience, however, the subscripts $j$ and $k$ have been omitted.

All the terms in equation (7.12) are known except $\Delta\hat{\mathbf{Q}}_1^n$. Solving,

$$\Delta\hat{\mathbf{Q}}_1^n = (\mathbf{B}'_1{}^n)^{-1}(\mathbf{S}'_1{}^n - \mathbf{C}'_1{}^n\Delta\hat{\mathbf{Q}}_2^n - \mathbf{A}'_1{}^n\Delta\hat{\mathbf{Q}}_3^n) \tag{7.13}$$

At the $\xi = 1$ boundary,

$$\mathbf{C}'_{N_1}{}^n\Delta\hat{\mathbf{Q}}_{N_1-2}^n + \mathbf{A}'_{N_1}{}^n\Delta\hat{\mathbf{Q}}_{N_1-1}^n + \mathbf{B}'_{N_1}{}^n\Delta\hat{\mathbf{Q}}_{N_1}^n = \mathbf{S}'_{N_1}{}^n \tag{7.14}$$

$$\Delta\hat{\mathbf{Q}}_{N_1}^n = (\mathbf{B}'_{N_1}{}^n)^{-1}(\mathbf{S}'_{N_1}{}^n - \mathbf{C}'_{N_1}{}^n\Delta\hat{\mathbf{Q}}_{N_1-2}^n - \mathbf{A}'_{N_1}{}^n\Delta\hat{\mathbf{Q}}_{N_1-1}^n) \tag{7.15}$$

An analogous procedure is followed to update values at the $\eta = 0$ and $\eta = 1$ boundaries.

Finally, note from Figure 7.2 that new values are not computed at the corners or edges of the computational domain during the solution algorithm. To make the edge values consistent with the rest of the flow field, in *Proteus* they are defined using the computed values at adjacent points. For example, at each $\zeta$ location along a $\xi$-$\eta$ edge (i.e., one of the four lines of intersection between the $\xi$ and $\eta$ boundary planes), the density $\rho$ and total energy $E_T$ are arbitrarily defined by linearly extrapolating from the two adjacent points in the $\xi$ and $\eta$ coordinate directions, and averaging the two results. The edge values of the velocities are updated by doing the same type of extrapolation. Instead of averaging, however, the extrapolated velocity whose absolute value is lower is used. This was done to maintain no-slip conditions at duct inlets and exits. The values at corners, where all three boundary planes intersect, are determined in an exactly analogous manner.

Updating boundary values from the first two sweeps is complicated somewhat when spatially periodic boundary conditions are used. Details are presented in the description of subroutine BVUP in Volume 3.

# 8.0 ARTIFICIAL VISCOSITY

With the numerical algorithm of Section 7.0, high frequency nonlinear instabilities can appear as the solution develops. For example, in high Reynolds number flows oscillations can result from the odd-even decoupling inherent in the use of second order central differencing for the inviscid terms. In addition, physical phenomena such as shock waves can cause instabilities when they are captured by the finite difference algorithm. Artificial viscosity, or smoothing, is normally added to the solution algorithm to suppress these high frequency instabilities. Two artificial viscosity models are currently available in the *Proteus* computer code - a constant coefficient model used by Steger (1978), and the nonlinear coefficient model of Jameson, Schmidt, and Turkel (1981). The implementation of these models in generalized nonorthogonal coordinates is described by Pulliam (1986b).

## 8.1 CONSTANT COEFFICIENT ARTIFICIAL VISCOSITY

The constant coefficient model uses a combination of explicit and implicit artificial viscosity. The standard explicit smoothing uses fourth order differences, and damps the high frequency nonlinear instabilities. Second order explicit smoothing, while not used by Steger or Pulliam, is also available in *Proteus*. It provides more smoothing than the fourth order smoothing but introduces a larger error, and is therefore not used as often. The implicit smoothing is second order and is intended to extend the linear stability bound of the fourth order explicit smoothing.

The explicit artificial viscosity is implemented in the numerical algorithm by adding the following terms to the right hand side of equation (7.5a) (i.e., the source term for the first ADI sweep.)

$$\frac{\varepsilon_E^{(2)}\Delta\tau}{J}(\nabla_\xi\Delta_\xi\mathbf{Q}+\nabla_\eta\Delta_\eta\mathbf{Q}+\nabla_\zeta\Delta_\zeta\mathbf{Q})-\frac{\varepsilon_E^{(4)}\Delta\tau}{J}[(\nabla_\xi\Delta_\xi)^2\mathbf{Q}+(\nabla_\eta\Delta_\eta)^2\mathbf{Q}+(\nabla_\zeta\Delta_\zeta)^2\mathbf{Q}] \tag{8.1}$$

where $\varepsilon_E^{(2)}$ and $\varepsilon_E^{(4)}$ are the second- and fourth-order explicit artificial viscosity coefficients. The symbols $\nabla$ and $\Delta$ are backward and forward first difference operators. Thus,

$$\begin{aligned}\nabla_\xi\mathbf{Q}_i &= \mathbf{Q}_i-\mathbf{Q}_{i-1}\\ \Delta_\xi\mathbf{Q}_i &= \mathbf{Q}_{i+1}-\mathbf{Q}_i\\ \nabla_\xi\Delta_\xi\mathbf{Q}_i &= \mathbf{Q}_{i+1}-2\mathbf{Q}_i+\mathbf{Q}_{i-1}\\ (\nabla_\xi\Delta_\xi)^2\mathbf{Q}_i &= \mathbf{Q}_{i+2}-4\mathbf{Q}_{i+1}+6\mathbf{Q}_i-4\mathbf{Q}_{i-1}+\mathbf{Q}_{i-2}\end{aligned}$$

Equivalent formulas are used for differences in the $\eta$ and $\zeta$ directions.

A few details should be noted at this point. First, the sign in front of the artificial viscosity term being added to equation (7.5a) depends on the sign of the "$i$" term in the difference formula. For damping, that term must be negative when added to the right hand side of the equations (i.e., explicit artificial viscosity), and positive when added to the left hand side (i.e., implicit artificial viscosity.) See Anderson, Tannehill, and Pletcher (1984) for details. Second, the terms being added are differences only, and not finite difference approximations to derivatives. They are therefore not divided by $\Delta\xi$, etc. Third, the variables being differenced are $\mathbf{Q}$, not $\hat{\mathbf{Q}}$. As noted by Pulliam (1986b), scaling the artificial viscosity terms by $1/J$ makes them consistent with the form of the remaining terms in the equations. Fourth, the terms are also scaled by $\Delta\tau$. This makes the steady state solution independent of the time step size (Pulliam, 1986b). And finally, note that the fourth-order difference formula cannot be used at grid points adjacent to boundaries. At these points, therefore, the appropriate fourth-order term in expression (8.1) is replaced by a second order term. Thus, for points adjacent to the $\xi = 0$ and $\xi = 1$ boundaries, $-\varepsilon_E^{(4)}\Delta\tau[(\nabla_\xi\Delta_\xi)^2\mathbf{Q}]/J$ is replaced by

$$+\frac{\varepsilon_E^{(4)}\Delta\tau}{J}\nabla_\xi\Delta_\xi Q \tag{8.2}$$

A similar expression is used at points adjacent to the $\eta$ and $\zeta$ boundaries.

The implicit artificial viscosity is implemented by adding the following terms to the left hand side of the equations specified.

$$\begin{aligned}
&-\frac{\varepsilon_I\Delta\tau}{J}[\nabla_\xi\Delta_\xi(J\Delta\hat{Q}^*)] && \text{to equation (7.5a)}\\
&-\frac{\varepsilon_I\Delta\tau}{J}[\nabla_\eta\Delta_\eta(J\Delta\hat{Q}^{**})] && \text{to equation (7.5b)}\\
&-\frac{\varepsilon_I\Delta\tau}{J}[\nabla_\zeta\Delta_\zeta(J\Delta\hat{Q}^{n})] && \text{to equation (7.5c)}
\end{aligned} \tag{8.3}$$

Note that the addition of the artificial viscosity terms, in effect, changes the original governing partial differential equations. At steady state, the difference equations with the artificial viscosity terms added actually correspond to the following differential equations.[5]

$$\frac{\partial\hat{E}}{\partial\xi}+\frac{\partial\hat{F}}{\partial\eta}+\frac{\partial\hat{G}}{\partial\zeta}=\frac{\partial\hat{E}_V}{\partial\xi}+\frac{\partial\hat{F}_V}{\partial\eta}+\frac{\partial\hat{G}_V}{\partial\zeta}+\frac{\varepsilon_E^{(2)}}{J}\left[(\Delta\xi)^2\frac{\partial^2(J\hat{Q})}{\partial\xi^2}+(\Delta\eta)^2\frac{\partial^2(J\hat{Q})}{\partial\eta^2}+(\Delta\zeta)^2\frac{\partial^2(J\hat{Q})}{\partial\zeta^2}\right]$$
$$-\frac{\varepsilon_E^{(4)}}{J}\left[(\Delta\xi)^4\frac{\partial^4(J\hat{Q})}{\partial\xi^4}+(\Delta\eta)^4\frac{\partial^4(J\hat{Q})}{\partial\eta^4}+(\Delta\zeta)^4\frac{\partial^4(J\hat{Q})}{\partial\zeta^4}\right]$$

The implicit terms do not appear, since they difference $\Delta\hat{Q}$, and in the steady form of the equations $\Delta\hat{Q}=0$. The artificial viscosity terms do not represent anything physical. The coefficients should therefore be as small as possible, but still large enough to damp any instabilities. Although optimum values will vary from problem to problem, recommended levels are $\varepsilon_E^{(4)}=O(1)$ and $\varepsilon_I=2\varepsilon_E^{(4)}$ (Pulliam, 1986b). The recommended level for $\varepsilon_E^{(2)}$, when used, is $\varepsilon_E^{(2)}=O(1)$.

## 8.2 NONLINEAR COEFFICIENT ARTIFICIAL VISCOSITY

The nonlinear coefficient artificial viscosity model is strictly explicit. Using the model as described by Pulliam (1986b), but in the current notation, the following terms are added to the right hand side of equation (7.5a).

$$\begin{aligned}
&\nabla_\xi\left\{\left[\left(\frac{\psi}{J}\right)_{i+1}+\left(\frac{\psi}{J}\right)_i\right](\varepsilon_\xi^{(2)}\Delta_\xi Q-\varepsilon_\xi^{(4)}\Delta_\xi\nabla_\xi\Delta_\xi Q)_i\right\}\\
&+\nabla_\eta\left\{\left[\left(\frac{\psi}{J}\right)_{j+1}+\left(\frac{\psi}{J}\right)_j\right](\varepsilon_\eta^{(2)}\Delta_\eta Q-\varepsilon_\eta^{(4)}\Delta_\eta\nabla_\eta\Delta_\eta Q)_j\right\}\\
&+\nabla_\zeta\left\{\left[\left(\frac{\psi}{J}\right)_{k+1}+\left(\frac{\psi}{J}\right)_k\right](\varepsilon_\zeta^{(2)}\Delta_\zeta Q-\varepsilon_\zeta^{(4)}\Delta_\zeta\nabla_\zeta\Delta_\zeta Q)_k\right\}
\end{aligned} \tag{8.4}$$

---

[5] These equations represent the use of the constant coefficient artificial viscosity model presented in this section. The nonlinear coefficient model to be presented in Section 8.2 is more complicated, but the same principle applies.

The difference operation $\Delta_\xi \nabla_\xi \Delta_\xi \mathbf{Q}$ is given by

$$\Delta_\xi \nabla_\xi \Delta_\xi \mathbf{Q}_i = \mathbf{Q}_{i+2} - 3\mathbf{Q}_{i+1} + 3\mathbf{Q}_i - \mathbf{Q}_{i-1}$$

In the expression (8.4), $\psi$ is defined as

$$\psi = \psi_x + \psi_y + \psi_z \tag{8.5}$$

where $\psi_x$, $\psi_y$ and $\psi_z$ are spectral radii defined by[6]

$$\begin{aligned}
\psi_x &= \frac{|U| + a\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}}{\Delta\xi} \\
\psi_y &= \frac{|V| + a\sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2}}{\Delta\eta} \\
\psi_z &= \frac{|W| + a\sqrt{\zeta_x^2 + \zeta_y^2 + \zeta_z^2}}{\Delta\zeta}
\end{aligned} \tag{8.6}$$

Here $U$, $V$, and $W$ are the contravariant velocities without metric normalization, defined by

$$\begin{aligned}
U &= \xi_t + \xi_x u + \xi_y v + \xi_z w \\
V &= \eta_t + \eta_x u + \eta_y v + \eta_z w \\
W &= \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w
\end{aligned} \tag{8.7}$$

and $a = \sqrt{\gamma RT}$, the speed of sound.

The parameters $\varepsilon^{(2)}$ and $\varepsilon^{(4)}$ are the second- and fourth-order artificial viscosity coefficients. Instead of being specified directly by the user, as they are in the constant coefficient model, in the nonlinear coefficient model they are a function of the pressure field. For the coefficients of the $\xi$ direction differences,

$$\left(\varepsilon_\xi^{(2)}\right)_i = \kappa_2 \Delta\tau \max(\sigma_{i+1},\ \sigma_i,\ \sigma_{i-1}) \tag{8.8a}$$

$$\left(\varepsilon_\xi^{(4)}\right)_i = \max\left[0,\ \kappa_4 \Delta\tau - \left(\varepsilon_\xi^{(2)}\right)_i\right] \tag{8.8b}$$

where

$$\sigma_i = \left| \frac{p_{i+1} - 2p_i + p_{i-1}}{p_{i+1} + 2p_i + p_{i-1}} \right| \tag{8.9}$$

Similar formulas are used for the coefficients of the $\eta$ and $\zeta$ direction differences.

The parameter $\sigma$ is a pressure gradient scaling parameter that increases the amount of second order smoothing relative to fourth order smoothing near shock waves. The logic used to compute $\varepsilon^{(4)}$ switches off the fourth order smoothing when the second order smoothing term is large.

The parameters $\kappa_2$ and $\kappa_4$ are user-specified constants. Like the coefficients in the constant coefficient model, the optimum values will be problem-dependent, and are best chosen through experience. Cases have been run with values of $\kappa_2$ ranging from from 0.01 for flows without shocks to 0.1 for flows with shocks, and $\kappa_4$ ranging from 0.0002 for flows computed with spatially constant second-order time differencing to

---

[6] It should be noted that the grid increments $\Delta\xi$, $\Delta\eta$, and $\Delta\zeta$ in these definitions do not appear in the corresponding formulas presented by Pulliam (1986b). This is because the grids used by Pulliam are constructed such that $\Delta\xi = \Delta\eta = \Delta\zeta = 1$, while in *Proteus* $\Delta\xi = 1/(N_1 - 1)$, $\Delta\eta = 1/(N_2 - 1)$, and $\Delta\zeta = 1/(N_3 - 1)$. The definitions used here for $\psi_x$, $\psi_y$, and $\psi_z$ result in an artificial viscosity level equivalent to that described by Pulliam.

0.005 for flows computed with spatially varying first-order time differencing. Pulliam (1986b) gives $\kappa_2 = 0.25$ and $\kappa_4 = 0.01$ as typical values for an Euler analysis.

Like the constant coefficient artificial viscosity model, the nonlinear coefficient model requires special formulas near boundaries. To apply (8.4) at $i = 2$, $\varepsilon_\xi^{(2)}$ is needed at $i = 1$. It is defined as

$$\left(\varepsilon_\xi^{(2)}\right)_1 = \kappa_2 \Delta\tau \max(\sigma_2,\ \sigma_1)$$

With the above definition, applying (8.4) at $i = 2$ and $i = N_1 - 1$ requires $\sigma$ at $i = 1$ and $i = N_1$. They are defined as

$$\sigma_1 = \left| \frac{-p_4 + 4p_3 - 5p_2 + 2p_1}{p_4 + 4p_3 + 5p_2 + 2p_1} \right|$$

$$\sigma_{N_1} = \left| \frac{-p_{N_1-3} + 4p_{N_1-2} - 5p_{N_1-1} + 2p_{N_1}}{p_{N_1-3} + 4p_{N_1-2} + 5p_{N_1-1} + 2p_{N_1}} \right|$$

And, finally, applying (8.4) at $i = 2$ and $i = N_1 - 1$ requires $\Delta_\xi \nabla_\xi \Delta_\xi \mathbf{Q}$ at $i = 1$ and $i = N_1 - 1$. There are numerous formulas that could be used. The ones currently in the *Proteus* code are

$$\Delta_\xi \nabla_\xi \Delta_\xi \mathbf{Q}_1 = -\mathbf{Q}_5 + 5\mathbf{Q}_4 - 9\mathbf{Q}_3 + 7\mathbf{Q}_2 - 2\mathbf{Q}_1$$

$$\Delta_\xi \nabla_\xi \Delta_\xi \mathbf{Q}_{N_1-1} = \mathbf{Q}_{N_1-4} - 5\mathbf{Q}_{N_1-3} + 9\mathbf{Q}_{N_1-2} - 7\mathbf{Q}_{N_1-1} + 2\mathbf{Q}_{N_1}$$

# 9.0 TURBULENCE MODEL

As noted briefly in Section 2.0, for turbulent flow the Reynolds stress and turbulent heat flux terms are modeled using the Boussinesq approach. An effective viscosity is thus defined as $\mu = \mu_l + \mu_t$, where $\mu_l$ is the laminar, or molecular, viscosity coefficient, and $\mu_t$ is the turbulent viscosity coefficient. Similarly, an effective second coefficient of viscosity is defined as $\lambda = \lambda_l + \lambda_t$, and an effective thermal conductivity coefficient is defined as $k = k_l + k_t$.

The turbulent coefficients must be computed using a turbulence model appropriate for the flow being computed. In *Proteus*, turbulence is modeled using either a generalized version of the Baldwin and Lomax (1978) algebraic eddy viscosity model, or the Chien (1982) low Reynolds number $k$-$\varepsilon$ model.

## 9.1 BALDWIN-LOMAX MODEL

For wall-bounded flows, (i.e., boundary layers), the Baldwin-Lomax turbulence model is a two-layer model, with

$$\mu_t = \begin{cases} (\mu_t)_{inner} & \text{for } y_n \le y_b \\ (\mu_t)_{outer} & \text{for } y_n > y_b \end{cases} \tag{9.1}$$

where $y_n$ is the normal distance from the wall, and $y_b$ is the smallest value of $y_n$ at which the values of $\mu_t$ from the inner and outer region formulas are equal. For free turbulent flows (i.e., mixing layers, jets, and wakes), $\mu_t = (\mu_t)_{outer}$. In the inner region, in addition to the Baldwin-Lomax model, an alternate expression first presented by Spalding (1961), and later by Kleinstein (1967), is also available.

### 9.1.1 Outer Region

The outer region turbulent viscosity is computed from

$$(\mu_t)_{outer} = KC_{cp}\rho F_{Kleb}F_{wake}Re_r \tag{9.2}$$

where $K$ is the Clauser constant, taken as 0.0168, $C_{cp}$ is a constant taken as 1.6, and $\rho$ is the static density.

The parameter $F_{wake}$ is computed from

$$F_{wake} = \begin{cases} y_{max}F_{max} & \text{for wall-bounded flows} \\ C_{wk}V_{diff}^2\dfrac{y_{max}}{F_{max}} & \text{for free turbulent flows} \end{cases} \tag{9.3}$$

where $C_{wk}$ is a constant taken as 0.25, and

$$V_{diff} = \left|\vec{V}\right|_{max} - \left|\vec{V}\right|_{min}$$

where $\vec{V}$ is the total velocity vector.

The parameter $F_{max}$ in equation (9.3) is the maximum value of

$$F(y_n) = \begin{cases} y_n \left|\vec{\Omega}\right| \left(1 - e^{-y^+/A^+}\right) & \text{for wall-bounded flows} \\ y_n \left|\vec{\Omega}\right| & \text{for free turbulent flows} \end{cases} \tag{9.4}$$

and $y_{max}$ is the value of $y_n$ corresponding to $F_{max}$.

In a simple boundary layer analysis, with only one solid surface, the procedure for computing $F(y_n)$ and $F_{max}$ is relatively straightforward. In a general Navier-Stokes analysis, in which any part of any boundary may be a solid surface, the problem is more complicated.

In *Proteus*, each grid point is labeled as either a wall-bounded point or a wake point. Wall-bounded points are those for which at least one of the three grid lines through the point intersects a solid wall. All other grid points are wake points.

For each wall-bounded point $P_{wall}$, the grid line intersecting the nearest wall is determined. $F(y_n)$ is then computed along that line, with $y_n$ equal to the distance to the wall nearest the point $P_{wall}$. $F_{max}$ is defined as the maximum value of $F$ along the line, and $y_{max}$ is the value of $y_n$ corresponding to $F_{max}$. It has been found that for wall-bounded flows the function $F(y_n)$ can have multiple peaks. In 3-D *Proteus*, the peak nearest the wall is used.

For each wake point $P_{wake}$, the grid line intersecting the nearest boundary is determined. Next, the values of $\left|\vec{V}\right|_{min}$ and $\left|\vec{V}\right|_{max}$ on that line are found. Two values of $F(y_n)$ are then computed - one with $y_n$ equal to the distance from the point $P_{wake}$ to the location of $\left|\vec{V}\right|_{min}$, and one with $y_n$ equal to the distance to the location of $\left|\vec{V}\right|_{max}$. Two values of $F_{max}$ and $y_{max}$ are determined, for the two $F(y_n)$ arrays. As in the wall-bounded case, $y_{max}$ is the value of $y_n$ corresponding to $F_{max}$. The smaller $y_{max}$ and the corresponding $F_{max}$ are the values finally used for computing $F_{wake}$.

In equation (9.4), $\left|\vec{\Omega}\right|$ is the magnitude of the total vorticity, defined as

$$\left|\vec{\Omega}\right| = \left[\left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}\right)^2 + \left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}\right)^2\right]^{1/2} \tag{9.5}$$

The parameter $A^+$ is the Van Driest damping constant, taken as 26.0. The coordinate $y^+$ is defined as

$$y^+ = \frac{\rho_w u_\tau y_n}{\mu_w} Re_r = \frac{\sqrt{\tau_w \rho_w Re_r}}{\mu_w} y_n \tag{9.6}$$

where $u_\tau = \sqrt{\tau_w / \rho_w Re_r}$ is the friction velocity, $\tau$ is the shear stress, and the subscript $w$ indicates a wall value. In *Proteus*, $\tau_w$ is set equal to $\mu_w \left|\vec{\Omega}\right|_w$.

The function $F_{Kleb}$ in equation (9.2) is the Klebanoff intermittency factor. For wake points, $F_{Kleb} = 1$. For wall-bounded points,

$$F_{Kleb} = (C_{Kleb})_{min} + [1 - (C_{Kleb})_{min}]\left[1 + B\left(\frac{C_{Kleb}\, y_n}{y_{max}}\right)^6\right]^{-1} \tag{9.7}$$

This factor accounts for the experimentally observed fact that, as the free stream is approached, the fraction of time the flow is turbulent decreases. In equation (9.7), $B$ and $C_{Kleb}$ are constants taken as 5.5 and 0.3, respectively. $(C_{Kleb})_{min}$ is a constant normally equal to 0.0. However, when using the Baldwin-Lomax model to generate initial turbulent viscosity values for the Chien $k$-$\varepsilon$ model (discussed in Section 9.2), $(C_{Kleb})_{min}$ is set equal to 0.1. This yields a small positive value for $\mu_t$ in the free stream, and has been found to minimize starting problems with the $k$-$\varepsilon$ model.

### 9.1.2 Inner Region

The inner region turbulent viscosity in the Baldwin-Lomax model is

$$(\mu_t)_{inner} = \rho l^2 \left|\vec{\Omega}\right| Re_r \tag{9.8}$$

where $l$ is the mixing length, normally given by

$$l = \kappa y_n \left(1 - e^{-y^+/A^+}\right) \tag{9.9}$$

and $\kappa$ is the Von Karman constant, taken as 0.4.

A modified form of equation (9.9), proposed by Launder and Priddin (1973), may also be used. This formula is most useful for flows with steep negative gradients of shear stress normal to the wall, such as accelerated flows or flows with suction. Their modified formula for $l$ is

$$l = \kappa y_n \left(1 - e^{-y^+(\tau^+)^n/A^+}\right) \tag{9.10}$$

where

$$\tau^+ = \frac{\tau}{\tau_w} = \frac{\mu\left|\vec{\Omega}\right|}{\mu_w\left|\vec{\Omega}\right|_w}$$

and $n$ is a constant taken as 1.7.

The inner region turbulent viscosity may also be computed using an alternate expression first presented by Spalding (1961), and later by Kleinstein (1967). In this model,

$$(\mu_t)_{inner} = \mu_l \kappa e^{-\kappa B}\left[e^{\kappa u^+} - 1 - \kappa u^+ - \frac{1}{2}(\kappa u^+)^2\right] \tag{9.11}$$

where

$$u^+ = \frac{\left|\vec{V}\right|}{u_\tau} = \frac{\left|\vec{V}\right|}{\sqrt{\tau_w/\rho_w Re_r}}$$

Again, in *Proteus*, $\tau_w$ is set equal to $\mu_w\left|\vec{\Omega}\right|_w$.

### 9.1.3 Turbulent Values of $\lambda$ and $k$

The turbulent second coefficient of viscosity is simply defined as

$$\lambda_t = -\frac{2}{3}\mu_t \tag{9.12}$$

The turbulent thermal conductivity coefficient is defined using Reynolds analogy as

$$k_t = \frac{c_p \mu_t}{Pr_t} Pr_r \tag{9.13}$$

where $c_p$ is the specific heat at constant pressure, and $Pr_t$ is the turbulent Prandtl number. In *Proteus*, the turbulent Prandtl number may be treated as constant, or as a variable using the following formula (Wassel and Catton, 1973):

$$Pr_t = \frac{C_{Pr3}}{C_{Pr1} Pr_l} \frac{1 - \exp\left( - \dfrac{C_{Pr4}}{\mu_t/\mu_l} \right)}{1 - \exp\left( - \dfrac{C_{Pr2}}{Pr_l \, \mu_t/\mu_l} \right)} \tag{9.14}$$

Here $C_{Pr1}$, $C_{Pr2}$, $C_{Pr3}$, and $C_{Pr4}$ are constants taken as 0.21, 5.25, 0.20, and 5.0, respectively, and $Pr_l = c_p \mu_l / k_l$ is the laminar Prandtl number.

## 9.2 CHIEN $k$-$\varepsilon$ TURBULENCE MODEL

### 9.2.1 $k$-$\varepsilon$ Equations

The low Reynolds number $k$-$\varepsilon$ formulation of K. Y. Chien (1982) was chosen because of its reasonable approximation of the near wall region and because of its numerical stability. Here $k$ and $\varepsilon$ are the turbulent kinetic energy and the turbulent dissipation rate, respectively.[7] In addition, the Chien $k$-$\varepsilon$ turbulence model was frequently used in past Navier-Stokes computations with good results (Nichols, 1990, 1991; Patel, Rodi, and Scheuerer, 1985; Sahu, 1984.) The set of $k$-$\varepsilon$ equations are lagged in time and solved separately from the Navier-Stokes equations to allow for code modularity in turbulence modeling. In Cartesian coordinates, the three-dimensional equations for the Chien $k$-$\varepsilon$ model can be written using vector notation as

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = \mathbf{S} + \mathbf{T} \tag{9.15}$$

where

$$\mathbf{W} = \begin{bmatrix} \rho k \\ \rho \varepsilon \end{bmatrix} \tag{9.16a}$$

$$\mathbf{F} = \begin{bmatrix} \rho u k - \dfrac{1}{Re_r} \mu_k \dfrac{\partial k}{\partial x} \\ \rho u \varepsilon - \dfrac{1}{Re_r} \mu_\varepsilon \dfrac{\partial \varepsilon}{\partial x} \end{bmatrix} \tag{9.16b}$$

$$\mathbf{G} = \begin{bmatrix} \rho v k - \dfrac{1}{Re_r} \mu_k \dfrac{\partial k}{\partial y} \\ \rho v \varepsilon - \dfrac{1}{Re_r} \mu_\varepsilon \dfrac{\partial \varepsilon}{\partial y} \end{bmatrix} \tag{9.16c}$$

$$\mathbf{H} = \begin{bmatrix} \rho w k - \dfrac{1}{Re_r} \mu_k \dfrac{\partial k}{\partial z} \\ \rho w \varepsilon - \dfrac{1}{Re_r} \mu_\varepsilon \dfrac{\partial \varepsilon}{\partial z} \end{bmatrix} \tag{9.16d}$$

$$\mathbf{S} = \begin{bmatrix} P_k - Re_r \rho \varepsilon \\ C_1 P_k \dfrac{\varepsilon}{k} - Re_r C_2 \rho \dfrac{\varepsilon^2}{k} \end{bmatrix} \tag{9.16e}$$

[7] It should be noted that in the Chien model, $\varepsilon$ is actually the isotropic portion of the turbulent dissipation rate. Throughout this manual, however, it is referred to as simply the turbulent dissipation rate.

$$\mathbf{T} = \begin{bmatrix} -\frac{2}{Re_r}\frac{\mu}{y_n^2}k \\ -\frac{2}{Re_r}\frac{\mu e^{-y^+/2}}{y_n^2}\varepsilon \end{bmatrix} \tag{9.16f}$$

and

$$\mu_k = \mu + \frac{\mu_t}{\sigma_k} \tag{9.17a}$$

$$\mu_\varepsilon = \mu + \frac{\mu_t}{\sigma_\varepsilon} \tag{9.17b}$$

$$C_1 = 1.35 \tag{9.17c}$$

$$C_2 = C_{2_r}\left(1 - \frac{2}{9}e^{-R_t^2/36}\right) \tag{9.17d}$$

$$\sigma_k = 1.0 \tag{9.17e}$$

$$\sigma_\varepsilon = 1.3 \tag{9.17f}$$

$$C_{2_r} = 1.8 \tag{9.17g}$$

$$R_t = \frac{\rho k^2}{\mu\varepsilon} \tag{9.17h}$$

$$P_k = \frac{\mu_t}{Re_r}P_1 - \frac{2}{3}\rho k P_2 \tag{9.18a}$$

$$P_1 = 2\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2\right] - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial u}{\partial z} + \frac{\partial v}{\partial x} + \frac{\partial v}{\partial z} + \frac{\partial w}{\partial x} + \frac{\partial w}{\partial y}\right)^2 \tag{9.18b}$$

$$P_2 = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \tag{9.18c}$$

The turbulent viscosity is given by

$$\mu_t = C_\mu \rho \frac{k^2}{\varepsilon} \tag{9.19a}$$

$$C_\mu = C_{\mu_r}\left(1 - e^{-C_3 y^+}\right) \tag{9.19b}$$

$$C_{\mu_r} = 0.09 \tag{9.19c}$$

$$C_3 = 0.0115 \tag{9.19d}$$

Note that the vectors **W**, **F**, **G**, **H**, and **S** are used in most standard $k$-$\varepsilon$ formulations (with different constants), and the vector **T** is unique to the low Reynolds number formulation of Chien. The parameter $y_n$ is the minimum distance to the nearest solid surface, and $y^+$ is computed from $y_n$. The production of turbulent kinetic energy $P_k$ includes the full Boussinesq approximation for compressible flows. All of the

above equations have been nondimensionalized using appropriate normalizing conditions. Nondimensionalization of mean flow properties is discussed in Section 2.1. The turbulent kinetic energy $k$ and the turbulent dissipation rate $\varepsilon$ have been nondimensionalized by $u_r^2$ and $\rho_r u_r^4/\mu_r$, respectively.

Following the procedure of Section 2.3, the following generalized grid transformation is used to transform the $k$-$\varepsilon$ equations from physical $(x, y, z, t)$ coordinates to computational $(\xi, \eta, \zeta, \tau)$ coordinates.

$$\begin{aligned} \xi &= \xi(x, y, z) \\ \eta &= \eta(x, y, z) \\ \zeta &= \zeta(x, y, z) \\ \tau &= t \end{aligned} \tag{9.20}$$

Applying the generalized grid transformation to equation (9.15) yields

$$\mathbf{W}_\tau + \mathbf{F}_\xi \xi_x + \mathbf{F}_\eta \eta_x + \mathbf{F}_\zeta \zeta_x + \mathbf{G}_\xi \xi_y + \mathbf{G}_\eta \eta_y + \mathbf{G}_\zeta \zeta_y + \mathbf{H}_\xi \xi_z + \mathbf{H}_\eta \eta_z + \mathbf{H}_\zeta \zeta_z = \mathbf{S} + \mathbf{T} \tag{9.21}$$

Although the above equations can not be put into exact strong conservation law form, the procedure used to do so for the mean flow equations, described in Section 2.4, is nonetheless applied to equation (9.21). The result is

$$\frac{\partial \hat{\mathbf{W}}}{\partial \tau} + \frac{\partial \hat{\mathbf{F}}}{\partial \xi} + \frac{\partial \hat{\mathbf{G}}}{\partial \eta} + \frac{\partial \hat{\mathbf{H}}}{\partial \zeta} = \hat{\mathbf{S}} + \hat{\mathbf{T}} \tag{9.22}$$

where

$$\hat{\mathbf{W}} = \frac{1}{J}\begin{bmatrix} \rho k \\ \rho \varepsilon \end{bmatrix} \tag{9.23a}$$

$$\hat{\mathbf{F}} = \hat{\mathbf{F}}_C - \hat{\mathbf{F}}_D - \hat{\mathbf{F}}_M \tag{9.23b}$$

$$\hat{\mathbf{F}}_C = \frac{1}{J}\begin{bmatrix} \xi_x \rho u k + \xi_y \rho v k + \xi_z \rho w k \\ \xi_x \rho u \varepsilon + \xi_y \rho v \varepsilon + \xi_z \rho w \varepsilon \end{bmatrix} \tag{9.23c}$$

$$\hat{\mathbf{F}}_D = \frac{1}{J}\frac{1}{Re_r}\begin{bmatrix} \mu_k(\xi_x^2 + \xi_y^2 + \xi_z^2)k_\xi \\ \mu_\varepsilon(\xi_x^2 + \xi_y^2 + \xi_z^2)\varepsilon_\xi \end{bmatrix} \tag{9.23d}$$

$$\hat{\mathbf{F}}_M = \frac{1}{J}\frac{1}{Re_r}\begin{bmatrix} \mu_k(\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z)k_\eta - \mu_k(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z)k_\zeta \\ \mu_\varepsilon(\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z)\varepsilon_\eta - \mu_\varepsilon(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z)\varepsilon_\zeta \end{bmatrix} \tag{9.23e}$$

$$\hat{\mathbf{G}} = \hat{\mathbf{G}}_C - \hat{\mathbf{G}}_D - \hat{\mathbf{G}}_M \tag{9.23f}$$

$$\hat{\mathbf{G}}_C = \frac{1}{J}\begin{bmatrix} \eta_x \rho u k + \eta_y \rho v k + \eta_z \rho w k \\ \eta_x \rho u \varepsilon + \eta_y \rho v \varepsilon + \eta_z \rho w \varepsilon \end{bmatrix} \tag{9.23g}$$

$$\hat{\mathbf{G}}_D = \frac{1}{J}\frac{1}{Re_r}\begin{bmatrix} \mu_k(\eta_x^2 + \eta_y^2 + \eta_z^2)k_\eta \\ \mu_\varepsilon(\eta_x^2 + \eta_y^2 + \eta_z^2)\varepsilon_\eta \end{bmatrix} \tag{9.23h}$$

$$\hat{\mathbf{G}}_M = \frac{1}{J}\frac{1}{Re_r}\begin{bmatrix} \mu_k(\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z)k_\xi - \mu_k(\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z)k_\zeta \\ \mu_\varepsilon(\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z)\varepsilon_\xi - \mu_\varepsilon(\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z)\varepsilon_\zeta \end{bmatrix} \tag{9.23i}$$

$$\hat{\mathbf{H}} = \hat{\mathbf{H}}_C - \hat{\mathbf{H}}_D - \hat{\mathbf{H}}_M \tag{9.23j}$$

$$\hat{\mathbf{H}}_C = \frac{1}{J}\begin{bmatrix} \zeta_x \rho u k + \zeta_y \rho v k + \zeta_z \rho w k \\ \zeta_x \rho u \varepsilon + \zeta_y \rho v \varepsilon + \zeta_z \rho w \varepsilon \end{bmatrix} \tag{9.23k}$$

$$\hat{\mathbf{H}}_D = \frac{1}{J}\frac{1}{Re_r}\begin{bmatrix} \mu_k(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)k_\zeta \\ \mu_\varepsilon(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)\varepsilon_\zeta \end{bmatrix} \tag{9.23l}$$

$$\hat{\mathbf{H}}_M = \frac{1}{J}\frac{1}{Re_r}\begin{bmatrix} \mu_k(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z)k_\xi - \mu_k(\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z)k_\eta \\ \mu_\varepsilon(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z)\varepsilon_\xi - \mu_\varepsilon(\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z)\varepsilon_\eta \end{bmatrix} \tag{9.23m}$$

$$\hat{\mathbf{S}} = \frac{1}{J}\begin{bmatrix} P_k - Re_r\rho\varepsilon \\ C_1 P_k \dfrac{\varepsilon}{k} - Re_r C_2 \rho \dfrac{\varepsilon^2}{k} \end{bmatrix} \tag{9.23n}$$

$$\hat{\mathbf{T}} = \frac{1}{J}\begin{bmatrix} -\dfrac{2}{Re_r}\dfrac{\mu}{y_n^2}k \\ -\dfrac{2}{Re_r}\dfrac{\mu e^{-y^+/2}}{y_n^2}\varepsilon \end{bmatrix} \tag{9.23o}$$

Note that in equation (9.23n), the term $P_k$ involves derivatives with respect to the Cartesian coordinate directions (see equations (9.18a-c.) These are evaluated using the chain rule.

### 9.2.2 Linearization of the $k$-$\varepsilon$ Equations

Solving equation (9.22) for $\partial\hat{\mathbf{W}}/\partial\tau$ and substituting the result into the time differencing scheme of Beam and Warming (1978), given by equation (3.1), for $\partial(\Delta\hat{\mathbf{W}}^n)/\partial\tau$ and $\partial\hat{\mathbf{W}}^n/\partial\tau$ yields

$$\begin{aligned}\Delta\hat{\mathbf{W}}^n = {} & \frac{\theta_1\Delta\tau}{1+\theta_2}\left(-\frac{\partial(\Delta\hat{\mathbf{F}}_C)}{\partial\xi} + \frac{\partial(\Delta\hat{\mathbf{F}}_D)}{\partial\xi} + \frac{\partial(\Delta\hat{\mathbf{F}}_M)}{\partial\xi} - \frac{\partial(\Delta\hat{\mathbf{G}}_C)}{\partial\eta} + \frac{\partial(\Delta\hat{\mathbf{G}}_D)}{\partial\eta} + \frac{\partial(\Delta\hat{\mathbf{F}}_M)}{\partial\eta} - \frac{\partial(\Delta\hat{\mathbf{H}}_C)}{\partial\zeta} + \frac{\partial(\Delta\hat{\mathbf{H}}_D)}{\partial\zeta} + \frac{\partial(\Delta\hat{\mathbf{H}}_M)}{\partial\zeta} + \Delta\hat{\mathbf{S}} + \Delta\hat{\mathbf{T}}\right)^n \\ & + \frac{\Delta\tau}{1+\theta_2}\left(-\frac{\partial\hat{\mathbf{F}}_C}{\partial\xi} + \frac{\partial\hat{\mathbf{F}}_D}{\partial\xi} + \frac{\partial\hat{\mathbf{F}}_M}{\partial\xi} - \frac{\partial\hat{\mathbf{G}}_C}{\partial\eta} + \frac{\partial\hat{\mathbf{G}}_D}{\partial\eta} + \frac{\partial\hat{\mathbf{G}}_M}{\partial\eta} - \frac{\partial\hat{\mathbf{H}}_C}{\partial\zeta} + \frac{\partial\hat{\mathbf{H}}_D}{\partial\zeta} + \frac{\partial\hat{\mathbf{H}}_M}{\partial\zeta} + \hat{\mathbf{S}} + \hat{\mathbf{T}}\right)^n \\ & + \frac{\theta_2}{1+\theta_2}\Delta\hat{\mathbf{W}}^{n-1} + O\left[\left(\theta_1 - \tfrac{1}{2} - \theta_2\right)(\Delta\tau)^2, (\Delta\tau)^3\right]\end{aligned} \tag{9.24}$$

Equation (9.24) is then linearized using the procedure described in Section 4.0. Let

$$A = \frac{\partial\hat{\mathbf{F}}_C}{\partial\hat{\mathbf{W}}},\quad B = \frac{\partial\hat{\mathbf{F}}_D}{\partial\hat{\mathbf{W}}},\quad C = \frac{\partial\hat{\mathbf{G}}_C}{\partial\hat{\mathbf{W}}},\quad D = \frac{\partial\hat{\mathbf{G}}_D}{\partial\hat{\mathbf{W}}},\quad E = \frac{\partial\hat{\mathbf{H}}_C}{\partial\hat{\mathbf{W}}},\quad F = \frac{\partial\hat{\mathbf{H}}_D}{\partial\hat{\mathbf{W}}},\quad M = \frac{\partial\hat{\mathbf{S}}}{\partial\hat{\mathbf{W}}},\quad N = \frac{\partial\hat{\mathbf{T}}}{\partial\hat{\mathbf{W}}} \tag{9.25}$$

be the Jacobian coefficient matrices, where

$$A = \begin{bmatrix} \xi_x u + \xi_y v + \xi_z w & 0 \\ 0 & \xi_x u + \xi_y v + \xi_z w \end{bmatrix} \tag{9.26}$$

$$B = \begin{bmatrix} \dfrac{1}{JRe_r}\mu_k(\xi_x^2 + \xi_y^2 + \xi_z^2)\left(\dfrac{J}{\rho}\right)_\xi & 0 \\ 0 & \dfrac{1}{JRe_r}\mu_\varepsilon(\xi_x^2 + \xi_y^2 + \xi_z^2)\left(\dfrac{J}{\rho}\right)_\xi \end{bmatrix} \tag{9.27}$$

$$C=\begin{bmatrix}\eta_x u+\eta_y v+\eta_z w & 0\\ 0 & \eta_x u+\eta_y v+\eta_z w\end{bmatrix} \tag{9.28}$$

$$D=\begin{bmatrix}\frac{1}{JRe_r}\mu_k(\eta_x^2+\eta_y^2+\eta_z^2)\left(\frac{J}{\rho}\right)_\eta & 0\\ 0 & \frac{1}{JRe_r}\mu_\varepsilon(\eta_x^2+\eta_y^2+\eta_z^2)\left(\frac{J}{\rho}\right)_\eta\end{bmatrix} \tag{9.29}$$

$$E=\begin{bmatrix}\zeta_x u+\zeta_y v+\zeta_z w & 0\\ 0 & \zeta_x u+\zeta_y v+\zeta_z w\end{bmatrix} \tag{9.30}$$

$$F=\begin{bmatrix}\frac{1}{JRe_r}\mu_k(\zeta_x^2+\zeta_y^2+\zeta_z^2)\left(\frac{J}{\rho}\right)_\zeta & 0\\ 0 & \frac{1}{JRe_r}\mu_\varepsilon(\zeta_x^2+\zeta_y^2+\zeta_z^2)\left(\frac{J}{\rho}\right)_\zeta\end{bmatrix} \tag{9.31}$$

$$M=\begin{bmatrix}2C_\mu\frac{k}{\varepsilon}\frac{P_k}{\mu_t} & -C_\mu\frac{k^2}{\varepsilon^2}\frac{P_k}{\mu_t}-Re_r\\ C_1C_\mu\frac{P_k}{\mu_t}+Re_rC_2\frac{\varepsilon^2}{k^2} & -2Re_rC_2\frac{\varepsilon}{k}\end{bmatrix} \tag{9.32}$$

$$N=\begin{bmatrix}-\frac{2\mu}{\rho y_n^2 Re_r} & 0\\ 0 & -\frac{2\mu}{\rho y_n^2 Re_r}e^{-y^+/2}\end{bmatrix} \tag{9.33}$$

The linearized form of equation (9.24) can now be written as

$$\begin{aligned}\Delta\hat{\mathbf{W}}^n+\frac{\theta_1\Delta\tau}{1+\theta_2}\left[\frac{\partial(A\Delta\hat{\mathbf{W}})}{\partial\xi}-\frac{\partial(B\Delta\hat{\mathbf{W}})}{\partial\xi}+\frac{\partial(C\Delta\hat{\mathbf{W}})}{\partial\eta}-\frac{\partial(D\Delta\hat{\mathbf{W}})}{\partial\eta}+\frac{\partial(E\Delta\hat{\mathbf{W}})}{\partial\zeta}-\frac{\partial(F\Delta\hat{\mathbf{W}})}{\partial\zeta}-M(\Delta\hat{\mathbf{W}})-N(\Delta\hat{\mathbf{W}})\right]^n=\\ \frac{\theta_1\Delta\tau}{1+\theta_2}\left[\frac{\partial(\Delta\hat{\mathbf{F}}_M)}{\partial\xi}+\frac{\partial(\Delta\hat{\mathbf{G}}_M)}{\partial\eta}+\frac{\partial(\Delta\hat{\mathbf{H}}_M)}{\partial\zeta}\right]^{n-1}\\ +\frac{\Delta\tau}{1+\theta_2}\left(-\frac{\partial\hat{\mathbf{F}}_C}{\partial\xi}+\frac{\partial\hat{\mathbf{F}}_D}{\partial\xi}+\frac{\partial\hat{\mathbf{F}}_M}{\partial\xi}-\frac{\partial\hat{\mathbf{G}}_C}{\partial\eta}+\frac{\partial\hat{\mathbf{G}}_D}{\partial\eta}+\frac{\partial\hat{\mathbf{G}}_M}{\partial\eta}-\frac{\partial\hat{\mathbf{H}}_C}{\partial\zeta}+\frac{\partial\hat{\mathbf{H}}_D}{\partial\zeta}+\frac{\partial\hat{\mathbf{H}}_M}{\partial\zeta}+\hat{\mathbf{S}}+\hat{\mathbf{T}}\right)^n+\frac{\theta_2}{1+\theta_2}\Delta\hat{\mathbf{W}}^{n-1}\\ +O\left[\left(\theta_1-\tfrac{1}{2}-\theta_2\right)(\Delta\tau)^2,(\Delta\tau)^3\right]\end{aligned} \tag{9.34}$$

### 9.2.3 LU Factorization Algorithm for the $k$-$\varepsilon$ Equations

The LU factorization scheme used to solve the $k$-$\varepsilon$ equations is essentially the same as that described by Hoffmann (1989). Letting RHS(9.34) represent the right hand side of equation (9.34), we can write

$$\left\{\mathbf{I}+\frac{\theta_1\Delta\tau}{1+\theta_2}\left[\frac{\partial}{\partial\xi}(A-B)+\frac{\partial}{\partial\eta}(C-D)+\frac{\partial}{\partial\zeta}(E-F)-(M+N)\right]\right\}^n\Delta\hat{\mathbf{W}}^n=\text{RHS(9.34)}\quad(9.35)$$

where $\mathbf{I}$ represents the identity matrix.

The Jacobian matrices $A$, $C$, and $E$ can each be split into two submatrices, such that each submatrix is associated with the positive and negative eigenvalues of the corresponding Jacobian matrix. Equation (9.35) can thus be rewritten as

$$\left\{\mathbf{I}+\frac{\theta_1\Delta\tau}{1+\theta_2}\left[\frac{\partial}{\partial\xi}(A^+ + A^-)+\frac{\partial}{\partial\eta}(C^+ + C^-)+\frac{\partial}{\partial\zeta}(E^+ + E^-)-\frac{\partial B}{\partial\xi}-\frac{\partial D}{\partial\eta}-\frac{\partial F}{\partial\zeta}-(M+N)\right]\right\}^n\Delta\hat{\mathbf{W}}^n=\text{RHS(9.34)}\quad(9.36)$$

Using first-order upwind differencing for the Jacobian matrices $A$, $C$, and $E$, and central differencing for $B$, $D$, $F$, and on the right hand side, equation (9.36) becomes

$$\begin{aligned}&\left\{\mathbf{I}+\frac{\theta_1\Delta\tau}{1+\theta_2}\left[\delta_\xi^- A^+ + \delta_\xi^+ A^- + \delta_\eta^- C^+ + \delta_\eta^+ C^- + \delta_\zeta^- E^+ + \delta_\zeta^+ E^-\right]\right.\\&\left.-\frac{\theta_1\Delta\tau}{1+\theta_2}\left[(\delta_\xi B)^+ + (\delta_\xi B)^- + (\delta_\eta D)^+ + (\delta_\eta D)^- + (\delta_\zeta F)^+ + (\delta_\zeta F)^- + (M+N)\right]\right\}^n\Delta\hat{\mathbf{W}}^n=\frac{\theta_1\Delta\tau}{1+\theta_2}\left[\delta_\xi(\Delta\hat{\mathbf{F}}_M)+\delta_\eta(\Delta\hat{\mathbf{G}}_M)+\delta_\zeta(\Delta\hat{\mathbf{H}}_M)\right]^{n-1}\\&+\frac{\Delta\tau}{1+\theta_2}\left(-\delta_\xi\hat{\mathbf{F}}_C+\delta_\xi\hat{\mathbf{F}}_D+\delta_\xi\hat{\mathbf{F}}_M-\delta_\eta\hat{\mathbf{G}}_C+\delta_\eta\hat{\mathbf{G}}_D+\delta_\eta\hat{\mathbf{G}}_M-\delta_\zeta\hat{\mathbf{H}}_C+\delta_\zeta\hat{\mathbf{H}}_D+\delta_\zeta\hat{\mathbf{H}}_M+\hat{\mathbf{S}}+\hat{\mathbf{T}}\right)^n\qquad(9.37)\\&+\frac{\theta_2}{1+\theta_2}\Delta\hat{\mathbf{W}}^{n-1}+O\left[\left(\theta_1-\tfrac{1}{2}-\theta_2\right)(\Delta\tau)^2,(\Delta\tau)^3\right]\end{aligned}$$

Note that the central differencing operators for $B$, $D$, and $F$ have been split into forward and backward differencing parts. Neglecting the temporal truncation and splitting errors, equation (9.37) can be approximately factored as

$$\begin{aligned}&\left\{\mathbf{I}+\frac{\theta_1\Delta\tau}{1+\theta_2}\left[\delta_\xi^- A^+ + \delta_\eta^- C^+ + \delta_\zeta^- E^+ - (\delta_\xi B)^- - (\delta_\eta D)^- - (\delta_\zeta F)^-\right]\right\}^n\cdot\\&\left\{\mathbf{I}+\frac{\theta_1\Delta\tau}{1+\theta_2}\left[\delta_\xi^+ A^- + \delta_\eta^+ C^- + \delta_\zeta^+ E^- - (\delta_\xi B)^+ - (\delta_\eta D)^+ - (\delta_\zeta F)^+ - (M+N)\right]\right\}^n\Delta\hat{\mathbf{W}}^n=\text{RHS(9.34)}\frac{\theta_1\Delta\tau}{1+\theta_2}\left[\delta_\xi(\Delta\hat{\mathbf{F}}_M)+\delta_\eta(\Delta\hat{\mathbf{G}}_M)+\delta_\zeta(\Delta\hat{\mathbf{H}}_M)\right]^{n-1}\\&+\frac{\Delta\tau}{1+\theta_2}\left(-\delta_\xi\hat{\mathbf{F}}_C+\delta_\xi\hat{\mathbf{F}}_D+\delta_\xi\hat{\mathbf{F}}_M-\delta_\eta\hat{\mathbf{G}}_C+\delta_\eta\hat{\mathbf{G}}_D+\delta_\eta\hat{\mathbf{G}}_M-\delta_\zeta\hat{\mathbf{H}}_C+\delta_\zeta\hat{\mathbf{H}}_D+\delta_\zeta\hat{\mathbf{H}}_M+\hat{\mathbf{S}}+\hat{\mathbf{T}}\right)^n\qquad(9.38)\\&+\frac{\theta_2}{1+\theta_2}\Delta\hat{\mathbf{W}}^{n-1}\end{aligned}$$

This equation can then be split into the following two-sweep sequence.

Sweep 1 (upward)

$$\left\{\mathbf{I}+\frac{\theta_1\Delta\tau}{1+\theta_2}\left[\delta_\xi^- A^+ + \delta_\eta^- C^+ + \delta_\zeta^- E^+ - (\delta_\xi B)^- - (\delta_\eta D)^- - (\delta_\zeta F)^-\right]\right\}^n\Delta\hat{\mathbf{W}}^*=\text{RHS(9.34)}\quad(9.39a)$$

Sweep 2 (downward)

$$\left\{\mathbf{I}+\frac{\theta_1\Delta\tau}{1+\theta_2}\left[\delta_\xi^+ A^- + \delta_\eta^+ C^- + \delta_\zeta^+ E^- - (\delta_\xi B)^+ - (\delta_\eta D)^+ - (\delta_\zeta F)^+\right]\right\}^n\Delta\hat{\mathbf{W}}^n=\Delta\hat{\mathbf{W}}^*\qquad(9.39b)$$

### 9.2.4 LU Sweeping Procedure for the $k$-$\varepsilon$ Equations

*Non-Periodic Boundary Conditions*

In the solution algorithm, the upward sweep is done first, then the downward sweep. When applied at an interior point in the computational domain, equations (9.39a) and (9.39b) each have just one unknown. These equations are therefore solved point-by-point.

The upward sweep starts at the lower left front corner of the computational domain, point (2, 2, 2), and marches along planes of constant $i+j+k$ to the upper right back corner, point $(N_1-1,\ N_2-1,\ N_3-1)$. Equation (9.39a) is solved for the intermediate unknown $\Delta\hat{\mathbf{W}}^*$ at each point $(i,j,k)$ using known information at points $(i-1,j,k)$, $(i,j-1,k)$, and $(i,j,k-1)$. This is possible because the left hand side of the equation contains only backward differencing operators.

The downward sweep is in the opposite direction, from point $(N_1-1,\ N_2-1,\ N_3-1)$ to point (2, 2, 2), again along planes of constant $i+j+k$. Equation (9.39b) is solved for the final unknown $\Delta\hat{\mathbf{W}}^*$ at each point $(i,j,k)$ using known information at points $(i+1,j,k)$, $(i,j+1,k)$, and $(i,j,k+1)$. This is possible because the left hand side of the equation contains only forward differencing operators.

*Spatially Periodic Boundary Conditions*

A spatially periodic boundary condition in the $\xi$ direction may be represented as shown in Figure 7.1. Following Section 7.2.2, an additional set of grid points is added at $i=N_1+1$, setting $\hat{\mathbf{W}}_{N_1+1}=\hat{\mathbf{W}}_2$. This allows us to use central differencing in the $\xi$ direction at $i=N_1$. The upward sweep therefore goes from point (2, 2, 2) to point $(N_1,\ N_2-1,\ N_3-1)$, and the downward sweep goes from point $(N_1,\ N_2-1,\ N_3-1)$ to point (2, 2, 2). An analogous procedure is used for the periodic boundary conditions in the $\eta$ and/or $\zeta$ directions.

### 9.2.5 Updating Boundary Values for $k$-$\varepsilon$ Equations

For easy modification and easy accommodation of complicated boundary conditions for $k$ and $\varepsilon$, non-periodic boundary conditions are treated explicitly in the solver. After the $k$ and $\varepsilon$ values at the interior points are advanced in time, the values at the boundaries are simply computed from the new interior values using the specified boundary conditions.

Spatially periodic boundary conditions in any sweep direction are treated implicitly, as described in the previous section. For a periodic boundary condition in the $\xi$ direction, the $k$ and $\varepsilon$ values at $i=1$ are easily updated by setting $\hat{\mathbf{W}}_1=\hat{\mathbf{W}}_{N_1}$. An analogous procedure is used for periodic boundary conditions in the $\eta$ and $\zeta$ directions.

### 9.2.6 Turbulent Values of $\lambda$ and $k$

The turbulent second coefficient of viscosity $\lambda_t$ and the turbulent thermal conductivity coefficient $k_t$ are defined as described previously in Section 9.1.3.

## APPENDIX A - EXPANSION OF VISCOUS TERMS

In Section 4.2, the viscous terms in the governing equations are linearized. To do this, the elements of $\hat{\mathbf{E}}_V$, $\hat{\mathbf{F}}_V$, and $\hat{\mathbf{G}}_V$, given in equations (2.17e) through (2.17g) must first be rewritten in terms of the dependent variables, and with derivatives in the Cartesian directions transformed to derivatives in the computational directions using the chain rule. The non-cross derivative terms, involving $\hat{\mathbf{E}}_{V_1}$, $\hat{\mathbf{F}}_{V_1}$, and $\hat{\mathbf{G}}_{V_1}$, are then linearized using Taylor series expansion. The cross derivative terms, involving $\hat{\mathbf{E}}_{V_2}$, $\hat{\mathbf{F}}_{V_2}$, and $\hat{\mathbf{G}}_{V_2}$, are simply lagged one time step. This Appendix presents the fully expanded viscous terms required in the linearization procedure.

The viscous term $\hat{\mathbf{E}}_V$ is given by equation (2.17e), which is repeated here.

$$\hat{\mathbf{E}}_V = \frac{1}{J}\frac{1}{Re_r}\begin{bmatrix} 0 \\ \tau_{xx}\xi_x + \tau_{xy}\xi_y + \tau_{xz}\xi_z \\ \tau_{xy}\xi_x + \tau_{yy}\xi_y + \tau_{yz}\xi_z \\ \tau_{xz}\xi_x + \tau_{yz}\xi_y + \tau_{zz}\xi_z \\ \beta_x\xi_x + \beta_y\xi_y + \beta_z\xi_z \end{bmatrix} \tag{A.1}$$

where

$$\tau_{xx} = 2\mu u_x + \lambda(u_x + v_y + w_z)$$

$$\tau_{yy} = 2\mu v_y + \lambda(u_x + v_y + w_z)$$

$$\tau_{zz} = 2\mu w_z + \lambda(u_x + v_y + w_z)$$

$$\tau_{xy} = \mu(u_y + v_x)$$

$$\tau_{xz} = \mu(u_z + w_x)$$

$$\tau_{yz} = \mu(v_z + w_y)$$

$$\beta_x = u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - \frac{1}{Pr_r}q_x$$

$$\beta_y = u\tau_{xy} + v\tau_{yy} + w\tau_{yz} - \frac{1}{Pr_r}q_y$$

$$\beta_z = u\tau_{xz} + v\tau_{yz} + w\tau_{zz} - \frac{1}{Pr_r}q_z$$

$$q_x = -kT_x$$

$$q_y = -kT_y$$

$$q_z = -kT_z$$

The chain rule is used to transform derivatives in the Cartesian directions into derivatives in the computational directions, resulting in

$$\tau_{xx} = (2\mu + \lambda)(\xi_x u_\xi + \eta_x u_\eta + \zeta_x u_\zeta) + \lambda(\xi_y v_\xi + \eta_y v_\eta + \zeta_y v_\zeta) + \lambda(\xi_z w_\xi + \eta_z w_\eta + \zeta_z w_\zeta)$$

$$\tau_{yy} = (2\mu + \lambda)(\xi_y v_\xi + \eta_y v_\eta + \zeta_y v_\zeta) + \lambda(\xi_x u_\xi + \eta_x u_\eta + \zeta_x u_\zeta) + \lambda(\xi_z w_\xi + \eta_z w_\eta + \zeta_z w_\zeta)$$

$$\tau_{zz} = (2\mu + \lambda)(\xi_z w_\xi + \eta_z w_\eta + \zeta_z w_\zeta) + \lambda(\xi_x u_\xi + \eta_x u_\eta + \zeta_x u_\zeta) + \lambda(\xi_y v_\xi + \eta_y v_\eta + \zeta_y v_\zeta)$$

$$\tau_{xy} = \mu(\xi_y u_\xi + \eta_y u_\eta + \zeta_y u_\zeta + \xi_x v_\xi + \eta_x v_\eta + \zeta_x v_\zeta)$$

$$\tau_{xz} = \mu(\xi_z u_\xi + \eta_z u_\eta + \zeta_z u_\zeta + \xi_x w_\xi + \eta_x w_\eta + \zeta_x w_\zeta)$$

$$\tau_{yz} = \mu(\xi_z v_\xi + \eta_z v_\eta + \zeta_z v_\zeta + \xi_y w_\xi + \eta_y w_\eta + \zeta_y w_\zeta)$$

$$\begin{aligned}\beta_x = {} & (2\mu + \lambda)(\xi_x uu_\xi + \eta_x uu_\eta + \zeta_x uu_\zeta) \\ & + \lambda(\xi_y uv_\xi + \eta_y uv_\eta + \zeta_y uv_\zeta + \xi_z uw_\xi + \eta_z uw_\eta + \zeta_z uw_\zeta) \\ & + \mu(\xi_y vu_\xi + \eta_y vu_\eta + \zeta_y vu_\zeta + \xi_x vv_\xi + \eta_x vv_\eta + \zeta_x vv_\zeta) \\ & + \mu(\xi_z wu_\xi + \eta_z wu_\eta + \zeta_z wu_\zeta + \xi_x ww_\xi + \eta_x ww_\eta + \zeta_x ww_\zeta) \\ & + k(\xi_x T_\xi + \eta_x T_\eta + \zeta_x T_\zeta)\end{aligned}$$

$$\begin{aligned}\beta_y = {} & (2\mu + \lambda)(\xi_y vv_\xi + \eta_y vv_\eta + \zeta_y vv_\zeta) \\ & + \lambda(\xi_x vu_\xi + \eta_x vu_\eta + \zeta_x vu_\zeta + \xi_z vw_\xi + \eta_z vw_\eta + \zeta_z vw_\zeta) \\ & + \mu(\xi_y uu_\xi + \eta_y uu_\eta + \zeta_y uu_\zeta + \xi_x uv_\xi + \eta_x uv_\eta + \zeta_x uv_\zeta) \\ & + \mu(\xi_z wv_\xi + \eta_z wv_\eta + \zeta_z wv_\zeta + \xi_y ww_\xi + \eta_y ww_\eta + \zeta_y ww_\zeta) \\ & + k(\xi_y T_\xi + \eta_y T_\eta + \zeta_y T_\zeta)\end{aligned}$$

$$\begin{aligned}\beta_z = {} & (2\mu + \lambda)(\xi_z ww_\xi + \eta_z ww_\eta + \zeta_z ww_\zeta) \\ & + \lambda(\xi_x wu_\xi + \eta_x wu_\eta + \zeta_x wu_\zeta + \xi_y wv_\xi + \eta_y wv_\eta + \zeta_y wv_\zeta) \\ & + \mu(\xi_z uu_\xi + \eta_z uu_\eta + \zeta_z uu_\zeta + \xi_x uw_\xi + \eta_x uw_\eta + \zeta_x uw_\zeta) \\ & + \mu(\xi_z vv_\xi + \eta_z vv_\eta + \zeta_z vv_\zeta + \xi_y vw_\xi + \eta_y vw_\eta + \zeta_y vw_\zeta) \\ & + k(\xi_z T_\xi + \eta_z T_\eta + \zeta_z T_\zeta)\end{aligned}$$

The above expressions for the $\tau$'s and $\beta$'s are next substituted into equation (A.1). The $\xi$ derivative terms become elements of $\hat{\mathbf{E}}_{V_1}$, and the $\eta$ and $\zeta$ derivative terms become elements of $\hat{\mathbf{E}}_{V_2}$. The resulting five elements of $\hat{\mathbf{E}}_{V_1}$ (excluding the $1/JRe_r$ coefficient) are

$$(\hat{\mathbf{E}}_{V_1})_1 = 0 \tag{A.2a}$$

$$(\hat{\mathbf{E}}_{V_1})_2 = 2\mu\xi_x^2 u_\xi + \lambda\xi_x(\xi_x u_\xi + \xi_y v_\xi + \xi_z w_\xi) + \mu\xi_y(\xi_y u_\xi + \xi_x v_\xi) + \mu\xi_z(\xi_z u_\xi + \xi_x w_\xi) \tag{A.2b}$$

$$(\hat{\mathbf{E}}_{V_1})_3 = 2\mu\xi_y^2 v_\xi + \lambda\xi_y(\xi_x u_\xi + \xi_y v_\xi + \xi_z w_\xi) + \mu\xi_x(\xi_y u_\xi + \xi_x v_\xi) + \mu\xi_z(\xi_z v_\xi + \xi_y w_\xi) \tag{A.2c}$$

$$(\hat{\mathbf{E}}_{V_1})_4 = 2\mu\xi_z^2 w_\xi + \lambda\xi_z(\xi_x u_\xi + \xi_y v_\xi + \xi_z w_\xi) + \mu\xi_x(\xi_z u_\xi + \xi_x w_\xi) + \mu\xi_y(\xi_z v_\xi + \xi_y w_\xi) \tag{A.2d}$$

$$\begin{aligned}(\hat{\mathbf{E}}_{V_1})_5 = {} & 2\mu(\xi_x^2 uu_\xi + \xi_y^2 vv_\xi + \xi_z^2 ww_\xi) + \lambda\xi_x(\xi_x uu_\xi + \xi_y uv_\xi + \xi_z uw_\xi) \\ & + \lambda\xi_y(\xi_x vu_\xi + \xi_y vv_\xi + \xi_z vw_\xi) + \lambda\xi_z(\xi_x wu_\xi + \xi_y wv_\xi + \xi_z ww_\xi) \\ & + \mu\xi_x(\xi_y vu_\xi + \xi_x vv_\xi + \xi_z wu_\xi + \xi_x ww_\xi) + \mu\xi_y(\xi_y uu_\xi + \xi_x uv_\xi + \xi_z vv_\xi + \xi_y ww_\xi) \\ & + \mu\xi_z(\xi_z uu_\xi + \xi_x uw_\xi + \xi_z vv_\xi + \xi_y vw_\xi) + k(\xi_x^2 + \xi_y^2 + \xi_z^2)T_\xi\end{aligned} \tag{A.2e}$$

For linearization it is convenient to rewrite the last element as

$$(\hat{\mathbf{E}}_{V_1})_5 = \frac{(2\mu+\lambda)}{2}[\xi_x^2(u^2)_\xi + \xi_y^2(v^2)_\xi + \xi_z^2(w^2)_\xi] + (\mu+\lambda)[\xi_x\xi_y(uv)_\xi + \xi_x\xi_z(uw)_\xi + \xi_y\xi_z(vw)_\xi]$$
$$+ \frac{\mu}{2}[\xi_x^2(v^2+w^2)_\xi + \xi_y^2(u^2+w^2)_\xi + \xi_z^2(u^2+v^2)_\xi] + k(\xi_x^2+\xi_y^2+\xi_z^2)T_\xi \qquad \text{(A.2f)}$$

The elements of $\hat{\mathbf{F}}_{V_1}$ and $\hat{\mathbf{G}}_{V_1}$ have exactly the same form as those of $\hat{\mathbf{E}}_{V_1}$, but with $\xi$ replaced by $\eta$ and $\zeta$, respectively.

The five elements of $\hat{\mathbf{E}}_{V_2}$ (again excluding the $1/JRe_r$ coefficient) are

$$(\hat{\mathbf{E}}_{V_2})_1 = 0 \qquad \text{(A.3a)}$$

$$(\hat{\mathbf{E}}_{V_2})_2 = 2\mu\xi_x(\eta_x u_\eta + \zeta_x u_\zeta) + \lambda\xi_x(\eta_x u_\eta + \eta_y v_\eta + \eta_z w_\eta + \zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta)$$
$$+ \mu\xi_y(\eta_y u_\eta + \eta_x v_\eta + \zeta_y u_\zeta + \zeta_x v_\zeta) + \mu\xi_z(\eta_z u_\eta + \eta_x w_\eta + \zeta_z u_\zeta + \zeta_x w_\zeta) \qquad \text{(A.3b)}$$

$$(\hat{\mathbf{E}}_{V_2})_3 = 2\mu\xi_y(\eta_y v_\eta + \zeta_y v_\zeta) + \lambda\xi_y(\eta_x u_\eta + \eta_y v_\eta + \eta_z w_\eta + \zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta)$$
$$+ \mu\xi_x(\eta_y u_\eta + \eta_x v_\eta + \zeta_y u_\zeta + \zeta_x v_\zeta) + \mu\xi_z(\eta_z v_\eta + \eta_y w_\eta + \zeta_z v_\zeta + \zeta_y w_\zeta) \qquad \text{(A.3c)}$$

$$(\hat{\mathbf{E}}_{V_2})_4 = 2\mu\xi_z(\eta_z w_\eta + \zeta_z w_\zeta) + \lambda\xi_z(\eta_x u_\eta + \eta_y v_\eta + \eta_z w_\eta + \zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta)$$
$$+ \mu\xi_x(\eta_z u_\eta + \eta_x w_\eta + \zeta_z u_\zeta + \zeta_x w_\zeta) + \mu\xi_y(\eta_z v_\eta + \eta_y w_\eta + \zeta_z v_\zeta + \zeta_y w_\zeta) \qquad \text{(A.3d)}$$

$$\begin{aligned}(\hat{\mathbf{E}}_{V_2})_5 = {} & 2\mu[\xi_x(\eta_x uu_\eta + \zeta_x uu_\zeta) + \xi_y(\eta_y vv_\eta + \zeta_y vv_\zeta) + \xi_z(\eta_z ww_\eta + \zeta_z ww_\zeta)] \\ & + \lambda\xi_x(\eta_x uu_\eta + \eta_y uv_\eta + \eta_z uw_\eta + \zeta_x uu_\zeta + \zeta_y uv_\zeta + \zeta_z uw_\zeta) \\ & + \lambda\xi_y(\eta_x vu_\eta + \eta_y vv_\eta + \eta_z vw_\eta + \zeta_x vu_\zeta + \zeta_y vv_\zeta + \zeta_z vw_\zeta) \\ & + \lambda\xi_z(\eta_x wu_\eta + \eta_y wv_\eta + \eta_z ww_\eta + \zeta_x wu_\zeta + \zeta_y wv_\zeta + \zeta_z ww_\zeta) \\ & + \mu\xi_x(\eta_y vu_\eta + \eta_x vv_\eta + \eta_z wu_\eta + \eta_x ww_\eta + \zeta_y vu_\zeta + \zeta_x vv_\zeta + \zeta_z wu_\zeta + \zeta_x ww_\zeta) \\ & + \mu\xi_y(\eta_y uu_\eta + \eta_x uv_\eta + \eta_z wv_\eta + \eta_y ww_\eta + \zeta_y uu_\zeta + \zeta_x uv_\zeta + \zeta_z wv_\zeta + \zeta_y ww_\zeta) \\ & + \mu\xi_z(\eta_z uu_\eta + \eta_x uw_\eta + \eta_z vv_\eta + \eta_y vw_\eta + \zeta_z uu_\zeta + \zeta_x uw_\zeta + \zeta_z vv_\zeta + \zeta_y vw_\zeta) \\ & + k(\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z)T_\eta + k(\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z)T_\zeta \end{aligned} \qquad \text{(A.3e)}$$

The elements of $\hat{\mathbf{F}}_{V_2}$ have exactly the same form as those of $\hat{\mathbf{E}}_{V_2}$, but with $\xi$ replaced by $\eta$ and $\eta$ replaced by $\xi$. Similarly, the elements of $\hat{\mathbf{G}}_{V_2}$ have exactly the same form as those of $\hat{\mathbf{E}}_{V_2}$, but with $\xi$ replaced by $\zeta$ and $\zeta$ replaced by $\xi$.

PAGE _____ INTENTIONALLY BLANK

## REFERENCES


Anderson, D. A., Tannehill, J. C., and Pletcher, R. H. (1984) *Computational Fluid Mechanics and Heat Transfer,* Hemisphere Publishing Corporation, McGraw-Hill Book Company, New York.

Baldwin, B. S., and Lomax, H. (1978) "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257.

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

Briley, W. R., and McDonald, H. (1977) "Solution of the Multidimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method," Journal of Computational Physics, Vol. 24, pp. 373-397.

Briley, W. R., and McDonald, H. (1980) "On the Structure and Use of Linearized Block Implicit Schemes," Journal of Computational Physics, Vol. 34, No. 1, pp. 54-73.

Cebeci, T., and Smith, A. M. O. (1974) *Analysis of Turbulent Boundary Layers,* Academic Press, New York.

Chen, S. C., and Schwab, J. R. (1988) "Three-Dimensional Elliptic Grid Generation Technique with Application to Turbomachinery Cascades," NASA TM 101330.

Chien, K. Y. (1982) "Prediction of Channel and Boundary-Layer Flows with a Low-Reynolds-Number Turbulence Model," AIAA Journal, Vol. 20, No. 1, pp. 33-38.

Douglas, J., and Gunn, J. E. (1964) "A General Formulation of Alternating Direction Methods. Part I - Parabolic and Hyperbolic Problems," Numerische Mathematik, Vol. 6, pp. 428-453.

Hoffmann, K. A. (1989) *Computational Fluid Dynamics for Engineers,* Engineering Educational System, Austin, Texas.

Hughes, W. F., and Gaylord, E. W. (1964) *Basic Equations of Engineering Science,* Schaum's Outline Series, McGraw-Hill Book Company, New York.

Jameson, A., Schmidt, W., and Turkel, E. (1981) "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259.

Kernighan, B. W., and Plauger, P. J. (1978) *The Elements of Programming Style,* McGraw-Hill Book Company, New York.

Kleinstein, G. (1967) "Generalized Law of the Wall and Eddy-Viscosity Model for Wall Boundary Layers," AIAA Journal, Vol. 5, No. 8, pp. 1402-1407.

Korn, G. A., and Korn, T. M. (1968) *Mathematical Handbook for Scientists and Engineers,* McGraw-Hill Book Company, New York.

Launder, B. E., and Priddin, C. H. (1973) "A Comparison of Some Proposals for the Mixing Length Near a Wall," International Journal of Heat and Mass Transfer, Vol. 16, pp. 700-702.

Nichols, R. H. (1990) "Two-Equation Model for Compressible Flows," AIAA Paper 90-0494.

Nichols, R. H. (1991) "Calculation of the Flow in a Circular S-Duct Inlet," AIAA Paper 91-0174.

Pulliam, T. H. (1986a) "Efficient Solution Methods for the Navier-Stokes Equations," *Numerical Techniques for Viscous Flow Calculations in Turbomachinery Bladings,* Lecture Series 1986-02, Von Karman Institute for Fluid Dynamics, Brussels, Belgium.

Patel, V. C., Rodi W., and Scheuerer, G. (1985) "Turbulent Models for Near-Wall and Low Reynolds Number Flows: A Review," AIAA Journal, Vol. 23, No. 9, pp. 1308-1319.

Pulliam, T. H. (1986a) "Efficient Solution Methods for the Navier-Stokes Equations," *Numerical Techniques for Viscous Flow Calculations in Turbomachinery Bladings,* Lecture Series 1986-02, Von Karman Institute for Fluid Dynamics, Brussels, Belgium.

Pulliam, T. H. (1986b) "Artificial Dissipation Models for the Euler Equations," AIAA Journal, Vol. 24, No. 12, pp. 1931-1940.

Pulliam, T. H., and Steger, J. L. (1978) "On Implicit Finite-Difference Simulations of Three Dimensional Flow," AIAA Paper 78-10.

Sahu, J. (1984) "Navier-Stokes Computational Study of Axisymmetric Transonic Turbulent Flows with a Two-Equation Model of Turbulence," Ph. D. Thesis, University of Delaware.

Schlichting, H. (1968) *Boundary-Layer Theory,* McGraw-Hill Book Company, New York.

Spalding, D. B. (1961) "A Single Formula for the Law of the Wall," Journal of Applied Mechanics, Vol. 28, pp. 455-457.

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

Thomas, P. D., and Lombard, C. K. (1979) "Geometric Conservation Law and Its Application to Flow Computations on Moving Grids," AIAA Journal, Vol. 17, No. 10, pp. 1030-1037.

Towne, C. E., Schwab, J. R., Benson, T. J., and Suresh, A. (1990) "PROTEUS Two-Dimensional Navier-Stokes Computer Code - Version 1.0, Volumes 1-3," NASA TM's 102551-3.

Vinokur, M. (1974) "Conservation Equations of Gasdynamics in Curvilinear Coordinate Systems," Journal f Computational Physics, Vol. 14, pp. 105-125.

Wassel, A. T., and Catton, I. (1973) "Calculation of Turbulent Boundary Layers Over Flat Plates With Different Phenomenological Theories of Turbulence and Variable Turbulent Prandtl Number," International Journal of Heat and Mass Transfer, Vol. 16, pp. 1547-1563.

White, F. M. (1974) *Viscous Fluid Flow,* McGraw-Hill Book Company, New York.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. AGENCY USE ONLY (*Leave blank*) | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | October 1993 | Technical Memorandum |

**4. TITLE AND SUBTITLE**

Proteus Three-Dimensional Navier-Stokes Computer Code–Version 1.0
Volume 1–Analysis Description

**5. FUNDING NUMBERS**

WU–505–62–52

**6. AUTHOR(S)**

Charles E. Towne, John R. Schwab, and Trong T. Bui

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

**8. PERFORMING ORGANIZATION REPORT NUMBER**

E–8106

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

National Aeronautics and Space Administration
Washington, D.C. 20546–0001

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

NASA TM–106337

**11. SUPPLEMENTARY NOTES**

Responsible person, Charles E. Towne, (216) 433–5851.

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**

Unclassified - Unlimited
Subject Category 34

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** (*Maximum 200 words*)

A computer code called *Proteus 3D* has been developed to solve the three-dimensional, Reynolds-averaged, unsteady compressible Navier-Stokes equations in strong conservation law form. The objective in this effort has been to develop a code for aerospace propulsion applications that is easy to use and easy to modify. Code readability, modularity, and documentation have been emphasized. The governing equations are solved in generalized nonorthogonal body-fitted coordinates, by marching in time using a fully-coupled ADI solution procedure. The boundary conditions are treated implicitly. All terms, including the diffusion terms, are linearized using second-order Taylor series expansions. Turbulence is modeled using either an algebraic or two-equation eddy viscosity model. The thin-layer or Euler equations may also be solved. The energy equation may be eliminated by the assumption of constant total enthalpy. Explicit and implicit artificial viscosity may be used. Several time step options are available for convergence acceleration. The documentation is divided into three volumes. This is the Analysis Description, and presents the equations and solution procedure. It describes in detail the governing equations, the turbulence model, the linearization of the equations and boundary conditions, the time and space differencing formulas, the ADI solution procedure, and the artificial viscosity models.

| 14. SUBJECT TERMS | 15. NUMBER OF PAGES |
|---|---|
| Navier-Stokes; Computational fluid dynamics; Viscous flow; Compressible flow | 72 |
| | 16. PRICE CODE A04 |

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | |