2011-07-14

Two-Dimensional Hydrodynamics of Swimming Rainbow Trout Using Navier-Stokes and Large Eddy Simulation Models

Donovan R. Chipman
Brigham Young University - Provo

Two-Dimensional Hydrodynamics of Swimming Rainbow Trout Using Navier-Stokes and Large Eddy Simulation Models

Donovan R. Chipman

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Rollin H. Hotchkiss, Chair
Daniel Maynes
Scott L. Thomson

Department of Civil and Environmental Engineering

Brigham Young University

August 2011

ABSTRACT

Two-Dimensional Hydrodynamics of Swimming Rainbow Trout Using
Navier-Stokes and Large Eddy Simulation Models

Donovan R. Chipman
Department of Civil and Environmental Engineering, BYU
Master of Science

Energy efficiency and propulsive characteristics of a 10 cm undulatory rainbow trout (*oncorhynchus mykiss*) swimming in a stationary position are considered. Two CFD simulations are performed utilizing dynamic grid meshing (FLUENT 6.3). The first simulation uses a laminar flow model with an added hydrofoil shape in order to test if thrust and drag can be brought to unity. The second simulation uses a Large-Eddy Simulation (LES) turbulence model to determine if transition to turbulence along the fish's surface leads to boundary layer separation. The expected results caused by adding these two features to earlier simulations do not occur. Thrust and drag are not found to be equal with usage of the thicker fish shape; instead both thrust and drag increase by 40-80% while diverging in value. Evidence of boundary layer separation is not present with usage of the LES turbulence model. Swimming energy efficiency is calculated to be 70% in both simulations. A brief analyses of boundary layer and downstream wake are included, showing general agreement with earlier studies. Limitations of the simulation are discussed. Future work regarding the author's preparation for an additional simulation of a rainbow trout utilizing a swimming method known as the Karman Gait is also considered. This preparation includes the creation of a 2-D grid domain and programs to define the kinematics of the fish and produce a specified vortex inlet condition.

Keywords: hydrodynamics, fish, rainbow trout, turbulence, power efficiency, thrust, drag, marine propulsion.

ACKNOWLEDGEMENTS

I wish to express thanks to my graduate advisor, Dr. Rollin H. Hotchkiss, my committee members, Dr. Daniel Maynes and Dr. Scott L. Thomson, and to Dr. Deryl Snyder, who first introduced me to this thesis project. I also wish to thank my wife and children and my parents for their support during this long process.

TABLE OF CONTENTS

LIST OF FIGURES vii

LIST OF SYMBOLS ix

1 Background and Objectives 1

1.1 Potential Application of Results 3

2 Introduction 5

2.1 Undulatory Propulsion 5

2.2 Flow Environment 5

2.3 Kinematics and Energy Efficiency 6

2.4 Boundary Layer 8

2.5 Robotic Studies 8

2.6 Computational Fluid Dynamics Studies 9

3 Materials and Methods 13

3.1 Governing Equations 13

3.2 Software 15

3.3 Geometry 15

3.4 Domain and Grid 16

3.5 Grid Fitness 17

3.6 Grid Updating Method 19

3.7 Inlet Conditions 22

3.8 Discretization 23

3.9 Thrust, Drag, and Power 23

4 Results 27

4.1 Wake Characteristics 27

4.2 Power, Thrust, and Drag 32

4.3 Boundary Layer Mechanism 35

5 Discussion 45

5.1 Vortex Shedding Mechanism 45

5.2 Energy Expended and Captured 45

5.3 Thrust and Drag 45

5.4 Power 48

5.5 Turbulent Effects 48

6 Conclusions and Future Work 51

6.1 Conclusions 51

6.2 Preparations for Karman Gait Simulation 52

6.3 Future Work 56

REFERENCES 57

Appendix A. UDF for Hydrofoil Motion (Uniform Flow Inlet) 63

Appendix B. UDF for Hydrofoil Motion (Von Karman Vortex Street Inlet) 73

Appendix C. UDF for Centerline Motion (Von Karman Vortex Street Inlet) 83

Appendix D. UDF for Von Karman Vortex Street Inlet 95

Appendix E. CFD Studies 101

Appendix F. Grid Convergence Calculations with Input from Current Study 103

LIST OF FIGURES

Figure 3-1: CFD Grid.. 16

Figure 3-2: Drag Based Grid Convergence. 18

Figure 3-3: NACA-0012 Hydrofoil Shape. 20

Figure 4-1: Vorticity Contours of Laminar Simulation. 28

Figure 4-2: Vorticity Contours of LES Simulation.. 29

Figure 4-3: Velocity Contours of Laminar Simulation.. 30

Figure 4-4: Velocity Contours of LES Simulation. 31

Figure 4-5: Thrust, Drag, and Power Coefficients of Laminar Simulation. 33

Figure 4-6: Thrust, Drag, and Power Coefficients of LES Simulation. 33

Figure 4-7: Hydrodynamic Coefficients for Additional Timestep (Laminar). 34

Figure 4-8: Pressure Contours of Laminar Simulation.. 36

Figure 4-9: Pressure Contours of LES Simulation. 37

Figure 4-10: Boundary Layer Profiles at 0.2 Chord Length. 38

Figure 4-11: Boundary Layer Profiles at 0.6 Chord Length. 39

Figure 4-12: Boundary Layer Profiles at 0.9 Chord Length 40

Figure 4-13: Time-Average Boundary Layer Profiles of Laminar Simulation................................ 41

Figure 4-14: Time-Average Boundary Layer Profiles of LES Simulation..................................... 41

Figure 4-15: Example Y+ Value Chart from LES Simulation... 42

Figure 5-1: Flat Plate Paramaters at 1 Degree Angle of Attack, Re = 45000…………………... 46

Figure 5-2: NACA-0012 Paramters at 4 Degree Angle of Attack, Re=45000. 46

Figure 6-1: Example CFD Grid for Karman Gait Simulation... 54

Figure 6-2: Vorticity Contours of Vortex Inlet Simulation. 55

Figure 6-3: Screenshot of Karman Gait Simulation. 55

LIST OF SYMBOLS

A	amplitude of tail beat
C_D	drag coefficient
C_P	power coefficient
C_T	thrust coefficient
d	local fish thickness
F	force, body force acting on fluid
f	tail beat frequency
Fr	Froude number
h	centerline vertical displacement
i	subscript of property in x-direction
j	subscript of property in y-direction
L	total fish length
Lrseg	length of regularized centerline segment
Lseg	length of centerline segment
ndiv	number of centerline segment divisions in modified region
P	pressure
R	radius of vortex core
r	linear displacement along radial line from vortex core
Re	Reynolds number
s	location along length of fish centerline
St	Strouhal number
T	tail beat time period

t	time
tmax	maximum fish thickness
U	free stream velocity in the x direction
V	total velocity vector
V_c	rotational velocity of vortex core
V_θ	rotational velocity at r
W	energy consumed doing work
y+	dimensionless wall value
λ	tail beat wavelength
μ	dynamic viscosity
ν	kinematic viscosity
ρ	density
σ	stress tensor
τ	shear stress, sub-grid stress scale
ω	vorticity

1 Background and Objectives

This study is an attempt to more fully understand the hydrodynamics of a swimming rainbow trout (*oncorhynchus mykiss*). This will entail analyzing the results of two original computational fluid dynamics (CFD) simulations, and describing work done in preparation for a third simulation to be carried out at a future time. Included are descriptions of the simulated fish's wall boundary layers, wake, thrust, drag, and energy efficiency. By extension, these results can be applied to any object that, like a rainbow trout, uses undulatory forms of motion for propulsion. The study will compare results of simulations using an expanded set of body and flow features to the results of a previous study carried out by Patrick Flanagan at Washington State University [Flanagan (2004)]. Of particular interest is a method devised by Flanagan to determine the Froude efficiency of a body that uses undulatory motion for propulsion. The Flanagan study, in turn, was a computer simulation of a particular case of an empirical study carried out by James Liao et al. of Harvard University [Liao (2003)].

In the Flanagan study, a rainbow trout was represented by a sinusoidally oscillating centerline. The position of the snout of the fish was defined at a fixed location within the simulated domain. The flow field was defined as water with a uniform velocity inlet of 0.45 m/s in the x-direction and a zero pressure outlet. Because the fish's position did not change, the values for thrust and drag should have been equal, but analysis of Flanagan's simulation revealed a minor deficiency between the thrust and drag of the simulated fish [Flanagan (2004)]. He hypothesized that this deficiency could be narrowed if the simulation used a fish body with the proper thickness in place of the centerline [Flanagan, 2004]. In order to test this hypothesis, this study adds a hydrofoil contour to approximate the shape of a rainbow trout.

It was additionally proposed in that study that boundary layer separation might be taking place near the fish's tail [Flanagan (2004)]. Separated boundary layers result in the creation of vortices along the length of the fish. It has been theorized that the energy efficiency of a body using an undulatory form of motion, such as the carangiform motion of a rainbow trout, might be benefited by the fish's control over the shedding of such vortices [Triantafyllou (1995)]. Laminar models, such as those used in the Flanagan simulation, cannot model transition to turbulence, so it could be necessary to use a turbulence model to better characterize the boundary layer and determine if separation is indeed taking place. This study has included a Large-Eddy Simulation (LES) turbulence model for that purpose.

An additional set of cases from the Liao study tested rainbow trout swimming in a turbulent wake downstream from a vertically-oriented half-cylinder rod [Liao (2003)]. The undulatory fish motion was characterized in the study as was the fish's position relative to vortices in the turbulent wake. The results suggested that in order to allow the fish to maintain its position downstream from the cylinder, the fish's kinematics within the wake needed to correspond to a different set of parameters (e.g. tail beat frequency, period, and amplitude) than it did in earlier experiments in which a uniform flow field was applied [Liao (2003)]. In effect, this different form of motion, referred to as the Karman gait, allowed the trout to "slalom" around the vortices in the wake [Liao, 2004]. It was hypothesized that by using such kinematic motion, the rainbow trout could capture energy from vortices in the environment [Liao (2004)]. This publication includes a description of work done to enable the simulation of such a scenario so that analysis can be performed to test that hypothesis.

1.1 Potential Application of Results

Better understanding of swimming and vortex control mechanisms could make it easier to design long duration underwater robots for use as scientific probes or surveillance drones.

Hydraulics passages and turbines can also potentially be made friendlier to fish by better understanding how to produce flow fields in which they can more easily swim.

2 Introduction

2.1 Undulatory Propulsion

A rainbow trout propels itself by oscillating with a transverse wave that starts at its front end and is passed backward towards the tail, ultimately resulting in high velocity jets of water that form into vortices after being shed into the wake. A vortex ring is shed every half tail-beat cycle [Blondeaux (2005)]. Carangiform swimmers, such as the rainbow trout, enlarge their vortex rings to accelerate and swim at a higher speed [Mueller et al. (2006)].

Lighthill has published an in-depth review of the topic of fluid dynamics for aquatic animals [Lighthill (1975)] and other more recent studies have included summaries of the state of aquatic propulsion research [Flanagan (2004), Lauder (2009)].

Heaving motion studies on rigid hydrofoils are often used to simulate the effects of undulatory motion [Dong et al. (2006), Ducoin et al. (2009), Techet (2008)]. Results of such studies within controlled environments have been valuable in understanding how various parameters such as oscillating frequency and heaving angles affect propulsive efficiency.

2.2 Flow Environment

The total effect of turbulence on undulatory fish motion is considered to be a question worthy of further research [Lauder (2009)]. That being said, turbulence is a 3-D phenomenon and is fundamentally different from its 2-D counterpart, primarily in the fact that 3-D turbulence tends to break into smaller disturbances, while 2-D turbulence tends to coalesce into larger disturbances [Canuto (2000)]. 3-D turbulent effects are particularly important for understanding the role of fins in aquatic locomotion [Tytell (2008), Webb (2010)]. Sensory of turbulence by the fish is done using its lateral-line sensing system, which sensory can be interfered with by the fish's own motion [Windsor (2009)]. A method to numerically simulate lateral-line sensing based on a Lagrangian perspective has been developed with the hope of better being able to design systems to aid fish passage in the northwestern United States [Goodwin et al. (2006)].

3-D simulations also provide a much more accurate picture of the flow environment than 2-D simulations do [Shen & Diplas (2008)].

Environments immediately upstream of swimming animals have been shown to contain significant fluid structures even if the upstream flow is generally quiescent [Peng & Dabiri (2008)].

2.3 Kinematics and Energy Efficiency

In the early twentieth century, a study published by J. Gray [Gray (1936)] found that a swimming dolphin utilizes several times more power than it was thought a dolphin should be able to produce based on the understanding of dolphin anatomy at that time. The supposed power deficiency came to be known as Gray's paradox, and became a driver for research in methods of propulsion used by fish and other marine life.

Metabolic rates provide one method to measure energy consumption during swimming, and there are multiple empirical studies that use this method that can provide a basis for comparison against CFD results [Enders et al. (2003), McNeil (2005), Claireaux et al. (2006), Taguchi (2011)]. In regards to this study, measurements made on salmon in open water tests suggest that forced flow tests in laboratory conditions under-predict power requirements by as much as 75% [Enders (2003)]. More generally, however, respiratory efficiency in animals such as fish and birds has been measured to increase with body mass faster than the body mass increases [McNeil, et al. (2005)], so that simple linear assumptions regarding respiratory efficiency are not accurate.

One study posited that, contrary to previous findings, the effects of turbulence on swimming performance could be negligible [Nikora (2003)]. The data was limited to one size of fish, however, and the authors recommended further studies be done using varying fish lengths. A later study [Lupandin (2005)] found that turbulence did indeed affect swimming performance, specifically concluding that longer fish required greater flow turbulence to reduce the critical flow velocity at which they could maintain their position. It has also recently been shown that rainbow trout consume less oxygen in turbulent conditions and that the energy they save in such flows by utilizing the Karman gait goes beyond that accounted for just by being in a region of reduced velocity [Taguchi (2011)].

It has been theorized that the most efficient oscillating motions are those that have a Strouhal number falling in the range of approximately .25 [Miao & Ho (2006)]. The Strouhal number is defined as:

$$St = \frac{f * A}{u} \qquad (2\text{-}1)$$

where f is the tailbeat frequency, A is the width of the tail beat, and u is the stream-wise velocity.

Besides continuous undulatory swimming, which this study considers, it is also possible to use a method of swimming known as “burst and coast”, which as the name suggests, rests between tail beats. This method of swimming is shown to save 56% of the energy required for continuous swimming [Chung (2009)], and as such, is worthy of further study.

2.4 Boundary Layer

Due to the effects of fluid viscosity and what is termed the no-slip condition (which requires that relative motion between a wall and the fluid at the wall boundary does not occur), a region of reduced flow velocity called a boundary layer forms near to a boundary wall. This boundary layer is responsible for increased drag along the length of a fish because of viscous shear resulting from the differences in the velocity at the wall and in the free stream.   If, due to an adverse pressure gradient, the shear stress along the wall drops to zero, then the boundary layer separates from the wall creating a region of reverse flow and increased vorticity.

It is possible that fish use vorticity control to reduce both their drag and energy requirements [Triantafyllou (2002)]. It is also possible, though, that drag control methods may not be necessary to explain the hydrodynamics of a fish, implying that inviscid models would be sufficient [Sears (1969)]. Using elongated-body theory, Lighthill’s analysis of his data suggested that drag is increased at the time that the fish’s swimming motion begins. This was considered to be the result of boundary layer thinning [Lighthill (1971)].

2.5 Robotic Studies

One of the major motivators for better understanding swimming hydrodynamics is its applicability to robotic vehicles. Various projects have been undertaken to construct robotic fish, some of which use whole body undulation [Triantafyllou (1995), Dogangil et al. (2005), McMasters et al. (2008), Suleman (2008), Low (2009)], while others investigate certain portions of fish motion, such as the fins [Y. Zhang et al. (2008), Hu et al. (2009)]. Results of these projects tend to be positive, showing good energy efficiency and improved maneuverability over current marine vehicles.

2.6 Computational Fluid Dynamics Studies

As opposed to inviscid numerical studies, viscous flow CFD simulations take into account the effects of viscosity and can find flow solutions using the full Navier-Stokes equations. CFD simulations with low Reynolds numbers have reduced computational requirements, as do 2-D simulations compared to 3-D scenarios. An example of an early low Reynolds number simulation was of a collective of swimming cells [Fauci (1996)]. Improvements in the models over time allowed for the consideration of skeletal structure interaction with the fish body to better approximate how real fish are affected by flow disturbances [Jordan (1996)].

A few CFD studies have been performed on relevant 2-D geometries in recent years. Miao, for example, performed a study that compared simulations of flexible airfoils in plunge motion with variations in flexure amplitude, Reynolds number, and reduced frequency. It was found that with Re = 40,000, flexure amplitudes of less than 0.5 produce wakes indicative of thrust production and that enhancements to the propulsive efficiency occur when the flexure amplitude is 0.3 times the chord length [Miao & Ho(2005)].

An optimization of the shape of a 2D swimming body with a prescribed vertical motion that was otherwise free to move horizontally led to the result that a semi-rounded front edge and inwardly tapering side towards the tailing edge, somewhat like that of a very narrow fish, is the most efficient shape. Reasoning as to why such a shape is optimal was concluded to require further study [Thomson (2009)]. A simulation of a 2D flapping dragon fly wing used a new method based on a leading edge suction analogy and assumed partial flow separation to calculate aerodynamic characteristics [Azuma (2005)]. Such a method could be useful in hydrodynamics as well, as it is able to obtain thrust and drag approximations on unsteady foil motions with much less computational effort than that which is required for Navier-Stokes simulations. Flapping dragon fly wings in flight can obtain a Reynolds number as high as Re = 35,600 [May (1991)], which is not much below the Reynolds number of the simulations performed for this study.

Since the Flanagan study was completed, there have also been several CFD simulations done on 3D fish shapes and even of small underwater vehicles. Adkins et al. used a Reynolds Averaged Navier-Stokes (RANS) turbulence model in their simulation of a 3D bio-mimetic fish. While they did not report thrust or other numerical hydrodynamic properties, their visualizations did show qualitatively that the flow can be considered two-dimensional in the longitudinal direction, but in the transverse direction, there are significant three-dimensional flow properties. The mix between the properties in the longitudinal and transverse directions is shown to be dependent upon the phase of oscillation [Adkins et al. (2006)]. Narasimhan et al. used three-dimensional CFD simulations to show that locating fins near to the center of mass improves an unmanned automated underwater vehicle's maneuverability [Narashimhan et al. (2006)].

Three-dimensional simulations of a pufferfish have utilized a combination of blade element analysis and CFD to determine thrust generation [Conroy & Gordon (2008)]. Another method for resolving thrust and power production is the artificial compressibility approach, which was utilized in the 3D simulation of a tuna fish of more than one meter in length with a Reynolds number of 710,000. The predicted thrust with this method is somewhat higher than that predicted with the panel method [Z. Zhang et al. (2008)]. Leroyer and Vissoneau used a method of coupling a RANS solver with Newton's laws, which allowed them to perform 3D simulations of fish performing turning motions with better accuracy than could have been done by just specifying the path of motion [Leroyer & Vissoneau (2005)].

Borzajani and Sotiropoulos. carried out two studies regarding the efficiency of carangiform and anguilliform motion. The studies used multiple 3D simulations with varying Reynolds and Strouhal numbers. It was determined that for carangiform motion, the critical Strouhal number (where constant velocity is achieved and the net longitudinal force becomes zero) increases as a function of the Reynolds number. Drag, however is shown to increase with the usage of undulatory motion when the Reynolds number is low (Re = 300 and Re =4,000). The wake structure is shown to be dependent on the Strouhal number as well [Borzajani & Sotiropoulos (2008)]. Anguilliform motion, on the other hand, was found to be more efficient than carangiform motion for a given Reynolds number [Borzajani and Sotiropoulos (2009)].

Appendix E contains a table that compares certain hydrodynamic characteristics of some of the studies mentioned.

3 Materials and Methods

Methods used to computationally simulate a rainbow trout swimming in a uniform inlet flow field corresponding to a specific case from Liao's experiments [Liao (2003)] and the methods used to analyze its hydrodynamics are described within this chapter. Two such simulations using different computational models were performed for this study.

3.1 Governing Equations

The first of these simulations used the full Navier-Stokes Equation with no consequential modeling of subscale turbulence or use of Reynolds averaging as follows:

$$\rho \frac{DV}{Dt} = F - \nabla P + \mu \nabla^2 V \tag{3-1}$$

where ρ is the fluid density, D/Dt is the total time respective derivative, V is the velocity vector, F is the body force acting on the fluid, P is the pressure, and μ is the fluid's dynamic viscosity.

The working fluid is liquid water at 20 degrees C. Because water is incompressible, it is subject to the condition of the continuity equation:

$$\nabla \bullet V = 0 \tag{3-2}$$

The dot product of the del operator and V equaling zero ensures that the volume of fluid within the simulation’s domain does not change, so that mass (for an incompressible fluid) is conserved.

In order to better understand conditions leading to possible flow separation, it is necessary to include a turbulence model. The second simulation will use such a model called Large-Eddy Simulation (LES). LES is a hybrid fluid model in that it uses filtered Navier-Stokes equations to calculate flow properties on a large scale, while utilizing simplified turbulence models to approximate flow on smaller scales. Such a model can be considered valid because vortex properties do not vary widely as vortex scales diminish.

The governing equations for Large-Eddy simulations used in the FLUENT software package are as follows [Fluent 6.3 User’s Guide (2006)]:

$$\frac{\partial \rho}{\partial t}+\frac{\partial}{\partial x_i}\left(\rho \bar{u}_i\right)=0 \tag{3-3}$$

$$\frac{\partial}{\partial t}\left(\rho \overline{u_i}\right)+\frac{\partial}{\partial x_j}\left(\rho \overline{u_i u_j}\right)=\frac{\partial}{\partial x_j}\left(\mu \frac{\partial \sigma_{ij}}{\partial x_j}\right)-\frac{\partial \overline{p}}{\partial x_i}-\frac{\partial \tau_{ij}}{\partial x_j} \tag{3-4}$$

$$\sigma_{ij} \equiv \left[\mu\left(\frac{\partial \overline{u_i}}{\partial x_j}+\frac{\partial \overline{u_j}}{\partial x_i}\right)\right] \tag{3-5}$$

where σ_{ij} is the stress tensor created as the result of molecular viscosity.

$$\tau_{ij} \equiv \rho \overline{u_i u_j}-\rho \overline{u_i}\,\overline{u_j} \tag{3-6}$$

where τ_{ij} is the sub-grid stress scale.

The overbars in equations (3-3) through (3-6) mean that the variables are averaged.

Sub-grid vortex modeling for the LES scenario was done using the Smagorinsky-Lily model.

3.2 Software

The two major software packages used in this analysis are the CFD solver FLUENT and the pre-processor GAMBIT.

FLUENT 6.3 is a product of Fluent, Inc., which is owned by ANSYS. Included within the package are solvers for such diverse conditions as laminar and turbulent flows, compressible and incompressible fluids, heat transfer, and multiphase flows. A notable feature that makes FLUENT suitable for this research project is the ability to adjust the mesh shape dynamically.

GAMBIT is a pre-processor used for creating discrete meshes and defining geometries. This program comes packaged with FLUENT.

3.3 Geometry

In Flanagan’s uniform inlet flow scenario (UIFS), the trout was modeled as a 10 cm undulating thin line with the upper and lower surfaces separated by a small one millimeter offset. A significant update in the present study is the addition of a hydrofoil shape (a NACA 0012) molded around the centerline. Although the contours of a four-digit NACA airfoil and that of a rainbow trout are not the same, the maximum length to width ratio of approximately 8.3 is maintained, as is the overall length of 10 cm. The hydrofoil is included to determine the effects that thickness has on the fish’s thrust and drag, and hence on its energy efficiency.

3.4 Domain and Grid

Grids for the initial time-step are created in GAMBIT using a journal file. The two-dimensional domain used is 30 cm high by 60 cm long, creating a four fish- length long region downstream of the fish to capture the wake. The upper and lower edges of the grid use a symmetry boundary condition. The inlet is a velocity inlet and will be described in further detail in a later section. Free stream pressure is also defined as being zero psi.

The grid for the UIFS case has approximately 530,000 cells, and was defined so that grid density was highest in the regions near to the moving fish wall and downstream of the fish in order to capture details of the boundary layer and the wake (figure 3-1).

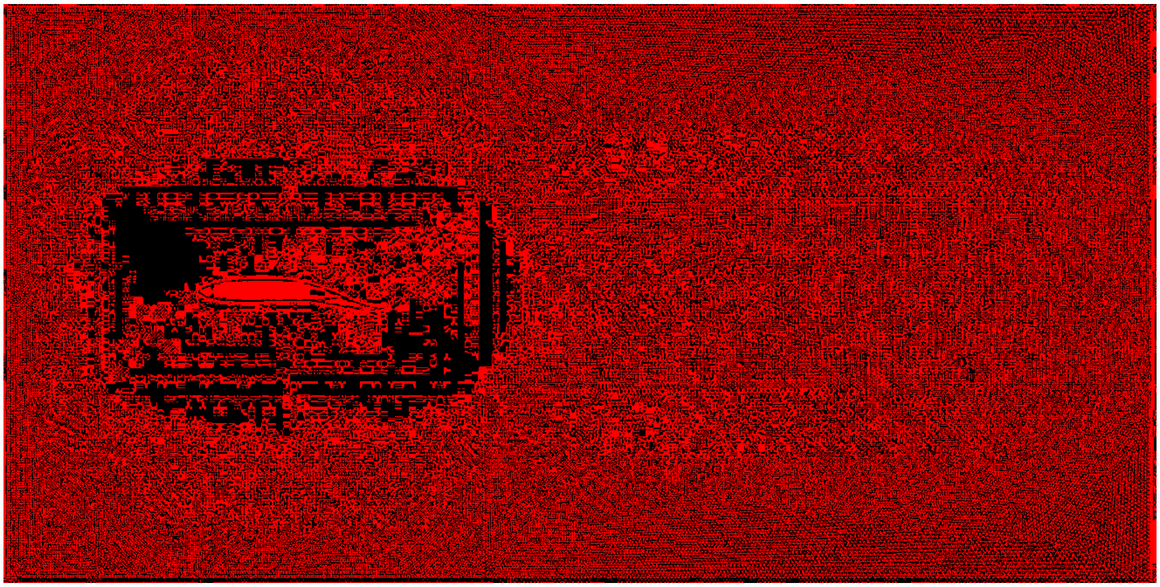


Figure 3-1: **Grid and initial fish shape used for both laminar and LES flow simulations.**

For turbulent simulations, the dimensionless wall value y+ is used to help determine if the grid near the wall is sufficiently fine.

The dimensionless wall value is defined as:

$$y^{+} = \frac{u * y}{\nu} \qquad (3\text{-}7)$$

where y is the distance away from the wall and ν is the kinematic viscosity (1 x 10^{-6} m^2/s for water). The asterisk is used as a multiplication sign in this study.

For LES models, it is recommended that y+ values should be close to 1 in order to resolve the laminar sub-layer and thus obtain the best results [Fluent 6.3 User's Guide].

3.5 Grid Fitness

Verification of basic grid fitness (i.e. no negative cell volume, no excessive skewness of cells, etc.) is done by the FLUENT software at the prompt of the user prior to initiation of a simulation.

Flanagan carried out calculations to determine grid uncertainty based on peak vorticity in grids of varying coarseness using methods outlined by Celik & Karatekin [Celik (1997)]. His results showed that for the laminar case, a 200-node fish centerline produces results accurate to within 0.6%. In order to use the same grid convergence method, three laminar flow simulations were performed with grids scaled to a fish with upper and lower surfaces each divided into 123 nodes, 163 nodes, and 203 nodes [see Appendix F and also Flanagan (2004) Appendix C for original source]. The data used is from time step 4,500, which equates to a simulation time of 0.9 seconds. Based on the input of vorticity of the second vortex and location of the eighth vortex downstream from the fish, the laminar simulation here considered showed the simulation

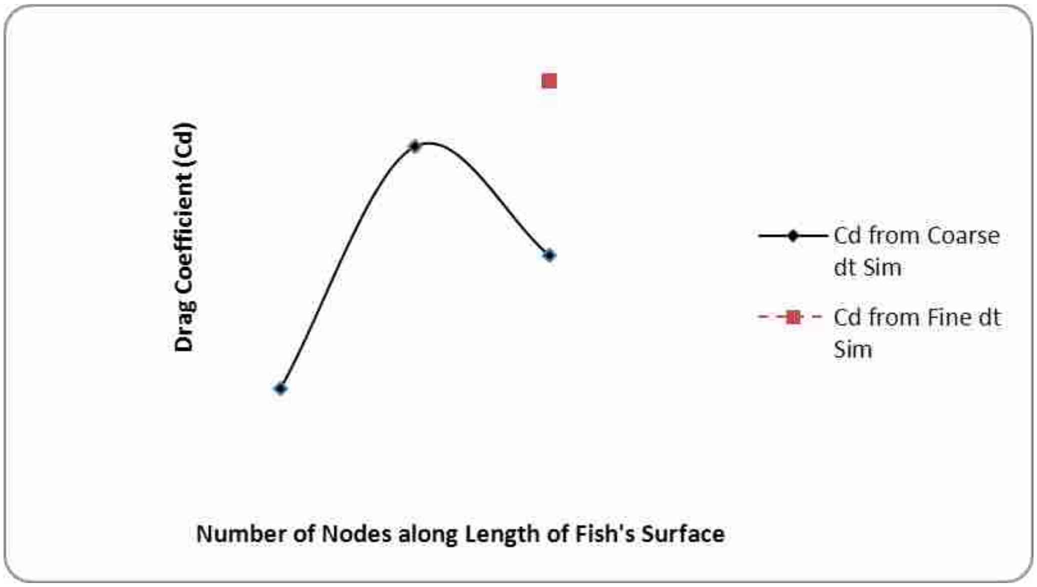


Figure 3-2: Grid convergence test for laminar flow simulation using drag coefficients.

should be accurate to within 1.4%. However, convergence did not occur based on the vorticity of the third downstream vortex, suggesting possible problems with grid independence.

Another method of determining grid convergence based on the drag coefficient was also utilized (figure 3-2). The average drag coefficient for the finest 203 node surface is 0.0808. For good convergence, given the pattern of increase in the drag coefficient of the other two coarser grid simulations, this value should be slightly higher than that of the 163 node surface; instead it is between the average values found for the 123 node surface (where $C_D = 0.0791$) and the 163 node surface (where $C_D = 0.0822$). Still, the overall change in drag coefficient between the 163 node case and the 203 node was less than 2%. Additionally, analysis of the drag coefficient from a subsequent 203 node surface simulation in which data was saved at every time step (see section 4.2 for details), produces a drag coefficient of approximately 0.083, which follows the expected pattern for convergence. This suggests that the usage of data from every time-step could show better convergence, but further simulations using data from every time step of the coarser 123 and 163 node grids would need to be performed to verify this hypothesis.

3.6 Grid Updating Method

The centerline of the fish was modeled by Flanagan using the following equation:

$$h(x,t) = a(x)\sin[2\pi\left(\frac{x}{\lambda} - \frac{t}{T}\right)] \qquad (3\text{-}8)$$

where a(x) is the lateral wave amplitude.

$$a(x) = L\left[.351\sin\left(\frac{x}{L} - 1.796\right) + .359\right] \qquad (3\text{-}9)$$

In the Flanagan simulation, both the upper and lower surfaces were divided into 200 segments, each .05 cm long. Node location updates at each time step utilized a looping method built into FLUENT's User-Defined Function (UDF) system called DEFINE_GRID_MOTION, which requires point locations to be specified by the user. The simulations used a method known as local remeshing [FLUENT 6.3 User's Guide (2006)] to update the grid at the start of each time step. If a cell has a skewness value greater than 0.7, or if its size is outside of the limits specified, it is marked for remeshing.

Flanagan's UDF utilized the Newton-Raphson method (because of its simplicity in programming and efficiency in reaching convergence) in order to verify that the length of each wall segment remained equal throughout the simulation.

For the current study, a width based on the NACA-0012 foil is added to each centerline point by the UDF. This is done in order to test the hypothesis that adding thickness to the simulated fish will help to bring the thrust to drag ratio closer to unity than that calculated in the Flanagan study.

The equation describing the outer line of a four- digit NACA foil is:

$$d(s) = \frac{t\max}{.2} L\left(.2969\sqrt{\frac{s}{L}} - .126\left(\frac{s}{L}\right) - .3516\left(\frac{s}{L}\right)^2 + .2843\left(\frac{s}{L}\right)^3 - .1015\left(\frac{s}{L}\right)^4 \right) \quad (3\text{-}10)$$

where d(s) is the distance of the foil's outer-line normal to and away from the centerline at position s, tmax is the maximum foil thickness, and L is the maximum chord length [Jacobs (1931)].

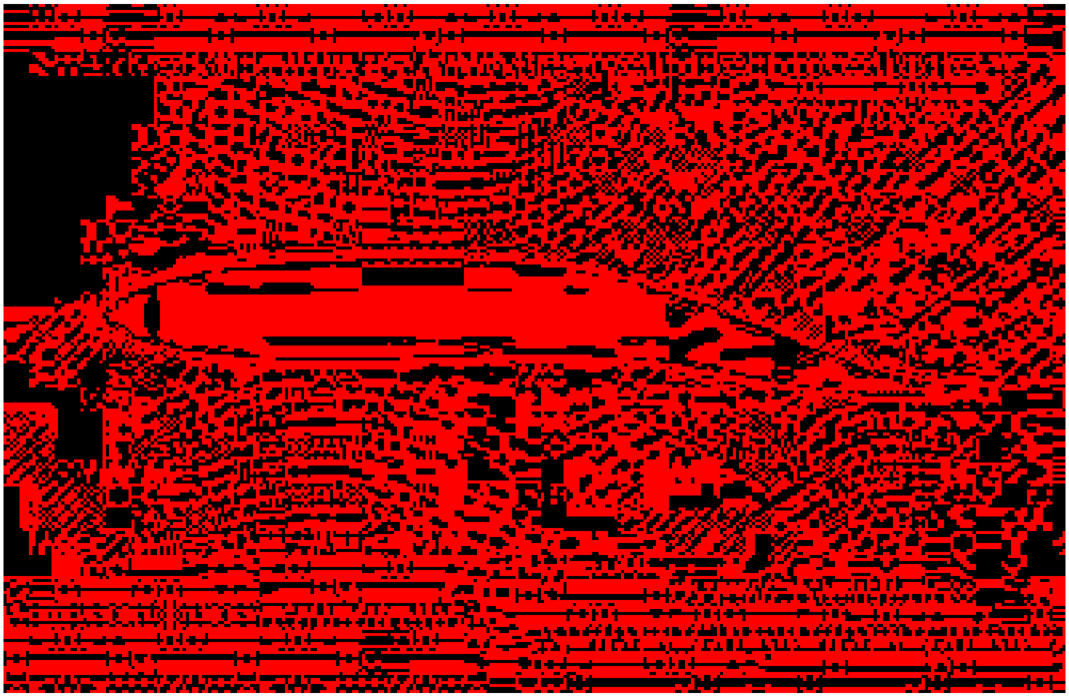


Figure 3-3: A comparison of the shape of a rainbow trout used by Liao [see Liao (2003)] to that of the hydrofoil used for this study (above) reveal differences in the leading edge radius and locations of maximum and minimum foil thicknesses.

Due to the addition of the foil length normal to the equally spaced centerline points (with four points at the start being exceptions), and the differences in slope along that centerline, the distance between grid points on the upper and lower surfaces differ somewhat. Along most of the foil, these length differences are quite minor and do not seriously impact the quality of the grid, but they are very significant at the front end where the foil is more perpendicular to the centerline than parallel to it. As such, it was necessary to divide the first centerline segment into four smaller sub-segments of geometrically increasing length according to the following formula:

$$Lseg(n) = \frac{n}{\sum_{i=1}^{ndiv} i} * L_{rseg} \qquad (3\text{-}11)$$

where L(n) is the length of centerline sub-segment n, ndiv is the number of sub-segments, and L_{rseg} is the length of a regular segment, in this case .05 cm.

The NACA foil shape used to define the fish in these simulations do not precisely match the actual shape of a rainbow trout. The ratio of the maximum thickness to length of approximately 8.3 was maintained, however.

A comparison of a fish shape taken from the Liao study and the one used for this simulation is here considered. Several differences are apparent just from a visual examination (See Figure 3-3 and Liao (2003)). For example, a real rainbow trout has a much larger radius of curvature at the nose than a NACA 0012 has at its front end. The maximum transverse thickness of a trout occurs less than half as far from its nose (about .15 body lengths) as it does for a standard NACA foil (which occurs at about 0.3 chord lengths from the front end). The fish has a minimum thickness that occurs upstream of the tail end, whereas NACA foils have a minimum thickness at the tail end. As a result of these differences, it is to be expected that the thrust and drag in the simulation will not precisely match that of a real fish.

This project utilizes the Newton-Raphson method to update node locations in the uniform inlet flow scenarios (UIFS).

The C program "segmentedfoilmoveudfud.c", which serve as the grid update UDF for the UIFS simulations, can be found in Appendix A. Based on the inputs to the UDF, the Strouhal number of the fish is 0.32.

3.7 Inlet Conditions

Inlet flow for the UIFS flow scenarios is set at a constant 0.45 m/s normal to the inlet location. The Reynolds number (Re) is a dimensionless value used to determine the importance of the viscosity of a working fluid is in relation to the inertia of a body moving within that fluid. It is defined as:

$$\mathrm{Re} = \frac{\rho * u * L}{\mu} \tag{3-12}$$

where ρ is the density of the working fluid (for water at STP, this is 1000 kg /m^3), U is the velocity of the working fluid relative to the considered body (0.45 m/s as defined above), L is a characteristic length (which in this case is the chord length of the fish, or 0.1 m), and μ is the dynamic viscosity (for water at STP, this is 1.002 x 10^{-3} N*s/m^2).

The Reynolds number of the simulations here considered is 45,000.

3.8 Discretization

The UIFS simulations covers a duration of 1.3 seconds, allowing enough time for water initially in the flow domain to be removed. Each time-step covers 0.0002 seconds, and over this duration, data was saved every fifty time-steps, creating 0.01 seconds of simulation time between data sets.

Time discretization is first-order implicit. Pressure discretization is second-order. Momentum discretization is second-order upwind. The pressure-velocity coupling is done using FLUENT's "simple" algorithm, which works well for the small time steps used in these simulations. The momentum under-relaxation factor is set to 0.7, while the pressure under-relaxation is set to 0.3 to allow scaled continuity residuals to go below 1.25 x 10^{-4} for the laminar simulation, and below 2 x 10^{-5} for the LES simulation.

3.9 Thrust, Drag, and Power

Thrust is the force that propels a body in the direction of its motion. So long as the boundary layer remains attached, thrust is solely the result of pressure differences acting normal to the surface of a body. For the purposes of this study, thrust can be mathematically defined as the component of the surface integral of the pressure over a differential unit of area acting in the direction of the body's motion.

The drag force is the force acting on a body in the direction opposite its motion. The drag force is composed of both friction drag and form drag, which are added together to obtain the total drag.

In this study, thrust and drag should be equal because the fish remains stationary.

For laminar cases, the hydraulic friction drag acting against a body can be defined in terms of the contacting fluid’s viscosity and the rate of shear strain as follows:

$$\tau = \mu \frac{\partial u}{\partial y} \tag{3-13}$$

Like thrust, the form drag is the result of pressure differences acting normal to the surface of a body and is defined as the component of the surface integral of the pressure over a differential unit of area acting in the opposite direction of the body’s motion.

Modified versions of Fortran90 programs written by Flanagan are used to determine the thrust and drag forces and power requirements. The major update required for these program is the replacement of an analytical solution of the slope on each cell face at each time-step with a central difference numerical approximation based on x-y location data output.

Property coefficients can be used to describe thrust and drag in a dimensionless form. The thrust and drag coefficients are defined as follows:

$$C_T = \frac{2 * Thrust}{\rho * u^2 * L} \tag{3-14}$$

$$C_D = \frac{2 * Drag}{\rho * u^2 * L} \tag{3-15}$$

The average power output is defined as the quotient of the integral of the work output and the period of time over which that work was done. For numerical studies such as the one considered here, this integral can be determined by summing the product of the force applied on the fish in a given direction by the displacement in that direction at each time step . Mathematically, these two processes can be represented as follows:

$$Power = \frac{1}{T}\int_0^T \frac{W}{\Delta t}\,dt = \frac{1}{T}\sum_{t=o}^{T}\sum_{surface}\left(F_x \Delta h_x + F_y \Delta h_y\right) \tag{3-16}$$

Power can also be defined in terms of a dimensionless power coefficient, which is found according to the following formula:

$$C_P = \frac{2*Power}{\rho * u^3 * L} \tag{3-17}$$

The Froude number (Fr) is a measure of power efficiency. It is found by dividing the product of thrust and velocity (the resultant power) by the total power output, or:

$$Fr = \frac{Thrust*u}{P} \tag{3-18}$$

Because power output changes with time, the average Froude number is here calculated for usage in comparison. It should be noted that equation 3-18 is only truly valid for this type of simulation when thrust equals drag, otherwise the fish will move up or down stream and the value for u will be different.

4 Results

This chapter will discuss the results of the simulations described in chapter 3. Particular consideration will be given to the characteristics of the wake downstream of the fish, the boundary layer along the length of the fish, and the resultant thrust, drag, and energy output.

4.1 Wake Characteristics

At the beginning of the fish's downstream wake, the fish's undulatory motion produces a jet of water that provides thrust. The portion of the flow field located downstream from the rainbow trout is composed of alternating shed vortices. The direction of vortex flow is opposite that which occurs in a wake created by a stationary object such as the rod used in Liao's reference study or a rock in a river bed.

Vorticity is used to measure the rotation of a flow-field. Mathematically speaking, vorticity is defined as the curl of the velocity vector, which for a two-dimensional flow field simplifies down to:

$$\omega_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \qquad (4\text{-}1)$$

The center of a vortex is determined by the location of maximum vorticity. The laminar model showed that peak vorticity in the wake is around 130 s^{-1} (figure 4-1). In the LES model, however, peak vorticity was almost 200 s^{-1} (figure 4-2), more than 50% higher than that of the

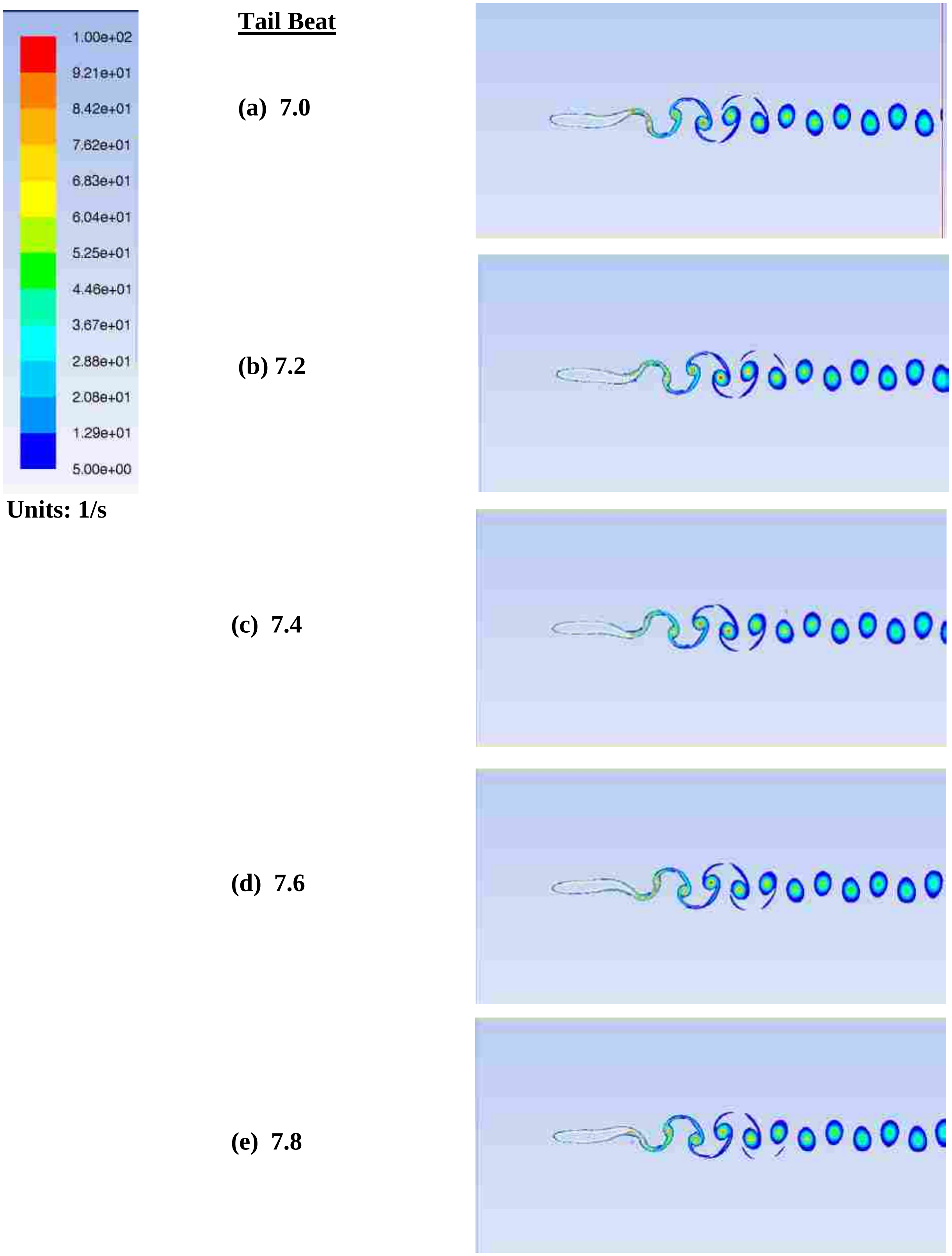


Figure 4-1: Absolute vorticity contour plot of the laminar simulation's flow domain at selected time steps over the course of one tail beat period.

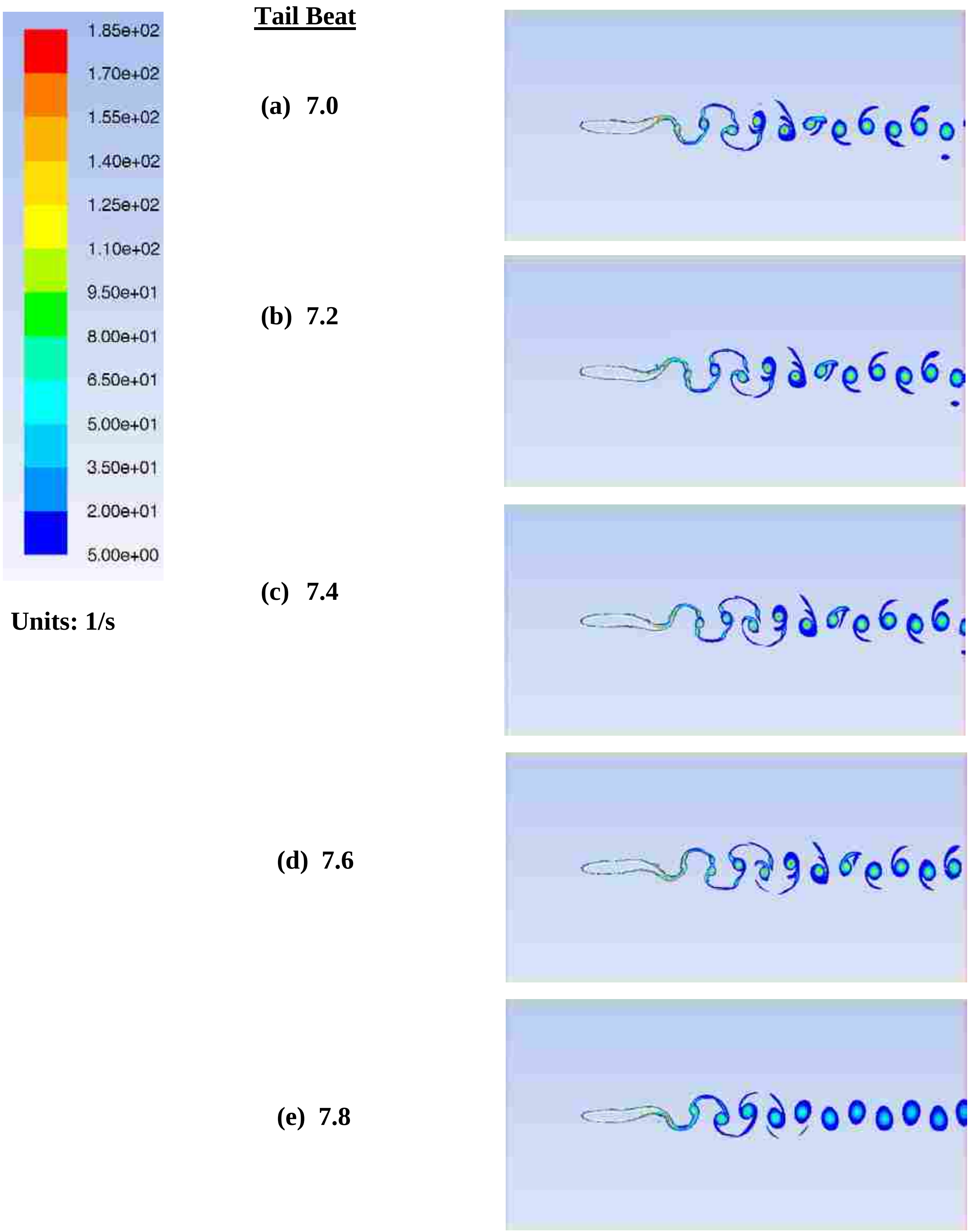


Figure 4-2: **Absolute vorticity contour plot of the LES simulation's flow domain at selected time steps over the course of one tail beat period.**

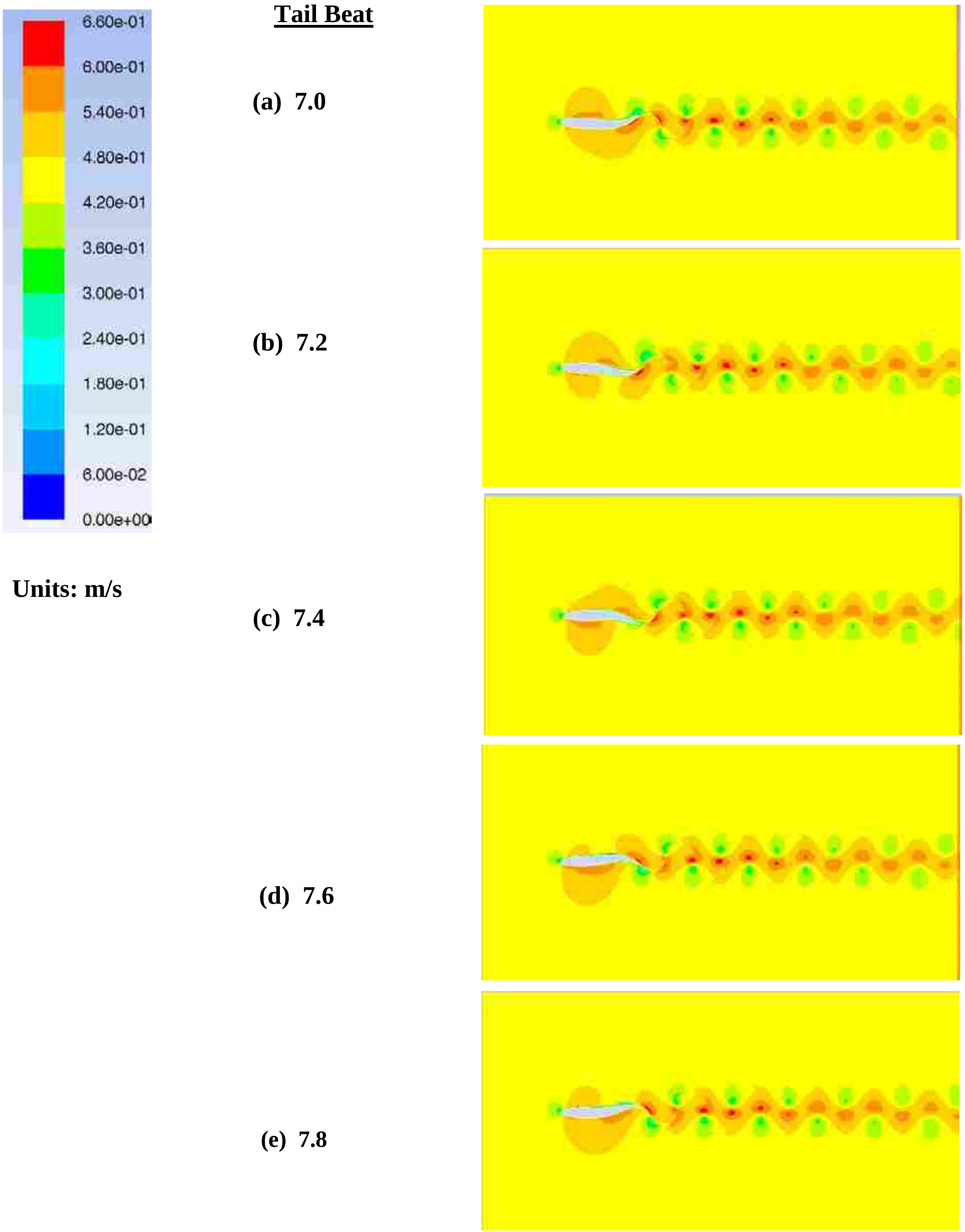


Figure 4-3: Velocity magnitude contours for the laminar flow simulation at selected time steps over the course of one tail beat.

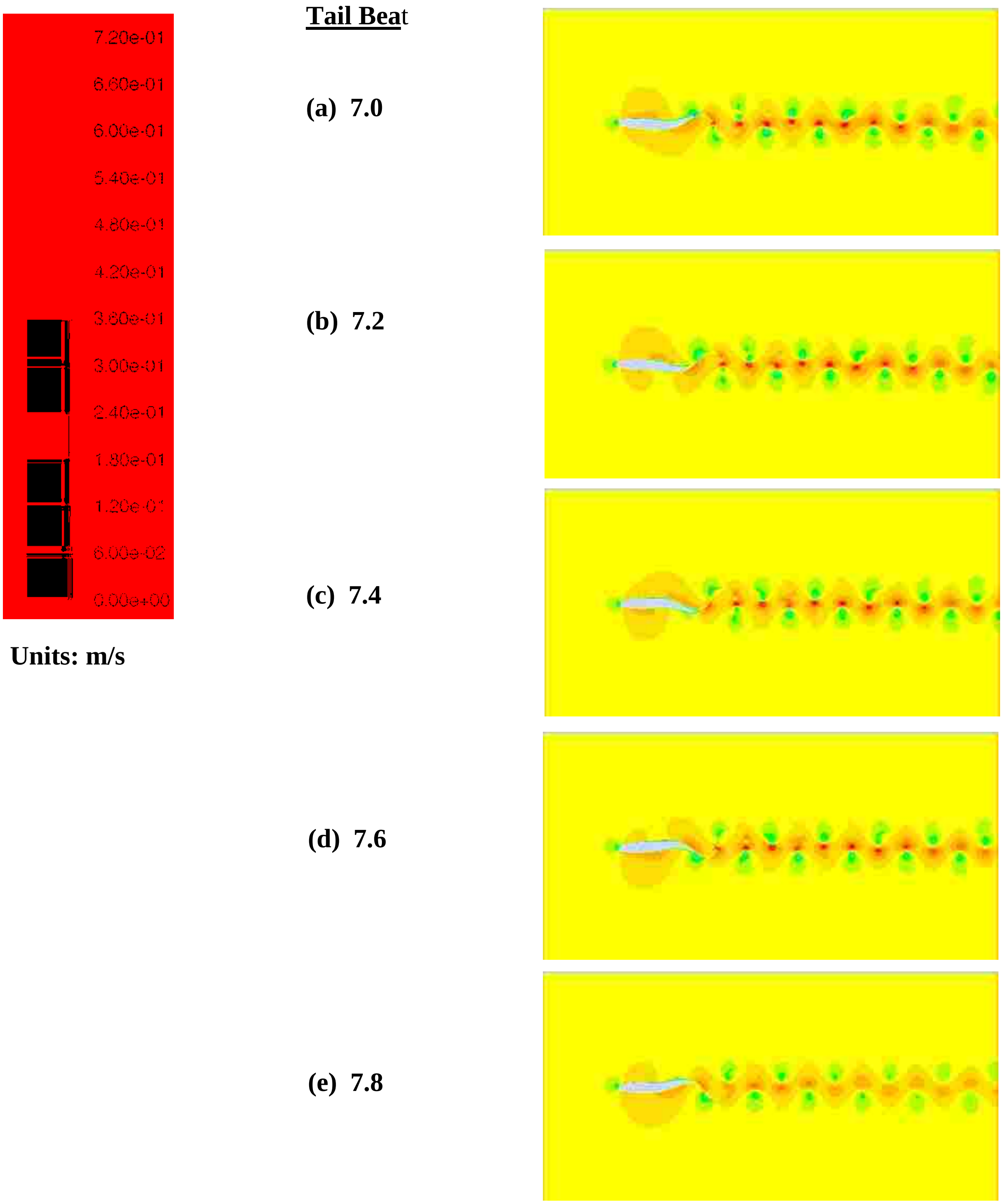


Figure 4-4: Velocity magnitude contours for the LES simulation at selected time steps over the course of one tail beat.

laminar simulations. The increase in vorticity found by the LES model is also indicated by the larger magnitude of maximum flow velocity of 0.72 m/s (figure 4-4), as compared to 0.66 m/s for the laminar simulation (figure 4-3), and also by the lower minimum pressure located within the shed vortices.

4.2 Power, Thrust, and Drag

Period-averaged thrust, drag, and power were calculated over a three tail beat time period for each simulation (see figures 4-5 and 4-6 for laminar and LES results, respectively). For the laminar and LES simulations thrust was calculated to be 1.06 N/m and 1.13N/m respectively. The difference between these two values is a little more than 7.2%. Period-averaged drag was calculated at .813 N/m and .868 N/m respectively for the laminar and LES flow simulations. The difference between these two values is a little less than 6.8%. In both cases, if the simulation were accurate, the thrust and drag would be equal to each other.

Friction drag is included in the total drag. For the laminar flow simulation, the average friction drag is 0.314 N/m, or 38.6% of the total drag. The LES simulation has an average friction drag of .264 N/m, or 30.4% of the total drag.

Average power output for the laminar flow simulation is 0.296 W/m. For the LES simulation, the average power output is 0.307 W/m. The difference in power output between the two simulations is 3.7%. These power averages are based on the sum of positive work done by the fish and negative work done on the fish. If only positive work is considered, then the average power output is 0.6858 W/m for the laminar flow simulation, and 0.7247 W/m for the LES simulation, a difference of 5.7%.

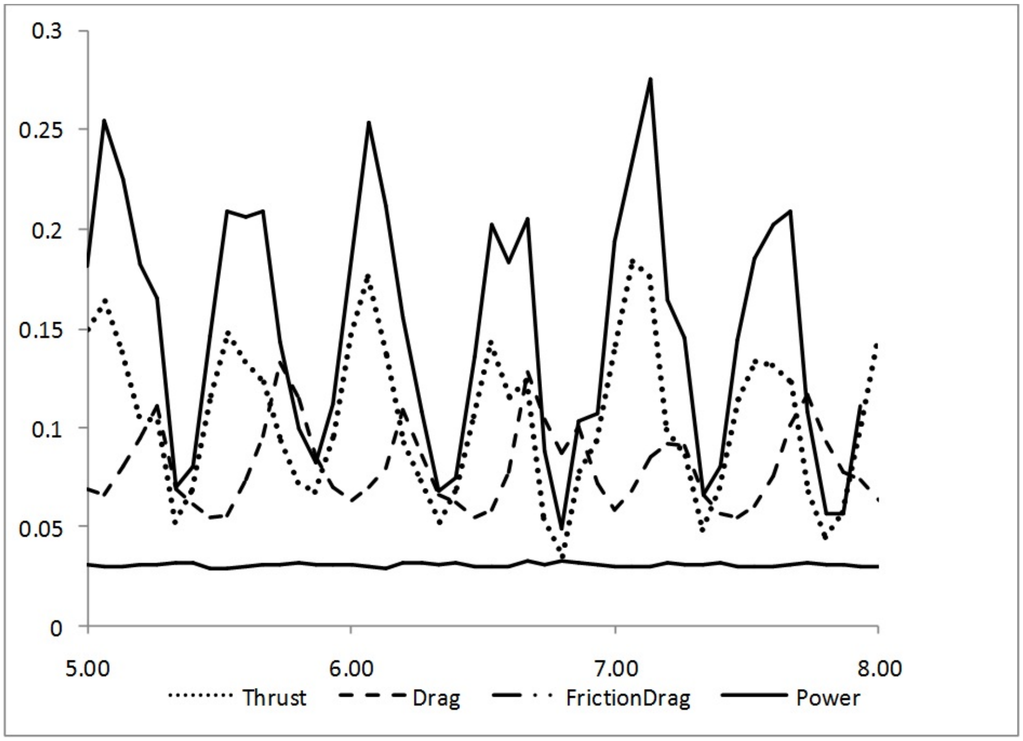


Figure 4-5: **Fluid dynamic property coefficients for the laminar flow simulation.**

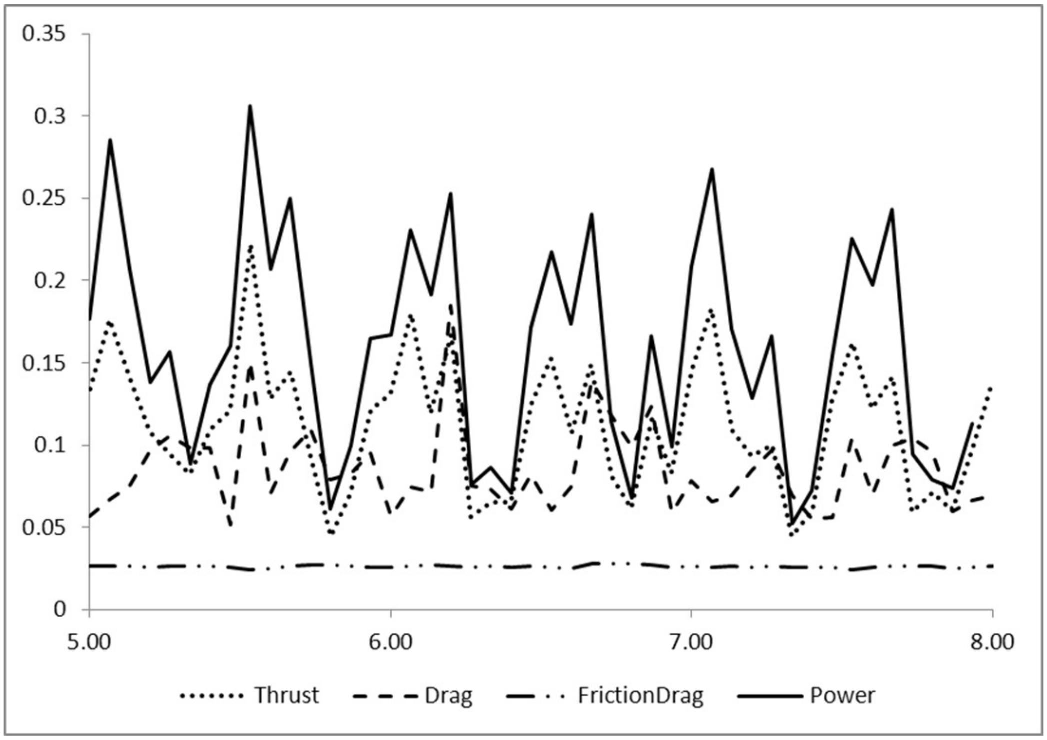


Figure 4-6: Fluid dynamic property coefficients for the LES simulation.

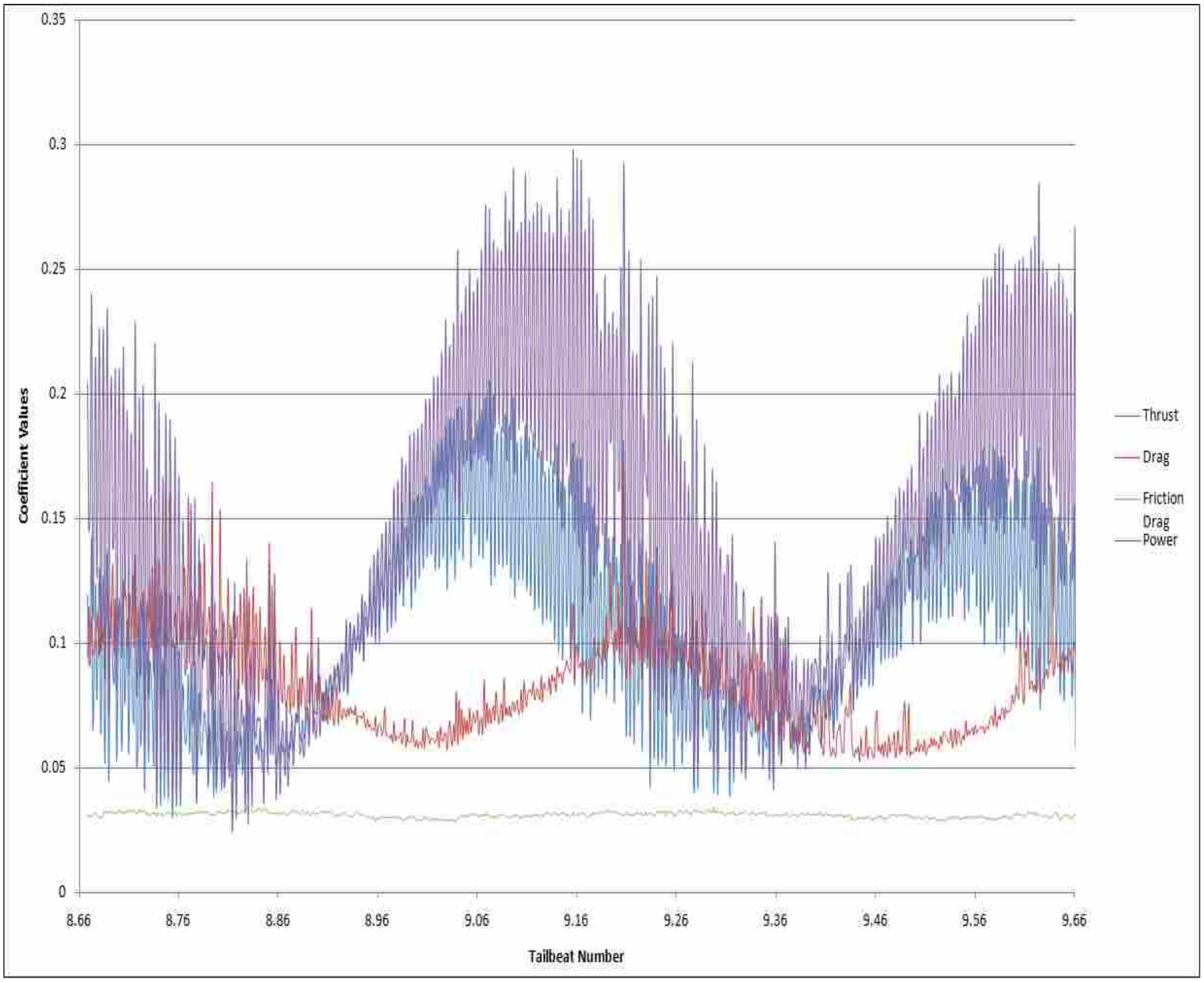


Figure 4-7: Hydrodynamic coefficients for additional tail beat. Shows coefficients for thrust, drag, friction drag, and power for every .0002 second time step.

As previous laminar simulation results have shown [Flanagan (2004)], the coefficient of power is always greater than the coefficient of thrust, peaks in drag occur shortly after peaks in thrust allowing for movement upstream, maximum thrust occurs at times at maximum tail beat amplitude, and maximum drag occurs when undulation at the mid-length of the chord reaches maximum amplitude. The patterns for the LES simulation are similar, and though it has a greater number of peaks in thrust and drag, the periods in which the drag is greater than the thrust lines up well with the results of the laminar simulation.

Both simulations (figures 4-5 and 4-6) exhibit an apparent irregularity in their wave patterns, though, suggesting that there are too many time steps between data sets to adequately allow for the proper calculation of the hydrodynamic parameters of the fish. Starting from the end of the 1.3 second laminar flow simulation described, an additional 0.15 second simulation using a laminar flow model was performed and data was saved at every time step. Analysis of the fish's hydrodynamic parameters was then carried using the same methods already described. Average thrust was found to be 1.0531 N/m, average drag was 0.8396 N/m, and average power (inclusive of negative work done) was 0.4909 W/m. Average power based only on work done by the fish was 0.6487 W/m. Power efficiency based on positive work done by the fish was 73.1%.

Plots of the hydrodynamic properties of this additional tail beat, however, reveal unacceptable levels of oscillation at each time step (figure 4-7). These oscillations show that the results of the entire simulation are non-physical in nature.

Figures 4-8 and 4-9 show contours of pressure for both simulations at various time steps during the seventh tail beat period. Pressure differentials between the left and right sides help contribute to the fish's propulsive thrust when the tail amplitude is greatest. These contours show that negative pressure regions are more intense than positive pressure regions and that, when considered in combination with the thrust and drag charts discussed earlier, maximum thrust occurs when the region of negative pressure is largest.

4.3 Boundary Layer Mechanism

Boundary layer profiles have been plotted for both UIFS cases over the time period of one tail beat at 2 cm (figure 4-10), 6 cm (figure 4-11), and 9 cm (figure 4-12) from the leading edge of the trout. It can be seen from these plots that velocity profiles variations over time increase with distance from the leading edge. Comparing the laminar profiles to the LES

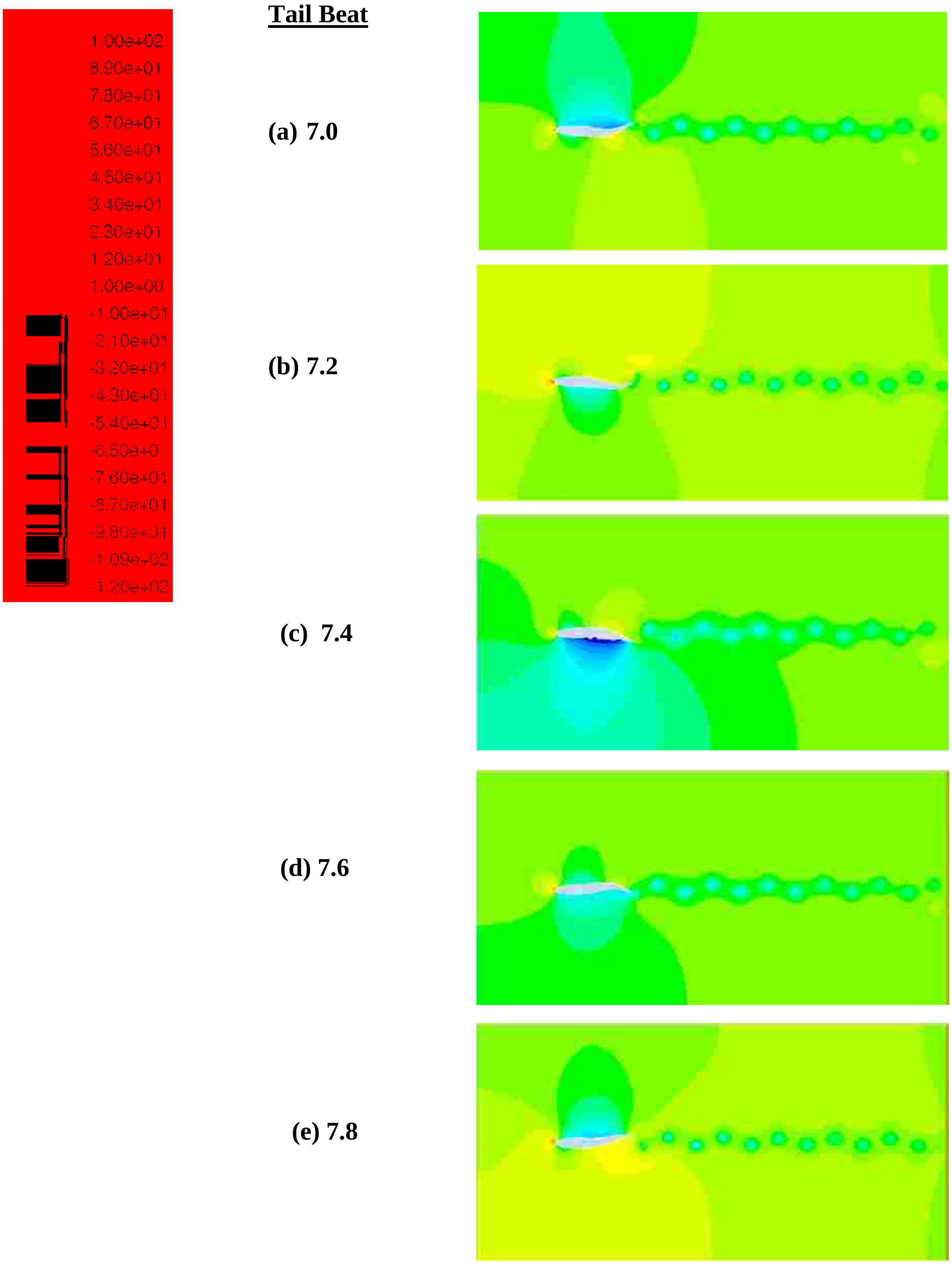


Figure 4-8: Pressure contour plots for the seventh tail beat of the laminar flow simulation.

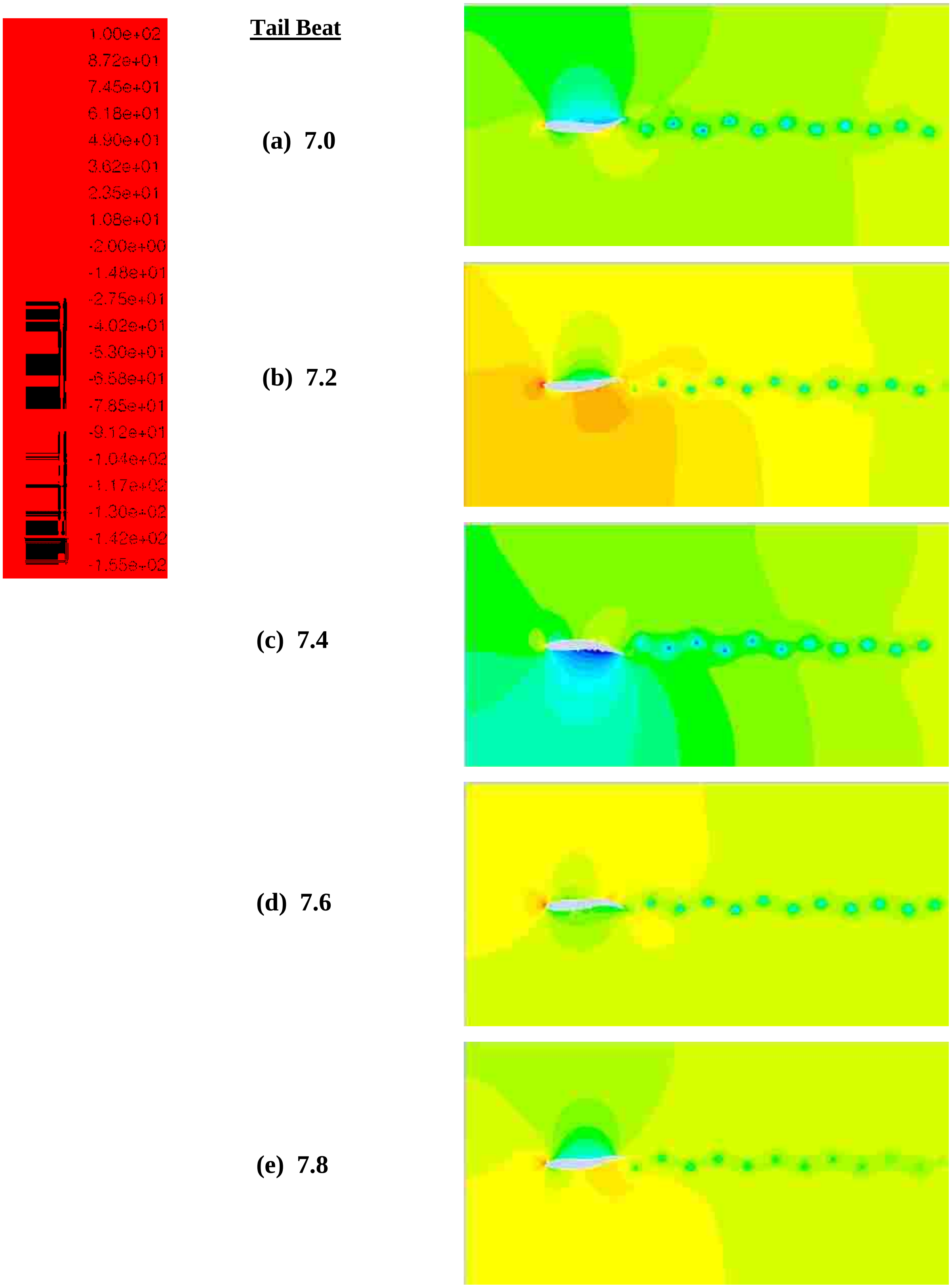


Figure 4-9: Pressure contour plots for the seventh tail beat of the LES simulation.

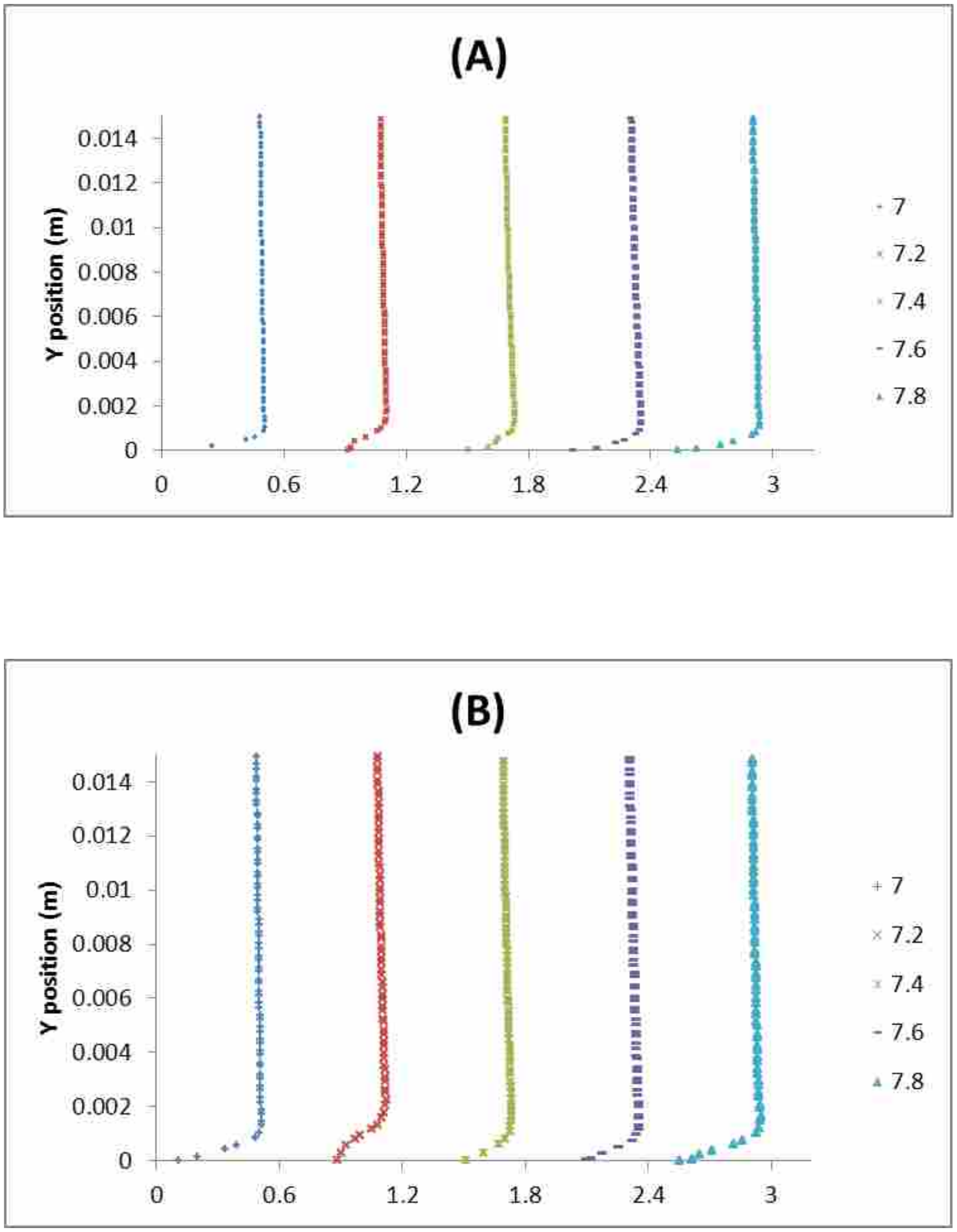


Figure 4-10: Velocity profile shapes of the upper (right hand) side of the fish at 0.2 chord lengths from the front end at selected time steps over the course of the seventh tail beat. (A) Laminar simulation. (B) LES simulation. Numbers in the legend refer to tail beat associated with each profile.

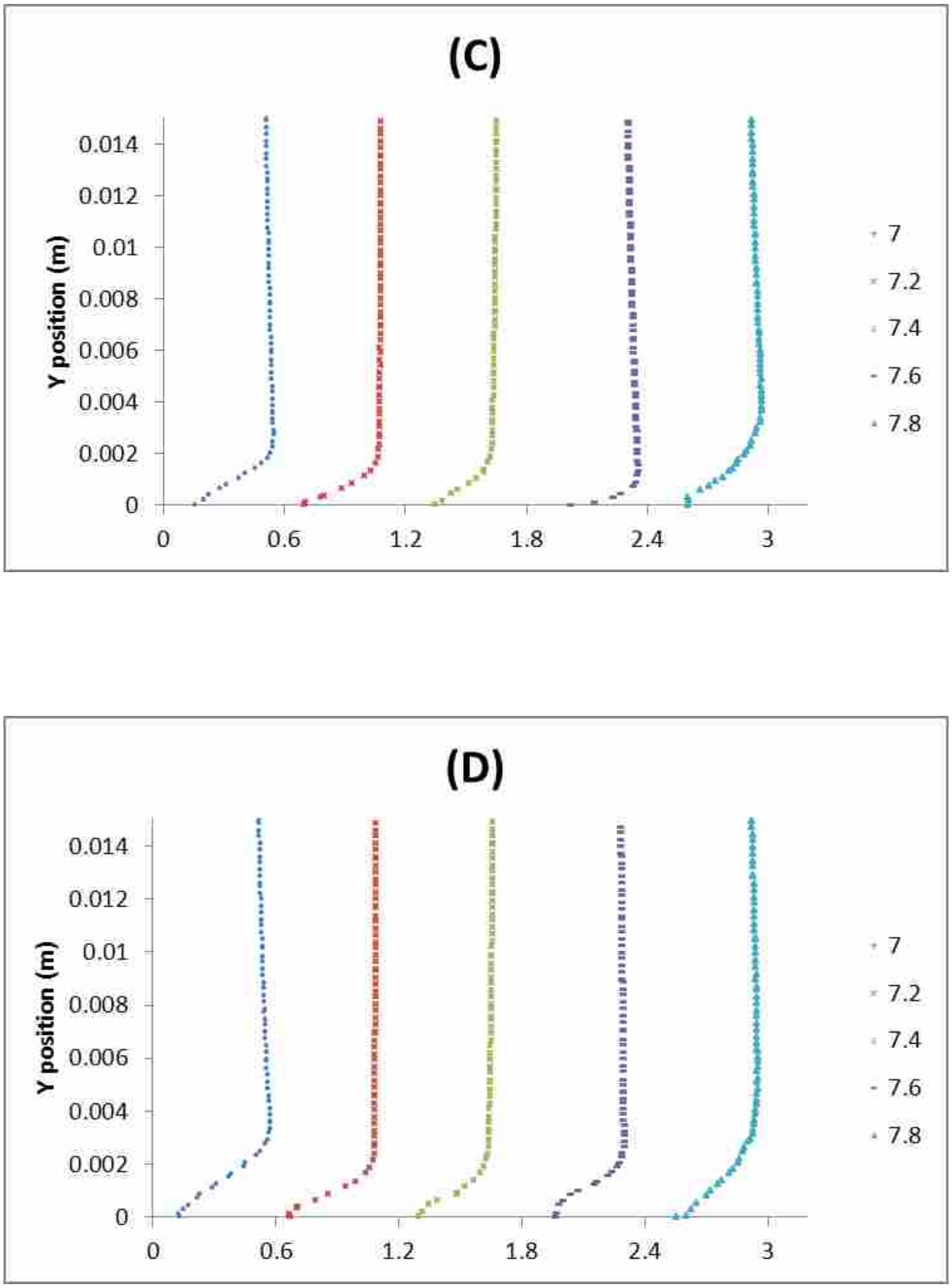


Figure 4-11: **Velocity profile shapes of the upper (right hand) side of the fish at 0.6 chord lengths from the front end at selected time steps over the course of the seventh tail beat. (C) Laminar simulation. (D) LES simulation. Numbers in the legend refer to tail beat associated with each profile.**

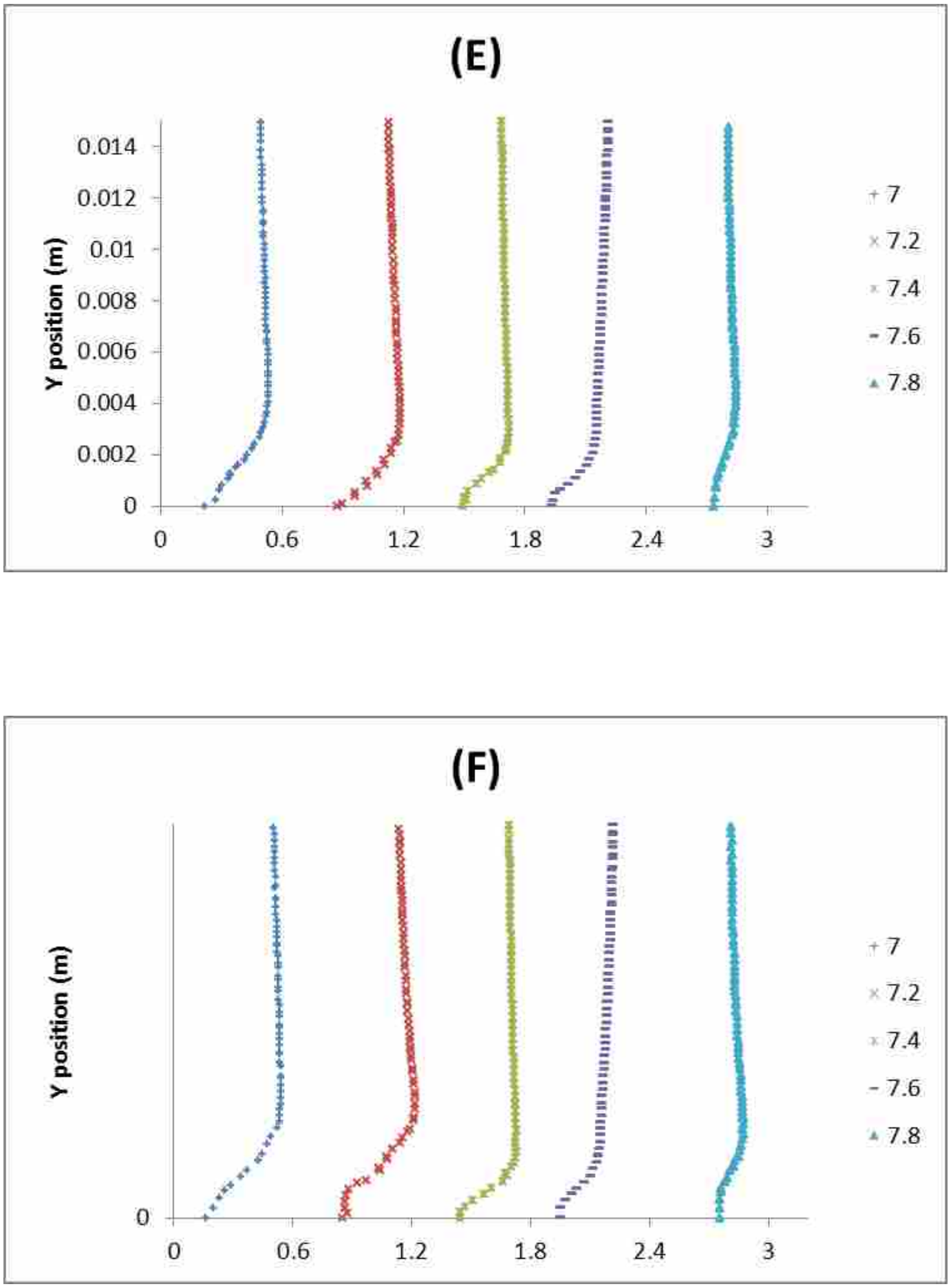


Figure 4-12: **Velocity profile shapes of the upper (right hand) side of the fish at 0.9 chord lengths from the front end at selected time steps over the course of the seventh tail beat. (E) Laminar simulation. (F) LES simulation. Numbers in the legend refer to tail beat associated with each profile.**

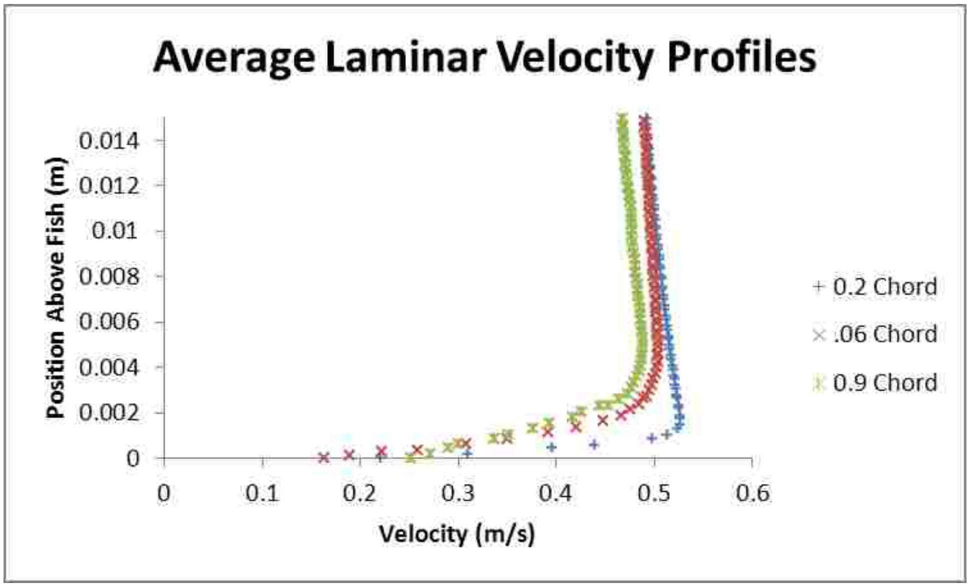


Figure 4-13: **Average velocity profile over one tail beat for the laminar flow simulation.**

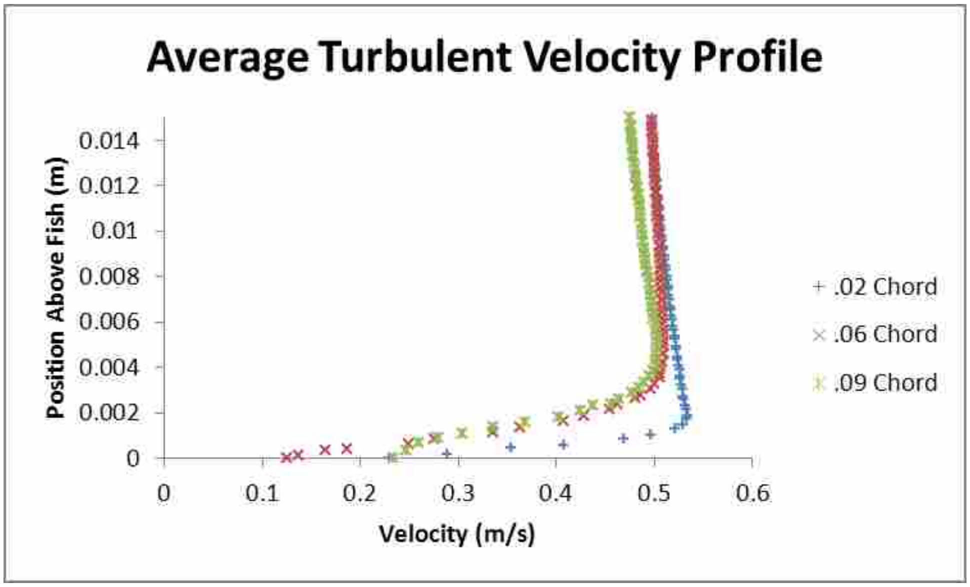


Figure 4-14: **Average velocity profile over one tail beat for the turbulent flow simulation.**

profiles, it is of note that the near wall velocities are lower in the LES simulation, indicating increased viscosity due to turbulence.

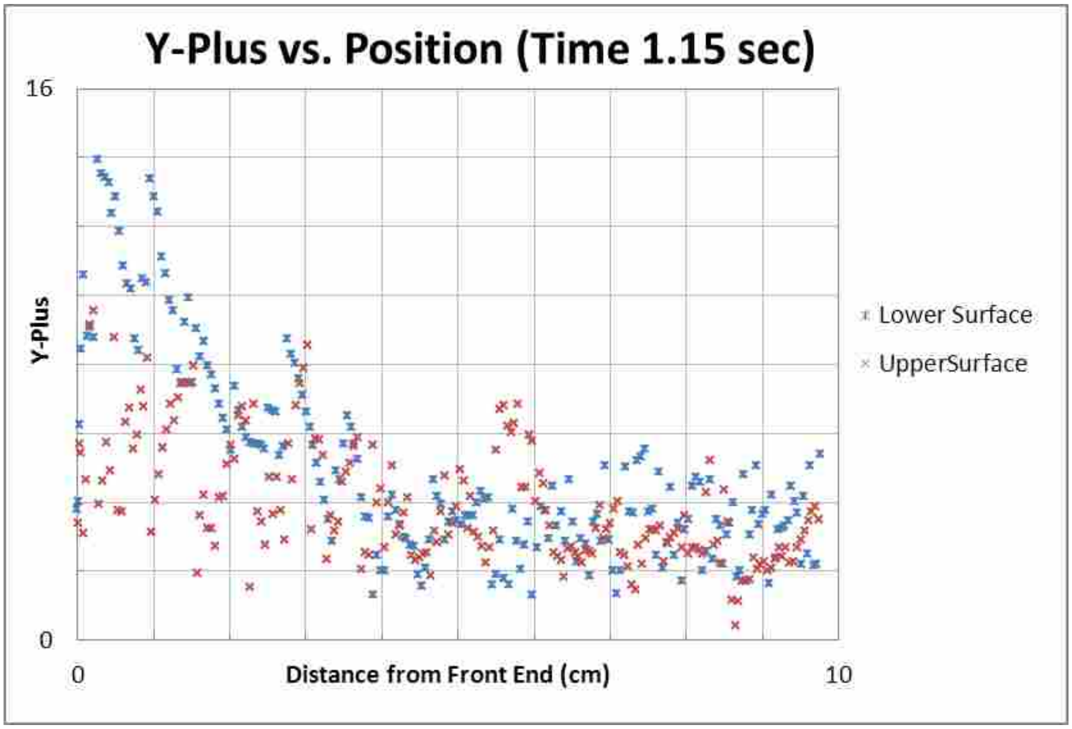


Figure 4-15: Example of y+ values along the length of both the upper and lower fish surfaces.

Averaging the velocity profiles over time at these same locations for the laminar flow simulation (see figure 4-13) and the LES simulation (see figure 4-14) to smooth out the random variations more clearly reveals a velocity overshoot in the profiles as the flow proceeds backwards along the fish. The boundary layer thickness is seen to increase from the front of the fish towards the back. The increase in flow velocity in the near wall regions is indicative of the creation of a high speed propulsive jet [Flanagan (2004)].

For LES cases, it is still possible to improve the grid sizing to achieve better results. Even though certain cells (especially near to the snout) had y+ values of as much as 16, the y+ value over most of the length of the actually fish fell between 2 and 6. Taking the y+ values at time 1.15 seconds for an example (figure 4-15), the upper surface had an average y+ value of 3.97 with a standard deviation of 1.76, and the lower surface had an average y+ value of 5.06

with a standard deviation of 2.94. Such a wide distribution of values in comparison to the mean is partially the result of local remeshing, as different cells are updated at different times depending on each cell’s size and skewness. As described in chapter 3, y+ values should be close to 1 for optimal flow resolution.

5 Discussion

5.1 Vortex Shedding Mechanism

Based on wall shear values, at no point during the simulations is boundary layer separation observed. Hence there are no vortices present to be shed along the length of the fish, meaning that vortex control is not being used to improve swimming efficiency in this particular case. There are, however, regions of high vorticity observed to be travelling backwards along the fish that eventually form into vortices shed from the tail into the downstream wake.

5.2 Energy Expended and Captured

Attempting to separate the work done by the fish from that done by the fluid can be complicated. First, it is empirically difficult to separate thrust and drag forces, and second, the Froude efficiency of the fluid should be zero [Schultz (2002)]. The method used to calculate power for this study is the same method used by Flanagan. The assumption used is that if the boundary wall moves in the same direction that the fluid pressure is pushing it, then the work is done by the fluid. If not, then the work is done by the fish.

5.3 Thrust and Drag

Analysis of the time -averaged data output from both of these simulations produced values for thrust and drag that are not equal to each other. In comparison to the Flanagan study, in which the thrust and drag were within 5% of each other, the percentage difference between the thrust and drag in the current study was about 30%, or six times larger.

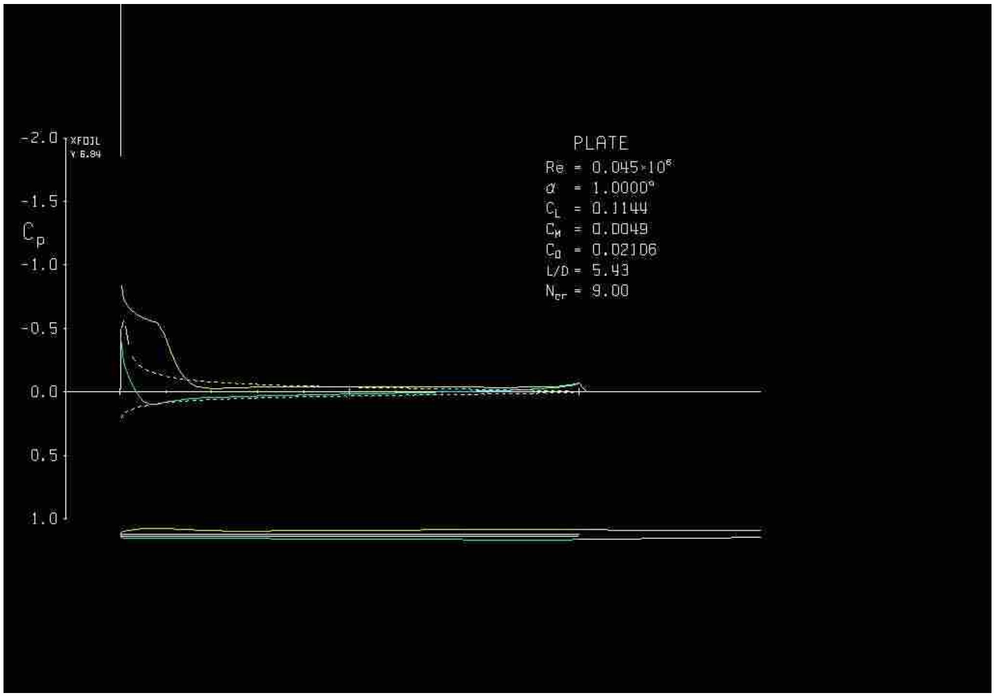


Figure 5-1: **XFOIL output of thin plate 1 degree angle of attack.**

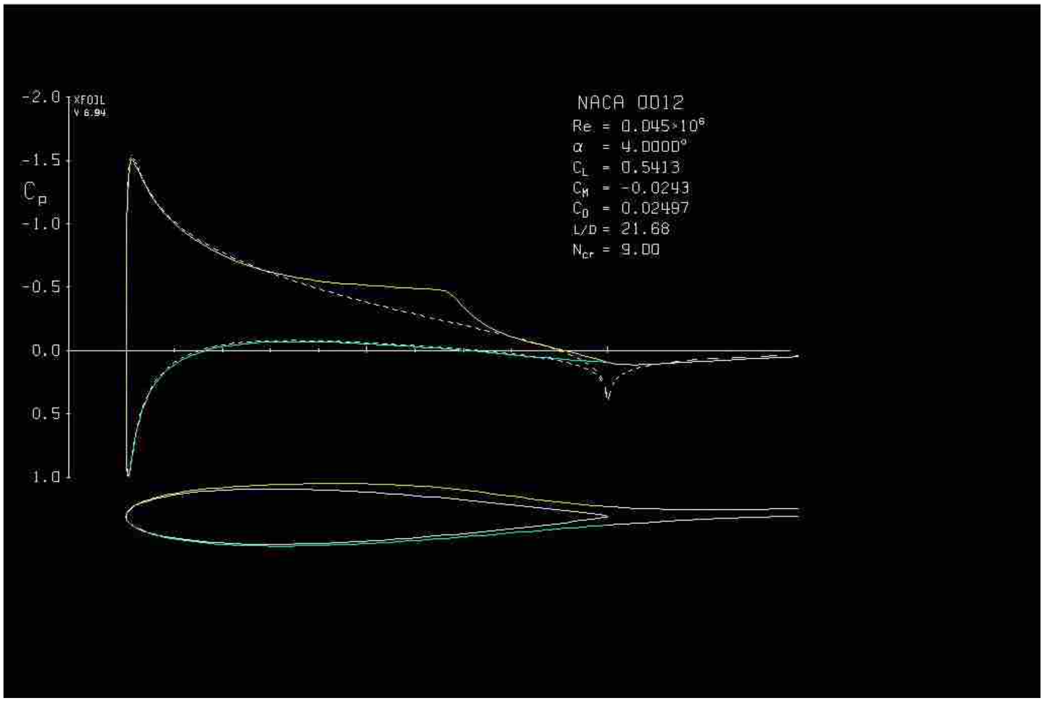


Figure 5-2: **XFOIL output of NACA-0012 at 4 degree angle of attack.**

Additional drag resulting from the foil shape used in this study was greater than that necessary to close the 5% thrust-to-drag ratio gap noted by Flanagan. Total drag increased from .611 N/m in the Flanagan study, to 0.813 N/m in this study's laminar case (an increase of 33%), and to .868 N/m in the turbulent LES case (an increase of 42%). The percentage of increase in thrust was even greater, increasing from 0.64 N/m the Flanagan study, to 1.05 N/m in this study's laminar case (an increase of 64%), and to 1.13 N/m in the turbulent case (an increase of 77%).

The reason for such a dramatic increase in calculated values for both the thrust and drag could be the improved hydrodynamic properties of the foil shape compared to those of a thin line. Because nearly all of the work done by the fish (about 97.6%) occurs in the transverse direction, any increase in lift should apply to the amount of thrust produced.

A series of simulations to obtain a comparison of the thrust and drag of a flat thin plate to that of a NACA-0012 airfoil was carried out using Mark Drela's XFOIL software package with a Reynolds number of 45,000 (Examples in figure 5-1 and figure 5-2). The simulations revealed that for a one degree angle of attack, the magnitude of the lift force of the NACA airfoil was 389% greater than that of the thin plate, though the "lift" force actually occurs in the opposite direction in that case. (At a zero degree angle of attack, a symmetric airfoil does not produce lift.) The lift-to-lift ratio of the airfoil to the plate decreases rapidly through two degrees, and above three degrees the value of the lift coefficient varies less drastically (only by about a factor of two) until stall occurs at a 10 degree angle of attack.

Drag values were not affected significantly below a two degree angle of attack, but can vary by as much as a factor of three at a six degree angle of attack. However, the flapping motion of the fish changes the hydrofoil's camber, and thus, with the snout remaining in-line with the direction of flow and the angle of the line between the snout and the tail consistently changing relative to the same flow, the effective angle of attack also changes. The maximum angle between the snout and the tail is about six degrees

In comparison to a flat plate, the change in thrust at a low angle of attack and the change in drag at a high angle of attack (the fish moves through both ranges), help to explain the differences between the values of thrust and drag obtained in this study and those of the reference study done by Flanagan [Flanagan (2004)].

5.4 Power

Power efficiency for both the laminar and turbulent simulations is calculated to be about 70%. This is somewhat larger than the efficiency of 62% that was calculated in the Flanagan study, though still considerably less than the 87% found by Anderson [Anderson, 1998].

With an assumed surface area of 36 cm^2 for a 10 cm long rainbow trout [Webb, 1985], and a calculated surface length of 20.4 cm, it can be approximated that the fish would extend approximately 1.78 cm into the screen. Using this value, we can calculate the approximate power output of the fish. In the laminar simulation, the fish produces 5.27 mW, while in the LES simulation, it produces 5.47 mW.

5.5 Turbulent Effects

A review of the velocity profile along the length of the fish for the LES simulation scenario did not reveal any instances of shear stress in the negative x-direction, indicating that boundary layer separation did not occur.

In general, the LES model produced a range of pressures that compare well with the laminar model on the positive pressure scale in that both had a maximum gauge pressure of about 100 Pa. However, the LES model had a lower minimum pressure, dropping down to around -155 Pa, whereas the laminar model tended to reach a minimum at about -120 Pa. This difference in pressure helps to account for the increase in drag on the LES model.

Overall, however, the actual variations in hydrodynamic properties (thrust, drag, and power) between the laminar and LES simulations were negligible. At least two reasons can be used to account for these similarities. First, the Reynolds number was only about 45,000, which is well below the value of Re = 500,000 commonly approximated for turbulent transition on a flat plate, thus turbulent effects may have been quite minimal. Second, even if turbulence effects do have an impact on the fish's swimming efficiency in reality, the inlet conditions were uniform with no prescribed inlet turbulence. This means that any turbulence created downstream of the fish would need to be numerically propagated back upstream over a number of time steps, and it is doubtful that there was enough simulation time for this to have adequately taken place.

6 Conclusions and Future Work

6.1 Conclusions

The objective of this study was to improve upon Flanagan's earlier CFD simulations of a two-dimensional, 10 cm long rainbow trout swimming in a 0.45 m/s steady inlet flow field. Primary improvements suggested by Flanagan's reference study were introduced one at a time in sequential simulations.

The reference study suggested that adding thickness to the thin centerline could close a 5% deficit between the calculated thrust and drag. The first simulation improved the simulated fish's geometry by adding thickness (in the form of a NACA-0012 foil) so that it better approximated that of a rainbow trout. While the different fish geometry did increase the drag, as suspected, the calculated increase was more than 5%. Additionally, thrust was increased more than the drag was, so that the final result was an even greater disparity between thrust and drag.

The reference study also suggested that adding a turbulence model might reveal some transition to turbulence which could allow for separation along the fish's boundary layer. Such a finding would suggest that the fish is using a method of vortex control to improve its swimming efficiency. The second simulation replaced the laminar flow model with an LES turbulence model for this purpose, as well as retaining the improved geometry from the first simulation. The results of this second simulation did not reveal the presence of separation along the boundary layer, strongly suggesting that perhaps rainbow trout do not use vortex control to enhance their swimming efficiency.

A third objective of modeling a two-dimensional rainbow trout swimming with a Karman gait in a vortex flow-field was partially completed and will be described later in this section.

Additional objectives were also completed. The boundary layer and the downstream wake were characterized, and an analysis of the power efficiency was presented.

6.2 Preparations for Karman Gait Simulation

Because part of the basis of this research is to eventually test the assertion by Liao et al that a rainbow trout utilizes the Karman gait [Liao (2003)] to conserve energy, further research should seek to simulate the conditions of turbulence in which the original research was performed. Work has been performed by this author to begin such a study, which for the purposes of this section shall be referred to as a Turbulent Inlet Flow Simulation (TIFS).

In order for the fish's motion in the simulation to approximate Liao's Karman gait, several kinematic changes need to be made for the TIFS case. These changes include increasing the centerline wavelength to 20.4 cm , increasing the period of each tail beat to 0.45 s, multiplying the tail beat amplitude profile by a factor of -2.3, and introducing a vertical motion path with an amplitude of 2.5 cm and a period of 0.45 s to match the tail beat [Liao (2003)]. While the UIFS uses the Newton-Raphson method for as a closure routine in determining the location of each node along the length of the fish at each time-step update, that method becomes unstable in the TIFS. This instability may be caused by the increased centerline amplitude. To overcome this difficulty, a more stable but less efficient closing approximation method is used. UDF files for both the hydrofoil and oscillating flat plate geometries can be found in Appendix B and Appendix C respectively.

The grid for the TIFS case has approximately 735,000 elements (figure 6-1), an increase of 40% over the UIFS case. The additional cells are the result of an enlarged region of fine cells necessary to ensure grid quality as the fish translates vertically on the screen around the inlet vortices.

The TIFS considered here utilizes a separate UDF to simulate an alternating Karman vortex street inlet. The individual vortices are modeled as Rankine vortices, meaning that a forced vortex core of radius R is surrounded by a free vortex. Rankine vortices better match the characteristics of vortices in a viscous fluid than do simple free vortices that are characteristic of inviscid fluids. They obey the following velocity profile definition (Acheson, 1990):

$$V_c(r) = \frac{V_0 r}{R} \qquad r<R \qquad (6\text{-}1)$$

$$V_c(r) = \frac{V_0 R}{r} \qquad r>R \qquad (6\text{-}2)$$

where $V_c(r)$ is the rotational fluid velocity at radius r from the center of the vortex and V_0 is the rotational velocity at the vortex core radius R.

The UDF creates two rows of vortices with counter-rotating velocity fields separated vertically and offset horizontally by pre-defined distances S and W, respectively. While the value for the horizontal offset can be easily found in Liao's publication, the values for R and V_0 as well as the row separation distance S still need to be defined following some trial and error. The C program "vortexgen.c" that serves as the velocity inlet UDF is found in Appendix D. A sample output with S = 5 cm, W=20.4 cm, R= 2 cm, and V0 = 15 rad/s is shown in figure 6-2.

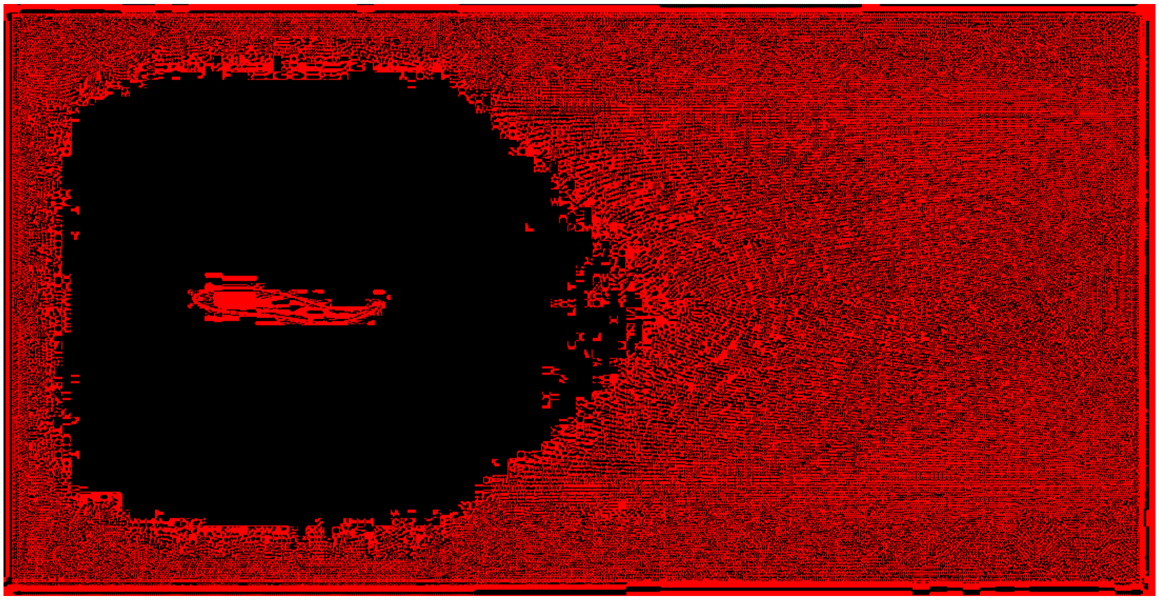


Figure 6-1: **Grid and potential fish shape used for a turbulent inlet flow scenario (TIFS) simulation. Notice the increased amplitude of the fish centerline and larger region of dense cells near the fish.**

The TIFS simulation FLUENT journal file as written covers 1.7 seconds to allow the inlet Karman vortex street to become fully developed within the flow domain and to allow for at least three full tail beat cycles in proximity to the inlet vortices.

A simulation using a 40 node centerline fish (without the hydrofoil shape) was run for 575 time steps of .001 seconds each (figure 6-3). Of note is the fact that the fish's undulatory motion still produces a series of shed vortices and there are signs that some vortex interaction may be taking place between the vortices shed by the fish and those introduced at the flow inlet.

A full length simulation of a 2D rainbow trout utilizing the Karman gait in a turbulent wake should be possible as the next step following the line of research of this study.

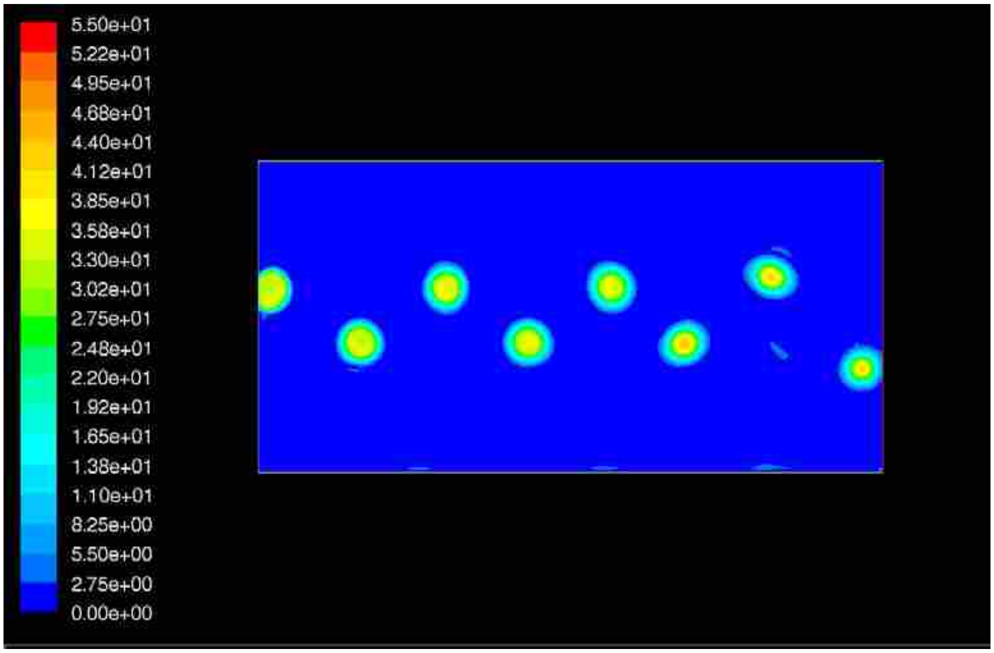


Figure 6-2: **Sample vortex flow simulation vorticity output from UDF "vortexgen.c" executed in FLUENT ver. 6.3. Flow pattern corresponds to time 1.62 seconds. Absolute vorticity in 1/s are displayed.**

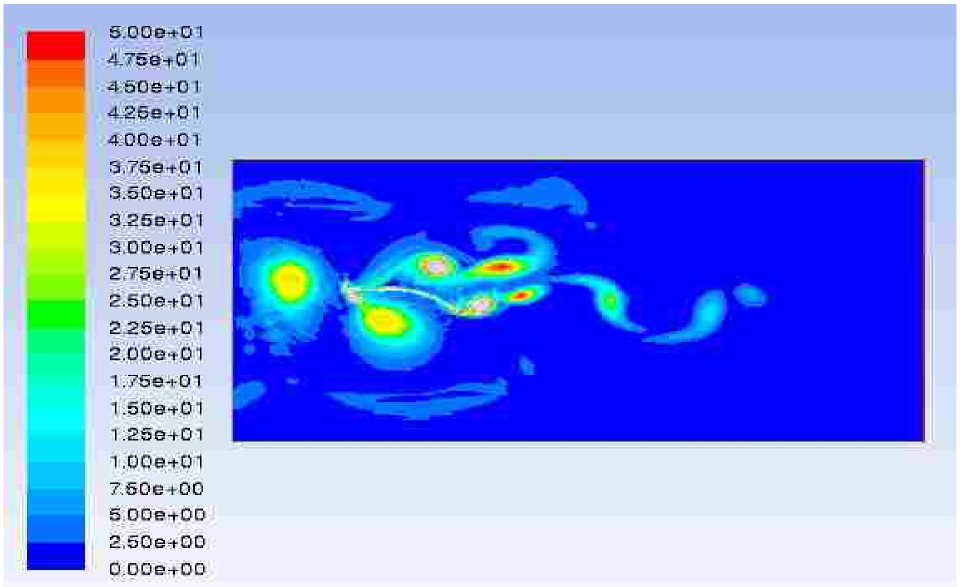


Figure 6-3: Vorticity contour plot of a 40-node centerline fish in a vortex inlet flow at time 0.575 seconds.

6.3 Future Work

Additional work in the future should include the simulation of different cases for the purpose of comparison. Such cases could include variations in the Reynolds number, tail beat frequency, maximum amplitude of motion, and centerline wavelength. Improvements to the shape of the hydrofoil to better match that of a rainbow trout should also be made. At some point, the position of the snout should be unpinned to allow the fish to move freely through the domain. The reason for non-physical thrust oscillation described in Ch. 4 should be solved.

Much work remains to be done in order to properly define the characteristics of inlet vortices for simulations in which the fish utilizes the Karman gait. The best methods for determining thrust and drag in cases in which the boundary layer has separated will need to be determined for these simulations as well. 3-D simulations should also be performed building both on 3-D simulations performed by others, and on the 2-D TIFS simulation described here.

REFERENCES

Acheson, D.J. *Elementary Fluid Dynamics*. New York, NY: Oxford University Press. 1990.

Adkins, D. and Yan, Y.Y. “CFD Simulation of Fish-like Body Moving in Viscous Liquid.” *Journal of Bionic Engineering*. vol. 3 (2006): 147-153.

Anderson, J.M. and Streitlein, K. and Barrett, D.S. and Triantafyllou, M.S. “Oscillating Foils of High Propulsive Efficiency”. *Journal of Fluid Mechanics*, vol. 360(1998): 41-72.

Azuma, A. and Okamoto, M. “Theoretical study on two-dimensional aerodynamic characteristics of unsteady wings.” *Journal of Theoretical Biology*. vol. 234 (May 2005): 67-78.

Blondeaux, P. et al. “Numerical Experiments on Flapping Foils Mimicking Fish-Like Motion”. *Physics of Fluids*. vol. 17 (November 2005): 113,601.

Borzajani, I. and Sotiropoulos, F. “Numerical Investigation of the Hydrodynamics of Carangiform Swimming in the Transitional and Inertial Flow Regimes.” *Journal of Experimental Biology*. vol. 211 (May 2008): 1541-1558.

Borzajani, I. and Sotiropoulos, F. “Numerical Investigation of the Hydrodynamics of Anguilliform Swimming in the Transitional and Inertial Flow Regimes.” *Journal of Experimental Biology*. vol. 212, Issue 4 (February 2009): 576-592.

Canuto, V.M. “Turbulent Convection: Is 2D a Good Proxy of 3D”. *Astronomy and Astrophysics*. vol. 357 (2000): 177-179.

Celik, I. and O. Karatekin. “Numerical Experiments on Application of Richardson Extrapolation with Nonuniform Grids.” *Journal of Fluids Engineering*, vol. 119 (1997): 584-590.

Chung, M-H. “On burst-and-coast swimming performance in fish-like locomotion.” *Bioinspiration and Biomimetics*. vol. 4 (2009): 036001.

Claireaux, G. and Couturier, C. and Groison, A. “Effect of Temperature on Maximum Swimming Speed and Cost of Transport in Juvenile European Sea Bass (*Dicentrarchus labrax)”*. *The Journal of Experimental Biology*, vol 209 (2006): 3420-3428.

Conroy, A. Wiktorowicz and Gordon, M. "How to swim the Pufferfish way: A biomechanical and CFD analysis." *Comparative Biochemistry and Physiology - Part A: Molecular and Integrative Physiology*. vol. 150, Issue 3, Supplement 1 (July 2008): S84.

Dogangil, G. and Ozicek, E. and Kuzucu, A. "Modeling, Simulation, and Development of a Robotic Dolphin Prototype". In *Proceedings of the IEEE International Conference on Mechatronics and Automation*. Niagara Falls, Canada: July 2005.

Dong H. and Mittal, R. and Najjar, F.M. "Wake topology and hydrodynamic performance of low-aspect-ratio flapping foils." *Journal of Fluid Mechanics*. vol. 566 (2006): 309–343.

Ducoin, A. and Astolfi, J.A. and Deniset, F. and Sigrist, J. "Computational and experimental investigation of flow over a transient pitching hydrofoil."   *European Journal of Mechanics B/Fluids*. vol. 28 (2009): 728–743.

Enders, E.C. and Boisclair, D. and Roy, A.G. "The Effect of Turbulence on the Cost of Swimming for Juvenile Atlantic Salmon (*Salmo salar*)". *Canadian Journal of Fisheries and Aquatic Sciences.* vol. 60 (2003): 1149-1160.

Fauci, L.J. "A Computational Model of Fluid Dynamics of Undulatory and Flagellar Swimming". *American Zoology*, vol. 36 (1996): 599-607.

Flanagan, P. "Unsteady Navier Stokes Simulation of Rainbow Trout Swimming Hydrodynamics." Master's Thesis, Washington State University, May 2004.

FLUENT 6.3 User's Guide. Lebanon, N.H.: FLUENT, Inc. September 2006.

Goodwin, R.A. and Nestler, J.M. and Anderson, J.J. and Weber, L.J. and Loucks, D.P. "Forecasting 3-D fish movement behavior using a Eulerian–Lagrangian–agent method (ELAM)."   *Ecological Modelling*, vol. 192 (2006): 197–223.

Gray, J. "The propulsive powers of the dolphin." *Studies in Animal Locomotion VI* (1936): Abstract.

Hu, T. and Shen, L. and Lin, L. and Xu, H. "Biological inspirations, kinematics modeling, mechanism design and experiments on an undulating robotic fin inspired by Gymnarchus niloticus." *Mechanism and Machine Theory*. vol. 44 (2009): 633–645.

Jacobs, Eastman. *Technical Notes for National Advisory Committee for Aeronautics No. 385 - Tests of Six Symmetrical Airfoils in the Variable Density Wind Tunnel*. Washington, DC: Langley Memorial Aeronautical Laboratory, July 1931.

Jordan, C.E. "Coupling Internal and External Mechanics to Predict Swimming Behavior: a General Approach." *American Zoology*, vol. 36 (1996): 710-722.

Lauder, G.V. “Swimming Hydrodynamics: Ten Questions and the Technical Approaches Needed to Resolve Them.” *Experimental Fluid*, vol. 47 (September 2009): 3-15.

Leroyer, A. and Visonneau, M. “Numerical methods for RANSE simulations of a self-propelled fish-like body.” *Journal of Fluids and Structures*. vol. 20 (2005): 975–991.

Liao, J.C. “The Karman Gait: Novel Body Kinematics of Rainbow Trout Swimming in a Vortex Street”. *The Journal of Experimental Biology*. vol. 206 (203): 1059-1073.

Liao, J.C. “Neuromuscular Control of Trout Swimming in a Vortex Street: Implications for Energy Economy during the Karman gait.” *Journal of Experimental Biology.* vol. 207 (September 2004): 3495-3506.

Lighthill, M.J. “Large-Amplitude Elongated-Body Theory of Fish Locomotion.” *Proceedings of the Royal Society of London. Series B, Biological Sciences*. vol. 179, Issue 1055 (1971): 125-138.

Lighthill, J. *Mathematical Biofluiddynamics*. Philadelphia, PA: Society for Industrial and Applied Mathematics,1975.

Low, K.H. “Modelling and parametric study of modular undulating fin rays for fish robots.” *Mechanism and Machine Theory*. vol. 44 (2009): 615–632.

Lupandin, A.I. “Effect of Flow Turbulence on Swimming Speed of Fish.” *Biology Bulletin*. vol. 32, Issue 5 (April 2005):. 461-466.

May, M.L. “Dragonfly Flight: Power Requirements at High Speed and Acceleration.” *Journal of Experimental Biology*. vol. 158 (February 1991): 325-342.

McMasters, R.L. and Grey, C.P. and Sollock, J.M. and Mukherjee, R. and Benard, A. and Diaz, A.R. “Comparing the Mathematical Models of Lighthill to the Performance of a Biomimetic Fish.” *Bioinspiration and Biomimetics.* vol. 3 (2008): 016002.

McNeill, A.R. “Models of the Scaling of Energy Costs for Locomotion.” *Journal of Experimental Biology*. vol. 208, Issue 9 (April 2005): 1645-1652.

Miao J.-M. and Ho M.-H. “Effect of Flexure on Aerodynamic Propulsive Efficiency of Flapping Flexible Airfoil.” *Journal of Fluids and Structures.* vol. 22 (2006): 401-419.

Mueller, U and Van Leuwen, J. “Undulatory Fish Swimming: From Muscles to Flow.” *Fish and Fisheries.* vol. 7 (2006): 84-103.

Narasimhan, M. and Dong, H. and Mittal, R. and Singh, S.N. “Optimal Yaw Regulation and Trajectory Control of Biorobotic AUV Using Mechanical Fins Based on CFD Parametrization.” *Transactions of the ASME 698*. vol. 128 (July 2006): 687-698.

Nikora, V.I. et al. “Effects of Fish Size, Time to Fatigue, and Turbulence on Swimming Performance. A Case Study of *Galaxias maculatas*.” *Journal of Fish Biology*. vol. 63 (2003): 1365-1382.

Peng and Dabiri. “The ‘Upstream Wake’ of Swimming and Flying Animals and Its Correlation with Propulsive Efficiency.” *Journal of Experimental Biology.* vol. 211 (2008): 2669-2677.

Schultz, W.W. and Webb, P.W. “Power Requirements of Swimming: Do new Methods Resolve Old Questions?” *Integrative and Comparative Biology*. vol. 42 (2002): p1018-1025.

Sears, W.R. and VanDyke, M. *Annual Review of Fluid Mechanics.* Palo Alto, CA: Eds. Annual Review, 1969.

Shen, Y. and Diplas, P. “Application of Two- and Three-Dimensional Computational Fluid Dynamics Models to Complex Ecological Stream Flows.” *Journal of Hydrology*. vol. 348 (2008): 195– 214.

Suleman, A. “Design and Testing of Biomimetic Tuna Using Shape Memory Alloy Induced Propulsion.” *Computers and Structures*. vol. 86 (2008): 491-499.

Taguchi, M. and Liao, J. “Rainbow Trout Consume Less Oxygen in Turbulence: The Energetics of Swimming Behaviors at Different Speeds.” *The Journal of Experimental Biology*. vol. 214 (January 2011): 1428-1436.

Techet, A. “Propulsive Performance of Biologically-Inspired Flapping Foils at High Reynolds Numbers.” *Journal of Experimental Biology*. Volume 211 (2008): pp. 274-279.

Thomson, S. “Shape Optimization and Fluid Dynamic Analysis of a Translating Flexible Body.” In *Proceedings of the 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* . Orlando, FL: July 2010.

Triantafyllou, M.S. and Triantafyllou, G.S. “An Efficient Swimming Machine.” *Scientific American.* vol. 272, Issue 3 (March 1995): 64.

Triantafyllou, M.S. et al. “Vorticity Control in Fish-Like Propulsion and Maneuvering.” *Integrative and Comparative Biology*. vol. 42 (January 2002): 1026-1051.

Tytell, E.D. and Standen, E.M. and Lauder, G.V. “Escaping Flatland: Three-dimensional Kinematics and Hydrodynamics of Median Fins in Fishes.” *Journal of Experimental Biology*. vol. 211 (2008): 187-195.

Webb, P. W. and Cotel, A. J. “Turbulence: Does Vorticity Affect the Structure and Shape of Body and Fin Propulsors?” *Integrative and Comparative Biology.* vol. 50, number 6 (April 2010): 1155-1166.

Windsor, S.P. and McHenry, M.J. “The Influence of Viscous Hydrodynamics on the Fish Lateral-Line System.” *Integrative and Comparative Biology*. vol. 49, Issue 6 (August 2009): 691-701.

Zhang, Y-H. and Song, Y. and Yang, J. and Low, K.H. “Numerical and Experimental Research on Modular Oscillating Fin.” *Journal of Bionic Engineering.* vol. 5 (2008): 13−23.

Zhang, Z. and Gil, A.J. and Hassan, O. and Morgan, K. “The Simulation of 3D Unsteady Incompressible Flows with Moving Boundaries on Unstructured Meshes.” *Computers and Fluids.* Volume 37 (2008): 620–631.

Appendix A. UDF for Hydrofoil Motion (Uniform Flow Inlet)

```
#include "/fslhome/drc53/udf.h"

static float dist=0.05;  /*IMPORTANT TO ADJUST rad(cm) DEPENDING ON INTERVAL
LENGTH */

DEFINE_GRID_MOTION(clinemvfishupper, domain, dt, time, dtime)
{
static float tcheck=0.0;

static int count;

Thread *tf = DT_THREAD (dt);

face_t f;

Node *v;

float NV_VEC (axis), NV_VEC (snout);

float len, divisions, segment, wave, per, alpha, per_snout, omega, rad, thick, snout_y, s, aloc,
bloc, xpos, ypos;

float theta, amp, fx, dfx, delx, conv, tempxcen, tempycen, slope, foil, xpos_u, ypos_u;
int n, i;

conv=1.0e-9;    /* set convergence tolerance (units are cm) */

 if (tcheck < time)

{count=0; tcheck=time;}                    /* initialize count */

/* set constants for kinematic model (use units of cm and sec) */

 len=10.0;    /*fish centerline length*/

divisions=4;  /*Number of divisions along the first regular node length*/
```

```
segment=divisions*(divisions+1)/2;  /*Weighted length of divisions along first node length (e.g.
length of fourth division is  4 x segment) */

 wave=11.5;   /*wavelength of centerline oscillation*/

per=0.15;    /*period of centerline oscillation*/

alpha = 0; /*amplitude of snout path*/

per_snout = 1; /*period of snout path oscillation*/

thick = 0.125;     /*maximum thickness of the fish's hydrofoil as a decimal part of centerline
length*/

/* set deforming flag on adjacent cell zone */

SET_DEFORMING_THREAD_FLAG (THREAD_T0 (tf));

Message ("\n time = %f\n", time);

Message ("tcheck = %f\n", tcheck);

 NV_D (axis, =, 0.0, 1.0, 0.0);

omega = 1/per_snout;

snout_y = 0.15 + alpha * sin(2 * M_PI * omega * time);

NV_D (snout, =, 0.10, snout_y, 0.0);

s = 0;

begin_f_loop (f, tf)
 {
f_node_loop (f, tf, n)
{
 v = F_NODE (f, tf, n);

 /* update node IF x position is greater than 0.10
and the current node has not been previously
visited when looping through previous faces */

 if ( /* NODE_X (v) > 0.100 && */  NODE_POS_NEED_UPDATE (v))
{
/* indicate that node position has been updated
so that it's not updated more than once */
```

```
NODE_POS_UPDATED (v);

count=count+1;

/*Message ("NODE_X(v)= %f,NODE_Y(v)=%f\n",NODE_X(v),NODE_Y(v));*/

if(count<=divisions+1)

 rad = count * dist / segment;

else
{
rad = dist;
}

 if (count==1)              /* find (x,y) for snout at new time */

 xpos=0.0;

else    /* Use Newton's method to iteratively solve for new (x,y) */
{
xpos=aloc+rad;

for (i=1; i<=200; i++)
{
 theta =2.0*M_PI*((xpos/wave)-(time/per));

/*Message ("i = %d\n", i);*/

amp = len*(0.351*sin((xpos/len)-1.796)+0.359);

fx = pow(xpos-aloc,2.0)+pow(amp*sin(theta)-bloc,2.0)-pow(rad,2.0);

dfx = 2.0*(xpos-aloc)+2.0*(amp*sin(theta)-bloc)*((2.0*M_PI*amp*cos(theta)/wave)+ 0.351 *
sin(theta) * cos((xpos/len)-1.796));

delx = fx/dfx;

xpos = xpos-delx;

if ( fabs(delx) <= conv)

i=201;              /* set i = max to break out of loop once solution is converged */

else
{
```

```
if (i==200)

Message ("Newton's method didn't converge");
}
}
}

ypos=len*(0.351*sin((xpos/len)-1.796)+0.359)*sin(2.0*M_PI*((xpos/wave)-(time/per)));

tempxcen = (xpos/100.0) + snout[0];

tempycen = (ypos/100.0) + snout[1];

slope = (ypos - bloc) / (xpos - aloc);   /*slope of centerline at current node*/

aloc=xpos;

bloc=ypos;       /* set a and b to calculate the next node */

/*Add geometry of the fish's hydrofoil*/

foil = .01 * len * (thick/0.2) * (.2969 * sqrt(s/len) - 0.126 * (s/len) - 0.3516 * pow(s/len, 2) +
0.2843 * pow(s/len, 3) - 0.1015 * pow(s/len, 4));

xpos_u = tempxcen + foil * cos(atan(slope) + M_PI/2);

ypos_u = tempycen + foil * sin(atan(slope) + M_PI/2);

NODE_X(v) = xpos_u;

NODE_Y(v) = ypos_u;

s = s + rad;
}
}
}
end_f_loop (f, tf);

Message ("\n count = %d, xpos = %e, ypos = %e\n", count, xpos, ypos);

}


DEFINE_GRID_MOTION(clinemvfishlower, domain, dt, time, dtime)
{
static float tcheck=0.0;
static int count;

Thread *tf = DT_THREAD (dt);

face_t f;

ode *v;

float NV_VEC (axis), NV_VEC (snout);

float len, divisions, segment, wave, per, alpha, per_snout, omega, rad, thick, snout_y, s, aloc,
bloc, xpos, ypos;

float theta, amp, fx, dfx, delx, conv, tempxcen, tempycen, slope, foil, xpos_u, ypos_u;

int n, i;

conv=1.0e-9;    /* set convergence tolerance (units are cm) */

if (tcheck < time)

{count=0; tcheck=time;}                    /* initialize count */

/* set constants for kinematic model (use units of cm and sec) */

len=10.0;    /*fish centerline length*/

divisions=4;  /*Number of divisions along the first regular node length*/

segment=divisions*(divisions+1)/2;  /*Weighted length of divisions along first node length (e.g.
length of fourth division is  4 x segment) */

wave=11.5;   /*wavelength of centerline oscillation*/

per=0.15;    /*period of centerline oscillation*/

alpha = 0; /*amplitude of snout path*/

per_snout = 1; /*period of snout path oscillation*/

thick = 0.125;     /*maximum thickness of the fish's hydrofoil as a decimal part of centerline
length*/

 /* set deforming flag on adjacent cell zone */
```

```
SET_DEFORMING_THREAD_FLAG (THREAD_T0 (tf));

essage ("\n time = %f\n", time);

Message ("tcheck = %f\n", tcheck);

 NV_D (axis, =, 0.0, 1.0, 0.0);

omega = 1/per_snout;

snout_y = 0.15 + alpha * sin(2 * M_PI * omega * time);

NV_D (snout, =, 0.10, snout_y, 0.0);

s = 0;

begin_f_loop (f, tf)
{
 f_node_loop (f, tf, n)
{
v = F_NODE (f, tf, n);

 /* update node IF x position is greater than 0.10 and the current node has not been previously
visited when looping through previous faces */

if ( /* NODE_X (v) > 0.100 && */  NODE_POS_NEED_UPDATE (v))

{/* indicate that node position has been updated so that it's not updated more than once */

NODE_POS_UPDATED (v);

count=count+1;

/*Message ("NODE_X(v)= %f, NODE_Y(v)=%f\n",NODE_X(v),NODE_Y(v));*/

if(count<=divisions+1)

rad = count * dist / segment;

else
{
rad = dist;
}

if (count==1)              /* find (x,y) for snout at new time */
{
```

```
aloc=0.0;

xpos = aloc + rad;

 bloc=len*(0.351*sin((xpos/len)-1.796)+0.359)*sin(2.0*M_PI*((xpos/wave)-(time/per)));

for (i=1; i<=200; i++)
{
theta =2.0*M_PI*((xpos/wave)-(time/per));

amp = len*(0.351*sin((xpos/len)-1.796)+0.359);

fx = pow(xpos-aloc,2.0)+pow(amp*sin(theta)-bloc,2.0)-pow(rad,2.0);

dfx = 2.0*(xpos-aloc)+2.0*(amp*sin(theta)-bloc)*((2.0*M_PI*amp*cos(theta)/wave)+
0.351*sin(theta)*cos((xpos/len)-1.796));

delx = fx/dfx;

xpos = xpos-delx;

if ( fabs(delx) <= conv)

 i=201;          /* set i = max to break out of loop once solution is converged */

else
{
if (i==200)

Message ("Newton's method didn't converge");
}
}
}
else    /* Use Newton's method to iteratively solve for new (x,y) */
{
xpos=aloc+rad;

for (i=1; i<=200; i++)
{
theta =2.0*M_PI*((xpos/wave)-(time/per));

amp = len*(0.351*sin((xpos/len)-1.796)+0.359);

fx = pow(xpos-aloc,2.0)+pow(amp*sin(theta)-bloc,2.0)-pow(rad,2.0);
dfx = 2.0*(xpos-aloc)+2.0*(amp*sin(theta)-bloc)*((2.0*M_PI*amp*cos(theta)/wave)+
0.351*sin(theta)*cos((xpos/len)-1.796));

delx = fx/dfx;

xpos = xpos-delx;

if ( fabs(delx) <= conv)

i=201;          /* set i = max to break out of loop once solution is converged */

{
if (i==200)

Message ("Newton's method didn't converge");
}
}
ypos=len*(0.351*sin((xpos/len)-1.796)+0.359)*sin(2.0*M_PI*((xpos/wave)-(time/per)));

tempxcen = (xpos/100.0) + snout[0];

tempycen = (ypos/100.0) + snout[1];

slope = (ypos - bloc) / (xpos - aloc);

aloc=xpos;

bloc=ypos;      /* set a and b to calculate the next node */

/*Add geometry of the fish's hydrofoil*/

foil = .01 * len * (thick/0.2) * (.2969 * sqrt(s/len) - 0.126 * (s/len) - 0.3516 * pow(s/len, 2) +
0.2843 * pow(s/len, 3) - 0.1015 * pow(s/len, 4));

xpos_u = tempxcen - foil * cos(atan(slope) + M_PI/2);

ypos_u = tempycen - foil * sin(atan(slope) + M_PI/2);

NODE_X(v) = xpos_u;
NODE_Y(v) = ypos_u;

s = s + rad;
}
}
}
end_f_loop (f, tf);
```

```
Message ("\n count = %d, xpos = %e, ypos = %e\n", count, xpos, ypos);
}
```

Appendix B. UDF for Hydrofoil Motion (Von Karman Vortex Street Inlet)

```
#include "/fsc/drc53/GradApps/udf.h"

static float dist=0.05;  /*IMPORTANT TO ADJUST rad(cm) DEPENDING ON INTERVAL
LENGTH */

DEFINE_GRID_MOTION(clinemvfishupper, domain, dt, time, dtime)
{
static float tcheck=0.0;

static int count;

Thread *tf = DT_THREAD (dt);

face_t f;

Node *v;

float NV_VEC (axis), NV_VEC (snout);

float len, divisions, segment, wave, per, alpha, per_snout, omega, ampfac, rad, thick, snout_y, s,
aloc, bloc, xpos, ypos;

float theta, amp, fx, delx, conv, tempxcen, tempycen, oldslope, slope, oldnorm, norm, b, c,
xinter, yinter, aroc, foil, xpos_u, ypos_u, change, del, sign;

int n, i, npoint;

conv=1.0e-7;    /* set convergence tolerance (units are cm) */

 if (tcheck < time)

{count=0; tcheck=time;}                    /* initialize count */

/* set constants for kinematic model (use units of cm and sec) */

len=10.0;    /*fish centerline length*/

divisions=4;  /*Number of divisions along the first regular node length*/
```

```
segment=divisions*(divisions+1)/2;  /*Weighted length of divisions along first node

length (e.g. length of fourth division is  4 x segment) */

wave=20.4;   /*wavelength of centerline oscillation*/

per=0.45;    /*period of centerline oscillation*/

alpha = 2.5; /*amplitude of snout path*/

per_snout = .45; /*period of snout path oscillation*/

ampfac = -2.3;      /*amplitude factor for the y position of the fish centerline*/

thick = 0.125;     /*maximum thickness of the fish's hydrofoil as a decimal part of centerline
length*/

 /* set deforming flag on adjacent cell zone */

SET_DEFORMING_THREAD_FLAG (THREAD_T0 (tf));

Message ("\n time = %f\n", time);

Message ("tcheck = %f\n", tcheck);

 NV_D (axis, =, 0.0, 1.0, 0.0);

omega = 1/per_snout;

snout_y = 0.15 + (alpha/100) * sin(2 * M_PI * omega * time);

NV_D (snout, =, 0.10, snout_y, 0.0);

npoint = len/dist;

s = 0;

begin_f_loop (f, tf)
{
f_node_loop (f, tf, n)
{
v = F_NODE (f, tf, n);

 /* update node IF x position is greater than 0.10 and the current node has not been previously
visited when looping through previous faces */
```

```
if ( /* NODE_X (v) > 0.100 && */  NODE_POS_NEED_UPDATE (v))
{
/* indicate that node position has been updated so that it's not updated more than once */

NODE_POS_UPDATED (v);

count=count+1;

 /*Message ("NODE_X(v)= %f, NODE_Y(v)=%f\n",NODE_X(v),NODE_Y(v));*/

if(count<=divisions+1)

 rad = (count-1) * dist / segment;

else
{
rad = dist;
}

s = s + rad;

 if (count==1)              /* find (x,y) for snout at new time */
{
xpos=0.0;

theta =2.0*M_PI*((xpos/wave)-(time/per));

amp = ampfac * len*(0.351*sin((xpos/len)-1.796)+0.359);

ypos = amp * sin(theta);
}

else    /* Use Newton's method to iteratively solve for new (x,y) */
{
change = rad/2;

del = rad/4;

for (i=1; i<=200; i++)
{
xpos = aloc + change;

theta =2.0*M_PI*((xpos/wave)-(time/per));

amp = ampfac * len*(0.351*sin((xpos/len)-1.796)+0.359);
ypos = amp * sin(theta);

fx = sqrt(pow(change,2)+pow(ypos-bloc,2));

delx = fx - rad;

if ( delx < 0)
{
sign = 1;
}
else
{
sign = -1;
}
change = change + sign * del;

del = del/2;

if ( fabs(delx) < conv)
{
i=201;          /* set i = max to break out of loop once solution is converged */
}
else
{
if (i==200)

Message ("Solution didn't converge\n");
}
}
}
tempxcen = (xpos/100.0) + snout[0];

tempycen = (ypos/100.0) + snout[1];

aloc=xpos;

bloc=ypos;      /* set a and b to calculate the next node */

/*Add geometry of the fish's hydrofoil*/

foil = .01 * len * (thick/0.2) * (.2969 * sqrt(s/len) - 0.126 * (s/len) - 0.3516 * pow(s/len, 2) +
0.2843 * pow(s/len, 3) - 0.1015 * pow(s/len, 4));

if(count == 1)
{
xpos_u = tempxcen;
```

```
ypos_u = tempycen;
}
else
{
xpos_u = tempxcen + foil * cos(atan(slope) + M_PI/2);

ypos_u = tempycen + foil * sin(atan(slope) + M_PI/2);
}
NODE_X(v) = xpos_u;

NODE_Y(v) = ypos_u;
}
}
}
end_f_loop (f, tf);

Message ("\n count = %d\n", count);
}

DEFINE_GRID_MOTION(clinemvfishlower, domain, dt, time, dtime)
{
static float tcheck=0.0;

static int count2;

Thread *tf = DT_THREAD (dt);

face_t f;

Node *v;

float NV_VEC (axis), NV_VEC (snout);

float len, divisions, segment, wave, per, alpha, per_snout, omega, ampfac, rad, thick, snout_y, s,
aloc, bloc, xpos, ypos;

float theta, amp, fx, delx, conv, tempxcen, tempycen, slope, foil, xpos_u, ypos_u, change, del,
sign;

int n, i, npoint;

conv=1.0e-7;    /* set convergence tolerance (units are cm) */

if (tcheck < time)

{count2=0; tcheck=time;}                 /* initialize count */
```

```
/* set constants for kinematic model (use units of cm and sec) */

len=10.0;    /*fish centerline length*/

divisions=4;  /*Number of divisions along the first regular node length*/

segment=divisions*(divisions+1)/2;  /*Weighted length of divisions along first node length (e.g.
length of fourth division is  4 x segment) */

wave=20.4;   /*wavelength of centerline oscillation*/

per=0.45;    /*period of centerline oscillation*/

alpha = 2.5; /*amplitude of snout path*/

per_snout = .45; /*period of snout path oscillation*/

ampfac = -2.3;      /*amplitude factor for the y position of the fish centerline*/

thick = 0.125;     /*maximum thickness of the fish's hydrofoil as a decimal part of centerline
length*/

/* set deforming flag on adjacent cell zone */

SET_DEFORMING_THREAD_FLAG (THREAD_T0 (tf));

Message ("\n time = %f\n", time);
Message ("tcheck = %f\n", tcheck);

NV_D (axis, =, 0.0, 1.0, 0.0);

omega = 1/per_snout;
snout_y = 0.15 + (alpha/100) * sin(2 * M_PI * omega * time);

NV_D (snout, =, 0.10, snout_y, 0.0);

npoint = len/dist;
s = 0;

begin_f_loop (f, tf)
{
f_node_loop (f, tf, n)
{
v = F_NODE (f, tf, n);
```

```
/* update node IF x position is greater than 0.10 and the current node has not been previously
visited when looping through previous faces */

if ( /* NODE_X (v) > 0.100 && */  NODE_POS_NEED_UPDATE (v))
{
/* indicate that node position has been updated so that it's not updated more than once */

NODE_POS_UPDATED (v);

count2=count2+1;

if(count2<=divisions)
{
rad = count2 * dist / segment;
}
else
{
rad = dist;
}
s = s + rad;
if (count2==1)            /* find (x,y) for snout at new time */
{
aloc = 0;

bloc = ampfac*len*(0.351*sin((aloc/len)-1.796)+0.359)*sin(2.0*M_PI*((aloc/wave)-
(time/per)));

change = rad/2;

del = rad/4;

for (i=1; i<=200; i++)
{
xpos = aloc + change;

theta =2.0*M_PI*((xpos/wave)-(time/per));

amp = ampfac * len*(0.351*sin((xpos/len)-1.796)+0.359);

ypos = amp * sin(theta);

fx = sqrt(pow(change,2)+pow(ypos-bloc,2));

delx = fx - rad;

if ( delx < 0)
```

```
{
sign = 1;
}
else
{
sign = -1;
}
change = change + sign * del;

del = del/2;

if ( fabs(delx) < conv)

 i=201;            /* set i = max to break out of loop once solution is converged */
else
{
if (i==200)

Message ("Solution didn't converge\n");
}
}
}
else    /* Use Newton's method to iteratively solve for new (x,y) */
{
change = rad/2;

del = rad/4;

for (i=1; i<=200; i++)
{
xpos = aloc + change;

theta =2.0*M_PI*((xpos/wave)-(time/per));

amp = ampfac * len*(0.351*sin((xpos/len)-1.796)+0.359);

ypos = amp * sin(theta);

fx = sqrt(pow(change,2)+pow(ypos-bloc,2));

delx = fx - rad;
 if ( delx < 0)
{
sign = 1;
}
else
```

```
{
sign = -1;
}

change = change + sign * del;

del = del/2;

if ( fabs(delx) < conv)

i=201;          /* set i = max to break out of loop once solution is converged */

else
{
if (i==200)

Message ("Solution didn't converge\n");
}
}
}
tempxcen = (xpos/100.0) + snout[0];

tempycen = (ypos/100.0) + snout[1];

slope = (ypos - bloc) / (xpos - aloc);

Message ("xpos = %e, aloc = %e\n", xpos, aloc);

aloc=xpos;

bloc=ypos;      /* set a and b to calculate the next node */

foil = .01 * len * (thick/0.2) * (.2969 * sqrt(s/len) - 0.126 * (s/len) - 0.3516 * pow(s/len, 2) +
0.2843 * pow(s/len, 3) - 0.1015 * pow(s/len, 4));

xpos_u = tempxcen - foil * cos(atan(slope) + M_PI/2);

ypos_u = tempycen - foil * sin(atan(slope) + M_PI/2);

NODE_X(v) = xpos_u;

NODE_Y(v) = ypos_u;
}
}
}
end_f_loop (f, tf);
```

```
Message ("\n count2 = %d\n", count2);
}
```

Appendix C. UDF for Centerline Motion (Von Karman Vortex Street Inlet)

```
#include "/fsc/drc53/GradApps/udf.h"

static float dist=0.05;  /*IMPORTANT TO ADJUST rad(cm) DEPENDING ON INTERVAL

 LENGTH */

DEFINE_GRID_MOTION(clinemvfishupper, domain, dt, time, dtime)

{

static float tcheck=0.0;

static int count;

Thread *tf = DT_THREAD (dt);

face_t f;

Node *v;

float NV_VEC (axis), NV_VEC (snout);

float len, wave, per, alpha, per_snout, omega, ampfac, rad, thick, snout_y, aloc, bloc, xpos,

ypos;

float theta, amp, fx, delx, conv, tempxcen, tempycen, change, del, sign;

int n, i, npoint;

conv=1.0e-7;    /* set convergence tolerance (units are cm) */

if (tcheck < time)

{count=0; tcheck=time;}                    /* initialize count */

/* set constants for kinematic model (use units of cm and sec) */
```

```
len=10.0;    /*fish centerline length*/

wave=20.4;   /*wavelength of centerline oscillation*/

per=0.45;    /*period of centerline oscillation*/

alpha = 0.5; /*amplitude of snout path*/

per_snout = .45; /*period of snout path oscillation*/

ampfac = -2.3;      /*amplitude factor for the y position of the fish centerline*/

thick = 0.12;     /*maximum thickness of the fish's hydrofoil as a decimal part of centerline

length*/

/* set deforming flag on adjacent cell zone */

SET_DEFORMING_THREAD_FLAG (THREAD_T0 (tf));

Message ("\n time = %f\n", time);

Message ("tcheck = %f\n", tcheck);

NV_D (axis, =, 0.0, 1.0, 0.0);

omega = 1/per_snout;

snout_y = 0.15 + (alpha/100) * sin(2 * M_PI * omega * time);

NV_D (snout, =, 0.10, snout_y, 0.0);

npoint = len/dist;

begin_f_loop (f, tf)

{

f_node_loop (f, tf, n)

{

v = F_NODE (f, tf, n);

/* update node IF x position is greater than 0.10 and the current node has not been previously
```

```
visited when looping through previous faces */

if ( /* NODE_X (v) > 0.100 && */  NODE_POS_NEED_UPDATE (v))

{

/* indicate that node position has been updated so that it's not updated more than once */

NODE_POS_UPDATED (v);

count=count+1;

/*Message ("NODE_X(v)= %f, NODE_Y(v)=%f\n",NODE_X(v),NODE_Y(v));*/

rad = dist;

if (count==1)               /* find (x,y) for snout at new time */

{

xpos=0.0;

theta =2.0*M_PI*((xpos/wave)-(time/per));

/*Message ("i = %d\n", i);*/

amp = ampfac * len*(0.351*sin((xpos/len)-1.796)+0.359);

ypos = amp * sin(theta);

}

else    /* Use Newton's method to iteratively solve for new (x,y) */

{

change = rad/2;

del = rad/4;

for (i=1; i<=200; i++)

{  xpos = aloc + change;

theta =2.0*M_PI*((xpos/wave)-(time/per));
/*Message ("i = %d\n", i);*/
amp = ampfac * len*(0.351*sin((xpos/len)-1.796)+0.359);
ypos = amp * sin(theta);
fx = sqrt(pow(change,2)+pow(ypos-bloc,2));
delx = fx - rad;
if ( delx < 0)
{
sign = 1;
}
else {
sign = -1;
}
change = change + sign * del;
del = del/2;
if ( fabs(delx) < conv)
{
i=201;          /* set i = max to break out of loop once solution is converged */
}
else
{if (i==200)
Message ("Solution didn't converge\n");
}
}
```

```
}
tempxcen = (xpos/100.0) + snout[0];
tempycen = (ypos/100.0) + snout[1];
Message ("node length = %f\n", rad);*/
aloc=xpos;
bloc=ypos;      /* set a and b to calculate the next node */
/*Add geometry of the fish's hydrofoil*/
NODE_X(v) = tempxcen;
NODE_Y(v) = tempycen + .0001;
/*Message ("node_x = %e, node_y = %e, count = %d, foil = %e\n", NODE_X(v), NODE_Y(v),
 count, foil);*/

}
}
}
end_f_loop (f, tf);
/*Message ("\n count = %d\n", count);*/
}
DEFINE_GRID_MOTION(clinemvfishlower, domain, dt, time, dtime)
{
static float tcheck=0.0;
static int count2;
Thread *tf = DT_THREAD (dt);
```

```
face_t f;
Node *v;
float NV_VEC (axis), NV_VEC (snout);
float len, divisions, segment, wave, per, alpha, per_snout, omega, ampfac, rad, thick, snout_y, s,
aloc, bloc, xpos, ypos;
 float theta, amp, fx, delx, conv, tempxcen, tempycen, change, del, sign;
 int n, i, npoint;
 conv=1.0e-7;    /* set convergence tolerance (units are cm) */
 if (tcheck < time)
 {count2=0; tcheck=time;}                 /* initialize count */
 /* set constants for kinematic model (use units of cm and sec) */
  len=10.0;    /*fish centerline length*/
wave=20.4;   /*wavelength of centerline oscillation*/
per=0.45;    /*period of centerline oscillation*/
alpha = 0.5; /*amplitude of snout path*/
per_snout = .45; /*period of snout path oscillation*/
ampfac = -2.3;      /*amplitude factor for the y position of the fish centerline*/
thick = 0.125;     /*maximum thickness of the fish's hydrofoil as a decimal part of centerline
length*/
/* set deforming flag on adjacent cell zone */
SET_DEFORMING_THREAD_FLAG (THREAD_T0 (tf));
Message ("\n time = %f\n", time);
Message ("tcheck = %f\n", tcheck);
```

```
NV_D (axis, =, 0.0, 1.0, 0.0);

omega = 1/per_snout;

snout_y = 0.15 + (alpha/100) * sin(2 * M_PI * omega * time);

NV_D (snout, =, 0.10, snout_y, 0.0);

npoint = len/dist;

begin_f_loop (f, tf)

{

f_node_loop (f, tf, n)

{

v = F_NODE (f, tf, n);

/* update node IF x position is greater than 0.10 and the current node has not been previously

visited when looping through previous faces */

 if ( /* NODE_X (v) > 0.100 && */  NODE_POS_NEED_UPDATE (v))

 {

/* indicate that node position has been updated so that it's not updated more than once */

NODE_POS_UPDATED (v);

count2=count2+1;

/*Message ("NODE_X(v)= %f, NODE_Y(v)=%f\n",NODE_X(v),NODE_Y(v));*/

rad = dist;

if (count2==1)            /* find (x,y) for snout at new time */

{

aloc = 0;

bloc = ampfac*len*(0.351*sin((aloc/len)-1.796)+0.359)*sin(2.0*M_PI*((aloc/wave)-
(time/per)));

change = rad/2;

del = rad/4;

for (i=1; i<=200; i++)

{  xpos = aloc + change;

theta =2.0*M_PI*((xpos/wave)-(time/per));

amp = ampfac * len*(0.351*sin((xpos/len)-1.796)+0.359);

ypos = amp * sin(theta);

fx = sqrt(pow(change,2)+pow(ypos-bloc,2));

delx = fx - rad;

if ( delx < 0)

{

sign = 1;

}

else

{

sign = -1;

}

change = change + sign * del;

del = del/2;

if ( fabs(delx) < conv)

i=201;             /* set i = max to break out of loop once solution is converged */

else
```

```
{if (i==200)

Message ("Solution didn't converge\n");

}

}

}

else    /* Use Newton's method to iteratively solve for new (x,y) */

{          change = rad/2;

del = rad/4;

for (i=1; i<=200; i++)

{  xpos = aloc + change;

theta =2.0*M_PI*((xpos/wave)-(time/per));

amp = ampfac * len*(0.351*sin((xpos/len)-1.796)+0.359);

ypos = amp * sin(theta);

fx = sqrt(pow(change,2)+pow(ypos-bloc,2));

delx = fx - rad;

if ( delx < 0)

{

sign = 1;

}

else {

sign = -1;

}

change = change + sign * del;
```

```
del = del/2;

if ( fabs(delx) < conv)

i=201;          /* set i = max to break out of loop once solution is converged */

else

{if (i==200)

Message ("Solution didn't converge\n");

}

}

}

tempxcen = (xpos/100.0) + snout[0];

tempycen = (ypos/100.0) + snout[1];

/*Message ("ypos = %e, bloc = %e\n", ypos, bloc);*/

/*Message ("xpos = %e, aloc = %e\n", xpos, aloc);*/

aloc=xpos;

bloc=ypos;      /* set a and b to calculate the next node */

NODE_X(v) = tempxcen;

NODE_Y(v) = tempycen-.0001;

/*Message ("node_x = %e, node_y = %e, count2= %d, foil= %e\n", NODE_X(v), NODE_Y(v),

count2,foil);*/


}

}

}
```

```
end_f_loop (f, tf);

Message ("\n count2 = %d\n", count2);
```

Appendix D. UDF for Von Karman Vortex Street Inlet

```
#include "/fsc/drc53/GradApps/Segmented/udf.h"

DEFINE_PROFILE(inlet_x_velocity, q, i)
{
float x[ND_ND];

Thread *t;

face_t f;

float R, Vort_max, Vmax, Vfs, tim, yin, xin, vxsum, vysum, ycen, dx, dy,
r, Vtan, xcomp, ycomp, vx, vy, y[1][2], xcent[8][2], xvec, yvec;

int a, b, nvor, nshed;

R = 0.02;  /*radius of vortex core (definition based on Rankine vortex) in meters*/

Vort_max = 15; /*radians per second at radius R*/

Vmax = R * Vort_max; /*maximum core vortex tangential velocity in m/s*/

Vfs = .45; /*free-stream velocity of the flow in m/s*/

nvor = 8; /*number of vortex shed cycles to be simulated*/

nshed = 1; /*number of locations from which vortices are shed alternately*/

y[0][0] = .175; /*location 1 of alternately shed vortex centroids along the y-axis*/

y[0][1] = .125;  /*location 2 of alternately shed vortex centroids along the y-axis*/

for(a = 0; a < nvor; a++)
                {
xcent[a][0] = .204*a - .102;

xcent[a][1] = .204*a;
}
```

```
tim = RP_Get_Real("flow-time");

xin = Vfs * tim;

begin_f_loop(f,q)
{
F_CENTROID(x,f,q);

yin = x[1];

vxsum = 0;

for(a = 0; a < nvor; a++)
{
for(b = 0; b < nshed; b++)
{
/*This portion will calculate the contribution to vorticity at the inlet made by upper vortices*/

ycen = y[b][0];

dx = xcent[a][0] - xin;

dy = yin - ycen;

r = pow((pow(dx,2) + pow(dy,2)),.5);

if(r <= R)

Vtan = Vmax*r/R;

else
{
Vtan = Vmax*R/r;
}
xvec = atan2(dy,dx);

xcomp = cos(xvec - M_PI/2);

vx = Vtan * xcomp;

vxsum = vxsum + vx;

/*This portion will calculate the contribution to vorticity at the inlet made by lower vortices*/

ycen = y[b][1];
dx = xcent[a][1] - xin;

dy = yin - ycen;

r = pow((pow(dx,2) + pow(dy,2)),.5);

if(r <= R)

Vtan = Vmax*r/R;

else
{
Vtan = Vmax*R/r;
}

xvec = atan2(dy,dx);

xcomp = cos(xvec + M_PI/2);

vx = Vtan * xcomp;

vxsum = vxsum + vx;
}
}
F_PROFILE(f, q, i) = vxsum + Vfs;
}
end_f_loop(f, q)
}
DEFINE_PROFILE(inlet_y_velocity, q, i)
{
float x[ND_ND];

Thread *t;

face_t f;

float R, Vort_max, Vmax, Vfs, tim, yin, xin, vxsum, vysum, ycen, dx, dy,
r, Vtan, xcomp, ycomp, vx, vy, y[1][2], xcent[8][2], xvec, yvec;

int a, b, nvor, nshed;

R = 0.02;  /*radius of vortex core (definition based on Rankine vortex) in meters*/

Vort_max = 15; /*radians per second at radius R*/

Vmax = R * Vort_max; /*maximum core vortex tangential velocity in m/s*/
```

```
Vfs = .45; /*free-stream velocity of the flow in m/s*/

nvor = 8; /*number of vortex shed cycles to be simulated*/
nshed = 1; /*number of locations from which vortices are shed alternately*/

y[0][0] = .175;  /*location 1 of alternately shed vortex centroids along the y-axis*/

y[0][1] = .125;  /*location 2 of alternately shed vortex centroids along the y-axis*/

for(a = 0; a < nvor; a++)
{
xcent[a][0] = .204*a - .102;

xcent[a][1] = .204*a;
}
tim = RP_Get_Real("flow-time");

xin = Vfs * tim;

begin_f_loop(f,q)
{
F_CENTROID(x,f,q);

yin = x[1];

vysum = 0;

for(a = 0; a < nvor; a++)
{
for(b = 0; b < nshed; b++)
{
/*This portion will calculate the contribution to vorticity at the inlet made by upper vortices*/

ycen = y[b][0];

dx = xcent[a][0] - xin;

dy = yin - ycen;

r = pow((pow(dx,2) + pow(dy,2)),.5);

if(r <= R)

Vtan = Vmax*r/R;
else
{
Vtan = Vmax*R/r;
}
yvec = atan2(dy,dx);

ycomp = sin(yvec - M_PI/2);

vy = Vtan * ycomp;

vysum = vysum + vy;

/*This portion will calculate the contribution to vorticity at the inlet made by lower vortices*/

ycen = y[b][1];

dx = xcent[a][1] - xin;

dy = yin - ycen;

r = pow((pow(dx,2) + pow(dy,2)),.5);

if(r <= R)

Vtan = Vmax*r/R;

else
{
Vtan = Vmax*R/r;
}

yvec = atan2(dy,dx);

ycomp = sin(yvec + M_PI/2);

vy = Vtan * ycomp;

vysum = vysum + vy;
}
}
F_PROFILE(f, q, i) = vysum;
}
end_f_loop(f, q)
}
```

Appendix E. CFD Studies.

	Dimensions (Additional Comments)	Organism	Re	St	Cd	Ct/Cf	Cp	Fr
Flanagan (2004)	2D (Laminar, Carangiform)	flat plate with rainbow trout kinematics	4.5e4	0.32	.00108 *	Ct = .00113 *	-	0.62
Miao (2006)	2D (One example)	flapping airfoil	1e4	-	-	Ct~ 0.2	~0.7	0.35
Zhang (2008)	3D	Tuna	7.1e5	0.41	-	Ct = 0.025	-	-
Borzajani (2008)	3D (Carangiform)	Mackerel	3e2	1.05	-	Cf = 0	-	.189
Borzajani (2008)	3D (Carangiform)	Mackerel	4e3	0.55	-	Cf = 0	-	.230
Borzajani (2009)	3D (Anguilliform)	Lamprey	3e2	1.25	-	Cf = 0	-	.176
Borzajani (2009)	3D (Anguilliform)	Lamprey	4e3	0.55	-	Cf = 0	-	.316
Borzajani (2009)	3D (Anguilliform)	Lamprey	∞	0.4	-	Cf = 0	-	.189
Present Study (2011)*	2D (Laminar, Carangiform)	hydrofoil with rainbow trout kinematics	4.5e4	0.32	.00145	Ct = .00188	.00117	0.7
Present Study (2011)*	2d (LES, Carangiform)	hydrofoil with rainbow trout kinematics	4.5e4	0.32	.00154	Ct = .00201	.00121	0.7

*Assumes average height of body is 1.8 cm.

Appendix F. Grid Convergence Calculations with Input from Current Study.

Richardson Extrapolation (MatchCAD File Created Originally by Patrick Flanagan. I have added values for vorticity and vortex locations from simulations included in this study.)

Richardson extrapolation is performed to estimate grid convergence, and these calculations are based on methods from Celik & Karatekin (1997). The extrapolation is performed for the 2D fish simulations using data from the 200, 160, and 120 face meshes.

$$F - F_1 = C(a_1 \cdot h)^n$$

$$F - F_2 = C(a_2 \cdot h)^n$$

$$F - F_3 = C(a_3 \cdot h)^n$$

These are the theoretical equations that this extrapolation scheme solves for. There are 3 unknowns:

F = the exact solution for zero grid size

C = the coefficient

n = the apparent order of the method

Characteristic grid size data:

$$a_1 := \frac{200}{200} \qquad a_1 = 1$$

$$a_2 := \frac{200}{160} \qquad a_2 = 1.25$$

$$a_3 := \frac{200}{120} \qquad a_3 = 1.667$$

Grid convergence of the parameter: Peak vorticity of the 2nd vortex downstream of trout

$$Data_{pv} := \begin{pmatrix} 132.583 \\ 129.099 \\ 114.041 \end{pmatrix}$$

$$Fpv_1 := Data_{pv_0}$$

$$Fpv_2 := Data_{pv_1}$$

$$Fpv_3 := Data_{pv_2}$$

Celik and Karatekin's solution formulation:

Guess Values of n and f(n) $\quad n_{pv} := 1 \quad fn_{pv} := 2$

Given

$$n_{pv} = \left| \frac{\ln\left(\frac{Fpv_2 - Fpv_3}{Fpv_1 - Fpv_2}\right)}{\ln(a_2)} - fn_{pv} \right| \qquad fn_{pv} = \frac{\ln\left[\frac{\left(\frac{a_3}{a_2}\right)^{n_{pv}} - 1}{a_2^{\,n_{pv}} - 1}\right]}{\ln(a_2)}$$

$$\begin{pmatrix} n \\ f_n \end{pmatrix} := \mathrm{Find}(n_{pv}, fn_{pv}) \qquad n = 4.624 \qquad f_n = 1.936$$

$$F_{ext} := \frac{a_2^{\,n} \cdot Fpv_1 - Fpv_2}{a_2^{\,n} - 1} \qquad F_{ext} = 134.512$$

Flanagan's solution formulation:

Guess Values: $F := 121$ $n_{pv} := 1$

Given

$$F = \frac{Fpv_1 - \left(\frac{a_1}{a_2}\right)^{n_{pv}} \cdot Fpv_2}{1 - \left(\frac{a_1}{a_2}\right)^{n_{pv}}} \qquad F = \frac{Fpv_2 - \left(\frac{a_2}{a_3}\right)^{n_{pv}} Fpv_3}{1 - \left(\frac{a_2}{a_3}\right)^{n_{pv}}}$$

$$\begin{pmatrix} F_{ext} \\ n \end{pmatrix} := \mathrm{Find}(F, n_{pv}) \qquad n = 4.624 \qquad F_{ext} = 134.512$$

Flanagan's solution formulation gives the same results as Celik's.

Now, calculate the actual error percentage for each grid size:

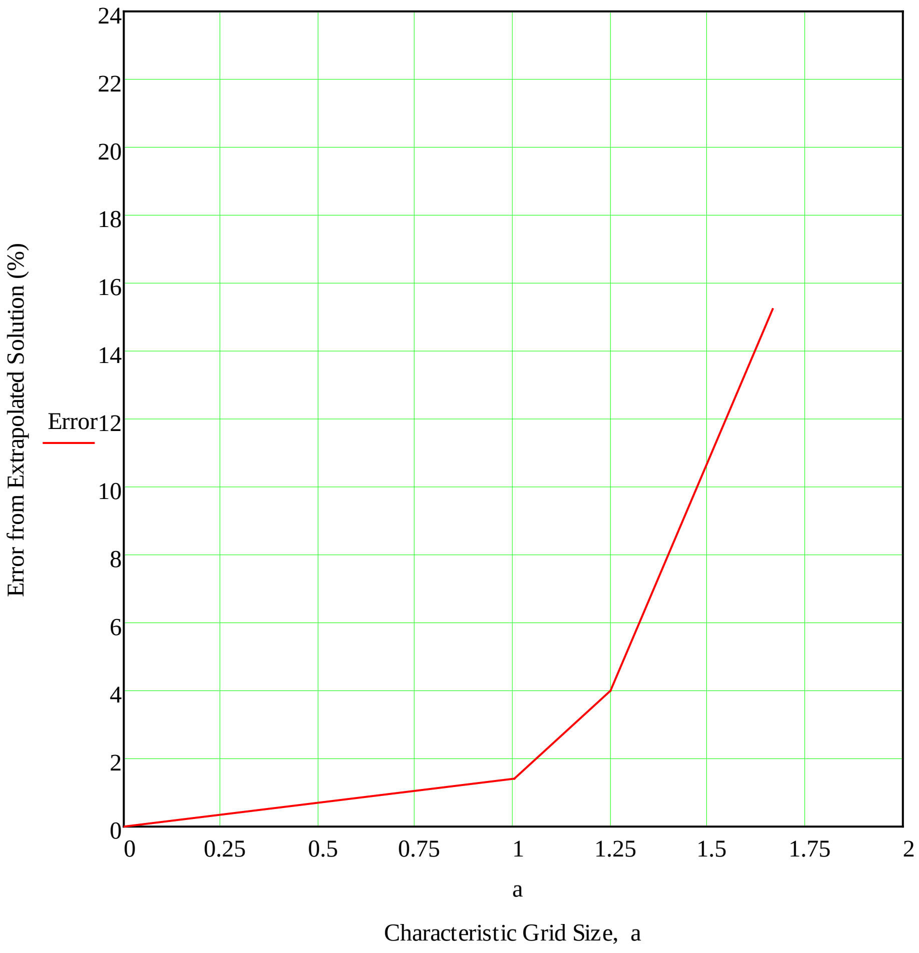
$$\mathrm{Error} := \begin{pmatrix} 0 \\ F_{ext} - Fpv_1 \\ F_{ext} - Fpv_2 \\ F_{ext} - Fpv_3 \end{pmatrix} \cdot \frac{100}{F_{ext}} \qquad a := \begin{pmatrix} 0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

$$\text{Error} = \begin{pmatrix} 0 \\ 1.434 \\ 4.024 \\ 15.219 \end{pmatrix}$$

This is the error percentages from the exact extrapolated solution.

Therefore, the error for the finest mesh (200 faces on trout) is only 1.43%.

Grid Convergence for Peak Vorticity of Developed Vortex



Grid convergence of the parameter: Location of the 8th vortex downstream of trout

$$\mathrm{Data}_x := \begin{pmatrix} 45.9559 \\ 46.0201 \\ 46.0459 \end{pmatrix}$$

$$Fx_1 := \mathrm{Data}_{x_0}$$
$$Fx_2 := \mathrm{Data}_{x_1}$$
$$Fx_3 := \mathrm{Data}_{x_2}$$

Using Flanagan’s solution formulation:

Guess Values:

$F := 2$ $\qquad n_x := 20$

Given

$$F = \frac{Fx_1 - \left(\frac{a_1}{a_2}\right)^{n_x} \cdot Fx_2}{1 - \left(\frac{a_1}{a_2}\right)^{n_x}} \qquad F = \frac{Fx_2 - \left(\frac{a_2}{a_3}\right)^{n_x} Fx_3}{1 - \left(\frac{a_2}{a_3}\right)^{n_x}}$$

$$\begin{pmatrix} F_{ext} \\ n \end{pmatrix} := \mathrm{Find}(F, n_x) \qquad n = -4.679 \qquad F_{ext} = 46.055$$

Now, calculate the actual error percentage for each grid size:

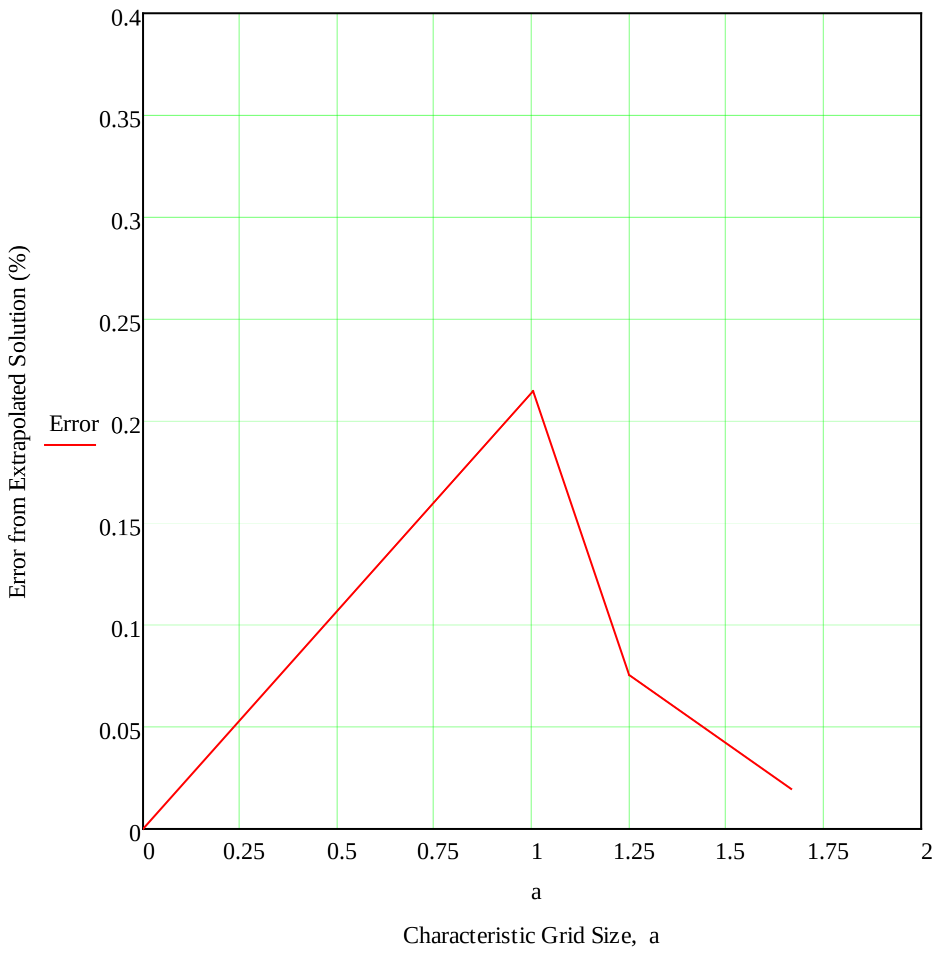
$$\mathrm{Error} := \begin{pmatrix} 0 \\ \left|F_{ext} - Fx_1\right| \\ \left|F_{ext} - Fx_2\right| \\ \left|F_{ext} - Fx_3\right| \end{pmatrix} \cdot \frac{100}{F_{ext}} \qquad a := \begin{pmatrix} 0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

$$\text{Error} = \begin{pmatrix} 0 \\ 0.215 \\ 0.076 \\ 0.02 \end{pmatrix}$$

This is the error percentages from the exact extrapolated solution.

Therefore, the error for the finest mesh (200 faces on trout) is only 0.02%.

Grid Convergence for Peak Vorticity of Developed Vortex


Error from Extrapolated Solution (%)
Error
a
Characteristic Grid Size, a