THE EFFECT OF BUTTERFLY-SCALE INSPIRED
PATTERNING ON LEADING-EDGE
VORTEX GROWTH

by

JACOB AARON WILROY

AMY W. LANG, COMMITTEE CHAIR
CHARLES O’NEILL
CHANG-KWON KANG

A THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Aerospace Engineering and Mechanics
in the Graduate School of
The University of Alabama

TUSCALOOSA, ALABAMA

2016
ABSTRACT

Leading edge vortices (LEVs) are important for generating thrust and lift in flapping flight, and the surface patterning (scales) on butterfly wings is hypothesized to play a role in the vortex formation of the LEV. To simplify this complex flow problem, an experiment was designed to focus on the alteration of 2-D vortex development with a variation in surface patterning. Specifically, the secondary vorticity generated by the LEV interacting at the patterned surface was studied, as well as the subsequent effect on the LEV’s growth rate and peak circulation. For this experiment, rapid-prototyped grooves based on the scale geometry of the Monarch butterfly (*Danaus plexippus*) were created using additive manufacturing and were attached to a flat plate with a chordwise orientation, thus increasing plate surface area. The vortex generated by the grooved plate was then compared to a smooth plate case in an experiment where the plate translated vertically through a 2 x 3 x 5 cubic foot tow tank. The plate was impulsively started in quiescent water and flow fields at $Re_c = 1416, 2833, \text{ and } 5667$ are examined using Digital Particle Image Velocimetry (DPIV). The maximum vortex formation number is 2.8 and is based on the flat plate travel length and chord length. Flow fields from each case show the generation of a secondary vortex whose interaction with the shear layer and LEV caused different behaviors depending upon the surface type. The vortex development process varied for each Reynolds number and it was found that for the lowest Reynolds number case a significant difference does not exist between surface types, however, for the other two cases the grooves affected the secondary vortex’s behavior and the LEV’s ability to grow at a rate similar to the smooth plate case.
LIST OF ABBREVIATIONS AND SYMBOLS

\( \varepsilon \) error
\( \Gamma \) circulation
\( \lambda \) eigenvalue
\( \nu \) statistical degrees of freedom or number of measurements taken
\( \Omega \) vorticity tensor
\( \omega_z \) vorticity
\( A \) area of integration
\( c \) chord length of the plate
\( F \) formation number
\( \text{FOV} \) field of view
\( \text{HDPE} \) high density polyethylene
\( l \) distance translated by the plate
\( \text{LED} \) light emitting diode
\( \text{LEV} \) leading edge vortex
\( M \) number of measurements taken
\( N \) ratio of vorticity threshold value to vorticity positive spatial average
\( p \) probability
\( \text{PVC} \) polyvinyl chloride
\( q \) velocity tensor
\( R \) radius of integration
Re  Reynolds number
S  rate-of-strain tensor
t  time
t_{v,p}  \( t \) estimator for uncertainty analysis
TR-DPIV  time resolved – digital particle image velocimetry
\( U_{\infty} \)  average plate velocity
u  component of velocity in the x-direction or velocity of flow field
v  component of velocity in the y-direction
x  location along x-axis
y  location along y-axis
ACKNOWLEDGMENTS

I would like to start by thanking my advisor and committee chair, Dr. Amy Lang, for not only making this research possible, but for giving me an opportunity in the Research Experience for Undergraduates program so many years ago and for always believing in me. With her guidance, my experience as a researcher and a graduate student has been more fruitful than I could have ever forseen when I first began this journey.

I am also extremely thankful for the assistance Dr. Redha Wahidi has provided me through all the years, here at home and from abroad. I will forever be indebted to you for your guidance and persistence to teach me about research and about life.

Additionally, I would like to acknowledge the students, faculty, and staff at the University of Alabama and abroad. I am very grateful to Dr. O’Neill and Dr. Kang for serving on my thesis committee and offering their expertise and wisdom on the subject of fluid mechanics, among many others. Without the experience of Jim Edmonds and James Yarbrough of the machine shop, the construction of the models and experimental setup would not have been possible. I am also very thankful to graduate students Leonardo Santos, Sashank Gautum, Jennifer Wheelus, and Michael Bradshaw for the assistance they provided me in the lab. I would like to thank the many students who have helped me along the way: Cassidy Elliott, Andrew Bonacci, Jared Hammerton, and Mark Koren. Finally, I would like to thank my family and friends for their support and patience over the last several years and the Creator who was there “in the beginning”.

v
CONTENTS

ABSTRACT ........................................................................................................................................ ii

LIST OF ABBREVIATIONS AND SYMBOLS .................................................................................. iii

ACKNOWLEDGMENTS ....................................................................................................................... v

LIST OF TABLES ................................................................................................................................. ix

LIST OF FIGURES ............................................................................................................................... x

1. INTRODUCTION ............................................................................................................................ 1

1.1 MOTIVATION ............................................................................................................................... 1

1.2 RESEARCH OBJECTIVES ............................................................................................................ 2

2. LITERATURE REVIEW ..................................................................................................................... 4

2.1 INTRODUCTION .......................................................................................................................... 4

2.2 BUTTERFLY SCALES .................................................................................................................. 5

2.3 INSECT WING LEADING EDGE VORTEX .................................................................................. 7

2.4 VORTEX BOUNDARY INTERACTION ......................................................................................... 10

2.5 EXPERIMENTAL OBJECTIVES ................................................................................................ 11

3. EXPERIMENTAL SETUP ................................................................................................................. 14

3.1 MODEL PARAMETERS ............................................................................................................... 14

3.2 TOW TANK AND MOTION MECHANISMS ............................................................................... 16
3.3 DPIV SYSTEM.......................................................................................... 18
3.4 EXPERIMENTAL PARAMETERS............................................................... 19

4. THEORY ........................................................................................................ 22
  4.1 VORTICITY ............................................................................................... 22
  4.2 VORTEX IDENTIFICATION...................................................................... 24
  4.3 VORTICITY THRESHOLD ........................................................................ 27
  4.4 UNCERTAINTY ANALYSIS....................................................................... 29
  4.5 CIRCULATION .......................................................................................... 31

5. RESULTS ....................................................................................................... 34
  5.1 BASELINE: SMOOTH PLATE ................................................................. 34
  5.2 EFFECTS OF SURFACE PATTERNING ................................................. 41

6. CONCLUSIONS.............................................................................................. 53
  6.1 DISCUSSION ............................................................................................ 53
  6.2 FUTURE WORK ........................................................................................ 55

REFERENCES ..................................................................................................... 57

APPENDIX A: SUPPLEMENTAL VORTICITY FIELDS AND CIRCULATION PLOTS..... 60

APPENDIX B: MATLAB CODES ....................................................................... 67
  B.1: PHASE AVERAGING CODE...................................................................... 67
  B.2: DOMAIN ASSEMBLY AND VORTEX TRACKING CODE........................ 70
  B.3: GLOBALIZER FUNCTION..................................................................... 78
<table>
<thead>
<tr>
<th>Section Number</th>
<th>Section Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.4</td>
<td>LOCAL CIRCULATION FUNCTION</td>
<td>79</td>
</tr>
<tr>
<td>B.5</td>
<td>LAMBDA CI CRITERION FUNCTION</td>
<td>79</td>
</tr>
<tr>
<td>B.6</td>
<td>SEARCH EXPANSION FUNCTION</td>
<td>80</td>
</tr>
<tr>
<td>B.7</td>
<td>SEARCH MATRIX THINNING FUNCTION</td>
<td>83</td>
</tr>
<tr>
<td>B.8</td>
<td>TOTAL CIRCULATION FUNCTION</td>
<td>83</td>
</tr>
<tr>
<td>B.9</td>
<td>VORTEX CIRCULATION FUNCTION</td>
<td>89</td>
</tr>
<tr>
<td>B.10</td>
<td>IMAGE SELECTION CODE FOR PROCESSING</td>
<td>94</td>
</tr>
<tr>
<td>B.11</td>
<td>IMAGE EXTRACTION CODE AFTER PROCESSING</td>
<td>95</td>
</tr>
<tr>
<td>B.12</td>
<td>DATA SKIP FUNCTION</td>
<td>97</td>
</tr>
<tr>
<td>C.1</td>
<td>IMAGINE ACQUISITION AND PLATE TRANSLATION CODE</td>
<td>98</td>
</tr>
<tr>
<td>C.2</td>
<td>BINARY FILE CONVERSION EVEN SEQUENCE 3</td>
<td>100</td>
</tr>
<tr>
<td>C.3</td>
<td>BINARY FILE CONVERSION ODD SEQUENCE 3</td>
<td>101</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 3.1. Settings used during acquisition of images for each Reynolds number case ............21
LIST OF FIGURES

Figure 2.1. Optical image of wing-scales on *M. rhetenor* .......................................................... 5
Figure 2.2. Image showing particle contaminating before and after fogging ............................... 6
Figure 2.3. The 2nd class of LEV established by the literature ...................................................... 8
Figure 2.4. Images of smoke-wire visualization over *V. atalanta* .............................................. 9
Figure 2.5. Monarch butterfly wing showing scales and veins....................................................... 12
Figure 2.6. Microscopic image showing geometry of the scales and direction of flow .................. 13
Figure 3.1. Thrust speed performance vs. speed for the A-LST-E linear stage by Zaber ............ 17
Figure 3.2. Typical accuracy of the A-LST-E linear stage models provided by Zaber............... 17
Figure 3.3. CAD drawing and image of the experimental setup ............................................... 18
Figure 3.4. Example images of raw and processed PIV data...................................................... 21
Figure 4.1. Grid of data surrounding a typical point of evaluation used in calculating the local
circulation ....................................................................................................................................... 23
Figure 4.2. Plot of $Q$-Criterion vortex criterion overlayed on a contour of vorticity ............... 26
Figure 4.3. Plot of $\Lambda$-Criterion vortex criterion overlayed on a contour of vorticity............. 26
Figure 4.4. Plot of $\lambda_{ci}$-Criterion vortex criterion overlayed on a contour of vorticity ........... 27
Figure 4.5. Total circulation curves for various threshold values............................................... 29
Figure 4.6. Vortex identification ($\lambda_{ci}$) scatter plot with vorticity contour plot .................... 33
Figure 4.7. Vortex circulation for the $Rec = 3000$ case with smooth plate showing the
convergence of data ..................................................................................................................... 33
Figure 5.1. Non-dimensional total circulation for all cases studied ............................................ 35
Figure 5.2. Circulation and ensemble-average vorticity field for the smooth plate case at $Re_c = 1500$...............................37
Figure 5.3. Circulation curves for the smooth plate case for all Reynolds numbers ..................38
Figure 5.4. Velocity magnitude contour plots showing the existence of a secondary vortex.......40
Figure 5.5. Vortex circulation for the smooth and grooved case at $Re_c = 1500$ ...............43
Figure 5.6. Vortex circulation curve for the smooth and groove case at $Re_c = 3000$ ...........45
Figure 5.7 Vorticity contour plots for $Re_c = 3000$ for the smooth and grooved case ..........47
Figure 5.8. Circulation for the grooved case at $Re_c = 6000$ with error bars .........................49
Figure 5.9. Smooth and groove vortex circulation for $Re_c = 6000$..............................49
Figure 5.10. Vorticity contour plots at $Re_c = 6000$ for the smooth and grooved case at $F = 0.25,$
0.40, and 0.70 ........................................................................................................50
Figure 5.11. Vorticity contour plots at $Re_c = 6000$ for the smooth and grooved case at $F = 0.82,$
1.40, and 1.55 ........................................................................................................52
Figure A.1. Vorticity contour plots for $Re_c = 1500$ for the smooth case .........................61
Figure A.2. Vorticity contour plots for $Re_c = 1500$ for the grooved case .............................62
Figure A.3. Vorticity contour plots for $Re_c = 3000$ for the smooth case .............................63
Figure A.4. Vorticity contour plots for $Re_c = 3000$ for the grooved case .............................64
Figure A.5. Vorticity contour plots for $Re_c = 6000$ for the smooth case .............................65
Figure A.6. Vorticity contour plots for $Re_c = 6000$ for the grooved case .............................66
1. INTRODUCTION

1.1 MOTIVATION

Nature has existed and evolved through millions of years and has solved many problems for the sake of its own survival. Some of those solutions have been an inspiration for common engineering difficulties throughout history and will continue to provide motivation for the creation of technology for many years to come. As a matter of fact, when faced with a problem, the engineer may first look to nature to see if a solution already exists, and by recognizing the solution, studying it, copying it, and enhancing it, mankind can take advantage of what has already been created rather than “reinventing the wheel”. This is called biomimetics, and the first goal is to understand the phenomena occurring in nature, ultimately leading to its re-creation according to how the engineer understands the physics (Strickland, 2009). By examining the flight characteristics that insects have developed through evolution, engineers can better understand why the characteristics are important for flapping flight and how they can be applied to solve the problems of today. The Monarch butterfly (*Danaus plexippus*) makes a great example of evolutionary development of exceptional flight capabilities. Like many insects, the butterfly uses its wing kinematics and wing shape to create thrust and lift for flight, but one of the characteristics that set butterflies apart from many other insects is the presence of surface patterning (scales) on its wings (Van den Burg et al., 1997). These scales form microgeometries (or cavities) on the wing surface, and it is hypothesized that the scales provide an aerodynamic purpose that is related to the development of LEVs.
The volume of literature created on the subject of LEVs and insect flight prove that there are many motivational reasons for studying insects and insect flight. The Monarch has been the focus of many research projects such as Slegers et al. (2016), Jones (2011), and Srygley et al. (2002), just to name a few. One interesting characteristic of the Monarch is its ability to conduct long flights during migration. Starting in the fall season in the United States, the Monarch butterfly begins to migrate to warmer locations like Mexico and southwest California to escape the cold winter temperatures of their summer breeding grounds in North America and Canada. This migration has been seen to occur up to 3,600 kilometers over the course of three months (Brower, 1996; Howard et al., 2009). This is the longest migration of any known insect, and it is an important feature that spurs the curiosity of an aerodynamicist; therefore researchers are interested in determining what characteristics of this butterfly allow it to fly for such long distances. The goal of this project is to learn if the surface patterning contributes to this ability and plays a role in the butterfly’s aerodynamics.

1.2 RESEARCH OBJECTIVES

This study experimentally investigates the unsteady, two-dimensional formation of a leading edge vortex (LEV) created by a plate undergoing translation at a fixed pitch angle. The objectives are to characterize the LEV development and growth and to determine differences between two types of surfaces, one smooth and the other grooved. There are certain phenomena that occur during the development of LEVs and in order to compare the two cases, these phenomena need to be identified and discussed. Some example features that appear during LEV formation are the creation of a secondary vortex and its vorticity, instability shedding from the shear layer, separation of the LEV from the shear layer, and saturation of the LEV. The goal of this study is to qualitatively describe what these phenomena look like, their behavior, and explain
how they are related to the overall development of a LEV and how they differ based on Reynolds number and when surface patterning is used. In addition to these characteristics, quantitative observations and comparisons of vortex growth are conducted, such as through the examination of characteristics like time-varying circulation and saturation of vortex growth.

To study these phenomena, time-resolved digital particle image velocimetry has been used. Noise and error are inherent in experimental work, and especially PIV, so in order to deduce meaning from the data collected, methods of noise analysis and uncertainty were utilized. Once the interval of uncertainty was determined, an examination of the characteristics and behaviors of the vortices was conducted and the effects of surface type and Reynolds number on the LEV were elucidated in an attempt to shed more light onto the physics of vortex formation.
2. LITERATURE REVIEW

2.1 INTRODUCTION

An important component of flapping flight for at least some if not most insects is the ability to create a LEV and maintain its attachment to the wing for the generation of thrust and lift; this topic has been subjected to numerous investigations (Van den Burg et al., 1997; Birch et al., 2001; Bomphrey et al., 2005; Srygley et al., 2002; Sane et al., 2002). Butterflies follow this same trend, and it is hypothesized that the patterned surface created by their scales provides an aerodynamic purpose. Studies conducted on butterflies have shown the scales to be a multifunctional surface, providing hydrophobicity, low adhesion to contamination, and creating photonic effects important for mating and predator evasion (Wagner et al., 1996; Ingram, 2009).

In the study of insect aerodynamics, butterflies and other flapping wing insects, such as the *Manduca sexta* (tobacco hawkmoth) and the *Drosophila melanogaster* (fruitfly), have revealed that the LEV is not only important for generating lift, but its stability, control, and manipulation through wing kinematics are important for creating different aspects of flight performance such as hovering, takeoff, landing, and fast forward flight (Bomphrey et al., 2005; Birch et al., 2001; Van den Burg et al., 1997). For the butterfly, some of this function may come from the scales and so a review of studies regarding vortex-boundary interactions are discussed in the sections to follow. As can be seen, insect flight is a complicated phenomenon and an understanding of the different aspects is required for the explanation of observed flight performance.
2.2 BUTTERFLY SCALES

There are many orders of insects in the animal kingdom, but one order in particular has a feature that sets it apart. The numerous species of butterflies are categorized under the order Lepidoptera, which means “scaly wing”, and each one has a vibrant or unique color pattern. The Monarch butterfly (*Danaus Plexippus*) is no exception from the exquisite work of nature, and as will be shown below, these scales provide them with a few distinct benefits. The scales on butterflies are hypothesized to be driven by mimicry, camouflage, and species-specific mate recognition (Cott et al., 1940). The color created, whether through pigmentation or structure, is owed to the scales that cover the wing. For many butterflies, the seat of coloration is within the microstructure of the scale rather than pigment, called photonic nanostructure, and refraction of light on the nanometer-level appears to the eye as a wide spectrum of colors and iridescence. Figure 2.1 is an optical image of the wing-scales from a Blue Morpho (*M. rhetenor*) butterfly, which shows that the scales are responsible for providing the color (Ingram, 2009). In regard to

![Figure 2.1](image)

Figure 2.1. (a) optical image of wing-scales on *M. rhetenor*, showing the scales as the seat of the color; (b) dorsal and (c) ventral surface of a single iridescent *M. rhetenor* scale [Scale bars: (a) 100 μm; (b) and (c) 25 μm]. (Ingram, 2009)
dimensions, the scales appear as dust to the naked eye, but under a microscope one can see that the scales on the wing overlap like shingles on a roof. Typical scale surface dimensions are of the order 75 µm by 200 µm, having a thickness between 2-5 µm. The many physical elements of the butterfly, including those that influence the refraction of light from scales, have been greatly studied by Ingram (2009).

Another purpose of the scales on a butterfly’s wing is to provide a low adhesion surface and offer it the ability to shed water easily, called hydrophobicity. These features allow the butterfly the ability to, for example, achieve flight in rainy conditions or escape from its web-building predators. Additionally, since the butterfly’s wings are so large, cleaning the wings with its legs is not possible. Wagner et al. (1996) conducted a test where silicate dust particles were used to contaminate a butterfly wing and examine the removal of dust by water droplets. After contamination, a mist was applied and a digital image analysis was conducted to determine the percentage of particles removed. For the order Lepidoptera, only 0.55-4.69% of the particles remained on the wing after fogging and subsequent rolling off of the water droplets. Images

![Image of butterfly wing](image1.png)

Figure 2.2. Left image shows the forewing of *Lysandra bellargus* after contamination. Right image shows the same wing but after fogging. (Wagner et al., 1996)
before and after fogging can be seen in Fig. 2.2. The study also examined the contact angles of water droplets on wing surfaces, where a contact angle of $0^\circ$ meant the surface is completely wetted, and on the other end of the spectrum, a contact angle of $180^\circ$ is found when the drop touches the surface at only one point (low adhesion). For the order Lepidoptera, values between $131.4^\circ$ and $141.4^\circ$ were found for the upper side of the forewings, proving the hydrophobic nature of the surface. The study also discusses the theoretical considerations for unwettable surfaces and determined that the scale-built ridge structure provides the mathematically and physically optimum requirements for an unwettable and “self cleaning” surface (Wagner et al., 1996).

2.3 INSECT WING LEADING EDGE VORTEX

For almost a century it has been known that a LEV held above a wing increases lift. Munk (1925) theoretically examined the effects of dynamic stall over an airfoil and determined that increased circulation was required to maintain the Kutta condition at the trailing edge. Farren (1935) experimentally investigated this phenomenon and using several different airfoils he found that the lift of an airfoil rotating at a certain pitch rate could generate a coefficient of lift up to 50% higher than the mean at fixed angles and that this lift could be achieved roughly $8^\circ$ beyond the angle of maximum lift. While these results can be related to the aerodynamics created by flapping insects, they were created at much higher Reynolds numbers in an attempt to simulate inviscid conditions and may not be directly applicable to the low-to-intermediate Reynolds numbers that are relevant to insect flight; therefore a more accurate model is needed (Sane et al., 2002). Initially, studies on insect flight were conducted using ‘quasi-steady’ models where the time variant properties of aerodynamic force coefficients were ignored (Weis-Fogh, 1973; Sane et al., 2002). However, the forces experienced in insect flight, especially during hover, cannot be
accurately predicted by these methods as they are unable to account for unsteady aerodynamic effects (Zbikowski, 2002). In a study conducted by Van den Berg et al. (1997), the LEV generated by a ‘hovering’ hawkmoth model was reportedly responsible for 2/3 the lift created. This means that in order to predict force generation, a method for including the unsteady effects of dynamic stall must be included in theoretical models. Additionally, LEVs have a tendency to detach from the wing in 2D flow depending on the Reynolds number, so an understanding of the mechanisms for the removal of energy from the vortex must be understood for correct modeling (Birch et al., 2001).

So far, the literature has produced three classes of LEVs, where each class differs qualitatively in terms of flow topology. Figure 2.3 shows the class of LEV (class II) for butterflies (Bomphrey et al., 2005). This class of LEV is the only structure where a continuous vortex is observed across the span and where no significant spanwise flow has been reported. Studies conducted with tethered insects and flapper models have shown the LEV to be relatively stable and capable of greatly increasing lift, but mechanisms for creating LEV stability and continuous attachment to the wing during the downstroke have been widely debated. One

![Figure 2.3](image)

Figure 2.3. The 2\textsuperscript{nd} class of LEV whose composition is that of a single vortex which extends between both wing tips and includes a free-slip critical point over the centerline of the thorax. (Bomphrey et al., 2005)
method described by Van den Berg et al. (1997) is analogous to delta wing aircraft and is associated with class III LEVs, where the removal of energy from the vortex is done through a spanwise flow in the core. However, other studies conducted with similar experimental conditions have reported that at Reynolds numbers matching insect flight, the LEV generated is not like that of delta wings (Srygley et al., 2002; Birch et al., 2001). In the first study it was shown that no spanwise flow existed or could be detected by flow visualization in the LEV generated by red admiral butterflies (Vanessa atalanta), which is shown in Fig. 2.4.

In the second study, a dynamically scaled Drosophila mechanical wing was analyzed using PIV, and it was shown that no significant spanwise flow in the LEV core existed. Instead, stability of the growing vortex was hypothesized to be achieved by a downward flow induced by tip vortices. One source that studied the effect of vorticity dynamics and wing tip vortices on the lift of a flat plate determined that tip vortices were not only important for increasing lift from the low pressure region, but were also capable of anchoring the LEV, corroborating the previous

Figure 2.4. Images of smoke-wire visualization show that at the midline there exists a continuous vortex, also called a free-slip critical point (saddle). (Srygley et al., 2002)
study’s findings (Shyy et al., 2009). In addition to these flow features, butterflies and other insects have been observed to use wing kinematics as a mechanism for manipulating flow for the generation of high lift. Srygley et al. (2002) observed several different aerodynamic mechanisms in a free-flying Vanessa atalanta including: wake capture, two different types of leading-edge vortex, active and inactive upstrokes, in addition to the use of rotational mechanisms and the Weis-Fogh ‘clap-and-fling’ mechanism.

Starting as early as 1977, Maxworthy, among others, began to examine vortex rings and their development in an attempt to describe the evolution of the ring’s circulation, position, and size (Gharib et al., 1998). Gharib et al. (1998) furthered this by developing a scaling parameter, called the formation number, to characterize the vortex formation process, and then later Milano et al. (2005) modified this scaling parameter for flapping plates. Milano et al. (2005) showed that optimal unsteady force generation is linked to the formation of a LEV with maximum circulation and that this maximum was achieved by a certain amount of rotation and pitching, linking the observations of wing kinematics with aerodynamics. A study similar to the one created here focused on several parameters including circulation and vorticity of a fluid created by a plunging plate. It was found that the maximum circulation and separation of vortices was independent of formation number and dependent on parametric variations (Rausch et al., 2009). Using the formation number developed by Milano et al., the experimental setup used in this study was designed so that the vortex had the potential to achieve a large formation number of 2.8, allowing the observation of most of the vortex growth and development stages.

2.4 VORTEX BOUNDARY INTERACTION

It is hypothesized that by using its scales, the butterfly can control the growth of this vortex, which will allow for larger flapping amplitudes and a lower flapping frequency. Preliminary data
from the statistical analysis of a large number of free-flying butterflies suggests this very concept (Slegers, et al., 2016). By using its scales to control the vortex growth, butterflies can achieve more efficient flight, e.g. travel further with the same amount of energy, which is especially important for the migratory Monarch butterfly. Akin to the study of the wing kinematics’ effect on the LEV is an interest of LEV interaction with a boundary, and with regards to the Monarch, interaction with a specially designed patterned surface. A study by Kramer et al. (2007) numerically examined the interaction and behavior of a vortex dipole that approached a wall. They found that the adverse pressure gradient created by the vortex dipole causes the formation of a secondary boundary layer with vorticity of opposite sign to that of the primary boundary layer. This would then create a circulation cell that would advect primary boundary layer vorticity into the interior of the domain. This experiment is not concerned with the vortex dipole created by the leading and trailing edge vortex (LEV and TEV), but it is interested in the LEV and shear layer created by the leading edge. They also found that the strength of the secondary vortex is highly dependent on the strength of the boundary layer, which is of interest since this experiment intends to increase the boundary layer strength with an increase in surface area, therefore increasing friction on the plate surface. Orlandi (1990) also studied the behavior of a vortex dipole rebounding from a wall and investigated the role of secondary vortices. His studied revealed complex interactions of secondary and tertiary vortices with the main vortex.

2.5 EXPERIMENTAL OBJECTIVES

To simulate vortex interaction with a patterned surface, this experiment used rapid-prototyped square grooves epoxied to a flat plate. On a butterfly wing like the one shown in Fig. 2.5, one will notice that the scales are arranged in evenly spaced rows and that the physical arrangement of the scales remains consistent in orientation relative to the supportive viens from
wingbase to wingtip. Closer examination shows that the rows of scales create grooves (or cavities) and that these grooves are perpendicular to the supportive veins on either side of them.

![Figure 2.5. Monarch butterfly wing showing scales and veins.](image)

From a fluid mechanics perspective, the orientation of the grooves relative to the LEV induced flow over the wing surface can allow for an increase in drag due to a larger surface area being exposed to the flow or a decrease in drag from the roller bearing effect (Jones, 2011). An example of this is shown in Fig. 2.6 where the red arrow represents flow normal to the cavities (roller bearing effect) and the green arrow represents flow parallel to the cavities (increased drag).

For this experiment, chordwise grooves were created to simulate flow parallel to the row of cavities. It should also be noted that on the hindwing of the Blue Pansy (*Junonia orithya*), Kusaba et al. (2009) found the scales to generally decrease in size up to 40% from the wing base towards the edge, which was ascribed to a maturation wave resulting from morphogenetic factors. Considering again a fluid mechanics perspective, this decrease is important for maintaining a certain cavity Reynolds number since wing rotation causes a variation of induced velocity over the wing leading edge. This experiment uses a translating plate to represent flapping and so the groove depth variation is not applicable and was not studied, but it does provide further evidence that suggests an aerodynamic purpose for the scales.
Figure 2.6. Microscopic image showing geometry of the scales and direction of flow.
3. EXPERIMENTAL SETUP

3.1 MODEL PARAMETERS

To generate a vortex, a flat plate with a chord of 7 inches and a total span of 34 inches was used. For the baseline case studied, a smooth surface plate was used which had a thickness of 0.75 inches, giving a thickness to chord ratio of approximately 11%. For the grooved flat plate, the same dimensions were used but a foundation plate with a thickness of 0.5 inch was used to compensate for the addition of the 0.25 inch thick 3-D printed grooved plates. Both flat plates were rounded along the top edges with a fillet radius of 3/8 inch, a dimensional feature necessary to eliminate undesired separation at the top edge due to the thickness of the plate. For this experimental setup, the flat plate was composed of a 24 inch middle section and two 4.75 inch plate end extensions. Initially, experiments were run with only the 24 inch middle section, but turbulence created by the mounting hardware on the plate ends quickly caused the destruction of the main vortex, so extensions were added to create a virtually infinite two-dimensional plate. Half inch by 0.25 inch door insulation foam was also glued to the plate extension to provide a closed seal between the plate ends and the tank wall, reducing any mass flux to negligible. This setup allowed the LEV to attach to the tank wall (boundary). Hermann von Helmholtz in 1858 provided a set of theorems or “rules” about vortex filaments which describe how the filament behaves in an inviscid flow acted on by conservative body forces (Bertin et al., 2014). While the flow in this experiment does not exactly meet those conditions, the theorems still provide some guidance for how to generate a stable vortex. After the addition of plate extensions, the LEV was allowed to attach to a boundary (one of Helmholtz’s theorems)
and as a result became more stable. The flow field was also confirmed to be highly two-dimensional by shining a light emitting diode (LED) flash light into the tow tank. Little to no spanwise flow was seen even within the low pressure core.

For the material, the smooth flat plate is made of black high-density polyethylene (HDPE) to reduce the reflectivity of laser light and to provide sufficient rigidity, while the grooved baseplate is made of grey polyvinyl chloride (PVC) material, with grey being the darkest color available. The epoxy chosen for gluing the grooved plates to the baseplate did not suitably adhere to HDPE according to the manufacturer, so PVC was chosen instead. Since plates were being glued to the PVC, the bottom surface of the baseplate could be painted black to reduce laser light reflectivity. The grooved plate was printed in the University of Alabama 3D Prototyping lab using their Stratasys Objet 30 Pro, which uses a proprietary, photo-active resin to create three-dimensional parts with a resolution of up to 900 dots per inch (dpi) in the z-axis and an accuracy of roughly 0.0039 inches (Stratasys). The transparent “VeroClear” resin material was used to print the grooved plates instead of the “VeroBlackPlus” due to its increased dimensional stability, which has been exhibited in other 3D printed parts and noted by the laboratory workers. The maximum x- and y-dimensions of the build tray are 11.57 inches by 7.55 inches, which required that the plate be printed in sections. Three 7 inches by 8 inches grooved plates were printed for the middle section and two 7 inch by 4.75 inch plates were printed for the plate extensions. These were attached using Loctite Epoxy for plastics.

The square-grooved plates have a cavity depth of 0.5 mm and an aspect ratio of 1. The cavities were created by extruding 0.2 mm thick protuberances upward along the chordwise direction. The protuberance thickness was chosen based on a trial and error solution to create a thin but rigid cavity wall. The groove model plates were also examined under a stereoscopic
microscope to examine the groove geometry and to ensure they were printed accurately. Using information concerning cavity size and Reynolds number based on depth ($Re_d$) from Jones (2011) and butterfly flight characteristics from Cranford (2015), the grooves were sized to yield $Re_d = 3.98, 7.97, \text{ and } 15.94$, which accompanied the three chosen flow speeds of 8, 16, and 32 mm s$^{-1}$, respectively. These flow parameters are discussed in further detail in § 3.4 below.

3.2 TOW TANK AND MOTION MECHANISMS

The experiment was conducted in a 36 inch by 24 inch by 60 inch free-surface tow tank that is composed of 1 inch thick Plexiglas® panes supported by 80/20® t-slotted extruded aluminum bars. The plate model was attached to a traverse carriage mount using two 54 inch stainless steel bars and the traverse was mounted to a custom made steel tubing frame. The camera was mounted to a PBC Linear Readi Rail and was placed adjacent to the tank. The camera was translated vertically with the plate via an adjustable cable attached to a hollow 2-inch by 1-inch by 0.25 inch aluminum cross bar, which was attached to the traverse carriage mount, allowing the camera to move in the same inertial reference frame as the plate. An A-LST0750B-E01-KT07U high load linear stage (traverse) produced by Zaber Technologies was used for vertical movement of the plate and camera. Limitation in the traverse’s lifting capacity (shown in Fig. 3.1 as a blue line) required the use of a counterweight system. Typical accuracy of the traverse is shown in Fig. 3.2 and was used for the uncertainty analysis. Since the plate model hung nearly 5 feet beneath the traverse, an internal rail system using PBC Linear’s Commercial Rail Linear Guide Series was attached to a rectangular frame of 80/20® bars, which were placed inside the tank to support the plate model. Figure 3.3 shows a schematic of the experimental setup.
Figure 3.1. Thrust speed performance vs. speed for the A-LST-E linear stage models provided by Zaber.

Figure 3.2. Typical accuracy of the A-LST-E linear stage models provided by Zaber.
3.3 DPIV SYSTEM

To measure the flow field a two-dimensional time-resolved digital particle image velocimetry (TR-DPIV) system was used. This system is composed of a laser to illuminate the measurement plane below the leading edge at the midspan, a custom LabVIEW program used to synchronously move the flat plate vertically and acquire images, seeding particles, a camera, and DPIV processing software. A Quantronix Darwin Series 527-30-M Nd:YLF diode-pumped solid-state laser was used to illuminate a 1 mm thick sheet of flow by utilizing an array of optics that reduced the laser beam diameter and then fanned the beam into a sheet. A LabVIEW code that combined Zaber Technologies’s linear stage control software and National Instrument’s Vision package for image acquisition software was developed for data collection. The flow was seeded with Potter Industries’s neutrally buoyant silver-coated hollow glass spheres, having an average diameter of 14 μm. To capture image data, a Basler A504k 8-bit high-speed digital
camera with a Nikon AF Micro Nikkor 20-105mm lens with f-number = 2.8 was used. The images collected have a resolution of 1024 x 1024 pixels², which gives a spatial resolution of 1.20 mm. The camera frame rates and exposure times and the laser repetition frequency were varied for each Reynolds number case to ensure that raw images were recorded with clear definition and that particles moved no further than 4-6 pixels between two consecutive images. An example of raw and processed images is shown in Fig. 3.4. A large pulse width relative to the selected camera exposure time and a high pulse repetition frequency allowed the laser to be operated independently of the camera while still providing sufficient illumination. The camera and laser settings were determined using trial-and-error and care was taken to obtain images with a minimum amount of particle blurring. The final settings chosen for this experiment allowed the PIV software to statistically obtain the most number of good correlations. These parameters, as well as the laser settings, are presented in Table 3.1. To process data we used TSI’s Insight 4G™ PIV processing software. A square recursive Nyquist grid with 50% overlap was used, which had a primary correlation window size of 32 x 32 pixels and a sub-correlation window of 16 x 16 pixels. This yielded 127 x 127 vectors for each image. MATLAB programs were written for post-processing and analysis of the data. The experimental parameters used in this experiment are presented in Table 3.1.

3.4 EXPERIMENTAL PARAMETERS

For this experiment, the plate was impulsively started in quiescent water with a 45° angle of attack at three different velocities. These velocities yield Reynolds numbers based on the plate chord \((Re_c)\) of 1416, 2833, and 5667. The decision for the angle of attack studied was established using several parameters. Butterfly climbing flight data taken from Cranford (2015) was used to conduct a simple analysis which determined that the butterflies’ wing could experience an angle
of attack around 45° for medium to fast forward flight speeds. The butterflies tested in that experiment achieved a Reynolds number based on the wingspan of one wing (root to tip) of approximately 6000, so this Reynolds number was used as the foundation for establishing the plate chord length and cavity depth in this experiment. Additionally, lower Reynolds numbers were studied to examine their effect on the LEV and the ability of the patterned surface to affect the vortex formation. The angle of attack mentioned above was found by using the average of the various kinematic metrics and reference lengths that were cited within the aforementioned source. This is, however, only one flight condition as the butterfly often combines climbing flight with hovering and gliding, as well as multiple types of wing beat mechanisms, all of which create different flow conditions over the wing. For instance, in hover the angle of attack could be increased up to 90° while for steady, gliding flight the angle of attack could be very small, so in addition to the above parameters, a preliminary trial of experiments were conducted for the project herein where multiple angles of attack were sampled. It was found that the 45° angle of attack case presented the most interesting flow phenomena, and so for this experiment, only the 45° case was studied due to time limitations. The LEV generated by the plate was fairly large in size relative to the field of view (FOV) which led to a limitation in the ability to view the vortex using the dual camera setup. This in turn limited the maximum formation number that could be reached, which is defined as the total travel length divided by the chord \( F = l/c \), so for this experiment the traverse that translated the plate was set to travel 500 mm yielding a maximum formation number of 2.8. This length of travel allowed the vortex to form and separate from the shear layer, thus enabling the investigation of a significant portion of the vortex development period. Twenty total runs were collected and every 10th raw image was analyzed to reduce the
amount of time needed for processing. An ensemble average is created for each case by averaging the twenty runs at each phase.

![Example images of raw and processed PIV data.](image)

**Table 3.1.** Settings used during acquisition of images for each Reynolds number case.

<table>
<thead>
<tr>
<th>Equipment</th>
<th>Metric</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plate</td>
<td>Velocity [mm/sec]</td>
<td>8 16 32</td>
</tr>
<tr>
<td></td>
<td>Reynolds Number</td>
<td>1416 2833 5667</td>
</tr>
<tr>
<td>Camera</td>
<td>Frame Rate [frames/sec]</td>
<td>30 60 120</td>
</tr>
<tr>
<td></td>
<td>Exposure time [μs]</td>
<td>20000 8333 5000</td>
</tr>
<tr>
<td>Laser</td>
<td>Amperage [A]</td>
<td>30 30 30</td>
</tr>
<tr>
<td></td>
<td>Pulse Rate [Hz]</td>
<td>250 480 1000</td>
</tr>
</tbody>
</table>
4. THEORY

4.1 VORTICITY

To calculate the out-of-plane vorticity ($\omega_z$), a Matlab code was written to use the two-component DPIV data in a “local circulation” method as defined by DeVoria et al. (2012) and discussed in Raffel et al. (2007). This method is reported to be more accurate due to the fact that it incorporates more data points into the calculation. Vorticity in two dimensions is defined as

$$\omega_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$  \hspace{1cm} (4.1)

However, instead of using a finite difference scheme, where two one-dimensional velocity gradients are examined, a local circulation calculation is used at each location ($x_0, y_0$) in the FOV. Using the contour integral definition of circulation, a circle with a radius equal to the spatial resolution is drawn at each data point. The eight points surrounding ($x_0, y_0$) are then used to create new velocity vectors along the arc length of the integration path at an equal angular interval of 45°. A schematic of this from DeVoria’s thesis (2011) is shown below in Fig. 4.1. Velocity vectors along the circle were calculated using an inverse distance weighted interpolation scheme given by Shepard (1968). After assuming a linear variation between the data points along the circle, an analytical integration was performed. Stokes’ Law was then used to obtain the area integral definition of circulation, and considering the radius of the integration area to be small when compared to the flow structure’s characteristic length, the following relation was used to obtain vorticity:
\[
\omega_z = \frac{\delta \Gamma}{\delta A}
\]  

(4.2)

The final result of this method derived by DeVoria (2011) for calculating the vorticity at the point \((x_0, y_0)\) is given below:

\[
\omega_z = \left( \frac{3 + 2\sqrt{2}}{24R} \right) \left\{ \left( 14 - 9\sqrt{2} \right) [(u_7 - u_3) + (v_1 - v_5)] + \left( \sqrt{2} - 1 \right) [(u_6 + u_8 + v_2 + v_8) - (u_2 + u_4 + v_4 + v_6)] \right\}
\]

(4.3)

Here the \(u_i\)'s and \(v_j\)'s are the velocity data and the subscripts indicate their relative location within the grid around the point \((x_0, y_0)\). Since this method is invalid for locations near the plate or at the FOV edge, those points were zeroed. In reality, a modified version of this method could be used where only part of the original integration circuit is used, but lower light intensity and loss of particles from the viewing area at the edges of the domain tend to increase error at those locations, so the grid points were simply zeroed.

Figure 4.1. Grid of data surrounding a typical point of evaluation (i.e. point 0). The solid circles are the locations of measured data and the open circles are the locations of interpolated data. The large circle is the integration circuit used in calculating the local circulation.

(DeVoria A., 2011)
4.2 VORTEX IDENTIFICATION

After calculating the vorticity at each point, the circulation of the vortex can be found, but before this can be done, the vortex must be identified and distinguished from the fluid around it. Several criteria were considered including the $Q$-Criterion as defined by Hunt et al. (1988), the $\Delta$-Criterion as defined by Chong et al. (1990), and the $\lambda_{ct}$-Criterion as defined by Zhou et al. (1999). All four methods utilize an analysis of the velocity gradient tensor, $\nabla q$, whose characteristic equation is given by

$$\lambda^3 + P\lambda^2 + Q\lambda + R = 0 \quad (4.4)$$

where $P$, $Q$, and $R$ are the three invariants of $\nabla q$ (Chakraborty et al., 2005). The $Q$-Criterion requires that the second invariant of the gradient of the velocity tensor be positive and was calculated in the following way:

$$Q = \frac{1}{2} (\|\Omega\|^2 - \|S\|^2) \text{ for } Q > 0 \quad (4.5)$$

where $\| \cdot \|$ is the Euclidean matrix norm and where

$$\Omega = \frac{1}{2} (\nabla q - \nabla q^T) \text{ and } S = \frac{1}{2} (\nabla q + \nabla q^T) \quad (4.6)$$

Here, $S$ is the rate-of-strain tensor and $\Omega$ is the vorticity tensor, which are also the symmetric and antisymmetric components of $\nabla q$, respectively. Therefore when the magnitude of vorticity is larger than the magnitude of the rate-of-strain, the value of $Q$ is kept, otherwise it is zeroed.

Another vortex criterion considered for vortex identification, the $\Delta$-criterion, is given in Eqn. 4.7.

$$\Delta = \left(\frac{1}{2} R\right)^2 + \left(\frac{1}{3} Q\right)^3 \quad (4.7)$$

Here $R = -\det(\nabla q)$, the third invariant of the velocity tensor gradient. Comparing Eqns. 4.5 and 4.7, it can be seen that $Q > 0$ is more restrictive than $\Delta > 0$. The third criterion is the $\lambda_{ct}$-Criterion, also known as the ‘swirling strength’ criterion, and it is based on the $\Delta$-Criterion and
uses the imaginary part of the complex conjugate eigenvalue of $\nabla q$ to identify vortices (Chakraborty et al., 2005).

Three different plots are shown below for a typical case in Figs. 4.2 through 4.4, where the contour shows levels of vorticity and the exes are located where the respective vortex criterion has identified that point as part of a vortex. Qualitatively, one can see that the $\Delta$-Criterion performs the poorest, while the other criteria seem to identify the vortex rather well. A quantitative analysis was needed to discern a credible difference, and this was done by calculating circulation, but before this could be done, the vortex had to be tracked through time and the points being identified as part of the vortex had to be clustered. A threshold was also applied to the vorticity to reduce the effects of noise in the measured flow field. In deciding which vortex criterion to use it was important that the vortex circulation and total circulation be very similar when the vortex is initially formed since the total circulation includes all of the vorticity in the FOV (vortex and shear layer) and very early in the formation the vortex and shear layer are still connected. This factor, along with the threshold analysis presented below, was used to determine which vortex criterion performed most suitably. A Matlab code was written to carry out this task and is presented in Appendix B. By identifying either the peak vorticity value or vortex criterion value when the vortex first forms, the trend of peak value locations was used to track the vortex as it moved through the FOV. After clustering the vortex points, Eqn. 4.2 was rearranged to solve for $\delta \Gamma$ and the vorticity was summed and multiplied by the differential area, and as stated by DeVoria (2011), the differential area used for this operation is the grid spacing rather than the circular area used initially.
Figure 4.2. Plot of $Q$-Criterion vortex criterion overlayed on a contour of vorticity.

Figure 4.3. Plot of $\Delta$-Criterion vortex criterion overlayed on a contour of vorticity.
Before circulation could be accurately examined, the noise due to ambient motion in the fluid had to be reduced. This was done by following the methods of DeVoria et al. (2012) and Wahidi et al. (2015), whereby a threshold was used to eliminate values from the circulation calculation. The threshold is defined as

$$\omega_{z,th} = N \omega_{z,avg}$$  \hspace{1cm} (4.9)

where $\omega_{z,avg}$ is the positive spatial average of the vorticity for each image and $N$ is some value between 0 and 1.60. To find the appropriate threshold, $N$ is incrementally increased from 0 in steps of 0.20 so that values of vorticity less than the specified percentage of the spatial average are thresholded out. The time-varying total circulation, $\Gamma_{total}$, generated by the LEV and plate leading edge is then calculated and the effect on $\Gamma_{total}$ of increasing the threshold is examined.

Figure 4.4. Plot of $\lambda_{ci}$-Criterion vortex criterion overlayed on a contour of vorticity.
Care was taken when selecting the appropriate threshold since too small of a threshold will lead to artificially high values of circulation, and too large of a threshold will affect the growth phase and plateau region of the total circulation (DeVoria et al., 2012). An overly large threshold will also begin to break the integrity of the vortex structure as explained by Wahidi et al. (2015). DeVoria et al. (2012) also used the criterion that $\Gamma_{\text{total}}$ be very close to 0 at $t = 0$, but in this study that criterion proved to be very difficult to achieve. This was caused by the fact that when the plate initially starts to move, a boundary layer of fluid moving toward the trailing edge of the plate formed along the chord and created a long strip of vorticity that was of the same sign as the vortex. This made the vorticity between the two indistinguishable, even through the use of vortex identification criterion. One likely reason this boundary layer appeared is due to the low angle of attack used in this study relative to other flapping wing experiments, which allowed some of the flow to remain attached to the plate just after the plate has begun to move, and this is especially true for the lowest Reynolds number case. For all Reynolds number cases studied, a TEV would form and then quickly separate causing fluid to be drawn toward the trailing edge. The interaction of the TEV with the LEV can be seen in Rausch et al. (2009). This issue quickly disappeared after $F = 0.04$, though, mainly because the LEV grew to a size that began to pull fluid along the plate chord back towards the leading edge of the plate. The results of applying a threshold are shown in Fig. 4.5 with normalized circulation values ($\Gamma^* = \frac{\Gamma}{c\bar{U}_\infty}$) versus formation number. When the value of $N$ was first incremented by 0.20 from 0, the change in circulation was small suggesting that a majority of the ambient noise is within this range. When it was incremented from 0.20 to 0.40 the change in circulation was large. This result is similar to DeVoria et al. (2012); however, the difference in circulation as the formation number increases for the first few curves quickly diminishes. This is due to the fact that for this experiment, the
LEV grew to eventually cover most of the FOV, therefore the effect of noise on the LEV changed. The same effect can be seen in Wahidi et al. (2015). The main criteria used to determine the appropriate threshold were selected based on the following: (1) the vortex circulation must match the total circulation (or at least have the same slope) initially since they are one in the same and (2) the slope of the total circulation must not be substantially changed. The second criterion can be observed above in Fig. 4.5. It was found for all Reynolds number cases that $N = 1.2$ allowed for the most appropriate reduction in noise while still maintaining the vortex development and growth characteristics.

![Figure 4.5](image)

Figure 4.5. Total circulation curves for various threshold values show the effect of thresholds on vortex growth. The threshold chosen is shown as a red, dashed line.

4.4 UNCERTAINTY ANALYSIS

To determine the uncertainty for this experiment, an examination of bias and precision error were conducted. Conducting an experiment using PIV involves the manipulation of a complex system of machines, optics, and software, where each component must accurately
extract and transfer the data that is physically observed and transform it into electrical signals. Due to this complexity, scientists and engineers have devoted their skills to understanding this subject, and it is from that library of information that understanding is extracted and used for this experimental setup. Precision error, however, is simply related to the scatter of the data and is easily understood through the use of statistics. The methods for analyzing both types of errors are given below.

To accurately extract information from the two-dimensional flow field, the camera and lens, and computer processing software must work together to recreate the flow field of “frozen” particles electronically. Particle image velocimetry, in this case, was used to break down the flow field into small segments, or interrogation regions, and use the pattern created by the particles to track the displacement of that specific pattern. In this work a 50% overlap of interrogation regions was used, and Raffel et al. (2007) explains that twice-oversampled PIV data provides more accurate data due to larger grid spacing for the same velocity measurement uncertainty. It also allows the data to remain uncorrelated, especially when using differentiation schemes (Raffel et al., 2007). It is important for the velocity gradient within the interrogation region to be minimal so that the particle image pair can be accurately correlated, and Forliti et al. (2000) shows that if the particle image diameters are nominally 2 pixels, a three point Gaussian peak estimator is used in the cross correlation, and the ratio given in the following equation is less than one, then the bias error is ensured to be less than 0.01 pixels.

\[
\frac{M \delta t \Delta u}{\sqrt{d_e^2 + d_r^2}} < 1
\]  

(4.10)

In Eqn. 4.10, M is the magnification, \( \delta t \) is the time difference between an image pair, \( \Delta u \) is the velocity variation from the mean velocity contained in the interrogation region, \( d_e \) is the particle image diameter, and \( d_r \) is the pixel size on the camera’s CCD chip. Using the nominal case
studied, it was ensured that the ratio given in Eqn. 4.10 is less than one, which yields a bias error of $\varepsilon_b = 0.0739 \text{ mm s}^{-1}$.

To find the precision error, principles from Figliola and Beasely’s *Theory and Design for Mechanical Measurements* was used, and the precision error was calculated for each case by analyzing the standard deviation of the mean at each time step examined.

$$\varepsilon_p = \frac{t_{\nu,p}}{\sqrt{M}} \sqrt{\frac{1}{M-1} \sum_{i=1}^{M} (x_i - \bar{x})^2}$$

The error was calculated using Eqn. 4.11 where $M$ is the number of measurements, $\nu = M - 1$ is the degrees of freedom, and $t_{\nu,p}$ is the $t$ estimator. For this experiment, $\nu = 19$ and a probability of 95% was used, yielding $t_{19,95} = 2.093$. The velocity precision error found using the nominal case is $\varepsilon_b = 0.2636 \text{ mm s}^{-1}$. The errors are combined using Eqn. 4.12 yielding $\varepsilon_b = 0.2738 \text{ mm s}^{-1}$ which is presented as error bars in the circulation plots.

$$\varepsilon_u = \sqrt{\varepsilon_b + \varepsilon_p}$$

4.5 CIRCULATION

Initially, circulation was calculated by ensemble averaging the PIV velocity data for all 20 runs and then finding total circulation and vortex circulation. It was found, however, that this yielded a non-physics based circulation curve where at certain instances of time the circulation curve would suddenly jump. These jumps occurred during the period when the vortex began to separate from the shear layer. Additionally, in the middle and highest Reynolds number cases, it was common for the shear layer to shed a cluster of vorticity, and in many cases, this cluster would temporarily connect the vortex identified region back to the shear layer, therefore instantaneously increasing the vortex circulation. Once the cluster was absorbed into the vortex
and the connection between the vortex and shear layer lost, the vortex circulation would instantaneously decrease. Therefore, defining an objective characterization for when the LEV is no longer connected to the shear layer can be rather difficult, and so it was determined that the best method for calculating circulation was to do so for each individual run and then average the curve over 20 runs. This method yielded a much smoother curve which better represents the physics observed in the PIV data. Another problem with accurately calculating the vorticity was the limitation of the criterion in identifying points that contained vorticity of similar magnitude. The vorticity contour plots shown in Figs. 4.2 through 4.4 and the plot below in Fig. 4.6 demonstrate examples of this when comparing the vortex criterion identified locations on the plot with the vorticity contour beneath it. It was found that the $\lambda_{ci}$-Criterion performed best, but in addition to using this criterion for the identification of points related to the vortex, it was also found that the inclusion of supplementary, non-zero vorticity valued points near points identified by the criterion was required. After clustering the points, data up to seven grid-spacings away from the identified points still contained vorticity of similar magnitude to that of the vortex and so these points were included with the initial points when circulation was calculated.

To ensure that 20 runs provided sufficient data to converge the circulation, an additional 10 runs were collected for the smooth plate middle Reynolds number case. A comparison of 15, 20, and 30 run circulation curves for this case are presented below in Fig. 4.7.
Figure 4.6. Vortex identification ($\lambda_{ci}$) scatter plot with vorticity contour plot beneath. The exes mark grid points where the criterion identified vortex points.

Figure 4.7. Vortex circulation for the $Re_c = 3000$ case with smooth plate showing the convergence of data.
5. RESULTS

5.1 BASELINE: SMOOTH PLATE

For this study, three Reynolds numbers were examined where two surface types were used on the lower surface of a plate used to generate a LEV. Data was collected using PIV and was processed to yield velocity data for the flow field. One of the most interesting and useful derivations from velocity in the study of vortex formation is vorticity. By examining the vorticity that is generated by the plate and then collected into the LEV, an understanding can be developed about how parameters like Reynolds number and surface type can affect the LEV growth.

One important characterization about LEVs is the point during the formation where the vortex separates from the shear layer or the “pinch off” point. This location in time is most often identified as the moment when the growth rate of the vortex circulation, or the slope, begins to differ from that of the total circulation, which includes all of the vorticity generated by the plate within the domain. Also of interest is the saturation of the LEV, which is usually the point after separation when the vortex will no longer accept vorticity. The non-dimensionalized total circulation for every case studied is presented in Fig. 5.1 and it shows that experimental conditions remained consistent for all cases tested. This collapse of the total circulation curves onto one curve after non-dimensionalization is expected. For the lowest Reynolds number case, the total circulation does deviate slightly from the other curves towards the end of the experiment. This deviation from the other two cases is especially true at later formation numbers after the vortex has separated from the shear layer and is no longer increasing in strength. This
may be a product of lower Reynolds number effects, i.e. viscous forces provide slightly more damping and dissipation as compared to the other two cases; however the Reynolds number is still much larger than unity. Regardless, the collapse of the total circulation curves from each case is expected since the mechanism for generating vorticity is the separation of flow over the leading edge, which remained the same for each experimental case tested.

In all of the cases examined, the flow field beneath the leading edge, which is dominated by the LEV, caused the growth of a secondary vortex. The creation of a secondary vortex can be explained in the context of plate boundary layers and separation due to adverse pressure gradients. As the LEV forms, it grows and this growth is complemented by the movement of the LEV core away from the plate. The core travels in two directions relative to the plate, one being along the plate surface and the other normal to it. In all cases, the core moves to a position located further beneath the plate and away from the leading edge, to some degree. The combination of this movement and increase in vortex strength induces a flow along the plate chord, either towards the leading edge or away from it, depending on the location of the core.

Figure 5.1. Non-dimensional total circulation for all cases studied. Solid line represents the smooth case. Dotted line represents the grooved case.
relative to the plate leading edge, which is constantly changing. Fluid that is between the leading edge and core is forced to accelerate towards the leading edge by the vortex, however, the existence of the shear layer and the rotation of the vortex causes an adverse pressure gradient to form, and this consequently causes a region of separation between the shear layer, leading edge, and vortex. The vortex and shear layer worked together to pull fluid away from the plate, causing an increase in static pressure near the leading edge. The no slip condition on the plate surface creates vorticity of opposite sign to that of the main vortex and shear layer, and this, coupled with the rotation of the LEV in a viscous fluid, causes a secondary vortex to form. This secondary vortex causes yet another velocity gradient on the plate surface beneath it such that vorticity of the same sign as the main vortex is generated. The resolution of this setup allowed for a small amount of observation of this vorticity, but only enough to be examined qualitatively, rendering quantification out of the scope of this project. It should be noted that care was taken to mask this vorticity so that only the vorticity generated by the leading edge was accounted for in the calculation of total circulation. The creation and magnitude of this secondary vortex relative to each case is discussed in a section below.

The objective identification of the separation point is not easily done for the middle and highest Reynolds number cases, but for the lowest Reynolds number case, this point is fairly clear. Examining Fig. 5.2a, which is the smooth case circulation curve of both vortex circulation and total circulation for $Re_c = 1500$, it can be seen that the LEV separates around a formation number $F = 1.3$. The slopes for the circulation curves are initially very similar, although the magnitude is slightly different, enough that they are not within the error. The cause of this discrepancy is most likely related to the difficulty in correctly identifying grid points as part of the vortex, but it can be seen that this limitation does not greatly affect the data as the growth
Figure 5.2. Circulation and ensemble-average vorticity field for the smooth plate case at $Re_c = 1500$. a) $\Gamma_{vortex}$ and $\Gamma_{total}$ b) $F = 0.28$ c) $F = 1.01$ d) $F = 1.31$ e) $F = 1.61$
Figure 5.3. Circulation curves for the smooth plate case: a) $Re_c = 6000$ b) $Re_c = 3000$ c) All three Reynolds number of $\Gamma_{vortex}$
rates are nearly the same. The separation point is easily identified when the slopes depart and the LEV circulation begins to dwindle while the total circulation continues to increase and eventually reach a plateau region. This separation point is also evident in the contour plots of non-dimensional vorticity, \( \omega_z^* = \omega_z c / U_\infty \), which are shown in Fig. 5.2b-5.2e. The connection is clear between the shear layer and vortex during the initial formation period, \( F = 0.0-1.2 \), but after this point it becomes apparent that the vorticity between the shear layer and the vortex is becoming thin. Examining Fig. 5.3, three different types of indication that represent pinching off of the shear layer from the vortex can be seen for the three different Reynolds number cases. For \( Re_c = 6000 \), separation of the vortex circulation curve from the total circulation curve begins to occur at an early formation number, as early as \( F = 0.5 \), while for \( Re_c = 3000 \), disparity between the curves begins to be present around \( F = 0.7 \). Again, the magnitudes of circulation are slightly different between the vortex circulation and the total circulation, but the slopes are clearly the same. For \( Re_c = 1500 \), separation of vortex circulation from the total occurs with a relatively smooth transition while the higher Reynolds number cases contain a more abrupt and sudden change in slope.

For all three cases a secondary vortex is generated beneath the LEV next to the shear layer and this can be seen in Fig. 5.4, but this secondary vortex, in the average sense, appears to provide a different effect for each Reynolds number. The secondary vortex does not seem to have a great effect on the LEV for the lowest Reynolds number case at any point during the vortex formation, but for the middle Reynolds number case an unstable flow field seems to be generated between the secondary vortex and the shear layer starting at \( F = 0.7 \) which can be directly correlated to the circulation curves in Fig. 5.3a and 5.3b. This instability not only causes a semi-abrupt disconnection of the shear layer from the LEV, but it also changes the way
Vorticity is fed to the still growing vortex. At this point in the formation process the vortex circulation begins to be fed by clusters of vorticity which causes undulations in the vortex circulation curve. It should be noted that these undulations are not within the error; rather the error also moves up and down with the provided points indicating that this is mostly a feature.

Figure 5.4. Velocity magnitude contour plots showing the existence of a secondary vortex at $F = 2.08$. a) $Re_c = 1500$ b) $Re_c = 3000$ c) $Re_c = 6000$
and is not simply scatter. These small segments of vorticity are quickly absorbed by the LEV indicating that the vortex is not saturated, providing a difference between the definitions of separation and saturation. This same feature is observed in the highest Reynolds number case whereby the vortex separates from the leading edge shear layer at an early formation number but the clusters of relatively large magnitude vorticity that are continuously fed by the shear layer are absorbed into the main vortex indicating that it is still growing. The difference in this phenomenon between the middle and highest Reynolds number cases is the observed intensity as measured by the qualitative size of the clusters and the quantitative increase in circulation of the vortex. The larger intensity in the highest Reynolds number cases is what drives the separation at an early formation number, but rather than continue to intensely feed the LEV with clusters of vorticity, this effect begins to subside and the shear layer begins to shed vorticity in a more stable way. This is qualitatively different from the middle Reynolds number case where the instability phenomenon develops at a later formation number and then returns to a more stable flow field. This difference indicates that there is a parameter that drives the secondary vortex to becoming unstable, causing it to more violently interact with the shear layer until it can return to a lower energy state, so to speak. By comparing these many different characteristics of vortex formation between the two surface types, the effect of the patterned surface on the LEV growth can be determined.

5.2 EFFECTS OF SURFACE PATTERNING

Now that baseline cases have been established, discussions of the comparisons found between the surface types can be carried out. Starting with the lowest Reynolds number cases, curves for vortex circulation can be found in Fig. 5.5a where it can be seen that a negligible difference appears to exist between the two cases. This is especially true when considering the
error. Additionally, by examining the vorticity contour plots in Fig. 5.5b through 5.5e at the same formation number, $F = 1.4$, where separation has begun to take place, one will notice only slight differences between the two cases. In the vorticity contour for the smooth case there appears to be a slight higher magnitude of the vorticity in the vortex core, and the shear layer also appears to be slightly more attached than in the grooved case. These differences are unfortunately very small and as stated earlier, they fall within the error. This does suggest that there could be a true physical difference, and the discovery of this may come with an increased number of runs that would eliminate some of the precision error, but this would have to be investigated in future work. Based on these results, one would infer that the surface patterning has no effect on the vortex formation process when developed at $Re_c = 1500$, but it was shown above that the development process as compared to the other Reynolds number cases is substantially different. It should be noted that this result indicates that the surface patterning used here, which in this case has a Reynolds number of $Re_d = 3.98$, has essentially no effect on the vortex possibly due to the cavity Reynolds number used. It does not rule out the fact that a similar surface patterning of a different cavity Reynods number could alter the vortex formation.

For the cases where $Re_c = 3000$, distinct differences can be found in the vortex formation process. Unlike the lower Reynolds number cases, distinguished interactions exist between the LEV and the secondary vortex. As stated above, the secondary vortex is a product of the main vortex since the flow induced by the LEV along the plate, coupled with the adverse pressure gradient, is what generates the negative vorticity. It is hypothesized that chordwise grooves, through an addition of surface area, increase the drag on the vortex, meaning less energy will be contained in the rotating flow field due to frictional losses at the plate surface. It has been reported that for flow parallel to rows of cavities which have $1 < Re_d < 75$, the surface drag is
Figure 5.5. Vortex circulation for the smooth and grooved case at $Re_c = 1500$. Smooth case contour on the left and groove case contour on the right. $F = 0.43$ and 1.4 for both.
increased (Jones, 2011). In this experiment, the smooth surface provided a total surface area of 238 in\(^2\), whereas the chordwise grooves increased that surface area to around 664 in\(^2\). With the larger surface area and increased cavity Reynolds number it is expected that the vortex growth rate should be decreased for this orientation when the LEV is forming, that the circulation should be a smaller value, the vortex development should be altered, or a combination of all three. For the grooved case there is a potential for a greater amount of vorticity near the plate since the surface drag is increased, therefore creating more secondary vorticity which can increase the relative size of the secondary vortex and effectively push the LEV away from the plate; however since the secondary vorticity is derived from the induced velocity of the vortex, who’s growth could be stunted by the patterned surface, the opposite effect could be seen, whereby the production of secondary vorticity is decreased. It should also be noted that the secondary vortex, like the LEV, is a low pressure region, therefore any changes in circulation due to an increase or decrease in core velocity could influence the flow of fluid near the vortex. It is the former concept that seems to be suggested by the data, that is, a stronger secondary vortex is generated for the grooved case and this increased secondary vorticity affects the shear layer’s ability to convect positive vorticity to the LEV. Examining Fig. 5.6 it can be seen that there is less area under the vortex circulation curve, similar to the \(Re_c = 1500\) cases, implying that the circulation averaged for the entire formation process is marginally reduced for the grooved case, but just like the \(Re_c = 1500\) case, the error interval is slightly too large to support this argument. Initially, the growth rates of the smooth and grooved cases are identical, but at approximately \(F = 0.55\) the grooved case’s vortex circulation suddenly departs from the smooth cases’s growth rate as well as the total circulation. This sudden departure is in response to the growing instability at the plate leading edge caused by the secondary vortex, which temporarily interrupts the transfer of
vorticity. This further imbalance in an already unsteady process initiates and lasts until almost $F = 2.3$, reducing in frequency from when it first starts. A similar process is seen for the smooth case, but in this instance, both the circulation plot and vorticity contour plots agree that this process initiates at a later time, around $F = 0.70$, and its effects aren’t as drastic as the grooved case. Balance is also quickly restored to the formation process indicating that the secondary vortex is quickly controlled allowing more positive vorticity to be fed to the separated but unsaturated LEV. For the smooth case, cyclic shedding was found to start at approximately $F = 0.75$, although the shear layer began to show signs of shedding at $F = 0.60$ but did not completely disconnect from the LEV. Shedding frequencies started at 0.60 Hz and reduced to 0.28 Hz after 3 cycles. In the grooved case, well defined shear layer shedding began at $F = 0.55$, which can be seen in Fig. 5.6. The shedding frequency began at a slightly lower value as compared to the smooth case, approximately 0.44 Hz and reduced to 0.27 Hz.

In the highest Reynolds number cases where the cavity Reynolds number is $Re_d = 15.94$ for the groove model, several similarities concerning how the grooves affected the vortex formation were found between this case and the middle Reynolds number case. The secondary
vortex providing most of the material of interest. The total and vortex circulation curve for the smooth case were presented in Fig. 5.3a above and the grooved case is presented below in Fig. 5.8. Studying Figs. 5.3a and 5.7 it can be seen that the error, whose main contribution is precision error, is not of large magnitude relative to the curve features, yielding an average value of roughly 4.5 cm$^2$ s$^{-1}$ between the two cases (or non-dimensionally, 0.0788). It can also be seen in these figures that the error is actually smaller than the average along certain portions of the curve where the slope changes rapidly, which indicates that the “scatter” that is especially present in the grooved case is actually a behavior, e.g. a repeatable pattern. Similar to the middle Reynolds number cases, the shear layer reaches an unstable state and begins to shed instabilities due to interactions with the secondary vortex, but instead of building to that state of flow, the instabilities happen immediately. During this initial period, positive vorticity is delivered to the LEV in spatially small but large magnitude vortices or, as stated in DeVoria (2011), Helmholtz-like-instabilities. These can be seen in Figs. 5.10 and 5.11 as the clusters of high magnitude vorticity that are not attached to either the shear layer or the vortex. Wahidi et al. (2015) reported similar instabilities occurring early in the formation process for the $Re_c = 6000$, but due to vibration within the setup, much uncertainty shielded as to whether the instability shedding was a naturally occurring phenomenon or a product of an ill-damped experimental setup. The setup used here is a refined version of the one used in Wahidi et al. (2015) whereby utilization of a rail system to stabilize the plate as explained in Chapter 3 ensured that large amplitude, low frequency vibration would not be permitted. Additionally, the presence of the instability shedding at a later formation number in the middle Reynolds number cases studied here provides
Figure 5.7. Vorticity contour plots for $Re_c = 3000$ with the smooth case on the left and groove on the right. From top to bottom: $F = 0.56, 0.70$, and $1.9$. 
further proof that the phenomenon is a product of fluid-fluid interaction rather than vibration induced fluid-structure interaction. This behavior in the middle Reynolds number cases for Wahidi et al. (2015) was not observed, but that experiment studied larger angles of attack than the one here.

For both the grooved case and smooth case where $Re_c = 6000$, the secondary vortex interaction with the shear layer and LEV commences at the very beginning of the experiment. Studying the vorticity contour plots in Fig. 5.10 for the grooved case reveals that the interaction causes the secondary vortex to cyclically displace a lateral distance of approximately 5-8 mm at a frequency of roughly 2 Hz in its position just below the leading edge. This event occurs for 8 cycles, with each cycle reducing in frequency due to an apparent damping or stabilizing force within the flow field, and at $F = 1.2$ the behavior ceases. After this point, the flow is less unsteady and the secondary vortex stabilizes. This behavior is what causes the undulations seen in Fig. 5.9 as sudden increases in vortex circulation. As for the smooth plate case, the secondary vortex does not exhibit nearly the same magnitude of activity. During the initial formation period up to $F = 0.60$, which is shown in Fig. 5.10, the secondary vortex has a comparably reduced back-and-forth motion with a similar frequency, but unlike the grooved case, the instability is quickly damped and the vortex formation returns to a more linear process. Interaction between the shear layer and secondary vortex still persists, however, causing the shear layer to deliver vorticity to the LEV in clusters rather than a contiguous layer. Nonetheless, except for at the very beginning ($F = 0.0 – 0.30$), the secondary vortex appears to remain fairly stationary beneath the leading edge unlike in the grooved case where the cyclic motion extends for a period four times longer.
Limited temporal and spacial resolution inhibits the ability of the computer codes written to resolve the small features that occur during the initial formation period, which is obvious from the scatter present in Fig. 5.8a before $F = 0.20$. The time step chosen for processing the data that is presented here was every 10$^{th}$ image, which allows the observation of the flow field in increments of $F = 0.02$ or in increments of time equal to 0.42 s, 0.21 s, and 0.10 s for the lower, middle, and highest Reynolds number cases, respectively. A more in-depth analysis of the flow field phenomenon that is present during the initial formation period would require a smaller time
Figure 5.10. Vorticity contour plots for $Re_c = 6000$ with the smooth case on the left and groove on the right. From top to bottom: $F = 0.25$, 0.40, and 0.70.
step as well as higher resolution. For both cases the shear layer sheds vorticity at a certain frequency even after the rapid motion of the secondary vortex has subsided. It emits vorticity clusters initially at 0.7 Hz and then slows to 0.55 Hz before reaching a steadier state of releasing vorticity in the shear layer. The smooth case has frequencies that are slightly higher than the grooved cases during this process. For the smooth case, the steady state is reached by $F = 1.30$ while for the grooved case the shear layer continues to emit clusters until $F = 1.69$, which is similar in length of time to the behavior described above concerning the secondary vortex. This behavior is shown below in Fig. 5.11. Interestingly enough, these times coincide with the formation number where the vortex circulation reaches its peak value, and so for the grooved case this means that the vortex growth is slightly inhibited due to the fact that both cases reach roughly the same peak value of vortex circulation. After this point, the LEV circulation in both cases reaches a plateau region that has a slightly negative slope indicating that the vortex has saturated and will be soon detaching from the plate. The point of complete separation is unfortunately out of the scope of this experiment, but it could be examined in future studies. The slightly negative slope that is visible in each Reynolds number case studied is most likely not a physical behavior of the LEV; for each case a new TEV begins to form and this TEV pushes the LEV and some of the positive vorticity from the shear layer out of the domain. This is artificially causing the total circulation and vortex circulation to plateau, however, by this point the vortex has concluded most of its formation and development which is the main focus of this study.
Figure 5.11. Vorticity contour plots for $Re_c = 6000$ with the smooth case on the left and groove on the right. From top to bottom: $F = 0.82$, 1.40, and 1.55.
6. CONCLUSIONS

6.1 DISCUSSION

A translating two-dimensional plate at a 45° angle of attack was used to model a simplified version of butterfly flapping flight. During butterfly flapping flight the wing produces a LEV similar to the one generated in this experiment. Digital particle image velocimetry was used to study the LEV at the midspan of the plate. Vorticity was extracted from the DPIV velocities by using a local circulation method developed by DeVoria (2011). A vortex identification method was then used to track the vortex and cluster the points that comprised it as it moved through the image. From this procedure, the time-varying circulation was calculated. It was found that by exchanging the smooth lower surface of the plate for a patterned surface comprised of chordwise grooves that dynamically matched those found on the Monarch butterfly, the vortex development could be altered in various ways.

When this experiment was first considered, it was postulated that the patterned surface on a butterfly’s wings served an aerodynamic purpose. The hypothesis was that the grooves altered the vortex development, but it was unknown in what way this change would occur. From this study it has been found that much of the vortex development is manipulated by the alteration of the secondary vortex, which is created from the vorticity generated by the LEV on the plate surface. At the lowest Reynolds number studied a very small amount of quantitative difference between the two surface types was found in the circulation generated by the LEV, which was studied up to a formation number of 2.8. Examination of vorticity contour plots revealed that no significant qualitative differences exist during the formation of the LEV as well. However, for
the higher two Reynolds number cases studied, it was found that relevant differences exist between the smooth surface and grooved surface, with the greatest amount of variance occurring with the Reynolds number closest to the butterfly Reynolds number, that is \( Re_c = 6000 \). This trend in increasing effects of the grooves on the LEV as Reynolds number increase indicates that the groove geometry and cavity Reynolds number found on the butterfly’s wing are sized specifically for the butterfly’s flight regime.

In some cases the secondary vortex, whose creation and existence seemed to depend on the LEV itself, acted to disrupt the convection of vorticity from the leading edge shear layer to the LEV. An increase in secondary vortex activity led to a decrease in the growth rate of the LEV. For the middle Reynolds number this interaction only occurred after a certain formation number, and for the grooved case the effect began sooner but ended at the same formation number as the smooth case. By increasing the Reynolds number, it was found that the secondary vortex interaction began when the vortex was initially formed, but in the grooved case the effect of modifying the LEV growth rate was found to be larger while the LEV was still able to achieve the same peak value of circulation. This in effect extended the period during which the vortex developed.

The total amount of circulation developed by the LEV did not differ significantly between the two cases, which is expected since a reduction in the ability to generate lift would not be efficient for the butterfly. Instead, it was found that the patterned surface acted to extend the development of the LEV over a longer period of time. This ability to control the LEV growth rate suggests that the same lift can be generated for a longer period of time and for the butterfly this could mean a reduction in the number of flaps needed. A collaboration of the work conducted in this thesis has been the flight test of free flying Monarch butterflies (Cranford,
2015). It was found that after the removal of their scales, the butterflies’ flapping amplitude decreased by 7% and the flapping efficiency also decreased by 38% (Slegers, et al., 2016). A decrease in flapping amplitude for similar energetic flight indicates that the butterflies’ ability to generate lift from the LEV that its wings develop as it flaps and flies through the air was altered. Since LEVs have a tendency to grow and then separate from the generating surface, the ability to control the vortex growth and allow it to remain attached to the wing for a longer period of time is a huge advantage for flapping insects. The work conducted here suggests that cavities of an appropriate geometry can play an important role in controlling vortex growth, which may allow the butterfly to flap less for the same amount of lift generated by its wings.

6.2 FUTURE WORK

Many aspects of this experiment could be improved upon, and some of the conclusions reached by this study also open the door for more investigations of vortex behavior. This experiment used nominal square grooves that were dynamically scaled to match those found on a butterfly wing and they successfully demonstrated that vortex manipulation during development could be achieved by use of a surface patterning. To further understand how much exactly the surface patterning can alter the vortex growth a more refined groove model could be built that more accurately represents the geometries found on the butterfly’s wing. Three-dimensional printing technology is on the rise, and the use of this technology to print smaller and more accurate grooves would be extremely beneficial.

Additionally, one of the difficulties associated with this experiment was the sheer size of the models being tested. The equipment limits were reached in order to be able to see the flow field beneath the plate, which only amounted to a width of roughly 5/6 of the chord. A smaller model would be highly beneficial as it would then be possible to study the entire flow field
beneath the surface simultaneously. A smaller model would also allow a more global view which would include the leading and trailing edge vortex within the same image. The effect of the surface pattern on the TEV or the effect of the TEV on the LEV could also be better understood.

The realm of flapping kinematics could also be explored with a plate model that pitches and/or translates using different velocity programs while also utilizing a plate that more accurately represents a particular insect’s wing. The size of the model, and hence the size of the vortex, limited the formation number achievable by the setup. Because of this, the formation number for which a vortex is theorized to be fully formed was out of reach for this study, so by reducing the model to a more manageable size, more of the vortex development can be studied. Finally, some researchers in the literature were able to make use of filtering techniques to create a smoother circulation curve. A better understanding and application of different filtering techniques could highly enrich the quantitative data from the experiment.
REFERENCES


APPENDIX A: SUPPLEMENTAL VORTICITY FIELDS AND CIRCULATION PLOTS

On the following pages contour plot sequences have been provided for each case studied using every 100\textsuperscript{th} image or nine multiples of $F = 0.19$. These plots are provided to increase clarity and to supply the reader with a more general picture of the vortex formation during a consistent sequence. The plots do not show the entire formation process, i.e. up to $F = 2.8$, but most of the important events occur before $F = 1.69$. The plots for each case are sequenced left-to-right, then top-to-bottom. They begin on the next page.
Figure A.1. Smooth $Re_c = 1500$. $F = 0.19, 0.37, 0.56, 0.75, 0.94, 1.12, 1.31, 1.50, 1.69$
Figure A.2. Groove $Re_c = 1500$. $F = 0.19, 0.37, 0.56, 0.75, 0.94, 1.12, 1.31, 1.50, 1.69$
Figure A.3. Smooth $Re_c = 3000$. $F = 0.19, 0.37, 0.56, 0.75, 0.94, 1.12, 1.31, 1.50, 1.69$
Figure A.4. Groove Re = 3000. F = 0.19, 0.37, 0.56, 0.75, 0.94, 1.12, 1.31, 1.50, 1.69
Figure A.5. Smooth $Re_c = 6000$. $F = 0.19, 0.37, 0.56, 0.75, 0.94, 1.12, 1.31, 1.50, 1.69$
Figure A.6. Groove $Re_c = 6000$. $F = 0.19, 0.37, 0.56, 0.75, 0.94, 1.12, 1.31, 1.50, 1.69$
APPENDIX B: MATLAB CODES

B.1: PHASE AVERAGING CODE

clear;
clc;
%
Creator: Jacob A. Wilroy. Build Date: June 2014
Purpose: Load .vec (or other forms of .txt) files for the averaging of vector data.
Instructions: Below are the paths where the files can be found, and file numbers. These need to be changed depending on file location.
%
%% Settings
SetNum = 1:1:20; % Set numbers to be uploaded
ImageNum = [1:1:49;50:25:1500]; % Images to be processed
% ImageNum = 10:10:1500; % Images to be processed
% Read Path - Delete # from end
A='I:\JW_FlappingPlate\Groove_45AoA_2Cam\32mmps\Top\R2\Analysis';
% Write Path - Delete # from end
B='I:\JW_FlappingPlate\Groove_45AoA_2Cam\32mmps\Top\R2\Averaged Files';

%% Create File Names for Reading
NoI = length(ImageNum); NoS = length(SetNum);
ReadPathnames = cell(NoS,1); WritePathnames = cell(NoS,1);
InFilenames = cell(NoI,1); OutFilenames = cell(NoI,1);
fprintf('Generating File Names...'

for n=1:NoS
    Num = sprintf(' %d',SetNum(n));
    RPathname = strcat(A,Num); WPPathname = B;
    ReadPathnames{n,1} = RPathname; WritePathnames{n,1} = WPPathname;
end

for n=1:NoI
    N = ImageNum(n);
    str2 = sprintf('img%04d.L.vec', N); str3 = sprintf('img%04d_AVG.dat', N);
    InFilenames{n,1} = str2; OutFilenames{n,1} = str3;
end

fprintf('Complete\n'

%% Read and Store Data for Calculations
Storage = cell(NoI,NoS); FileText = cell(NoI,NoS);
fprintf('Reading Files...
'

for m=1:NoS
    cd(ReadPathnames{m,1});
    for k=1:NoI
        fileID = fopen(InFilenames{k,1});
        if fileID == -1
            fprintf('Error at file %s\n', InFilenames{k,1});
            error('Error: File does not exist');
        end
        FileText{k,m} = fgetl(fileID);
    end
end

if m==1 & & k==1
str = FileText{1,1};
[mstart,mend]=regexp(str,'I=.*?(?=,)', 'start', 'end');
I=str2double(str(mstart+2:mend));
[mstart,mend]=regexp(str,'J=.*?(?=,)', 'start', 'end');
J=str2double(str(mstart+2:mend));
[~,mend]=regexp(str,'SourceImageWidth="', 'start', 'end');
ImgW=str2double(str(mend+1:mend+4));
[~,mend]=regexp(str,'SourceImageHeight="', 'start', 'end');
ImgH=str2double(str(mend+1:mend+4));
[Xppx,mend]=regexp(str,'MicrometersPerPixelX="', 'start', 'end');
Xppx=str2double(str(mend+1:mend+9));
[Xppy,mend]=regexp(str,'MicrometersPerPixelY="', 'start', 'end');
Xppy=str2double(str(mend+1:mend+9));
dx=(Xppx/1000)*ImgW/I; dy=(Xppy/1000)*ImgH/J;
[msstart,mend] = regexp(str,'MicrosecondsPerDeltaT="', 'start', 'end');
deltaT=(str2double(str(mend+1:mend+10)))/1000000;
end

FileNumbersCell = textscan(fileID, '%f', 5*I*J, 'delimiter', ',');
fclose(fileID);

FileNumbersMatrix = cell2mat(FileNumbersCell);
MatrixXYUVchc = reshape(FileNumbersMatrix,[5,I*J]);
Storage{k,m} = transpose(MatrixXYUVchc);
end
fprintf('\t[Folder %d of %d Read]n', m, NoS)
end
clear FileNumbersCell FileNumbersMatrix MatrixXYUVchc;

fprintf('...Complete\n')

%% Calculate U and V Average Velocities
TransferMatrix = Storage{1,1};
X = TransferMatrix(:,1); X = transpose(reshape(X,[I,J]));
Y = TransferMatrix(:,2); Y = transpose(reshape(Y,[I,J]));
clear TransferMatrix; AvgStorage1 = cell(NoI,1);
fprintf('Averaging Velocities...')

for Phase=1:NoI
    DM = zeros(I,J,NoS,3); %DataMatrix
    for Set=1:NoS
        U = Storage{Phase,Set}(:,3); U = transpose(reshape(U,[I,J]));
        V = Storage{Phase,Set}(:,4); V = transpose(reshape(V,[I,J]));
        CHC = Storage{Phase,Set}(:,5); CHC = transpose(reshape(CHC,[I,J]));
        DM(:,:,Set,1) = U;
        DM(:,:,Set,2) = V;
        DM(:,:,Set,3) = CHC;
        clear U V CHC;
    end
    %Checking each (i,j) for negative CHC's and marking (u,v) zeros
    for Set=1:NoS
        for i=1:I
            for j=1:J
                if DM(j,i,Set,3)<0; DM(j,i,Set,1)=0; DM(j,i,Set,2)=0; end
            end
        end
    end
end
DM(:,,:,3) = zeros(I,J,NoS); %HC no longer needed

%Averaging the sets for one phase
DMAvg = zeros(I,J,3);
for i=1:I
    for j=1:J
        USet(:,1) = DM(j,i,:,1); VSet(:,1) = DM(j,i,:,2);
        DMAvg(j,i,1) = sum(USet)/nnz(USet);
        DMAvg(j,i,2) = sum(VSet)/nnz(VSet);
        if isnan(DMAvg(j,i,1)) == 1; DMAvg(j,i,1) = 0; end  
        if isnan(DMAvg(j,i,2)) == 1; DMAvg(j,i,2) = 0; end  
        DMAvg(j,i,3) = sqrt(DMAvg(j,i,1).^2+DMAvg(j,i,2).^2); %Velocity Mag.
    end
end

%Entire data set (all images) is stored in DM. Each DM (data set) is then
%placed in AvgStorage
AvgStorage1{Phase,1} = DMAvg*1000; %U,V originally m/s --> *1000 --> mm/s
clear DMAvg DM USet VSet;
end

clear StorageCell; fprintf('Complete
'
)

%% Vorticity
fprintf('Calculating Vorticity...'
)
AvgStorage2 = cell(NoI,1);
for Phase=1:NoI
    DMAvg = AvgStorage1{Phase}; U = DMAvg(:,1); V = DMAvg(:,2);
    wz = fcn_LocalCirculation(U,V,I,J,dx);
    wz(1,1:J) = 0; wz(I,1:J) = 0; %top and bottom of the image
    wz(1:I,1) = 0; wz(1:I,J) = 0; %left and right side of the image
    DMAvg(:,:,4) = wz; AvgStorage2{Phase,1} = DMAvg;
    clear DMAvg wz U V dUdy dVdx;
end

clear AvgStorage1; fprintf('Complete
'
)

%% Reformat Files
fprintf('Reformatting Data...'
)
FinalStorage = cell(NoI,1);
for Phase=1:NoI %Rewrite data matrix into original format
    Temp = zeros(I*J,6); n=1; DMAvg = AvgStorage2{Phase,1};
    for j=1:J
        for i=1:I
            Temp(n,1) = X(j,i); %X
            Temp(n,2) = Y(j,i); %Y
            Temp(n,3) = DMAvg(j,i,1); %U
            Temp(n,4) = DMAvg(j,i,2); %V
            Temp(n,5) = DMAvg(j,i,3); %Velocity Magnitude
            Temp(n,6) = DMAvg(j,i,4); %Vorticity
            n = n+1;
        end
    end
    FinalStorage{Phase,1} = Temp;
    clear Temp DMAvg;
end
clear AvgStorage2; fprintf('Complete\n')

%% Save Average Velocity Files
fprintf('Saving Files...')
cd(WritePathnames{1,1}); %Write to file
for i=1:NoI
    fileID = fopen(OutFilenames{i,1},'wt'); MatrixWrite = FinalStorage{i,1};
    temp=strcat('VAR
ABLES= "X mm", "Y mm", "U mm/s", "V mm/s", 
"Velocity Magnitude", "Z Vorticity", 
" DATASETAUXDATA Common.Density="0.000001", 
" DATASETAUXDATA Common.Incompressible="TRUE", ZONE T="T1", I=",
num2str(I),', J=',num2str(J),'", F=POINT');
    fprintf(fileID,'%s
%d
',temp);
    fclose(fileID);
    dlmwrite(OutFilenames{i,1}, MatrixWrite, '-append', 'newline', 'pc');
    clear MatrixWrite;
end
fprintf('Complete\n')
load handel.mat; sound(y, 2*Fs);

B.2: DOMAIN ASSEMBLY AND VORTEX TRACKING CODE

clear; clc;
{%
Creator: Jacob Wilroy
Build Date: June 2016
Purpose: Used to assemble a domain created with multiple cameras.
Use this to read averaged files (must already include vorticity) and then
either plot contours or write the combined/global data to file for opening
in Tecplot.
%
% Settings
% A = 10:10:1500; ImageNum = {A; A;};
% A = 1:1:50; ImageNum = {A; A;};
% A = 10:10:50; B = 75:25:1500; C = [A,B]; ImageNum = {C; C;};
% A = 25:25:500; ImageNum = {A; A;};
SetNum = [1;2;];
% [Set #, X offset (mm), Y offset (mm) (down)]
XYOff = [1,0,0;
       2,0,121.35;
       2,0,120.61]; %Groove
% 2,0,120.61]; %Smooth
NonDimThreshold = 1.2;
set(0,'defaultfigureposition',[50 50 1500 925])
SavePlotToFile = 'Y'; Plot = 'Y';
quiverScale = 3; quiverColor = 'w'; quiverLineWidth = 0.5;
SaveLocation = ...
'I:\JW_FlappingPlate\Groove_45AoA_2Cam\32mmps\Global Averaged Data 2';
%'VelMag' 'Vorticity' 'Q_Criterion' 'Track_Vortex' 'Delta2_Criterion'
%'WriteDataFile'
Run = 'WriteDataFile';
% Read Files
NoS = length(SetNum); NoI = length(ImageNum{1,1});
InputFilename = cell(NoI,NoS); First = ImageNum{1,1}(1,1);
Step = (ImageNum{1,1}(1,2) - ImageNum{1,1}(1,1));
Last = ImageNum{NoS,1}(1,length(ImageNum{NoS,1}));
OutFilenames = cell(NoI,1);

ReadPathNameCell{1,1} = ...
    'I:\JW_FlappingPlate\Groove_45AoA_2Cam\32mmps\Top\Averaged Files';
ReadPathNameCell{2,1} = ...
    'I:\JW_FlappingPlate\Groove_45AoA_2Cam\32mmps\Bottom\Averaged Files';

ReadPathNameCell{1,1} = ...
    'I:\JW_FlappingPlate\Smooth_45AoA_2Cam\16mmps\Top\Averaged Files';
ReadPathNameCell{2,1} = ...
    'I:\JW_FlappingPlate\Smooth_45AoA_2Cam\16mmps\Bottom\Averaged Files';

fprintf('Running ''%s'' code for images %g through %g.
',Run,First,Last)
fprintf('Non-dimensional threshold selected: %g.
',NondimThreshold)
fprintf('Plotting: %s. Save plots to file: %s
',Plot,SavePlotToFile);
fprintf('Plot save location: %s
',SaveLocation);

for m=1:NoS
    for n=1:NoI
        N = ImageNum{m,1}(1,n);
        str1 = sprintf('img%04d_AVG.dat', N);
        str2 = sprintf('img%04d_GBLAVG.dat', N);
        InputFilename{n,m} = str1;
        OutFilenames{n,m} = str2;
    end
end

fprintf('File Reading in Progress........................\n');
StorageCell = cell(NoI,NoS);
for m=1:NoS
    cd(ReadPathNameCell{m,1});
    for k=1:NoI
        fileID = fopen(InputFilename{k,m});
        if fileID == -1
            fprintf('Error at file %s\n', InputFilename{k,m});
            error('Error: File does not exist');
        end
        FileText{k,m} = fgetl(fileID);
        if m == 1 && k == 1
            str = FileText{1,1};
            [matchstart,matchend] = regexp(str, 'I=.*?(?=,)' , 'start', 'end');
            I= str2double(str(matchstart+2:matchend));
            [matchstart,matchend] = regexp(str, 'J=.*?(?=,)' , 'start', 'end');
            J= str2double(str(matchstart+2:matchend));
        end
        FileNumbersCell = textscan(fileID, '%f', 6*I*J, 'Delimiter', ',', ' ');
        fclose(fileID);

        FileNumbersMatrix = cell2mat(FileNumbersCell);
        MatrixXYUVMagWz = reshape(FileNumbersMatrix,[6,I*J]);
        StorageCell{k,m} = transpose(MatrixXYUVMagWz);
        dx = StorageCell{1,1}(1,1); dy = dx;
        fprintf('%t[File %d of %d Read]\n', k, NumofImages)
    end
    fprintf('[Stage %d of %d Read]\n', m, NoS)
end
fprintf('..............................Complete\n');

% Matrix Formatting
fprintf('Matrix Formatting in Progress...................
');
X = StorageCell{1,1}(:,1); Y = StorageCell{1,1}(:,2);
X = reshape(X,[I,J]); X = transpose(X);
Y = reshape(Y,[I,J]); Y = transpose(Y);
DM = zeros(I,J,6,NoI,NoS);
for m=1:NoS
    for n=1:NoI
        %Format data into IxJ matrices
        U(:,1) = StorageCell{n,m}(:,3); V(:,1) = StorageCell{n,m}(:,4);
        VelMag(:,1) = StorageCell{n,m}(:,5);
        Wz(:,1) = StorageCell{n,m}(:,6);
        U = reshape(U,[I,J]); U = transpose(U);
        V = reshape(V,[I,J]); V = transpose(V);
        VelMag = reshape(VelMag,[I,J]); VelMag = transpose(VelMag);
        Wz = reshape(Wz,[I,J]); Wz = transpose(Wz);
        %Delete values along left and right borders
        U(1:I,1:2) = 0; U(1:I,J-1:J) = 0;
        V(1:I,1:2) = 0; V(1:I,J-1:J) = 0;
        VelMag(1:I,1:2) = 0; VelMag(1:I,J-1:J) = 0;
        Wz(1:I,1:2) = 0; Wz(1:I,J-1:J) = 0;
        %Delete values along top and bottom borders
        if m == 1 %Top camera
            U(1:2,1:J) = 0; U(I-3:I,1:J) = 0;
            V(1:2,1:J) = 0; V(I-3:I,1:J) = 0;
            VelMag(1:7,1:J) = 0; VelMag(1:7,1:J) = 0;
            Wz(1:7,1:J) = 0; Wz(I-3:I,1:J) = 0;
        elseif m == 2 %Bottom camera
            U(1:4,1:J) = 0; U(I-1:I,1:J) = 0;
            V(1:4,1:J) = 0; V(I-1:I,1:J) = 0;
            VelMag(1:4,1:J) = 0; VelMag(I-1:I,1:J) = 0;
            Wz(1:4,1:J) = 0; Wz(I-1:I,1:J) = 0;
        end
        DM(:,:,1,n,m) = U;
        DM(:,:,2,n,m) = V;
        DM(:,:,3,n,m) = VelMag;
        DM(:,:,4,n,m) = Wz;
    end
end
fprintf('Complete
');

%% Global Matrix Creation
fprintf('Global Matrix Meshing in Progress...............');
J2 = J+round(abs(sum(XYOff(:,2))/dx));
I2 = I+round(sum(XYOff(:,3))/dy);
ImagesTotDom = First:Step:Last;
ImagesTotDom = ImageNum{1,1};
[XGBl,YGBl] = meshgrid(dx:dx:(J2*dx),dy:dy:(I2*dy));
YGBl = -1*YGBl;

istart(1) = 1; jstart(1) = 1;
iend(1) = I; jend(1) = J;

if strcmp(Run,'VelMag') == 1 || strcmp(Run,'Track_Vortex') == 1
    UGBl = fcn_Globalizer(DM(:,:,1,:,:),J2,I2,...
        ImagesTotDom,NoS,istart,iend,jstart,jend);
    VGBl = fcn_Globalizer(DM(:,:,2,:,:),J2,I2,...
        ImagesTotDom,NoS,istart,iend,jstart,jend);
    VMGBl = fcn_Globalizer(DM(:,:,3,:,:),J2,I2,...
        ImagesTotDom,NoS,istart,iend,jstart,jend);
end
if strcmp(Run,'Vorticity') == 1 || strcmp(Run,'Track_Vortex') == 1
    [WzGbl] = fcn_Globalizer(DM(:,:,4,:,:),J2,I2,...
    ImagesTotDom,NoS,istart,iend,jstart,jend);
end
if strcmp(Run,'WriteDataFile') == 1
    [WzGbl] = fcn_Globalizer(DM(:,:,4,:,:),J2,I2,...
    ImagesTotDom,NoS,istart,iend,jstart,jend);
    WzGbl = (WzGbl*177.8)/32;
    [UGbl] = fcn_Globalizer(DM(:,:,1,:,:),J2,I2,...
    ImagesTotDom,NoS,istart,iend,jstart,jend);
    [VGbl] = fcn_Globalizer(DM(:,:,2,:,:),J2,I2,...
    ImagesTotDom,NoS,istart,iend,jstart,jend);
    [VMGbl] = fcn_Globalizer(DM(:,:,3,:,:),J2,I2,...
    ImagesTotDom,NoS,istart,iend,jstart,jend);
    VortexCritGbl = zeros(I2,J2,NoI);
    for n=1:NoI
        In1 = UGbl(:,:,n); In2 = VGbl(:,:,n);
        VortexCritGbl(:,:,n) = fcn_LambdaCICriterion(In1,In2,I2,J2,dx,dy,0.01);
    end
end
fprintf('Complete\n');
%% Vorticity Transfer
% Transfer only positive vorticity above the threshold to the new matrix
if strcmp(Run,'Vorticity') == 1  || strcmp(Run,'Track_Vortex') == 1
    fprintf('Vorticity Value Transfer in Progress............');
    PosAvgVort = zeros(NoI,1); PosWzGbl = zeros(I2,J2,NoI);
    for n=1:NoI
        Temp = zeros(I2,J2,NoI);
        for i=1:I2
            for j=1:J2
                if WzGbl(i,j,n) > 0
                    Temp(i,j,n) = WzGbl(i,j,n);
                end
            end
        end
        PosAvgVort(n)=sum(sum(Temp(:,:,n)))/nnz(Temp(:,:,n));
        clear Temp;
    end
end
fprintf('Complete\n');
%% Calculate Vortex Criterion
TF = 'False';
if strcmp(Run,'Q_Criterion') == 1;      TF = 'True'; end
if strcmp(Run,'Delta2_Criterion') == 1; TF = 'True'; end
if strcmp(Run,'Track_Vortex') == 1;     TF = 'True'; end
if strcmp(TF,'True') == 1
    fprintf('Vortex Identification Calculation in Progress...');
    VortexCritGbl = zeros(I2,J2,NoI);
    if strcmp(Run,'Q_Criterion') == 1 || strcmp(Run,'Track_Vortex') == 1
        xQ = zeros(I2*J2,NoI); yQ = xQ;
        for n=1:NoI
            In1 = UGbl(:,:,n); In2 = VGbl(:,:,n); Count = 0;
            VortexCritGbl(:,:,n) = fcn_LambdaCICriterion(In1,In2,I2,J2,dx,dy,0.01);
for i=1:I2
    for j=1:J2
        Count = Count+1;
        if VortexCritGbl(i,j,n) > 0 && PosWzGbl(i,j,n) ~= 0
            xQ(Count,n) = j*dx; yQ(Count,n) = i*dy;
        end
    end
end end

if strcmp(Run, 'Delta2_Criterion') == 1
    for n=1:NoI
        In1 = UGbl(:,:,n); In2 = VGbl(:,:,n);
        VortexCritGbl(:,:,n) = fcn_DeltaCriterion2(In1,In2,I2,J2,dx,dy);
    end
end fprintf('Complete\n');
end

%% Vortex Tracking
if strcmp(Run, 'Track_Vortex') == 1;
    fprintf('Vortex Tracking in Progress....................\n');
    UMat = UGbl; VMat = VGbl; Wz = PosWzGbl; VelMag = VMGbl;
    I = I2; J = J2;
    PackageDeal = [I,J,NoI,dx,dy];

    [Circulation(:,1),VortexTracking, Zone1,CircCoordinates1] = ...
        fcn_VortexCirculation(UMat,VMat,Wz,PackageDeal,VelMag);
    [Circulation(:,2),~,Zone2,CircCoordinates2] = ...
        fcn_TotalCirculation(UMat,VMat,Wz,PackageDeal,VelMag,ImagesTotDom);
    fprintf('.................................Complete\n');
end

%% Reformat Files
if strcmp(Run, 'WriteDataFile') == 1
    fprintf('Reformatting Data...............................\n');
    FinalStorage = cell(NoI,1);
    for Phase=1:NoI
        %Rewrite data matrix into original format
        Temp = zeros(I2*J2,3); n=1;
        for i=1:I2
            for j=1:J2
                Temp(n,1) = XGbl(i,j); %X
                Temp(n,2) = YGbl(i,j); %Y
                Temp(n,3) = UGbl(i,j,Phase); %U
                Temp(n,4) = VGbl(i,j,Phase); %V
                Temp(n,5) = VMGbl(i,j,Phase); %Velocity Magnitude
                Temp(n,6) = WzGbl(i,j,Phase); %Vorticity
                Temp(n,7) = VortexCritGbl(i,j,Phase); %Lambda CI
                n = n+1;
            end
        end
        FinalStorage{Phase,1} = Temp;
        clear Temp;
    end
    fprintf('Complete\n');
    % Write to File
    fprintf('Writing Data to File.............................\n');
    cd(SaveLocation);
    Name = 'Global Averaged Data';
    [status,message,messageID] = mkdir(Name);
    if strcmp(message, 'Directory already exists.') == 1
n1 = 1;
while strcmp(message,'Directory already exists.') == 1
    NewName = strcat(Name,sprintf('%d',n1));
    [status,message,messageID] = mkdir(NewName);
    n1 = n1+1;
end
fprintf('
Specified folder already exists.
')
fprintf('New folder: %s.
',NewName)
cd(NewName);
else
    cd(Name);
end

for n=1:NoI
    fileID = fopen(OutFilenames{n,1},'wt');
    MatrixWrite = FinalStorage{n,1};
    temp=strcat('VARIABLES= "X mm", "Y mm", "U mm/s", "V mm/s",',',
        '"Velocity Magnitude","V",
        '"DATASETAUXDATA Common.Density="0.000001",
        '"DATASETAUXDATA Common.Incompressible="TRUE", ZONE T="T1", I=',
        'num2str(J2),', J=',num2str(I2),', F=POINT');
    fprintf(fileID, '%s
%d
', temp);
    fclose(fileID);
    dlmwrite(OutFilenames{n,1}, MatrixWrite, '-append', 'newline', 'pc');
    % fprintf(fileID, MatrixWrite, 'double');
    % fclose(fileID);
    clear MatrixWrite;
end
fprintf('Complete
');
return;
end

%% Plotting
if strcmp(Plot,'Y') == 1
    figure('visible','off');
    [XGblSkip] = fcn_DataSkip(XGbl,size(XGbl,1),size(XGbl,2));
    [YGblSkip] = fcn_DataSkip(YGbl,size(YGbl,1),size(YGbl,2));
    for n=1:length(ImagesTotDom)
        Num = sprintf('%d',ImagesTotDom(n));
        handle = figure(n);
        if strcmp(Run,'VelMag') == 1
            HiValue = 16; % 16, 20, 60
            Name = 'Global_VelMag'; PlotName = strcat('VelMag', Num);
            handle = figure(1); set(handle,'NextPlot','replace')
            [UgblSkip] = fcn_DataSkip(UGbl(:,:,n),size(UGbl(:,:,n),1),size(UGbl(:,:,n),2));
            [VgblSkip] = fcn_DataSkip(VGbl(:,:,n),size(VGbl(:,:,n),1),size(VGbl(:,:,n),2));
            contourf(XGbl,YGbl,VMGbl(:,:,n), 'LevelList',0:4:HiValue,...
                'LineColor','none'); grid on; colorbar; hold on;
            quiver(XGblSkip,YGblSkip,UGblSkip,VGblSkip,...
                quiverScale,'Color',quiverColor, 'LineWidth',quiverLineWidth);
title(strcat('Velocity Magnitude', Num));
xlim([dx, J2*dx]); ylim([I2*-dy, -dy]);
set(gca, 'XTick', 0:10:round(J2*dx+10));
set(gca, 'YTick', -round((J2*dx)/10)*10:10:0);
xlabel('x [mm]'); ylabel('y [mm]')
end

%%%%%%%%%%---VORTICITY---%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if strcmp(Run, 'Vorticity') == 1
    HiValue = 10; Step = HiValue/15;
    Name = 'Global Vorticity'; PlotName = strcat('Vorticity', Num);
    handle = figure(1); set(handle, 'NextPlot', 'replace');
    contourf(XGbl, YGbl, WzGbl(:,:,n), 'LevelList', 0:Step:HiValue,...
             'LineColor', 'none'); grid on; colorbar; hold on;
    title(strcat('Vorticity', Num));
xlim([dx, J2*dx]); ylim([I2*-dy, -dy]);
    set(gca, 'XTick', 0:10:round(J2*dx+10));
    set(gca, 'YTick', -round((J2*dy)/10)*10:10:0);
    xlabel('x [mm]'); ylabel('y [mm]')
end

%%%%%%%%%%---Q Criterion---%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if strcmp(Run, 'Q_Criterion') == 1
    HiValue = 2; Step = HiValue/20;
    Name = 'Global Q Criterion'; PlotName = strcat('Q Criterion', Num);
    handle = figure(1); set(handle, 'NextPlot', 'replace');
    contourf(XGbl, YGbl, VortexCritGbl(:,:,n), 'LevelList', 0.01:Step:HiValue,...
             'LineColor', 'none'); grid on; colorbar; hold on;
    title(strcat('Q Criterion', Num));
xlim([dx, J2*dx]); ylim([I2*-dy, -dy]);
    set(gca, 'XTick', 0:10:round(J2*dx+10));
    set(gca, 'YTick', -round((J2*dy)/10)*10:10:0);
    xlabel('x [mm]'); ylabel('y [mm]')
end

%%%%%%%%%%---Delta2 Criterion---%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if strcmp(Run, 'Delta2_Criterion') == 1
    HiValue = 10; Step = HiValue/20;
    Name = 'Global Delta2 Criterion'; PlotName = strcat('Delta2 Criterion', Num);
    handle = figure(1); set(handle, 'NextPlot', 'replace');
    contourf(XGbl, YGbl, VortexCritGbl(:,:,n), 'LevelList', 0.1:Step:HiValue,...
             'LineColor', 'none'); grid on; colorbar; hold on;
    title(strcat('Delta2 Criterion', Num));
xlim([dx, J2*dx]); ylim([I2*-dy, -dy]);
    set(gca, 'XTick', 0:10:round(J2*dx+10));
    set(gca, 'YTick', -round((J2*dy)/10)*10:10:0);
    xlabel('x [mm]'); ylabel('y [mm]')
end

%%%%%%%%%%---Track Vortex---%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if strcmp(Run, 'Track_Vortex') == 1
    figure(1); Name = 'Global Vorticity_Total and Vortex';
    PlotName = strcat('Vorticity_Total and Vortex', Num);
    HiValue = 5; Step = HiValue/20;
    Zone1(1,1,n) = HiValue; Zone2(1,1,n) = HiValue; PosWzGbl(1,1,n) = HiValue;

    handle1 = subplot(2,2,1); %VORTEX CIRCULATION CONTOUR
    contourf(XGbl, YGbl, Zone1(:,:,n), 'LevelList', 0.0001:Step:HiValue,...
             'LineColor', 'none'); colorbar; grid on; hold on;
    plot(CircCoordinates1(1,:,n), CircCoordinates1(2,:,n), 'r')
    title(strcat('\omega_z Vortex', Num));
xlim([dx, J2*dx]); ylim([I2*-dy, -dy]);
    set(gca, 'XTick', 0:15:round(J2*dx+10));
    set(gca, 'YTick', -round((J2*dy)/10)*10:10:0);
    xlabel('x [mm]'); ylabel('y [mm]')
end
handle2 = subplot(2,2,2);  %TOTAL CIRCULATION CONTOUR
contourf(XGbl,YGbl,Zone2(:,:,n),'LevelList',0.0001:Step:HiValue,...
 'LineColor','none'); colorbar; grid on; hold on;
plot(CircCoordinates2(:,1,:),CircCoordinates2(:,2,:), 'r');
title(strcat('\omega_{z} Total',Num));
xlim([dx,J2*dx]); ylim([I2*-dy,-dy]);
set(gca,'XTick',0:15:round(J2*dx+10));
set(gca,'YTick',-round((I2*dy)/10)*10:15:0);
xlabel('x [mm]'); ylabel('y [mm]')

% handle3 = subplot(2,2,3);  %VORTEX CRITERION CONTOUR
HiValue = 5; Step = HiValue/20;
contourf(XGbl,YGbl,VortexCritGbl(:,:,n),'LevelList',0.1:Step:HiValue,...
 'LineColor','none');colorbar;grid on;hold on;
title(strcat('Delta2 Criterion_{2D}',Num));
xlim([dx,J2*dx]); ylim([I2*-dy,-dy]);
set(gca,'XTick',0:15:round(J2*dx+10));
set(gca,'YTick',-round((I2*dy)/10)*10:15:0);
xlabel('x [mm]'); ylabel('y [mm]')

handle3 = subplot(2,2,3);  %VORTICITY PLOT
contourf(XGbl,YGbl,PosWzGbl(:,:,n),'LevelList',0.001:Step:HiValue,...
 'LineColor','none'); colorbar; grid on; hold on;
scatter(xQ(:,n),yQ(:,n),'MarkerEdgeColor',[0 0 0],... 
 'Marker','x','LineWidth',1);
title(strcat('\omega_{z} ',Num));
xlim([dx,J2*dx]); ylim([I2*-dy,-dy]);
set(gca,'XTick',0:15:round(J2*dx+10));
set(gca,'YTick',-round((I2*dy)/10)*10:15:0);
xlabel('x [mm]'); ylabel('y [mm]')

if n == 1
 handle4 = subplot(2,2,4);  %CIRCULATION PLOT
plot(ImagesTotDom,Circulation(:,1),'b*',... 
 ImagesTotDom,Circulation(:,2),'ro');
grid on; xlabel('Formation Number'); ylabel('\Gamma');
legend('\Gamma_{vortex}', '\Gamma_{total}','Location','northwest')
title(strcat('Circulation'));
%ylim([0 120]);
xlim([ImagesTotDom(1) ImagesTotDom(NoI)]);
set(gca,'XTick',100:200:1500);
%set(gca,'YTick',0:10:100);
end

set(handle1,'NextPlot','replace'); set(handle2,'NextPlot','replace');
set(handle3,'NextPlot','replace'); set(handle4,'NextPlot','replace')
end

%Save plots to file in folder. Create new folder if needed.
if strcmp(SavePlotToFile,'Y') == 1
 if n == 1;
   cd(SaveLocation);
   [status,message,messageID] = mkdir(Name);
   if strcmp(message,'Directory already exists.') == 1
     n1 = 1;
     while strcmp(message,'Directory already exists.') == 1
       newName = strcat(Name,sprintf( ' _%d',n1));
       [status,message,messageID] = mkdir(newName);
       n1 = n1+1;
     end
   end
end
fprintf('Specified folder already exists.\n')
fprintf('New folder: %s.\n', NewName)
cd(NewName);
else
cd(Name);
end
end

%Write ASCII file
if n == 1 && strcmp(Run, 'Track_Vortex') == 1
  ImageNumTr = transpose(ImagesTotDom);
  FileName = 'CirculationDataSet.txt';
  WriteMatrix = horzcat(ImageNumTr, Circulation);
  fileID = fopen(FileName, 'wt');
  fprintf(fileID, '%s\n', '
', 'ImageNum, Vortex, Total'); fclose(fileID);
  dlmwrite(FileName, WriteMatrix, '-append', 'newline', 'pc');
end

set(gcf, 'PaperPosition', [300 300 12 10])
%   set(gcf,'PaperPosition', [300 300 8 7])
saveas(gcf, PlotName, 'png');
%   close(gcf);
end
end
cd('C:\Users\Admin\Documents\A Jacob Wilroy\MATLAB\Research');

B.3: GLOBALIZER FUNCTION

function [Output] = fcn_Globalizer(In1, In2, In3, In4, In5, In6, In7, In8, In9)

Data = In1; MaxWidth = In2; MaxHeight = In3; ImagesTotDom = In4; NoS = In5;
istart = In6; iend = In7; jstart = In8; jend = In9;

Output = zeros(MaxHeight, MaxWidth, length(ImagesTotDom));
for m = 1:length(ImagesTotDom)
  Locall = zeros(MaxHeight, MaxWidth, NoS);
  Locall(istart(1):iend(1), jstart(1):jend(1), 1) = Data(:, :, 1, m, 1);
  Locall(istart(2):iend(2), jstart(2):jend(2), 2) = Data(:, :, 1, m, 2);

  for i = 1:MaxHeight
    for j = 1:MaxWidth
      Output(i, j, m) = sum(Locall(i, j, :))/nnz(Locall(i, j, :));
      if isnan(Output(i, j, m)) == 1; Output(i, j, m) = 0; end
    end
  end
  fprintf('Image set %g globalized\n', m);
end
end
B.4: LOCAL CIRCULATION FUNCTION

function [Out1] =fcn_LocalCirculation(In1,In2,In3,In4,In5)
%{
Purpose: Calculate vorticity using the "Local Circulation Method" as define
in Adam DeVoria's thesis (2011). Instead of conducting a one-dimensional
finite difference scheme for vorticity, a line integral is performed around
each data point in the flow field. See the appendix of his thesis for more
information.
%}
%% Settings and Initializations
U = In1; V = In2; I = In3; J = In4; dx = In5;
%% Computations
Vorticity = zeros(I,J);
R = dx;
Constant = ((3+2*sqrt(2))/(24*R));
for i=2:I-1
    for j=2:J-1
        u2 = U(i-1,j+1); u3 = U(i-1,j); u4 = U(i-1,j-1); u6 = U(i+1,j-1);
        u7 = U(i+1,j); u8 = U(i+1,j+1);
        v1 = V(i,j+1); v2 = V(i,j-1);
        v4 = V(i-1,j-1); v5 = V(i-1,j); v6 = V(i-1,j+1); v8 = V(i+1,j+1);
        Vorticity(i,j) = Constant*((14-9*sqrt(2))*((u7-u3)+(v1-v5))...
            +(sqrt(2)-1)*((u6+u8+v2+v8)-(u2+u4+v4+v6)));
    end
end
%% Outputs
Out1 = Vorticity;
end

B.5: LAMBDA CI CRITERION FUNCTION

function [Out1] =fcn_LambdaCICriterion(In1,In2,In3,In4,In5,In6,In7)
%{
Purpose: Calculate the Lambda CI criterion for a given flow field.
%}
%% Settings and Initializations
U = In1; V = In2; I = In3; J = In4; dx = In5; dy = In6;
N = In7;
%% Computations
LambdaCI = zeros(I,J);
[dUdx,dUdy] = gradient(U,dx,-dy); [dVdx,dVdy] = gradient(V,dx,-dy);
for i=1:I
    for j=1:J
        Delq = [dUdx(i,j), dUdy(i,j); dVdx(i,j), dVdy(i,j)];
        Vec = eig(Delq);
        if isreal(Vec(1)) == 0 || isreal(Vec(2)) == 0;
            LambdaCI(i,j) = imag(Vec(1));
        end
    end
end
LambdaCI(1:2,1:J) = 0; LambdaCI(I-1:I,1:J) = 0;
LambdaCI(1:1,I:2) = 0; LambdaCI(1:1,J-1:J) = 0;
AvgLambdaCI = sum(sum(LambdaCI))/nnz(LambdaCI);
for i=1:I
  for j=1:J
    if LambdaCI(i,j) < N*AvgLambdaCI; LambdaCI(i,j) = 0; end
  end
end

%% Outputs
Out1 = LambdaCI;
end

B.6: SEARCH EXPANSION FUNCTION

function [Out1,Out2,Out3]=fcn_SearchExpansion4(In1,In2,In3,In4,In5,In6)

% Purpose: Conduct a search whereby the code expands outward in the north, south, east, and west directions from an initial location. The NSEW expansion continues for each direction until a certain number of zero vorticity i,j locations are found, specified by the variable BlankLimit. At every i,j location that is identified as part of the cluster, a new search will begin from that location. This will continue until the connected region of values has been completely searched.

%% Settings & Initializations
i_start = In1; j_start = In2; SearchMatrix = In3; I = In4; J = In5; BlankLimit = In6;
Tracker = zeros(I,J); Tracker(i_start,j_start) = 1; i = 0; j = 0;
NorthSearch = 0; SouthSearch = 0; EastSearch = 0; WestSearch = 0;
up = 0; down = 0; left = 0; right = 0; FinishSearch = 'N'; NewSearch = 'N';

%% North-South-East-West Expansion Search
% Do a search from each point where contiguous, nonzero data is found
while strcmp(FinishSearch, 'N') == 1
  for ii = 1:I
    for jj = 1:J
      % Based on what has been found, look for i,j of where to start
      if Tracker(ii,jj) == 1
        Tracker(ii,jj) = 2;
        Tracker(ii,jj) = 2;
        NewSearch = 'Y'; i = ii; j = jj;
        break;
      end
    end
    if strcmp(NewSearch, 'Y') == 1; break; end;
  end

  % If no starting locations are found, stop searching
  if strcmp(NewSearch, 'N') == 1 && ii==I && jj==J
    FinishSearch = 'Y'; break;
  end

  % Walk north
  while NorthSearch <= BlankLimit
    up = up+1; idexU = i-up;
    if idexU < 1; break; end

    if SearchMatrix(idexU,j) > 0 && Tracker(idexU,j) == 0
      Tracker(idexU,j) = 1;
    end
  end
end

end
else
    NorthSearch = NorthSearch+1;
end

if j-1 > 0 && j-1 < J+1 %Catty-corner location
    if SearchMatrix(idexU,j-1) > 0 && Tracker(idexU,j-1) == 0
        Tracker(idexU,j-1) = 1;
    end
end

if j+1 > 0 && j+1 < J+1 %Catty-corner location
    if SearchMatrix(idexU,j+1) > 0 && Tracker(idexU,j+1) == 0
        Tracker(idexU,j+1) = 1;
    end
end

end

%Walk south
while SouthSearch <= BlankLimit
    down = down+1; idexD = i+down;
    if idexD > I; break; end

    if SearchMatrix(idexD,j) > 0 && Tracker(idexD,j) == 0
        Tracker(idexD,j) = 1;
    else
        SouthSearch = SouthSearch+1;
    end

    if j-1 > 0 && j-1 < J+1
        if SearchMatrix(idexD,j-1) > 0 && Tracker(idexD,j-1) == 0
            Tracker(idexD,j-1) = 1;
        end
    end

    if j+1 > 0 && j+1 < J+1
        if SearchMatrix(idexD,j+1) > 0 && Tracker(idexD,j+1) == 0
            Tracker(idexD,j+1) = 1;
        end
    end

end

%Walk east
while EastSearch <= BlankLimit
    right = right+1; jdexR = j+right;
    if jdexR > J; break; end

    if SearchMatrix(i,jdexR) > 0 && Tracker(i,jdexR) == 0
        Tracker(i,jdexR) = 1;
    else
        EastSearch = EastSearch+1;
    end

    if i-1 > 0 && i-1 < I+1
        if SearchMatrix(i-1,jdexR) > 0 && Tracker(i-1,jdexR) == 0
            Tracker(i-1,jdexR) = 1;
        end
    end
end
if i+1 > 0 && i+1 < I+1
    if SearchMatrix(i+1,jdexR) > 0 && Tracker(i+1,jdexR) == 0
        Tracker(i+1,jdexR) = 1;
    end
end

%Walk west
while WestSearch <= BlankLimit
    left = left+1; jdexL = j-left;
    if jdexL < 1; break; end

    if SearchMatrix(i,jdexL) > 0 && Tracker(i,jdexL) == 0
        Tracker(i,jdexL) = 1;
    else
        WestSearch = WestSearch+1;
    end

    if i-1 > 0 && i-1 < I+1
        if SearchMatrix(i-1,jdexL) > 0 && Tracker(i-1,jdexL) == 0
            Tracker(i-1,jdexL) = 1;
        end
    end

    if i+1 > 0 && i+1 < I+1
        if SearchMatrix(i+1,jdexL) > 0 && Tracker(i+1,jdexL) == 0
            Tracker(i+1,jdexL) = 1;
        end
    end

end

%Reset the search criteria
NorthSearch = 0; SouthSearch = 0; EastSearch = 0; WestSearch = 0;
NewSearch = 'N'; up = 0; down = 0; left = 0; right = 0;
end

%% Calculate the centroid of the area where positive vorticity was found
Weight = zeros(I,J);
g_ix = zeros(I,1); g_i = zeros(I,1);
g_jx = zeros(1,J); g_j = zeros(1,J);

for i=1:I %Transfer only the vorticity of the connected region
    for j=1:J
        if Tracker(i,j) == 2; Weight(i,j) = SearchMatrix(i,j); end
    end
end

for i=1:I %i centroid
    g_ix(i,1) = i*sum(Weight(i,:));%nnz(Tracker(i,:))*;
    g_i(i,1) = sum(Weight(i,:));
end
i_centroid = round(sum(g_ix)/sum(g_i));

for j=1:J %j centroid
    g_jx(1,j) = j*sum(Weight(:,j));%nnz(Tracker(:,j))*;
    g_j(1,j) = sum(Weight(:,j));
end
end
j_centroid = round(sum(g_jx)/sum(g_j));

% Outputs
Out1 = i_centroid;
Out2 = j_centroid;
Out3 = Tracker;
end

B.7: SEARCH MATRIX THINNING FUNCTION

function [O1] = fcn_ThinSearchMatrix(In1,In2,In3,In4,In5,In6,In7)
%
Purpose: Reduce the number of locations clustered by zeroing values of the
weight matrix where the value is less than some % of the peak value.
%
% Settings and Initializations
i_start = In1; j_start = In2; Weight1 = In3; Tracker = In4;
I = In5; J = In6; Thin = In7;
% Delete values from the Weight matrix that do not meet the criterion
Weight2 = zeros(I,J); PeakValue = 0;

for i=1:I %Transfer only the vorticity of the connected region
    for j=1:J
        if Tracker(i,j) == 2; Weight2(i,j) = Weight1(i,j); end;
    end
end

i_start_peak = (i_start-8); i_end_peak = (i_start+8);
j_start_peak = (j_start-8); j_end_peak = (j_start+8);
if i_start_peak<1; i_start_peak = 1;
elseif i_end_peak>I; i_end_peak = I;
end
if j_start_peak<1; j_start_peak = 1;
elseif j_end_peak>J; j_end_peak = J;
end

for i=i_start_peak:i_end_peak %Look for the peak value
    for j=j_start_peak:j_end_peak
        Value = Weight2(i,j);
        if Value > PeakValue; PeakValue = Value; end
    end
end

for i=1:I %Delete values < __% of peak value
    for j=1:J
        if Tracker(i,j) == 2 && Weight2(i,j) < Thin*PeakValue
            Tracker(i,j) = 3; Weight2(i,j) = 0;
        end
    end
end

% Outputs
O1 = Weight2;
end

B.8: TOTAL CIRCULATION FUNCTION

function [O1,O2,O3,O4]=fcn_TotalCirculation_SS(In1,In2,In3,In4,In5,In6)
%
Purpose: This function is used for global circulation calculation (shear
layer and vortex). Vortex and shear layer tracked using Q criterion. Shear layer and vortex discovered using vorticity, but only after the plate vorticity has been deleted. Integration area defined by vorticity.

%% Settings & Initializations
UMat = In1; VMat = In2; Vorticity = In3; PackageDeal = In4; VelMag = In5; ImageNum = In6;

I = PackageDeal(1,1); J = PackageDeal(1,2); NumofImages = PackageDeal(1,3); dx = PackageDeal(1,4); dy = PackageDeal(1,5); Count = 0; VortexTracking = zeros(NumofImages,11); PV = zeros(NumofImages,3); Zone = zeros(I,J,NumofImages); Circulation = zeros(NumofImages,1); CircCoordinates = zeros(2,5,NumofImages);

% PlateTo
p = [2 51]; PlateBot = [78 127]; %Plate Location = [i j], Smooth
% PlateTop = [9 49]; PlateBot = [87 127]; %Plate Location = [i j], Groove
PlateTop = [3 41]; PlateBot = [88 127]; %Plate Location = [i j], Smooth
PlateTop1 = [10 52]; PlateBot1 = [84 127];
PlateTop2 = [12 52]; PlateBot2 = [56 97];
% PlateTop = [9 49]; PlateBot = [87 127]; %Plate Location = [i j], Groove

%SearchArea = 'Limited_Manual'; %Choose this one for stage 2, stage 3
SearchArea = 'Unlimited'; %Choose this one for stage 1

%% Define Plate Area
PlateArea1 = zeros(I,J); PlateArea2 = zeros(I,J); PlateArea3 = zeros(I,J); PlateArea4 = zeros(I,J);
PlateLength = PlateBot(1,1) - PlateTop(1,1) + 1;
for m=1:12
    i = (PlateTop(1,1)+m-1):1:(PlateBot(1,1)+m-1);
    j = PlateTop(1,2):1:PlateBot(1,2);
    for k=1:PlateLength
        PlateArea1(i(k),j(k)) = 2;
    end
end
PlateLength1 = PlateBot1(1,1) - PlateTop1(1,1) + 1;
for m=1:9
    i = (PlateTop1(1,1)+m-1):1:(PlateBot1(1,1)+m-1);
    j = PlateTop1(1,2):1:PlateBot1(1,2);
    for k=1:PlateLength1
        if m <= 2; PlateArea2(i(k),j(k)) = 2; end
        PlateArea3(i(k),j(k)) = 2;
    end
end
PlateLength2 = PlateBot2(1,1) - PlateTop2(1,1) + 1;
for m=1:5
    i = (PlateTop2(1,1)+m-1):1:(PlateBot2(1,1)+m-1);
    j = PlateTop2(1,2):1:PlateBot2(1,2);
    for k=1:PlateLength2
        PlateArea4(i(k),j(k)) = 2;
    end
end

%% Peak Value Search
fprintf('Total Circulation ');
for n=1:NumofImages
    %Process data for each image
    Count = Count+1; PV(n,1) = 0; U = UMat(:, :, n); V = VMat(:, :, n);
    if n <= 4; PlateArea = PlateArea3; else PlateArea = PlateArea2; end
%Calculate Vortex Criterion for image
[LamCI] = fcn_LambdaCICriterion(U,V,I,J,dx,dy,2);

%Evaluate vortex location, search entire image for peak value
if strcmp(SearchArea,'Unlimited')
    for i=8:13
        % for i=8:25
        for j=44:50
            % for j=40:55
                if LamCI(i,j) > PV(n,1)
                    PV(n,1) = LamCI(i,j); PV(n,2) = j; PV(n,3) = i;
                end
            end
        end
    end

%Evaluate vortex location, search manually limited area for peak value
elseif strcmp(SearchArea,'Limited_Manual')
    for i=round(0.1*I):round(0.9*I)
        for j=round(0.3*J):round(0.7*J)
            if LamCI(i,j) > PV(n,1)
                PV(n,1) = LamCI(i,j); PV(n,2) = j; PV(n,3) = i;
            end
        end
    end

%Evaluate vortex location, search predicted area for peak value
elseif strcmp(SearchArea,'Limited')
    for i=I_search_start:I_search_end
        for j=J_search_start:J_search_end
            if LamCI(i,j) > PV(n,1)
                PV(n,1) = LamCI(i,j); PV(n,2) = j; PV(n,3) = i;
            end
        end
    end

% Conduct search to find contiguous data and calculate centroid

%Using peak value location, cluster, then calculate centroid
j_start = PV(n,2); i_start = PV(n,3);
if j_start == 0; error('Error with peak value search. j_start = 0'); end;
if i_start == 0; error('Error with peak value search. i_start = 0'); end;

%Masking points near the plate
VortexTracking(n,1) = PV(n,1);
VortSearch1 = Vorticity(:,:,n); %VortSearch = VortSearch1;
LamCI11 = LamCI;
for i=PlateTop1(1,1):PlateBot1(1,1)
    for j=PlateTop1(1,2):PlateBot1(1,2)
        if LamCI(i,j) ~= 0 && Vorticity(i,j,n) == 0; LamCI11(i,j) = 0; end
        if PlateArea(i,j) == 2;
            if LamCI(i,j) == 0; Vorticity(i,j,n) = 0; end
        LamCI(i,j) = 0; Vorticity(i,j,n) = 0;
    end
end

%Vortex Search - Cluster vortex's criterion points
[i_centroid_Vort,j_centroid_Vort,Tracker_Vortex1] = ...
    fcn_SearchExpansion4(i_start,j_start,LamCI11,I,J,1);
%Shear Layer Search
if n>1
    PV_SL = 0;
    for i=8:15
        for j=42:49
            if LamCI11(i,j) > PV_SL && Vorticity(i,j,n) ~= 0
                PV_SL = LamCI11(i,j); i_start_SL = i; j_start_SL = j;
            end
        end
    end
end

%Cluster shear layer's criterion points
 [~,~,Tracker_ShearLayer1] = ...
 fcn_SearchExpansion4(i_start_SL,j_start_SL,LamCI11,I,J,1);
if nnz(Tracker_ShearLayer1)<4;
    PV_SL = 0;
    for i=8:15
        for j=42:49
            if Vorticity(i,j,n) > PV_SL
                PV_SL = Vorticity(i,j,n); i_start_SL = i; j_start_SL = j;
            end
        end
    end
    [~,~,Tracker_ShearLayer1] = ...
    fcn_SearchExpansion4(i_start_SL,j_start_SL,LamCI11,I,J,1);
end
end

%Plate Search - Set this manually, based on prior observation of matrix
i_start_Plate = 86; j_start_Plate = 121;
PV_Plate = 0;
for i=60:80
    for j=110:127
        if VortSearch1(i,j) > PV_Plate
            PV_Plate = VortSearch1(i,j);
            i_start_Plate = i; j_start_Plate = j;
        end
    end
end

%Cluster plate BL vorticity
 [~,~,Tracker_Plate1] = ...
 fcn_SearchExpansion4(i_start_Plate,j_start_Plate,VortSearch1,I,J,1);
for i=1:I
    for j=1:J
        %Delete plate tracking points where the vortex could exist
        if Tracker_Vortex1(i,j) == 2 && Tracker_Plate1(i,j) == 2
            Tracker_Plate1(i,j) = 0;
        end
        %Delete plate tracking points where the shear layer could exist
        if n>1
            if Tracker_ShearLayer1(i,j) == 2 && Tracker_Plate1(i,j) == 2
                Tracker_Plate1(i,j) = 0;
            end
        end
    end
end
end
%Thin the identified points by 15% of the maximum value, then recluster
[VortSearch2] = fcn_ThinSearchMatrix(i_start_Plate,j_start_Plate,...
VortSearch1,Tracker_Plate1,I,J,0.15);
[~,~,Tracker_Plate2] = ... 
fcn_SearchExpansion4(i_start_Plate,j_start_Plate,VortSearch2,I,J,1);

%Remove BL vorticity from vorticity matrix used in Circ. calculation
for i=PlateTop(1,1):PlateBot(1,1)
  for j=PlateTop(1,2):PlateBot(1,2)
    %Plate BL leaves domain after approximately image 600
    if ImageNum(n) <= 600
      if Tracker_Plate2(i,j) == 2 && PlateArea1(i,j) == 2
        VortSearch1(i,j) = 0;
      end
    end
    if LamCI(i,j) == 0 && PlateArea1(i,j) == 2
      VortSearch1(i,j) = 0;
    end
  end
end

%Re-cluster shear layer and vortex using vorticity
[~,~,Tracker_Vortex2] = ... 
fcn_SearchExpansion4(i_start,j_start,VortSearch1,I,J,1);
if n>1
  [~,~,Tracker_ShearLayer2] = ...  
fcn_SearchExpansion4(i_start_SL,j_start_SL,VortSearch1,I,J,1);
end

%Choose centroid
j_centroid = j_centroid_Vort; i_centroid = i_centroid_Vort;
VortexTracking(n,2)=j_centroid;
VortexTracking(n,3)=VortexTracking(n,2)*dx;
VortexTracking(n,4)=i_centroid;
VortexTracking(n,5)=VortexTracking(n,4)*dy;

% Predict the search location for next image
if Count == 2
  SearchArea = 'Limited';
  I_diff = VortexTracking(n,4)-VortexTracking(n-1,4);
  J_diff = VortexTracking(n,2)-VortexTracking(n-1,2);
  %Vortex does not travel backwards and usually does not move more than 4
  %points in either the i or j direction. But the algorithm will
  %unfortunately identify a random hot point and follow it off another
  %direction, i.e. screw up the prediction
  if I_diff > 8 || J_diff > 8 || I_diff < 0 || J_diff < 0
    I_diff = PV(n,3)-PV(n-1,3);
    J_diff = PV(n,2)-PV(n-1,2);
  end
  if I_diff > 8 || J_diff > 8 || I_diff < 0 || J_diff < 0
    I_diff = 3; J_diff = 3;
  end

  I_next_center = VortexTracking(n,4)+I_diff;
  J_next_center = VortexTracking(n,2)+J_diff;
  I_search_start = I_next_center-4;
  J_search_start = J_next_center-4;
  I_search_end = I_next_center+4;
  J_search_end = J_next_center+4;
  if I_search_start < 1; I_search_start = 1; end

end
if J_search_start < 1; J_search_start = 1; end
if I_search_end > I; I_search_end = I; end
if J_search_end > J; J_search_end = J; end
VortexTracking(n,6) = J_search_start*dx;
VortexTracking(n,7) = I_search_start*dy;
VortexTracking(n,8) = J_search_end*dx;
VortexTracking(n,9) = I_search_end*dy;
Count = Count-1;
end

%% Circulation Calculations
% Search Tracker matrix for farthest reaching points (NSEW)
i_north = I+1; i_south = 0; j_west = J+1; j_east = 0;

if n == 1
    Track = Tracker_Vortex2;
else
    Track = Tracker_Vortex2+Tracker_ShearLayer2;
end

% Define integration box
for i=1:I
    for j=1:J
        if Track(i,j) == 2 || Track(i,j) == 4
            if i<i_north; i_north = i; end
            if i>i_south; i_south = i; end
            if j<j_west; j_west = j; end
            if j>j_east; j_east = j; end
        end
    end
end

% Include any vorticity shed from shear layer towards vortex to the west
% if ImageNum(n) >= 400
    lowerlimit = i_south*1.5; if lowerlimit > I; lowerlimit = I; end;
    if saferlimit = j_west*1.5; if saferlimit < 1; saferlimit = 1; end;
    saferlimit = 4; safererlimit = j_west;
    for i=i_north:lowerlimit
        for j=safererlimit:safererlimit
            if Track(i,j) == 0 && LamCI(i,j) ~= 0 && Vorticity(i,j,n) ~= 0
                Track(i,j) = 3;
            end
        end
    end
end

% Remove vorticity on plate caused by secondary vortex
if ImageNum(n) >= 200
    for i=PlateTop2(1,1):PlateBot2(1,1)
        for j=PlateTop2(1,2):PlateBot2(1,2)
            if PlateArea4(i,j) == 2; Vorticity(i,j,n) = 0; end
        end
    end
end

% Summing vorticity for circulation calculation
for i=i_north:i_south
    for j=j_west:j_east
        Zone(i,j,n) = Vorticity(i,j,n);
    end
end
CircCoordinates(1,:,n) = [j_west,j_east,j_east,j_west,j_west]*dx;
CircCoordinates(2,:,n)= -1*[i_north,i_north,i_south,i_south,i_north]*dy;
Circulation(n,1) = (sum(sum(Zone(:,:,n)))/100)*dx*dy;

%% Outputs
O1 = Circulation;
O2 = VortexTracking;
O3 = Zone;
O4 = CircCoordinates;
end

B.9: VORTEX CIRCULATION FUNCTION

function [O1,O2,O3,O4]=fcn_VortexCirculation_SS(In1,In2,In3,In4,In5)
Purpose: This function is used to track the vortex by searching for the peak value, and then by using the first two images, predict where the next peak value will occur.

%% Settings & Initializations
UMat = In1; VMat = In2; Vorticity = In3; PackageDeal = In4; VelMag = In5;

I = PackageDeal(1,1); J = PackageDeal(1,2); NoI = PackageDeal(1,3);
dx = PackageDeal(1,4); dy = PackageDeal(1,5);
VortTracking = zeros(NoI,11); Count = 0;
Zone = zeros(I,J,NoI); CircCoordinates = zeros(2,5,NoI);
Circulation = zeros(NoI,1); PV = zeros(NoI,3);
DiffStore = zeros(NoI,2);

% SearchArea = 'Limited_Manual'; %Choose this one for stage 2, stage 3
SearchArea = 'Unlimited'; %Choose this one for stage 1

% PlateTop = [2 51]; PlateBot = [78 127]; %Plate Location = [i j], Smooth
% PlateTop = [9 49]; PlateBot = [87 127]; %Plate Location = [i j], Groove
PlateTop = [12 52]; PlateBot = [86 127]; %Plate Location = [i j], Smooth
% PlateTop = [9 49]; PlateBot = [87 127]; %Plate Location = [i j], Groove

%% Define Plate Area
PlateArea2 = zeros(I,J); PlateArea3 = zeros(I,J);
PlateLength = PlateBot(1,1) - PlateTop(1,1)+1;
for m=1:8
    i = (PlateTop(1,1)+m-1):1:(PlateBot(1,1)+m-1);
    j = PlateTop(1,2):1:PlateBot(1,2);
    for k=1:PlateLength
        if m <= 2; PlateArea2(i(k),j(k)) = 2; end;
        PlateArea3(i(k),j(k)) = 2;
    end
end
%PlateArea(75:127,122:127) = 2;

%% Peak Value Search
fprintf('Vortex Circulation ');
for n=1:NoI
    Count = Count+1; PV(n,1) = 0; U = UMat(:,;,,n); V = VMat(:,;,,n);
    %A = Vorticity(:,;,,n); B = VelMag(:,;,,n);
    if n < 2; PlateArea = PlateArea3; else PlateArea = PlateArea2; end

    %Calculate LambdaCI Criterion for image
    [LamCI] = fcn_LambdaCICriterion(U,V,I,J,dx,dy,0.1);

    %Evaluate vortex location, search entire image for peak value
    if strcmp(SearchArea,'Unlimited')
        for i=8:13
            for j=44:50
                if LamCI(i,j) > PV(n,1)
                    PV(n,1) = LamCI(i,j);
                    PV(n,2) = j; PV(n,3) = i;
                end
            end
        end
    end

    %Evaluate vortex location, search manually limited area for peak value
    elseif strcmp(SearchArea,'Limited_Manual')
        for i=round(0.1*I):round(0.9*I)
for j=round(0.3*J):round(0.7*J)
    if LamCI(i,j) > PV(n,1)
        PV(n,1) = LamCI(i,j);
        PV(n,2) = j; PV(n,3) = i;
    end
end
end

%Evaluate vortex location, search predicted area for peak value
elseif strcmp(SearchArea, 'Limited')
    for i=I_search_start:I_search_end
        for j=J_search_start:J_search_end
            if LamCI(i,j) > PV(n,1)
                PV(n,1) = LamCI(i,j);
                PV(n,2) = j; PV(n,3) = i;
            end
        end
    end
end

% Conduct search to find contiguous data and calculate centroid
% Using peak value location, expand outward to find bounds, then centroid
j_start = PV(n,2);
i_start = PV(n,3);
if j_start == 0; j_start = 1; end; if i_start == 0; i_start = 1; end;
VortTracking(n,1) = PV(n,1);
for i=1:I
    for j=1:J
        if LamCI(i,j) ~= 0 && Vorticity(i,j,n) == 0; LamCI(i,j) = 0; end
        if PlateArea(i,j) == 2;
%         if LamCI(i,j) == 0; Vorticity(i,j,n) = 0; end
            LamCI(i,j) = 0; Vorticity(i,j,n) = 0;
        end
    end
end

%Cluster vortex points
 [~,~,VorTrak1] = ...
 fcn_SearchExpansion4(i_start,j_start,LamCI,I,J,1);
%Cluster additional points within the identified group. Set by "Ext".
VorTrak11 = zeros(I,J); Ext = 7;
for i=(Ext+1):(I-Ext)
    for j=(Ext+1):(J-Ext)
        if VorTrak1(i,j) == 0 && Vorticity(i,j,n) ~= 0
            if sum(LamCI(i:i-Ext,j)) > 0 %North by Ext grid points
                VorTrak11(i,j) = 2;
            elseif sum(LamCI(i:i+Ext,j)) > 0 %South by Ext grid points
                VorTrak11(i,j) = 2;
            elseif sum(LamCI(i,j:j+Ext)) > 0 %East by Ext grid points
                VorTrak11(i,j) = 2;
            elseif sum(LamCI(i,j:j-Ext)) > 0 %West by Ext grid points
                VorTrak11(i,j) = 2;
            end
        end
    end
end
VorTrak111 = VorTrak1+VorTrak11;
 [~,~,VorTrak1111] = ...
 fcn_SearchExpansion4(i_start,j_start,VorTrak111,I,J,1);
%This section used to for tracking purposes, not circulation.

[LamCI2] = fcn_ThinSearchMatrix(i_start, j_start, LamCI, VorTrak1, I, J, .2);
[i_centroid_VC, j_centroid_VC, ~] = ...
    fcn_SearchExpansion4(i_start, j_start, LamCI2, I, J, 1);
if isnan(i_centroid_VC) == 1 || isnan(j_centroid_VC) == 1
    PV(n,1) = 0; PV(n,2) = 0; PV(n,3) = 0;
    i_t_s = (i_start-6); i_t_e = (i_start+6);
    j_t_s = (j_start-6); j_t_e = (j_start+6);
    if i_t_s <=0; i_t_s = 0; end
    if i_t_e <=0; i_t_e = 0; end
    if j_t_s <=0; j_t_s = 0; end
    if j_t_e <=0; j_t_e = 0; end
    for i=i_t_s:i_t_e
        for j=j_t_s:j_t_e
            if LamCI2(i,j) > PV(n,1)
                PV(n,1) = LamCI2(i,j);
                PV(n,2) = j; PV(n,3) = i;
            end
        end
    end
    j_start = PV(n,2); i_start = PV(n,3);
    [i_centroid_VC, j_centroid_VC, ~] = ...
        fcn_SearchExpansion4(i_start, j_start, LamCI2, I, J, 1);
    end

{%
% For debugging
set(0,'defaultfigureposition',[100 100 800 800])
[II,JJ] = meshgrid(1:1:J,1:1:I); JJ = -1*JJ;
% h1 = subplot(1,3,1);
% if n>=80
%     h1 = figure(1);
%     A=[j_start,j_start];B=[0,-1];C = [0,J];D=[-i_start,-i_start];
%     contourf(II,JJ,LamCI,'LevelList',[0.0000001:0.5:5],'LineColor','none');
%     hold on; grid on; plot(A,B,'r',C,D,'r'); set(h1,'NextPlot','replace');
% end
%
% quiver(II,JJ,U,V,2,'Color','w');
% colorbar; title('Vortex Criterion')
% h2 = subplot(1,3,2);
% contourf(II,JJ,Vorticity(:,:,n),'LevelList',[0:0.5:5],'LineColor','none');
% hold on;grid on;
% quiver(II,JJ,U,V,2,'Color','w');
% colorbar; title('Vorticity')
% h3 = subplot(1,3,3);
% contourf(II,JJ,VelMag(:,:,n),'LevelList',[0:2:30],'LineColor','none');
% hold on; grid on;
% quiver(II,JJ,U,V,2,'Color','w');
% colorbar; title('VelMag')
% set(h1,'NextPlot','replace');set(h2,'NextPlot','replace');
% set(h3,'NextPlot','replace')
%
% Choose centroid
if n>1
    if abs(j_centroid-j_centroid_VC)>10||abs(i_centroid-i_centroid_VC)>10
        j_centroid = j_centroid+2; i_centroid = i_centroid+2;
    else
        j_centroid = j_centroid_VC; i_centroid = i_centroid_VC;
    end
else
    j_centroid = j_centroid_VC; i_centroid = i_centroid_VC;
end
%}

92
%% Predict the search location for next image
if Count == 2
    SearchArea = 'Limited';
    I_diffnow = round(VortTracking(n,4)-VortTracking(n-1,4));
    J_diffnow = round(VortTracking(n,2)-VortTracking(n-1,2));
    %Vortex does not travel backwards and usually does not move more than 4
    %points in either the i or j direction. But the algorithm will
    %unfortunately identify a random hot point and follow it off another
    %direction, i.e. screw up the prediction
    DiffStore(n,1) = I_diffnow; DiffStore(n,2) = J_diffnow;
    if n > 3
        if I_diffnow>3 || J_diffnow>3 || I_diffnow<0 || J_diffnow<0
            I_diffavg = sum(DiffStore(n-3:n,1))/nnz(DiffStore(n-3:n,1));
            J_diffavg = sum(DiffStore(n-3:n,2))/nnz(DiffStore(n-3:n,2));
            I_diff = round(I_diffavg); J_diff = round(J_diffavg);
            if isnan(I_diff) == 1; I_diff = 3; end
            if isnan(J_diff) == 1; J_diff = 3; end
        end
    else
        I_diff = I_diffnow; J_diff = J_diffnow;
    end
    if I_diff > 3 || J_diff > 3 || I_diff < 0 || J_diff < 0
        DiffStore(n,1) = 1; DiffStore(n,2) = 1;
        I_diff = PV(n,3)-PV(n-1,3);
        J_diff = PV(n,2)-PV(n-1,2);
    end
    if I_diff > 3 || J_diff > 3 || I_diff < 0 || J_diff < 0
        I_diff = 1; J_diff = 1;
    end
    I_next_center = VortTracking(n,4)+I_diff;
    J_next_center = VortTracking(n,2)+J_diff;
    I_search_start = I_next_center-4;
    J_search_start = J_next_center-4;
    I_search_end = I_next_center+4;
    J_search_end = J_next_center+4;
    if I_search_start < 1; I_search_start = 1; end
    if J_search_start < 1; J_search_start = 1; end
    if I_search_end > I; I_search_end = I; end
    if J_search_end > J; J_search_end = J; end
    VortTracking(n,6) = J_search_start*dx;
    VortTracking(n,7) = I_search_start*-dy;
    VortTracking(n,8) = J_search_end*dx;
    VortTracking(n,9) = I_search_end*-dy;
    Count = Count-1;
end

%% Circulation Calculations
i_north = I+1; i_south = 0; j_west = J+1; j_east = 0;
for i=1:I
    for j=1:J
        if VorTrak1111(i,j) ~= 0;
            Zone(i,j,n) = Vorticity(i,j,n);
            if i<i_north; i_north = i; end
            if i>i_south; i_south = i; end
            if j<j_west; j_west = j; end
            if j>j_east; j_east = j; end
        end
    end
end
% CircCooridnates(1,:,n) = [j_west,j_east,j_east,j_west,j_west]*dx;  
CircCoordinates(2,:,n) = [-i_north,i_north,i_north,i_north,i_north]*dy;  
Circulation(n,1) = (sum(sum(Zone[:,:,n]))/100)*dx*dy;

if n >= 118
    set(0,'defaultfigureposition',[100 100 800 800])
    [II,JJ] = meshgrid(1:1:J,1:1:I); JJ = -1*JJ;
    h1 = figure(1);
    contourf(II,JJ,Zone(:,:,n),'LevelList',[0:0.5:5], 'LineColor','none');
    hold on; grid on; set(h1,'NextPlot','replace');
end

if n == 1; fprintf('Files Processed: 1');
else fprintf(', %g', n);
end
Jacob = 'dumb';

fprintf('
');

clear;
%
{ Creator: Jacob Wilroy.                  Build Date: July 2016
Description: Code is designed to copy images from folder "Name" to folder "RawData_Sink" (Insight 4G Raw Data folder). Code is setup so that multiple data sets having the same image number can be processed together.  
}%
%% Settings
%IMAGE SELECTION
Preprocessing = 50:50:500;   %Images for Avg Min Intensity Image
ImageNumRead = [5:5:500];   %Stage 1 Processing
% ImageNumRead = [25:25:2250];   %Stage 2 Processing
% ImageNumRead = [1000:25:2800];   %Stage 3 Processing
% ImageNumRead = [1600:25:2800];   %Stage 4 Processing
ImageNumWrite = ImageNumRead;
%DATA SET SELECTION
DataSets = 19:1:20;
% FileNum = 4:1:6;
%FILE/FOLDER PATH - Data Storage
Name = 'I:\JW_FlappingPlate\Smooth_45AoA_2Cam\32mmps\Bottom\Raw Data';
% Name = 'I:\JW_FlappingPlate\Smooth_45AoA_2Cam\32mmps\Top\Raw Data';
%FILE/FOLDER PATH - Data Analysis (these don't change)
RawData_Sink = 'C:\Experiments10\Jacob\PIV Data\RawData';
% RawData_Sink = 'F:\SmoothTemp\32mmps\Top\4t6';
%% Computations
NoP = length(ImageNumRead);NoPreProcs = length(Preprocessing);
NoS = length(DataSets); ImageMatrix = cell(NoP,2,NoS);
FileNameA = cell(NoP,2,NoS);
FileNameB = cell(NoP,2,NoS);
ReadPathName = cell(NoS,1);
for k=1:NoS
  ImageNumReadM = ImageNumRead+ImageNumRead(length(ImageNumRead))*(k-1);
  for n=1:NoP %Generate all file names
    N = ImageNumRead(n); M = ImageNumReadM(n);
    FileNameA{n,1,k} = sprintf('img%04d.LA.tif',N);
    FileNameA{n,2,k} = sprintf('img%04d.LB.tif',N);
    FileNameB{n,1,k} = sprintf('img%04d.LA.tif',M);
    FileNameB{n,2,k} = sprintf('img%04d.LB.tif',M);
    ImageMatrix{n,1,k} = zeros(1024,1024,'uint8');
    ImageMatrix{n,2,k} = zeros(1024,1024,'uint8');
  end
  Num1 = sprintf(' %d',DataSets(k));
  ReadPathName{k,1} = strcat(Name,Num1);
end
% Read new image files
for k=1:NoS
  cd(ReadPathName{k,1});
  for n=1:NoP
    ImageMatrix{n,1,k} = imread(FileNameA{n,1,k});
    ImageMatrix{n,2,k} = imread(FileNameA{n,2,k});
  end
  fprintf('Set %g read
',DataSets(k));
end
% Write new pre-processing image files
cd(RawData_Sink);
for k=1:NoS
  for i=1:NoPreProcs
    n = 0; ImageNum = 0;
    while ImageNum ~= Preprocessing(i)
      n = n+1; ImageNum = ImageNumWrite(n);
    end
    imwrite(ImageMatrix{n,1,k},FileNameB{n,1,k},'WriteMode','overwrite')
    imwrite(ImageMatrix{n,2,k},FileNameB{n,2,k},'WriteMode','overwrite')
  end
  fprintf('Paused, press enter to continue...
')
  pause;
  disp('Continuing')
end
% Write new image files
for k=1:NoS
  for i=1:NoP
    imwrite(ImageMatrix{i,1,k},FileNameB{i,1,k},'WriteMode','overwrite')
    imwrite(ImageMatrix{i,2,k},FileNameB{i,2,k},'WriteMode','overwrite')
  end
  fprintf('Set %g written
',DataSets(k));
end
cd('C:\Users\Admin\Documents\A Jacob Wilroy\MATLAB\Research');
fprintf('Finished\n')
disp('******************************************************************************

B.11 IMAGE EXTRACTION CODE AFTER PROCESSING

clear;
%
Creator: Jacob Wilroy. Build Date: February 2016
Description: Code is designed to move vector files from "Analysis_Source"
folder (Insight 4G Analysis folder) to "Name" (usually an external
hard drive where experimental data is being stored). Code then deletes
previous run's images from the Raw Data folder (RawData_Sink) and Analysis folder (Analysis_Source). Change the variable "Delete_RawData" to 'N' if there are no files to delete.

%% Settings
% IMAGE SELECTION
ImageNumRead = [5:5:500]; % Stage 1 Processing
% ImageNumRead = [25:25:2250]; % Stage 2 Processing
% ImageNumRead = [1000:25:2800]; % Stage 3 Processing
% ImageNumRead = [1600:25:2800]; % Stage 4 Processing
% DATA SET SELECTION
DataSets = 19:1:20; Delete_RawData = 'Y'; % Images to delete? 'Y' or 'N'
% FileNum = 4:1:6; Delete_RawData = 'N'; % Images to delete? 'Y' or 'N'
% FILE/FOLDER PATH - Data Storage
Name = 'I:\JW_FlappingPlate\Smooth_45AoA_2Cam\32mmps\Bottom\R3\Analysis';
% Name = 'I:\JW_FlappingPlate\Smooth_45AoA_2Cam\32mmps\Top\R2\Analysis';
% FILE/FOLDER PATH - Data Analysis (these don't change)
RawData_Sink = 'C:\Experiments10\Jacob\PIV Data\RawData';
Analysis_Source = 'C:\Experiments10\Jacob\PIV Data\Analysis';
%% Settings
% FILE/FOLDER PATH
Name = 'I:\JW_FlappingPlate\Smooth_45AoA_2Cam\32mmps\Top\4t6A';
% Generate all file names
NoP = length(ImageNumRead); NoS = length(DataSets);
FileNameA = cell(NoP,NoS);
FileNameB = cell(NoP,NoS);
FileNameC = cell(NoP,NoS);
WritePathName = cell(NoS,1);
for k=1:NoS
    ImageNumReadM = ImageNumRead+ImageNumRead(length(ImageNumRead))*(k-1);
    for n=1:NoP
        N = ImageNumRead(n); M = ImageNumReadM(n);
        FileNameA{n,k} = sprintf('img%04d.L.vec',N);
        FileNameB{n,k} = sprintf('img%04d.L.vec',M);
        FileNameC{n,1,k} = sprintf('img%04d.LA.tif',M);
        FileNameC{n,2,k} = sprintf('img%04d.LB.tif',M);
    end
    Num1 = sprintf('%d',DataSets(k));
    WritePathName{k,1} = strcat(Name,Num1);
end
% Move vector files to storage
for k=1:NoS
    cd(Analysis_Source)
    for n=1:NoP
        for m=1:NoS
            movefile(FileNameB{n,k},WritePathName{k,1})
        end
        fprintf('Set %g moved\n',DataSets(k));
    end
end
% Rename
for k=1:NoS
    % Rename files where needed
    cd(WritePathName{k,1})
    if k>1
        for n=1:NoP
            % Move vector files to storage
            movefile(FileNameB{n,k},FileNameA{n,k})
        end
        fprintf('Set %g renamed\n',DataSets(k));
    end
end
% Delete Files
if strcmp(Delete_RawData,'Y') == 1
    cd(RawData_Sink);
    for k=1:NoS
        for n=1:NoP
            % Delete images from previous run
            delete(FileNameC{n,1,k}, FileNameC{n,2,k});
```
end
disp('Remaining contents of "Raw Data" folder deleted');
cd(Analysis_Source)
for k=1:NoS
  for n=1:NoP
    % Empty remaining contents of Analysis folder
    delete(FileNameC{n,1,k}, FileNameC{n,2,k});
  end
end
% Delete the two generated Minimum Intensity Images
delete('Generated000000.T-001.D-001.RAW.H-001.LA.tif',...
      'Generated000000.T-001.D-001.RAW.H-001.LB.tif');
disp('Remaining contents of "Analysis" folder deleted');
end
cd('C:\Users\Admin\Documents\A Jacob Wilroy\MATLAB\Research');
fprintf('Finished\n')
disp('*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*')

B.12: DATA SKIP FUNCTION

function [O1] = fcn_DataSkip(In1,In2,In3)
  %{ Purpose: Used to skip data points like in TecPlot. Self explanatory. }%
  %% Settings and Initializations
  Data = In1; I = In2; J = In3;
  
  Skip_X = 3; Skip_Y = Skip_X;
  % Grab velocity data using index skip
  Data_Skip = zeros(I,J);
  Indices_I = [1:Skip_Y:round(I/Skip_Y)*Skip_Y,0];
  Indices_J = [1:Skip_X:round(J/Skip_X)*Skip_X,0];
  NumofIndices_I = length(Indices_I); NumofIndices_J = length(Indices_J);
  counti = 1; countj = 1; I_go = 'N';

  for i=1:I
    if i == Indices_I(counti); I_go = 'Y'; counti = counti+1; end
    for j=1:J
      if strcmp(I_go,'Y') == 1;
        if j == Indices_J(countj)
          Data_Skip(i,j) = Data(i,j); countj = countj+1;
        end
      end
    end
  end
  I_go = 'N'; countj = 1;
end

  %% Outputs
  O1 = Data_Skip;
end
APPENDIX C: LABVIEW VIRTUAL INSTRUMENT CODES

C.1: IMAGE ACQUISITION AND PLATE TRANSLATION CODE
C.2: BINARY FILE CONVERSION EVEN SEQUENCE 3

<table>
<thead>
<tr>
<th>String 1</th>
<th>F:\UW_FlappingPlate\Smooth_45\CoA\Stage 3\8mmps\Run 2\img</th>
</tr>
</thead>
<tbody>
<tr>
<td>String 2</td>
<td>F:\UW_FlappingPlate\Smooth_45\CoA\Stage 3\8mmps\Raw Data 1\img</td>
</tr>
<tr>
<td>String 3</td>
<td>F:\UW_FlappingPlate\Smooth_45\CoA\Stage 3\8mmps\Run 1\img</td>
</tr>
<tr>
<td>String 4</td>
<td>F:\UW_FlappingPlate\Smooth_45\CoA\Stage 3\8mmps\Raw Data 1\img</td>
</tr>
</tbody>
</table>
C.3: BINARY FILE CONVERSION ODD SEQUENCE 3