The Pennsylvania State University

The Graduate School

A CARTESIAN IMMERSED BOUNDARY METHOD STUDY OF DAMAGED AND UNDAMAGED AIRFOIL PERFORMANCE

A Thesis in

Aerospace Engineering

by

Adam Goss

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2009

The thesis of Adam Goss was reviewed and approved* by the following:

Philip J. Morris
Boeing/A.D Welliver Professor of Aerospace Engineering
Thesis Advisor

Dennis K. McLaughlin
Professor of Aerospace Engineering

George A. Lesieutre
Professor of Aerospace Engineering
Head of the Department of Aerospace Engineering

*Signatures are on file in the Graduate School

ABSTRACT

A computational study is performed to develop a capability of analyzing the effect damage has on the aerodynamic performance of airfoils. A Cartesian immersed boundary method is implemented in 2D simulations of the compressible Navier-Stokes equations. These equations are discretized using a Weighted Essentially Non-Oscillatory (WENO) scheme for spatial derivatives and a 4^{th} order Runge-Kutta scheme for temporal derivatives. Results from time-accurate, parallel computations are presented for a NACA 0009 airfoil in both undamaged and damaged states. Time histories of lift, drag, and moment coefficients are shown, along with Mach contours and turbulent kinetic energy contours at three times throughout the simulations. Based on these initial results, it was determined that the boundary conditions used in the simulations were causing reflections of propagating disturbances which contaminate the solution. Further studies must be completed with the current methodology using alternate boundary conditions to validate its results prior to continuing its development for this application in aircraft survivability.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This work was funded by the Air Force's Science and Engineering Palace Acquire (S&E PAQ) training program, in coordination with the Aerospace Vehicle Survivability Facility (AVSF) at Wright-Patterson Air Force Base, OH. The author would like to thank Dr. Philip Morris for his time in advising the research. Input from Mr. Steve Miller, a Ph.D. student at Penn State, regarding computational questions is appreciated.

Chapter 1

Introduction and Motivation

In the airspace of modern warfare, aviators and their aircraft are subjected to a diversity of hostile threats posing a challenge to the aircraft's integrity and the pilot's life. Whether it be manned aircraft providing close air support, such as an A-10 "Thunderbolt", or unmanned aerial vehicles (UAVs) performing reconnaissance missions, these assets enter hostile environments where a strong probability exists for them to sustain damage from an array of small arms fire, to MAN-Portable Air Defense (MANPAD) rockets, to Surface-to-Air Missiles (SAMs).

One aircraft that engages and is likely to sustain more ballistic damage than any other aircraft in the U.S. Air Force is the A-10 Thunderbolt II. Engineered to be a flying tank, it can withstand severe damage and still return safely after completing its mission; thus making it one of the Air Force's most survivable aircraft [1]. In the unmanned sector, the effectiveness of combat UAVs has been clearly demonstrated in reconnaissance, surveillance, and targeting roles, facilitated largely by the recent War on Terror in the Middle East.

It is evident that these and other aerial assets of the military need to survive numerous missions throughout their lifetime. Hence, one burden an aviator has when operating a damaged aircraft is making the decision to abandon or somehow salvage the plane, which certainly heightens the inherent risk to him or her. Knowledge of the aircraft's state and remaining capabilities at these junctures is crucial to what decision is made.

In 1983, an Israeli Air Force pilot, Captain Zivi Nadivi was faced with such a decision. As he was intercepting a hostile aircraft in a simulated air defense mission, his F-15D Eagle collided with an A-4 Skyhawk, resulting in the Skyhawk's destruction, and the right wing torn completely from the Eagle. In Nadivi's account of the collision, his decision to apply afterburner

to recover the spinning aircraft was "totally opposite instinct." The increased speed however, leveled the aircraft and enabled Nadivi to land the F-15 on one wing. McDonnel Douglass investigated the incident, after first saying it was "aerodynamically impossible" for the aircraft to fly on one wing according to the latest computer simulations. Their findings showed that the wide fuselage of the F-15D produced enough lift that, with the afterburners lit, made the aircraft into a rocket with no need of wings [2,3]. It remains speculation as to whether this outcome could have been simulated with better computer technology and whether such technology could have enlightened Nadivi such that his decision to light the afterburners would have been his first response.

More than two decades later, vast improvements have been made in computational aerodynamic simulations. Applying new technology to the survivability arena, may allow decisions similar to that faced by Captain Nadivi to be easier to make. Should an aircraft sustain damage, knowledge of how its aerodynamic characteristics change may enable the pilot to alter the handling of the vehicle in order to bring the asset safely home for repair. Consider a wing that has ballistic damage, as in Figure 1.1. How would this altered state of the structure affect the envelope the pilots or operators have to operate in? Would they encounter a flutter condition sooner than expected – possibly at normal cruising speeds? If so, could the aircraft be salvaged if flown more conservatively; or less conservatively as the case may be? Answers to questions like these require a strong understanding of fluid-structure interactions, as well as ample resources to analyze them.

Figure 1.1: Ballistic damage of aircraft wing in a range test

Current engineering communities see a void in damage-assessment technology and a need for new capabilities in support of the war-fighter, which has motivated the research endeavor introduced here. An aerodynamic analysis is performed on damaged airfoils that involve an interaction between a moving body and its surrounding fluid. Much research has been conducted in the past on modeling such fluid-structure interactions. Several approaches have been formulated and used throughout industry. But the underlying theme among them all is the expensive computational price to be paid.

Computational Aeroelasticity History

Sadeghi [8] reviewed various methodologies that have been used for dealing with fluid-structure interaction numerically. Early modeling approaches relied on linear potential theory to formulate the mean flow and unsteady perturbations of the structure. Some research studies investigated airfoil flutter using harmonic, indicial, and time-domain approaches. Others solved the Transonic Small Disturbance (TSD) equation coupled to a spring-mass model to investigate shock wave motion and its relevance to the dip in flutter speeds seen in experimental data through

the transonic region. All these linear approaches do not account for nonlinear behavior where the richness of the physics lies. To account for nonlinearity, the Navier-Stokes equations must be solved, using an appropriate model for turbulence. Turbulence, Sadeghi states, plays a key role where shock-boundary-layer interference influences shock motion.

Nonlinear approaches were developed for investigating areas for which the earlier approaches were insufficient. For example, the three-dimensional Euler equations were used to study the aeroelastic response of wings to unsteady flow. Other investigators took an Eulerian-Lagrangian approach to the solution of the fluid-structure dynamics as a single system.

Sadeghi points out that with time-marching approaches, the computational grid must be updated each time step to account for the structural motion, which results in a significant computational cost even with modern computational resources. Body-fitted grids are especially taxing to reproduce, which has spurred development of Cartesian grid based methods. Cho et. al.[5] mention one such formulation recently implemented by Marshall and Ruffin [6] who solved the near-wall flow using a "cut-cell" methodology. This methodology replaces reduced size cells intersecting the body with a boundary interpolation technique. However, smoothness and stability issues in this cut-cell methodology, as the authors describe, renders it ineffective in viscous simulations where general bodies are modeled. This motivates the use of "immersed boundary methods" which essentially removes the structural body from the model but represents its influence with fictitious body forces in the fluid-governing equations.

Peskin originated an immersed body method in the 1940s by modeling flow through heart valves [7]. The technique had to be tuned for each unique application thereafter which exposed its limitation through equation stiffness and spurious oscillations arising in the results [5]. In the late 1990s, Mohd-Yusof [8] successfully addressed these issues by applying the fictitious body force to the discretized equations of motion. Cho et. al.[5] review numerous examples using this

modification, including flow around square cylinders, flexible structures, and flapping wings, demonstrating the applicability of the technique.

In the research presented here, the immersed boundary technique is applied to a study of effects of damage on airfoil performance. The computational approach taken stems from the work completed by Dr. Cho at the Pennsylvania State University in developing a CARtesian Immersed BOUndary (CARIBOU) algorithm for fluid-structure interaction solutions. This code is applied to solving the 2D compressible Navier-Stokes equations to an airfoil in undamaged and damaged states.

The following chapter describes the mathematical framework within CARIBOU and how the body force representing the airfoil is included in the governing equations. The discussion includes how the equations are averaged and the methodology for modeling the extra terms arising from the averaging process. Chapter 3 explains how the equation formulation is implemented numerically. A description is given of the computational domain and how the governing equations are transformed to discrete form. Spatial and temporal discretization strategies are explained, in addition to the various initial and boundary conditions available in the current version of CARIBOU. Finally, chapter 4 presents the results of the numerical simulations, including time histories of aerodynamic coefficients (C_D, C_L, C_M) and images of the flow solutions at selected intervals throughout the simulations. These results are drawn upon in chapter 5 for conclusions on how CARIBOU performed towards the intended objective, and what could be improved upon for future work.

Chapter 2

Equation Framework

Chopra [9] recently developed a parallel implementation of an immersed boundary method for the simulation of oscillating airfoils, in close coordination to the work described by Cho et. al. [5]. The present research extends the CARtesian Immersed BOUndary (CARIBOU) code to modeling flow around damaged airfoils, using two-dimensional cases for a preliminary study. Underlying its framework is a methodology closely resembling that of Mohd-Yusof [8], but using the compressible Navier-Stokes equations, written below in Cartesian tensor form.

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}\left(\rho u_i\right) = 0 \tag{2.1a}$$

$$\frac{\partial}{\partial t}\left(\rho u_i\right) + \frac{\partial}{\partial x_j}\left(\rho u_i u_j + p\delta_{ij}\right) = \frac{\partial \tau_{ij}}{\partial x_j} + f_i \tag{2.1b}$$

$$\frac{\partial}{\partial t}\left(\rho e_t\right) + \frac{\partial}{\partial x_j}\left[\left(\rho e_t + p\right)u_j\right] = \frac{\partial}{\partial x_j}\left(u_i \tau_{ij} - q_j\right) \tag{2.1c}$$

An equation of state is also used as a closing equation to the above system, relating the pressure, temperature, and density. The ideal gas law can be written as

$$p = \frac{\rho T}{\gamma} \tag{2.2}$$

where γ is the ideal gas constant evaluated by the ratio of specific heats at constant pressure (c_p) and volume (c_v) according to

$$\gamma = c_p / c_v \,. \tag{2.3}$$

As written, the variables in the governing equation are non-dimensionalized by free-stream quantities, namely ρ_∞^* for density; a_∞^* for velocity; L_∞^* for length (airfoil chord); μ_∞^* for viscosity; T_∞^* for temperature; $\rho_\infty^* a_\infty^{*2}$ for pressure; and $\mu_\infty^* a_\infty^* / L_\infty^*$ for stress.

In the energy equation (2.1c), the total energy is defined by

$$e_t = \frac{T}{\gamma(\gamma-1)} + \frac{1}{2}(u_i u_i) , \tag{2.4}$$

and according to Fourier's law of heat conduction, the heat flux term is given by

$$q_j = -\frac{1}{\gamma-1}\frac{M}{\mathrm{Re}}\frac{\mu}{\mathrm{Pr}_L}\left(\frac{\partial T}{\partial x_j}\right) . \tag{2.5}$$

The three nondimensional quantities *M, Re,* and Pr_L represent Mach number, Reynold's number, and the laminar Prandtl number respectively, defined using the above reference scales in addition to the mean freestream velocity ($\overline{U}_\infty^*$) and thermal conductivity (k_∞^*). That is,

$$M = \overline{U}_\infty^* / a_\infty^* \tag{2.6a}$$

$$\mathrm{Re} = \rho_\infty^* \overline{U}_\infty^* L_\infty^* / \mu_\infty^* \tag{2.6b}$$

$$\mathrm{Pr}_L = \mu_\infty^* c_p^* / k_\infty^* . \tag{2.6c}$$

The dynamic coefficient of viscosity, μ, is approximated by Sutherland's law in dimensional form as,

$$\mu^* = \frac{A^* T^{3/2}}{T + S^*} \tag{2.7}$$

where A^* and S^* are empirical, dimensional constants. In non-dimensional form,

$$\mu = \frac{(1+S)T^{3/2}}{T+S} \tag{2.8}$$

with $S = S^* T_\infty^*$ and $A^* T_\infty^{1/2} = 1 + S$.

Considering the momentum equation (2.1b), it is assumed the fluid is Newtonian and is thus characterized by a linear stress-strain relationship, expressed by

$$\tau_{ij} = \frac{M}{\mathrm{Re}}\left[\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)+\lambda\frac{\partial u_k}{\partial x_k}\delta_{ij}\right]. \tag{2.9}$$

Terms in the stress tensor expression may be consolidated using Stoke's hypothesis which relates the bulk coefficient of viscosity, λ, to the dynamic coefficient of viscosity, μ, by

$$\lambda = -\frac{2}{3}\mu\ . \tag{2.10}$$

Also seen in the momentum equation is the distributed force, f_i, which equalizes the fluid and body velocities inside the body. This "penalization" force is derived from the Brinkman Penalization method stemming from studies on the permeability of porous media [10]. In the discrete form of the momentum equation, written as

$$\frac{\rho u_i^{n+1}-\rho u_i^{n}}{\Delta t} = RHS_i + f_i\,, \tag{2.11}$$

the force is defined by

$$f_i = \begin{Bmatrix} 0 & outside\,the\,body \\ -RHS_i+\left(\rho v_{bi}^{n+1}-\rho u_i^{n}\right)/\Delta t & inside\,the\,body \end{Bmatrix} \tag{2.12}$$

where RHS_i represents the convective, viscous, and pressure gradient terms. Comparison of equations (2.11) and (2.12) shows that at the end of each time step, the fluid velocity inside the body is equal to the body velocity (v_{bi}), $u_i^{n+1} = v_{bi}^{n+1}$. Using this force to represent the influence of the body in such a fashion is the essence of the "immersed boundary method" implemented in this airfoil study.

Turbulent flows contain an immense amount of physical detail, far beyond the ability to be completely described by computational modeling. The statistical nature of turbulence however, can be described well. The predictability of the average values provides for a more

tractable analysis of turbulence. Hence the flow variables in the governing equations are decomposed into mean and fluctuating quantities:

$$\phi(\vec{x},t) = \bar{\phi}(\vec{x},t) + \phi'(\vec{x},t). \tag{2.13}$$

The prime denotes a fluctuation or perturbation of the quantity and the over-bar denotes the time average defined by

$$\phi(\vec{x}) = \lim_{T\to\infty} \frac{1}{T}\int_0^T \phi(\vec{x},t)dt. \tag{2.14}$$

This decomposition (Reynolds time average) is performed for all the variables except for the velocity components and energy. Time averaging the equations of motion in compressible flow would lead to correlations between velocity and density requiring further information to model these terms [11]. Hence, for velocity, a different form of the Reynolds average, called Favre averaging, is used to eliminate these correlation terms in the averaged equations. In Favre averaging, the variable is decomposed into a mass averaged component (denoted by a tilde) and a fluctuation (denoted by a double prime). That is,

$$\phi(\vec{x},t) = \tilde{\phi}(\vec{x},t) + \phi''(\vec{x},t). \tag{2.15}$$

The mass averaged, or density weighted term is given by,

$$\tilde{\phi}(\vec{x}) = \frac{1}{\bar{\rho}} \lim_{T\to\infty} \frac{1}{T}\int_0^T \rho(\vec{x},t)\phi(\vec{x},t)dt. \tag{2.16}$$

Chopra explains that in both averages, the averaging time is significantly larger than the small eddy turn-over time, but significantly smaller than the large eddy turn-over time. Making this assumption allows the governing equations to retain time-dependent terms that describe the evolution of the low frequency or large scale components of the turbulence.

The following equations show each variable's decomposition according to the above definitions.

.

$$\rho = \bar{\rho} + \rho' \tag{2.17a}$$

$$p = \bar{p} + p' \tag{2.17b}$$

$$q_j = \bar{q}_j + q'_j \tag{2.17c}$$

$$u_i = \tilde{u}_i + u_i'' \tag{2.17d}$$

$$T = \tilde{T} + T'' \tag{2.17e}$$

$$\tau = \tilde{\tau} + \tau'' \tag{2.17f}$$

$$e = \tilde{e} + e'' \tag{2.17g}$$

Once these decompositions are substituted into the governing equations, the time or mass average applied to each term results in the following equations,

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i}\left(\bar{\rho}\tilde{u}_i\right) = 0 \tag{2.18a}$$

$$\frac{\partial}{\partial t}\left(\bar{\rho}\tilde{u}_i\right) + \frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_i\tilde{u}_j + \bar{p}\delta_{ij}\right) = \frac{\partial}{\partial x_j}\left(\tilde{\tau}_{ij} - \overline{\rho u_j'' u_i''}\right) + f_i \tag{2.18c}$$

$$\frac{\partial}{\partial t}\left(\bar{\rho}\tilde{e}_t\right) + \frac{\partial}{\partial x_j}\left[\left(\bar{\rho}\tilde{e}_t + \bar{p}\right)\tilde{u}_j\right] = \frac{\partial}{\partial x_j}\left[\tilde{u}_j\left(\tilde{\tau}_{ij} - \overline{\rho u_j'' u_i''}\right)\right] - \frac{\partial}{\partial x_j}\left(\bar{q}_j + \overline{\rho u_j'' e''}\right) + \frac{\partial}{\partial x_j}\left(\overline{u_i''\tau_{ij}} - \frac{1}{2}\overline{\rho u_j'' u_i'' u_i''}\right) \tag{2.18c}$$

$$\bar{p} = \frac{\bar{\rho}\tilde{T}}{\gamma} \tag{2.19d}$$

where

$$\bar{\rho}\tilde{e}_t = \bar{\rho}\left[\frac{\tilde{T}}{\gamma(\gamma-1)} + \frac{1}{2}\left(\tilde{u}_i\tilde{u}_i\right)\right] + \frac{1}{2}\left(\overline{\rho u_i'' u_i''}\right). \tag{2.19}$$

The last two terms in the energy equation ($\overline{u_i'' \tau_{ij}}$ and $\overline{\rho u_j'' u_i'' u_i''}$) can be neglected as they have been shown to be only applicable to hypersonic flow regimes. Solving additional equations for the remaining terms produced by the averaging process (i.e. Reynolds stress terms)

will only produce additional unknowns, thereby resulting in the turbulence closure problem. Therefore, models must be developed for these terms.

The Reynolds stress tensor term can be modeled using the Boussinesq eddy viscosity approximation, which relates the stress to the mean strain rate according to

$$\tau_{ij} = -\overline{\rho u_j'' u_i''} = \frac{M}{\mathrm{Re}} \mu_t \left[\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial \bar{u}_k}{\partial x_k} \delta_{ij} \right]. \tag{2.20}$$

This approximation reduces the number of unknowns in the stress tensor to one: the turbulent eddy viscosity, μ_t, for which Menter's Shear Stress Tensor (SST) [12] provides closure. The eddy viscosity parameter also appears in the gradient hypothesis used to model the turbulent heat flux term in the energy equation, given by

$$q_{Tj} = \overline{\rho u_j'' e''} = -\frac{1}{\gamma - 1} \frac{M}{\mathrm{Re}} \frac{\mu_t}{\mathrm{Pr}_t} \left(\frac{\partial \tilde{T}}{\partial x_j} \right). \tag{2.21}$$

Then with these approximations and Menter's SST turbulence model, the set of governing equations are closed, and simplify to

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i} \left(\rho \bar{u}_i \right) = 0 \tag{2.22a}$$

$$\frac{\partial}{\partial t} \left(\bar{\rho} \tilde{u}_i \right) + \frac{\partial}{\partial x_j} \left(\bar{\rho} \tilde{u}_i \tilde{u}_j + \bar{p} \delta_{ij} \right) = \frac{\partial}{\partial x_j} \left(\tilde{\tau}_{ij} - \tau_{ij} \right) + f_i \tag{2.22b}$$

$$\frac{\partial}{\partial t} \left(\bar{\rho} \tilde{e}_t \right) + \frac{\partial}{\partial x_j} \left[\left(\bar{\rho} \tilde{e}_t + \bar{p} \right) \tilde{u}_j \right] = \frac{\partial}{\partial x_j} \left[\tilde{u}_j \left(\tilde{\tau}_{ij} - \tau_{ij} \right) \right] - \frac{\partial}{\partial x_j} \left(\bar{q}_j + q_{Tj} \right) \tag{2.22c}$$

$$\bar{p} = \frac{\bar{\rho} \tilde{T}}{\gamma}. \tag{2.22d}$$

Menter's SST Turbulence Model

Modeling the turbulent eddy viscosity is facilitated by Menter's Shear Stress Tensor (SST) model, derived from Wilcox's *k-ω* two-equation model. Menter's approach removes from the original model its strong dependence on free-stream values and includes transport of principal shear stress for adverse pressure gradient boundary layers [12]. The two-equations comprising the model are

$$\frac{\partial}{\partial t}(\rho k)+\frac{\partial}{\partial x_j}(\rho u_j k)=S_{P,k}+S_{D,k}+D_k \tag{2.23a}$$

$$\frac{\partial}{\partial t}(\rho \omega)+\frac{\partial}{\partial x_j}(\rho u_j \omega)=S_{P,\omega}+S_{D,\omega}+D_\omega \tag{2.23b}$$

where the Production (S_P), Dissipation (S_D) and Diffusion (D) terms for each equation are defined by,

$$S_{P,k}=\mu_t \Omega^2\left(\frac{M}{\mathrm{Re}}\right) \tag{2.24}$$

$$S_{D,k}=-\rho\beta' k\omega\left(\frac{M}{\mathrm{Re}}\right) \tag{2.25}$$

$$D_k=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]\left(\frac{M}{\mathrm{Re}}\right) \tag{2.26}$$

$$S_{P,\omega}=\rho\gamma\Omega^2\left(\frac{M}{\mathrm{Re}}\right) \tag{2.27}$$

$$S_{D,\omega}=-\rho\beta\omega^2\frac{\mathrm{Re}}{M}+2\rho(1-F_1)\sigma_{\omega_2}\frac{1}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}\left(\frac{M}{\mathrm{Re}}\right) \tag{2.28}$$

$$D_\omega=\frac{\partial}{\partial x_j}\left[\left(\mu+\frac{\mu_t}{\sigma_k}\right)\frac{\partial \omega}{\partial x_j}\right]\left(\frac{M}{\mathrm{Re}}\right) \tag{2.29}$$

Values for the parameters in the above definitions can be found in Menter [12]. Finally, the turbulent eddy viscosity can be evaluated from the kinetic energy and specific dissipation rate, according to

$$\mu_t = \min\left[\frac{\rho k}{\omega}, \frac{a_1 \rho k}{\Omega F_2}\left(\frac{\mathrm{Re}}{M}\right)\right]. \tag{2.30}$$

This overview of the equations governing the fluid motion provides the framework on which the current research study is built. It was shown how the compressible Navier-Stokes equations were averaged, using both Reynolds and Favre averages, for exploiting the predictability of averaged quantities. In so doing, extra terms appear in the framework, which are modeled using Menter's SST turbulence model. The following chapter explains how these equations are transformed to discrete space for computation.

Chapter 3

Numerical Implementation

The governing equations discussed in the previous chapter are now discretized for numerical simulation. This chapter begins with a description of the stretched Cartesian grid on which the computations are performed, followed by an explanation of how the equations are transformed to a uniform grid through generalized coordinates. Details for the discretization process are provided, including the construction of the Weighted Essentially Non-Oscillatory (WENO) scheme used for spatial derivatives and the 4 stage Runge-Kutta scheme used for temporal derivatives. Concluding the chapter is a summary of the use of the body definition function, along with initial and boundary conditions used for the simulations.

Computational Grid

CARIBOU solves the two-dimensional governing equations on a stretched Cartesian grid in physical space. The stretching is formulated using trigonometric hyperbolic functions with parameters defining the extent of clustering. Various degrees of stretching can be used on the grid in both *x*- and *y*-directions according to the level of refinement necessary in that particular region. Figure 3.1(a) shows the stretched grid used around a NACA0009 airfoil. Sections in the *x*-direction, around the leading and trailing edges, are more finely resolved compared to outer regions, as is the section surrounding the airfoil between -0.1 and 0.1 in the *y*-direction. Clustering the mesh in this fashion provides a reasonable resolution of the boundary layer. The computational grid extends from -20 to 20 in both directions corresponding to 40 total chords, so the figure shows a close-up view of the grid in the vicinity of the airfoil body. Figure 3.1(b)

shows the same airfoil, but with a semi-circular portion removed to simulate a damaged state. The mesh was adjusted to be uniform in the *x*-direction surrounding this damage to achieve sufficient smoothness of the curve, as shown in Figure 3.1(c). In both cases, the number of grid points was 330 in the *x*-direction and 380 in the *y*-direction.

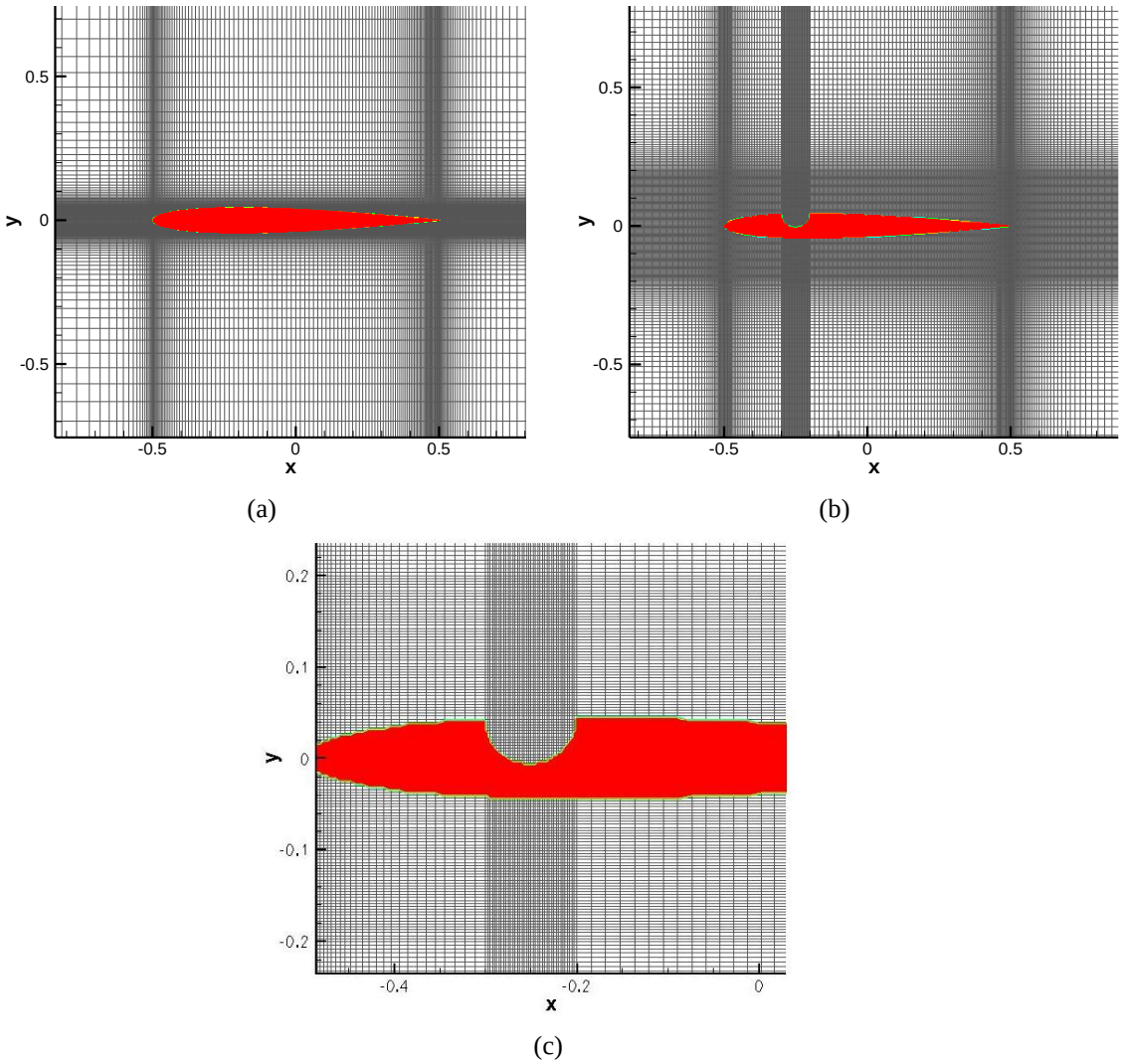


Figure 3.1: Stretched Cartesian grid used for (a) undamaged and (b) damaged NACA 0009 airfoil; (c) uniform grid in x-direction in vicinity of damage

This grid was used in both serial and parallel versions of CARIBOU. Parallel computation significantly enhanced the efficiency of the simulations. Twelve processors on COCOA3, a cluster of 128 nodes and dual 2.4 GHz processors, each having 1 GB RAM have been used. In parallel simulations, the domain is broken into blocks according to the number of processors and the number of grid points in each of the coordinate directions. For 12 processors and the previously stated number of grid points, the domain is divided into 4 blocks in the *x*-direction and 3 blocks in the *y*-direction. Message Passage Interfacing (MPI) routines are used to establish communication among these blocks.

Discretization of Equations

Discretization of the governing equations requires the physical grid in (x,y) coordinates to be transformed to a computational domain with coordinates (ξ,η). Transforming the physical coordinates to this generalized coordinate system can be explained by considering the governing equations in vector form (with no source terms)

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \boldsymbol{E}}{\partial x} + \frac{\partial \boldsymbol{F}}{\partial y} = 0 \tag{2.31}$$

where,

$$\boldsymbol{Q} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e_t \end{Bmatrix} \tag{2.32a}$$

$$\boldsymbol{E} = \begin{Bmatrix} \rho u \\ \rho uu + p + \tau_{xx} \\ \rho uv - \tau_{xy} \\ (\rho e_t + p)u - u\tau_{xx} - v\tau_{xy} + q_x \end{Bmatrix} \tag{2.32b}$$

$$\boldsymbol{F} = \begin{Bmatrix} \rho v \\ \rho uv + p + \tau_{yx} \\ \rho vv - \tau_{yy} \\ (\rho e_t + p)v - u\tau_{yx} - v\tau_{yy} + q_y \end{Bmatrix}. \tag{2.32c}$$

Subscripts in this definition, and those following, represent partial derivatives. The spatial derivatives can be expanded in terms of the generalized coordinates according to,

$$\frac{\partial E}{\partial x} = \frac{\partial E}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial E}{\partial \eta}\frac{\partial \eta}{\partial x} \tag{2.33a}$$

$$\frac{\partial F}{\partial y} = \frac{\partial F}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial F}{\partial \eta}\frac{\partial \eta}{\partial y} \tag{2.33b}$$

After expanding these terms, the entire equation can be multiplied by the Jacobian of the transformation defined as,

$$J = x_\xi y_\eta - x_\eta y_\xi . \tag{2.34}$$

Note that,

$$J\begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = \begin{bmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{bmatrix}, \tag{2.35}$$

Manipulation of the resulting terms leads to the following equation.

$$\frac{\partial}{\partial t}(JQ) + \frac{\partial}{\partial \xi}\left[J(E\xi_x + F\xi_y)\right] + \frac{\partial}{\partial \eta}\left[J(E\eta_x + F\eta_y)\right] + E\left[\frac{\partial}{\partial \eta}(y_\xi) - \frac{\partial}{\partial \xi}(y_\eta)\right] + F\left[\frac{\partial}{\partial \xi}(x_\eta) - \frac{\partial}{\partial \eta}(x_\xi)\right] = 0 \tag{2.36}$$

The last two bracketed terms contain second-order mixed derivatives which are equal on the basis of continuity. Thus these terms cancel, leaving the transformed equation,

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} = 0 \tag{2.37}$$

where the transformed solution vector ($\hat{\boldsymbol{Q}}$) and flux vectors ($\hat{\boldsymbol{E}}$, $\hat{\boldsymbol{F}}$) are defined as

$$\hat{\boldsymbol{Q}} = J\boldsymbol{Q} \tag{2.38a}$$

$$\hat{\boldsymbol{E}} = J\left(\boldsymbol{E}\xi_x + \boldsymbol{F}\xi_y\right) \tag{2.38b}$$

$$\hat{\boldsymbol{F}} = J\left(\boldsymbol{E}\eta_x + \boldsymbol{F}\eta_y\right). \tag{2.38c}$$

WENO Scheme

Transformation to a uniform grid is necessary for the use of a Weighted Essentially Non-Oscillatory (WENO) scheme to represent the fluxes in discrete form. Due to the presence of the body, discontinuities arise in conventional differencing schemes. These contaminate the solution with numerical oscillations. Implementation of the WENO scheme reduces these numerical oscillations occurring at the fluid-body interface and in regions of rapid change (such as close to a shock) [13].

Letting *q* represent a flux in either direction, the derivative of the flux at node *i* is defined by

$$\left(\frac{\partial q}{\partial \xi}\right)_{\xi=\xi_i} = \frac{\tilde{q}_{i+1/2} - \tilde{q}_{i-1/2}}{\Delta\xi}. \tag{2.39}$$

CARIBOU uses a 5th-order WENO scheme where the flux, *q*, is weighted according to

$$\tilde{q}_{i+1/2} = w_1\tilde{q}^{(1)}_{i+1/2} + w_2\tilde{q}^{(2)}_{i+1/2} + w_3\tilde{q}^{(3)}_{i+1/2} \tag{2.40}$$

and each of the weighted fluxes are defined by,

$$\tilde{q}_{i+1/2}^{(1)} = \frac{1}{3}\tilde{q}_{i-2} - \frac{7}{6}\tilde{q}_{i-1} + \frac{11}{6}\tilde{q}_i \tag{2.41a}$$

$$\tilde{q}_{i+1/2}^{(2)} = -\frac{1}{6}\tilde{q}_{i-1} + \frac{5}{6}\tilde{q}_i + \frac{1}{3}\tilde{q}_{i+1} \tag{2.41b}$$

$$\tilde{q}_{i+1/2}^{(3)} = \frac{1}{3}\tilde{q}_i + \frac{5}{6}\tilde{q}_{i+1} - \frac{1}{6}\tilde{q}_{i+2}\,. \tag{2.41c}$$

Fluxes on the RHS of these expressions are averages at the respective node, where Lax-Friedrich flux splitting is used for the computation. Specifically, the flux q is split into positive and negative components [5],

$$q = q^+ + q^- \tag{2.42}$$

where

$$(q^+)' \geq 0\,, (q^+)' \leq 0\,. \tag{2.43}$$

The individual weights in the expressions are computed according to,

$$w_j = \alpha_j \Big/ \sum_{k=1}^{3} \alpha_k\,, \quad \alpha_k = \frac{d_k}{(\varepsilon + \beta_k)^2} \tag{2.44}$$

where

$$d_1 = 1/10, \quad d_2 = 3/5, \quad d_3 = 3/10\,. \tag{2.45}$$

Lastly, the β_k terms in Equation (2.38) are smoothness indicators and are represented by

$$\beta_1 = \frac{13}{12}(q_{i-2} - 2q_{i-1} + q_i)^2 + \frac{1}{4}(q_{i-2} - 4q_{i-1} + 3q_i)^2 \tag{2.46a}$$

$$\beta_2 = \frac{13}{12}(q_{i-1} - 2q_i + q_{i+1})^2 + \frac{1}{4}(q_{i-1} - q_{i+1})^2 \tag{2.46b}$$

$$\beta_3 = \frac{13}{12}(q_i - 2q_{i+1} + q_{i+2})^2 + \frac{1}{4}(3q_i - 4q_{i+1} + q_{i+2})^2\,. \tag{2.46c}$$

Runge-Kutta Scheme

Time derivative terms are discretized according to a 4th-order explicit Runge-Kutta (RK) scheme. The formulation of this scheme for one dimension, using the variables introduced previously, can be written as,

$$\frac{\boldsymbol{Q}' - \boldsymbol{Q}^{\boldsymbol{n}}}{\Delta t} = a_1 \frac{\partial \boldsymbol{E}^{n}}{\partial x} \tag{2.47a}$$

$$\frac{\boldsymbol{Q}'' - \boldsymbol{Q}'}{\Delta t} = a_2 \frac{\partial \boldsymbol{E}'}{\partial x} \tag{2.47b}$$

$$\frac{\boldsymbol{Q}''' - \boldsymbol{Q}''}{\Delta t} = a_3 \frac{\partial \boldsymbol{E}''}{\partial x} \tag{2.47c}$$

$$\frac{\boldsymbol{Q}^{n+1} - \boldsymbol{Q}'''}{\Delta t} = a_4 \frac{\partial \boldsymbol{E}}{\partial x}. \tag{2.47d}$$

The four coefficients, a_1, a_2, a_3, a_4, are assigned the standard values (1/4, 1/3, ½, 1) respectively for time-accurate simulations, but for faster convergence in steady-state simulations, the values (3/8, ½, 1, 1) are assigned [14].

Body Definition Function

CARIBOU defines the region containing the body using a Body Definition Function (BDF) which assigns a value of 0 or 1 to each grid point in the domain depending on whether that point lies outside or inside, respectively, a prescribed boundary. So, in Figure 3.1 for example, points inside the airfoil (colored red) would have a BDF value of 1, while points outside (colored gray) would have a BDF value of 0.

Modeling damage can be achieved by prescribing an additional boundary, superimposed over the airfoil, along with an additional condition to check before assigning a value to a

particular point. This additional check switches the value of a body point to zero if this point also falls within the superimposed boundary. Any number of damage shapes and sizes can be prescribed this way, as well as at any location. The semi-circular shape used in this preliminary study was selected for simplicity, and the quarter chord location on the airfoil's upper surface was selected as a position that was expected to cause the most notable effect in the results. The radius for the damage shown in Figure 3.1 is 0.05, which is 5% of the airfoil chord.

Initial Conditions

Several initial conditions are available for selection in CARIBOU. An integer value is assigned to the variable *ic_case* in the input file according to the desired condition. Each condition is listed in Table 3.1 with its corresponding value.

Table 3.1: Selections for initial conditions

ic_case	Corresponding initial condition
1	Gaussian pulse
2	Quiescent condition
3	Quiescent condition with body velocity equal to grid velocity (in moving grid simulations)
4	Initial uniform mean flow
5	Initial uniform mean flow with zero velocity inside body.
11	Restart condition

For each simulation reported in the following chapter, the initial condition used was that corresponding to *ic_case* = 5, uniform mean flow with zero velocity inside the body.

Boundary Conditions

Boundary conditions (BC) are selected in CARIBOU in the same fashion as the initial conditions. Namely the variable *bc_case* is given a value in the input file according to the following definitions in Table 3.2.

Table 3.2: Selections for boundary conditions

bc_case	Corresponding boundary condition
1	Periodic BC, imposed in the inflow and outflow
2	Mean flow BC, imposed in the inflow 3 points
3	Riemann BC, imposed in the inflow and outflow
31	Riemann BC, imposed in all directions (in, out, bottom, top)
4	Mean flow BC, imposed in the inflow
5	left: mean flow BC right: extrapolation BC bottom: partly symmetric, partly no-slip BC top: extrapolation BC

For each simulation in the present research, the boundary condition corresponding to *bc_case* = 5 was used. For the left and bottom boundaries, each variable is imposed, and for the right and top boundaries, they are extrapolated. Along the bottom boundary up to a specified point, the symmetric condition imposes values on each node which are equal to the neighboring, interior point values. Beyond this specified point, the velocity components are set to zero while the symmetric condition is imposed for the remaining variables.

By embedding multiple selections of initial and boundary conditions within the code, numerous simulations using different conditions can be performed to see the effect such conditions have on the results. The numerical results presented in the following chapter show this effect for the conditions chosen. Having presented the numerical implementation here, including a discussion of the stretched Cartesian grid, the transformation to a generalized computational grid, and the manner by which the governing equations have been discretized, the groundwork has been provided for the numerical results of the following chapter to be presented.

Chapter 4

Numerical Results

Results of the developments discussed through the previous two chapters are now presented. Various simulation cases are provided, showing time histories of lift, drag, and moment coefficients of an airfoil in damaged and undamaged states. Experimental data for these coefficients are shown for comparison; in those cases where the data exists. Contours of Mach number and turbulent kinetic energy are also presented to show the time evolution of the simulations. These results are drawn upon in the following chapter for conclusions on how CARIBOU performed in this study along with avenues for future development.

An NACA 0009 airfoil was the primary body shape used for running 2D simulations with CARIBOU. The simulation cases covered three angles-of-attack of the airfoil (0°, 4°, 9°) for both undamaged and damaged states, for a total of six configurations. Angle-of-attack is specified as a parameter in the input file to the main code. Adjusting this value causes the flow to be angled relative to the airfoil, which itself stays aligned with the grid (see Figure 3.1). Keeping the airfoil aligned with the grid and rotating the flow allows the near wall regions to be well resolved. In contrast, if the airfoil were rotated relative to the flow, portions of the surface would approach coarser regions of the grid thereby decreasing flow resolution.

The first simulation ran the undamaged airfoil at zero angle of attack (AOA), with a Mach number of 0.2 and Reynolds number of 100,000. Figure 4.1 shows values of the aerodynamic coefficients for this case. This was a time accurate simulation using a CFL of 0.5 on 12 processors. The time scale, t, used in the figure is nondimensionalized by free stream quantities as discussed previously. Using a sound speed of approximately 340 m/s, the corresponding dimensional scale (t^*) would cover approximately the first 0.5 seconds in physical

time. The results show an initial transient solution that progresses to a nearly steady state behavior, where there are zero lift and moment coefficients and a drag coefficient of approximately 0.025. Relative to the experimental data shown in the drag polar of Figure 4.2(b), this steady state drag is approximately twice than the minimum drag coefficient measured at a Reynolds number of 1.0×10^5.

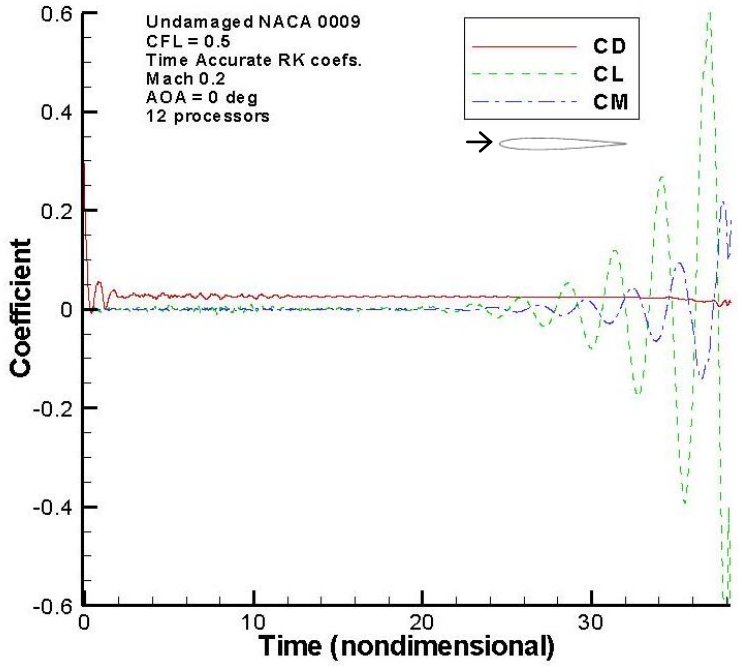
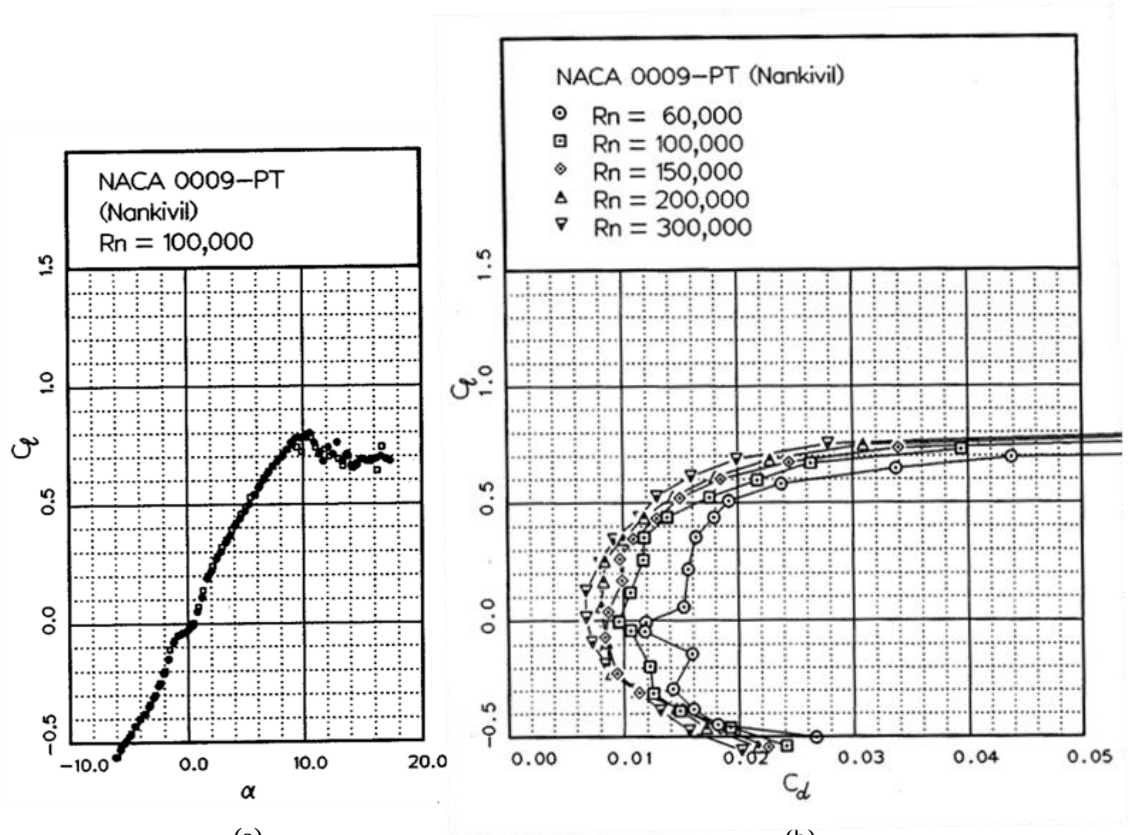


Figure 4.1: Aerodynamic coefficient results: Undamaged NACA 0009, 0° AOA, Re = 1.0×10^5



(a) (b)

Figure 4.2: (a) NACA 0009 Lift Curves[15] (b) NACA 0009 Drag Polar[15]

From t = 20 (t^* = 0.3 s) onwards, the simulation developed non-physical oscillations that indicates an underlying instability in the code. This can also be seen in Figure 4.3 where flow solutions were selected from each period to show contours of Mach number (x-component) and nondimensional turbulent kinetic energy within the vicinity of the airfoil. The last row of the figure shows a wavering energy trail, which is an artifact of the computational instability rather than a physical instability. At 0° AOA, the flow should remain in a steady state condition as shown in the middle row of the figure.

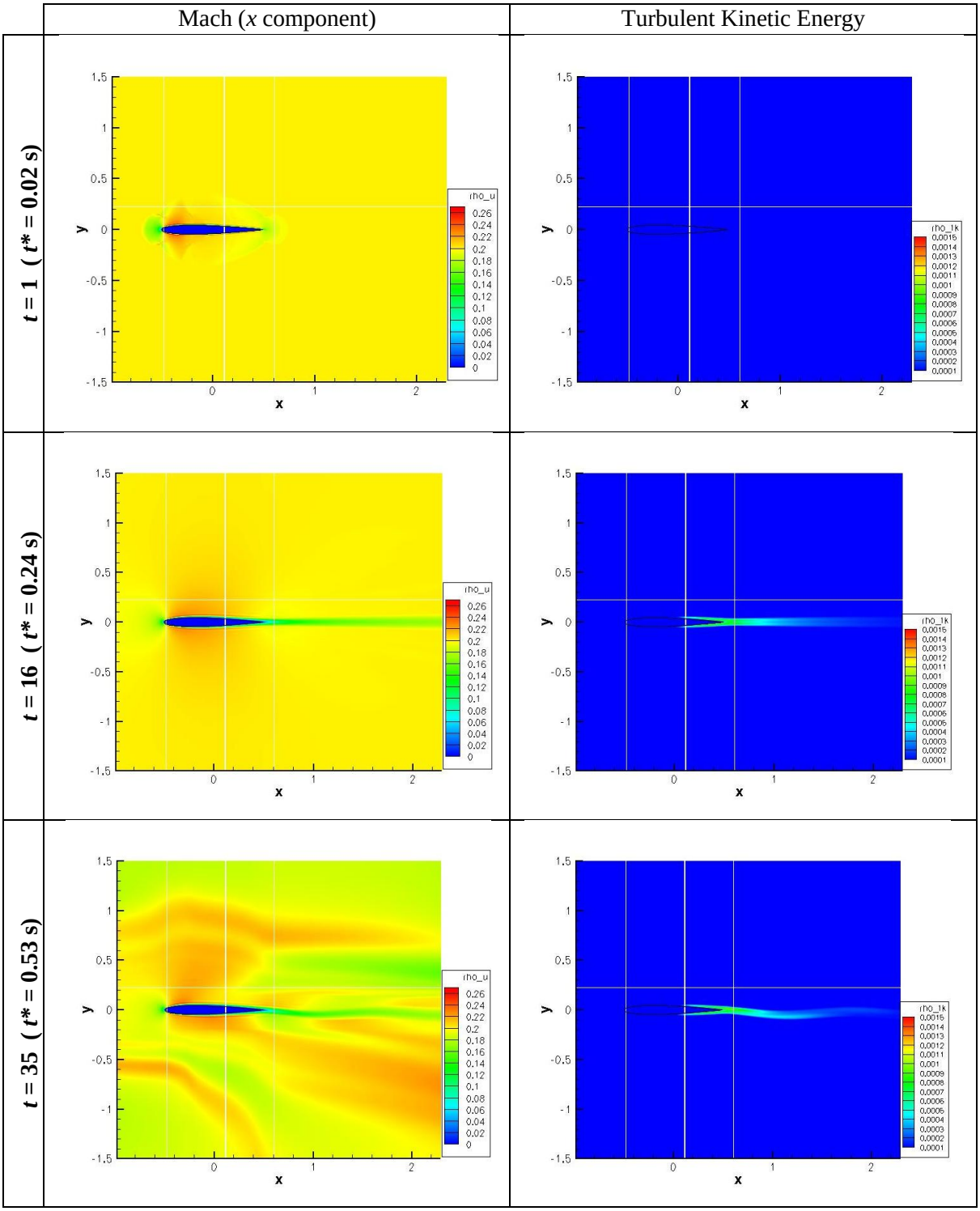


Figure 4.3: Mach and turbulent kinetic energy contours of undamaged NACA 0009 airfoil at 0° AOA during transient, steady, and unstable simulation times

In comparison to the damaged airfoil under the same conditions, the same three regions (initial transient, steady state, and oscillatory) are observed in the results, as shown in Figure 4.4. The only difference is a slight reduction in lift.

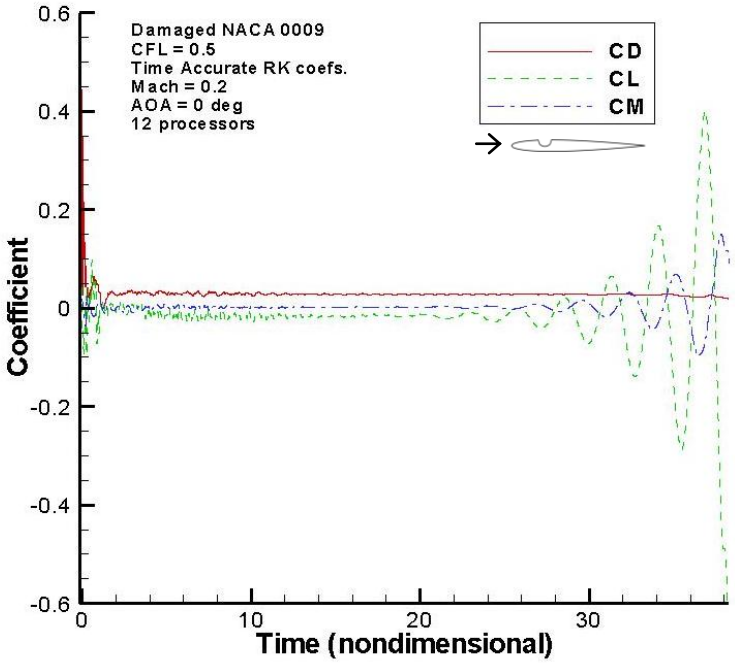


Figure 4.4: Aerodynamic coefficient results: Damaged NACA 0009, 0° AOA, Re = 1.0×10^5

Contours of Mach number and turbulent kinetic energy are shown in Figure 4.5 for the three separate periods in this simulation. An effect of the damage can be seen in the steady state energy plot (middle row) with turbulent energy being generated at the back edge of the cutout and trailing along the upper surface. In comparison to the undamaged case at the same time period, the same level of turbulent energy is produced farther aft on the upper surface and is symmetrical with the lower surface. The final plots on the last row of the figure correspond to the unstable region which is emphasized by points of strong energy concentration being shed from the airfoil. From the previous results of the undamaged airfoil in this region, the zero degree case has an

inherent instability embedded within it which prevents the effects of damage from being accurately described by these results.

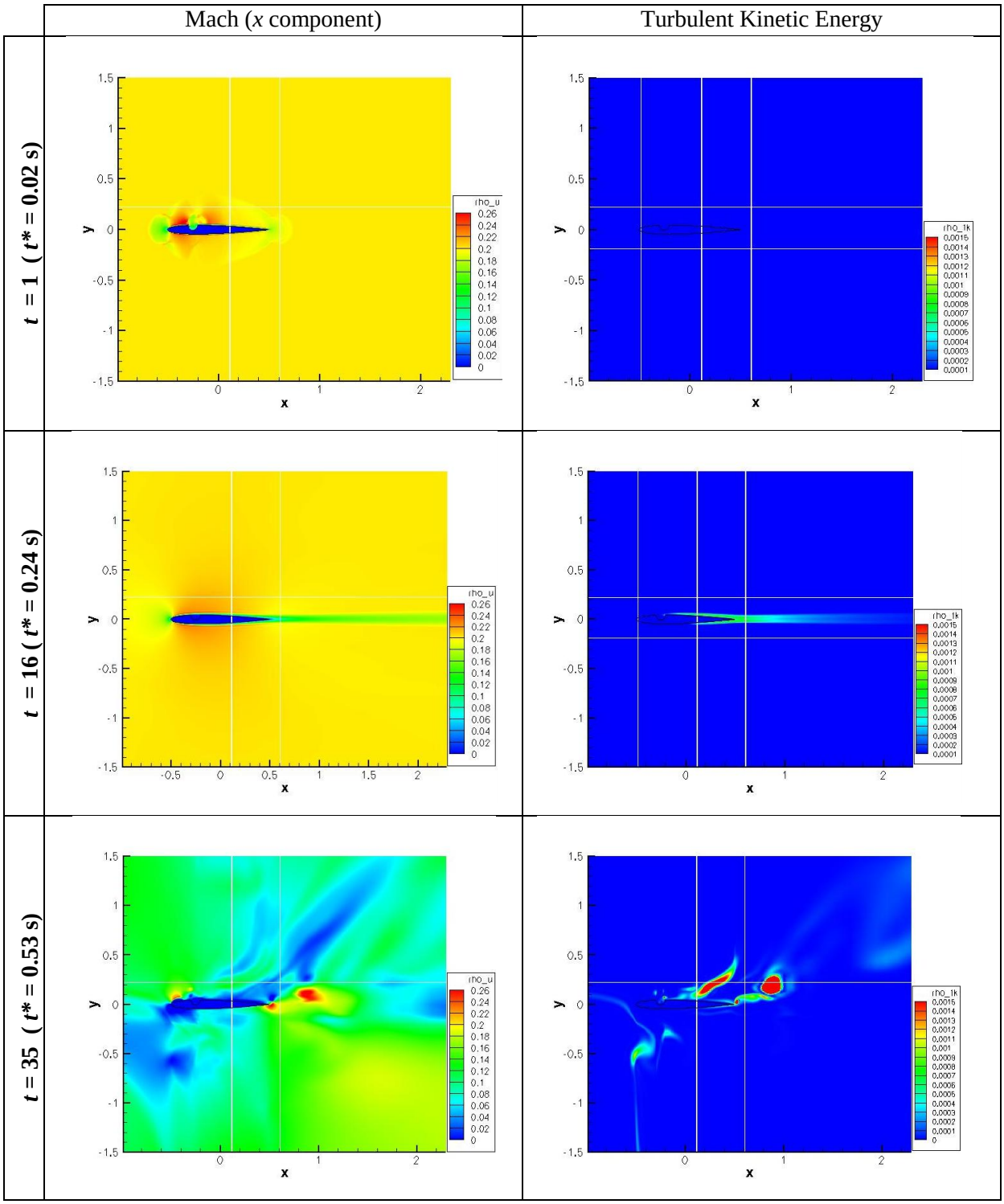


Figure 4.5: Mach and turbulent kinetic energy contours of damaged NACA 0009 airfoil at 0° AOA during transient, steady, and unstable simulation times

A second simulation looked at the undamaged and damaged cases of the same NACA 0009 airfoil at 4° AOA. Experimental data is presented in Table 4.1 for the undamaged state, with interpolated values for 4° AOA.

Table 4.1: Experimental lift and drag coefficients for a NACA0009 airfoil[15] and (linearly interpolated values) at 4° AOA

α	C_L	C_D
3.15	0.353	0.0118
(4.00)	(0.424)	(0.0134)
4.19	0.440	0.0138

Figure 4.6 shows CARIBOU's results for the undamaged case. There is not a long steady state region prior to the simulation going unstable, but the initial transient portion does settle at values closely matching the experimental values for lift. The drag coefficient is approximately 0.04 which again is significantly larger than the experimental value of 0.01.

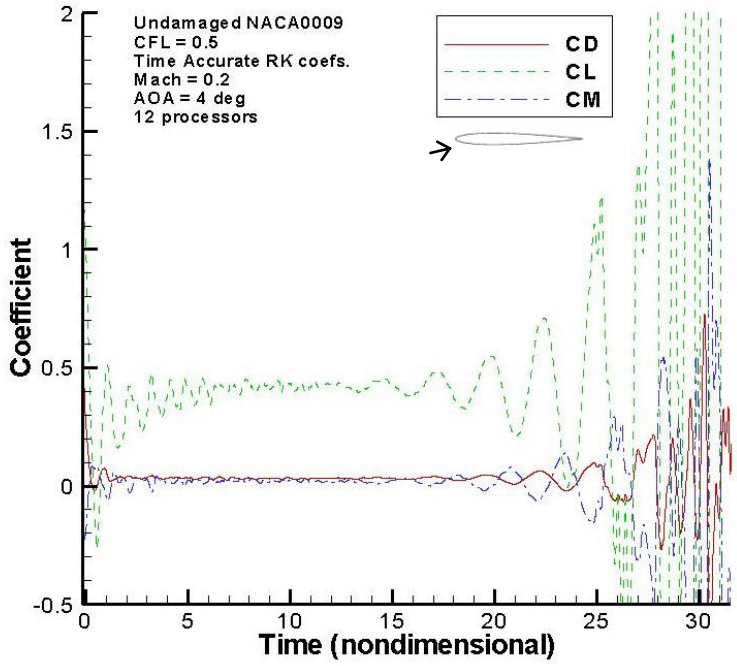


Figure 4.6: Aerodynamic coefficient results: Undamaged NACA 0009, 4° AOA, Re = 1.0x10^5

Figure 4.7 presents flow solutions for this case at times in the transient region, the steady region, and the unstable region to show the progression of the simulation through each of these periods. The first row of pictures corresponds to the transient phase, where the flow has not yet settled. The second row corresponds to approximately 0.15 seconds into the simulation (t = 10), where the flow has settled to an approximate steady state. Considering the Mach contours, the angle of attack can be seen by the flow trailing upwards aft of the airfoil. Also, the flow is accelerating over the airfoil as would be expected. The turbulent kinetic energy for this time shows several vortical structures beginning to shed from the upper surface. These structures intensify as seen in the final row, corresponding to the region with strong oscillations near the end of the simulation. The Mach number in this region shows an increase in velocity fluctuations which is obviously unstable as the steady state contour should still be seen at this time in the simulation.

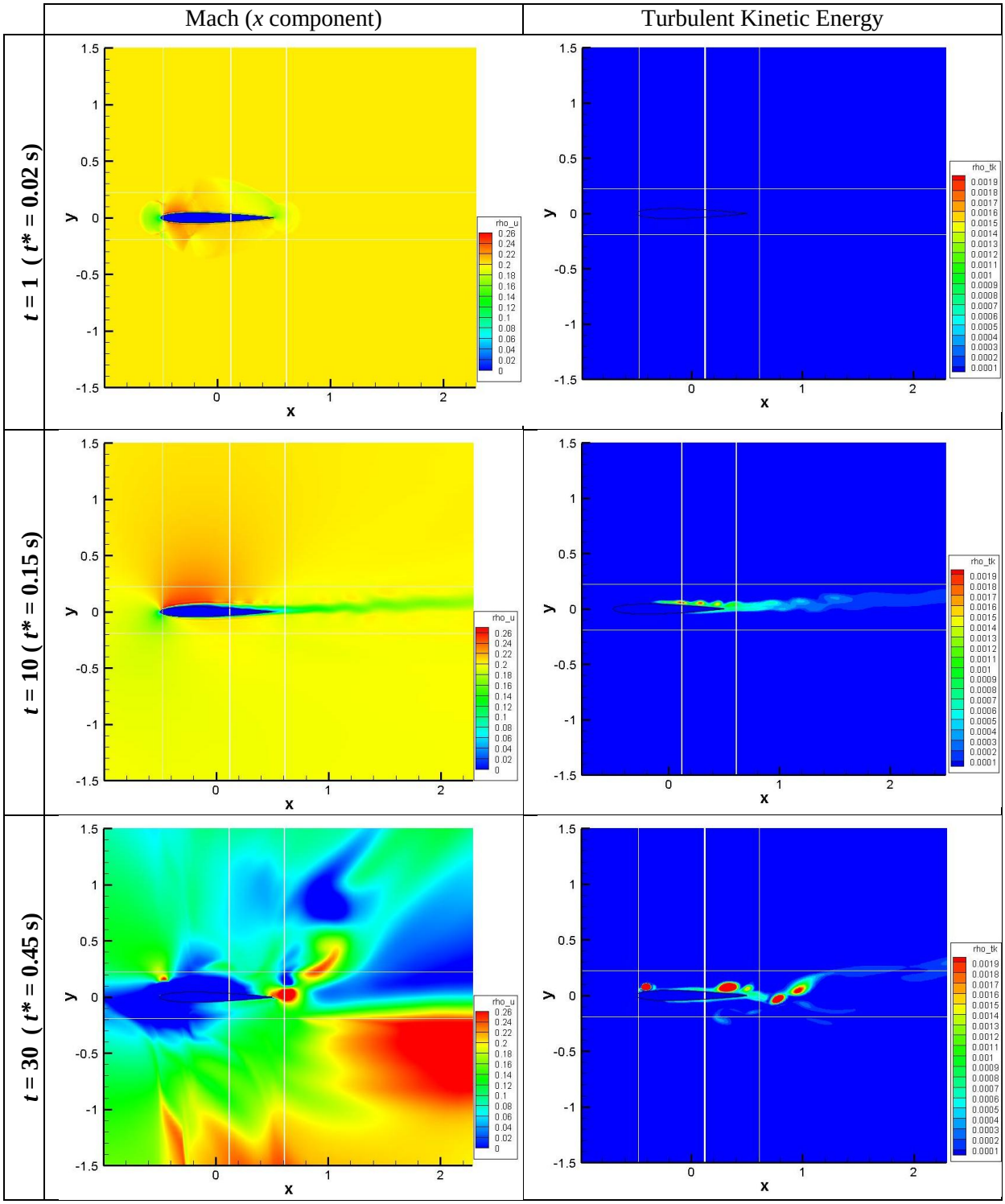


Figure 4.7: Mach and turbulent kinetic energy contours of undamaged NACA 0009 airfoil at 4° AOA

Considering the damaged state for this case, there is no apparent effect on the aerodynamic characteristics shown in Figure 4.8. The time history of these quantities shows the same trends and values as the undamaged state.

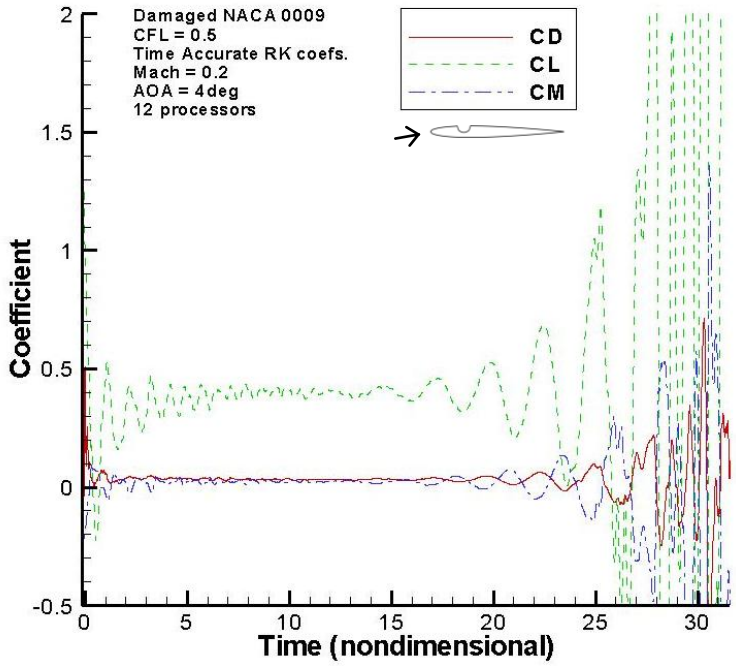


Figure 4.8: Aerodynamic coefficient results: Damaged, NACA 0009, 4° AOA, Re = 1.0×10^5

Minor effects from the damage can be seen in the contours of the *x*-component of Mach number and the nondimensional turbulent kinetic energy, shown in Figure 4.9. In particular, the Mach contours have more oscillations in the wake in the steady state period than the undamaged case. This is due to the cutout shedding small vortices downstream, as was seen in the 0° AOA results.

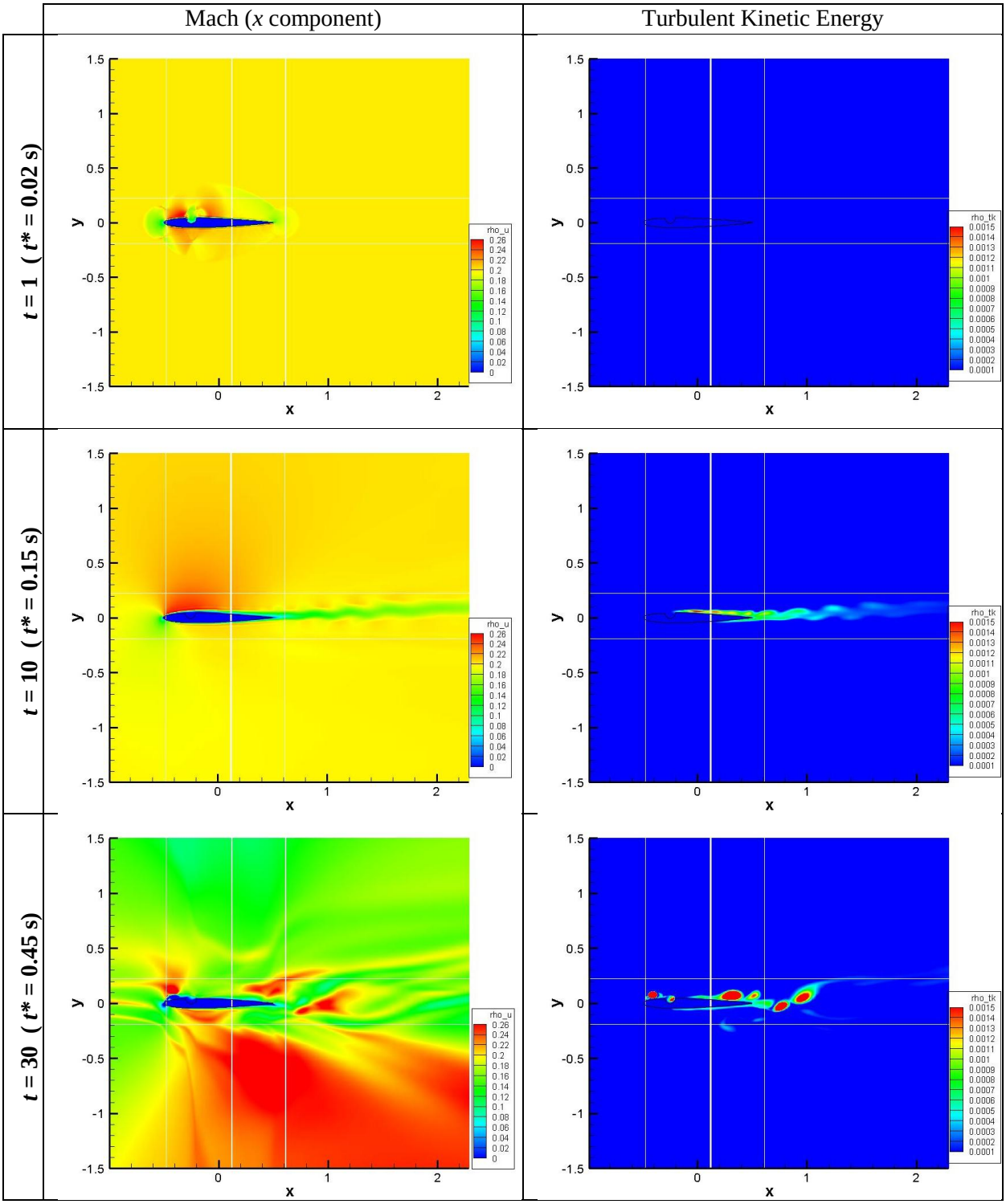


Figure 4.9: Mach and turbulent kinetic energy contours of damaged NACA 0009 airfoil at 4° AOA

The final simulation considered the undamaged and damaged states of the NACA 0009 airfoil near its stall angle of 9° AOA. Figure 4.10 shows clearly different profiles for the aerodynamic coefficients than those at lower angles of attack. Namely, there is a rise in lift and drag and a corresponding dip in moment where there was otherwise steady state behavior. Again however, the simulation transitions into growing oscillations that eventually produce "Not a Number" (NaN) as a result. This progression can be seen in the contours of Mach number and turbulent kinetic energy for the initial, intermediate, and unstable periods of the simulation in Figure 4.11.

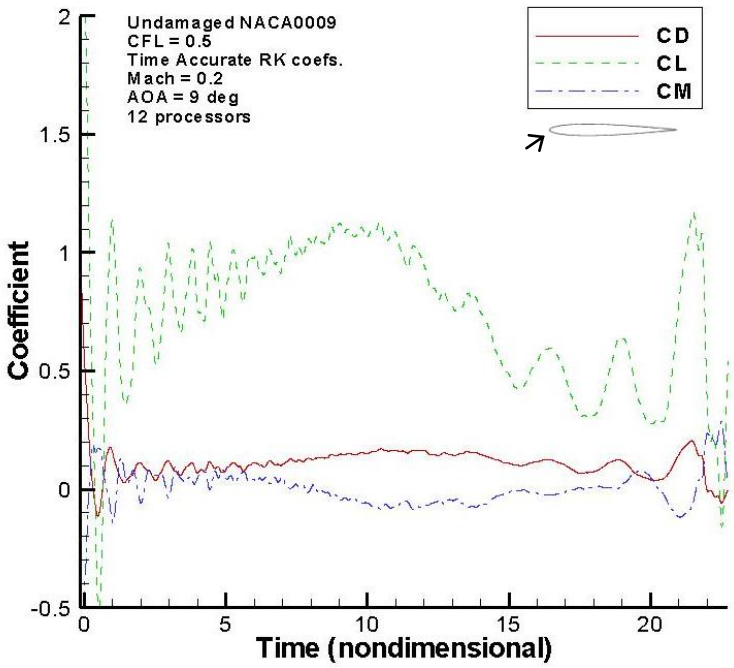


Figure 4.10: Aerodynamic coefficient results: Undamaged NACA 0009, 9° AOA, Re = 1.0×10^5

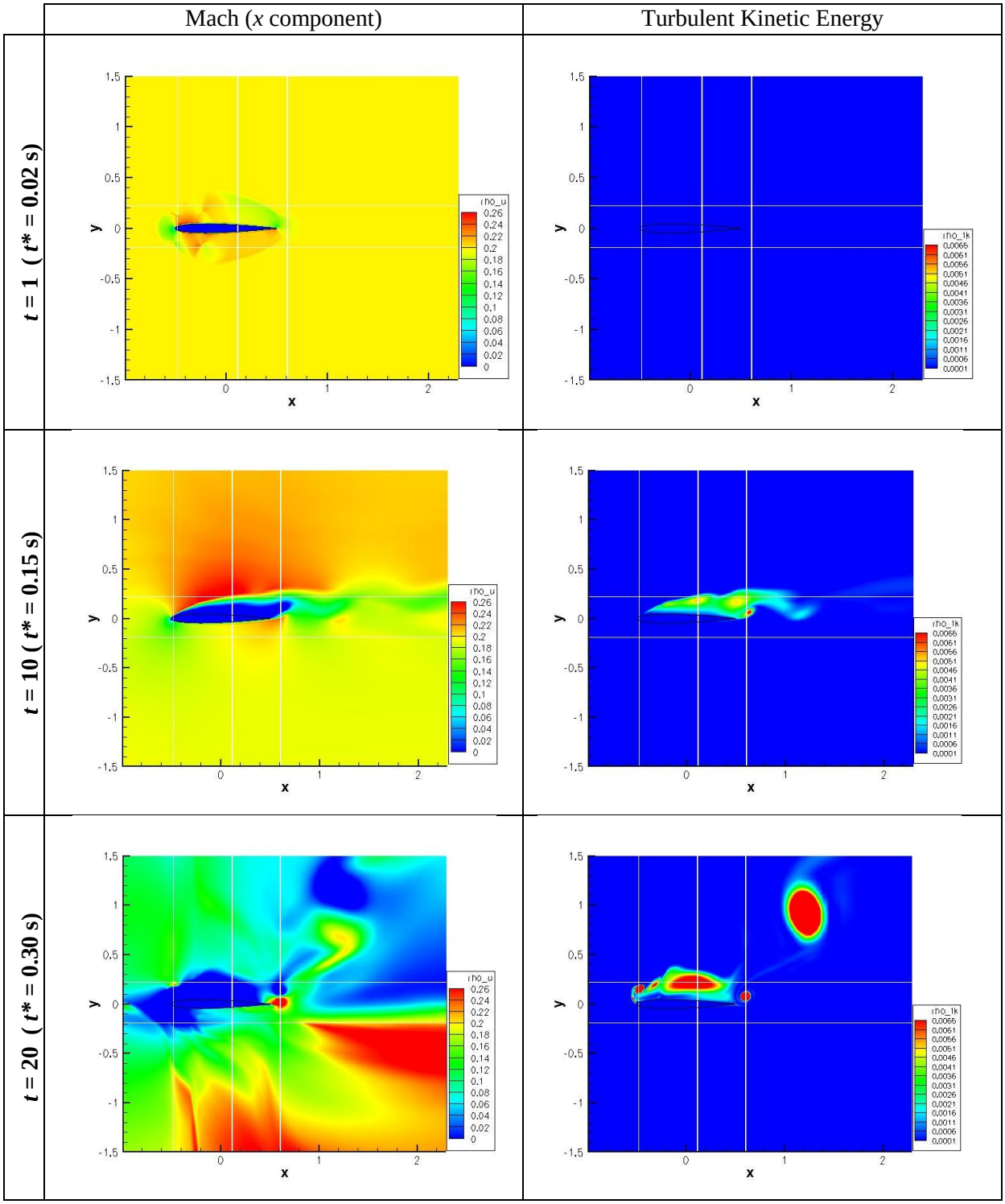


Figure 4.11: Mach and turbulent kinetic energy contours of undamaged NACA 0009 airfoil at 9° AOA during transient, intermediate, and unstable simulation times

Similar results for the damaged airfoil at 9° AOA can be seen in the coefficient time histories in Figure 4.12 and the contours of Figure 4.13.

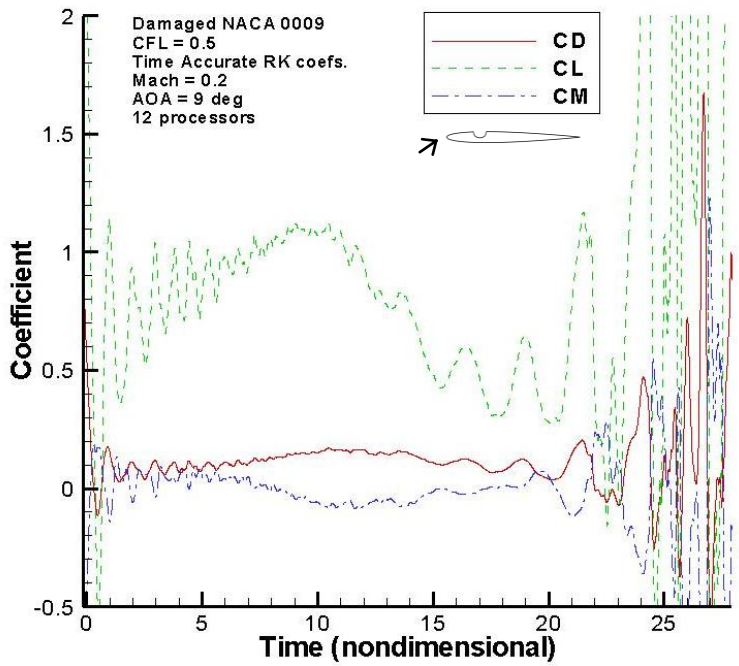


Figure 4.12: Aerodynamic coefficient results: Damaged NACA 0009, 9° AOA, Re = 1.0×10^5

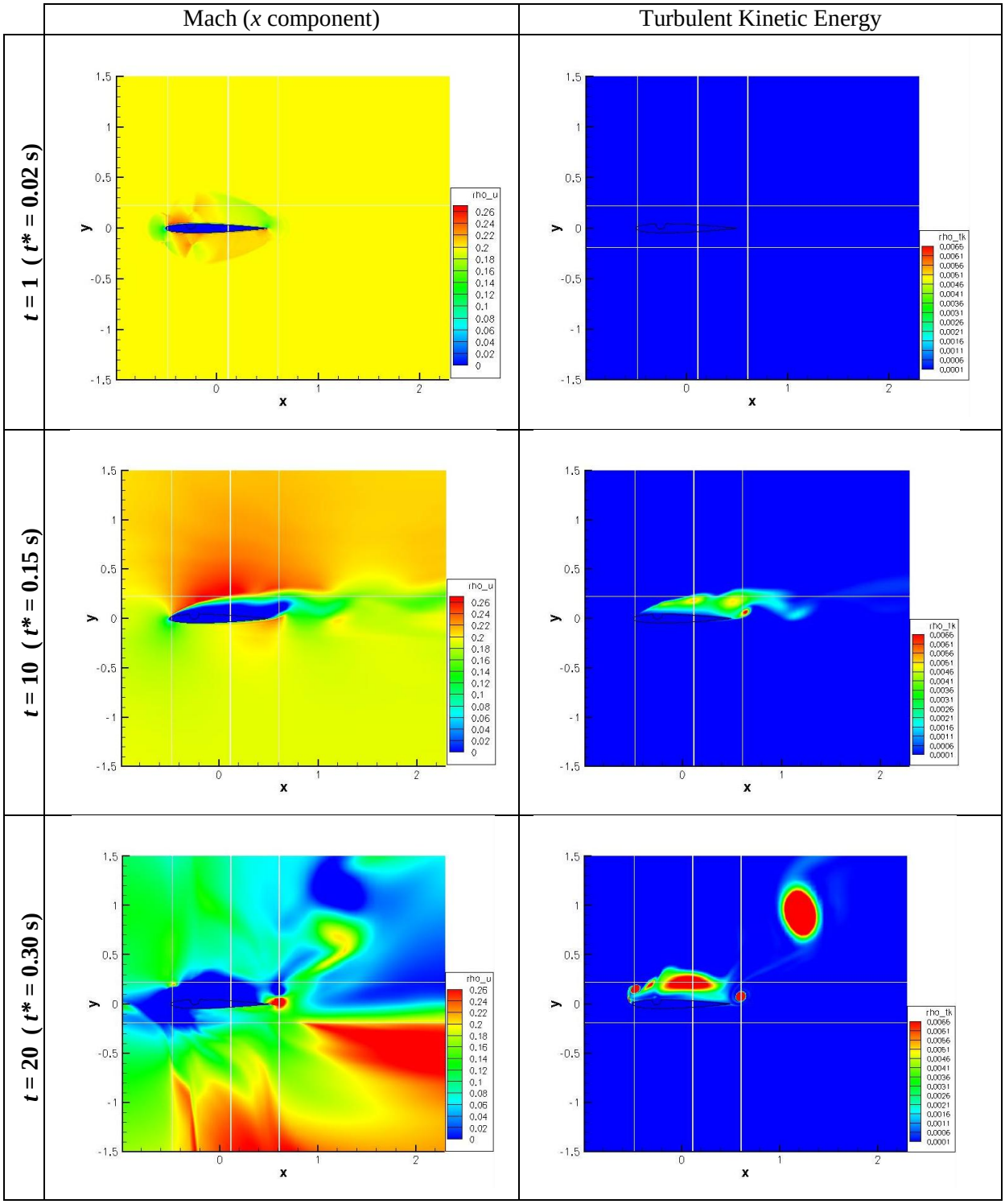


Figure 4.13: Mach and turbulent kinetic energy contours of damaged NACA 0009 airfoil at 9° AOA during transient, intermediate, and unstable simulation times

The results of the damaged and undamaged cases near the stall angle show very little difference. At the stall angle, the flow has detached from the upper surface prior to reaching the damage location so any effects of the cutout will not be propagated downstream. This would likely change with different shapes of damage, such as portions of the lower surface removed or sections of both upper and lower surfaces removed to simulate a hole through the wing.

According to the results presented, there is a numerical instability that contaminates the solution. This instability appears as large oscillations in the aerodynamic coefficients at large times. Clearly, the source of the instability should be the focus of future work, but it is proposed here that the source lies with boundary conditions. Boundary conditions often need more attention in time-accurate compressible codes such as used here. Compressible disturbances propagating to the domain boundaries must be handled correctly, or they may reflect and contaminate the interior flow.

It is likely in these results that disturbances are reflecting from the boundaries rather than radiating out of the domain as they should. To check this, the zero degree case was re-executed using a smaller domain, measuring 10 chords by 10 chords rather than the original 40 by 40 chords. Figure 4.14 shows the comparison of lift coefficients between these cases.

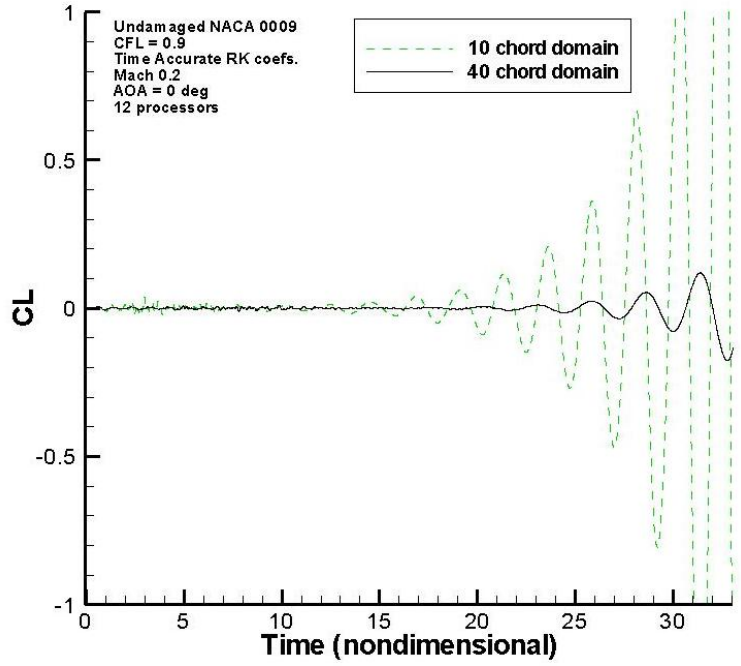


Figure 4.14: Comparison of lift using two different size computational domains

Disturbances have less distance to travel to the boundary in the smaller domain, and less distance to reflect back, if they indeed are reflecting. Therefore, their effect would be seen earlier in the time-accurate results. Based on Figure 4.14, it is observed that the unstable oscillations begin sooner using the smaller domain, and so it is concluded that the outflow boundary conditions are a primary source for the instability.

The next chapter summarizes the results of the present study and provides suggestions for future work.

Chapter 5

Conclusions and Future Work

Previous developments using the Cartesian immersed boundary approach have been extended to the survivability field, where an opportunity exists to further understanding on the effect that damage has on the aerodynamic characteristics of flight surfaces. The results presented here are a preliminary step in the development of a way to predict, in real time, an aircraft's performance upon sustaining damage. This study considered a NACA 0009 airfoil with a semi-circular portion of its upper surface removed at the quarter chord, with a radius amounting to 5% of its chord. Several simulations were conducted at various angles of attack to compare the lift, drag, and moment coefficients of the airfoil in the undamaged and damaged states. These were time-accurate, compressible simulations conducted using parallel processing.

Based on the results, it was observed that the semi-circular damage on the upper surface did not have a significant impact on the airfoil's characteristics. The only effect that the results show is a small reduction of lift at 0° angle of attack, although this reduction was not seen at the 4° and 9° angles of attack. For 0° and 4° angles of attack, CARIBOU computed higher drag than the experimental measurements. It is likely that the lack of effect of the damage in the 4° and 9° cases was due to the minimum pressure being forward of the quarter chord location in the 4° case and the flow being separated over most of the upper surface in the 9° case.

Each simulation consistently produced unbounded oscillations in the results at large times, indicating the presence of an instability in the code. It is likely that this instability is linked to the outflow boundary conditions, where compressible disturbances are being reflected rather

than radiated from the domain. Such reflections contaminate the flow solutions, thereby preventing any solid conclusions being possible regarding the effect of damage.

Future work should continue the development of this application, by first investigating the source of the instability and its connection to boundary conditions. There are several other options programmed in CARIBOU for these conditions, which should be exercised. This will allow confidence to be developed with CARIBOU when consistent matching is obtained between numerical and experimental results. Then various damage sizes, shapes, and locations can be explored.

Though the eventual results presented in this thesis were unsatisfactory due to the numerical instability, the use of an immersed boundary method to examine the effects of different damage to an airfoil, or eventually to a wing and aircraft, offers considerable promise

References

[1] http://www.globalsecurity.org/military/systems/aircraft/a-10.htm, November 2008, GlobalSecurity.org, © 2000-2009,

[2] http://www.f-16.net/varia_article8.html, November 2008, Copyright © 2008 Lieven Dewitte and Stefaan Vanhastel.

[3] http://duoquartuncia.blogspot.com/2007/05/landing-f15-with-only-one-wing.html, November 2008.

[4] Sadeghi, Yang, and Liu., "Parallel Computation of Wing Flutter with a Coupled Navier-Stokes/CSD Method," AIAA Paper 2003-1347, 2003.

[5] Cho, Yong; Boluriaan, Said; and Morris, Philip J., "Immersed Boundary Method for Viscous Flow Around Moving Bodies," AIAA Paper 2006-1089, 2006.

[6] Marshall, D. and Ruffin, S.M., "A new inviscid wall boundary condition treatment for embedded boundary cartesian grid schemes," AIAA Paper 2004-0581, 2004.

[7] Peskin, C. S., "Flow patterns around heart valves: a numerical method," *Journal of Computational Physics*, Vol. 10, 1972, pp. 252-271.

[8] Moud-Yusof, J., "Combined immersed-boundary/ B-spine methods for simulations of flow in complex geometries," *CTR Annual Research Briefs*, NASA Ames Research/Stanford University, 1997, pp. 317-327.

[9] Chopra, Jogesh, "An immersed boundary method for the prediction of the aeroelastic behavior of oscillating airfoils," Master of Science Thesis, The Pennsylvania State University, 2007.

[10] Brinkman, H. C., "On the permeability of media consisting of closely packed porous particles," *Applied Scientific Research*, Vol. A1, 1947, pp. 81-86

[11] Wilcox, D. C., *Basic Fluid Mechanics*, DCW Industries, Inc., La Canada, CA, 2003, pg 662.

[12] Menter, F. R., "Improved two-equation k-ω turbulence models for aerodynamic flows," NASA TM-103975, 1992.

[13] Shu, C. W., "Essentially non-oscillatory and weighted essentially non oscillatory schemes for hyperbolic conservation laws," ICASE Report No. 97-65, NASA/CR-97-206253, 1997.

[14] Lockard, D. P., *Simulations of the Loading and Radiated Sound of Airfoils and Wings in Unsteady Flow Using Computational Aeroacoustics and Parallel Computers*, Ph.D. thesis, Penn State University, 1997.

[15] Selig, M.S., Donovan, J. F., and Fraser, D. B., *Airfoils at Low Speeds in Soar Tech Vol. 8*, Herk Stokely, 1504 Horseshoe Circle, Virginia Beach, VA 23451, 1989.