Hydrodynamics of Dolphin Caudal Fins

Eric Zacharia[1], Melissa Green[2]
Syracuse University, Syracuse, NY, 13244

Experiments were conducted on a panel shaped like a dolphin caudal fin to study the wake structure behind its pitching motion. The experiments vary in flow velocity, pitching amplitude, and span location along the caudal fin, which vary the Strouhal and Reynolds numbers. The flow behind the pitching panel is composed of a reverse von Kármán vortex street and a converging wake structure. Wakes with well-developed vortices had Strouhal numbers within and around the 0.2 - 0.4 range. This supports the results found by Triantafyllou, Triantafyllou, and Grosenbaugh, when they claimed that values within this range correspond to optimal propulsive efficiencies for oscillating foils. Additionally, at higher Strouhal numbers, the wake breaks down further upstream, which supports results found by Green, Rowley, and Smits.

Keywords: dolphins, vortex dynamics, Strouhal number, flow visualization.

NOMENCLATURE

t	time, s
f	pitching frequency, Hz
L	pitching amplitude, m
v	flow velocity, m/s
ρ	water density at room temperature, kg/m^3
μ	water viscosity at room temperature, Pa
c	averaged chord length, m
St	Strouhal number
Re	Reynolds number

[1]Undergraduate Research Student, Department of Aerospace and Mechanical Engineering, Syracuse University, Syracuse, NY, 13244. Member of AIAA.
[2]Assistant Professor, Department of Aerospace and Mechanical Engineering, Syracuse University, Syracuse, NY, 13244. Member of AIAA.

I. Introduction

Hydrodynamics is a branch of physics dealing with the motion of fluids and the forces resulting on bodies immersed in the fluid due to their motion relative to the fluid. The inspiration behind this research comes from observing the fascinating locomotion of aquatic mammals in their ability to propel themselves through water powerfully and efficiently. These animals propel themselves through water by borrowing and kicking their fins. A dolphin is one of the ocean's great examples that is of particular interest from the point of view of getting bio-inspiration in order to improve propulsive efficiency of manmade marine applications. The dolphin has a very powerful tail, or caudal fin, that acts a wing in the sense that it has the ability to create a lift force that propels the dolphin forwards after each stroke. For the purpose of relating the propulsive ability of aquatic mammals to bio-inspired marine applications, the swimming efficiency can be quantified using the Strouhal number. Previous experiments have been conducted to show that most fish and cetaceans achieve optimal propulsion within a Strouhal number range between 0.2 and 0.4 (Triantafyllou, Triantafyllou, and Grosenbaugh 1993). Before the experiments, this result was predicted to be between 0.25 and 0.35. Another study, conducted on odontocete cetaceans, showed that the greatest number of Strouhal values, out of a total of 248 calculated Strouhal numbers, was found between 0.225 and 0.275 (44%) (Rohr and Fish 2004).

The pure pitching motion of a dolphin's caudal fin creates a wake that can be analyzed for the purpose of understanding thrust generation. With the panel immersed in a flowing water channel, the pitching panel creates a reverse von Kármán vortex street. A regular von Kármán vortex street generates drag, as the resultant flow of the vortices points in the opposite direction as the freestream flow. The pitching panel reverses the direction of these vortices by pitching back and forth, which generates thrust. Thrust generation is dependent on Reynolds number, Strouhal number. The Strouhal number is defined as the frequency of the vortex shedding, multiplied by the panel's chord length, and divided by the velocity of the flow as seen by the pitching panel. The Reynolds number signifies the relative strength of inertial forces and viscous forces and is defined as the density of the fluid, multiplied by the velocity of the flow as seen by the pitching panel, multiplied by the panel's chord length, and divided by the dynamic viscosity of the fluid. The vortex breakdown is another prominent characteristic of the wake; all structure eventually becomes chaotic, with the distance downstream depending on Strouhal number. Both Reynolds and Strouhal numbers are dimensionless and are displayed mathematically below.

$$Re = \frac{\rho v L}{\mu} \qquad \textbf{\textit{Eq. 1}}$$

$$St = \frac{f L}{v} \qquad \textbf{\textit{Eq. 2}}$$

Previous work has been completed with pitching panels, both rigid and flexible. Experimental results on the role of flexibility on aquatic propulsion showed that flexible pitching panels amplified thrust production by 100-200% and propulsive efficiency by 100% when compared to rigid pitching panels (Dewey, Boschitsch, Moored, Stone, and Smits 2013). Experiments were performed for Strouhal numbers between 0.17 and 0.56 for a rigid trapezoidal shaped panel, followed by a Lagrangian coherent structure (LCS) analysis to investigate the formation of the wake (Green, Rowley, and Smits 2011). For this rigid panel, with higher Strouhal numbers, the complexity of the wake increased downstream of the trailing edge, and the vortices lost coherence as the wake compressed and split. This analysis was done using Finite-time Lyapunov exponent (FTLE). In current work, more experiments have been performed using stereoscopic particle image velocimetry (PIV) to characterize the wake produced by a rigid trapezoidal pitching panel, which support that the location of wake breakdown is dependent on Strouhal number (King and Green, publication pending). It would be of engineering value to study the wake breakdown of a pitching panel that more closely resembles a cetacean caudal fin and compare the results to that of the trapezoidal shaped panel.

This report presents results from a study conducted on realistically shaped, bio-inspired dolphin caudal fins. The study serves as a supplement to studies conducted by Dr. Melissa Green and her group on trapezoidal shaped panels that approximately mimicked the caudal fins of fish and other aquatic mammals. This reports discusses the effects of individually varying Reynolds number and Strouhal number on the panel's wake structure. The results are visually analyzed using pictures taken directly from experiments, and the varied scenarios are compared. The results found by Dr. Melissa Green and her research group are referenced throughout the report.

II. Experimental Setup

Experiments were carried out in the flow visualization water channel located in Link Hall 280. It has a test section size of 6x6x18 inches and has the ability to control the freestream flow velocity of the water with a keypad. A picture of the water channel can be seen in the figure below.

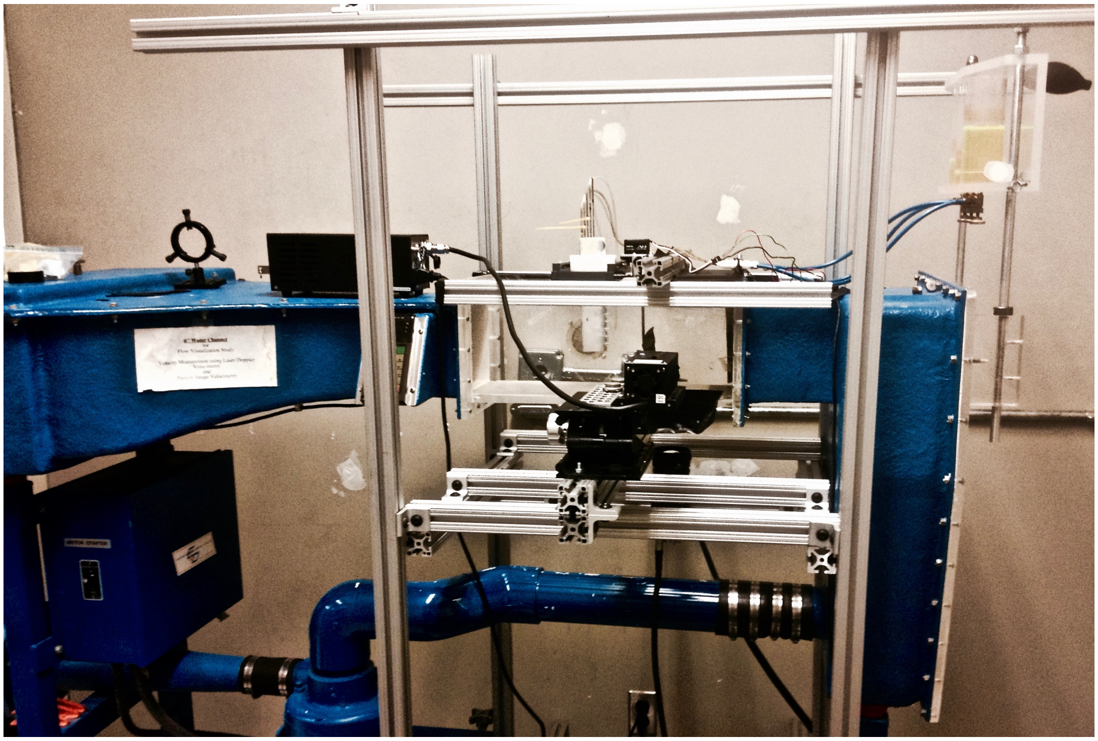


Figure 1 Flow visualization water channel.

The experimental setup includes a laser that emits a blue laser sheet through the test section of the water channel, which can be set at various heights using a mechanical lift, an Arduino Uno board that connects, via USB, to a computer with a working Arduino software code, a pitching RC servo motor that can be controlled using Arduino, a 62-millimeter long test piece that replicates the geometry of a dolphin's caudal fin, which can be easily attached and removed from the RC motor, a fluorescent dye that flows from a separated reservoir, through pump and valve controlled tubing, and out through an airfoil shaped nozzle so that it may be steadily induced into the flow upstream from the test section, a LDT MotionXtra NX8 compact high-speed camera that is mounted below the transparent test section and is able to record the flow at 30-30,000 frames per second, a structural support consisting of steel beams, brackets, and bolts, which can hold all of this equipment in a steady and level position, and a number of computer software programs including: Arduino, MotionInspector, MotionViewer, SolidWorks, MATLAB, Excel, QuickTime Player, and iMovie. The schematic in Figure 2 displays a labeled representation of the experimental setup.

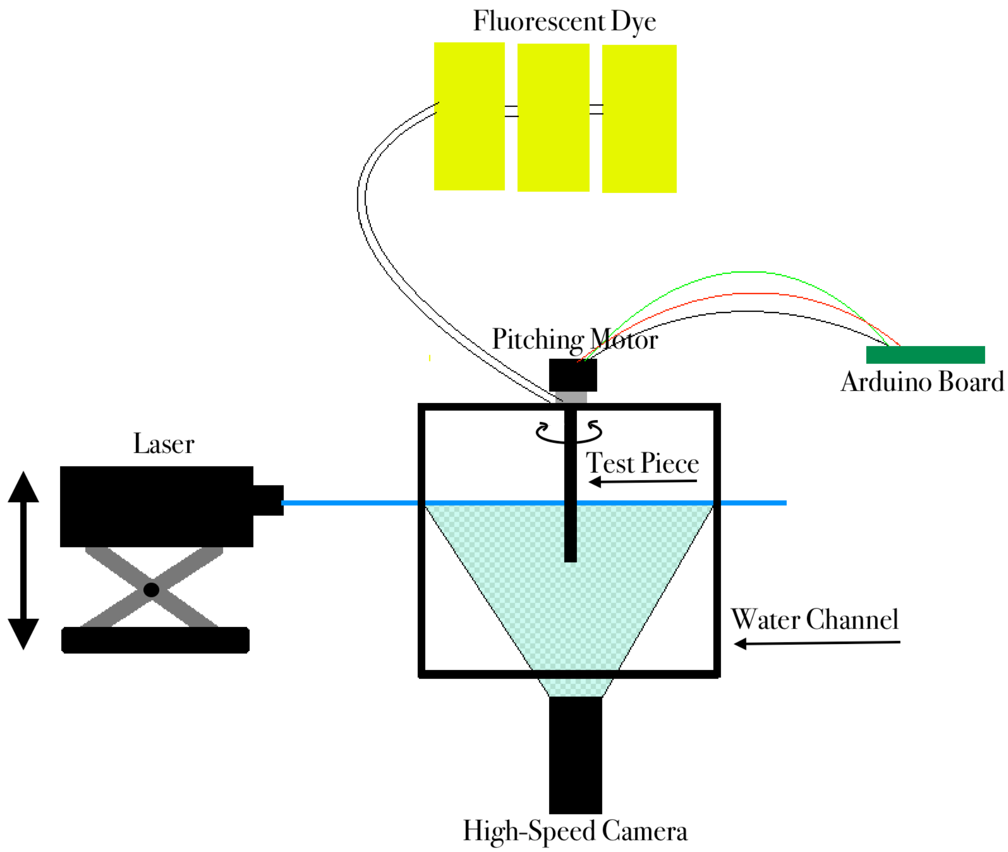


Figure 2 Experimental setup.

III. Model Design and Fabrication

The fabrication of an accurate caudal fin test piece required the implementation of a few processes. First, a two-dimensional shape was created on MATLAB with help from researchers at Lehigh University (Ayancik, Fatma, and Moored). The functions and scripts used for generating the shape can be found in the appendix section of this report. The chord distribution and mid-chord line were approximated using fourth and second-order polynomials, respectively. Next, the shape was replicated on SolidWorks, and a thickness profile of a NACA 0012 was applied. This thickness profile was chosen to keep a consistent trend with previous experiments (Dewey, Boschitsch, Moored, Stone, and Smits 2013). A rod designed with screw holes was then attached to the caudal fin shape so that it could easily be attached to, and removed from, the RC motor. The test pieces were then manufactured using a MakerBot 3D printer.

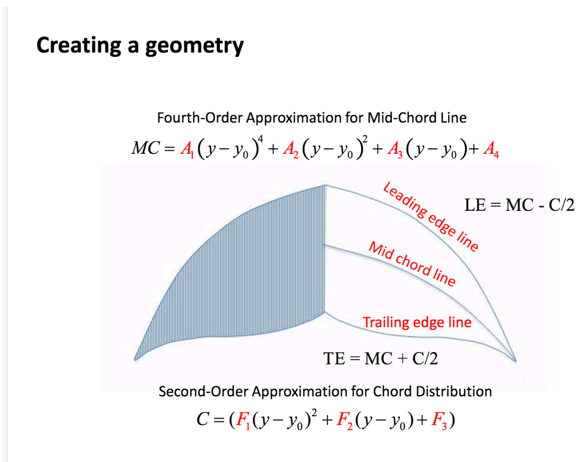


Figure 3 Creating the geometry. (Ayancik, Fatma, and Moored)

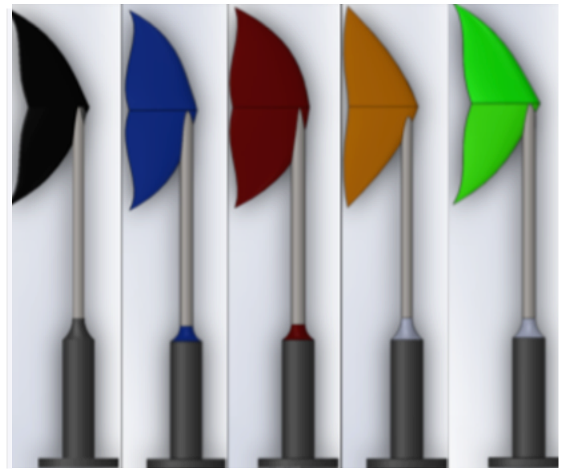


Figure 4 Test pieces shown on SolidWorks.

VI. Procedure

Data was collected from experiments using a technique called flow visualization by planar laser induced fluorescence (PLIF). Each experiment was performed by recording the flow visualization at seven different planar locations along the lower half span of the test piece. The experiments were run at different Strouhal and Reynolds numbers by changing three parameters, where each parameter was changed one at a time. First, all parameters were fixed, while changing only the planar location along half of the span of the test piece. Next, all parameters were fixed, while changing only the flow speed between each trial. Finally, all parameters were fixed, while changing only the amplitude of the tip of the pitching panel's sweep. The plan for collecting data can be seen in the table below, which represents a three dimensional matrix. The matrix includes 9 (sweep angles) x 7 (planar locations) x 5 (channel frequencies), for a total of 315 different experiments.

Table 1: 5 Channel Frequencies: 3-9 Hertz (steps of 1.5)	**1. 70°**	**2. 60°**	**3. 50°**	**4. 40°**	**5. 30°**	**6. 20°**	**7. 15°**	**8. 10°**	**9. 5°**
1. 0% (Root Chord)	✓	✓	✓	✓	✓	✓	✓	✓	✓
2. 20%	✓	✓	✓	✓	✓	✓	✓	✓	✓
3. 40%	✓	✓	✓	✓	✓	✓	✓	✓	✓
4. 60%	✓	✓	✓	✓	✓	✓	✓	✓	✓
5. 80%	✓	✓	✓	✓	✓	✓	✓	✓	✓
6. 100% (Tip)	✓	✓	✓	✓	✓	✓	✓	✓	✓
7. 120%	✓	✓	✓	✓	✓	✓	✓	✓	✓

The planar laser induced fluorescence data acquisition technique, is executed in the following way. The high-speed camera, set up below the test section of the water channel, takes multiple pictures per second of the flow. The major points of interest in the flow are the vortices that are being shed from the trailing edge of the pitching test piece. The fluorescent dye outlines the vortices in the flow, which are highlighted by the planar shaped laser. This creates a clear visual of the two dimensional profiles of the vortices for the high-speed camera to record. The view from the camera is recorded using the MotionInspector software and then saved on the MotionViewer software as multiple images. These images are then installed into MATLAB so that two-dimensional visualizations can be created by plotting these images in a "for loop." The visualizations that are run on MATLAB are then recorded using the QuickTime Player screen recorder. The related recordings are then pieced together in iMovie to create videos that display the similarities and differences between the varied experiments.

VII. Results

The experimental data takes its form as multiple photographs. Photographs with similar parameters are compared to see the effect of Reynolds number and Strouhal number on the wake's structure.

Experiments that vary only in Reynolds number do not appear to be incredibly different. Figure 5 displays three trials that differ greatly by Reynolds number, and slightly by Strouhal number. No two trials shared the same Strouhal number, so trials with similar Strouhal values were compared. All three trials in the figure display the flow around a panel at the 40% location along half of the span. The top photograph in the figure displays the panel pitching at an angle of 20 degrees and water flowing at a water tunnel frequency of 6 Hertz, the middle photograph displays the panel pitching at an angle of 15 degrees and water flowing at a water tunnel frequency of 4.5 Hertz, and the bottom photograph displays the panel pitching at an angle of 10 degrees and water flowing at a water tunnel frequency of 3 Hertz. The three trials display a similar oscillation of flow in the wake, and none of them seem to have any distinct vortices, probably due to their relatively low Strouhal numbers. The top picture does look a little less similar than the other two; its wake appears to lose shape sooner, but this may be an effect of having a slightly higher Strouhal number.

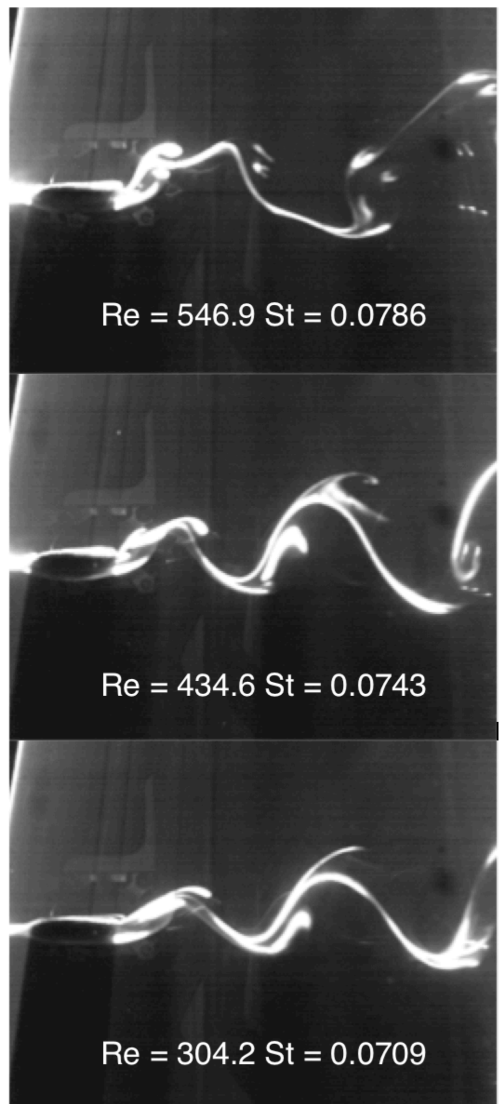


Figure 5 Comparing Reynolds numbers.

Experiments that vary only in Strouhal number do in fact look very different. Figure 6 displays four trials that only vary by Strouhal number. All four trials in the figure display the flow around a panel at the 40% location along half of the span, and they all share the same flow velocity since they are all set at a water tunnel frequency of 4.5 Hertz. The trials differ in pitching sweep angle, which is the parameter that alters the Strouhal number. There are two very important characteristics that can be compared in these four photographs. First, as Strouhal number is increased, the wake breaks down sooner than later. The drastic difference can be seen clearly by comparing the 20-degree trial with the 50-degree trial. As seen in the 20-degree trial, the wake does not appear to break down within the frame of reference, but as seen in the 50-degree trial, the wake shows loss of connectivity. Additionally, smaller Strouhal numbers appear to correspond to a narrower wake. Once again comparing extreme values of 50-degrees versus 20-degrees, the width of the wake approximately doubles with an approximately doubled Strouhal number. This is not to say that the width of the wake always increases linearly with Strouhal number; rather it is an observation found in this scenario.

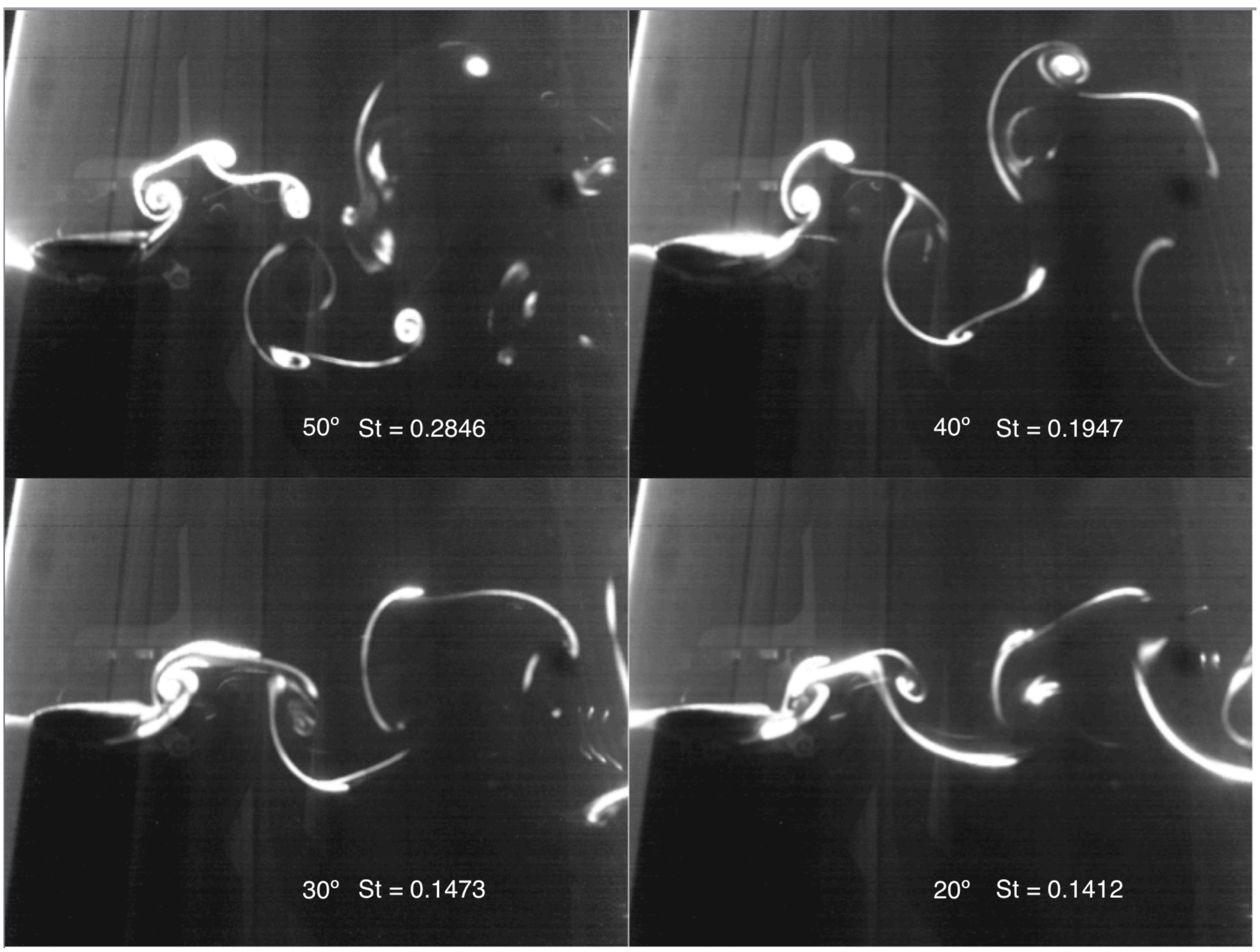


Figure 6 Comparing Strouhal numbers.

Previous researchers have shown that a wake also converges spanwise behind a pitching panel, which is a function of Strouhal number (Green, Rowley, and Smits 2011). The three-dimensional vortices in the wake shrink in spanwise length as they flow down stream and eventually break down. As Strouhal number is increased, vortex breakdown occurs sooner. The figure below shows an example of a converging wake. The six photographs all display a pitching panel at a 50-degree angle of sweep and a 6-Hertz water channel frequency, but each was taken at different spanwise locations. Each trial shown has equivalent Reynolds and Strouhal numbers.

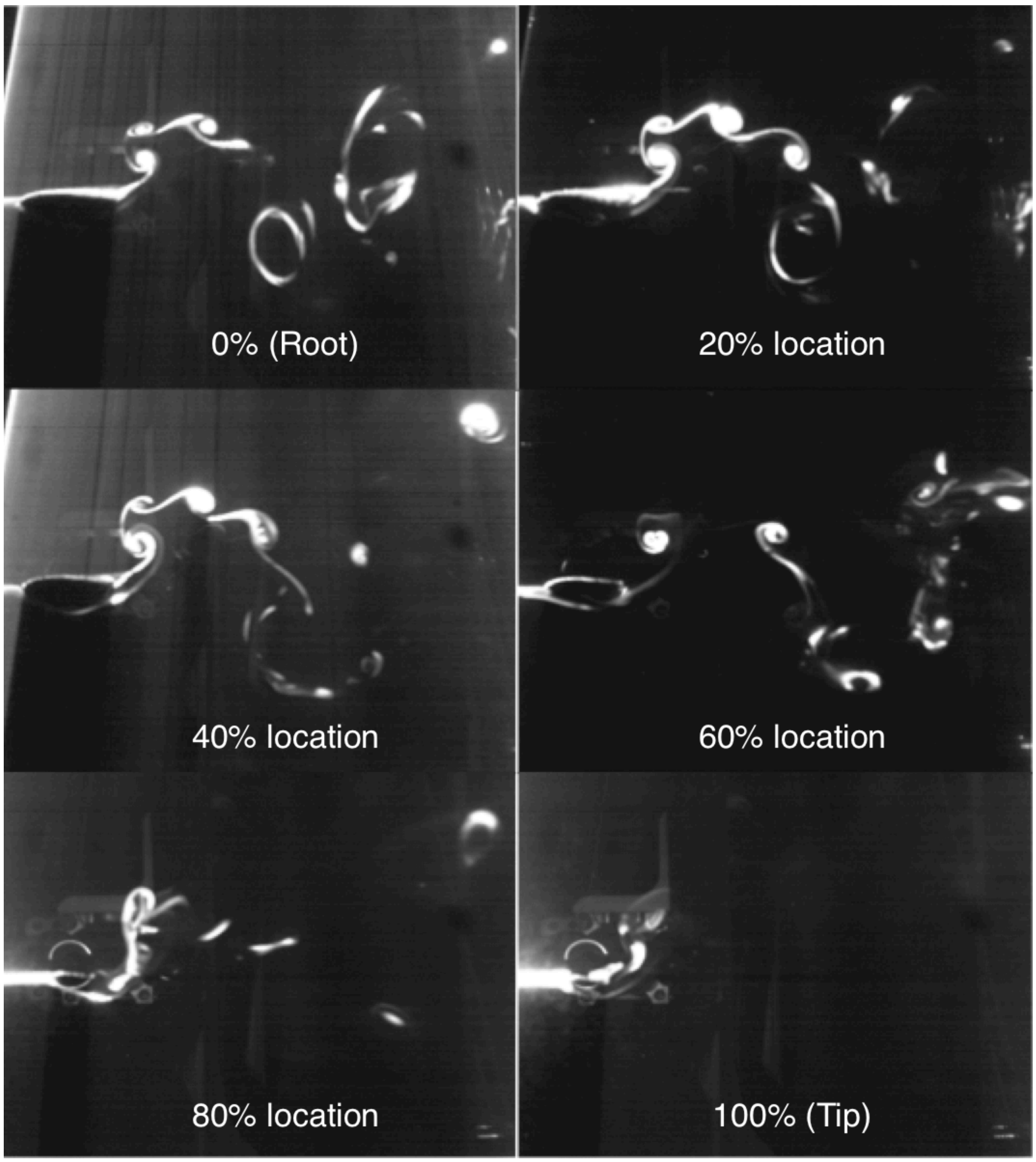


Figure 7 Comparing spanwise locations.

As the spanwise location approaches the tip of the pitching panel, the length of the wake decreases. It can be inferred from these pictures that the wake converges from the tips towards the root of the panel.

VIII. Conclusions

The purpose of experimenting with different parameters is to figure out the most optimal balance between power and efficiency while swimming through water. Power and efficiency can be observed by looking at the flow around the pitching panel. There appears to be a correlation between wakes that have well-developed vortices and efficient swimming. Wakes with well-developed vortices appear to correspond to Strouhal numbers within and around the 0.2 and 0.4 range, which supports the claim that previous researchers have made about oscillating swimmers having optimal propulsive efficiencies within the 0.25 and 0.35 range (Triantafyllou, Triantafyllou, and Grosenbaugh 1993). Therefore, wakes that are more chaotic and do not appear to have well developed vortices are less efficient. Additionally, when observing how long it takes for the flow to separate from the panel, it can be inferred from theory that a longer period of contact means for a greater amount of skin friction drag.

Observing the strength of a wake is more complicated. What does a powerful wake look like? Do larger, more powerful vortices behind the panel mean that the parameters are set up to have the system be more propulsive? Answers to these questions are still in the works, as researchers like Professor Green and her research group are currently pursuing meaningful results.

The flexible manner in which aquatic mammals swim does not seem practical for large boats or submarines, but perhaps could be applied to smaller modes of transportation or diving equipment. This is due to the likely possibility that larger vessels will create huge amounts of side-to-side propulsive force in the water due to having a kicking only motion instead of traveling forwards like fish and cetaceans that swim with both a kicking and borrowing motion.

IX. Future Work

The current goal being pursued is to create three-dimensional visualizations of the experimental data with the help of postdoctoral researcher, Rajeev Kumar. The three-dimensional view will provide a clear representation of the results, but unlike the two-dimensional views, it will include a view of how quickly the wake converges along the spanwise axis. Next, it would be of value to run experiments on the other dolphin caudal shapes that have been created to see how the shapes compare.

In the future, if this work were to be continued, it would be interesting to see how shark and whale shaped caudal fins compare and contrast to the dolphin's results. More conclusions could be made if both heaving and rolling motions were added to the overall movement of the fin. These additional motions would more accurately replicate how aquatic mammals swim. Additionally, it would be beneficial to make the fin as efficient as possible by adding a skin-like material to the panels for the purpose of reducing skin-friction drag and imperfections to the panel geometries that act as vortex generators. The purpose of vortex generation is to promote turbulent flow, which helps delay flow separation from the surface. A delayed surface separation would likely increase skin-friction drag slightly, while greatly decreasing the size of the wake, or "pressure drag." Theoretically, these vortex generators and a skin-like material could help the panel swim through water more efficiently.

References

1. Green, Melissa. "Green Fluids Laboratory." Green Fluid Dynamics Lab. *http://greenfluids.syr.edu/research.html*.

2. King, Justin T., and Melissa A. Green. "Experimental Study of the Three-Dimensional Wake of a Trapezoidal Pitching Panel."

3. Ayancik, Fatma, and Keith Moored. "Investigation of the Role of Planform Shape in Cetacean Swimming." Department of Mechanical Engineering and Mechanics. Lehigh University. Presentation.

4. Dewey, Peter A., Birgitt M. Boschitsch, Keith W. Moored, Howard A. Stone, and Alexander J. Smits. "Scaling laws for the thrust production of flexible pitching panels." *J. Fluid Mech*732 (2013): 29-46. Print.

5. Fish, F.E., and G.V. Lauder. "Passive and Active Flow Control by Swimming Fishes and Mammals." *Annu. Rev. Fluid Mech* 38.1 (2006): 193-224. Print.

6. Green, Melissa A., Clarence W. Rowley, and Alexander J. Smits. "The unsteady three-dimensional wake produced by a trapezoidal pitching panel." *J. Fluid Mech* 685 (2011): 117-145. Print.

7. Lauder, G. V., P. G. Madden, J. L. Tangorra, E. Anderson, and T. V. Baker. "Bioinspiration from fish for smart material design and function." *Smart Mater. Struct* 20.9 (2011): 094014. Print.

8. Triantafyllou, G.S., M.S. Triantafyllou, and M.A. Grosenbaugh. "Optimal Thrust Development in Oscillating Foils with Application to Fish Propulsion." *Journal of Fluids and Structures* 7.2 (1993): 205-224. Print.

Appendix

Table A.1 Reynolds and Strouhal numbers (0.2 < St < 0.4 are highlighted).

70 Degree Sweep			
Channel Frequency (Hz)	*Free Stream Velocity* (mm/s)*	*Reynolds Number*	*Strouhal Number*
3.0	21.00	304.2	0.4664
4.5	30.00	434.6	0.3265
6.0	37.75	546.9	0.2595
7.5	43.75	633.8	0.2239
9.0	47.80	692.5	0.2049
60 Degree Sweep			
Channel Frequency (Hz)	*Free Stream Velocity* (mm/s)*	*Reynolds Number*	*Strouhal Number*
3.0	21.00	304.2	0.4066
4.5	30.00	434.6	0.2846
6.0	37.75	546.9	0.2262
7.5	43.75	633.8	0.1952
9.0	47.80	692.5	0.1786
50 Degree Sweep			
Channel Frequency (Hz)	*Free Stream Velocity* (mm/s)*	*Reynolds Number*	*Strouhal Number*
3.0	21.00	304.2	0.3437
4.5	30.00	434.6	0.2406
6.0	37.75	546.9	0.1912
7.5	43.75	633.8	0.1650
9.0	47.80	692.5	0.1510

40 Degree Sweep			
Channel Frequency (Hz)	*Free Stream Velocity* (mm/s)*	*Reynolds Number*	*Strouhal Number*
3.0	21.00	304.2	0.2781
4.5	30.00	434.6	0.1947
6.0	37.75	546.9	0.1547
7.5	43.75	633.8	0.1335
9.0	47.80	692.5	0.1222
30 Degree Sweep			
Channel Frequency (Hz)	*Free Stream Velocity* (mm/s)*	*Reynolds Number*	*Strouhal Number*
3.0	21.00	304.2	0.2105
4.5	30.00	434.6	0.1473
6.0	37.75	546.9	0.1171
7.5	43.75	633.8	0.1010
9.0	47.80	692.5	0.0925
20 Degree Sweep			
Channel Frequency (Hz)	*Free Stream Velocity* (mm/s)*	*Reynolds Number*	*Strouhal Number*
3.0	21.00	304.2	0.1412
4.5	30.00	434.6	0.0988
6.0	37.75	546.9	0.0786
7.5	43.75	633.8	0.0678
9.0	47.80	692.5	0.0620
15 Degree Sweep			
Channel Frequency (Hz)	*Free Stream Velocity* (mm/s)*	*Reynolds Number*	*Strouhal Number*
3.0	21.00	304.2	0.1061
4.5	30.00	434.6	0.0743
6.0	37.75	546.9	0.0590
7.5	43.75	633.8	0.0509
9.0	47.80	692.5	0.0466

10 Degree Sweep			
Channel Frequency (Hz)	*Free Stream Velocity* (mm/s)*	*Reynolds Number*	*Strouhal Number*
3.0	21.00	304.2	0.0709
4.5	30.00	434.6	0.0496
6.0	37.75	546.9	0.0394
7.5	43.75	633.8	0.0340
9.0	47.80	692.5	0.0311
5 Degree Sweep			
Channel Frequency (Hz)	*Free Stream Velocity* (mm/s)*	*Reynolds Number*	*Strouhal Number*
3.0	21.00	304.2	0.0355
4.5	30.00	434.6	0.0248
6.0	37.75	546.9	0.0197
7.5	43.75	633.8	0.0170
9.0	47.80	692.5	0.0156

Arduino Code (5 Degree Sweep Only):

```
#include <VarSpeedServo.h>

VarSpeedServo myservo;  // create servo object to control a servo
                        // a maximum of eight servo objects can be
created

const int servoPin = 10; // the digital pin used for the servo

void setup() {
  myservo.attach(servoPin);  // attaches the servo on pin 10 to the
servo object
  myservo.write(0,255,true); // set the intial position of the servo,
as fast as possible, wait until done
}

void loop() {
  // write(degrees 0-180, speed 1-255, wait to complete true-false)

  myservo.write( 87.5000, 2.05,true);  // All the Way to the Left

  myservo.write( 87.5871, 2.05,true);
  myservo.write( 87.6818, 2.05,true);
  myservo.write( 87.7847, 2.05,true);
  myservo.write( 87.8966, 2.05,true);
  myservo.write( 88.0183, 2.05,true);
  myservo.write( 88.1505, 2.05,true);
  myservo.write( 88.2943, 2.05,true);
  myservo.write( 88.4506, 2.05,true);
  myservo.write( 88.6205, 2.05,true);
  myservo.write( 88.8052, 2.05,true);
  myservo.write( 89.0060, 2.05,true);
  myservo.write( 89.2243, 2.05,true);
  myservo.write( 89.4616, 2.05,true);
  myservo.write( 89.7196, 2.05,true);

  myservo.write( 90.0000, 2.05,true); // Max Speed and Mid Stroke

  myservo.write( 90.2804, 2.05,true);
  myservo.write( 90.5384, 2.05,true);
  myservo.write( 90.7757, 2.05,true);
  myservo.write( 90.9940, 2.05,true);
  myservo.write( 91.1948, 2.05,true);
  myservo.write( 91.3795, 2.05,true);
  myservo.write( 91.5494, 2.05,true);
  myservo.write( 91.7057, 2.05,true);
  myservo.write( 91.8495, 2.05,true);
  myservo.write( 91.9817, 2.05,true);
  myservo.write( 92.1034, 2.05,true);
  myservo.write( 92.2153, 2.05,true);
  myservo.write( 92.3182, 2.05,true);
  myservo.write( 92.4129, 2.05,true);
```

```
  myservo.write( 92.5000, 2.05,true); // All the Way to the Right

  myservo.write( 92.4129, 2.05,true);
  myservo.write( 92.3182, 2.05,true);
  myservo.write( 92.2153, 2.05,true);
  myservo.write( 92.1034, 2.05,true);
  myservo.write( 91.9817, 2.05,true);
  myservo.write( 91.8495, 2.05,true);
  myservo.write( 91.7057, 2.05,true);
  myservo.write( 91.5494, 2.05,true);
  myservo.write( 91.3795, 2.05,true);
  myservo.write( 91.1948, 2.05,true);
  myservo.write( 90.9940, 2.05,true);
  myservo.write( 90.7757, 2.05,true);
  myservo.write( 90.5384, 2.05,true);
  myservo.write( 90.2804, 2.05,true);

  myservo.write( 90.0000, 2.05,true); // Max Speed and Mid Stroke

  myservo.write( 89.7196, 2.05,true);
  myservo.write( 89.4616, 2.05,true);
  myservo.write( 89.2243, 2.05,true);
  myservo.write( 89.0060, 2.05,true);
  myservo.write( 88.8052, 2.05,true);
  myservo.write( 88.6205, 2.05,true);
  myservo.write( 88.4506, 2.05,true);
  myservo.write( 88.2943, 2.05,true);
  myservo.write( 88.1505, 2.05,true);
  myservo.write( 88.0183, 2.05,true);
  myservo.write( 87.8966, 2.05,true);
  myservo.write( 87.7847, 2.05,true);
  myservo.write( 87.6818, 2.05,true);
  myservo.write( 87.5871, 2.05,true);
}
```

MATLAB Code:

Contents

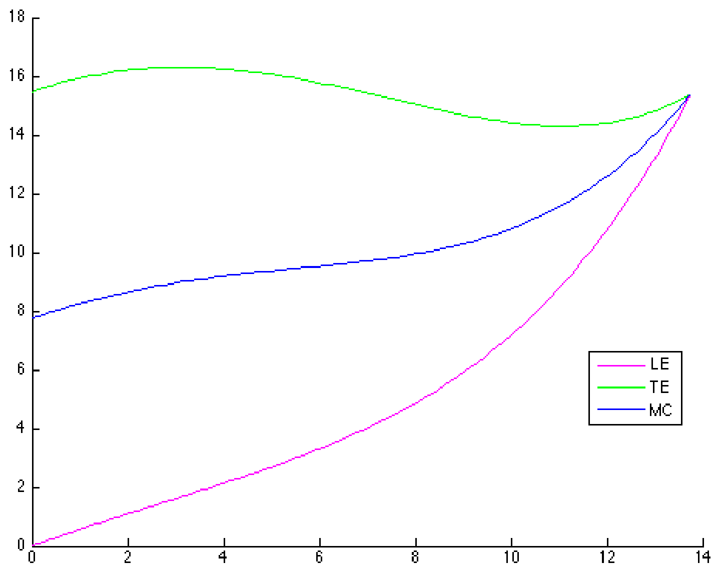

```
% Eric Zacharia
% Hydrodynamics of Caudal Fins
% Section 1: Plot of 11 different test pieces
% Section 2: Reynolds and Strouhal numbers
% Section 3: Arduino Code's Pitching Angles and Velocities
% Section 4: Experimental Visualization (Camera, LCS, and Q Criterion views)
clear all
close all
clc
TopRight = [880   385   560   420];
set(0, 'DefaultFigurePosition', TopRight)
```

Legend for all test pieces

```
0: tursiops_right4
1: 1common
2: 2Common
3: 3Common
4: 4Common
5: 2Tursiops
6: 3Tursiops
7: 4Tursiops
8: Delphinus
9: Grampus
10: Striped
```

```
[b,b_b,c_r,c_b,c_ms,c_t,x_ms,x_3qs,x_t,tmax_f,tmax_b,M ] = Parameters(1);

yfluke = linspace(0,b,100);
Nx     = 20;
Ny     = 20;

[MC,LE,TE,thick,c,xf,tmax] = Geometry(b,c_r,c_ms,c_t,x_ms,x_3qs,...
    x_t,yfluke,Nx,Ny,tmax_f);

figure(1)
hold on
plot(yfluke,LE,'m')
plot(yfluke,TE,'g')
plot(yfluke,MC,'b')
legend('LE','TE','MC','Location','best')
```

```
Warning: Polynomial is badly conditioned. Add points with distinct X values,
reduce the degree of the polynomial, or try centering and scaling as described
in HELP POLYFIT.
Warning: Polynomial is not unique; degree >= number of data points.
```



1common --- All Code below is only for the "1common" Test Piece.

Chord Approximations (Did not use these because they are a function of S, and I create S using chord) SMC = S/b % m Standard Mean Chord MAC = (2/S)*(c^2)*(b/2-0) % m Mean Average Chord

```
% Chord Length (m)
midchord  = 21.8694E-3;    % m Measured mid chord line of 1common
c         = TE - LE;       % Chord lengths at 100 half-span locations, not to scale
cmid      = max(c);        % Mid Chord, not to scale
c         = (mean(c)/cmid)*midchord; % m Averaged chord, scaled appropriately

% Pertinent Values
T         = 1.7;           % Seconds (Period of Pitching Motion)
f         = 1/T;           % Hz (Frequency of Pitching Motion)
mu        = 1.002E-3;      % Pascals (Dynamic Viscosity of Water at Room Temperature)
rho       = 999.97;        % kg/m^3 (Density of Water at Room Temperature)
b         = 62.2808E-3;    % m (Span length of Test Piece)
S         = c*b;           % m^2 Area of total test piece using an averaged chord

% Free Stream Velocity (Hz -> m/s) (From Jacob's plot)
v30       = 21.00E-3;      % m/s
v45       = 30.00E-3;      % m/s
v60       = 37.75E-3;      % m/s
v75       = 43.75E-3;      % m/s
v90       = 47.80E-3;      % m/s
```

Reynolds and Strouhal Numbers

```
theta70   = 70*(pi/180);  % Degrees converted to Radians
theta60   = 60*(pi/180);  % Degrees converted to Radians
theta50   = 50*(pi/180);  % Degrees converted to Radians
theta40   = 40*(pi/180);  % Degrees converted to Radians
theta30   = 30*(pi/180);  % Degrees converted to Radians
theta20   = 20*(pi/180);  % Degrees converted to Radians
theta15   = 15*(pi/180);  % Degrees converted to Radians
theta10   = 10*(pi/180);  % Degrees converted to Radians
theta5    = 5 *(pi/180);  % Degrees converted to Radians
w70       = theta70/T;    % Radians/Second
w60       = theta60/T;    % Radians/Second
w50       = theta50/T;    % Radians/Second
w40       = theta40/T;    % Radians/Second
w30       = theta30/T;    % Radians/Second
w20       = theta20/T;    % Radians/Second
w15       = theta15/T;    % Radians/Second
w10       = theta10/T;    % Radians/Second
w5        = theta5 /T;    % Radians/Second
L70       = sqrt(2*max(c)^2*(1-cos(theta70))); %inches converted to m
L60       = sqrt(2*max(c)^2*(1-cos(theta60))); %inches converted to m
L50       = sqrt(2*max(c)^2*(1-cos(theta50))); %inches converted to m
```

```
L40       = sqrt(2*max(c)^2*(1-cos(theta40))); %inches converted to m
L30       = sqrt(2*max(c)^2*(1-cos(theta30))); %inches converted to m
L20       = sqrt(2*max(c)^2*(1-cos(theta20))); %inches converted to m
L15       = sqrt(2*max(c)^2*(1-cos(theta15))); %inches converted to m
L10       = sqrt(2*max(c)^2*(1-cos(theta10))); %inches converted to m
L5        = sqrt(2*max(c)^2*(1-cos(theta5 ))); %inches converted to m

Re30      = rho*v30*c/mu;  % Reynolds number at 3.0 Hz
Re45      = rho*v45*c/mu;  % Reynolds number at 4.5 Hz
Re60      = rho*v60*c/mu;  % Reynolds number at 6.0 Hz
Re75      = rho*v75*c/mu;  % Reynolds number at 7.5 Hz
Re90      = rho*v90*c/mu;  % Reynolds number at 9.0 Hz

% 70 Sweep
St30_70   = f*L70/v30;     % Strouhal number at 3.0 Hz
St45_70   = f*L70/v45;     % Strouhal number at 4.5 Hz
St60_70   = f*L70/v60;     % Strouhal number at 6.0 Hz
St75_70   = f*L70/v75;     % Strouhal number at 7.5 Hz
St90_70   = f*L70/v90;     % Strouhal number at 9.0 Hz
% 60 Sweep
St30_60   = f*L60/v30;
St45_60   = f*L60/v45;
St60_60   = f*L60/v60;
St75_60   = f*L60/v75;
St90_60   = f*L60/v90;
% 50 Sweep
St30_50   = f*L50/v30;
St45_50   = f*L50/v45;
St60_50   = f*L50/v60;
St75_50   = f*L50/v75;
St90_50   = f*L50/v90;
% 40 Sweep
St30_40   = f*L40/v30;
St45_40   = f*L40/v45;
St60_40   = f*L40/v60;
St75_40   = f*L40/v75;
St90_40   = f*L40/v90;
% 30 Sweep
St30_30   = f*L30/v30;
St45_30   = f*L30/v45;
St60_30   = f*L30/v60;
St75_30   = f*L30/v75;
St90_30   = f*L30/v90;
% 20 Sweep
St30_20   = f*L20/v30;
St45_20   = f*L20/v45;
St60_20   = f*L20/v60;
St75_20   = f*L20/v75;
St90_20   = f*L20/v90;
% 15 Sweep
St30_15   = f*L15/v30;
St45_15   = f*L15/v45;
St60_15   = f*L15/v60;
St75_15   = f*L15/v75;
St90_15   = f*L15/v90;
% 10 Sweep
St30_10   = f*L10/v30;
St45_10   = f*L10/v45;
St60_10   = f*L10/v60;
St75_10   = f*L10/v75;
St90_10   = f*L10/v90;
% 5 Sweep
St30_5    = f*L5/v30;
St45_5    = f*L5/v45;
St60_5    = f*L5/v60;
St75_5    = f*L5/v75;
St90_5    = f*L5/v90;
```

Pitching Angles, Velocities, and Animations

The purpose of this code was to increase the angle of pitch exponentially so that there would be a higher concentration of points along the ends of the pitching motion. T = 1.7 seconds (f = 0.5882 Hz) (Position) 90 degrees is parallel to flow. (Position) 0 and 180 Degrees is perpendicular to flow.

```
dt          = 0:15;                              % Number of Time Steps
% 70 Degree Pitch
Degree70_1  =         54  + 1.2698532.^dt;       % From 55 to 90  degrees.
Degree70_2  = fliplr(126  - 1.2698532.^dt);      % From 90 to 125 degrees.
```

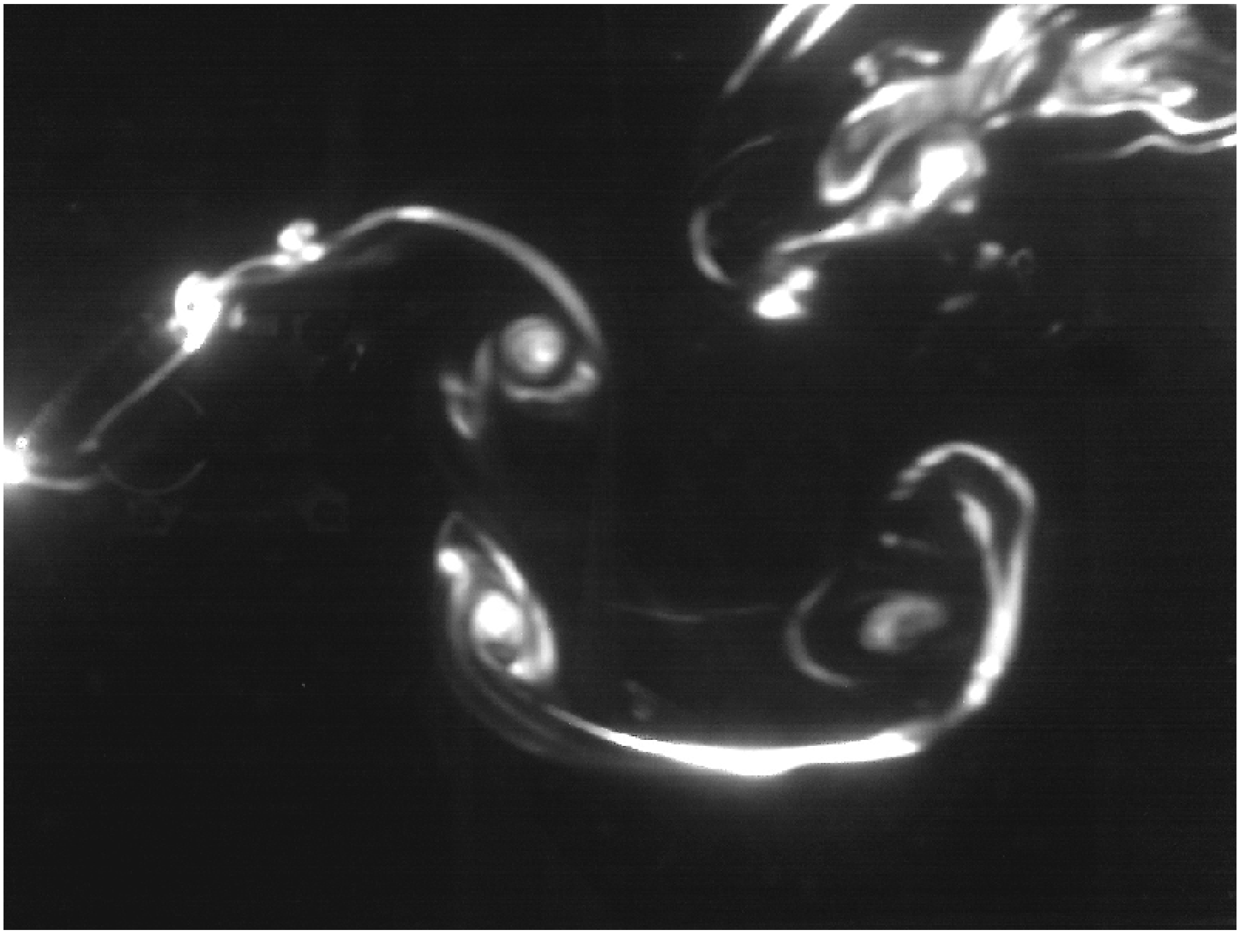
```
dV70        = linspace(0,3.568364,16);          % Velocity Array
Velocity70  = linspace(48,48,16);               % (48 degrees/time).
% 60 Degree Pitch
Degree60_1  =         59   + 1.2572572.^dt;     % From 60 to 90  degrees.
Degree60_2  = fliplr(121   - 1.2572572.^dt);    % From 90 to 120 degrees.
dV60        = linspace(0,3.52870100,16);        % Velocity Array
Velocity60  = linspace(39,39,16);               % (39 degrees/time).
% 50 Degree Pitch
Degree50_1  =         64   + 1.2426007.^dt;     % From 65 to 90  degrees.
Degree50_2  = fliplr(116   - 1.2426007.^dt);    % From 90 to 115 degrees.
Velocity50  = linspace(30,30,16);               % (30 degrees/time).
% 40 Degree Pitch
Degree40_1  =         69   + 1.2250333.^dt;     % From 70 to 90  degrees.
Degree40_2  = fliplr(111   - 1.2250333.^dt);    % From 90 to 110 degrees.
Velocity40  = linspace(21,21,16);               % (21 degrees/time).
% 30 Degree Pitch
Degree30_1  =         74   + 1.203025.^dt;      % From 75 to 90 degrees.
Degree30_2  = fliplr(106   - 1.203025.^dt);     % From 90 to 105 degrees.
Velocity30  = linspace(17.25,17.25,16);         % (17.25 degrees/time).
% 20 Degree Pitch
Degree20_1  =         79   + 1.173346.^dt;      % From 80 to 90 degrees.
Degree20_2  = fliplr(101   - 1.173346.^dt);     % From 90 to 100 degrees.
Velocity20  = linspace(10.25,10.25,16);         % (10.25 degrees/time).
% 15 Degree Pitch
Degree15_1  =        81.5 + 1.15335.^dt;        % From 82.5 to 90 degrees.
Degree15_2  = fliplr(98.5 - 1.15335.^dt);       % From 90 to 97.5 degrees.
Velocity15  = linspace(8.8,8.8,16);             % (8.8 degrees/time).
% 10 Degree Pitch
Degree10_1  =        84   + 1.126877.^dt;       % From 85 to 90 degrees.
Degree10_2  = fliplr(96   - 1.126877.^dt);      % From 90 to 95 degrees.
Velocity10  = linspace(5.05,5.05,16);           % (5.05 degrees/time).
% 5 Degree Pitch
Degree5_1   =        86.5 + 1.087104.^dt;       % From 87.5 to 90 degrees.
Degree5_2   = fliplr(93.5 - 1.087104.^dt);      % From 90 to 92.5 degrees.
Velocity5   = linspace(2.05,2.05,16);           % (2.05 degrees/time).
```

Experimental Visualization: "1common" Test piece: 0:299 pictures

dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 0% span/2015-11-03 20.46.18/'; %3.0 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 0% span/2015-11-03 20.46.54/'; %4.5 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 0% span/2015-11-03 20.47.28/'; %6.0 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 0% span/2015-11-03 20.48.01/'; %7.5 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 0% span/2015-11-03 20.48.37/'; %9.0 Hz %-- dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 20% span/2015-11-03 20.35.24/'; %3.0 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 20% span/2015-11-03 20.35.58/'; %4.5 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 20% span/2015-11-03 20.36.32/'; %6.0 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 20% span/2015-11-03 20.37.13/'; %7.5 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 20% span/2015-11-03 20.37.46/'; %9.0 Hz %-- dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 40% span/2015-11-03 19.20.35/'; %3.0 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 40% span/2015-11-03 19.21.31/'; %4.5 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 40% span/2015-11-03 19.22.04/'; %6.0 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 40% span/2015-11-03 19.22.39/'; %7.5 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 40% span/2015-11-03 19.23.14/'; %9.0 Hz %-- dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 60% span/2015-11-03 19.10.33/'; %3.0 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 60% span/2015-11-03 19.11.12/'; %4.5 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 60% span/2015-11-03 19.11.46/'; %6.0 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 60% span/2015-11-03 19.12.19/'; %7.5 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 60% span/2015-11-03 19.12.57/'; %9.0 Hz %-- dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 80% span/2015-11-03 17.33.43/'; %3.0 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 80% span/2015-11-03 17.34.24/'; %4.5 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 80% span/2015-11-03 17.35.44/'; %6.0 Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 80% span/2015-11-03 17.36.24/'; %7. Hz dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 80% span/2015-11-03 17.36.58/'; %9.0 Hz %-- dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 100% span//';%not done dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 100% span//';%not done dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 100% span//';%not done dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 100% span//';%not done dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 100% span//';%not done %-- dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/10 degree Sweep 0% span/2015-11-03 20.51.52/'; dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/10 degree Sweep 0% span/2015-11-03 20.51.15/'; dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/10 degree Sweep 0% span/2015-11-03 20.50.33/'; dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/10 degree Sweep 0% span/2015-11-03 20.49.57/'; dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/10 degree Sweep 0% span/2015-11-03 20.49.11/'; %-- dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/10 degree Sweep 20% span/2015-11-03 20.34.17/'; dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/10 degree Sweep 20% span/2015-11-03 20.33.40/'; dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/10 degree Sweep 20% span/2015-11-03 20.32.58/'; dir = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/10 degree Sweep 20% span/2015-11-03 20.32.24/'; dir =

```
% %-----------------------------------------------------------------------------------------------------------------------------
------------
% %-----------------------------------------------------------------------------------------------------------------------------
------------
% %-----------------------------------------------------------------------------------------------------------------------------
------------

% 40 percent half-span, 6 Hz channel frequency, All angles of sweep (5-70 degrees)
% dir     = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/5 degree Sweep 40% span/2015-11-0
3 19.22.04/'; %6.0 Hz
% dir     = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/10 degree Sweep 40% span/2015-11-
03 19.25.01/';
% dir     = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/15 degree Sweep 40% span/2015-11-
03 19.28.30/';
% dir     = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/20 degree Sweep 40% span/2015-11-
03 19.32.12/';
% dir     = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/30 degree Sweep 40% span/2015-11-
03 19.35.20/';
% dir     = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/40 degree Sweep 40% span/2015-11-
03 19.40.51/';
% dir     = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/50 degree Sweep 40% span/2015-11-
03 19.43.11/';
% dir     = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/60 degree Sweep 40% span/2015-09-
24 13.30.45/';
dir     = '/Users/ericzacharia/Desktop/Research/All Data Recordings/1common Data Recordings/70 degree Sweep 40% span/2015-09-24
 13.25.00/';

figure(2) % Actual Camera View
for  m  = 0:1:0
pillows  = [dir,'1commonJuly2315',sprintf('%6.6d',m),'.jpg'];
bacon    = imread(pillows);
imshow(bacon)
pause(1E-50)
legend('1common Test piece: 70 Degree Sweep at 60% half-span')
clear thing
end

figure(3) % Edge "LCS" View
for  m  = 0:1:0
pillows  = [dir,'1commonJuly2315',sprintf('%6.6d',m),'.jpg'];
bacon    = imread(pillows);
edgebob = edge(bacon);
imshow(edgebob)
pause(1E-50)
legend('1common Test piece: 70 Degree Sweep at 60% half-span')
clear thing
end

figure(4) % Center "Q criterion" View
% 0   = black
% 255 = white
blkwht  = zeros(100); % Preallocation for speed
for  m  = 0:1:0
pillows  = [dir,'1commonJuly2315',sprintf('%6.6d',m),'.jpg'];
bacon    = imread(pillows);
for i    = 1:1:1200;
    for j = 1:1:1600;
        if bacon(i,j) < 80;
        blkwht(i,j) = 255;
        else
        blkwht(i,j) = 0;
        end
    end
end
imshow(blkwht);
pause(1E-50);
clear thing
end
```

```
Warning: Image is too big to fit on screen; displaying at 50%
Warning: Plot empty.
Warning: Image is too big to fit on screen; displaying at 50%
Warning: Plot empty.
Warning: Image is too big to fit on screen; displaying at 50%
```



Contents


```
function[MC,LE,TE,thick,c,xf,tmax] = Geometry(b,c_r,c_ms,c_t,x_ms,x_3qs,x_t,yfluke,Nx,Ny,tm
ax_f)
```

```
%Geometry(0.2200,0.1710,0.1268,0.001,-0.0471,-0.0801,-0.1496,~,yfluke,Nx,Ny,tmax_f)
airfoil_shape='NACA';
```

Geometry parameters

b - span length of a single fluke (root to tip length). b_b - half-span of the body (x-z plane through the origin to the fluke root length). c_b - chord length of the body. c_r - root chord length of the fluke. c_ms - mid-span chord length. c_t - tip chord length (must be non-zero, but can be very small). x_ms - x position of the mid-chord line at the fluke root minus the x position of the mid-chord line at the fluke mid-span (_ms). x_3qs - x position of the mid-chord line at the fluke root minus the x position of the mid-chord line at the fluke three-quarter-span (_3qs). x_t - x position of the mid-chord line at the fluke root minus the x position of the mid-chord line at the fluke tip (_t). tmax_b - maximum thickness in chords of the body. tmax_f - maximum thickness in chords of the fluke. Atip - amplitude of the fluke tip measured from the maximum deflection to the x-y plane through the origin.

Pertinent Equations

```
%Fourth order approximation of mid chord line
A1 = (32*(6*x_ms - 8*x_3qs + 3*x_t))/(27*b^4);
A2 = -(444*x_ms - 448*x_3qs + 114*x_t)/(27*b^2);
A3 = (84*x_ms - 64*x_3qs + 15*x_t)/(9*b);
A4 = -c_r/2;
MC = -(A1*(yfluke - yfluke(1)).^4 + A2*(yfluke - yfluke(1)).^2 + A3*(yfluke - yfluke(1)) +
A4);

%A second-order approximation of the chord distribution along the span length.
F1 = (2*(c_r - 2*c_ms + c_t))/b^2;
F2 = -(3*c_r - 4*c_ms + c_t)/b;
F3 = c_r;
CFluke = F1*(yfluke - yfluke(1)).^2 + F2*(yfluke - yfluke(1))+ F3;

%Defining the leading-edge and trailing-edge lines.
LEFluke = MC - CFluke/2;
TEFluke = MC + CFluke/2;

%Stepping through each spanwise position to calculate the positions of the fluke neutral pl
ane at the given time step.
thick = zeros(Nx,Ny);
xf = zeros(Nx,Ny);
c = CFluke;
LE = LEFluke;
TE = TEFluke;
tmax = tmax_f*ones(length(yfluke),1);
```

```
for j=1:Ny
%Creating Ny points along the chord for each spanwise position for atotal of Nx*Ny points.
The points are spaced more closely at the leading and trailing edges.

        beta = linspace(0,pi,Nx);
        x_c = 1/2*(1 - cos(beta));   %half-cosine approximation for fine distribution
        xf(:,j) = LE(j) + c(j)*x_c;
%       xf(:,j) = LE(j) - c(j)*x_c;  %x-values on 3D plane gives Nx panel distribution rang
e

        % NACA four series shape coefficients.
        a0 = 0.2969;
        a1 = -0.1260;
        a2 = -0.3516;
        a3 = 0.2843;
        a4 = -0.1015;

            thick(:,j) = tmax(j)*c(j)/0.2*(a0*sqrt(x_c) + a1*(x_c) + a2*(x_c).^2 + a3*(x_c)
.^3 + a4*(x_c).^4);
            thick(:,j) = [0;thick(2:end,j) - thick(end,j)*ones(Nx-1,1)];

end
```

```
Error using Geometry (line 21)
Not enough input arguments.
```

```
end
```

Published with MATLAB® R2013a

Contents


```
function [ b,b_b,c_r,c_b,c_ms,c_t,x_ms,x_3qs,x_t,tmax_f,tmax_b,M ] = Parameters(Fluke)
```

```
% [ b,b_b,c_r,c_b,c_ms,c_t,x_ms,x_3qs,x_t,tmax_f,tmax_b,M ] = Parameters_v2( i,j,button, XA
,YA,XB,YB,x,y,minya,minxa,minyb,minxb)
%This code calculates the fluke parameters from image data and then it converts the units t
o the real data and appropriate unit by using scale factors convert parameters from pixels
to meters

if Fluke == 100
    load('common_right');
elseif Fluke == 0
    load('/Users/ericzacharia/Documents/MATLAB/Research/Geometry files/tursiops_right4.mat'
);
elseif Fluke == 1
    load('/Users/ericzacharia/Documents/MATLAB/Research/Geometry files/1common.mat');
elseif Fluke == 2
    load('/Users/ericzacharia/Documents/MATLAB/Research/Geometry files/2Common.mat');
elseif Fluke == 3
    load('/Users/ericzacharia/Documents/MATLAB/Research/Geometry files/3Common.mat');
elseif Fluke == 4
    load('/Users/ericzacharia/Documents/MATLAB/Research/Geometry files/4Common.mat');
elseif Fluke == 5
    load('/Users/ericzacharia/Documents/MATLAB/Research/Geometry files/2Tursiops.mat');
elseif Fluke == 6
    load('/Users/ericzacharia/Documents/MATLAB/Research/Geometry files/3Tursiops.mat');
elseif Fluke == 7
    load('/Users/ericzacharia/Documents/MATLAB/Research/Geometry files/4Tursiops.mat');
elseif Fluke == 8
    load('/Users/ericzacharia/Documents/MATLAB/Research/Geometry files/Delphinus.mat');
elseif Fluke == 9
    load('/Users/ericzacharia/Documents/MATLAB/Research/Geometry files/Grampus.mat');
elseif Fluke == 10
    load('/Users/ericzacharia/Documents/MATLAB/Research/Geometry files/Striped.mat');
end
```

```
Error using Parameters (line 6)
Not enough input arguments.
```

Geometry parameters

b - span length of a single fluke (root to tip length). b_b - half-span of the body (x-z plane through the origin to the fluke root length). c_b - chord length of the body. c_r - root chord length of the fluke. c_ms - mid-span chord length. c_t - tip chord length (must be non-zero, but can be very small). x_ms - x position of the mid-chord line at the fluke root minus the x position of the mid-chord line at the fluke mid-span (_ms). x_3qs - x position of the mid-chord line at the fluke root minus the x position of the mid-chord line at the fluke three-quarter-span (_3qs). x_t - x position of the mid-chord line at the fluke root minus the x position of the mid-chord line at the fluke tip (_t). tmax_b - maximum thickness in chords of the body. tmax_f - maximum thickness in chords of the fluke. Atip - amplitude of the fluke tip measured from the maximum deflection to the x-y plane through the origin.

```
pA      = polyfit(XA,YA,4);
pB      = polyfit(XB,YB,4);
y_data  = [min(XA):1:max(XA)];
fA      = polyval(pA,y_data);
fB      = polyval(pB,y_data);

% Define x data for all calculations
y_ms    = (max(y_data)+ min(y_data))/2;
y_3qs   = (3*max(y_data))/4;
y_t     = max(y_data);

% Define y data for all calculations
xA_data = fA;  %leading edge line
xB_data = fB;  %trailing edge line

xC_data = (xA_data+xB_data)/2; %mid-span line
b       = (max(y_data)-min(y_data));   % b      - span length of a single fluke (root to ti
p length).
b_b     = 0;                           % b_b    - half-span of the body (x-z plane through
the origin to the fluke root length).
c_r     = (max(xA_data)-max(xB_data)); % c_r    - root chord length of the fluke.
c_b     = c_r;                         % c_b    - chord length of the body.
XA_ms   = polyval(pA,y_ms);
XB_ms   = polyval(pB,y_ms);
XC_ms   = (XA_ms+XB_ms)/2;
c_ms    = XA_ms-XB_ms;                 % c_ms   - mid-span chord length.
c_t     = 0.001;                       % c_t    - tip chord length (must be non-zero, but c
an be very small).

x_ms    = -(max(xC_data)-XC_ms);       % x_ms   - x position of the mid-chord line at the f
luke root minus the x position of the mid-chord line at the fluke mid-span (_ms).
    XA_3qs = polyval(pA,y_3qs);
    XB_3qs = polyval(pB,y_3qs);
    XC_3qs = (XA_3qs+XB_3qs)/2;

x_3qs   = -(max(xC_data)-XC_3qs);      % x_3qs  - x position of the mid-chord line at the f
luke root minus the x position of the mid-chord line at the fluke three-quarter-span (_3qs)
.
    XA_t   = polyval(pA,y_t);
    XB_t   = polyval(pB,y_t);
    XC_t   = (XA_t+XB_t)/2;

x_t     = -(max(xC_data)-XC_t);        % x_t    - x position of the mid-chord line at the f
luke root minus the x position of the mid-chord line at the fluke tip (_t).

tmax_b  = 0.21;                        % tmax_b - maximum thickness in chords of the body.
tmax_f  = 0.21;                        % tmax_f - maximum thickness in chords of the fluke.
M       = 110;

% This part converts pixels to meters for Tursiops
    b = b*0.01;
    b_b = b_b*0.01;
    c_b = c_b*0.01;
    c_r = c_r*0.01;
```

```
    c_ms = c_ms*0.01;
    x_ms = x_ms*0.01;
    x_3qs = x_3qs*0.01;
    x_t = x_t*0.01;
```

```
end
```

Published with MATLAB® R2013a