(1)

THIN-LAYER NAVIER-STOKES SOLUTIONS FOR A CRANKED DELTA WING

THESIS

Francis R. Smith
Captain, USAF

AFIT/GAE/AA/88D-34

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

(1)

AFIT/GAE/AA/88D-34

THIN-LAYER NAVIER-STOKES SOLUTIONS FOR A CRANKED DELTA WING

THESIS

Francis R. Smith
Captain, USAF

AFIT/GAE/AA/88D-34

Approved for public release; distribution unlimited

AFIT/GAE/AA/88D-34

THIN-LAYER NAVIER-STOKES SOLUTIONS FOR A CRANKED DELTA WING

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Aeronautical Engineering

Francis R. Smith, B.S.

Captain, USAF

December 1988

Preface

Many current fighter aircraft utilize thin delta wings with forebody strakes, which produce strong vortical flows at moderate to high angles of attack. The resulting lift increment is highly nonlinear and cannot be accurately predicted by current design methods. The objective of this thesis is to calculate the flow over a cranked delta wing using the thin-layer Navier-Stokes equations. Emphasis is placed on determining the effects of the strake vortex on the wing vortex and the ability of the code to reproduce the secondary vortex structure.

The algorithm used in this study was the ARC3D code, written by Dr. Thomas Pulliam of the NASA Ames Research Center. It has been extensively modified by Dr. Philip Webster of the Flight Dynamics Laboratory. The calculations were done using the CDC Cyber and Cray XMP-12 computers at Wright-Patterson AFB, Ohio.

I would like to thank Dr. Joseph Shang, of the Flight Dynamics Laboratory, not only for his financial support but also his technical and moral support. I would also like to thank Dr. Philip Webster for his patience and technical expertise on the ARC3D computer code. I would like to thank Dr. Halim for his support of this effort. Finally I would like to thank my biggest supporter, my wife [redacted] who was always there when I needed her.

This thesis was prepared using a Macintosh Plus computer using Microsoft Word word processing software and Expressionist equation writing software. It was printed using an Apple Laserwriter Plus printer.

Table of Contents

List of Figures

List of Symbols

a	speed of sound (m/sec)
C_p	specific heat at constant pressure, or coefficient of pressure
E	inviscid flux vector in ξ direction
E_v	viscous flux vector in the ξ direction
e	total energy
e_i	specific internal energy
F	inviscid flux vector in η direction
F_v	viscous flux vector in the η direction
G	inviscid flux vector in ζ direction
G_v	viscous flux vector in the ζ direction
I	identity matrix
J	transformation Jacobian
k	thermal conductivity
L	model length, mm
M	local Mach number
p	pressure
Pr	Prandtl number, 0.72 for air
Q	flow variable vector
q	heat flux vector
R	gas constant
Re	Reynolds number
t	time
T	absolute temperature (°K)
u,v,w	Cartesian velocity components in the x,y,and z directions.
x,y,z	Cartesian coordinates in streamwise,spanwise, and normal directions

α	angle of attack
γ	ratio of specific heats
μ	molecular viscosity coefficient
ξ,η,ζ	transformed body fitted coordinates
ρ	density
τ	shear stress

Subscripts

aw	adiabatic wall
le	evaluated at the wing leading edge
n	normal to the body surface
T	pitot condition
x	partial derivative with respect to x
y	partial derivative with respect to y
z	partial derivative with respect to z
ξ	partial derivative with respect to ξ
η	partial derivative with respect to η
ζ	partial derivative with respect to ζ

Abstract

For thin, highly swept wings operating at moderate to high angles of attack, ~~the~~ flow over the wing is dominated by ~~the~~ a formation of leading edge vortices. These vortices produce a minimum pressure ~~and~~ this results in an additional lift increment. ~~This~~ lift increment is nonlinear with angle of attack and cannot be accurately predicted using present design methods.

The thin-layer Navier-Stokes equations were used to calculate ~~the~~ flow over a straight delta wing and a cranked delta wing. The straight delta wing was used as the test case due to the availability of both experimental and numerical data. Results are compared with this data in order to validate the numerical procedure. The computer code uses an implicit, time marching algorithm developed by Beam and Warming. The solution is marched in time until a steady state is achieved. The code is approximately factored and diagonalized in order to reduce computational work. A solid state disk is used ~~in order~~ to allow for the large grid needed for a 3-D ~~three dimensional~~ solution.

The thin-layer Navier-Stokes equations are capable of accurately calculating vortical flows. The cranked delta wing exhibited flow similar to a straight delta wing upstream of the crank. The vortex generated at the crank quickly became paired with the vortex from the front of the wing. ~~The~~ vortex location aft of the crank changes with streamwise location. ~~The~~ grid resolution is important when trying to calculate vortical flows, due to the large gradients in both ~~the~~ spanwise and normal directions. The solid state disk can be used to run problems that require more computer memory than is available. Optimization of the program input/output should be done for running the code with the solid state disk ~~in order~~ to reduce ~~the~~ central processor unit time and job cost.

[illegible]

[illegible]

THIN-LAYER NAVIER-STOKES SOLUTIONS FOR A CRANKED DELTA WING

I Introduction

The current class of high performance fighter aircraft must be able to operate over a wide range of flight conditions. A single point design aircraft is no longer acceptable. They must have efficient supersonic cruise performance and yet be able to maneuver transonicly for dogfighting. The supersonic requirement has driven aircraft designers to use thin, highly swept wings. Although these are desirable for cruise performance, they greatly degrade the maneuverability of the aircraft. This performance degradation is caused by flow separation at moderate to high angles of attack. This difficulty in maintaining attached flow, has led designers to explore the possibility of using vortex lift for maneuver enhancement (1:20).

Current fighters, such as the F-16 and the F-18, utilize vortex lift to enhance their performance by the use of wing-strake planforms. The strakes generate strong vortices, which sweep back over the wing and generate a significant increase in lift. This lift increment is a result of the decrease in pressure on the upper surface. It is highly nonlinear and cannot be accurately predicted using current linear design methods.

Newsome and Kandil (2:2-3) have classified the flow over a body into four general categories. The first category is for attached flow which occurs at low angles of attack. The second category, for moderate to high angles of attack, is identified by the formation of large vortices on the lee-side of delta wings, swept wings, low aspect ratio wings, and slender bodies. A typical flow pattern is shown in Figure 1. The flow is dominated by the large primary vortices generated at the wing leading edge. In addition, a secondary vortex is formed because the spanwise flow is unable to negotiate the adverse pressure gradient. The vortices in this region are both stable and symmetric and it is for this reason that most of the research done on vortex flows has been in this region. The third category of flow is

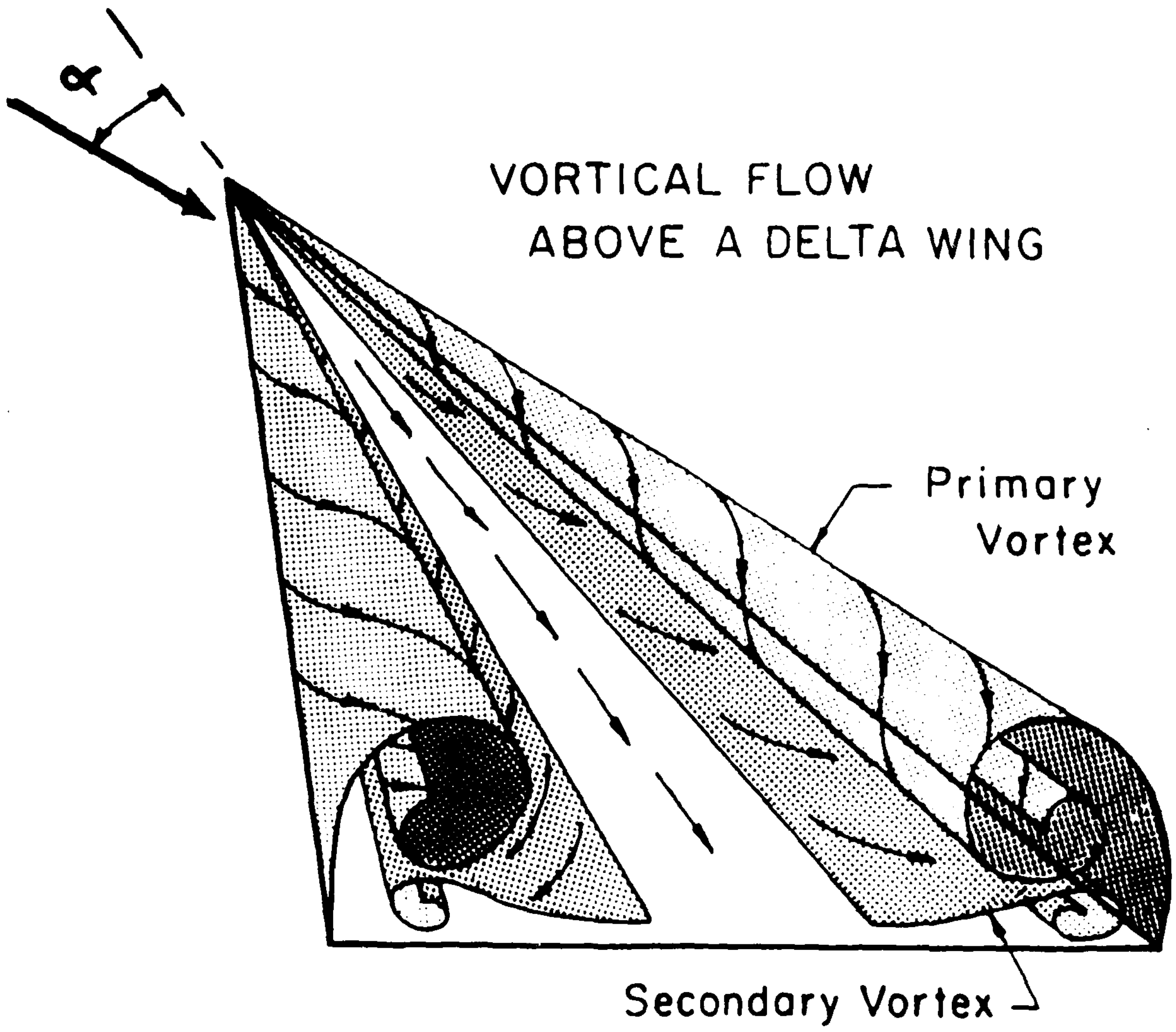


Figure 1. Vortex Flow Over a Delta Wing (3)

for very high angles of attack. The flow in this region produces either unstable or asymmetric vortices. The vortices may burst or become asymmetric, producing large changes in the body forces and moments. This region can be characterized as a transition region from steady stable vortex flows to unsteady, asymmetric flows. The majority of the research in this area is related to predicting vortex breakdown or vortex asymmetry. The final region is for extreme angles of attack. The flow is characterized by an unsteady diffuse wake that may produce vortex shedding. This area is not of much interest, since most flight vehicles are not likely to operate in it.

For sharp delta wings, Stanbrook and Squire (4) have shown that the flow can be classified according to the normal Mach number (M_n) and the normal angle of attack.(α_n), given by

$$M_n = M_\infty \sqrt{1 - \sin^2\Lambda \cos^2\alpha} \qquad (1)$$

$$\alpha_n = \tan^{-1}\left(\frac{\tan\alpha}{\cos\Lambda}\right) \qquad (2)$$

where Λ is the wing leading edge sweep, α is the angle of attack, and M_∞ is the freestream Mach number. The Stanbrook-Squire boundary classifies leading edge flows into either separated flows or attached flows depending upon whether the normal Mach number is less than or greater than one.

Miller and Wood (5) have refined this boundary and identified six possible types of flow, which are shown in Figure 2. For normal Mach numbers less than one and small angles of attack, the flow is characterized by a separation bubble at the leading edge. As angle of attack is increased the flow separates at the leading edge and forms a vortex, which is often referred to as "classical leading-edge separation" (2:3). Normally a secondary vortex will form under the the primary vortex. This is due to the spanwise flow being

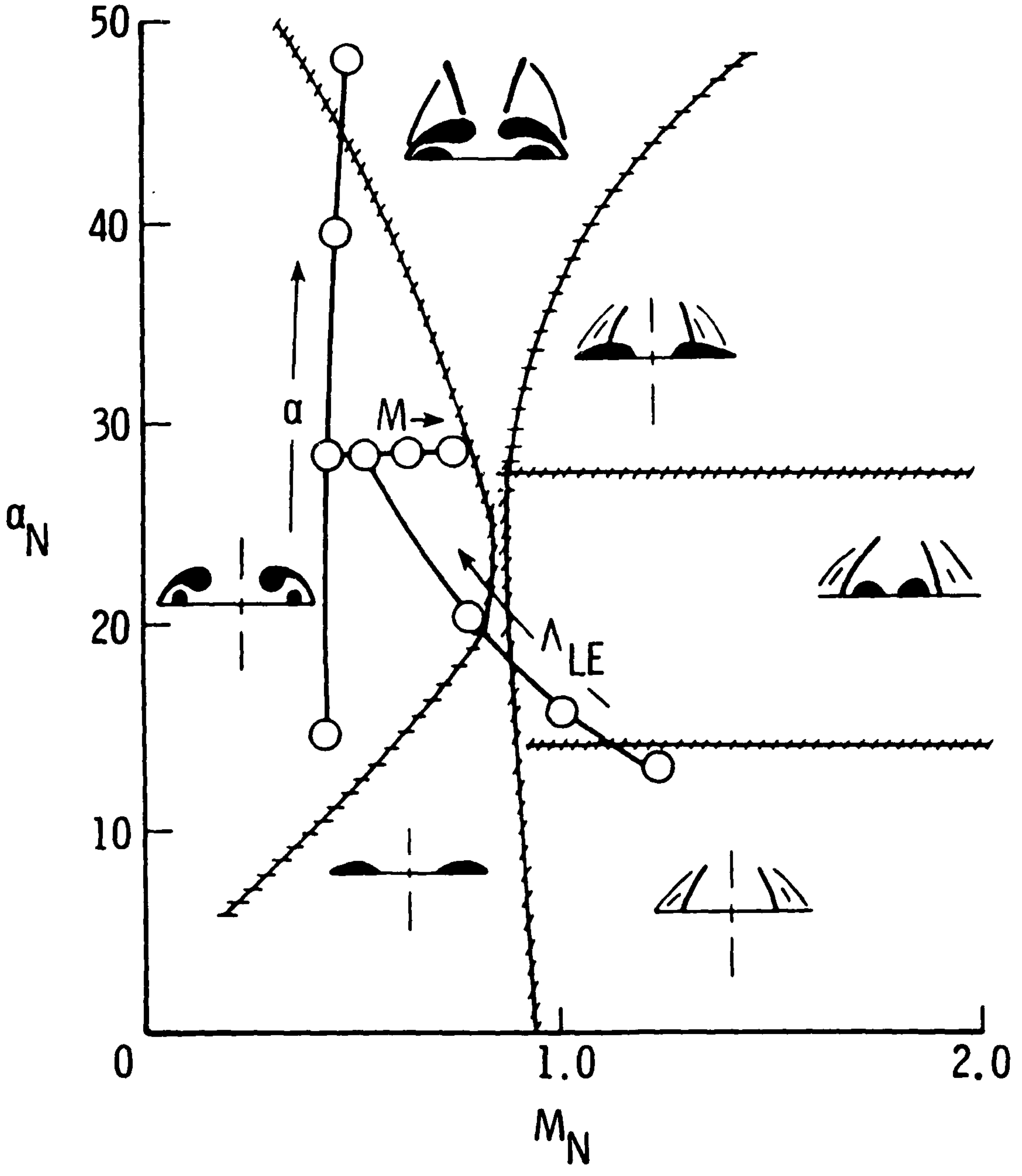


Figure 2. Possible Types of Vortex Flows Over Sharp Delta Wings (5)

unable to negotiate the adverse pressure gradient, thus separating from the wing surface. For normal Mach numbers greater than one, there are three types of flows possible. At low angles of attack the flow is attached and the leading edge expansion is terminated by a crossflow shock. With increasing angle of attack the shock strengthens and causes the formation of a shock induced separation bubble in board on the delta wing. Further increase in angle of attack causes the strengthing of a thin separation bubble with a crossflow shock coalesced on top of it. The final category of flow occurs for very high angles of attack, with either subsonic or supersonic normal Mach numbers. This flow is characterized by strong leading edge vortices and strong crossflow shocks.

Methods which predict vortical flows can be generally classified into two categories. The first category models the vortex in an approximate manner and includes such methods as the Polhamus suction analogy, vortex lattice methods and panel methods. The second category captures the vortex as a solution to the governing equations and includes Euler solutions and Navier-Stokes solutions. The suction analogy (1,6) is an empirical method which can predict forces and moments but is unable to predict pressures and velocities. Vortex lattice methods (7-8) are able to predict pressure and velocities by modeling the vortex sheet as discrete line vortices. In panel methods (9-13) the vortex sheet is more accurately modeled, but prior knowledge of the vortex structure is required (14:825-826). The Euler solutions (15-19) can reproduce the primary vortex but are unable to reproduce the secondary structure. Although the Navier-Stokes solutions (20-25) reproduce all facets of the flow, they have the drawback of being difficult to obtain and computationally expensive.

Most of the early research done on vortical flows used one of the methods that approximately modeled the vortex core. As the speed and size of computers increased, Euler and Navier-Stokes solutions became more feasible. The Euler equations have the advantage of being simpler to solve than the Navier-Stokes equations. The major disadvantage to using the Euler equations to model vortical flows is, that an inviscid approach is used to model a predominatly viscous phenomenon. The Euler equations can only capture the primary vortex and it is for this reason that the Navier-Stokes equations are chosen as the governing equations for this investigation.

The objective of this research is to use the Navier-Stokes equations to calculate the flow over a cranked delta wing. A partial listing of the available experimental data is contained in References 25-31. The configuration chosen corresponds to that tested by Henke (31). This configuration was chosen because its sharp leading edges and flat upper surface, will produce a strong vortex which is free of body influences. The test conditions are for a freestream Mach number of 2.5 and an angle of attack of 10°.

The solution of the full Navier-Stokes equations requires a substantial amount of computing resources, therefore the following assumptions were made. The first was that the viscous effects are only significant in a thin layer near the body. The viscous derivatives normal to the body are large compared to the viscous derivatives along the body, as long as the flow is attached or only mildly separated. The only viscous derivatives retained are those normal to the body and the resulting equations are known as the thin-layer Navier-Stokes equations (32:9). The second assumption was that the flow was laminar. Although this may seem restrictive, the laminar numerical results of Rizzetta and Shang (33) and Buter and Rizzetta (34) were in good agreement with experimental data. The laminar flow assumption appears to be valid for freestream Reynolds numbers up to about one million.

The computer code used is the ARC3D code written by Dr. Thomas Pulliam of the NASA Ames Research Center (32). The code is based on the Beam and Warming implicit approximate factorization algorithm and uses the thin-layer approximation. It has been modified by Dr. Phil Webster of the Flight Dynamics Laboratory. The most significant coding change has been the incorporation of boundary conditions that allow for a branch cut, between the upper and lower body surfaces. This allows for the use of an H-grid topology which gives better modeling of the sharp leading edge.

This version of the code has never been run on the Cray XMP-12 computer at Wright-Patterson AFB. Before the cranked wing calculations could be performed, the code and hardware needed to be checked. This was done by running a test case for which experimental and numerical data were available. The test case corresponds to the configuration tested by Monnerie and Werle (35). Available numerical data includes the full Navier-Stokes calculations of Rizzetta and Shang (33) and Buter and Rizzetta(34), and the conical solutions of Thomas and Newsome (36) and Vigneron, Rakich, and Tannehill (37).

II Analysis

Governing Equations

The time-dependent compressible Navier-Stokes (NS) equations describe the conservation of mass, momentum, and energy for a flowing fluid. Ignoring body forces and external heat addition, the equations can be written in non-dimensional, strong conservation-law form as

$$\partial_t \mathbf{Q} + \partial_x \mathbf{E} + \partial_y \mathbf{F} + \partial_z \mathbf{G} = \frac{1}{Re}\left(\partial_x \mathbf{E}_v + \partial_y \mathbf{F}_v + \partial_z \mathbf{G}_v\right) \tag{3}$$

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \tag{4}$$

$$\mathbf{E} = \begin{bmatrix} \rho u \\ \rho uu+p \\ \rho uv \\ \rho uw \\ (e+p)u \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho vv+p \\ \rho vw \\ (e+p)v \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho ww+p \\ (e+p)w \end{bmatrix} \tag{5}$$

$$\mathbf{E}_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{bmatrix}$$

$$\mathbf{F}_v = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} - q_y \end{bmatrix} \tag{6}$$

$$\mathbf{G}_v = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} - q_z \end{bmatrix}$$

where

$$\begin{aligned} \tau_{xx} &= \lambda(u_x + v_y + w_z) + 2\mu u_x \\ \tau_{yy} &= \lambda(u_x + v_y + w_z) + 2\mu v_y \\ \tau_{zz} &= \lambda(u_x + v_y + w_z) + 2\mu w_z \\ \tau_{xy} &= \tau_{yx} = \mu(u_y + v_x) \\ \tau_{xz} &= \tau_{zx} = \mu(u_z + w_x) \\ \tau_{yz} &= \tau_{zy} = \mu(v_z + w_y) \end{aligned} \tag{7}$$

$$\begin{aligned} q_x &= -k\frac{\partial T}{\partial x} = -\frac{C_p\mu}{Pr}\frac{\partial}{\partial x}\left[\frac{(\gamma-1)e_i}{R}\right] = -\frac{\gamma\mu}{Pr}\frac{\partial e_i}{\partial x} \\ q_y &= -k\frac{\partial T}{\partial y} = -\frac{C_p\mu}{Pr}\frac{\partial}{\partial y}\left[\frac{(\gamma-1)e_i}{R}\right] = -\frac{\gamma\mu}{Pr}\frac{\partial e_i}{\partial y} \\ q_z &= -k\frac{\partial T}{\partial z} = -\frac{C_p\mu}{Pr}\frac{\partial}{\partial z}\left[\frac{(\gamma-1)e_i}{R}\right] = -\frac{\gamma\mu}{Pr}\frac{\partial e_i}{\partial z} \end{aligned} \tag{8}$$

$$e_i = \frac{e}{\rho} - \frac{1}{2}(u^2 + v^2 + w^2) \tag{9}$$

The equation of state is used to close the system of equations. For a perfect gas, it can be expressed in terms of the internal energy and the density as

$$p = (\gamma-1)\rho e_i \tag{10}$$

The molecular viscosity (μ) is related to the temperature through Sutherland's formula and has the units of kg/(m·s)

$$\mu = 1.458(10)^{-6}\frac{T^{1.5}}{T + 110.4} \tag{11}$$

The preceding equations have been nondimensionalized using the following relations

$$\begin{array}{lll} x = \dfrac{x^*}{L} & y = \dfrac{y^*}{L} & z = \dfrac{z^*}{L} \\ u = \dfrac{u^*}{a_\infty} & v = \dfrac{v^*}{a_\infty} & w = \dfrac{w^*}{a_\infty} \\ p = \dfrac{p^*}{\rho_\infty a_\infty^2} & e = \dfrac{e^*}{\rho_\infty a_\infty^2} & t = \dfrac{t^*}{(L/a_\infty)} \\ \mu = \dfrac{\mu^*}{\mu_\infty} & \rho = \dfrac{\rho^*}{\rho_\infty} & Re = \dfrac{\rho_\infty a_\infty L}{\mu_\infty} \end{array} \tag{12}$$

where the dimensional variables are denoted by an asterisk, freestream conditions by ∞, and L is the body reference length. Stokes's hypothesis is used to relate λ to μ with λ equal to $-2/3\ \mu$ (38:160-161).

Coordinate Transformation

In order to make the governing equations applicable for arbitrary geometries, they are transformed into generalized coordinates (ξ,η,ζ,τ) by means of the following transformation

$$\begin{aligned} \tau &= t \\ \xi &= \xi(x,y,z,t) \\ \eta &= \eta(x,y,z,t) \\ \zeta &= \zeta(x,y,z,t) \end{aligned} \tag{13}$$

Pulliam (38:160-161) has shown that the transformed equations written in strong conservation-law form and in terms of the contravariant velocities (U,V,W) are given by

$$\partial_\tau \widehat{\mathbf{Q}} + \partial_\xi \widehat{\mathbf{E}} + \partial_\eta \widehat{\mathbf{F}} + \partial_\zeta \widehat{\mathbf{G}} = \frac{1}{\mathrm{Re}}\left(\partial_\xi \widehat{\mathbf{E}}_v + \partial_\eta \widehat{\mathbf{F}}_v + \partial_\zeta \widehat{\mathbf{G}}_v\right) \tag{14}$$

$$\widehat{\mathbf{Q}} = J^{-1}\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \tag{15}$$

$$\widehat{\mathbf{E}} = J^{-1}\begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ (e+p)U + \xi_t p \end{bmatrix} \quad \widehat{\mathbf{F}} = J^{-1}\begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ (e+p)V + \eta_t p \end{bmatrix} \quad \widehat{\mathbf{G}} = J^{-1}\begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ (e+p)W + \zeta_t p \end{bmatrix} \tag{16}$$

$$\widehat{\mathbf{E}}_v = J^{-1}\begin{bmatrix} 0 \\ \xi_x \tau_{xx} + \xi_y \tau_{xy} + \xi_z \tau_{xz} \\ \xi_x \tau_{yx} + \xi_y \tau_{yy} + \xi_z \tau_{yz} \\ \xi_x \tau_{zx} + \xi_y \tau_{zy} + \xi_z \tau_{zz} \\ \xi_x \beta_x + \xi_y \beta_y + \xi_z \beta_z \end{bmatrix}$$

$$\widehat{\mathbf{F}}_v = J^{-1}\begin{bmatrix} 0 \\ \eta_x \tau_{xx} + \eta_y \tau_{xy} + \eta_z \tau_{xz} \\ \eta_x \tau_{yx} + \eta_y \tau_{yy} + \eta_z \tau_{yz} \\ \eta_x \tau_{zx} + \eta_y \tau_{zy} + \eta_z \tau_{zz} \\ \eta_x \beta_x + \eta_y \beta_y + \eta_z \beta_z \end{bmatrix} \tag{17}$$

$$\widehat{G}_v = J^{-1} \begin{bmatrix} 0 \\ \zeta_x\tau_{xx} + \zeta_y\tau_{xy} + \zeta_z\tau_{xz} \\ \zeta_x\tau_{yx} + \zeta_y\tau_{yy} + \zeta_z\tau_{yz} \\ \zeta_x\tau_{zx} + \zeta_y\tau_{zy} + \zeta_z\tau_{zz} \\ \zeta_x\beta_x + \zeta_y\beta_y + \zeta_z\beta_z \end{bmatrix}$$

$$\begin{aligned} \beta_x &= \gamma\mu Pr^{-1}\,\partial_x e_i + u\tau_{xx} + v\tau_{xy} + w\tau_{xz} \\ \beta_y &= \gamma\mu Pr^{-1}\,\partial_y e_i + u\tau_{yx} + v\tau_{yy} + w\tau_{yz} \\ \beta_z &= \gamma\mu Pr^{-1}\,\partial_z e_i + u\tau_{zx} + v\tau_{zy} + w\tau_{zz} \end{aligned} \tag{18}$$

$$\begin{aligned} U &= \xi_t + \xi_x u + \xi_y v + \xi_z w \\ V &= \eta_t + \eta_x u + \eta_y v + \eta_z w \\ W &= \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w \end{aligned} \tag{19}$$

The details of the transformation to generalized coordinates is contained in Appendix A.

Thin-Layer Approximation

The thin-layer approximation applied to the NS equations consists of neglecting the viscous derivatives parallel to the body and retaining the derivatives normal to the body. This is reasonable for high Reynolds number flows, as long as the flow is attached or only mildly separated (38:160). The approximation assumes that the body is mapped on to a ζ=constant plane, and the viscous derivatives in the ξ and η directions are neglected. All the convective terms are retained along with the unsteady terms. Applying the thin-layer approximation to Equations 14-17, Pulliam and Steger have shown that the equations can be written as

$$\partial_\tau\widehat{Q} + \partial_\xi\widehat{E} + \partial_\eta\widehat{F} + \partial_\zeta\widehat{G} = \frac{1}{Re}\partial_\zeta\widehat{S} \tag{20}$$

where **Q**, **E**, **F**, and **G** are the same as before. The viscous flux terms parallel to the body are neglected ($\partial_\xi \mathbf{E}_v = \partial_\eta \mathbf{F}_v = 0$) and the viscous flux term normal to the body is given by

$$\hat{S} = J^{-1} \begin{bmatrix} 0 \\ \mu m_1 u_\zeta + (\mu/3) m_2 \zeta_x \\ \mu m_1 v_\zeta + (\mu/3) m_2 \zeta_y \\ \mu m_1 w_\zeta + (\mu/3) m_2 \zeta_z \\ \mu m_1 m_3 + (\mu/3)\, m_2 m_4 \end{bmatrix} \tag{21}$$

$$\begin{aligned} m_1 &= \zeta_x^2 + \zeta_y^2 + \zeta_z^2 \\ m_2 &= \zeta_x u_\zeta + \zeta_y v_\zeta + \zeta_z w_\zeta \\ m_3 &= \frac{1}{2}\partial_\zeta\left(u^2 + v^2 + w^2\right) + \gamma \mathrm{Pr}^{-1} \partial_\zeta (e_i) \\ m_4 &= \zeta_x u + \zeta_y v + \zeta_z w \end{aligned} \tag{22}$$

The thin-layer Navier-Stokes (TLNS) equations are a mixed set of hyperbolic-parabolic partial differential equations (PDE) in time. These have the same form as the NS equations, and can be solved using the same type of methods. If the unsteady terms are dropped, the equations become a mixed set of hyperbolic-elliptic equations. These equations are more difficult to solve and therefore most solutions for the compressible NS equations have used the unsteady form. The steady-state solution is obtained by marching the solution in time until converged, and is known as the time-dependent approach. The time-dependent approach can be explicit or implicit and is usually second-order accurate in space. If time accuracy is desired, the algorithm should be second-order accurate in time. For steady-state calculations, a first-order accurate algorithm can be used to accelerate convergence (39:424,482).

Boundary Conditions

The pressure at the body surface can be found using a normal pressure momentum relation. Pulliam has shown (38:161) by combining the three transformed momentum equations, the normal pressure is given by

$$P_n\left(\zeta_x^2 + \zeta_y^2 + \zeta_z^2\right) = \rho\left(\partial_\tau \zeta_t + u\partial_\tau \zeta_x + v\partial_\tau \zeta_y + w\partial_\tau \zeta_z\right) - \rho U\left(\zeta_x u_\xi + \zeta_y v_\xi + \zeta_z w_\xi\right) - \rho V\left(\zeta_x u_\eta + \zeta_y v_\eta + \zeta_z w_\eta\right) \quad (23)$$

and the Cartesian velocities can be determined from

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = J^{-1} \begin{bmatrix} (\eta_y\zeta_z - \eta_z\zeta_y) & -(\xi_y\zeta_z - \xi_z\zeta_y) & (\xi_y\eta_z - \xi_z\eta_y) \\ -(\eta_x\zeta_z - \eta_z\zeta_x) & (\xi_x\zeta_z - \xi_z\zeta_x) & -(\xi_x\eta_z - \xi_z\eta_x) \\ (\eta_x\zeta_y - \eta_y\zeta_x) & -(\xi_x\zeta_y - \xi_y\zeta_x) & (\xi_x\eta_y - \xi_y\eta_x) \end{bmatrix} \begin{bmatrix} U - \xi_t \\ V - \eta_t \\ -\zeta_t \end{bmatrix} \quad (24)$$

The above expressions are valid for inviscid or viscous flows and for steady or unsteady body motion.

The no slip boundary condition is specified at the body surface, where U=V=W=0 and u,v, and w are determined from Equation 24. In the case of a stationary grid where $\xi_t=\eta_t=\zeta_t=0$, u,v, and w are zero. The surface pressure is calculated by integrating Equation 23. For viscous flow over a stationary grid, Equation 23 reduces to the pressure gradient normal to the surface being zero ($P_n=0$). The body is assumed to be adiabatic ($T_n=0$) and the density is found by linear extrapolation from the point above. The farfield boundary conditions are specified to be freestream values, except for the downstream condition. A first-order extrapolation was used as the downstream boundary condition. A plane of symmetry was imposed at the centerline.of the body, where

$$\frac{\partial \rho}{\partial \eta} = \frac{\partial \rho u}{\partial \eta} = \frac{\partial \rho w}{\partial \eta} = \frac{\partial e}{\partial \eta} = v = 0 \qquad \text{at } \eta = 0 \tag{25}$$

Explicit boundary conditions are used for their ease in application.

III Numerical Procedure

The numerical procedure used to obtain solutions of the governing equations is outlined. Topics include the choice of algorithm, and formulation of the finite-difference equations. Emphasis is placed on detailing the simplifications used (approximate factorization and diagonalization) and their impact on code efficiency and accuracy. The implementation of the nonlinear dissipation model is discussed. The solution process for one time step is outlined for running the code, in core or out of core with the Solid-State Disk (SSD).

Implicit Time Marching Algorithm

A time marching finite-difference scheme, developed by Beam and Warming (40:118-120), is used to solve the thin-layer Navier-Stokes equations. This is an alternating-direction implicit (ADI) scheme and is similar to schemes developed by Lindemuth and Killeen (41) and McDonald and Briley (42). The scheme uses an implicit, three-point, time-differencing formula in the form

$$\Delta\hat{Q}^n = \frac{\vartheta\Delta t}{1+\varphi}\frac{\partial}{\partial t}\left(\Delta\hat{Q}^n\right) + \frac{\Delta t}{1+\varphi}\frac{\partial}{\partial t}\left(\hat{Q}^n\right) + \frac{\varphi}{1+\varphi}\left(\Delta\hat{Q}^{n-1}\right) + O\left[\left(\vartheta - \frac{1}{2} - \varphi\right)(\Delta t)^2 + (\Delta t)^3\right] \quad (26)$$

the parameters ϑ and φ can be chosen to produce a scheme which is either first- or second-order accurate in time. Since we are primarily interested in steady-state solutions, a first-order scheme is chosen. If $\vartheta=1$ and $\varphi=0$, this results in the Euler implicit scheme (39:490)

$$\Delta\hat{Q}^n = \Delta t\frac{\partial}{\partial t}\left(\Delta\hat{Q}^n\right) + \Delta t\frac{\partial}{\partial t}\left(\hat{Q}^n\right) + O\left[(\Delta t)^2\right] \quad (27)$$

and $\Delta\hat{Q}^n = \hat{Q}^{n+1} - \hat{Q}^n$, thus Equation 27 can be rewritten as

$$\Delta\hat{Q}^n - \Delta t\frac{\partial}{\partial t}\left(\hat{Q}^{n+1}\right) = 0 \quad (28)$$

Writing Equation 20 at the n+1 time level yields

$$\partial_\tau \hat{Q}^{n+1} = -\partial_\xi \hat{E}^{n+1} - \partial_\eta \hat{F}^{n+1} - \partial_\zeta \hat{G}^{n+1} + \frac{1}{Re}\partial_\zeta \hat{S}^{n+1} \tag{29}$$

Finally, substitution of Equation 29 into Equation 28 yields

$$\Delta \hat{Q}^n + \Delta t\left(\partial_\xi \hat{E}^{n+1} + \partial_\eta \hat{F}^{n+1} + \partial_\zeta \hat{G}^{n+1} - \frac{1}{Re}\partial_\zeta \hat{S}^{n+1}\right) = 0 \tag{30}$$

<u>Linearization</u>

For Equation 30 to be solved for **Q** the flux vectors, **E**, **F**, **G**, and **S**, which are nonlinear functions of **Q**, must be linearized. The inviscid flux vectors can be linearized in time by using a Taylor series about Q^n (32:12):

$$\begin{aligned}
\hat{E}^{n+1} &= \hat{E}^n + \frac{\partial \hat{E}^n}{\partial \hat{Q}^n}\Delta \hat{Q}^n = \hat{E}^n + \hat{A}^n \Delta \hat{Q}^n \\
\hat{F}^{n+1} &= \hat{F}^n + \frac{\partial \hat{F}^n}{\partial \hat{Q}^n}\Delta \hat{Q}^n = \hat{F}^n + \hat{B}^n \Delta \hat{Q}^n \\
\hat{G}^{n+1} &= \hat{G}^n + \frac{\partial \hat{G}^n}{\partial \hat{Q}^n}\Delta \hat{Q}^n = \hat{G}^n + \hat{C}^n \Delta \hat{Q}^n
\end{aligned} \tag{31}$$

Steger has shown (43:3-4) that the viscous flux vector can be linearized using a Taylor series and is given by

$$\hat{S}^{n+1} = \hat{S}^n + J^{-1}\frac{\partial \hat{S}^n}{\partial \hat{Q}^n} J\,\Delta \hat{Q}^n = \hat{S}^n + J^{-1}\hat{M}^n J\,\Delta \hat{Q}^n \tag{32}$$

The flux Jacobians are given in Appendix B. Substituting Equations 31 and 32 into Equation 30 yields the "delta form" of the algorithm

$$\left[I + \Delta t \partial_\xi \hat{A}^n + \Delta t \partial_\eta \hat{B}^n + \Delta t \partial_\zeta \hat{C}^n - Re^{-1} \Delta t \partial_\zeta J^{-1} \hat{M}^n J\right] \Delta \hat{Q}^n = -\Delta t \left(\partial_\xi \hat{E}^n + \partial_\eta \hat{F}^n + \partial_\zeta \hat{G}^n - Re^{-1} \partial_\zeta \hat{S}^n\right) \tag{33}$$

In Equation 33 that the left-hand-side contains the unknown ΔQ and is sometimes referred to as the "implicit" part. The right-hand-side contains the known quantities and is referred to as the "explicit" part (32:13). Applying second-order, central-difference operators to equation 33 yields the final form of the time marching algorithm:

$$\left[I + \Delta t \delta_\xi \hat{A}^n + \Delta t \delta_\eta \hat{B}^n + \Delta t \delta_\zeta \hat{C}^n - Re^{-1} \Delta t \delta_\zeta J^{-1} \hat{M}^n J\right] \Delta \hat{Q}^n = -\Delta t \left(\delta_\xi \hat{E}^n + \delta_\eta \hat{F}^n + \delta_\zeta \hat{G}^n - Re^{-1} \delta_\zeta \hat{S}^n\right) \tag{34}$$

<u>Approximate Factorization</u>

To integrate the full three-dimensional operator would be prohibitively expensive. One simplification that can be used is to approximately factor the three-dimensional operator into three one-dimensional operators. If terms on the $O(\Delta t^2)$ are neglected the left-hand-side of Equation 34 is factored into (32:79)

$$\left[I + \Delta t \delta_\xi \hat{A}^n\right]\left[I + \Delta t \delta_\eta \hat{B}^n\right]\left[I + \Delta t \delta_\zeta \hat{C}^n - Re^{-1} \Delta t \delta_\zeta J^{-1} \hat{M}^n J\right] \Delta \hat{Q}^n \tag{35}$$

Neglecting the $O(\Delta t^2)$ terms will not degrade the accuracy of the scheme since it is first order accurate in time. Equation 34 can now be written using approximate factorization as

$$\left[I + \Delta t \delta_\xi \hat{A}^n\right]\left[I + \Delta t \delta_\eta \hat{B}^n\right]\left[I + \Delta t \delta_\zeta \hat{C}^n - Re^{-1} \Delta t \delta_\zeta J^{-1} \hat{M}^n J\right] \Delta \hat{Q}^n =$$

$$-\Delta t\left(\delta_\xi \hat{E}^n + \delta_\eta \hat{F}^n + \delta_\zeta \hat{G}^n - Re^{-1}\,\delta_\zeta \hat{S}^n\right) \qquad (36)$$

The approximate factorization reduces the (Jmax·Kmax·Lmax·5) x (Jmax·Kmax·Lmax·5) banded matrix down to a set of three block tridiagonal matrices. The size of any matrix is now at most (max | Jmax,Kmax,Lmax |·5) x (max | Jmax,Kmax,Lmax |·5). The solution now consists of three sweeps, one in the ξ direction, one in the η direction, and one in the ζ direction. Each block tridiagonal matrix can be solved using a block lower-upper decomposition (LUD) (32:17).

Diagonalization

Approximate factorization was used to reduce the block three dimensional implicit operator of Equation 34 down to three one-dimensional block tridiagonal matrices. The solution of these matrices are still computationally complex, since they involve the solution of 5x5 blocks. One way to decrease the work is to decouple the equations. If the operators are diagonalized, the block structure is decomposed into five scalar operators.

Diagonalization of the Euler equations is presented and then the method is extended to the Navier-Stokes equations. This is due to the fact that " the viscous flux Jacobian M^n is not simultaneously diagonalizable with the flux Jacobian C^n (32:19)." Warming, Beam. and Hyett (44:1037-1041) have shown that the inviscid flux Jacobians can be diagonalized since they have real eigenvalues and a complete set of eigenvectors. The flux Jacobians are written in terms of their eigenvalues and eigenvectors as

$$\begin{aligned} \hat{\mathbf{A}}^n &= \left(\mathbf{T}_\xi \Lambda_\xi \mathbf{T}_\xi^{-1}\right)^n \\ \hat{\mathbf{B}}^n &= \left(\mathbf{T}_\eta \Lambda_\eta \mathbf{T}_\eta^{-1}\right)^n \qquad (37) \\ \hat{\mathbf{C}}^n &= \left(\mathbf{T}_\zeta \Lambda_\zeta \mathbf{T}_\zeta^{-1}\right)^n \end{aligned}$$

where T_ξ is the eigenvector matrix of **A** and likewise, T_η for **B**, and T_ζ for **C**. The eigenvector and eigenvalue matrices are written out in Appendix C. Equation 36 can now be written neglecting the viscous flux Jacobian as

$$\left[\left(T_\xi T_\xi^{-1}\right)^n + \Delta t\delta_\xi\left(T_\xi\Lambda_\xi T_\xi^{-1}\right)^n\right]\left[\left(T_\eta T_\eta^{-1}\right)^n + \Delta t\delta_\eta\left(T_\eta\Lambda_\eta T_\eta^{-1}\right)^n\right]$$
$$\times\left[\left(T_\zeta T_\zeta^{-1}\right)^n + \Delta t\delta_\zeta\left(T_\zeta\Lambda_\zeta T_\zeta^{-1}\right)^n\right]\Delta\hat{Q}^n = \hat{R}^n \qquad (38)$$

where

$$\hat{R}^n = -\Delta t\left(\delta_\xi\hat{E}^n + \delta_\eta\hat{F}^n + \delta_\zeta\hat{G}^n - Re^{-1}\delta_\zeta\hat{S}^n\right) \qquad (39)$$

The eigenvector matrices are factored outside of the difference operators and this yields the "diagonal" form of the algorithm.

$$T_\xi^n\left[I + \Delta t\delta_\xi\Lambda_\xi^n\right]\hat{N}^n\left[I + \Delta t\delta_\eta\Lambda_\eta^n\right]\hat{P}^n\left[I + \Delta t\delta_\zeta\Lambda_\zeta^n\right]\left(\hat{T}_\zeta^{-1}\right)^n\Delta\hat{Q}^n = \hat{R}^n \qquad (40)$$

where

$$\hat{N}^n = \left(T_\xi^{-1}T_\eta\right)^n$$
$$\hat{P}^n = \left(T_\eta^{-1}T_\zeta\right)^n \qquad (41)$$

Since the eigenvector matrices are functions of ξ,η, and ζ, factoring them outside of the operator introduces an error. Pulliam and Chaussee (45:356-359) have shown that for steady-state solutions, the accuracy of the solution is not affected. The reason being, for steady-state solutions ΔQ approaches zero. Since the right-hand-side is unchanged by the diagonalization the error is determined by the order of the differencing chosen. They have also shown the time accuracy of the algorithm to be at most first-order.

The above discussion only applies to the Euler equations since the viscous flux Jacobian has been neglected. To extend diagonalization to the Navier-Stokes equations, Pulliam suggests (32:20) including a diagonal term on the implicit side to approximate the viscous Jacobian eigenvalues. The current estimates are

$$\begin{aligned}\lambda_v(\xi) &= \rho\mu Re^{-1}J^{-1}\left(\xi_x^2 + \xi_y^2 + \xi_z^2\right)\\ \lambda_v(\eta) &= \rho\mu Re^{-1}J^{-1}\left(\eta_x^2 + \eta_y^2 + \eta_z^2\right)\\ \lambda_v(\zeta) &= \rho\mu Re^{-1}J^{-1}\left(\zeta_x^2 + \zeta_y^2 + \zeta_z^2\right)\end{aligned} \qquad (42)$$

For the thin-layer approximation the ξ and η terms are ignored and the $\lambda_v(\zeta)$ term is added to the Λ_ζ eigenvalues.

Artificial Dissipation

By the use of linear stability analysis, it can be shown that the diagonalized algorithm, Equation 40, is unconditionally stable. In reality this is not true, especially when the system is nonlinear. "Scales of motion appear which cannot be resolved by the numerics and are due to the nonlinear interactions in the convection terms of the momentum equations (32:26)."

One way of dealing with these numerical instabilities is to add some numerical dissipation to the algorithm which does not alter the accuracy of the solution. The dissipation model chosen is a mixed second order, fourth order model due to Jameson (46). The nonlinear model for the ζ coordinate is given by

$$D_\zeta = \nabla_\zeta\left(\sigma_{j,k,l+1}J^{-1}_{j,k,l+1} + \sigma_{j,k,l}J^{-1}_{j,k,l}\right)\left(\varepsilon^{(2)}_{j,k,l}\Delta_\zeta - \varepsilon^{(4)}_{j,k,l}\Delta_\zeta\nabla_\zeta\Delta_\zeta\right)J \qquad (43)$$

where

$$\varepsilon^{(2)}_{j,k,l} = \kappa_2 \Delta t \max\left(\Upsilon_{j,k,l+1}, \Upsilon_{j,k,l}, \Upsilon_{j,k,l-1}\right)$$

$$\Upsilon_{j,k,l} = \frac{|P_{j,k,l+1} - 2P_{j,k,l} + P_{j,k,l-1}|}{|P_{j,k,l+1} + 2P_{j,k,l} + P_{j,k,l-1}|} \tag{44}$$

$$\varepsilon^{(4)}_{j,k,l} = \max\left(0, \kappa_4 \Delta t - \varepsilon^{(2)}_{j,k,l}\right)$$

$$\sigma_{j,k,l} = |W| + a\left(\zeta_x^2 + \zeta_y^2 + \zeta_z^2\right)^{1/2}$$

and σ is the spectral radius scaling for C. ∇_ζ and Δ_ζ are the first order accurate backward and forward difference operators. Typical values of the constants κ_2 and κ_4 are .25 and .01, respectively. Similar terms are used for the ξ and η directions. Applying this model to both the implicit and explicit sides of Equation 40 yields

$$T^n_\xi\left[I + \Delta t \delta_\xi \Lambda^n_\xi - \Delta t D_\xi I\right]\widehat{N}^n\left[I + \Delta t \delta_\eta \Lambda^n_\eta - \Delta t D_\eta I\right]\widehat{P}^n\left[I + \Delta t \delta_\zeta \Lambda^n_\zeta - \Delta t D_\zeta I\right]\left(\widehat{T}_\zeta^{-1}\right)^n \Delta\widehat{Q}^n =$$

$$-\Delta t\left(\delta_\xi \widehat{E}^n + \delta_\eta \widehat{F}^n + \delta_\zeta \widehat{G}^n - Re^{-1}\delta_\zeta \widehat{S}^n\right) - \left(D_\xi + D_\eta + D_\zeta\right)\widehat{Q}^n \tag{45}$$

Since the dissipation is a fourth order model, this necessitates the use of scalar penta-diagonal solvers. Although pentadiagonal equations are more complicated to solve than tridiagonal equations, the use of implicit dissipation improves convergence and "robustness" of the algorithm (47:16).

Solution Procedure

Since the algorithm makes use of approximate factorization, the solution can be obtained by sweeping in the ξ,η, and ζ directions sequentially. The solution process becomes a series of matrix-vector multiplies and solutions of scalar pentadiagonal equations.This process consists of the following eight steps at each time level.

1. A matrix-vector multiplication at each grid point.

$$\hat{S}_1 = \left(T_\xi^{-1}\right)^n\left[\hat{R}^n - (D_\xi + D_\eta + D_\zeta)\hat{Q}^n\right]$$

2. The solution of five scalar pentadiagonal equations for the ξ direction.

$$\hat{S}_2 = \left[I + \Delta t\delta_\xi\Lambda_\xi^n - \Delta t D_\xi I\right]^{-1}\hat{S}_1$$

3. A second matrix-vector multiplication at each grid point.

$$\hat{S}_3 = \left(\hat{N}^{-1}\right)^n\hat{S}_2$$

4. The solution of five pentadiagonal equations,this time in the η direction.

$$\hat{S}_4 = \left[I + \Delta t\delta_\eta\Lambda_\eta^n - \Delta t D_\eta I\right]^{-1}\hat{S}_3$$

5. A third matrix-vector multiplication at each grid point.

$$\hat{S}_5 = \left(\hat{P}^{-1}\right)^n\hat{S}_4$$

6. The solution of five scalar pentadiagonal equations, in the ζ direction.

$$\hat{S}_6 = \left[I + \Delta t\delta_\zeta\Lambda_\zeta^n - \Delta t D_\zeta I\right]^{-1}\hat{S}_5$$

7. The final matrix-vector multiplication at each grid point.

$$\Delta\hat{Q}^n = (T_\zeta)^n\hat{S}_6$$

8. The solution is then updated by

$$\hat{Q}^{n+1} = \hat{Q}^n + \Delta\hat{Q}^n$$

Solid State Disk Implementation

The calculations were done on a Cray XMP-12 computer using a Solid State Disk (SSD). The SSD was used because of the large grid and large amounts of storage required for the solution of a three dimensional problem.

The solution process using the SSD, entails two sweeps of the domain for each time step. The first sweep is for planes of constant η (k planes), where the differencing in the ζ direction is done. The second sweep is for planes of constant ζ (l planes), where the ξ and η differencing is done.

The SSD is set up with two working files. The first file contains the metrics and the second file contains the flow variables. At each plane the metrics and the flow variables are read into the in-core memory. The calculations are carried out and the updated flow variables are loaded back on to the SSD. There is no need to unload the metrics since they are invarient with time. The algorithm advances to the next plane and the process is repeated.

Using the SSD is one way of running problems which require more core memory than may be available. The trade off is that the code is not as efficient as a code that runs totally in core. Appendix D contains a comparison of Central Processor Unit (CPU) times, input/output (I/O), and storage required to run the code in core and out of core using the SSD.

IV Results and Discussion

The diagonalized algorithm is used to calculate the flow over a 75° delta wing and a 80°/69° cranked delta wing at angle of attack. The delta wing calculations are compared with the experimental data of Monnerie and Werle (35) and the numerical results of Buter and Rizzetta (34). The cranked wing solution is compared with experimental data of Henke (31). The calculations were done on the Cray XMP-12 computer at Wright-Patterson Air Force Base (AFB), Ohio. The information and files required to run this code at the Air Force Institute of Technology, are contained in Appendix E.

Delta Wing

Since this version of the code had not been previously run using the Cray at Wright-Patterson AFB, it was necessary to insure there were no hardware or system software incompatibilities. A 75° delta wing at a freestream Mach number of 1.95 and angle of attack of 10° was chosen as the test case. This configuration was tested by Monnerie and Werle (35) and the model geometry is given in Figure 3. Numerical solutions using the full Navier-Stokes equations and the conical approximation to the Navier-Stokes equations are contained in references 31-34. Comparisons will be made with the experimental data and with the numerical results of Buter and Rizzetta (34).

Due to the simple geometry of the delta wing, the grid was generated using an algebraic grid generator. The grid consisted of 30 points in the streamwise (ξ) direction, 55 points in the spanwise (η) direction and 65 points in the normal (ζ) direction. A typical streamwise grid plane (η,ζ) is shown in Figure 4. The grid incorporates a branch cut between the upper and lower body surfaces to allow for the use of an H-grid. This approach was used to eliminate the small radius of curvature of the grid, at the wingtip, that results from using a C grid. Since the freestream Mach number is supersonic, the upstream propagation of disturbances is extremely limited. This implies that there is no requirement

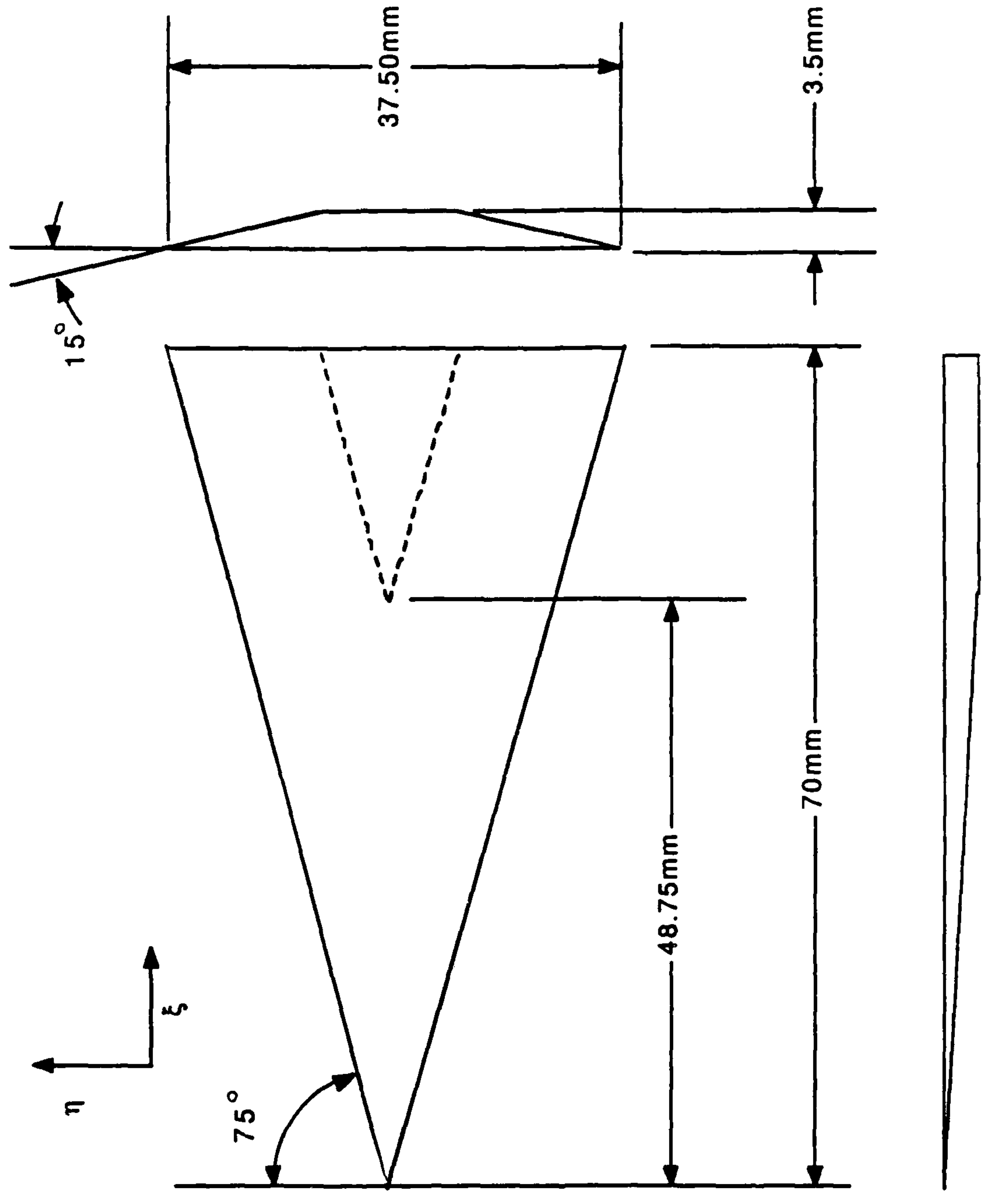


Figure 3. 75° Delta Wing Geometry

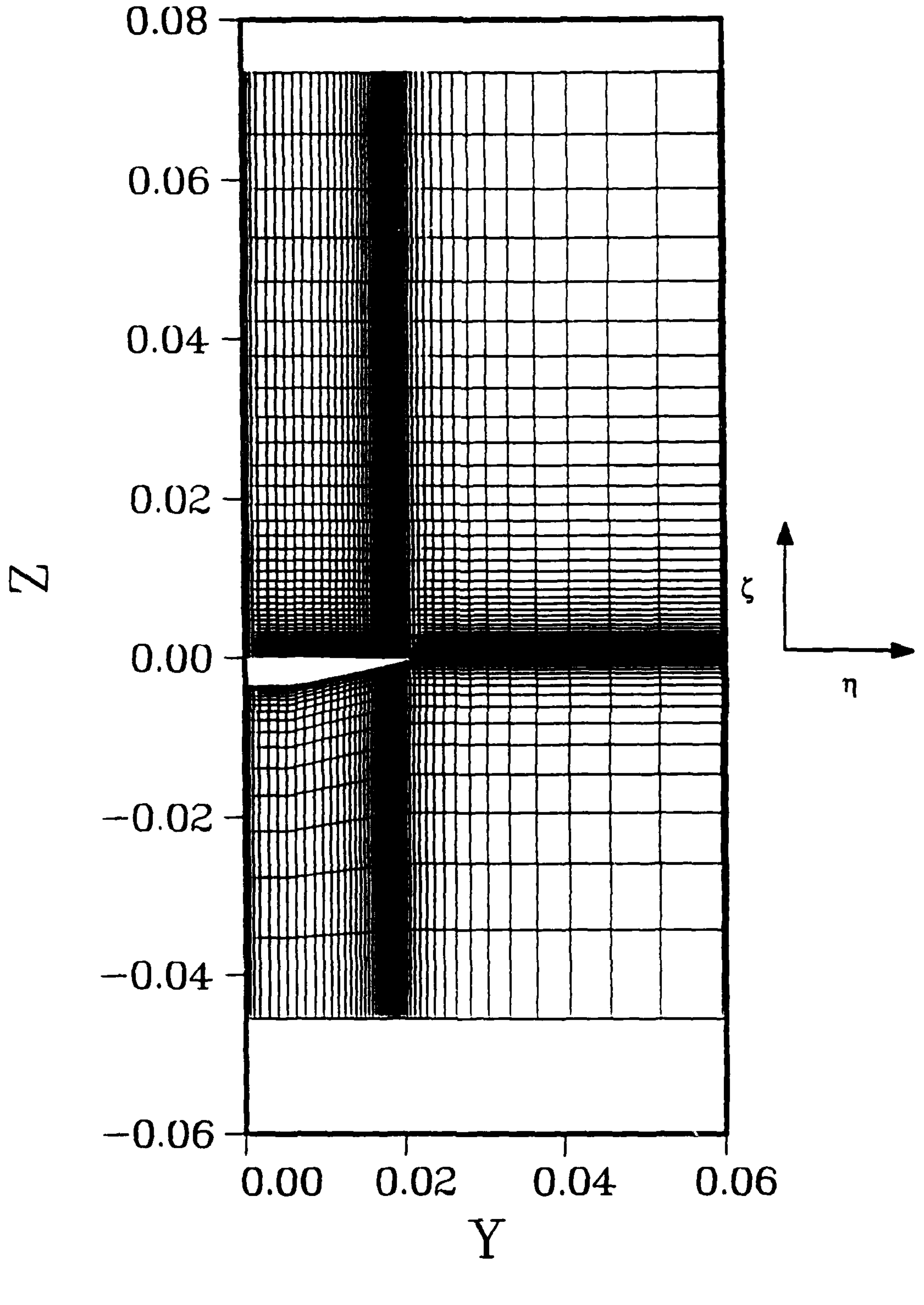
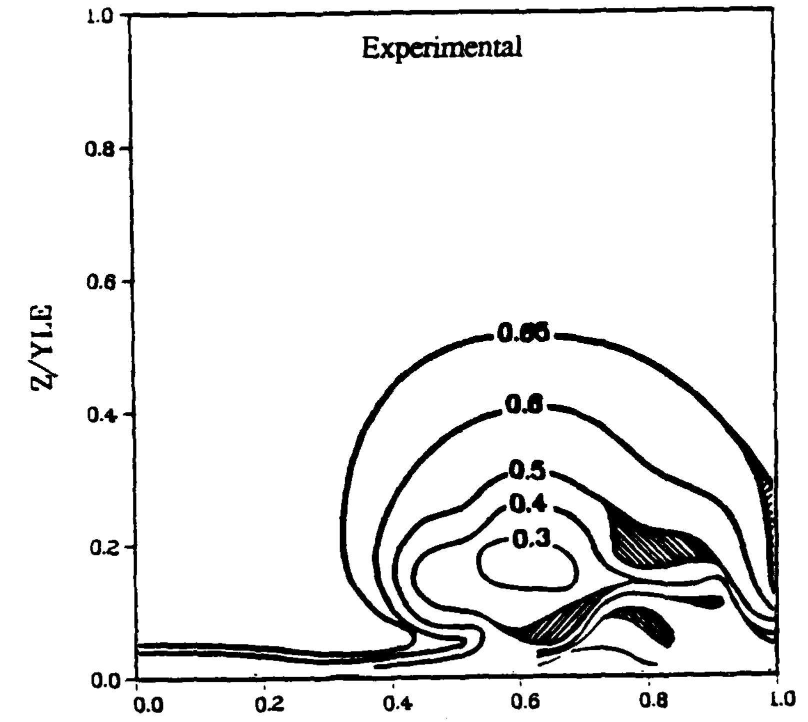


Figure 4. Trailing Edge Grid Plane for Delta Wing

for a wake region and the domain ends at the wing trailing edge. Orthogonality is not enforced at the boundaries, but previous work (33,34) has shown that good results can be obtained using this approach. Points were clustered in the high gradient areas such as, the nose, the wingtip and at the body surface, by the use of exponential functions.

Figure 5 shows the comparison of Pitot pressures with the experimental data.The Pitot pressures have been normalized by the freestream Pitot pressure. The thin-layer Navier-Stokes calculations show good agreement on the location and the strength of the primary vortex. The comparison with the work of Buter and Rizzetta are shown in Figure 6. Again good agreement is seen with the full Navier-Stokes calculations. As was seen in the previous comparison, the thin-layer calculations exhibit more total pressure loss, but the crossplane velocities and the wing surface pressures agree well. The present study exhibits a little less development in the secondary separation region than do the full Navier-Stokes solutions as seen in the surface pressure plot.

The streamwise development of the leading edge vortex is given in Figure 7. Traveling from the nose to the trailing edge, the vortex is seen to strengthen. This is marked by a decrease in pressure of the primary vortex core and an increase in the crossplane velocities. Also from the crossplane velocities, the formation and strengthening of the secondary separation is seen, although this is not significant until aft of 60 percent of the wing length. Upon closer examination of the crossplane velocities, it is noted that the center of the vortex and gross features of the flow do not change rapidly in the streamwise direction. Although the flow is not truly conical, it is changing slowly enough that ignoring the viscous effect will not significantly affect the overall flow field structure. The same trend can be seen in the surface pressure plots of Figure 8. The primary vortex is seen to increase in strength with aft movement.



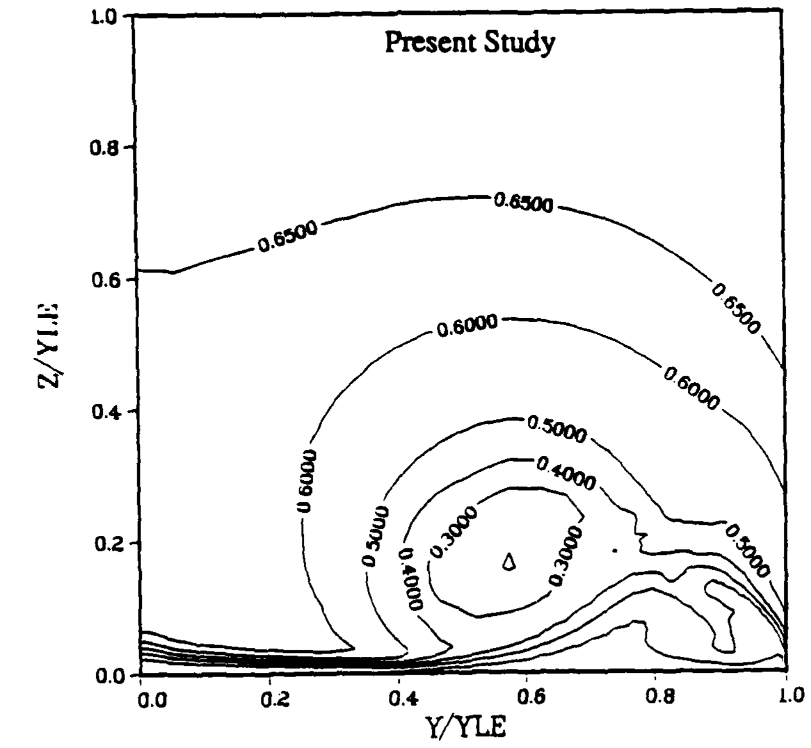
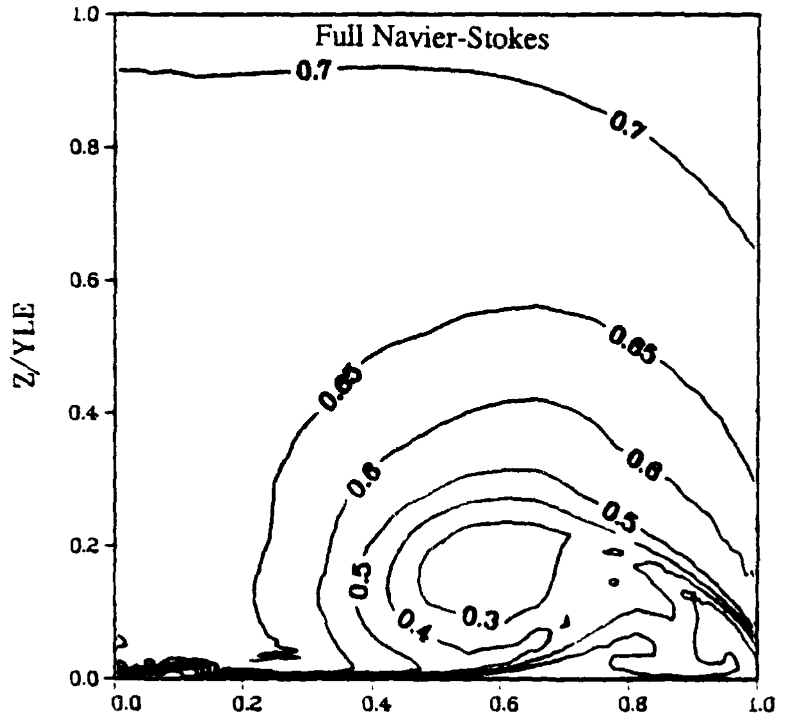


Figure 5. Comparison with Experimental Pitot Pressures at x/L=0.8



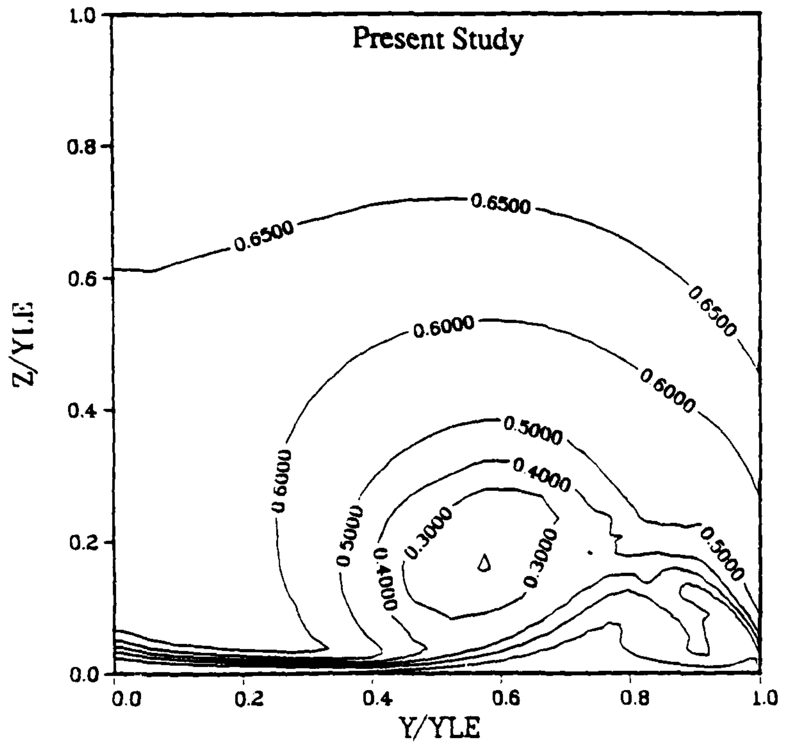
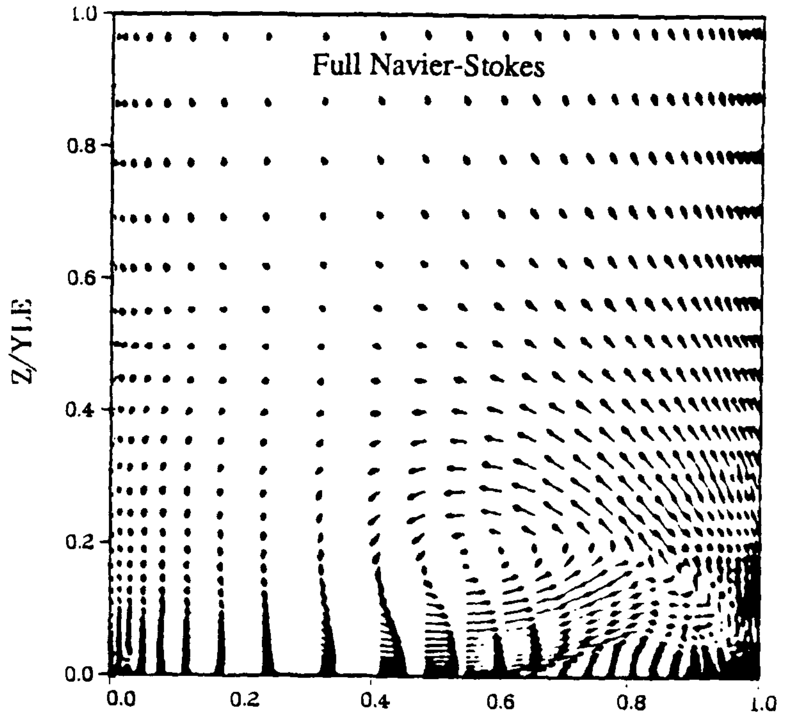


Figure 6. Comparison with Full Navier-Stokes Solutions,
Pitot Pressures at x/L=0.8



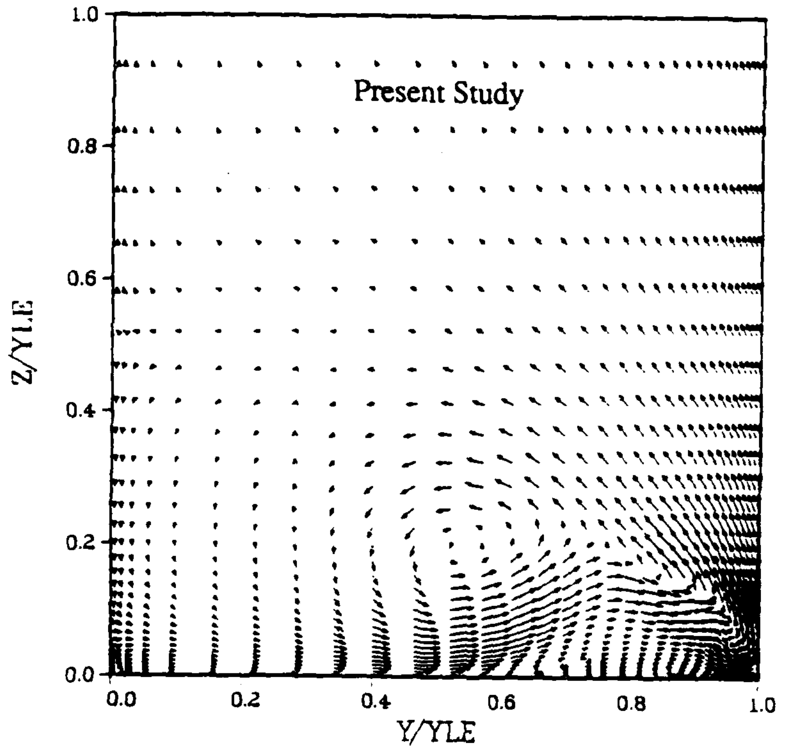


Figure 6. (cont) Crossplane Velocities at x/L=0.8

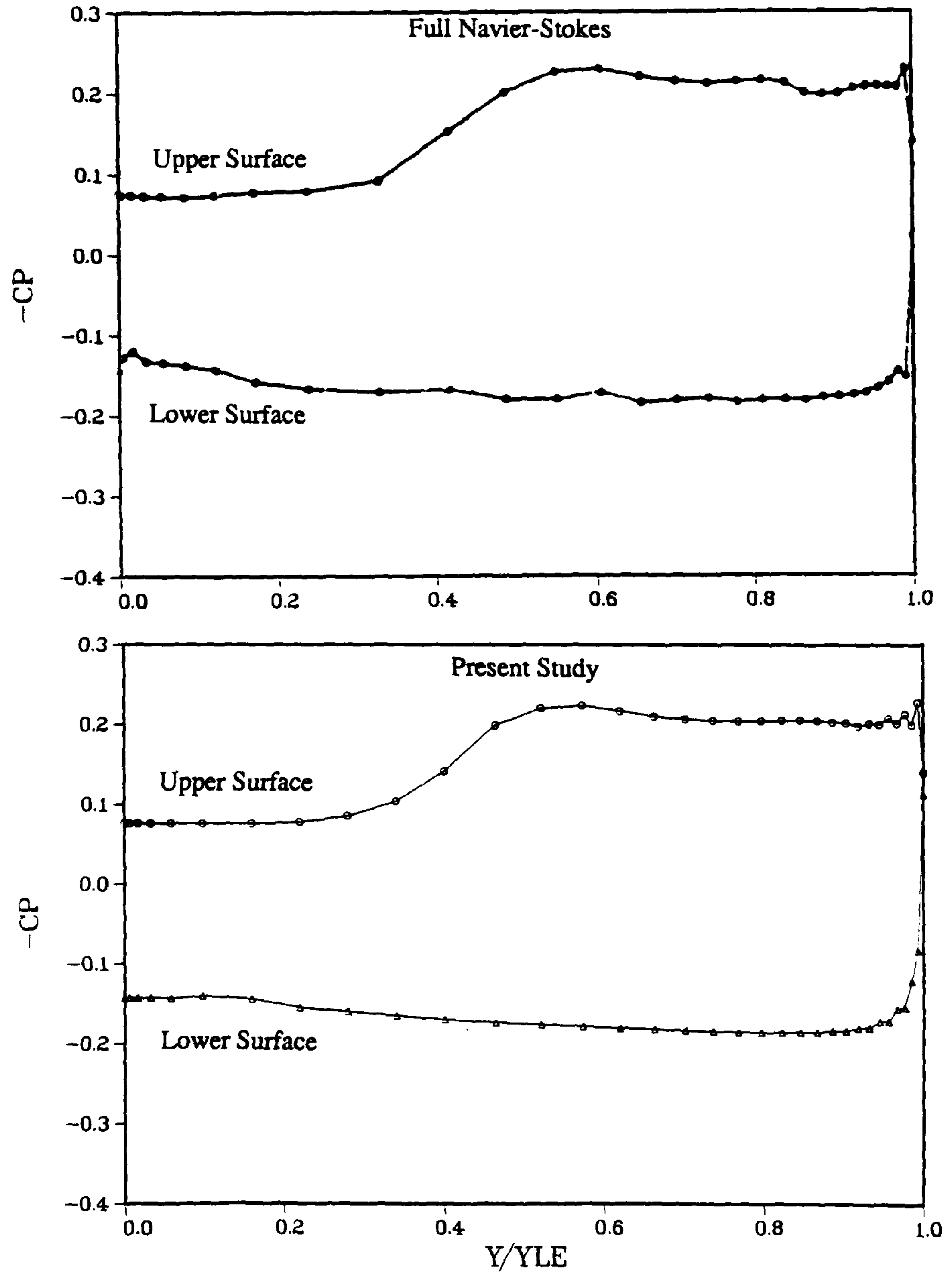


Figure 6. (cont) Surface Pressures at x/L=0.8

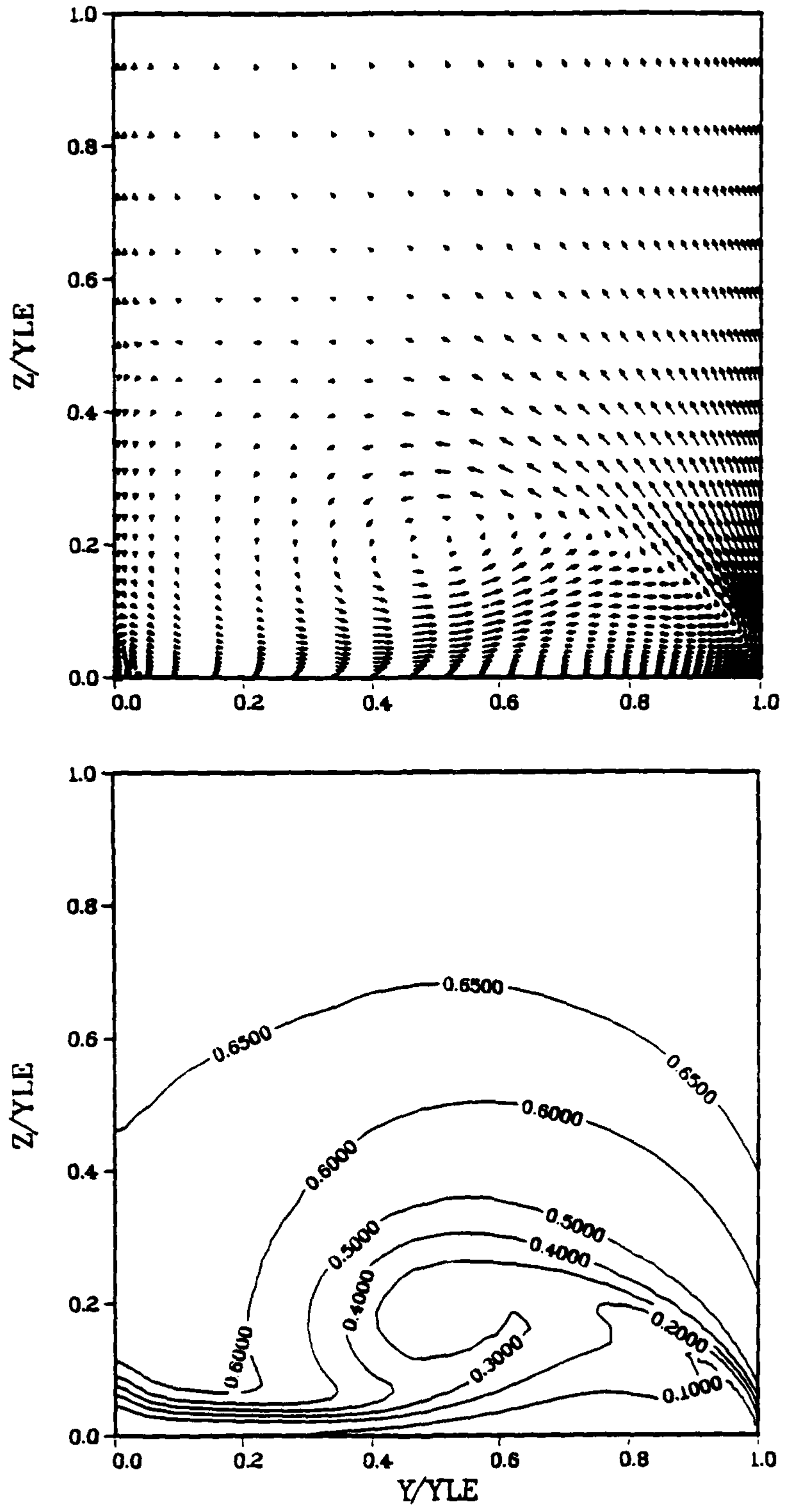


Figure 7. Pitot pressures and Crossplane Velocities for the Delta Wing at x/L=0.2

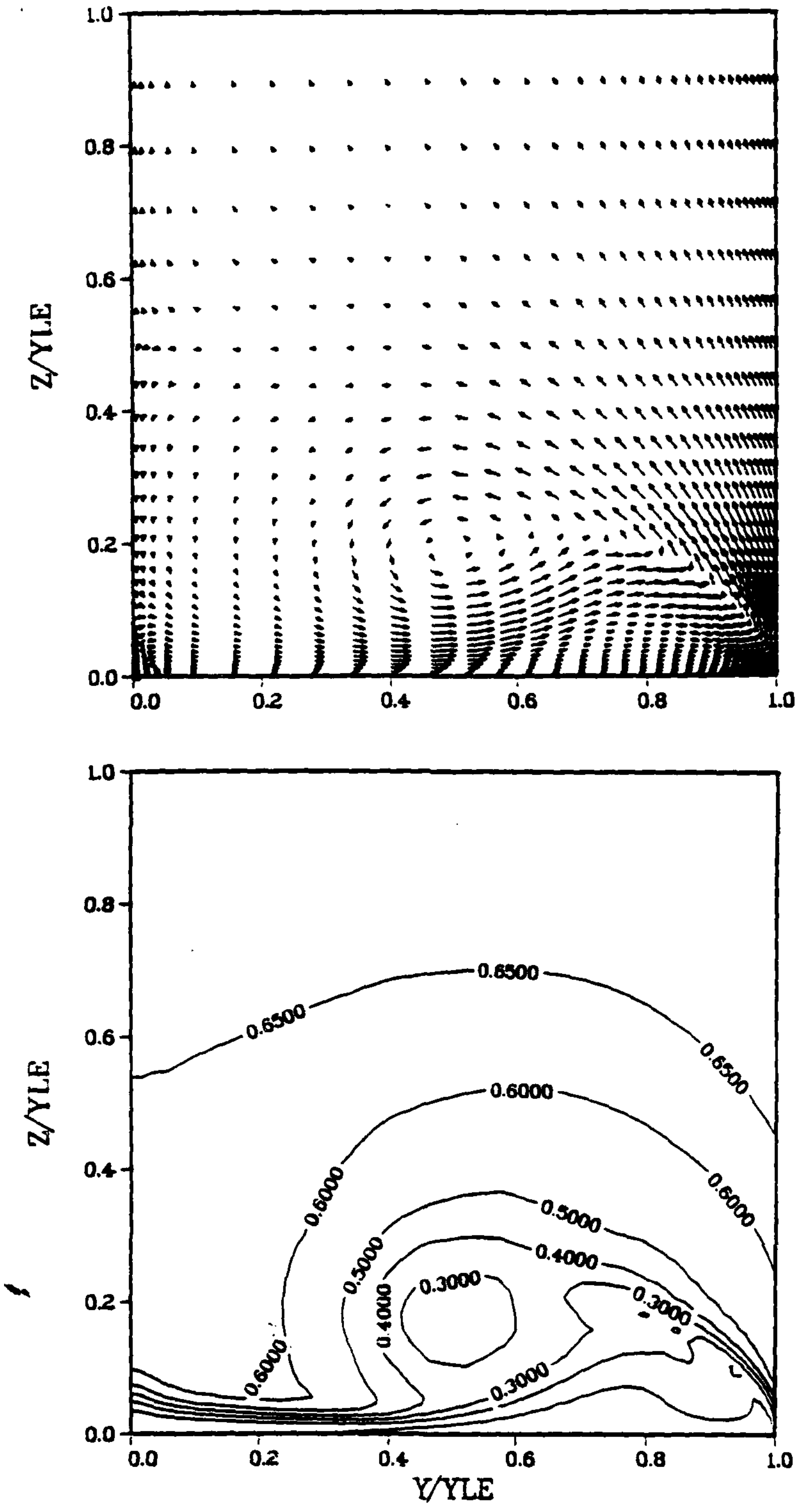


Figure 7. (cont) x/L=0.4

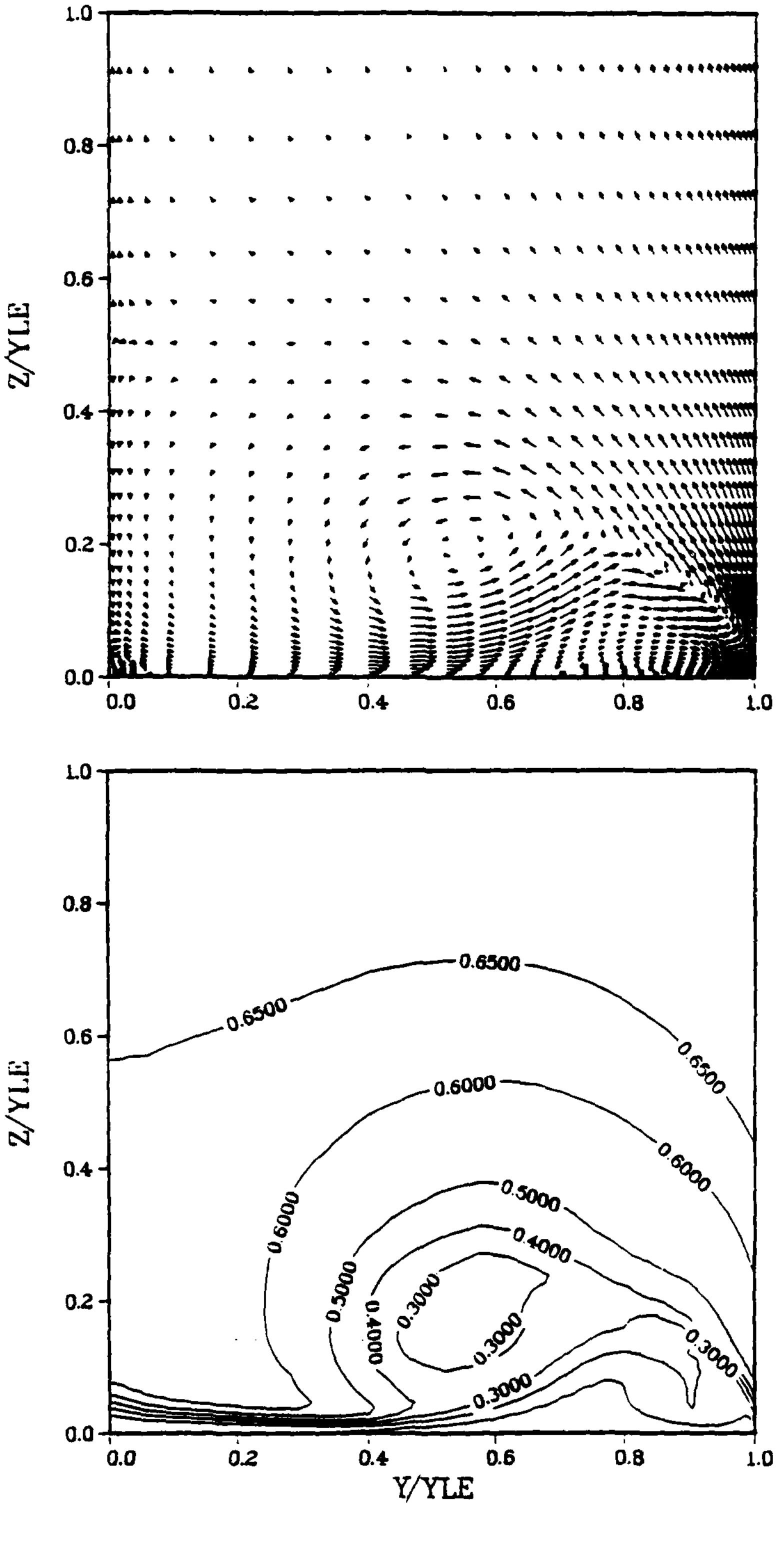
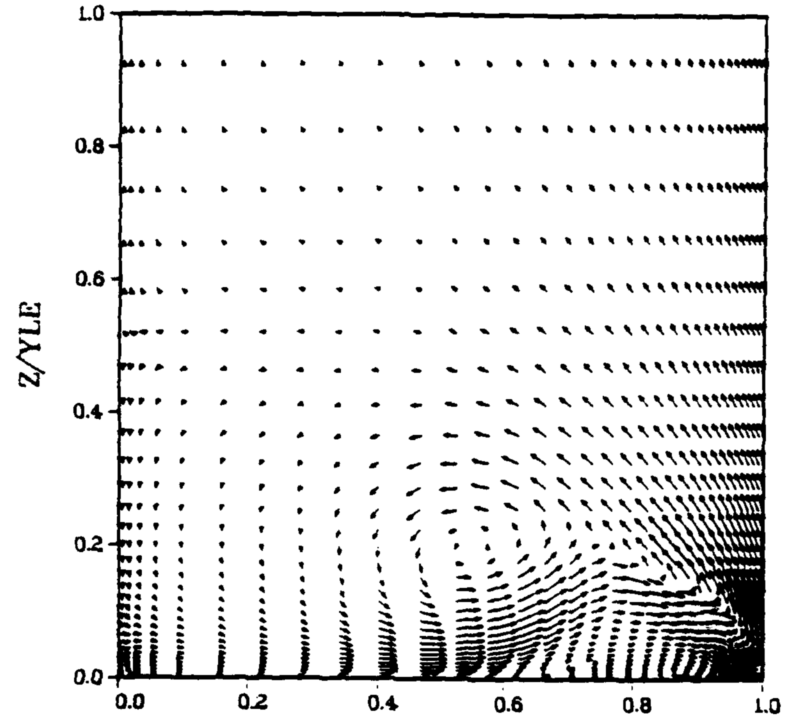


Figure 7. (cont) x/L=0.6



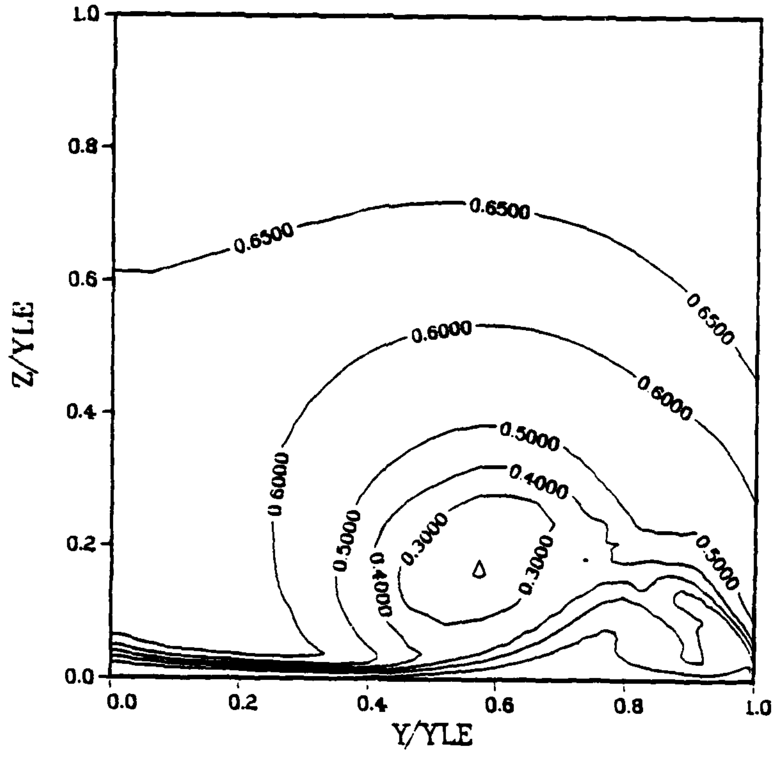
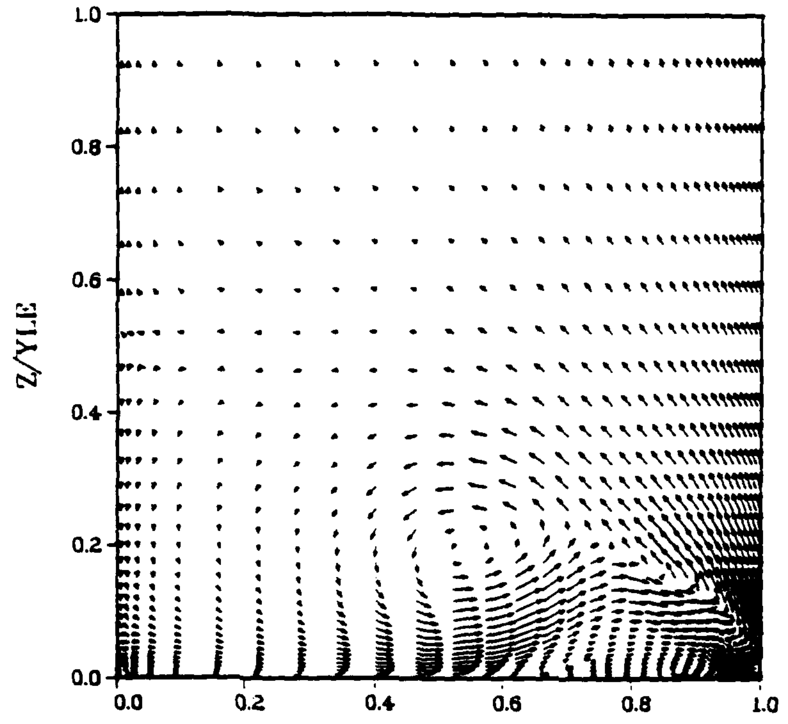


Figure 7. (cont) x/L=0.8



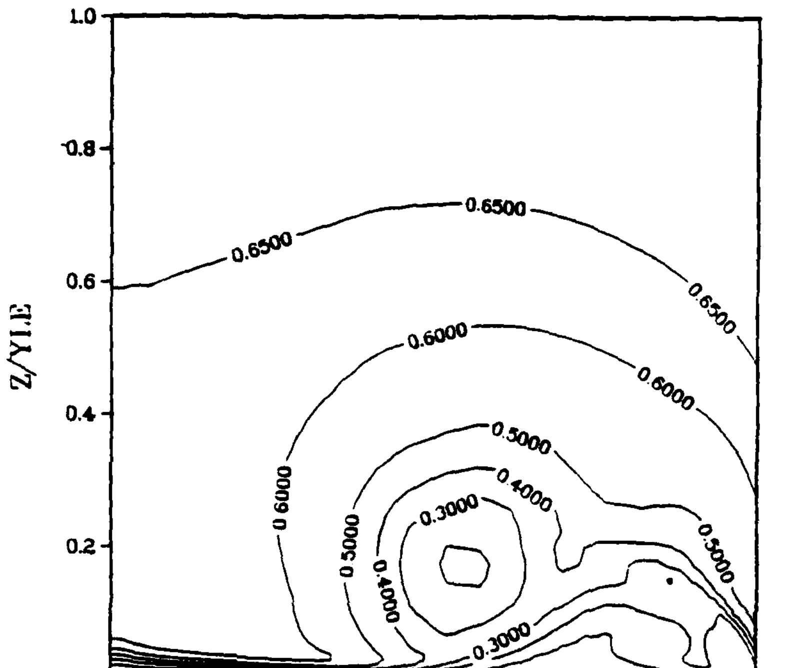


Figure 7. (cont) x/L=1.0

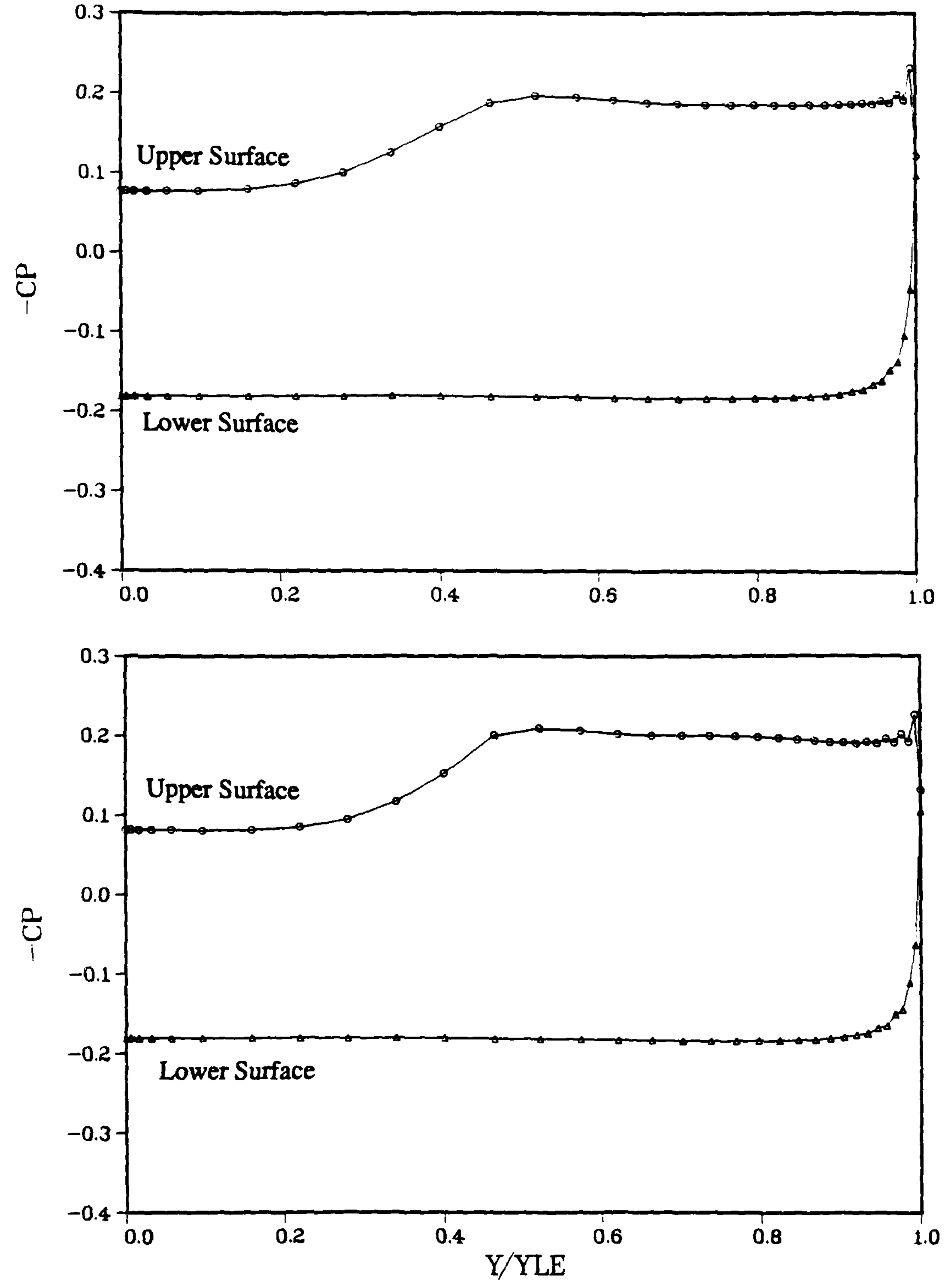
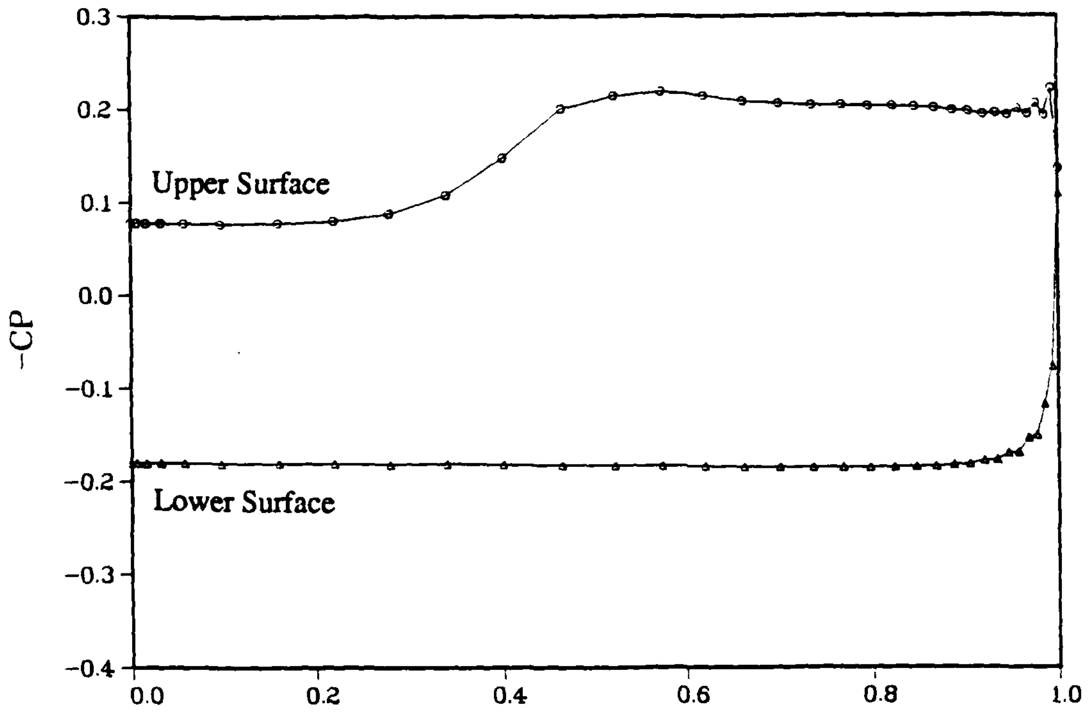


Figure 8. Delta Wing Surface Pressures
x/l=0.2 (upper), x/l=0.4 (lower)



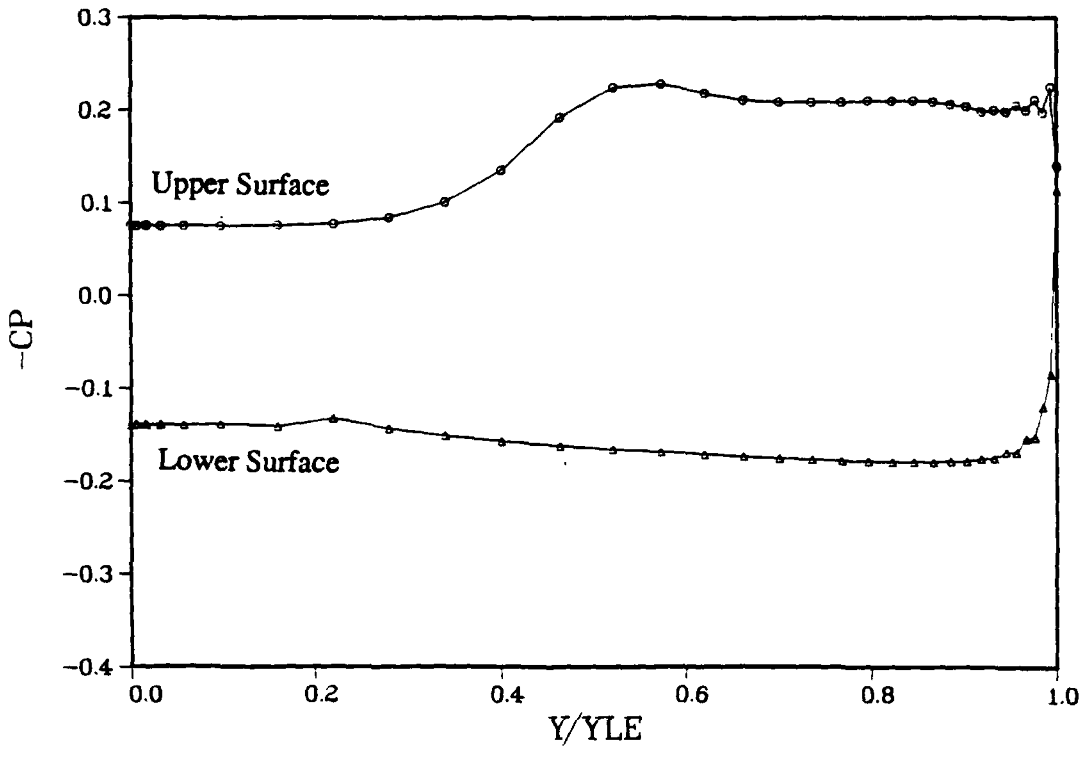


Figure 8. (cont) x/l=0.6 (upper), x/l=1.0 (lower)

Cranked Delta Wing

The configuration chosen for this study was that tested by Henke (31) and is shown in Figure 9. It has an 80° leading edge sweep forward of the crank and is followed by a 69° sweep. The crank occurs at 40 percent of the body length. The wing has a triangular cross section with a flat upper surface and sharp leading edges. The angle between the upper and lower surfaces is 40° and this results in a relatively thick body at the trailing edge.

An algebraic grid generator was also used to fit a grid around the cranked delta wing. The grid generator was originally written to generate grids for delta wings and has been modified by the author to generate grids for cranked delta wings. Appendix F contains the listing of the grid generator used for these calculations. The grid consists of 127,400 points, 28 in the streamwise direction, 65 in the spanwise direction, and 70 normal to the body. A typical streamwise grid plane is shown in Figure 10. As can be seen from the figure, points have been clustered at the body surface and the wing tip. The grid consists of 41 points on the wing surface in the spanwise direction and 25 points in the streamwise direction. In addition to clustering at the nose, points were clustered in the vicinity of the wing crank. This was necessary because of the large changes in the flow, resulting from the start up of the crank vortex. As before a branch cut is used to allow for an H grid topology. Since the freestream Mach number is supersonic, there is no significant signal propagation from the wake region, therefore the domain ends at the wing trailing edge.

The first plane of the grid on the body is located at $x/l=0.05$ from the apex of the delta wing. It was found that placing a plane of data farther forward than this caused the program to crash. It is thought that the points become compressed in the η-ζ plane and this causes large unsmooth variations in the metrics. The correction to this problem was to move the initial plane rearward and open up the η-ζ plane.

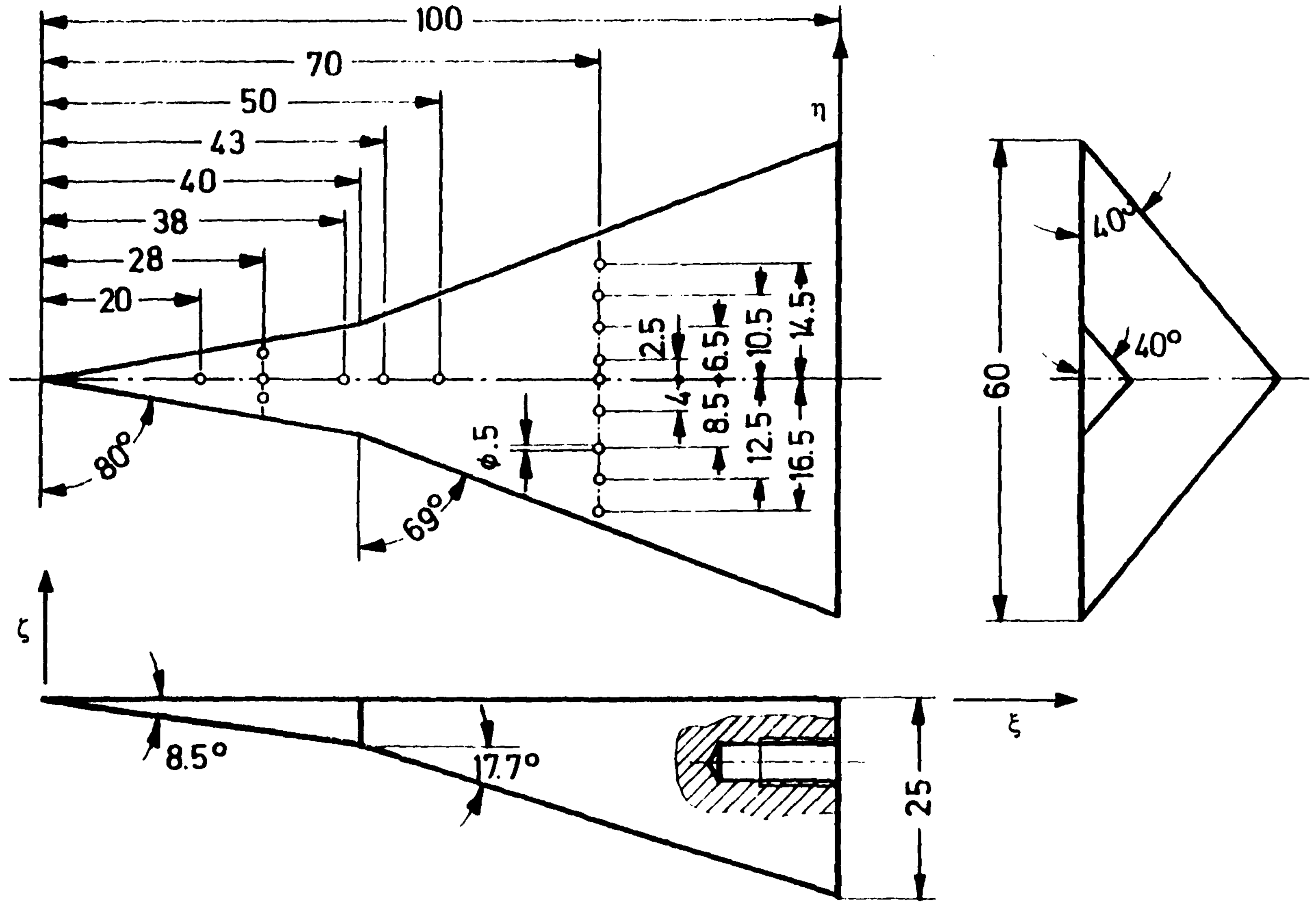


Figure 9. 80°/69° Cranked Delta Wing Geometry

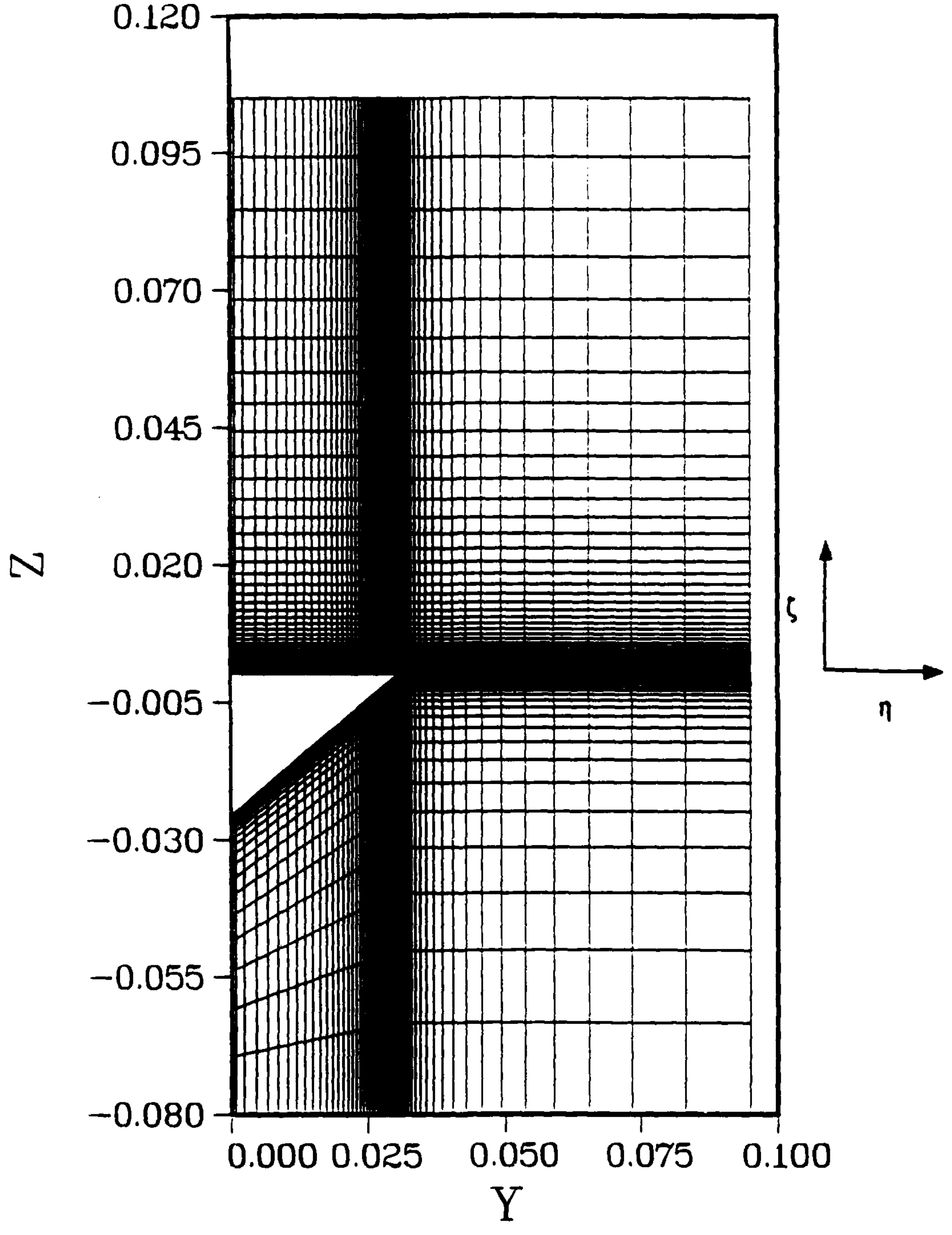


Figure 10. Trailing Edge Grid Plane for Cranked Delta Wing

The configuration chosen was for a freestream Mach number of 2.5 and an angle of attack of 10 degrees. This angle of attack was chosen because it was thought to be high enough to develop stable vortical flow. The freestream Reynolds number is 686,000 and the wind tunnel stagnation conditions are 10 psia and 20^{o} C.

Figure 11 shows the development of the Pitot pressures and cross plane velocities along the wing. The flow over the wing in front of the crank is similar to that found on straight delta wings. From the crossplane velocity plot, for a streamwise location of x/l=0.398, it is seen that the center of the primary vortex is located at approximately 50 percent of the wing span. The location of the vortex center remains fairly constant with streamwise location. This was also true for the flow over the 75° delta wing that was shown previously. In addition to the primary vortex structure, there was also a secondary structure located at about 85 percent of the wing span.

Moving aft of the crank, a second vortex forms at the wing crank. This second vortex can be seen in the plots of Figure 11, at a wing location of x/L=0.414. The vortex center is located at approximately 85 percent of the wing span. The area of secondary separation seen in front of the crank, has started to diminish. Moving aft on the wing (x/L=0.453) shows no evidence of the second vortex. The vortex generated at the crank appears to become paired with the first vortex and the secondary vortex has now completely disappeared as a seperate identity. Finally, looking at a plane well aft of the crank , x/L=0.88, an area of low Pitot pressure has formed between the first vortex and the wing leading edge.

The vortex location is seen not to change with streamwise location in front of the crank but this is not true aft of the crank. Figure 12 shows a comparison between the flow in front of the crank and well downstream of the crank. Aft of the crank the vortex center is seen to move inboard and towards the wing surface. The area of secondary separation present in the flow in front of the crank has been replaced by an area of low Pitot pressure.

The surface pressures show that the overall magnitude of the pressure is approximately the same for both planes. The primary difference is that the flow over the cranked portion must negotiate a larger pressure gradient than the flow in front of the crank.

Figure 13 shows a comparison of the upper surface pressures with experimental data. The upper surface pressure exhibits the same trend as the experimental data but is about 10 percent less for the outboard wing section. It appears that the grid may be too coarse in the spanwise and normal directions to adequately resolve all the details of the flow field.

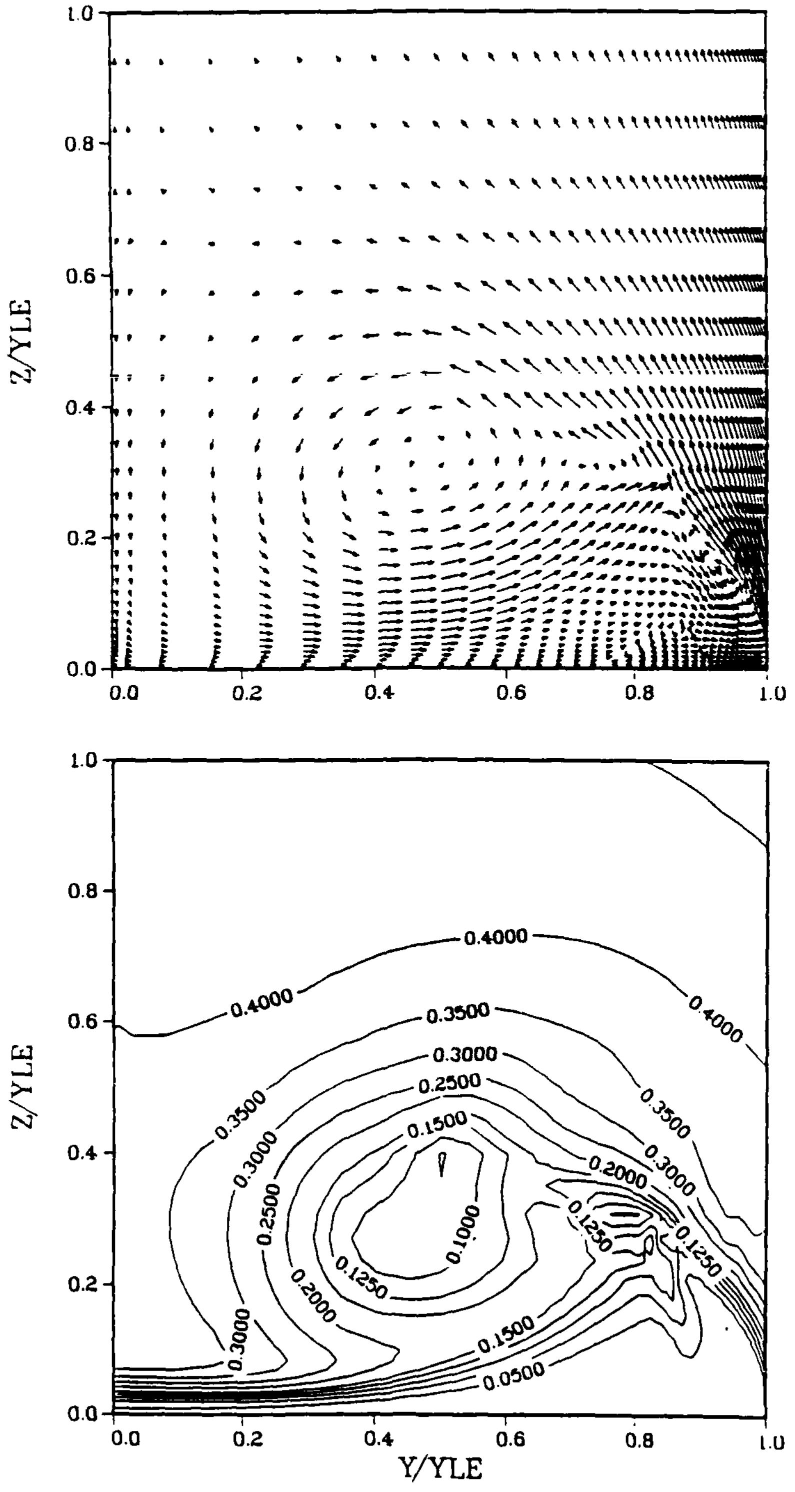
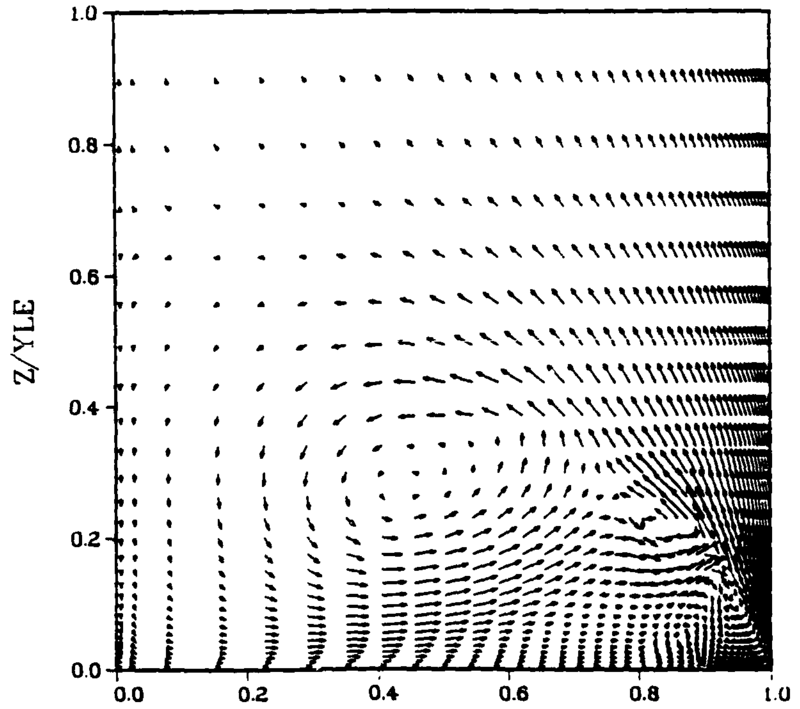


Figure 11. Pitot Pressures and Crossplane Velocities for the Cranked Delta Wing, x/L=0.398



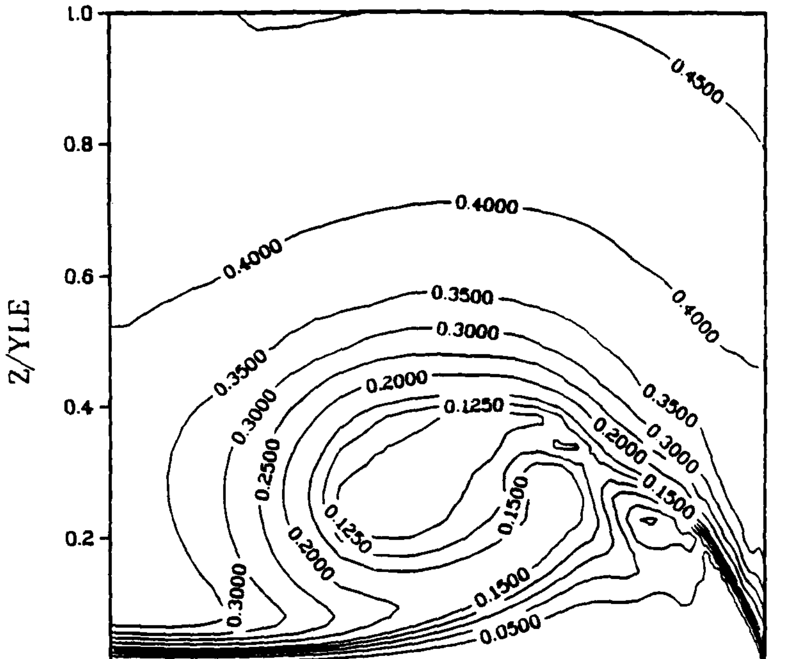
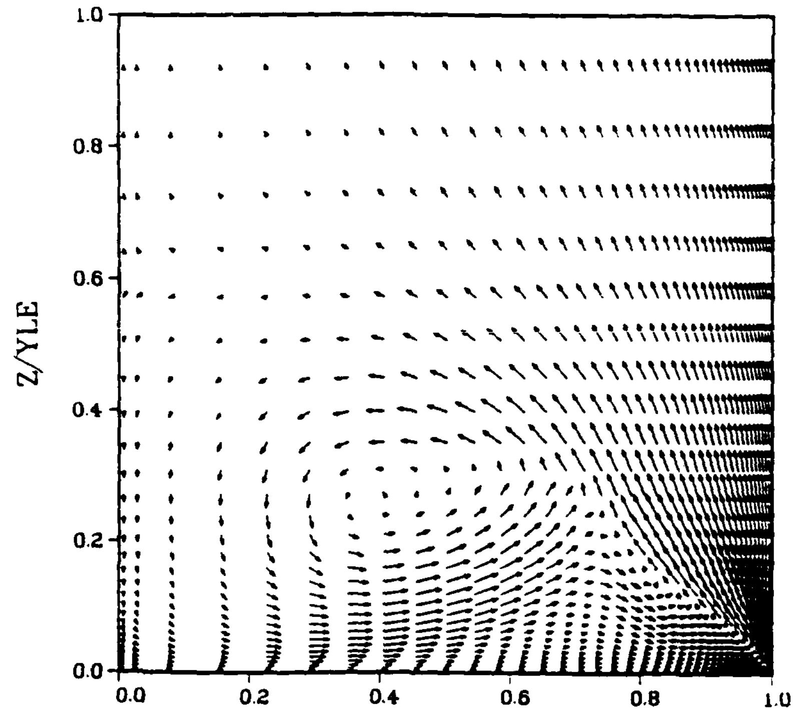


Figure 11. (cont) x/L=0.414



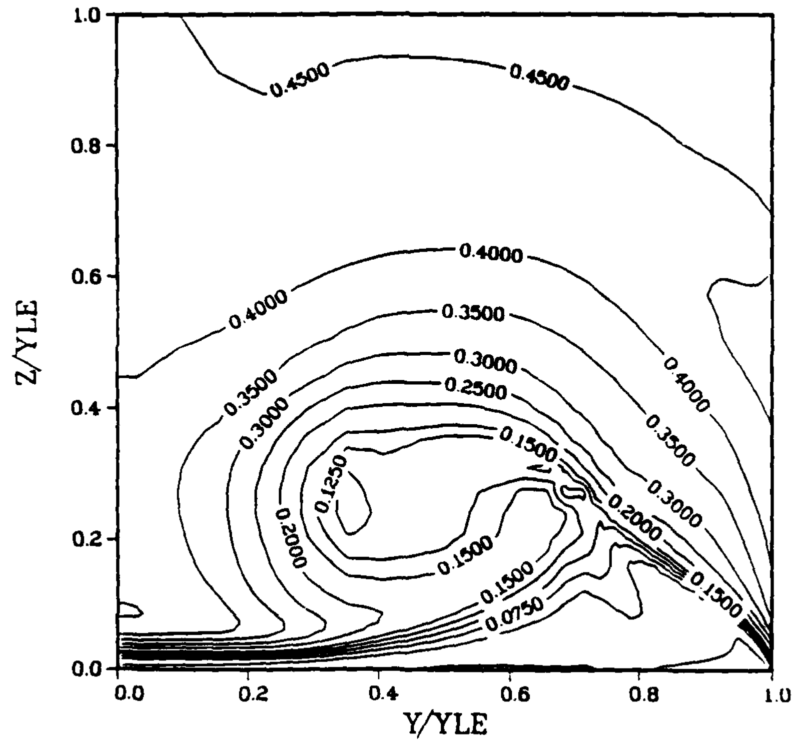
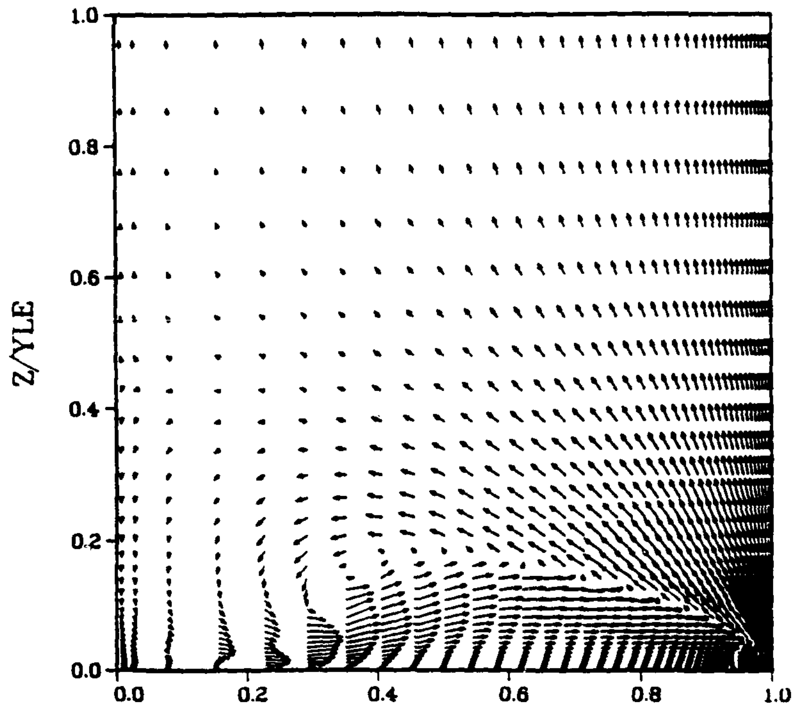


Figure 11. (cont) x/L=0.453



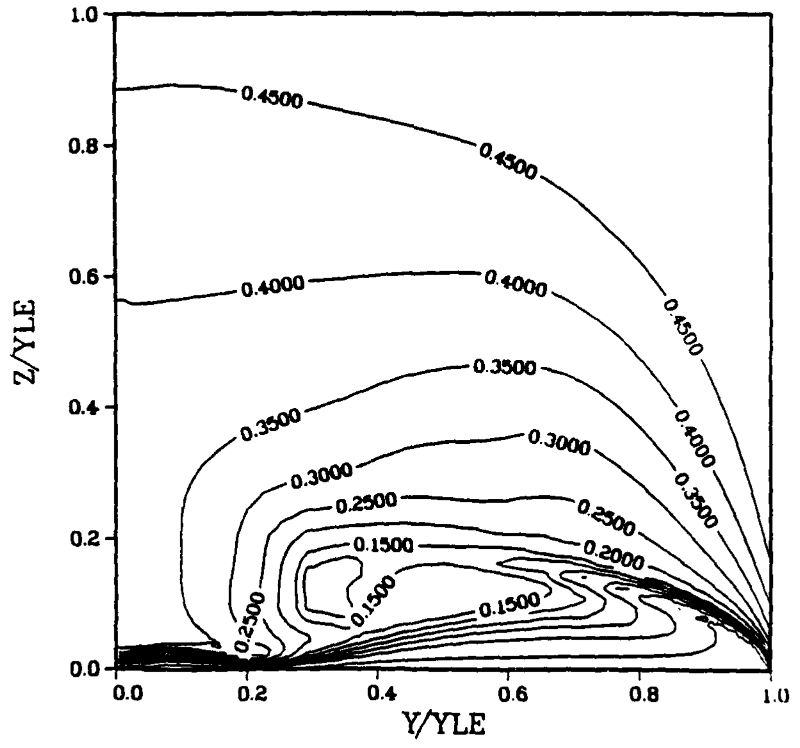
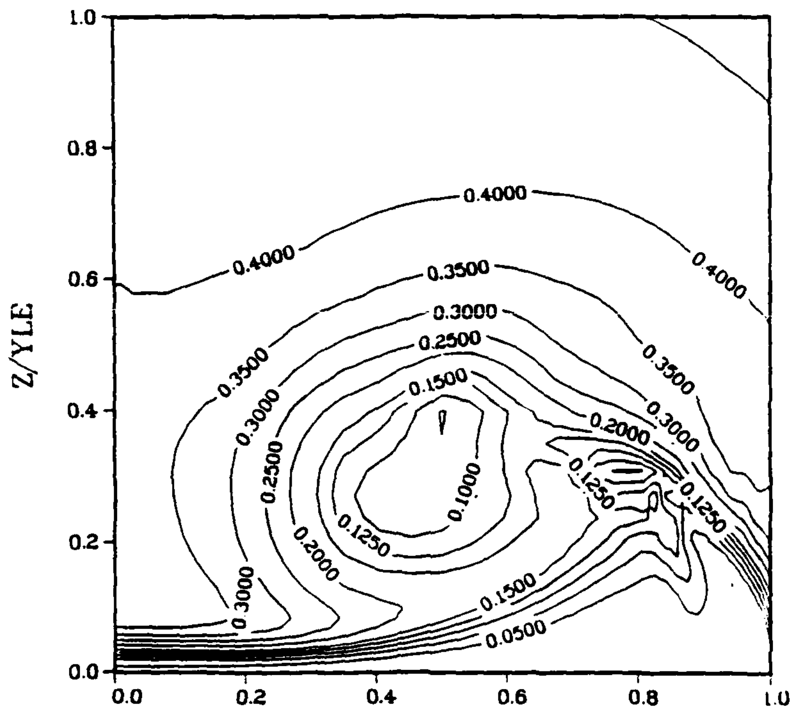


Figure 11. (cont) x/L=0.880



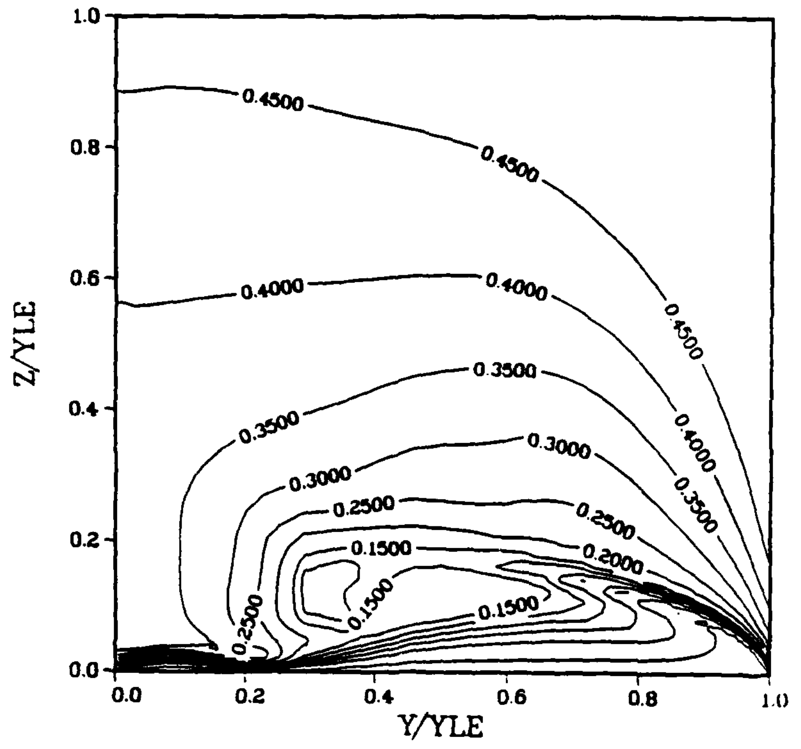
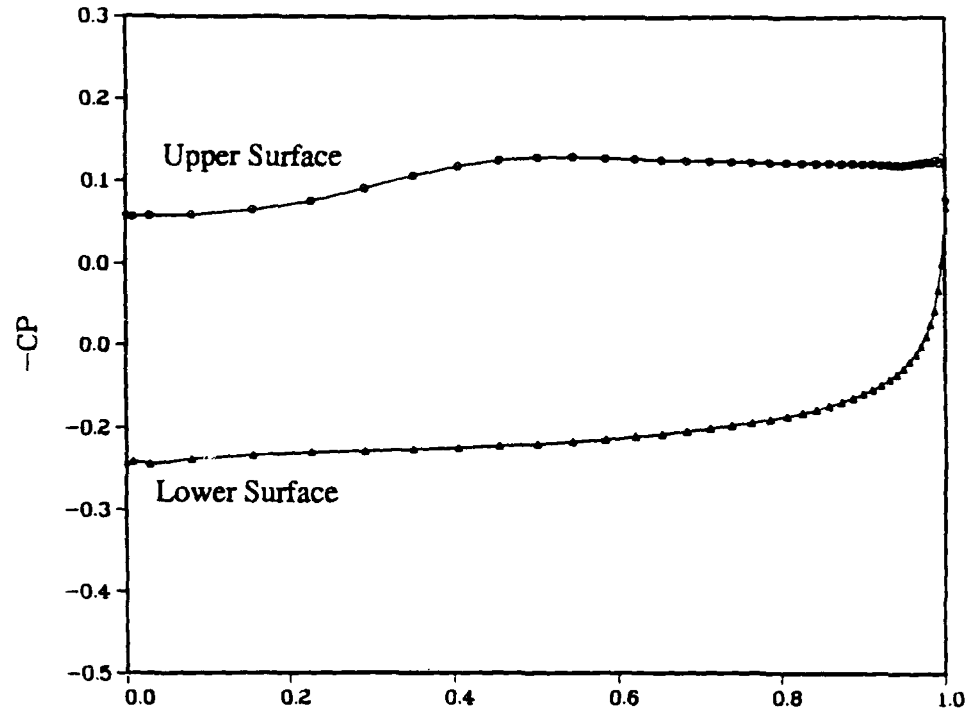


Figure 12. Comparison of Pressures Upsream and Downstream of the Crank
Pitot Pressures x/L=0.398 (upper), x/L=0.88 (lower)



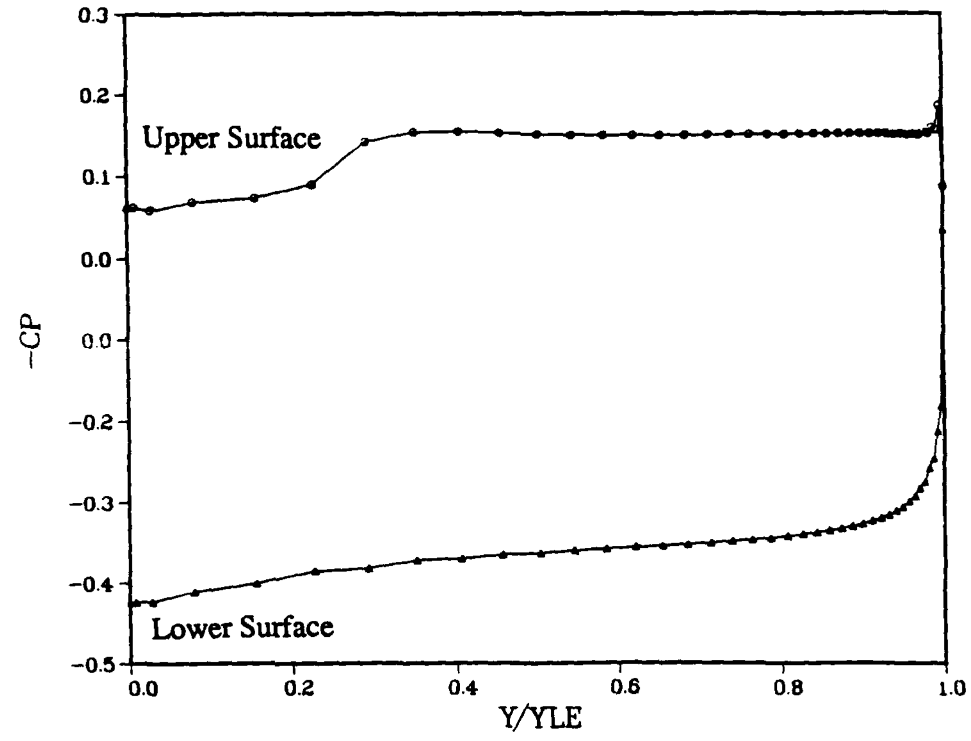


Figure 12. (Cont) Surface Pressures x/l=0.398 (upper), x/L=0.88 (lower)

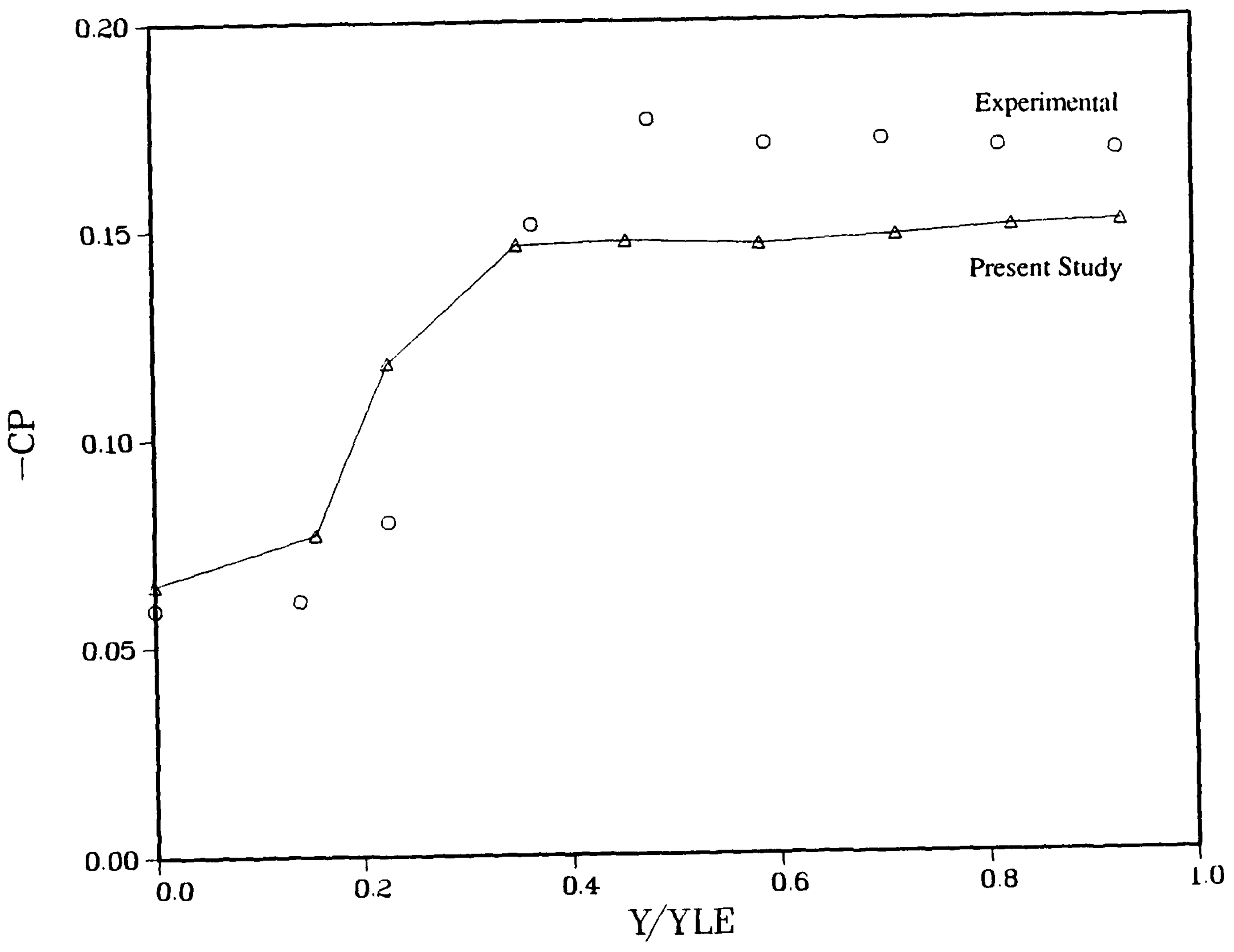


Figure 13. Cranked Wing Surface Pressures Compared with Experiment x/L=0.7

V Conclusions and Recommendations

The Thin-layer Navier-Stokes equations were shown to be capable of reproducing both the primary and secondary structures of vortical flows. For flow over a straight delta wing, good agreement was obtained with the full Navier-Stokes solutions of Buter and Rizzetta. The present study did exhibit slightly more Pitot pressure loss than either the experimental data or the full Navier-Stokes solutions.

The cranked delta wing exhibited flow similar to a straight delta wing for locations upstream of the crank. Vortex location did not vary greatly with streamwise location. A second vortex formed at the crank and then quickly became paired with the first vortex. Moving downstream the vortex center moves inboard and towards the wing surface. The secondary separation present upstream of the crank has been replaced by an area of low Pitot pressure. The magnitude of the wing surface pressures are approximately the same for the flow fore and aft of the crank. The major difference is the flow downstream of the crank must negotiate a larger pressure gradient than the flow upstream of the crank.

Grid resolution appears to be critical when trying to calculate vortical flows. It is necessary to cluster points not only normal to the body but also in the spanwise direction near the leading edge. This is required in order to accurately resolve the large gradients in the η direction. The result of insufficient numerical resolution is degraded definition of the secondary vortical structure or even a failure to resolve the primary structure. For a delta wing the grid can be fairly coarse in the streamwise direction as long as the vortices are stable. If the flow becomes unsteady, such as with vortex bursting, then the resolution in the streamwise direction should be fairly fine. The above is only true when there is little or no trailing edge effects propagating upstream. The cranked delta wing will require a fine grid in the streamwise direction due to the flow changing downstream of the crank.

The use of the Solid State Disk (SSD) allows for the use of grids that require more computer memory than is available. The accuracy of the solution is not affected when using the SSD. The main drawback to using the SSD with the ARC3D code seems to be the amount of input/output required. If the input/output can be better optimized this would reduce the required central processor unit time and also reduce the job cost.

Appendix A: Coordinate Transformation of the Governing Equations

The governing equations can be transformed from Cartesian coordinates (x,y,z,t) into generalized coordinates (ξ,η,ζ,τ) by the following transformation

$$\begin{aligned} \tau &= t \\ \xi &= \xi(x,y,z,t) \\ \eta &= \eta(x,y,z,t) \\ \zeta &= \zeta(x,y,z,t) \end{aligned} \qquad \text{(A1)}$$

Using the chain rule, the Cartesian partial differentials can be written in terms of the transformed coordinates

$$\begin{aligned}
\frac{\partial}{\partial t} &= \frac{\partial \tau}{\partial t}\frac{\partial}{\partial \tau} + \frac{\partial \xi}{\partial t}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial t}\frac{\partial}{\partial \eta} + \frac{\partial \zeta}{\partial t}\frac{\partial}{\partial \zeta} = \partial_\tau + \xi_t\partial_\xi + \eta_t\partial_\eta + \zeta_t\partial_\zeta \\
\frac{\partial}{\partial x} &= \frac{\partial \tau}{\partial x}\frac{\partial}{\partial \tau} + \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta} + \frac{\partial \zeta}{\partial x}\frac{\partial}{\partial \zeta} = \xi_x\partial_\xi + \eta_x\partial_\eta + \zeta_x\partial_\zeta \\
\frac{\partial}{\partial y} &= \frac{\partial \tau}{\partial y}\frac{\partial}{\partial \tau} + \frac{\partial \xi}{\partial y}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial y}\frac{\partial}{\partial \eta} + \frac{\partial \zeta}{\partial y}\frac{\partial}{\partial \zeta} = \xi_y\partial_\xi + \eta_y\partial_\eta + \zeta_y\partial_\zeta \qquad \text{(A2)} \\
\frac{\partial}{\partial z} &= \frac{\partial \tau}{\partial z}\frac{\partial}{\partial \tau} + \frac{\partial \xi}{\partial z}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial z}\frac{\partial}{\partial \eta} + \frac{\partial \zeta}{\partial z}\frac{\partial}{\partial \zeta} = \xi_z\partial_\xi + \eta_z\partial_\eta + \zeta_z\partial_\zeta
\end{aligned}$$

Applying Equation A2 to Equation 3 yields

$$\begin{aligned}
&\partial_\tau \mathbf{Q} + \partial_\xi\left(\xi_t\mathbf{Q} + \xi_x\mathbf{E} + \xi_y\mathbf{F} + \xi_z\mathbf{G}\right) + \partial_\eta(\eta_t\mathbf{Q} + \eta_x\mathbf{E} + \eta_y\mathbf{F} + \eta_z\mathbf{G}) + \\
&\qquad \partial_\zeta\left(\zeta_t\mathbf{Q} + \zeta_x\mathbf{E} + \zeta_y\mathbf{F} + \zeta_z\mathbf{G}\right) = Re^{-1}\left[\partial_\xi\left(\xi_x\mathbf{E}_v + \xi_y\mathbf{F}_v + \xi_z\mathbf{G}_v\right) + \right. \\
&\qquad\qquad \left.\partial_\eta(\eta_x\mathbf{E}_v + \eta_y\mathbf{F}_v + \eta_z\mathbf{G}_v) + \partial_\zeta\left(\zeta_x\mathbf{E}_v + \zeta_y\mathbf{F}_v + \zeta_z\mathbf{G}_v\right)\right] \qquad \text{(A3)}
\end{aligned}$$

the metrics appearing in the above equation are defined as

$$\begin{aligned}
\xi_x &= J\left(y_\eta z_\zeta - y_\zeta z_\eta\right)\\
\xi_y &= J\left(z_\eta x_\zeta - x_\eta z_\zeta\right)\\
\xi_z &= J\left(x_\eta y_\zeta - y_\eta x_\zeta\right)\\
\eta_x &= J\left(z_\xi y_\zeta - y_\xi z_\zeta\right)\\
\eta_y &= J\left(x_\xi z_\zeta - x_\zeta z_\xi\right)\\
\eta_z &= J\left(y_\xi x_\zeta - x_\xi y_\zeta\right)\\
\zeta_x &= J\left(y_\xi z_\eta - z_\xi y_\eta\right)\\
\zeta_y &= J\left(x_\eta z_\xi - x_\xi z_\eta\right)\\
\zeta_z &= J\left(x_\xi y_\eta - y_\xi x_\eta\right)\\
\xi_t &= -x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z\\
\eta_t &= -x_\tau \eta_x - y_\tau \eta_y - z_\tau \eta_z\\
\zeta_t &= -x_\tau \zeta_x - y_\tau \zeta_y - z_\tau \zeta_z
\end{aligned} \tag{A4}$$

where J is the Jacobian of the transformation and is given by

$$J = 1/J^{-1} \tag{A5}$$

$$J^{-1} = \frac{\partial(x,y,z,t)}{\partial(\xi,\eta,\zeta,\tau)} = \begin{vmatrix} 1 & 0 & 0 & 0\\ x_\tau & x_\xi & x_\eta & x_\zeta\\ y_\tau & y_\xi & y_\eta & y_\zeta\\ z_\tau & z_\xi & z_\eta & z_\zeta \end{vmatrix} \tag{A6}$$

$$J^{-1} = x_\xi y_\eta z_\zeta + x_\zeta y_\xi z_\eta + x_\eta y_\zeta z_\xi - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi \tag{A7}$$

If the transformation is a result of grid generation then the metrics and the Jacobian can be computed numerically using central differences.

Vinokur (48) showed that the governing equations can be transformed into strong conservation law form, by first dividing Equation A3 by the Jacobian and then using the chain rule to bring the Jacobian inside the differential operators. Equation A3 becomes

$$\begin{aligned}
&\frac{\partial}{\partial\tau}\left(\frac{\mathbf{Q}}{J}\right)+\frac{\partial}{\partial\xi}\left(\frac{\xi_t\mathbf{Q}+\xi_x\mathbf{E}+\xi_y\mathbf{F}+\xi_z\mathbf{G}}{J}\right)+\frac{\partial}{\partial\eta}\left(\frac{\eta_t\mathbf{Q}+\eta_x\mathbf{E}+\eta_y\mathbf{F}+\eta_z\mathbf{G}}{J}\right)+\\
&\frac{\partial}{\partial\zeta}\left(\frac{\zeta_t\mathbf{Q}+\zeta_x\mathbf{E}+\zeta_y\mathbf{F}+\zeta_z\mathbf{G}}{J}\right)-\mathbf{Q}\left[\left(\frac{\xi_t}{J}\right)_\xi+\left(\frac{\eta_t}{J}\right)_\eta+\left(\frac{\zeta_t}{J}\right)_\zeta\right]-\mathbf{E}\left[\left(\frac{\xi_x}{J}\right)_\xi+\left(\frac{\eta_x}{J}\right)_\eta+\left(\frac{\zeta_x}{J}\right)_\zeta\right]\\
&-\mathbf{F}\left[\left(\frac{\xi_y}{J}\right)_\xi+\left(\frac{\eta_y}{J}\right)_\eta+\left(\frac{\zeta_y}{J}\right)_\zeta\right]-\mathbf{G}\left[\left(\frac{\xi_z}{J}\right)_\xi+\left(\frac{\eta_z}{J}\right)_\eta+\left(\frac{\zeta_z}{J}\right)_\zeta\right]=\frac{1}{\mathrm{Re}}\left\{\frac{\partial}{\partial\xi}\left(\frac{\xi_x\mathbf{E}_v+\xi_y\mathbf{F}_v+\xi_z\mathbf{G}_v}{J}\right)+\right.\\
&+\frac{\partial}{\partial\eta}\left(\frac{\eta_x\mathbf{E}_v+\eta_y\mathbf{F}_v+\eta_z\mathbf{G}_v}{J}\right)+\frac{\partial}{\partial\zeta}\left(\frac{\zeta_x\mathbf{E}_v+\zeta_y\mathbf{F}_v+\zeta_z\mathbf{G}_v}{J}\right)+\mathbf{E}_v\left[\left(\frac{\xi_x}{J}\right)_\xi+\left(\frac{\eta_x}{J}\right)_\eta+\left(\frac{\zeta_x}{J}\right)_\zeta\right]\\
&\left.+\mathbf{F}_v\left[\left(\frac{\xi_y}{J}\right)_\xi+\left(\frac{\eta_y}{J}\right)_\eta+\left(\frac{\zeta_y}{J}\right)_\zeta\right]+\mathbf{G}_v\left[\left(\frac{\xi_z}{J}\right)_\xi+\left(\frac{\eta_z}{J}\right)_\eta+\left(\frac{\zeta_z}{J}\right)_\zeta\right]\right\} \qquad \text{(A8)}
\end{aligned}$$

the last four terms on the left hand side and the last three terms of the right hand side are known as the invarients of the transformation. It can be shown, by using the definition of the metrics, that these terms are zero (32:6).

The contravarient velocities U, V, and W are defined to be the velocities in directions normal to planes of constant ξ, η, and ζ respectively and are given by

$$\begin{aligned}
U &= \xi_t+\xi_x u+\xi_y v+\xi_z w\\
V &= \eta_t+\eta_x u+\eta_y v+\eta_z w \qquad \text{(A9)}\\
W &= \zeta_t+\zeta_x u+\zeta_y v+\zeta_z w
\end{aligned}$$

Applying these definitions to Equation A8, Pullian (38:159-160) showed that the governing equations written in strong conservation-law form and generalized coordinates are given by

$$\partial_\tau \hat{\mathbf{Q}} + \partial_\xi \hat{\mathbf{E}} + \partial_\eta \hat{\mathbf{F}} + \partial_\zeta \hat{\mathbf{G}} = \frac{1}{Re}\left(\partial_\xi \hat{\mathbf{E}}_v + \partial_\eta \hat{\mathbf{F}}_v + \partial_\zeta \hat{\mathbf{G}}_v\right) \tag{A10}$$

$$\hat{\mathbf{Q}} = J^{-1}\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \tag{A11}$$

$$\hat{\mathbf{E}} = J^{-1}\begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ (e+p)U + \xi_t p \end{bmatrix} \hat{\mathbf{F}} = J^{-1}\begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ (e+p)V + \eta_t p \end{bmatrix} \hat{\mathbf{G}} = J^{-1}\begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ (e+p)W + \zeta_t p \end{bmatrix} \tag{A12}$$

$$\hat{\mathbf{E}}_v = J^{-1}\begin{bmatrix} 0 \\ \xi_x\tau_{xx} + \xi_y\tau_{xy} + \xi_z\tau_{xz} \\ \xi_x\tau_{yx} + \xi_y\tau_{yy} + \xi_z\tau_{yz} \\ \xi_x\tau_{zx} + \xi_y\tau_{zy} + \xi_z\tau_{zz} \\ \xi_x\beta_x + \xi_y\beta_y + \xi_z\beta_z \end{bmatrix}$$

$$\hat{\mathbf{F}}_v = J^{-1}\begin{bmatrix} 0 \\ \eta_x\tau_{xx} + \eta_y\tau_{xy} + \eta_z\tau_{xz} \\ \eta_x\tau_{yx} + \eta_y\tau_{yy} + \eta_z\tau_{yz} \\ \eta_x\tau_{zx} + \eta_y\tau_{zy} + \eta_z\tau_{zz} \\ \eta_x\beta_x + \eta_y\beta_y + \eta_z\beta_z \end{bmatrix} \tag{A13}$$

$$\widehat{G}_v = J^{-1} \begin{bmatrix} 0 \\ \zeta_x \tau_{xx} + \zeta_y \tau_{xy} + \zeta_z \tau_{xz} \\ \zeta_x \tau_{yx} + \zeta_y \tau_{yy} + \zeta_z \tau_{yz} \\ \zeta_x \tau_{zx} + \zeta_y \tau_{zy} + \zeta_z \tau_{zz} \\ \zeta_x \beta_x + \zeta_y \beta_y + \zeta_z \beta_z \end{bmatrix}$$

$$\begin{aligned} \beta_x &= \gamma \mu Pr^{-1}\, \partial_x e_i + u\tau_{xx} + v\tau_{xy} + w\tau_{xz} \\ \beta_y &= \gamma \mu Pr^{-1}\, \partial_y e_i + u\tau_{yx} + v\tau_{yy} + w\tau_{yz} \\ \beta_z &= \gamma \mu Pr^{-1}\, \partial_z e_i + u\tau_{zx} + v\tau_{zy} + w\tau_{zz} \end{aligned} \tag{A14}$$

and the cartesian derivatives become

$$\begin{aligned} u_x &= \xi_x u_\xi + \eta_x u_\eta + \zeta_x u_\zeta \\ u_y &= \xi_y u_\xi + \eta_y u_\eta + \zeta_y u_\zeta \\ u_z &= \xi_z u_\xi + \eta_z u_\eta + \zeta_z u_\zeta \\ v_x &= \xi_x v_\xi + \eta_x v_\eta + \zeta_x v_\zeta \\ v_y &= \xi_y v_\xi + \eta_y v_\eta + \zeta_y v_\zeta \\ v_z &= \xi_z v_\xi + \eta_z v_\eta + \zeta_z v_\zeta \\ w_x &= \xi_x w_\xi + \eta_x w_\eta + \zeta_x w_\zeta \\ w_y &= \xi_y w_\xi + \eta_y w_\eta + \zeta_y w_\zeta \\ w_z &= \xi_z w_\xi + \eta_z w_\eta + \zeta_z w_\zeta \end{aligned} \tag{A15}$$

Appendix B: Flux Jacobian Matrices

Pulliam has shown (32:99-100) that the invisicid flux Jacobians, **A**, **B**, and **C**, are obtained by time linearizations of the inviscid flux vectors, **E**, **F**, and **G**, and are given by

$$\widehat{\mathbf{A}} = \frac{\partial \widehat{\mathbf{E}}}{\partial \widehat{\mathbf{Q}}} = \begin{bmatrix} \xi_t & \xi_x & \xi_y & \xi_z & 0 \\ \xi_x\phi^2 - u\theta_1 & \xi_t + \theta_1 - \xi_x(\gamma-2)u & \xi_y u - \xi_x(\gamma-1)v & \xi_z u - \xi_x(\gamma-1)w & \xi_x(\gamma-1) \\ \xi_y\phi^2 - v\theta_1 & \xi_x v - \xi_y(\gamma-1)u & \xi_t + \theta_a - \xi_y(\gamma-2)v & \xi_z v - \xi_y(\gamma-1)w & \xi_y(\gamma-1) \\ \xi_z\phi^2 - w\theta_1 & \xi_x w - \xi_z(\gamma-1)u & \xi_y w - \xi_z(\gamma-1)v & \xi_t + \theta_a - \xi_z(\gamma-2)w & \xi_z(\gamma-1) \\ -\theta_1\left(\gamma e/\rho - 2\phi^2\right) & \xi_x\psi - (\gamma-1)u\theta_1 & \xi_y\psi - (\gamma-1)v\theta_1 & \xi_z\psi - (\gamma-1)w\theta_1 & \xi_t + \gamma\theta_1 \end{bmatrix} \tag{B1}$$

$$\widehat{\mathbf{B}} = \frac{\partial \widehat{\mathbf{F}}}{\partial \widehat{\mathbf{Q}}} = \begin{bmatrix} \eta_t & \eta_x & \eta_y & \eta_z & 0 \\ \eta_x\phi^2 - u\theta_{2b} & \eta_t + \theta_2 - \eta_x(\gamma-2)u & \eta_y u - \eta_x(\gamma-1)v & \eta_z u - \eta_x(\gamma-1)w & \eta_x(\gamma-1) \\ \eta_y\phi^2 - v\theta_2 & \eta_x v - \eta_y(\gamma-1)u & \eta_t + \theta_2 - \eta_y(\gamma-2)v & \eta_z v - \eta_y(\gamma-1)w & \eta_y(\gamma-1) \\ \eta_z\phi^2 - w\theta_2 & \eta_x w - \eta_z(\gamma-1)u & \eta_y w - \eta_z(\gamma-1)v & \eta_t + \theta_2 - \eta_z(\gamma-2)w & \eta_z(\gamma-1) \\ -\theta_2\left(\gamma e/\rho - 2\phi^2\right) & \eta_x\psi - (\gamma-1)u\theta_2 & \eta_y\psi - (\gamma-1)v\theta_2 & \eta_z\psi - (\gamma-1)w\theta_2 & \eta_t + \gamma\theta_2 \end{bmatrix} \tag{B2}$$

$$\widehat{C} = \frac{\partial \widehat{G}}{\partial \widehat{Q}} = \left[\begin{array}{ll} \zeta_t & \zeta_x \\ \zeta_x\phi^2 - u\theta_3 & \zeta_t + \theta_3 - \zeta_x(\gamma-2)u \\ \zeta_y\phi^2 - v\theta_3 & \zeta_x v - \zeta_y(\gamma-1)u \\ \zeta_z\phi^2 - w\theta_3 & \zeta_x w - \zeta_z(\gamma-1)u \\ -\theta_3\left(\gamma e/\rho - 2\phi^2\right) & \zeta_x\psi - (\gamma-1)u\theta_3 \end{array}\right.$$

$$\left.\begin{array}{lll} \zeta_y & \zeta_z & 0 \\ \zeta_y u - \zeta_x(\gamma-1)v & \zeta_z u - \zeta_x(\gamma-1)w & \zeta_x(\gamma-1) \\ \zeta_t + \theta_3 - \zeta_y(\gamma-2)v & \zeta_z v - \zeta_y(\gamma-1)w & \zeta_y(\gamma-1) \\ \zeta_y w - \zeta_z(\gamma-1)v & \zeta_t + \theta_3 - \zeta_z(\gamma-2)w & \zeta_z(\gamma-1) \\ \zeta_y\psi - (\gamma-1)v\theta_3 & \zeta_z\psi - (\gamma-1)w\theta_3 & \zeta_t + \gamma\theta_3 \end{array}\right] \tag{B3}$$

where

$$\phi^2 = (\gamma-1)\left(\frac{u^2+v^2+w^2}{2}\right) \tag{B4}$$

$$\psi = \gamma e/\rho - \phi^2 \tag{B5}$$

$$\begin{aligned} \theta_1 &= \xi_x u + \xi_y v + \xi_z w \\ \theta_2 &= \eta_x u + \eta_y v + \eta_z w \\ \theta_3 &= \zeta_x u + \zeta_y v + \zeta_z w \end{aligned} \tag{B6}$$

The viscous flux Jacobian (M) is obtained from the linearization of the viscous flux vector (S). Using a Taylor series, Steger has shown the Jacobian is given by (43:3-4)

$$\widehat{\mathbf{M}} = \frac{\partial \widehat{S}}{\partial \widehat{\mathbf{Q}}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ m_{21} & \alpha_1\delta_\zeta(\rho^{-1}) & \alpha_2\delta_\zeta(\rho^{-1}) & \alpha_3\delta_\zeta(\rho^{-1}) & 0 \\ m_{31} & \alpha_2\delta_\zeta(\rho^{-1}) & \alpha_4\delta_\zeta(\rho^{-1}) & \alpha_5\delta_\zeta(\rho^{-1}) & 0 \\ m_{41} & \alpha_3\delta_\zeta(\rho^{-1}) & \alpha_5\delta_\zeta(\rho^{-1}) & \alpha_6\delta_\zeta(\rho^{-1}) & 0 \\ m_{51} & m_{52} & m_{53} & m_{54} & \alpha_0\delta_\zeta(\rho^{-1}) \end{bmatrix} \tag{B7}$$

$$\begin{gathered} m_{21} = \alpha_1\delta_\zeta(-u/\rho) + \alpha_2\delta_\zeta(-v/\rho) + \alpha_3\delta_\zeta(-w/\rho) \\ m_{31} = \alpha_2\delta_\zeta(-u/\rho) + \alpha_4\delta_\zeta(-v/\rho) + \alpha_5\delta_\zeta(-w/\rho) \\ m_{41} = \alpha_3\delta_\zeta(-u/\rho) + \alpha_5\delta_\zeta(-v/\rho) + \alpha_6\delta_\zeta(-w/\rho) \\ m_{51} = \alpha_1\delta_\zeta(-u^2/\rho) + \alpha_2\delta_\zeta(-2uv/\rho) + \alpha_3\delta_\zeta(-2uw/\rho) + \alpha_4\delta_\zeta(-v^2/\rho) + \\ \alpha_6\delta_\zeta(-w^2/\rho) + \alpha_0\delta_\zeta(-e/\rho^2) + \alpha_0\delta_\zeta[(u^2 + v^2 + w^2)/\rho] \\ m_{52} = -m_{21} - \alpha_0\delta_\zeta(u/\rho) \\ m_{53} = -m_{31} - \alpha_0\delta_\zeta(v/\rho) \\ m_{54} = -m_{41} - \alpha_0\delta_\zeta(w/\rho) \end{gathered} \tag{B8}$$

$$\begin{gathered} \alpha_0 = \gamma\mu Pr^{-1}(\zeta_x^2 + \zeta_y^2 + \zeta_z^2) \\ \alpha_1 = \mu\left[(4/3)\zeta_x^2 + \zeta_y^2 + \zeta_z^2\right] \\ \alpha_2 = (\mu/3)\zeta_x\,\zeta_y \\ \alpha_2 = (\mu/3)\zeta_x\,\zeta_z \\ \alpha_4 = \mu\left[\zeta_x^2 + (4/3)\zeta_y^2 + \zeta_z^2\right] \\ \alpha_5 = (\mu/3)\zeta_y\,\zeta_z \\ \alpha_6 = \mu\left[\zeta_x^2 + \zeta_y^2 + (4/3)\zeta_z^2\right] \end{gathered} \tag{B9}$$

Appendix C: Flux Jacobian Eigenvalue and Eigenvector Matrices

Warming, Beam, and Hyett (44:1037-1041) have shown that the invisicid flux Jacobians can be diagonalized since they have real eigenvalues and a complete set of eigenvectors. The flux Jacobians are written in terms of their eigenvalues and eigenvectors as

$$\begin{aligned}\widehat{\mathbf{A}}^n &= \left(\mathbf{T}_\xi \Lambda_\xi \mathbf{T}_\xi^{-1}\right)^n \\ \widehat{\mathbf{B}}^n &= \left(\mathbf{T}_\eta \Lambda_\eta \mathbf{T}_\eta^{-1}\right)^n \\ \widehat{\mathbf{C}}^n &= \left(\mathbf{T}_\zeta \Lambda_\zeta \mathbf{T}_\zeta^{-1}\right)^n\end{aligned} \tag{C1}$$

where $\mathbf{T}_\xi$ is the eigenvector matrix of **A** and likewise $\mathbf{T}_\eta$ for **B** and $\mathbf{T}_\zeta$ for **C**. The eigenvalue matrices are given by

$$\begin{aligned}\Lambda_\xi &= \begin{bmatrix} U & 0 & 0 & 0 & 0 \\ 0 & U & 0 & 0 & 0 \\ 0 & 0 & U & 0 & 0 \\ 0 & 0 & 0 & U+a\kappa_1 & 0 \\ 0 & 0 & 0 & 0 & U-a\kappa_1 \end{bmatrix} \\ \Lambda_\eta &= \begin{bmatrix} V & 0 & 0 & 0 & 0 \\ 0 & V & 0 & 0 & 0 \\ 0 & 0 & V & 0 & 0 \\ 0 & 0 & 0 & V+a\kappa_2 & 0 \\ 0 & 0 & 0 & 0 & V-a\kappa_2 \end{bmatrix} \\ \Lambda_\zeta &= \begin{bmatrix} W & 0 & 0 & 0 & 0 \\ 0 & W & 0 & 0 & 0 \\ 0 & 0 & W & 0 & 0 \\ 0 & 0 & 0 & W+a\kappa_3 & 0 \\ 0 & 0 & 0 & 0 & W-a\kappa_3 \end{bmatrix}\end{aligned} \tag{C2}$$

where a is a characteristic speed and

$$\kappa_1 = \left(\xi_x^2 + \xi_y^2 + \xi_z^2\right)^{1/2}$$

$$\kappa_2 = \left(\eta_x^2 + \eta_y^2 + \eta_z^2\right)^{1/2} \qquad \text{(C3)}$$
$$\kappa_3 = \left(\zeta_x^2 + \zeta_y^2 + \zeta_z^2\right)^{1/2}$$

The eigenvector matrices are

$$T_\xi = \begin{bmatrix} \tilde{\xi}_x & \tilde{\xi}_y & \tilde{\xi}_z & \alpha & \alpha \\ \tilde{\xi}_x u & \tilde{\xi}_y u - \tilde{\xi}_z \rho & \tilde{\xi}_z u + \tilde{\xi}_y \rho & \alpha\left(u + \tilde{\xi}_x a\right) & \alpha\left(u - \tilde{\xi}_x a\right) \\ \tilde{\xi}_x v + \tilde{\xi}_z \rho & \tilde{\xi}_y v & \tilde{\xi}_z v - \tilde{\xi}_x \rho & \alpha\left(v + \tilde{\xi}_y a\right) & \alpha\left(v - \tilde{\xi}_y a\right) \\ \tilde{\xi}_x w - \tilde{\xi}_y \rho & \tilde{\xi}_y w + \tilde{\xi}_x \rho & \tilde{\xi}_z w & \alpha\left(w + \tilde{\xi}_z a\right) & \alpha\left(w - \tilde{\xi}_z a\right) \\ \left[\tilde{\xi}_x \phi^2/(\gamma-1) + \rho\left(\tilde{\xi}_z v - \tilde{\xi}_y w\right)\right] & \left[\tilde{\xi}_y \phi^2/(\gamma-1) + \rho\left(\tilde{\xi}_x w - \tilde{\xi}_z u\right)\right] & \left[\tilde{\xi}_z \phi^2/(\gamma-1) + \rho\left(\tilde{\xi}_y u - \tilde{\xi}_x v\right)\right] & \alpha\left[\left(\phi^2 + a^2\right)/(\gamma-1) + \tilde{\theta}_1 a\right] & \alpha\left[\left(\phi^2 + a^2\right)/(\gamma-1) - \tilde{\theta}_1 a\right] \end{bmatrix}$$

$$T_\eta = \begin{bmatrix} \tilde{\eta}_x & \tilde{\eta}_y & \tilde{\eta}_z & \alpha & \alpha \\ \tilde{\eta}_x u & \tilde{\eta}_y u - \tilde{\eta}_z \rho & \tilde{\eta}_z u + \tilde{\eta}_y \rho & \alpha\left(u + \tilde{\eta}_x a\right) & \alpha\left(u - \tilde{\eta}_x a\right) \\ \tilde{\eta}_x v + \tilde{\eta}_z \rho & \tilde{\eta}_y v & \tilde{\eta}_z v - \tilde{\eta}_x \rho & \alpha\left(v + \tilde{\eta}_y a\right) & \alpha\left(v - \tilde{\eta}_y a\right) \\ \tilde{\eta}_x w - \tilde{\eta}_y \rho & \tilde{\eta}_y w + \tilde{\eta}_x \rho & \tilde{\eta}_z w & \alpha\left(w + \tilde{\eta}_z a\right) & \alpha\left(w - \tilde{\eta}_z a\right) \\ \left[\tilde{\eta}_x \phi^2/(\gamma-1) + \rho\left(\tilde{\eta}_z v - \tilde{\eta}_y w\right)\right] & \left[\tilde{\eta}_y \phi^2/(\gamma-1) + \rho\left(\tilde{\eta}_x w - \tilde{\eta}_z u\right)\right] & \left[\tilde{\eta}_z \phi^2/(\gamma-1) + \rho\left(\tilde{\eta}_y u - \tilde{\eta}_x v\right)\right] & \alpha\left[\left(\phi^2 + a^2\right)/(\gamma-1) + \tilde{\theta}_2 a\right] & \alpha\left[\left(\phi^2 + a^2\right)/(\gamma-1) - \tilde{\theta}_2 a\right] \end{bmatrix} \qquad \text{(C4)}$$

$$T_\zeta = \begin{bmatrix} \tilde{\zeta}_x & \tilde{\zeta}_y & \tilde{\zeta}_z & \alpha & \alpha \\ \tilde{\zeta}_x u & \tilde{\zeta}_y u - \tilde{\zeta}_z \rho & \tilde{\zeta}_z u + \tilde{\zeta}_y \rho & \alpha\left(u + \tilde{\zeta}_x a\right) & \alpha\left(u - \tilde{\zeta}_x a\right) \\ \tilde{\zeta}_x v + \tilde{\zeta}_z \rho & \tilde{\zeta}_y v & \tilde{\zeta}_z v - \tilde{\zeta}_x \rho & \alpha\left(v + \tilde{\zeta}_y a\right) & \alpha\left(v - \tilde{\zeta}_y a\right) \\ \tilde{\zeta}_x w - \tilde{\zeta}_y \rho & \tilde{\zeta}_y w + \tilde{\zeta}_x \rho & \tilde{\zeta}_z w & \alpha\left(w + \tilde{\zeta}_z a\right) & \alpha\left(w - \tilde{\zeta}_z a\right) \\ \left[\tilde{\zeta}_x \phi^2/(\gamma-1) + \rho\left(\tilde{\zeta}_z v - \tilde{\zeta}_y w\right)\right] & \left[\tilde{\zeta}_y \phi^2/(\gamma-1) + \rho\left(\tilde{\zeta}_x w - \tilde{\zeta}_z u\right)\right] & \left[\tilde{\zeta}_z \phi^2/(\gamma-1) + \rho\left(\tilde{\zeta}_y u - \tilde{\zeta}_x v\right)\right] & \alpha\left[\left(\phi^2 + a^2\right)/(\gamma-1) + \tilde{\theta}_3 a\right] & \alpha\left[\left(\phi^2 + a^2\right)/(\gamma-1) - \tilde{\theta}_3 a\right] \end{bmatrix}$$

$$T_\xi^{-1} = \begin{bmatrix} \tilde{\xi}_x\left(1-\phi^2/a^2\right) - \left(\tilde{\xi}_z v - \tilde{\xi}_y w\right)/\rho & \tilde{\xi}_x(\gamma-1)u/a^2 & \tilde{\xi}_x(\gamma-1)v/a^2 + \tilde{\xi}_z/\rho & \tilde{\xi}_x(\gamma-1)w/a^2 - \tilde{\xi}_y/\rho & -\tilde{\xi}_x(\gamma-1)/a^2 \\ \tilde{\xi}_y\left(1-\phi^2/a^2\right) - \left(\tilde{\xi}_x w - \tilde{\xi}_z u\right)/\rho & \tilde{\xi}_y(\gamma-1)u/a^2 + \tilde{\xi}_x/\rho & \tilde{\xi}_y(\gamma-1)v/a^2 & \tilde{\xi}_y(\gamma-1)w/a^2 + \tilde{\xi}_x/\rho & -\tilde{\xi}_y(\gamma-1)/a^2 \\ \tilde{\xi}_z\left(1-\phi^2/a^2\right) - \left(\tilde{\xi}_y u - \tilde{\xi}_x v\right)/\rho & \tilde{\xi}_z(\gamma-1)u/a^2 + \tilde{\xi}_y/\rho & \tilde{\xi}_z(\gamma-1)v/a^2 + \tilde{\xi}_x/\rho & \tilde{\xi}_z(\gamma-1)w/a^2 & -\tilde{\xi}_z(\gamma-1)/a^2 \\ \beta\left(\phi^2 - \tilde{\theta}_1 a\right) & -\beta(\gamma-1)u - \xi_x a & -\beta\left[(\gamma-1)v - \xi_y a\right] & -\beta\left[(\gamma-1)w - \xi_z a\right] & \beta(\gamma-1) \\ \beta\left(\phi^2 + \tilde{\theta}_1 a\right) & -\beta(\gamma-1)u + \xi_x a & -\beta\left[(\gamma-1)v + \xi_y a\right] & -\beta\left[(\gamma-1)w + \xi_z a\right] & \beta(\gamma-1) \end{bmatrix}$$

$$T_\eta^{-1} = \left[\begin{array}{cc} \tilde{\eta}_x\left(1-\phi^2/a^2\right) - \left(\tilde{\eta}_z v - \tilde{\eta}_y w\right)/\rho & \tilde{\eta}_x(\gamma-1)u/a^2 \\ \tilde{\eta}_y\left(1-\phi^2/a^2\right) - \left(\tilde{\eta}_x w - \tilde{\eta}_z u\right)/\rho & \tilde{\eta}_y(\gamma-1)u/a^2 + \tilde{\eta}_x/\rho \\ \tilde{\eta}_z\left(1-\phi^2/a^2\right) - \left(\tilde{\eta}_y u - \tilde{\eta}_x v\right)/\rho & \tilde{\eta}_z(\gamma-1)u/a^2 + \tilde{\eta}_y/\rho \\ \beta\left(\phi^2 - \tilde{\theta}_2 a\right) & -\beta(\gamma-1)u - \eta_x a \\ \beta\left(\phi^2 + \tilde{\theta}_2 a\right) & -\beta(\gamma-1)u + \eta_x a \end{array} \right.$$

$$\left.\begin{matrix} \tilde{\eta}_x(\gamma-1)v/a^2+\tilde{\eta}_z/\rho & \tilde{\eta}_x(\gamma-1)w/a^2-\tilde{\eta}_y/\rho & -\tilde{\eta}_x(\gamma-1)/a^2 \\ \tilde{\eta}_y(\gamma-1)v/a^2 & \tilde{\eta}_y(\gamma-1)w/a^2+\tilde{\eta}_x/\rho & -\tilde{\eta}_y(\gamma-1)/a^2 \\ \tilde{\eta}_z(\gamma-1)v/a^2+\tilde{\eta}_x/\rho & \tilde{\eta}_z(\gamma-1)w/a^2 & -\tilde{\eta}_z(\gamma-1)/a^2 \\ -\beta[(\gamma-1)v-\eta_y a] & -\beta[(\gamma-1)w-\eta_z a] & \beta(\gamma-1) \\ -\beta[(\gamma-1)v+\eta_y a] & -\beta[(\gamma-1)w+\eta_z a] & \beta(\gamma-1) \end{matrix}\right] \qquad \text{(C5)}$$

$$T_\zeta^{-1} = \left[\begin{matrix} \tilde{\zeta}_x\left(1-\phi^2/a^2\right)-\left(\tilde{\zeta}_z v-\tilde{\zeta}_y w\right)/\rho & \tilde{\zeta}_x(\gamma-1)u/a^2 \\ \tilde{\zeta}_y\left(1-\phi^2/a^2\right)-\left(\tilde{\zeta}_x w-\tilde{\zeta}_z u\right)/\rho & \tilde{\zeta}_y(\gamma-1)u/a^2+\tilde{\zeta}_x/\rho \\ \tilde{\zeta}_z\left(1-\phi^2/a^2\right)-\left(\tilde{\zeta}_y u-\tilde{\zeta}_x v\right)/\rho & \tilde{\zeta}_z(\gamma-1)u/a^2+\tilde{\zeta}_y/\rho \\ \beta\left(\phi^2-\tilde{\theta}_3 a\right) & -\beta(\gamma-1)u-\zeta_x a \\ \beta\left(\phi^2+\tilde{\theta}_3 a\right) & -\beta(\gamma-1)u+\zeta_x a \end{matrix}\right.$$

$$\left.\begin{matrix} \tilde{\zeta}_x(\gamma-1)v/a^2+\tilde{\zeta}_z/\rho & \tilde{\zeta}_x(\gamma-1)w/a^2-\tilde{\zeta}_y/\rho & -\tilde{\zeta}_x(\gamma-1)/a^2 \\ \tilde{\zeta}_y(\gamma-1)v/a^2 & \tilde{\zeta}_y(\gamma-1)w/a^2+\tilde{\zeta}_x/\rho & -\tilde{\zeta}_y(\gamma-1)/a^2 \\ \tilde{\zeta}_z(\gamma-1)v/a^2+\tilde{\zeta}_x/\rho & \tilde{\zeta}_z(\gamma-1)w/a^2 & -\tilde{\zeta}_z(\gamma-1)/a^2 \\ -\beta[(\gamma-1)v-\zeta_y a] & -\beta[(\gamma-1)w-\zeta_z a] & \beta(\gamma-1) \\ -\beta[(\gamma-1)v+\zeta_y a] & -\beta[(\gamma-1)w+\zeta_z a] & \beta(\gamma-1) \end{matrix}\right]$$

where

$$\begin{matrix} \tilde{\xi}_x = \dfrac{\xi_x}{\kappa_1} & \tilde{\xi}_y = \dfrac{\xi_y}{\kappa_1} & \tilde{\xi}_z = \dfrac{\xi_z}{\kappa_1} & \tilde{\theta}_1 = \dfrac{\theta_1}{\kappa_1} \\ \tilde{\eta}_x = \dfrac{\eta_x}{\kappa_2} & \tilde{\eta}_y = \dfrac{\eta_y}{\kappa_2} & \tilde{\eta}_z = \dfrac{\eta_z}{\kappa_2} & \tilde{\theta}_2 = \dfrac{\theta_2}{\kappa_2} \\ \tilde{\zeta}_x = \dfrac{\zeta_x}{\kappa_3} & \tilde{\zeta}_y = \dfrac{\zeta_y}{\kappa_3} & \tilde{\zeta}_z = \dfrac{\zeta_z}{\kappa_3} & \tilde{\theta}_3 = \dfrac{\theta_3}{\kappa_3} \end{matrix} \qquad \text{(C6)}$$

$$\alpha = \frac{\rho}{\sqrt{2}a} \qquad \beta = \frac{1}{\sqrt{2}\rho a} \qquad \text{(C7)}$$

Appendix D: In-Core Versus Solid State Disk Comparison

The use of the solid state disk allows for the calculations to be performed on grids that require more computer memory than is available. The purpose of this appendix is to evaluate the overhead associated with using the solid state disk. The code was run for 50 iterations in order to get a representative feel for differences in central processor unit (CPU) times, input/output (I/O) requirements, and job size.

A grid was generated for a delta wing that was small enough to fit entirely in the core of the Cray XMP-12 computer. The grid consisted of 32,000 points which is about the maximum number of points that can be processed at one time. The grid had 20 points in the streamwise direction, 40 points in the spanwise direction, and 40 points in the normal direction.

The accuracy of the solution is not affected by the choice of methods. The only difference in the operation of the code is, the solid state disk loads and unloads planes of data as needed for calculations. This resulted in a 40 percent increase in CPU time. This number will vary depending on the number of data planes that are processed per iteration.

The in-core run required a job size of 1,420,000 words to execute. This used approximately 80 percent of the available core memory. The SSD run required 260,000 words of core to execute. The problem encountered when running a job that requires a large block of core memory is that the job may sit for extended periods. The in-core job sat for almost 30 hours, while the out-of-core job sat for about 5 minutes, and was finished in about 30 minutes.

The last area examined was the cost of running the job. The in-core cost was $40 dollars, while the out-of-core jobcost was $375. It is believed that this cost is due to an input/output charge. The SSD made almost 250,000 I/O requests, while the in-core had 150. If this is the case, then using the SSD may become prohibitive from a financial point of view.

Appendix E: Local Application

This appendix contains the job control language and parameter input files necessary to run the ARC3D code on the Cray XMP-12 computer at Wright-Patterson AFB,Ohio. The first file is used for compiling the code and storing it on the Cray. The second file is the file used for submitting a job to the Cray. The code will look for three files before it begins execution, 1) a restart file, if requested, 2) a grid file, and 3) a parameter input file. The third file is an example of a parameter input file. Upon termination the program returns a restart file and a parameter output file. Due to large amount of CPU time required, the code was run in batches of 200 time iterations.

```
******************** File To Compile ARC3D Code ************************

JOB,JN=CRANKCMP,T=50,MFL=150000.  SMITH-AFIT/ENY-54731
ACCOUNT,AC=xxxxxxx,APW=xxxxxx,US=xxxxxxx,UPW=xxxxxx.
*.
*. THIS FILE IS USED TO COMPILE THE ARC3D CODE ON THE CRAY XMP
*. A COPY OF THE COMPILED PROGRAM IS STORED ON THE CRAY.
*.
*. PURGE THE PREVIOUS COPY.
*.
ACCESS,DN=ARC3D,ID=D880165,UQ.
DELETE,DN=ARC3D.
RELEASE,DN=ARC3D.
*.
*. GET SOURCE CODE FROM THE CFS.
*.
FETCH,DN=CODE,SDN=ARC3D,MF=CB,DF=CB,^
TEXT='SREAD,ARC3D,ARC3D.CTASK,ALL.'.
*.
*. COMPILE CODE AND WRITE TO BINCODE
*.
CFT,I=CODE,B=BINCODE,ON=Z,OPT=FULLIFCON,MAXBLOCK=4000,L=0.
*.
*. SAVE EXECUTABLE ON THE CRAY
*.
SAVE,DN=BINCODE,PDN=ARC3D,ID=D880165.
*.
*. SEND BACKUP TO THE CFS
*.
*. DISPOSE,DN=BINCODE,SDN=ARC3DEXE,MF=CB,DC=ST,DF=TR,^
*. TEXT='CTASK,ALL.SWRITE,ARC3DEXE,ARC3DEXE'.
/EOF
```

************************ ARC3D Input File ****************************

```
JOB,JN=CRANK,T=1600,MFL=500000,SSD=11000,CL=P2.   SMITH/54731
ACCOUNT,AC=xxxxxxx,APW=xxxxxx,US=xxxxxxx,UPW=xxxxxx.
*.
*. THIS FILE RUNS THE ARC3D COMPUTER CODE ON THE ASD
*. CRAY XMP USING THE SOLID STATE DISK.
*.
*. SET UP FILES ON THE SSD.
*.
ASSIGN,DN=FT11,DV=SSD-0-20,RDM,U.
ASSIGN,DN=FT12,DV=SSD-0-20,RDM,U.
*.
*. GET THE RESTART FILE FROM THE CFS.
*.
FETCH,DN=RESTIN,SDN=RESTIN,MF=CB,DF=TR,^
TEXT='SREAD,RESTIN,RESTIN.CTASK,ALL.'.
*.
*. GET THE GRID FROM THE CFS.
*.
FETCH,DN=GRIDIN,SDN=BNGRID,MF=CB,DF=TR,^
TEXT='SREAD,BNGRID,BINGRID.CTASK,ALL.'.
*.
*. GET INPUT FILE FROM THE CYBER.
*.
FETCH,DN=PARAIN,SDN=PARAIN,MF=CB,DF=CB,^
TEXT='GET,PARAIN.CTASK,ALL.'.
*.
*. GET COMPILED ARC3D CODE FROM THE CRAY.
*.
ACCESS,DN=$BLD,PDN=ARC3D,ID=D880165.
OPTION,STAT=ON.
*.
*. LOAD AND RUN.
*.
LDR.
*.
*. SAVE THE RESTART TO THE CFS.
*.
DISPOSE,DN=RESTOUT,SDN=RESTOUT,MF=CB,DC=ST,DF=TR,DEFER,^
TEXT='CTASK,ALL.SWRITE,RESTOUT,RESTOUT'.
*.
*. SAVE THE GRID OUTPUT TO THE CFS.
*.
*.  DISPOSE,DN=GRIDOUT,SDN=GRIDPLOT,MF=CB,DC=ST,DF=TR,^
*.  TEXT='CTASK,ALL.SWRITE,GRIDPLOT,GRIDPLOT'.
*.
*. BATCH PARAMETER OUTPUT FILE TO THE WAIT QUEUE.
*.
DISPOSE,DN=PAROUT,SDN=PAROUT,MF=CB,DC=ST,DF=CB,DEFER,^
TEXT='CTASK,ALL.ROUTE,PAROUT,DC=WT,UJN=PAROUT,UN=D880165.'.
```

```
*.
EXIT.
DISPOSE,DN=RESTOUT,SDN=RESTOUT,MF=CB,DC=ST,DF=TR,DEFER,^
TEXT='CTASK,ALL.SWRITE,RESTOUT,RESTOUT'.
DISPOSE,DN=PAROUT,SDN=PAROUT,MF=CB,DC=ST,DF=CB,DEFER,^
TEXT='CTASK,ALL.ROUTE,PAROUT,DC=WT,UJN=PAROUT,UN=D880165.'.
DUMPJOB.
DEBUG.
*.
/EOF
```

```
******************** Parameter Input File ***********************

F                     INCORE(T/F)
28,65,70              JMAX,KMAX,LMAX
T T                   RESTART,STORE
T T F                 READGRID,READALL,WRITGRID
50                    NP
T                     PJLINE (ENTER NJLINE(<11),THEN EACH PAIR OF K,L)
1                     NJLINE
21 25                 JLINK(NJ),JLINL(NJ)
T                     PKLINE (ENTER NKLINE(<11),THEN EACH PAIR OF J,L)
2                     NKLINE
24 25                 KLINJ(NK),KLINL(NK)
24 24
T                     PLLINE (ENTER NLLINE(<11),THEN EACH PAIR OF J,K)
1                     NLLINE
24 21                 LLINJ(NL),LLINK(NL)
T F                   IFSCOR,METAV
T F                   VISCOUS,TURBULNT (IF VISCOUS THEN ENTER
                      RE,PR,TINF)
686650, 0.72, 234.4   RE,PR,TINF
2.5,10.,0.            FSMACH,ALP,YAW
1                     METH
3                      ISPEC      ARTIFICIAL VISCOSITY STUFF
0.50,0.75             DIS2X,DIS4X
0.50,0.75             DIS2Y,DIS4Y
0.50,0.75             DIS2Z,DIS4Z
1                      IVARDT  0-CONST 1-VARIABLE TIME STEP
F                     PERIODIC IN K
4                     IBC(IF IBC=3 THEN ENTER JTAIL1,JTAIL2)
                      (THERE IS NO 4!!)
41  25  3  29         KEDGE LSURF JSTART JEND      IBC=4 STUFF
1.0                   CLENGTH    (CHECK THIS NUMBER)
2                     IORDER
1                     NUMDT    (NUMBER OF TIMESTEP SEQUENCES TO
                       FOLLOW)
2.00 200              DTSEQ(I),ITERDT(I)  (TIME STEP AND NUMBER OF
                       TIMESTEPS)
/EOF
```

Appendix F : Algebraic Grid Generator

This program was originally written by Dr. Phil Webster of the Air Force Flight Dynamics Laboratory to generate a grid for a delta wing. It has since been modified by the author to generate the grid for a cranked delta wing. A branch cut between the upper and lower surfaces of the body allows for the use of an H grid. This approach was used to eliminate the small radius of curvature generated at the leading edge using a C grid. Points are clustered in high gradient areas using an exponential stretching function. There is no provision for a wake region, therefore this code is only used to generate grids for supersonic flows.

```
      PROGRAM DELGRD
C
C  THIS PROGRAM WAS MODIFIED TO CLUSTER POINTS AT THE WING
C  LEADING EDGE BY USING ELLIPSES FOR THE UPPER AND LOWER
C  BOUNDARIES OF THE DOMAIN.  FRS  8 AUG 88
C
C  THIS PROGRAM WAS MODIFIED TO GENERATE A GRID FOR A CRANKED
C  DELTA WING CONFIGURATION. FRS 28 AUG 88
C
C  PROGRAM GENERATES A GRID (HOPEFULLY) TO FIT THE DELTA WING
C  MODIFIED FOR A DELTA WING WITH NO WAKE REGION
C
CDIR$ NOVECTOR
      REAL LEN
       COMMON X(28,65,70),Y(28,65,70),Z(28,65,70)
      LOGICAL LSTOP,LCHECK,LCRANK
      LCHECK = .TRUE.
      LSTOP = .FALSE.
      LCRANK = .TRUE.
C
C OUTPUT FILES FOR IRIS (MUST BE COMMENTED OUT)
C
C       OPEN (UNIT=10,FILE='GRID.BIN',FORM='BINARY')
C      OPEN (UNIT=9,FILE='GRID.DAT')
C
C FILES FOR THE KIRTLAND AFB CRAY
C
       OPEN (10,FILE='CGRID',STATUS='NEW',FORM='UNFORMATTED')
       IF (LCHECK) OPEN (9,FILE='CKGRID',STATUS='NEW',
     &       FORM='FORMATTED')
      PI = ASIN(1.0) * 2.0
      LEN = .100
      ALPHA = 10.0
      ALPHA = (ALPHA / 180.0) * PI
      OMEGA = 40.0
      OMEGA = (OMEGA / 180.0) * PI
      DELTA = 1.0E-3 * LEN
      XSTART = 0.05 * LEN
      XEND = XSTART + LEN
      XTOTAL = XEND
      JSTART = 3
      KMID = 4
C
C  INPUT GEOMETRY FOR DELTA WING
C
       IF(.NOT.LCRANK)THEN
         SWEEP = 80.0
         SWEEP = ((90.0 - SWEEP) / 180.0) * PI
         BASE = LEN * TAN(SWEEP)
         XMID = XSTART + 0.90 * LEN
C
         NJ = 30
         JMID = 28
         JEND = 30
         NK=55
         KEDGE=34
         NL=65
         LSURF=21
C
C  INPUT GEOMETRY FOR CRANKED DELTA WING
C
      ELSE
        XCRANK=.402*LEN+XSTART
        SWEEP1=80.
        SWEEP1=((90.-SWEEP1)/180.)*PI
        SWEEP2=69.
        SWEEP2=((90.-SWEEP2)/180.)*PI
        BASE1=(XCRANK-XSTART)*TAN(SWEEP1)
        BASE2=(XEND-XCRANK)*TAN(SWEEP2)+BASE1
        XMID1=(XCRANK-XSTART)*.70+XSTART
        XMID2=XSTART+.70*LEN
C
        NJ=28
        JMID1=8
        JCRANK=13
        JMID2=24
        JEND=28
        NK=65
        KEDGE=41
        NL=70
        LSURF=25
       END IF
C
       IF (LCHECK) THEN
         WRITE (9,*) ' DELTA WING LENGTH IS  ',LEN
      IF(.NOT.LCRANK)THEN
        WRITE (9,*) ' HALF THE WING ROOT CHORD IS  ',BASE
       THICK=BASE*TAN(OMEGA)
        WRITE (9,*) ' ROOT THICKNESS IS  ',THICK
      ELSE
        WRITE (9,*) ' HALF THE WING ROOT CHORD IS  ',BASE2
       THICK=BASE2*TAN(OMEGA)
        WRITE (9,*) ' ROOT THICKNESS IS  ',THICK
      END IF
     ENDIF
C
     DO 1 J = 1,NJ
       DO 1 K = 1,NK
         DO 1 L = 1,NL
           X(J,K,L) = 0.0
           Y(J,K,L) = 0.0
           Z(J,K,L) = 0.0
   1 CONTINUE
C
C CONSTANTS FOR THE CALCULATION OF THE LIMITS OF THE DOMAIN
C
     YMAX1 = 0.15 * LEN
     ZMAX1T = 0.15 * LEN
     ZMAX1B = -0.15 * LEN
     IF(.NOT.LCRANK)THEN
       YMAX2 = 0.85 * LEN
       ZMAX2T = 1.05 * LEN
       ZMAX2B = -0.65 * LEN
     ELSE
       YMAX2 = 0.95 * LEN
       ZMAX2T = 1.05 * LEN
       ZMAX2B = -0.80 * LEN
     END IF
     XTOTSQ = XTOTAL * XTOTAL
     XTOTCU = XTOTSQ * XTOTAL
C
C  THE CONSTANT CURVE CONTROLS THE AMOUNT OF CURVATURE IN THE
C  MAX AND MIN Z PLANES.  CURVE MUST BE GREATER THAN OR EQUAL
C  TO 0 AND LESS THAN 1. IF CURVE IS EQUAL TO 0 A RECTANGULAR
C  GRID RESULTS. IF CURVE EQUALS 1 A SINGULARITY OCCURS AT
C  Y=YMAX.
C
     CURVE=0.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                     C
C  X DIRECTION                                        C
C                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C DIST X POINTS ALONG THE L = LSURF AND K = 1
C
C AHEAD OF BODY
```

```
C
   IF (LCHECK) WRITE (9,*) '  X DIRECTION'
   J = 1
   DO 90 K = 1,NK
    DO 90 L = 1,NL
      X(J,K,L) = 0.0
 90 CONTINUE
   IF (LCHECK) WRITE (9,*) J,X(J,1,LSURF)
C
   NUMB = JSTART - 1
   DIST = XSTART
   DETA = 1.0 / FLOAT(NUMB)
   DX = 20.0*DELTA
   CALL FINDC(DX,DIST,NUMB,C,1)
   DO 100 J = 2,JSTART-1
     JJ = JSTART - J
     ETA = DETA * FLOAT(JJ)
     FAC = (EXP(C*ETA) - 1.0) / (EXP(C) - 1.0)
     XTEMP = XSTART - FAC * DIST
     DO 99 K = 1,NK
      DO 99 L = 1,NL
        X(J,K,L) = XTEMP
 99   CONTINUE
     IF (LCHECK) THEN
      Q = (XSTART - XTEMP) / DX
       WRITE (9,*) J,X(J,1,LSURF),Q
     ENDIF
 100 CONTINUE
C
   J = JSTART
   DO 101 K = 1,NK
    DO 101 L = 1,NL
      X(J,K,L) = XSTART
 101 CONTINUE
   IF (LCHECK) WRITE (9,*) J,X(J,1,LSURF)
C
   IF(.NOT.LCRANK)THEN
C
C FIRST SECTION OF BODY
C
   NUMB = JMID - JSTART
   DIST = XMID - XSTART
   DETA = 1.0 / FLOAT(NUMB)
   DX = 15.0 * DELTA
   CALL FINDC(DX,DIST,NUMB,C,1)
   DO 110 J = JSTART+1,JMID-1
     JJ = J - JSTART
     ETA = DETA * FLOAT(JJ)
     FAC = (EXP(C*ETA) - 1.0) / (EXP(C) - 1.0)
     XTEMP = FAC * DIST + XSTART
     DO 109 K = 1,NK
      DO 109 L = 1,NL
        X(J,K,L) = XTEMP
 109   CONTINUE
        IF (LCHECK) THEN
         Q = (XTEMP - XSTART) / DX
          WRITE (9,*) J,X(J,1,LSURF),Q
        ENDIF
 110 CONTINUE
C
    J = JMID
    DO 111 K = 1,NK
     DO 111 L = 1,NL
       X(J,K,L) = XMID
 111 CONTINUE
     IF (LCHECK) WRITE (9,*) J,X(J,1,LSURF)
C
C NOW FOR THE SECOND PART OF THE BODY
C
    NUMB = JEND - JMID
    DIST = XEND - XMID
    DETA = 1.0 / FLOAT(NUMB)
    DX = 5.0 * DELTA
    CALL FINDC(DX,DIST,NUMB,C,1)
    DO 120 J = JMID+1,JEND-1
     JJ = JEND - J
     ETA = DETA * FLOAT(JJ)
     FAC = (EXP(C*ETA) - 1.0) / (EXP(C) - 1.0)
     XTEMP = XEND - FAC * DIST
     DO 119 K = 1,NK
      DO 119 L = 1,NL
        X(J,K,L) = XTEMP
 119   CONTINUE
        IF (LCHECK) THEN
         Q = (XEND - XTEMP) / DX
          WRITE (9,*) J,X(J,1,LSURF),Q
        ENDIF
 120 CONTINUE
C
    J = JEND
    DO 121 K = 1,NK
     DO 121 L = 1,NL
       X(J,K,L) = XEND
 121 CONTINUE
     IF (LCHECK) WRITE (9,*) J,X(J,1,LSURF)
C
C DO THE CRANKED WING
C
    ELSE
C
C  DISTRIBUTE POINTS FROM JSTART TO JMID1
C
    NUMB = JMID1 - JSTART
    DIST = XMID1 - XSTART
    DETA = 1.0 / FLOAT(NUMB)
    DX = 50.0 * DELTA
```

```
      CALL FINDC(DX,DIST,NUMB,C,1)
      DO 125 J = JSTART+1,JMID1-1
        JJ = J - JSTART
        ETA = DETA * FLOAT(JJ)
        FAC = (EXP(C*ETA) - 1.0) / (EXP(C) - 1.0)
        XTEMP = FAC * DIST + XSTART
        DO 124 K = 1,NK
         DO 124 L = 1,NL
  124       X(J,K,L) = XTEMP
        IF (LCHECK) THEN
         Q = (XTEMP - XSTART) / DX
          WRITE (9,*) J,X(J,1,LSURF),Q
        ENDIF
  125 CONTINUE
      J = JMID1
      DO 126 K = 1,NK
       DO 126 L = 1,NL
  126    X(J,K,L) = XMID1
       IF (LCHECK) WRITE (9,*) J,X(J,1,LSURF)
C
C  DISTRIBUTE POINTS FROM JMID1 TO JCRANK
C
      NUMB = JCRANK-JMID1
      DETA = 1.0 / FLOAT(NUMB)
      DX = 8.0 * DELTA
      DIST = XCRANK-.5*DX-XMID1
      CALL FINDC(DX,DIST,NUMB,C,1)
      DO 130 J = JMID1+1,JCRANK-1
        JJ = JCRANK - J
       ETA = DETA * FLOAT(JJ)
        FAC = (EXP(C*ETA) - 1.0) / (EXP(C) - 1.0)
        XTEMP = XCRANK-.5*DX-FAC*DIST
        DO 129 K = 1,NK
         DO 129 L = 1,NL
  129       X(J,K,L) = XTEMP
        IF (LCHECK) THEN
         Q = (XCRANK-.5*DX-XTEMP)/DX
          WRITE (9,*) J,X(J,1,LSURF),Q
        ENDIF
  130 CONTINUE
C
      J = JCRANK
      DO 131 K = 1,NK
       DO 131 L = 1,NL
         X(J,K,L) = XCRANK-.5*DX
  131    X(J+1,K,L) = XCRANK+.5*DX
      IF (LCHECK) THEN
        WRITE(9,*)J,X(J,1,LSURF)
        WRITE(9,*)J+1,X(J+1,1,LSURF)
      END IF
C
C  DISTRIBUTE POINTS FROM JCRANK+1 TO JMID2
C
```

```
      NUMB = JMID2 - (JCRANK+1)
      DIST = XMID2 - X(JCRANK+1,1,LSURF)
      DETA = 1.0 / FLOAT(NUMB)
      DX = 8.0 * DELTA
      CALL FINDC(DX,DIST,NUMB,C,1)
      DO 135 J = JCRANK+2,JMID2-1
        JJ = J - (JCRANK+1)
        ETA = DETA * FLOAT(JJ)
        FAC = (EXP(C*ETA) - 1.0) / (EXP(C) - 1.0)
        XTEMP = FAC * DIST + X(JCRANK+1,1,LSURF)
        DO 136 K = 1,NK
         DO 136 L = 1,NL
  136       X(J,K,L) = XTEMP
        IF (LCHECK) THEN
          Q = (XTEMP - X(JCRANK+1,1,LSURF))/DX
          WRITE (9,*) J,X(J,1,LSURF),Q
        ENDIF
  135 CONTINUE
C
      J = JMID2
      DO 137 K = 1,NK
       DO 137 L = 1,NL
  137   X(J,K,L) = XMID2
      IF (LCHECK) WRITE (9,*) J,X(J,1,LSURF)
C
C  DISTRIBUTE POINTS FROM JMID2 TO JEND
C
      DX = 5.0 * DELTA
      DO 140 K = 1,NK
       DO 140 L = 1,NL
         X(J+1,K,L) = XMID2+0.08*LEN
         X(J+2,K,L) = XMID2+0.18*LEN
  140    X(J+3,K,L) = XMID2+0.30*LEN-DX
      IF(LCHECK)THEN
       DO 141 J=JMID2+1,JEND-1
         Q=(XEND-X(J,1,LSURF))/DX
         WRITE(9,*)J,X(J,1,LSURF),Q
  141  CONTINUE
      END IF
C
      J = JEND
      DO 142 K = 1,NK
       DO 142 L = 1,NL
  142   X(J,K,L) = XEND
      IF (LCHECK) WRITE(9,*)J,X(J,1,LSURF)
      END IF
C
C CALCULATE THE LIMITS OF THE DOMAIN
C
      L = LSURF
      IF (LCHECK) WRITE (9,*) 'LIMITS   YMAX      ZMIN     ZMAX'
      DO 150 J = 1,NJ
        XSQ  = X(J,1,LSURF) * X(J,1,LSURF)
```

```
     XCU = XSQ * X(J,1,LSURF)
     FAC1 = 3.0 * XSQ / XTOTSQ
     FAC2 = 2.0 * XCU / XTOTCU
     YTEMP = (YMAX2 - YMAX1) * (FAC1 - FAC2) + YMAX1
     ZTEMPB = (ZMAX2B - ZMAX1B) * (FAC1 - FAC2) + ZMAX1B
     ZTEMPT = (ZMAX2T - ZMAX1T) * (FAC1 - FAC2) + ZMAX1T
     DO 148 L=1,NL
       Y(J,NK,L) = YTEMP
 148  CONTINUE
     DO 149 K = 1,NK
       Z(J,K,1)=ZTEMPB
       Z(J,K,NL)=ZTEMPT
 149  CONTINUE
      IF (LCHECK) WRITE (9,151) J,X(J,1,1),Y(J,NK,1),
    &     Z(J,1,1),Z(J,1,NL)
 150 CONTINUE
 151 FORMAT (I3,4E12.5)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                C
C   Y DIRECTION                                  C
C                                                C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C DISTRIBUTE POINTS ALONG THE Y AXIS AT X = XEND
C
    IF (LCHECK) WRITE (9,*) '  Y DIRECTION'
C
    IF(.NOT.LCRANK)THEN
    J = JEND
    L = LSURF
    K = 1
     Y(J,K,L) = 0.0
     IF (LCHECK) WRITE (9,*) K,Y(J,K,L)
C
C INNER SECTION OF WING
C
    YMID = BASE * .10
    NUMB = KMID-1
    DIST = YMID
    DETA = 1.0 / FLOAT(NUMB)
    DY = 0.04 * DIST
    CALL FINDC(DY,DIST,NUMB,C,1)
    DO 180 K = 2,KMID-1
      KK = K - 1
      ETA = DETA * FLOAT(KK)
      FAC = (EXP(C*ETA) - 1.0) / (EXP(C) - 1.0)
      Y(J,K,L) = FAC * DIST
      IF (LCHECK) THEN
        Q = Y(J,K,L) / DY
        PCT = (Y(J,K,L) - Y(J,K-1,L)) / BASE
        WRITE (9,*) K,Y(J,K,L),Q,PCT
      ENDIF
 180 CONTINUE
```

```
C
    K=KMID
    Y(J,K,L) = YMID
    IF(LCHECK) WRITE(9,*)K,Y(J,K,L)
C
C OUTER SECTION OF THE WING
C
    NUMB = KEDGE - KMID
    DIST = BASE - YMID
    DETA = 1.0 / FLOAT(NUMB)
    DY = 1.0 * DELTA / TAN(OMEGA)
    CALL FINDC(DY,DIST,NUMB,C,1)
    DO 190 K = KMID+1,KEDGE-1
      KK = KEDGE - K
      ETA = DETA * FLOAT(KK)
      FAC = (EXP(C*ETA) - 1.0) / (EXP(C) - 1.0)
      Y(J,K,L) = BASE - FAC * DIST
      IF (LCHECK) THEN
       Q = (BASE - Y(J,K,L)) / DY
       PCT = (Y(J,K,L) - Y(J,K-1,L)) / BASE
        WRITE (9,*) K,Y(J,K,L),Q,PCT
      ENDIF
  190 CONTINUE
C
    K = KEDGE
    Y(J,K,L) = BASE
    IF (LCHECK) WRITE (9,*) K,Y(J,K,L)
    IF (LSTOP) CALL EXIT
C
C BASED ON THE Y VALUES AT X = XEND DISTRIBUTE Y VALUES FROM
C X = XSTART TO XEND FOR K LESS THAN OR EQUAL TO KEDGE
C
    DO 210 J = JSTART+1,JEND
      FAC = (X(J,1,LSURF) - XSTART) / LEN
      DO 210 K = 2,KEDGE
        YTEMP = Y(JEND,K,LSURF) * FAC
        DO 210 L = 1,NL
          Y(J,K,L) = YTEMP
  210 CONTINUE
C
C FOR THE REGION BETWEEN 0 AND XSTART WITH K LESS THAN KEDGE,
SET ALL
C Y TO THE VALUE AT JSTART+1
C
    JP = JSTART + 1
    DO 220 J = 1,JSTART
      DO 220 K = 2,KEDGE
        DO 220 L = 1,NL
          Y(J,K,L) = Y(JP,K,LSURF)
  220 CONTINUE
C
C  DO THE CRANKED WING
C
```

```
     ELSE
C
C  DISTRIBUTE POINTS ALONG JEND
C
     J = JEND
     L = LSURF
     K = 1
     Y(J,K,L) = 0.0
     IF (LCHECK) WRITE (9,*) K,Y(J,K,L)
C
C INNER SECTION OF WING
C
     YMID2 = BASE2 * 0.08
     NUMB = KMID-1
     DIST = YMID2
     DETA = 1.0 / FLOAT(NUMB)
     DY = 0.10 * DIST
     CALL FINDC(DY,DIST,NUMB,C,1)
     DO 235 K = 2,KMID-1
       KK = K - 1
       ETA = DETA * FLOAT(KK)
       FAC = (EXP(C*ETA) - 1.0) / (EXP(C) - 1.0)
       Y(J,K,L) = FAC * DIST
       IF (LCHECK) THEN
         Q = Y(J,K,L) / DY
         PCT = (Y(J,K,L) - Y(J,K-1,L)) / BASE2
         WRITE (9,*) K,Y(J,K,L),Q,PCT
       ENDIF
 235 CONTINUE
C
     K=KMID
     Y(J,K,L) = YMID2
     IF(LCHECK) WRITE(9,*)K,Y(J,K,L)
C
C OUTER SECTION OF THE WING
C
     NUMB = KEDGE - KMID
     DIST = BASE2 - YMID2
     DETA = 1.0 / FLOAT(NUMB)
     DY = 1.0 * DELTA / TAN(OMEGA)
     CALL FINDC(DY,DIST,NUMB,C,1)
     DO 240 K = KMID+1,KEDGE-1
       KK = KEDGE - K
       ETA = DETA * FLOAT(KK)
       FAC = (EXP(C*ETA) - 1.0) / (EXP(C) - 1.0)
       Y(J,K,L) = BASE2 - FAC * DIST
       IF (LCHECK) THEN
         Q = (BASE2 - Y(J,K,L)) / DY
         PCT = (Y(J,K,L) - Y(J,K-1,L)) / BASE2
         WRITE (9,*) K,Y(J,K,L),Q,PCT
       ENDIF
 240 CONTINUE
C
```

```
      K = KEDGE
      Y(J,K,L) = BASE2
      IF (LCHECK) WRITE (9,*) K,Y(J,K,L)
C
C BASED ON THE Y VALUES AT X = XEND DISTRIBUTE Y VALUES FROM
C X = XSTART TO XEND FOR K LESS THAN OR EQUAL TO KEDGE
C
      DO 244 J = JSTART+1,JEND
       IF(J.LE.JCRANK) THEN
        Y(J,KEDGE,LSURF)=(X(J,1,LSURF)-XSTART)*TAN(SWEEP1)
       ELSE
        Y(J,KEDGE,LSURF)=(XCRANK-XSTART)*TAN(SWEEP1)+
     1      (X(J,1,LSURF)-XCRANK)*TAN(SWEEP2)
       END IF
       FAC=Y(J,KEDGE,LSURF)/Y(JEND,KEDGE,LSURF)
       DO 244 K = 2,KEDGE
        YTEMP = Y(JEND,K,LSURF) * FAC
        DO 244 L = 1,NL
         Y(J,K,L) = YTEMP
  244 CONTINUE
C
C FOR THE REGION BETWEEN 0 AND XSTART WITH K LESS THAN KEDGE,
SET ALL
C Y TO THE VALUE AT JSTART+1
C
      JP = JSTART + 1
      DO 246 J = 1,JSTART
       DO 246 K = 2,KEDGE
        DO 246 L = 1,NL
          Y(J,K,L) = Y(JP,K,LSURF)
  246 CONTINUE
      END IF
C
C SET Y FROM KEDGE TO NK
C
      DO 250 J = 1,NJ
       DIST = Y(J,NK,LSURF) - Y(J,KEDGE,LSURF)
       DY = Y(J,KEDGE,LSURF) - Y(J,KEDGE-1,LSURF)
       NUMB = NK - KEDGE
       DETA = 1.0 / FLOAT(NUMB)
       CALL FINDC(DY,DIST,NUMB,C,1)
       DO 250 K = KEDGE+1,NK
        KK = K - KEDGE
        ETA = DETA * FLOAT(KK)
        FAC = (EXP(C*ETA) - 1.0) / (EXP(C) - 1.0)
        YTEMP = Y(J,KEDGE,LSURF) + FAC * DIST
        DO 250 L = 1,NL
         Y(J,K,L) = YTEMP
  250 CONTINUE
C
C DEFINE MAX AND MIN PLANES IN THE Z DIRECTION
C
      DO 260 J=1,NJ
      DO 260 K=1,NK
        Z(J,K,1)=Z(J,K,1)*((1.-(CURVE*Y(J,K,1)/
     1           Y(J,NK,1))**2)**.5)
        Z(J,K,NL)=Z(J,K,NL)*((1.-(CURVE*Y(J,K,NL)/
     1           Y(J,NK,NL))**2)**.5)
  260 CONTINUE
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                    C
C  Z DIRECTION                                       C
C                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C CALCULATE THE Z VALUES ABOVE THE SURFACE FROM THE ORIGIN
C TO XTOTAL
C
      NUMB = NL - LSURF
      DETA = 1.0 / FLOAT(NUMB)
      FAC1 = DELTA * (X(JSTART+1,1,LSURF) - XSTART) / LEN
      DO 280 J = 1,NJ
        IF (J .LE. JSTART) THEN
          DZ = FAC1
         ELSEIF (J .LT. JEND) THEN
          FAC = (X(J,1,LSURF) - XSTART) / LEN
          DZ = DELTA * FAC
         ELSE
          DZ = DELTA
        ENDIF
        DO 280 K=1,NK
         DIST=Z(J,K,NL)
         CALL FINDC(DZ,DIST,NUMB,C,1)
         DO 280 L = LSURF+1,NL
          LL = L - LSURF
          ETA = DETA * FLOAT(LL)
          FAC = (EXP(C*ETA) - 1.0) / (EXP(C) - 1.0)
          Z(J,K,L)= FAC * DIST
  280 CONTINUE
      IF (LCHECK) THEN
        WRITE (9,*) ' Z DIRECTION'
        J = JEND
        K = 1
        DZ = Z(J,K,LSURF+1) - Z(J,K,LSURF)
        DO 285 L = LSURF,NL
         Q = Z(J,K,L) / DZ
         WRITE (9,*) L,Z(J,K,L),Q
  285   CONTINUE
      ENDIF
C
C NOW FOR THE BOTTOM
C
      S2 = TAN(OMEGA)
      NUMB = LSURF - 2
      DETA = 1.0 / FLOAT(NUMB)
```

```
    DO 290 J = JSTART+1,JEND
     DO 290 K = 1,NK
C
C FIRST LOCATE THE Z VALUE OF THE BOTTOM SURFACE
C
       IF(K.LT.KEDGE)THEN
         B2 = -S2 * Y(J,KEDGE,LSURF)
         Z(J,K,LSURF-1) = Y(J,K,LSURF) * S2 + B2
       ELSE
         Z(J,K,LSURF-1)=Z(J,KEDGE-1,LSURF-1)
       END IF
C
C CALCULATE THE LOCAL DELTA
C
       FAC = (X(J,1,LSURF) - XSTART) /LEN
       DZ = DELTA * FAC
        DIST = ABS(Z(J,K,1) - Z(J,K,LSURF-1))
       CALL FINDC(DZ,DIST,NUMB,C,1)
       DO 290 L = 1,LSURF-2
         LL = LSURF - L - 1
        ETA = DETA * FLOAT(LL)
         FAC = (EXP(C*ETA) - 1.0) / (EXP(C) - 1.0)
         Z(J,K,L)=Z(J,K,LSURF-1)-FAC*DIST
 290 CONTINUE
C
C REGION BETWEEN 0 AND XSTART, Z VALUES ARE THOSE AT XSTART+1
C
    JP = JSTART + 1
    Z1 = Z(JP,1,LSURF+1) - Z(JP,1,LSURF)
    Z2 = Z(JP,1,LSURF) - Z(JP,1,LSURF-1)
    DZ = Z2 - Z1
    DO 310 L = 1,LSURF-1
     ZTEMP = Z(JP,1,L) + DZ
     DO 310 J = 1,JSTART
      DO 310 K = 1,NK
        Z(J,K,L) = ZTEMP
 310 CONTINUE
C
C OUTPUT LOOPS
C
C ARC3D LOOP
C
    WRITE (10) NJ,NK,NL
    DO 999 L = 1,NL
       WRITE (10) ((X(J,K,L),J = 1,NJ),K = 1,NK),
    *           ((Y(J,K,L),J = 1,NJ),K = 1,NK),
    *           ((Z(J,K,L),J = 1,NJ),K = 1,NK)
 999 CONTINUE
C
C  WRITE OUT GRID PLANES FOR XYGRID PLOTTING
C
     OPEN(11,FILE='JPLANE',STATUS='NEW',FORM='FORMATTED')
     OPEN(12,FILE='KPLANE',STATUS='NEW',FORM='FORMATTED')
```

```
      OPEN(13,FILE='LPLANE',STATUS='NEW',FORM='FORMATTED')
C
      J=JEND
      DO 1000 K=1,NK
        DO 1000 L=1,NL
 1000   WRITE(11,1030)Y(J,K,L),Z(J,K,L)
      K=1
      DO 1010 J=1,NJ
        DO 1010 L=1,NL
 1010   WRITE(12,1030)X(J,K,L),Z(J,K,L)
      L=LSURF
      DO 1020 J=1,NJ
        DO 1020 K=1,NK
 1020   WRITE(13,1030)X(J,K,L),Y(J,K,L)
 1030 FORMAT(2E15.7)
C
C IRIS LOOP
C       WRITE (10) NJ,NK,NL
C       WRITE (10) (((X(J,K,L),J = 1,NJ),K = 1,NK),L = 1,NL),
C     &            (((Y(J,K,L),J = 1,NJ),K = 1,NK),L = 1,NL),
C     &            (((Z(J,K,L),J = 1,NJ),K = 1,NK),L = 1,NL)
C
      CALL EXIT
C      STOP
      END
C
C  SUBROUTINE FINDC
C
      SUBROUTINE FINDC(DELTA,DIST,ICELL,C,IDIV)
      C = 2.00
      ETA = FLOAT(IDIV) / FLOAT(ICELL)
      DIV = IDIV
       DO 100 I = 1,20
        EC = EXP(C)
        ECM1 = EC - 1.0
        ECD = EXP(C * ETA)
        ECDM1 = ECD - 1.0
        BOTTOM = ECM1 * ECD * ETA - ECDM1 * EC
        TOP = DELTA - DIST * ECDM1 / ECM1
        C = C + TOP / BOTTOM / DIST * ECM1 * ECM1
  100  CONTINUE
      RETURN
      END
```

Bibliography

1. Lamar J. E.,"Analysis and Design of Strake-Wing Configurations", Journal of Aircraft. Vol 17: pp20-27, (1980).

2. Newsome R. W. and Kandil O. A., "Vortical Flow Aerodynamics-Physical and Numerical Simulation", AIAA Paper 87-0205, (1987).

3. Payne F. M. and Nelson R. C., " An Experimental Investigation of Vortex Breakdown on a Delta Wing", NASA CP-2416, (1985).

4. Stanbrook A. and Squire L. C., "Possible Types of Flow at Swept Leading Edges", Aeronautical Quarterly, Vol 15, Pt 1: pp 72-82 (Feb 1964).

5. Miller D. S. and Wood R. M., "Leeside Flow Over Delta Wings at Supersonic Speeds", NASA TP-2430, (1985).

6. Polhamus E. C., "A Concept of the Vortex Lift of a Sharp-Edge Delta Wing Based on a Leading-Edge Suction Analogy", NASA TN D-3767, (1966)

7. Morchoisne Y., "Modeling Unsteady Flows by the Method of Point Vortices", ONERA TP 1986-30, (1986).

8. Xieyman Y., "Roll Up of Strake Leading/Trailing-Edge Vortex Sheets for Double - Delta Wings", Journal of Aircraft, Vol 22: pp 87-89, (1985).

9. Fornasier L. and Rizzi A., "Comparison of Results from a Panel Method and an Euler Code for Cranked Delta Wing", AIAA Paper 85-4091, (1985).

10. Wood R. M. And Colvell P. F., "Experimental and Theoretical Study of the Longitudinal Aerodynamic Characteristics of Delta and Double-Delta Wings at Mach Numbers of 1.6, 1.9, and 2.16", NASA TP-2433, (1985).

11. Hewitt B. L. and Kellaway W., "Developments in the Lifting Surface Theory Treatment of Symmetric Planforms with a Leading Edge Crank in Subsonic Flow", Aeronautical Research Council CP-1323, (1972).

12. Strang W. Z., Berdahl C. H., Nutley E. L., and Murn A. J., "Evaluation of Four Panel Aerodynamic Prediction Methods", AIAA Paper 85-4092, (1985).

13. Nathman J. K., "Analysis of Leading-Edge Vortices on Complex Configurations", AIAA Paper 85-4054, (1985).

14. Hoeijmakers H. W., Vaatstra W., and Verhaagen N. G., "Vortex Flow Over Delta and Double-Delta Wings", Journal of Aircraft Vol 20: pp825-832, (1983).

15. Rizzi A. and Purcell C., "Numerical Experiment with Inviscid Vortex-Streched Flow Around a Cranked Delta Wing: Supersonic Speed", Engineering Computations, Vol 3: pp 230-234, (1986).

16. Rizzi A. and Purcell C., "Numerical Experiment with Inviscid Vortex-Streched Flow Around a Cranked Delta Wing: Subsonic Speed", AIAA Paper 85-4091, (1985).

17. Rizzi A., "On the Computation of Transonic Leading-Edge Vortices Using the Euler Equations", Journal of Fluid Mechanics, Vol 181: pp 163-195, (1987).

18. Eriksson L. E., Smith R. W., and Wiese M. R., "Grid Generation and Inviscid Flow Computation About Crank-Winged Airplane Geometries", AIAA Paper 87-1125, (1987).

19. Fujii K., Gavali S., and Holst T., "Evaluation of Navier-Stokes and Euler Solutions for Leading Edge Seperation Vortices", NASA TM-89458, (1987).

20. Kerlick G. D., Klopfer G. H., and Mion D., "A Numerical Study of Strake Aerodynamics"; Nielson Engineering and Research TR-270, (1985).

21. Hsu C. H. and Hartwick P. M., "Computation of Vortical Interaction for a Sharp Edged Double Delta Wing", AIAA Paper 87-0206, (1987).

22. Hsu C. H., "Navier-Stokes Computation of Flow Around a Round-Edged Double-Delta Wing", AIAA Paper 88-2560, (1988).

23. Hsu C. H. and Hartwick P. M., "Incompressible Navier-Stokes Computations of Vortical Flows Over Double-Delta Wings", AIAA Paper 87-1341, (1987).

24 Fujii K. and Schiff L., "Numerical Simulation of Vortical Flows Over a Straked Wing", AIAA Paper 87-1229, (1987).

25. Fujii K. and Kutler P., "Numerical Simulation of the Leading-Edge Seperation Vortex for a Wing and Strake Wing Configuration", AIAA Paper 83-1908, (1983).

26. Ganzer U. and Szodruch J., "Vortex Formation over Delta, Double Delta, and Wave Rider Configurations at Supersonic Speeds", AGARD CP-428, (1987).

27. Ausherman D. W., "Surface Pressure Measurements on a Double Delta Wing/Body Configuration at Mach 2 and Mach 3", NSWC/MP/86-240, (1986).

28. Sevier J. R., "Aerodynamic Characteristics at Mach Numbers of 1.41 and 2.01 of a Series of Cranked Wings Ranging in Aspect Ratio From 4.00 to 1.74 in Combination With a Body", NASA

29. Nathman J. K., Norton D. J., and Rao B. M., "An Experimental Investigation of Incompressible Flow Over Delta and Double-Delta Wings", ONR CR 215-231-3, (1976).

30. Brennenstuhl U. and Hummel D., "Vortex Formation Over Double-Delta Wings", ICAS Paper 82-6.6.3, (1982).

31. Henke R., "Wind Tunnel Tests of Delta Wings", College of Aeronautics Memo 8216, Cranfield Institute of Technology, (1982).

32. Pulliam T. H., "Efficient Solution Methods for the Navier-Stokes Equations", Lecture Notes for the Von Karmen Institute for Fluid Dynamics Lecture Series: Numerical Techniques for Viscous Flow Computation in Turbomachinery Bladings, Brussels, Belgium, January 1986.

33. Rizzetta D. P. and Shang J. S. "Numerical Simulation of Leading-Edge Vortex Flows", AIAA Paper 84-1544, (1984).

34. Buter T. A. and Rizzetta D. P. "Steady Supersonic Navier-Stokes Solutions of a 75° Delta Wing", NASA CP-2416, (1985).

35. Monnerie, B. and Werle H., "Study of Supersonic and Hypersonic Flow About a Slender Wing at an Angle of Attack", AGARD CP-30, (1968). (in French)

36. Thomas J. L and Newsome R. W., "Navier-Stokes Computations of Leeside Flows Over Delta Wings", AIAA Paper 86-1049, (1986).

37. Vigneron Y. C., Rakich J. V. and Tannehill J. C. "Calculation of Supersonic Viscous Flow over Delta Wings with Sharp Subsonic Leading Edges", AIAA Paper 78-1137, (1978).

38. Pulliam T. H. and Steger J. L., "Implicit Finite-Difference Simulations of Three Dimensional Compressible Flow", AIAA Journal, Vol 18: pp159-167, (1980).

39. Anderson D. A., Tannehill J. C. and Pletcher R. H.. Computational Fluid Dynamics and Heat Transfer. New York: Hemisphere Publishing Corporation, 1984.

40. Warming R. F. and Beam R. M., "On the Construction and Application of Implicit Factored Schemes for Conservation Laws", SIAM-AMS Proceedings, Vol 11 : pp 85-129, (1978).

41 Lindemuth I. and Killeen J., "Alternating Direction Implicit Techniques for Two Dimensional Magnetohydrodynamics Calculations", Journal of Computational Physics, Vol 13: pp181-208, (1973).

42. McDonald H. and Briley W., "Three Dimensional Supersonic Flow of a Viscous or Inviscid Gas", Journal of Computational Physics, Vol 19: pp150-178, (1975).

43. Steger J. L., "Implicit Finite-Difference Simulation of Flow About Arbitrary Geometries with Application to Airfoils", AIAA Paper 77-665, (1977).

44. Warming R., Beam R., and Hyett B., "Diagonalization and simultaneous Symmetrization of the Gas-Dynamic Matrices", Mathematics of Computation, Vol 22: pp 1037-1045, (1975).

45. Pulliam T. H. and Chaussee D. S., "A Diagonal Form of an Implicit Approximate Factorization Algorithm", Journal of Computational Physics, Vol 39: pp347-363, (1981).

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

47. Pulliam T. H., "Artificial Dissipation Models for the Euler Equations", AIAA Paper 85-0438, (1985).

48. Vinokur M., "Conservation Equations of Gas-Dynamics in Curvilinear Coordinate Systems", Journal of Computational Physics, Vol 14: pp105-125, (1974).

Vita

Captain Francis R. Smith was born [redacted]
He graduated from Ocean City High School in Ocean City, New Jersey in 1976. He enlisted in the United States Air Force and served as a Jet Engine Mechanic until August 1978. He attended Syracuse University under the Airman Scholarship and Commissioning Program and received his Bachelor of Science Degree in Aerospace Engineering. Upon graduation he received his commission through the Reserve Officers Training Corps and was assigned to the Air Force Wright Aeronautical Laboratories at Wright-Patterson AFB. Ohio. He then served as a project engineer on the Mission Adaptive Wing Program. In May 1987 he entered the School of Engineering, Air Force Institute of Technology.

[redacted]

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188
1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimted.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GAE/AA/88D-34	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering; 6b. OFFICE SYMBOL (If applicable) AFIT/ENY	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
See Box 19

12. PERSONAL AUTHOR(S) Smith, Francis R., B.S., Captain USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
MS Thesis	FROM ________ TO ______	1988 December	97

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Numerical Analysis, Applied Aerodynamics Computational Fluid Dynamics, Aircraft Performance
20	04		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: THIN-LAYER NAVIER-STOKES SOLUTIONS FOR A CRANKED DELTA WING

Thesis Chairman: Ahmad Halim
Associate Professor of Aerospace Engineering

[illegible]
10 Jan 89

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Ahmad Halim	22b. TELEPHONE (Include Area Code) 513-255-2040; 22c. OFFICE SYMBOL AFIT/ENY

DD Form 1473, JUN 86 Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE

Block 19

For thin, highly swept wings operating at moderate to high angles of attack, the flow over the wing is dominated by the formation of leading edge vortices. These vortices produce a minimum pressure and this results in an additional lift increment. This lift increment is nonlinear with angle of attack and cannot be accurately predicted using present design methods.

The thin-layer Navier-Stokes equations were used to calculate the flow over a straight delta wing and a cranked delta wing. The straight delta wing was used as the test case due to the availability of both experimental and numerical data. Results are compared with this data in order to validate the numerical procedure. The computer code uses an implicit, time marching algorithm developed by Beam and Warming. The solution is marched in time until a steady state is achieved. The code is approximately factored and diagonalized in order to reduce computational work. A solid state disk is used in order to allow for the large grid needed for a three dimensional solution.

The thin-layer Navier-Stokes equations are capable of accurately calculating vortical flows. The cranked delta wing exhibited flow similar to a straight delta wing upstream of the crank. The vortex generated at the crank quickly became paired with the vortex from the front of the wing. The vortex location aft of the crank changes with streamwise location. The grid resolution is important when trying to calculate vortical flows, due to the large gradients in both the spanwise and normal directions. The solid state disk can be used to run problems that require more computer memory than is available. Optimization of the program input/output should be done for running the code with the solid state disk in order to reduce the central processor unit time and job cost.