NASA-CR-194795

Implementation of the Baldwin-Barth Turbulence Model into the ZETA Code and its Diagnosis

A Thesis

Presented to the Faculty of

California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree of

Master of Science in Aeronautical Engineering

by

Scott L. Low

June 1993

Implementation of the Baldwin-Barth Turbulence Model into the ZETA Code and its Diagnosis

A Thesis

Presented to the Faculty of

California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree of

Master of Science in Aeronautical Engineering

by

Scott L. Low

June 1993

APPROVAL PAGE

TITLE : Implementation of Baldwin-Barth Turbulence Model into a Time-Accurate Code for Unsteady Flows

AUTHOR : Scott L. Low

DATE SUBMITTED : June 16, 1993

Dr. Jin Tso
Thesis Advisor

Dr. Russell M. Cummings
Committee Member

Prof. Jon A Hoffmann
Committee Member

Dr. Chee Tung
Committee Member

Dr. Russell M. Cummings
Department Head

PAGE ii INTENTIONALLY BLANK

ABSTRACT

IMPLEMENTATION OF BALDWIN-BARTH TURBULENCE MODEL
INTO A TIME-ACCURATE CODE FOR UNSTEADY FLOWS

Scott L. Low

June 1993

The Baldwin-Barth turbulence model was implemented into Zeta, a time-accurate, zonal, integro-differential code for incompressible laminar and turbulent flows. The implementation procedure patterned after the model subroutine in ARC2D. The results of ZETA with the Baldwin-Barth turbulence model were compared with experimental data, with ZETA using Baldwin-Lomax model, and with ARC2D using the Baldwin-Barth model. The Baldwin-Barth model subroutine was tested by inputting an ARC2D velocity solution of an NACA-0012 airfoil at $R_e = 3.91 \times 10^6$ and $\alpha = 5^o$. The resultant turbulent viscosity and Reynolds stresses compared favorably with the original data. For the same grid having grid points inside the laminar sublayer, which is necessary due to the one-equation nature of the model, ZETA however predicts early separation. It was found that the current ZETA has problem with such a fine grid. Further work is in progress to solve this problem.

ACKNOWLEDGEMENTS

I am grateful to my advisor, Dr. Jin Tso, and Dr. Chee Tung of NASA Ames Research Center for their guidance and support throughout this project. I would also like to thank Dr. Clin Wang of George Institute of Technology and Dr. Timothy Barth of NASA Ames Research Center for their assistance and helpful discussions on the numerical aspects of the project. I also acknowledge the Army's Fluid Mechanics Division (Aeroflightdynamics Directorate) at NASA Ames Research Center for the use of their facilities.

This project was funded by NASA Grant number NCA2-722.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

A	Cross-sectional area of an airfoil.
A^+	Constant equal to 26 in the turbulence models.
$A_{i,j}$	Lower elements of the tridiagonal matrix of the vorticity transport equation.
An	Components making up $A_{i,j}$ where n is an index.
a_n	Velocity Fourier coefficients where n is an index.
$B_{i,j}$	Diagonal elements of the tridiagonal matrix of the vorticity transport equation.
Bn	Components making up $B_{i,j}$ where n is an index.
B_s	Internal boundary.
B_∞	External boundary.
b_n	Velocity Fourier coefficients where n is an index.
$C_{i,j}$	Upper elements of the tridiagonal matrix of the vorticity transport equation.
Cn	Components making up $C_{i,j}$ where n is an index.
C_p	Pressure coefficient.
c,γ	Airfoil transformation parameters.
c_n	Velocity Fourier coefficients where n is an index.
c_1, c_2	Grid stretching parameters.
$D_{i,j}$	RHS matrix elements of the discretized vorticity transport equation.
Dn	Components making up $D_{i,j}$ where n is an index.
D_1,D_2	Damping functions in the Baldwin-Barth model.
d_n	Velocity Fourier coefficients where n is an index.
E	Extent of the grid.
$F(y)$	A function in the Baldwin-Lomax model.
F_{kleb}	Klebanoff intermittence function.
f_k	A Damping function in the Baldwin-Barth model where k is an index or variable.

H	Transformation factor.
JR	Radial index of the grid.
L, AL	Airfoil chordlength.
$M(\vec{R})$	A M-type matrix operator in the discretized Baldwin-Barth model equation.
P	Production of κ in the Baldwin-Barth turbulence model.
R	Flow region or radius of a unit circle.
R_e	Reynolds number.
$R_{i,j}$	Solution vector of the Baldwin-Barth subroutine.
R_T	Turbulent Reynolds number.
$\vec{r}$	Position vector with respect to axis of rotation.
r,θ	Physical radial and angular coordinates.
$S\omega$	Source term of the vorticity transport equation.
s	Distance from the unit circle in the computational grid.
t	Time in second.
U_∞	Freestream velocity.
u^+	Scaled velocity.
u',v'	Cartesian velocity components in the physical plane and inertial system.
u,v	Cartesian velocity components in the physical plane and rotating system.
$\mathbf{u}',\mathbf{v}'$	Cartesian velocity components in the computational plane and inertial system.
$\mathbf{u},\mathbf{v}$	Cartesian velocity components in the computational plane and rotating system.
u_τ	Friction velocity.
$\vec{v}$	Velocity vector in the physical plane and inertial coordinate system.
$\vec{v}$	Velocity vector in the physical plane and rotating coordinate system.
v'_ρ, v'_ϕ	Cylindrical velocity components in the computational plane and inertial system.
v_ρ, v_ϕ	Cylindrical velocity components in the computational plane and rotating system.
x,y	Cartesian coordinates of the grid.
y^+	Length scale.

Z Physical plane: $Z = x + iy$, $Z = re^{i\theta}$.

α Airfoil angle of attack.

α_n Vorticity Fourier coefficients where n is an index.

α_a^x, α_a^y The first coefficient s of the advective terms in the Baldwin-Barth model.

α_d^x, α_d^y The first coefficient s of the diffusive terms in the Baldwin-Barth model.

β_n Vorticity Fourier coefficients where n is an index.

β_a^x, β_a^y The second coefficient s of the advective terms in the Baldwin-Barth model.

β_d^x, β_d^y The second coefficient s of the diffusive terms in the Baldwin-Barth model.

Δm Increment of m where m is a dummy variable.

δ Boundary layer thickness.

ε Dissipation terms in the turbulence models.

γ_a^x, γ_a^y The third coefficient s of the advective terms in the Baldwin-Barth model.

γ_d^x, γ_d^y The third coefficient s of the diffusive terms in the Baldwin-Barth model.

κ Von Karman constant or Production terms in the turbulence models.

ν Kinematic viscosity.

ν_t, ν_e Turbulent and effective viscosities.

ν_{ti} Inner turbulent viscosity in the Baldwin-Lomax model.

ν_{to} Outer turbulent viscosity in the Baldwin-Lomax model.

ρ, ϕ Computational radial and angular coordinates.

σ Physical plane origin shift in the transformation.

ς Computational plane: $\varsigma = \xi + i\eta$, $\varsigma = \rho e^{i\phi}$.

Ω Body or airfoil angular speed, positive counter-clockwise.

$\omega, \vec{\omega}$ Scalar and vector vorticities.

ξ, η General computational coordinates.

ψ' Stream function in either plane and the inertial coordinate system.

ψ Stream function in either plane and the rotating coordinate system.

CHAPTER 1

INTRODUCTION

1.1 Motivation

For years,unsteady flow has been the subject of numerous and continuing studies, in the field of helicopter or rotorcraft aerodynamics. The performance of a helicopter such as the UH-60A Black Hawks (Fig. 1.1) depends greatly on the aerodynamic lift of the rotor. Since the angle of attack of any rotor blade section oscillates as it travels through the rotor plane (Fig. 1.2), the resulting flow over the blade has complex and periodically changing characteristics. Understanding and accurately predicting the unsteady aerodynamics of flows over airfoils are, therefore, critical for designing new rotor blade airfoils and improving helicopter performance.

1.2 Previous Work

Much experimental work has been done and many numerical flow solvers have been developed to study unsteady flows. Experimental data from unsteady flow investigations on geometries such as rotor system [1] and advanced airfoil sections [2-4] are readily available for review and comparison. Reference [2] contains the two dimensional dynamic stall characteristics of eight arifoils in sinusoidal pitch oscillations over a wide range of unsteady flow conditions. Numerical codes ranging from panel methods, coupling between time-dependent inviscid panel method and an unsteady boundary layer code [5], a full-potential code [6], an Euler code [7], a zonal integro-differential method [8] and other finite difference codes based on the Navier-Stokes equations, have produced results of reasonable agreement with experimental results. The code in Ref. [6] predicted comparable surface pressures for helicopter rotors and the Euler code of Reference [7] was used to investigate the rotor blade-vortex interactions.

One recent study using the zonal integro-differential method for a two-element airfoil [9] and other single airfoils has demonstrated that this procedure is effective in treating general viscous flows, even with large separation regions. The code, ZETA [10] utilizes an integral representation of the velocity vector [11], a velocity-vorticity formulation of the Navier-Stokes Equations, and a Fourier series expansion. Using an integral representation of velocity, the flow computation may be confined to viscous zones. For turbulent flows, the Baldwin-Lomax [12] algebraic model is used to determine the eddy viscosity. A typical grid has 80x50 grid points, which is coarse compared to prevailing grid requirements for Navier-Stokes solutions. The general performance of this numerical procedure is satisfactory. However, further evaluations and refinements are required prior to using it for designing high-lift rotor blade airfoils.

1.3 Present Works

The Baldwin-Barth turbulence model [13] is a self-consistent one-equation model that does not require an algebraic length scale. It is derived from a simplified form of the standard κ-ε model equations [13]. This robust model was found to give significantly better results than the algebraic Baldwin-Lomax model in a recent study of four popular turbulence models [14]. This and other good results obtained with this model prompted the present investigation. The objectives are to implement the Baldwin-Barth turbulence model, and examine the effect of finer grids and Reynolds number(R_e) on ZETA. No known time-accurate case has be run with the Baldwin-Barth model. The computational results will be compared with that of ARC2D and with the experimental results of McAlister, et al [2].

This report presents the numerical procedures used and the results. Chapter 2 and 3 describe the governing equations, grids, and boundary conditions. The code's numerical formulation is explained in Chapter 4, and the two turbulence models are presented in Chapter 5. Numerical results for the NACA-0012 airfoil and comparisons are presented in Chapter 6. Chapter 7 contains major conclusions and recommendations for future study.

CHAPTER 2

GOVERNING EQUATIONS

2.1 Equations of Motion

The behavior of a viscous, incompressible, turbulent flow is described by the continuity equation and the vorticity transport equation in the code. The vorticity transport equation is derived by taking the curl of each term of the incompressible Navier-Stokes equations in its familiar pressure-velocity form. Since the flow is incompressible, no energy equation is required. With body forces such as gravity and heat transfer are negligible, these equations can be expressed as

$$\nabla \cdot \vec{v} = 0 \tag{2.1}$$

$$\frac{\partial \vec{\omega}}{\partial t} = -(\vec{v} \cdot \nabla)\vec{\omega} + (\vec{\omega} \cdot \nabla)\vec{v} + \nabla^2(\nu_e \vec{\omega}) + S_\omega \tag{2.2}$$

where $\vec{v}$ denotes the velocity and the vorticity vectors:

$$\vec{\omega} = \nabla \times \vec{v} \tag{2.3}$$

The effective viscosity ν_e is composed of both eddy viscosity ν_t and kinematic viscosity ν:

$$\nu_e = \nu_t + \nu$$

The physical processes of convection, stretching and rotation, and diffusion of vorticity are represented by the right hand side terms of Equation 2.2 respectively. For a two-dimensional flow, the scalar source term S_ω reduces to

$$S_\omega = 2\left[\frac{\partial^2}{\partial x^2}\left(\nu_e \frac{\partial u}{\partial y}\right) - \frac{\partial^2}{\partial y^2}\left(\nu_e \frac{\partial v}{\partial x}\right) + \frac{\partial^2}{\partial x \partial y}\left(\nu_e \frac{\partial v}{\partial y}\right) - \frac{\partial^2}{\partial x \partial y}\left(\nu_e \frac{\partial u}{\partial x}\right)\right]$$

The S_ω term is negligible relative to the other terms of the equation.

These equations are appropriate for both external and internal flow, but external flow will be emphasized. Equation 2.1 to Equation 2.3 are also three-dimensional but their

two-dimensional forms will be applied in ZETA. The corresponding velocity field is computed from the vorticity field with Equations 2.1 and 2.2. Along with initial and boundary velocity conditions, these equations will uniquely determine the time-dependent flow field of an incompressible fluid.

2.1.1 Kinematics and Kinetics of the Flow

The equations of motions are divided into a kinematic aspect and a kinetic aspect. The kinematic aspect of the problem relates the velocity field to the vorticity field at any instant of time. For a given vorticity field, the velocity field can be uniquely determined [15]. This aspect of the flow is governed by Equations 2.1 and 2.3 which are linear and elliptic. The solution of these equations will require prescribed boundary conditions about the flow field. Since the flow field is known only at infinity, the entire flow field must be included in the solution procedure.

With the vorticity-velocity formulation, Equations 2.1 and 2.3 may be reformulated as an integral representation for velocity vector at time t [11]

$$\vec{v}(\vec{r},t) = -\int_R \vec{\omega}_o \times \nabla_o P dR_o + \oint_B \left[(\vec{v}_o \cdot \vec{n}_o) - (\vec{v}_o \times \vec{n}_o)\times\right] \nabla_o P dB_o \tag{2.4}$$

where B includes the internal boundary B_S and external boundary B_∞ of the region R. $\vec{n}$ is the unit normal on B facing away from the region R. The subscript 'o' denotes that the operators and variables are in the viscous region, and P is the fundamental solution of Poisson's equation.

$$P(\vec{r},\vec{r}_o) = \frac{1}{2\pi} \ln \frac{1}{|\vec{r}-\vec{r}_o|}$$

With an integral representation, the integral over the fluid domain, R, does not need to include the inviscid region since $\vec{\omega}_o$ is zero there. The solution of velocity is then confined to the viscous region.

The kinetic aspect of the problem deals with the change of the vorticity field with

time and it is described by Equation 2.2. This equation is nonlinear, parabolic in time, and elliptic in space. The equation describes the transport, not the generation or depletion, of vorticity. Again, the solution of vorticity may be confined to the viscous region since $\vec{\omega}_o$ is zero in the inviscid region of the flow. This is the distinct feature of the numerical method. Taking advantage of this feature, the flow field is divided into three zones: an inviscid zone constituting the majority of the field, an attached viscous zone, and a detached viscous zone (Fig. 2.1). The detached viscous zone may include the wake, starting vortex assembly and separated regions of the airfoil.

2.2 Coordinate Transformation

The grid generation procedure employs a modified Joukowski transformation. The geometry and governing equations in Cartesian coordinates(x,y) are transformed to a generalized, body-conforming, curvilinear coordinate system(ρ,ϕ) [16]. The grid points have a one-to-one correspondence with the physical points. Unlike a conventional grid generator, this procedure works backward in that a specified computational grid is first constructed and then comformally mapped into the physical plane. The computational grid is composed of concentric circles and radial lines about a unit circle which represents the airfoil (Fig. 2.2). The grid transformation used is

$$Ze^{i\alpha} = \zeta + \gamma + \frac{c^2}{\zeta + \gamma} + \sigma \tag{2.5}$$

where

Z = physical plane

$= x + iy$

$= re^{i\theta}$

ς = computational plane

$= \xi + i\eta$

$= \rho e^{i\phi}$

α = airfoil angle of attack

c,γ = airfoil parameters

σ = physical plane origin shift

r,θ = physical radial and angular coordinates

ρ,ϕ = computational radial and angular coordinates

The airfoil parameters are

$$c = (\xi + \sqrt{1-\eta^2})(1-\delta) \tag{2.6}$$

$$\gamma = \xi + i\eta \tag{2.7}$$

The parameter c is the numerical chord length of the airfoil. γ is a complex translation of the origin to the center of the unit circle in the computational plane. δ is a real number close to zero which prevents the transformation from becoming singular. It also specifies the curvature of the trailing edge. ξ will always be negative and η will be zero for symmetric airfoils [15].

Based on the above transformation, the metrics and scale factor, H, of the transformations are defined as

$$\frac{\partial x}{\partial \xi} = \frac{\partial y}{\partial \eta} \tag{2.8}$$

$$\frac{\partial x}{\partial \eta} = -\frac{\partial y}{\partial \xi} \tag{2.9}$$

$$H = \left|\frac{\partial Z}{\partial \zeta}\right| \tag{2.10}$$

$$\begin{aligned} H^2 &= \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial x}{\partial \eta}\right)^2 \\ &= \left(\frac{\partial y}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2 \end{aligned} \tag{2.11}$$

The invariant of the transformation is the stream function. The integral representation for

the velocity vector is transformed into the computational plane by multiplying the vorticity vector by the square of the scale factor H^2,

$$\vec{\omega}_\zeta = H^2\vec{\omega} \tag{2.12}$$

The vorticity transport equation is transformed by differentiating the physical coordinates with respect to the computational coordinates. The vorticity is computed in the rotating coordinate system so the grid parameters need not be recomputed as the airfoil angle of attack changes. The velocity is computed in a body-fixed inertial reference frame. When needed, the velocity values are transformed to the rotating reference frame.

2.3 Transformed Equations

The governing equations for the kinematic and kinetic aspects of the flow are presented in both the inertial and rotating coordinate systems. These equations are summarized in their differential forms. The two aspects of the flow problem are computed in different reference frames. Both coordinate systems are considered body-fixed. The transformed equations are presented in their respective frames of reference. In this section, a primed variable is associated with the inertial coordinate system while a non-primed variable is associated with the rotating coordinate system. The following list of variables are used:

- $\vec{v}'$ Velocity vector in the physical plane and inertial coordinate system.
- $\vec{v}$ Velocity vector in the physical plane and rotating coordinate system.
- u',v' Cartesian velocity components in the physical plane and inertial system.
- u,v Cartesian velocity components in the physical plane and rotating system.
- $\mathbf{u}',\mathbf{v}'$ Cartesian velocity components in the computational plane and inertial system.
- $\mathbf{u},\mathbf{v}$ Cartesian velocity components in the computational plane and rotating system.
- v_ρ',v_ϕ' Cylindrical velocity components in the computational plane and inertial system.
- v_ρ,v_ϕ Cylindrical velocity components in the computational plane and rotating

system.

ψ' Stream function in either plane and the inertial coordinate system.

ψ Stream function in either plane and the rotating coordinate system.

Ω Body or airfoil angular speed, positive counter-clockwise.

$\vec{r}$ Position vector with respect to axis of rotation.

The velocities defined in terms of the the stream functions are

$$\begin{aligned} u' &= \frac{\partial \psi'}{\partial y} & v' &= -\frac{\partial \psi'}{\partial x} \\ u &= \frac{\partial \psi}{\partial y} & v &= -\frac{\partial \psi}{\partial x} \\ u' &= \frac{\partial \psi'}{\partial \eta} & v' &= -\frac{\partial \psi'}{\partial \xi} \\ u &= \frac{\partial \psi}{\partial \eta} & v &= -\frac{\partial \psi}{\partial \xi} \end{aligned} \tag{2.13}$$

The governing equations of the code are presented in the inertial reference frame below.

Kinematics:

$$\nabla' \cdot \vec{v}' = 0 \tag{2.14}$$

$$\nabla' \times \vec{v}' = \vec{\omega}' \tag{2.15}$$

$$\vec{v}_B = \vec{\Omega} \times \vec{r}_B \tag{2.16}$$

$\vec{v}_B$ is the velocity at any station on the body or airfoil surface.

Kinetics:

$$\frac{\partial \vec{\omega}'}{\partial t} = -\left(\vec{v}' \cdot \nabla'\right)\vec{\omega}' + \nabla'^2\left(\nu_e \vec{\omega}'\right) \tag{2.17}$$

Transforming to the computational plane, the kinematic governing equations become

$$\nabla'_\zeta \cdot \vec{v}'_\zeta = 0 \tag{2.18}$$

$$\nabla'_\zeta \times \vec{v}'_\zeta = \vec{\omega}'_\zeta \tag{2.19}$$

The two dimensional vorticity in the computational plane is

$$\vec{\omega}_\zeta = \frac{\partial v'}{\partial \xi} - \frac{\partial u'}{\partial \eta}$$

Numerically, the kinematic aspect of the flow will be solved using the integral representation of the velocity vector and a Fourier series expansion. The governing equations of the code are now presented in the rotating coordinate system.

Kinematics:

$$\nabla \cdot \vec{v} = 0 \tag{2.20}$$

$$\nabla \times \vec{v} = \vec{\omega} \tag{2.21}$$

$$\vec{v} = \vec{v}' - \vec{\Omega} \times \vec{r} \tag{2.22}$$

$$\vec{\omega} = \vec{\omega}' - 2\vec{\Omega} \tag{2.23}$$

$$\vec{v}_B = 0 \tag{2.24}$$

Kinetics:

$$\frac{\partial \vec{\omega}}{\partial t} = -(\vec{v} \cdot \nabla)\vec{\omega} + \nabla^2(\nu_e \vec{\omega}) - 2\dot{\vec{\Omega}} \tag{2.25}$$

The transformation of the governing equation of the kinetic aspect of the flow will be presented in more details. With the above equation and Equation 2.2, the vorticity can be referenced to the inertial system. The vorticity vector has been replaced by its scalar value.

$$\frac{\partial \omega'}{\partial t} = -(\vec{v} \cdot \nabla)\omega' + \nabla^2(\nu_e \omega') \tag{2.26}$$

The vorticity is in the physical plane and the velocities are still referenced to the rotating coordinate system. The divergence of velocity and Laplace operators are transformed as described in reference [10]:

$$\vec{v} \cdot \nabla = \frac{1}{H^2}(\vec{v}_\zeta \cdot \nabla_\zeta) \tag{2.27}$$

$$\nabla^2 = \frac{1}{H^2}\nabla_\zeta^2 \tag{2.28}$$

The resulting vorticity transport equation in the computational plane and in its conservative

form is

$$H^2\frac{\partial\omega'}{\partial t} + \nabla_\zeta \cdot \left(\vec{v}_\zeta \omega'\right) = \nabla_\zeta^2\left(\nu_e \omega'\right) \tag{2.29}$$

To obtain the numerical form of this equation, it is first non-dimensionalized.

$$x^* = \frac{x}{R} \qquad y^* = \frac{y}{R}$$
$$u^* = \frac{u}{U_\infty} \qquad v^* = \frac{v}{U_\infty}$$
$$\omega^* = \frac{\omega' R}{U_\infty} \qquad t^* = \frac{tU_\infty}{R}$$

$R = 1$ is the radius of the unit circle in the computational plane which represents the airfoil surface. $U_\infty = 1$ is the velocity at infinity. The non-dimensionalized vorticity transport equation is

$$H^2\frac{\partial\omega^*}{\partial t^*} + \nabla^* \cdot \left(\vec{v}^* \omega^*\right) = \nabla^{*2}\left(\frac{\nu_e \omega^*}{U_\infty R}\right) \tag{2.30}$$

With L defined as the airfoil chord length, the Reynolds number R_e is

$$R_e = \frac{U_\infty L}{\nu} \tag{2.31}$$

The numerical kinematic viscosity ν is therefore defined as L/R_e since $U_\infty = 1$. Dropping the superscripts and writing in cylindrical coordinates, the vorticity transport equation to be discretized becomes

$$\rho H^2\frac{\partial\omega}{\partial t} + \frac{1}{(\rho - c_2)\partial s}\frac{\partial}{}\left(\rho v_\rho \omega\right) + \frac{\partial}{\partial\phi}\left(v_\phi \omega\right) =$$
$$\frac{1}{(\rho - c_2)\partial s}\frac{\partial}{}\left(\frac{\rho}{(\rho - c_2)}\frac{\partial\omega_\nu}{\partial s}\right) + \frac{1}{\rho}\frac{\partial^2\omega_\nu}{\partial\phi^2} \tag{2.32}$$

where

$$\omega_\nu = \left(\frac{L}{R_e} + \nu_t\right)\omega$$

$$\frac{\partial}{\partial\rho} = \frac{\partial s}{\partial\rho}\frac{\partial}{\partial s}$$

$$\rho = e^{(s+c_1)+c_2}$$

$$\frac{\partial \rho}{\partial s} = \rho - c_2$$

The variable s, c_1, and c_2 will discussed in details in the next chapter.

Finally, the kinetic part of the problem requires velocity values in the rotating reference frame. Since the velocity is solved in the inertial reference frame in the kinematic part of the problem, the velocity correlations in the computational plane are given here. Having obtained the cylindrical velocity components, v'_ρ and v'_ϕ, in the inertial coordinate system, the Cartesian velocity components, $\mathbf{u}'$ and $\mathbf{v}'$, are computed as

$$\mathbf{u}' = v'_\rho \cos\phi - v'_\phi \sin\phi \tag{2.33}$$

$$\mathbf{v}' = v'_\rho \sin\phi + v'_\phi \cos\phi \tag{2.34}$$

The Cartesian velocity components are then transformed to the rotating coordinate system.

$$\mathbf{u} = \mathbf{u}' + \frac{\partial x}{\partial \eta}\Omega_x + \frac{\partial y}{\partial \eta}\Omega_y \tag{2.35}$$

$$\mathbf{v} = \mathbf{v}' - \frac{\partial x}{\partial \xi}\Omega_x - \frac{\partial y}{\partial \xi}\Omega_y \tag{2.36}$$

Transforming back to the cylindrical coordinates, the velocity components in the rotating coordinate system become

$$v_\rho = \mathbf{u}\cos\phi + \mathbf{v}\sin\phi \tag{2.37}$$

$$v_\phi = \mathbf{v}\cos\phi - \mathbf{u}\sin\phi \tag{2.38}$$

2.4 Aerodynamic Loads

Use of the vorticity-velocity form of the Navier-Stokes equation has the disadvantage of not having the pressure distribution computed from the equations of motion. Thus, special attention is required . The total head, h, of an incompressible flow can be derived from the pressure-velocity form of the Navier-Stokes equations [10]. An integral representation of the total head, h, for a two-dimensional external flow is

$$h(\vec{r},t) = \frac{-1}{2\pi}\int_R \frac{(\vec{v}_o\times\vec{\omega}_o)\cdot(\vec{r}_o-\vec{r})}{|\vec{r}_o-\vec{r}|^2}dR_o - \frac{1}{2\pi}\oint_{B_S} \frac{h_o(\vec{r}_o-\vec{r})\cdot\vec{n}_o}{|\vec{r}_o-\vec{r}|^2}dB_o + \frac{\nu}{2\pi}\oint_{B_S} \frac{\vec{\omega}_o(\vec{r}_o-\vec{r})\cdot\vec{s}_o}{|\vec{r}_o-\vec{r}|^2}dB_o + h_\infty \quad (2.39)$$

where B_s is the internal boundary, and $\vec{n}$ and $\vec{s}$ are the unit normal and tangent vectors, respectively. The pressure coefficient, Cp, is calculated as

$$C_p(\rho=1,\phi) = h(\rho=1,\phi) - h_\infty + 1 \quad (2.40)$$

where ρ and ϕ are the radial and angular coordinates of the computational plane. Having determined the pressure coefficient, the components of aerodynamic force coefficients on the body(ρ = 1), in the computational plane, are computed as follow:

$$C_{N_p} = \frac{1}{L}\int_0^{2\pi} C_p(\phi)\frac{\partial x}{\partial \phi}d\phi \quad (2.41)$$

$$C_{N_f} = \frac{2}{R_e}\int_0^{2\pi} \omega(1,\phi)\frac{\partial y}{\partial \phi}d\phi \quad (2.42)$$

$$C_{T_p} = \frac{1}{L}\int_0^{2\pi} C_p(1,\phi)\frac{\partial y}{\partial \phi}d\phi \quad (2.43)$$

$$C_{T_f} = \frac{2}{R_e}\int_0^{2\pi} \omega(1,\phi)\frac{\partial x}{\partial \phi}d\phi \quad (2.44)$$

$$C_N = C_{N_p} + C_{N_f} \quad (2.45)$$

$$C_T = C_{T_p} + C_{T_f} \tag{2.46}$$

The subscripts N and T denote the normal and tangential components, and p and f the pressure and friction components. The lift and drag coefficients are

$$C_L = C_N \cos\alpha - C_T \sin\alpha \tag{2.47}$$

$$C_D = C_N \sin\alpha + C_T \cos\alpha \tag{2.48}$$

where α is the airfoil angle of attack. The moments are computed with respect to the quarter-chord and are positive counterclockwise since ZETA uses a left-handed coordinate system. The moment coefficients are

$$C_{M_p} = \frac{1}{L^2}\int_0^{2\pi} C_p(\phi)\left[x\frac{\partial x}{\partial \phi} + y\frac{\partial y}{\partial \phi}\right]d\phi \tag{2.49}$$

$$C_{M_f} = \frac{1}{R_e \cdot L}\int_0^{2\pi} \omega(1,\phi)\left[x\frac{\partial y}{\partial \phi} - y\frac{\partial x}{\partial \phi}\right]d\phi \tag{2.50}$$

$$C_M = C_{M_p} + C_{M_f} \tag{2.51}$$

All of the above coefficients are normalized with respect to the airfoil chord length L.

CHAPTER 3

GRID AND BOUNDARY CONDITIONS

3.1 Grid Generation

The grid generation procedure as mentioned in Section 2.2 uses a modified Joukowski transformation. Though a numerical transformation is also available, only the well tested Joukowski transformation will be discussed. This transformation conformally maps the computational plane about the specified airfoil in the physical plane. The grid is composed of concentric circles and radial lines in cylindrical coordinates about a unit circle which is the transformed airfoil surface. The grid generator constructs the physical and computational grids, and computes the parameters of the grid transformation and all the variables which depend only on the specified grid.

The radial lines in the computational plane are evenly spaced about the unit circle. When conformally mapped into the physical plane, the radial lines are more concentrated near the leading and trailing edges (Fig. 3.1). The concentric circles in the computational plane are, however, stretched in the radial direction. With conformal mapping, the stretching will give better resolution near the leading and trailing edges. The grid generator, presently, does not have any adaptive feature like clustering, but the conformal transformation accounts for it by concentrating lines near the surface and the leading and trailing edges.

The stretching formula in the computational plane is

$$\rho = e^{s + c_1} + c_2 \tag{3.1}$$

where

$$s = (n - 1)\Delta s$$

n is the index of the radial grid line and s is the distance from the unit circle. The radial grid extent E in the computational plane and the desired spacing between the first two azimuthal

lines Δs_1 must be specified in order to determine the unknown stretching parameters Δs, c_1 and c_2. A minimal non-dimensional grid extent of 18 that corresponds to approximately 5 chordlengths is recommended. Typically, the boundary layer thickness and the desired number of grid lines within the boundary layer will determine the first grid spacing Δs_1. For some applications, the Δs_1 will have to be inside the laminar sublayer. The laminar boundary layer thickness is [17]

$$\delta = \frac{5 \cdot AL}{\sqrt{R_e}} \tag{3.2}$$

where AL is the non-dimensional airfoil chordlength. The turbulent boundary layer thickness is

$$\delta = \frac{.37 \cdot AL}{R_e^{1/5}} \tag{3.3}$$

Knowing the grid extent E, the radial grid dimension JR, and the first grid spacing Δs_1, along with defining the unit circle will give a system of three equations to determine the three unknown parameters. An area ratio of less than 1.2 was maintained and experience has shown that Δs should be less than 0.15 for an effective grid [10]. The system of equations is

$$\begin{aligned} e^{c_1} + c_2 &= 1.0 \\ e^{\Delta s + c_1} + c_2 &= 1.0 + \Delta s_1 \\ e^{JR\Delta s + c_1} + c_2 &= E \end{aligned} \tag{3.4}$$

3.2 Boundary Conditions

For a given airfoil geometry, the combination of initial and boundary conditions distinguishes the flow patterns. The airfoil and the surrounding fluid are initially at rest. Immediately after time t = 0, the airfoil instantaneously translates at a velocity of $-\vec{v}_\infty$. The flow is potential at this instant. A sheet of vorticity acts as a velocity discontinuity between

the airfoil surface and the undisturbed surrounding fluid. This vorticity sheet will diffuse and convect away with time.

For the solution of the vorticity and velocity vectors, four boundary conditions are required. These are the vorticity and velocity boundary conditions on the flow field's internal and external boundaries. Each of these is a Dirichlet type boundary condition. The internal boundary is the airfoil surface and the external boundary is taken to be at infinity.

The velocity on the external boundary is zero since the fluid far from the airfoil is at rest. The external boundary condition is satisfied exactly. The $\vec{V}_\infty$ term in Equation 2.4 accounts for the relative velocity between the airfoil and the fluid. The velocities on the internal boundary are known through the no slip condition and the prescribed airfoil forward speed, angle of attack, and oscillatory motion. The integral over the internal boundary in Equation 2.4 will generally be non-zero since the airfoil is free to rotate about some body-fixed origin. This boundary integral requires both normal and tangential velocity components, but one is sufficient to uniquely determine the incompressible velocity field in R.

At the external boundary, the vorticity is simply zero. The vorticity external boundary can be located anywhere inside the inviscid zone. Computation of vorticity is confined to just inside the inviscid zone. A zero vorticity gradient normal to the boundary is applied when cutting through the vortical wake is required. The internal boundary vorticities are accurately computed with Equation 2.4 which is part of the kinematic aspect of the problem. Knowing either the normal or tangential velocity component on the airfoil surface, the near-surface vorticity ω can be computed uniquely [18]. However, the same reference shows that using the normal velocity component may introduce numerical difficulties, so the tangential component is used.

CHAPTER 4

NUMERICAL METHOD

The numerical formulation of the governing equations is an integro-differential and zonal methods. Applying the zonal procedure does not require modification of the governing equations. As previously stated, the kinetic and the kinematic aspect of the flow are treated separately through the integro-differential approach.The great advantage of using this approach is that it permits the flow solution to be confined to the viscous regions. The kinematic aspect or the velocity vector equation utilizes the integral approach while the kinetic aspect or the vorticity transport equation utilizes the differential approach. There are three major steps in the computation loop. The loop starts with one kinetic part and followed by two kinematic parts. It may be summarized as [7]

1) Solving the vorticity transport equation, the interior vorticity values at the new time level are computed with the known vorticity and velocity values at the previous time level.
2) Using the newly computed interior vorticity values, new boundary vorticity values are established.
3) A new velocity field is computed with the new vorticity field.

The velocity is computed at grid points and the vorticity is computed at half points in the radial direction. Details of the numerical methods used for the two aspects of the problem will be discussed.

4.1 Kinematic Aspect Of The Flow

The integral representation for the velocity vector is

$$\vec{v}(\vec{r},t) = \frac{-1}{2\pi}\int_R \frac{\vec{\omega}_o\times(\vec{r}_o-\vec{r})}{|\vec{r}_o-\vec{r}|^2}dR_o + \frac{1}{2\pi}\oint_{B_s} \frac{(\vec{v}_o\cdot\vec{n}_o)(\vec{r}_o-\vec{r})}{|\vec{r}_o-\vec{r}|^2}dB_o - \frac{1}{2\pi}\oint_{B_s} \frac{(\vec{v}_o\times\vec{n}_o)\times(\vec{r}_o-\vec{r})}{|\vec{r}_o-\vec{r}|^2}dB_o + \vec{v}_\infty \tag{4.1}$$

The boundary conditions and two-dimensional restrictions have been imposed on this equation. Since the computational grid is in polar coordinates, the velocity components may be expressed as

$$v_r(r,\theta) = \frac{1}{2\pi}\int_R \frac{\omega_o r_o \sin(\theta_o-\theta)}{r_o^2+r^2-2r_o r\cos(\theta_o-\theta)}dR_o - \frac{1}{2\pi}\int_0^{2\pi} \frac{v_{r_o}[r_o\cos(\theta_o-\theta)-r]}{r_o^2+r^2-2r_o r\cos(\theta_o-\theta)}r_o d\theta_o + \frac{1}{2\pi}\int_0^{2\pi} \frac{v_{\theta_o} r_o \sin(\theta_o-\theta)}{r_o^2+r^2-2r_o r\cos(\theta_o-\theta)}r_o d\theta_o + V_{\infty_r} \tag{4.2}$$

$$v_\theta(r,\theta) = \frac{-1}{2\pi}\int_R \frac{\omega_o[r_o\cos(\theta_o-\theta)-r]}{r_o^2+r^2-2r_o r\cos(\theta_o-\theta)}dR_o - \frac{1}{2\pi}\int_0^{2\pi} \frac{v_{r_o} r_o \sin(\theta_o-\theta)}{r_o^2+r^2-2r_o r\cos(\theta_o-\theta)}r_o d\theta_o - \frac{1}{2\pi}\int_0^{2\pi} \frac{v_{\theta_o}[r_o\cos(\theta_o-\theta)-r]}{r_o^2+r^2-2r_o r\cos(\theta_o-\theta)}r_o d\theta_o + V_{\infty_\theta} \tag{4.3}$$

where

$$V_{\infty_r} = v_\infty \cos(\alpha)\cos(\theta) + v_\infty \sin(\alpha)\sin(\theta)$$
$$V_{\infty_\theta} = v_\infty \sin(\alpha)\cos(\theta) + v_\infty \cos(\alpha)\sin(\theta)$$

and $\vec{\omega}$ is again replaced by it sole component ω. As in Equation 4.1, the last two integrals of Equation 4.2 and 4.3 are over the interior boundary B_s which is the airfoil surface. These two equations are also valid in the computational plane if ω_o is replace by $\omega_{o\zeta}$, which is transformed as

$$\omega_{o\zeta} = \omega_o H_o^2 \tag{4.4}$$

This transformation was discussed in the last chapter. In the computational plane, the position components r and θ are replaced by ρ and ϕ. The angle of attack α is positive nose-up as before.

4.1.1 Fourier Series Expansion

Using conventional methods to compute the integrals for the velocity components will be inefficient numerically. If the finite Fourier series expansion is used for the integrals, the vorticity and the velocity components can be evaluated explicitly. The modified equations are

$$\omega_\zeta(\rho,\phi) = \frac{\alpha_o(\rho)}{2} + \sum_{n=1}^{N-1} \left(\alpha_n(\rho)\cos(n\phi) + \beta_n(\rho)\sin(n\phi)\right) + \frac{\alpha_N(\rho)}{2}\cos(N\phi) \tag{4.5}$$

$$v_\rho(\rho,\phi) = \frac{a_o(\rho)}{2} + \sum_{n=1}^{N-1} \left(a_n(\rho)\cos(n\phi) + b_n(\rho)\sin(n\phi)\right) + \frac{a_N(\rho)}{2}\cos(N\phi) \tag{4.6}$$

$$v_\phi(\rho,\phi) = \frac{c_o(\rho)}{2} + \sum_{n=1}^{N-1} \left(c_n(\rho)\cos(n\phi) + d_n(\rho)\sin(n\phi)\right) + \frac{c_N(\rho)}{2}\cos(N\phi) \tag{4.7}$$

Each of the Fourier coefficients is a function of ρ only, assuming the coefficients are constant along any azimuthal grid line. There are $2N = I - 1$ terms in each series. I is the total number of discrete points in the tangential direction about the grid. Since an "O" grid is used, the first and last points overlap. Knowing ω_ζ at all grid points, in particular, at 2N points along an azimuthal line, the Fourier coefficients can be determined as

$$\alpha_k = \frac{1}{N}\sum_{p=0}^{2N-1} \omega_\zeta(\phi_p)\cos(k\phi_p) \qquad k = 0,1,...,N \tag{4.8}$$

$$\beta_k = \frac{1}{N}\sum_{p=0}^{2N-1} \omega_\zeta(\phi_p)\sin(k\phi_p) \qquad k = 0,1,...,N-1 \tag{4.9}$$

Substituting Equations 4.5 through 4.7 into Equations 4.2 and 4.3 in the computational plane and evaluating the integrals, the following relations between the known vorticity Fourier coefficients and the velocity Fourier coefficients are: [12]

$$a_o(\rho) = \frac{a_o(1)}{\rho} \tag{4.10}$$

$$a_1(\rho) = \frac{1}{2}\int_1^\rho \beta_1(\rho_o)\left(\frac{\rho_o}{\rho}\right)^2 d\rho_o + \frac{1}{2}\int_\rho^\infty \beta_1(\rho_o)d\rho_o + \frac{1}{2}\left(\frac{1}{\rho}\right)^2[a_1(1) + d_1(1)] + v_\infty\cos(\alpha) \tag{4.11}$$

$$a_n(\rho) = \frac{1}{2}\int_1^\rho \beta_n(\rho_o)\left(\frac{\rho_o}{\rho}\right)^{n+1} d\rho_o + \frac{1}{2}\int_\rho^\infty \beta_n(\rho_o)\left(\frac{\rho_o}{\rho}\right)^{n-1} d\rho_o + \frac{1}{2}\left(\frac{1}{\rho}\right)^{n+1}[a_n(1) + d_n(1)] \qquad 2 \le n \le N \tag{4.12}$$

$$b_1(\rho) = \frac{1}{2}\int_1^\rho \alpha_1(\rho_o)\left(\frac{\rho_o}{\rho}\right)^2 d\rho_o - \frac{1}{2}\int_\rho^\infty \alpha_1(\rho_o)d\rho_o - \frac{1}{2}\left(\frac{1}{\rho}\right)^2[c_1(1) - b_1(1)] + v_\infty\sin(\alpha) \tag{4.13}$$

$$b_n(\rho) = \frac{1}{2}\int_1^{\rho} \alpha_n(\rho_o)\left(\frac{\rho_o}{\rho}\right)^{n+1} d\rho_o - \frac{1}{2}\int_{\rho}^{\infty} \alpha_n(\rho_o)\left(\frac{\rho_o}{\rho}\right)^{n-1} d\rho_o$$
$$- \frac{1}{2}\left(\frac{1}{\rho}\right)^{n+1}[c_n(1) - b_n(1)] \qquad 2 \le n \le N \tag{4.14}$$

$$c_o(\rho) = \int_1^{\rho} \alpha_o(\rho_o)\left(\frac{\rho_o}{\rho}\right) d\rho_o + \frac{c_o(1)}{\rho} \tag{4.15}$$

$$c_1(\rho) = \frac{1}{2}\int_1^{\rho} \alpha_1(\rho_o)\left(\frac{\rho_o}{\rho}\right)^{2} d\rho_o - \frac{1}{2}\int_{\rho}^{\infty} \alpha_1(\rho_o) d\rho_o$$
$$+ \frac{1}{2}\left(\frac{1}{\rho}\right)^{2}[c_1(1) - b_1(1)] + v_\infty \sin(\alpha) \tag{4.16}$$

$$c_n(\rho) = \frac{1}{2}\int_1^{\rho} \alpha_n(\rho_o)\left(\frac{\rho_o}{\rho}\right)^{n+1} d\rho_o - \frac{1}{2}\int_{\rho}^{\infty} \alpha_n(\rho_o)\left(\frac{\rho_o}{\rho}\right)^{n-1} d\rho_o$$
$$+ \frac{1}{2}\left(\frac{1}{\rho}\right)^{n+1}[c_n(1) - b_n(1)] \qquad 2 \le n \le N \tag{4.17}$$

$$d_1(\rho) = \frac{1}{2}\int_1^{\rho} \beta_1(\rho_o)\left(\frac{\rho_o}{\rho}\right)^{2} d\rho_o - \frac{1}{2}\int_{\rho}^{\infty} \beta_1(\rho_o) d\rho_o$$
$$+ \frac{1}{2}\left(\frac{1}{\rho}\right)^{2}[a_1(1) + d_1(1)] - v_\infty \cos(\alpha) \tag{4.18}$$

$$d_n(\rho) = \frac{1}{2}\int_1^{\rho} \beta_n(\rho_o)\left(\frac{\rho_o}{\rho}\right)^{n+1} d\rho_o - \frac{1}{2}\int_{\rho}^{\infty} \beta_n(\rho_o)\left(\frac{\rho_o}{\rho}\right)^{n-1} d\rho_o$$
$$+ \frac{1}{2}\left(\frac{1}{\rho}\right)^{n+1}[a_n(1) + d_n(1)] \qquad 2 \le n \le N \tag{4.19}$$

$a_n(1)$, $b_n(1)$, $c_n(1)$, and $d_n(1)$ are the known Fourier coefficients of the transformed velocities on the unit circle. By applying these Fourier coefficient equations on the tangential velocity component on the unit circle, the constraints on the vorticity Fourier coefficients are found.

$$\int_1^{\infty} \alpha_1(\rho_o) d\rho_o = -c_1(1) - b_1(1) + 2v_\infty \sin\alpha \tag{4.20}$$

$$\int_1^{\infty} \beta_1(\rho_o)d\rho_o = -d_1(1) + a_1(1) - 2v_\infty \cos\alpha \tag{4.21}$$

$$\int_1^{\infty} \alpha_n(\rho_o)\left(\frac{1}{\rho_o}\right)^{n-1}d\rho_o = -c_n(1) - b_n(1) \tag{4.22}$$

$$\int_1^{\infty} \beta_n(\rho_o)\left(\frac{1}{\rho_o}\right)^{n-1}d\rho_o = -d_n(1) + a_n(1) \tag{4.23}$$

The principle of conservation of total vorticity is employed to add the final constraint to the kinematic aspect of the flow. Using the finite Fourier series expansion of vorticity (Eq. 4.5), the equation can be re-expressed for numerical application

$$\int_R \omega_o dR_o = -2A\Omega$$

$$\int_1^{\infty} \alpha_o(\rho_o)\rho_o d\rho_o = -\frac{A}{\pi}\Omega \tag{4.24}$$

where A is the airfoil cross-sectional area and Ω the airfoil angular speed. The velocity and the vorticity field are uniquely determined with these constraints.

4.2 Kinetic Aspect of the Flow

A combination of finite difference schemes is used to compute a numerical solution of the kinetic part of the problem in the computational plane. Discretization of the vorticity transport equation (Eq. 2.32) generates a set of finite difference equations(FDE) which are algebraic. The working variable is the vorticity as mentioned. As time advances, the employed time-accurate numerical method solves for the vorticity at the half grid points in the radial direction, based on known vorticity values at the previous time level. A first order accurate backward difference scheme is used on time, and a second order accurate central difference scheme in space is used on the diffusive terms. An upwind difference scheme is used on the convective terms. The differencing is second order accurate in the stream-wise

direction and first order accurate in the reverse direction. Collecting the terms of the FDE into a tridiagonal matrix for successive line under-relaxation along the radial line, the FDE becomes

$$A_{i,j}\omega_{i,j+1}^{k+1} + B_{i,j}\omega_{i,j}^{k+1} + C_{i,j}\omega_{i,j-1}^{k+1} = D_{i,j}\left(\omega_{i,j}^{k}, \omega_{i-1,j}^{k+1}, \omega_{i+1,j}^{k}\right) \tag{4.25}$$

The superscripts, k+1, indicate the new time level. The matrix elements for the laminar terms are

$$A_{i,j} = A2 + A4$$

$$= \frac{\rho_{j+1/2}}{(\rho_j - c_2)\Delta s}\left[v_{\rho R}<0\right] - \frac{L/R_e}{(\rho_j - c_2)\Delta s^2}\frac{\rho_{j+1/2}}{(\rho_{j+1/2} - c_2)} \tag{4.26}$$

$$B_{i,j} = B1 + B2 + B3 + B4 + B5$$

$$= \frac{H^2}{\Delta t}\rho_j - \frac{\rho_{j-1/2}}{(\rho_j - c_2)\Delta s}\left[v_{\rho L}<0\right] + \frac{\rho_{j+1/2}}{(\rho_j - c_2)\Delta s}\left[v_{\rho R}<0\right]$$

$$+ \frac{1}{\Delta\phi}\left[v_{\phi R}>0\right] - \frac{1}{\Delta\phi}\left[v_{\phi L}<0\right]$$

$$+ \frac{L/R_e}{(\rho_j - c_2)\Delta s^2}\left[\frac{\rho_{j+1/2}}{(\rho_{j+1/2} - c_2)\Delta s} + \frac{\rho_{j-1/2}}{(\rho_{j-1/2} - c_2)\Delta s}\right] + \frac{2L/R_e}{\rho_j \Delta\phi^2} \tag{4.27}$$

$$C_{i,j} = C2 + C4$$

$$= \frac{\rho_{j-1/2}}{(\rho_j - c_2)\Delta s}\left[v_{\rho L}>0\right] - \frac{L/R_e}{(\rho_j - c_2)\Delta s^2}\frac{\rho_{j-1/2}}{(\rho_{j-1/2} - c_2)} \tag{4.28}$$

$$D_{i,j} = D1 + D3 + D5$$

$$= \frac{H^2}{\Delta t}\rho_j\left(\omega_{i,j}^{k}\right) + \frac{1}{\Delta\phi}\left(\omega_{i+1,j}^{k}\right)\left[v_{\phi R}<0\right]$$

$$- \frac{1}{\Delta\phi}\left(\omega_{i-1,j}^{k}\right)\left[v_{\phi L}>0\right] + \frac{L/R_e}{\rho_j \Delta\phi^2}\left(\omega_{i+1,j}^{k} + \omega_{i-1,j}^{k}\right) \tag{4.29}$$

where

$$v_{\rho R} = v_{\rho_{i,j+1}}$$

$$v_{\rho L} = v_{\rho_{i,j}}$$

$$v_{\phi R} = \frac{1}{4}\left(v_{\phi_{i,J}} + v_{\phi_{i,J+1}} + v_{\phi_{i+1,J}} + v_{\phi_{i+1,J+1}}\right)$$

$$v_{\phi L} = \frac{1}{4}\left(v_{\phi_{i,J}} + v_{\phi_{i,J+1}} + v_{\phi_{i-1,J}} + v_{\phi_{i-1,J+1}}\right)$$

The index j and J are for the vorticity and velocity grids respectively. The conditions in the brackets indicate when the term is included. The relationships between some of the variables are

$$B2 = C2$$

$$B3 = A2$$

$$B4 = -A4 - C4$$

$$B5 = 2 \times D5 \qquad (4.30)$$

When the flow at a grid point is considered turbulent, the diffusion terms, A4, B4, B5, C4, and D5, are multiplied by

$$\left(1 + \frac{\nu_e R_e}{L}\right)$$

This modification is possible by utilizing the definition of ω_ν used in Equation 2.32. In the boundary layer, the discretized vorticity transport equation may be simplified to the boundary layer equation which is parabolic, and the solution is marched forward without iterating. With the zonal approach, the numerical procedure is made even more efficient.

The numerical procedure used for the aerodynamic loads are very similar to that of the velocity vector equation in its integral form. Greater details are found in reference [10].

CHAPTER 5

TURBULENT MODELS

In examining the differential form of the incompressible turbulent, Reynolds-averaged Navier-Stokes momentum equation, there are apparent stress gradients due to transport of momentum by turbulent fluctuations and deformations attributed to fluctuations. To solve these equations by a finite-difference method, some closing assumptions have to be made about these apparent turbulent stresses. In 1877, Boussinesq suggested that the apparent turbulent shearing stresses might be related to the rate of mean strain through an apparent scalar turbulent or "eddy" viscosity [16]. Many turbulence models have been developed but all have limitations. The models range from algebraic to κ–ϵ formulations and the accuracy of these models' predictions varies.

The present code employs the Baldwin-Lomax algebraic turbulence model [12]. This model requires small amount of computational time and has been used extensively. Its accuracy has been found to be comparable to more complex turbulence models [20]. The accuracy of the model will change depending on the flow conditions since it was calibrated and verified with experimental data. The present code using the Baldwin-Lomax turbulence model has been generally under-predicting the aerodynamic loads such as C_L. This differs from most turbulence models which over-predict the value of C_L. The model is used for attached and separated flow regions. Beside the inherent shortcomings of the zero-equation model [16], the calculation of the length scale will not be accurate for separated flow regions. To further develop the code and to improve upon its performance, the Baldwin-Barth turbulence model is to be added to the code. Since the major aim of this investigation is to implement the Baldwin-Barth one-equation turbulence model, this model will be presented in greater details.

5.1 The Baldwin-Lomax Turbulence Model

The Baldwin-Lomax turbulence model is a two-layer, zero-equation, or algebraic eddy viscosity model. This model is patterned after that of Cebeci [21] with modification that avoid the necessity for finding the edge of the boundary layer [12]. The eddy viscosity ν_t is computed as follow

$$\nu_t = \begin{cases} \nu_{ti} & y \le y_{crossover} \\ \nu_{to} & y_{crossover} \le y \end{cases} \tag{5.1}$$

where the subscripts i and o denote the inner and outer layers respectively. $y_{crossover}$ is the smallest value of y at which the eddy viscosity values from both the inner and outer regions are equal.

The inner eddy viscosity ν_{ti} is based on the Prandtl-Van Driest formulation.

$$\nu_{ti} = l^2|\omega| \tag{5.2}$$

where

$$l = ky[1 - \exp(-y^+/A^+)] \tag{5.3}$$

$|\omega|$ is the vorticity magnitude and y^+ is the length scale.

$$y^+ = \frac{yu_\tau}{\nu} \tag{5.4}$$

where u_τ is the friction velocity and ν is the kinematic viscosity. The friction velocity is used to scale the tangential velocity. The scaled tangential velocity is given as reference [22] which is the 'law of the wall'.

$$\begin{aligned} u^+ &= y^+ && \text{in viscous layer} \\ u^+ &= \frac{1}{k}\ln y^+ + C && \text{in inertial layer} \end{aligned} \tag{5.5}$$

C is found by assuming that $y^+ = 10$ is the matching point between the two layers.

The eddy viscosity in the outer region, ν_{to} is given as

$$\nu_{to} = KC_{cp}F_{wake}F_{kleb}(y) \tag{5.6}$$

with

$$F_{wake} = \min(y_{max}F_{max}, C_{wk}Y_{max}u_{dif}^2/F_{max}) \tag{5.7}$$

The value of F_{max} is determined from this function:

$$F(y) = y|\omega|[1 - \exp(-y^+/A^+)] \tag{5.8}$$

and y_{max} is the corresponding y location. F_{kleb} is the Klebanoff intermittency function:

$$F_{kleb}(y) = \left[1 + 5.5\left(\frac{C_{kleb}y}{y_{max}}\right)^6\right]^{-1} \tag{5.9}$$

The quantity u_{dif}^2 is the difference between the maximum and the minimum total velocity in the profile. The constants used are

$A^+ = 26.0$ $\quad C_{cp} = 1.6$ $\quad C_{kleb} = 0.3$

$C_{wk} = 1.0$ $\quad k = 0.4$ $\quad K = 0.0168$

Numerically, the model begins with the determination of friction velocity, u_τ, iteratively. Consequently, the inner and outer eddy viscosity, ν_{ti} and ν_{to}, is computed along the direction normal to the surface. The final eddy viscosity value is assigned according to the $y_{crossover}$ location.

5.2 The Baldwin-Barth Turbulence Model

The originally proposed Baldwin-Barth turbulence model [13] is chosen over the recently modified formulation [23] because applications of the model showed that the original model formulation gave significantly better results [14]. The finite difference method is used to determine the turbulent Reynolds number $\tilde{R}_T$ of the partial differential R_T equation. The turbulent Reynolds number is directly related to eddy viscosity ν_t. The derivation of the κ-R_T and then the R_T model equation begins with a standard form of the k-ε equations.

$$\frac{D\kappa}{Dt} = \nabla \cdot \left(\nu + \frac{\nu_t}{\sigma_\kappa}\right)\nabla\kappa + P - \varepsilon \tag{5.10}$$

$$\frac{D\varepsilon}{Dt} = \nabla \cdot \left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\nabla\varepsilon + c_{\varepsilon_1}\frac{\varepsilon}{\kappa}P - c_{\varepsilon_2}\frac{\varepsilon^2}{\kappa} \tag{5.11}$$

where the total or substantive derivative is

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + V \cdot \nabla \tag{5.12}$$

and P is the production of κ in the equation,

$$P = \nu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \frac{\partial U_i}{\partial x_j} - \frac{2}{3} \nu_t \left(\frac{\partial U_k}{\partial x_k} \right)^2 \tag{5.13}$$

By considering R_T and its differentials,

$$R_T = \frac{\kappa^2}{\nu \varepsilon} \tag{5.14}$$

$$\frac{dR_T}{R_T} = 2\frac{d\kappa}{\kappa} - \frac{d\varepsilon}{\varepsilon} \tag{5.15}$$

a κ-R_T equation is derived from the κ-ε equations with a valid simplification of the diffusion terms.

$$\frac{D(\nu R_T)}{Dt} = (2 - c_{\varepsilon_1}) \frac{\nu R_T}{\kappa} P + (c_{\varepsilon_2} - 2)\kappa + \left(\nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \nabla^2 (\nu R_T) - \frac{1}{\sigma_\varepsilon} (\nabla \nu_t) \cdot \nabla (\nu R_T) \tag{5.16}$$

where

$$\nu_t = c_\mu (\nu R_T) \tag{5.17}$$

and νR_T is the appropriate field variable rather than R_T. To obtain the R_T equation, it will first require the rearrangement of Equation 5.14

$$\varepsilon = \frac{\kappa^2}{\nu R_T} = \frac{(\kappa_1 + \kappa_2)^2}{\nu R_T} \tag{5.18}$$

where $\kappa^2 = (\kappa_1 + \kappa_2)^2$ and assigning a value to κ_1 without loss of generality, at large R_T.

$$\kappa_1^2 = \nu R_T P \tag{5.19}$$

With the above two equations, a relation among k, k_1, and k_2 is obtained.

$$\kappa = \sqrt{\nu R_T P} \left(1 + \frac{\kappa_2}{\kappa_1} \right) \tag{5.20}$$

Substituting Equation 5.20 into Equation 5.16 and rearranging the terms leads to

$$\frac{D(\nu R_T)}{Dt} = (c_{\varepsilon_2} - c_{\varepsilon_1})\sqrt{\nu R_T P} + \left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\nabla^2(\nu R_T) - \frac{1}{\sigma_\varepsilon}(\nabla\nu_t)\cdot\nabla(\nu R_T)$$
$$- (2 - c_{\varepsilon_1})\frac{\kappa_2}{\kappa_1 + \kappa_2}\sqrt{\nu R_T P} - (2 - c_{\varepsilon_1})\kappa_2 \tag{5.21}$$

Neglecting the last two terms in this equation, a self-consistent one-equation model is obtained [13]. This equation is valid over a major portion of the shear layer at sufficiently high R_T. For the model to be applicable in near-wall regions, the turbulent Reynolds number, R_T is re-expressed as

$$R_T = \tilde{R}_T f_3(\tilde{R}_T) \tag{5.22}$$

where f_3 is a damping function that approximates $R_T \approx \tilde{R}_T$ at large R_T. Applying commonly used damping functions in the κ-ε models [13], the eddy viscosity is

$$\nu_t = \nu c_\mu f_\mu R_T = \nu c_\mu f_\mu \tilde{R}_T \tag{5.23}$$

and

$$\varepsilon' = \varepsilon - D = \frac{\kappa^2}{\nu R_T} = \frac{(\kappa_1 + \kappa_1)^2}{\nu f_3 \tilde{R}_T} \tag{5.24}$$

κ_1 is also applicable at small R_T when R_T is replaced with $\tilde{R}_T$ in its definition. The resulting model equation for $\tilde{R}_T$ is

$$\frac{D(\nu\tilde{R}_T)}{Dt} = (c_{\varepsilon_2} f_2 - c_{\varepsilon_1})\sqrt{\nu\tilde{R}_T P} + \left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\nabla^2(\nu\tilde{R}_T) - \frac{1}{\sigma_\varepsilon}(\nabla\nu_t)\cdot\nabla(\nu\tilde{R}_T) \tag{5.25}$$

Subjecting this equation to the thin shear layer approximation and further developments [13], the following list of functions are obtained to determine $\nu\tilde{R}_T$.

$$\frac{1}{\sigma_\varepsilon} = \frac{1}{\kappa^2}(c_{\varepsilon_2} - c_{\varepsilon_1})\sqrt{c_\mu} \tag{5.26}$$

$$\nu_t = c_\mu(\nu\tilde{R}_T)D_1D_2 \tag{5.27}$$

$$\mu_t = \rho\nu_t \tag{5.28}$$

$$D_1 = 1 - \exp(-y^+/A^+) \tag{5.29}$$

$$D_2 = 1 - \exp(-y^+/A_2^+) \tag{5.30}$$

$$f_\mu f_3 = D_1 D_2 \tag{5.31}$$

$$P = \nu_t\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\frac{\partial U_i}{\partial x_j} - \frac{2}{3}\nu_t\left(\frac{\partial U_k}{\partial x_k}\right)^2 \tag{5.32}$$

$$f_2(y^+) = \frac{c_{\varepsilon_1}}{c_{\varepsilon_2}} + \left(1 - \frac{c_{\varepsilon_1}}{c_{\varepsilon_2}}\right)\left(\frac{1}{\kappa y^+} + D_1 D_2\right)\left(\sqrt{D_1 D_2} + \frac{y^+}{\sqrt{D_1 D_2}}\left(\frac{1}{A^+}\exp(-y^+/A_2^+)D_2 + \frac{1}{A_2^+}\exp(-y^+/A_2^+)D_1\right)\right) \tag{5.33}$$

The constants used for the preceding equations are

$k = 0.41$ $\quad c_{\varepsilon_1} = 1.2$ $\quad c_{\varepsilon_2} = 2.0$

$c_\mu = 0.09$ $\quad A^+ = 26$ $\quad A_2^+ = 10$

5.2.1 Numerical Formulation

The numerical solution of the one-equation model is de-coupled from the flow equations. An implicit factored ADI solver for scalar equation is employed on a two-dimensional, logically rectangular mesh. This computational grid, which is different from the cylindrical computational grid of the main flow solver, is chosen to simplify and quicken the implementation process. The finite difference formulation of the model equation is patterned after that of ARC2D [24]. Having to use an "O" grid and computing time-accurately requires, respectively, periodicity and modifications for a time-accurate and integro-differential code for incompressible turbulent flows.

Defining the solution vector R where $R_{i,j} \approx \nu\tilde{R}_T(x_{i,j}, y_{i,j})$, the model equation with discretized advection and diffusion terms becomes a system of ordinary differential equations of the form [13]

$$\vec{R}_t + M(\vec{R})\vec{R} = D\vec{R} \tag{5.34}$$

where $M(\vec{R})$ is a M-type matrix operator representing the discretization of advection and diffusion terms and D is a diagonal matrix. A M-type matrix is diagonally dominant, has positive diagonal entries and negative off-diagonal entries, and has zero row sum.

Since the source term of the one-equation model can be computed explicitly once the velocity gradient is known, only the advection and diffusion terms are discretized. Assuming ν is constant, the model equation may be rewritten as

$$\frac{\partial(\nu\tilde{R}_T)}{\partial t} + V\cdot\nabla(\nu\tilde{R}_T) = 2\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\nabla^2(\nu\tilde{R}_T) - \frac{1}{\sigma_\varepsilon}\nabla\cdot\nu_t\nabla(\nu\tilde{R}_T) + \left(c_{\varepsilon_2}f_2 - c_{\varepsilon_1}\right)\sqrt{c_\mu f_\mu f_3}\,S\tilde{R}_T \tag{5.35}$$

where S is

$$S^2 = \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\frac{\partial U_i}{\partial x_j} \tag{5.36}$$

The advective terms is approximated by first-order accurate upwind differencing:

$$\begin{aligned} V\cdot\nabla(\nu\tilde{R}_T) \approx\ & \alpha_a^x R_{i+1,j} + \beta_a^x R_{i,j} + \gamma_a^x R_{i-1,j} \\ & + \alpha_a^y R_{i,j+1} + \beta_a^y R_{i,j} + \gamma_a^x R_{i,j-1} \end{aligned} \tag{5.37}$$

where

$$\alpha_a^x = \frac{1}{\Delta x}u_{i,j}^- \qquad \gamma_a^x = \frac{1}{\Delta x}u_{i,j}^+ \qquad \beta_a^x = -\left(\alpha_a^x + \gamma_a^x\right)$$

$$\alpha_a^y = \frac{1}{\Delta y}v_{i,j}^- \qquad \gamma_a^y = \frac{1}{\Delta y}v_{i,j}^+ \qquad \beta_a^y = -\left(\alpha_a^y + \gamma_a^y\right)$$

The diffusive terms are approximated by second-order accurate central differencing and may be combined as

$$\begin{aligned} 2\left(\nu + \frac{\nu_t}{\sigma}\right)\nabla^2(\nu\tilde{R}_T) - \nabla\cdot\frac{\nu_t}{\sigma}\nabla(\nu\tilde{R}_T) \approx\ & \alpha_d^x R_{i+1,j} + \beta_d^x R_{i,j} + \gamma_d^x R_{i-1,j} \\ & + \alpha_d^y R_{i,j+1} + \beta_d^y R_{i,j} + \gamma_d^x R_{i,j-1} \end{aligned} \tag{5.38}$$

where

$$\alpha_d^x = \frac{1}{\Delta x^2}\left[2\left(\nu + \frac{\nu_t}{\sigma}\right)_{i,j} - \left(\frac{\nu_t}{\sigma}\right)_{i+1/2,j}\right] \qquad \alpha_d^y = \frac{1}{\Delta y^2}\left[2\left(\nu + \frac{\nu_t}{\sigma}\right)_{i,j} - \left(\frac{\nu_t}{\sigma}\right)_{i,j+1/2}\right]$$

$$\gamma_d^x = \frac{1}{\Delta x^2}\left[2\left(\nu + \frac{\nu_t}{\sigma}\right)_{i,j} - \left(\frac{\nu_t}{\sigma}\right)_{i-1/2,j}\right] \qquad \gamma_d^y = \frac{1}{\Delta y^2}\left[2\left(\nu + \frac{\nu_t}{\sigma}\right)_{i,j} - \left(\frac{\nu_t}{\sigma}\right)_{i,j-1/2}\right]$$

$$\beta_d^x = -\left(\alpha_d^x + \gamma_d^x\right) \qquad \beta_d^y = -\left(\alpha_d^y + \gamma_d^y\right)$$

The matrix operator will not be a M-type matrix due to the coefficients of the diffusive terms if the grid resolution is poor. This condition is strictly enforced in the algorithm.

Typically, a grid wall or first grid spacing of $y^+ = 3.5$ is required since $\tilde{R}_T$ is designed to behave linearly in the near-wall region for zero gradient boundary layers [13]. Therefore, a grid of adequate resolution is critical to the model's performance. The recommended boundary conditions are

1) Solid walls: $\tilde{R}_T = 0$.
2) Inflow: Specify $\tilde{R}_T$ to match experimental ν_t.
3) Outflow: Extrapolate $\tilde{R}_T$ from interior values.
4) Freestream: Set to a small value $\tilde{R}_{T_\infty} < 1$.

A listing of the one-equation model is in Appendix A.

CHAPTER 6

RESULTS AND DISCUSSION

Implementation and verification of the Baldwin-Barth one-equation turbulence model and evaluation of ZETA are described. ZETA is a zonal integro-differential code for incompressible, laminar and turbulent flows. The emphasis will be on the verification and evaluation processes since the field equations and numerical method for the implementation have been described in Section 5.2. The new constraints introduced by the Baldwin-Barth model and modifications of the ZETA require further evaluation of the code. The test cases were chosen for their simple flow characteristics and for correlation with the experimental results [2]. Numerical results are first compared with corresponding results of ZETA with the Baldwin-Lomax algebraic turbulence model and of ARC2D using the Baldwin-Barth model. The comparison will also utilize experimental results when applicable.

Computations are performed on a NACA-0012 airfoil which has been extensively studied. Most cases have a Reynolds number R_e of 3.91×10^6, an angle of attack α of 5.0 degrees, and a freestream Mach number M_∞ of 0.301. This flow condition corresponds to an experimental case of Ref. [2]. It was chosen to minimize compressibility effects so using the incompressible flow solver, ZETA, was justified.

This flow case with simple attached flow features is selected so the newly implemented Baldwin-Barth subroutine can be easily tested. Although ZETA is capable of computing unsteady flow over oscillating airfoils, the computed cases are steady at a given angle of attack. The code is time-accurate and both laminar and turbulent computations were performed (Table I). All computations were done on the CRAY Y-MP 8/832 or C-90 but the code can be run on VAX, IRIS, or SUN workstations.

ZETA can only compute on the "O" grid which is created with a grid generator that is used exclusively for this code. The grid generator also provides the transformation

factors and other grid dependent variables for the flow solver. Creating a grid normally requires a CPU time of less than 1.0 second. The computation begins when the airfoil starts impulsively from rest at a time immediately after t = 0. For most flows, the solutions are considered steady at a non-dimensional time of about 150. This time is equivalent to having the airfoil travel 42 chord-lengths through the fluid since the non-dimensional freestream velocity is 1.0 and the chord length of an airfoil is 3.6190589 due to the conformal mapping scheme. The typical number of iterations to reach a non-dimensional time of 150 is about 7,600. This corresponds to a total CPU time of about 4.5 hours, or a real time of about 2 to 3 days. The convergence rates vary with grids, angles of attack, and Reynolds numbers.

6.1 Implementation of the Baldwin-Barth Turbulence Model

The implementation of the Baldwin-Barth one-equation turbulence model is patterned after that of ARC2D [13]. Modification of the ZETA code was also necessary. The major change in ZETA involved the computation of physical velocity for the Baldwin-Barth subroutine. Presently, the model subroutine is capable of handling the periodicity of an "O" grid, the left-handed coordinate system, and the time-accurate aspects of ZETA. Since the model subroutine is de-coupled from the flow solver, a conventional rectangular computational grid was chosen for convenience.

The numerical procedures in the model subroutine include computations of the metrics and Jacobians of transformation [16], a generalized distance function which gives the minimum distance to the solid wall [25], and the gradient of the mean velocity. The solution of the model equation utilizes an alternating-direction implicit(ADI) scheme in a half-stagger grid. The eddy viscosity is provided at half points in the radial direction. The "O" grid is periodic in the tangential direction which requires a periodic scalar tridiagonal solver for the system of finite difference equations in the computational plane. A Thomas tridiagonal solver is used for the non-periodic, normal direction.

For a given iteration, the model subroutine will require the input of R_e, velocity components(u, v), and the physical grid geometry(x,y) from the flow solver. Using these informations, the subroutine will model the corresponding eddy viscosity of the flow. A flowchart of the Baldwin-Barth one-equation model subroutine is presented in Figure 6.1.

6.2 Verification of the Baldwin-Barth Turbulence Model

Testing was conducted to verify that the subroutine works properly. The constraints imposed by the "O" grid and conformal mapping complicated the verification process. The ideal test case would be a turbulent flow on a flat plate. With no pressure gradient, the flow characteristic is dominated by eddy viscosity since the molecular viscosity is small. Flow on a flat plate has been thoroughly investigated and analytical descriptions of the flow are available for comparisons. However, numerical computation of flow on a flat plate with "O" grid has proven to be extremely difficult. The grid spacing of a zero-thickness, flat plate collapses near the surface and especially at the leading and trailing edges. Difficulties in calculating the transformation metrics makes proper transformation of such grids nearly impossible. A pseudo-flat plate was not used due to similar restrictions associated with conformal mapping. Therefore, a NACA-0012 airfoil was chosen as the test body.

ARC2D and ZETA, using the Baldwin-Barth turbulence model, computed cases for comparison. The test cases ran at $R_e = 3.91 \times 10^6$ and $\alpha = 5.0$ degrees on a 120x80 grid. The velocity magnitude contours and eddy viscosity distributions of ZETA and ARC2D are shown in Figure 6.2 through Figure 6.5. Although the figures show global similar features, ZETA with the Baldwin-Barth model predicts flow separation early. Separation is indicated by flow reversal. As in the ARC2D case, ZETA with Baldwin-Lomax does not show any flow separation (Fig. 6.6). The corresponding vorticity is presented in Figure 6.7. The unexpected flow separation could be caused by the incorrect implementation of the Baldwin-Barth model or undesirable interactions between ZETA and constraints imposed by the turbulence subroutine. The ZETA code and these probable causes were examined.

6.2.1 Test of the Isolated Turbulence Subroutine

The field variable of the model subroutine is the turbulent Reynolds number, $\tilde{R}_T$ which is directly proportional to eddy viscosity. Thus, the prediction of $\tilde{R}_T$ must be correct. An effective way of testing the turbulence subroutine is to isolate and place it in a small test program. Receiving a converged ARC2D velocity solution field(u,v) along with the grid geometry(x,y), the subroutine would restart and iterate until $\tilde{R}_T$ converged.

The eddy viscosity, ν_t, was calculated with the converged $\tilde{R}_T$ of the test program and compared with the ν_t of ARC2D. Figure 6.8 shows the global features of the test case which are in good agreement with that of ARC2D (Fig. 6.5). The trailing edge features of the ν_t distribution for both ARC2D and test case also compare well (Fig. 6.9 and Fig. 6.10). A more detailed comparison of the $\tilde{R}_T$ profiles at six locations on the more critical top surface of the airfoil is in Figure 6.11. The matching ν_t distributions and $\tilde{R}_T$ profiles indicate that the subroutine in ZETA predicts eddy viscosity correctly.

Another convincing comparison is the velocity distributions u^+ vs y^+, which is computed in the subroutine with the mean velocity and geometry, at mid-chord for both ZETA and ARC2D cases (Fig. 6.12). The good agreement of the velocity profiles at mid-chord as well as other chord stations also affirms that the model subroutine in ZETA is producing the correct eddy viscosity for a given flow. It is also interesting to note that the velocity profile is similar to that of a flat plate (Fig. 6.13). Although ZETA with the Baldwin-Barth model predicts flow separation early, the above tests show that the model subroutine is implemented correctly. The focus will be shifted to the ZETA code and its numerical interactions with the Baldwin-Barth model subroutine.

However, it is important to note that ARC2D and the isolated model test cases are time-average runs but ZETA is time-accurate at all time. Recommended modifications for the time-accurate code were made to the model subroutine in ZETA. Since no known time-accurate computation with the Baldwin-Barth model is available, the model subroutine's numerical capability to compute time-accurately may require further examination.

6.3 Omission of S_ω

The governing equations of ZETA are the continuity equation and the vorticity transport equation. The vorticity transport equation (Equation 2.2) has neglected a scalar term, S_ω:

$$S_\omega = 2\left[\frac{\partial^2}{\partial x^2}\left(\nu_e\frac{\partial u}{\partial y}\right) - \frac{\partial^2}{\partial y^2}\left(\nu_e\frac{\partial v}{\partial x}\right) + \frac{\partial^2}{\partial x\partial y}\left(\nu_e\frac{\partial v}{\partial y}\right) - \frac{\partial^2}{\partial x\partial y}\left(\nu_e\frac{\partial u}{\partial x}\right)\right]$$

This is justified below by the order-of-magnitude analysis of S_ω.

For a laminar boundary layer, the boundary layer thickness is proportional to the square root of the kinematic viscosity [17]:

$$\delta \sim \sqrt{\nu}$$

This boundary layer thickness is very small compared to the chord of the airfoil. The dimensionless δ is therefore very small compared to unity. ν_e is of the order δ as indicated in the numerical computations using both Baldwin-Lomax and Baldwin-Barth turbulence models. This implies that the inverse of the turbulent Reynolds number:

$$\frac{1}{R} = \frac{\nu_e}{V_\infty C}$$

is also of the order δ. Therefore, the orders of magnitude of the individual terms of S_ω, which is nondimensionalized with the free-stream velocity, V_∞, and the chord, C, are as follow:

$$S_\omega = 2\left[\frac{\partial^2}{\partial x^2}\left(\frac{1}{R}\frac{\partial u}{\partial y}\right) - \frac{\partial^2}{\partial y^2}\left(\frac{1}{R}\frac{\partial v}{\partial x}\right) + \frac{\partial^2}{\partial x\partial y}\left(\frac{1}{R}\frac{\partial v}{\partial y}\right) - \frac{\partial^2}{\partial x\partial y}\left(\frac{1}{R}\frac{\partial u}{\partial x}\right)\right]$$

$$1 \quad \delta\,\frac{1}{\delta} \quad \frac{1}{\delta^2}\,\delta\,\delta \qquad \frac{1}{\delta} \quad \delta\,\frac{\delta}{\delta} \qquad \frac{1}{\delta} \quad \delta\,1$$

Thus, S_ω is of order one. The vorticity transport equation is of the order $1/\delta$. Therefore, the omission of S_ω in the vorticity transport equation is justified.

Having shown that the turbulence model subroutine works properly and the derivation of the governing equations was correct, the numerical aspects of ZETA and its interactions with the turbulence model subroutine will then be investigated.

6.4 The Effects of Grid

The grid has the utmost effect on computational accuracy of any case. Depending on the numerical methods of a code, constraints or demands on the grid will be different and sometimes conflicting. Grid requirements will also change with flow conditions. For conventional finite difference codes, a finer grid will generally increase the accuracy of the computation. This grid effect applies to the Baldwin-Barth model subroutine which utilizes the finite difference formulation. However, the integro-differential approach with Fourier series expansions of ZETA places unique demands on grid resolution.

The Baldwin-Barth numerical formulation requires good grid resolution in both tangential and normal directions of the airfoil. To capture the flow features in the boundary layer requires proper grid spacing, especially in the flow direction and in the near-wall region. Having the appropriate first grid spacing off the airfoil surface is also essential. The Baldwin-Barth subroutine requires a grid point to be inside the laminar sublayer for proper estimation of the velocity profile [13]. The slope of the profile is used in computing the friction velocity in the subroutine. Inaccurate approximation of the slope will cause the calculation of y^+, the damping factors, and hence eddy viscosity to be incorrect (Fig. 6.1). Computations with a grid having the first velocity grid point off the airfoil surface outside of the laminar sublayer will give inaccurate flow solutions.

In contrast, the flow solver, ZETA, prefers a coarse grid with a minimum of only about 10 to 20 grid points inside the boundary layer [10]. Testings also found that the first radial grid point off the airfoil surface should be outside of the laminar sublayer. This unusual grid requirement is a result of the numerical formulation in the code. When the airfoil is impulsively started to move in the fluid, the kinematic aspect of the code computes boundary vorticity on the first ring of vorticity grid points off the surface (Fig. 2.2). This procedure integrates over the viscous zone outside of the first ring to obtain the boundary vorticity. Using the known velocities at the previous time level and the boundary vorticities, the vorticity transport equation computes the vorticity in the remaining viscous

zone. The boundary condition is updated at each time level. However, the kinematic aspect of the code could not compute constant vorticity at grid points inside the laminar sublayer. The numerical error will then accumulate as the time level advances and will have a great impact on the accuracy of the boundary vorticity and the overall flow solution.

This premise is tested by running ZETA in the laminar mode where the effect of the turbulence model is removed. Changing the first radial velocity grid spacings off the surface has significant effects on the solution of the given flow (Fig. 6.14 to 6.16). The flow separation region progressively increases as the first grid spacing decreases from 0.001 to 0.00006. The amount of separation should not change with the first radial grid spacing. The case with more grid points inside the laminar sublayer show unreasonably large amount of separations (Fig. 6.16). The numerical problem of updating the vorticity boundary inside the laminar sublayer appeared to cause the flow differences. Another probable reason for the separation differences is the higher numerical dissipation rate of coarse grids. Numerically, the flow is more likely to remain attached on coarser grid on which disturbances quickly dissipate.

For the same flow condition and a first grid spacing of 0.0001, systematically reducing all the numerical eddy viscosity to 20%, 10%, and zero percent of the original Baldwin-Barth subroutine value does not change the flow features significantly (Fig. 6.17 to 6.19). The flows with reduced eddy viscosity behave like the normal turbulent flow. Such behavior suggests that the flow solver may be introducing some interferences with greater effects than the computed eddy viscosity.

ZETA with the Baldwin-Barth model running on a grid with the first ring of grid points outside of the laminar sublayer did not show separation (Fig. 6.20 and 6.21) but the predicted aerodynamic loads are low. This is expected since the model requires a grid point within the laminar sublayer. However, having any grid point in the laminar sublayer will lead to inaccurate flow solution with ZETA.

The conflicting grid requirements of ZETA and the Baldwin-Barth model make

accurate flow computation difficult unless a solution to the problem is found. For the above case using the Baldwin-Barth model (Fig. 6.2), a systematic investigation of grid size and first grid spacing showed that a grid of 120x80 with a first grid spacing of 0.0001 gives the best but still incorrect solution. Adding grid lines in the tangential direction did not improve the flow solution. On the same 120x80 grid, ZETA with the Baldwin-Lomax model under-predict the aerodynamic loads such as C_L by about 20% although it showed correct global features(Fig. 6.6). A coarser grid is recommended when using the Baldwin-Lomax turbulence model. A typical grid size of 80x45 was used for previous investigations of airfoils. The first grid spacing varies slightly with R_e.

6.5 The Effects of Reynolds Number

Most numerical codes have a operating range of Reynolds numbers for which the resulting flow solutions are accurate. Generalizing from previous work, ZETA is not expected to perform well for cases where R_e is less than $2x10^5$. No other known Reynolds number effect on the flow solution is reported. Codes with the Baldwin-Barth model generally run at a R_e of $2x10^6$ or greater. Further investigation of Reynolds number effect was performed since the Baldwin-Barth model introduces new R_e and grid demands. By running the NACA-0012 airfoil at $\alpha = 5^o$ through a range of Reynolds numbers, it was found that R_e affects ZETA with both Baldwin-Lomax and Baldwin-Barth models.

R_e from $1x10^4$ to $4x10^6$ were tested with ZETA to examine the tangential velocity and vorticity along the top surface of the airfoil. The tangential velocity and vorticity along the airfoil surface start to fluctuate between R_e of one and two millions with the Baldwin-Lomax model (Fig. 6.22 and 6.23). The normal velocity and vorticity exhibit no such fluctuations. There is also no fluctuation on the lower airfoil surface. The fluctuations did not disappeared when more lines were added in the tangential direction of the grid to improve the velocity gradient calculation. Cases with the Baldwin-Barth model showed more severe fluctuations in the tangential surface velocity and vorticity. Fluctuations begin

at a R_e of about $1x10^5$ and becomes proportionally worse as R_e increases (Fig. 6.24 and Fig. 6.25). The Baldwin-Barth model equation is a transport equation which may be the cause of the more severe fluctuations. Transport equations tend to accentuate disturbances in a code. Fluctuations also existed in the laminar mode using a fine grid. There is no evidence of fluctuations when using a coarse grids even at high Reynolds number (Fig. 6.21). Having fluctuations in the laminar mode and with both turbulence models hint that the flow solver presently may not be able to accommodate a fine grid at higher R_e. However, the grids used for ZETA could be considered coarse for most finite difference Navier-Stokes codes.

6.6 The Performance of Turbulence Models

For most applications of turbulence models including Baldwin-Lomax, Navier-Stokes codes would normally over-predict C_L. ZETA with the Baldwin-Lomax turbulence model however has been under-predicting C_L and the predicted C_L is generally lower at higher R_e. The lift slope in Figure 6.26 was computed at $R_e = 3.91x10^6$ and $\alpha = 5$ degrees on a 80x45 grid. The experimental result on the figure is from Reference 2. The lower C_L values are consistent with ZETA's performance but the cause of the difference is not known. The performance of ZETA with the zero-equation model is generally acceptable for flows where separation is minimum. Flows with large separation region are common in unsteady aerodynamic investigations where dynamic stall is often encountered. As expected, ZETA's performance for these flows and high angle of attack flows are not as reliable. Two of the possible causes are compressibility effect and poor performance of the turbulence model in separated flows. The calculation of the length scale in the zero-equation model is known to be inaccurate for separated flows and this is true in ZETA.

Though the overall performance of ZETA with the Baldwin-Lomax model is reasonable when used properly, it remains questionable. As shown in Figure 6.17 to 6.19, changing the eddy viscosity magnitude did not change the Baldwin-Barth flow solutions

which are slightly separated. When the Baldwin-Lomax model is applied, the flow remains attached. Beside the eddy viscosity magnitude, the only other factor involved is the eddy viscosity distribution since the Baldwin-Barth subroutine can only affects the flow solution by the computed eddy viscosity. The eddy viscosity distribution of the Baldwin-Lomax case is typically like that of Fig. 6.7. Note that the eddy viscosity, ν_t, is constant in large area near the airfoil. This is unrealistic. Examining the velocity distribution, the Baldwin-Lomax case is also different from that of ZETA and ARC2D with the Baldwin-Barth model (Fig. 6.27). The reason for the attached flow when using the Baldwin-Lomax model is unclear. ZETA with the Baldwin-Lomax model however tends to delay flow separation (Fig. 6.26).

There are advantages in using the Baldwin-Lomax model in ZETA. It is not time or memory intensive. A coarse grid(80x45) can be used and tends to provide better flow solutions than finer grids. With Baldwin-Lomax, steady and dynamic flow cases can also be ran quickly and interactively on the CRAY Y-MP.

The Baldwin-Barth one-equation turbulence model has been implemented in mainly finite difference codes. The solution of the model equation requires a relatively significant addition of memory and time due to its added complexity and grid resolution requirement. Although the Baldwin-Barth model has been proven to be reliable in many flow conditions over airfoils, it has not been performing well in ZETA. As discussed in the previous sections, disruptive fine grids and R_e factors might of contributed to the poor performance of ZETA with the Baldwin-Barth model. Therefore, extensive quantitative comparison with experimental results was not conducted.

For a flow at $R_e = 3.91 \times 10^6$ and $\alpha = 20$ degrees, the velocity contour shapes with both turbulence models are comparable (Fig. 6.28 and 6.29) but the values are different. At this condition, the airfoil is stalled and the surface boundary layer effects are negligible. Flow solvers are not expected to do well in these stall cases. However, the Baldwin-Barth turbulence model seems to perform better when the disruptive grid effect at higher R_e is

small. There is no fluctuation in the flow solution. The eddy viscosity distribution of the Baldwin-Barth model seems to be more reasonable than that of the Baldwin-Lomax model for separated flows (Fig. 6.30 and 6.31). One reason for the unrealistic eddy viscosity distribution of Baldwin-Lomax (Fig. 6.31) is that the subroutine imposed a maximum eddy viscosity limit of 100 times the laminar viscosity. In the development process [10], the Baldwin-Lomax model performed better with this imposed upper limit on eddy viscosity.

6.6.1 Aerodynamic Loads Prediction

In the prediction of aerodynamic loads: C_L, C_D, and C_M, ZETA with Baldwin-Lomax performs well with a coarse, 80x45 grid (Fig. 6.26). Even the suction peak of the C_p distribution matches well with the experimental data (Fig. 6.32). The computation of C_p however deteriorates with finer grids like those used for the Baldwin-Barth model. In the same figure, the C_p distribution computed with the Baldwin-Barth model indicates that it performs well especially near the leading edge but shows early flow separation. The C_p distribution fluctuates on the upper surface of the trailing edge. The fluctuation problem is most obvious in the time convergence history of C_L (Fig. 6.33). ZETA with the Baldwin-Lomax model shows no fluctuation on a coarse grid(80x45). The C_L compares well but it is expectedly lower than the experimental value of 0.58. It is much lower when using a finer grid(120x80). In the Baldwin-Barth case (Fig. 6.33), the average $C_L = 0.48$ is low but the peak $C_L = 0.53$ is comparable to the experimental $C_L = 0.58$. C_L does not fluctuate on grids with the first grid point outside of the laminar sublayer (dy = 0.01) but this model is not expected to perform well on such grid. The average of the fluctuating C_L values is also comparable to the experimental data before stall occurs (Fig. 6.34). However, ZETA with the Baldwin-Barth model is presently ineffective. The model would improve ZETA's performance in separated flows when the conflicting grid requirement is solved. Without the high frequency fluctuations, the results with the Baldwin-Barth model are promising.

CHAPTER 7

CONCLUDING REMARKS

7.1 Conclusions

The Baldwin-Barth one-equation turbulence model has been implemented into ZETA. The isolated turbulence model subroutine works correctly. ZETA with the Baldwin-Barth turbulence model can not be used for practical flow applications due to their conflicting grid requirements. Baldwin-Barth requires a point inside the laminar sublayer but ZETA does not work well with such grid. The implementation process revealed some limitations of ZETA and the integro-differential scheme. Major conclusions of this investigation includes:

(1) The first grid point off the solid body surface should be outside of the laminar sublayer to compute accurately with ZETA.

(2) For airfoils, ZETA should be run at R_e greater than $2x10^5$.

(3) A fine grid and higher R_e will cause vorticity and velocity fluctuations.

(4) The grid generator used for ZETA is simple but rigid. The inability to cluster grid lines and restriction to the use of the "O" grid hampered the study.

(5) ZETA with the Baldwin-Lomax turbulence model seems to give better result, but the computed eddy viscosity distribution remains questionable.

7.2 Recommendations

Future development and work with ZETA should consider the use of an effective grid generator that is independent of the code and has adaptive capability. The conflicting grid requirement may be resolved by refining the computation of vorticity in the laminar sublayer in ZETA. The code should be expanded to include the compressible flow case. Recent work on vortex trapping to counteract unwanted dissipations has proven to be promising and ZETA should have such a feature since a zonal procedure is used.

TABLE I: Summary of Selected Test Cases

Study Case #	Reynolds Number, R_e	Angle of Attack, α	Grid Size	First Grid Spacing, dy	Code and Comments
1	$3.91x10^6$	5^o	120x80	0.0001	Z/BB
2	$3.91x10^6$	5^o	120x80	0.00006	ARC2D
3	$3.91x10^6$	5^o	120x80	0.0001	Z/BL
4	$3.91x10^6$	5^o	120x80	0.00006	Isolated BB with ARC2D inputs
5	$3.91x10^6$	5^o	120x80	0.001	Laminar
6	$3.91x10^6$	5^o	120x80	0.0001	Laminar
7	$3.91x10^6$	5^o	120x80	0.00006	Laminar
8	$3.91x10^6$	5^o	120x80	0.0001	Z/BL with 20% of the computed ν_t
9	$3.91x10^6$	5^o	120x80	0.0001	Z/BL with 10% of the computed ν_t
10	$3.91x10^6$	5^o	120x80	0.0001	Z/BL with 0% of the computed ν_t
11	$3.91x10^6$	5^o	120x80	0.01	Z/BB
12	$1x10^6$	5^o	120x80	0.0001	Z/BL
13	$2x10^6$	5^o	120x80	0.0001	Z/BL
14	$1x10^4$	5^o	120x80	0.0001	Z/BB
15	$5x10^5$	5^o	120x80	0.0001	Z/BB
16	$2x10^6$	5^o	120x80	0.0001	Z/BB
17	$3.91x10^6$	3^o	80x45	0.001	Z/BL
18	$3.91x10^6$	5^o	80x45	0.001	Z/BL
19	$3.91x10^6$	10^o	80x45	0.001	Z/BL
20	$3.91x10^6$	13^o	80x45	0.001	Z/BL

Z/BB: ZETA with Baldwin-Barth(BB) turbulence model.

Z/BL: ZETA with Baldwin-Lomax(BL) turbulence model.

TABLE I, Continued

Study Case #	Reynolds Number, R_e	Angle of Attack, α	Grid Size	First Grid Spacing, dy	Code and Comments
21	3.91×10^6	15^o	80x45	0.001	Z/BL
22	3.91×10^6	16^o	80x45	0.001	Z/BL
23	3.91×10^6	20^o	80x45	0.001	Z/BL
24	1×10^6	20^o	120x80	0.0001	Z/BB
25	1×10^6	20^o	120x80	0.0001	Z/BL

Z/BB: ZETA with Baldwin-Barth(BB) turbulence model.

Z/BL: ZETA with Baldwin-Lomax(BL) turbulence model.

Figure 1.1: UH-60A Black Hawks in forward flight.

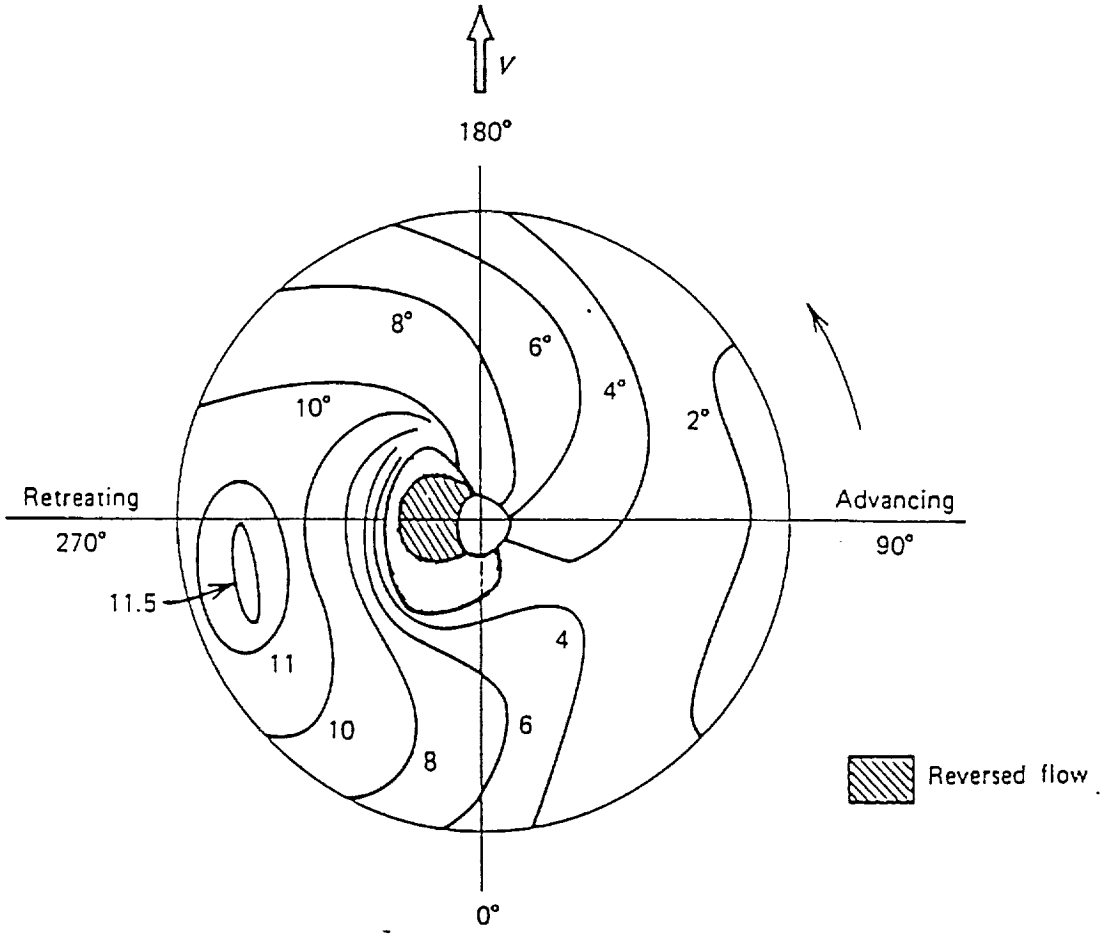


Figure 1.2: Example of angle of attack distribution in forward flight of twisted blade.

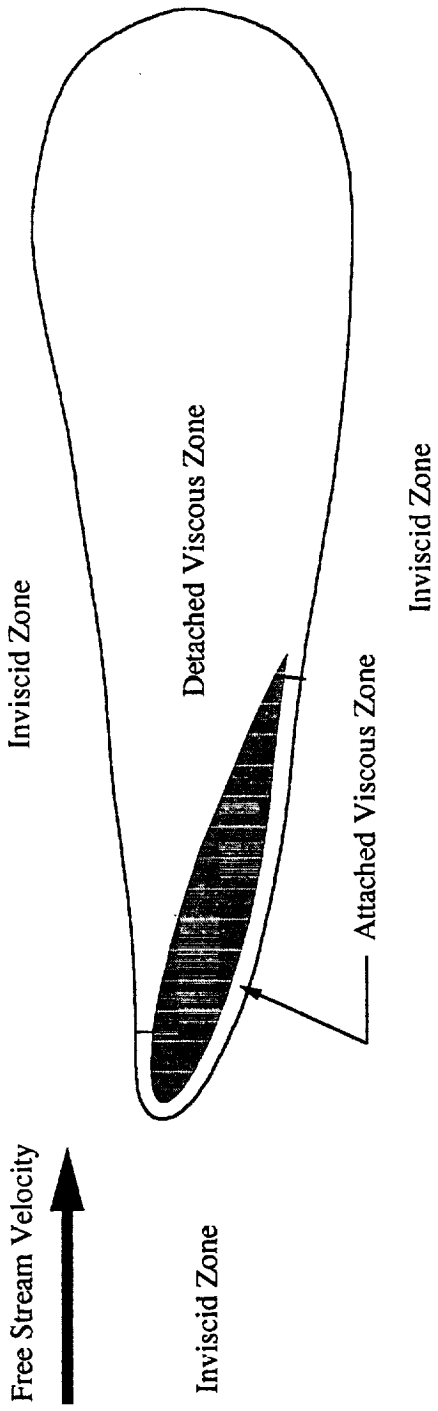
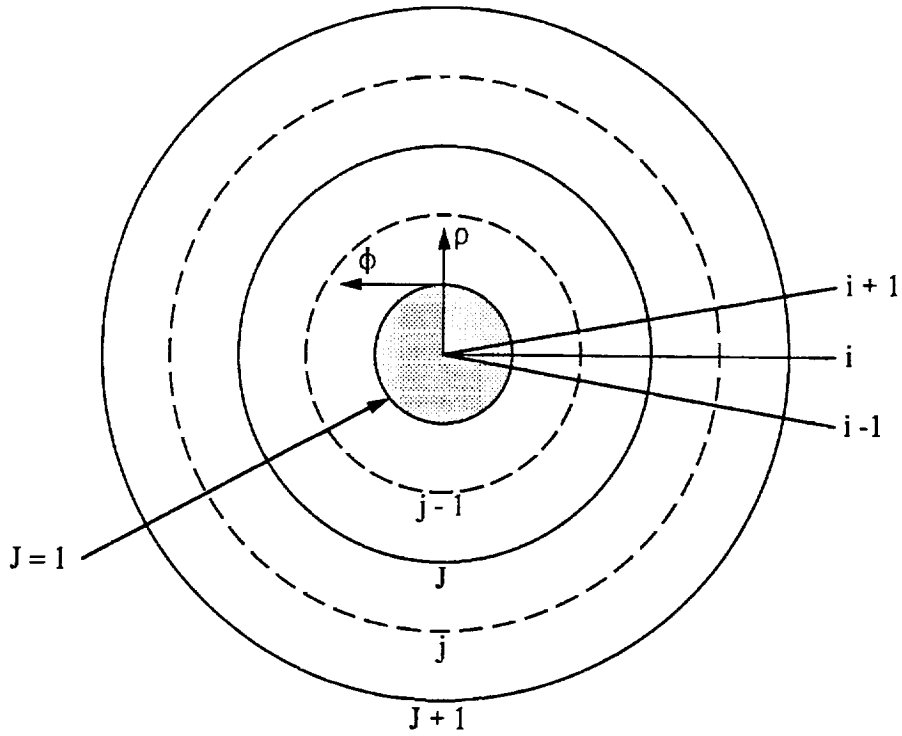
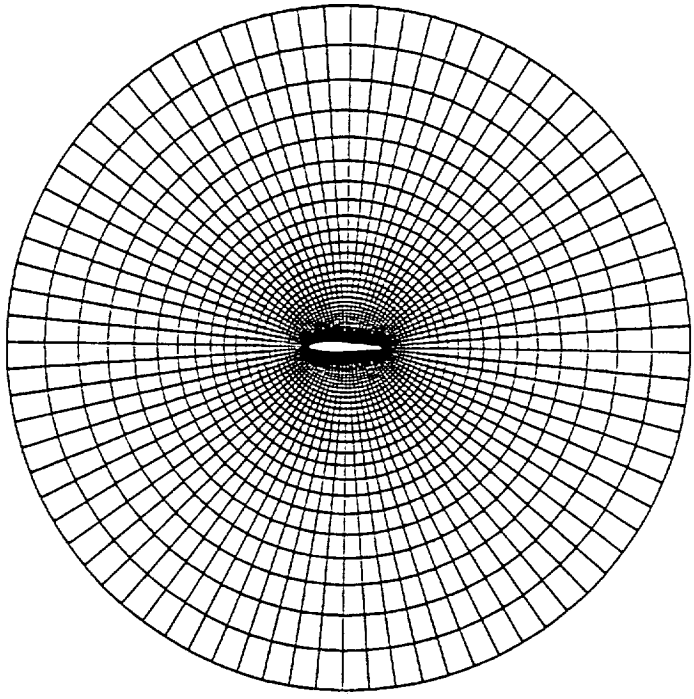


Figure 2.1: Division of flow zones.



J => velocity grid, j => vorticity grid

Computational grid



Physical grid

Figure 2.2: Computational and physical grids.

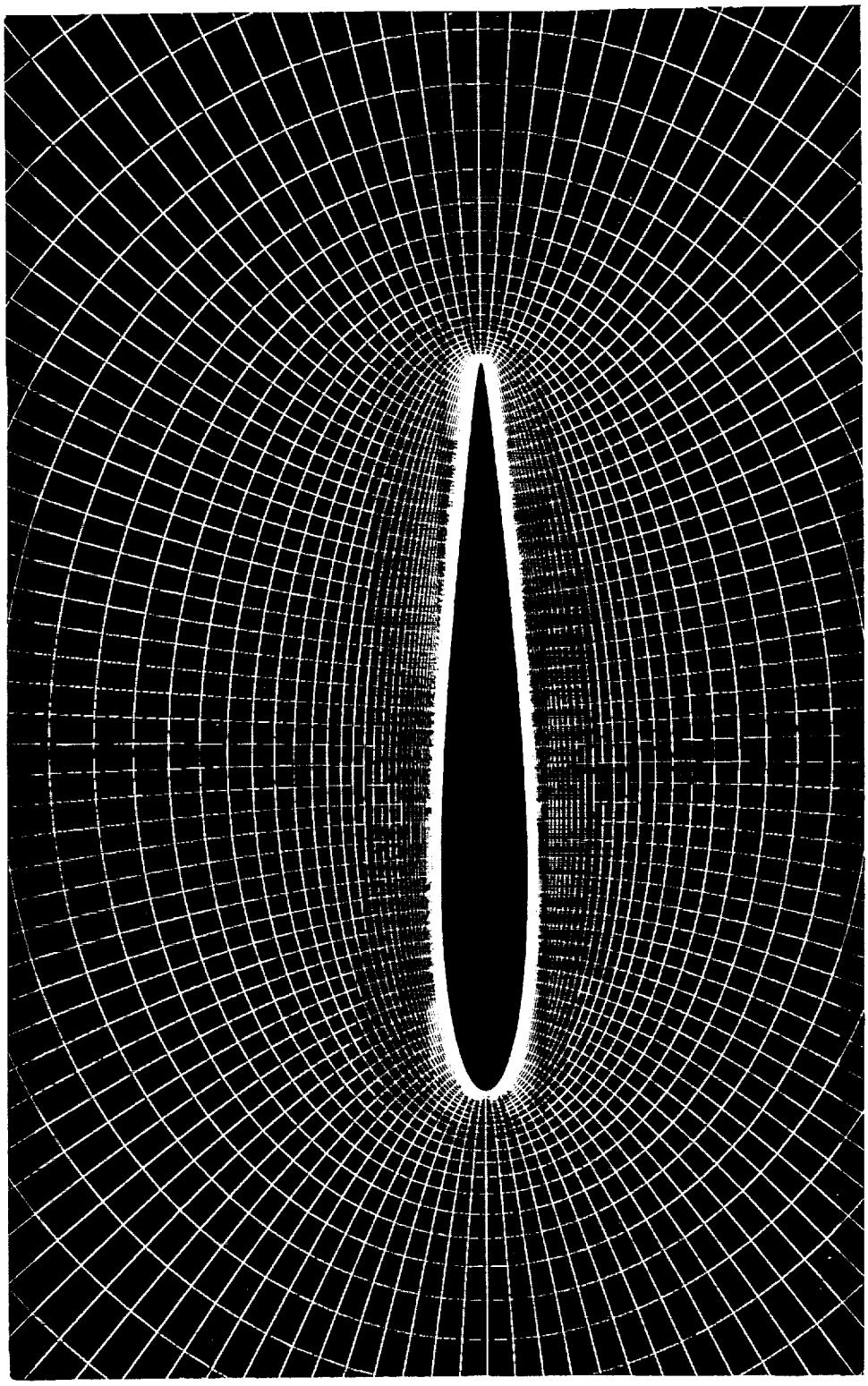


Figure 3.1: Clustering of radial lines at the leading and trailing edges.

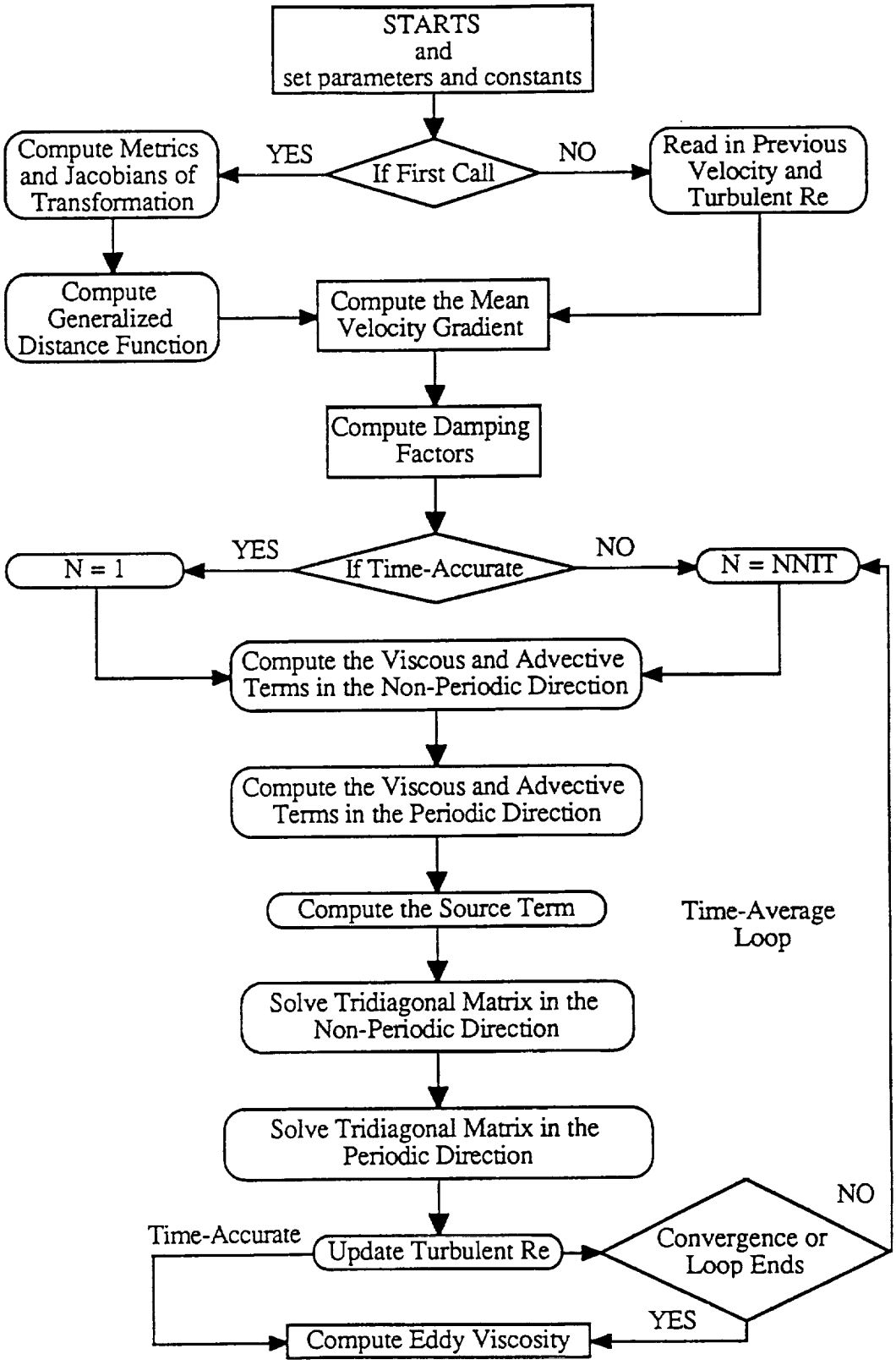


Figure 6.1: Flowchart of the Baldwin-Barth subroutine.

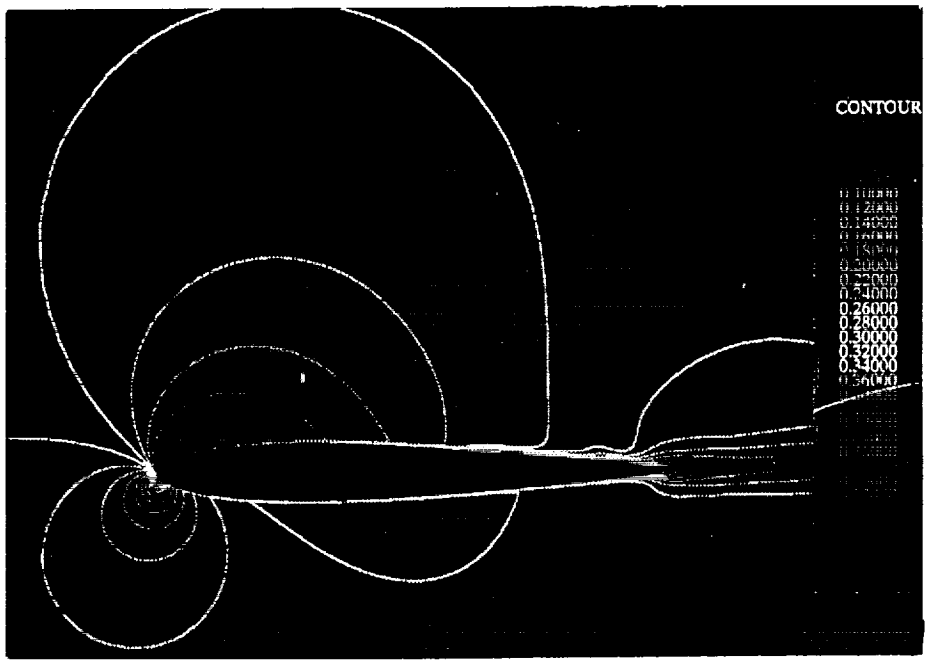


Figure 6.2: Velocity contours of an NACA-0012 airfoil at R_e =3.91x10^6 and α = 5° from ZETA with Baldwin-Barth turbulence model.

Figure 6.3: Eddy viscosity contours of an NACA-0012 airfoil at R_e =3.91x10^6 and α = 5° from ZETA with Baldwin-Barth turbulence model.

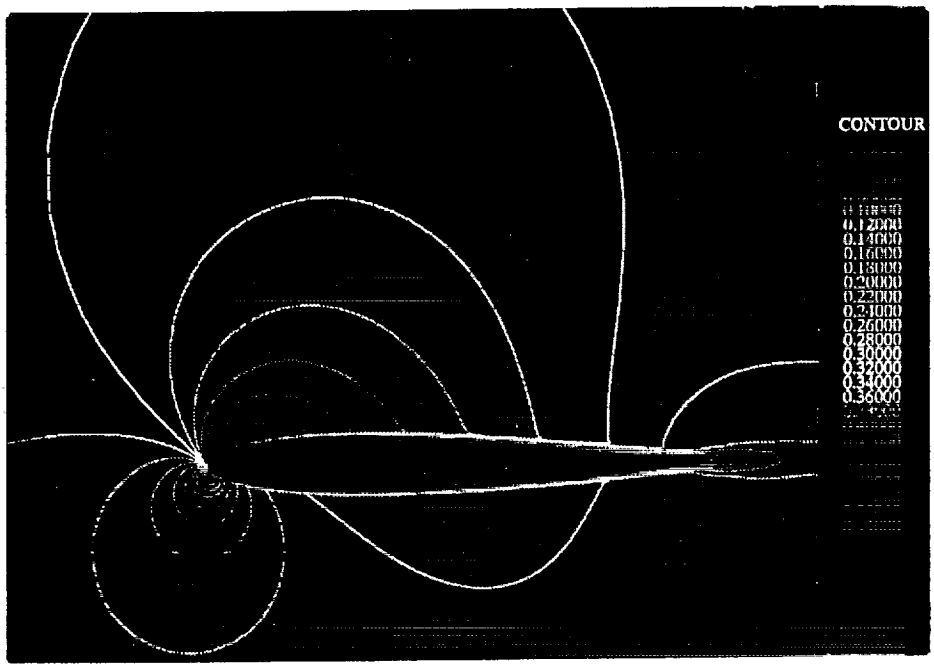


Figure 6.4: Velocity contours of an NACA-0012 airfoil at R_e =3.91x10^6 and $\alpha = 5^o$ from ARC2D with Baldwin-Barth turbulence model.

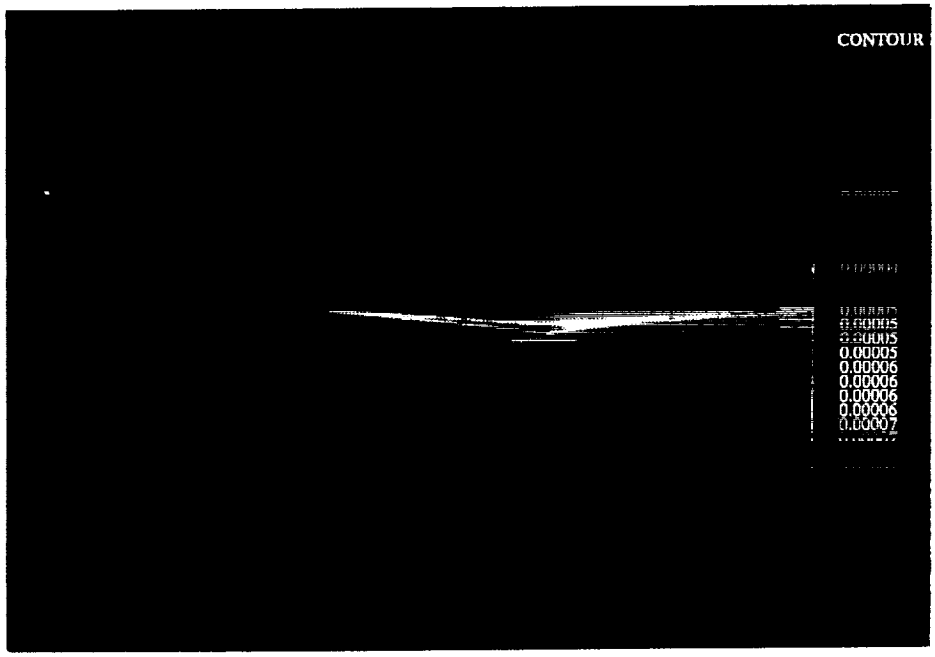


Figure 6.5: Eddy viscosity contours of an NACA-0012 airfoil at R_e =3.91x10^6 and $\alpha = 5^o$ from ARC2D with Baldwin-Barth turbulence model.

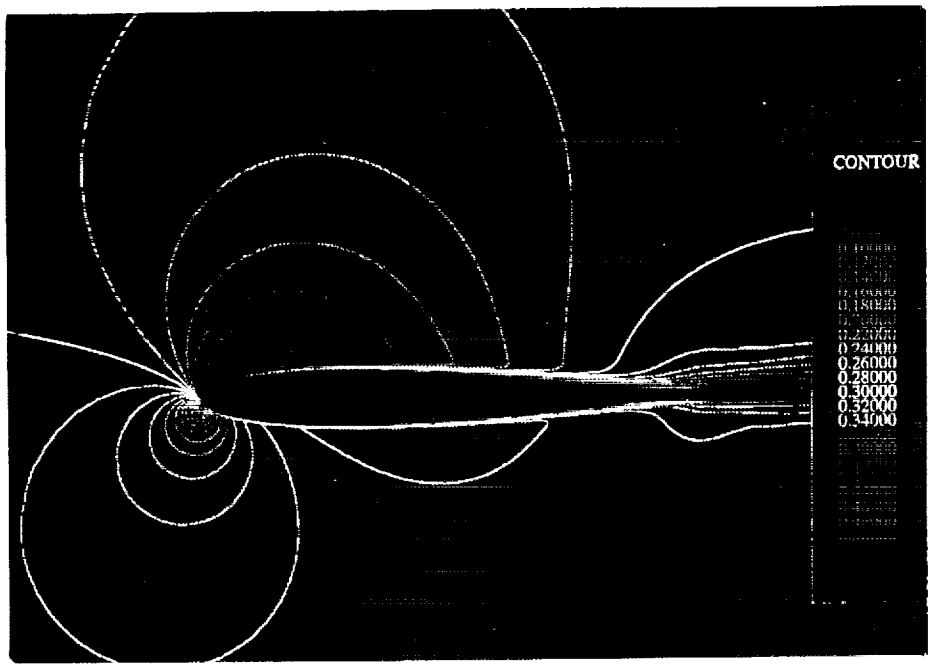


Figure 6.6: Velocity contours of an NACA-0012 airfoil at R_e =3.91x10^6 and α = 5^o from ZETA with Baldwin-Lomax turbulence model.

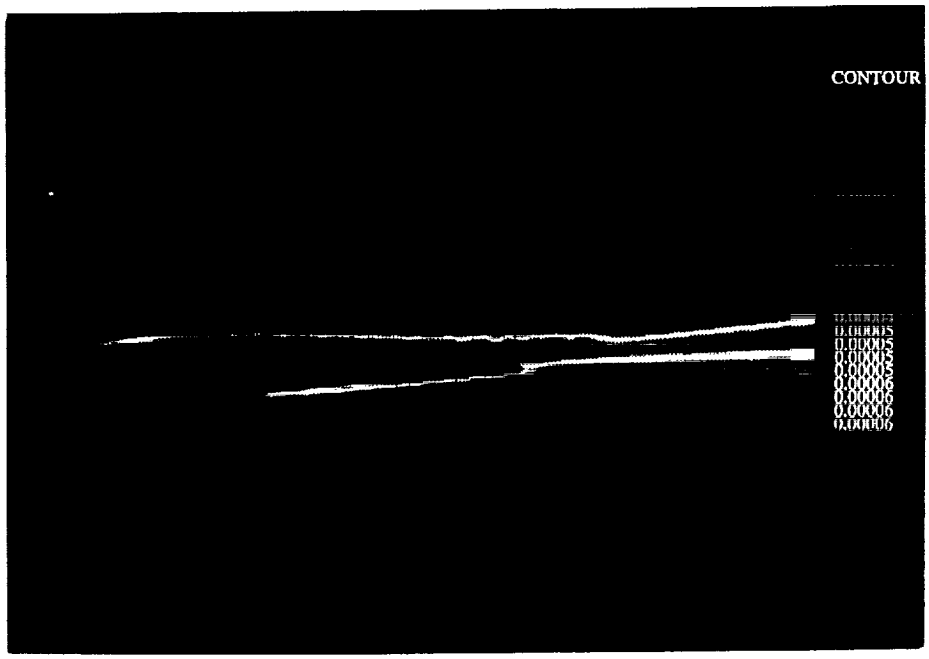


Figure 6.7: Eddy viscosity contours of an NACA-0012 airfoil at R_e =3.91x10^6 and α = 5^o from ZETA with Baldwin-Lomax turbulence model.

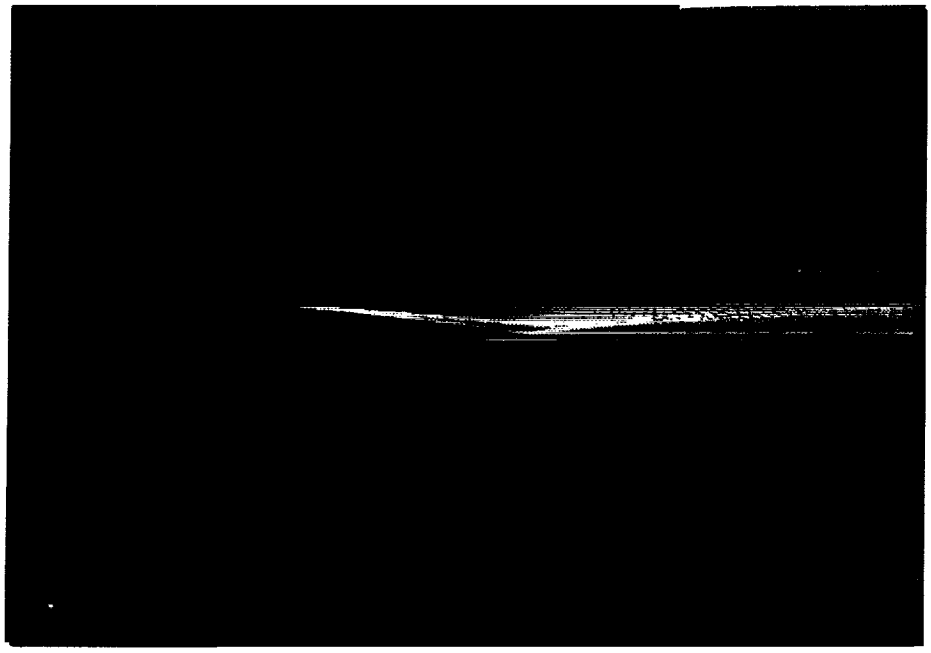


Figure 6.8: Eddy viscosity contours of an NACA-0012 airfoil at $R_e = 3.91 \times 10^6$ and $\alpha = 5^o$ from isolated Baldwin-Barth turbulence model test program.

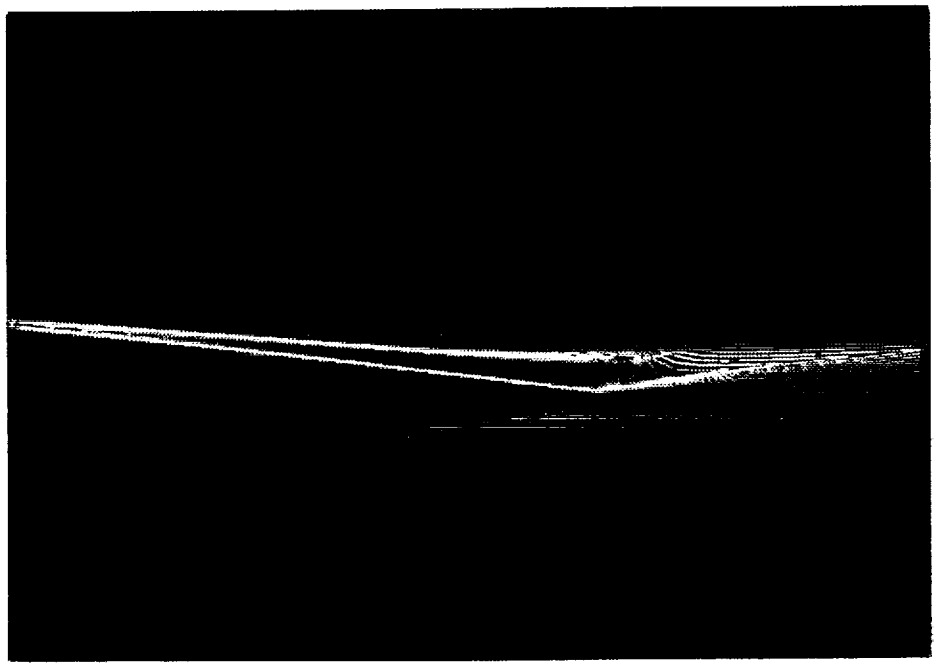


Figure 6.9: Trailing edge eddy viscosity contours of an NACA-0012 airfoil at $R_e = 3.91 \times 10^6$ and $\alpha = 5^o$ from ARC2D with Baldwin-Barth model.

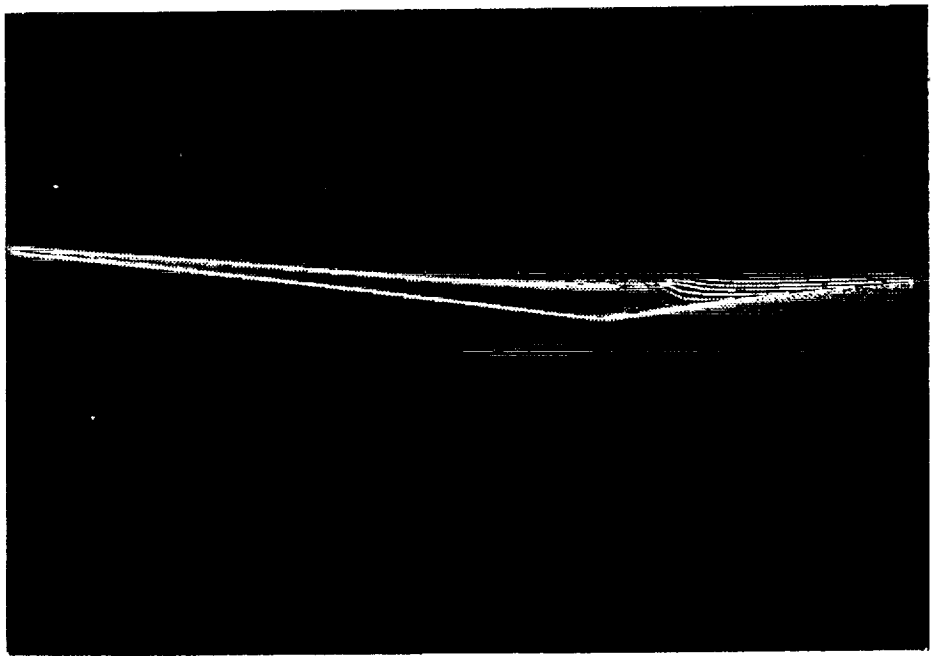


Figure 6.10: Trailing edge eddy viscosity contours of an NACA-0012 airfoil at $R_e = 3.91 \times 10^6$ and $\alpha = 5^o$ from the isolated Baldwin-Barth subroutine.

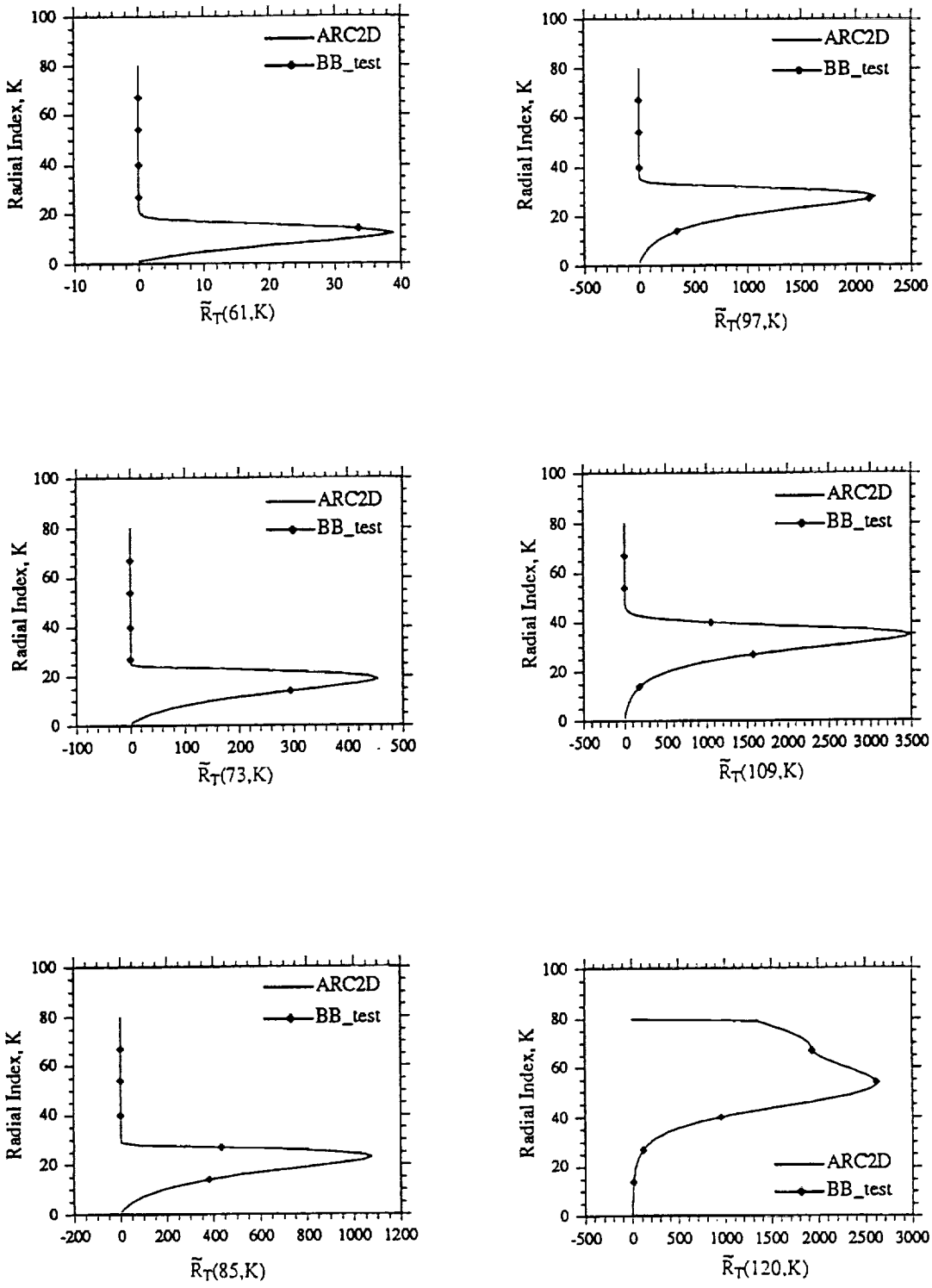


Figure 6.11: $\tilde{R}_T$ profiles at six locations on the top surface of the airfoil from ARC2D and the isolated Baldwin-Barth test cases.

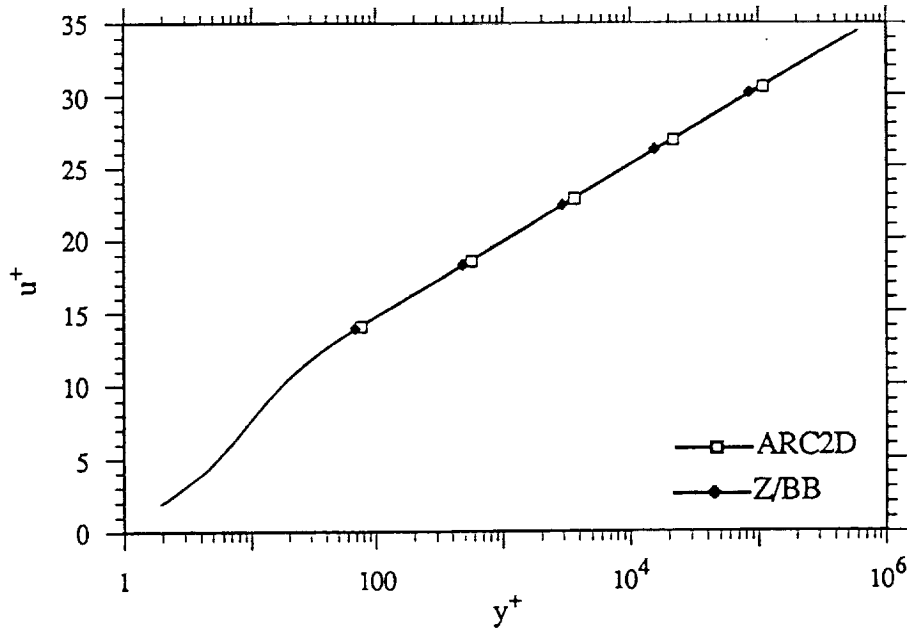


Figure 6.12: u^+ vs y^+ at mid-chord for both ZETA and ARC2D solutions.

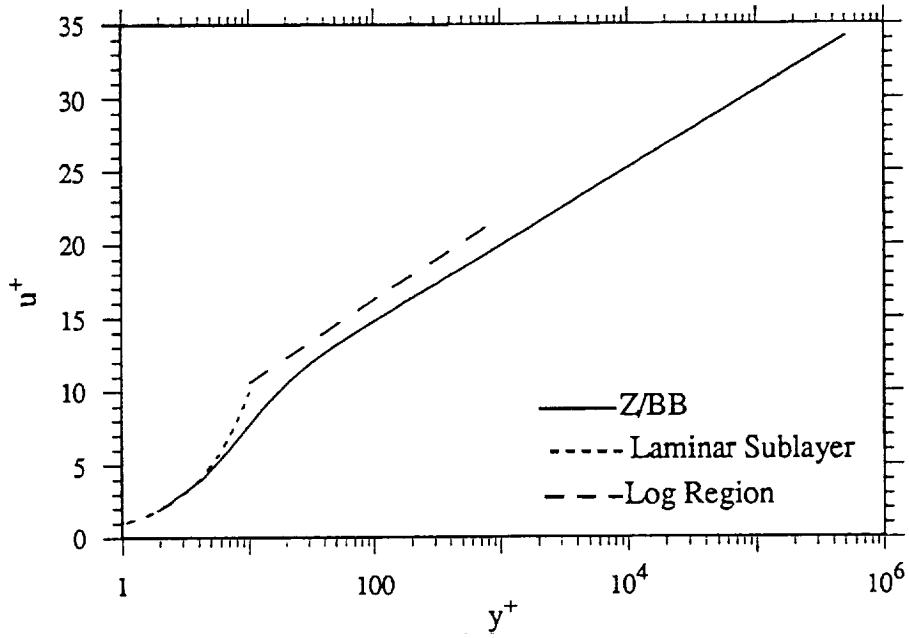


Figure 6.13: u^+ vs y^+ at mid-chord of an NACA-0012 airfoil and on a flat plate.

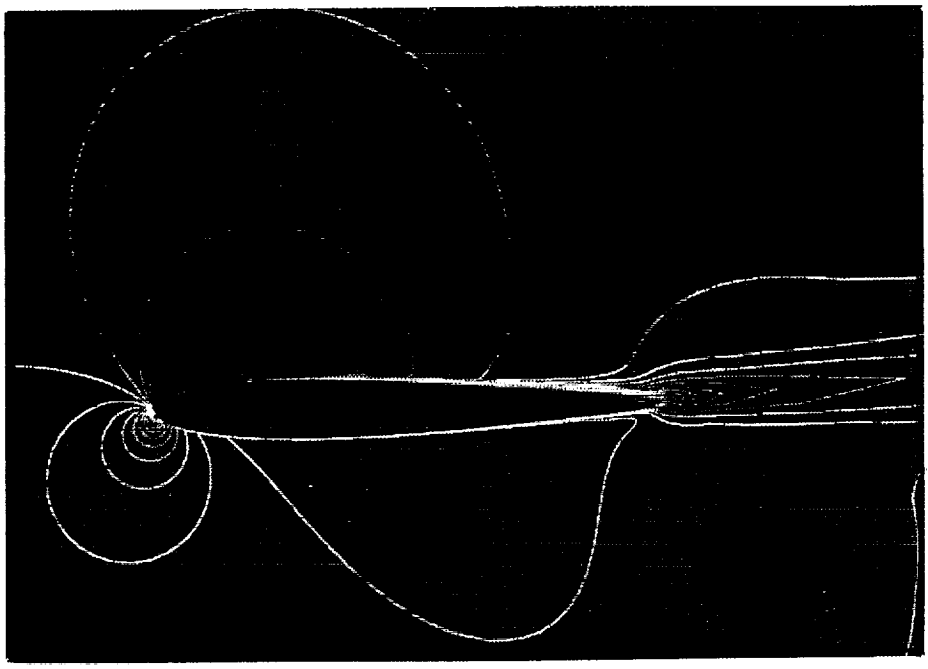


Figure 6.14: Velocity contours with first grid spacing, dy = 0.001.

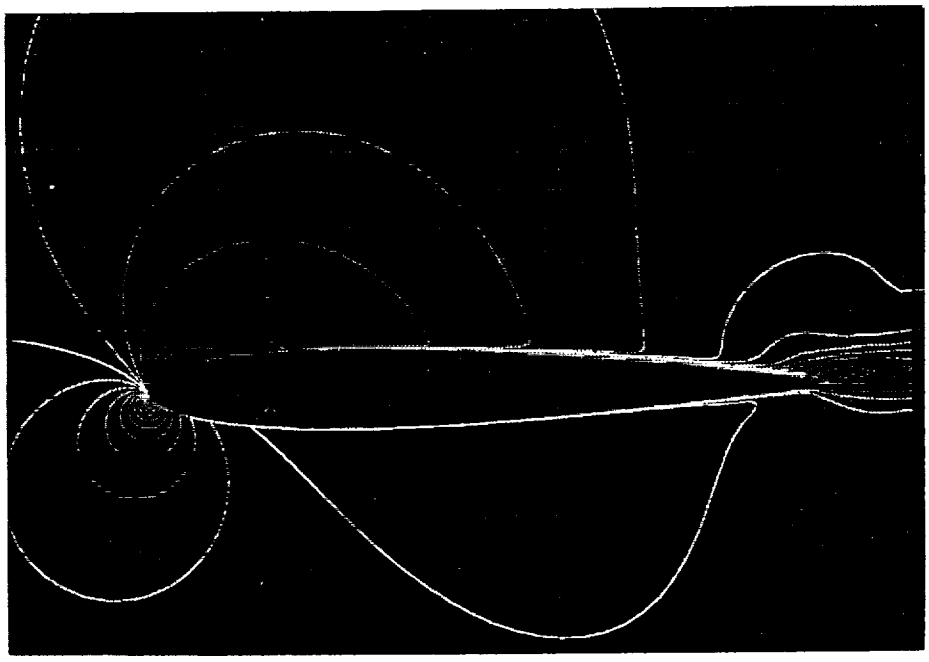


Figure 6.15: Velocity contours with first grid spacing, dy = 0.0001.

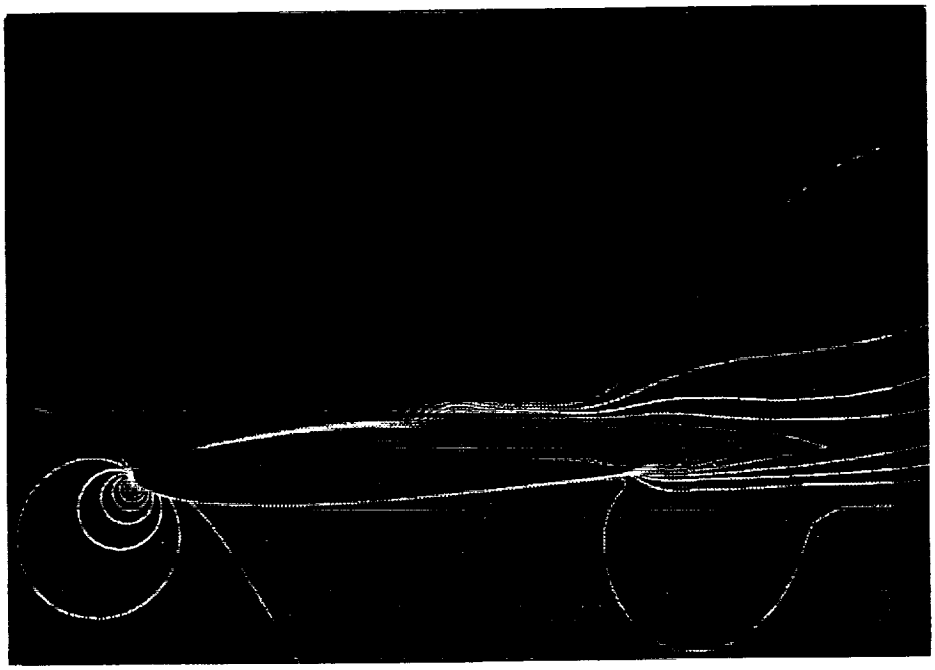


Figure 6.16: Velocity contours with first grid spacing, dy = 0.00006.

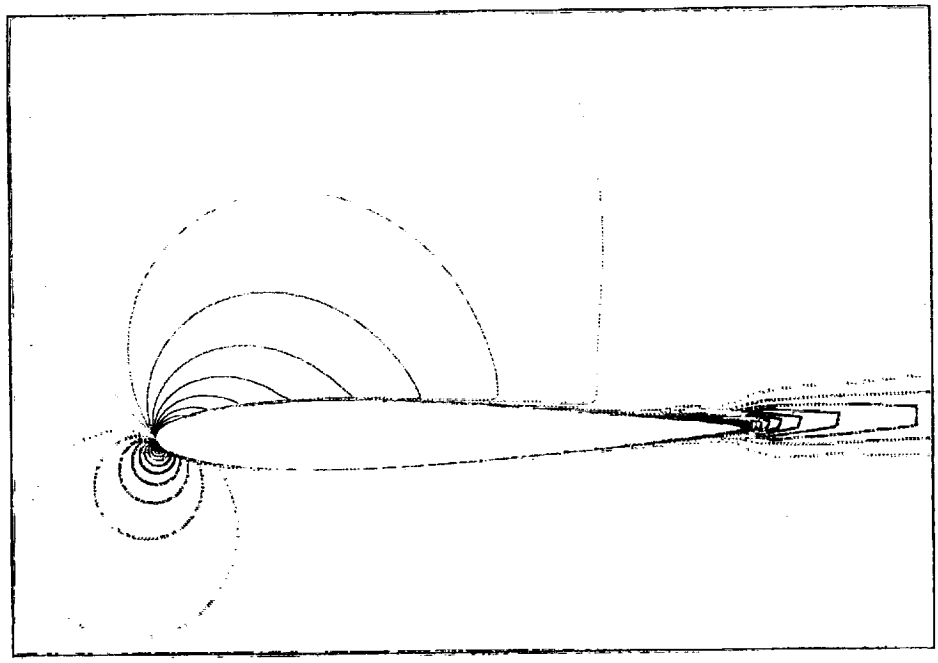


Figure 6.17: Velocity contours with 20% of computed eddy viscosity from the Baldwin-Barth turbulence model.

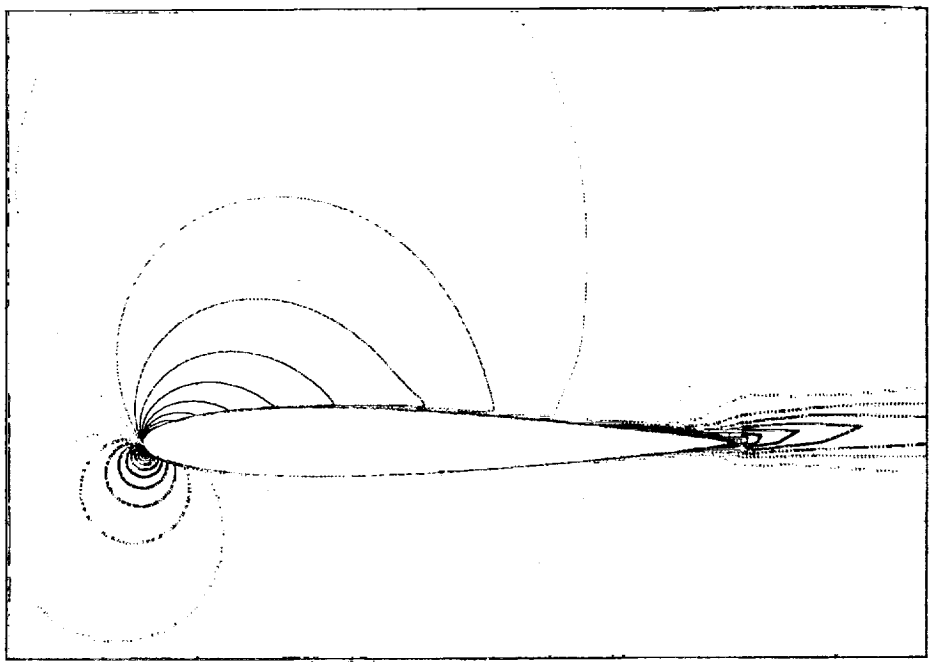


Figure 6.18: Velocity contours with 10% of computed eddy viscosity from the Baldwin-Barth turbulence model.

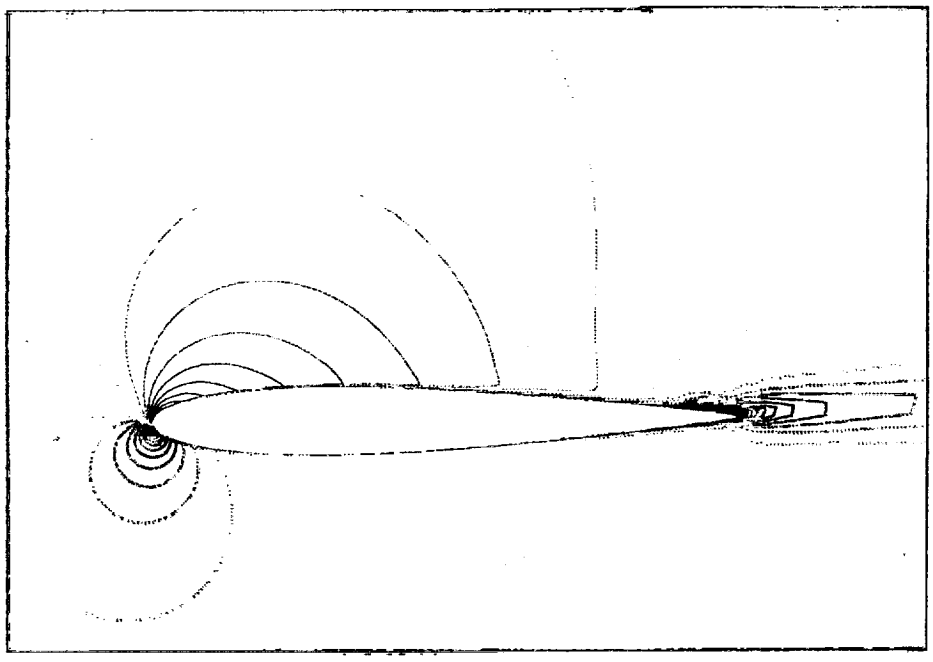


Figure 6.19: Velocity contours with 0% of computed eddy viscosity from the Baldwin-Barth turbulence model.

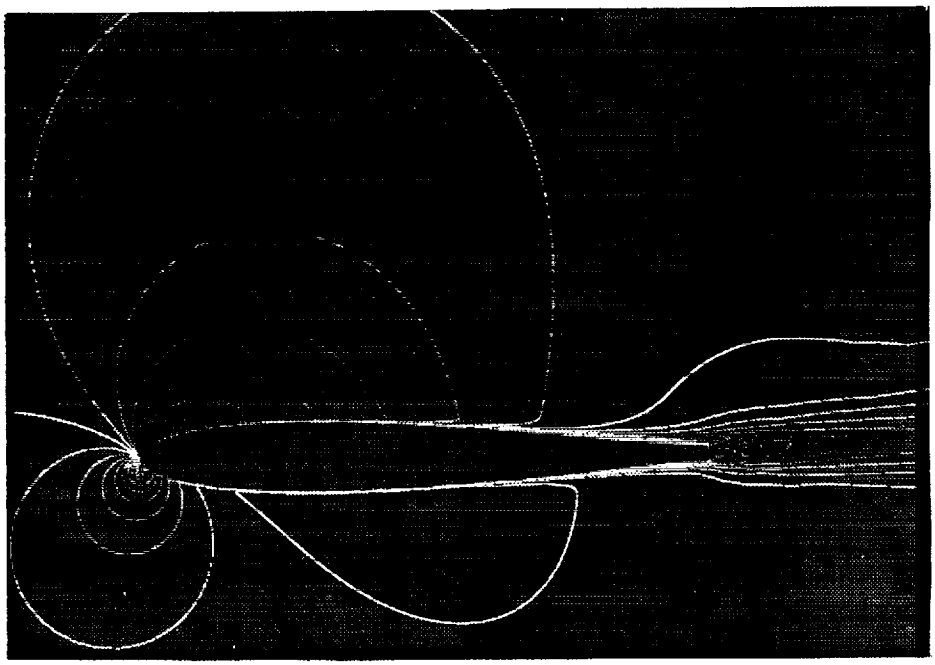


Figure 6.20: Velocity contours of an NACA-0012 airfoil at R_e =3.91x10^6 and $\alpha = 5^o$ from ZETA with Baldwin-Barth turbulence mode. dy = 0.01

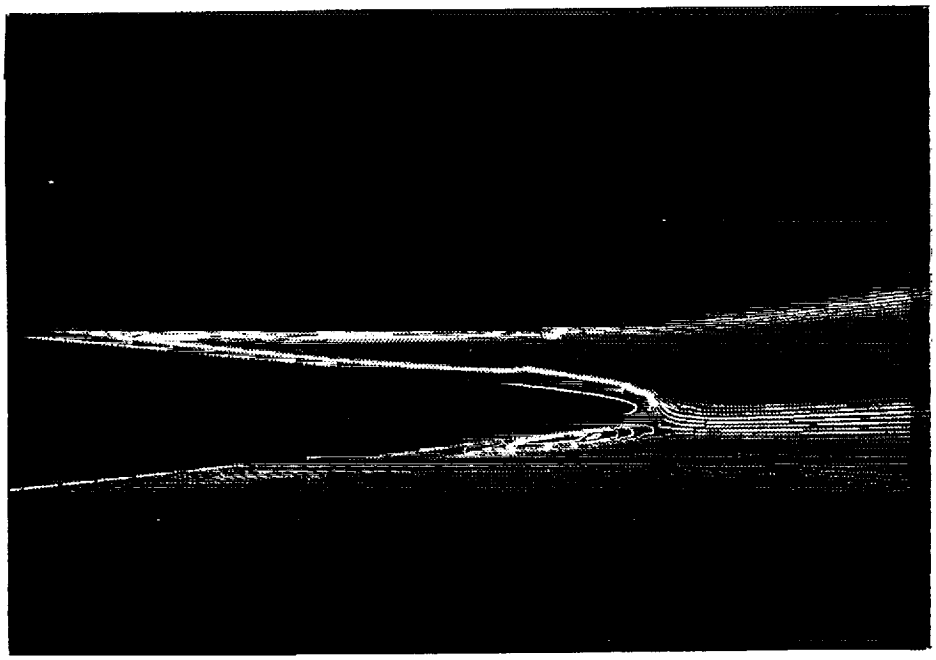


Figure 6.21: Eddy viscosity contours of a NACA0012 at R_e =3.91x10^6 and $\alpha = 5^o$ from ZETA with Baldwin-Barth turbulence model. dy = 0.01.

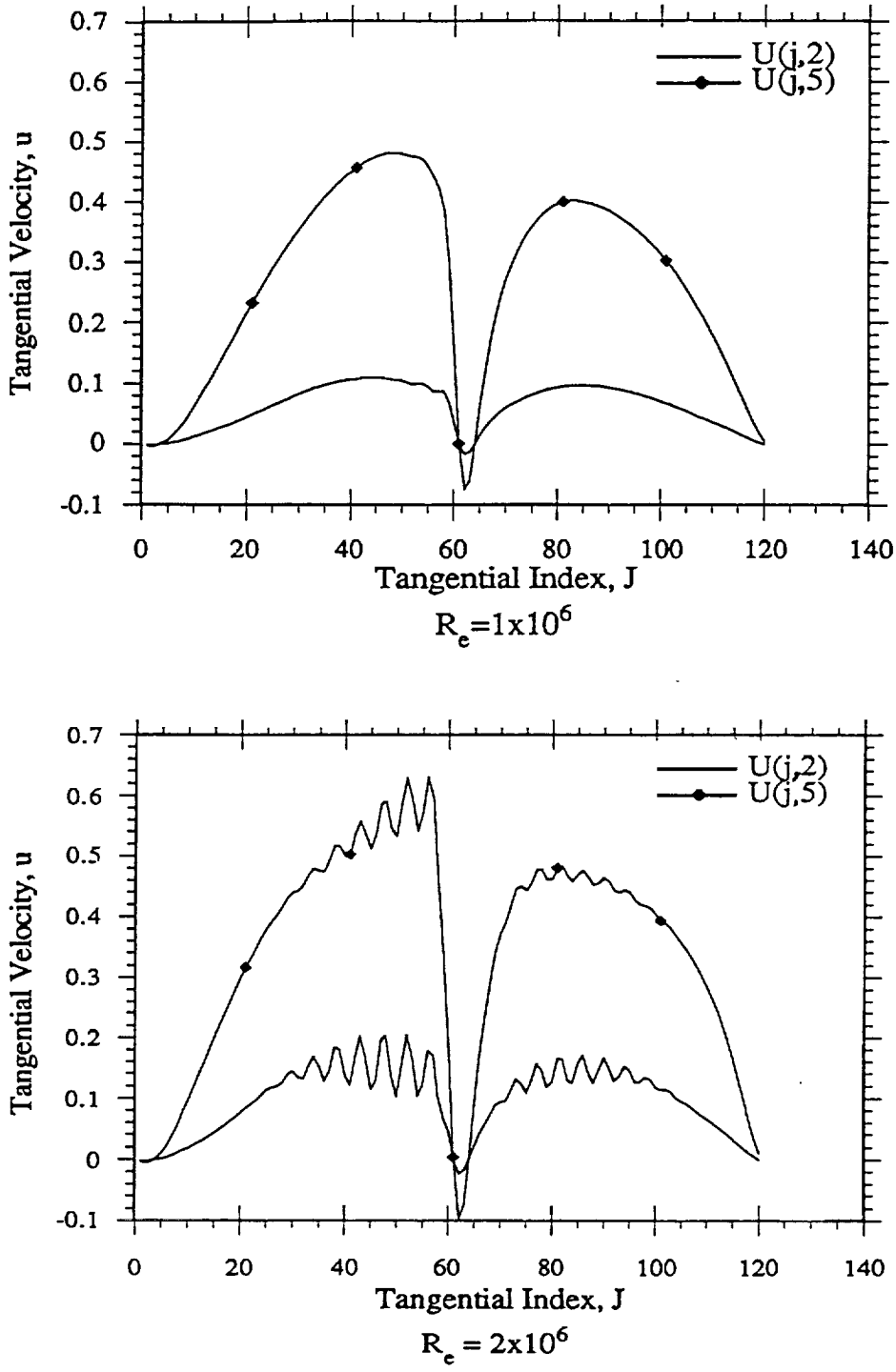


Figure 6.22: Surface velocities at R_e =1x10^6 and 2x10^6 and α = 5° from ZETA with Baldwin-Lomax turbulence model. dy = 0.0001.

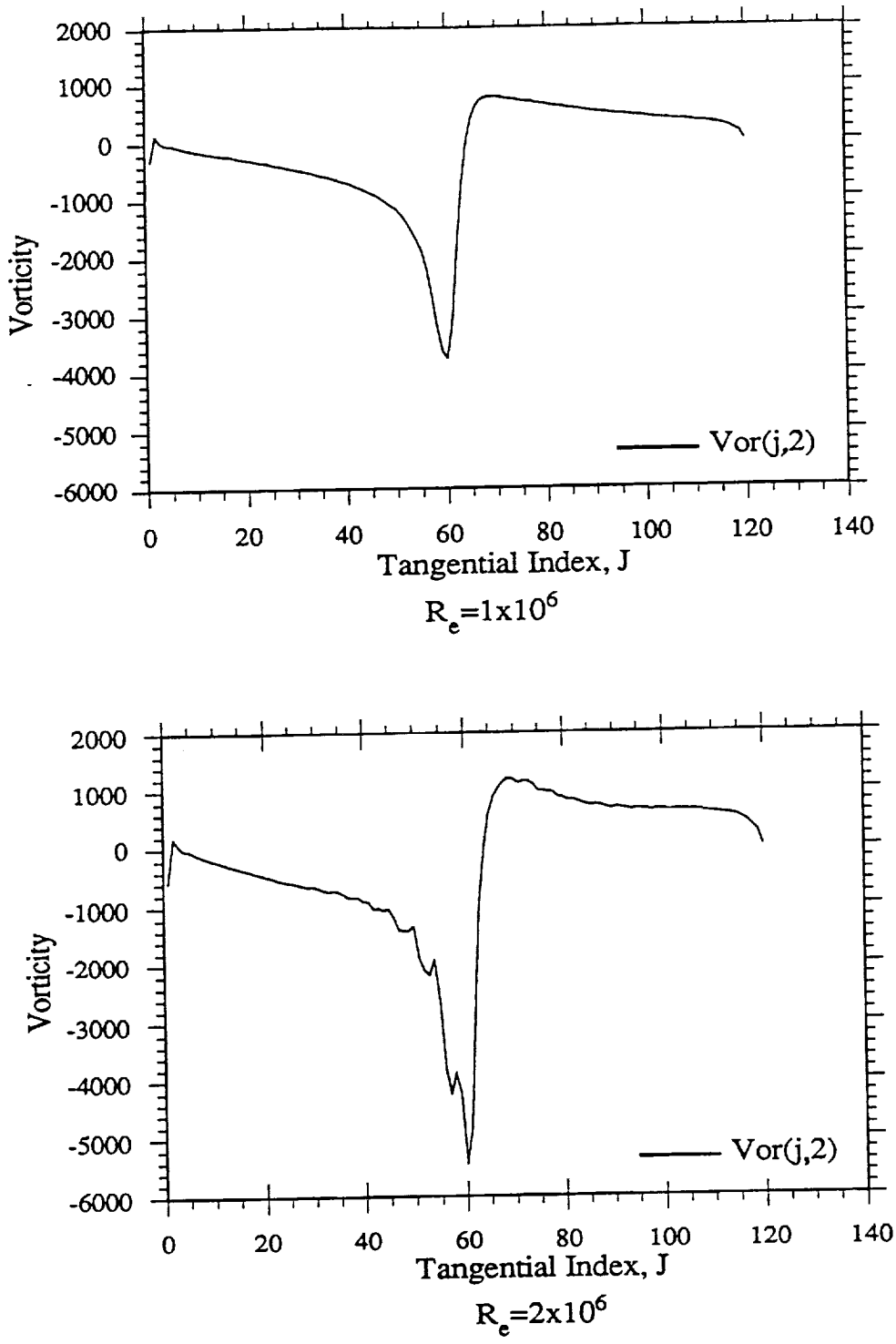


Figure 6.23: Surface vorticities at R_e =1x10^6 and 2x10^6 and α = 5° from ZETA with Baldwin-Lomax turbulence model. dy = 0.0001.

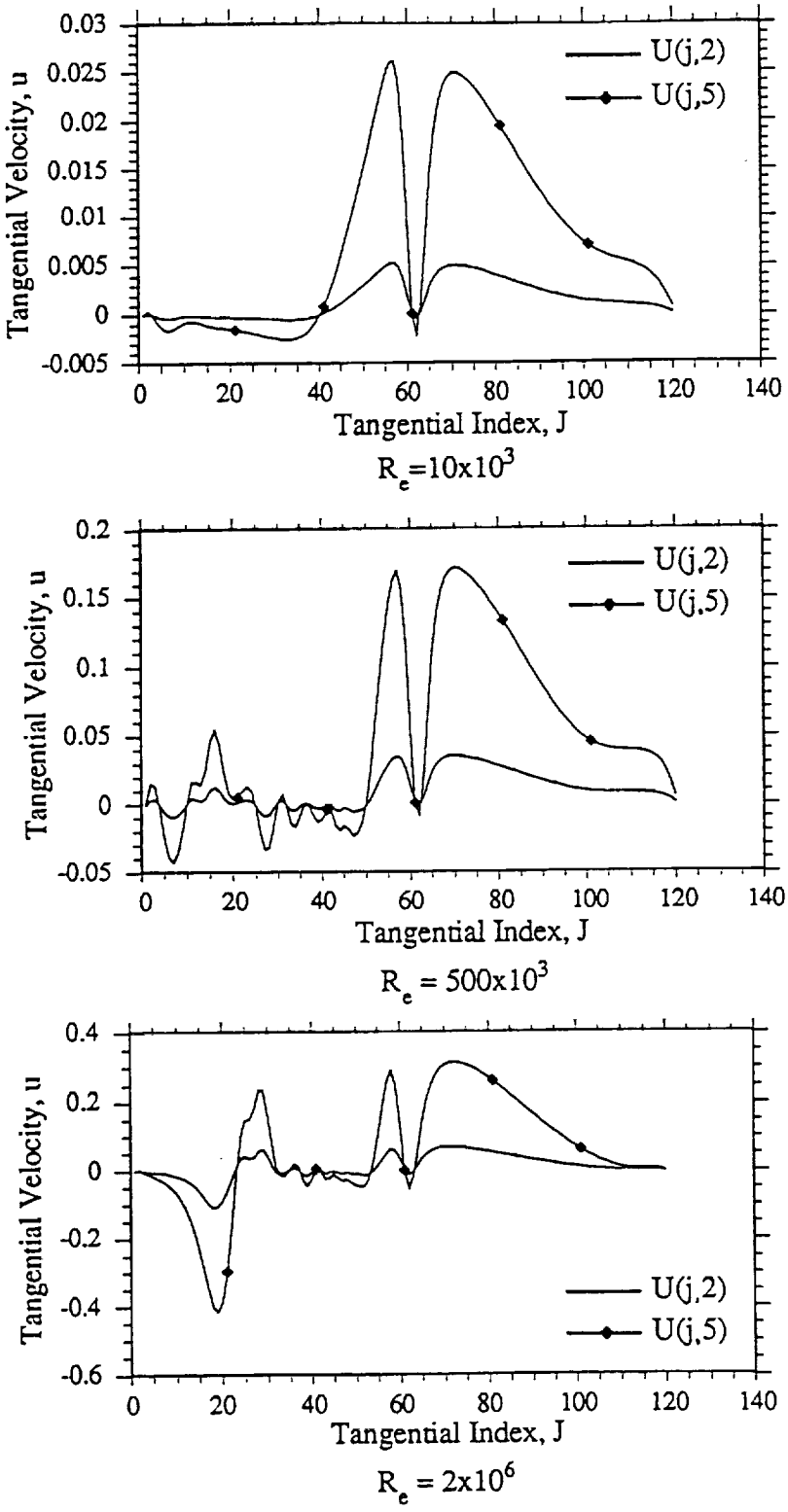


Figure 6.24: Surface velocity fluctuations of ZETA with Baldwin-Barth turbulence model at different R_e. dy = 0.0001

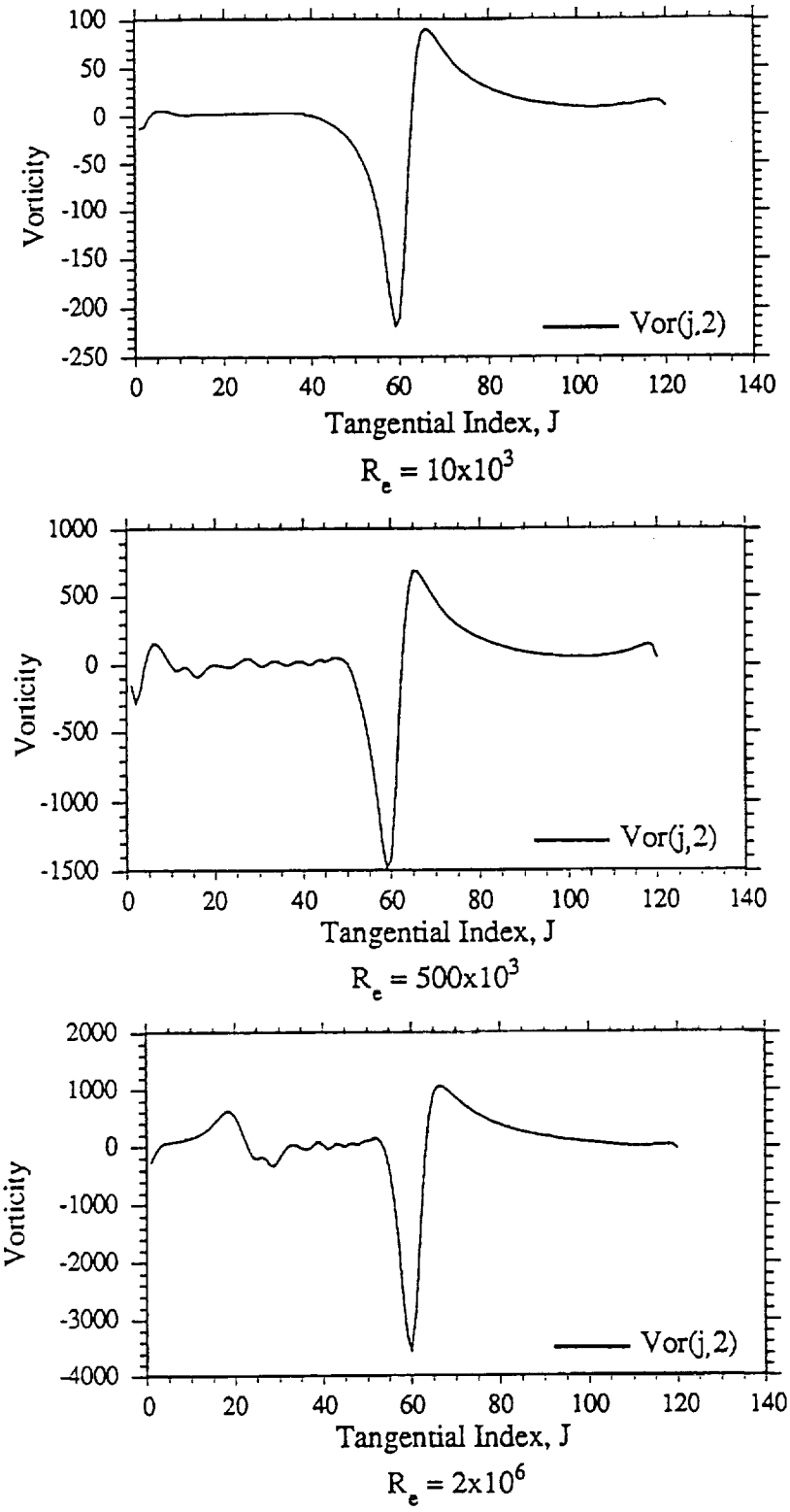


Figure 6.25: Surface vorticity fluctuations of ZETA with Baldwin-Barth turbulence model at different R_e. dy = 0.0001

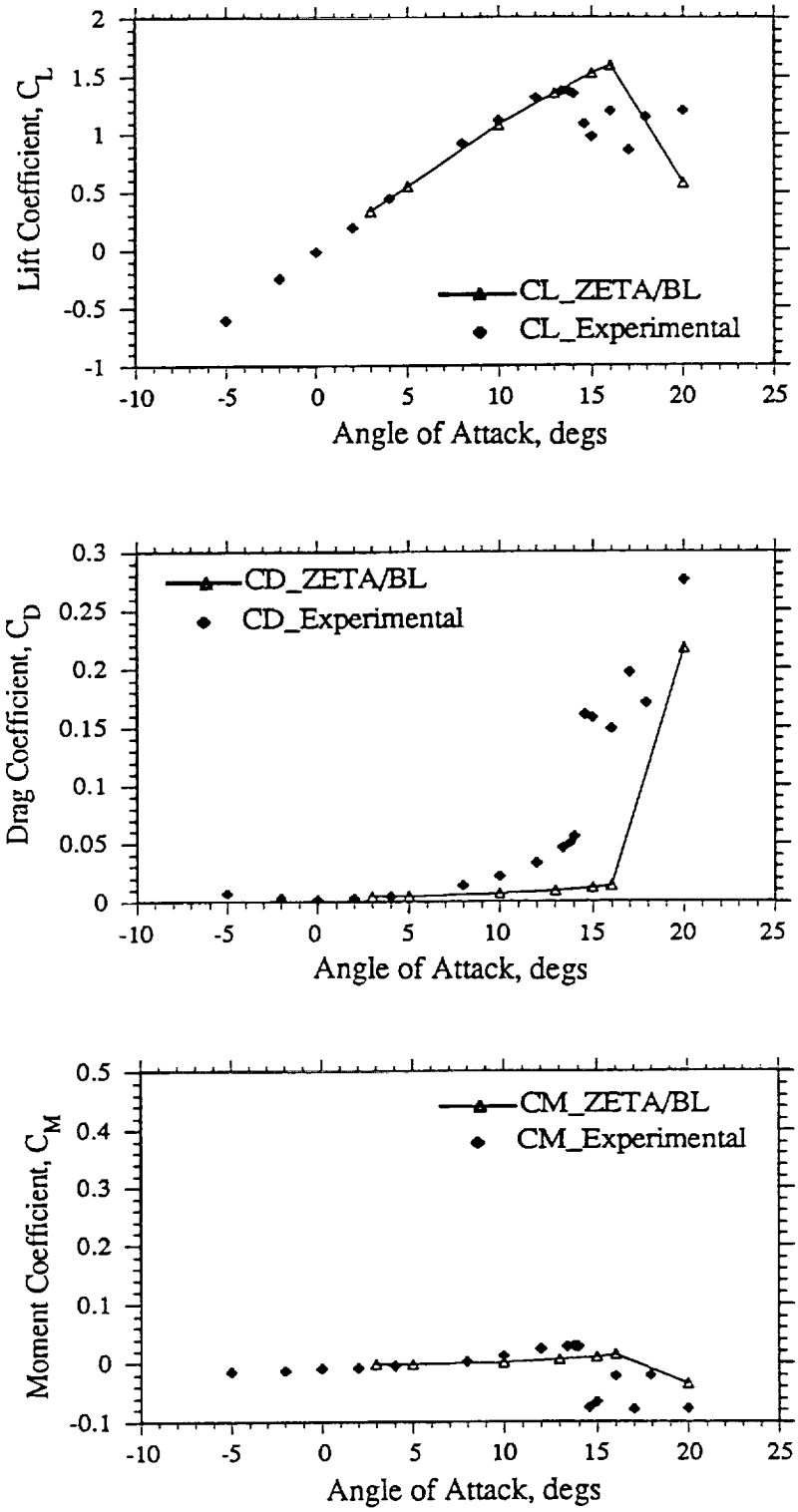


Figure 6.26: Computed and experimental aerodynamic loads of an NACA-0012 airfoil at $R_e = 3.91 \times 10^6$ and $\alpha = 5^o$.

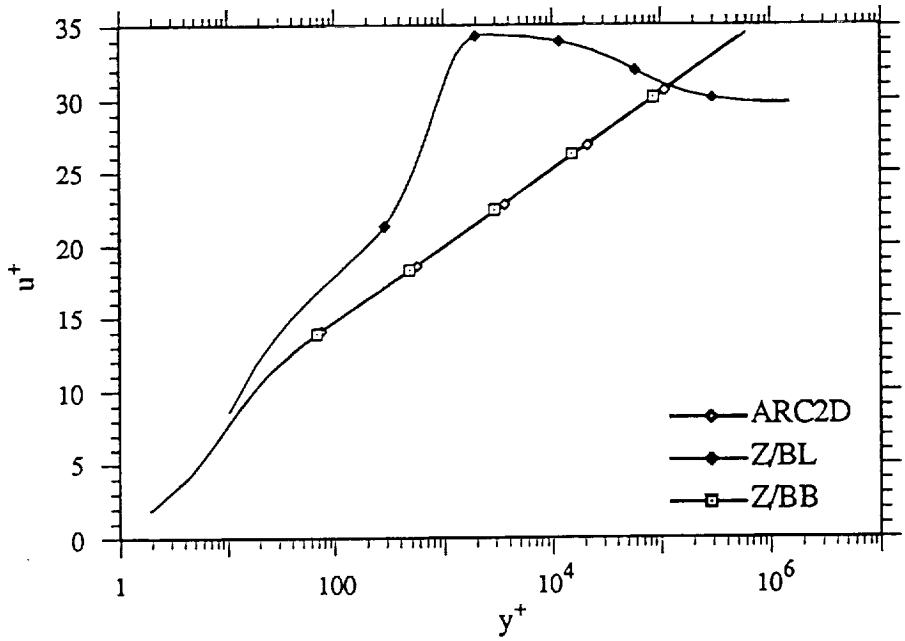


Figure 6.27: u^+ vs y^+ at mid-chord of ARC2D with Baldwin-Barth turbulence model and ZETA with Bladwin-Barth and Bladwin-Lomax turbulence models.

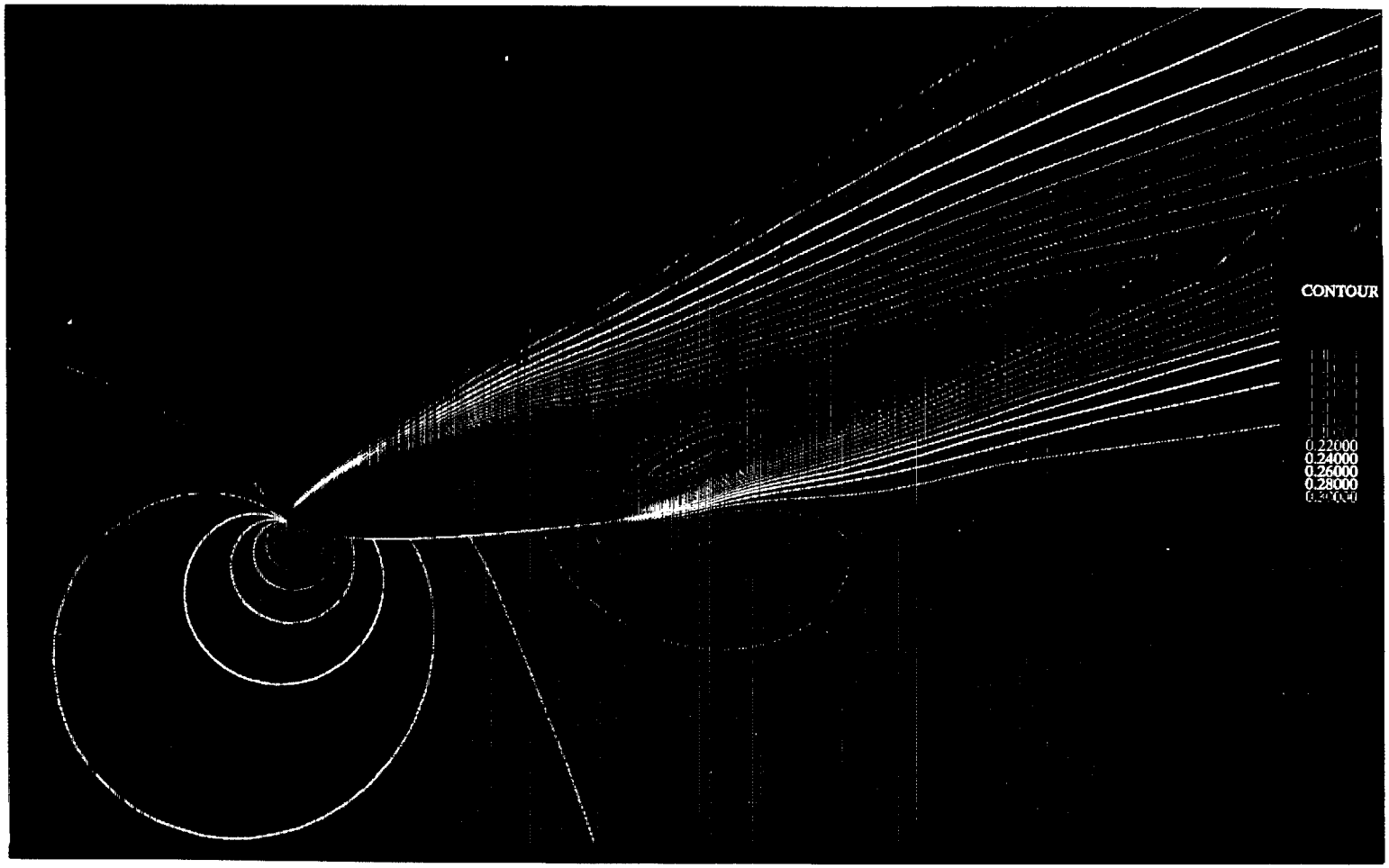


Figure 6.28: Velocity contours of an NACA-0012 airfoil at $R_e = 1x10^6$ and $\alpha = 20^o$ from ZETA with Baldwin-Barth turbulence model.

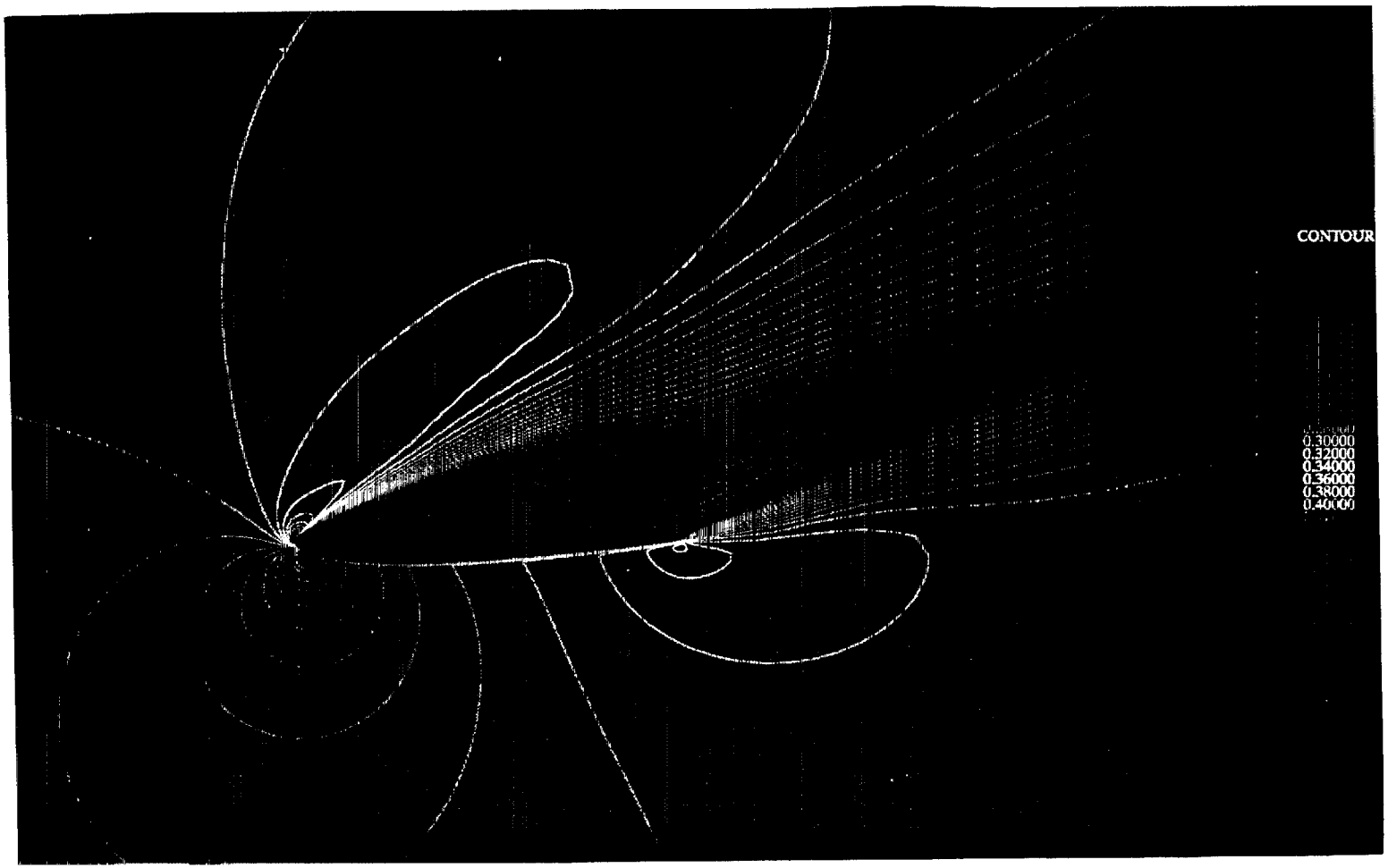


Figure 6.29: Velocity contours of an NACA-0012 airfoil at $R_e = 1x10^6$ and $\alpha = 20^o$ from ZETA with Baldwin-Lomax turbulence model.

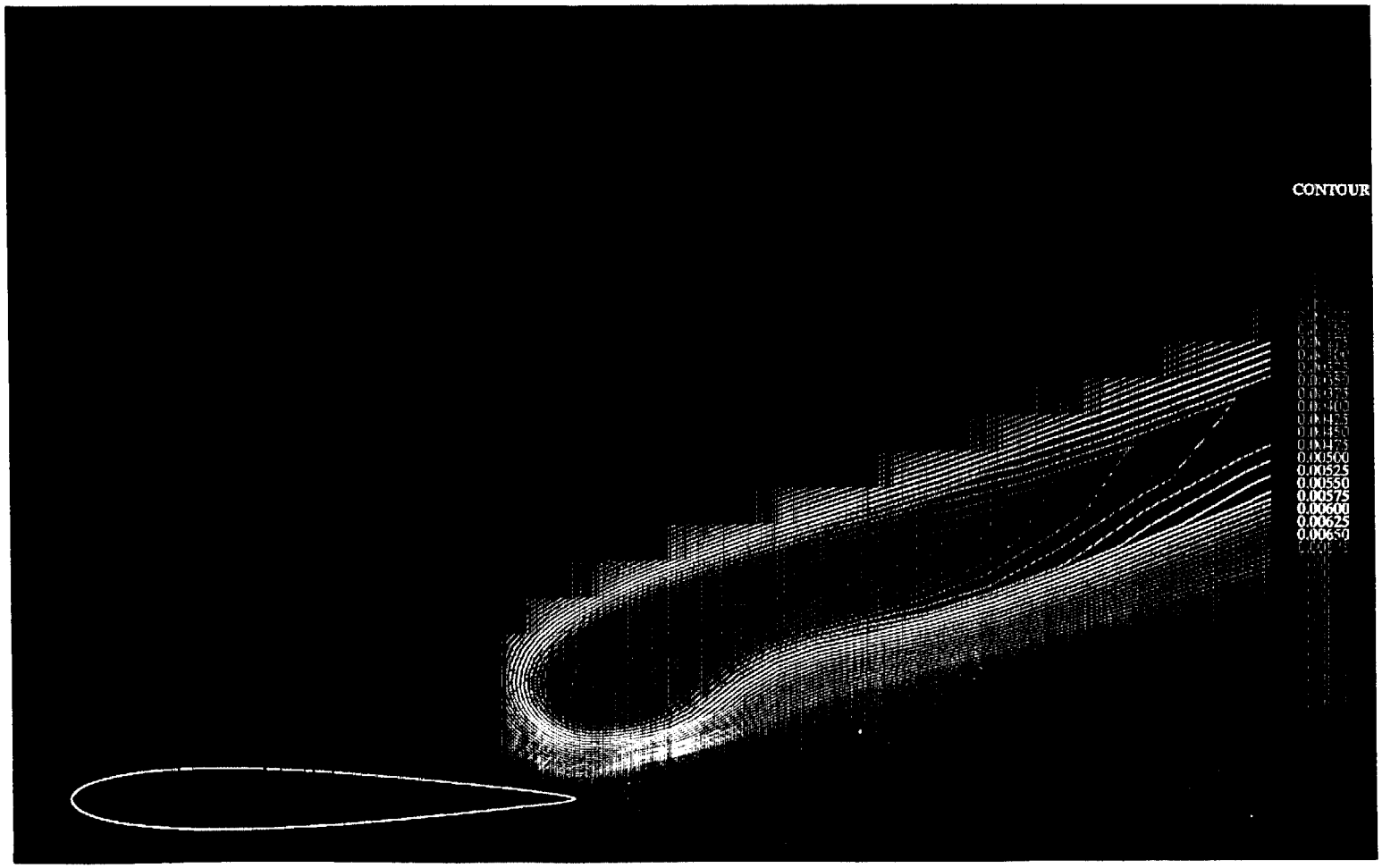


Figure 6.30: Eddy viscosity contours of an NACA-0012 airfoil at $R_e = 1 \times 10^6$ and $\alpha = 20^o$ from ZETA with Baldwin-Barth turbulence model.

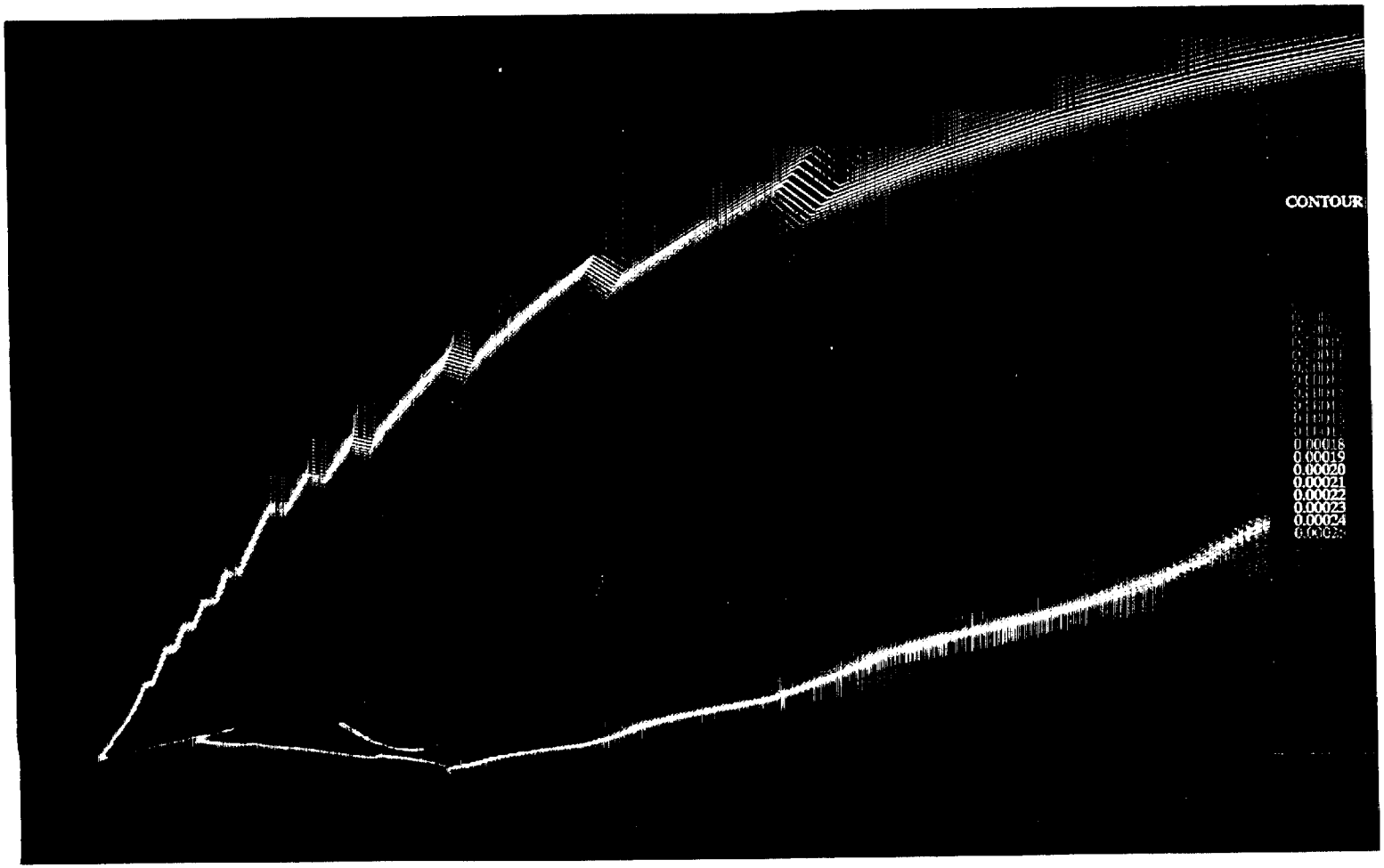


Figure 6.31: Eddy viscosity contours of an NACA-0012 airfoil at $R_e = 1 \times 10^6$ and $\alpha = 20^o$ from ZETA with Baldwin-Lomax turbulence model.

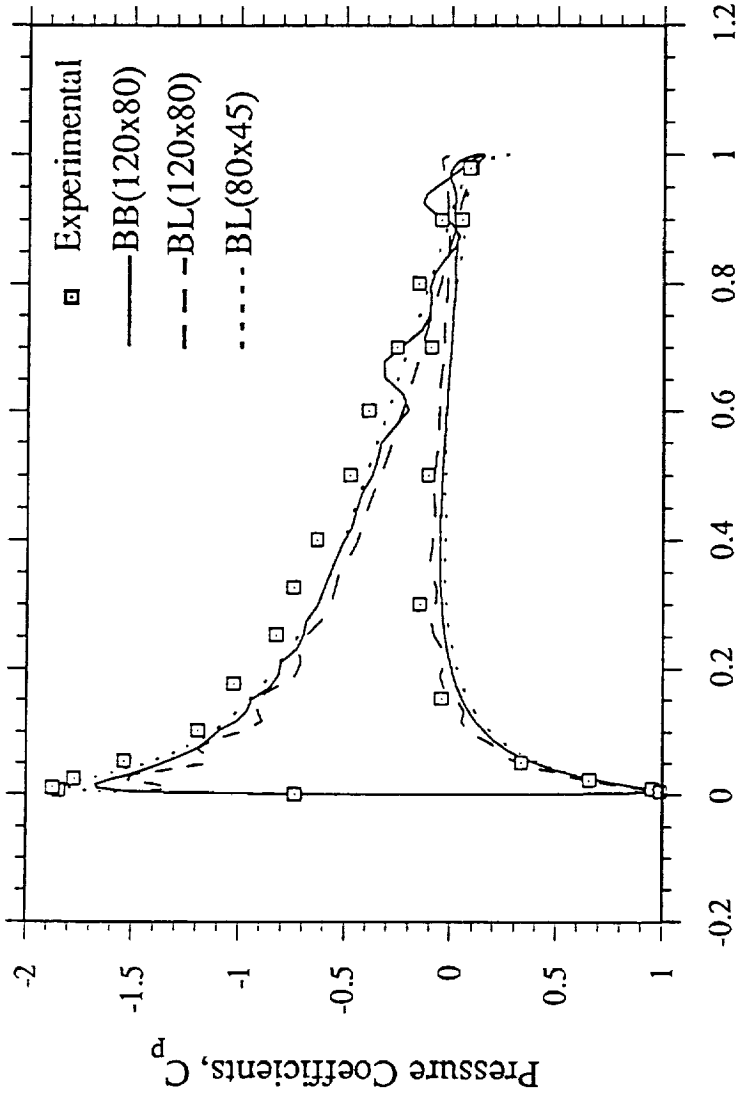


Figure 6.32: Computed and experimental pressure coefficients of an NACA-0012 airfoil at $R_e = 3.91 \times 10^6$ and $\alpha = 5^o$.

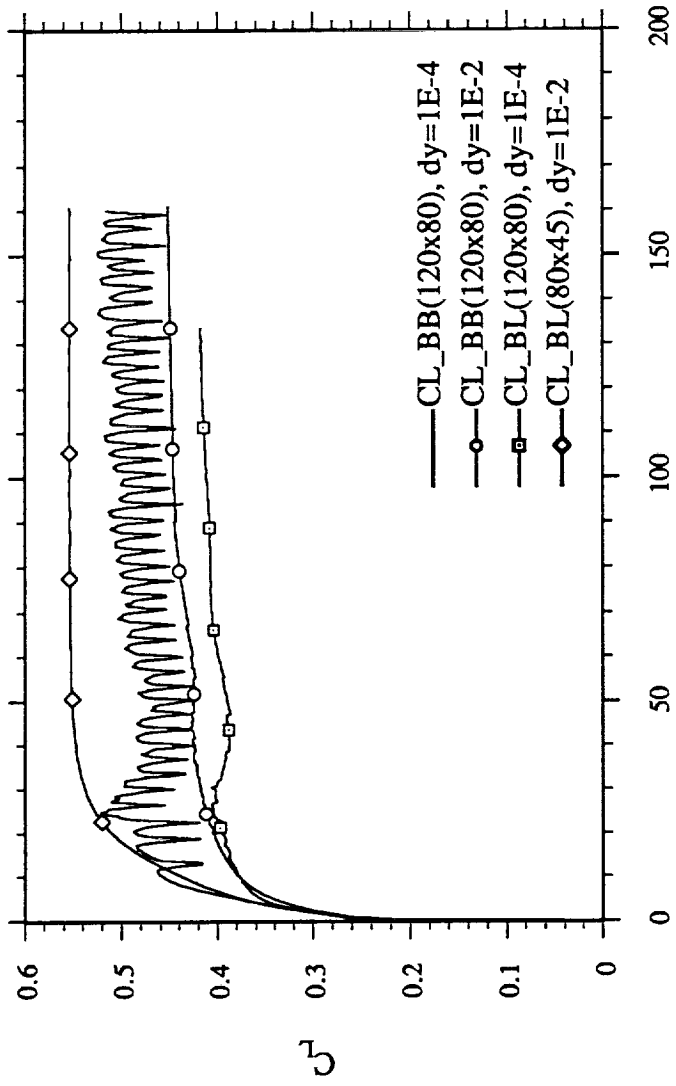


Figure 6.33: Examples of time convergence history of ZETA at $R_e = 3.91 \times 10^6$ and $\alpha = 5°$.

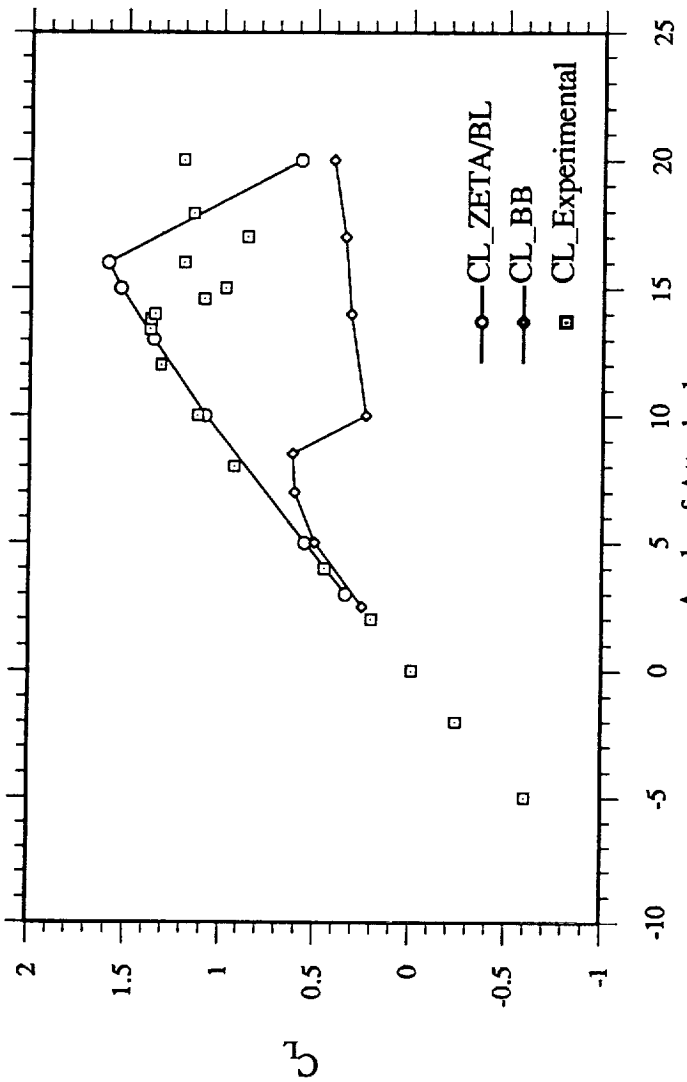


Figure 6.34: Computed and experimental C_L of an NACA-0012 airfoil at $R_e = 3.91 \times 10^6$ and $\alpha = 5^o$.

APPENDIX A

LISTING OF THE BALDWIN-BARTH SUBROUTINE

```
C***********************************************************************
C*************** TURBULENT VISCOSITY ********************************
C***********************************************************************
      SUBROUTINE BBEDDY

C     ONE EQUATION TURBULENCE MODEL
C      VERSION 1.0
C     BY BALDWIN AND BARTH  NASA AMES R.C.

C            ********* GENERAL NOTE ***********
C      CONSTANTS ARE GIVEN BELOW FOR THE MODEL THAT APPEARED IN
C      NASA TM-102847. THE SECOND VERSION THAT APPEARED IN
C      AIAA PAPER 91-0610 IS NOT USED.

      PARAMETER(JDIM=120,KDIM=80)
      PARAMETER(MAXJ=121,MAXK=81)

      COMMON /GRD/JMAX,IM2,KFC,KMAX,N
      COMMON /BB_1/TURMU(JDIM,KDIM),TURRE(JDIM,KDIM),RE,FIRST
      COMMON /BB_2/ SMIN(MAXJ,MAXK), BWT(MAXJ,MAXK),
     $              IB(MAXJ,MAXK),X(JDIM,KDIM),Y(JDIM,KDIM),
     $              XY(JDIM,KDIM,4),XYJ(JDIM,KDIM),
     $              AA(0:MAXJ,0:MAXK),AAN(MAXJ,MAXK)
      COMMON/BB_3/U(MAXJ,MAXK),V(MAXJ,MAXK),FSMACH,ALPI,T,AL
      COMMON/DELTA/DZ,DTET,DT
      COMMON /TUR1/USTAR(JDIM),EDDY(JDIM,KDIM),IOTUR(JDIM),
     $              YN(MAXJ,KDIM),IOT,IOTB
      COMMON /COE/AF,UI,VI,OMG,VSC,NPL,ICTUR,ICST,ICPL

      DIMENSION Q(JDIM,KDIM,4),VORT(JDIM,KDIM)
      DIMENSION PRESS(JDIM,KDIM)
      DIMENSION FNU(JDIM,KDIM),DS(JDIM,KDIM)

      DIMENSION UD(MAXJ,MAXK),VD(MAXJ,MAXK),QD(JDIM,KDIM),
      DIMENSION FND(KDIM),TEMPU(MAXJ,MAXK), TEMPV(MAXJ,MAXK),
     >          EMPS(MAXJ,MAXK), TEMPMU(JDIM,KDIM),
     >          GQ(0:JDIM,0:KDIM,2,2),
     >          GQN(JDIM,KDIM,2,2),
     >          DAMP1(JDIM,KDIM), DAMP1M(JDIM,KDIM),
     >          DAMP2(JDIM,KDIM), WORKX(MAXK,MAXJ),
     >          WORKY(MAXJ,MAXK)
      DIMENSION  AX(KDIM,JDIM),BX(KDIM,JDIM),CX(KDIM,JDIM),
     >          DX(KDIM,JDIM),EX(KDIM,JDIM),FX(KDIM,JDIM),
     >          AY(JDIM,KDIM),BY(JDIM,KDIM),CY(JDIM,KDIM),
     >          DY(JDIM,KDIM),EY(JDIM,KDIM),FY(JDIM,KDIM)

      LOGICAL FIRST

      DATA FIRST /.TRUE. /
```

```
C       THE FOLLOWING TWO VARIABLES ARE FOR 'O' GRID.
C       THEY ARE 2 AND JMAX-1 FOR 'C' GRID.

        JTAIL1 = 1
        JTAIL2 = JMAX

C       THESE ARE THE CONSTANTS FOR THE MODEL APPEARING IN NASA
C       TM-102847.

        AKARMAN = .41
        CMU  = .09
        C1E  = 1.20
        C2E  = 2.00
        SIGE = AKARMAN**2/((C2E-C1E)*SQRT(CMU))
        APLUS1 = 26.
        APLUS2 = 10.
        RETINF = 1.E-12

C       DTM IN THE MODEL IS EQUAL TO DT OF THE SOLVER AND  NNIT IS
C       ONE FOR TIME ACCURATE SOLUTION.

        NNIT   = 1
        DTM    = DT

C       FOR STEADY-STATE CALCULATIONS WE USUALLY TAKE A TIME STEP
C       IN THE TURBULENCE MODEL WHICH IS LARGER THAN THE FLOW
C       SOLVER(BUT NOT TOO LARGE).

C       IF(ICST .EQ. 0 .OR. ICST .EQ. 1)DTM  = 200.*DT
C       IF(DTM.GT.50)DTM = 50.

C       **THESE CONSTANTS FOR THE SUTHERLAND'S LAW ARE NOT USED
C       IN THE INCOMPRESSIBLE FLOW
C       C2B=198.6/TINF
C       C2BP = C2B + 1.

        IF(FIRST)THEN

        FIRST = .FALSE.
        NNIT  = 1
        WRITE(6,*)' KARMAN CONSTANT = ',AKARMAN
        WRITE(6,*)' CMU             = ',CMU
        WRITE(6,*)' C1E             = ',C1E
        WRITE(6,*)' C2E             = ',C2E
        WRITE(6,*)' SIGMA E         = ',SIGE
        WRITE(6,*)' APLUS1          = ',APLUS1
        WRITE(6,*)' APLUS2          = ',APLUS2
        WRITE(6,*)' RE_T INF        = ',RETINF
        WRITE(6,*)' RE              = ',RE
        WRITE(6,*)' JDIM,KDIM       = ',JDIM,KDIM
        WRITE(6,*)' JMAX,KMAX       = ',JMAX,KMAX
        WRITE(6,*)' MAXJ,MAXK       = ',MAXJ,MAXK
        WRITE(6,*)' JTAIL1,JTAIL2   = ',JTAIL1,JTAIL2
        WRITE(6,*)' DTM             = ',DT

        IF(JTAIL2 .EQ. JMAX)WRITE(6,*)' JTAIL2 = JMAX '
```

```
C       COMPUTE METRICS

        CALL XYMETS(JMAX,KMAX,X,Y,XY,XYJ)
        DO J=1,JMAX
        DO K=1,KMAX
C       ****FOR TIME ACCURATE SOLUTION****
        DS(J,K) = 1.
C       ****FOR NON-TIME ACCURATE SOLUTION****
C       DS(J,K) = 1./(1. + SQRT(XYJ(J,K)))
        ENDDO
        ENDDO

C   COMPUTE GENERALIZED DISTANCE FUNCTION WHICH IS
C   MINIMUM DISTANCE TO WALL.
C   SAME DISTANCE FUNCTION APPEARS IN AIAA PAPER 91-0721

        DO J=1,JMAX
          DO K=1,KMAX
            DISTMS = 10000.
            DISTML = 10000.
            SMIN(J,K) = 10000.
            IB(J,K)  = 0
            DO JB=JTAIL1,JTAIL2-1
                JJ1 = JB
                JJ2 = JB + 1
                DXA = X(JJ2,1) - X(JJ1,1)
                DYA = Y(JJ2,1) - Y(JJ1,1)
                SNX = -DYA
                SNY = DXA
                DXX1 = X(J,K)-X(JB,1)
                DYY1 = Y(J,K)-Y(JB,1)
                DXX2 = X(J,K)-X(JB+1,1)
                DYY2 = Y(J,K)-Y(JB+1,1)
                DXX  = .5*(DXX1+DXX2)
                DYY  = .5*(DYY1+DYY2)
                DIST1 = (DXX1**2+DYY1**2)
                DIST2 = (DXX2**2+DYY2**2)
                DISTS = MIN(DIST1,DIST2)
                DISTL = MAX(DIST1,DIST2)
                DOT   = SNX*DXX + SNY*DYY
                IF((DISTS.LE.DISTMS).AND.(DOT.GT.-1.E-10))THEN
                DISTMS = DISTS
                DISTML = DISTL
                SS
     >              = (DXA*(X(J,K)-X(JJ1,1))
     >              + DYA*(Y(J,K)-Y(JJ1,1)))/(DXA**2+DYA**2)
                SS = MAX(0.0,MIN(SS,1.0))
                XPT = X(JB,1) + SS*(X(JB+1,1)-X(JB,1))
                YPT = Y(JB,1) + SS*(Y(JB+1,1)-Y(JB,1))
                SSMIN = SQRT((X(J,K)-XPT)**2+(Y(J,K)-YPT)**2)
                IF(SSMIN .LT. SMIN(J,K))THEN
                  SMIN(J,K) = SSMIN
                  BWT(J,K) = SS
                  IB(J,K) = JB
                  DISTMS = DISTS
                  DISTML = DISTL
                ENDIF
              ENDIF
```

```
      ENDDO
    ENDDO
  ENDDO

C     COMPUTE WEIGHING FACTORS FOR VELOCITY GRADIENT
C     CALCULATION.

      DO K=1,KMAX
        DO J=1,JMAX
          AA(J,K) = 0.0
          AAN(J,K) = 0.0
        ENDDO
      ENDDO

      DO J=1,JMAX
        DO K=1,KMAX-1
          DXA = X(J,K+1)-X(J,K)
          DYA = Y(J,K+1)-Y(J,K)
          SNX = DYA
          SNY = -DXA
              JM1 = J-1
              IF (J .EQ. 1) JM1=JMAX
          AA(JM1,K) = AA(JM1,K) + SNX*.5*(X(J,K+1)+X(J,K))
          AA(J,K) = AA(J,K) - SNX*.5*(X(J,K+1)+X(J,K))
        ENDDO
      ENDDO

      DO J=1,JMAX
              JP1=J+1
              IF(J .EQ. JMAX) JP1 = 1
        DO K=1,KMAX
          DXA = X(JP1,K)-X(J,K)
          DYA = Y(JP1,K)-Y(J,K)
          SNX = DYA
          SNY = -DXA
          AA(J,K-1) = AA(J,K-1) - SNX*.5*(X(JP1,K)+X(J,K))
          AA(J,K) = AA(J,K)  + SNX*.5*(X(JP1,K)+X(J,K))
        ENDDO
      ENDDO
      DO J=1,JMAX
              JP1=J+1
              IF(J .EQ. JMAX) JP1 = 1
        DO K=1,KMAX-1
          AAN(J,K) = AAN(J,K) + AA(J,K)
          AAN(JP1,K) = AAN(JP1,K) + AA(J,K)
          AAN(J,K+1) = AAN(J,K+1) + AA(J,K)
          AAN(JP1,K+1) = AAN(JP1,K+1) + AA(J,K)
        ENDDO
      ENDDO

C     IF(JTAIL2 .NE. JMAX)THEN
C       K = 1
C       DO J=1,JTAIL1
C         JJ = JMAX - J + 1
C         TEMP1 = AAN(J,K)
C         TEMP2 = AAN(JJ,K)
C         AAN(J,K) = TEMP1 + TEMP2
C         AAN(JJ,K)= TEMP2 + TEMP1
```

```
C          ENDDO
C        ENDIF

         IF(ICST .EQ. 0)THEN
                PRINT*,'********INITIAL RUN********'
           DO K=1,KMAX
             DO J=1,JMAX
               TURRE(J,K) =  RETINF
             ENDDO
           ENDDO
         ELSE
         NNIT = 1
         PRINT*,'*******RESTART********'
           OPEN(16)
           READ(16,*) JA,KA
           READ(16,*) ((TURRE(J,K), J=1,JA), K=1,KA)
           REWIND 16
         ENDIF
           WRITE(*,*)' FINISHED TURB MODEL INIT '
         ENDIF

C        DO J=JTAIL1,JTAIL2
C          TURRE(J,1) = 0.0
C        ENDDO

         JX=JMAX+1
         DO K=1,KMAX
         SMIN(JX,K)=SMIN(1,K)
         ENDDO

C        *********CHANGE J DIRECTION FOR VELOCITIES*********
         DO K = 1, KMAX
           DO J = 2, JMAX
             TEMPU(J,K) = U(J,K)
             TEMPV(J,K) = V(J,K)
           ENDDO
         ENDDO
C
         DO K = 1, KMAX
           DO J = 2, JMAX
             UD(J,K) = TEMPU(JMAX+2-J,K)
              VD(J,K) = TEMPV(JMAX+2-J,K)
           ENDDO
           UD(1,K) = U(1,K)
           VD(1,K) = V(1,K)
         ENDDO
C
C        NOW THE MODEL
C        COMPUTE THE VELOCITY GRADIENT VECTOR

         DO K=1,KMAX
           DO J=1,JMAX
             Q(J,K,1) = 1.
             Q(J,K,2) = UD(J,K)
             Q(J,K,3) = VD(J,K)
             FNU(J,K) = 1.
           ENDDO
         ENDDO
```

```
C       THIS ROUTINE ACTUALLY COMPUTES THE GRADIENT OF THE MEAN
C       VELOCITY

        CALL GRADV(MAXJ,MAXK,JDIM,KDIM,JMAX,KMAX,JTAIL1,JTAIL2,
     >                   X,Y,AA,AAN,Q,GQ,GQN)

C       GENERAL NOTE: THE EDDY VISCOSITY IS SUPPLIED AT HALF POINTS
C       IN K, NU(J,K+1/2). THIS MEANS THAT SOME QUANTITIES ARE NEEDED
C       AT HALF POINTS AND OTHERS ARE NOT.

C       STEP 1.  CALCULATE THE DAMPING FACTORS AT 1/2 POINTS

        RATT= C1E/C2E

        DO J=1,JMAX
          DO K=1,KMAX-1
            JB  = IB(J,K)
            WT  = BWT(J,K)
            JJ1 = JB
            JJ2 = JB + 1
            IF(JB .EQ. JMAX) JJ2 = 1
            WM1   = FNU(JJ1,1)
            WM2   = FNU(JJ2,1)
            SSW1  = ABS(GQ(JJ1,1,1,2) - GQ(JJ1,1,2,1))
            SSW2  = ABS(GQ(JJ2,1,1,2) - GQ(JJ2,1,2,1))
            SSW   = ABS(SSW1 + WT*(SSW2-SSW1))
            WNU   = WM1 + WT*(WM2-WM1)
            RHOW  = Q(JJ1,1,1) + WT*(Q(JJ2,1,1)-Q(JJ1,1,1))
            RA    = SQRT( RE*(SSW/WNU + 1.E-10*FSMACH**2))
            YPLS  = RA*SMIN(J,K)
            YPLS  = MAX(YPLS,.0001)
            YPLSM = RA*.5*(SMIN(J,K)+SMIN(J,K+1))
               IF(J .EQ. 91 .AND. K .LE. 5) PRINT*,'Y+= ',YPLSM
            EXP1    =     EXP(-(YPLS/APLUS1))
            EXP2    =     EXP(-(YPLS/APLUS2))
            DAMP1(J,K)  = (1.-EXP1)*(1.-EXP2)
            DAMP1M(J,K) = (1. - EXP(-(YPLSM/APLUS1)))*
     >                          (1. - EXP(-(YPLSM/APLUS2)))

            DDY = EXP1/APLUS1*(1.-EXP2)
     >          + EXP2/APLUS2*(1.-EXP1)
            DD = DAMP1(J,K)
            SDD = SQRT(DD)
            DAMP2(J,K) =
     >              RATT + (1.-RATT)*( 1./(AKARMAN*YPLS) + DD )*
     >              (SDD + YPLS*DDY/SDD)

          ENDDO
          DAMP1 (J,KMAX) = 1.0
          DAMP1M(J,KMAX) = 1.0
        ENDDO
```

```
C       NOW SOLVE THE EQUATION

        DO 500 NIT=1,NNIT

        DO K=1,KMAX
          DO J=1,JMAX
            TURMU(J,K) = CMU*DAMP1(J,K)*TURRE(J,K)
          ENDDO
        ENDDO

C       F_ETA_ETA VISCOUS TERMS

        DO J=1,JMAX
          DO K=2,KMAX-1
            KP1 = K+1
            KM1 = K-1
            XY3P = .5*(XY(J,K,3)+XY(J,KP1,3))
            XY4P = .5*(XY(J,K,4)+XY(J,KP1,4))
            TTP =  (XY3P*XY(J,K,3)+XY4P*XY(J,K,4))

            XY3M = .5*(XY(J,K,3)+XY(J,KM1,3))
            XY4M = .5*(XY(J,K,4)+XY(J,KM1,4))
            TTM = (XY3M*XY(J,K,3)+XY4M*XY(J,K,4))

            CNUD = ( FNU(J,K)+TURMU(J,K)/SIGE
     >                 + CMU*DAMP1(J,K)*TURRE(J,K)/SIGE )/RE

            CDP  = TTP*CNUD
            CDM = TTM*CNUD

            TREP = .5*(TURRE(J,KP1)+TURRE(J,K))
            TREM = .5*(TURRE(J,KM1)+TURRE(J,K))

            CAP  =  CMU*TREP*DAMP1M(J,K  )*TTP/(SIGE*RE)
            CAM  =  CMU*TREM*DAMP1M(J,KM1)*TTM/(SIGE*RE)

C           THIS COMES FROM MAXIMUM PRINCIPLE ANALYSIS

            BY(J,K) = MIN(-CDM + CAM,0.0)
            CY(J,K) = MAX( CDP - CAP,0.0) + MAX( CDM - CAM,0.0)
            DY(J,K) = MIN(-CDP + CAP,0.0)

            FY(J,K) = - BY(J,K)*TURRE(J,KM1)
     >                - CY(J,K)*TURRE(J,K  )
     >                - DY(J,K)*TURRE(J,KP1)

          ENDDO
        ENDDO

C       ADVECTIVE TERMS IN ETA

        DO J=1,JMAX
          DO K=2,KMAX-1
            UU = (XY(J,K,3)*Q(J,K,2)+XY(J,K,4)*Q(J,K,3))
            SGNU = SIGN(1.,UU)
            APP  = .5*(1.+SGNU)
            APM  = .5*(1.-SGNU)
            FY(J,K) = FY(J,K) -
```

```
     >                    UU*( APP*(TURRE(J,K)-TURRE(J,K-1))
     >                    +APM*(TURRE(J,K+1)-TURRE(J,K)) )
            BY(J,K) = BY(J,K) - UU*APP
            CY(J,K) = CY(J,K) + UU*(APP-APM)
            DY(J,K) = DY(J,K) + UU*APM
          ENDDO
        ENDDO
        PRINT*,'ETA TERMS DONE'

C       E_XI_XI VISCOUS TERMS

        DO J=1,JMAX
          JP1 = J+1
          JM1 = J-1
          IF(J .EQ. 1) JM1 = JMAX
          IF(J .EQ. JMAX) JP1 = 1
          DO K=2,KMAX-1
            XY1P = .5*(XY(J,K,1)+XY(JP1,K,1))
            XY2P = .5*(XY(J,K,2)+XY(JP1,K,2))
            TTP =  (XY1P*XY(J,K,1)+XY2P*XY(J,K,2))
            XY1M = .5*(XY(J,K,1)+XY(JM1,K,1))
            XY2M = .5*(XY(J,K,2)+XY(JM1,K,2))
            TTM =  (XY1M*XY(J,K,1)+XY2M*XY(J,K,2))

            CNUD=( FNU(J,K)+TURMU(J,K)/SIGE
     >             + CMU*DAMP1(J,K)*TURRE(J,K)/SIGE )/RE


            CDP = TTP*CNUD
            CDM = TTM*CNUD

            TREP =.5*(TURRE(JP1,K)+TURRE(J,K))
            TREM =.5*(TURRE(JM1,K)+TURRE(J,K))

            CAP=CMU*TREP*.5*(DAMP1(J,K)+DAMP1(JP1,K))*TTP/(SIGE*RE)
            CAM=CMU*TREM*.5*(DAMP1(JM1,K)+DAMP1(J,K))*TTM/(SIGE*RE)


C       THIS COMES FROM MAXIMUM PRINCIPLE ANALYSIS

            BX(K,J) = MIN(-CDM + CAM,0.0)
            CX(K,J) = MAX( CDP - CAP,0.0) + MAX( CDM - CAM,0.0)
            DX(K,J) = MIN(-CDP + CAP,0.0)

            FY(J,K) = FY(J,K) - BX(K,J)*TURRE(JM1,K)
     >                - CX(K,J)*TURRE(J,K  )
     >                - DX(K,J)*TURRE(JP1,K)

          ENDDO
        ENDDO

C       ADVECTIVE TERMS IN XI

        DO J=1,JMAX
          P1 = J+1
          JM1 = J-1
          IF(J .EQ. 1) JM1 = JMAX
```

```
          IF(J .EQ. JMAX) JP1 = 1
          DO K=2,KMAX-1
            UU = (XY(J,K,1)*Q(J,K,2)+XY(J,K,2)*Q(J,K,3))
            SGNU = SIGN(1.,UU)
            APP  = .5*(1.+SGNU)
            APM  = .5*(1.-SGNU)
            FY(J,K)= FY(J,K) -
     >                 UU*(APP*(TURRE(J,K)-TURRE(JM1,K))
     >                 +APM*(TURRE(JP1,K)-TURRE(J,K)) )
            BX(K,J) = BX(K,J) - UU*APP
            CX(K,J) = CX(K,J) + UU*(APP-APM)
            DX(K,J) = DX(K,J) + UU*APM
          ENDDO
        ENDDO
        PRINT*,'XI TERMS DONE'
C
        DO K=2,KMAX-1
          KM1 = K - 1
          DO J=1,JMAX
            UX = GQN(J,K,1,1)
            UY = GQN(J,K,1,2)
            VX = GQN(J,K,2,1)
            VY = GQN(J,K,2,2)

C       REAL PRODUCTION OF K

C           SS = SQRT(ABS(
C    >            2.*(UX**2+VX*UY+VY**2)+UY**2+VX**2
C    >            -.666666*(UX + VY)**2))


C       THIN LAYER APPROXIMATION

            SS = ABS(UY-VX)

            TT=(C2E*DAMP2(J,K)-C1E)*SQRT(CMU*DAMP1(J,K))*SS

            FACT = DS(J,K)*DTM
            BX(K,J) = BX(K,J)*FACT
            CX(K,J) = CX(K,J)*FACT + 1.
            DX(K,J) = DX(K,J)*FACT
            BY(J,K) = BY(J,K)*FACT
            CY(J,K) = CY(J,K)*FACT + 1.
            DY(J,K) = DY(J,K)*FACT
            FY(J,K) = (FY(J,K)+TT*TURRE(J,K))*FACT
          ENDDO
        ENDDO

        CALL TRIV(JDIM,KDIM,1,JMAX,2,KMAX-1,BY,CY,DY,FY)

        DO K=2,KMAX-1
          DO J=1,JMAX
            BY(J,K) = BX(K,J)
            CY(J,K) = CX(K,J)
            DY(J,K) = DX(K,J)
          ENDDO
        ENDDO
```

```
      CALL VTRIP(JDIM,KDIM,1,JMAX,2,KMAX-1,
     >                   BY,CY,DY,FY)

      DO K=2,KMAX-1
        DO J=1,JMAX
          FX(K,J) = FY(J,K)
        ENDDO
      ENDDO
      NEGN = 0
      SUMN = 0.0
      JMIN = 1
      KMIN = 1
      FXM  = ABS(FX(1,1))

      DO K=2,KMAX-1
        DO J=1,JMAX
          TURRE(J,K) = TURRE(J,K) + FX(K,J)
          IF(ABS(FX(K,J)).GT.FXM)THEN
            FXM = ABS(FX(K,J))
            MIN = J
            KMIN = K
          ENDIF

C       THIS NEXT CHECK IS NEVER NEEDED IN PRACTICE
          IF(TURRE(J,K).LT. 1.E-12)THEN
            NEGN = NEGN + 1
            TURRE(J,K) = 1.E-12
            FX(K,J) = 0.0
          ENDIF
          SUMN = SUMN + (FX(K,J))**2
        ENDDO
      ENDDO

      SUMN = SQRT(SUMN)/FLOAT((JMAX-1)*(KMAX-1))

C     OUT FLOW BOUNDARY CONDITION FOR A POLAR MESH.
C     DO J=1,JMAX
C       TURRE(J,1) = 0.
C       IF(Q(J,KMAX,2)*XY(J,KMAX,3)+Q(J,KMAX,3)*XY(J,KMAX,4) .GE. 0.)
C    *                TURRE(J,KMAX) = TURRE(J,KMAX-1)
C       ENDDO

      XMU=0.0
      TMAX = 0.0
      DO K=1,KMAX-1
        DO J=1,JMAX
          TMAX = MAX(TMAX,TURRE(J,K))
          TURMU(J,K) = CMU*DAMP1M(J,K)*.5*(
     >                       TURRE(J,K)*Q(J,K,1)
     >                       + TURRE(J,K+1)*Q(J,K+1,1))

          IF (TURMU(J,K) .GT. XMU) THEN
          XMU = TURMU(J,K)
          JMU=J
          KMU=K
          ENDIF
        ENDDO
      ENDDO
```

```
      WRITE(*,'(A,E12.6,A,E12.6,A,E12.6,I5,1X,I5)')
     $          ' RESID RE_T = ',SUMN,' MAX RT = ',TMAX,
     $          ' RESID MAX  = ',FXM,JMIN,KMIN

      WRITE(*,*) 'MAX TURMU = ', XMU,XMU*AL/RE,'   (J,K)= ',JMU,KMU

500   CONTINUE

C     ********CHANGE J DIRECTION OF TURMU AND SET TO EDDY**********
      DO K = 1, KMAX
        DO J = 2, JMAX
          TEMPMU(J,K) = TURMU(J,K)
        ENDDO
      ENDDO
C
      DO K = 1, KMAX
        DO J = 2, JMAX
          EDDY(J,K) = TEMPMU(JMAX+2-J,K)*AL/RE
        ENDDO
        EDDY(1,K) = TURMU(1,K)*AL/RE
      ENDDO
      DO K=1,KMAX
        EDDY(1,K)=.5*(EDDY(2,K)+EDDY(JMAX,K))
      ENDDO
C
C     DO J = 1, JMAX
C       EDDY(J,1) = 0.0
C     ENDDO

      RETURN
      END
```

```
C*********************************************************************
C*******************MEAN VELOCITY GRADIENT****************************
C*********************************************************************
      SUBROUTINE GRADV(MAXJ,MAXK,JDIM,KDIM,JMAX,KMAX,
     >                          JTAIL1,JTAIL2,X,Y,AA,AAN,Q,GQ,GQN)

      DIMENSION AA(0:MAXJ,0:MAXK),AAN(MAXJ,MAXK)
      DIMENSION Q(JDIM,KDIM,4)
      DIMENSION X(JDIM,KDIM),Y(JDIM,KDIM)
      DIMENSION GQ(0:JDIM,0:KDIM,2,2),GQN(JDIM,KDIM,2,2)

C     NOTE THE ZERO STARTING INDEX IN THE DIMENSIONS

      DO 10 N=1,2
        DO 10 K=1,KMAX
          DO 10 J=1,JMAX
            GQ(J,K,N,1) = 0.0
            GQ(J,K,N,2) = 0.0
            GQN(J,K,N,1) = 0.0
            GQN(J,K,N,2) = 0.0
10    CONTINUE

      DO 20 NN=2,3
      N = NN-1
        DO 20 K=1,KMAX-1
          DO 20 J=1,JMAX
            JM1 = J-1
            IF(J .EQ. 1) JM1 = JMAX
            DX = X(J,K+1)-X(J,K)
            DY = Y(J,K+1)-Y(J,K)
            SNX =  DY
            SNY = -DX
      GQ(JM1,K,N,1) = GQ(JM1,K,N,1) + SNX*.5*(Q(J,K+1,NN)+Q(J,K,NN))
      GQ(J,K,N,1)   = GQ(J,K,N,1)   - SNX*.5*(Q(J,K+1,NN)+Q(J,K,NN))
      GQ(JM1,K,N,2) = GQ(JM1,K,N,2) + SNY*.5*(Q(J,K+1,NN)+Q(J,K,NN))
      GQ(J,K,N,2)   = GQ(J,K,N,2)   - SNY*.5*(Q(J,K+1,NN)+Q(J,K,NN))
20    CONTINUE

      DO 30 NN=2,3
      N = NN-1
        DO 30 K=1,KMAX
          DO 30 J=1,JMAX
            P1=J+1
            IF(J .EQ. JMAX) JP1 = 1
            DX = X(JP1,K)-X(J,K)
            DY = Y(JP1,K)-Y(J,K)
            SNX =  DY
            SNY = -DX
      GQ(J,K-1,N,1) = GQ(J,K-1,N,1) - SNX*.5*(Q(JP1,K,NN)+Q(J,K,NN))
      GQ(J,K,N,1)   = GQ(J,K,N,1)   + SNX*.5*(Q(JP1,K,NN)+Q(J,K,NN))
      GQ(J,K-1,N,2) = GQ(J,K-1,N,2) - SNY*.5*(Q(JP1,K,NN)+Q(J,K,NN))
      GQ(J,K,N,2)   = GQ(J,K,N,2)   + SNY*.5*(Q(JP1,K,NN)+Q(J,K,NN))
30    CONTINUE
```

```
      DO 90 M=1,2
      DO 90 N=1,2
      DO 90 K=1,KMAX-1
      DO 90 J=1,JMAX
        JP1=J+1
        IF(J .EQ. JMAX) JP1 = 1
        GQN(J,K,N,M) = GQN(J,K,N,M) + GQ(J,K,N,M)
        GQN(JP1,K,N,M) = GQN(JP1,K,N,M) + GQ(J,K,N,M)
        GQN(J,K+1,N,M) = GQN(J,K+1,N,M) + GQ(J,K,N,M)
        GQN(JP1,K+1,N,M) = GQN(JP1,K+1,N,M) + GQ(J,K,N,M)
90    CONTINUE

C     IF(JTAIL2 .NE. JMAX)THEN
C     K = 1
C     DO 95 N=1,2
C     DO 95 J=1,JTAIL1
C       JJ = JMAX - J + 1
C       TEMP1 = GQN(J,K,N,1)
C       EMP2 = GQN(JJ,K,N,1)
C       GQN(J,K,N,1) = TEMP1 + TEMP2
C       GQN(JJ,K,N,1)= TEMP2 + TEMP1
C       TEMP1 = GQN(J,K,N,2)
C       TEMP2 = GQN(JJ,K,N,2)
C       GQN(J,K,N,2) = TEMP1 + TEMP2
C       GQN(JJ,K,N,2)= TEMP2 + TEMP1
C95   CONTINUE
C     ENDIF

      DO 110 N=1,2
      DO 110 K=1,KMAX-1
      DO 110 J=1,JMAX
        GQ(J,K,N,1) = GQ(J,K,N,1)/AA(J,K)
        GQ(J,K,N,2) = GQ(J,K,N,2)/AA(J,K)
110   CONTINUE


      DO 100 N=1,2
      DO 100 K=1,KMAX
      DO 100 J=1,JMAX
        GQN(J,K,N,1) = GQN(J,K,N,1)/AAN(J,K)
        GQN(J,K,N,2) = GQN(J,K,N,2)/AAN(J,K)
100   CONTINUE

      RETURN
      END
```

```
C*********************************************************************
C*************** SCALAR TRIDIAGONAL **********************************
C*********************************************************************
      SUBROUTINE TRIV(JDIM,KDIM,JL,JU,KL,KU,A,B,C,F)
C
      DIMENSION A(JDIM,KDIM),B(JDIM,KDIM),C(JDIM,KDIM)
      DIMENSION X(JDIM,KDIM),F(JDIM,KDIM)
C
      DO 10 J=JL,JU
        X(J,KL)=C(J,KL)/B(J,KL)
        F(J,KL)=F(J,KL)/B(J,KL)
10    CONTINUE
      KLP1 = KL +1
      DO 1 I=KLP1,KU
        DO 20 J=JL,JU
          Z=1./(B(J,I)-A(J,I)*X(J,I-1))
          X(J,I)=C(J,I)*Z
          F(J,I)=(F(J,I)-A(J,I)*F(J,I-1))*Z
20      CONTINUE
1     CONTINUE
C
      KUPKL=KU+KL
      DO 2 I1=KLP1,KU
        I=KUPKL-I1
        DO 30 J=JL,JU
          F(J,I)=F(J,I)-X(J,I)*F(J,I+1)
30      CONTINUE
2     CONTINUE
C
      RETURN
      END
```

```
C*********************************************************************
C*************** SCALAR PERIODIC TRIDIAGONAL ************************
C*********************************************************************
      SUBROUTINE VTRIP(JDIM,KDIM,J1,J2,K1,K2,A,B,C,F)
C
      DIMENSION A(JDIM,KDIM),B(JDIM,KDIM),C(JDIM,KDIM),F(JDIM,KDIM),
     >          QD(JDIM,KDIM),S(JDIM,KDIM),FND(KDIM)

      JA = J1 + 1
C
C     FORWARD ELIMINATION SWEEP
C
      DO 1 K = K1,K2
        FND(K) = F(J2,K)
        QD(J1,K) = -C(J1,K)/B(J1,K)
        F(J1,K) = F(J1,K)/B(J1,K)
        S(J1,K) = - A(J1,K)/B(J1,K)
1     CONTINUE
C
      DO 10 J=JA,J2
        DO 2 K = K1,K2
          P =1./( B(J,K) + A(J,K)*QD(J-1,K))
          QD(J,K) = - C(J,K)*P
          F(J,K) = ( F(J,K) - A(J,K)*F(J-1,K))*P
          S(J,K) = - A(J,K)*S(J-1,K)*P
2       CONTINUE
10    CONTINUE
C
C     BACKWARD PASS
      JJ = J1 + J2
      DO 3 K = K1,K2
        QD(J2,K) = 0.
        S(J2,K) = 1.
3     CONTINUE
C
      DO 11 I=JA,J2
        J = JJ - I
        DO 4 K = K1,K2
          S(J,K) = S(J,K) + QD(J,K)*S(J+1,K)
          QD(J,K) = F(J,K) + QD(J,K)*QD(J+1,K)
4       CONTINUE
11    CONTINUE
      DO 5 K = K1,K2
        F(J2,K) = ( FND(K) - C(J2,K)*QD(J1,K) - A(J2,K)*QD(J2-1,K))/
     1              ( C(J2,K)*S(J1,K) + A(J2,K)*S(J2-1,K)  +B(J2,K))
5     CONTINUE
C
C     BACKWARD ELIMINATION PASS
      DO 12 I=JA,J2
        J = JJ -I
        DO 6 K = K1,K2
          F(J,K) = F(J2,K)*S(J,K) + QD(J,K)
6       CONTINUE
12    CONTINUE
      RETURN
      END
```

```
C*************************************************************
C***************METRICS SUBROUTINE*************************
C*************************************************************
      SUBROUTINE XYMETS(JMAX,KMAX,X,Y,XY,XYJ)
C
      PARAMETER(JDIM=120,KDIM=80,MAXJ=121,MAXK=81)
      DIMENSION X(JDIM,KDIM),Y(JDIM,KDIM),XY(JDIM,KDIM,4)
      DIMENSION XYJ(JDIM,KDIM), WORKX(JDIM,KDIM),
     $                WORKY(JDIM,KDIM)
      LOGICAL PERIODIC, SHARP, CUSP
C
C     READ IN X, Y COORDINATES
C

C     **FORMAT FOR ONE-ELEMENT GEOM**
C           USE FORT.10 FOR PLOT3D

      OPEN(12)
      READ(12,*) JDUM,KDUM, AL
      READ(12,100) ((X(J,K), J=1,JDUM), K=1,KDUM)
      READ(12,100) ((Y(J,K), J=1,JDUM), K=1,KDUM)
      REWIND 12
100   FORMAT(10E19.12)
C     ** FORMAT FOR TWO-ELEMENTS GEOM**
C     OPEN(12)
C     READ(12,*) JDUM,KDUM
C     READ(12,*) ((X(J,K), J=1,JDUM), K=1,KDUM),
C    %                ((Y(J,K), J=1,JDUM), K=1,KDUM)
C     REWIND 12
C
C     CHANGE DIRECTION OF 'J' TO CLOCKWISE FOR BB MODEL
C
      DO K = 1, KMAX
        DO J = 2, JMAX
          WORKX(J,K) = X(J,K)/AL
          WORKY(J,K) = Y(J,K)/AL
        ENDDO
        X(1,K)=X(1,K)/AL
        Y(1,K)=Y(1,K)/AL
      ENDDO
C
      DO K = 1, KMAX
        DO J = 2, JMAX
          X(J,K) = WORKX(JMAX+2-J,K)
          Y(J,K) = WORKY(JMAX+2-J,K)
        ENDDO
      ENDDO
C
      JLOW=1
      JUP=JMAX
C     JLOW=2
C     JUP=JMAX-1
C     XY4 =XXI,XY3 = YXI
C
```

```
      DO 11 K=1,KMAX
        DO 10 J=JLOW,JUP
          JP1 = J + 1
          JM1 = J - 1
          IF(J .EQ. 1) JM1 = JUP
          IF(J .EQ. JUP) JP1 = JLOW
          XY(J,K,4) = ( X(JP1,K) - X(JM1,K))*.5
          XY(J,K,3) = ( Y(JP1,K) - Y(JM1,K))*.5
10      CONTINUE
C
      JM=JMAX-1
C     IF(.NOT.PERIODIC)THEN
C         XY(1,K,4) = .5*( -3.*X(1,K) +4.*X(2,K) - X(3,K))
C         XY(1,K,3) = .5*( -3.*Y(1,K) +4.*Y(2,K) - Y(3,K))
C         XY(JMAX,K,4) = ( 3.*X(JMAX,K) -4.*X(JM,K) + X(JM-1,K))*.5
C         XY(JMAX,K,3) = ( 3.*Y(JMAX,K) -4.*Y(JM,K) + Y(JM-1,K))*.5
C     ENDIF
C
11    CONTINUE
C
C     XY2 = XETA, XY1 = YETA
C
      DO 21 J=1,JMAX
        DO 20 K=2,KMAX-1
          XY(J,K,2) = ( X(J,K+1) - X(J,K-1))*.5
          XY(J,K,1) = ( Y(J,K+1) - Y(J,K-1))*.5
20      CONTINUE
        XY(J,1,2) = ( -3.*X(J,1) +4.*X(J,2) - X(J,3))*.5
        XY(J,1,1) = ( -3.*Y(J,1) +4.*Y(J,2) -Y(J,3))*.5
        KM=KMAX-1
        XY(J,KMAX,2) = ( 3.*X(J,KMAX) -4.*X(J,KM) + X(J,KM-1))*.5
        XY(J,KMAX,1) = ( 3.*Y(J,KMAX) -4.*Y(J,KM) + Y(J,KM-1))*.5
21    CONTINUE
C
C     FOR PERIODIC GRIDS WITH SHARP OR CUSP TRAILING EDGES USE
C     FIRST ORDER DERIVATIVE FOR ETA TERMS, SECOND ORDER
C     SOMETIMES LEADS TO NEGATIVE JACOBIANS
      PERIODIC = .TRUE.
      SHARP = .TRUE.
      CUSP = .FALSE.
      IF( PERIODIC )THEN
        J = 1
        IF( SHARP .OR. CUSP)THEN
          XY(J,1,2) =  -X(J,1) + X(J,2)
          XY(J,1,1) =  -Y(J,1) + Y(J,2)
        ENDIF
      ENDIF
C
      DO 30 K=1,KMAX
        DO 30 J=1,JMAX
          DINV = 1./ ( XY(J,K,4) * XY(J,K,1) - XY(J,K,3) * XY(J,K,2) )
          IF(DINV .LE. 0.) THEN
            PRINT*, J,K
            PRINT*, XY(J,K,1),XY(J,K,2), XY(J,K,3),XY(J,K,4)
          ENDIF
          IF(DINV .LE. 0.0)WRITE(6,*)'JACOBIAN( ',J,K,' )= ',DINV
          XYJ(J,K) = DINV
30    CONTINUE
```

```
      DO 32 K=1,KMAX
      DO 32 J=1,JMAX
        DINV = XYJ(J,K)
C
C     XIX = XY1, XIY=XY2,ETAX=XY3, ETAY = XY4
C
        XY(J,K,1) =  XY(J,K,1)*DINV
        XY(J,K,2) = - XY(J,K,2)*DINV
        XY(J,K,3) = - XY(J,K,3)*DINV
        XY(J,K,4) =  XY(J,K,4)*DINV
32    CONTINUE
C
      RETURN
      END
```

```
C*********************************************************************
C***************DEFINITION OF MAJOR VARIABLES*************************
C*********************************************************************
```

AA	WEIGHTS FOR VELOCITY GRADIENT COMPUTATION AT CELL CENTER
AAN	WEIGHTS FOR VELOCITY GRADIENT COMPUTATION AT GRID POINT
AL	CHORD LENGTH
ALPI	ANGLE OF ATTACK OF THE AIRFOIL
BX	LHS MATRIX ENTRIES IN THE XI DIRECTION
BY	LHS MATRIX ENTRIES IN THE ETA DIRECTION
BWT	INTERPOLATION RATIO OF SMIN LOCATION ON THE SURFACE
CX	LHS MATRIX ENTRIES IN THE XI DIRECTION
CY	LHS MATRIX ENTRIES IN THE ETA DIRECTION
DAMP1	THE PRODUCT OF THE DAMPING FACTORS IN BB MODEL, D1 * D2
DAMP1M	THE PRODUCT OF THE DAMPING FACTORS IN BB MODEL, D1 * D2 AT MID K POINT
DAMP2	DAMPING FACTOR, F2(Y^+)
DS	SPATIALLY VARIABLE TIME STEP, SCALED ON METRIC JACOBIAN
DT	TIME STEP
DX	LHS MATRIX ENTRIES IN THE XI DIRECTION
DY	LHS MATRIX ENTRIES IN THE ETA DIRECTION
EDDY	EDDY VISCOSITY IN ZETA
FIRST	LOGICAL: (T) THE SUBROUTINE IS CALLED FOR THE FIRST TIME
FMU	LAMINAR EDDY VISCOSITY
FNU	KINEMATIC VISCOSITY
FSMACH	FREE STREAM MACH NUMBER
FX	RHS MATRIX ENTRIES IN THE XI DIRECTION
FY	RHS MATRIX ENTRIES IN THE ETA DIRECTION
GAMI	GAMMA - 1
GAMMA	GAS CONSTANT (1.4)
GQ	VELOCITY GRADIENTS AT CELL CENTER
GQN	VELOCITY GRADIENTS AT GRID POINT
IB	POINTER FOR SMIN
ICST	FLOW INDICATOR: 0 - INITIAL RUN 1 - STEADY CASE 2/3 - PITCHING UP 4 - OSCILLATING
JDIM	J DIMENSION OF THE GRID
JMAX	TOTAL NUMBER OF POINTS IN XI DIRECTION
JTAIL1	FIRST SOLID BODY POINT
JTAIL2	LAST SOLID BODY POINT
KDIM	K DIMENSION OF THE GRID
KMAX	TOTAL NUMBER OF POINTS IN ETA DIRECTION
NNIT	INDEX FOR ITERATIVE SOLUTION OF TURRE, NNIT = 1 FOR TIME ACCURATE SOLUTION
PERIODIC	PERIODIC(T) AND NON-PERIODIC(F) OPTIONS
PI	4.*ATAN(1.)
PRESS	PRESSURE (PLOT3D FORMAT)
Q	CONSERVATIVE VARIABLES (PLOT3D FORMAT)
RE	REYNOLDS NUMBER
SMIN	MINIMUM DISTANCE TO THE WALL
SS	PRODUCTION OF K OR TEMPORARY VARIABLE FOR BWT
TINF	INFINITY TEMPERATURE
TURMU	TURBULENT EDDY VISCOSITY
TURRE	TURBULENT REYNOLDS NUMBER
U	TANGENTIAL VELOCITY
V	NORMAL VELOCITY
VORT	VORTICITY

X	CARTESIAN X COORDINATES OF THE GRID
XY	METRIC TRANSFORMATIONS
	XY(J,K,1) = D XI / DX
	XY(J,K,2) = D XI / DY
	XY(J,K,3) = D ETA / DX
	XY(J,K,4) = D ETA / DY
XYJ	JACOBIANS OF METRIC TRANSFORMATIONS
Y	CARTESIAN Y COORDINATES OF THE GRID

LIST OF REFERENCES

[1] Tung, C. and Branum, L., "Model Tilt-Rotor Hover Performance and Surface Pressure Measurement," AHS, Annual Forum, 46th, 1990.

[2] McAlister, K.W., Pucci, S.L., McCroskey, W.J. and Carr, L.W., "An Experimental Study of Dynamic Stall on Advance Airfoil Sections, Vol. 2 - Pressure and Force Data," NASA TM-84245, 1982.

[3] Carr, L.W., Chandraekhara, M.S. and Brock, N.J. "A Quantitative Study of Unsteady Compressible flow on an Oscillating Airfoil," AIAA Paper 91-1683.

[4] McAlister, K.W., Carr, L.W., "Water Tunnel Visualizations of Dynamic Stall," ASME, 1978, p 103-110.

[5] Geissler, W., Carr, L.W., and Cebeci, T, "Unsteady Separation Characteristics of Airfoil Operating under Dynamic stall Conditions," DGLR, European Rotorcraft Forum, 1986, Paper.

[6] Strawn, R.C., and Tung, C., "Prediction of Unsteady Transonic Rotor Loads with a Full-Potential Rotor Code," AHS, Annual Forum, 43rd, 1987.

[7] Hassan, A.A, Tung, C. and Sankar, L.N., "Euler Solutions for Self-Generated Rotor Blade-Vortex Interactions," AIAA Paper 90-1588.

[8] Gulcat, U., Wu, J.C. and Wang, C.M., "Zonal Solution of Unsteady Viscous Flow Problems," AIAA Paper 84-1637.

[9] Wang, C.M., Wu, J.C. and Tung, C., "A Numerical Study of General Viscous Flows Around Multi-Element Airfoils," AIAA Paper 90-0572.

[10] Pattern, M.T., Wu, J.C. and Wang, C.M., "ZETA - A Manual for a Computer Code that Uses a Zonal Procedure for Evaluating Turbulent and Laminar Flows," Georgia Institute of Technology, 1987.

[11] Wu, J.C. and Thompson, J.F., "Numerical Solution of Time Dependent Incompressible Navier-Stokes Equations Using an Integro-Differential Formulations," J. Computers and Fluids, vol. 1, No. 2, pp 197-215.

[12] Baldwin, B.S. and Lomax, H., "Thin Layer Approximations and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257.

[13] Baldwin, B.S. and Barth, T., " A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows," NASA TM-102847, 1990.

[14] Menter, F.R., "Performance of Popular Turbulence Models for Attached and Separated Adverse Pressure Gradient Flows," AIAA Paper 91-1784.

[15] Mehta, U.B., "Starting Vortex, Separation Bubbles and Stall - A Numerical Study of Laminar Unsteady Flow Around an Airfoil," Ph.D. Thesis, Illinois Institute of Technology, December, 1972.

[16] Anderson, D.A., Tannehill, J.C. and Pletcher, R.H., Computational Fluid Mechanics and Heat Transfer, Hemisphere Publishing Co., New York, c. 1984.

[17] Schlichting, H., Boundary Layer Theory, McGraw-Hill, Seventh Edition, 1979.

[18] Wu, J.C., "Problems of General Viscous Flow", Chapter 4 of Developments in Boundary Element Methods-2, Editors: R. Shaw and P. Banerjee, Applied Science Publishers LTD, pp. 69-109, 1982.

[19] Wang, C.M., "Computation of Steady Internal Flows Using Integral Representation Methods with Finite Series Expansions," Ph.D. Thesis, Georgia Institute of Technology, 1983.

[20] York, B. and Knight, D., "Calculation of a Class of Two-Dimensional Turbulent Boundary Layers Using the Baldwin-Lomax Model," AIAA Paper 85-0126.

[21] Cebeci, T., "Calculatin of Compressible Turbulent Boundary Layers with Heat and Mass transfer," AIAA Paper 70-741.

[22] Tennekes, H., and Lumley, J.L., A First Course in Turbulence, MIT Press, Ninth Printing, 1983.

[23] Baldwin, B.S. and Barth, T., "A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows," AIAA Paper 91-0610.

[24] Pulliam, T.H., "Euler and Thin Layer Navier-Stokes Codes: ARC2D, ARC3D," Notes for Computational Fluid Dynamics User's Workshop. 1984.

[25] Barth, T.,"Numerical Aspects of Computing Viscous High Reynolds Number Flows on Unstructured Meshes," AIAA Paper 91-0721.