FIRST STAGES OF A VISCOUS FINITE ELEMENT SOLVER FOR NON-INERTIAL AND AEROELASTIC PROBLEMS

By

NICHOLAS J MOFFITT

Bachelor of Science

Oklahoma State University

Stillwater, Oklahoma

2002

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December 2004

FIRST STAGES OF A VISCOUS FINITE ELEMENT

SOLVER FOR NON-INERTIAL AND

AEROELASTIC PROBLEMS

Thesis Approved:

F. S. Arena

Thesis Advisor

D. G. Lilley

F. W. Chambers

A. G. Emslie

Dean of Graduate College

ACKNOWLEDGEMENTS

I would like to thank several people for their contributions to my thesis work and life during this research:

Dr. Arena: I would like to begin by thanking you for the unending encouragement and enthusiasm. Thank you for pushing me through the long hours and grueling loads. Thank you for most of all for your ever present enthusiasm to generate better viscous solutions. And thank you for the discussions that solidified this research.

Dr. Chambers and Dr. Lilley: Thank you for sitting as my committee and pushing me to achieve to higher standards of research.

Tim Cowan: Thank you for answering all of my questions about the development of equations and implementation in *Euler2d*. I am also grateful that you made many allowances for future expansions so that this research could be possible.

To my friends and coworkers: Thank you, Charles, for all of the long talks that prompted much of the discussions in this work. And thank you for assisting me as I learned to work with *Euler2d* and other programs. Thank you, Miranda and Robert, for helping to run several test cases, without which I might not have finished. Mark, Deric, Austin, Steve, and Aaron, thank you for being there when I needed to complain or just needed to get away for a while.

To my family: Thank you, Mom and Dad, for supporting me, not only emotionally, but by helping with all my needs when the time arose. Thank you for standing by me as I worked through my Masters' work, and thank you for being supportive of all the other decisions that I have made in my academic career. And to my sister Heather, thank you for living your life with that never say die attitude. You have been an encouragement to me as you compete to live life to its fullest.

I would also like to thank Dr. Gupta, the NASA Dryden STARS group, and NASA Space Grant for supporting this research; and for the contributions of Oklahoma State University and the Hanson Trust for contributions to my education.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

NOMENCLATURE

The names and descriptions of control variables and array names used in the code are given in Table 4-4, 4-5, and 4-6.

Common Acronyms

CFD	Computational Fluid Dynamics
CFL	Courant-Fredrich-Lewy Number
FD	Finite Difference Method(s)
FEM	Finite Element Method(s)
FV	Finite Volume Method(s)
GMRES	Generalized Minimal Residual Algorithm
GSMAC	Generalized and Simplified Marker-and-Cell Method
LSM	Least Squares Method
RHS	Right-hand-side (in reference to right-hand-side of governing equations or right-hand-side vector)
SUPG	Streamline Upwind Petrov-Galerkin Method

Arabic Letters:

$[A]$	Inverse of Jacobean matrix ($[A]=[J]^{-1}$)
A_e	Area of two-dimensional element
a_x, a_y	Acceleration in the x- and y-directions, respectively (non-inertial frame)
c	Chord of airfoil, ellipse, or plate
c_f	Coefficient of friction
C_m	Aerodynamic moment coefficient (about the origin of mesh)
c_p, c_v	Specific heat at constant pressure and constant volume
C_x, C_y	Aerodynamic force coefficients (in mesh x- and y-directions)
C_θ	Cosine of angle between global and local (boundary edge) coordinates
D	Diameter of cylinder
e	Internal energy of the flow
f_i	Generalized force vector for mode i
F_i, G_i	Euler advection/pressure flux terms
F_v, G_v	Viscous flux terms
h	Total energy or enthalpy of the flow ($h = e + p / \rho$)
k	Coefficient of thermal conductivity
Kn	Knudson number ($Kn = \ell / L$)
$[J]$	Jacobean matrix

ℓ	Mean free path of the fluid
L	Characteristic length of the flow
LBE	Limits to Boundary Edge groups (4 groups; 8 limits)
Ma	Mach number ($Ma = U / a$)
$[M_e]$	Element "consistent" mass matrix
$[M_L]$	Lumped mass matrix
$\hat{n}$	Outward facing normal vector on Γ for 2-D domain
NSD	Number of singular nodes (inviscid and viscous)
$NSDV$	Number of singular viscous nodes
n_x, n_y	Components of outward facing normal vector
NWL	Number of wall nodes (inviscid and viscous)
$NWLV$	Number of viscous wall nodes
p	Thermodynamic pressure; dimensionless after Section
Pr	Prandtl number ($Pr = c_p \mu_0 / k_0$)
q	Heat flux
R	Gas constant for a particular fluid
Re	Reynolds number ($Re = \rho_0 U_0 L / \mu_0$)
R_x, R_y	Radius from origin of mesh to center of boundary edge
S	Sutherland's constant
S_{mod}	Modified Sutherland's constant ($S_{mod} = c_v S / U_0^2$)
S_r	Source term
S_θ	Sine of angle between global and local (boundary edge) coordinates
T	Absolution temperature
$\boldsymbol{U}$	Unknowns vector ($\boldsymbol{U}^{\mathrm{T}} = \{ \rho, u, v, p, \rho h \}$)
u, v	Velocities in the x- and y-directions, respectively
U_e	External velocity
U_i	Euler flux term
V	Velocity vector ($V = \{ u, v \}$)
VM_{real}	Virtual memory required for 8 byte float variables
$VM_{integer}$	Virtual memory required for integer variables
w_j	Gaussian quadrature weighting factor
W_{ij}	Segment weighting factor for element edge data, between nodes i and j

Greek Letters:

α , β	van der Walls coefficients
$\dot{\alpha}$	Rotational velocity of the body (non-inertial frame)
γ	Ratio of specific heats
Γ	Boundary of the spatial domain
δ_{ij}	Kronecker delta
Δt	Time step
Δt_{adv}	Time for fluid to move across an element
Δx	Segment length
ε	Strain

η	Local coordinate for one-dimensional elements ($-1 \leq \eta \leq 1$)
θ	Angle of rotation between global and local (boundary edge) coordinates
κ	Coefficient of bulk viscosity
λ	Second coefficient of viscosity
λ_{ij}	Eigenvalue at node *i* for segment spanning between node *i* and *j*
μ	First coefficient of viscosity; dynamic viscosity
ν	Kinematic viscosity ($\nu = \mu / \rho$)
ξ_1, ξ_2, ξ_3	Local coordinate for two-dimensional elements ($0 \leq \xi_1, \xi_2, \xi_3 \leq 1$)
ρ	Density of the fluid
τ_{ij}	Stress tensor on element of fluid
τ'_{ij}	Amended stress tensor (i.e., excludes pressure)
Φ	Viscous dissipation term
ϕ_i	Displacement of mode *i* at node *j*
Ω	Spatial domain

<u>Miscellaneous Symbols</u>:

$[\ ]_0$	Free stream condition
$[\ ]_e$	Referenced to the element
$[\ ]_L$	Referenced to the reference length, *L*
$[\ ]_r$	Relative velocity or energy (non-inertial frame)
$[\ ]_t$	Transformation acceleration or velocity (non-inertial frame)
$[\ ]_x$	Referenced to local segment length
$[\ ]_{xy}$	Referenced to xy-plane
$[\ ]_\delta$	Referenced to boundary layer displacement thickness
$[\ ]^*$	Dimensionless quantity
$[\ ]^{(0)}$	Reference condition
$[\ ]^T$	Transpose of matrix or vector
$\lvert\ \rvert$	Magnitude of a vector
$\hat{x}$	Boundary edge local coordinate or velocity (does not apply to $\hat{n}$)
$\vec{x}$ or $\{\ \}$	Vector notation

CHAPTER 1

1. Introduction

This thesis presents the appropriate adaptations to a non-inertial finite element Euler solver to include all of the terms of the two-dimensional Navier-Stokes equations. The original solver was written by Cowan (2003) in order to simulate aeroservoelastic applications in the non-inertial reference frame. This thesis expands the two-dimensional solver of Cowan to include the viscous and conduction terms present in the full Navier-Stokes equations.

This chapter will continue by discussing the background of viscous finite element solvers and will give examples as to the state-of-the-art. The next section will define the scope of the research objective, and the final section will give a brief overview of this thesis.

1.1. Background

The full set of Navier-Stokes equations is a complex system of nonlinear partial differential equations. Because of the complexity of their representation, no exact solution for the equations is available. Only a handful of exact solutions are available for simplified forms of the equations: Couette flow, Stokes' first and second problems, flow between two rotating cylinder, and various others. Several numerical solutions are available to less simplified, approximate forms. One such numerical solution is the

Blasius boundary layer. In order to represent the Navier-Stokes equations for more general geometries and boundary conditions, several discrete methods have been formed to solve the multi-dimensional partial differential equations using algebraic equations. Three of the most popular methods are the finite difference, finite volume, and finite element methods. The finite difference approach uses Taylor series expansions to approximate the time and spatial derivatives. This method is the most simple to implement but requires a structured mesh in order to retain the simplicity. Finite volume increases the complexity of the solver while expanding the capabilities to more complex geometry and unstructured meshes. But finite volume requires either many small elements or higher order approximations in order to represent complicated flow fields. Finite element, being the most complex method, can be used to represent complicated flows with fewer, less complex elements. For these reasons, Cowan (2003) chose to implement a finite element scheme to solve the compressible Euler equations.

Many sources can be found in the literature covering finite element solutions in general. Most of these references handle structural finite element solvers. Using the knowledge base at the time, Baker (1983) wrote a book for those who had previously worked on structural FEM (Finite Element Methods) and wanted to begin solving fluid dynamics equations. Similarly, many other applications can be made from the structural FEM literature to fluid solvers, including numerical procedures, finite element methods, and common problems with finite elements. The majority of fluids research involving FEM has involved incompressible, inviscid solvers. These solvers were expanded into compressible Euler solvers and then viscous terms were explored in recent years. The majority of research began in fluids FEM solvers that were written to solve specified

geometries. As time progressed, the solution of more and more complex configurations were tested until a few general fluids FEM solvers were assembled. Because this research expands Cowan's Euler solver to include the full Navier-Stokes equations, viscous finite element solvers are reviewed in the following paragraphs.

Table 1-1 shows a quick overview of some of the viscous finite element models that have been documented in the literature in the last 15 years. The table lists the name of the author and date of publication shown in the bibliography. The table also includes the finite element method used, temporal and spatial orders, element shape, dissipation model, and whether the solver is implicit, explicit, or mixed. An overview of these references is given below, followed by specifics of an key points from these papers.

Table 1-1: Overview of Viscous Finite Element References

Date	Author	Method	Imp/Exp	Temporal Order	Spatial Order	Element Shape	Dissipation Model
1990	Bristeau	G, GMRES	Imp	1st & 2nd	1st	TRI	
1990	Jiang	LSM			1st	QUAD	
1990	Tanahashi	GSMAC	Exp	1st or P-C	1st	QUAD	
1991	Fernandez	G	Imp	1st	1st	TRI	Upwinding
1992	Marcum	G	Exp	RK or L-W	1st	TRI	2nd & 4th order RK; 2nd L-W
1992	Tworzydlo	TG	Imp/Exp	2nd	1st	QUAD	Art. Diss.
1994	Kallinderis	G	Imp/Exp		1st	QUAD, TRI (sub.)	4th order smoothing
1994	Soulaimani	LSM, GMRES	Imp	2nd	1st; 2nd vel.	TRI	SUPG
1995	Barsoum	PG	Imp	1st	1st	QUAD	
1996	Mittal	SUPG, GMRES		2nd		QUAD	SUPG
1997	Masud	SUPG & LSM	Imp	1st	1st, H.O.	TRI	SUPG
1998	Bonhaus	SUPG, GMRES	Imp	1st	1st, H.O.	TRI	SUPG
1999	Arminjon	MFVFE	Exp	2 step P-C	1st	TRI	
1999	Whiting	SUPG & LSM	Imp	2nd or P-MC	1st, H.O.	TRI	SUPG
2000	Jakobsen	SUPG				QUAD	SUPG
2000	N'dri	G	Imp	1st	1st	TRI	
2000	Wood	TG	Exp	1st	1st	TRI	
2004	Luo	MFVFE				TRI	

G = Galerkin
TG = Taylor-Galerkin
PG = Petrov-Galerkin
SUPG = Streamline Upwind Petrov Galerkin
LSM = Least-Squares Method
GSMAC = Generalized Simplified Marker-And-Cell
MFVFE = Mixed Finite Volume & Finite Element Method
GMRES = Generalized Minimal Residual

TRI = Triangular elements
QUAD = Quadralateral elements

P-C = Predictor-Corrector
P-MC = Predictor-Multicorrector
RK = Runge Kutta
L-W = Lax-Wandroff

All of the examples shown in Table 1-1used some form of Galerkin methodology. Fernandez (1991) illustrates a basic Galerkin method mixed with an upwinding scheme.

(Galerkin's method has been left for a full discussion in Section 3.1). The upwinding scheme was implemented in order to help alleviate instability problems in the advection terms. This method is more common in early Euler literature than in the viscous FEM literature shown here. Most of the cases above utilized the SUPG, or Streamline Upwind Petrov-Galerkin, method (Mittal, 1998; Masud, 1997; Bonhaus, 1998; Whiting, 1999; Jakobsen, 2000). SUPG is similar to the upwinding of Fernandez, but SUPG uses an adaptive shape function to implement the upwinding in only the momentum equations. Similar methods are illustrated in the literature for pressure and density. SUPG is considered to be a finite element methodology because SUPG defines the shape function used when developing the finite element equations; but, SUPG is also specified as the dissipation methodology by many papers while Petrov-Galerkin is stringently referenced as a method for developing the FEM equations. Even though the Galerkin or variation thereof, are the most widely used methodologies, Jiang (1990) says that the Least-Squares Methodology (LSM) is the only means to a "general-purpose code…[and] unified method". Jiang goes on to say that all of the *different* Galerkin methods are useful in different aspects and flow regimes, whereas LSM can be used in all aspects and regimes of computational fluid dynamics. This idea is not shown anywhere else in the literature, but other authors, such as Masud (1998) and Whiting (1999) use LSM as an alternative methology. The final methodology to discuss is the GMRES, or generalized minimal residual, solution "method". The GMRES algorithm is outlined in non-linear form by Bristeau (1990) and is better described as a numerical technique to solver the equations established by the various solution methods above. Although Bristeau is the

only reference shown to outline the technique, many other sources implement GMRES in their solvers, including Soulaimani (1994), Mittal (1998), and Bonhaus (1998).

Since finite element is rooted in finite difference (FD) and finite volume (FV) methods, one would expect to see many similarities other than those previously mentioned. Arminjon (1999) combines a finite volume approximation of the advection terms of the N-S equations with a finite element approximation of the diffusion terms. This combination allowed Arminjon to represent the diffusion terms with a lower order shape function than would have been possible with a full finite volume solver. This method illustrates the power of FEM versus the simplicity of FV while keeping the unstructure mesh capabilities of both methods. Mixed FEM and FV methods are also used by Luo (2004). Tanahashi (1990) illustrates the application of a finite difference concept MAC (marker-and-cell) to finite elements. Tanahashi used the velocity vectors on the edges (or vertices) and total energy at the center of non-orthogonal, quadrilateral elements to represent the GSMAC (generalized and simplified MAC) method. GSMAC shows how finite element can be used to expand the structured methods of FD to the unstructured applications of FEM.

Dealing with the solution technique to solve the FEM equations, Tworzydlo (1992) implemented a mixed implicit-explicit scheme on a Taylor-Galerkin method. Three parameters were used to adjust the amount of implicit or explicit used to solve the advection, diffusion, and other terms of the equations. Using these parameters, Tworzydlo explored the linearized analytical stability of all combinations of these parameters. Tworzydlo found that fully explicit schemes were only stable for CFL factors less than unity and that the fully implicit scheme was surprisingly not the most stable

configuration. Tworzydlo shows that some combinations of implicit schemes are unconditionally unstable, while others are stable but may exhibit underdamped oscillations. Tworzydlo was able to use these findings to fine tune the solution techniques used in different areas of the flow field according to the needs of the flow. Kallinderis (1994) also implements a mixed implicit-explicit method but does not go into as much detail as Tworzydlo.

Most of the methods above use a first or second order finite difference approximation, whether implicit or explicit, to time step their solutions. Second order time approximations can be found in Tworzydlo (1992), Soulaimani (1994), and Mittal (1998). Marcum (1992) gives the option of either Runge-Kutta scheme or two-step Lax-Wendroff in order to step solutions through time. Marcum also found that two different dissipation models were required to accommodate the different time stepping schemes. Tanahashi (1990), Kallinderis (1994), and Jakobsen (2000) illustrate a means of local time stepping that can be used to accelerate the convergence of steady solutions using unsteady methods.

Most of the references of Table 1-1Table 1- use a first order shape function with a triangular or quadrilateral shaped element. Kallinderis (1994) illustrates the use of triangular elements to subdivide the quadrilateral elements in order to gain a better resolution or to merge to different sized meshes. Kallinderis, like many of the other sources, uses a structured meshing algorithm. Marcum (1992), on the other hand, illustrates the applications and features of both structured and unstructured solvers. Unstructured solvers can also be found in Bristeau (1990), Soulaimani (1994), Masud (1997) , Bonhaus (1998), Arminjon (1999), Wood (2000), and Luo (2004). Masud and

Wood also use moving mesh techniques to represent moving fluid boundaries. Because Masud allows for free surface moving boundaries, source terms are included to represent the relative motion of the moving elements.

All of the sources above, except for four, utilize a linear shape function. Soulaimani (1994) uses a quadratic shape function to represent the velocity distributions on the element and linear function for all other properties. Masud (1997), Bonhaus (1998), and Whiting (1999) leave their derivations open to higher order shape functions in order to accommodate further research.

The governing equations and boundary conditions are often the defining or limiting features of a solution method. All of the references above implement the viscous stress terms of the momentum equation along with the viscous dissipation and heat transfer terms of the energy equation. Ecer (1988) and Luo (2004) use Sutherland's equation to allow for variable viscosity in the flow; Bonhaus (1998) uses a 0.76 power law in order to solve for viscosity. Ecer, Luo, Barsoum(1995), and Mittal (1998) include some form of Stokes' hypothesis to relate the second coefficient of viscosity to the first coefficient. All of the sources above also implement a strong no slip condition of viscous solid wall surfaces (i.e., $u = v = 0$). Although, these boundary conditions do vary from isothermal to adiabatic walls. Bonhaus (1998) and Tworzydlo (1992) also develop their far field boundary conditions. In both descriptions, the far field is assumed to contain no viscous flux terms; both sources also assume no shear terms on inviscid wall boundary conditions. Tworzydlo further describes the conditions for a symmetry plane as lacking normal flow, shear stress, and heat transfer across the boundary condition. Tworzydlo

also notes that finite difference methods have received considerable attention in the area of boundary conditions; but FEM, especially implicit methods, are not yet complete.

Finally, all of the references above make use of primitive variables except for Soulaimani (1994). Soulaimani uses conservative variables to represent flow properties because the application of primitive variables is "questionable in the presence of shocks." Conservative variables increase the stability of the finite element approximation in compressible flow fields. All of the incompressible references use velocity and pressure as their primitive variables; the compressible sources used velocity, density, and enthalpy (or total energy) to represent the primitive variables and use the ideal gas equation of state to implicitly solve for pressure. Two of the sources, Jiang (1990) and Tanahashi (1990) adjust the viscous terms from velocity gradients to vorticity. Vorticity is then solved for as a "primitive variable".

As a conclusion to the literature review, an outline of the capabilities of solver to be expanded is given below. Cowan's (2003) two-dimensional solver *Euler2d* is a Galerkin finite element solver used to solve the compressible Euler equations in conservative variable form. The solver uses an unstructured grid of piece-wise linear triangle elements. A local-time stepped steady solver along with both linear and second order unsteady times steps are available to step the solution. The equations are derived in an implicit format but solved through an explicit iterative algorithm that utilizes the Courant-Fredrich-Lewy (CFL) number in order to relax the iterations. Two dissipation models are also used to stabilize the solution: A low order artificial dissipation model controlled by an input dissipation constant; and, a high order model, similar to the first,

that contains gradient limiters that reduce shock smearing and artificial flow solutions. Three boundary conditions are available: Far field, solid wall, and symmetry plane.

1.2. Research Objective

The objective of this research is to expand Cowan's finite element Euler solver into a full viscous solver. This thesis will develop the full Navier-Stokes equation for two-dimensional finite elements and their corresponding boundary conditions. Boundary layers are the most basic building block of all real aerodynamics problems; therefore the proper spacing of elements within the boundary layer are investigated to find an appropriate rule of thumb.

The original Euler solver was written to handle aeroelastic and non-inertial problems. The non-inertial portion of the solver was generated by adding non-inertial source terms to the original Euler equations. These source terms will be kept in their original form. On the other hand, the aeroelastic portion of the solver relied on a transpiration boundary condition in order to simulate motion in the boundary conditions without moving the structural surfaces. The transpiration condition is applied by moving the norms for the solid surfaces and reapplying the tangential flow to the new normals. In the viscous flow solver, all velocities on the solid surfaces must match the surface velocities themselves; therefore, changing the normals will not effect the flow field. The scope of this thesis is *not* to adapt the transpiration condition for viscous flows but rather to allow for future work to occur in this area.

Keeping all of this in mind, the new viscous solver needs to be tested for added run time, user complexity, and validity. The solver needs to be verified by comparisons

with several solutions to the simplified Navier-Stokes equations. Then the solver must be validated using data from several subsonic, supersonic, and transonic test cases that have been generated to validate computational fluids solvers.

1.3. Overview

This paper will continue by discussing the finite element method in Chapter 2. Continuing from the background previously presented, Chapter 2 will discuss the added complexities and requirements of viscous solvers versus Euler solvers and the additional problems that can be addressed using a viscous solver. The chapter will then focus on the development of the Navier-Stokes equations and the adjustments that are required to convert from a finite element Euler solver to a full viscous solver. The chapter will also discuss other topics including variable viscosity, dimensionless forms, and limitations of the second coefficient of viscosity.

Chapter 3 will introduce the finite element method used by Cowan (2003) and the need for a full Navier-Stokes solver. The chapter will discuss the representation of time derivatives, piece-wise linear spatial discretization, and stabilization routines of the original Euler solver. The chapter will also use the dimensionless form of the Navier-Stokes equations to develop the equations for each element and the required changes to the boundary conditions.

Chapter 4 will discuss the changes in computer implementation from the Euler solver to the new viscous solver. The chapter will cover the amendments to the data structures, any algorithm changes, and additional supporting programs. The chapter will also discuss the addition of shear stress to the force and moment summations, and the

chapter will end by covering the increased memory requirements, run times, and complexities to using the viscous solver over the original code.

Chapter 5 will begin with several examples to verify the adaptations to the code. The chapter will conclude with a validation of the code. Chapter 6 recaps the previous four chapters and gives recommendations for future work.

CHAPTER 2

2. Problem Definition

In this chapter the governing equations of the problem will be defined. To begin, the difference between Euler and Navier-Stokes problems will be defined. This discussion will also include a brief literature review of viscous aeroservoelastic applications and additional difficulties of such a solver. In the next sections, the Euler equations that defined the original problem of Cowan (2003) will be expanded to include the additional terms of the full Navier-Stokes equation and the non-inertial form of those equations. The three sections following will show that all of the properties, aside from viscosity and thermal conductivity, can be treated as constant properties. The equations and properties will be used to develop the dimensionless forms of the governing equations. Finally, Stokes' hypothesis will be developed to give the lower limit to the second coefficient of viscosity.

2.1. Choice of Governing Equations

The Euler equations can be used to solve a wide variety of aerodynamic and aeroelastic cases. The Euler equations are also simpler and easier to solve; and, for flows that are governed primarily by advection or pressure distributions, the Euler equations are a good assessment of the flow field. But, if the shear stresses, vortical nature, or boundary layers are desired, the full Navier-Stokes equations must be used. The full

viscous equations might also be helpful in a few Euler applications: A Navier-Stokes solver could be used to sustain a stable solution behind bluff bodies, as additional dissipation near leading and trailing edges and near shock, or to help capture vorticity in pure lift calculations. Several applications where a Navier-Stokes solver is more applicable are given below. Following those applications is a short discussion on the difficulties that can arise in a viscous solver.

2.1.1. Specific Applications of a Viscous Solver

Examples of solutions requiring viscous approximations include high angle of attack aerodynamics, multi-component wing interactions, stores and interference calculations, drag assessment (although shown to not yet be accurate enough by the literature), boundary layer controls, boundary layer-shock interactions in transonic and supersonic cases, transonic aeroelasticity, and engine spill over. This list is not definitive but contains many of the exemplary cases.

Drag prediction is the most popular application of viscous solvers. Many references can be found in the literature concerning inaccuracies of current viscous solutions. These solutions arise because of numerical problems or bad representation of the flow fields and boundary conditions. Paparone (2003) gives methods for eliminating "spurious drag" without excessive meshing. Although the prediction of viscous drag may not be fully accurate using reasonable meshes, the change in drag due to changes in configurations is often found to be very accurate. Brodersen (2002) used a Navier-Stokes solver to predict the changes in drag due to different engine installations and found the differences between the installations to be accurate. When drag predictions are necessary

for specified geometries, often Euler models along with empirical data have been used in the past. An example of such drag prediction is given by Oktay (2000), which predicted drag on missile geometries using an Euler solver, skin friction models, and empirical bluff body data. Oktay found that the method worked well for many missile configurations. On the other hand, configurations exist which have no empirical data or contain flow oscillations or other means that cannot be modeled properly using the Euler solvers. Filippone (2001) assessed the drag on a helicopter fuselage while uninstalled from the rotors. Filippone found that a viscous solver gave good drag predictions for the uninstalled drag. The installed drag varied dramatically due to rotor effects on the fuselage. Given the non-inertial capabilities of Cowan's Euler solver, viscous terms, and flow field partitioning between the fuselage and rotor, the installed drag on the helicopter could be predicted without empirical means.

Shock-boundary layer interactions are a big interest in transonic aerodynamics and aeroelasticity. Gai (2000) shows that the effects of a shock are actually felt upstream of Euler predicted locations because of effects in the boundary layer. The subsonic flow in the boundary layer allows the effects to propagate upstream effectively changing the location of the shock foot, bending the shock. Lee (1990) found distinct frequencies in a shock-boundary layer solution. These frequencies corresponded to the propagation time from the shock to the trailing edge of the airfoil and from the trailing edge (through the boundary layer) to the shock. These oscillations were found to be self-sustaining.

Other examples of shock-boundary layer interaction are illustrated in Jameson (1998), Gefroh (2002), and Lee (2004). Jameson addresses the proper grid refinement to capture shock-boundary layer interactions. Gefroh and Lee address the use of mesoflaps

in reducing drag due to shock-boundary layer interactions. Mesoflaps are small flexible surfaces covering an otherwise closed chamber. The mesoflaps downstream of the shock deflect into the chamber causing flow to be diverted out of the boundary layer; upstream mesoflaps deflect into the boundary layer returning the flow. The flow in the chamber then recirculates from downstream to upstream. Lee found that mesoflaps could be used to not only reduce wave drag caused by the shock but also the residual viscous drag caused by boundary layer thickening. Gefroh obtained similar results and found that the boundary layer was thicker upstream of the shock but thinner downstream of the shock due to the mesoflaps. Gefroh also suggested variation of mesoflap stiffness upstream of the shock. Using a compressible, viscous, aeroelastic solver, the mesoflaps could be modeled so that stiffness and area of coverage could be cheaply and efficiently tested before further experiments were to take place.

Vortices are another aspect of viscous fluid flow. An Euler solver can sustain vortical motion and convect vorticity downstream; but, without the aid of artificial dissipation, an Euler solver cannot generate vortices or dissipate existing vorticity. Without viscosity, vortices must be modeled in some way similar to that of Panaras (1990). Panaras used finite vortices in a flow field. The vortices were released in a given distribution upstream of either an airfoil or a corner of a step. The vorticity was allowed to interact with the surface as a potential flow field would mathematically. Panaras found that the "finite-area vorticity model" showed similar results to that in the laboratory, but Panaras was unable to model the dissipation effects and generation of new vortices from a separated boundary layer. Using a full viscous solver, the vorticity could be generated from a shear layer or other source, similar in geometry to the laboratory device. The

vorticity, convection flow, and boundary layer interactions could then be naturally modeled to see the effects of dissipation and boundary layer separation on the results. Similarly, vortical behavior is found to occur in lifting bodies. Kandil (1990) illustrates the application of a compressible Navier-Stokes solver on a rounded-edge delta wing configuration. The delta wing was held at a large mean angle of attack and oscillated about the roll axis of the wing. Kandil found that the primary and secondary vortices along with the associated shocks could be modeled quite well. Kandil was also able to find the oscillating pressure distribution on the wing and lift and moment coefficient histories. Kandil's computations are highly three dimensional, but the same vortical behavior can be found behind sharp edged objects held normal to the flow and bluff bodies at high enough Reynolds numbers.

Turbomachinery often contains shock-boundary layer interactions, wake effects, and problems that are isolated to the low speed boundary layer propagation. Gottfried (2002) used an Euler solver to explore the effects of high subsonic and transonic flows over an IGV (inlet guide vane) and rotor setup. Gottfried found the solver to simulate shocks slightly ahead of experiment and found that the solution lacked the upper harmonics seen in the experimental unsteady lift. Gottfried suggested the addition of viscous effects to capture the unsteady shock-boundary layer interactions on these two phenomena. In another situation, Chernobrovkin (1999) successfully approximated the shock-boundary layer interactions in a turbomachinery situation. Wadia (1998) used a viscous flow solver to experiment on the sweep angle of rotor blades in a compressor. Wadia found that three-dimensional viscous effects existed everywhere in the flow and when these effects were used properly shock-boundary layer could be minimized. Wadia

also found that forward swept rotors were more efficient because of reduced shock-boundary layer losses, lower suction at the leading edge, and less accumulation of a boundary layer near the blade tips. Shock-wake influences, which can also be important, can be modeled in a viscous solver. Darbe (2004) investigated IGV rotor interactions and found that shocks from the rotor disturb the IGV flow, causing unsteady effects in the IGV wake. The IGV wake in turn propagates downstream and disturbs the rotor.

The final application that will be addressed is high angle of attack flow. Erickson (1995) gives an excellent review of high angle of attack flow. Some of the examples given by Erickson are a general aviation plane landing in a flare, separated flow generated by flow over a wing-mounted engine nacelle, a fighter jet making a large-scale combat maneuver, and the blanketing effect seen by the tail of the space shuttle on reentry. Kwon (1992) simulated the flow over a F-15 wing-body-inlet configuration at high angle of attack, resulting in a model of the vortical flow leaving the strake and that seen by the twin tails. Similar tests were ran on a F-18 configuration by Rizk (1992) and Ghaffari (1993). Rogers (2001) used a viscous solver to model the landing configuration of a Boeing 777-200. The results of Rogers' analysis show the pressure distributions over the wing, slats, flaps, flaperons, and Krugers all at fully extended conditions. The lift and drag verse angle of attack was also found to match wind tunnel model data for most of the range. The computational results did stall at a lower angle of attack than expected, but Rogers was able to make adjustments to improve the computational model in order to stall nearer the experimental limit.

2.1.2. Difficulties with Viscous Solvers

Difficulties arise in modeling flows with a viscous finite element solver in the areas of spurious oscillations, local time stepping, numerical dissipation (in many forms), and appropriately capturing the viscous portions of the flow field. Bristeau (1990) shows that spurious oscillations can occur because matched orders of shape functions used to approximate the flow velocities versus other flow properties. Bristeau suggests using a higher order shape function to represent the velocities and lower order to represent density in compressible viscous solvers; no suggestions were made for the other flow properties. Tanahashi (1990) makes similar suggestions for velocity and pressure in a incompressible viscous solver. Soulaimani (1994) found that "unsteady flows appear at relatively low Reynolds numbers". Soulaimani says that these unsteady flows are due to numerical errors generated by the low Reynolds numbers. (Suggestions are made below for local time stepping, local segment Reynolds numbers, and artificial dissipation in order to address these problems.)

A different problem was identified by Shapira (1990). Shapira modeled flows through a two-dimensional converging channel. At Reynolds numbers less than the specified critical Reynolds numbers, the flows were found to be symmetric about the symmetry plane of the channel. At Reynolds numbers greater than the critical value, an asymmetric condition occurred where the flow separated on one side of the channel earlier than the other side. Shapira found that the asymmetric conditions were dependent upon the initial conditions of the solution. Although these asymmetric conditions are not undermining the validity of the solution, the conditions can be set off by small asymmetries in the mesh or held back by symmetric meshes. Therefore vortex shedding

on a cylinder or other body might be effected by the nature of the mesh itself. The onset of transition, which is outside of the scope of this thesis, could even be accelerated by disturbances in the grid and surface representation.

Many suggestions in the literature can be used to handle the above problems. The most frequent suggestion is a homogeneous or near homogeneous grid. For an unstructured grid, homogeneity is hard to ensure but steps can be taken in order to smooth the grid as much as possible. Marcum (1996) comments that unstructured meshes for viscous flow fields has received little attention. Additionally Marcum gives suggestions for viscous meshes and adaptive viscous meshes. Burton (1993) and Jameson (1998) also address the topic of well constructed viscous meshes. Jameson further assesses the error due to improper meshing. Thomasset (1981) suggests that numerical oscillations can be avoided by considering the local Reynolds number on the longest side of each element to be limited to values less than two. The computational cost to evaluate a mesh generated according to Thomasset's rule is too high to endure, therefore, artificial dissipation is most often chosen to alleviate such problems. Another suggestion is to decrease the local time step for steady flows and global time step for unsteady flows. The suggestions of Tanahashi (1990) and Kallinderis (1994) are represented in Equations 2.1 and 2.2, respectively. Jakobsen (2000) gives a similar idea for quadrilateral elements.

(2.1) $$\Delta t \le \min\left(\frac{\Delta x_e}{\left|\vec{V}_e\right|}, \frac{1}{8}\mathrm{Re}_L\,\Delta x_e^2\right)$$

(2.2) $$\Delta t = 0.9\frac{\Delta x_e}{\left|\vec{V}_e\right| + \frac{2\nu}{\Delta x_e}}$$

The suggestion made by Tanahashi (Eq. 1.1) requires the minimum of two values. The first is the advection limit for reasonable approximations, where the fluid is not allowed to cross the entire element in one time step. The second is a function of Reynolds number of the flow and the element size. Tanahashi equates the first to a CFL number and calls the second the "diffusion number". Kallinderis' suggestion contains the local Reynolds number hidden in the denominator.

Finally, the numerical viscosity of a solution is discussed by several authors. Numerical viscosity, as defined by Bristeau (1990), is the "upwinding, artificial viscosity, [or] viscosity introduced via time discretization". Bristeau further explains that numerical viscosity is commonly used to settle the oscillations caused by mesh sizing, time stepping, or similar shape functions for velocity and the other flow properties, instead of handling the problems directly. All aspects of numerical viscosity can be seen as numerical errors in the solver; numerical viscosity is so named to give these errors a positive spin because such errors are helpful and desired. Concerning artificial dissipation alone, Kallinderis (1995) gives a method for measuring the effects of artificial dissipation compared to the final solution. The method involves finding the ratio of changes due to artificial dissipation and the actual advective-viscous contributions. Kallinderis further found that the use of too much artificial dissipation can change the boundary layer solution and reduce the local skin friction coefficient significantly. Using Kallinderis' analysis, caution must be taken in the amount of artificial dissipation used in the viscous solutions.

2.1.3. Continuum Mechanics

The Navier-Stokes equations have been developed from the idea that a fluid can be treated as a continuum, or a body with continuous properties. In other words, the equations were not developed by thinking of the interaction between individual particles, but by treating the fluid as a finite volume and considering the conservation of mass, energy, and momentum across that volume.

A fluid can be treated as a continuum as long as the mean free path of the fluid particles is much smaller than the characteristic length of the body. The relationship between these two length scales is given by the Knudsen number: $Kn = \ell / L$ where ℓ is the mean free path and L is the characteristic length of the body. For continuum flow to exist, the Knudsen number must be much less than unity, or $Kn \leq 0.01$, as a rule of thumb. John (1984) presents the Knudsen number as a relationship to the Mach and Reynolds numbers of the flow:

$$Kn \approx \frac{Ma}{\sqrt{\mathrm{Re}_\delta}} \tag{2.3}$$

where Ma is the Mach number and Re_δ is the Reynolds number for the boundary layer thickness. (Note: This equation is only appropriate for Reynolds numbers larger than 1.)

2.1.4. Definition of a Newtonian Fluid

The stress-strain behavior of fluids can be fit to several models: Plastics, pseudoplastics, ideal Bingham plastics, and Newtonian fluids. Simple fluids, such as water, oils, and gases, are most often modeled as Newtonian fluids. The Newtonian fluid model says that (1) the stress is linearly proportional to the rate of strain of the fluid,

(2) the fluid is isotropic, and (3) in the absence of strain, the stress tensor reduces to the hydrostatic condition (White, 1991). The linear stress-strain relationship is illustrated in the following shear stress equation:

$$\tau_{xy} = 2\mu\varepsilon_{xy} \tag{2.4}$$

where τ_{xy} is the shear stress in the *xy*-plane, ε_{xy} is the rate of shear strain in the *xy*-plane, and μ is the viscosity of the fluid.

The isotropic assumption says that the fluid properties are independent of the coordinate system. The stress tensor developed in a *xy*-coordinate system is represented by the following indical equation (White, 1991):

$$\tau_{ij} = -p\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \delta_{ij}\lambda(\nabla \cdot V) \tag{2.5}$$

where τ_{ij} is the stress tensor, δ_{ij} is the Kronecker delta, and λ is the second coefficient of viscosity. This relationship also meets the third criteria in that the equation reduces to the hydrostatic condition in the absence of strain.

The stress tensor is often amended so that the pressure term is separated from the strain terms in the following form:

$$\tau'_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \delta_{ij}\lambda(\nabla \cdot V) \tag{2.6}$$

where τ'_{ij} is the amended stress tensor. This amended relationship is used in the formation of the governing equations below so that the pressure term can be included in the inviscid portion of the equations.

2.1.5. Ideal Equation of State

The ideal gas equation is often used to equate the changes in pressure of a fluid to the changes in density and temperature. The ideal gas equation is developed from kinetic-molecular theory (Brown et al., 1997): (1) The gas consists of a large number of molecules in continuous random motion, (2) the volume of the molecules is negligible when compared to the entire volume, (3) molecular forces are negligible, (4) molecular collisions are perfectly elastic, and (5) the average kinetic energy of the molecules is proportional to the absolute temperature and independent of time. The ideal gas equation is given below:

$$p = \rho RT \tag{2.7}$$

These assumptions are correct for gases under most conditions. But as the pressure increases or the temperature decreases, kinetic-molecular theory begins to break down. As these conditions occur, the density of the gas increases, causing the volume of the molecules and interplay of molecular forces to become more important. In 1873, Johannes van der Waals suggested corrections to the ideal gas equation to account for these conditions (Moran and Shapiro, 1996):

$$\left(p + \alpha\rho^2\right)\left(1/\rho - \beta\right) = RT \tag{2.8}$$

where $\alpha\rho^2$ is the correction for molecular forces and β is the correction for the volume of the molecules.

Using order of magnitude analysis, the orders of the terms in parentheses can be compared to determine whether the van der Waals corrections are necessary. Typical values of α are on the order of 0.001 bars $(m^3/kg)^2$; β is typically on the order of 0.001

m^3/kg. Under aerodynamic conditions, pressures can range from 0.1 to 10 bars while densities vary from 0.001 to 0.1 kg/m^3. The order of magnitude analysis, presented below, shows that the van der Waals corrections are not necessary:

$$\left.\begin{matrix}0.1\\10\end{matrix}\right\} \equiv O(p) >> O(\alpha\rho^2) \equiv \begin{cases}1x10^{-9}\\1x10^{-5}\end{cases} \tag{2.9}$$

and

$$\left.\begin{matrix}1000\\10\end{matrix}\right\} \equiv O\left(1/\rho\right) >> O(\beta) \equiv 0.001 \tag{2.10}$$

The previous relationships of the order of magnitude analysis can also be used to check the flow field solution for similar validity of the van der Waals equations. To eliminate the need for two extra variables, the ideal gas equation should be rewritten to depend on flow energy, *e*, and the ratio of specific heats, γ, instead of the gas constant, *R*, and temperature, *T*. The converted equation takes the following form:

$$p = \rho(\gamma - 1)\left(e - \frac{1}{2}u^2 - \frac{1}{2}v^2\right) \tag{2.11}$$

2.2. The Compressible Navier-Stokes Equations

The original governing equations of Cowan (2003) are represented by Equation 2.12. The vectors U_i, F_i, and G_i are given in Equations 2.13 through 2.15. The first term of the vectors represent the continuity equation, the second and third terms correspond to the x- and y-momentum equations, and the fourth to the energy equation. Cowan represents the following equation in conservative form. The equation has been written here in expanded form so that the viscous terms can be more readily added. Here the subscript *i* represents the inviscid vector:

(2.12) $$\frac{\partial U_i}{\partial t} + \frac{\partial F_i}{\partial x} + \frac{\partial G_i}{\partial y} = 0$$

where

(2.13) $$U_i = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{Bmatrix}$$

(2.14) $$F_i = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho ue + pu \end{Bmatrix}$$

(2.15) $$G_i = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho ve + pv \end{Bmatrix}$$

The equations can be modified by adding a viscous vector to each of the spatial derivative terms. The original vectors are represented by the subscript *i*, and the new viscous vectors are represented by the subscript *v*:

(2.16) $$\frac{\partial U_i}{\partial t} + \frac{\partial (F_i - F_v)}{\partial x} + \frac{\partial (G_i - G_v)}{\partial y} = 0$$

where

(2.17) $$F_v = \begin{Bmatrix} 0 \\ \tau'_{xx} \\ \tau'_{xy} \\ u\tau'_{xx} + v\tau'_{xy} + q_x \end{Bmatrix}$$

(2.18) $$G_v = \begin{Bmatrix} 0 \\ \tau'_{xy} \\ \tau'_{yy} \\ u\tau'_{xy} + v\tau'_{yy} + q_y \end{Bmatrix}$$

The stress and conduction terms are then given by the following equations:

(2.19) $$\tau'_{xx} = 2\mu \frac{\partial u}{\partial x} + \lambda(\nabla \cdot V)$$

(2.20) $$\tau'_{yy} = 2\mu \frac{\partial v}{\partial y} + \lambda(\nabla \cdot V)$$

(2.21) $$\tau'_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)$$

(2.22) $$q_x = k\frac{\partial T}{\partial x} = \frac{k}{c_v}\frac{\partial e}{\partial x}$$

(2.23) $$q_y = k\frac{\partial T}{\partial y} = \frac{k}{c_v}\frac{\partial e}{\partial y}$$

Cowan also presents two additional equations for the ideal gas equation of state and the enthalpy relationship, shown below:

(2.24) $$p = \rho(\gamma - 1)\left(e - \frac{1}{2}u^2 - \frac{1}{2}v^2\right)$$

(2.25) $$h = e + \frac{p}{\rho} = \frac{\gamma p}{\rho(\gamma - 1)} + \frac{1}{2}u^2 + \frac{1}{2}v^2$$

2.3. Non-Inertial Form of the Equations

In order to handle the additional effects of spinning and super-maneuvering vehicles, Cowan introduced a fourth vector to the Euler equations. This fourth vector is called the non-inertial source vector, shown in Equation 2.27:

$$\frac{\partial U_i}{\partial t} + \frac{\partial F_i}{\partial x} + \frac{\partial G_i}{\partial y} = S_r \tag{2.26}$$

where

$$S_r = -\rho \begin{Bmatrix} 0 \\ a_{t,x} - \dot{\alpha} v_r \\ a_{t,y} + \dot{\alpha} u_r \\ a_{t,x}(u_t + u_r) + a_{t,y}(v_t + v_r) \end{Bmatrix} \tag{2.27}$$

where a is the acceleration and $\dot{\alpha}$ is the rotational velocity of the body. The subscripts r and t above represent the relative and transformation conditions, respectively. Cowan also places the relative subscript on all velocities in the other three vectors.

Because the non-inertial source term is not the topic of the current research, the reader is referred to the dissertation written by Cowan that covers the topic in full. Instead, the non-inertial source vector is discussed only to show that the same adjustments can be made to the viscous terms to arrive at a non-inertial viscous solution.

The new viscous terms would first need to be adjusted similarly to Cowan's inviscid terms to allow for the relative velocities. In doing so, all velocity terms in the equations are given a similar relative subscript:

$$F_v = \begin{Bmatrix} 0 \\ \tau'_{xx} \\ \tau'_{xy} \\ u_r \tau'_{xx} + v_r \tau'_{xy} + q_x \end{Bmatrix} \tag{2.28}$$

$$G_v = \begin{Bmatrix} 0 \\ \tau'_{xy} \\ \tau'_{yy} \\ u_r \tau'_{xy} + v_r \tau'_{yy} + q_y \end{Bmatrix} \tag{2.29}$$

where

$$\tau'_{xx} = 2\mu \frac{\partial u_r}{\partial x} + \lambda(\nabla \cdot V_r) \tag{2.30}$$

$$\tau'_{yy} = 2\mu \frac{\partial v_r}{\partial y} + \lambda(\nabla \cdot V_r) \tag{2.31}$$

$$\tau'_{xy} = \mu\left(\frac{\partial u_r}{\partial y} + \frac{\partial v_r}{\partial x}\right) \tag{2.32}$$

$$q_x = \frac{k}{c_v}\frac{\partial e_r}{\partial x} \tag{2.33}$$

$$q_y = \frac{k}{c_v}\frac{\partial e_r}{\partial y} \tag{2.34}$$

where e_r is the relative energy of the flow.

These equations will not effect the non-inertial source term; and, the viscous terms themselves will remain unaffected because the relative gradients are equivalent to their inertial counterparts. In other words, if the x- and y-directions of the non-inertial and inertial frames correspond to each other, the gradient in one frame is equal to that of the other frame, which can be safely said about the u- and v-velocities and energy.

2.4. Constant Properties

A calorically perfect gas has thermodynamic properties that are linearly proportional to the absolute temperature. For calorically perfect properties, the property can be represented as a linear function of the absolution temperature that goes to zero at an absolute temperature of zero: $f(T) = k\,T$. To test the validity of such an assumption, atmospheric pressures and temperatures for Earth (Bertin, 1998) and Mars (Benson, 2004) were represented on both sides of normal shocks ranging from Mach 1.5 to 3.5.

For simplification, Martian atmosphere was considered to be composed of pure carbon dioxide. To obtain functions to represent c_p and γ, empirical data was taken from three sources, and 6th order polynomials were fit to all of the points (Moran, 1996; Incropera, 2002; John, 1984). (Plots of the pressure and temperature profiles from 0 to 80 kilometers of altitude are shown for Earth and Mars in Appendix B; plots are also shown for c_p and γ.)

The pressure and temperature downstream of the shocks were generated from the normal shock equations of John (1984). The changes in c_p and γ were generated for the corresponding conditions across the shock. The percent change in properties is plotted in Figures B-9 through B-12 in Appendix B. Each plot shows the variation of c_p and γ across a normal shock at various altitudes and Mach numbers between 1.5 and 3.5. The largest percent difference in c_p was found to be 12.9% for air and 9.7% for carbon dioxide. The largest percent difference in γ was found to be 4.4% for air and 12.0% for carbon dioxide. All four of these conditions existed at Mach 3.5, and all four percentages are small enough to consider these properties to be constant across the flow field. Small errors will be generated due to this assumption, but the computational costs will also be greatly reduced.

2.5. Variable Viscosity and Thermal Conductivity

A similar procedure was used to test the validity of a constant viscosity flow field. Again Earth and Martian atmospheres were tested with supersonic shocks ranging from Mach 1.5 to 3.5. (Plots of the percent change across the shock are given in Figures B-13 and B-14 in Appendix B.) Sutherland's law was used to model the coefficient of

viscosity versus temperature (White, 1991). The model predicted a maximum change of 139% difference for air and 148% for carbon dioxide. These changes are drastically different than those for c_p and γ.

A 150% increase in viscosity across a shock means that the Reynolds number would decrease by a factor of 2 ½. This could be damaging when trying to predict the transition location or thickness of any boundary layer on the downwind side of the shock. Therefore, Sutherland's law will be used to model the viscosity of the fluid at all points in the flow:

$$\frac{\mu}{\mu^{(0)}} = \left(\frac{T}{T^{(0)}}\right)^{1.5} \frac{T^{(0)} + S}{T + S} \tag{2.35}$$

where $\mu^{(0)}$ is the reference viscosity, $T^{(0)}$ is the reference (absolute) temperature, and S is Sutherland's constant

Since the flow field will be considered to have variable viscosity, the necessity for variable thermal conductivity must also be assessed. Prandtl number is the dimension-less ratio of the effects of viscosity to thermal conductivity:

$$\mathrm{Pr} = \frac{c_p \mu}{k} \tag{2.36}$$

where k is the coefficient of thermal conductivity. Since c_p is a assumed to be constant for the flow, the Prandtl number can be used to relate the thermal conductivity to viscosity (now a variable property). Looking at the temperatures in the Earth atmosphere and those after the Mach 3.5 shock, the temperature ranges from 200 to 950 K. Across this range the Prandtl number averages 0.703 and only deviates by 5%. This variation is a reasonable percentage to assume a constant Prandtl number. Similarly, temperature for the Martian atmosphere range from 200 to 800 K. Across this range the Prandtl number

averages 0.733 and again only deviates by 5%. Again this variation is reasonable to assume a constant Prandtl number. Therefore, the changes in the viscosity can be represented using Sutherland's law while changes in thermal conductivity will be seen by scaling viscosity by the Prandtl number of the flow field.

2.6. Dimensionless Forms

The original solver of Cowan utilized the governing equations in dimensionless form. Therefore, the additional viscous terms need to be non-dimensionalized in a similar manner. Equations 2.37 through 2.44 were used by Cowan; the remaining five equations were added by this research.

(2.37) $$\rho^* = \frac{\rho}{\rho_0}$$

(2.38) $$u^* = \frac{u}{U_0};\ v^* = \frac{v}{U_0}$$

(2.39) $$p^* = \frac{p}{\rho_0 U_0^2}$$

(2.40) $$e^* = \frac{e}{U_0^2};\ h^* = \frac{h}{U_0^2}$$

(2.41) $$t^* = \frac{tU_0}{L}$$

(2.42) $$x^* = \frac{x}{L};\ y^* = \frac{y}{L}$$

(2.43) $$\gamma = \frac{c_p}{c_v}$$

(2.44) $$Ma_0 = \frac{U_0}{a_0}$$

(2.45) $$\mathrm{Re} = \frac{\rho_0 U_0 L}{\mu_0}$$

(2.46) $$\mathrm{Pr} = \frac{\mu c_p}{k}$$

(2.47) $$\mu^* = \frac{\mu}{\mu_0}$$

(2.48) $$\lambda^* = \frac{\lambda}{\mu}$$

(2.49) $$S_{\mathrm{mod}} = \frac{c_v S}{U_0^2}$$

Using the previously defined dimensionless ratios, the governing equations can be specified in the following form:

(2.50) $$\frac{\partial U_i^*}{\partial t^*} + \frac{\partial\left(F_i^* - F_v^*\right)}{\partial x^*} + \frac{\partial\left(G_i^* - G_v^*\right)}{\partial y^*} = 0$$

where

(2.51) $$U_i^* = \begin{Bmatrix} \rho^* \\ \rho^* u^* \\ \rho^* v^* \\ \rho^* e^* \end{Bmatrix}$$

(2.52) $$F_i^* = \begin{Bmatrix} \rho^* u^* \\ \rho^* \left(u^*\right)^2 + p^* \\ \rho^* u^* v^* \\ \rho^* u^* e^* + p^* u^* \end{Bmatrix}$$

(2.53) $$G_i^* = \begin{Bmatrix} \rho^* v^* \\ \rho^* u^* v^* \\ \rho^* \left(v^*\right)^2 + p^* \\ \rho^* v^* e^* + p^* v^* \end{Bmatrix}$$

(2.54) $$F_v^* = \begin{Bmatrix} 0 \\ \tau_{xx}^* \\ \tau_{xy}^* \\ u^*\tau_{xx}^* + v^*\tau_{xy}^* + q_x^* \end{Bmatrix}$$

(2.55) $$G_v^* = \begin{Bmatrix} 0 \\ \tau_{xy}^* \\ \tau_{yy}^* \\ u^*\tau_{xy}^* + v^*\tau_{yy}^* + q_y^* \end{Bmatrix}$$

and the stress and conduction terms are given by:

(2.56) $$\tau_{xx}^* = \frac{2\mu^*}{\mathrm{Re}}\frac{\partial u^*}{\partial x^*} + \frac{\lambda^*\mu^*}{\mathrm{Re}}\left(\frac{\partial u^*}{\partial x^*} + \frac{\partial v^*}{\partial y^*}\right)$$

(2.57) $$\tau_{yy}^* = \frac{2\mu^*}{\mathrm{Re}}\frac{\partial v^*}{\partial y^*} + \frac{\lambda^*\mu^*}{\mathrm{Re}}\left(\frac{\partial u^*}{\partial x^*} + \frac{\partial v^*}{\partial y^*}\right)$$

(2.58) $$\tau_{xy}^* = \frac{\mu^*}{\mathrm{Re}}\left(\frac{\partial u^*}{\partial y^*} + \frac{\partial v^*}{\partial x^*}\right)$$

(2.59) $$q_x^* = \frac{\mu^*}{\gamma\,\mathrm{Pr}\,\mathrm{Re}}\frac{\partial e^*}{\partial x^*}$$

(2.60) $$q_y^* = \frac{\mu^*}{\gamma\,\mathrm{Pr}\,\mathrm{Re}}\frac{\partial e^*}{\partial y^*}$$

Further, the ideal gas equation of state and enthalpy equation can be non-dimensionalized into the following forms (repeated from Cowan, 2003).

(2.61) $$p^* = \rho^*(\gamma - 1)\left(e^* - \frac{1}{2}(u^*)^2 - \frac{1}{2}(v^*)^2\right)$$

(2.62) $$h^* = e^* + \frac{p^*}{\rho^*} = \frac{\gamma p^*}{\rho^*(\gamma - 1)} + \frac{1}{2}(u^*)^2 + \frac{1}{2}(v^*)^2$$

Finally, Sutherland's equation can be non-dimensionalized by dividing the equation at any state by that at the free stream condition. The rational form eliminates the need for the reference temperature and viscosity. The temperatures can then be converted to internal energy and non-dimensionalized into the following equation:

$$\mu^* = \left(\frac{e^*}{e_0^*}\right)^{1.5} \frac{e_0^* + S_{\text{mod}}}{e^* + S_{\text{mod}}} \tag{2.63}$$

From this point onward, the asterisks are dropped to simplify the equations; nevertheless, the equations are still represented in dimensionless form.

2.7. Stokes Hypothesis for Second Viscosity

The dissipation term in the energy equation represents the energy dissipated by viscous effects. Since viscous effects cannot create energy, the viscous dissipation term must be non-negative (either positive or zero). This term is given Equation 2.64 where the subscripts *x* and *y* represent the partial derivatives with respect to the *x*- and *y*-directions:

$$\Phi = \mu\left[2u_x^2 + 2v_y^2 + \left(v_x + u_y\right)^2\right] + \lambda\left(u_x + v_y\right)^2 \tag{2.64}$$

Stokes hypothesis says that the second coefficient of viscosity can be represented as a direction function of dynamic viscosity: $\lambda = -\frac{2}{3}\mu$. Recent findings in the literature include values of the second coefficient greater than zero and some even greater than their corresponding dynamic viscosity. These experiments are a matter of controversy themselves but seem to suggest than the second viscosity should be $\lambda \geq -\frac{2}{3}\mu$. This condition can be proven by showing that a non-negative dissipation term is ensured for the following conditions (White, 1991; Baum, 1997):

(2.65) $$\mu \geq 0 \ ; \ 3\lambda + 2\mu \geq 0$$

Anderson (1984) gives a similar discussion by relating the first and second coefficients of viscosity to the combined viscosity, κ:

(2.66) $$\kappa \equiv \tfrac{2}{3}\mu + \lambda$$

Anderson further explains that κ is only helpful when studying the structure of shocks and/or the absorption and attenuation of acoustic waves. Anderson further assumes that κ is negligible for his discussion and arrives at Stokes' hypothesis. Rearranging the previous equation, the lower limit of the second coefficient is again illustrated:

(2.67) $$\lambda = \kappa - \tfrac{2}{3}\mu \geq -\tfrac{2}{3}\mu$$

CHAPTER 3

3. Methodology

The previous chapter started with the Euler equations used by Cowan (2003) and developed the full Navier-Stokes equations. Using Cowan's Euler solver and his original formulation as a model, the Navier-Stokes equations will now be developed into a finite element methodology. First, the theory behind finite element methods will be illustrated by discussing the terms involved and benefits of using integration by parts on the original governing equations. The space-time discretization established by Cowan is reviewed. The equations for one- and two-dimensional elements will then be generated to be used to represent the boundary edges and internal flow field, respectively. Next, the boundary conditions used by Cowan will be reviewed along with the needed changes to establish the boundary conditions as viscous boundaries. And finally, the chapter will conclude with a discussion of the artificial dissipation model used by Cowan.

3.1. Overview of Finite Element Methods

The partial differential equations represented by Equations 2.10 through 2.13 represent the compressible Euler equations. No closed form solution to these equations exists in their full representation, although solutions have been found for more simplified forms. In order to solve the full compressible Euler equations, a numerical method is required. Cowan (2003) chose to solve the equations using a triangular finite element

method with three temporal models. Cowan selected a finite element method to represent the geometry without requiring body fit coordinate systems. Instead, the finite element method uses an unstructured mesh of triangles. The three temporal models include one steady-state model and two unsteady time stepping models. The space-time formulations and dissipation models used by Cowan are left for discussion in later sections.

In the previous chapter, the Euler equations were expanded to the Navier-Stokes equations given by Equations 2.14 through 2.21. These equations can be represented in the same form as Cowan's Euler equations. In order to begin, three terms must be definied: (1) the spatial domain, (2) the spatial boundary, and (3) the element shape-weighting function. The spatial domain, Ω, encompasses the entire flow field to be represented by the spatial elements. The spatial boundary, Γ, represents the edges of the flow field and the edges of the spatial domain. The spatial boundary is always one dimension less than that of the corresponding spatial boundary. For example, since the spatial domain used in this thesis will always be the two-dimensional domain, the spatial boundary will always be one-dimensional. Once the spatial domain has been established and bound by the spatial boundary, the domain must be broken into many finite elements. Each element uses a shape function, Φ, to represent the distribution of flow field variables across the element. The triangular elements used here contain three points of information at the three vertices. The most simple shape function is given by a linear relationship between these three values. In the Galerkin method, the shape function is also used as the weighting function (explained below).

In order to find a solution, the finite element method integrates the governing equations at a specific set of flow variables. The equations are said to be solved when the

integral of the governing equation is minimized or zero. If the integral is minimized to a value other than zero, the difference between the final value of the integral and zero is the residual numerical error. Baker (1983) describes this process as solving equations by minimizing the overall error on the flow field. A weighting function is used to scale the governing equations to aid in the minimization process. Several methods have been developed over the years; the Galerkin method is the particular method selected by Cowan and will therefore be illustrated here. The Galerkin method uses a weighting function that is equivalent to the element shape function. The value of the weighting function is equal to the shape function on the element and equivalently zero at all other points in the domain. The vector form of the Galerkin integration is given below for the governing equations. For more details on Galerkin formation, Lapidus and Pinder (1982) give a step-by-step presentation of the Galerkin integral along with several other methods.

$$\int_{\Omega} \Phi_e^T \left(\frac{\partial U_i}{\partial t} + \frac{\partial (F_i - F_v)}{\partial x} + \frac{\partial (G_i - G_v)}{\partial y} - S_r \right) d\Omega = 0 \tag{3.1}$$

where U_i, F_i, F_v, G_i, G_v, and S_r represent the inviscid, viscous, and non-inertial source vectors previously given in Equations 2.10 through 2.16 and 2.25. Using the Gauss divergence theorem, or integral by parts, the equation can be broken into two integrals:

$$\int_{\Omega} \left(\Phi_e^T \frac{\partial U_i}{\partial t} - \frac{\partial \Phi_e^T}{\partial x}(F_i - F_v) - \frac{\partial \Phi_e^T}{\partial y}(G_i - G_v) - \Phi_e^T S_r \right) d\Omega + \int_{\Gamma} \left(\Phi_e^T \left(F_i n_x + G_i n_y - F_v n_x - G_v n_y \right) \right) d\Gamma = 0 \tag{3.2}$$

The first integral represents the integration on the elements interior to the flow field domain, Ω; the second integral represents the integration of the boundary elements of the domain boundary, Γ. The two integrals can be used to split the solution of the

flow domain into two separate parts: The first part uses the domain integral to find the necessary contribution to the solution by the triangular elements within the domain. The second part uses the boundary integral to find the necessary contributions given by the different boundary conditions around the boundary.

The solution of the terms with a subscript *I* has already been accomplished by Cowan in the original solver. The non-inertial source vector S_r is also included in Cowan's problem definition. The remaining two vectors (denoted by subscript *V*) are added by this thesis. In the sections that follow, the original formulation of Cowan will be discussed to bring the reader up to speed, then an explanation of the additions made by this thesis will be given.

3.2. Space-Time Formulation

Cowan (2003) uses an implicit predictor multi-corrector algorithm to march the solution of Equation 3.2 to zero. This algorithm is implicit because all of the flow properties in the equation are defined at the current time except for those used to approximate the time derivative. The algorithm begins with the flow properties at the last time step and corrects for Equation 3.2. The new flow properties at the current time condition are then used to iterate the solution to convergence. The current solver does not specify a convergence condition but rather a number of cycles to convergence.

The original space-time formulation used by Cowan can be separated into three different types: (1) steady, (2) first order unsteady, and (3) second order unsteady, all with a linear piece-wise spatial formulation. The spatial formulation is linear on each element and discontinuous from element to element, therefore piece-wise linear. The

temporal formulation originates when the two-dimensional spatial domain is given a third dimension in time.

3.2.1. Steady Solutions

For steady solutions Cowan uses a constant-in-time approximation for the time step, similar to the piece-wise linear spatial domain. Cowan names this condition the "jump condition" because each domain is considered to be constant in time for each space-time domain, but discontinuous between domains. The "jump condition" is illustrated in Figure 3-1. The vector **U** represents the unknowns vector at various times *t*. The superscripts represent the discontinuity in unknowns across the jump condition.

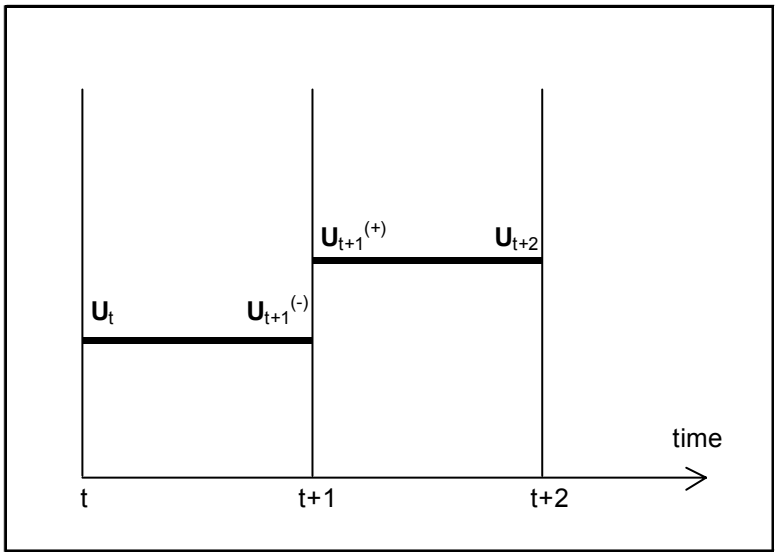

Figure 3-1: Constant-in-Time Representation (or Jump Condition)

The steady solution can be progressed through time just as that of an unsteady solution. The time step of the solution march is limited by stability. The stability is, in part, ensured by using a Courant type (CFL) stability criteria, used to scale the time step.

The time step is also limited to the amount of time for a fluid particle to cross the smallest element. The second limitation can be lifted by allowing each element to have its own local time instead of enforcing a global time on the entire flow field. In this way, a CFL of unity will allow information to progress across each element in the domain at every solution step. The CFL factor can then be used to scale this local time step back.

The local time step is developed according to a energy stability analysis methodology. The method relates the time step at a node to the eigenvalues of the segments (e.g., element edges, boundary edges) connected to that node. For a given node *i* and CFL factor, the local time step is given by Equation 3.3:

$$\Delta t_i = 2 \; CFL \; \left[M_L\right]_i \left[\sum_{k=1}^{nsg} \lambda_{ij,k}\right]^{-1} \tag{3.3}$$

where $\left[M_L\right]_i$ is the global lumped mass matrix at node *i* and $\lambda_{ij,k}$ is the eigenvalue for the k^{th} segment attached to node *i*, spanning between nodes *i* and *j*. Further, the eigenvalue is found using the velocity at node *j*, $\vec{V}_{j,k}$, speed of sound for the segment, $a_{ij,k}$, and geometric weighting vector, $\vec{W}_{ij,k}$. The eigenvalue for segment *i,j* is illustrated in Equation 3.4.

$$\lambda_{ij,k} = \left|\vec{V}_{j,k} \cdot \vec{W}_{ij,k}\right| + a_{ij,k}\left|\vec{W}_{ij,k}\right| \tag{3.4}$$

3.2.2. 1st Order Unsteady Solutions

The first order unsteady solver still uses the jump conditions along with a CFL factor, but steps the solution at a time accurate global time step. In order to speed up convergence, the solver still calculates the local time step and steps the inner iterations

with the local time step. Each time step is still calculated using the global time step, but local time steps are used to stabilize the solution and converge the solution more quickly. Cowan further warns that more inner cycles per time step are required for global time steps that are larger than the local time step.

The local time step is also adjusted for stability reasons when the local time is greater than the global time step. Cowan describes the ratio of local to global time steps as a relaxation factor. If the relaxation factor (local over global) is less than unity, the element is under-relaxed. For a relaxation factor greater than unity, the element is over-relaxed and could diverge. Therefore, the unsteady solver is limited to local time steps less than or equal to the global time step. The only remaining concern with the unsteady solver is ensuring that the number of cycles per time step is greater than the global time step divided by the local time step for the worst case scenario, generally the smallest element of the field.

3.2.3. 2nd Order Unsteady Solutions

The second order unsteady solution is generated by replacing the first order jump condition with a second order jump condition approximation. The new approximation is a second order finite difference equation given below.

$$\frac{\frac{3}{2}U_t^i - 2U_{t-1} + \frac{1}{2}U_{t-2}}{\Delta t} \tag{3.5}$$

where U_t^i is the flow properties at the i^{th} iteration toward a solution of U_t; and, U_{t-1} and U_{t-2} are the flow properties at the past two time steps, respectively.

Cowan also warns that because the local time stepping is used to numerically stabilize the second order solution, this solution is often found to be unstable and require more artificial dissipation to converge. But the two unsteady solvers use about the same amount of CPU time to generate their solutions.

3.3. Finite Element Discretization

Now that the temporal discretization has been discussed, the spatial equations need to be shown in their original Euler form and their current viscous form. First the unknowns vector is developed here. The shape function is also then generated in one and two dimensions to find the unknowns at any point across the field element or boundary edge. The shape vector and corresponding finite element equations can then be developed in both one and two dimensions.

The unknowns vector is a vector containing the five unknown properties that will be solved for at each point in the flow. These five properties include density, velocity in the x- and y-directions, pressure, and enthalpy. The unknowns vector is illustrated in Equation 3.6. From this point on the bold faced **U** will represent the unknowns vector, whereas the lowercase *u* will represent the velocity in the x-direction.

$$\mathbf{U} = \begin{Bmatrix} \rho \\ u \\ v \\ p \\ \rho h \end{Bmatrix} \tag{3.6}$$

3.3.1. One-Dimensional Elements

This section will develop the necessary shape function and derivatives along a one-dimensional element. The 1-D element is will be used as a simplified version of the two-dimensional element and also in the development of boundary edge integrals.

The shape function for a one-dimensional element can be defined most simply by two lines. The two lines span between nodes *i* and *j*. One of the lines must go to zero at node *i* and unity at node *j*; the remaining line will go to zero and unity at the opposite nodes. An illustration of the shape function is given by the Figure 3-2 below. The first line will be represented by the first term of the shape vector, Φ_1; while the second line is represented by the remaining term, Φ_2.

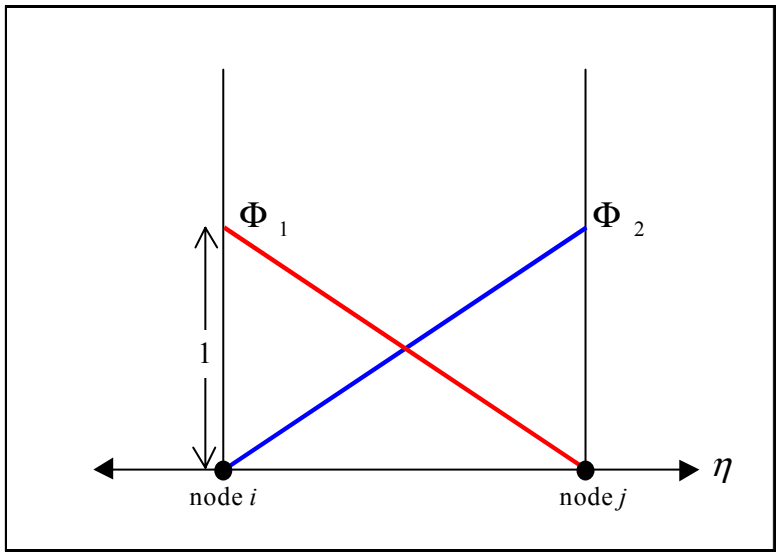

Figure 3-2: Shape Vector for a Linear One-Dimensional Element

If the scalar product of the shape vector and the vector representing the flow properties is found, the scalar flow property at any location η along the element can be

found by Equation 3.7. This scalar product is represented graphically in Figure 3-3 as the full linear shape function of the one-dimensional element.

(3.7) $$u = \Phi_e u_e = \{\Phi_1 \quad \Phi_2\}\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}$$

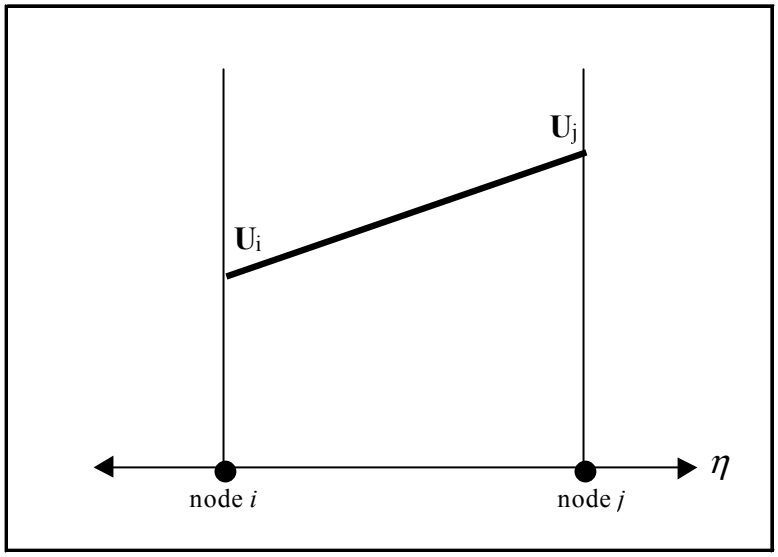

Figure 3-3: Shape Function for a Linear One-Dimensional Element

If the natural coordinate η of the element spans from η = -1 at node *i* to η = 1 at node *j*, then the shape function vector for the element is given by Equation 3.8.

(3.8) $$\Phi_e = \frac{1}{2}\{1-\eta \quad 1+\eta\}$$

Since any property can be generated by a scalar product with the shape function vector, the x-location along the element can be found by Equation 3.9.

(3.9) $$x = x_e \Phi_e = \frac{1}{2}(x_1 + x_2) + \frac{1}{2}\eta x_{21}$$

where

(3.10) $$x_{21} = x_2 - x_1 = \Delta x_e$$

Here, x_1 represents the x-location of node *i* and x_2 represents the x-location of node *j*. If the line were sloped through an xy-plane, the y-location at any η could be found through a similar approach.

The gradient of the shape function is constant because the element has been represented by a linear shape function. The relationship between the gradient in the natural coordinates and the gradient in the global coordinate can be related by the chain rule, shown in Equation 3.11.

$$\frac{\partial}{\partial \eta} = \frac{\partial x}{\partial \eta}\frac{\partial}{\partial x} \tag{3.11}$$

where the Jacobean of the element can be found by either differentiating Equation 3.9 or by inspection by looking at the change in both the local and global coordinates across the element. Both of these methods are illustrated below.

$$\frac{\partial x}{\partial \eta} = \frac{\partial}{\partial \eta}\left(\frac{1}{2}(x_1 + x_2) + \frac{1}{2}\eta x_{21}\right) = \frac{1}{2}x_{21} \tag{3.12}$$

$$\frac{\partial x}{\partial \eta} = \frac{\Delta x}{\Delta \eta} = \frac{\Delta x_e}{1-(-1)} = \frac{1}{2}\Delta x_e \tag{3.13}$$

The relationship of Equation 3.11 can be reversed to find the global derivatives in terms of the local derivatives in the natural coordinate frame, illustrated in Equation 3.14. The inverse of the Jacobean is used to reverse the equation.

$$\frac{\partial}{\partial x} = \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta} = \frac{2}{\Delta x_e}\frac{\partial}{\partial \eta} \tag{3.14}$$

Using Equation 3.14, the gradient of the shape function can be found.

$$\frac{\partial \Phi_e^T}{\partial x} = \frac{2}{\Delta x_e}\frac{\partial \Phi_e^T}{\partial \eta} = \frac{1}{\Delta x_e}\begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \tag{3.15}$$

The gradient on the element can then be used to represent the global gradient vector. The gradient of u-velocity and internal energy are illustrated in Equations 3.16 and 3.17 below. The gradient is constant across the element, therefore the gradient of a property will be expressed as a constant scalar for the element.

(3.16) $$\frac{\partial u}{\partial x} = \frac{\partial(\Phi_e u_e)}{\partial x} = \frac{1}{\Delta x_e}\{-1 \quad 1\}\begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix} = \frac{1}{\Delta x_e}(u_2 - u_1)$$

(3.17) $$\frac{\partial e}{\partial x} = \frac{\partial(\Phi_e e_e)}{\partial x} = \frac{1}{\Delta x_e}\{-1 \quad 1\}\begin{Bmatrix} e_1 \\ e_2 \end{Bmatrix} = \frac{1}{\Delta x_e}(e_2 - e_1)$$

Now that the shape function has been defined, and the element gradients can be found in the global frame, several integrals must be examined in order to understand how the governing equations are converted to finite element form. The governing equation in one dimension (a simplified version of Equation 3.2) is given by the following:

(3.18) $$\int_\Omega \left(\Phi_e^T \frac{\partial U_i}{\partial t} - \frac{\partial \Phi_e^T}{\partial x}(F_i - F_v) \right) d\Omega + \int_\Gamma \left(\Phi_e^T (F_i - F_v) n_x \right) d\Gamma = 0$$

The first term of this integration can be broken down into the vector representation given in Equation 3.19. Since the shape function is considered to be constant in time by the jump condition, the shape function can be pulled out of the time derivative.

(3.19) $$\int_\Omega \left(\Phi_e^T \frac{\partial U_i}{\partial t} \right) d\Omega = \frac{\Delta x_e}{2} \int_{-1}^{1} \Phi_e^T \frac{\partial (U_{i,e} \Phi_e)}{\partial t} d\eta = \frac{\Delta x_e}{2} \int_{-1}^{1} \Phi_e^T \Phi_e d\eta \frac{\partial U_{i,e}}{\partial t}$$

The last term of Equation 3.19 gives a matrix, called the element mass matrix, and the time derivative of the vector U_i for the element. The integration of the element mass matrix is given in Equation 3.20. This matrix can be used for any of the three time

derivatives mentioned in Section 1.2; the time derivative needs only be represented as a column vector.

$$[M_e] = \frac{\Delta x_e}{2}\int_{-1}^{1} \Phi_e^T \Phi_e \, d\eta = \frac{\Delta x_e}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \tag{3.20}$$

If U_i can be represented by Equation 3.21 for one-dimensional problems, then the time derivative of the vector can be approximated for first order accuracy by Equation 3.22. Using the time derivative of the momentum flux as an example, Equations 3.19 through 3.22 can be combined to derive the finite element time derivative term for momentum flux, shown in Equation 3.23.

$$U_i = \begin{Bmatrix} \rho \\ \rho u \\ \rho e \end{Bmatrix} \tag{3.21}$$

$$\frac{\partial U_i}{\partial t} \equiv \frac{1}{\Delta t}\begin{Bmatrix} \rho_{t+1} - \rho_t \\ (\rho u)_{t+1} - (\rho u)_t \\ (\rho e)_{t+1} - (\rho e)_t \end{Bmatrix} \tag{3.22}$$

$$\int_{\Omega}\left(\Phi_e^T \frac{\partial(\rho u)}{\partial t}\right) d\Omega = \frac{\Delta x_e}{6\Delta t}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}\begin{Bmatrix} (\rho_1 u_1)_{t+1} - (\rho_1 u_1)_t \\ (\rho_2 u_2)_{t+1} - (\rho_2 u_2)_t \end{Bmatrix} \tag{3.23}$$

The second term in Equation 3.18 can be separated into two parts: An inviscid integration and a viscous integration term. The inviscid integration term will be handled next. This term is given in Equation 3.24. The gradient of the shape function is constant for the element and can therefore be pulled out of the integration.

$$\int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial x} F_i\right) d\Omega = \frac{\Delta x_e}{2}\int_{1}^{-1}\left(\frac{2}{\Delta x_e}\frac{\partial \Phi_e^T}{\partial \eta} F_i\right) d\eta = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}\int_{1}^{-1} F_i \, d\eta \tag{3.24}$$

Cowan uses Gauss quadrature to integrate the final integral form of the previous equation. Numerical integration was not necessary for the inviscid integrals, but Cowan

saw fit to do so in order to accommodate the addition of viscous terms. This equation is given below along with the Gauss weights, w_j, and Gauss points, η_j, for one- and two-point Gauss quadrature.

$$\int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial x}F_i\right)d\Omega = \left\{\begin{matrix}-1\\1\end{matrix}\right\}\sum_{j=1}^{NP} w_j F_i(\eta_j) \tag{3.25}$$

Table 3-1: Gauss Quadrature for One-Dimensional Elements (Chand., 1991)

Number of Points (*NP*)	Gauss Points (η_j)	Gauss Weights (w_j)
1	0.0	2.0
2	-0.577350269 0.577350269	1.0 1.0

Similar to the representation of U_i, F_i can be represented by Equation 3.26. Using the energy flux as an example, Equations 3.24 through 3.26 can be combined to derive the finite element energy flux term with one point Gauss quadrature, shown in Equation 3.27.

$$F_i = \left\{\begin{matrix}\rho u\\ \rho u^2 + p\\ (\rho e + p)u\end{matrix}\right\} \tag{3.26}$$

$$\int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial x}(\rho e + p)u\right)d\Omega = \left\{\begin{matrix}-1\\1\end{matrix}\right\}\left(2\left(\frac{\rho_1 e_1 + \rho_2 e_2}{2} + \frac{p_1 + p_2}{2}\right)\frac{u_1 + u_2}{2}\right) \tag{3.27}$$

The remaining viscous term to integrate is given in Equation 3.28 below. Notice that, similar to Equation 3.21, the gradient can be moved outside of the integral. Therefore, Gauss quadrature can be used to integrate the remaining equation resulting in Equation 3.29.

(3.28) $$\int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial x}F_v\right)d\Omega=\frac{\Delta x_e}{2}\int_{1}^{-1}\left(\frac{2}{\Delta x_e}\frac{\partial \Phi_e^T}{\partial \eta}F_v\right)d\eta=\begin{Bmatrix}-1\\1\end{Bmatrix}\int_{1}^{-1}F_v d\eta$$

(3.29) $$\int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial x}F_v\right)d\Omega=\begin{Bmatrix}-1\\1\end{Bmatrix}\sum_{j=1}^{NP}w_j F_v(\eta_j)$$

The only complication occurs when trying to apply the integration to a mixture of gradients and variables. This complication is easily resolved with the understanding that each gradient in the equation is a constant for the element, and each variable in the equation can be generated as a scalar value by using Equation 3.7. Therefore, F_v can be represented by Equations 3.30 through 3.32. Using the normal stress as an example, Equation 3.16 and Equations 3.29 through 3.32 can be combined to derive Equation 3.33 shown below.

(3.30) $$F_v=\begin{Bmatrix}0\\ \tau'_{xx}\\ u\tau'_{xx}+q_x\end{Bmatrix}$$

where

(3.31) $$\tau'_{xx}=\frac{(2+\lambda)}{\mathrm{Re}}\mu\frac{\partial u}{\partial x}$$

(3.32) $$q_x=\frac{1}{\gamma\,\mathrm{Pr}\,\mathrm{Re}}\mu\frac{\partial e}{\partial x}$$

(3.33) $$\int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial x}\tau'_{xx}\right)d\Omega=\begin{Bmatrix}-1\\1\end{Bmatrix}\left(2\left(\frac{(2+\lambda)}{\mathrm{Re}}\frac{\mu_1+\mu_2}{2}\frac{u_2-u_1}{\Delta x_e}\right)\right)$$

3.3.2. Two-Dimensional Elements

Now that the one-dimensional element has been explored, the two-dimensional element is just an expansion on the shape function and the full two-dimensional

governing equations. In the same way that the one-dimensional shape function was defined by a line between two points (Figure 3-3), the shape function for a two-dimensional element can be described as three points used to establish a plane.

The basic shape and natural coordinates of a two-dimensional element are illustrated in Figure 3-4 below. The natural coordinate system of the two-dimensional triangle element is defined by three nodes. At each of the nodes, two of the natural coordinates go to zero while one of the coordinates goes to unity. Along the edges of the element, one of the coordinates remains zero while the other two span from zero to unity.

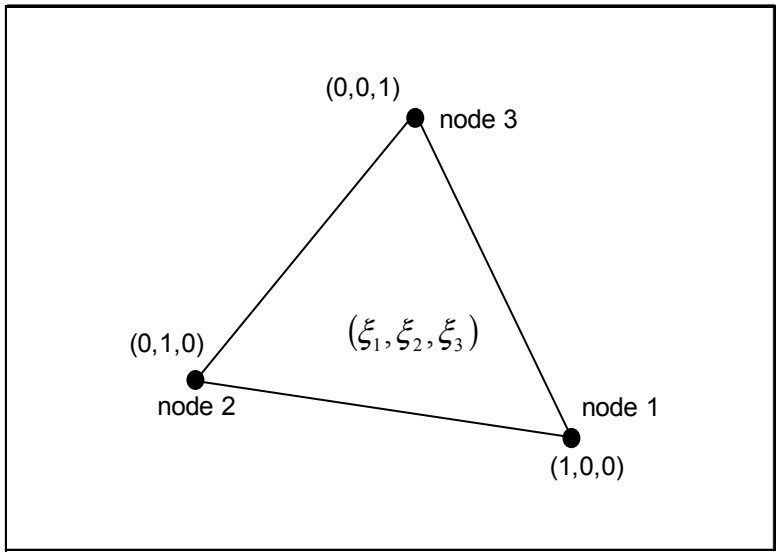

Figure 3-4: Geometry and Natural Coordinates for a Two-Dimensional Element

In order to limit the possible combinations on the natural coordinates, Equation 3.34 is used to calculate a third coordinate from the two known coordinates.

(3.34) $$\xi_1 + \xi_2 + \xi_3 = 1$$

(3.35) $$\begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix} \begin{Bmatrix} \xi_1 \\ \xi_2 \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix}$$

The three-space coordinate system of Figure 3-4 can therefore be transformed into a two-space coordinate system using Equation 3.34. Equation 3.35 is the result of converting from a three coordinate system to a two natural coordinate system. From this point forward, the two natural coordinates will be used to express the location throughout the triangle element. In this way, there are two global coordinates and now two local coordinates on each element.

If the scalar product of the shape vector and the flow properties vector is again made, as in Equation 3.36, a scalar property can be found at any position on the element. This scalar product is represented by a plane in Figure 3-5.

(3.36) $$u = \Phi_e u_e = \{\xi_1 \quad \xi_2 \quad \xi_3\} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

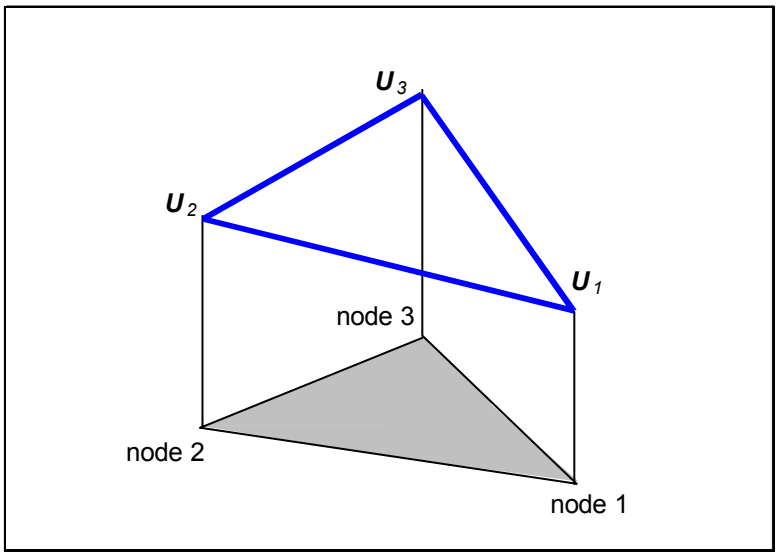

Figure 3-5: Shape Function for a Linear Two-Dimensional Element

Since any property can be generated by a scalar product with the shape function vector, the xy-location on the element can be found by such a product. In Equation 3.37 the three natural coordinates are used to find the two global coordinates.

(3.37)
$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix}$$

Equation 3.35 can be substituted into Equation 3.37 and simplified into Equation 3.38 below.

(3.38)
$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} x_{13} & x_{23} \\ y_{13} & y_{23} \end{bmatrix} \begin{Bmatrix} \xi_1 \\ \xi_2 \end{Bmatrix} + \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}$$

where

(3.39)
$$x_{ij} = x_i - x_j$$
$$y_{ij} = y_i - y_j$$

Again the gradient of the shape function is constant because the element has been represented by a linear shape function. The relationship between the gradient in the local and global coordinates can be found through the chain rule. In Equation 3.40, the chain rule has been illustrated in an *i*-index notation where *i* is either 1 or 2 for the natural coordinates. The index notation has then been expanded into Equation 3.41 into matrix notation, where the Jacobean matrix is shown.

(3.40) $$\frac{\partial}{\partial \xi_i} = \frac{\partial x}{\partial \xi_i}\frac{\partial}{\partial x} + \frac{\partial y}{\partial \xi_i}\frac{\partial}{\partial y}$$

(3.41) $$\begin{Bmatrix} \frac{\partial}{\partial \xi_1} \\ \frac{\partial}{\partial \xi_2} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi_1} & \frac{\partial y}{\partial \xi_1} \\ \frac{\partial x}{\partial \xi_2} & \frac{\partial y}{\partial \xi_2} \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{Bmatrix} = \begin{bmatrix} x_{13} & y_{13} \\ x_{23} & y_{23} \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{Bmatrix}$$

The Jacobean can be inverted and moved to the other side of the equation, resulting in Equation 3.42, such that the global gradient is now defined by the local gradient. Notice that the determinant of the Jacobean is set equal to twice the area of the element, A_e, in Equation 3.43. The element area is used this way for notation purposes only.

(3.42) $$\begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial}{\partial \xi_1} \\ \frac{\partial}{\partial \xi_2} \end{Bmatrix} = \frac{1}{2A_e} \begin{bmatrix} y_{23} & -y_{13} \\ -x_{23} & x_{13} \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial \xi_1} \\ \frac{\partial}{\partial \xi_2} \end{Bmatrix} = \frac{1}{2A_e} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial \xi_1} \\ \frac{\partial}{\partial \xi_2} \end{Bmatrix}$$

where

(3.43) $$2A_e = |\det[J]| = |x_{13}y_{23} - x_{23}y_{13}|$$

From Equation 3.42, the x-derivative of the shape function can be defined as:

$$\frac{\partial \Phi_e^T}{\partial x} = \frac{1}{2A_e}\left(A_{11}\frac{\partial \Phi_e^T}{\partial \xi_1} + A_{12}\frac{\partial \Phi_e^T}{\partial \xi_2} \right) \tag{3.44}$$

$$\frac{\partial \Phi_e^T}{\partial x} = \frac{1}{2A_e}\left(A_{11}\frac{\partial}{\partial \xi_1}\begin{Bmatrix} \xi_1 \\ \xi_2 \\ 1-\xi_1-\xi_2 \end{Bmatrix} + A_{12}\frac{\partial}{\partial \xi_2}\begin{Bmatrix} \xi_1 \\ \xi_2 \\ 1-\xi_1-\xi_2 \end{Bmatrix} \right) \tag{3.45}$$

$$\frac{\partial \Phi_e^T}{\partial x} = \frac{1}{2A_e}\left(A_{11}\begin{Bmatrix} 1 \\ 0 \\ -1 \end{Bmatrix} + A_{12}\begin{Bmatrix} 0 \\ 1 \\ -1 \end{Bmatrix} \right) = \frac{1}{2A_e}\begin{Bmatrix} A_{11} \\ A_{12} \\ -A_{11}-A_{12} \end{Bmatrix} \tag{3.46}$$

Similarly, the y-derivative can be defined as:

$$\frac{\partial \Phi_e^T}{\partial y} = \frac{1}{2A_e}\left(A_{21}\begin{Bmatrix} 1 \\ 0 \\ -1 \end{Bmatrix} + A_{22}\begin{Bmatrix} 0 \\ 1 \\ -1 \end{Bmatrix} \right) = \frac{1}{2A_e}\begin{Bmatrix} A_{21} \\ A_{22} \\ -A_{21}-A_{22} \end{Bmatrix} \tag{3.47}$$

Using these two relationships, the gradient on the element can then be used to represent the global gradient vector. The x-gradient of u-velocity and y-gradient of internal energy are illustrated in Equations 3.48 and 3.49, respectively. Remember that the gradient is constant across the element, therefore the gradient of a property will be expressed as a constant vector for the element.

$$\frac{\partial u}{\partial x} = \frac{\partial(\Phi_e u_e)}{\partial x} = \frac{A_{11}u_1 + A_{12}u_2 - (A_{11}+A_{12})u_3}{2A_e} \tag{3.48}$$

$$\frac{\partial e}{\partial y} = \frac{\partial(\Phi_e e_e)}{\partial y} = \frac{A_{21}e_1 + A_{22}e_2 - (A_{21}+A_{22})e_3}{2A_e} \tag{3.49}$$

Now that the two-dimensional shape function has been defined and used to develop the gradient vectors, there are several integrals to be solved. Equation 3.2 can again be used as the developing equation; but, this time Equation 3.2 need not be

simplified in order to represent the integrals. The first term of the equation can be broken down into the vector representation of Equation 3.50. Since the shape function is considered to be constant in time by the jump condition, the shape function can be pulled out of the time derivative, as it was in Equation 3.19.

$$(3.50)\quad \int_{\Omega}\left(\Phi_e^T\frac{\partial U_i}{\partial t}\right)d\Omega = 2A_e\int_0^1\int_0^{1-\xi_2}\Phi_e^T\frac{\partial\left(U_{i,e}\Phi_e\right)}{\partial t}d\xi_1 d\xi_2 = 2A_e\int_0^1\int_0^{1-\xi_2}\Phi_e^T\Phi_e d\xi_1 d\xi_2\frac{\partial U_{i,e}}{\partial t}$$

The last term of this equation can again be broken into the mass matrix and the time derivative vector. The mass matrix is integrated in Equation 3.51. The mass matrix can be used for any of the three time derivatives mentioned in Section 3.2; the time derivative needs only be represented as a column vector.

$$(3.51)\qquad \left[M_e\right] = 2A_e\int_0^1\int_0^{1-\xi_2}\Phi_e^T\Phi_e d\xi_1 d\xi_2 = \frac{A_e}{12}\begin{bmatrix}2 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 2\end{bmatrix}$$

The vector U_I is given by Equation 2.1 for two-dimensional problems; and, the time derivative of the vector can be represented to first order accuracy by Equation 3.52. Using the time derivative of the energy flux as an example, Equations 3.50 through 3.52 can be combined to derive the finite element time derivative term, shown in Equation 3.53.

$$(3.52)\qquad \frac{\partial U_i}{\partial t} \equiv \frac{1}{\Delta t}\begin{Bmatrix}\rho_{t+1}-\rho_t\\ (\rho u)_{t+1}-(\rho u)_t\\ (\rho v)_{t+1}-(\rho v)_t\\ (\rho e)_{t+1}-(\rho e)_t\end{Bmatrix}$$

$$(3.53)\qquad \int_{\Omega}\left(\Phi_e^T\frac{\partial U_i}{\partial t}\right)d\Omega = \frac{A_e}{12\Delta t}\begin{bmatrix}2 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 2\end{bmatrix}\begin{Bmatrix}(\rho_1 e_1)_{t+1}-(\rho_1 e_1)_t\\ (\rho_2 e_2)_{t+1}-(\rho_2 e_2)_t\\ (\rho_3 e_3)_{t+1}-(\rho_3 e_3)_t\end{Bmatrix}$$

The second and third terms of Equation 3.2 can be broken into two separate parts: An inviscid and a viscous integration term. The inviscid term is the term used on each element of the Euler solver; the viscous term is the additional computations added by this thesis. The inviscid terms are given in Equations 3.54 and 3.55. Notice that the gradient of the shape function is constant for the element and can therefore be pulled out of the integration.

$$(3.54) \qquad \int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial x}F_i\right)d\Omega = 2A_e\int_0^1\int_0^{1-\xi_2}\left(\frac{\partial \Phi_e^T}{\partial x}F_i\right)d\xi_1 d\xi_2 = \begin{Bmatrix} A_{11} \\ A_{12} \\ -A_{11}-A_{12} \end{Bmatrix}\int_0^1\int_0^{1-\xi_2} F_i d\xi_1 d\xi_2$$

$$(3.55) \qquad \int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial y}G_i\right)d\Omega = 2A_e\int_0^1\int_0^{1-\xi_2}\left(\frac{\partial \Phi_e^T}{\partial y}G_i\right)d\xi_1 d\xi_2 = \begin{Bmatrix} A_{21} \\ A_{22} \\ -A_{21}-A_{22} \end{Bmatrix}\int_0^1\int_0^{1-\xi_2} G_i d\xi_1 d\xi_2$$

As with the one-dimensional element integration, Cowan used Gauss quadrature to numerically integrate the inviscid flux terms. Equations 3.54 and 3.55 are repeated below using Gauss quadrature; the Gauss weights and points for one- and three-point Gauss quadrature are given for the following equations.

$$(3.56) \qquad \int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial x}F_i\right)d\Omega = \begin{Bmatrix} A_{11} \\ A_{12} \\ -A_{11}-A_{12} \end{Bmatrix}\sum_{j=1}^{NP} w_j F_i\left(\xi_{1,j},\xi_{2,j}\right)$$

$$(3.57) \qquad \int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial y}G_i\right)d\Omega = \begin{Bmatrix} A_{21} \\ A_{22} \\ -A_{21}-A_{22} \end{Bmatrix}\sum_{j=1}^{NP} w_j G_i\left(\xi_{1,j},\xi_{2,j}\right)$$

Table 3-2: Gauss Quadrature for Two-Dimensional Elements (Chand., 1991)

Number of Points (NP)	Gauss Points ($\xi_{1,j}, \xi_{2,j}$)	Gauss Weights (w_j)
1	(1/3, 1/3)	1/2
3	(2/3, 1/6)	1/6
	(1/6, 2/3)	1/6
	(1/6, 1/6)	1/6

The vectors F_i and G_i can further be represented by Equations 3.58 and 3.59. Using the mass flux as an example, Equations 3.56 through 3.59 can be combined to derive the finite element mass flux terms with one point Gauss quadrature, shown in the equations that follow.

(3.58) $$F_i = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho ue + pu \end{Bmatrix}$$

(3.59) $$G_i = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho ve + pv \end{Bmatrix}$$

(3.60) $$\int_\Omega \left(\frac{\partial \Phi_e^T}{\partial x} F_i \right) d\Omega = \begin{Bmatrix} A_{11} \\ A_{12} \\ -A_{11} - A_{12} \end{Bmatrix} \frac{1}{2} \left(\frac{\rho_1 u_1 + \rho_2 u_2 + \rho_3 u_3}{3} \right)$$

(3.61) $$\int_\Omega \left(\frac{\partial \Phi_e^T}{\partial y} G_i \right) d\Omega = \begin{Bmatrix} A_{21} \\ A_{22} \\ -A_{21} - A_{22} \end{Bmatrix} \frac{1}{2} \left(\frac{\rho_1 v_1 + \rho_2 v_2 + \rho_3 v_3}{3} \right)$$

The remaining viscous terms to integrate (from the element integrals) is given in Equations 3.62 and 3.63 below. Notice, the gradient can again be moved outside of the

integral. Gauss quadrature can again be used to numerically solve the double integral, resulting in the equations that follow.

$$\int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial x}F_v\right)d\Omega = 2A_e\int_0^1\int_0^{1-\xi_2}\left(\frac{\partial \Phi_e^T}{\partial x}F_v\right)d\xi_1 d\xi_2 = \begin{Bmatrix} A_{11} \\ A_{12} \\ -A_{11}-A_{12} \end{Bmatrix}\int_0^1\int_0^{1-\xi_2} F_v d\xi_1 d\xi_2 \tag{3.62}$$

$$\int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial y}G_v\right)d\Omega = 2A_e\int_0^1\int_0^{1-\xi_2}\left(\frac{\partial \Phi_e^T}{\partial y}G_v\right)d\xi_1 d\xi_2 = \begin{Bmatrix} A_{21} \\ A_{22} \\ -A_{21}-A_{22} \end{Bmatrix}\int_0^1\int_0^{1-\xi_2} G_v d\xi_1 d\xi_2 \tag{3.63}$$

$$\int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial x}F_v\right)d\Omega = \begin{Bmatrix} A_{11} \\ A_{12} \\ -A_{11}-A_{12} \end{Bmatrix}\sum_{j=1}^{NP} w_j F_v\left(\xi_{1,j},\xi_{2,j}\right) \tag{3.64}$$

$$\int_{\Omega}\left(\frac{\partial \Phi_e^T}{\partial y}G_v\right)d\Omega = \begin{Bmatrix} A_{21} \\ A_{22} \\ -A_{21}-A_{22} \end{Bmatrix}\sum_{j=1}^{NP} w_j G_v\left(\xi_{1,j},\xi_{2,j}\right) \tag{3.65}$$

The vectors F_v and G_v can further be represented by Equations 3.66 through 3.72. Using the x-momentum flux as an example, Equations 3.64 through 3.72 can be combined to derive the finite element x-momentum flux terms with one point Gauss quadrature, shown in Equations 3.73 through 3.78.

$$F_v = \begin{Bmatrix} 0 \\ \tau'_{xx} \\ \tau'_{xy} \\ u\tau'_{xx} + v\tau'_{xy} + q_x \end{Bmatrix} \tag{3.66}$$

$$G_v = \begin{Bmatrix} 0 \\ \tau'_{xy} \\ \tau'_{yy} \\ u\tau'_{xy} + v\tau'_{yy} + q_y \end{Bmatrix} \tag{3.67}$$

where

$$\tau'_{xx} = \frac{(2+\lambda)}{\mathrm{Re}}\mu\frac{\partial u}{\partial x} + \frac{\lambda}{\mathrm{Re}}\mu\frac{\partial v}{\partial y} \tag{3.68}$$

$$\tau'_{yy} = \frac{\lambda}{\mathrm{Re}}\mu\frac{\partial u}{\partial x} + \frac{(2+\lambda)}{\mathrm{Re}}\mu\frac{\partial v}{\partial y} \tag{3.69}$$

$$\tau'_{xy} = \frac{\mu}{\mathrm{Re}}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{3.70}$$

$$q_x = \frac{\mu}{\gamma\,\mathrm{Pr}\,\mathrm{Re}}\frac{\partial e}{\partial x} \tag{3.71}$$

$$q_y = \frac{\mu}{\gamma\,\mathrm{Pr}\,\mathrm{Re}}\frac{\partial e}{\partial y} \tag{3.72}$$

$$\int_\Omega \left(\frac{\partial \Phi_e^T}{\partial x} F_v\right) d\Omega = \begin{Bmatrix} A_{11} \\ A_{12} \\ -A_{11} - A_{12} \end{Bmatrix} \left(\frac{1}{2}\frac{\mu_1 + \mu_2 + \mu_3}{3}\left(\frac{(2+\lambda)}{\mathrm{Re}}\frac{\partial u}{\partial x} + \frac{\lambda}{\mathrm{Re}}\frac{\partial v}{\partial y}\right)\right) \tag{3.73}$$

$$\int_\Omega \left(\frac{\partial \Phi_e^T}{\partial y} G_v\right) d\Omega = \begin{Bmatrix} A_{21} \\ A_{22} \\ -A_{21} - A_{22} \end{Bmatrix} \left(\frac{1}{2}\frac{1}{\mathrm{Re}}\frac{\mu_1 + \mu_2 + \mu_3}{3}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right) \tag{3.74}$$

where

$$\frac{\partial u}{\partial x} = \frac{A_{11}u_1 + A_{12}u_2 - (A_{11} + A_{12})u_3}{2A_e} \tag{3.75}$$

$$\frac{\partial v}{\partial y} = \frac{A_{21}v_1 + A_{22}v_2 - (A_{21} + A_{22})v_3}{2A_e} \tag{3.76}$$

$$\frac{\partial u}{\partial y} = \frac{A_{21}u_1 + A_{22}u_2 - (A_{21} + A_{22})u_3}{2A_e} \tag{3.77}$$

$$\frac{\partial v}{\partial x} = \frac{A_{11}v_1 + A_{12}v_2 - (A_{11} + A_{12})v_3}{2A_e} \tag{3.78}$$

The remaining terms of Equation 3.2 below to the boundary edge integrals or boundary conditions. These integrals are based on the one-dimensional element equations from the previous section. The only difference here comes from the evaluation of the derivatives in the viscous integrals. The remaining four terms of Equation 3.2 will be handled below in two groups: Inviscid and viscous terms. The inviscid terms are examples of the derivations used by Cowan; the viscous terms are examples of the terms used in this thesis.

The inviscid boundary integral terms are given by Equations 3.79 and 3.80.

$$\int_{\Gamma} \left(\Phi_e^T F_i n_x\right) d\Gamma = \frac{\Delta s}{2} n_x \int_{-1}^{1} \Phi_e^T \Phi_e F_{i,e} ds = \frac{\Delta s}{2} n_x \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{Bmatrix} F_{i,1} \\ F_{i,2} \end{Bmatrix} \tag{3.79}$$

$$\int_{\Gamma} \left(\Phi_e^T G_i n_y\right) d\Gamma = \frac{\Delta s}{2} n_y \int_{-1}^{1} \Phi_e^T \Phi_e G_{i,e} ds = \frac{\Delta s}{2} n_y \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{Bmatrix} G_{i,1} \\ G_{i,2} \end{Bmatrix} \tag{3.80}$$

The two previous equations can be used by evaluating the values of F_i and G_i at nodes 1 and 2 using Equations 3.58 and 3.59. Both equations assume that the inviscid fluxes vary linearly across the surface; this assumption is not bad and often helps with the stability of the boundary condition. The values of the inviscid flux terms along with the boundary normal can be used to calculate the vectors from Equation 3.79 and 3.80. The boundary normal, given by (n_x, n_y), faces into the flow field by convention.

The viscous boundary integral terms are given by Equations 3.81 and 3.82. This time instead of assuming that the viscous fluxes are linear along the boundary edge, the properties are assumed to vary linearly along the edge as previously represented on the element.

(3.81) $$\int_{\Gamma}\left(\Phi_e^T F_v n_x\right)d\Gamma = \frac{\Delta s}{2} n_x \int_{-1}^{1} \Phi_e^T F_v ds = \frac{\Delta s}{4} n_x \sum_{j=1}^{NP} w_j \begin{Bmatrix} 1-\eta_j \\ 1+\eta_j \end{Bmatrix} F_v\left(\eta_j\right)$$

(3.82) $$\int_{\Gamma}\left(\Phi_e^T G_v n_y\right)d\Gamma = \frac{\Delta s}{2} n_y \int_{-1}^{1} \Phi_e^T G_v ds = \frac{\Delta s}{4} n_y \sum_{j=1}^{NP} w_j \begin{Bmatrix} 1-\eta_j \\ 1+\eta_j \end{Bmatrix} G_v\left(\eta_j\right)$$

F_v and G_v can be found in a similar manner as with the entire element, but the numerical integration must also include the shape function on the boundary. Using the energy flux as an example, Equations 3.81 and 3.82 can be combined with Equations 3.66 through 3.72 to derive the finite element energy flux terms with one point Gauss quadrature, shown in Equations 3.83 and 3.84. Since the gradient on the element is constant across the element, the gradient is also constant for the boundary edge. Therefore, the gradient for the element that corresponds to the boundary edge is used in the integration of Equations 3.83 and 3.84. This element gradient is still found using Equations 3.69 through 3.72. (Note: The indices 1, 2, and 3 in Equations 3.83 and 3.84 may not align exactly with the indices 1 and 2 in the following equations. The indices below are with respect to the boundary edge nodes; those in Equations 3.83 and 3.84 belong to the element that corresponds to that boundary edge.)

(3.83) $$\int_{\Gamma}\left(\Phi_e^T F_v n_x\right)d\Gamma = \frac{\Delta s}{4} n_x \left(2\begin{Bmatrix} 1 \\ 1 \end{Bmatrix}\left(\frac{u_1+u_2}{2}\tau'_{xx} + \frac{v_1+v_2}{2}\tau'_{xy} + q_x \right)\right)$$

(3.84) $$\int_{\Gamma}\left(\Phi_e^T G_v n_y\right)d\Gamma = \frac{\Delta s}{4} n_y \left(2\begin{Bmatrix} 1 \\ 1 \end{Bmatrix}\left(\frac{u_1+u_2}{2}\tau'_{xy} + \frac{v_1+v_2}{2}\tau'_{yy} + q_y \right)\right)$$

where

(3.85) $$\tau'_{xx} = \frac{(2+\lambda)}{\mathrm{Re}}\frac{\mu_1+\mu_2}{2}\frac{\partial u}{\partial x} + \frac{\lambda}{\mathrm{Re}}\frac{\mu_1+\mu_2}{2}\frac{\partial v}{\partial y}$$

(3.86) $$\tau'_{yy} = \frac{\lambda}{\mathrm{Re}}\frac{\mu_1+\mu_2}{2}\frac{\partial u}{\partial x} + \frac{(2+\lambda)}{\mathrm{Re}}\frac{\mu_1+\mu_2}{2}\frac{\partial v}{\partial y}$$

(3.87) $$\tau'_{xy} = \frac{1}{\mathrm{Re}} \frac{\mu_1 + \mu_2}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)$$

(3.88) $$q_x = \frac{1}{\gamma \mathrm{Pr}\, \mathrm{Re}} \frac{\mu_1 + \mu_2}{2} \frac{\partial e}{\partial x}$$

(3.89) $$q_y = \frac{1}{\gamma \mathrm{Pr}\, \mathrm{Re}} \frac{\mu_1 + \mu_2}{2} \frac{\partial e}{\partial y}$$

3.4. Boundary Conditions and Boundary Integrals

The original Euler solver written by Cowan (2003) contained three boundary condtions: Solid wall, symmetry plane, and far field. The solid wall condition is a slip conditions that restricts the flow to that tangential to the surface of the wall. Displacement and velocity of the solid wall is simulated through transpiration, which changes the normal of the surface rather moving the mesh to establish motion. The symmetry condition also restricts flow to the tangential direction, but does so in a weakly through the boundary integral. The symmetric condition cannot handle curvature well whereas the solid wall condition handles curvature quite well. The far field condition uses Reimann invariants to adjust the boundary integral so that the freestream is weakly enforced at all far field edges.

In order to apply the viscous and conduction terms to the flow field, these terms must also be applied to the boundary integrals through the boundary conditions. All boundary conditions within the scope of this thesis are considered to be adiabatic, therefore all of the following sections will automatically neglect any heat transfer across the boundaries. This assumption is good since no heat transfer can occur across a symmetry plane by definition, the far field should not contain gradients by definition, and

the solid surfaces have no solid conduction model to receive the heat transfer therefore it is only safe to assume none in the first place. All other assumptions will be listed below with their corresponding boundary condition.

All three original boundary conditions will be discuss further along with any adaptations, and the new viscous solid wall will be discussed in entirety. Because a new viscous wall is being generated by this thesis, the original solid wall condition of Cowan will be referred to as an inviscid wall or Euler solid wall condition from this point forward.

3.4.1. Symmetry Boundary Condition

In order to implement the symmetry boundary condition, the definition of symmetry must be met across the boundary. Therefore, any gradients perpendicular to the boundary must be zero, and the fluid can only flow tangent to the boundary. Cowan enforces the symmetry boundary condition by weakly applying the symmetry constrains to the boundary integrals themselves. The inviscid boundary flux vectors are given in Equation 3.90 below.

$$F_i n_x + G_i n_y = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho e + p)u \end{Bmatrix} n_x + \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho e + p)v \end{Bmatrix} n_y \tag{3.90}$$

To apply the symmetry condition weakly, all velocities normal to the boundary must be neglected, shown in Equation 3.91. If this condition is used to simplify the previous equation, the boundary flux vectors are given by Equation 3.92.

(3.91) $$\vec{V} \cdot \hat{n} = un_x + vn_y = 0$$

(3.92) $$F_i n_x + G_i n_y = \begin{Bmatrix} (\rho)(un_x + vn_y) \\ (\rho u)(un_x + vn_y) + pn_x \\ (\rho v)(un_x + vn_y) + pn_y \\ (\rho e + p)(un_x + vn_y) \end{Bmatrix} = \begin{Bmatrix} 0 \\ pn_x \\ pn_y \\ 0 \end{Bmatrix}$$

For viscous flow, the symmetry boundary condition must again apply zero velocity normal to the boundary, while also requiring that no velocity gradient occur across the symmetry plane. These conditions mean that viscous shear stress cannot occur on the boundary. The symmetry condition can, on the other hand, allow heat and viscous stresses parallel to the symmetry plane. Therefore, starting with Equations 2.52 through 2.58, the only remain terms are shown below:

(3.93) $$F_v n_x + G_v n_y = \begin{Bmatrix} 0 \\ \tau^*_{xx} \\ 0 \\ u\tau^*_{xx} \end{Bmatrix} n_x + \begin{Bmatrix} 0 \\ 0 \\ \tau^*_{yy} \\ v\tau^*_{yy} \end{Bmatrix} n_y$$

where

(3.94) $$\tau^*_{xx} = \frac{\lambda}{\mathrm{Re}} \mu \frac{\partial v}{\partial y}$$

(3.95) $$\tau^*_{yy} = \frac{\lambda}{\mathrm{Re}} \mu \frac{\partial u}{\partial x}$$

The shear stresses have been dropped altogether, and the components of the normal stresses and heat transfer terms that occur parallel to their respective normal vector (i.e., n_x, n_y) have also been dropped. The remaining terms are the components of the velocity divergence tangential to the symmetry plane. Equations 3.93 and 3.95 are only good for symmetry planes parallel to the x- and y-directions. Any symmetry plane

used in a direction not parallel to the cardinal directions would need to consider the gradients in the local frame as a reference to dropping terms.

3.4.2. Far-Field Boundary Condition

The original far-field boundary conditions use one-dimensional Reimann invariants to encourage the flow to return to the freestream conditions on the far-field boundaries of the flow. This condition is a weakly applied boundary conditions since the boundary conditions are not strongly imposed on the points but rather are suggested using the Reimann invariants.

For a viscous far-field boundary condition, the velocity gradients should go to zero at the boundary. In this case the Reimann invariants can be left in their original form. This assumption works well until the gradients at the boundary become large. Large gradients only exist in shocks, boundary layers, wakes, and vortices. This thesis will therefore assess the relative breakdown of the solution near the far-field boundary and make suggestions from that assessment.

3.4.3. Inviscid Wall Boundary Condition

The inviscid wall boundary of the original solver assume no flow normal to the surface. This condition is strongly applied on the boundary using Equation 3.96. The boundary integral is applied using Equation 3.90.

(3.96) $$\vec{V} = \vec{V} - \left(\vec{V} \cdot \hat{n}\right)\hat{n}$$

Because this boundary condition is for an inviscid wall, viscous terms were not included in the boundary integrals. This boundary condition does remain for two general

purposes: (1) To generate shocks and other conditions that need to be present around viscous boundaries, when a viscous boundary is not necessary; (2) to generate inviscid surfaces which can be used to simulate flow geometry, such as engine exhausts.

3.4.4. Viscous Wall Boundary Condition

The viscous wall boundary condition is a new addition generated by this thesis. The inviscid integrals, F_i and G_i, are taken from Equation 3.90. For the remaining viscous integrals, all of the remaining viscous terms are included along with a strong condition represented by the no-slip boundary conditions. For the no-slip boundary condition to exist, the velocity of the fluid on the surface must be equal to the velocity of the surface, represented by the following equation:

(3.97) $$\vec{V}_{vs} = \vec{V}_{bel}$$

where $\vec{V}_{vs}$ is the velocity of the fluid on the viscous surface and $\vec{V}_{bel}$ is the velocity of the boundary element, or velocity of the solid surface.

If the surface is not allowed to move, Equation 3.97 is simplified to a zero velocity condition. This condition can then be used to assess the viscous dissipation near a solid surface. Figure 3-6 shows the distribution of velocity, shear stress, and viscous dissipation in the Blasius solutions. The Blasius solution was generated using a 4^{th} order Runge-Kutta algorithm to solve Equation 3.98. In Figure 3-6, the velocity distribution was created by plotting f', the shear stress from f'', and the viscous dissipation by multiplying the previous two plots.

(3.98) $$f''' + ff'' = 0$$

(3.99) $$f = f(\eta)$$

(3.100) $$\eta = y\sqrt{\frac{U_e}{2\nu x}}$$

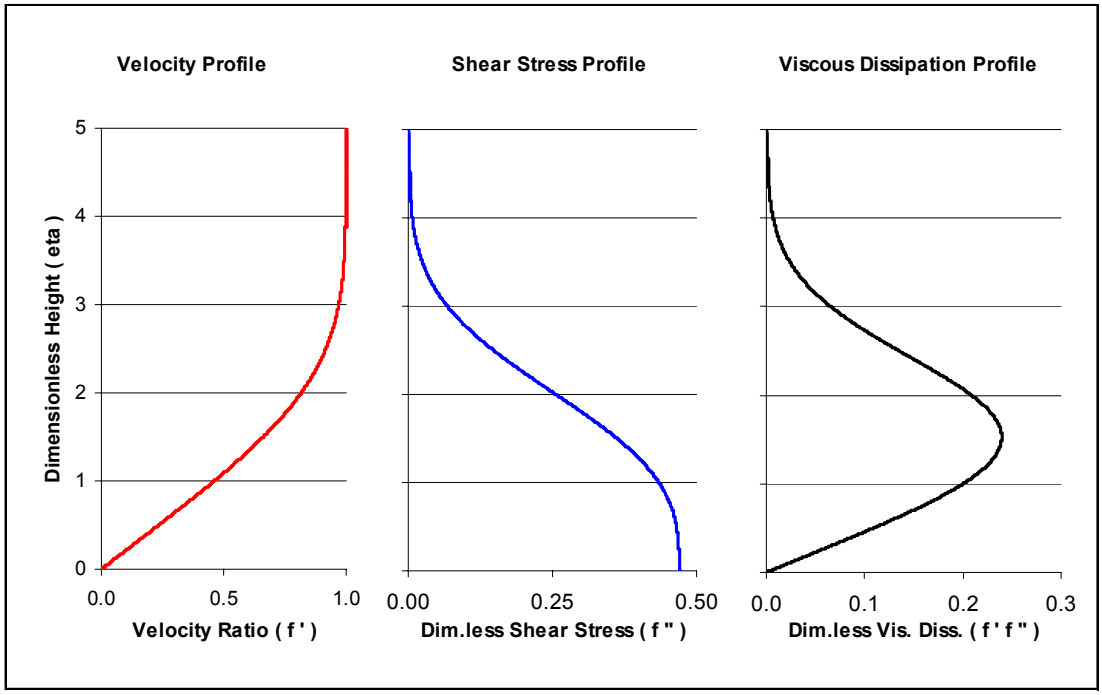

Figure 3-6: Viscous Dissipation near a Viscous Surface

The viscous dissipation plot goes to zero showing that there is no need to include viscous dissipation in the viscous wall boundary integral. Therefore, these terms are excluded from the solution. Again using Equations 2.52 through 2.58 and dropping the unnecessary terms, the viscous wall boundary integrals are solved using the following viscous terms, F_v and G_v:

(3.101) $$F_v n_x + G_v n_y = \begin{Bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ 0 \end{Bmatrix} n_x + \begin{Bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ 0 \end{Bmatrix} n_y$$

where

$$\tau_{xx} = \frac{2\mu}{\mathrm{Re}}\frac{\partial u}{\partial x} + \frac{\lambda\mu}{\mathrm{Re}}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \tag{3.102}$$

$$\tau_{yy} = \frac{2\mu}{\mathrm{Re}}\frac{\partial v}{\partial y} + \frac{\lambda\mu}{\mathrm{Re}}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \tag{3.103}$$

$$\tau_{xy} = \frac{\mu}{\mathrm{Re}}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{3.104}$$

After implementing the above equations in the viscous solid wall boundary integrals, the residual for the momentum equations is limited to a minimum value that far exceeds that of the continuity and energy residual. The previous viscous wall equations contribute only to the right-hand-side vector for the nodes on the boundary. The strong boundary condition is later applied to these conditions, therefore the viscous wall contributions in right-hand-side vector only add to overall residual of the flow field. Given the lack of application for the boundary integral values, the integrals were removed from the solver to help converge the solution residual and decrease the overall run time.

3.5. Stabilization: Artificial Dissipation

The original Euler solver contains two different artificial dissipation models: Low-order and high-order dissipation. The high-order dissipation model is similar to the low-order model except that the high-order model contains gradient limits to allow real flow discontinuities. Because of the calculation time to generate the gradient limiters, the low-order model is much more efficient than the other model.

Cowan found the high-order model to be unstable for high Mach number; but for subsonic and transonic Mach numbers with or without vorticity, the high-order model is

much more useful. The high-order model is also required for the non-inertial calculations because the low-order model will remove any relative velocities from the flow field. Because of the effects of the low-order model on real flow gradients, all of the viscous test cases and their Euler counter parts have been generated using the high-order model (unless otherwise specified).

CHAPTER 4

4. Computer Implementation

The Euler governing equations were extended from their original form to the Navier-Stokes equations in Chapter 2 and then applied to the finite element method in Chapter 3. This chapter will discuss how the code was expanded to handle the changes outlined by the two previous chapters. The discussion will begin with the additional controls and storage required to handle the new viscous terms. The optimal implementation of Sutherland's equation will also be discussed. Further, the original algorithm will be outlined along with sections added by this research. The supporting programs generated or amended by this research will be outlined.

One major obstacle was encountered during startup and similar situations: The solution may become unstable due to the coupling of the local time stepping and the viscous forces applied on elements. This problem occurs mainly for low Reynolds numbers and suggestions are given to help handle the problem. Finally, this chapter will be concluded with an overview of memory requirements and computational runtime compared with the original solver.

4.1. Data Structures and Controls

This section will begin by discussing the variables used to control the Euler solver and additional controls required for the new viscous solver. Then the data structures used

by the original solver are outlined along with additional storage required to evaluate the viscous terms. Finally, this section will wrap up with a discussion of how Sutherland's equation is handled in the new solver.

4.1.1. Control Variables

The variables used to control the number of values stored in the geometry arrays are shown in Table 4-1. The geometry controls are also used to compute the appropriate number of iterations when evaluating the elemental and boundary integrals. Two new geometry controls are used by the viscous solver. These variables are discussed below. The overall solver control variables are represented in Table 4-2. These variables are used to control the flow properties, number of calculations, and output files. Four additional parameters are needed in the new solver.

Table 4-1: Geometry Control Variables

Control Name	Array Description
NND	Number of nodes
NEL	Number of elements
NSG	Number of segments
NBE	Number of boundary edges
NBP	Number of boundary points
NWL	Number of nodes on solid walls (Euler and viscous)
NSD	Number of singular nodes on solid walls (Euler and viscous)
NWLV	Number of nodes on viscous solid walls
NSDV	Number of singular nodes on viscous solid walls
LBE	Limits for boundary edge groups

All of the controls shown in Table 4-1 can be found in the geometry file, *.g2d. The arrangement of the geometry file can be seen in Appendix A. The original solver required eight geometry controls; two new parameters have been added by the viscous

solver. The Euler solver requires the number of solid wall nodes to evaluate the flow tangency condition. The original solver also requires the number of singular nodes so that the tangency condition can be omitted from these nodes. The new viscous solver requires the Euler and viscous surfaces to be separated in order to streamline the evaluation of boundary integrals. In order to accommodate both solvers with the same controls, the number of viscous nodes *NWLV* has been included in the number of solid wall nodes *NWL*. Similarly, the number of singular viscous nodes *NSDV* has been included in the number of singular nodes *NSD*. Because *NWL* and *NSD* contain all of the solid surface counts, the original solver will treat all of the surfaces equally whereas the new viscous solver will treat the nodes according to the appropriate solid wall conditions. This idea is further expanded in the next section where the order of points and boundary edges is discussed. Figure 4-1 and 4-2 are used to illustrate each principles.

Table 4-2: Solver Control Variables

Control Name	Array Description
DT	Global time step (dimensionless)
GAMMA	Ratio of specific heats
DISS	Dissipation factor
CFL	Local time step stability factor
LAMB	Ratio of viscosity coefficients
SMOD	Modified Sutherland's coefficient
MACH	Freestream Mach number
RE	Freestream Reynolds number
PR	Prandtl number
ALPHA	Freestream angle of attack
REFDIM	Reference dimension
NSTP	Number of solution steps
NOUT	Output frequency: Number of steps between output files
NCYC	Iterative cycles per solution step
ISOL	CFD solution type
IDISS	Dissipation type
IPNT	Number of points for Gauss quadrature used on elemental integrals
ISTRT	Restart flag
IAERO	Aerodynamic forces flag
IDYNM	Dynamics and/or non-inertial flag
IELAST	Elastic flag
IFREE	Freestream flag
IFORCE	External forcing flag
NR	Number of elastic modes
AINF	Freestream speed of sound (dimensional)
RHOINF	Freestream density (dimensional)

All of the controls shown in Table 4-2 can be found in the controls file, *.con. The arrangement of the controls file can also be seen in Appendix A. The original solver required twenty-two controls; four new controls have been added by the viscous solver. These new controls include the freestream Reynolds number *RE*, which is used to control the coefficient of viscosity in the solution through dimensionless means. The local coefficient of viscosity is controlled by Sutherland's equation. This equation is represented in Equation 2.61, where the modified Sutherland's constant is given by

Equation 2.47. The modified Sutherland's coefficient *SMOD* represents the dimensionless form of the Sutherland's constant. The local coefficient of viscosity is further related to the second coefficient of viscosity through their ratio *LAMB*, and the local conduction coefficient is related to local viscosity through the Prandtl number *PR* of the flow. Through these four variables all of the viscous and conduction effects can be controlled.

4.1.2. Data Structures

The original data structures, i.e. storage arrays, can be summarized in the table that follows. No new data structures have been added by this research, but the data structures have been amended to include more of the required data. The first five structures are read in from various input files; the next ten arrays are generated within the preprocessing section of the solver and used in the aerodynamics solution; and, the last three are developed during preprocessing and used in the elastics solution. Each of these three groups are overviewed below, followed by amendments to the data structures.

Table 4-3: Data Structures

Structure Name	Array Description
COOR	Nodal coordinates
IELM	Element connectivity
ISEG	Segment connectivity
IBEL	Boundary edge points,
PHIA	Nodal mode shape vector
G2D	Area and gradient of shape function for each element
WSG	Segment weights
DM	Lumped mass matrix
DELT	Local time step vector
RBE	Edge length and unit outward facing normal
ANOR	Area weighted wall normals
GRX, GRY	Nodal gradient in the x- and y-directions, respectively
RHS	Right hand side vector
RSD	Residual vector
UNO, UN, UN1	Previous, current, and predicted future unknowns vectors, respectively
BVEL	Elastic wall deformation and velocities
XN, XN1	Current and future conditions for mode shapes
FA	Generalized forces for mode shapes
RM, RC, RK	Mass, damping, and stiffness matrices, respectively

Input Data Structures: The first four structures listed here are read in from the geometry file, or *.g2d. *COOR* contains an x- and y-coordinate for each node in the mesh. The nodes in the mesh are indexed from 1 to *NND*. *IELM* and *ISEG* represent the nodal connectivity for the elements and segments, respectively. *IELM* contains three nodal indices for each element, representing the three vertices of the triangle; *ISEG* contains two indices for the end points of each segment. The elements of *IELM* are numbered from 1 to *NEL*, and the segments of *ISEG* are numbered from 1 to *NSG*.

The original *IBEL* contains the indices for the two nodes on each boundary edge. In order to increase the efficiency in evaluating boundary integral effects, Cowan chose to organize *IBEL* so that the first group of edges correspond to solid wall conditions,

followed by a group of symmetry plane edges, and then the far field conditions. The limits of these three groups are set by the control *LBE*. *LBE*(1) marks the beginning of the solid wall boundary edges, and *LBE*(2) marks its end. *LBE*(3) and *LBE*(4) designate the symmetry boundary edges while the far field edges are bounded by *LBE*(5) and *LBE*(6). The new viscous boundary edges are bounded by *LBE*(7) and *LBE*(8), such that these two constraints lie within the bounds of *LBE*(1) and *LBE*(2). Therefore, the Euler surfaces are listed first, followed by the viscous boundary edges. The original solver will only read the first six *LBE*s, treating both solid wall surfaces as Euler boundary edges. The new viscous solver differentiates between the two types, evaluating all four boundary conditions. The order of *IBEL* is illustrated in Figure 4-1 below.

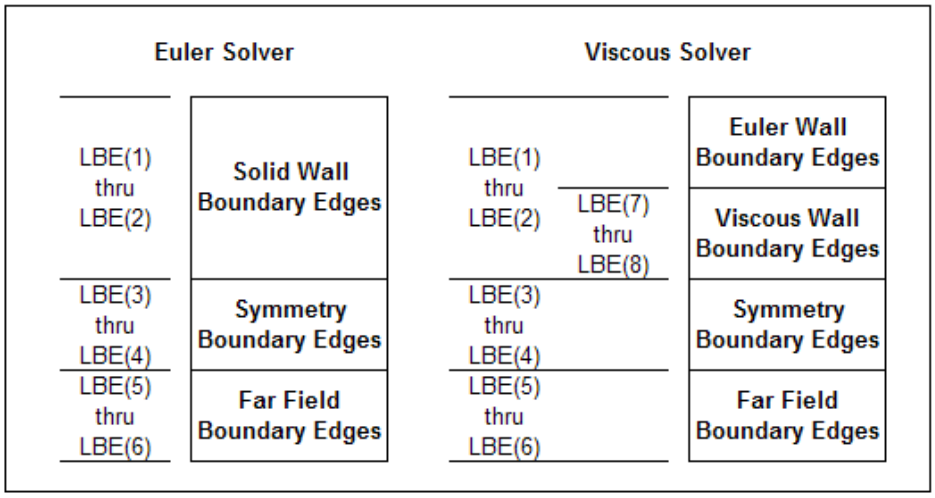

Figure 4-1: Order of Boundary Edges in Euler and Viscous Solvers

Similarly, the nodes in *COOR* are ordered to aid in the evaluation of the tangency condition. The original solver separates the first *NWL* nodes to lie on the solid wall edges. The solver applies the strong tangential boundary conditions to all of these nodes except the last *NSD* singular nodes. To accommodate the differences in the Euler and

viscous solvers, the viscous nodes are placed so that they overlap both of these areas. *NWLV* viscous wall nodes are contained within the *NWL* solid wall nodes. The first *NSDV* nodes of the singular nodes are also viscous nodes. The node order is best represented by Figure 4-2. Therefore, both solvers apply the tangency condition to the Euler wall group of nodes, whereas only the Euler solver applies the condition to the viscous wall group. Finally, the viscous groups are set equal to the boundary velocity in the viscous solver.

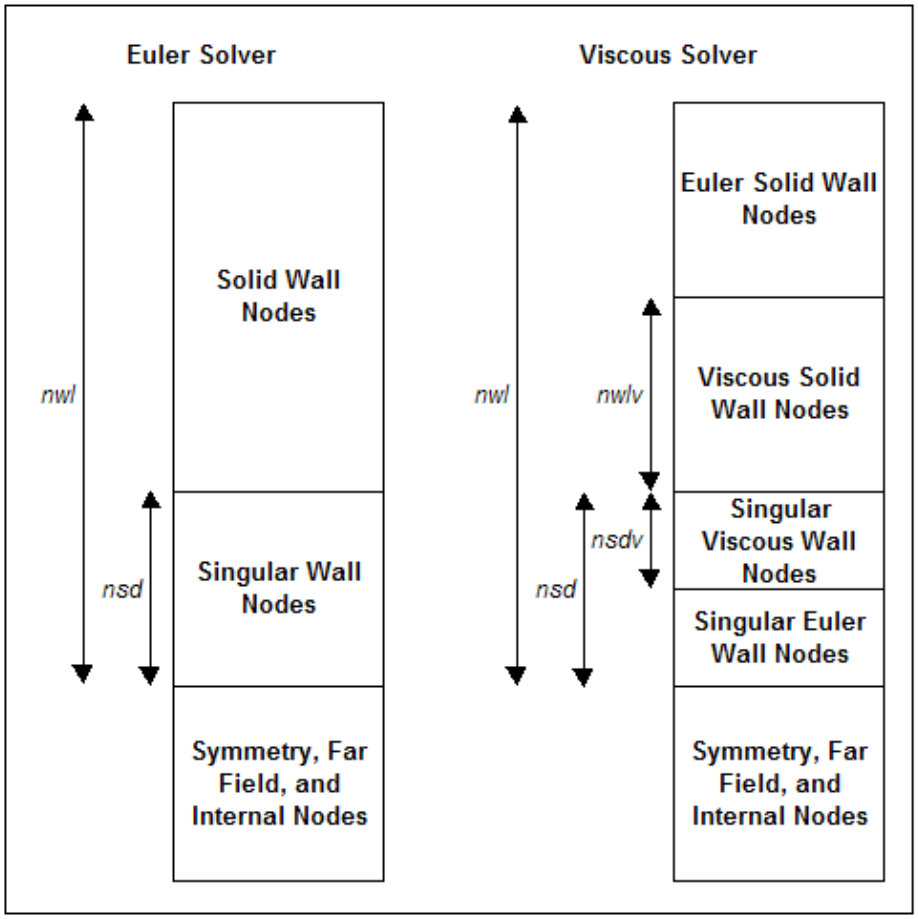

Figure 4-2: Order of Nodes in Euler and Viscous Solvers

The final structure is read from the mode shape vectors file, or *.vec. *PHIA* contains the nodal shape displacement of each mode in the structural model. Each mode shape is represented by a node-by-node distribution of the mode shape.

The content and size of *COOR*, *IELM*, *ISEG*, and *PHIA* were left in their original form. *IBEL* needed to be adapted to include information about the boundary edge gradient. The original Euler solver only needed to contain the indices for the two nodes on the boundary edge. An index for the attached element was added in order to evaluate the new viscous gradients on the boundary edges. The gradient on the linear element is constant and therefore can be applied as a constant to the attached boundary edge. The gradients are evaluated on all element in the flow field during the preprocessing portion of the run. These gradients are stored in the *G2D* array. Using the index for the attached element, the gradients can be called out of the *G2D* array without any additional preprocessing or storage.

Aerodynamic Data Structures: As previously mentioned, the *G2D* array is used to store the element area and 2 by 2 gradient matrix. The element gradient matrix and area were developed previously in Equations 3.42 and 3.43, respectively. *RBE* contains the boundary edge normal vector and edge length; *ANOR* contains the area weighted normal for solid boundary nodes. *RBE* is used to hold the values of Δx , n_x, and n_y in the Equations of Section 3.3.2. *ANOR* is evaluated using the normals for the two boundary edges on either side of each node on a solid surface. *WSG* contains the length weighted gradients for each segment. *GRX* and *GRY* are the nodal gradient in the x- and y-directions, respectively. These segment weights and nodal gradients are used to evaluate

the amount of dissipation needed at each node. These six data structures are used to define the complete geometry of the flow field.

DM and *DELT* are the element mass matrix and local time steps for each node in the flow field. The mass matrix was previously defined by Equation 3.51, and the local time step was developed in Equations 3.3 and 3.4. *UNO*, *UN*, and *UN1* are the unknowns values at all nodes in the flow field at the previous, current, and future time steps, respectively. The unknowns vector was previously defined by Equation 3.6. *BVEL* is the boundary velocity evaluated by applying the structural solver or force surface motion to each node on the solid surfaces. Equation 3.2 represents the right-hand side vector, stored in *RHS*. The vector is evaluated by summing all of the contributions of Equation 3.2 for every node in the flow field. (The time derivative vector in Equation 3.2 is neglected for evaluation of the steady right-hand side vector.) This vector is then used to solve for the unknowns at the next iteration. The right-hand side vector is also used to evaluate the residuals vector, *RSD*. The residuals vector contains one residual for each of the four equations evaluated. The residual for each equation is defined as the root-mean-square of all of the corresponding *RHS* contributions. Because the flow velocities on the viscous surfaces are generated by setting the flow velocity equation to the surface velocity, the *RHS* for the viscous surfaces does not contribute to the solution. Therefore, before the residual vector is evaluated the *RHS* at each node on the viscous surfaces is set to zero.

Structural Arrays: *XN* and *XN1* represents the current and future displacement mode shape factors. These factors are multiplied by the *PHIA* mode shape vector in order to find the total displacement at each node. *FA* is the generalized force for each

mode shape; *RM*, *RC*, and *RK* are the mass, damping, and stiffness matrices for the structural solver. The generalized force vector contains a generalized force component for each of the *NR* modes. Each of the three matrices is a square matrix *NR* in rank. All of the structural data structures were left to their original size and content.

4.1.3. Sutherland's law

Sutherland's law has been previously given in Equation 2.61. There are two methods to store Sutherland's equation in the solver: (1) Function calls or (2) look up tables. Because the actual function would include several divisions and rational powers, the function could be computationally expensive. Another method is to store the proper values in an array during the preprocessing stage of the algorithm. The array could then be used to find the most appropriate viscosity for a given energy using various interpolation routines. In other words, the array is used as a "look up table".

The solver was first written so that the dimensionless viscosity of Equation 2.61 is always equated to unity. Then four cases were tested for the increase computational cost for the two methods. The first two cases explored the use of function calls. The first function used the rational power of Equation 2.61 directly; the second function used the ratio of energies times the square root of the ratio. The final two cases investigated look up tables. One cases tested linear interpolation with an 100,000 element array; the second used quadratic interpolation on the same array. Both arrays were stored so that the energy could be converted to an appropriate index and the index could be used to look up the viscosity ratio. A laminar boundary layer in subsonic flow was used as the test

bed for all five test cases. Table 4-4 shows the run time results and percent increase over the baseline of $\mu = 1.0$ at all locations.

Table 4-4: Results of Sutherland's Equation Tests

	Run Time (sec)	Percent Increase
Mu = 1	103.11	N / A
Function Call with Rational	239.39	132%
Function Call with Square Root	119.89	16%
Array Call with Linear Interp.	132.88	29%
Array Call with Quadratic Interp.	158.05	53%

The results show that the function call is the best method when the square root is used in place of the rational power. This method still increases the overall run time by a factor of 16%. Because of this cost, an option is included in the solver so that the $\mu = 1.0$ case can be used to accelerate the run time when a variable viscosity problem is not required. The viscosity ratio can be set to unity everywhere by entering a value of 0.0 for *SMOD*. Any value greater than 0.0 will represent the ratio using Equation 2.61.

4.2. Aerodynamic Forces

The original Euler solver computed all three of the aerodynamic coefficients using on the average pressure distributed over the boundary edge. Each edge is linearly distributed; therefore, the average pressure on each edge can be found by averaging the pressure at the two nodes on either side of the edge. Using this average pressure and the

components of the surface normal, the components of the aerodynamic force can be found. These force components can also be used to find the moment about the origin. The non-dimensional form of these three coefficients is shown in the following equations:

$$C_x = \sum_{i=LBE(1)}^{LBE(2)} 2\Delta x\left(p^*_{avg} - p^*_0\right)n_x \tag{4.1}$$

$$C_y = \sum_{i=LBE(1)}^{LBE(2)} 2\Delta x\left(p^*_{avg} - p^*_0\right)n_y \tag{4.2}$$

$$C_m = \sum_{i=LBE(1)}^{LBE(2)} 2\Delta x\left(p^*_{avg} - p^*_0\right)\left(R_x n_y - R_y n_x\right) \tag{4.3}$$

where p^* is the dimensionless pressure used in Section 2.6. The generalized aerodynamic force used by the structural solver can be found similar to Equations 4.1 through 4.3. The generalize force vector can be found by summing the pressure for all nodes on the solid surfaces dotted with the mode shape at that point. The generalized force vector is illustrated in the following equation:

$$f_i = \sum_{j=1}^{NWL} 2\Delta x\left(p^*_j - p^*_0\right)\left(n_x\phi_{ij,x} + n_y\phi_{ij,y}\right) \tag{4.4}$$

where *NWL* is the number of wall nodes, p^*_j is the dimensionless pressure at node *j*, and ϕ_{ij} is the mode shape displacement of mode *i* at node *j*.

For the new viscous solver, the shear stress along viscous solid surfaces must also be addressed. The shear stress vector is developed in full in Appendix C, but a short version of that development is given here. The local shear stress on a boundary edge is given with reference to the local elemental coordinate system. The magnitude of the local shear stress is given by the following equation:

(4.5) $$\hat{\tau}_{xy} = \mu\left(\frac{\partial \hat{u}}{\partial \hat{y}} + \frac{\partial \hat{v}}{\partial \hat{x}}\right)$$

The local gradients can be found in terms of the global gradients on the element. The conversion from global to local gradient is given by:

(4.6) $$\begin{Bmatrix} \frac{\partial \hat{u}}{\partial \hat{y}} \\ \frac{\partial \hat{v}}{\partial \hat{x}} \end{Bmatrix} = \begin{Bmatrix} -C_\theta S_\theta \frac{\partial u}{\partial x} - S_\theta^2 \frac{\partial v}{\partial x} + C_\theta^2 \frac{\partial u}{\partial y} + S_\theta C_\theta \frac{\partial v}{\partial y} \\ -C_\theta S_\theta \frac{\partial u}{\partial x} + C_\theta^2 \frac{\partial v}{\partial x} - S_\theta^2 \frac{\partial u}{\partial y} + S_\theta C_\theta \frac{\partial v}{\partial y} \end{Bmatrix}$$

Combining Equations 4.5 and 4.6, the magnitude of the local shear stress is given by:

(4.7) $$\hat{\tau}_{xy} = \mu\left(2C_\theta S_\theta\left(\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right) + \left(C_\theta^2 - S_\theta^2\right)\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right)$$

The magnitude of the local shear stress can be assumed in the local x-direction of the boundary edge. Using the same local to global coordinate transform, the local shear stress can be rotated back into the global reference frame to find the actual shear stress vector, given by Equation 4.8.

(4.8) $$\vec{\tau} = \hat{\tau}_{xy}\begin{Bmatrix} \cos\theta \\ \sin\theta \end{Bmatrix}$$

The equations can be converted to their final form by using the components of the surface normal to replace the trigonometric ratios: $\sin\theta = -n_x$ and $\cos\theta = n_y$. Combining Equations 4.7 and 4.8 and using the surface normals, the shear stress vector for a viscous boundary edge is represented by the following:

(4.9) $$\vec{\tau} = \hat{\tau}_{xy}\begin{Bmatrix} n_y \\ -n_x \end{Bmatrix} = \mu\left(-2n_y n_x\left(\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right) + \left(n_y^2 - n_x^2\right)\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right)\begin{Bmatrix} n_y \\ -n_x \end{Bmatrix}$$

Equation 4.9 can be non-dimensionalized by dividing the shear stress vector by the freestream dynamic pressure. Equation 4.10 represents the coefficient of friction as a vector for a boundary edge.

$$\vec{c}_f = \frac{\mu^*}{\mathrm{Re}}\left(-2n_y n_x\left(\frac{\partial v^*}{\partial y^*} - \frac{\partial u^*}{\partial x^*}\right) + \left(n_y^2 - n_x^2\right)\left(\frac{\partial v^*}{\partial x^*} + \frac{\partial u^*}{\partial y^*}\right)\right)\begin{Bmatrix} n_y \\ -n_x \end{Bmatrix} \tag{4.10}$$

The coefficient of friction can be summed in vector form across all of the viscous boundary edges to find the three aerodynamic coefficients. This process is similar to that of summing the coefficient of pressure across solid wall elements. The three aerodynamic coefficients are equated to the shear stress below:

$$C_x = \sum_{i=LBE(7)}^{LBE(8)} 2\Delta x c_f n_y \tag{4.11}$$

$$C_y = \sum_{i=LBE(7)}^{LBE(8)} -2\Delta x c_f n_x \tag{4.12}$$

$$C_m = \sum_{i=LBE(7)}^{LBE(8)} -2\Delta x c_f \left(R_x n_x + R_y n_y\right) \tag{4.13}$$

The generalize force vector can be found by summing the shear stresses for all nodes on the viscous surfaces dotted with the mode shape at that point. The generalize force vector is illustrated in the following equation:

$$f_i = \sum_{j=1}^{NWLV} 2\Delta x c_f \left(n_y \phi_{ij,x} - n_x \phi_{ij,y}\right) \tag{4.14}$$

where *NWLV* is the number of viscous solid wall nodes, c_f is the coefficient of friction at node *j*, and ϕ_{ij} is the mode shape displacement of mode *i* at node *j*.

The final form of the aerodynamic coefficients can be found by summing up the contributions due to pressure across all solid wall boundary edges (Equations 4.1 through

4.4) and by summing the contributions due to shear stress across all viscous wall edges (Equations 4.11 through 4.14). Because of the overlapping nature of the boundary edge limit controls, *LBE*s, the equations can be used in their given form and both the inviscid and viscous surfaces will be handled in the pressure summation while only the viscous surfaces will be used in the shear stress summation.

4.3. Basic Algorithm

The algorithm for both the Euler and Navier-Stokes solvers is represented by the pseudocode below. All lines of the algorithm previous to the advancement of the first time step are considered part of the preprocessing of the algorithm. There are six lines of the algorithm that are marked as conditional. For non-inertial solutions, the dynamics file must be read during preprocessing while the elastic vectors file must be read for aeroelastic cases. Both of the previous two cases require the unsteady solver for time accuracy. Therefore, both non-inertial and elastic cases apply the transpiration boundary conditions, assemble the time derivatives using the jump condition, calculate forces on solid surfaces, and step the structural/dynamics solver to generate boundary conditions inputs for the next step of the CFD solver.

Adaptations to generate the viscous solver include input/output files, initial and boundary conditions, shear forces, and the governing equations. There were several changes made to the control and geometry files; these changes were reflected in the preprocessing section of the code. The unknowns output file was amended to include the freestream Reynolds number in the header. These changes are discussed in Section 4.1 and are shown in the file formats in Appendix A.

```
Read control data
  Read geometry data
  Assemble additional geometry data
     Compute element Jacobeans & areas
     Invert global lumped mass matrix
     Compute length & normal of boundary edges
     Add element & boundary edge weights to segments
  Open and read non-inertial / dynamic file (if necessary)
  Open and read elastic vectors file (if necessary)
  Compute elastic deformation & velocity for wall nodes
  Apply initial conditions
  Apply flow tangency
  Compute forces on solid surfaces
  Solve structural dynamics equations
  Apply transpiration boundary conditions (optional for inviscid walls)
  Advance CFD solution one time step
     Loop over all segments and compute nodal time steps
     Loop through inner cycles
        Reapply solid wall boundary conditions
        Assemble jump condition (unsteady)
        Assemble element flux integrals
        Assemble source vector integrals
        Assemble inviscid solid wall boundary integrals
        Assemble symmetry boundary integrals
        Assemble far-field boundary integrals
        Add dissipation to RHS vector to smooth solution
        Apply flow tangency to inviscid wall nodes
        Update solution for next cycle
        Apply no slip condition for viscous solid walls
     Repeat for NCYC times
     Output residuals
     Compute generalized forces acting on walls (optional)
     Advance structural dynamics for the next time step (optional)
     Apply transpiration B.C. for elastic modes
     Output contents of unknowns vector
  Repeat for NSTP times
```

Figure 4-3: Pseudocode for Euler and Navier-Stokes Solvers

The initial and boundary conditions for the viscous solver include a strong application of the no slip condition. Therefore, the velocity at all viscous wall nodes is

set equal to the wall velocity when the initial conditions are applied on the flow field. Further, the strong boundary conditions are reapplied at the end of each inner cycle.

In the preprocessing section and at the end of each CFD iteration, the forces are summed on all solid wall surfaces. The original code summed the contributions due to pressure on all solid surfaces. The new code includes this summation over both the inviscid and viscous solid surfaces. Then the code sums the shear forces on all viscous solid surfaces. The application of element gradients to find the shear force vector is discussed in Section 4.2. This application is made not only for the aerodynamic forces and moment coefficients but also for the generalized forces used to march the structural/dynamics solver.

Finally, changes to the governing equations and their finite element forms are given in Chapters 2 and 3. These changes are reflected in the assembly of element integrals and the boundary flux integral terms. Since the far-field, source vector, and inviscid wall integrals were not changed, the code was left in its original in these sections. The same can be said for the dissipation models. The final adaptation for the governing equation was made just before the calculation of the residuals vector. As previously discussed, the viscous wall boundary condition is strongly applied therefore any changes to the velocity terms of the right-hand side vector contribute to the residual vector without actually affecting the solution. Therefore, just before the evaluation of the residual vector, the right-hand side terms for u- and v-velocities at all viscous wall nodes is set to zero. Using this method, the residual vector should converge as for the original code.

4.4. Primary and Supporting Programs

There are five programs used in this research. Each of the five programs are outlined below. The first program is the viscous solver explicitly outlined in this thesis. The second program is the original Euler solver. The third program was originally written to generate the geometry file for the Euler solver; this thesis amended the program to generate geometry files that are valid to both the Euler and viscous solvers. The final two programs are used to edit the unknowns binary file and to interpolate solutions onto new meshes.

NS2d: The viscous solver is the program explicitly outlined in this thesis. The program was adapted from *Euler2d* to include viscous terms. The viscous solver has the capability to handle steady and unsteady viscous flows. No turbulence model is available in the current *NS2d* program, therefore the solver can only handle laminar flows. The solver does retain the non-inertial source terms from *Euler2d*. On the other hand, the aeroelastic routines of *NS2d* cannot use transpiration to model the deformation of viscous surfaces; yet, the structural model was retained, leaving the ability to expand the solver to include a viscous transpiration model. The format of all input and output files used by *NS2d* is illustrated in Appendix A.

Euler2d: The original Euler solver can be used to generate flow field solutions to seed the viscous solver. The same geometry file can be used to generate an Euler solution, locating shocks, expansion, and large pressure gradients. Then the solution can be used to seed the viscous solver so that the boundary layers and other viscous flow solutions can be generated. The Euler solver requires less run time and is much more stable at generating start up solutions. The Euler solver has the capability to generate

steady and unsteady flow solutions in either the inertial or non-inertial reference frames. *Euler2d* also has the capability to use transpiration boundary conditions to model aeroelastic responses using linearized mode shapes. The format of all input and output files used by *Euler2d* is illustrated in Appendix A.

Makeg2d: Both the Euler and Navier-Stokes solvers require a prescribed geometry file. The meshing software available in the STARS package is called *surface*. *Surface* can be used to mesh a surface bounded by two or more curves. As long as the mesh lies in only two of the cardinal directions, *makeg2d* can be used to rearrange the nodes and boundary edges according to the requirements of both the Euler and Navier-Stokes solvers. The output file, or *.g2d, can be seen in Appendix A. The input files are called the front file, or *.fro, and the boundary conditions file, or *.bco. The front file can be generated by the *surface* mesh generator. The file formats for both the front and boundary conditions files are also given in Appendix A.

Unkadapt2d: The unknowns files, *.unk or *.un#, contain the dimensionless density, pressure, velocity components, and enthalpy at all of the nodes in the flow. The files also contains several control variables in the header of each file. Both solvers required the number of nodes, ratio of specific heats, Mach number, angle of attack, reference length, and current dimensionless time. The Navier-Stokes solver also requires the freestream Reynolds number. In order to convert an *Euler2d* unknowns file to a *NS2d* unknowns file, *unkadapt2d* can be used to read in the unknowns file and edit the header. *Unkadapt2d* can be used to read in any unknowns file and edit any of the control values except the number of nodes and the reference length. The file can then be written in either the *Euler2d* or *NS2d* formats.

Unkinterp2d: *Unkinterp2d* can be used to interpolate the solutions from one mesh to another mesh. The elements used by both *Euler2d* and *NS2d* are linear triangular elements. The interpolation program requires a geometry *.g2d file for both the old and new meshes, and a unknowns *.unk file for the old mesh. The program sorts the elements in the old geometry file and puts the elements into an optimal arrangement of clusters. The clusters are used to accelerate the search and interpolation process. If the new nodes do not lie within the old mesh elements, the program extrapolates the solutions from the closest boundary edge. Therefore, *unkinterp2d* can be used to interpolate solutions during the mesh convergence process. The program can also be used to place a finer mesh over shocks and other natural flow gradients. And, because of the extrapolation features of unkinterp2d, the program can be used on meshes. The program closes by allowing the user to amend the header of the unknowns file, similar to *unkadapt2d*.

4.5. Suggestions for Local Time Stepping Constraints

Instabilities have been found to occur when starting up a viscous solution from either freestream conditions or from an Euler solution. These problems can also occur if the freestream Reynolds number is decreased too quickly. The instability can occur near viscous surfaces and near anomalies in the flow mesh.

The instability problem arises because of the coupling of the local time step and viscous stresses. At start up, the velocities on the surface of viscous walls are set equal to zero, either absolutely or relative to the non-inertial effects. The elements lying along the viscous surfaces then contain the largest gradient that will be seen in the boundary layer

as the viscous effects develop. This large gradient equates to a large change in the right-hand side vector for the nodes just off the viscous surfaces.

The problem would be much simpler if not for the coupling of the local time step. The local time step is calculated using Equation 4.15, which is a combination of Equations 3.3 and 3.4. Notice that the two velocities used are the average acoustic speed on the segment a_{ij} and the local velocity at the opposite node V_j. For the start up condition, the velocity along the surface is zero, therefore the node just off of the surface suddenly loses this velocity contribution. The summation of velocity contributions on the segment is then inverted; therefore, the decrease in velocity results in an increase in the inverse. This inverse is then used to scale the local time step. The combined effects of the increase in local time step and viscous stresses result in a much larger impulse at the point just off the viscous surface.

$$\Delta t_i = 2\ CFL\ \left[M_L\right]_i \left[\sum_{k=1}^{nsg} \left|\vec{V}_{j,k} \cdot \vec{W}_{ij,k}\right| + a_{ij,k}\left|\vec{W}_{ij,k}\right|\right]^{-1} \tag{4.15}$$

Problems with the local time step have also been found to occur when an anomaly in the mesh occurs. Such an anomaly is illustrated in Figure 4-4. The anomaly allows for some of the segments connected to one particular node to decrease enough that an effect is seen in the velocity summation and therefore the inverse. Problems are not seen at such anomalies unless a gradient or shock wave passes directly over the anomaly. As the wave passes over the anomaly, the forces are increased at the node. Combined with the increased local time, the larger impulse again occurs.

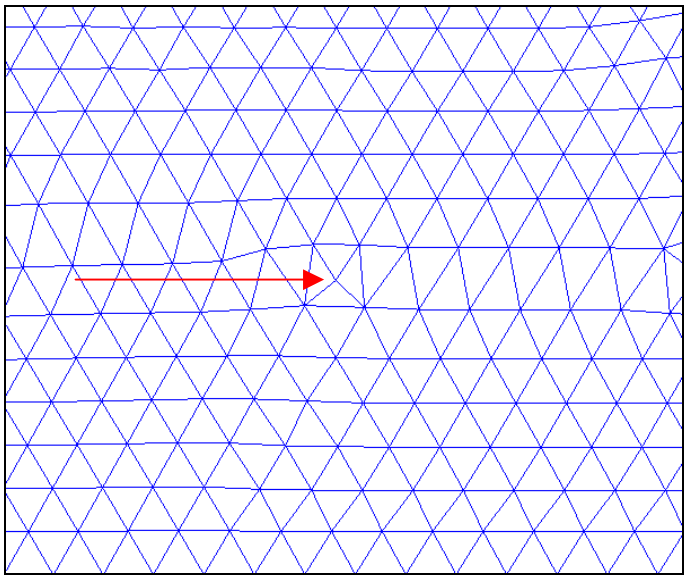

Figure 4-4: Mesh Anomaly Causing an Increase in Local Time Step

The properties at the point of interest are adversely affected by the large impulse values. The properties are over-adjusted in the first iteration. Then the properties are adjusted too far and the gradient increases. An even larger impulse occurs. The properties continue to oscillate in a diverging manner. This process is illustrated for the viscous surface in Figure 4-5 and for a mesh anomaly in Figure 4-6.

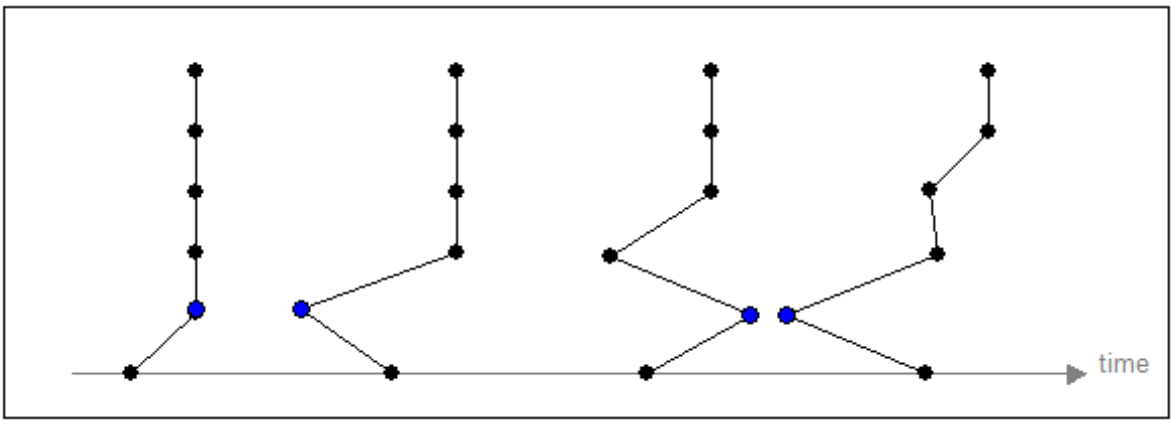

Figure 4-5: Divergence of Along a Viscous Surface

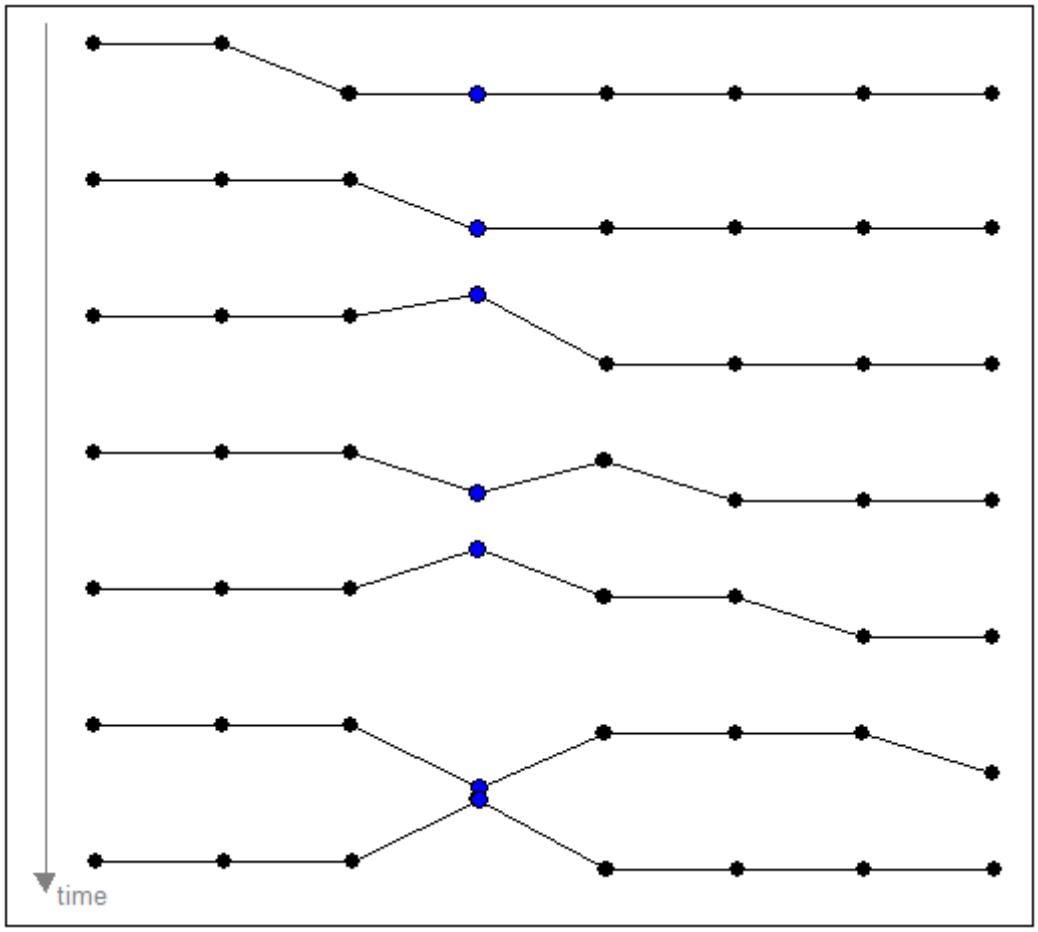

Figure 4-6: Divergence at a Mesh Anomaly

Suggestions for Steady Solutions: Two methods were explored for generating stable steady solutions when the unstable conditions exist. The first methods involves increasing the freestream Reynolds number to decrease the initial viscous effects seen by the solver. This method would decrease the impulse on the problematic nodes. A stable Reynolds number can be found by starting at the desired Reynolds number and stepping by order of magnitude (i.e., $Re_{new} = 10\ Re_{old}$). A stable Reynolds number should result in a converging residual. The stable Reynolds number can be used for several thousand

iterations or until the viscous gradients have settled out of the problem areas of the flow field. The Reynolds number can then be decreased. The residual output must be watched because the new solution can often converge for a short time and then diverge to infinity. Also, the Reynolds number in the unknowns file header must be updated at each change. This process was found to be tedious and required much attention by the user. Therefore, this process is not suggested unless the second method would take too much time.

The second method requires a global stabilization of all of the local impulses. This stabilization can occur by lowering the local time step at each node in the field. Equation 4.15 shows that the *CFL* number can be used to scale all of the local time steps. Figure 4-7 illustrates the results of nine tests ran to find the appropriate *CFL* number for various Reynolds numbers. The tests varied the geometry, freestream Reynolds number, dissipation, and number of inner cycles.

Two geometries were used in the tests: A laminar boundary layer over a flat plate and a circular cylinder. The boundary layer was started with freestream conditions at all locations in the flow. The smallest segment in the boundary layer meshes was varied through a range of 0.0008, 0.0043, and 0.0208 of the reference dimension. The cylinder was tested under a freestream and Euler start up solutions. The smallest mesh for the cylinder was varied through a range of 0.0022, 0.0024, and 0.0027 of the reference dimension. The freestream Reynolds number, dissipation constant, and number of inner cycles was varied for each Reynolds number. The freestream Reynolds number was varied from 1 to 10^9 by orders of magnitude; the dissipation constant was varied from 1.0 to 2.0 by steps of 0.2; and, the tests were given one or five inner cycles.

The results of the tests showed that the dissipation constant had little effect on the stability at start up. In Figure 4-7, the stable *CFL* number varies slightly at the high end of the Reynolds number range. This variation is due solely to changes in dissipation. One inner cycle was found to be slightly less stable than five cycles per solution step. Figure 4-7 represents only the results for five cycles per step. The plot for one cycle shows the same linear trend at low Reynolds numbers and a decrease in maximum CFL available at higher Reynolds numbers, but the decrease is only from *CFL* of 0.7 to 0.3.

The Reynolds number has the greatest effect of the three parameters tested. The effects of Reynolds number are explicitly shown in Figure 4-7. The two regions of the figure represent a stable upper limit of CFL for the solver and stable *CFL*s at unstable Reynolds numbers. The upper limit of CFL for the viscous solver is between 0.3 and 0.7, which is comparable to 0.8 recommended by Cowan (2003).

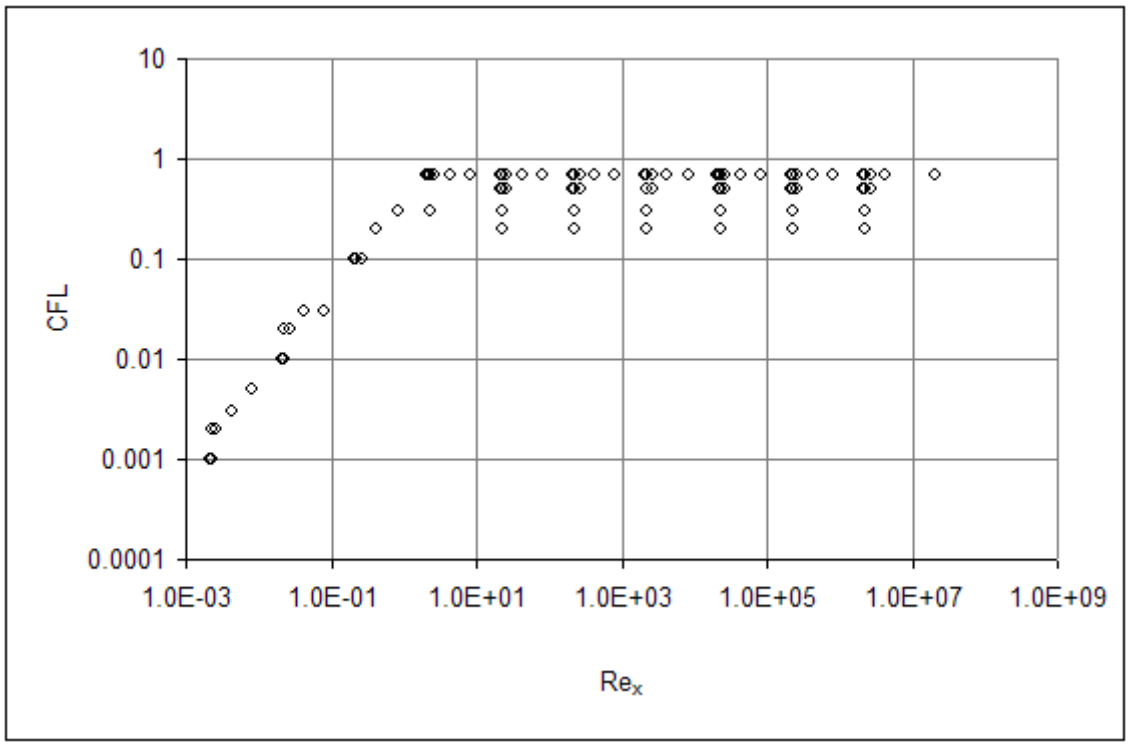

Figure 4-7: Highest Stable CFL vs. Minimum Local Reynolds Number

The linear region of Figure 4-7 has been isolated in Figure 4-8 and approximated by a power law trend. The trend is shown on the graph below. The power generated by the trend is 0.94, which is very close to unity; therefore, the relationship between CFL and local Reynolds number is almost linear. As a final note on these two figures, both figures have been plotted with Re_x on the abscissa. Re_x was derived from the freestream Reynolds number Re_L, the minimum segment length x, and the reference length L. The file organization program *makeg2d* outputs the minimum segment length x as a standard output while the routine compiles the necessary data to generate the geometry file. The other two values can be taken from the control file. Using these three parameters, Equation 4.16 can be used to generate Re_x.

(4.16) $$\mathrm{Re}_x = \frac{x}{L}\mathrm{Re}_L$$

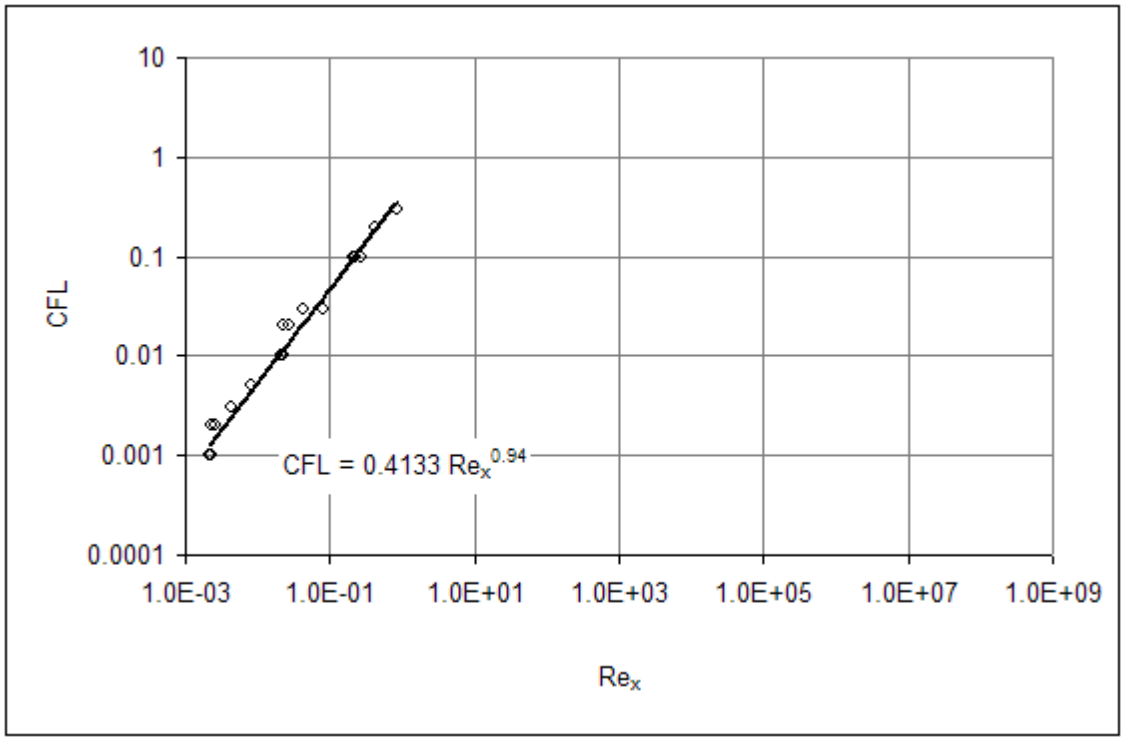

Figure 4-8: Stable CFL Trend for Low Reynolds Numbers

Suggestions for Unsteady Solutions:

Using the trend in Figure 4-8 and the suggestions of Tanahashi (1990) shown in Equation 2.1, the following relationship is suggested to select a stable unsteady time step:

$$\Delta t \le \min\left(\Delta t_{adv}, \frac{0.4\,\mathrm{Re}_x}{CFL}\Delta t_{adv}\right) \tag{4.17}$$

where Δt_{adv} is approximated by the shortest time for fluid to cross any element in the flow. One safe method for solving for Δt_{adv} is to divide the smallest element length by the largest velocity in the flow field. Once the solver has been running for a long enough duration, a subsonic viscous flow field will contain the smallest elements in the boundary layer, where the smallest velocities are also found. On the other hand, the initial steps of the solver may not have small formed the boundary layer yet and the previous suggestion

would give a conservative estimate of the advection time. Supersonic viscous flow fields will contain shocks that will use small elements to represent fast changes in the high speed flow. The global time step will most likely be most restricted by the elements near the shock.

4.6. Summary

To summarize the previous sections, there were several adjustments made to the control variables, data structures, and sections of the algorithm. The basic algorithm of the CFD and structural/dynamic algorithms were not changed, but rather the equations applied to the right-hand side vector were adapted so that the elemental and boundary integrals included not only the Euler fluxes but also the viscous terms.

Changes to the equations were also adapted in the control variables and data structures. Four new control variables were added to include viscous and conduction terms. Reynolds number was added as a dimensionless form of viscosity. The local viscosity was found from the freestream Reynolds number *RE* using Sutherland's equation and the modified Sutherland's constant, *SMOD*. The ratio of viscosity coefficients *LAMB* and the Prandtl number *PR* are considered constant for the flow: The ratio of viscosity coefficients is used to find the second coefficient of viscosity at every point in the flow. Likewise, Prandtl number is used to ratio the effects of viscosity to conduction. The viscous and conduction terms can be controlled completely through these four dimensionless numbers.

All of the data structures were left in their original form except for the boundary element information *IBEL*. Originally *IBEL* included the indices for the two nodes on

each boundary. The new viscous terms require gradients to be evaluated on the boundaries. These gradients are constant across the adjacent element and are therefore found through that element. Therefore, a third component of IBEL was generated to store the element corresponding to the appropriate boundary edge.

Changes to the calculated equations included the evaluation of Sutherland's equation and the calculation of shear stresses along viscous wall surfaces. Sutherland's equation requires several divisions and rational power be accessed. Both processes are computationally expensive; therefore, computationally efficiency was tested to find the best method for evaluating Sutherland's equation. The aerodynamic coefficients and generalized force vector needed to include contributions due to shear stress. The magnitude of the local shear stress was calculated using the global gradients and an off-axis correction. The local shear stress magnitude was then applied to a vector parallel to the surface. This vector could then be transformed back into the global coordinate frame. The shear stress vectors could then be summed for all boundary edges as their contributions to the force and moment coefficients and generalized force vector.

An instability was shown to occur due to the coupling of an increased local time step and viscous stresses. Several suggestions were made to elevate the instability at the cost of computational efficiency and complication. Suggested CFL values were given for various Reynolds numbers in order to stabilize steady solutions. Similar values were suggested for unsteady solutions along with smaller global time steps. A warning was also given concerning the number of inner cycles appropriate to ensure time accuracy. These problems only persist during the start up of a solution and when gradients convect across anomalies in the mesh. Suggestions were also made to identify these areas.

4.6.1. Memory Requirements

The virtual memory requirements of the original code are summed up in Equations 4.17 and 4.18 for the real (8 byte float) and integer storage elements, respectively. The actual size of the compiled executable is 512 kilobytes.

(4.18) $$VM_{real} = 37nnd + 5nel + 2nsg + 3nbe + 4nwl + 5nr + 34nr^2 + 2nr \cdot nwl + 572$$

(4.19) $$VM_{integer} = 34nnd + 3nel + 2nsg + 2nbe + nr + 69$$

The size of the compiled *NS2d* executable is 548 kilobytes, and the virtual memory requirements are given up in Equations 4.19 and 4.20 below, again for real and integer requirements, respectively.

(4.20) $$VM_{real} = 37nnd + 5nel + 2nsg + 3nbe + 4nwl + 5nr + 34nr^2 + 2nr \cdot nwl + 626$$

(4.21) $$VM_{integer} = 34nnd + 3nel + 2nsg + 3nbe + nr + 69$$

Therefore, the viscous terms to the original solver add an additional 54 real (8 byte float) and *NBE* more integer values to the virtual memory and 7% increase in the file size.

4.6.2. Computational Performance

The computational time is dependent upon the number of elements, boundary edges (and type), and segments. The number of segments *NSG* contributes to the time that the solver spends adjusting the dissipation portion of the solution; likewise, the number of boundary edges can be used to assess the time spent solving for various boundary conditions. Both of these contributions are small in comparison to the time spent evaluating the element integrals. In fact, the computational time is almost a linear

relationship of the number of elements in a solution mesh. This linear trend is shown in Figure 4-9. Variations on the relationship are dependent upon the geometry and are related to the time spent evaluating the dissipation and boundary conditions. Figure 4-9 gives two linear trend lines with zero intercepts. These trends represent the computational cost per element in Euler2d and NS2d. The addition of viscous elements has therefore added approximately 31.2% more computational cost for the same mesh.

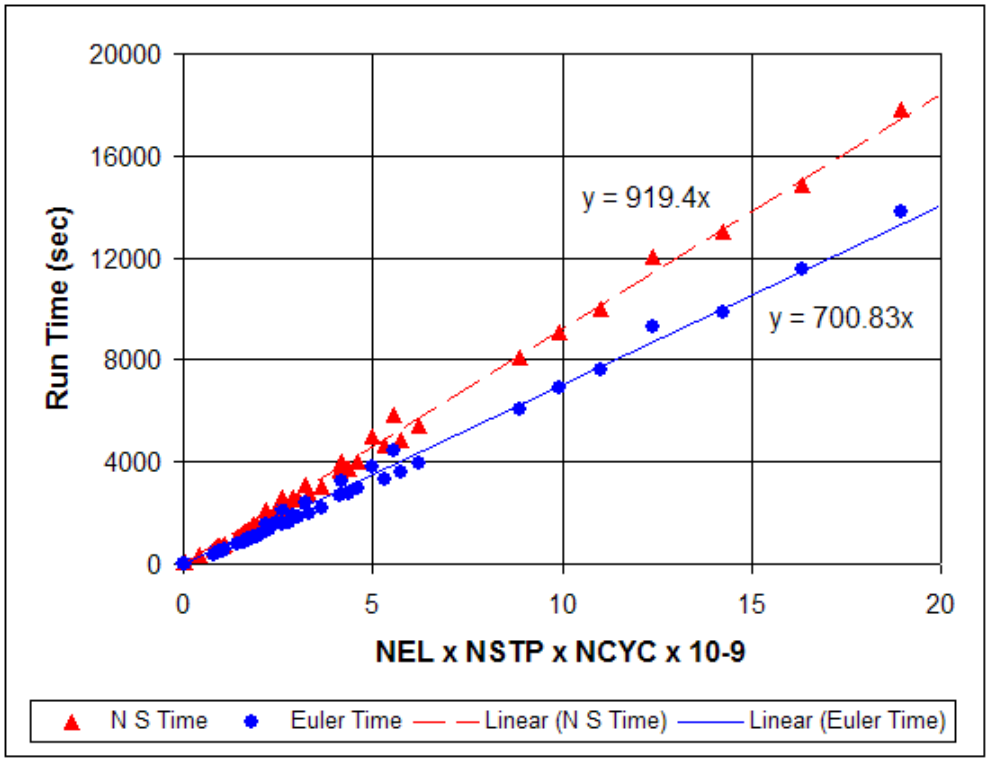

Figure 4-9: Euler2d and NS2d Efficiency Comparison

Most of the previously mentioned time stepping problems occur where the viscous effects are large, generally in low Reynolds number flows. If the Reynolds number is large enough, the natural dissipation may be high enough to make artificial dissipation unnecessary. The artificial dissipation was turned off for several test cases in

this research after the solution was allowed to settle out. To reduce the computational time required by the dissipation models, a bypass option has been added so that the solver will bypass the dissipation subroutine altogether. The bypass option is triggered by a dissipation factor *DISS* of zero.

Using the very low Reynolds number cylinder, three tests were ran for 100 iterations. Each test had a different dissipation model: (1) Low order model, (2) high order model, and (3) no dissipation model. The times are reflected as 172.12, 110.64, and 69.50 seconds. Therefore, the low order model requires 59% more computational time whereas the second order model requires 148% more time. Further studies need to be made into improving the current dissipation models.

As for the increase in the elements required to represent the boundary layer and wakes in test cases used in this thesis, the number of elements has increased by one order of magnitude in most situations. The number of elements for the cylinder cases was increased by a factor of 2.5 to capture the boundary layer and wake. Similarly the boundary layer and wake of the NACA 0012 was increased by a factor of 5. Larger factors can be attributed to large areas of separated flow and shock boundary layer interactions.

CHAPTER 5

5. Results

There are two steps to proving that the viscous terms are both modeling the governing equations and the overall flow physics properly. These two steps are verification and validation. Slater (2004) defines verification as demonstrating that the program and implementation correctly represent the conceptual description of the model or governing equations. Oberkampf and Trucano (2002) describe the process as using simplified analytical models of the governing equations, simplified numerical ODE and PDE models, and general features of the model to show that the model can generate the qualitative results. The verification section below will illustrate a Blasius boundary layer (ODE) to show that *NS2d* models the governing equations properly for simplified flows. Flow visualizations from van Dyke (1982) are also used to show that the general physics of the flow fields are properly handled.

Slater (2004) defines validation as demonstrating that the computational model agrees with physical reality. Oberkampf and Trucano (2002) define validation in the same way. Both sources also require the use of experimental data for validation. Conforming to these standards, *NS*2d is validated through the use of cylinder flows and four airfoil experiments. The tests cover a wide range of Reynolds numbers and subsonic Mach numbers. *NS2d* is also compared with other viscous solvers when applicable.

5.1. Verification

The verification section begins with a laminar boundary layer at a Reynolds number of 90000. The boundary layer is compared to Blasius' solution for both velocity profile, shear stress, and drag on a flat plate. Many mesh configurations were run in order to find the appropriate number of elements to represent the boundary layer. Suggestions are given for obtaining a good engineering estimate of the local shear stress (aeroelastic applications) and overall drag (aerodynamic applications). The next three cases are flow visualizations compared with van Dyke (1982). A forward and backward facing step at a Reynolds number of 1 are shown. Then the two cases are combined into a rectangular block setting on a flat plate with a Reynolds number of 1. These four cases will show that the code is solving simplified viscous geometries as designed.

The next three groups will show that the solver is capable of modeling the flow over immersed bodies. An NACA airfoil at a relatively high Reynolds number is used to illustrate flow over a streamlined body. The effects of dissipation on the airfoil boundary layer are also discussed along with "induced vortex shedding" when the dissipation is decreased. A 6:1 ellipse is then used to illustrate flow over a "bluff" body. A vortex street is shown behind the ellipse, and dissipation induced shedding is again illustrated. The next group of immersed bodies is circular cylinders. The Reynolds number for the cylinders ranges from 1.54 to 200. A vortex street is illustrated for the two highest Reynolds numbers.

To conclude this section, one problematic case is illustrated. The case is a 2% thick flat plate held at an angle of attack of 2.5 degrees. The plate illustrates leading edge

separation on thin airfoils. The test case shows significant dependence upon artificial dissipation and requires the extra dissipation to retain a stable solution.

5.1.1. Laminar (Blasius) Profile

Blasius' solution to the laminar boundary layer problem is a classical ordinary differential equation (ODE) approximation of the Navier-Stokes equations. Other more simplified, analytical solutions to the Navier-Stokes equations exist, but these solutions require the solver to model attributes such as given downstream pressures or wall motion, which the solver does not currently utilize. These features can be added to the solver at a later date, and cases such as Hagen-Poiselle and Couette flow can be used as verification tests for those features. For now, the laminar boundary layer is the most simplified form of the governing equations.

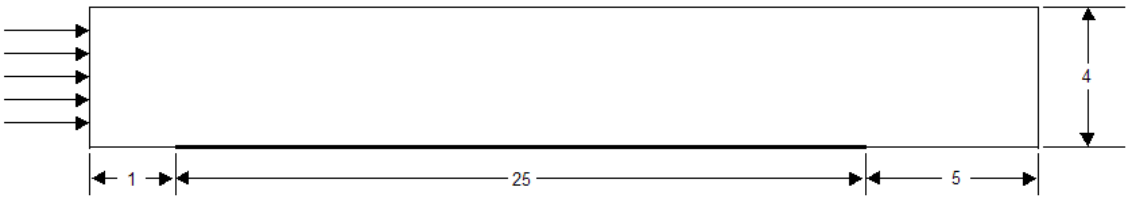

Figure 5-1: Layout of Blasius Boundary Layer Test Case

The test case geometry is given in Figure 5-1 above along with the flow properties and solver controls in Table 5-1 below. The freestream Reynolds number is 3600 for the test case, and the plate is 25 units long. Therefore, the Reynolds number for the entire plate is 90000, as specified previously. The freestream Mach number is 0.3, and the

highest local Mach number in the flow was 0.31, found over the thickest part of the boundary layer. Given the maximum Mach number, the flow can be considered incompressible and can be modeled using Blasius' solution.

Table 5-1: Boundary Layer Test Case Parameters

Gamma	1.4
Dissipation	0.0 - 1.0
CFL	0.50
Visc. Ratio	-0.66
Mod. Suther.	0.0
Mach Num.	0.3
Reynolds Num.	3600
Prandtl Num.	0.7
Angle of Attack	0.0
Ref. Dimen.	1.0
Num. of Steps	40000
Cycles per Step	4
Solution Type	Steady
Diss. Type	Grad. Limiters
Integration Type	1 Point

Blasius' solution can be found by using Equations 5.1 through 5.3 below (Bertin, 1998). The profile ends with 99% of the external velocity given at η = 3.5. This condition also ensures that the velocity at η = 5.0 can be approximated as external, or freestream. Blasius' solution gives a boundary layer thickness (to 99% of external velocity) represented by Equation 5.4.

(5.1) $$f''' + ff'' = 0$$

(5.2) $$f = f(\eta) = \frac{u}{U_e}$$

(5.3) $$\eta = y\sqrt{\frac{U_e}{2\nu x}}$$

(5.4) $$\delta = x\frac{5.0}{\sqrt{\mathrm{Re}_x}}$$

This approximation was used to define several parameters for the mesh spacing. The meshing software used with this research uses "sources" (i.e., points, lines, and triangles) to define the areas of concentrated meshes. Each source is defined by points: One in the case of the point source, two for lines, and three for triangles. Each source also contains three control values: Spacing, constant radius, and radius to double. The constant radius, or inner radius, is the radius about the source in which the spacing is constant. The radius to double, or outer radius, is the radius where the spacing doubles. From these two radii, an exponential curve is fit to define the spacing at any radius away from the source. The meshing routine looks at all of the sources given and the background mesh spacing to define the local spacing. Using the local spacing and the marching front method, triangle elements are placed throughout the flow field.

For this test case, the boundary layer was defined by three line sources. The first stretches from the leading edge of the plate to 3 reference lengths downstream; the second from 3 to 12 reference lengths; and, the final from 12 to 32 reference lengths. The additional length on the final line source was used to capture the wake behind the plate. Each source was placed direction along the plate. The first radius of the source was used to define the curvature of the boundary layer. Figure 5-2 shows that the greatest curvature in the boundary layer occurs between η = 1.0 and 3.5. Therefore, the first radius is chosen to cover 50% of the boundary layer height, given by Equation 5.4. The second radius is then chosen to be 2.5 times the first. Using the exponential decay of the spacing, enough elements are distributed across the boundary layer to represent the curvature.

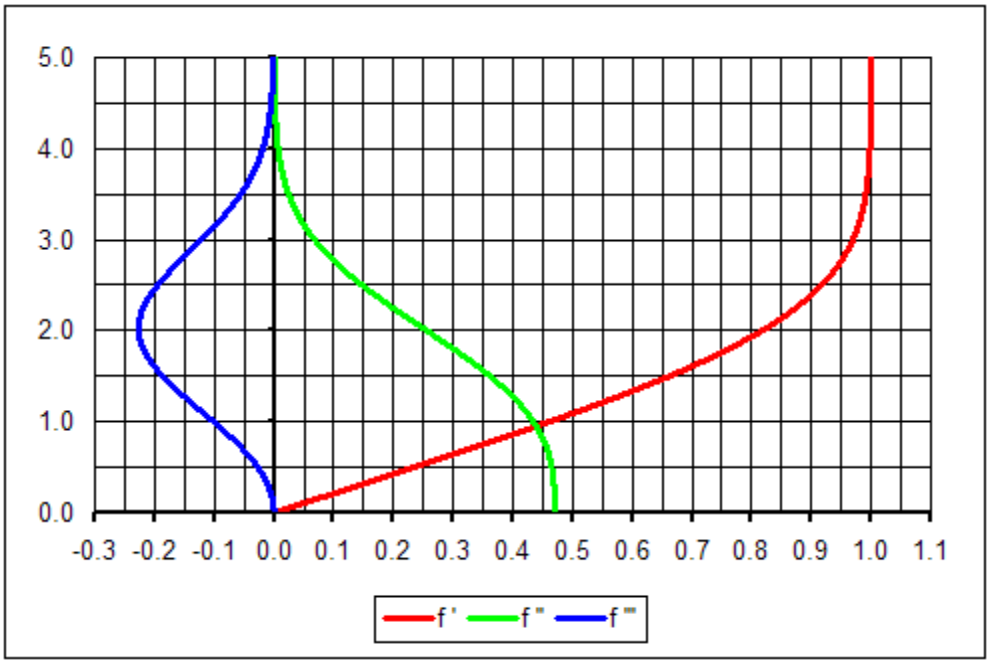

Figure 5-2: First Three Derivatives of Blasius Function *f*

Finally, the spacing must be defined. The nominal spacing is defined as one-sixth of the radius at the four locations. Because the boundary layer thickness is zero at the leading edge, a fictitious location was used to find an appropriate radius. This fictitious location was defined as 0.5 reference lengths downstream; the boundary layer thickness there was equated to the radius at the leading edge. Test cases were then constructed as percentages of the nominal case spacing. These percentages could then be tested to find the appropriate spacing to represent the boundary layer. The nominal case is given in Table 5-2 below; the other cases are specified in Appendix D. The nominal case was found to have 10 elements spanning the boundary layer. The finest test case ran (60%) was found to have approximately 16 $^{2}/_{3}$ elements while the most coarse case had only 1 ¼ elements to represent the boundary layer. An example of these three cases along with the 160% case are illustrated in Figures 5-3 through 5-6.

Table 5-2: Nominal Boundary Layer Test Case

100% Spacing Test Case					
x-Location	Fict. Local	Thickness	In. Rad.	Out. Rad.	Spacing
0.0	0.50	0.0589	0.0295	0.0737	0.0049
3.0	3.00	0.1443	0.0722	0.1804	0.012
12.0	12.00	0.2887	0.1443	0.3608	0.0241
32.0	32.00	0.4714	0.2357	0.5893	0.0393

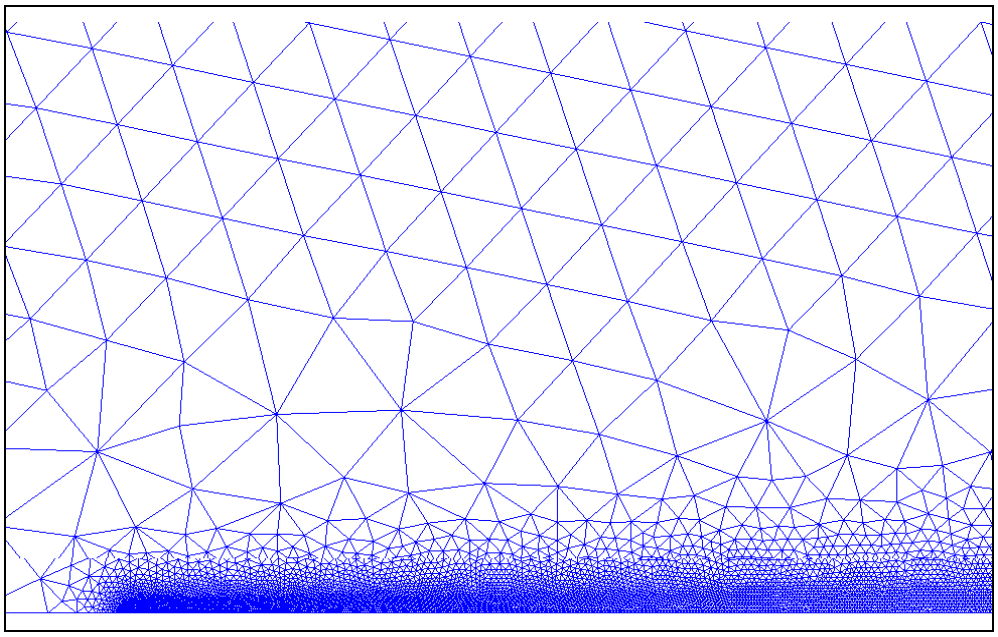

Figure 5-3: Nominal 100% Boundary Layer Cases

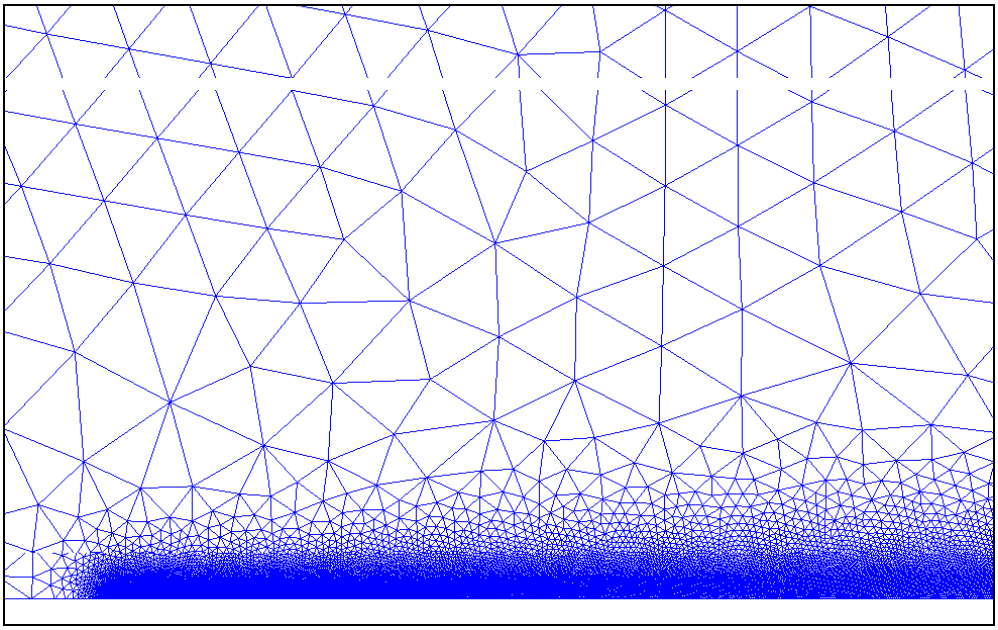

Figure 5-4: Finest 60% Boundary Layer Case

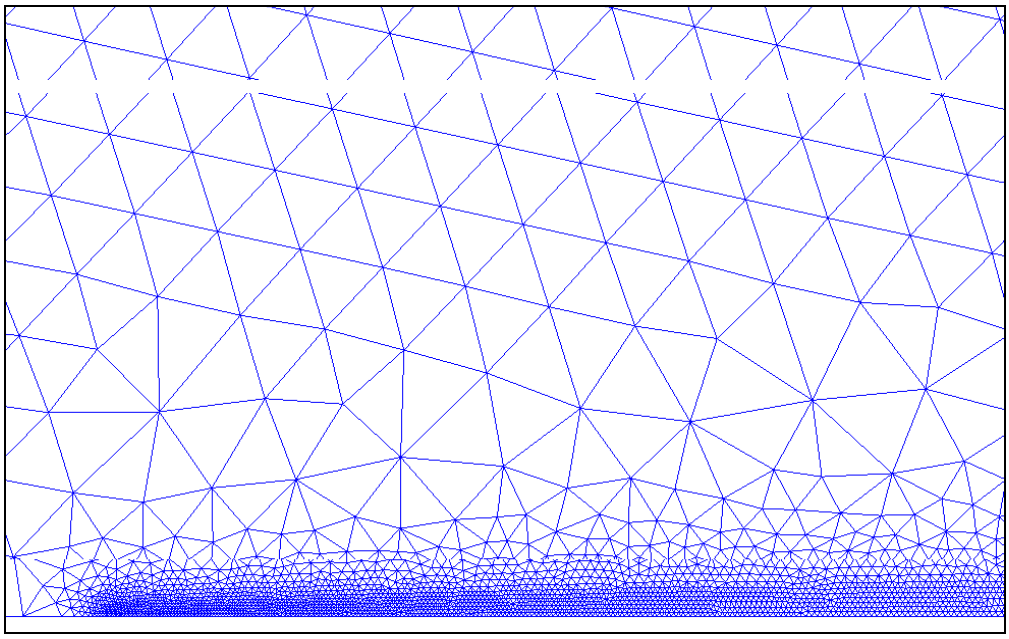

Figure 5-5: Medium 160% Boundary Layer Case

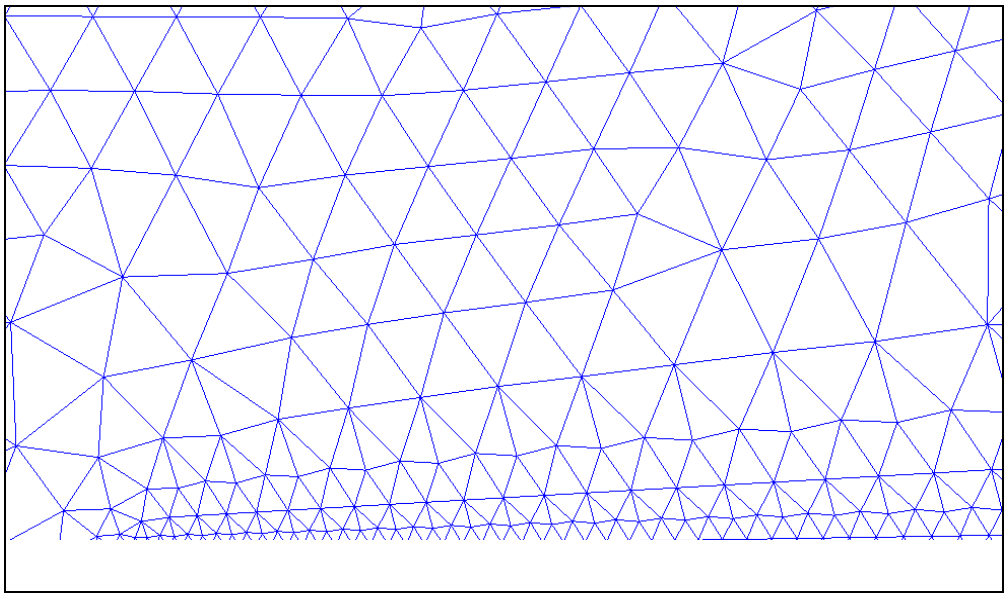

Figure 5-6: Most Coarse 800% Boundary Layer Case

Each case was swept through dissipations from 1.0 to 0.0 by steps of 0.1. The same trend was found to occur in all of the cases: Artificial dissipation decreases the curvature in the boundary layer. This decrease in curvature is shown in Figurs 5-8 and 5-9. Figure 5-8 shows the boundary layer with a dissipation of 1.0; Figure 5-9 shows no dissipation. With no dissipation, the laminar boundary layer profile was found to match Blasius' solution for all finer meshes. As the mesh spacing increased, the velocity profile overshoots at the point of greatest curvature. Figure 5-10 illustrates this slight overshoot. Yet, even as the elements grow large, the slope at the base of the profile is relatively intact (see Figure 5-11). The spread on the points increases as the elements grow. This scatter around the solution can only be expected; larger elements mean larger jumps in the linear approximations. The overshoots at the point of greatest curvature can be seen in the 160% mesh. Therefore, the 150% mesh represents the dynamics of the boundary layer within engineering approximations; although, for better accuracy, elements can be added. The 150% mesh has only been tested on the flat plate, so engineering judgment must be used when generating other meshes.

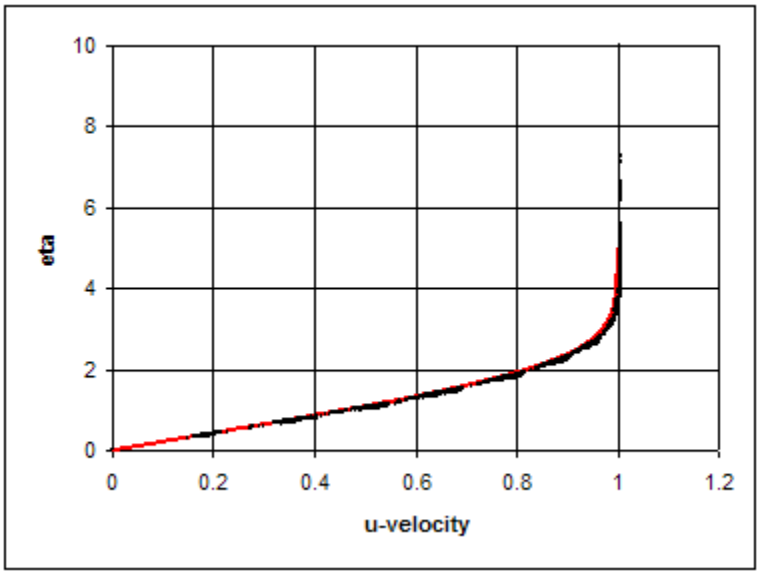

Figure 5-7: 150% Mesh Velocity Profile (*DISS* = 0.0)

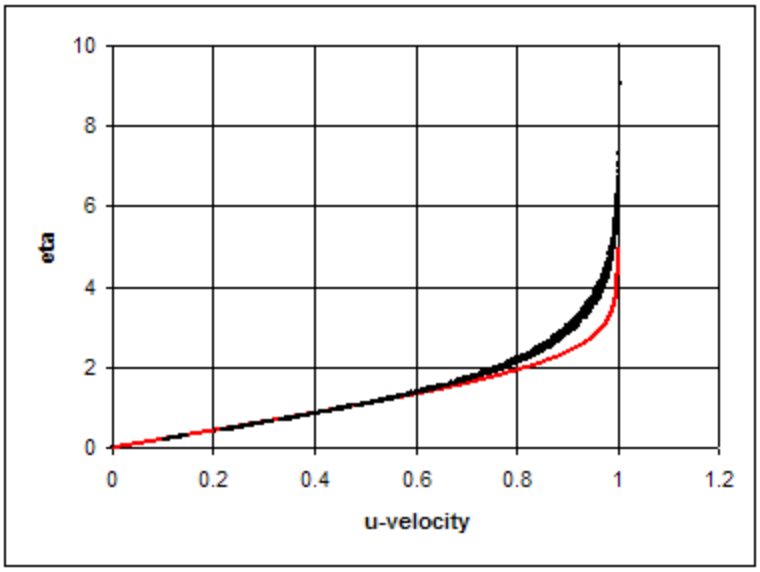

Figure 5-8: Nominal Mesh with Dissipation of 1.0

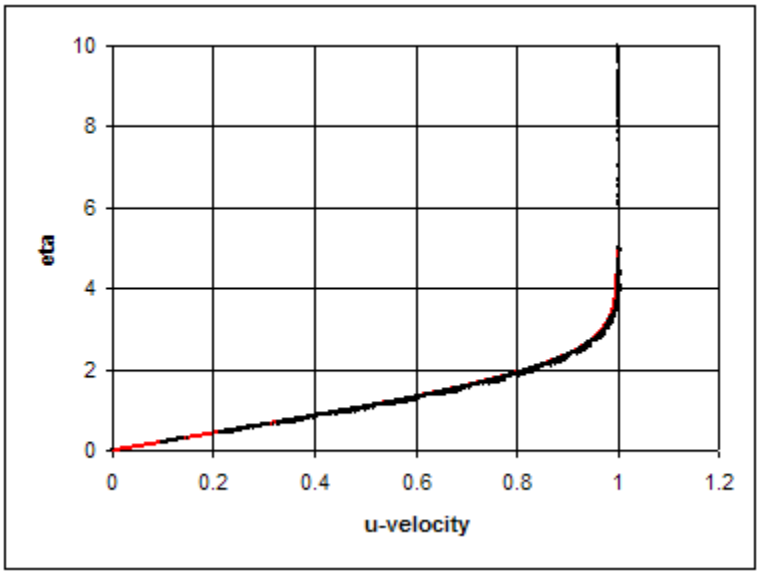

Figure 5-9: Nominal Mesh with Dissipation of 0.0

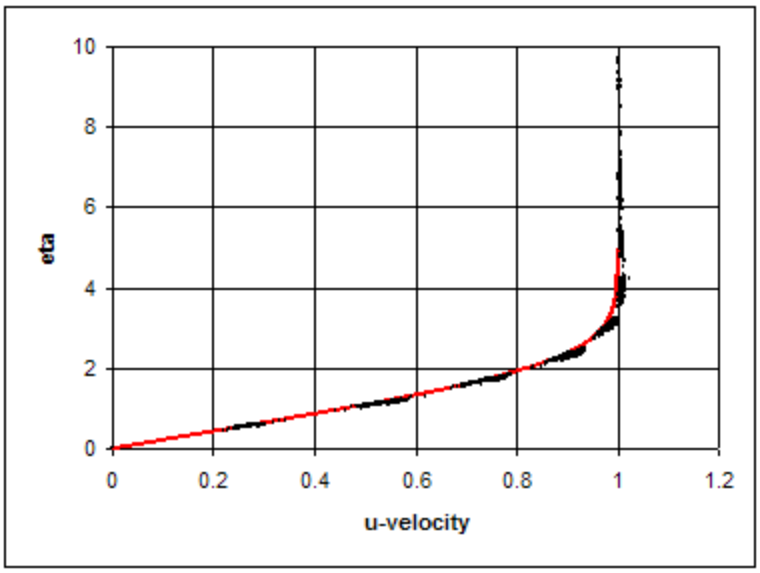

Figure 5-10: 240% Mesh – Overshoot in Velocity Profile (*DISS* = 0.0)

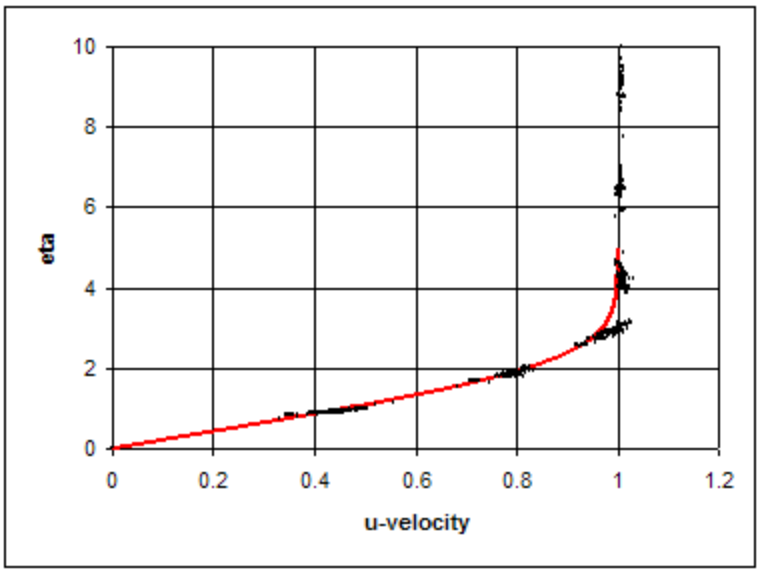

Figure 5-11: 400% Mesh – Scattering in Velocity Profile (*DISS* = 0.0)

Since the lower 20% of the boundary layer stays intact even for large element sizes, the shear stress is also expected to show good results. The shear stress for the nominal (100%), 240%, and 400% cases are shown in Figures 5-12 through 5-15; Figures 5-12 and 5-13 illustrate the effects of artificial dissipation on shear stress.

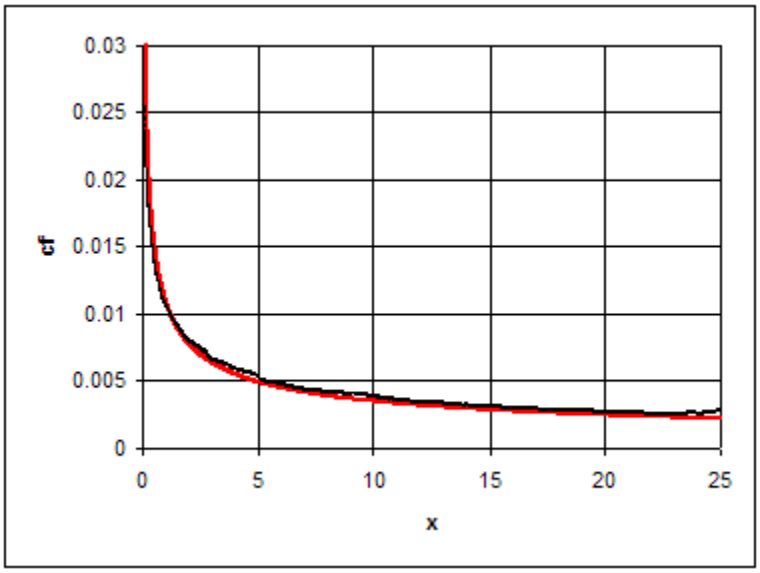

Figure 5-12: Nominal Mesh Shear Stress – Dissipation of 1.0

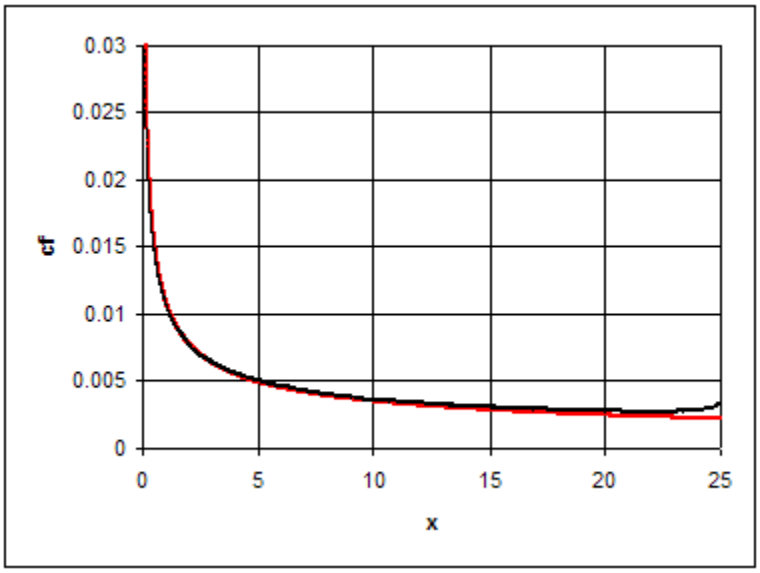

Figure 5-13: Nominal Mesh Shear Stress – Dissipation of 0.0

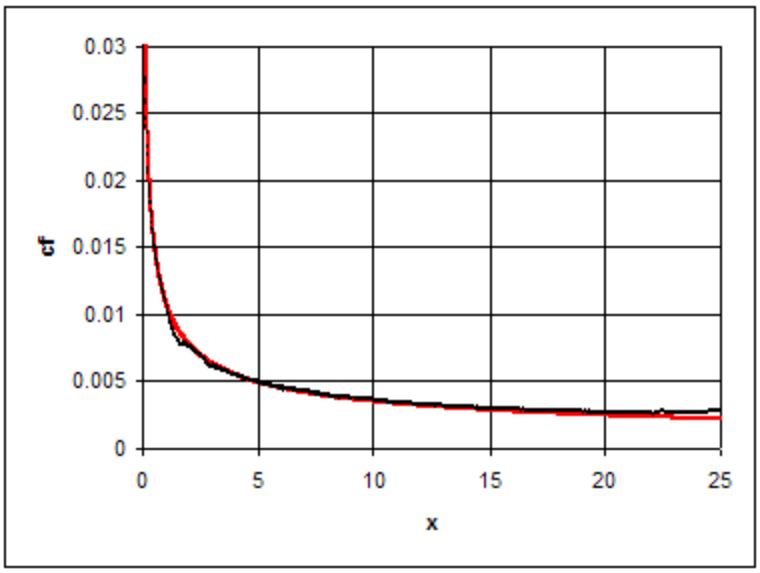

Figure 5-14: 240% Mesh – Shear Stress Distribution along Plate (*DISS* = 0.0)

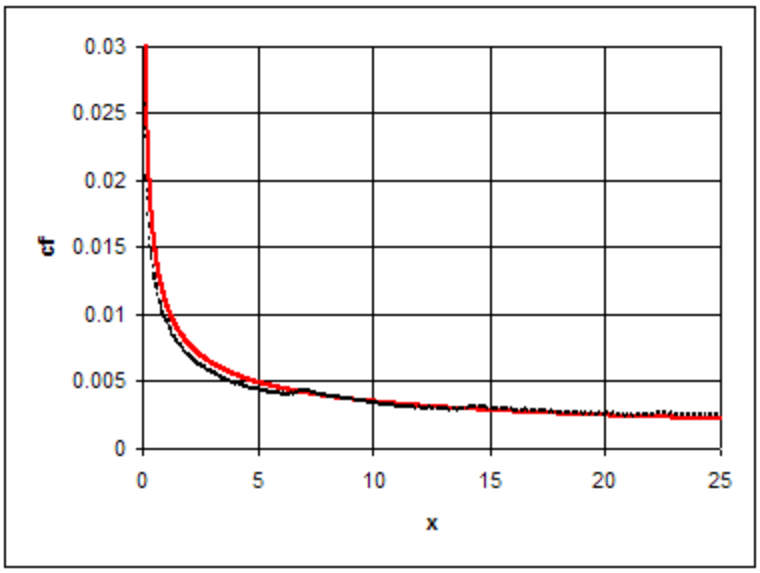

Figure 5-15: 400% Mesh – Shear Stress Distribution along Plate (*DISS* = 0.0)

In Figures 5-12 through 5-14, the shear stress increases at the end of the plate (i.e., x = 25). The increase is due to the exit plane. Figure 5-16 shows the Mach number profile from near the end of the plate to the exit plane. The velocities near the exit plane decrease to return to the freestream condition due to the far field boundary.

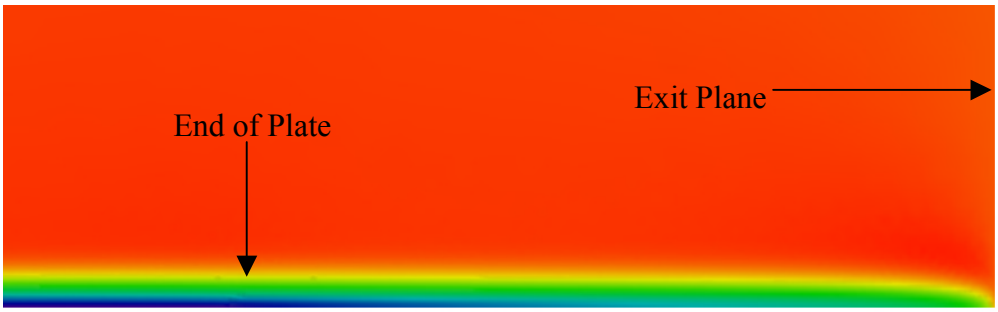

Figure 5-16: Mach Number Profile Near the Exit Plane of Nominal Mesh

Small decreases in the local shear stress near the leading edge of the plate can be seen in Figures 5-14 and 5-15. The leading edge differences in the 175% mesh and larger. Remembering again that these tests were only run for a flat plate, the 170% mesh case is considered to represent the local shear stress well within engineering estimates.

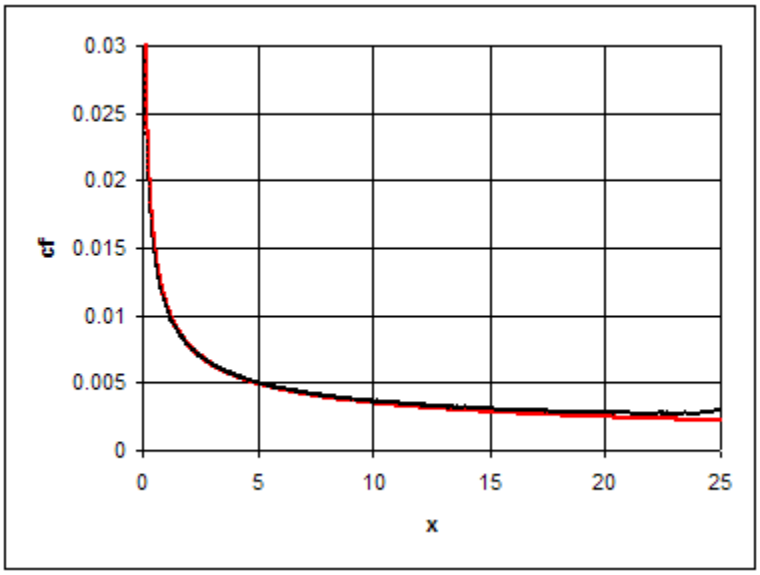

Figure 5-17: 170% Mesh Shear Stress (*DISS* = 0.0)

5.1.2. Forward and Backward Facing Steps

The forward and backward facing steps have been generated at freestream Reynolds numbers of 1. The geometry for both cases is shown in Figure 5-18; for the backward step, the flow comes in a 9 unit high section and leaves out the 10 unit high section from left to right. The control parameters for both cases are given in Table 5-3.

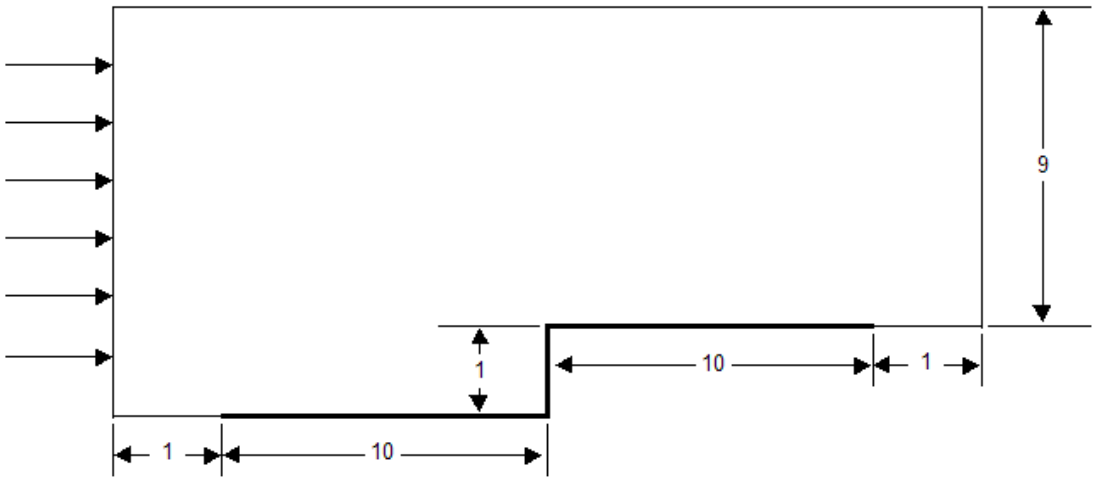

Figure 5-18: Layout of Forward Facing Step Test Case

In order to best illustrate the flow over the backward and forward steps, velocity vectors have been plotted for both cases in Figure 5-19 and 5-21, respectively. These velocity vector fields can be compared to the flow visualization from van Dyke (1982), given in Figure 5-20. The vector fields were generated with a constant spacing of 0.1 over the entire flow field. The actual grids for the forward and backward facing steps can be seen in Figures 5-22 and 5-23.

Table 5-3: Forward (Left) and Backward (Right) Step Control Parameters

Forward Step	
Gamma	1.4
Dissipation	1.0
Dim. Time	2.3E-05
CFL	0.50
Visc. Ratio	-0.66
Mod. Suther.	0.0
Mach Num.	0.3
Reynolds Num.	1.0
Prandtl Num.	0.7
Angle of Attack	0.0
Ref. Dimen.	1.0
Ref. Density	1.0
Ref. Accoustic. Speed	1.0
Num. of Steps	94000
Cycles per Step	15
Solution Type	Unsteady
Diss. Type	Grad. Limiters
Integration Type	1 Point

Backward Step	
Gamma	1.4
Dissipation	1.0
Dim. Time	2.0E-05
CFL	0.50
Visc. Ratio	-0.66
Mod. Suther.	0.0
Mach Num.	0.3
Reynolds Num.	1.0
Prandtl Num.	0.7
Angle of Attack	0.0
Ref. Dimen.	1.0
Ref. Density	1.0
Ref. Accoustic. Speed	1.0
Num. of Steps	122000
Cycles per Step	25
Solution Type	Unsteady
Diss. Type	Grad. Limiters
Integration Type	1 Point

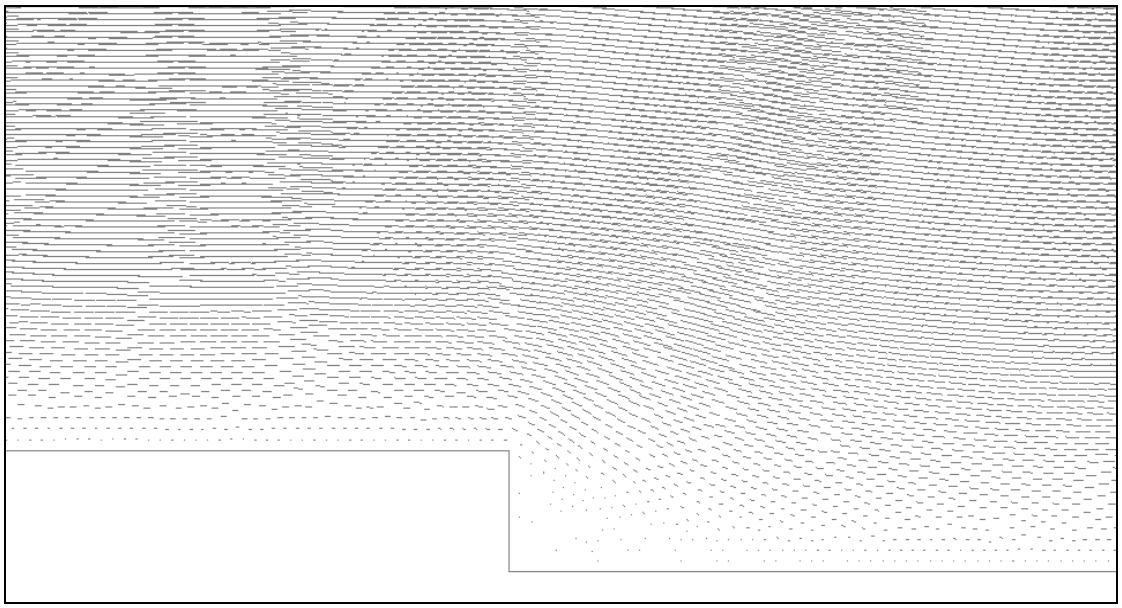

Figure 5-19: Backward Facing Step Velocity Vectors

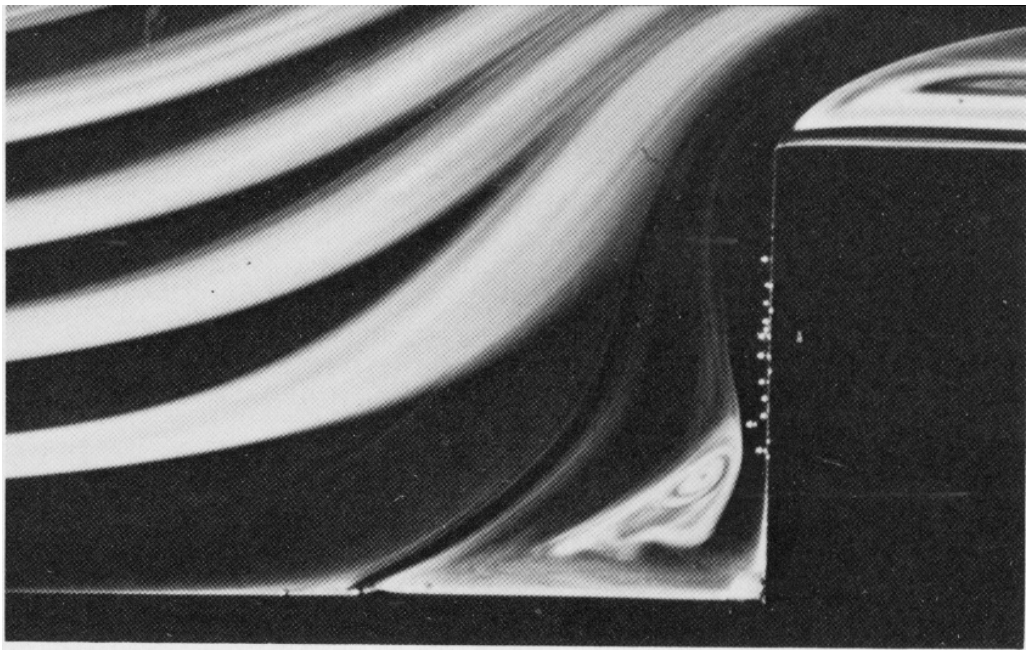

Figure 5-20: Flow over a Forward/Backward Facing Step (van Dyke, 1982)

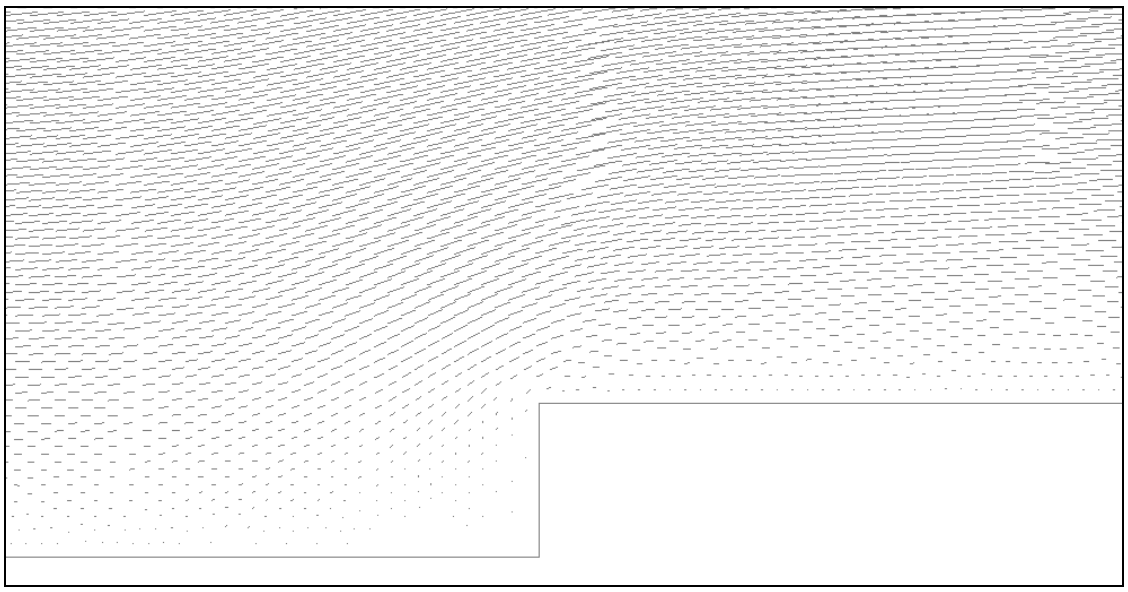

Figure 5-21: Forward Facing Step Velocity Vectors

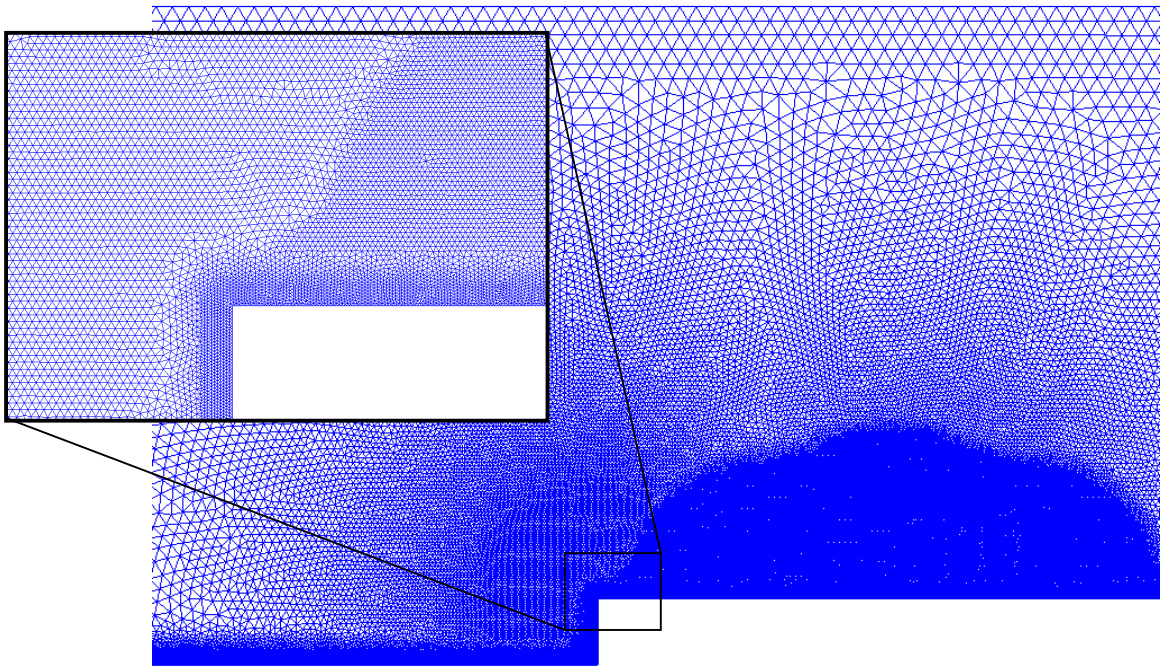

Figure 5-22: Unstructured Mesh for Forward Facing Step

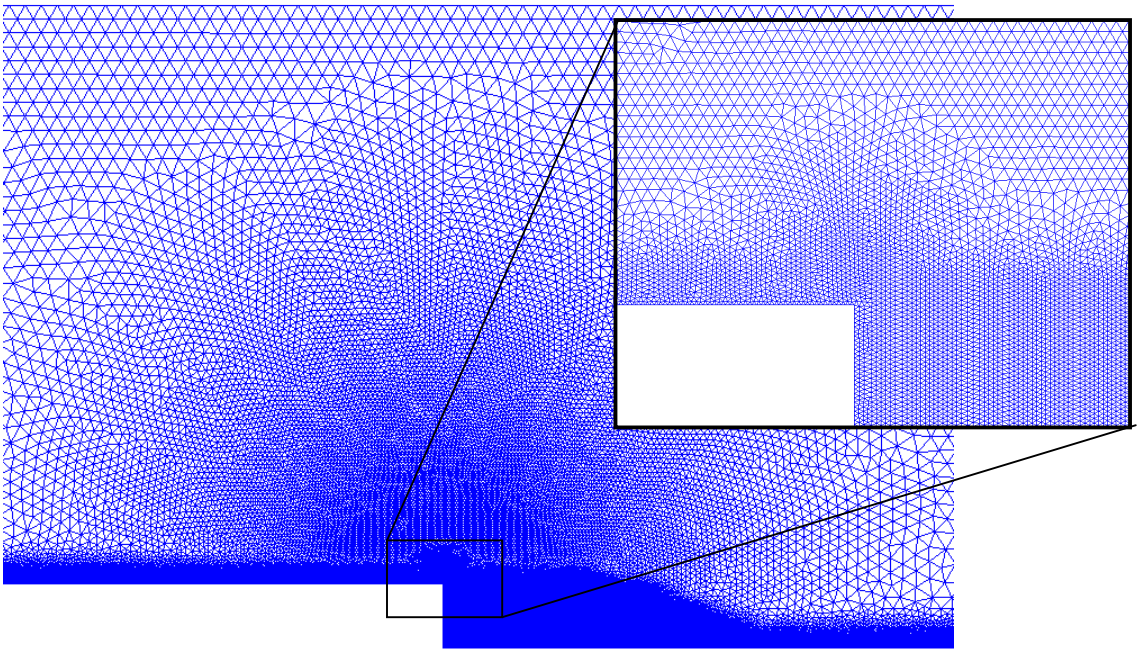

Figure 5-23: Unstructured Mesh for Backward Facing Step

5.1.3. Rectangular Block

Combining the forward and backward facing steps, the next cases is a rectangular block setting on a flat viscous surface, illustrated in Figure 5-24. The block measures one unit in height and length, as the previous steps. Similarly a vector plot has been generated in comparison with a flow visualization from van Dyke (1982). The flow visual and vector plot are given in Figures 5-26 and 5-27, respectively. Further, the mesh for this test case is given in Figure 5-25.

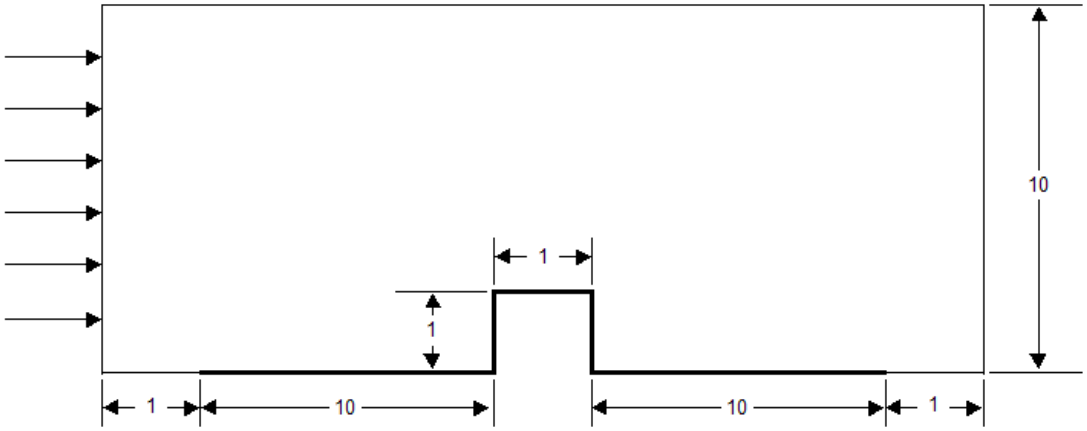

Figure 5-24: Layout of Rectangular Block Test Case

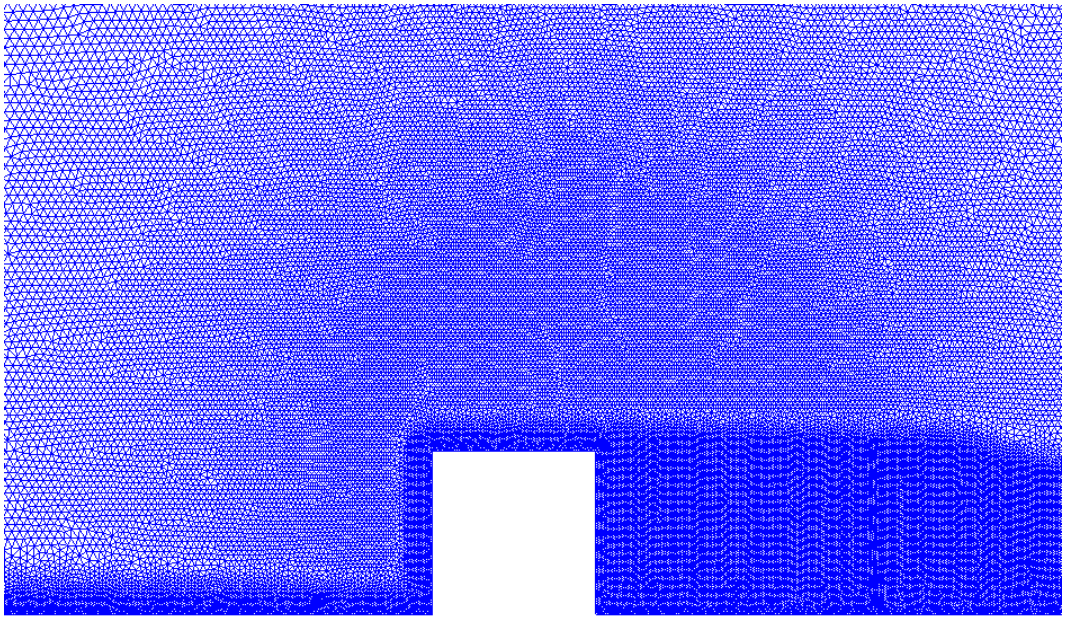

Figure 5-25: Unstructured Mesh for Rectangular Block Case

Table 5-4: Control Parameters for Rectangular Block Case

Gamma	1.40
Dissipation	1.0 & 0.3
Dim. Time	2.0E-05
CFL	0.5
Visc. Ratio	-0.66
Mod. Suther.	0.0
Mach Num.	0.3
Reynolds Num.	1.0
Prandtl Num.	0.7
Angle of Attack	0.0
Ref. Dimen.	2.0
Ref. Density	1.0
Ref. Accoustic Speed	1.0
Num. of Steps	122000
Cycles per Step	25
Solution Type	Unsteady
Diss. Type	Grad. Limiters
Integration Type	1 point

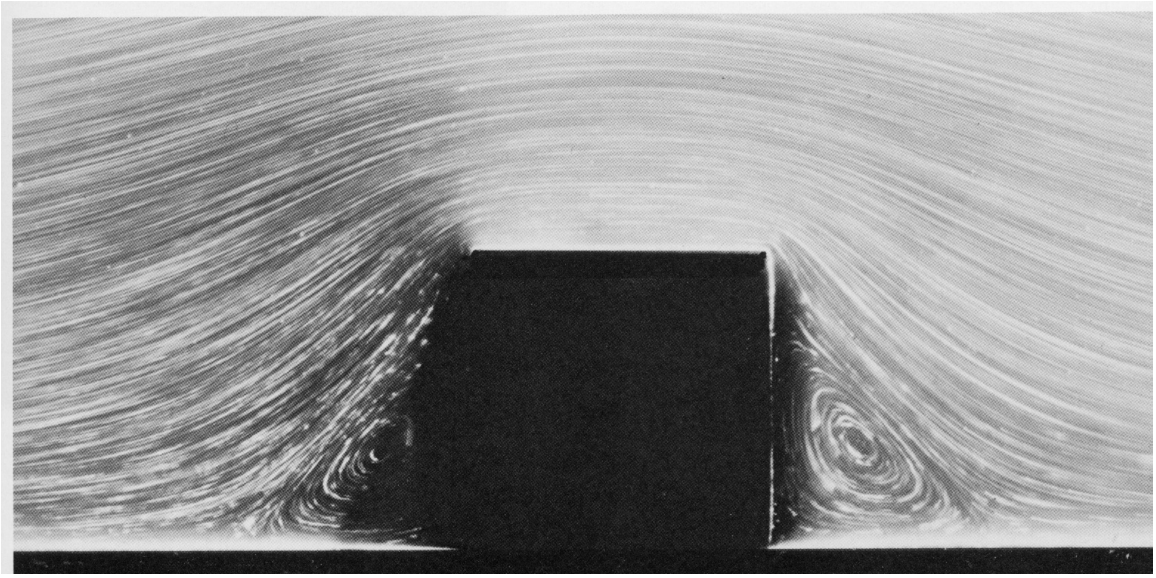

Figure 5-26: Rectangular Block on a Flat Plate, Re = 0.02 (van Dyke, 1982)

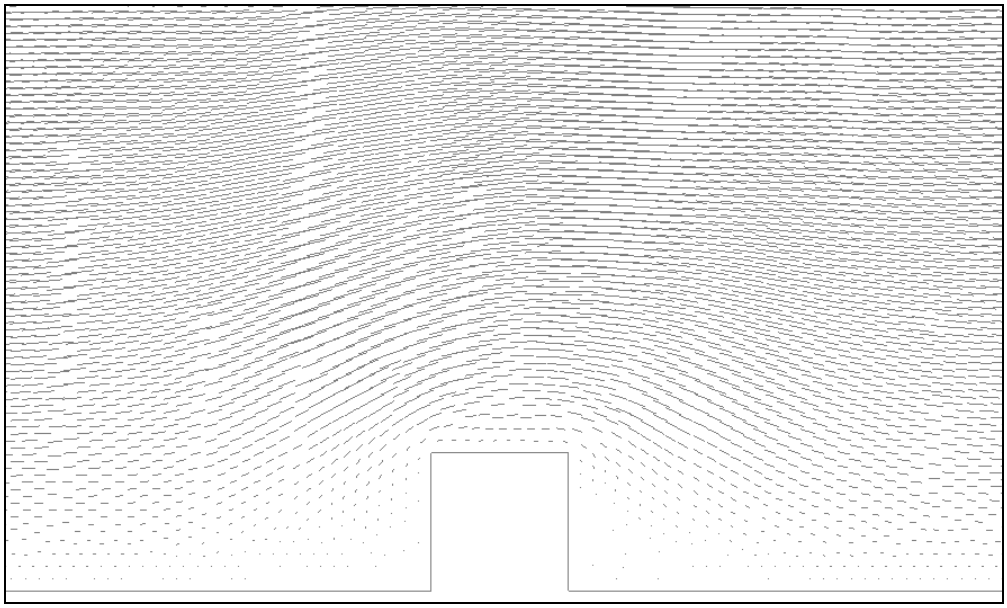

Figure 5-27: Rectangular Block Vector Plot

5.1.4. NACA 0012

The next case illustrates that NS2d can handle a streamlined body in the flow field. An NACA 0012 is generated using the geometry of Figure 5-28. The mesh was generated using the 150% rule of Section 5.1.1. That mesh is shown in Figure 5-29

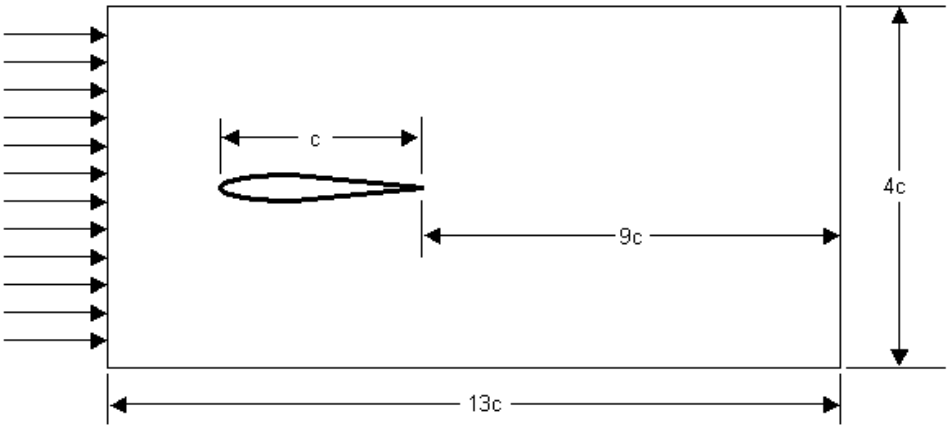

Figure 5-28: Layout of Airfoil Test Cases

Table 5-5: Control Parameters for NACA 0012 Airfoil

Gamma	1.40
Dissipation	1.0 & 0.3
Dim. Time	2.5E-04
CFL	0.5
Visc. Ratio	-0.66
Mod. Suther.	0.0
Mach Num.	0.3
Reynolds Num.	3.0E+06
Prandtl Num.	0.7
Angle of Attack	0.0
Ref. Dimen.	2.0
Ref. Density	1.0
Ref. Accoustic Speed	1.0
Num. of Steps	40000
Cycles per Step	25
Solution Type	Unsteady
Diss. Type	Grad. Limiters
Integration Type	1 point

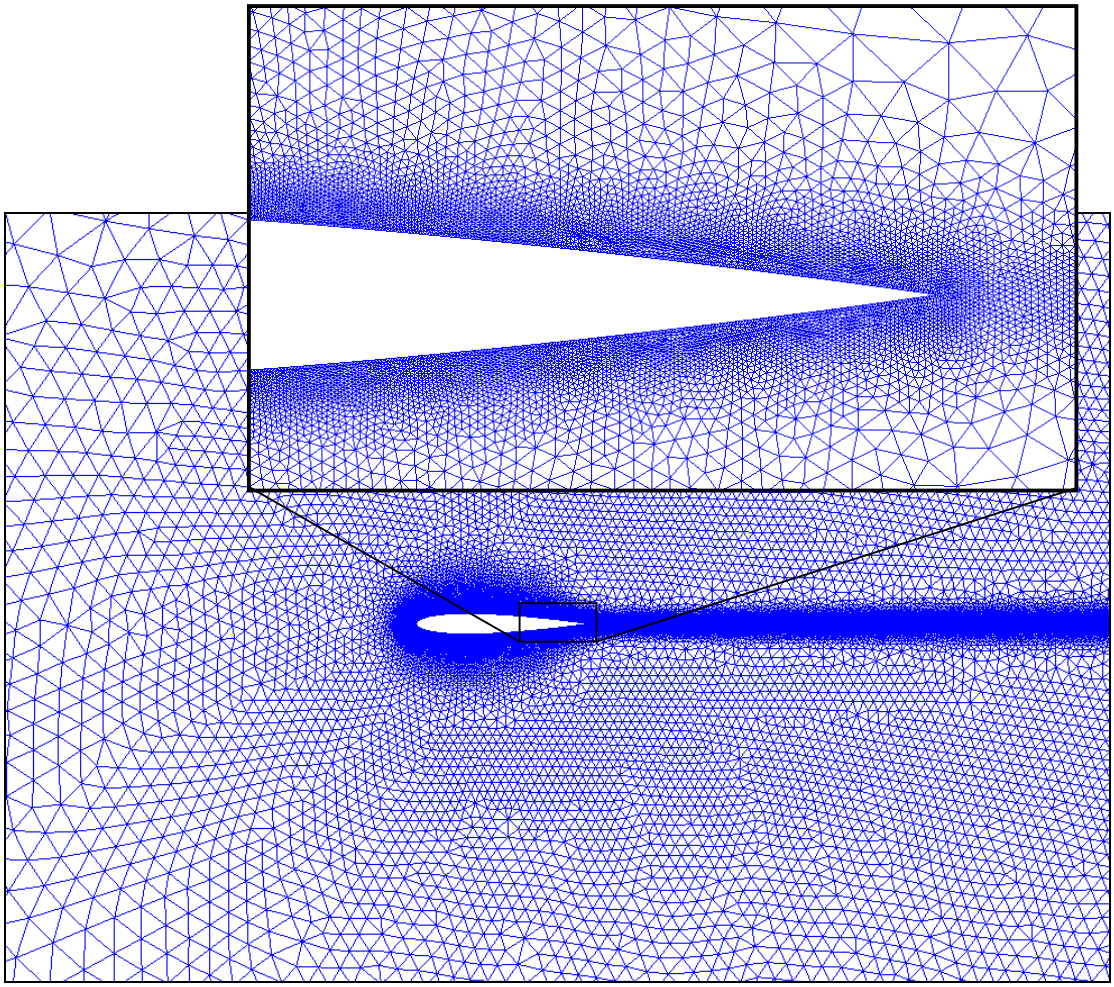

Figure 5-29: NACA 0012 Airfoil Mesh

The solution to the NACA 0012 case is given in Figure 5-30. Figure 5-30 ranges from a Mach number of 0.36 to zero on the airfoil surface. The solution shows a well defined boundary layer, slight separation near the trailing edge, and a distinct wake area. The solution shows problems capturing the boundary layer on the leading edge because of limitations on mesh spacing.

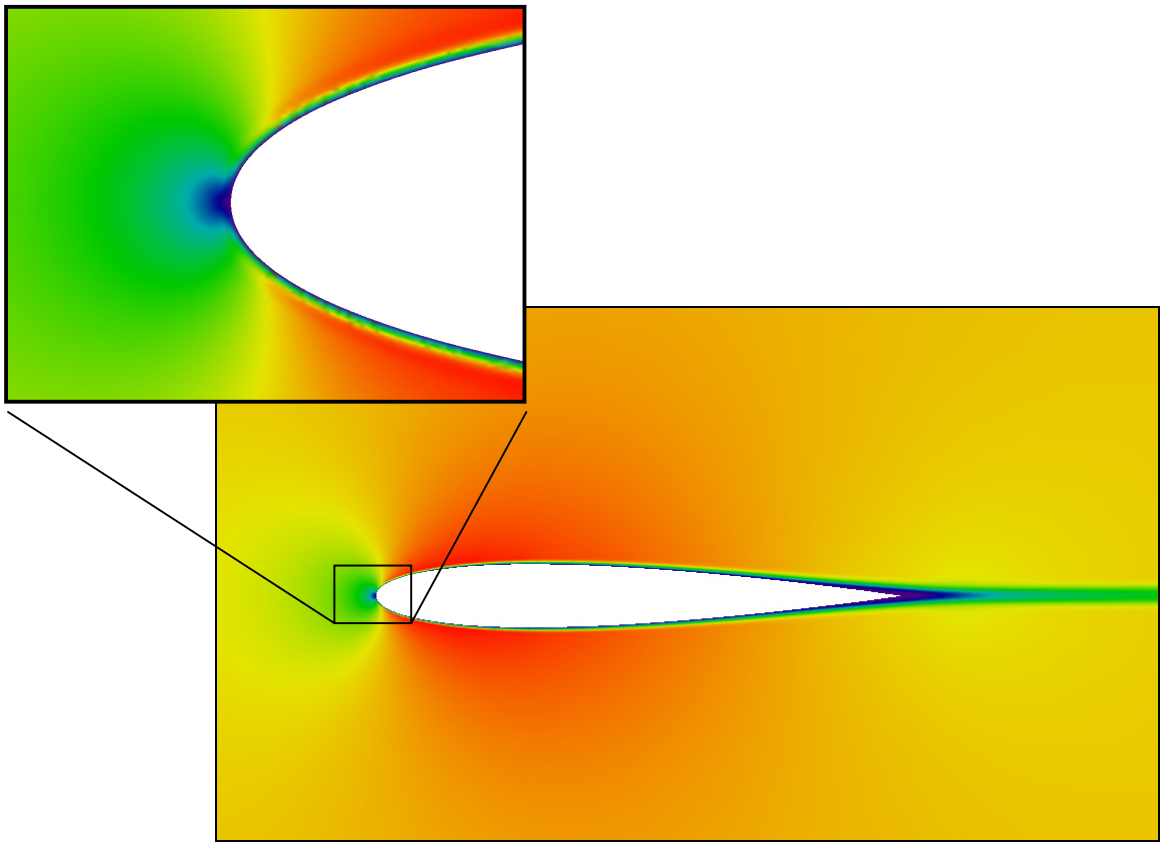

Figure 5-30: Mach Number Distribution over NACA 0012 Airfoil (*DISS* = 1.0)

Two different dissipation levels were tested with the NACA 0012 airfoil. The first is shown in Figure 5-30. A dissipation of 1.0 was used to converge the solution; then the dissipation was decreased to 0.3. Because the artificial dissipation acts as viscosity in the solution, the extra "viscosity" had to be shed into the flow. The shedding can be seen in the entropy plots of Figures 5-31 through 5-33.

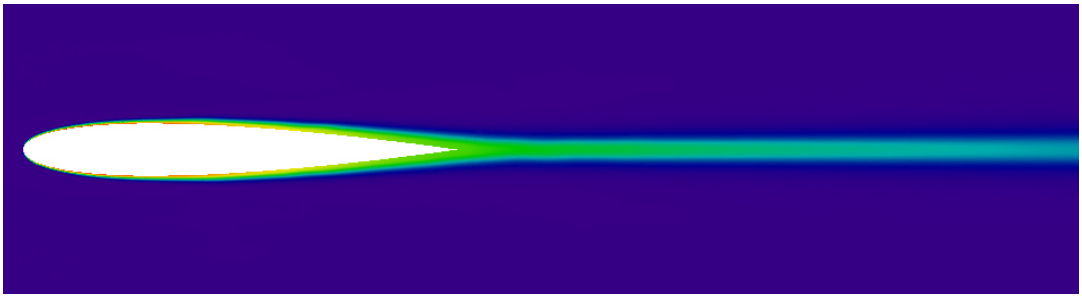

Figure 5-31: Entropy for NACA 0012 Airfoil with *DISS* = 1.0 and *RE* = 3.0x10^6

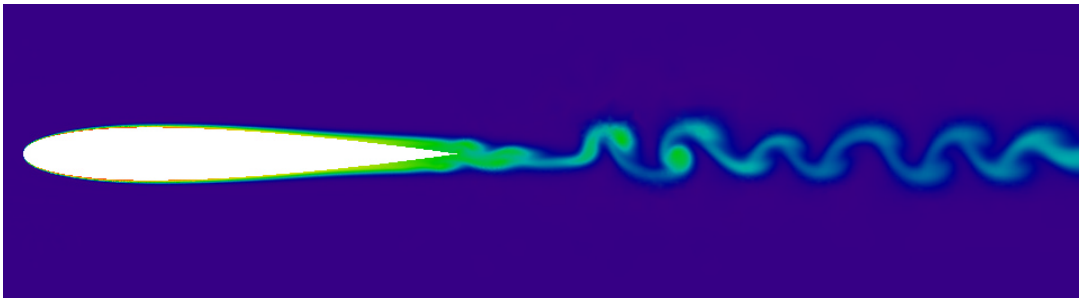

Figure 5-32: Entropy for NACA Airfoil 1.5 *dt* After Dissipation Lowered

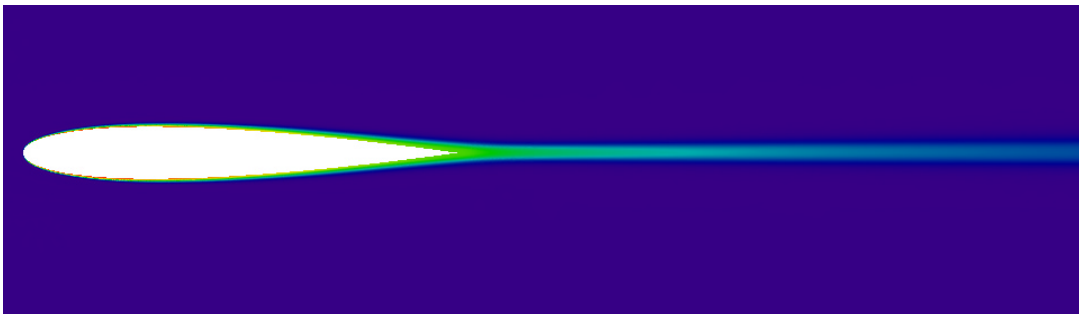

Figure 5-33: Entropy for NACA Airfoil with *DISS* = 0.3 and *RE* = 3.0x10^6

Not only can the shed vortices be seen in the intermediate picture, but the wake and boundary layer are clearly reduced due to the decrease in "viscosity". The shed vorticity goes to collaborate the effects of dissipation seen in the boundary layer studies.

This artificial “viscosity” acts as viscosity to the flow physics while limiting the gradients in the flow, even in the boundary layer itself.

5.1.5. Thin Ellipse

A thin ellipse of ratio 6 to 1 is used to generate a more complicated immersed body. The ellipse is similar to the NACA 0012 along the leading edge and mid-surfaces. The trailing edge is more like the circular cylinder. The ellipse leaves behind an oscillating wake because of the bluntness and the high Reynolds number of the test case. The geometry and a sample of the mesh are given in Figures 5-34 and 5-35, respectively.

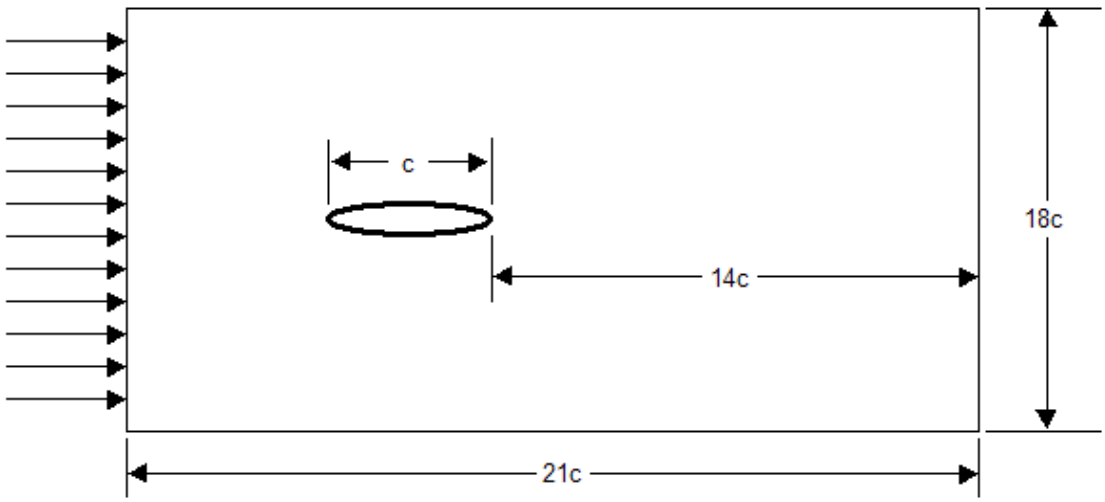

Figure 5-34: Layout of 6:1 Ellipse Test Case

Table 5-6: Control Parameters for 6:1 Ellipse

Gamma	1.40
Dissipation	1.0 & 0.6
Dim. Time	1.0E-03
CFL	0.5
Visc. Ratio	-0.66
Mod. Suther.	0.0
Mach Num.	0.3
Reynolds Num.	4000
Prandtl Num.	0.7
Angle of Attack	0.0
Ref. Dimen.	1.0
Ref. Density	1.0
Ref. Accoustic Speed	1.0
Num. of Steps	80000
Cycles per Step	20
Solution Type	Unsteady
Diss. Type	Grad. Limiters
Integration Type	1 point

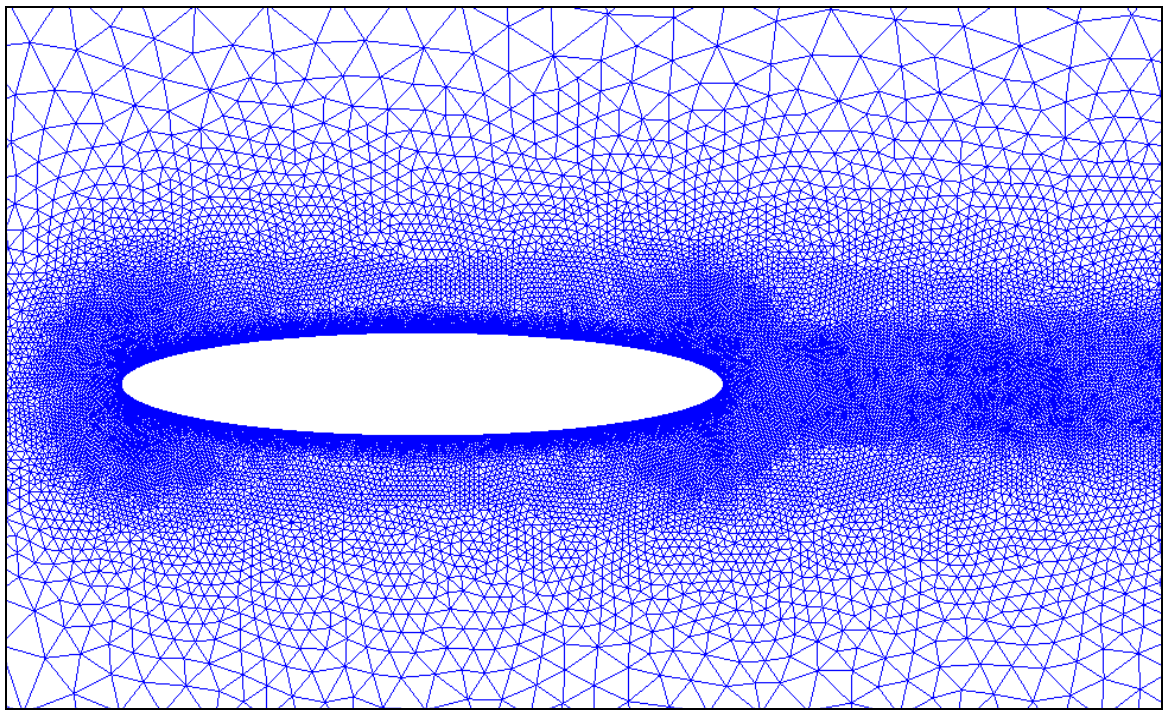

Figure 5-35: Unstructured Mesh for 6:1 Ellipse Case

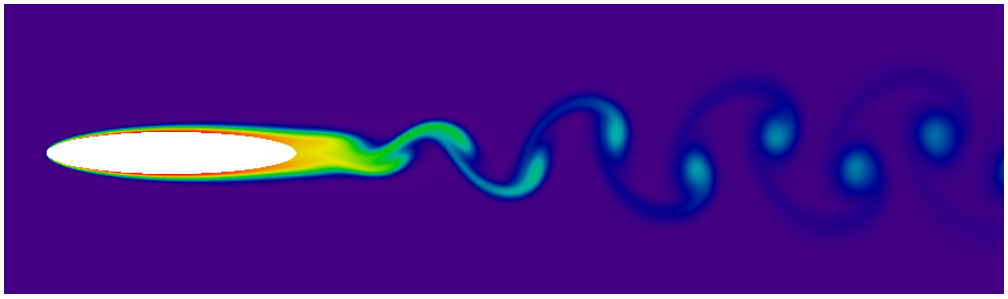

Figure 5-36: Oscillating Wake Behind 6:1 Ellipse for *DISS* = 0.6 and *RE* = 4000

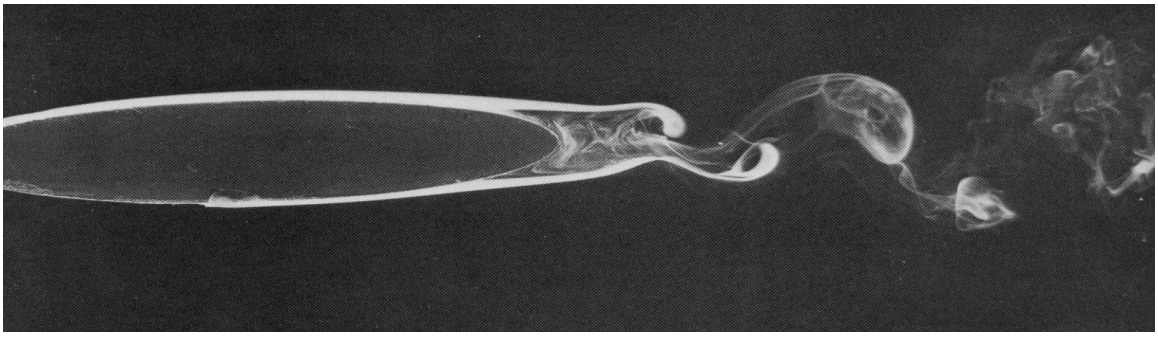

Figure 5-37: Flow Visualization of 6:1 Ellipse, Re = 4000 (van Dyke, 1982)

Figure 5-36 is an entropy plot, revealing the oscillating wake behind the ellipse. The Strouhal number for this particular ellipse was found by *NS2d* to be 4.8. This picture can be compared to that of Figure 5-37, again from van Dyke (1982). The ellipse was simulated at two different dissipation factors: 1.0 and 0.6. Using the previous pattern of behavior, reducing the dissipation should reduce the "viscosity" of the flow over the ellipse. The lift coefficient is shown in Figure 5-38 to oscillate as the ellipse shed vortices. This oscillation changes in amplitude by a factor of 68% but not in frequency for a reduction in dissipation. Although the lift coefficient oscillates, the drag coefficient

is constant at 0.0522 for *diss* = 1.0 and 0.0538 for *diss* = 0.6. Therefore, the drag is increased by 3% by decreasing the artificial dissipation of the solver.

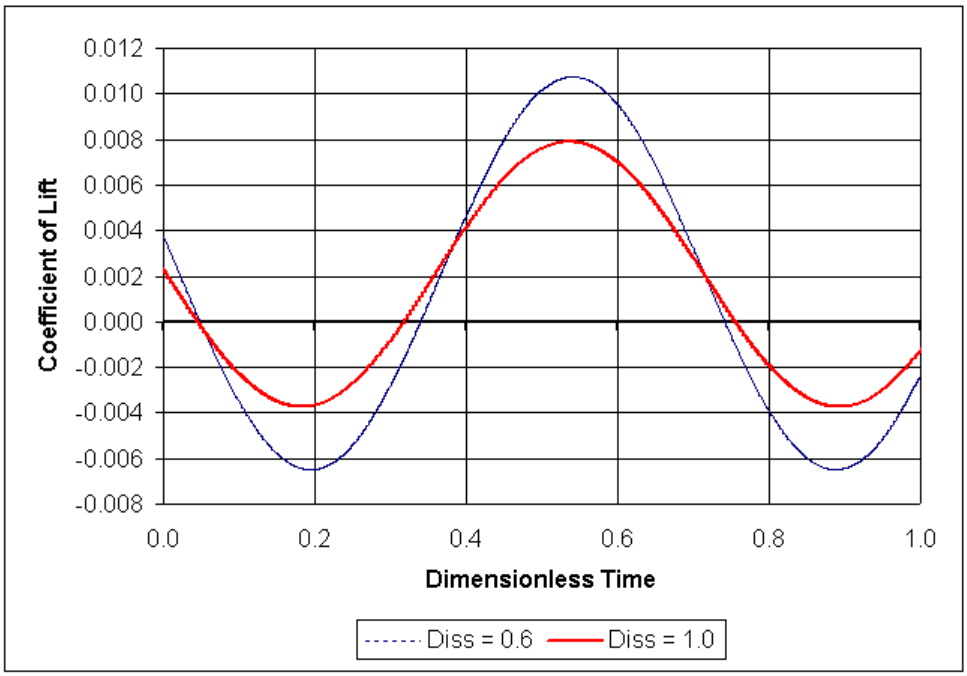

Figure 5-38: Oscillating Lift Coefficient for Ellipse with *diss* = 0.6 and 1.0

5.1.6. Circular Cylinder

The geometry for the circular cylinder test case is given below.

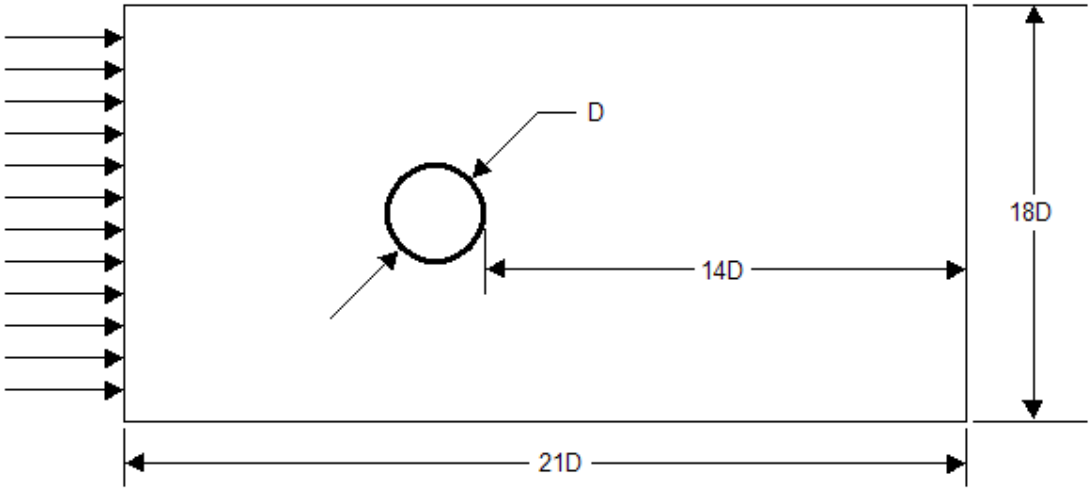

Figure 5-39: Layout of Circular Cylinder Test Case

Six Reynolds numbers where tested and are shown on the following pages. Each cylinder flow is used to characterize a different aspect of laminar bluff body flow. The first Reynolds number is 1.94, followed by 9.6, 2.6, and 4.1, the suggested limit of a stable wake. Those cases are followed by Reynolds numbers of 105 and 200, which show a definite von Karman vortex street. Each of these flows is verified according to pictures from van Dyke (1982).

The control parameters for these cases are given in Table 5-7, and a sample of the unstructured mesh for the cylinders is given in Figure 5-40.

Table 5-7: Control Parameters for Circular Cylinders

Gamma	1.40					
Dissipation	1.00	1.00	1.00	1.00	1.0 & 0.5	1.0, 0.5, 0.25
Dim. Time	1.00E-05	5.00E-05	1.00E-04	1.00E-04	5.00E-04	5.00E-04
CFL	0.5	0.5	0.5	0.5	0.5	0.5
Visc. Ratio	-0.66					
Mod. Suther.	0.0					
Mach Num.	0.3					
Reynolds Num.	1.94	9.6	26	41	105	200
Prandtl Num.	0.7					
Angle of Attack	0.0					
Ref. Dimen.	1.0					
Ref. Density	1.0					
Ref. Accoustic Speed	1.0					
Num. of Steps	400000	400000	225000	225000	100000	100000
Cycles per Step	15	15	15	15	15	15
Solution Type	Unsteady					
Diss. Type	Grad. Limiters					
Integration Type	1 point					

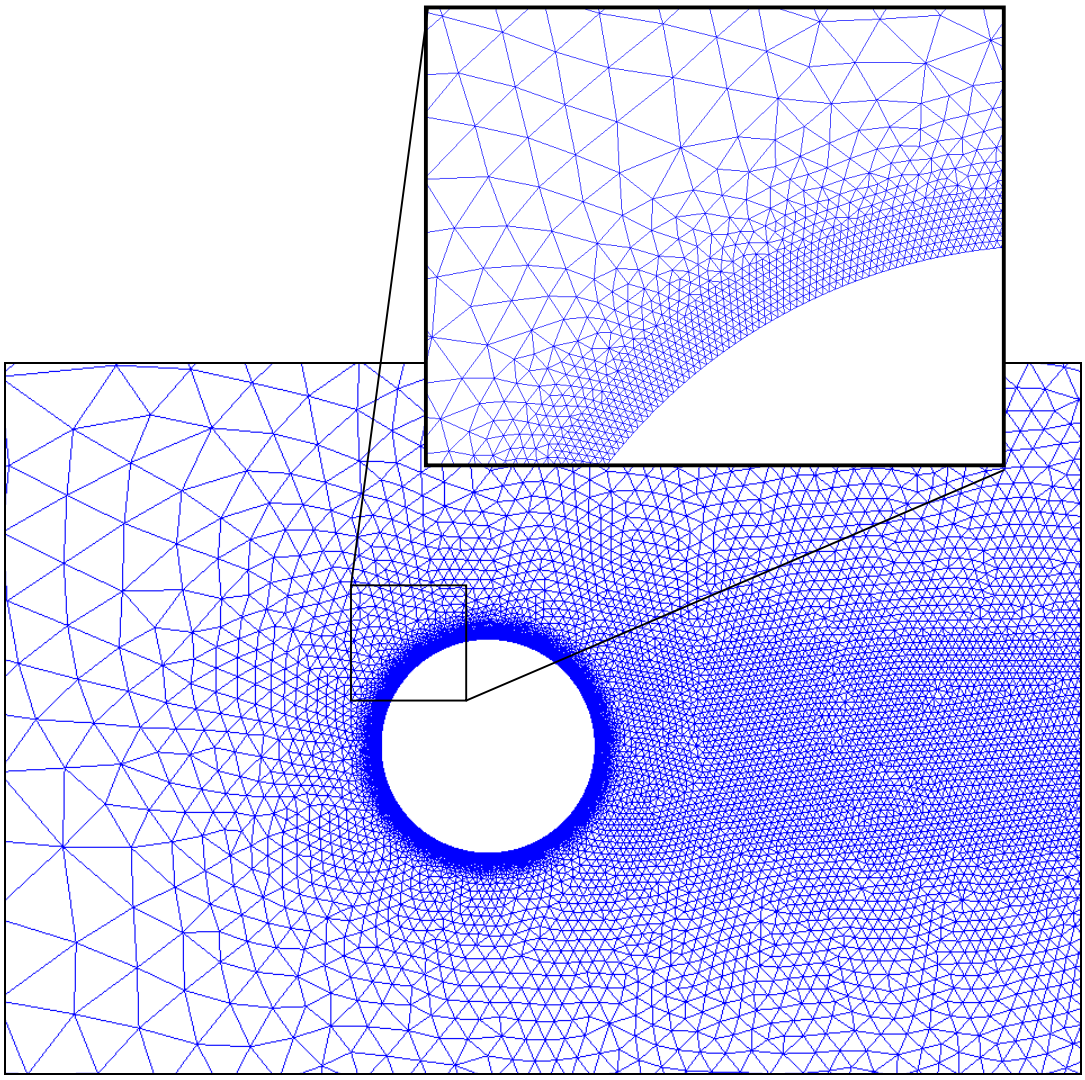

Figure 5-40: Unstructured Mesh for Circular Cylinder Cases

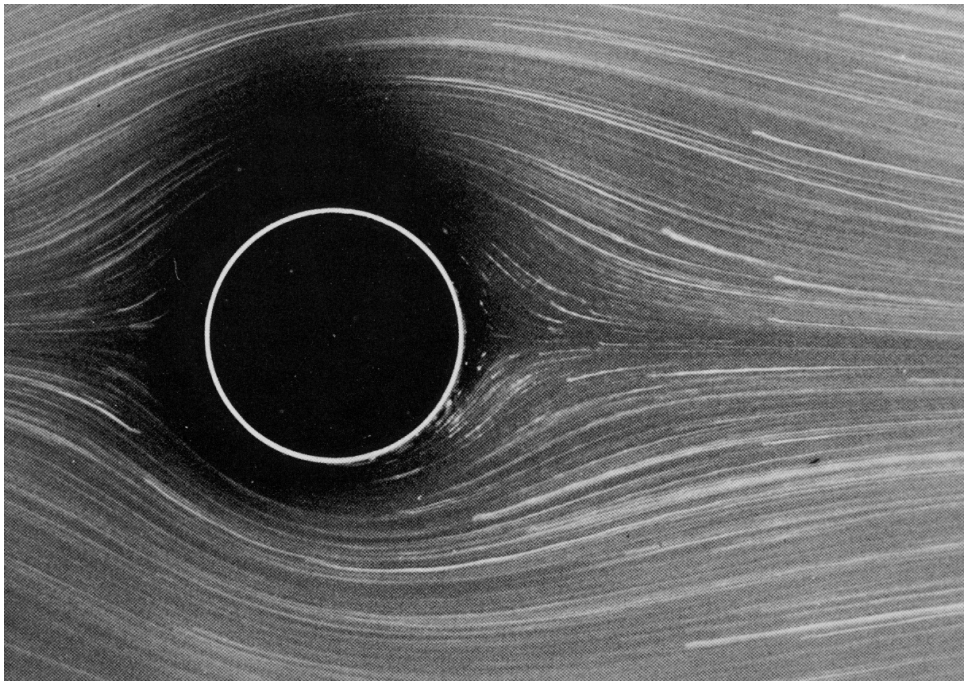

Figure 5-41: Circular Cylinder at Re = 1.54. (van Dyke, 1982)

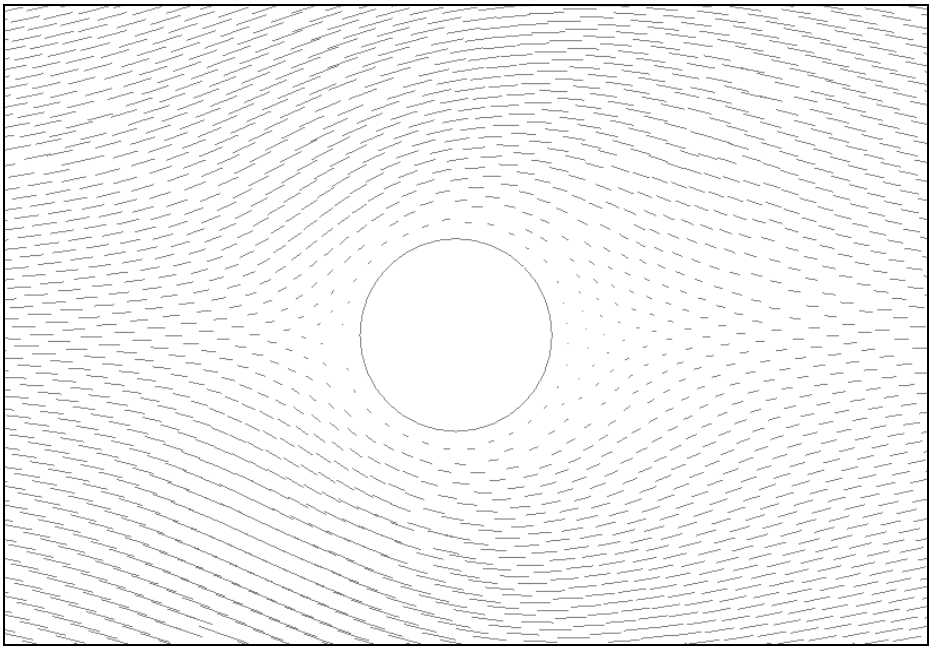

Figure 5-42: Computational Circular Cylinder at *RE* = 1.54

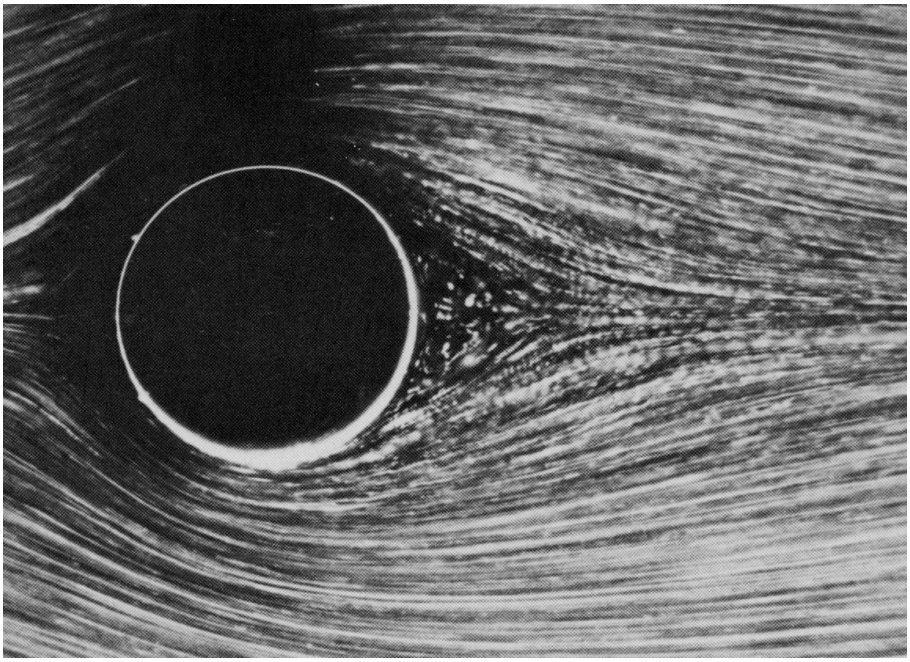

Figure 5-43: Circular Cylinder at Re = 9.6 (van Dyke, 1982)

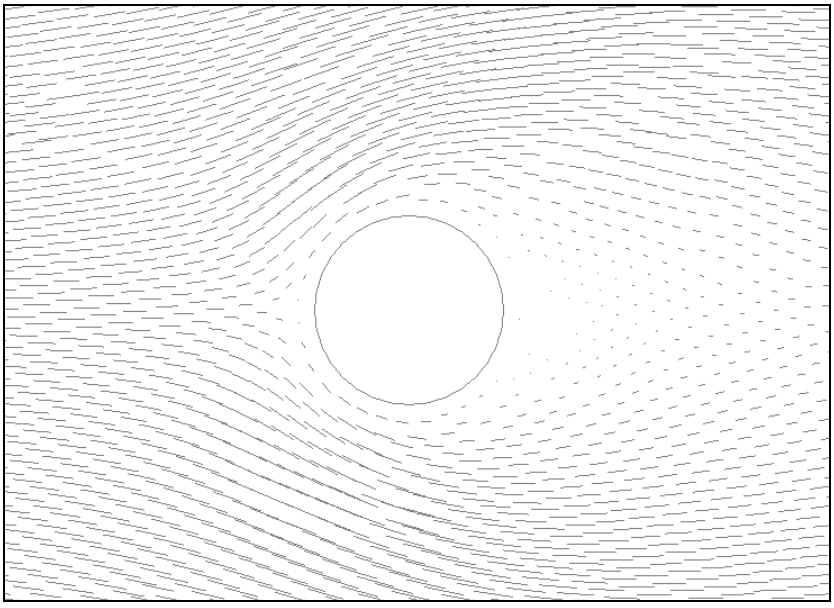

Figure 5-44: Computational Circular Cylinder at *RE* = 9.6

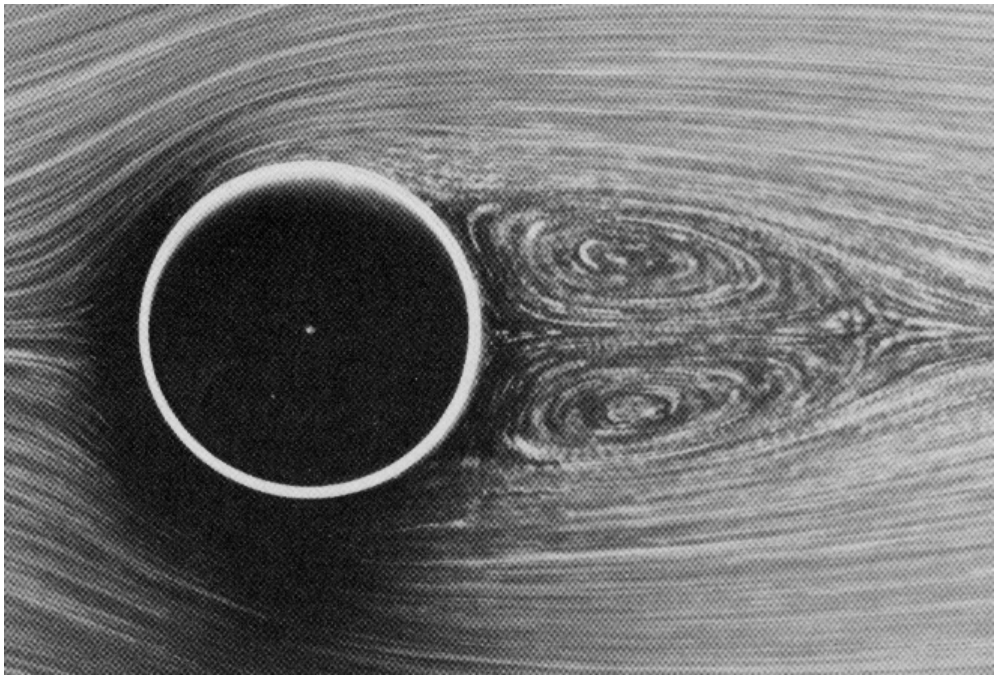

Figure 5-45: Circular Cylinder at Re = 26.0. (van Dyke, 1982)

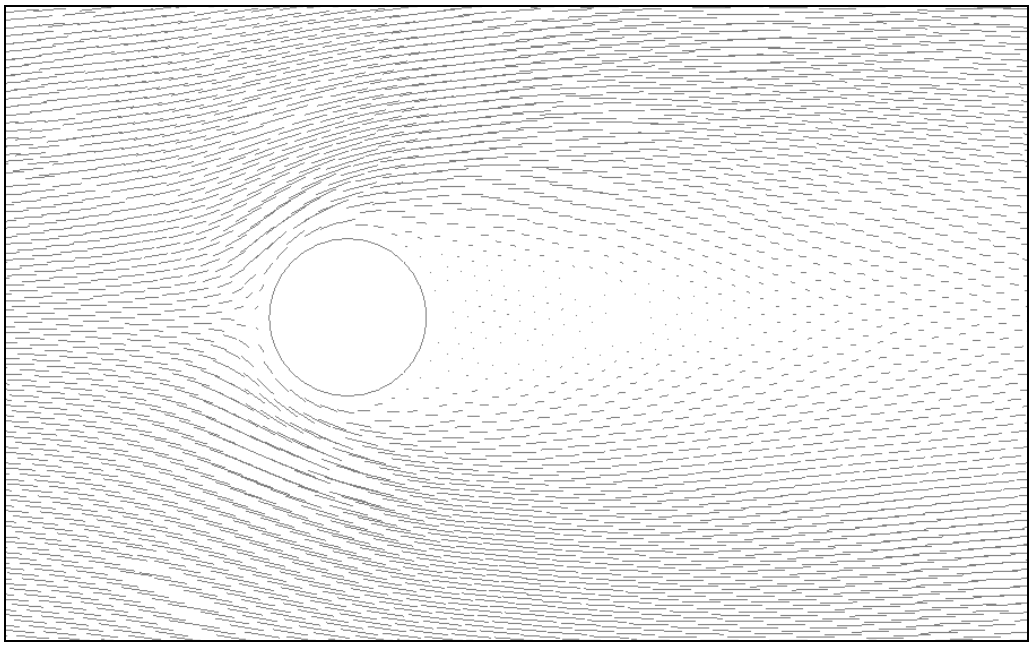

Figure 5-46: Computational Circular Cylinder at *RE* = 26.0

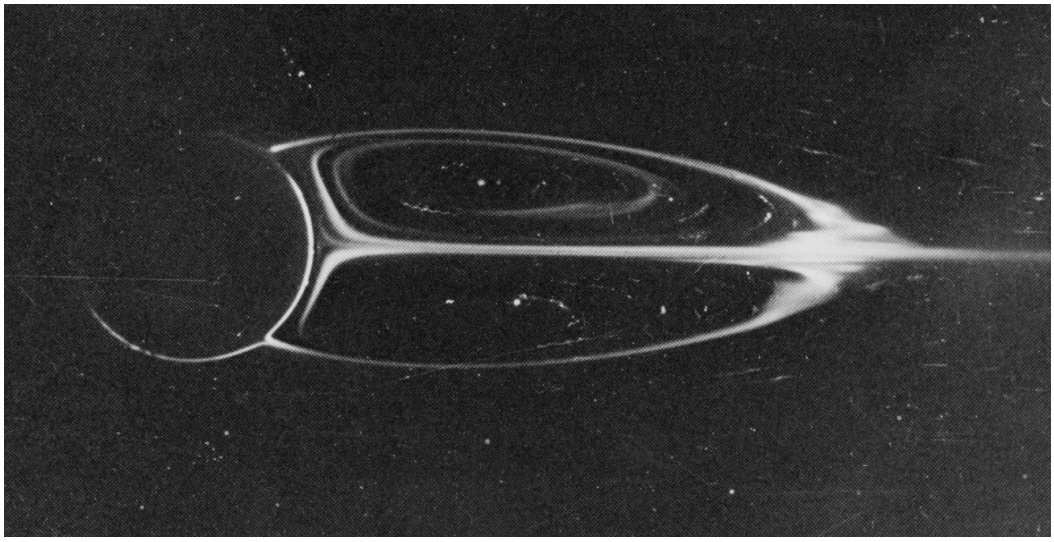

Figure 5-47: Circular Cylinder at Re = 41.0. (van Dyke, 1982)

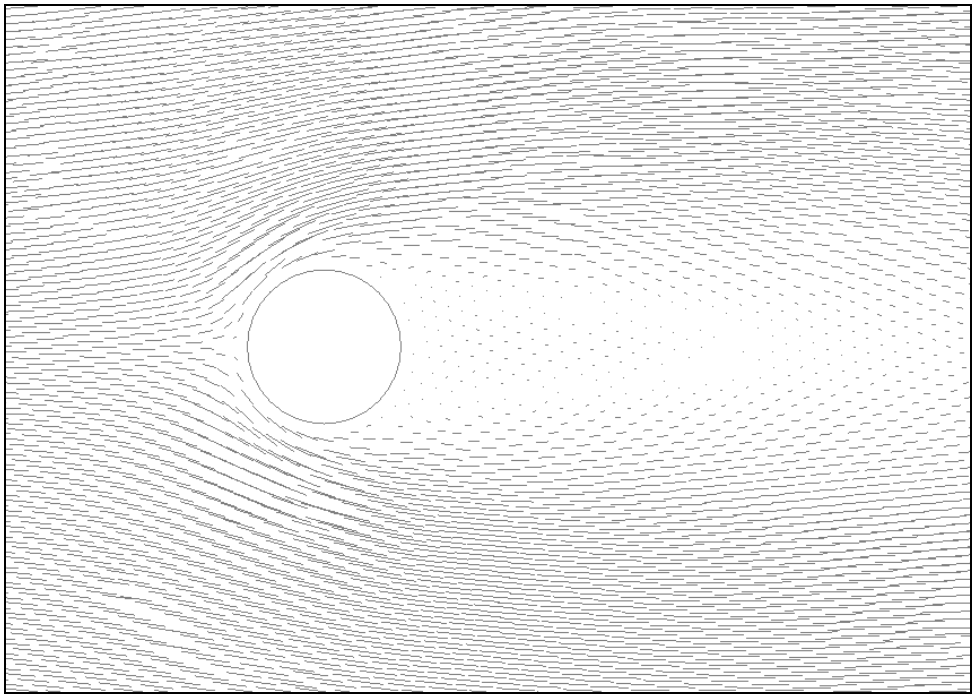

Figure 5-48: Computational Circular Cylinder at *RE* = 41.0, Velocity Vectors

Van Dyke comments in the caption under Figure 5-47 that "this [Reynolds number 41.0] is the approximate upper limit for steady flow. Far downstream the wake has already begun to oscillate sinusoidally." The sinusoidal oscillations are far enough down stream that variations in velocity, density, and pressure are hard to see, and oscillations in the forces on the cylinder are negligible. But Figure 5-49 shows the distribution of entropy in the wake. From that figure, the oscillations can be distinguished quite easily.

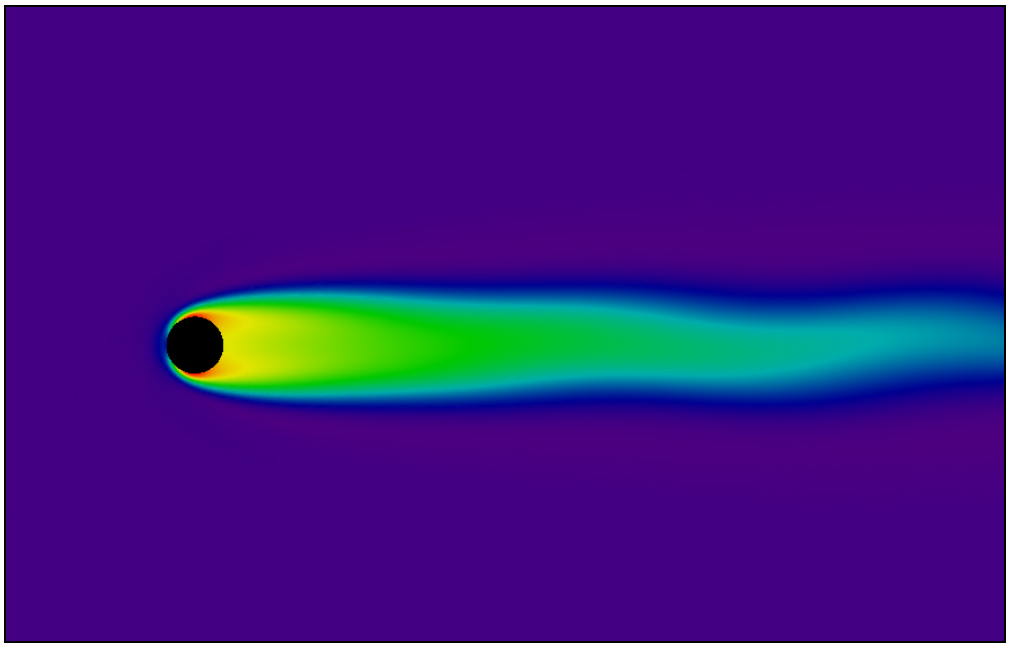

Figure 5-49: Computational Circular Cylinder at *RE* = 41.0 (Entropy Production)

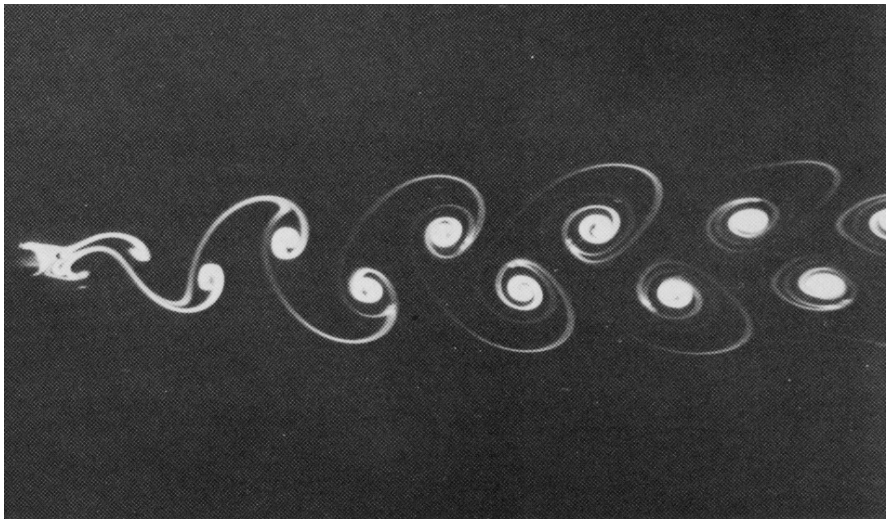

Figure 5-50: Circular Cylinder at Re = 105, Streamlines Shown by Electrolytic Precipitation in Water (van Dyke, 1982)

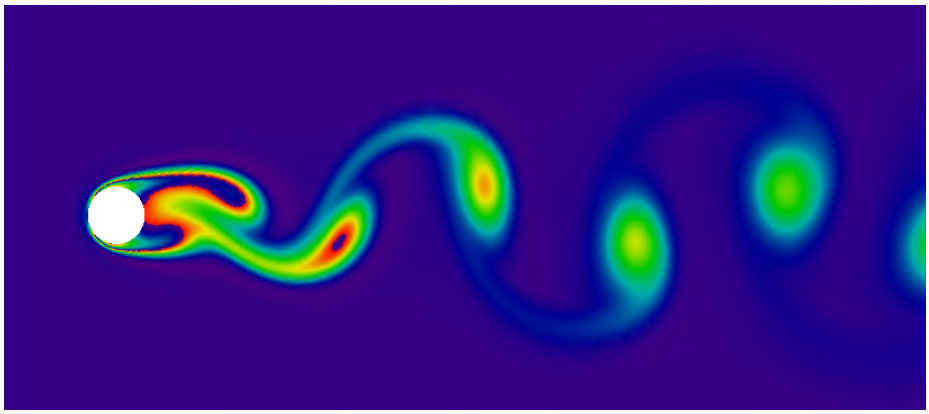

Figure 5-51: Computational Circular Cylinder at *RE* = 105 (Entropy Production)

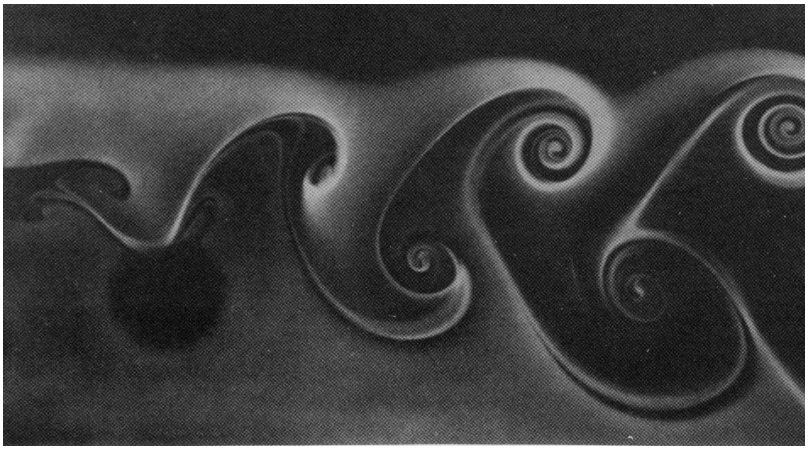

Figure 5-52: Circular Cylinder at Re = 200, Streaklines Shown by a Thin Sheet of Tobacco Smoke (van Dyke, 1982)

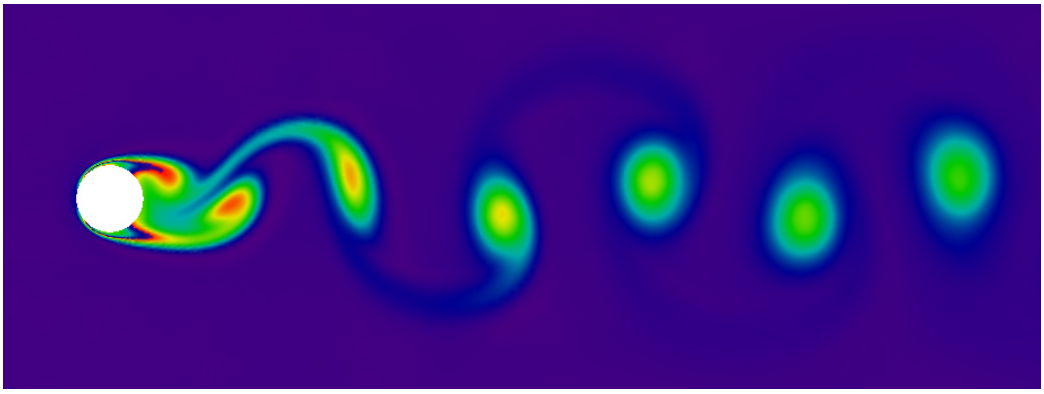

Figure 5-53: Computational Circular Cylinder at *RE* = 200 (Entropy Production)

The vortex streets generated by *NS2d* are shown to be confined to the wake directly behind the cylinder. Figure 5-54 shows the vector field for the Re = 200 case. Movement of the vortex centers is explained by the effects of the relative freestream velocity. Figure 5-55 illustrates the velocity distribution between two opposing vortices. The picture on the right of Figure 5-55 shows the velocity distribution shifted by the freestream. With the application of a freestream, the center of rotation is moved closer to the centerline of the flow. This phenomenon explains differences in the previous figures.

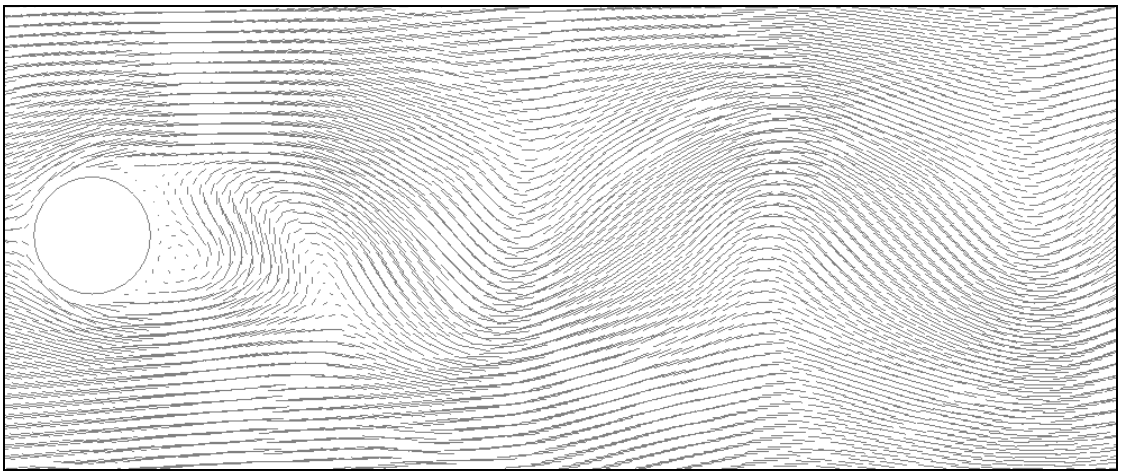

Figure 5-54: Vector Plot of Circular Cylinder at *RE* = 200

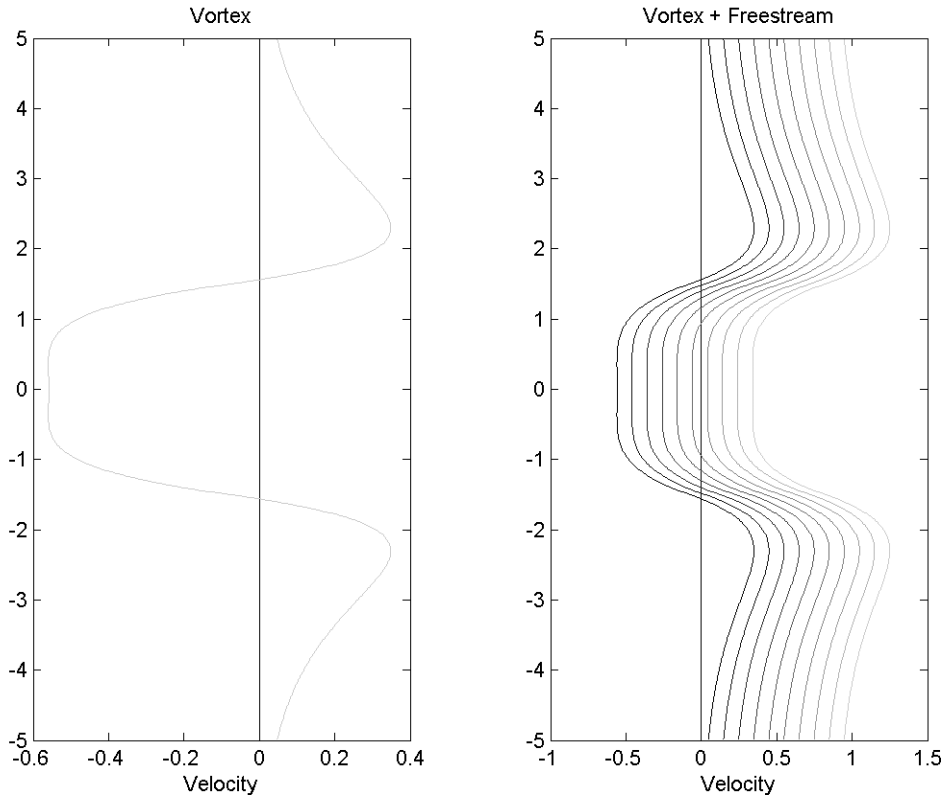

Figure 5-55: Velocity Distribution for Two Opposing Vortices (Left) With the Addition of a Freestream at Various Velocities (Right)

5.1.7. Leading Edge Separation on Flat Plate

The flat plate was originally generated to investigate separation and laminar reattachment, especially in the leading edge separation case. The case was unsuccessful at generating the flow geometry similar to that shown in Figure 5-57. Instead three different cases were run to investigate the problem.

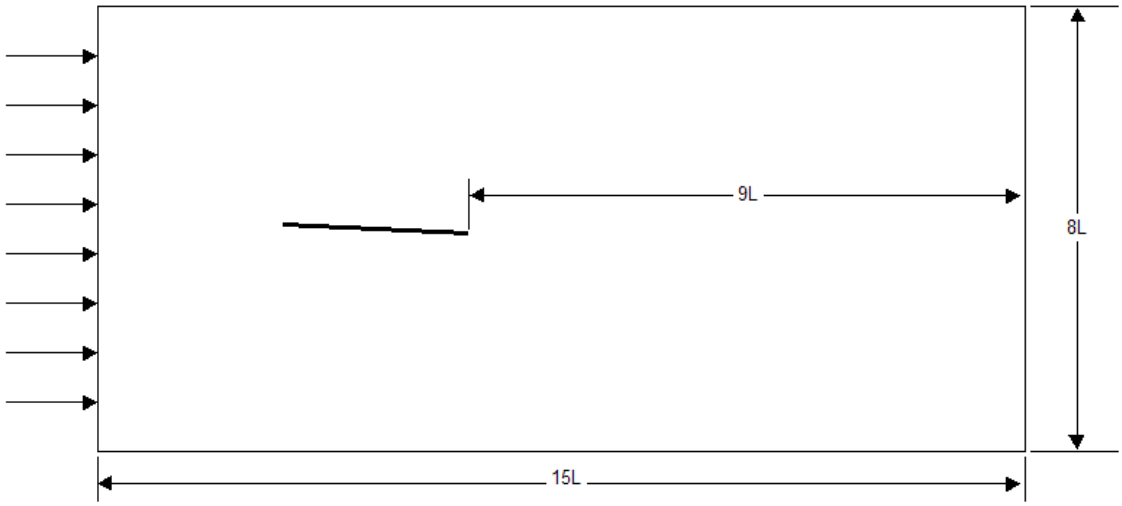

Figure 5-56: Layout of Leading Edge Separation Test Case

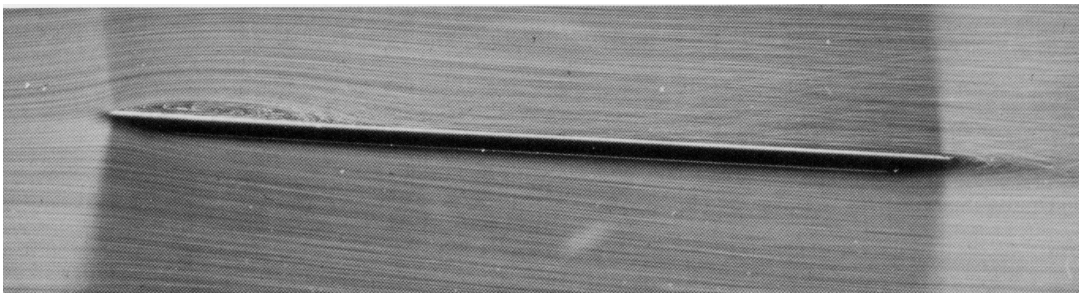

Figure 5-57: Leading Edge Separation on Flat Plate, Re = 10,000 (van Dyke, 1982)

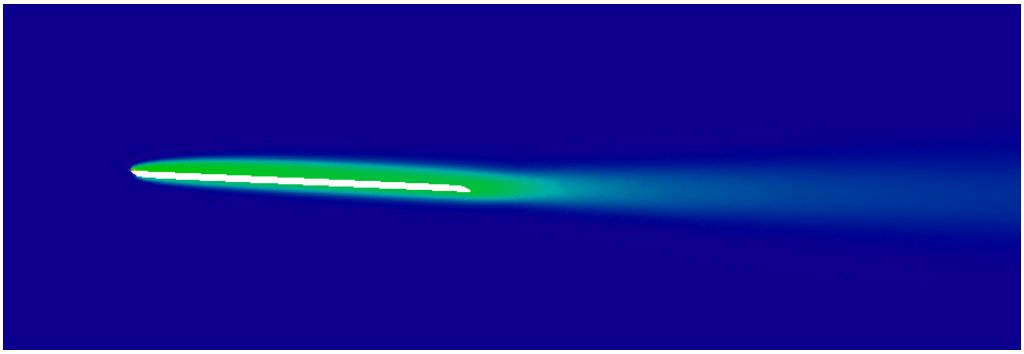

Figure 5-58: Plate for *DISS* = 1.0, *RE* = 10,000

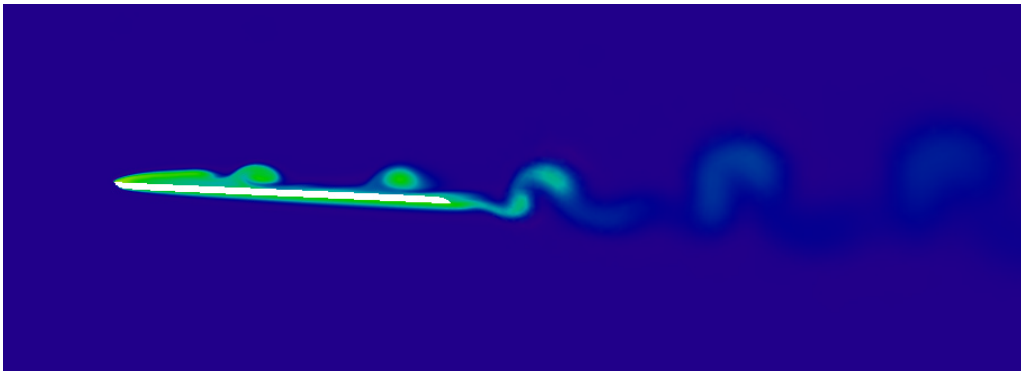

Figure 5-59: Plate for *DISS* = 0.6, *RE* = 10,000

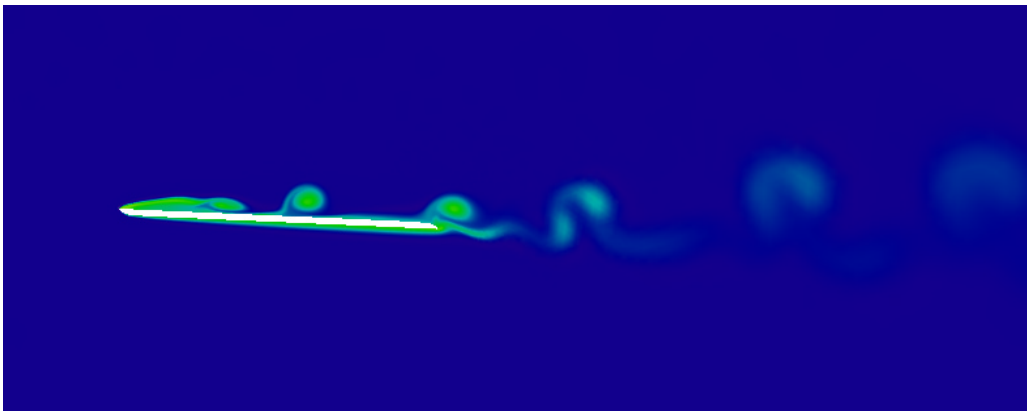

Figure 5-60: Plate for *DISS* = 0.3, *RE* = 10,000

The plate was found to be highly dependent on the dissipation factor. In fact, with too much dissipation, the entire plate separates (shown in Figure 5-58). As the dissipation was decreased, the flow formed a leading edge separation bubble and began to release vortices into the flow. These vortices persisted through the variation of dissipation until the dissipation dropped low enough that the solution became unstable.

The problem with the leading edge separation bubble is hard to understand. While dissipation seems to corrupt the solution, the lack of dissipation disrupts the solution altogether. Two suggestions arise: (1) Add a turbulence model and see if the problem persists; or, (2) try a different type of dissipation model. Both of these ideas are outside of the scope of this thesis and are therefore left for future research.

5.2. Validation

Without a turbulence model, good validation prospects are less frequent than verification. The validation section opens with the circular cylinders from the previous section. The drag on the cylinders and their Strouhal number is compared to experiment. A low subsonic Selig-Donavan 7080 airfoil is then used to compare lift and drag at various angles of attck. Finally, one high subsonic and two transonic airfoil cases are illustrated. The first two are a NASA 10% Supercritical airfoil run at Mach numbers of 0.6 and 0.79. The Euler solver shows both airfoils to contain shocks at different locations than that of the Navier-Stokes solver. Problems with these cases are addressed along with explanations. And the third case is a Boeing A4 airfoil at a Mach number of 0.789. This case shows similar difficulties and little results.

5.2.1. Circular Cylinder

The drag coefficients for the six cylinder cases are given in Table 5-8. These values are further plotted against other reputable numerical solutions. The two numerical solutions originate from Lei (2004) and Nakabayashi (1993). Lei used a finite difference solver to estimate the drag coefficient and Strouhal numbers for various Reynolds numbers from 10 to 1000; Nakabayashi used iterated Oseen's approximation on various shear flows to find the drag, lift, and flow visualizations for Reynolds numbers from 2 to 40. Both solvers were verified and validated in their own right.

Table 5-8: Drag Coefficient vs. Reynolds Number for Cylinder Cases

Re	Cd
1.54	8.15E+00
9.6	2.52E+00
26.0	1.65E+00
41.0	1.40E+00
105	1.27E+00
200	1.31E+00

The values are presented above for future comparisons, but Figure 5-61 shows good agreement between the three viscous solvers.

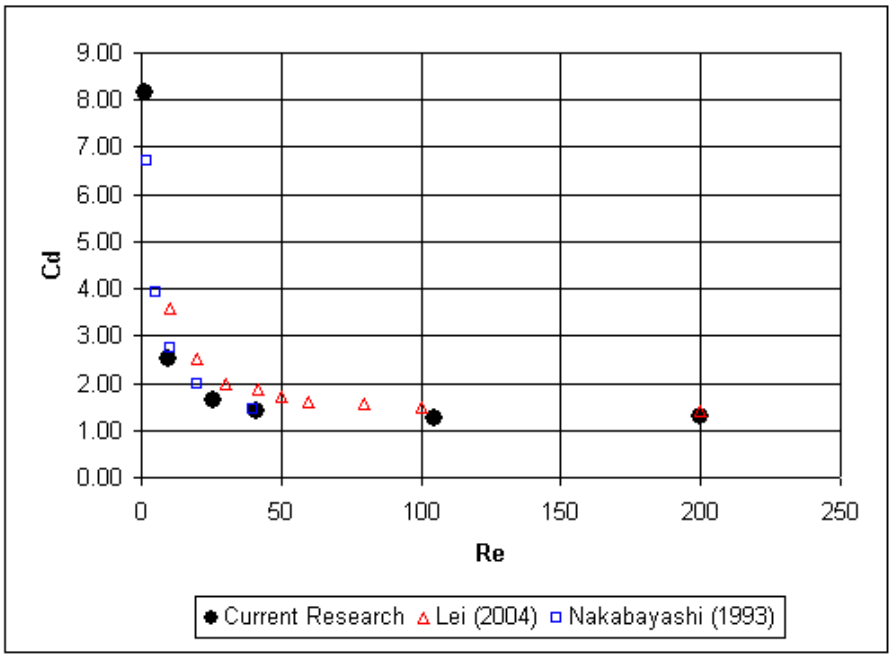

Figure 5-61: Drag Coefficient vs. Reynolds Number for Various Viscous Codes

The Strouhal number is also plotted versus Lei (2004) and Williamson (1998). Strouhal numbers for the current research were taken form the lift history plots shown in Figures 5-60 and 5-61. Williamson uses a square root approximation to find the relationship between Strouhal and Reynolds numbers. The current results and those of Lei are presented as points in Figure 5-64, while Williamson's empirical equation is used to generate a trend line through the data.

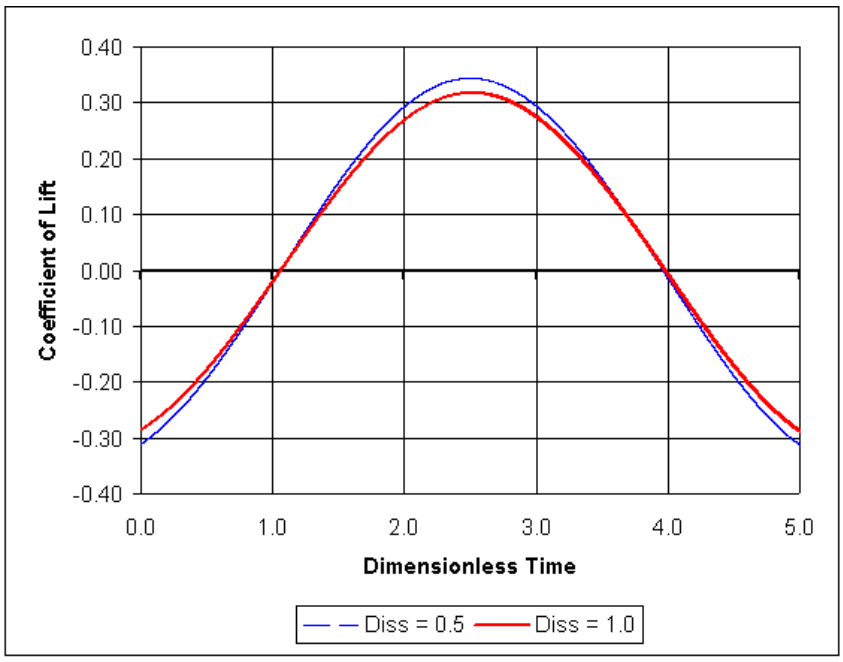

Figure 5-62: Lift Coefficient for Cylinder at Re = 105 with *diss* = 0.5 and 1.0

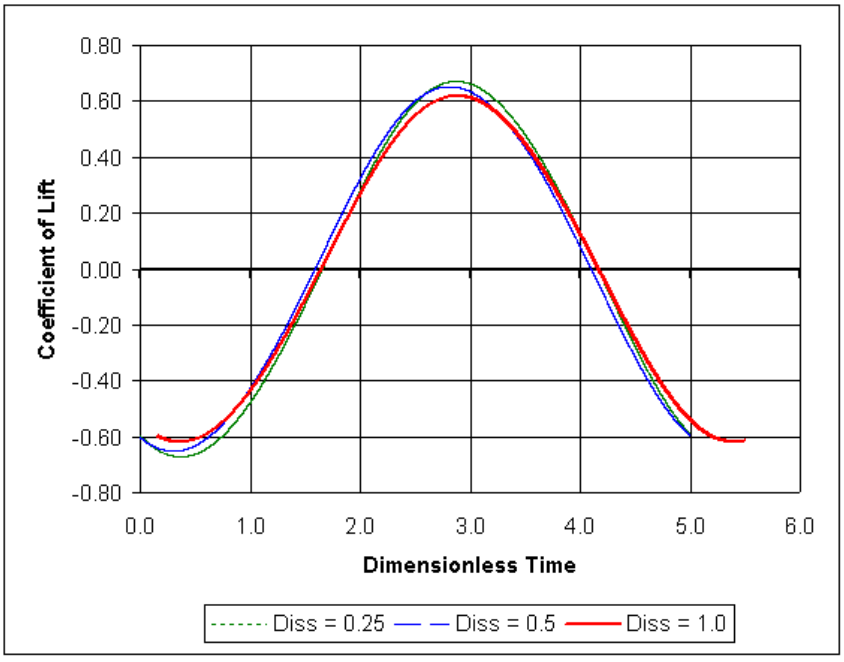

Figure 5-63: Lift Coefficient for Cylinder at Re = 200 with *diss* = 0.25, 0.5, & 1.0

NS2d predicts the Strouhal number to be 0.17 for the Re = 105 case and 0.19 for Re = 200. These numbers correspond well to White's (1991) approximation of 0.2 for Reynolds numbers between 100 and 10^5. The lack of dissipation in the Re = 105 case increased the amplitude of lift oscillations by 8% (see Figure 5-62); the Re = 200 case tested three dissipation values showing an increase of 5% and 8% in the 0.5 and 0.25 dissipation cases.

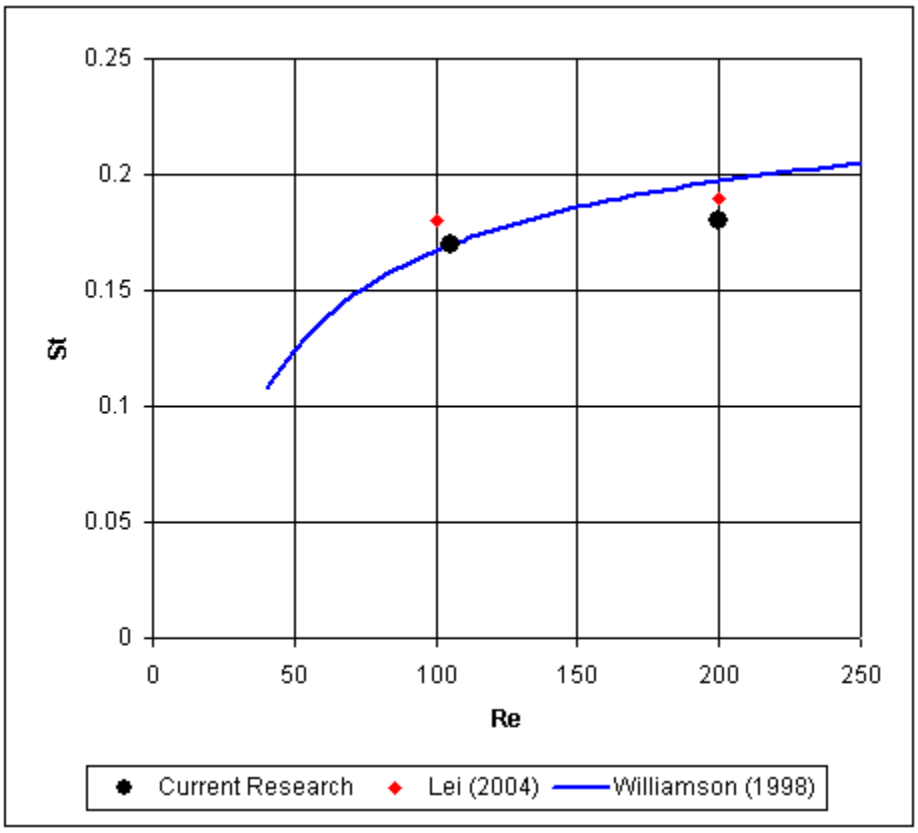

Figure 5-64: Strouhal Number vs. Reynolds Number for Various Cases

As a final note, the empirical curve of Williamson has been terminated on the low end by Re = 40. This limit was specified by Lei to be the lower limit of Strouhal oscillations. The limit also verifies the findings of Figure 5-49 and comment by van Dyke (1982).

5.2.2. SD 7080 Airfoil

The Selig-Donavan 7080 airfoil geometry is given in Figure 5-65. The airfoil ordinates and experimental data were both taken from the UIUC Airfoil Data Site (Selig, 2004). The lift and drag curves are illustrated in Figure 5-66 and Figure 5-67, respectively. The lift data from *Euler2d* and *NS2d* was plotted versus experimental data for Reynolds numbers of 99943, 100018, 149842, and 150082 to obtain a good average. Likewise, the drag data was plotted versus Reynolds numbers of 99943 and 149842.

The lift curve generated by *Euler2d* is linear until the flow separates over the top surface at an angle of 16 degrees. The separation is caused by "numerical viscosity" in the dissipation model. The SD 7080 was then generated in *NS2d* at a Reynolds number of 10^5 and a Mach number of 0.3. The lift curve from *NS2d* is more reasonable at lower angles of attack, but separates at higher angles. Because no turbulence model was implemented in the viscous code, the flow must remain purely laminar. At higher angles of attack, the lift is not suppressed by a higher pressure that could occur if turbulence was being modeled.

Figure 5-65: SD 7080 Airfoil Geometry

Table 5-9: Control Parameters for SD 7080 Airfoil

Gamma	1.4
Dissipation	1.0
Dim. Time	1.0E-03
CFL	0.50
Visc. Ratio	-0.66
Mod. Suther.	0.0
Mach Num.	0.3
Reynolds Num.	1.0E+05
Prandtl Num.	0.7
Angle of Attack	0.0 to 16.0
Ref. Dimen.	1.0
Ref. Density	1.0
Ref. Accoustic. Speed	1.0
Num. of Steps	100000
Cycles per Step	15
Solution Type	Unsteady
Diss. Type	Grad. Limiters
Integration Type	1 Point

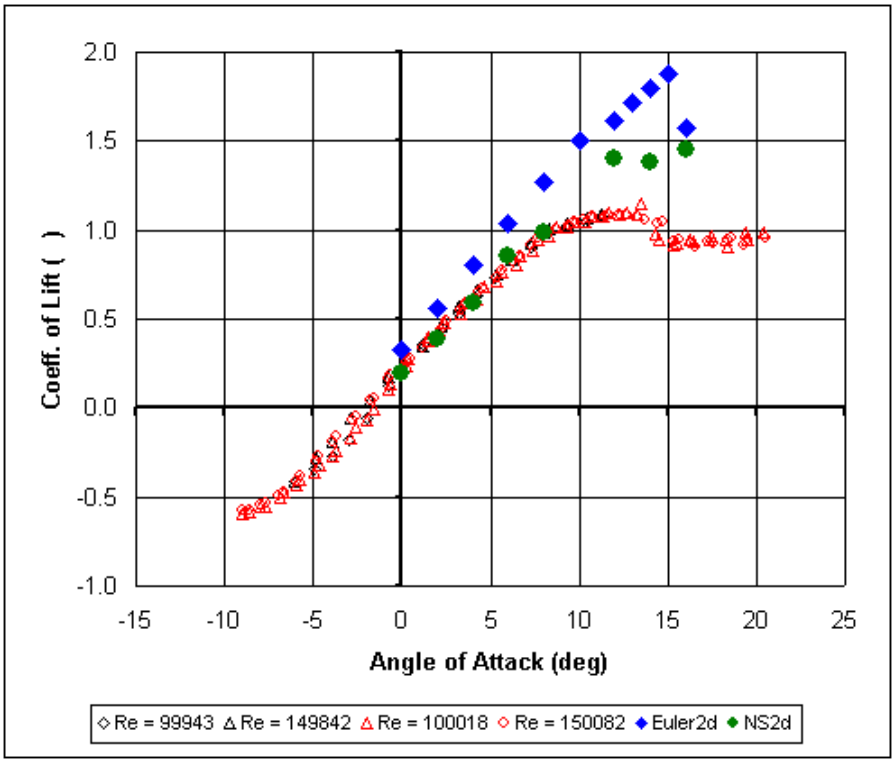

Figure 5-66: Lift Curve for SD 7080 Airfoil

Similar to problems with the lift, the drag curve is lower at small angles of attack and increases faster than the experimental data. The lower drag at small angles originates in the turbulence problem again. Turbulence is not modeled and the Reynolds numbers are high enough that transition, if not turbulence, should be occurring on the airfoil. At mid angles, the drag is increased because the flow separates slightly from the trailing edge and also sheds an unsteady wake (see Figure 5-68). At higher angles of attack, the flow separates off the airfoil at the leading edge instead of staying attached (see Figure 5-69). This vortical separation causes an increase in drag.

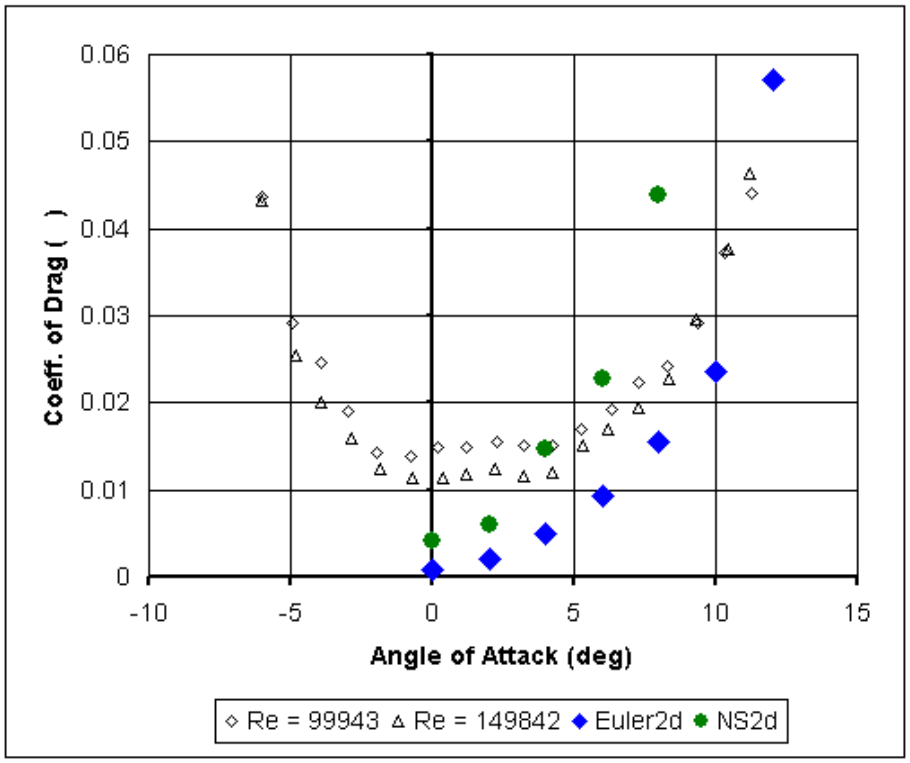

Figure 5-67: Drag Curve for SD 7080 Airfoil

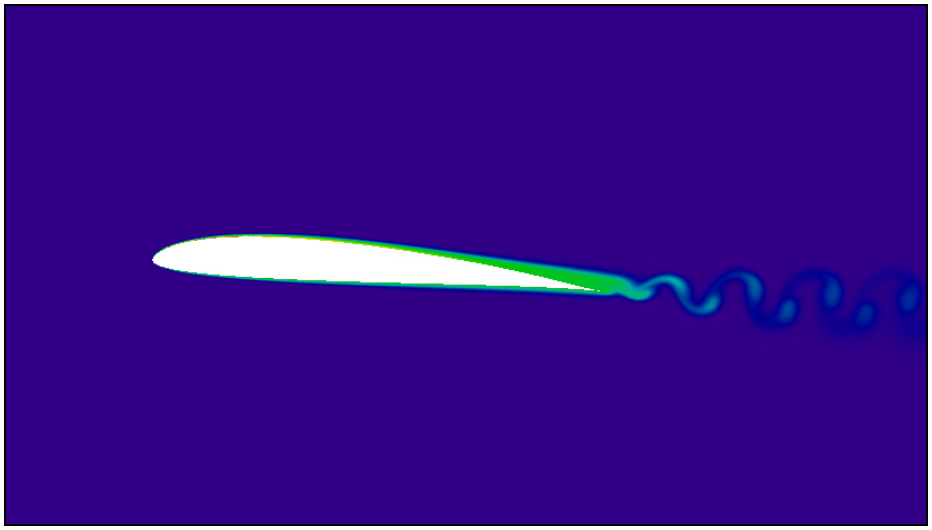

Figure 5-68: SD 7080 at 4^{o} Angle of Attack, RE = 10^{5} (Entropy Production)

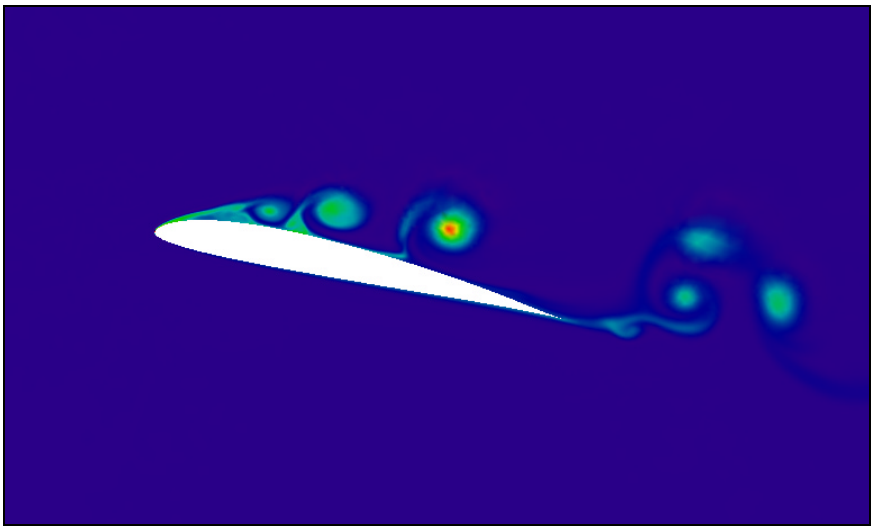

Figure 5-69: SD 7080 at 10^{o} Angle of Attack, RE = 10^{5} (Entropy Production)

5.2.3. 10% Thick NASA Supercritical Airfoil

The airfoil geometry tested is shown in Figure 5-70. The airfoil was run through to Mach number of 0.6 and 0.79. The corresponding Reynolds numbers were 2.35×10^{7} and 2.79×10^{7}. The resulting coefficient of pressure plots are shown in Figures 5-71 and 5-74. The computational pressure coefficients were taken from both Euler2d and NS2d and experimental data was taken from AGARD (Barche, 1979).

Figure 5-70: NASA 10% Thick Supercritical Airfoil Geometry

Table 5-10: Control Parameters for Both 10% Thick NASA Supercritical Airfoils

Subcritical Case	
Gamma	1.4
Dissipation	1.0
Dim. Time	1.0E-03
CFL	0.50
Visc. Ratio	-0.66
Mod. Suther.	0.0
Mach Num.	0.6
Reynolds Num.	2.35E+07
Prandtl Num.	0.7
Angle of Attack	2.0
Ref. Dimen.	1.0
Ref. Density	1.0
Ref. Accoustic. Speed	1.0
Num. of Steps	40000
Cycles per Step	15
Solution Type	Unsteady
Diss. Type	Grad. Limiters
Integration Type	1 Point

Supercritical Case	
Gamma	1.4
Dissipation	1.0
Dim. Time	1.0E-03
CFL	0.50
Visc. Ratio	-0.66
Mod. Suther.	0.0
Mach Num.	0.79
Reynolds Num.	2.79E+07
Prandtl Num.	0.7
Angle of Attack	2.0
Ref. Dimen.	1.0
Ref. Density	1.0
Ref. Accoustic. Speed	1.0
Num. of Steps	30000
Cycles per Step	15
Solution Type	Unsteady
Diss. Type	Grad. Limiters
Integration Type	1 Point

Looking at Figure 5-71, Euler2d predicts a shock at 10% of the chord; *NS2d* predicts all of the velocities on the airfoil to be below sonic speeds. This shock is not shown in the experimental data therefore the airfoil must be running at below the critical Mach number. The variation of pressure across the airfoil is represented well by *NS2d*, but problems are shown to occur near the leading edge, upper trailing edge, and under the trailing edge cusp. These problems are discussed below.

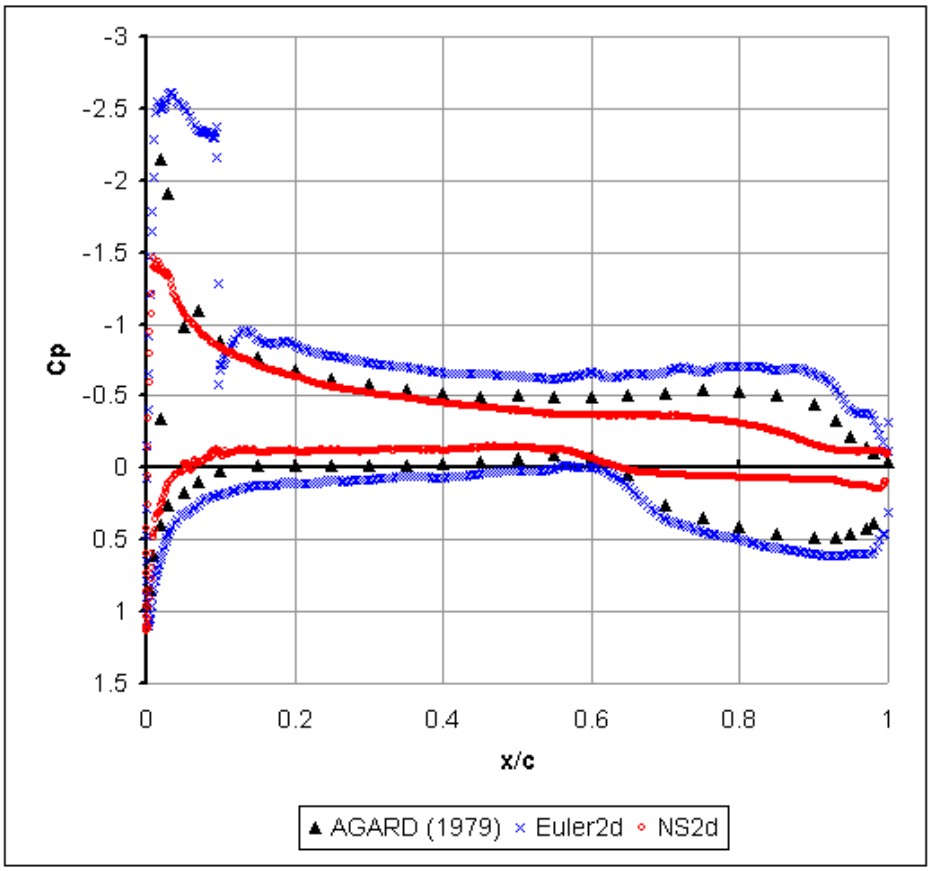

Figure 5-71: Coefficient of Pressure for NASA Supercritical Airfoil at M = 0.6

The flow over the cusp is not being predicted correctly by *NS2d* because *NS2d* lacks a turbulence model needed to retain attachment around the underside of the cusp. Separation under the cusp can be seen in Figure 5-72. The decrease in pressure on the top surface of the trailing edge can also be visualized using Figure 5-72. Again a turbulence model would most likely keep the flow attached to the upper surface for longer.

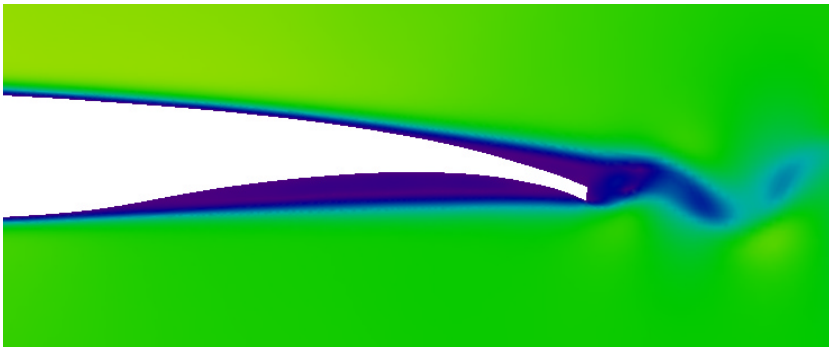

Figure 5-72: Mach Number for Cusp of NASA Supercritical Airfoil (M = 0.6)

Finally problems with the leading edge occur because the flow field cannot be properly represented by the current meshing software. The software was originally written for Euler solvers and has been used in that application since. Only recently has the meshing software been pushed to the limit of viscous meshing. The mesh around the leading edge is shown in Figure 5-73. As the source spacing at the leading edge, the mesh become less desirable. Notice the non-homogenous nature of the mesh in the figure and how the elements are skewed from equilateral to isosceles triangles. If the mesh spacing is decreased beyond that shown in Figure 5-73, the mesh become unusable.

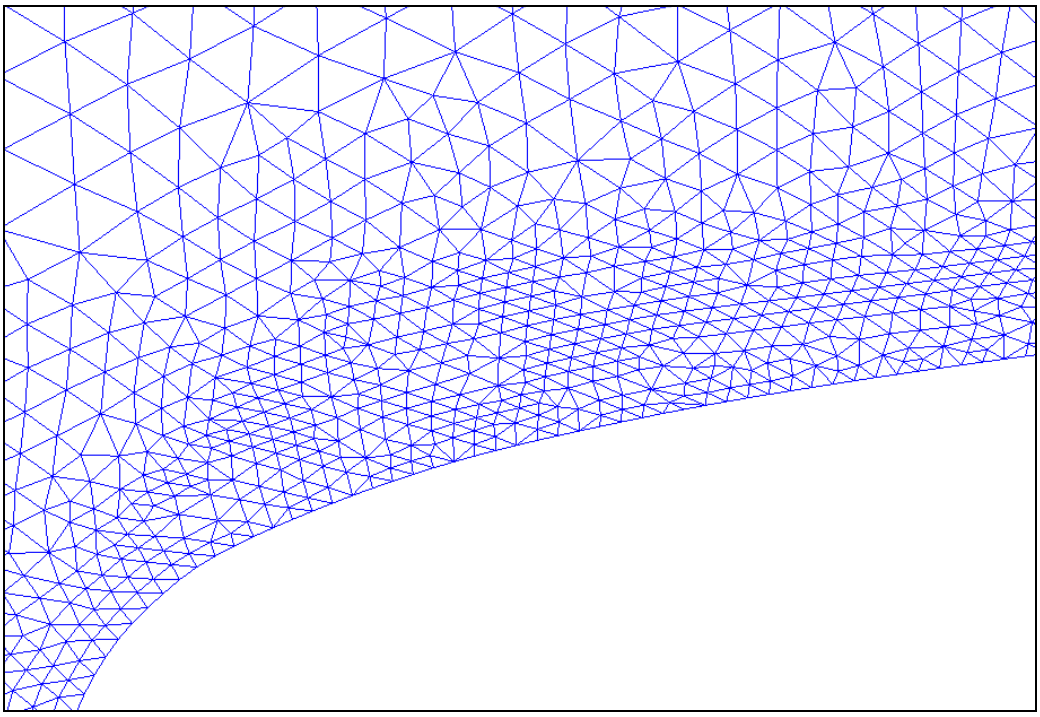

Figure 5-73: Leading Edge Mesh for NASA Supercritical Airfoil

Now examining the Mach 0.79 case, *Euler2d* and *NS2d* predicts were again run. The Euler solver predicted a shock very near the trailing edge (see Figure 5-75). The viscous solver also predicts a shock, but the shock quickly separates the flow and is forced forward on the airfoil. Again, because of a lack of turbulence, the boundary layer is separated by the pressure gradient across the shock. The separation causes a change in "geometry", which in turn moves the shock forward. The separation points moves with the shock until a lambda-like structure appears around the quarter chord of the airfoil. The shock structure is represented in Figure 5-76. In the figure, an oscillating wake is also shown to be shedding vortices from under the separation zone. The motion of the shock-separation region is described in more detail in the discussion of the Boeing A4 airfoil, below.

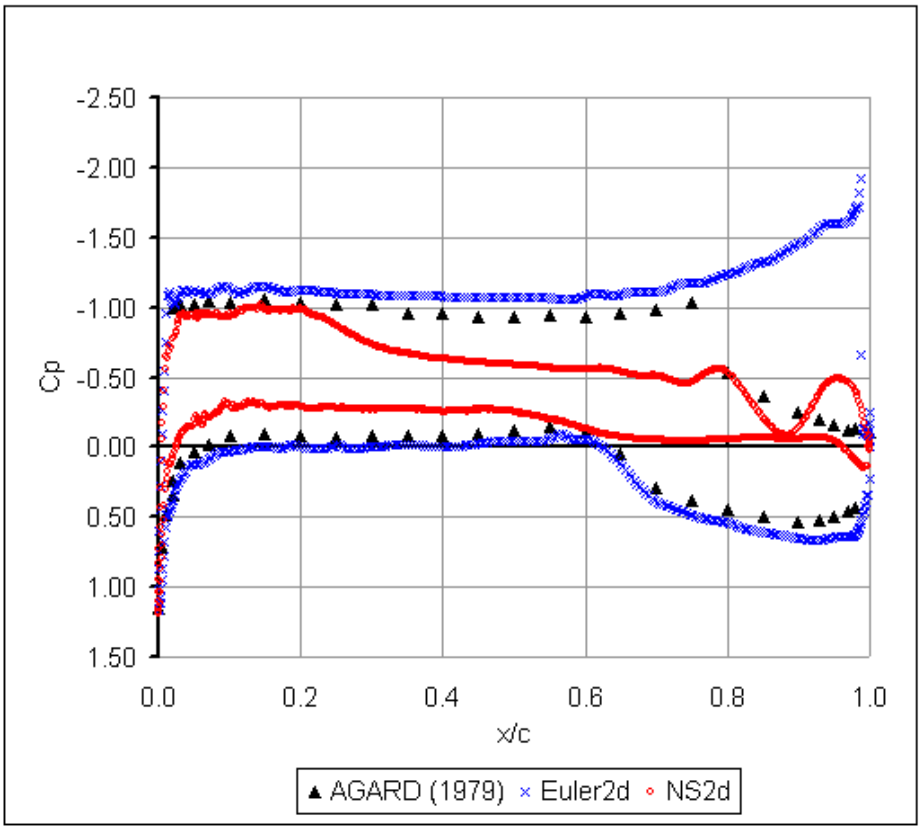

Figure 5-74: Coefficient of Pressure for NASA Supercritical Airfoil at M = 0.79

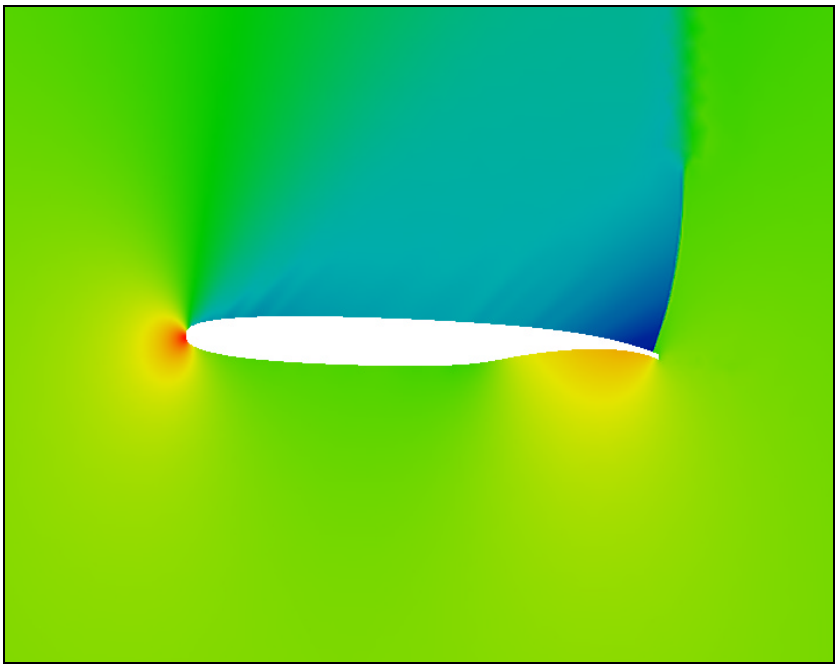

Figure 5-75: Euler2d Density Plot to NASA Supercritical Airfoil, *MACH* = 0.79

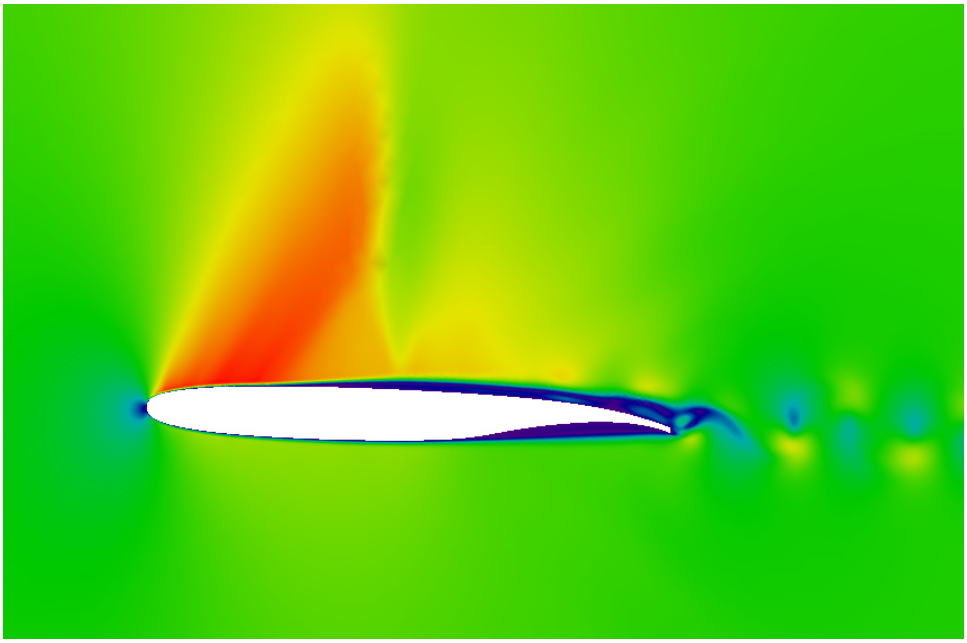

Figure 5-76: NS2d Mach Number Plot to NASA Supercritical Airfoil, *MACH* = 0.79

5.2.4. Boeing A4 Airfoil

Figure 5-77 gives the pressure distribution predicted by AGARD (Elsenaar, 1994) for validation of CFD codes. A clear shock is shown to be present on the upper surface at 65% of the chord. Figures 5-78 through 5-81 show the Boeing A4 airfoil at four progressions as the flow is marched through time. The first image shows the formation of the boundary layer and trailing edge shocks. The second image shows the shocks moving forward on the airfoil. Separation occurs just behind the shocks. As time progresses the shocks continue to move forward until the shocks move past the quarter chord at $dt = 16.0$. Only one suggestion arises at this time: The addition of a turbulence model could reattach the boundary layer and stop the shock from sliding forward.

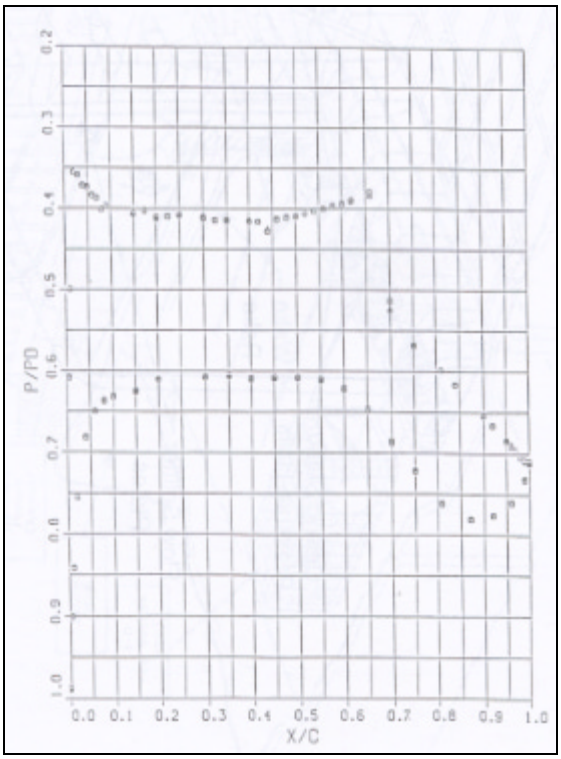

Figure 5-77: Pressure Distribution Measured by AGARD (1994)

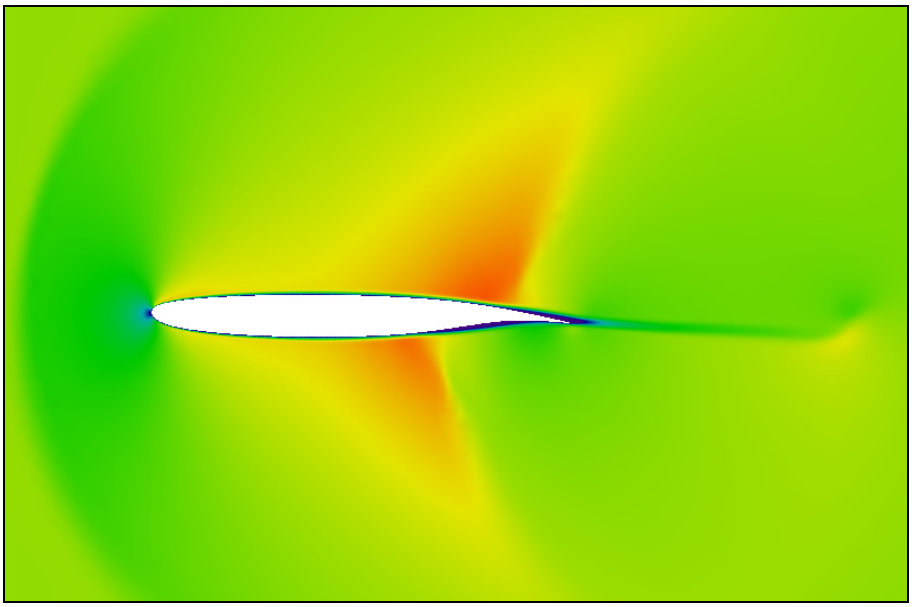

Figure 5-78: Boeing A4 Airfoil at *dt* = 1.5 (Mach Number Plot)

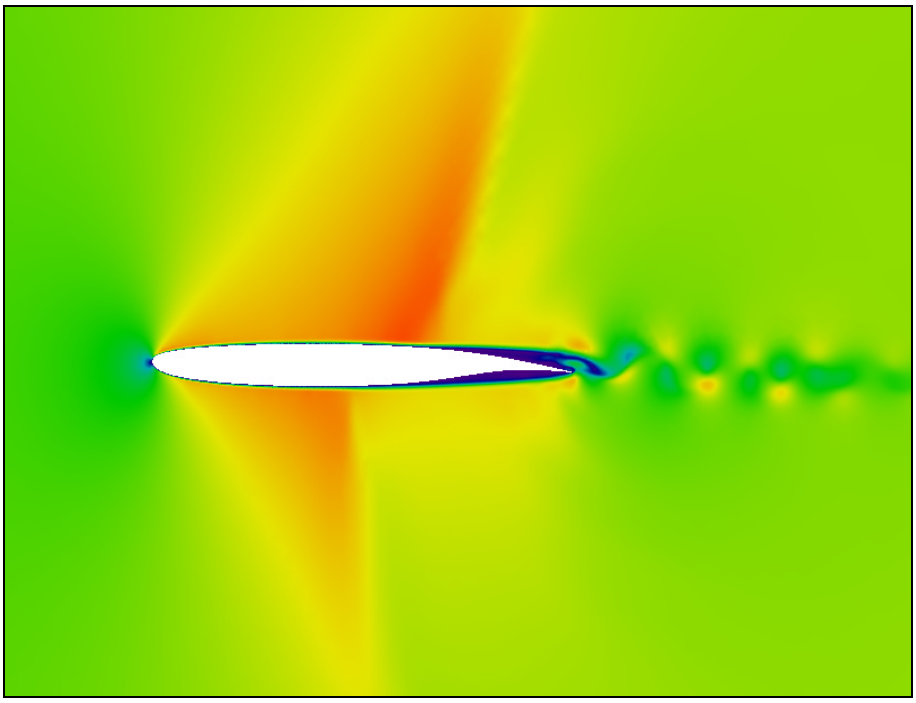

Figure 5-79: Boeing A4 Airfoil at *dt* = 4.5 (Mach Number Plot)

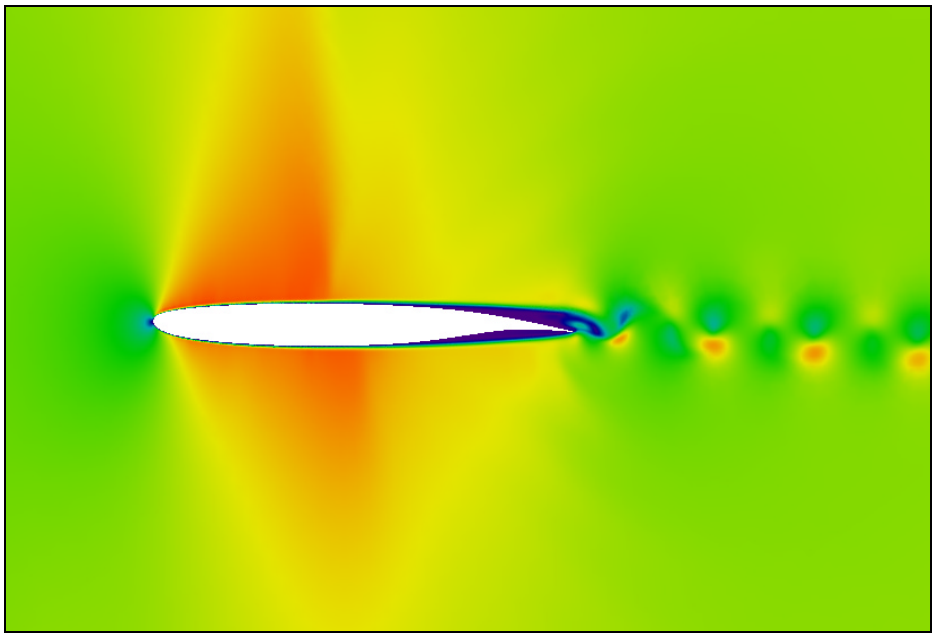

Figure 5-80: Boeing A4 Airfoil at *dt* = 10. 5 (Mach Number Plot)

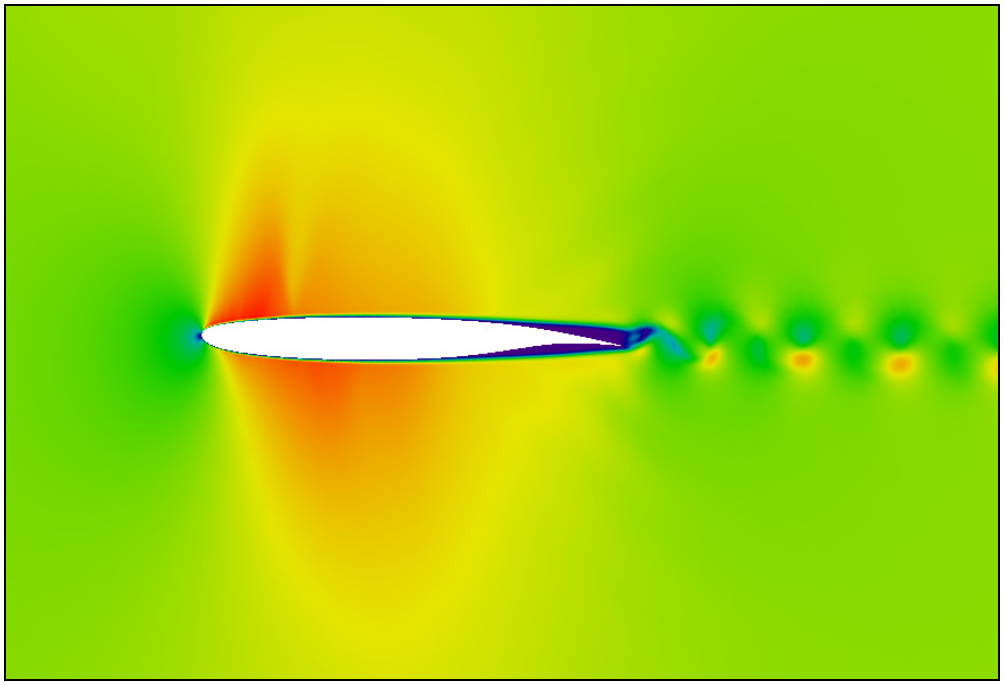

Figure 5-81: Boeing A4 Airfoil at *dt* = 16.0 (Mach Number Plot)

CHAPTER 6

6. Conclusion

This chapter will wrap up all of the previous discussion and results in the first section. The validity of the results and precautions to be used are also discussed below. The chapter ends with recommendations for future research. These recommendations include adaptations to the meshing software, further exploration into problems encountered by this research, and expansion of the current research beyond the current scope.

6.1. Discussion of Results

Several test cases were used to both verify and validate the *NS2d* code. A Blasius laminar boundary layer was used to verify that the solver was solving the governing equations for incompressible flow over a flat plate. Several tests were also run on the appropriate number of elements to represent the boundary layer. Six to seven elements give good results for both the velocity profile and the shear stress distribution. Several slow flow cases were run and compared to pictures from van Dyke (1982) in order to show that the solver could handle more complex geometries. These tests included the forward and backward facing step, rectangular block, 6:1 ellipse, and several circular cylinders. An NACA 0012 airfoil was used to represent a baseline flow over a streamlined body, while the flat plate airfoil showed that problems occur due to the

coupling of the natural and numerical viscosities. Several numerical dissipation effects were also shown including, vortex shedding after the dissipation is changed and change in unsteady lift amplitude for a cylinder.

The validation cases began with the circular cylinders and showed that the drag coefficient could be matched to that of other low Reynolds number computer codes and that the Strouhal number could be correctly modeled for higher laminar Reynolds numbers. A Selig-Donavan 7080 airfoil was used to generate a lift and drag plot for comparison to experimental data. The lift and drag data generated by *NS2d* was found to be better than that of *Euler2d*. The lift predicted by the viscous solver was found to be closer at small angles of attack, but separated late. The drag was also shown to be over predicted at all mid to high angles of attack. This excess drag is due to an enlarged, unsteady wake in mid angle cases and large vortical separation in high angle of attack cases. All of these problems are due to lack of turbulence in the solution.

The code was then run for a NASA 10% Thick Supercritical airfoil at both subcritical and supercritical Mach numbers. The subcritical case was found to model the pressure distribution on the leading edge better than *Euler2d*, which predicted a transonic shock. The subcritical case was hindered by the lack of a turbulence model and limitations on grid size and homogeneity. The supercritical case experienced problems with separation that moved forward with shock. The same problem arose for the Boeing A4 airfoil. Images were presented for the motion of the shock at four times in the unsteady flow.

This thesis has therefore shown that *NS2d* can appropriately model flow around moderately complex geometry at low Reynolds number regime. As the Reynolds number

is increased, the solver cannot handle large separated flow nor predict properties correctly when the boundary layer should be transitioning or turbulent. Problems also arose due to limitations in the meshing software available for the solver. Mesh size and homogeneity were both important issues. And finally, suggestions for stable steady-local and unsteady-global time steps were made according to empirical information from *NS2d*.

6.2. Recommendations for Future Research

Several recommendations have been made in the paragraphs that follow. These recommendations include suggestions for future investigations that lie outside the scope of this thesis, adaptation of the current solver, changes to the support programs, and continuation of research within the scope of this thesis.

Expansion of the Current Solver: The scope of this research included the addition of viscous terms to a two-dimensional, non-inertial Euler solver. This research should be continued by expanding the solver to contain one or several turbulent models and a viscous transpiration boundary condition. Several other additions to the solver are mentioned below, but these two are explicitly excluded from the scope of the current research. Also excluded is the expansion of Cowan's (2003) other solver from a three-dimensional Euler solver to include similar viscous, turbulent, and transpiration terms. The two-dimensional solver is needed as a staging ground for complex cross-sections and future additions to the solvers. The three-dimensional solver is further needed to generate solutions to complex geometries at high angles of attack, transonic conditions, and other viscous related aeroelastic cases.

Local Time Stepping: There are two problems discussed in this thesis concerning local time stepping: Instabilities and proper time accurate solutions for unsteady cases. The instabilities were addressed in Section 4.5 where an empirical equation was found for stable CFL numbers and suggestions for stable global time steps. These suggestions should be advanced through further research in search of an appropriate local time step. Tanahashi (1990), Kallinderis (1994), and Jakobsen (2000) should be used as guidelines in this development. Time accuracy in unsteady cases is only ensured if the number of inner cycles times the local time step is greater than or equal to the global time step. This comparison should be made for each node in the solution field in order to flag the user when single occurrences and/or excessive violation of time accuracy is found.

Number of Gauss Points: This research concentrated on the one-point Gauss quadrature when generating aerodynamic coefficient results. Several examples were given to compare the two integration models used by the solvers. Several interesting tests still need to be made on the number of Gauss points that could be generated for both the Euler solver and new Navier-Stokes solver. Using the various test cases of this thesis and others as needed, the three aerodynamic coefficients should be generated for one and three point Gauss integration in both the inviscid and viscous solvers. These tests could then be used to investigate the necessity of the higher order integration model on certain test cases and the sufficiency of the lower order model on other test cases. This investigation should then be expanded to their effects on the aerodynamic forcing function used in the aeroelastic routine to step the structural solver. This thesis also presented the effects of the dissipation model on various Euler results. Similar tests

should be ran to investigate the effects of the higher order integration scheme on dissipation and other "solver" attributes.

Second Viscosity: This thesis presented Stokes' hypothesis as the lower limit to the second coefficient of viscosity. Several experiments have been conducted to increase the understanding of this second coefficient, but these investigations themselves have been brought into question (White, 1991). The overall effects on all output unknowns of the solver should be investigated to better understand the effects of the second coefficient of viscosity. The lower limit, mentioned previously, was used as the default condition of the ratio of viscosity coefficients. The effects of this lower limit on flow conditions near the blunt leading and trailing edges should be tested along with conditions near shocks and any other areas of the flow that have volumetric changes. This default condition should be adjusted as needed and suggestions made for the nominal ranges for the second coefficient of viscosity.

Third dissipation model: The two dissipation models presented by Cowan (2003) and used in this research use artificial dissipation to reduce ill effects in the flow solutions. Both models use the gradients between nodes to generate an appropriate amount of change at each node. The higher order model uses an additional gradient limiter to decrease the smearing of shocks and the effects on actual flow gradients. Regretfully, both models affect the shape and shear stress of boundary layers. A third dissipation model should be generated that does not affect boundary layers but still controls shocks and other flow problems. This model might have a null area, such that gradients lying in this area are not effected by the third model. In order to generate such a model, investigations would need to be pursued to find the limits of the natural flow

gradients in both laminar and turbulent flows. Kallinderis (1995) and Naterer (1994) should be used as guidelines during this investigation.

Second Viscosity Instead of Dissipation: The current dissipation models require much of the needed run time to calculate the contributions of every segment on their corresponding nodes. The overall runtime of both the Euler and Navier-Stokes solvers could be decreased significantly if the dissipation model could be replaced. Combining the two previous recommendations: The second coefficient of viscosity might be capable, within reasonable limits of the natural coefficient, to generate stable shocks without changing the other areas of the flow. If such a capability is available, the second coefficient should be investigated as a replacement for the dissipation models in some or all flows. The second coefficient is already required for several calculations within the new solver, therefore turning off the dissipation model for available test cases could decrease their runtimes significantly. The Navier-Stokes solver could be used to generate Euler results in shorter periods of time using the second coefficient and inviscid solid walls instead of the Euler solver with the dissipation model.

Residual Watching: An appropriate number of inner cycles is often hard to judge when generating an unsteady solution. The same can be said for global iterations for steady solutions. The residual vector could be used to limit both of these problems. As the residual decreases for a steady solutions, the residuals are converging on an appropriate solution for the given flow field and boundary conditions. A limit can be set in order to judge the convergence of the steady solution. When convergence is met, the global iterations would be terminated. Using this system, the number of global iterations could be set extremely high, knowing that the solver will terminate when the residuals

decrease beyond a specified value. Similarly, the number of inner cycles could be set to a very large number and the residual could be used to judge when an unsteady time step has converged to an appropriate value. Residual watching could also be used to safe guard against instability problems. The solver could be instructed to decrease the CFL and repeat a global step in order to generate a stable steady solution. As the instabilities dissipate, the solver could again increase the CFL number to an appropriate value with an upper limit given by the value specified in the control file. A similar but much more complex methodology could be used to adjust the dimensionless time step in order to arrive at a more stable solution. The issue of time accuracy must be kept in mind, and the user input time step must also be met at all loads, residual, and other outputs from the CFD solver. Care must also be taken to handle the structural solver as well. Suggestions to dissipation watching should also be used from Kallinderis (1995).

Improved Far Field Condition: The far field condition used by Cowan (2003) implements Reimann invariants on the far field to weakly adjust the boundary to the free stream conditions. This boundary condition was originally generated to be used at large distances from the test cases in order to bound the flow. In the original assumptions, the effects of the body and/or flow gradients would be greatly reduced across this distance so that a far field assumption could be used. This research has shown that vortices, wakes, and other flow conditions are smeared to some extent by the current far field condition. The effects of the current far field condition on confined outflow conditions (i.e., nozzles, pipe flow, etc.) has not been undertaken. Research is needed to improve the outflow far field condition or to generate a new outflow condition that will not produce numerical

effects near the outflow plane. Srinivas (1984) illustrates a "non-reflecting" boundary condition and the appropriate references.

Engine Boundary Conditions: No engine inlet and outlet boundary conditions currently exist in either the Euler or Navier-Stokes solvers. Such boundary conditions were not the focus of either investigation. In the future these boundary conditions need to be available to the solver. Inlet boundary conditions might include a required mass flow rate seen on the boundary; similar conditions could be used on the outflow plane. These two planes could be assigned indices so that the mass flow rate from one outlet matches the inflow of one or several inlets. Further, to model inlet spill over in choked flow conditions, a maximum mass flow condition could be used on the inlet. This condition would allow any mass flow rate up to a certain value to pass through the inlet plane; any flow rate above this value would be limited so that spill over would occur.

Improve Meshing Routine: This thesis illustrated problems that occur when small segment are generated. The current meshing software starts by generating nodes on the bounds of the flow field. Then the routine uses the marching front method to progress the "front" of unconnected elements to the inside of the flow field, one element at a time. When the front collapses on itself, a node is placed in order to complete the mesh. Several mesh adaptation techniques are used to smooth the mesh as the elements are being placed. Still heterogeneous areas of the mesh are allowed to occur. Investigations need to be made into the control parameters of the current meshing routine to minimize the number of small triangles generated in the mesh. Adaptations to the current routine should be added to help minimize the number of anomalies along the boundaries and within the mesh itself. Suggestions can be found in Burton (1993) and Jameson (1998).

Boundary Layer Remeshing: This research generated several laminar boundary layers and was able to conclude an appropriate number of elements and spacing within the boundary layer for several necessary conditions. These conditions include a good engineering approximation of the elemental shear stress along a surface, a good flow physics approximation of the curvature of the boundary layer profile, and the appropriate aerodynamic loads for the aeroelastic solver; all without applying too many elements and requiring too much run time. Applying these rules of thumb to two-dimensional geometries can be a grueling task using points, lines, and triangles as sources to communicate the necessity for a finer mesh to the meshing software. In the future, this research will be expanded to a full three-dimensional Navier-Stokes solver using the three-dimensional solver of Cowan (2003) as a jumping off point. With three-dimensional surfaces, the task of generating boundary layer meshes will be even more difficult to the point of being impossible for the complex geometry of entire aircraft. One such answer to the meshing problem could be to include the viscous surfaces themselves as sources for finer nodes clusters. The draw back of this method occurs with complex geometries and knowing a predicted height of the boundary layer and direction of growth. Further complications occur when using non-inertial and aeroelastic solutions with a given freestream Reynolds number. Therefore, adaptive remeshing should be explored at the two-dimensional level for viscous gradients near the surface of bodies and in their wakes. This adaptive remeshing should also include the influence of turbulent effects if applicable. The previous method could be used to generate a coarse viscous mesh near surfaces after running an Euler solution of the geometry. The remeshing process could then continue by running a Navier-Stokes solution through an adaptive remesh process

similar to that mentioned above. Further research will be needed to access the problem of a moving surface and morphing boundary layer seen on a aeroelastic surface or body in non-inertial rotation or translation. Suggestions can be taken from Marcum (1996).

BIBLIOGRAPHY

Anderson, D.A.; Tannehill, J.C.; Pletcher, R.H. *Computational Fluid Mechanics and Heat Transfer*. Hemisphere Publishing Company. New York. 1984.

Arminjon, P.; Madrane, A. "Staggered Mixed Finite Volume/Finite Element Method for the Navier-Stokes Equations." *AIAA Journal*. Vol. 37. No. 12. Dec 1999. pp. 1558-1571.

Baker, A.J. *Finite Element Computational Fluid Mechanics*. Hemisphere Publishing Company. New York. 1983.

Barche, J. (Chair) *Experimental Data Base for Computer Program Assessment*. *AGARD-AR-138*. Report of the Fluid Dynamics Panel, Work Group 04. Specialised Printing Services Limited. Loughton, Essex. May 1979.

Barsoum, M.E.; Alexandrou, A.N. "Stable Finite Element Solutions for Fully Viscous Compressible Flows." *Finite Elements in Analysis and Design*. Vol. 19. 1995. pp. 69-87.

Baum, S. Dynamical Oceanography. Part I: Fundamental Principles. http://stommel.tamu.edu/~baum/reid/book1/book/node102.html#SECTION00630000000000000000. December 1, 1997.

Benson, T.J. NASA Glenn Research Center. http://www.grc.nasa.gov/WWW/K-12/airplane/atmosmrm.html. March 4, 2004.

Bertin, J.J.; Smith, M.L. *Aerodynamics for Engineers*. 3rd Ed. Prentice Hall. Upper Saddle River, New Jersey. 1998.

Bonhaus, D.L. *A Higher Order Accurate Finite Element Method for Viscous Compressible Flow*. Dissertation. Virginia Polytechnic Institute and State University. Nov. 1998.

Bristeau, M.O.; Glowinski, R.; Dutto, L.; Periaux, J.; Roge, G. "Compressible Viscous Flow Calculations Using Compatible Finite Element Approximations." *International Journal for Numerical Methods in Fluids*. Vol. 11. 1990. pp. 719-749.

Brodersen, O. "Drag Prediction of Engine-Aircraft Interference Effects Using Unstructured Navier-Stokes Calculations." *Journal of Aircraft*. Vol. 39. No. 6. Nov-Dec 2002. pp. 927-935.

Brown, T.L.; LeMay, H.E.; Bursten, B.E. *Chemistry: The Central Science*. 7th Ed. Prentice Hall. Upper Saddle River, New Jersey. 1997.

Burtin, K.L.; Pepper, D.W. "Use of Adaptive Finite Elements for Compressible Flow." *Journal of Thermophysics and Heat Transfer*. Vol. 7. No. 4. 1993. pp. 744-747.

Chandruptla, T.R.; Belegundu, A. *Introduction to Finite Element in Engineering*. Prentice Hall. 1991.

Chernobrovkin, A.; Lakshminarayana, B. "Development and Validation of Navier-Stokes Procedure for Turbomachinery Unsteady Flow." *AIAA Journal*. Vol. 37. No. 5. May 1999. pp. 557-563.

Cowan, T.J. *Finite Element CFD Analysis of Super-Maneuvering and Spinning Structures*. Dissertation. Oklahoma State University. May 2003.

Darbe, R.P. *Computational Study of Trailing-Edge Geometry on Inlet Guide Vane Forced Response in a Transonic Compressor*. Thesis. Oklahoma State University. May 2004.

Ecer, A.; Akay, H.U. "A Finite Element Method for Solution of Viscous Compressible Flows." *Recent Developments in Computational Fluid Dynamics*. *AMD-Vol. 95*. Ed: Tezduyar, T.E.; Hughes, T.J.R. ASME. 1988.

Elsenaar, A. (Chair*) A Selection of Experimental Test Cases for the Validation of CFD Codes*. *AGARD-AR-303*. Vol. I. Report of the Fluid Dynamics Panel. Canada Communications Group. Quebec, Canada. August 1994.

Elsenaar, A. (Chair*) A Selection of Experimental Test Cases for the Validation of CFD Codes*. *AGARD-AR-303*. Vol. II. Report of the Fluid Dynamics Panel. Canada Communications Group. Quebec, Canada. August 1994.

Erickson, G.E. "High Angle-of-Attack Aerodynamics." *Annual Review of Fluid Mechanics*. Vol. 27. 1995. pp.45-88.

Fernandez, G.; Hafez, M. "Finite Element Solution of Compressible Navier-Stokes Equations." *Advances in Finite Element Analysis in Fluid Dynamics*. FED-Vol. 123. ASME. 1991.

Filippone, A.; Michelsen, J.A. "Aerodynamic Drag Prediction of Helicopter Fuselage." *Journal of Aircraft*. Vol. 38. No. 2. March-April 2001. pp. 326-333.

Gai, S.L.; The, S.L. "Interaction Between a Conical Shock Wave and a Plane Turbulent Boundary Layer." *AIAA Journal*. Vol. 38. No. 5. May 2000. pp. 804-811.

Gefroh, D.; Loth, E.; Dutton, C.; McIlwain, S. "Control of an Oblique Shock/Boundary Layer Interaction with Aeroelastic Mesoflaps." *AIAA Journal*. Vol. 40. No. 12. Dec 2002. pp. 2456-2466.

Ghaffari, F.; Luckring, J.M.; Thomas, J.L.; Bates, B.L; Biedron, R.T. "Multiblock Navier-Stokes Solutions about the F/A-18 Wing-LEX-Fuselage Configuration." *Journal of Aircraft*. Vol. 30. No. 3. May-June 1993. pp. 293-303.

Gottfried, D.A.; Fleeter, S. "Turbomachine Blade Row Interaction Predictions with a Three-Dimensional Finite Element Method." *Journal of Propulsion and Power*. Vol. 18. No. 5. Sept-Oct. 2002. pp. 978-989.

Incropera, F.P.; DeWitt, D.P. *Introduction to Heat Transfer*. 4th Ed. John Wiley & Sons, Inc. New York. 2002.

Jakobsen, L.A.; Moller, H.; Lund, E. "Sensitivity Analysis of Convection Dominated Steady 2D Fluid Flow Using SUPG FEM." *AIAA-2000-4822*. 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization. Long Beach, CA. 2000.

Jameson, A.; Martinelli, L. "Mesh Refinement and Modeling Errors in Flow Simulation." *AIAA Journal*. Vol. 36. No. 5. May 1998. pp. 676-686.

Jiang, B.-N.; Povinelli, L.A. "Least-Squares Finite Element Method for Fluid Dynamics." *Computer Methods in Applied Mechanics and Engineering*. Vol. 81. 1990. pp. 13-37.

John, J.E.A. *Gas Dynamics*. 2nd Ed. Prentice Hall. Englewood Cliffs, New Jersey. 1984.

Kallinderis, Y.; McMorris, H. "Magnitude of Artificial Dissipation for Numerical Simulations." *AIAA Journal*. Vol. 33. No. 8. Aug 1995. pp. 1526-1529.

Kallinderis, Y.; Nakajima, K. "Finite Element Method for Incompressible Viscous Flows with Adaptive Hybrid Grids." *AIAA Journal*. Vol. 32. No. 8. Aug 1994. pp.1617-1625.

Kandil, O.A.; Chuang, H.A. "Unsteady Navier-Stokes Computations Past Oscillating Delta Wing at High Incidence." *AIAA Journal*. Vol. 28. No. 9. Sept 1990. Pp. 1565-1572.

Kwon, O.J.; Sankar, L.N. "Viscous Flow Simulation of a Fighter Aircraft." *Journal of Aircraft*. Vol. 29. No. 5. Sept-Oct 1992. pp. 886-891.

Lapidus, L.; Pinder, G.F. *Numerical Solution of Partial Differential Equations in Science and Engineering*. John Wiley & Sons, Inc. New York. 1982.

Lee, B.H.K. "Oscillatory Shock Motion Caused by Transonic Shock Boundary-Layer Interaction." *AIAA Journal*. Vol. 28. No. 5. May 1990. pp. 942-944.

Lee, Y. "Skin Friction Measurements for Recirculating Normal-Shock/Boundary Layer Interaction Control." *AIAA Journal*. Vol. 42. No. 4. April 2004. pp. 806-814.

Lei, C.; Cheng, L.; Kavanagh, K. "A Finite Difference Solution of Shear Flow over a Circular Cylinder." *Ocean Engineering*. Vol. 27. 2000. pp. 271-290.

Luo, H.; Baum, J.D.; Lohner, R. "High-Reynolds Number Viscous Flow Computations Using an Unstructured-Grid Method." *AIAA-2004-1103*. 42nd AIAA Aerospace Sciences Meeting and Exhibit. Reno, Nevada. 2004.

Marcum, D.L. "Adaptive Unstructured Grid Generation for Viscous Flow Applications." *AIAA Journal*. Vol. 34. No. 11. Nov 1996. pp. 2440-2443.

Marcum, D.L.; Agarwal, R.K. "Finite Element Navier-Stokes Solver for Unstructured Grids." *AIAA Journal*. Vol. 30. No. 3. March 1992. pp. 648-654.

Masud, A.; Hughes, T.J.R. "A Space-Time Galerkin Least-Squares Finite Element Formulation of the Navier-Stokes Equations for Moving Domain Problems." *Computational Methods in Applied Mechanics and Engineering*. Vol. 146. 1997. pp. 91-126.

Mittal, S. "Finite Element Computation of Unsteady Viscous Compressible Flows." *Computational Methods in Applied Mechanics and Engineering*. Vol. 157. 1998. pp. 151-175.

Moran, M.J.; Shapiro, H.N. *Fundamentals of Engineering Thermodynamics*. 3rd Ed. John Wiley & Sons, Inc. New York. 1996.

N'dri, D.; Garon, A.; Fortin, A. "Stable Space-Time Formulation for the Navier-Stokes Equaitons." *AIAA-2000-0394*. 38th AIAA Aerospace Sciences Meeting and Exhibit. 2000.

Nakabayashi, K.; Yoshida, N. "Numerical Analysis for Viscous Shear Flows Past a Circular Cylinder at Intermediate Reynolds Numbers." *JSME International*

Journal, Series B: Fluids and Thermal Engineering. Vol. 36. No. 1. 1993. pp. 34-41.

Naterer, G.F.; Schneider, G.E. "Use of the Second Law for Artificial Dissipation in Compressible Flow Discrete Analysis." *Journal of Thermophysics and Heat Transfer*. Vol. 8. No. 3. July-Sept 1994. pp. 500-506.

Oberkampf, W.L.; Trucano, T.G. "Verification and Validation in Computational Fluid Dynamics." *SAND2002–0529*. Sandia National Laboratories. Albuquerque, New Mexico. 2002.

Oktay, E.; Asma, C.O. "Fast Drag Prediction Method Using Euler Equations." *Journal of Spacecraft and Rockets*. Vol. 37. No. 5. Sept 2000. pp. 692-697.

Panaras, A.G. "Shear-Layer-Edge Interaction: Simulation by Finite-Area Vortices." *AIAA Journal*. Vol. 28. No. 8. Sept 1990. Pp. 1557-1564.

Paparone, L.; Tognaccini, R. "Computational Fluid Dynamics-Based Drag Prediction and Decomposition." *AIAA Journal*. Vol. 41. No. 9. Sept 2003. p. 1647-1657.

Peiro, J.; Peraire, J.; Morgan, K. *FELISA System Reference Manual: Part 2 – User Manual*. December 1993.

Rizk, Y.M.; Gee, K. "Unsteady Simulation of Viscous Flowfield Around F-18 Aircraft at Large Incidence." *Journal of Aircraft*. Vol. 29. No. 6. Nov-Dec 1992. pp. 986-992.

Rogers, S.E.; Roth, K.; Cao, H.V.; Slotnick, J.P.; Whitlock, M.; Nash, S.M.; Baker, M.D. "Computation of Viscous Flow of a Boeing 777 Aircraft in Landing

Configuration." *Journal of Aircraft*. Vol. 38. No. 6. Nov-Dec 2001. pp. 1060-1068.

Selig, M. UIUC Airfoil Data Site. http://www.aae.uiuc.edu/m-selig/ads.html. March 16, 2004.

Shapira, M.; Degani, D.; Weihs, D. "Stability and Existence of Multiple Solutions for Viscous Flow in Suddenly Enlarged Channels." *Computers and Fluids*. Vol. 18. No. 3. 1990. pp. 239-258.

Slater, J.W. NPARC Alliance CFD Verification and Validation Web Site. http://www.grc.nasa.gov/WWW/wind/valid/validation.html. May 14, 2004.

Soulaimani, A.; Fortin, M. "Finite Element Solution of Compressible Viscous Flows Using Conservative Variables." *Computer Methods in Applied Mechanics and Engineering*. Vol. 118. 1994. pp. 319-350.

Srinivas, K.; Fletcher, C.A.J. "A Generalized Coordinate Time-Split Finite Element Method for Compressible Viscous Flow." *Computational Techniques and Applications*. *CTAC-83*. Ed: Noye, J.; Fletcher, C. Elsevier Science Publishers, North-Holland. 1984.

Tanahashi, T.; Okanaga, H.; Saito, T. "GSMAC Finite Element Method for Unsteady Incompressible Navier-Stokes Equations at High Reynolds Numbers." *International Journal for Numerical Methods in Fluids*. Vol. 11. 1990. pp.479-499.

Thomasset, F. *Implementation of Finite Element Methods for Navier-Stokes Equations*. Spinger-Verlag. New York. 1981.

Topol, D.A. "Rotor Wake/Stator Interaction Noise – Predictions vs Data." *Journal of Aircraft*. Vol. 30. No. 5. Sept-Oct 1993. pp. 728-735.

Tworzydlo, W.W.; Oden, J.T.; Thornton, E.A. "Adaptive Implicit/Explicit Finite Element Method for Compressible Viscous Flow." *Computer Methods in Applied Mechanics and Engineering*. Vol. 95. 1992. pp. 397-440.

Van Dyke, M. *Album of Fluid Motion*. Parabolic Press. Stanford, CA. 1982.

Wadia, A.R.; Szucs, P.N.; Crall, D.W. "Inner Workings of Aerodynamic Sweep." *Journal of Turbomachinery*. Vol. 120. No. 4. Oct. 1998. pp. 671-682.

White, F.M. *Viscous Fluid Flow*. 2nd Ed. McGraw-Hill. Boston. 1991.

Whiting, C.H. *Stablilized Finite Element Methods for Fluid Dynamics using a Hierarchical Basis*. Dissertation. Rensselaer Polytechnic University. Troy, New York. Sept. 1999.

Williamson, C.H.K.; Brown, G.L. "A Series of 1/Root Re to Represent the Strouhal-Reynolds Number Relationship to the Cylinder Wake." *Journal of Fluids and Structures*. Vol. 12. 1998. pp. 1073-1085.

Wood, B.; Loth, W.; Geubelle, P.; McIlwain, S. "A Numerical Methodology for an Aeroelastic SBLI Flow." *AIAA-2000-0552*. 38th Aerospace Sciences Meeting and Exhibit. Reno, Nevada. 2000.

APPENDIX A: Summary of File Formats

This appendix provides a summary of the input and output file formats used for this research. This appendix is similar in layout and content to that of Cowan (2003) but has been modified to include the new terms, definitions, and notes appropriate to the modifications of this research.

The following input files are defined:

- case.fro (reuquired) contains the surface mesh generated by *surface* or some other meshing software. Resorted by *makeg2d* in order to generate the *.g2d file (ASCII) (See Peiro, 1993 for details)
- case.bco (required) contains the boundary conditions and singular point conditions needed to resort the *.fro file into the *.g2d file (ASCII)
- case.g2d (required) contains the geometry data structures representing the computational mesh as required by the flow solver (ASCII)
- case.con (required) contains values for the solver control parameters and flow conditions (ASCII)
- case.unk (optional) contains the nodal values of the primitive flow variables (density, velocity, and pressure) for each node of the computational mesh to be used as the initial conditions for the flow solution (Binary)

The following output files are defined:

- case.un# contains the nodal values of the primitive flow variables (density, velocity, and pressure) for each node of the computational mesh (Binary)
- case.rsd contains a history of the solution residuals for the conservation variables (density, momentum, and total energy) (ASCII)
- case.lds contains a history of the dimensionless aerodynamic forces acting on the solid walls of the geometry (ASCII)

Boundary Conditions Input File (`case.bco`)

Basic File Format

```
Line of Text
  ncv
Line of Text
  (bccv(i,1), bccv(i,2), i = 1, ncv)
Line of Text
  (sccv(i,1), sccv(i,2), i = 1, ncv)
```

Definition of Terms

`ncv`: (int) number of curves (in *.fro file)
`bccv(i,1)`: (int) index for curve *i*
`bccv(i,2)`: (int) boundary condition for curve *i*
= 1, inviscid wall condition
= 2, symmetry plane condition
= 3, far field condition
= 4, far field condition
= 9, viscous wall condition
`sccv(i,1)`: (int) index for curve *i*
`sccv(i,2)`: (int) singular condition for curve *i*
= 0, no singular nodes
= 2, first and last nodes singular
= 3, only first node singular
= 4, only last node singular

Geometry Input File (`case.g2d`)

Basic File Format

```
Line of Text
  nnd nel nsg nbe nbp nwl nsd nwlv nsdv
Line of Text
  (LBE(i), i = 1, 8)
Line of Text
  COOR(i,1) COOR(i,2)
                          ( i = 1,…,nnd )
Line of Text
  IELM(i,1) IELM(i,2) IELM(i,3)
                          ( i = 1,…,nel )
Line of Text
  ISEG(i,1) ISEG(i,2)
                          ( i = 1,…,nsg )
Line of Text
  IBEL(i,1) IBEL (i,2)
                          ( i = 1,…,nbe )
Line of Text
  IBEL(i,3)
                          ( i = 1,…,nbe )
```

Definition of Terms

`nnd`: (int) number of nodes
`nel`: (int) number of elements
`nsg`: (int) number of segments
`nbe`: (int) number of boundary elements
`nbp`: (int) number of boundary points
`nwl`: (int) number of wall nodes
`nsd`: (int) number of singular nodes
`nwlv`: (int) number of viscous wall nodes
`nsdv`: (int) number of viscous singular nodes
`LBE(i)`: (int) boundary element starting/stopping indexes for four BC types
`COOR(i,1)`: (real) x-coordinate for node i
`COOR(i,2)`: (real) y-coordinate for node i
`IELM(i,1)`: (int) node 1 for element i
`IELM(i,2)`: (int) node 2 for element i
`IELM(i,3)`: (int) node 3 for element i
`ISEG(i,1)`: (int) node 1 for segment i
`ISEG(i,2)`: (int) node 2 for segment i
`IBEL(i,1)`: (int) node 1 for boundary elem. i
`IBEL(i,2)`: (int) node 2 for boundary elem. i
`IBEL(i,3)`: (int) index for triangle elem.

Comments

- This is a plain text (ASCII) file.
- Nodal data is sorted such that the first `nwl` nodes are defined as solid wall nodes. Out of the first `nwl` nodes, the last `nsd` nodes are defined as singular nodes. The middle `nwlv` nodes are defined as viscous wall nodes and segmented by `LBE`.
- The nodal coordinates in this file are treated as dimensional values and are non-dimensionalized using the reference dimension specified in the solver control file.
- The element connectivity data must define elements that are oriented clockwise.
- Boundary element data is sorted based on the starting/stopping indexes for the three BC types, i.e. boundary elements `LBE(1)` through `LBE(2)` are solid wall elements, `LBE(3)` through `LBE(4)` are symmetry elements, `LBE(5)` through `LBE(6)` are far-field elements, and `LBE(7)` through `LBE(8)` are viscous solid wall elements. Note that $\texttt{LBE(1)} \leq \texttt{LBE(7)} < \texttt{LBE(8)} \leq \texttt{LBE(2)}$.
- The program `makeg2d` is used to convert a standard STARS surface triangulation file and modified boundary conditions file into an appropriately sorted two-dimensional geometry file.

Sample File

```
$  nnd,  nel,  nsg,  nbe,  nbp,  nwl,  nsd, nwlv, nsdv
     8     6    13     8     8     3     0     2     0
$ LBE(8)
     1     2     3     2     3     8     2     2
$ Nodal coordinates
 -.100000E+01 -.100000E+01
 0.100000E+01 -.100000E+01
 0.000000E+00 -.100000E+01
 0.100000E+01 0.100000E+01
 -.100000E+01 0.100000E+01
 0.100000E+01 0.000000E+00
 0.000000E+00 0.100000E+01
 -.100000E+01 0.000000E+00
$ Element connectivity
     1     3     8
     3     2     6
     5     8     7
     6     4     7
     8     3     6
     6     7     8
$ Segment connectivity
     1     3
     1     8
     2     3
     2     6
     3     8
     3     6
     4     6
     4     7
     5     8
     5     7
     6     7
     6     8
     7     8
$ Boundary edge data
     1     3
     3     2
     2     6
     6     4
     4     7
     7     5
     5     8
     8     1
$ Boundary edge element indices data
     2
     2
     4
     4
     3
     3
     1
```

Solver Control Input File (`case.con`)

Basic File Format

```
&control
      dt          = 0.1d0,
      gamma       = 1.4d0,
      diss        = 1.0d0,
      cfl         = 0.5d0,
      lamb        = -0.6d0,
      smod        = 2.45d-3,

      mach        = 0.6d0,
      Re          = 9.0d4,
      Pr          = 0.7d0,
      alpha       = 0.0d0,
      refdim      = 1.0d0,

      nstp        = 100,
      nout        = 50,
      ncyc        = 3,
      isol        = 0,
      idiss       = 0,
      ipnt        = 1,

      istrt       = .false.,
      iaero       = .false.,
      idynm       = .false.,
      ielast      = .false.,
      ifree       = .true.,
      iforce      = .false.,

      nr          = 0,
      ainf        = 1.0d0,
      rhoinf      = 1.0d0,
/
```

Definition of Terms

`dt`: (real) dimensionless global time step
`gamma`: (real) ratio of specific heats
`diss`: (real) dissipation factor or constant
`cfl`: (real) local time step stability factor
`lamb`: (real) ratio of second to first viscosity
`smod`: (real) modified Sutherland's constant

`mach`: (real) free-stream Mach number
`Re`: (real) free-stream Reynolds number
`Pr`: (real) Prandtl number for entire flow
`alpha`: (real) free-stream angle of attack
`refdim`: (real) reference dimension

`nstp`: (int) total solution steps
`nout`: (int) output frequency, steps/output
`ncyc`: (int) iterative cycles per solution step
`isol`: (int) CFD solution type
`idiss`: (int) dissipation type
`ipnt`: (int) number of points for numerical integration of flux/source vectors

`istrt`: (logical) restart flag
`iaero`: (logical) aerodynamic forces flag
`idynm`: (logical) dynamic/non-inertial flag
`ielast`: (logical) elastic flag
`ifree`: (logical) free-stream velocity flag
`iforce`: (logical) external force flag

`nr`: (int) number of elastic modes
`ainf`: (real) dimensional free-stream sonic speed
`rhoinf`: (real) dimensional free-stream density

Comments

- This is a plain text (ASCII) file formatted as a Fortran namelist.
- The default values for each parameter are given in the basic file format above. Any term not appearing in the file is set to the default value by the solver. Any term can be "commented out" by placing an exclamation mark (`!`) before the name of the term.
- The global time step is only used for unsteady solutions.
- Appropriate values for the dissipation factor are in the range $0.0 < \texttt{diss} \leq 2.0$. Some dissipation is required to stabilize the solution, but too much dissipation will corrupt the solution and possibly be a destabilizing influence.
- The local time step stability factor is a safety factor used to compute local time steps for each solution step. For steady solutions, a stability factor of 0.8 is typically acceptable for most problems. For unsteady solutions, the stability factor is typically in the range $0.3 \leq \texttt{cfl} \leq 0.8$.

- The values of `refdim`, `mach`, `ainf`, and `rhoinf` are used to non-dimensionalize all values read in by the flow solver.
- The values of `Re` and `Pr` are used by the flow solver as non-dimensional forms of free-stream viscosity and thermal conductivity. Reynolds number `Re` is input at its free-stream value; Prandtl number `Pr` is constant for the flow field.
- Variable viscosity is handled through Sutherland's equation with relation to free-stream values. The modified Sutherland's constant `smod` is non-dimensionalized by c_v and the free-stream velocity, given by $c_v S/U_0^2$.
- Constant viscosity can be specified by `smod` = 0.0d0. Any positive value of `smod` is considered to specify a variable viscosity condition.
- `lamb` is the constant ratio of second viscosity to first viscosity. The ratio of viscosities is given by the following equation: `lamb` = λ/μ. In order to maintain a positive viscous dissipation (not create energy with viscosity), `lamb` must be less than $-2/3$.
- The free-stream angle of attack is ignored for dynamic (non-inertial) problems.
- The number of iterative cycles should be set to 3 for steady solutions. For unsteady solutions, use a sufficient number of cycles to allow for an appropriate level of convergence at each step.
- There are four available CFD solution types defined as follows:
  - `isol` = 0 is a steady solution (not time accurate)
  - `isol` = 1 is a first-order unsteady solution
  - `isol` = 2 is a second-order unsteady solution
  - `isol` = 3 is a supersonic piston perturbation solution
- There are two available dissipation types defined as follows:
  - `idiss` = 0 is a low order dissipation
  - `idiss` = 1 is a high order dissipation with gradient limiters
- The low-order dissipation is typically overly diffuse and should be used in conjunction with low values of the dissipation factor. Low-order dissipation works best for problems without strong vortices and for supersonic/hypersonic flows.
- The high-order dissipation is more CPU intensive than the low-order dissipation and less stable. Larger values for the dissipation factor are typically required for stabilization. The high-order dissipation works best for subsonic to transonic flows with strong gradients or vortices. Rotating domains will typically require high-order dissipation to resolve the circulating pattern of the relative flow velocities.
- There are two types of numerical integration defined as follows:
  - `ipnt` = 1 uses a one-point gauss quadrature
  - `ipnt` = 3 uses a three-point symmetric gauss quadrature
- When the restart flag is set to `.true.`, the solver will read one set of solution unknowns from the *`case`*`.unk` file and apply this set of unknowns as the initial conditions for the new iterative solution.
- A restarted solution assumes that the time gradient of the initial state is zero, i.e. the solution stored in the *`case`*`.unk` file is a converged, steady state solution. This has a significant impact on the second-order unsteady solution since it relies on two sets of solution unknowns for advancement to the next time step, i.e. a second-order unsteady solution should not be restarted from the last time step of a similar unsteady

solution that was stopped because both sets of unsteady data from the last solution step are not available for accurate evaluation of the time gradients in the flow.

- If the free-stream velocity flag is set to `.false.`, the free-stream velocity is set to zero, and relative flow velocities must be generated through dynamic rotation or translation of the non-inertial coordinate system.
- If the external force flag is set to `.true.`, the solver will read the user defined external force vector for each global time step from the input file *`case`*`.frc`. If the solver reaches the end of the input file before completing the solution, the last force vector in the file carries over to each of the remaining global time steps if it was non-zero.

Unknowns (Initial Conditions) Input File (`case.unk`)

Basic File Format

nnd gam xmi alp ref t Re

((UN(i,j), i = 1,…,nnd), j = 1,…,5)

Definition of Terms

`nnd:` (int) number of nodes
`gam:` (real) ratio of specific heats
`xmi:` (real) freestream Mach number
`alp:` (real) freestream angle of attack
`ref:` (real) reference dimension
`t:` (real) dimensionless time
`Re:` (real) freestream Reynolds number

`UN(i,1):` (real) density for node *i*
`UN(i,2):` (real) x-velocity for node *i*
`UN(i,3):` (real) y-velocity for node *i*
`UN(i,4):` (real) pressure for node *i*
`UN(i,5):` (real) enthalpy for node *i*

Comments

- This is an unformatted (binary) file.
- The solution unknowns stored in this file are dimensionless quantities.
- For dynamic (non-inertial) problems, the solution unknowns stored in the file are relative quantities referenced to the body-fixed coordinate system.

Unknowns Output File (`case.un#`)

Basic File Format

nnd gam xmi alp ref t Re

((UN(i,j), i = 1,…,nnd), j = 1,…,5)

Definition of Terms

`nnd:` (int) number of nodes
`gam:` (real) ratio of specific heats
`xmi:` (real) freestream Mach number
`alp:` (real) freestream angle of attack
`ref:` (real) reference dimension
`t:` (real) dimensionless time
`Re:` (real) freestream Reynolds number

`UN(i,1):` (real) density for node *i*
`UN(i,2):` (real) x-velocity for node *i*
`UN(i,3):` (real) y-velocity for node *i*
`UN(i,4):` (real) pressure for node *i*
`UN(i,5):` (real) enthalpy for node *i*

Comments

- This is an unformatted (binary) file.
- The solution unknowns stored in this file are dimensionless quantities.
- For dynamic (non-inertial) problems, the solution unknowns stored in the file are relative quantities referenced to the body-fixed coordinate system.
- The name of this file is iterated for each output file: *.un1, *.un2, … , *.un10, …

Solution Residuals Output File (`case.rsd`)

Basic File Format

1	(RSD(i), i = 1,…,4)
⋮	⋮
istp	(RSD(i), i = 1,…,4)
⋮	⋮
nstp	(RSD(i), i = 1,…,4)

Definition of Terms

`istp`: (int) current solution step
`nstp`: (int) total or last solution step

`RSD(1)`: (real) density solution residual
`RSD(2)`: (real) x-momentum solution residual
`RSD(3)`: (real) y-momentum solution residual
`RSD(4)`: (real) energy solution residual

Comments

- This is a plain text (ASCII) file.
- For steady problems, the solution residuals indicate the degree of convergence to the final steady state solution. All four solution residuals should converge to approximately the same order of magnitude.
- For unsteady problems, the solution residuals indicate the degree of convergence for each global step of the solution, or the degree of convergence for the steady solution that is solved at each step.

Sample File

1	0.38320E-05	0.10743E-04	0.69854E-05	0.10598E-03
2	0.20317E-05	0.50694E-05	0.40436E-05	0.56307E-04
3	0.12024E-05	0.35187E-05	0.26241E-05	0.32195E-04
4	0.91334E-06	0.25166E-05	0.23637E-05	0.24240E-04
5	0.73183E-06	0.19442E-05	0.22228E-05	0.19376E-04
6	0.59870E-06	0.16179E-05	0.20889E-05	0.15963E-04
7	0.51663E-06	0.14311E-05	0.19719E-05	0.13946E-04
8	0.44924E-06	0.12989E-05	0.18536E-05	0.12398E-04
9	0.39510E-06	0.12095E-05	0.17283E-05	0.11156E-04
10	0.34726E-06	0.11478E-05	0.15878E-05	0.99450E-05
11	0.30775E-06	0.10746E-05	0.14329E-05	0.88159E-05
12	0.26207E-06	0.98700E-06	0.12833E-05	0.76280E-05
13	0.22418E-06	0.87924E-06	0.11245E-05	0.65113E-05
14	0.18904E-06	0.77764E-06	0.98148E-06	0.54617E-05
15	0.15809E-06	0.69345E-06	0.84471E-06	0.44739E-05
16	0.13411E-06	0.62203E-06	0.72991E-06	0.37422E-05
17	0.11564E-06	0.55717E-06	0.64350E-06	0.32661E-05
18	0.10516E-06	0.50502E-06	0.57520E-06	0.30152E-05
19	0.10101E-06	0.46193E-06	0.53100E-06	0.29279E-05
20	0.98711E-07	0.43618E-06	0.49934E-06	0.28901E-05
⋮	⋮	⋮	⋮	⋮

Aerodynamic Loads Output File (`case.lds`)

Basic File Format

0	0.0	(FD(i), i = 1,…,3)
⋮	⋮	⋮
istp	t_{istp}	(FD(i), i = 1,…,3)
⋮	⋮	⋮
nstp	t_{nstp}	(FD(i), i = 1,…,3)

Definition of Terms

`istp`: (int) current solution step
`nstp`: (int) total or last solution step
t_i: (real) dimensionaless time at step *i*

`FD(1)`: (real) x-force coefficient
`FD(2)`: (real) y-force coefficient
`FD(3)`: (real) moment coefficient

Comments

- This is a plain text (ASCII) file.
- The force coefficients in this output file are dimensionless values based on the reference conditions specified in the solver control file.
- For dynamic (non-inertial) problems, the force coefficients stored in this file are referenced to the body-fixed coordinate system.

Sample File

0	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
1	0.10000E+00	0.29186E+01	0.40893E-04	0.20386E-04
2	0.20000E+00	0.53878E+01	0.74036E-04	0.36920E-04
3	0.30000E+00	0.73426E+01	0.10798E-03	0.53867E-04
4	0.40000E+00	0.87754E+01	0.14611E-03	0.72920E-04
5	0.50000E+00	0.97221E+01	0.18835E-03	0.94033E-04
6	0.60000E+00	0.10250E+02	0.23574E-03	0.11772E-03
7	0.70000E+00	0.10444E+02	0.26257E-03	0.13112E-03
8	0.80000E+00	0.10391E+02	0.25348E-03	0.12657E-03
9	0.90000E+00	0.10173E+02	0.25302E-03	0.12632E-03
10	0.10000E+01	0.98630E+01	0.23377E-03	0.11668E-03
11	0.11000E+01	0.95136E+01	0.20482E-03	0.10219E-03
12	0.12000E+01	0.91543E+01	0.19370E-03	0.96640E-04
13	0.13000E+01	0.88118E+01	0.23365E-03	0.11662E-03
14	0.14000E+01	0.85116E+01	0.28553E-03	0.14256E-03
15	0.15000E+01	0.82553E+01	0.37539E-03	0.18747E-03
16	0.16000E+01	0.80367E+01	0.55544E-03	0.27749E-03
17	0.17000E+01	0.78461E+01	0.76662E-03	0.38306E-03
18	0.18000E+01	0.76747E+01	0.10095E-02	0.50449E-03
19	0.19000E+01	0.75147E+01	0.12664E-02	0.63292E-03
20	0.20000E+01	0.73607E+01	0.15058E-02	0.75262E-03
⋮	⋮	⋮	⋮	⋮

APPENDIX B: Investigation of Constant vs. Variable Properties

This appendix provides all of the graphs used to study the change in properties across a normal shock at various Mach numbers from Sections 2.4 and 2.5. Figures B-1 and B-2 show the standard pressure and temperature at various altitudes above the surface of the Earth. Figures B-3 and B-4 show the corresponding pressures and temperatures above the surface of Mars, as presented on the NASA Glenn Research Center web site. Figures B-5 through B-8 give the thermodynamic properties of both air and carbon dioxide (Marsian atmosphere). Figures B-9 through B-14 show the percent change in specific heat, the ratio of specific heats, and the coefficient of viscosity. Each percent change in property is shown for both Earth and Martian at various altitudes and Mach numbers. Mach numbers range from Mach 1.5 to 3.5 in steps of 0.5, and altitudes range from 0 to 80 kilometers from the surface of either planet.

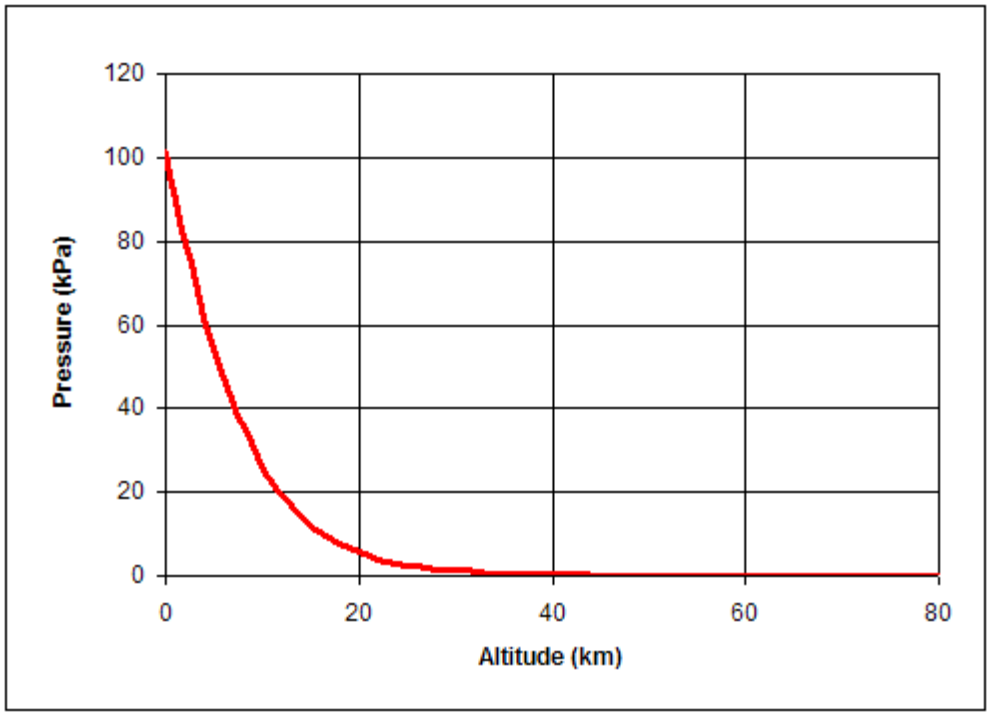

Figure B-1: Variation of Pressure in Standard Earth Atmosphere (Bertin, 1998)

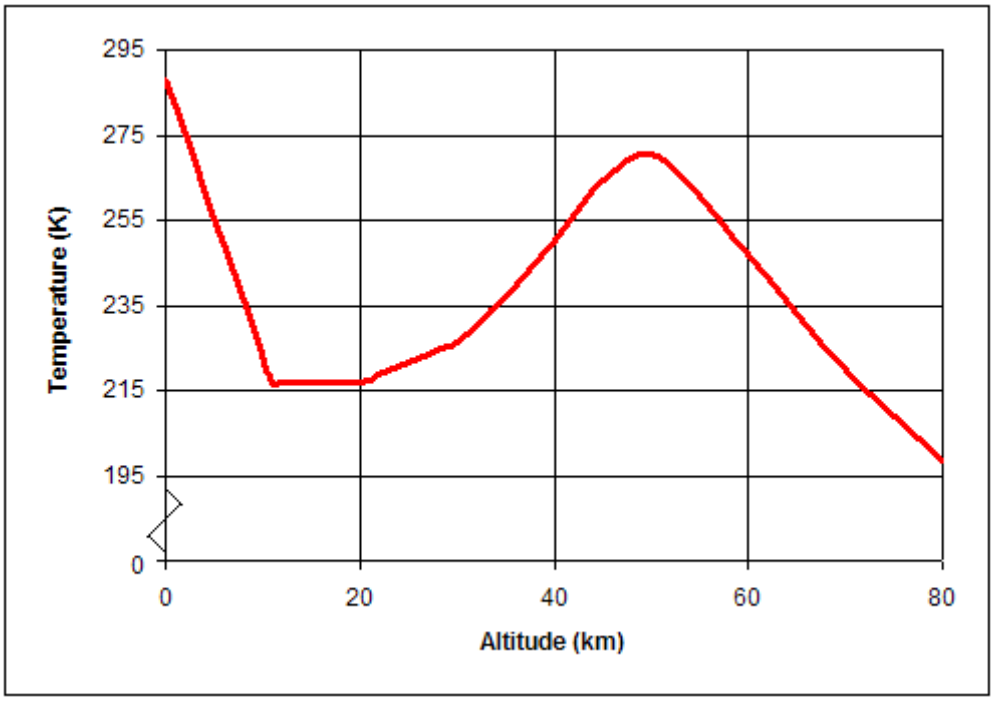

Figure B-2: Temperature Variation in Standard Earth Atmosphere (Bertin, 1998)

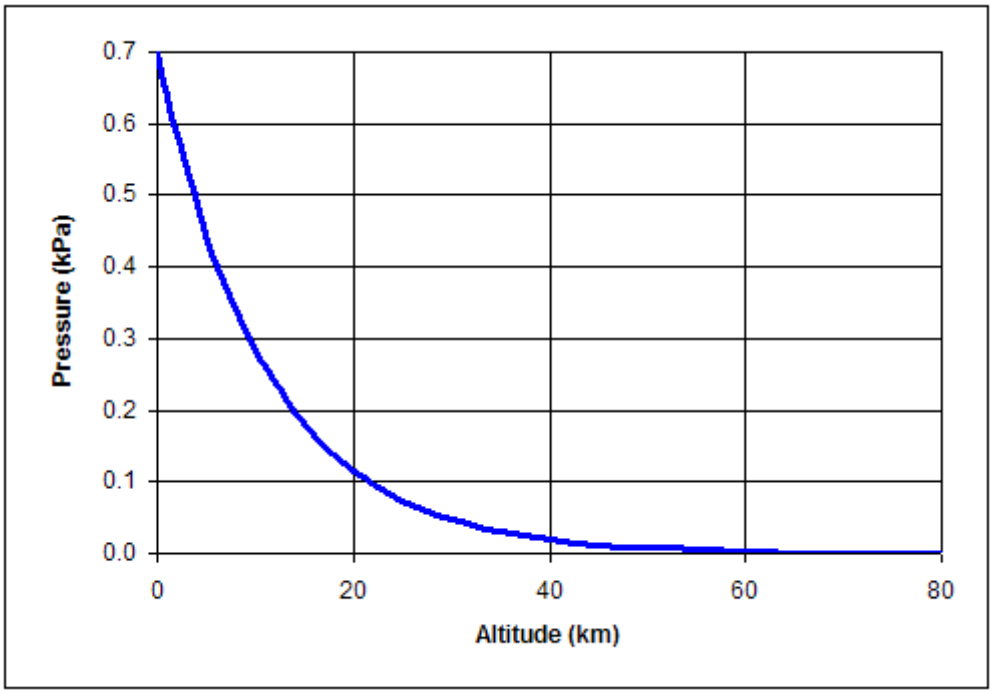

Figure B-3: Pressure Variation in Martian Atmosphere (Benson, 2004)

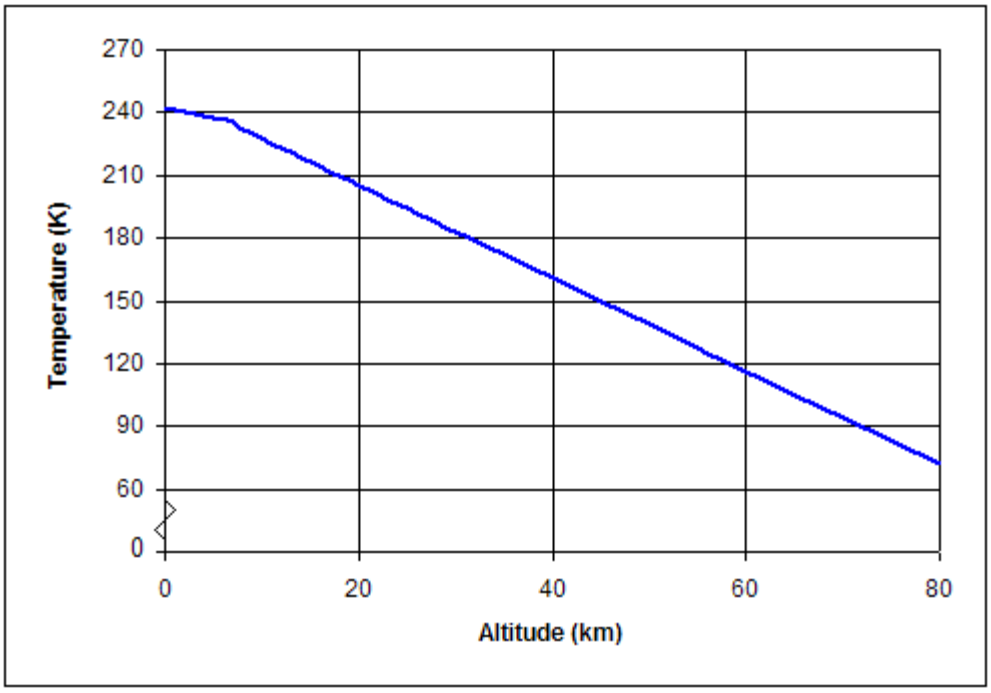

Figure B-4: Temperature Variation in Martian Atmosphere (Benson, 2004)

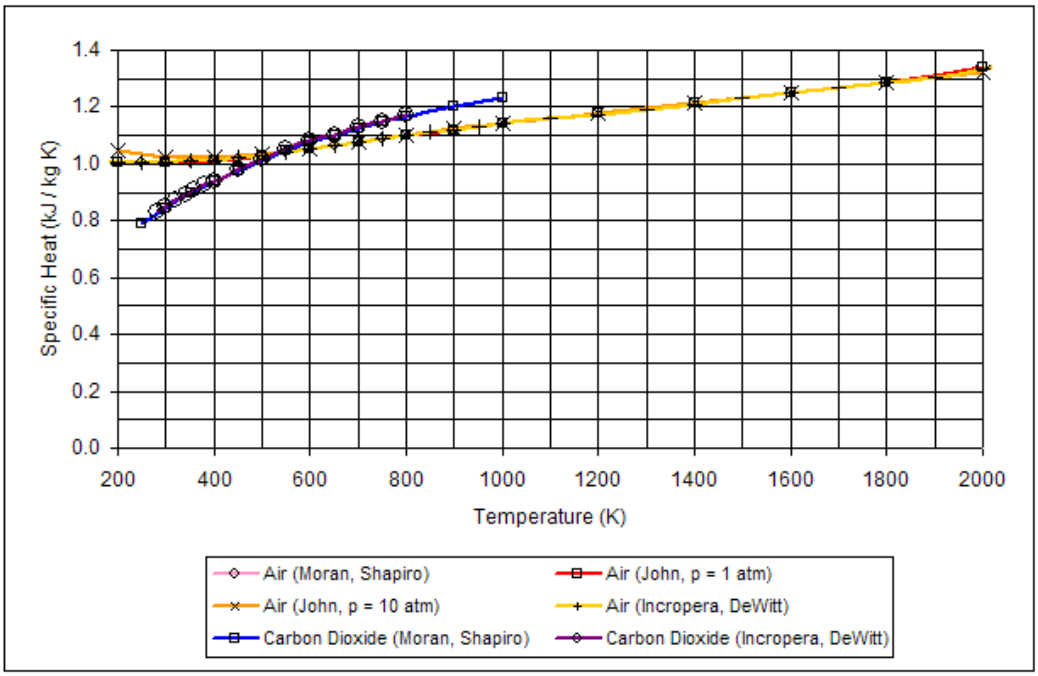

Figure B-5: Variation of Specific Heat at Constant Pressure for Air and Carbon Dioxide (John, 1984; Moran, 1996; Incropera, 2002)

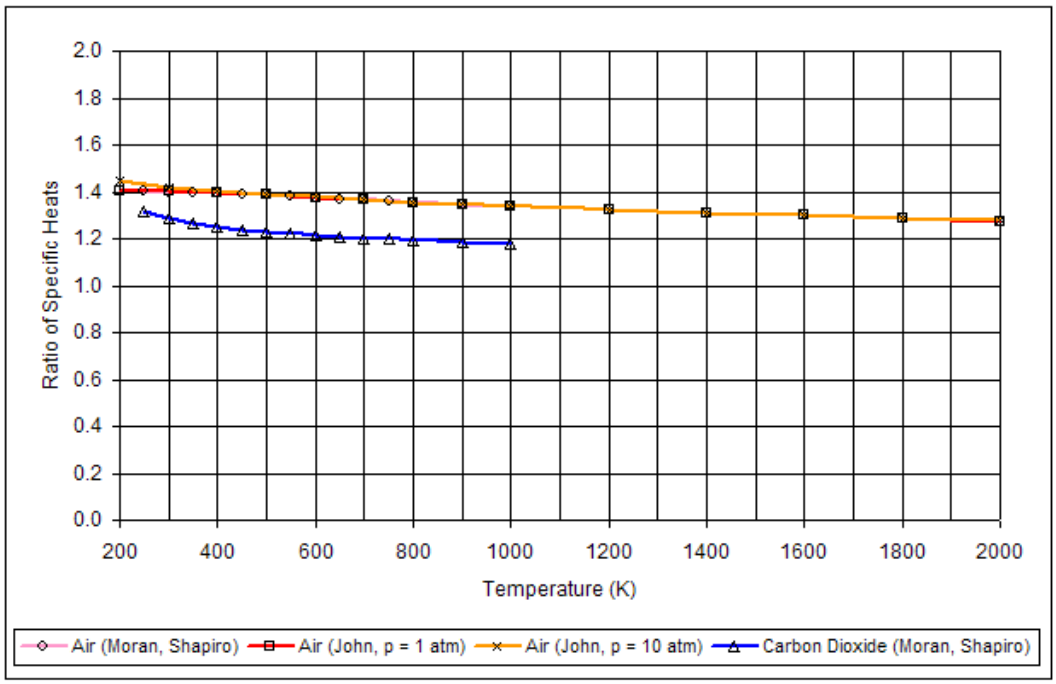

Figure B-6: Variation of the Ratio of Specific Heats for Air and Carbon Dioxide (John, 1984; Moran, 1996)

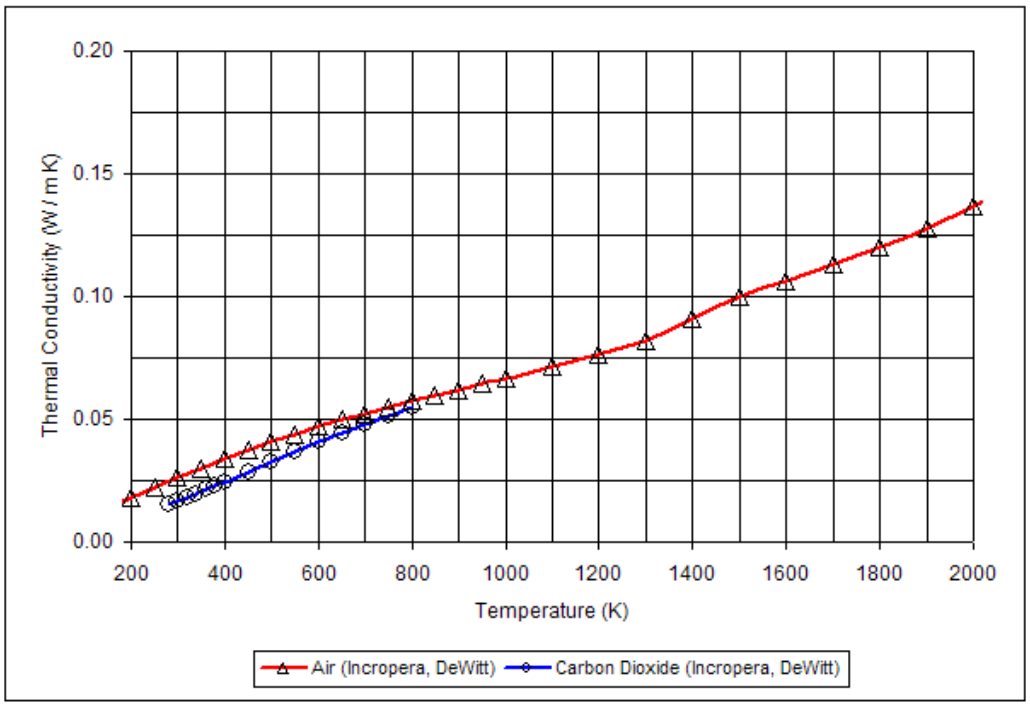

Figure B-7: Variation of the Thermal Conductivity for Air and Carbon Dioxide (Incropera, 2002)

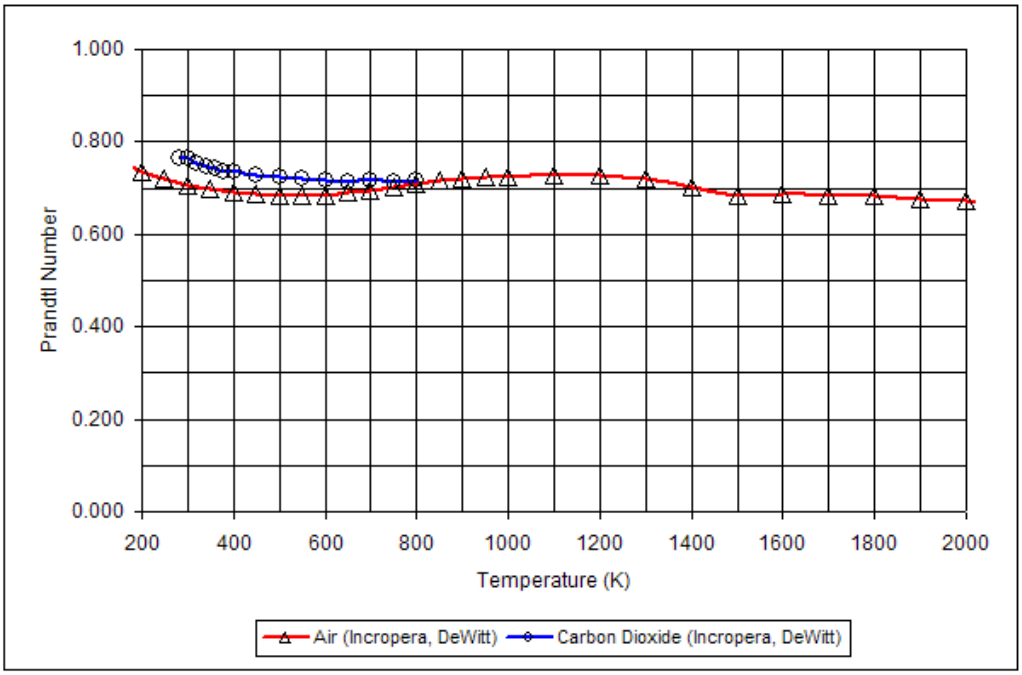

Figure B-8: Variation of the Prandtl Number for Air and Carbon Dioxide (Incropera, 2002)

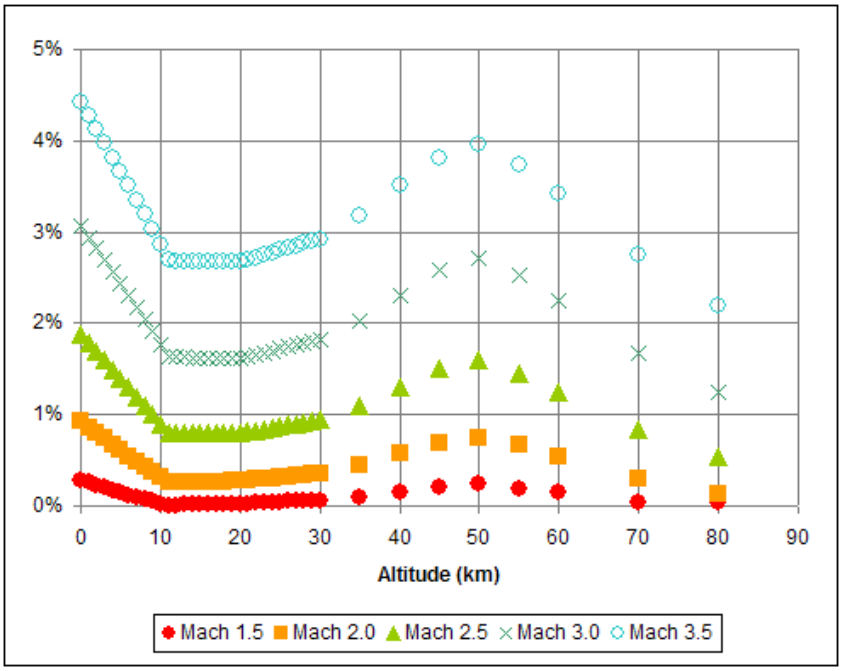

Figure B-9: Percent Increase in Specific Heat across a Normal Shock in Earth Atmosphere

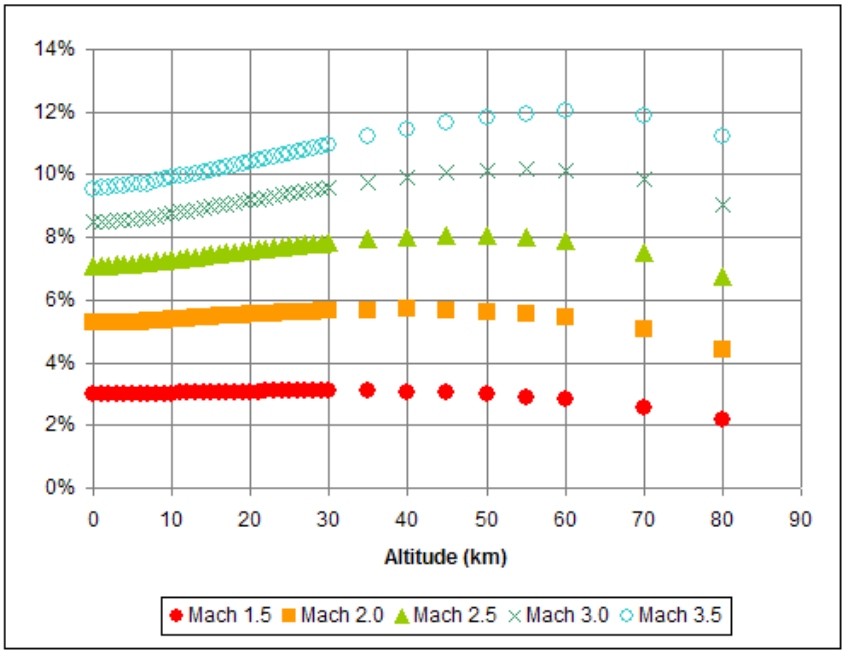

Figure B-10: Percent Increase in Specific Heat across a Normal Shock in Martian Atmosphere

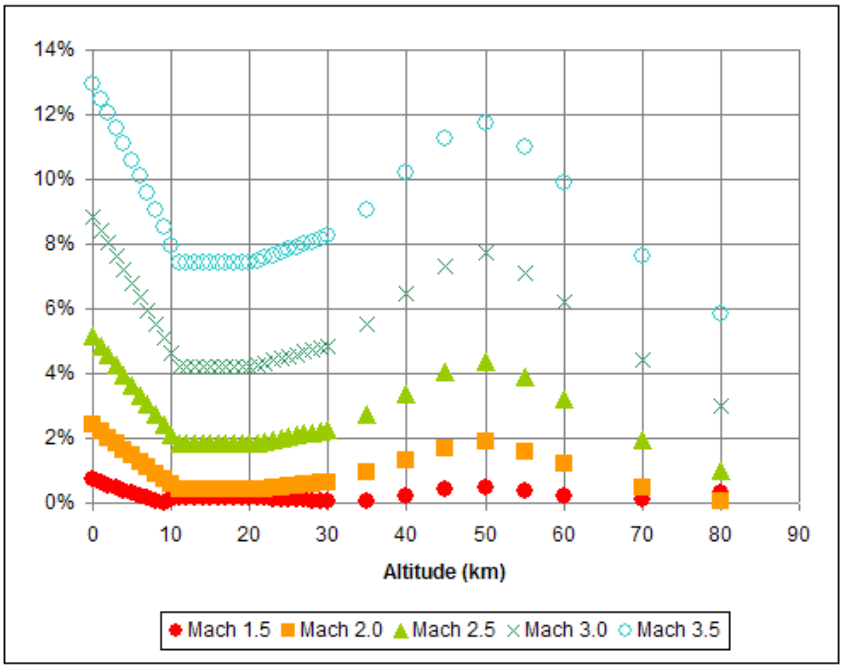

Figure B-11: Percent Increase in the Ratio of Specific Heats across a Normal Shock in Earth Atmosphere

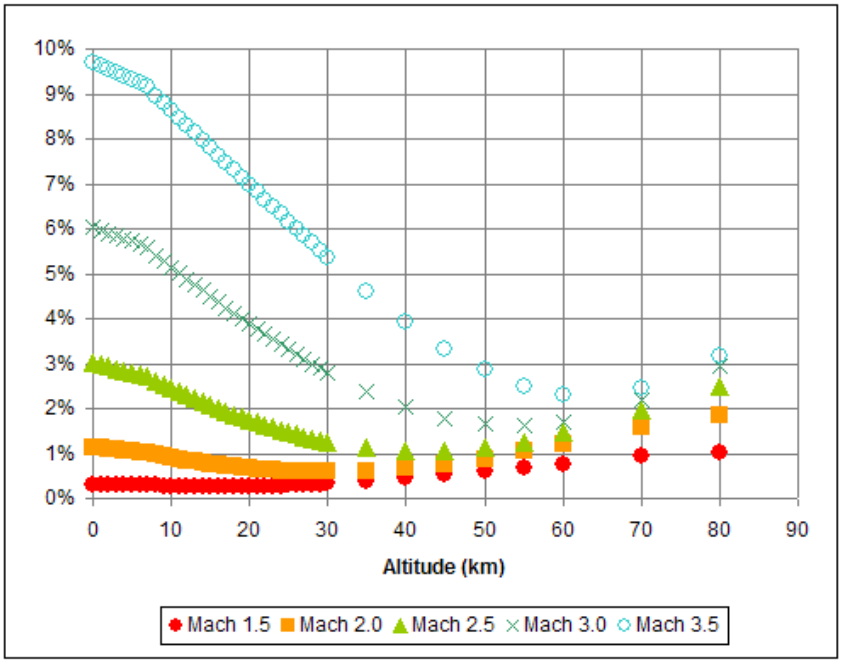

Figure B-12: Percent Increase in the Ratio of Specific Heats across a Normal Shock in Martian Atmosphere

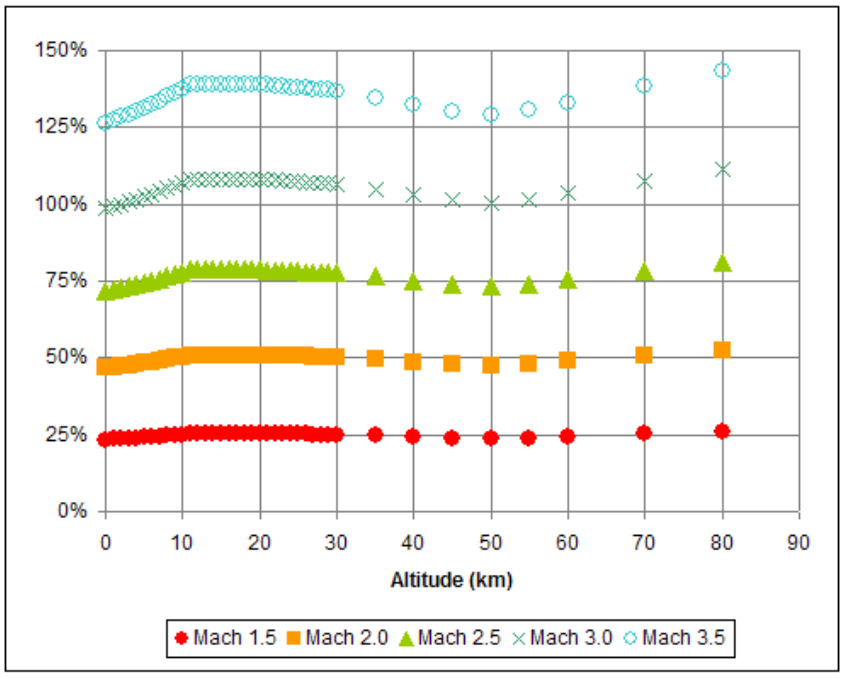

Figure B-13: Percent Increase in the Coefficient of Viscosity across a Normal Shock in Earth Atmosphere

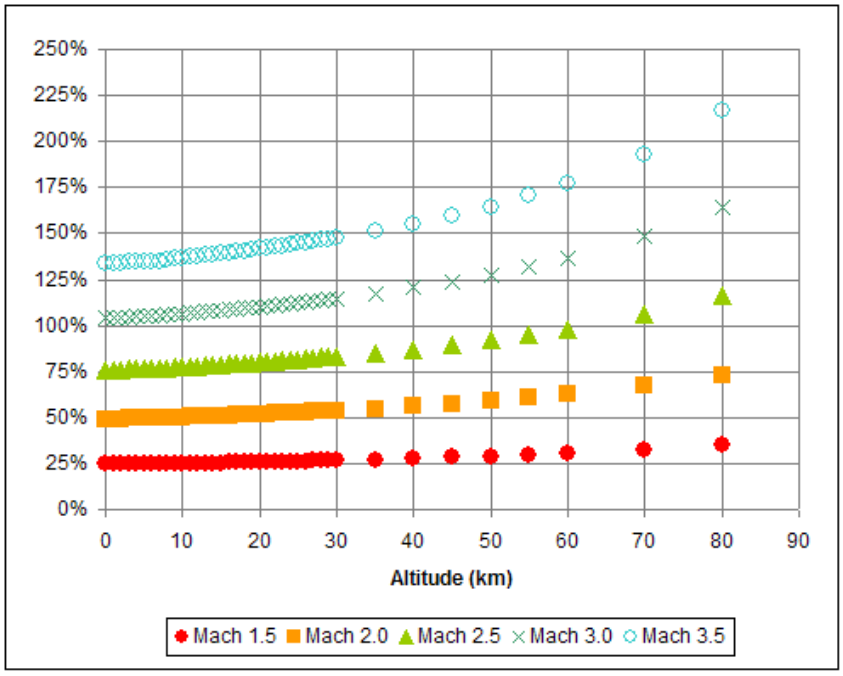

Figure B-14: Percent Increase in the Coefficient of Viscosity across a Normal Shock in Martian Atmosphere

APPENDIX C: Development of Off-Axis Shear Stress Vector

This appendix develops the shear stress vector in several simplified and the off-axis orientation. The development begins with the purely horizontal and vertical solid surface cases. These two cases can be later used to verify that the off-axis shear stress equation works correctly. The text then develops the transformation matrices for the velocity and gradients between the on- and off-axis orientations. The shear stress vector is then assembled in the off-axis, or local coordinate frame, and then rotated back into the on-axis, or global frame of the flow field.

The magnitude of the shear stress on a surface is given by the following scalar:

$$\tau_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{C-1}$$

For a surface which lies only in the *x*-direction, the velocity into the surface is always zero (i.e., $v = 0$). Because all of the *v*-velocities along the surface are zero, the *x*-derivative of the *v*-velocity will also always be zero. The remaining terms of the shear stress equation give:

$$\tau_{xy} = \mu\frac{\partial u}{\partial y} \tag{C-2}$$

which is the form generally used when approaching viscous flow over a flat plate at zero angle of attack. From this equation, the shear stress is positive if the *u*-velocity is in the positive *x*-direction and has a positive gradient in the positive *y*-direction, or both are negative. The shear stress is negative if either the velocity or gradient is positive and the other is negative.

If the surface is vertical (i.e., only lies in the *y*-direction), similar assumptions to the *u*-velocity and the *y*-derivative of the *u*-velocity, resulting in the following equation:

$$\tau_{xy} = \mu\frac{\partial v}{\partial x} \tag{C-3}$$

If the *v*-velocity is positive and decreases in the positive *x*-direction, the shear stress will be negative. Likewise, if the *v*-velocity is negative and increases in the positive *x*-direction, the shear stress will also be negative; otherwise, the shear stress is positive.

For the general case of a surface at some angle of incidence with respect to the *x*-direction, the shear stress must be formulated in the local frame of reference. The previous two equations were developed in the global reference frame. The general case

of the shear stress should be checked using the vertical surface (i.e., ninety-degree rotation from global to local reference frames). The rotation from the global to local reference frames is given by:

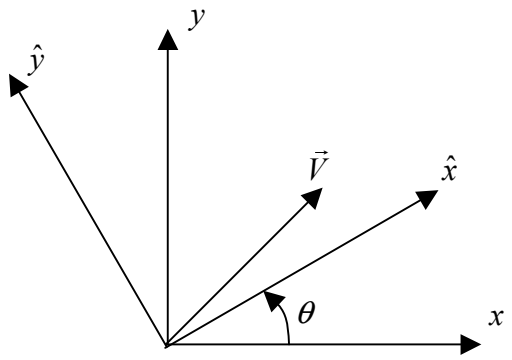

Figure C-1: Diagram of Global and Local Axes

The conversion from the global to local coordinate frames and the reverse can be generated by the following matrix mathematics:

$$\text{(C-4)} \qquad \begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix} \begin{Bmatrix} \hat{x} \\ \hat{y} \end{Bmatrix} \qquad \begin{Bmatrix} \hat{x} \\ \hat{y} \end{Bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix}^{-1} \begin{Bmatrix} x \\ y \end{Bmatrix}$$

And the transformation from global to local velocities, and vice versa, can be generated using the following matrix forms:

$$\text{(C-5)} \qquad \begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix} \begin{Bmatrix} \hat{u} \\ \hat{v} \end{Bmatrix} \qquad \begin{Bmatrix} \hat{u} \\ \hat{v} \end{Bmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix}^{-1} \begin{Bmatrix} u \\ v \end{Bmatrix}$$

where

$$\text{(C-6)} \qquad [\theta] = \begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

The derivative in the local frame can be generated using the Jacobean to convert from the global to local frame:

$$\begin{Bmatrix} \frac{\partial}{\partial \hat{x}} \\ \frac{\partial}{\partial \hat{y}} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{x}} \\ \frac{\partial x}{\partial \hat{y}} & \frac{\partial y}{\partial \hat{y}} \end{bmatrix} \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{Bmatrix} = [\theta]^T \begin{Bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{Bmatrix} \tag{C-7}$$

Because of the definition of $[\theta]$, the inverse of $[\theta]$ is given by the opposing rotation:

$$[\theta]^{-1} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} = [\theta]^T \tag{C-8}$$

The previous equation is further used to define the *u*- and *v*-velocity gradients:

$$\begin{Bmatrix} \frac{\partial \hat{u}}{\partial \hat{x}} \\ \frac{\partial \hat{u}}{\partial \hat{y}} \end{Bmatrix} = [\theta]^T \begin{Bmatrix} \frac{\partial \hat{u}}{\partial x} \\ \frac{\partial \hat{u}}{\partial y} \end{Bmatrix} \qquad \begin{Bmatrix} \frac{\partial \hat{v}}{\partial \hat{x}} \\ \frac{\partial \hat{v}}{\partial \hat{y}} \end{Bmatrix} = [\theta]^T \begin{Bmatrix} \frac{\partial \hat{v}}{\partial x} \\ \frac{\partial \hat{v}}{\partial y} \end{Bmatrix} \tag{C-9}$$

Using the Equation C-5b to define the velocity transform, Equation C-9 can be converted to the following vector equations:

$$\begin{Bmatrix} \frac{\partial \hat{u}}{\partial \hat{x}} \\ \frac{\partial \hat{u}}{\partial \hat{y}} \end{Bmatrix} = [\theta]^T \begin{Bmatrix} \frac{\partial}{\partial x}(\theta_{11}u + \theta_{21}v) \\ \frac{\partial}{\partial y}(\theta_{11}u + \theta_{21}v) \end{Bmatrix} \tag{C-10}$$

$$\begin{Bmatrix} \frac{\partial \hat{v}}{\partial \hat{x}} \\ \frac{\partial \hat{v}}{\partial \hat{y}} \end{Bmatrix} = [\theta]^T \begin{Bmatrix} \frac{\partial}{\partial x}(\theta_{12}u + \theta_{22}v) \\ \frac{\partial}{\partial y}(\theta_{12}u + \theta_{22}v) \end{Bmatrix} \tag{C-11}$$

Substituting the shorthand notation of Equation C-12 into Equations C-10 and C-11 and multiplying out the two vector equations results in the following forms:

(C-12) $$[\theta] = \begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix} = \begin{bmatrix} C_\theta & -S_\theta \\ S_\theta & C_\theta \end{bmatrix}$$

(C-13) $$\begin{Bmatrix} \frac{\partial \hat{u}}{\partial \hat{x}} \\ \frac{\partial \hat{u}}{\partial \hat{y}} \end{Bmatrix} = \begin{Bmatrix} C_\theta^2 \frac{\partial u}{\partial x} + C_\theta S_\theta \frac{\partial v}{\partial x} + C_\theta S_\theta \frac{\partial u}{\partial y} + S_\theta^2 \frac{\partial v}{\partial y} \\ -C_\theta S_\theta \frac{\partial u}{\partial x} - S_\theta^2 \frac{\partial v}{\partial x} + C_\theta^2 \frac{\partial u}{\partial y} + S_\theta C_\theta \frac{\partial v}{\partial y} \end{Bmatrix}$$

(C-14) $$\begin{Bmatrix} \frac{\partial \hat{v}}{\partial \hat{x}} \\ \frac{\partial \hat{v}}{\partial \hat{y}} \end{Bmatrix} = \begin{Bmatrix} -C_\theta S_\theta \frac{\partial u}{\partial x} + C_\theta^2 \frac{\partial v}{\partial x} - S_\theta^2 \frac{\partial u}{\partial y} + S_\theta C_\theta \frac{\partial v}{\partial y} \\ S_\theta^2 \frac{\partial u}{\partial x} - S_\theta C_\theta \frac{\partial v}{\partial x} - S_\theta C_\theta \frac{\partial u}{\partial y} + C_\theta^2 \frac{\partial v}{\partial y} \end{Bmatrix}$$

The local shear stress is given by Equation C-15. Equations C-13 and C-14 can be substituted into the local shear stress equation:

(C-15) $$\hat{\tau}_{xy} = \mu \left(\frac{\partial \hat{u}}{\partial \hat{y}} + \frac{\partial \hat{v}}{\partial \hat{x}} \right)$$

(C-16) $$\hat{\tau}_{xy} = \mu \left(2C_\theta S_\theta \left(\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} \right) + \left(C_\theta^2 - S_\theta^2 \right) \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \right)$$

The shear stress in the local frame can be converted back to the global frame by the following transform:

(C-17) $$\vec{\tau} = [\theta] \hat{\tau}_{xy} \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} = \hat{\tau}_{xy} [\theta] \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} = \hat{\tau}_{xy} \begin{Bmatrix} \cos\theta \\ \sin\theta \end{Bmatrix}$$

Equation C-17 can be verified using the vertical surface as a check:

$$\hat{\tau}_{xy} = \mu\left(2(0)(1)\left(\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right) + (0-1)\left(\frac{\partial v}{\partial x} + 0\right)\right) = -\mu\frac{\partial v}{\partial x} \tag{C-18}$$

$$\vec{\tau} = \hat{\tau}_{xy}\begin{Bmatrix} 0 \\ 1 \end{Bmatrix} = -\mu\frac{\partial v}{\partial x}\begin{Bmatrix} 0 \\ 1 \end{Bmatrix} \tag{C-19}$$

which matches the previous solution.

Using the definition of the surface normal vector (outward facing by convention), the relationship between the surface normal components and the sine/cosine of the orientation angle can be found by the following steps:

$$\hat{n} = \begin{Bmatrix} n_x & n_y \end{Bmatrix}$$

where $n_x = |\hat{n}|\sin\theta$ and $n_y = -|\hat{n}|\cos\theta$ and $|\hat{n}| = 1$. Therefore, $\sin\theta = n_x$ and $\cos\theta = -n_y$. From this, the shear stress can now be represented as:

$$\vec{\tau} = \hat{\tau}_{xy}\begin{Bmatrix} -n_y \\ n_x \end{Bmatrix} = \mu\left(-2n_y n_x\left(\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right) + \left(n_y^2 - n_x^2\right)\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right)\begin{Bmatrix} -n_y \\ n_x \end{Bmatrix} \tag{C-20}$$

Non-dimensionalizing, the previous equation:

$$\vec{c}_f = \frac{\mu^*}{\mathrm{Re}}\left(-2n_y n_x\left(\frac{\partial v^*}{\partial y^*} - \frac{\partial u^*}{\partial x^*}\right) + \left(n_y^2 - n_x^2\right)\left(\frac{\partial v^*}{\partial x^*} + \frac{\partial u^*}{\partial y^*}\right)\right)\begin{Bmatrix} -n_y \\ n_x \end{Bmatrix} \tag{C-21}$$

The contribution to the *x*- and *y*-forces and moment is given by:

(C-22) $$C_x = \sum_{i=1}^{NVBE} 2\left(c_p n_x + c_f n_y\right)\Delta x$$

(C-23) $$C_y = \sum_{i=1}^{NVBE} 2\left(c_p n_y - c_f n_x\right)\Delta x$$

(C-24) $$C_m = \sum_{i=1}^{NVBE} 2\left(c_p \left(R_x n_y - R_y n_x\right) - c_f \left(R_x n_x + R_y n_y\right)\right)\Delta x$$

where *NVBE* is the number of viscous boundary edges and R_x and R_y are the distances in the *x*- and *y*-directions, respectively, from the origin to the center of the boundary edge

APPENDIX D: Laminar Boundary Layer Test Cases and Results

This appendix gives a detailed description of all of the test cases used in the laminar boundary layer tests. The test cases are named by the percentage of the nominal spacing used to generate their boundary layer meshes. Therefore, the 100% case is the nominal case while fine cases are represented by percentages less than 100%, and vice versa for the courser cases.

Procedure for developing cases from 50% inner rule:

- Using a plate of length 25 units
- Using a freestream Reynolds number of 3600 (for 1 unit length)
- Full plate Reynolds number of 90000
- Using Blasius solution to initially size the thickness of the boundary layer (i.e. del = 5 * x / sqrt(Re_x))
- Find thickness at 0.5, 3, 12, and 32 units; 50% of these thicknesses correspond to the inner radius of the line source at 0, 3, 12, and 32 units from the leading edge
- Find the outer radius using the 2.5 rule (i.e. outer radius = inner radius * 2.5)
- Justify the initial spacing so that six steps are required (in the least) to cross 50% of the boundary layer (i.e., spacing = inner radius / 6). This corresponds to the 100% spacing test case.

100% Spacing Test Case					
x-Location	Fict. Local	Thickness	In. Rad.	Out. Rad.	Spacing
0	0.5	0.0589	0.0295	0.0737	0.0049
3	3	0.1443	0.0722	0.1804	0.0120
12	12	0.2887	0.1443	0.3608	0.0241
32	32	0.4714	0.2357	0.5893	0.0393

Spacing for Finer Test Cases						
90%	85%	80%	75%	70%	65%	60%
0.0044	0.0042	0.0039	0.0037	0.0034	0.0032	0.0029
0.0108	0.0102	0.0096	0.0090	0.0084	0.0078	0.0072
0.0217	0.0204	0.0192	0.0180	0.0168	0.0156	0.0144
0.0354	0.0334	0.0314	0.0295	0.0275	0.0255	0.0236

Spacing for Medium Test Cases						
110%	115%	120%	125%	130%	135%	140%
0.0054	0.0056	0.0059	0.0061	0.0064	0.0066	0.0069
0.0132	0.0138	0.0144	0.0150	0.0156	0.0162	0.0168
0.0265	0.0277	0.0289	0.0301	0.0313	0.0325	0.0337
0.0432	0.0452	0.0472	0.0491	0.0511	0.0531	0.0550

Spacing for Medium-Course Test Cases						
150%	160%	170%	175%	180%	190%	200%
0.0074	0.0078	0.0083	0.0086	0.0088	0.0093	0.0098
0.0180	0.0192	0.0204	0.0210	0.0216	0.0228	0.0240
0.0362	0.0386	0.0410	0.0422	0.0434	0.0458	0.0482
0.0590	0.0629	0.0668	0.0688	0.0707	0.0747	0.0786

Spacing for Course Test Cases							
210%	220%	230%	240%	250%	260%	270%	280%
0.0103	0.0108	0.0113	0.0118	0.0123	0.0127	0.0132	0.0137
0.0252	0.0264	0.0276	0.0288	0.0300	0.0312	0.0324	0.0336
0.0506	0.0530	0.0554	0.0578	0.0603	0.0627	0.0651	0.0675
0.0825	0.0865	0.0904	0.0943	0.0983	0.1022	0.1061	0.1100

Spacing for Most Course Test Cases							
300%	350%	400%	450%	500%	600%	700%	800%
0.0147	0.0172	0.0196	0.0221	0.0245	0.0294	0.0343	0.0392
0.0360	0.0420	0.0480	0.0540	0.0600	0.0720	0.0840	0.0960
0.0723	0.0844	0.0964	0.1085	0.1205	0.1446	0.1687	0.1928
0.1179	0.1376	0.1572	0.1769	0.1965	0.2358	0.2751	0.3144

Actual Number of Elements to Span the Lower 60% of the Boundary Layer

100% case	10 elements	180% case	5.56 elements
90% case	11.11 elements	190% case	5.26 elements
85% case	11.76 elements	200% case	5 elements
80% case	12.5 elements	210% case	4.76 elements
75% case	13.33 elements	220% case	4.55 elements
70% case	14.29 elements	230% case	4.35 elements
65% case	15.38 elements	240% case	4.17 elements
60% case	16.67 elements	250% case	4 elements
110% case	9.09 elements	260% case	3.85 elements
115% case	8.7 elements	270% case	3.7 elements
120% case	8.33 elements	280% case	3.57 elements
125% case	8 elements	300% case	3.33 elements
130% case	7.69 elements	350% case	2.86 elements
135% case	7.41 elements	400% case	2.5 elements
140% case	7.14 elements	450% case	2.22 elements
150% case	6.67 elements	500% case	2 elements
160% case	6.25 elements	600% case	1.66 elements
170% case	5.88 elements	700% case	1.43 elements
175% case	5.71 elements	800% case	1.25 elements

Flow external to the boundary layer is represented by a grid of 0.25 spacing (6.4 time that of the largest spacing used above to represent flow in the boundary layer)

Laminar Profiles for the Various Mesh Spacings:

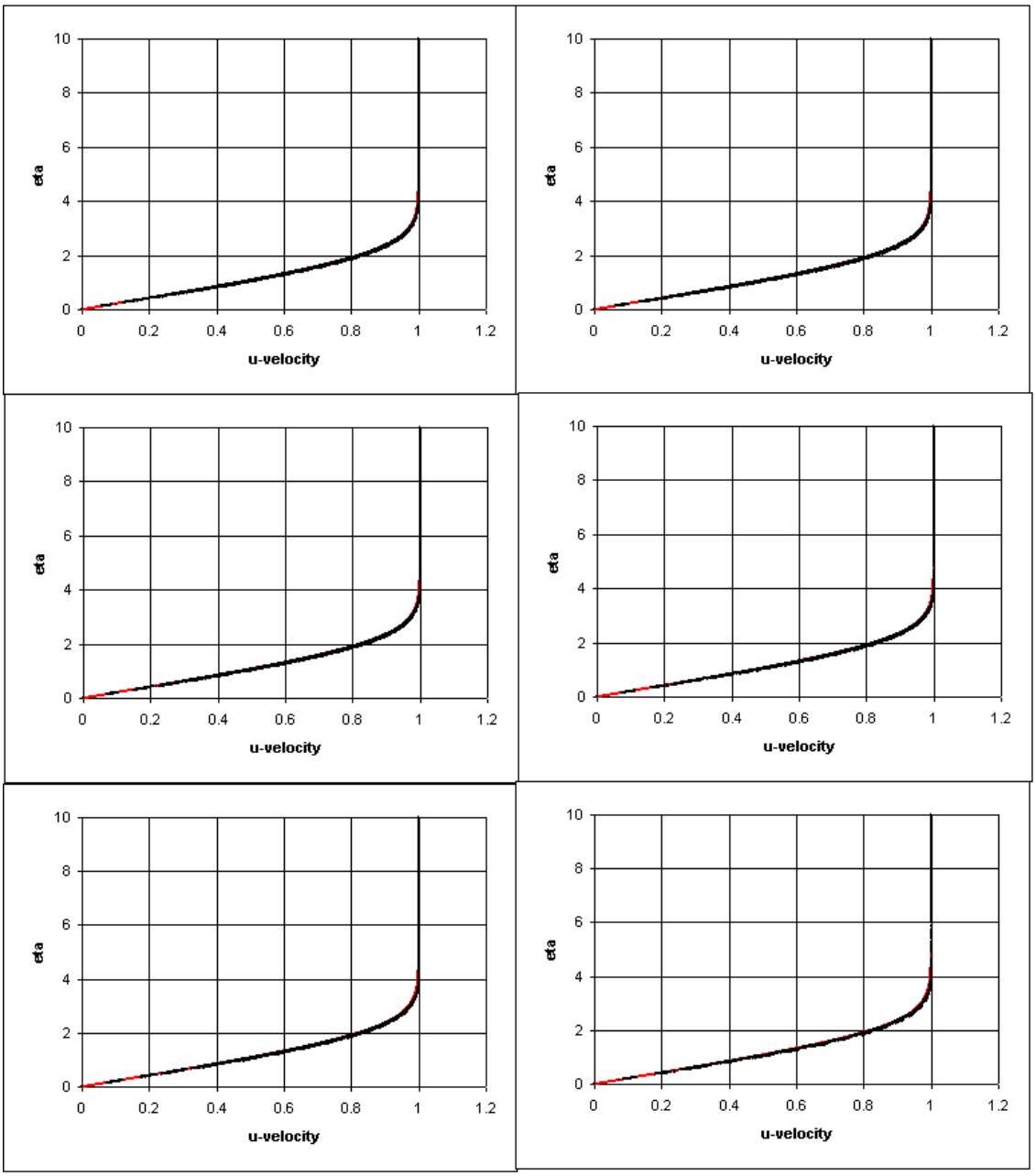

Figure D-1: Laminar Velocity Profiles for 60%, 65%, 70%, 75%, 80%, and 85% Mesh Cases. (from left to right; top to bottom)

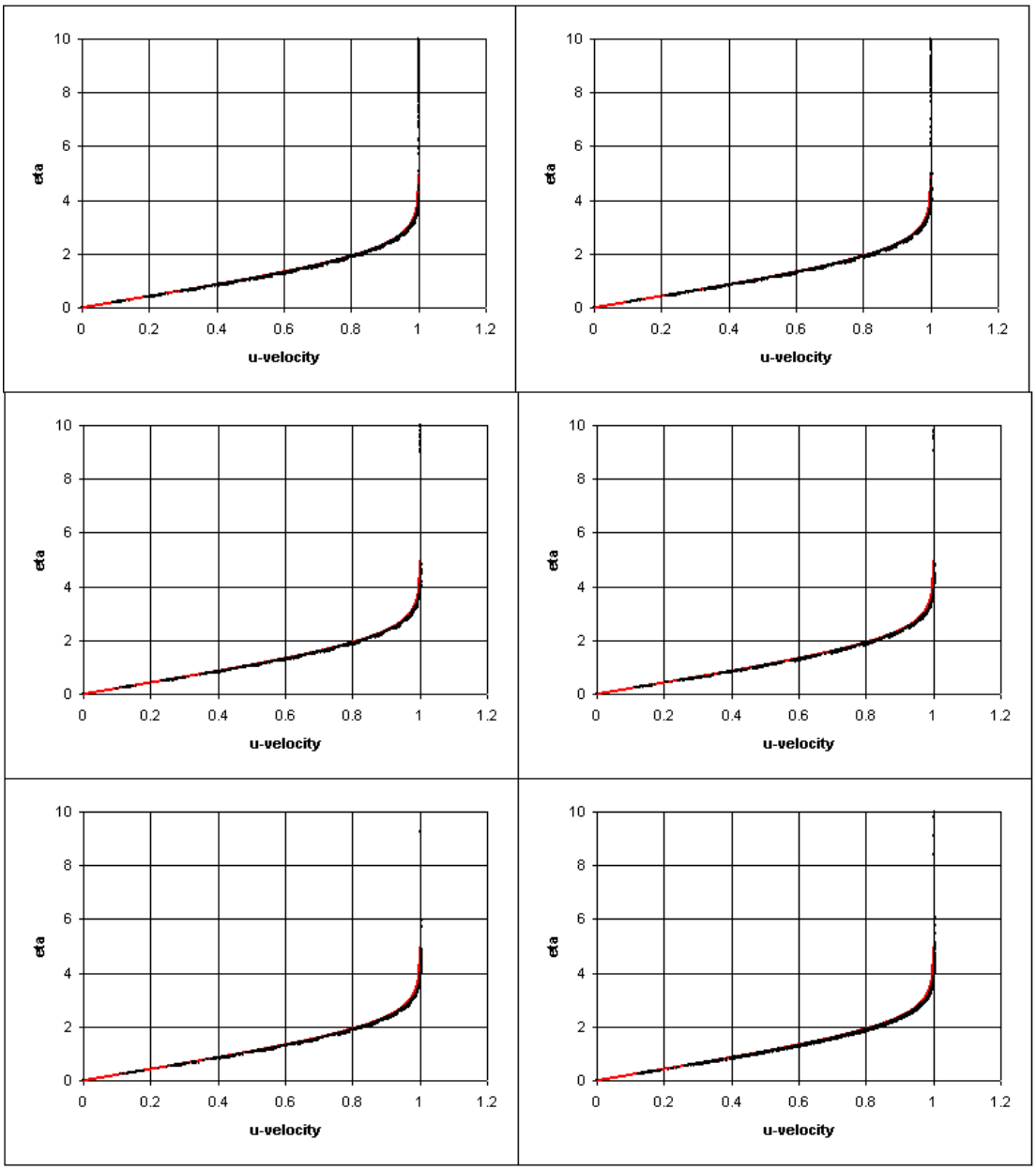

Figure D-2: Laminar Velocity Profiles for 90%, 100%, 110%, 115%, 120%, and 125% Mesh Cases. (from left to right; top to bottom)

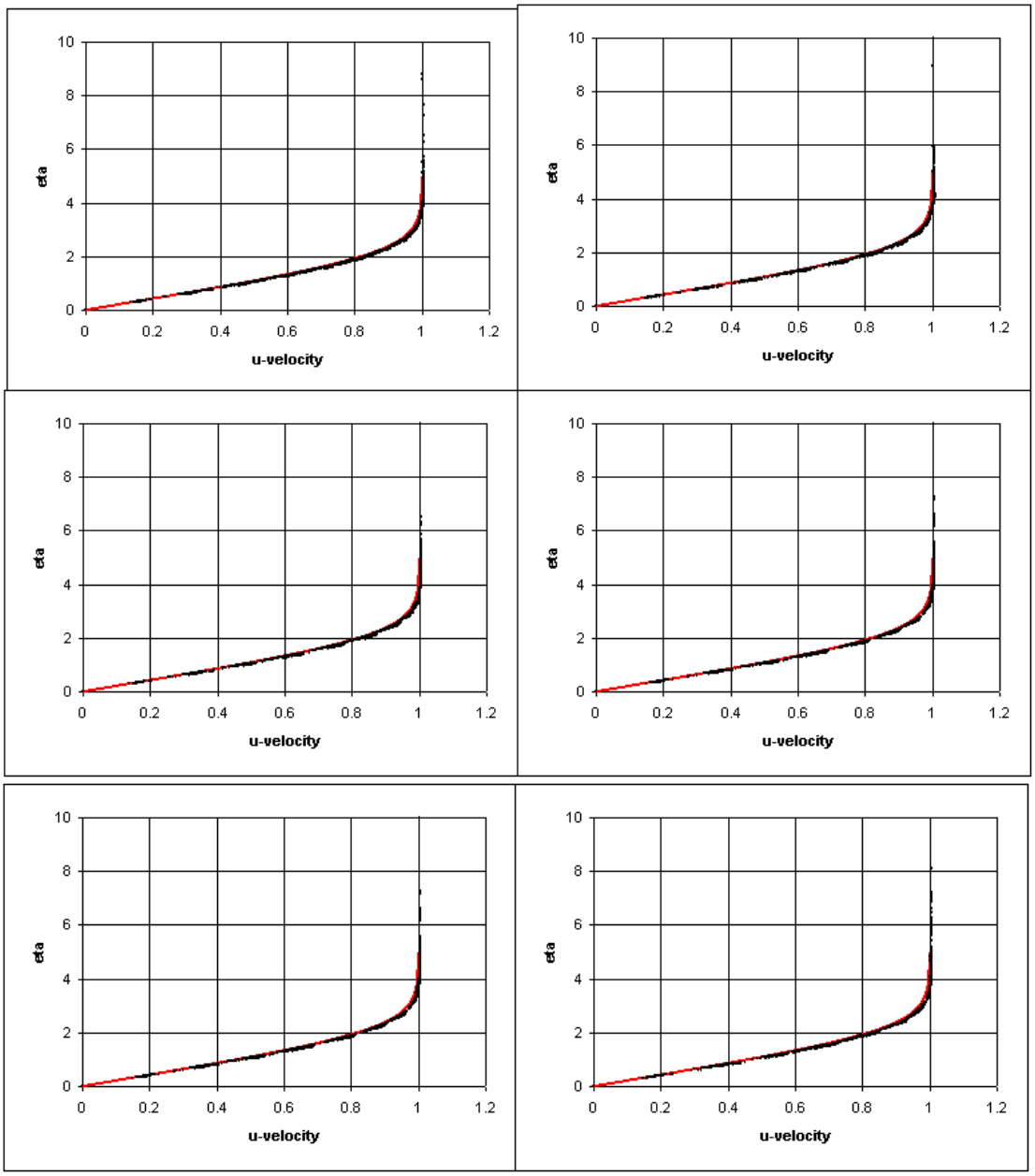

Figure D-3: Laminar Velocity Profiles for 130%, 135%, 140%, 145%, 150%, and 160% Mesh Cases. (from left to right; top to bottom)

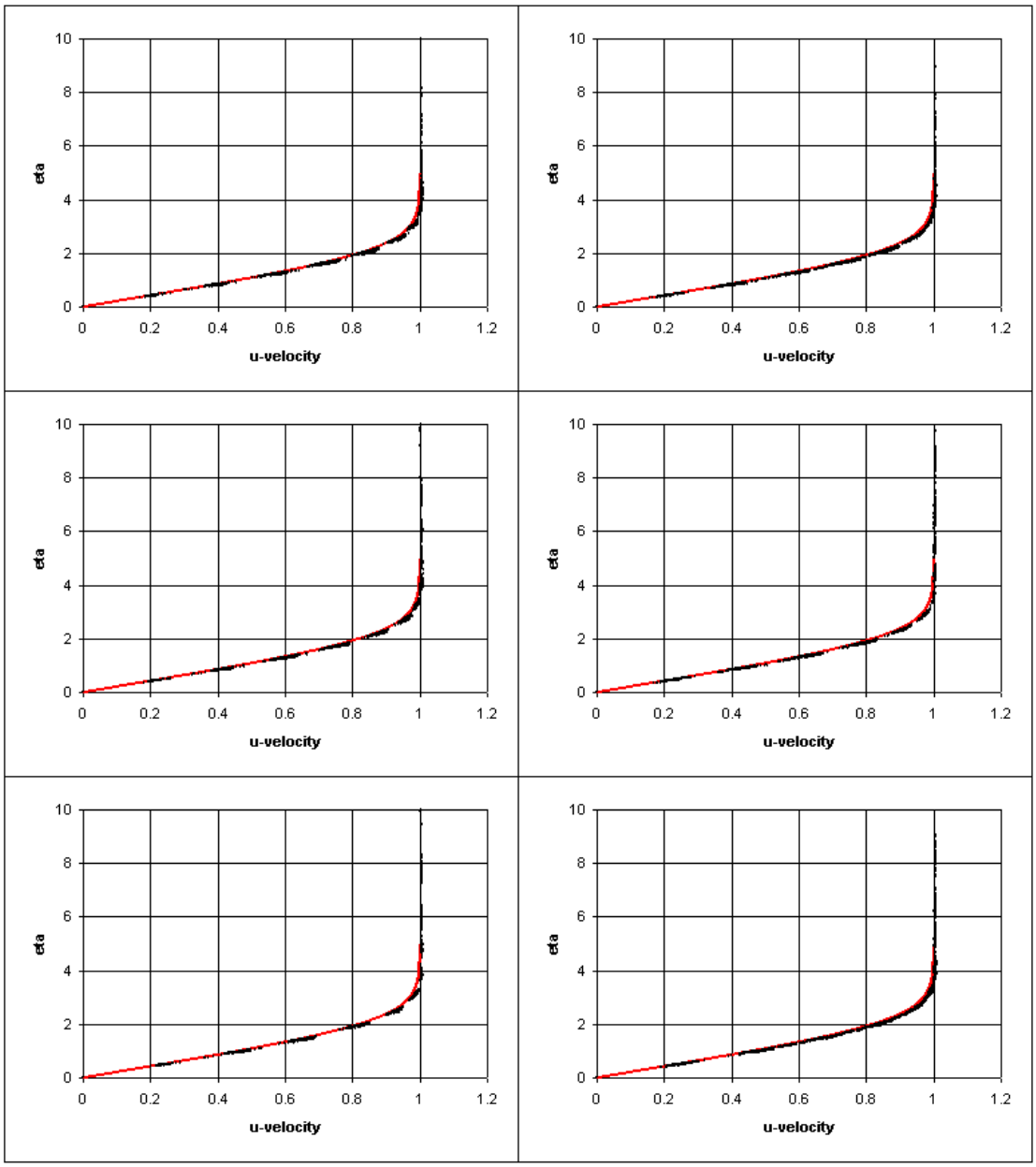

Figure D-4: Laminar Velocity Profiles for 170%, 175%, 180%, 190%, 200%, and 210% Mesh Cases. (from left to right; top to bottom)

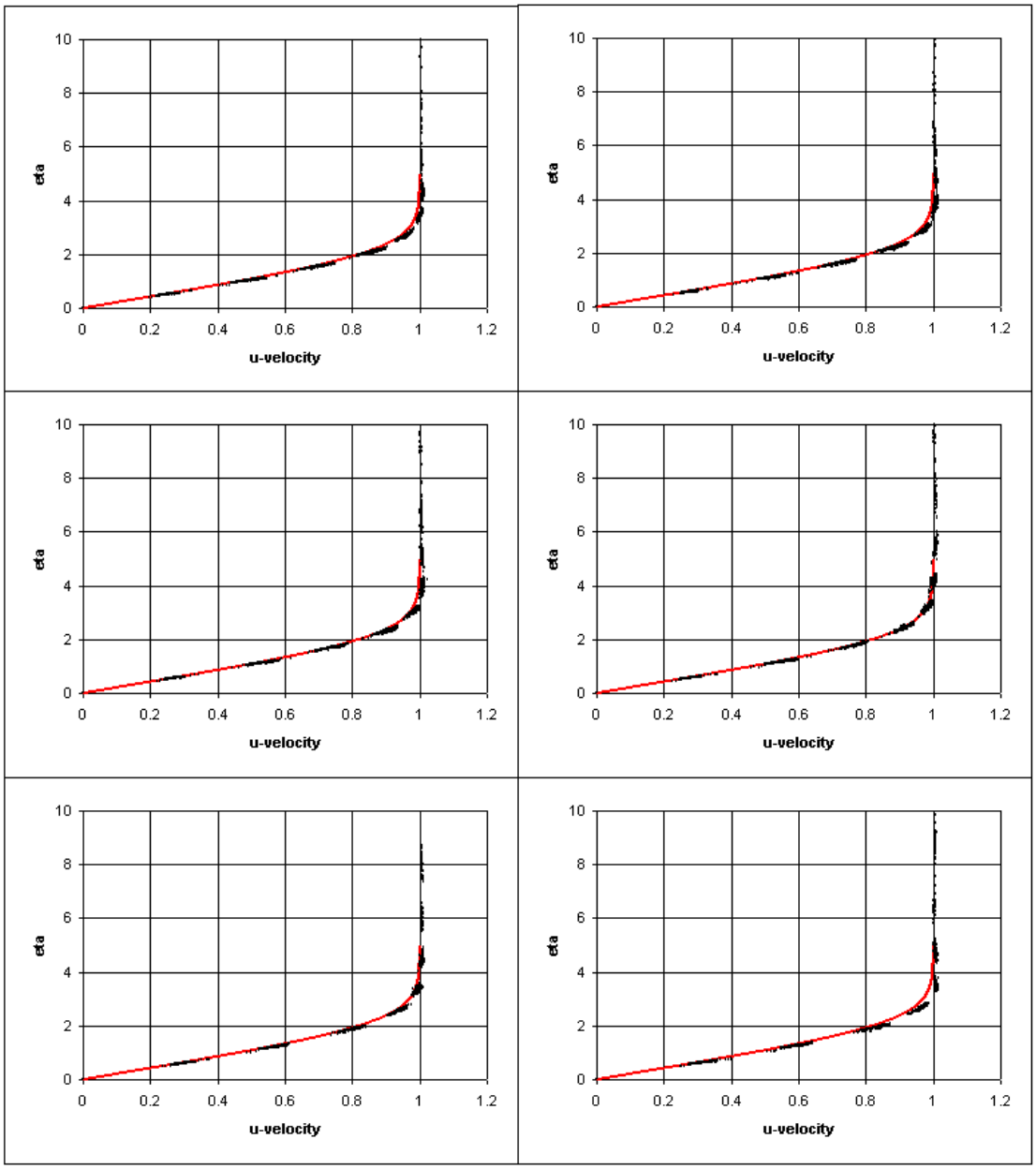

Figure D-5: Laminar Velocity Profiles for 220%, 230%, 240%, 250%, 260%, and 270% Mesh Cases. (from left to right; top to bottom)

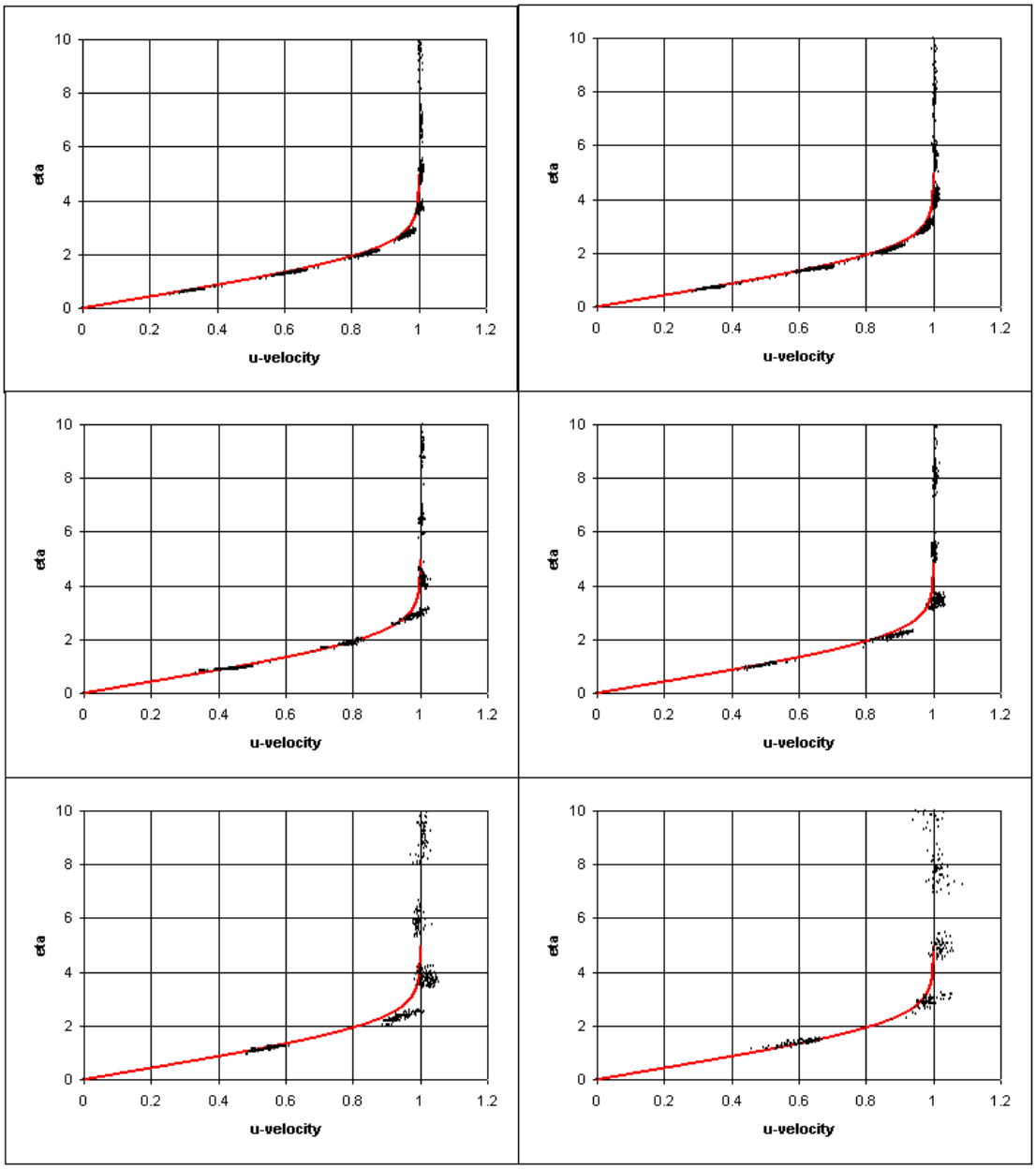

Figure D-6: Laminar Velocity Profiles for 280%, 350%, 400%, 450%, 500%, and 600% Mesh Cases. (from left to right; top to bottom)

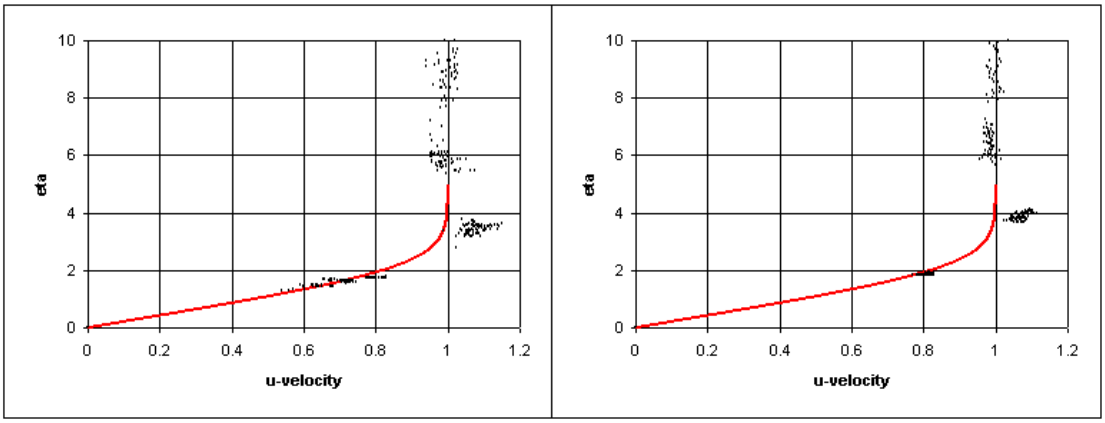

Figure D-7: Laminar Velocity Profiles for 700% and 800% Mesh Cases. (from left to right; top to bottom)

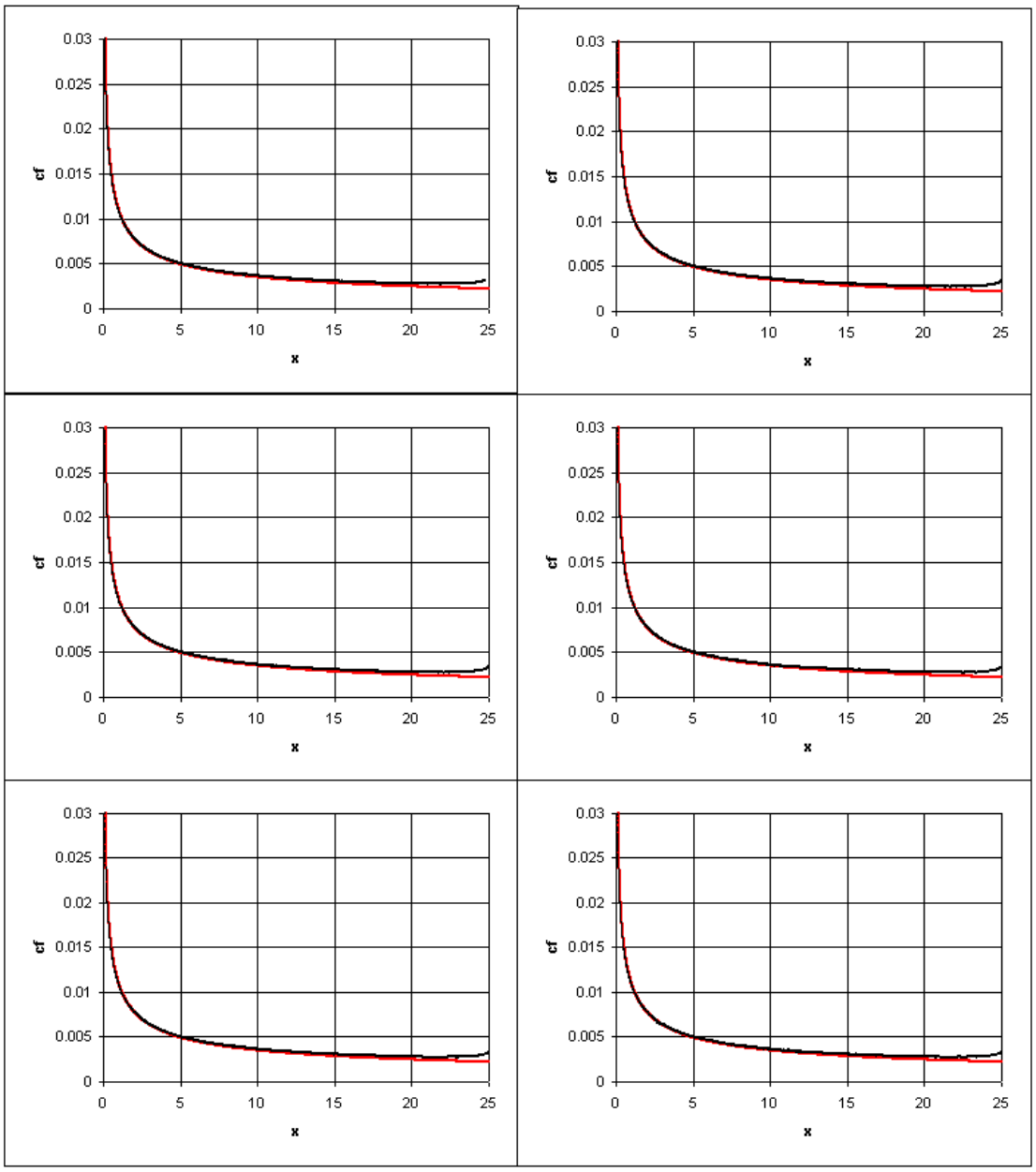

Figure D-8: Shear Stress Distribution for 60%, 65%, 70%, 75%, 80%, and 85% Mesh Cases. (from left to right; top to bottom)

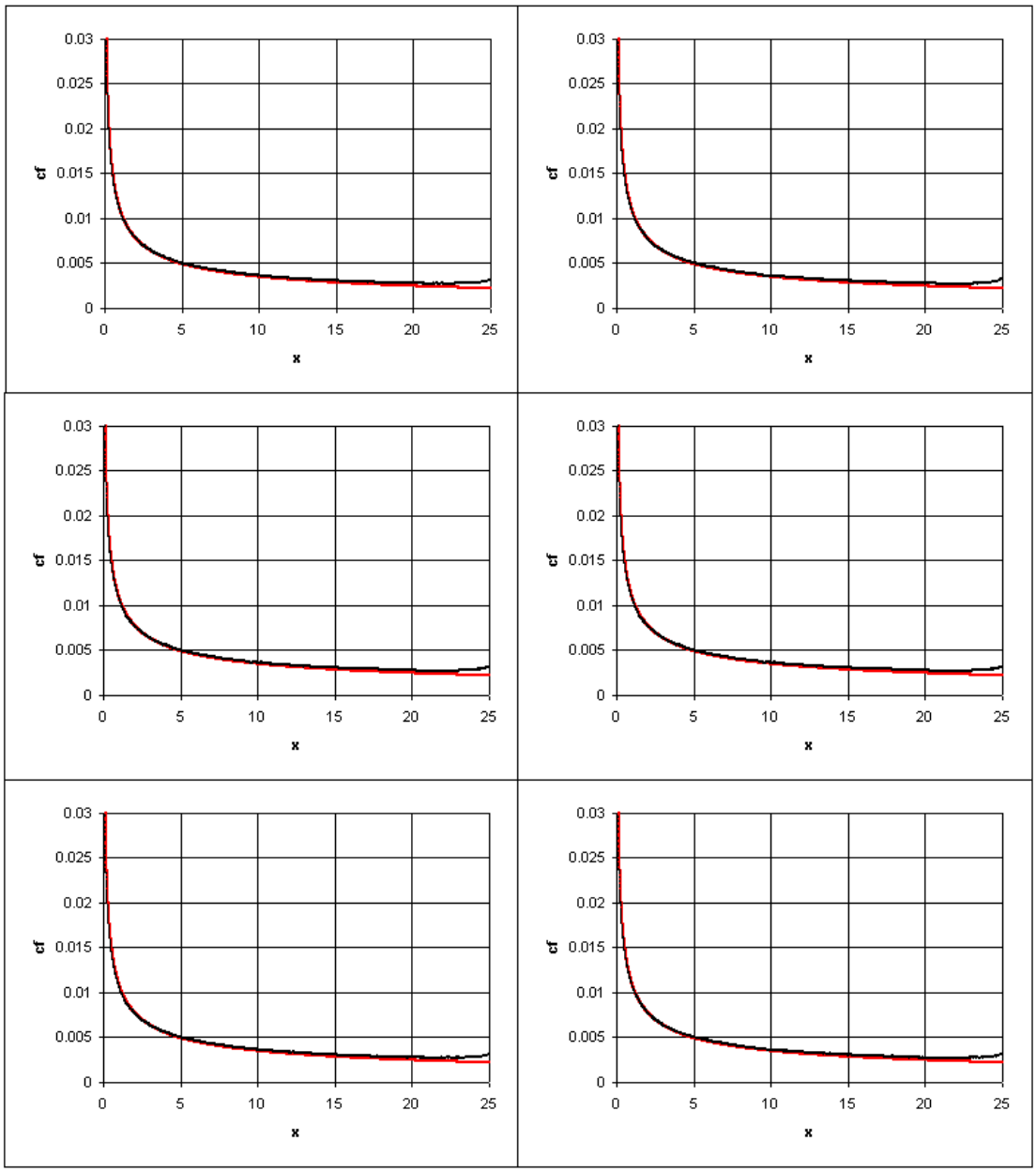

Figure D-9: Shear Stress Distribution for 90%, 100%, 110%, 115%, 120%, and 125% Mesh Cases. (from left to right; top to bottom)

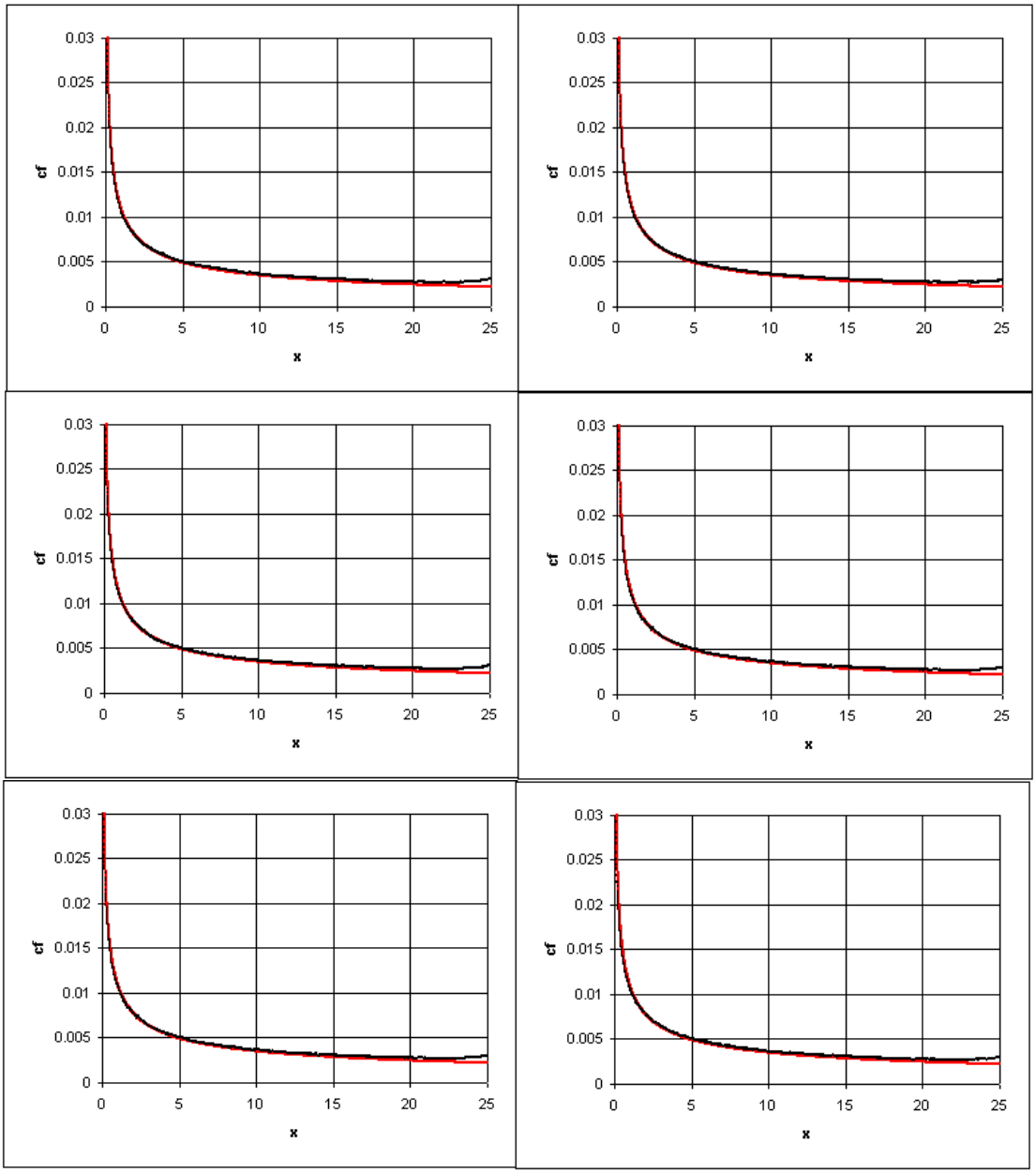

Figure D-10: Shear Stress Distribution for 130%, 135%, 140%, 145%, 150%, and 160% Mesh Cases. (from left to right; top to bottom)

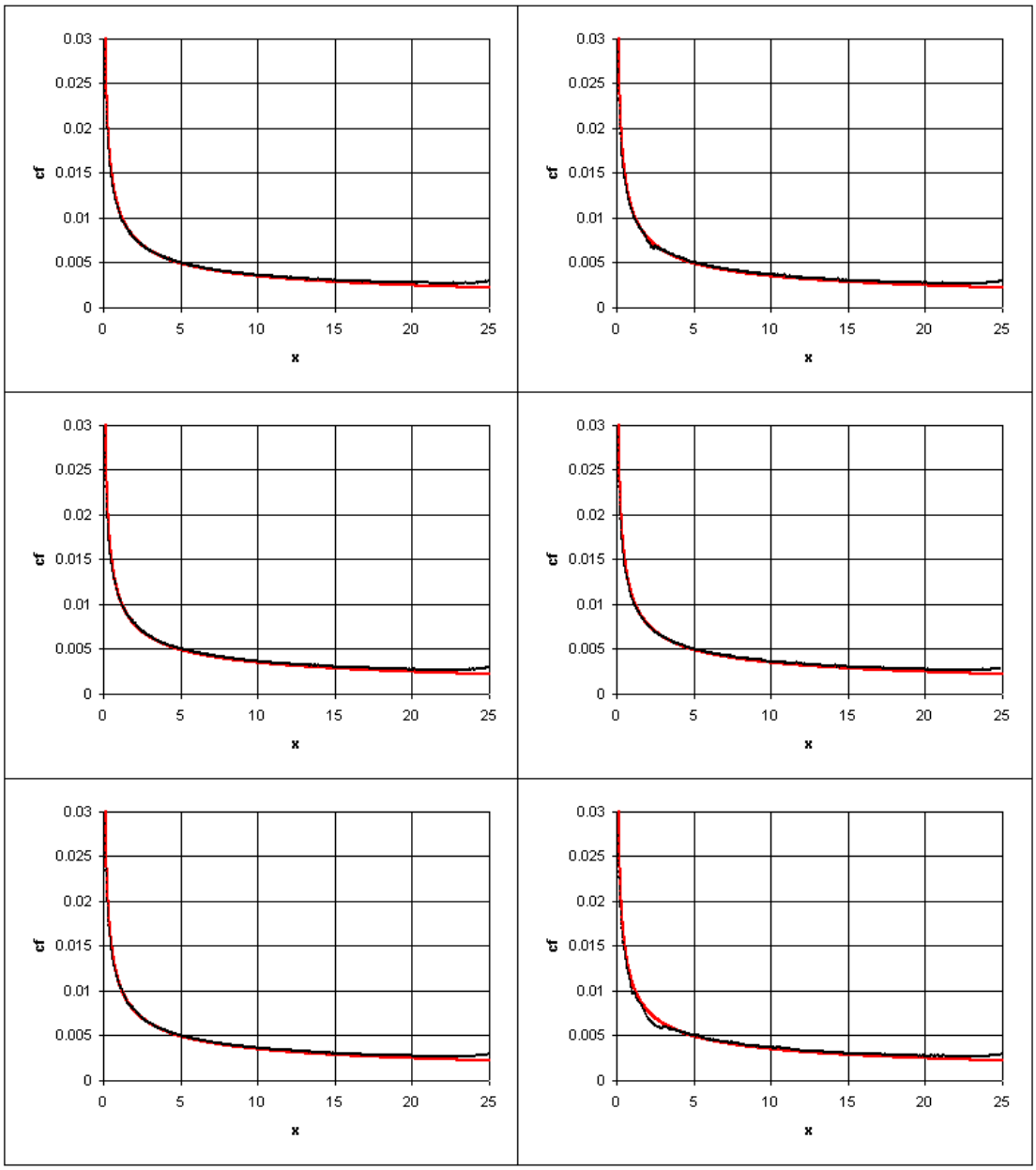

Figure D-11: Shear Stress Distribution for 170%, 175%, 180%, 190%, 200%, and 210% Mesh Cases. (from left to right; top to bottom)

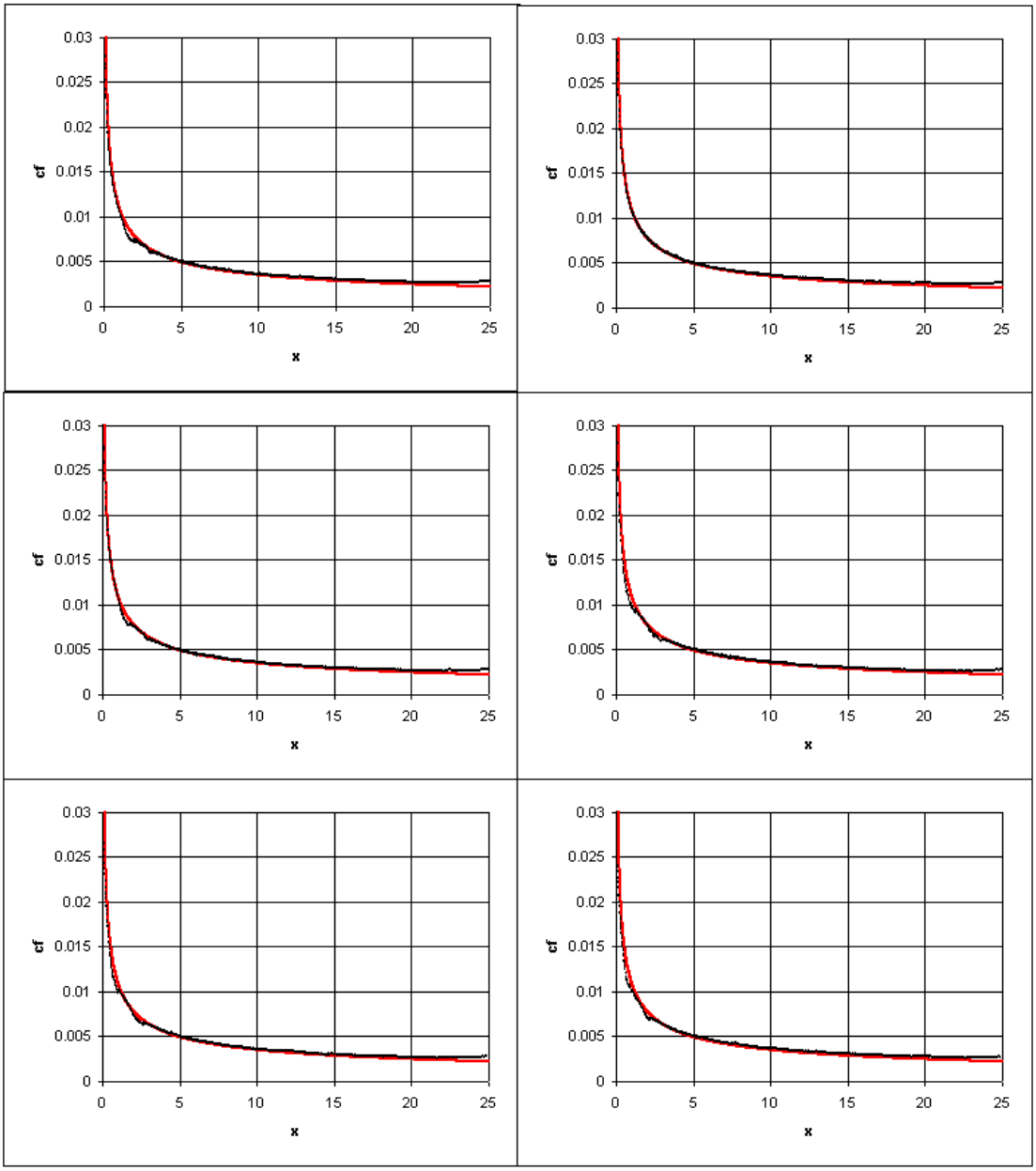

Figure D-12: Shear Stress Distribution for 220%, 230%, 240%, 250%, 260%, and 270% Mesh Cases. (from left to right; top to bottom)

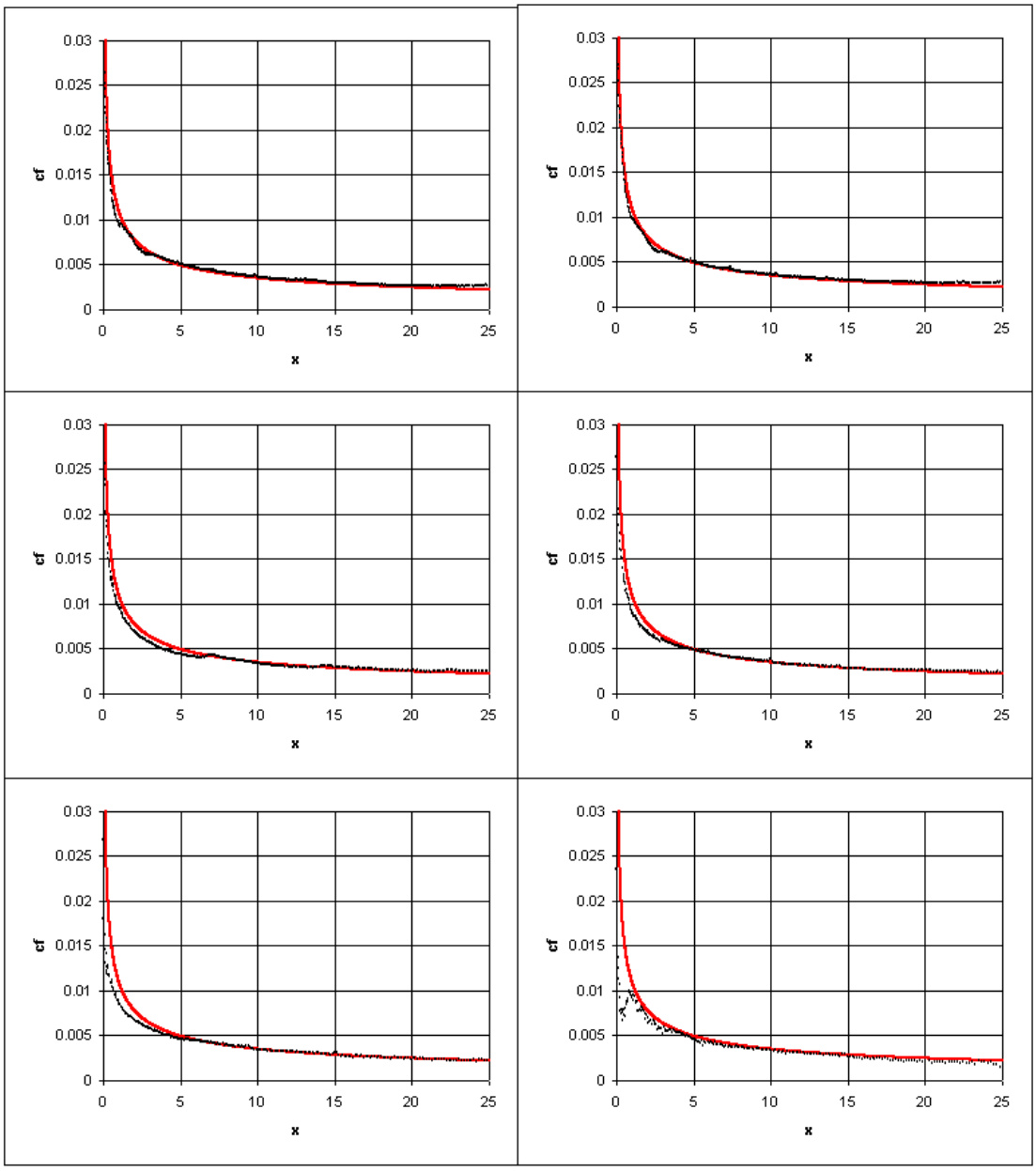

Figure D-13: Shear Stress Distribution for 280%, 350%, 400%, 450%, 500%, and 600% Mesh Cases. (from left to right; top to bottom)

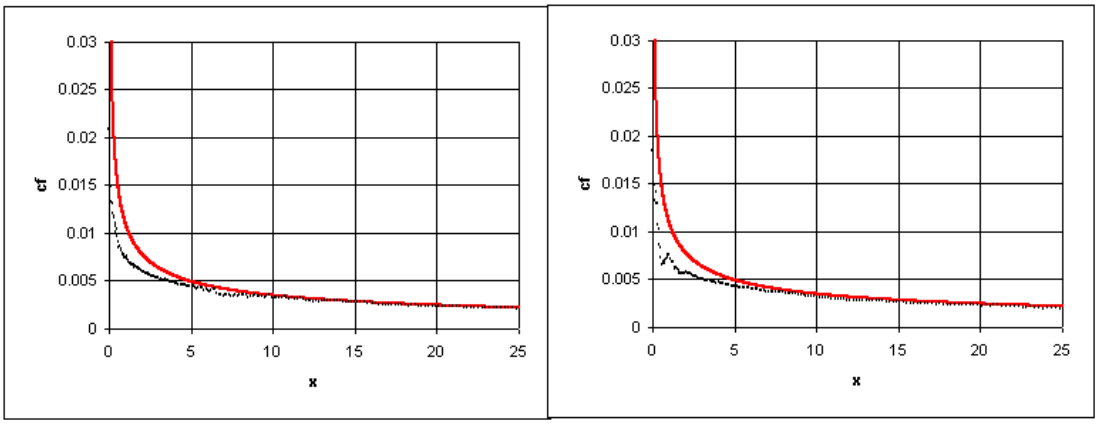

Figure D-14: Shear Stress Distribution for 700% and 800% Mesh Cases. (from left to right; top to bottom)

VITA

Nicholas J Moffitt

Candidate for the Degree of

Master of Science

Thesis: FIRST STAGES OF A VISOUS FINITE ELEMENT SOLVER FOR NON-INERTIAL AND AEROELASTIC PROBLEMS

Major Field: Mechanical Engineering

Biography:

Personal Data: Born in Enid, Oklahoma March 12, 1980. The son of Edward L. and Diahanna R. Moffitt.

Education: Graduated from Cleveland High School, Cleveland, Oklahoma in May 1998; received a Bachelor of Science degree in Aerospace Engineering from Oklahoma State University in December 2002. Completed the Requirements for the Master of Science degree at Oklahoma State University in December 2004.

Experience: Graduate Research Assistant, Oklahoma State University School of Mechanical and Aerospace Engineering, 2002-2004; Teaching Assistant, Oklahoma State University School of Mechanical and Aerospace Engineering, 2002-2004; Intern, NORDAM Group, Nacelles and Thrust Reversers, 1997-1998.

Professional Memberships: American Institute of Aeronautics and Astronautics (AIAA), Pi Tau Sigma, and Golden Key National Honor Society.

Name: Nicholas J Moffitt Date: December 2004

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: FIRST STAGES OF A VISCOUS FINITE ELEMENT SOLVER FOR NON-INERTIAL AND AEROELASTIC PROBLEMS

Pages of Study: 232 Candidate for the Degree of Master of Science

Major Field: Mechanical Engineering

Scope and Method of Study:

In 2003, Tim Cowan generated a finite element Euler solver for aeroelastic and non-inertial problems. This thesis concentrates on the expansion of the Euler equations to the full Navier-Stokes equations for laminar cases. The study develops the needed terms to adapt the equations, implements the equations in a Fortran program, and then finishes by verifying and validating the code.

Findings and Conclusions:

Suggestions are given for the appropriate mesh to calculate the proper flow attributes and local shear stress distribution of a laminar boundary layer. Flow attributes are important if an understanding of the complexities of the flow are to be investigated, but the local shear stress and overall drag on the surface are used to calculate the forcing functions for the structural and dynamics solver contained in the package. Suggestions were also made for stable CFL values used to evaluate steady solutions and stable time steps to generate unsteady solutions.

Several viscous test cases were used varying across the subsonic, transonic, and supersonic regimes. Test cases include the Blasius solution, circular cylinder, and several airfoil solutions. Also several solutions were found to match pictures of very slow flow and viscous flows with Reynolds numbers up to the onset of turbulence. Several investigations were made past the onset of transition and turbulence to investigate the ability of the solver in these areas.

After verification and validation of the viscous solver, the solver was found to work well for the cases test in the subsonic, transonic, and supersonic laminar solutions. But the solver does not possess a turbulence model, and therefore cannot properly predict separation or properties in a transitional or turbulent boundary layer. The inclusion of as many test cases in all areas of subsonic, transonic, and supersonic flow was sought; but, caution is given for solutions investigated that do not resemble the verification or validation test cases in any way

ADVISOR'S APPROVAL: A. S. Arena