

# ANALYSIS OF BIO-INSPIRED PROPULSORS

MELISSA A. GREEN

A DISSERTATION  
PRESENTED TO THE FACULTY  
OF PRINCETON UNIVERSITY  
IN CANDIDACY FOR THE DEGREE  
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE  
BY THE DEPARTMENT OF  
MECHANICAL AND AEROSPACE ENGINEERING

ADVISORS: C.W. ROWLEY AND A.J. SMITS

APRIL 2009

© Copyright by Melissa A. Green, 2009.  
All Rights Reserved



# Abstract

The locomotion of fish and aquatic animals is achieved by the oscillation of fins and flukes, which creates highly three-dimensional, unsteady flow fields that are not yet well-understood. The principal non-dimensional parameter presently used to describe these flows is the Strouhal number,  $St = fA/U$ , which depends on the frequency of oscillation ( $f$ ), the width of the wake ( $A$ ), and the freestream velocity ( $U$ ). In previous work on two-dimensional foils, wake structure and thrust performance have been shown to scale with this parameter, but it does not include considerations of three-dimensionality, which become important in the study of low-aspect ratio propulsors. In the current work, a new nondimensional scaling parameter is proposed which, for a rigid rectangular pitching panel, collapses measurements of pressure, thrust, and circulation when plotted again Strouhal number.

Dye flow visualization and Digital Particle Image Velocimetry (DPIV) were used to investigate the wakes of rigid pitching panels with a trapezoidal panel geometry, chosen to model idealized fish caudal fins. A Lagrangian Coherent Structure (LCS) analysis was employed to investigate the formation and evolution of the panel wake. The LCS analysis, based on calculations of the Direct Lyapunov Exponent (DLE) has several advantages over Eulerian methods, including greater detail and the ability to define structure boundaries without relying on a preselected threshold. A bifurcation of the LCS structure in the wake of the trapezoidal pitching panels coincided with an observed transition of the wake structure. Also, the LCS analysis provided evidence of the “trapping” of vortices on trapezoidal panel surfaces with low sweep angles or at low Strouhal numbers.

# Acknowledgements

Thanks to Lex and Clancy, who have taught me about life and work and fluid mechanics. I'll always hold myself to the standards I've learned here. To Mom and Dad, I keep realizing that much on which I pride myself is a product of how I was raised. To Sue and Michelle, my sisters and my best friends. To John, my favorite big brother. To Jess, who will kill me if she doesn't see her name here (but who deserves my gratitude anyway.) To Sean, whose friendship has made me a better researcher, coffee drinker, and crossword puzzler. To James, who was a good mentor and friend (and who built a very nice experiment for me to steal). To Radd and Joe and Glenn and Vocaturo, who are quick to help and to chat. To Juan and to Euan, who made me think I won some sort of roommate lottery. To Janel and space shuttles, Vroom! To Marcus and wine on the porch. To Megan and lab girl-power. To Steve and semi-regular Monday morning meetings. To Chrissie and Sarah and the Woody Wu Fountain. To all my labmates and classmates in all my years in all corners of the Equad. And to Stupid Thursdays.

This work was supported by the National Science Foundation Graduate Research Fellowship, the National Science Foundation CAREER award #CMS-0347239 (Clarence W. Rowley), the National Institute of Health CNRS Grant 1R01NS054271, and the Office of Naval Research MURI Grant ONR N00014-08-1-0642.

This dissertation carries the number T-3198T in the records of the Department of Mechanical and Aerospace Engineering.

# Contents

Abstract . . . . .	iii
Acknowledgements . . . . .	iv
List of Tables . . . . .	vii
List of Figures . . . . .	viii
<b>1 Introduction</b>	<b>4</b>
1.1 Motivation and Goals . . . . .	4
1.2 Approach and Objectives . . . . .	5
1.3 Previous Work . . . . .	6
1.4 Overview and resulting publications . . . . .	12
<b>2 Lagrangian coherent structure analysis</b>	<b>14</b>
2.1 Eulerian methods . . . . .	14
2.2 Lagrangian criterion . . . . .	15
2.2.1 Example: Hill’s spherical vortex . . . . .	16
2.3 Isolated hairpin vortex . . . . .	19
2.3.1 Formation of secondary hairpin . . . . .	23
2.3.2 Positive-time LCS . . . . .	27
2.4 Fully turbulent channel . . . . .	30
2.4.1 Comparison with Eulerian criteria . . . . .	32
2.5 Summary . . . . .	32
<b>3 Experimental methods</b>	<b>34</b>
3.1 Experimental facility . . . . .	34
3.2 Panel geometry and kinematics . . . . .	36
3.3 Time-resolved pressure measurements: rigid rectangular panel . . . . .	37
3.3.1 Flow visualization and DPIV: rigid trapezoidal panel . . . . .	40
3.3.2 Dye flow visualization . . . . .	41
3.3.3 Digital Particle Image Velocimetry . . . . .	43
<b>4 Three-dimensional effects</b>	<b>48</b>
4.1 Pressure measurements . . . . .	48
4.1.1 Temporal analysis . . . . .	48
4.1.2 Spatial analysis . . . . .	50
4.1.3 Scaling . . . . .	51
4.2 Application of scaling . . . . .	54

4.2.1	Summary . . . . .	61
<b>5</b>	<b>Three-dimensional panel wake</b>	<b>64</b>
5.1	Dye flow visualization . . . . .	64
5.1.1	White light illumination . . . . .	65
5.1.2	Planar laser-induced fluorescence . . . . .	67
5.2	Digital particle image velocimetry . . . . .	76
5.2.1	Eulerian analysis . . . . .	76
5.2.2	LCS analysis . . . . .	77
5.3	Summary . . . . .	93
<b>6</b>	<b>Conclusions and Future Work</b>	<b>95</b>
6.1	Discussion and conclusions . . . . .	95
6.1.1	Coherent structure identification . . . . .	95
6.1.2	Three-dimensionality . . . . .	96
6.1.3	Trapezoidal planform geometry . . . . .	96
6.2	Future work . . . . .	98
<b>A</b>	<b>Accelerated DLE calculation</b>	<b>105</b>
<b>B</b>	<b>DLE program</b>	<b>107</b>
B.1	Fortran90 code to compute DLE . . . . .	107
B.1.1	Main program to compute DLE in turbulent channel . . . . .	107
B.1.2	Supporting modules . . . . .	109
B.1.3	Main program to compute the intermediary flow maps for the panel data . . . . .	129
B.1.4	Main program to compute DLE using a composition of the in- termediary flow maps . . . . .	131
B.2	Fortran90 code to compute hyperbolicity criteria . . . . .	132
B.2.1	Main program . . . . .	132
B.2.2	Supporting modules . . . . .	133

# List of Tables

3.1	Rectangular panel dimensions . . . . .	38
3.2	Summary of pressure experiment parameters . . . . .	38
3.3	Trapezoidal panel geometries . . . . .	40
4.1	Rectangular panel dimensions . . . . .	49
5.1	Trapezoidal panel geometries . . . . .	64

# List of Figures

1.1	Terminology of fins and other features of fish. . . . .	5
1.2	Classes of body-caudal fin (BCF) locomotion. Shaded areas of the fish bodies indicate the sections used in locomotion. . . . .	5
1.3	Different tail configurations for ray-finned fish, from heterocercal (with uneven upper and lower lobes) to homocercal (symmetric). . . . .	6
1.4	Different tail configurations for ray-finned fish, from heterocercal (with uneven upper and lower lobes) to homocercal (symmetric). . . . .	7
1.5	Wake of a NACA 0012 airfoil pitching sinusoidally about the quarter-chord point. Flow is from right to left. . . . .	8
1.6	Schematic comparing (top) a drag producing 2P wake structure and (bottom) a thrust-producing 2S wake structure (reverse von Kármán vortex street). Freestream flow is from right to left. . . . .	9
1.7	Proposed sketch of flow structure around a finite aspect ratio airfoil pitching with $St = 0.35$ , $Re_c = 164$ , based on dye flow visualization. . . . .	10
1.8	Isosurfaces of vorticity magnitude for a trapezoidal fin at $St = 0.3$ . (a) Rigid trapezoidal panel, (b) spanwise symmetric flexibility, and (c) spanwise asymmetric flexibility. . . . .	11
1.9	Top and side views of dye flow visualization of low-aspect-ratio rectangular panel wakes at different Strouhal numbers, $Re_c = 640$ . (a) $St = 0.23$ , (b) $St = 0.43$ , and (c) $St = 0.64$ . . . . .	12
1.10	Vortex skeletons of the wake behind rigid rectangular pitching panels. (a) $St = 0.23$ and (b) $St = 0.43$ . . . . .	13
2.1	Two-dimensional cut through Hill's spherical vortex with radius $z^2 + r^2 = 1$ . (a) Streamlines with vortex radius indicated by a green circle, (b) Q criterion, and (c) swirling strength. . . . .	17
2.2	Negative-time DLE plots of Hill's spherical vortex, with white showing maximum values. (a) short integration time, and (b) long integration time. . . . .	18
2.3	Positive- and negative-time DLE field for Hill's spherical vortex. Contour is only shown for $DLE > 50\% DLE_{max}$ . . . . .	19
2.4	Evolved isolated hairpin vortex generated by linear stochastic estimation, plotted using $10\% \max \lambda_{ci}^2$ . (a) Initial condition, (b) $t^+ = 63$ , and (c) $t^+ = 171$ . . . . .	20

2.5	Two dimensional nDLE plots of the isolated hairpin evolved to $t^+ = 63$ . (a) 10% max $\lambda_{ci}^2$ superimposed with location of the three planes, (b) constant-streamwise ( $y$ - $z$ ) plane, (c) constant wall-normal ( $x$ - $z$ ) plane ( $y^+ = 98$ ), and (d) constant-spanwise ( $x$ - $y$ ) plane . . . . .	21
2.6	Sketches from Falco (1977) of the “typical eddy” in planes (a) parallel to the streamwise flow and normal to the wall, and (b) normal to the streamwise flow and normal to the wall. . . . .	21
2.7	(a) Isosurfaces of 10% max $\lambda_{ci}^2$ (black) superimposed on three planes of nDLE $\geq 60\%$ max (white) (b) Translucent isosurfaces of 60% max nDLE superimposed on three planes of nDLE (black) $\geq 60\%$ max (white)	23
2.8	Two dimensional nDLE plots of the isolated hairpin evolved to $t^+ = 171$ . (a) 10% max $\lambda_{ci}^2$ superimposed with location of the three planes, (b) constant-streamwise ( $y$ - $z$ ) plane, (c) constant wall-normal ( $x$ - $z$ ) plane ( $y^+ = 95$ ), and (d) constant-spanwise ( $x$ - $y$ ) plane . . . . .	24
2.9	nDLE constant-spanwise ( $x$ - $y$ ) planes for three integration times of the isolated hairpin evolved to $t^+ = 171$ . (a) $t^+ = 45$ , (b) $t^+ = 90$ , and (c) $t^+ = 135$ . . . . .	25
2.10	Magnitude of $\langle \mathbf{n}, \mathbf{S}\mathbf{n} \rangle$ along constant-spanwise ( $x$ - $y$ ) cross sections of nLCS surfaces at times (a) $t^+ = 45$ , (b) $t^+ = 63$ , and (c) $t^+ = 99$ . Negative strain rate (compression normal to the surface) is black; positive strain rate (expansion) is white with black outline. . . . .	26
2.11	Three-dimensional hyperbolic pLCS (gray) and nLCS (black) of the isolated hairpin head, plotted by using isosurfaces of 50% max DLE that satisfy the corresponding hyperbolicity criteria. . . . .	28
2.12	Hyperbolic pLCS (gray) and nLCS (black) of the isolated hairpin head in a two dimensional slice in (a) a constant-spanwise ( $x$ - $y$ ) plane and (b) a constant wall-normal ( $x$ - $z$ ) plane, plotted for regions of DLE $> 50\%$ maximum value that satisfy the corresponding hyperbolicity criteria. Time-dependent saddle points are highlighted using a pink box. Blue arrows indicate fluid behavior in figure (a). . . . .	29
2.13	(a) Vector field around the hairpin head in a frame that is convecting with the speed of the pLCS and nLCS intersection point. (b) Flowfield sketch of a large scale motion by Falco (1977), with the observer moving with the velocity of the saddle point. . . . .	30
2.14	Two dimensional nDLE plots of the fully turbulent channel. (a) 1% max $Q$ superimposed with location of the three planes, (b) constant wall-normal ( $x$ - $z$ ) plane, (c) constant-spanwise ( $x$ - $y$ ) plane ( $y^+ = 33$ ), and (d) constant-streamwise ( $y$ - $z$ ) plane. White boxes highlight structures that resembles corresponding cuts through the isolated hairpin. . . . .	31
2.15	Hyperbolic pLCS (gray) and nLCS (black) of the isolated hairpin head in a two dimensional slice in a constant-spanwise ( $x$ - $y$ ) plane through the fully turbulent channel, plotted for regions of DLE $> 50\%$ maximum value that satisfy the corresponding hyperbolicity criteria. A possible hairpin vortex is highlighted by a white box. . . . .	31
2.16	Constant-streamwise ( $y$ - $z$ ) planes of (a) nDLE and (b) $Q$ criterion . . . . .	32

3.1	Hydrogen bubble wire experiments, flow is from left to right. (a) $U = 0.012$ m/s, (b) $U = 0.018$ m/s, and (c) $U = 0.012$ m/s, showing disruption from thermal plumes. . . . .	35
3.2	Fairing and panel assembly. . . . .	37
3.3	Panel and fairing position in water channel test section. . . . .	38
3.4	Four bar linkage mechanism used to actuate the pitching panel. . . .	39
3.5	Pressure port placement on the low-aspect ratio ( $AR = S/C = 0.54$ ) panel (Panel 1). . . . .	40
3.6	Step response of the pressure transducer and tubing. $\Delta$ : experimental response; — optimized transfer function model. . . . .	41
3.7	Bode plots of the model transfer function. (a) Bode magnitude plot with log-log axes, (b) Bode phase plot with log-log axes, (c) Bode magnitude plot with linear axes zoomed in on frequencies of interest in the current work, and (d) Bode phase plot with linear axes zoomed in on frequencies of interest. . . . .	42
3.8	Diagram of trapezoid dimensions . . . . .	43
3.9	White light flow visualization orientations. Blue arrow indicates freestream flow direction. (a) top view, and (b) angles view. . . . .	43
3.10	Schematic of PLIF orientation and setup . . . . .	44
3.11	DPIV acquisition setup. . . . .	45
3.12	DPIV 2D planes and resulting 3D grids . . . . .	46
3.13	Illustration of optical effects. The distance between the camera and the image plane ( $h$ ) is the same in both cases. The width of the image plane when the camera is closer to the water surface is smaller than the width of the image plane when the camera is farther away from the water surface ( $d_2 > d_1$ .) . . . . .	47
4.1	Schematic of the panel advancing, and its associated trailing edge amplitude curve. . . . .	48
4.2	Propulsive efficiency variation with Strouhal number for Panel 1. . . .	49
4.3	Propulsive efficiency variation with Strouhal number for Panel 2. . . .	50
4.4	Unsteady pressure variation at $0.875c$ on Panel 1, $A = 20$ mm, $St = 0.27$ . —, trailing edge amplitude; $\circ$ , pressure deviation from the time-averaged mean. Panel is “retreating” for $90^\circ < \phi < 270^\circ$ . . . . .	51
4.5	Unsteady pressure variation at $0.875c$ on (a) Panel 2, $A = 20$ mm, $St = 0.33$ , and (b) Panel 1, $A = 40$ mm, $St = 0.33$ . —, trailing edge amplitude; $\circ$ , pressure deviation from the time-averaged mean. . . .	52
4.6	Streamwise distribution of pressure along the midspan for Panel 1 pitching at $St = 0.27$ and $A = 20$ mm. (a) $\phi = 0^\circ$ ; (b) $\phi = 90^\circ$ ; (c) $\phi = 180^\circ$ ; (d) $\phi = 270^\circ$ . In each figure, the upper part shows the pressure distribution and the lower part shows the position of the panel. *, pressure deviation from the time-averaged mean; $\Delta$ , location of pressure port. . . . .	53



4.7	Peak-to-peak pressure amplitude at $x = 0.875c$ : (a) $C_p$ ; (b) $C_p^*$ with $C_1 = 7$ . . . . .	55
4.8	Pressure gradient on advancing surface of panel. Magnitude indicated by arrow length. . . . .	56
4.9	Flow visualization of panel 1 pitching at $St=0.6$ , with the panel currently at motion extremum. The spanwise ejection of fluid highlighted with a white box. These images are from Buchholz (2006). . . . .	57
4.10	Previously measured thrust coefficient. (a) $C_T$ ; (b) $C_T^*$ with $C_2 = 7$ . . . . .	58
4.11	(a) Previously calculated circulation coefficient non-dimensionalized using panel pitching frequency and freestream velocity ( $C_{\Gamma_{Uf}}$ ); (b) Circulation coefficient with additional scaling ( $C_{\Gamma_{Uf}}^*$ with $C_3 = 7$ ). . . . .	59
4.12	(a) Previously calculated circulation coefficient non-dimensionalized using panel pitching amplitude and frequency ( $C_{\Gamma_{Af}}$ ); (b) Circulation coefficient with additional scaling ( $C_{\Gamma_{Af}}^*$ with $C_3 = 7$ ). . . . .	60
4.13	PIV at the midspan of a panel at $\phi = 90^\circ$ : (a) Panel 2, $A = 31$ mm, $St = 0.36$ , (b) Panel 1, $A = 31$ mm, $St = 0.26$ , and (c) Panel 1, $A = 20$ mm and $St = 0.27$ . Contour levels are $\pm n^2 s^{-1}$ , $n = 2, 3, 4, \dots, 10$ . . . . .	62
4.14	Unsteady pressure variation on the centerline at $0.875c$ with $A = 20$ mm and $St = 0.27$ : (—) Panel 1, $A = 20$ mm; (····) Panel 1, $A = 40$ mm; and (- - -) Panel 2, $A = 20$ mm. . . . .	63
5.1	White light flow visualization of panel C pitching at $St = 0.33$ , $Re_c = 2690$ , and $A/S = 0.009$ , viewed from above. (a) $\phi \approx 0^\circ$ , (b) $\phi \approx 90^\circ$ , (c) $\phi \approx 180^\circ$ , and (d) $\phi \approx 270^\circ$ . . . . .	65
5.2	White light flow visualization of panel C pitching at two different Strouhal numbers, viewed from above. (a) $St = 0.42$ , $Re_c = 2150$ , $A/S = 0.009$ , $\phi \approx 0^\circ$ , (b) $St = 0.42$ , $Re_c = 2150$ , $A/S = 0.009$ , $\phi \approx 90^\circ$ , (c) $St = 0.55$ , $Re_c = 1615$ , $A/S = 0.009$ , $\phi \approx 0^\circ$ , and (d) $St = 0.55$ , $Re_c = 1615$ , $A/S = 0.009$ , $\phi \approx 90^\circ$ . . . . .	66
5.3	Angled view white light flow visualization of panel C pitching at $St = 0.33$ , $Re_c = 2690$ , and $A/S = 0.009$ . Yellow boxes highlight the line of dye stretching as fluid that was pulled onto the far surface in the previous half cycles remains trapped there, and does not wash back on the near surface as the panel pitches into the page. (a) $\phi \approx 90^\circ$ , (b) $\phi \approx 180^\circ$ , and (c) $\phi \approx 270^\circ$ . . . . .	68
5.4	Angled view white light flow visualization of panel C pitching at $St = 0.55$ , $Re_c = 1615$ , $A/S = 0.009$ . Yellow box highlights fluid pulled from the far surface to the near surface as the panel pitches into the page. (a) $\phi \approx 330^\circ$ , (b) $\phi \approx 30^\circ$ , and (c) $\phi \approx 90^\circ$ . . . . .	69
5.5	PLIF images on panel C pitching at $St = 0.33$ , $Re_c = 2690$ , and $A/S = 0.009$ , acquired at $x = 0.25c$ . Blue boxes highlight counterrotating vortex pairs rolling up at the swept edge. (a) $\phi \approx 90^\circ$ , (b) $\phi \approx 180^\circ$ , (c) $\phi \approx 270^\circ$ , and (d) $\phi \approx 360^\circ$ . . . . .	70

5.6	PLIF images on panel C pitching at $St = 0.33, Re_c = 2690$ , and $A/S = 0.009$ , acquired at different streamwise locations along the chord. Phases are chosen to correspond to the crossing of the swept edge structure through the laser plane. (a) $x = 0.25c, \phi \approx 90^\circ$ , (b) $x = 0.5c, \phi \approx 180^\circ$ , (c) $x = 0.75c, \phi \approx 270^\circ$ , and (d) $x \approx c, \phi \approx 360^\circ$ . . . . .	71
5.7	PLIF images on panel C pitching at $St = 0.55, Re_c = 1615$ , and $A/S = 0.009$ , acquired at $x = 0.5c$ . Blue boxes highlight the quasi-streamwise structure as it is created and then pulled over the swept edge. (a) $\phi \approx 270^\circ$ and (b) $\phi \approx 315^\circ$ . . . . .	73
5.8	PLIF image of panel C pitching at $St = 0.55, Re_c = 1615$ , and $A/S = 0.009$ , acquired $x \approx c$ and $\phi \approx 180^\circ$ . . . . .	74
5.9	PLIF images of swept edge structure roll-up on panels B (at $x = 0.25c, St = 0.42, Re_c = 2390$ , and $A/S = 0.012$ ) and C (at $x = 0.5c, St = 0.42, Re_c = 2150$ , and $A/S = 0.009$ ). Blue box highlights the structure of interest. (a) Panel C, $\phi \approx 180^\circ$ , (b) panel C, $\phi \approx 270^\circ$ , (c) panel B, $\phi \approx 90^\circ$ , and (d) panel B, $\phi \approx 180^\circ$ . . . . .	75
5.10	Schematic of quasi-streamwise structure dynamics. Structure outlined in blue, distance from the swept edge indicated in red. . . . .	76
5.11	Two-dimensional planes of spanwise vorticity in four planes along the span of the trailing edge. $St = 0.33, Re_c = 2390$ , and $A/S = 0.012$ . Plane 1: $0.5S$ , plane 2: $8.67S$ , plane 3: $0.79S$ , and plane 4: $0.9S$ . . . . .	78
5.12	Two-dimensional planes of spanwise vorticity in four planes along the span of the trailing edge. $St = 0.55$ . Plane 1: $0.5S$ , plane 2: $0.79S$ , plane 3: $0.84S$ , and plane 4: $0.95S$ . . . . .	79
5.13	Spanwise vorticity for the panel pitching at $St = 0.33$ . Red and blue surfaces are 18% maximum and minimum $\omega_z$ , respectively. (a) $\phi = 0^\circ$ and (b) $\phi = 180^\circ$ . . . . .	80
5.14	Spanwise vorticity for the panel pitching at $St = 0.55$ . Red and blue surfaces are 17% maximum and minimum $\omega_z$ , respectively. (a) $\phi = 0^\circ$ and (b) $\phi = 180^\circ$ . . . . .	81
5.15	Angled view of the isosurfaces of $\omega_z$ at both Stouhal numbers. Red and blue surfaces are 18% maximum and minimum $\omega_z$ , respectively. (a) $St = 0.33, \phi = 0^\circ$ and (b) $St = 0.55, \phi = 180^\circ$ . . . . .	82
5.16	pLCS (red) and nLCS (blue) in three two-dimensional planes along the span of the panel pitching at $St = 0.33$ . LCS are represented by DLE fields that are blank for values less than 33% maximum of the field. (a) $0.5S$ , (b) $0.79S$ , and (c) $0.95S$ . . . . .	83
5.17	Hyperbolic pLCS and nLCS in three two-dimensional planes along the span of the panel pitching at $St = 0.33$ . Hyperbolic LCS are represented by DLE fields that are blank for values less than 33% maximum of the field, and if the applicable strain criterion is not satisfied. (a) $0.5S$ and (b) $0.79S$ . . . . .	84

5.18	Isosurfaces of spanwise vorticity and two-dimensional slices of nLCS around the panel pitching at $St = 0.33$ . Red and blue surfaces are 18% maximum and minimum $\omega_z$ , respectively. LCS are represented by regions where $nDLE > 43\%$ maximum value. . . . .	85
5.19	nLCS superimposed on two-dimensional planes of spanwise vorticity. nLCS are represented by regions where $nDLE > 33\%$ maximum value. (a) $0.5S$ (pink box highlights secondary vorticity sheet roll-up), (b) $0.79S$ , and (c) $0.84S$ . . . . .	87
5.20	pLCS and nLCS in three two-dimensional planes along the span of the panel pitching at $St = 0.33$ . LCS are represented by DLE fields that are blank for values less than 33% maximum of the field. (a) $0.5S$ , (b) $0.67S$ , and (c) $0.79S$ . . . . .	88
5.21	Hyperbolic pLCS and nLCS in three two-dimensional planes along the span of the panel pitching at $St = 0.55$ . Hyperbolic LCS are represented by DLE fields that are blank for values less than 32% maximum of the field, and if the applicable strain criterion is not satisfied. (a) $0.5S$ and (b) $0.67S$ . . . . .	89
5.22	Hyperbolic pLCS and nLCS at the midspan of the panel pitching at $St = 0.55$ . Hyperbolic LCS are represented by DLE fields that are blank for values less than 32% maximum of the field, and if the applicable strain criterion is not satisfied. (a) $\phi = 100^\circ$ , (b) $\phi = 260^\circ$ , (c) $\phi = 345^\circ$ , and (d) $\phi = 100^\circ$ . . . . .	90
5.23	Isosurfaces of spanwise vorticity and two-dimensional slices of nLCS around the panel pitching at $St = 0.33$ . Red and blue surfaces are 18% maximum and minimum $\omega_z$ , respectively. LCS are represented by regions where $nDLE > 43\%$ maximum value. Yellow boxes indicate locations of time-dependent saddle merges in the two planes closest to the midspan. . . . .	91
5.24	nLCS superimposed on two-dimensional planes of spanwise vorticity. nLCS are represented by regions where $nDLE > 32\%$ maximum value. (a) $0.5S$ (time-dependent saddle merge highlighted by yellow box), (b) $0.67S$ , and (c) $0.79S$ . . . . .	92
5.25	Two-dimensional nDLE field at $x = 0.83c$ on the panel pitching at both Strouhal numbers. (a) $St = 0.33$ and (b) $St = 0.55$ . . . . .	93
6.1	Proposed schematic of swept edge structure dynamics in the trapping situation. (a) As panel pitches into the page, a negative spanwise vortex is shed from the trailing edge, while a swept edge structure (bold) with components in the positive streamwise direction and positive spanwise direction. (b) In the next half-cycle as the panel pitches out of the page, the swept edge structure undergoes a like-sign interaction with the positive spanwise vortex being shed from the trailing edge. The swept edge structure forming on the far side of the panel is indicated by a dotted line. . . . .	97
6.2	Schematic of trapezoidal panel geometry with trailing edge notch. . .	98

# Nomenclature

$A$	peak-to-peak amplitude of trailing edge
$a$	spherical vortex radius
$AR$	panel aspect ratio
$C$	constant
$c$	chord length of panel
$C_p$	pressure coefficient
$C_T$	thrust coefficient
$C_\Gamma$	circulation coefficient
$D$	cylinder diameter
$d$	image plane width
DLE	Direct Lyapunov Exponent
$f$	pitching frequency
FTLE	Finite-Time Lyapunov Exponent
$h$	height above the water channel
$i$	integer
LCS	Lagrangian Coherent Structure
$k$	number of intermediary flow maps
$\mathbf{n}$	unit normal vector
$n_a$	refractive index of air
$n_w$	refractive index of water
nDLE	negative-time Direct Lyapunov Exponent
nLCS	negative-time Lagrangian Coherent Structure
$p$	pressure
pDLE	positive-time Direct Lyapunov Exponent
pLCS	positive-time Lagrangian Coherent Structure
$Re_c$	Reynolds number based on chord
$Re_\tau$	Reynolds number based channel half-width and friction velocity
$r$	radial coordinate
$\mathbf{S}$	rate of strain tensor
$S$	span of panel
$St$	Strouhal number
$t$	time
$T_e$	period of oscillation
$T_v$	stationary cylinder shedding frequency
$\mathbf{u}$	velocity vector

$U$	freestream velocity
$u$	streamwise velocity
$u_\tau$	friction velocity
$v$	transverse velocity
$w$	spanwise velocity
$\mathbf{x}$	location vector
$x$	streamwise coordinate
$y$	transverse coordinate
$z$	spanwise coordinate
$\beta$	refractive angle
$\delta$	channel half-width
$\Gamma$	circulation
$\theta$	trapezoidal panel sweep angle
$\nu$	kinematic viscosity
$\rho$	fluid density
$\phi$	phase
$\varphi$	azimuthal coordinate
$\Phi$	flow map
$\boldsymbol{\omega}$	vorticity vector
$\boldsymbol{\Omega}$	rate of rotation tensor
$\omega_z$	spanwise vorticity

For instance, on the planet Earth, man had always assumed that he was more intelligent than dolphins because he had achieved so much - the wheel, New York, wars and so on - whilst all the dolphins had ever done was muck about in the water having a good time. But conversely, the dolphins had always believed that they were far more intelligent than man - for precisely the same reasons.

Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

# Chapter 1

## Introduction

### 1.1 Motivation and Goals

The locomotion of fish enables high speed, efficiency, and maneuverability. To accomplish this, fish undulate their bodies and an array of fins, shown in figure 1.1, which create highly three-dimensional, unsteady flow fields that are not yet well-understood. In the field of engineering and fluid mechanics, the first step towards exploiting the evolutionary advances in fish locomotion has been to mimic the shape and motion of underwater swimmers. However, different fish have evolved not only to swim quickly or efficiently over large distances, but also to fit into different niches in the underwater environment. Therefore, a thorough analysis is needed to determine characteristics of fish form and function from which man-made applications could benefit.

The main goal of this work is to understand the fluid dynamics of the simple underlying shapes and motions of fish swimming, and to distill the characteristic phenomena that lead to efficient propulsion, large thrust, or increased maneuverability. A comprehension of the simple physics is expected to lead to advances in both the mechanical design of underwater vehicles and the design of control algorithms and actuators.

Figure 1.2 shows the range of locomotory classes for fish that create thrust by oscillating a combination of their bodies and caudal fins (BCF locomotion). At the far left of this spectrum is anguilliform motion, in which fish (eels and lampreys) pass a traveling wave down their bodies that has a wavelength less than the body length. As we move to the right towards more oscillatory motion, less of the body is undulated. For thunniform swimmers (sharks), less than half a wavelength is present on the body, and for ostraciiform swimmers (boxfish) the caudal fin is actuated in pure oscillation.

In the current work, we are interested in locomotion from the far right of this spectrum: fish that propel themselves primarily by an oscillation of their caudal fin. The caudal fin propulsor has a low aspect ratio, and therefore creates highly three-dimensional flowfields.

While thunniform and ostraciiform swimmers are similar in that they oscillate their caudal fin for propulsion, a wide variety of caudal fin planform geometries are

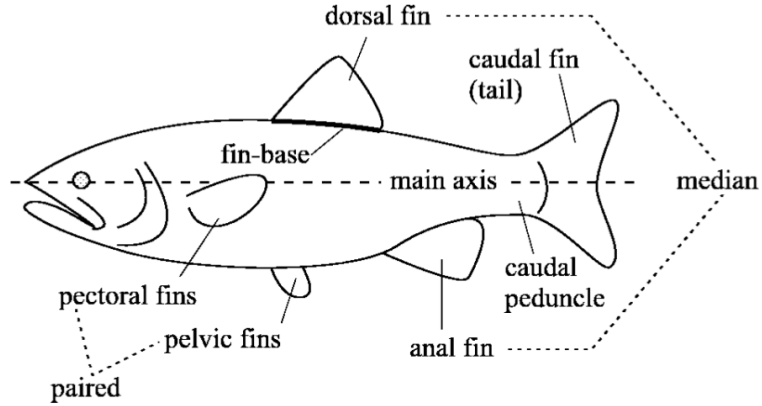


Figure 1.1: Terminology of fins and other features of fish. From Sfakiotakis *et al.* (1999).

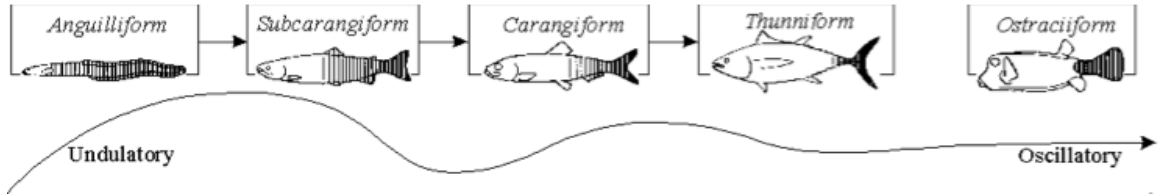


Figure 1.2: Classes of body-caudal fin (BCF) locomotion. Shaded areas of the fish bodies indicate the sections used in locomotion. From Lindsey (1978).

used, as shown in figure 1.3. The heterocercal caudal fins, which are externally asymmetric, are considered to be the more primitive configuration for sharks and ray-finned (teleost) fish. The homocercal, or symmetric, caudal fin is a derived morphology prevalent in teleost fishes (Lauder, 2000).

Here, we will focus on the simple symmetric geometries in a low-amplitude purely pitching motion as models of thunniform or ostraciiform swimmers with homocercal tails. In particular, we will also investigate what (if any) the benefits are of the trapezoidal planform geometry of teleost fishes. In order to separate the effects of geometric planform and the effects of flexibility, we will conduct experiments using rigid panels. Future work may look to include flexibility as a parameter. Previous work has been done to characterize the wakes of rigid rectangular low aspect ratio propulsors, and here we will look to build on that foundation and develop additional tools for the analysis of these wakes, as well as the wakes of trapezoidal propulsors.



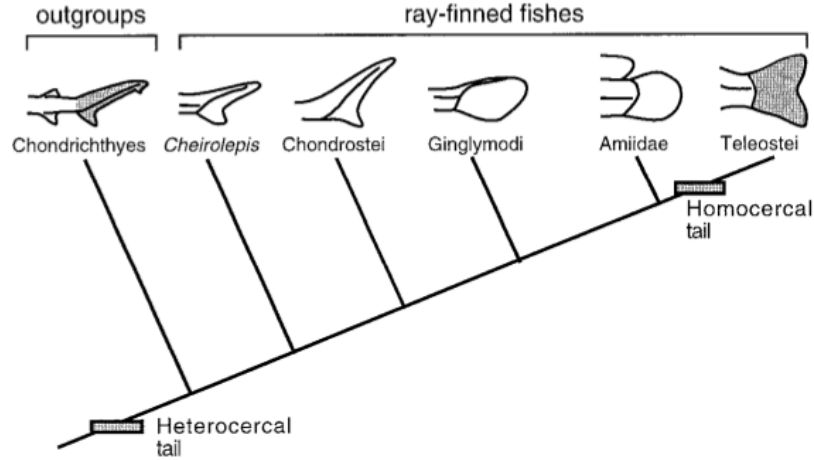


Figure 1.3: Different tail configurations for ray-finned fish, from heterocercal (with uneven upper and lower lobes) to homocercal (symmetric). From Lauder (2000).

## 1.2 Approach and Objectives

In the course of this work, we explore the use of Lagrangian Coherent Structures (LCS) as a tool for vortex identification and analysis. We use LCS to look at numerical data sets of previously studied flowfields and demonstrate its capability to identify coherent structures and provide additional detail about vortex dynamics.

To determine the parameters that govern three-dimensional unsteady flows, pressure measurements are conducted on the surface of simple rectangular panels. The wakes of trapezoidal panel are also investigated, and dye flow visualization and Digital Particle Image Velocimetry (DPIV) are used to examine the vortex production and evolution both around the panels and in their wakes. The velocity data from the DPIV was used to enable an LCS analysis of the vortex dynamics in the flow around the trapezoidal panels as well.

## 1.3 Previous Work

In order to describe the qualitative structure of the wake of the pitching panels in this and previous work, observations of the organization of vorticity shed by the panel are reported. Vorticity is the curl of the velocity field ( $\omega = \nabla \times \mathbf{u}$ ), and isolated regions of concentrated vorticity are often referred to as individual vortices or *coherent structures*.

A precise definition of a vortex remains elusive. Different authors have derived mathematical criteria for the definition of a vortex from the velocity gradient tensor ( $\nabla \mathbf{u}$ ), but while these criteria are able to locate vortex cores in a fluid flow, all require the application of an arbitrary threshold to determine those regions that are inside the vortex and those that are not. This is discussed in more detail in chapter 2. In

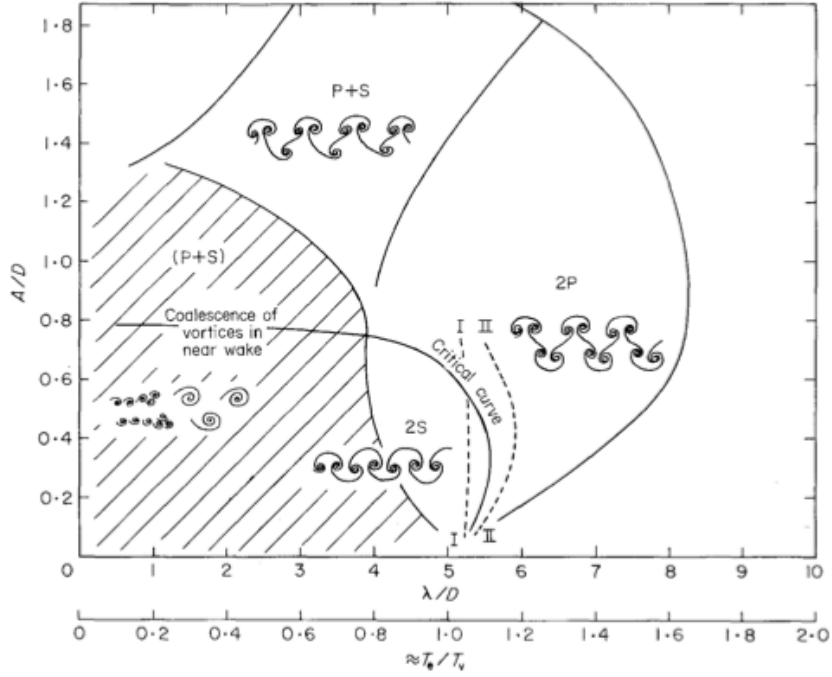


Figure 1.4: Wake structures of a transversely oscillating circular cylinder.  $A$  is the oscillation amplitude,  $D$  is the cylinder diameter,  $\lambda = T_e U$  where  $T_e$  is the period of oscillation and  $U$  is the freestream velocity, and  $T_v$  is the shedding frequency of a stationary cylinder. From Williamson & Roshko (1988).

the current work, we use a variety of methods in concert to identify and describe the behavior of the coherent structures in the wake of rigid pitching panels.

The work of Williamson & Roshko (1988) was the first to characterize the organization of vortical structures downstream of bodies oscillating in a freestream flow by investigating the wake of a circular cylinder oscillating transversely. Depending on the frequency of oscillation, different patterns of vortex shedding were observed, and their summary of the patterns is shown in figure 1.4. In each half-cycle of oscillation, either a single vortex (“S”) or a pair of vortices (“P”) were shed. Hence, the naming convention for a wake in which single vortices are shed each half-cycle is “2S,” as two single vortices are being shed each period. In the drag-producing wake, this is also referred to as a “von Kármán vortex street.” Similarly, when two pairs of vortices are shed each period, the wake is said to have a “2P” structure. A combination of these two basic structures (“P+S”) were also observed.

Early experimental work on fish swimming used two-dimensional airfoil sections or flat plates as models of fish fins, and showed that the basic wake structures behind oscillating cylinders could be used to describe the wakes behind the model fins. In particular, Oshima & Natsume (1980) and Koochesfahani (1989) visualized the flow over NACA 0012 airfoils, and observed a chain of vortical structures in the wake. An example of the flow visualizations of Koochesfahani (1989) is shown in figure 1.5.

At low nondimensional frequencies, the vortices were aligned similarly to the drag-producing 2S bluff-body wake, with the induced velocity between each subsequent pair of vortices in the upstream direction, causing a momentum deficit in the wake.

As the frequency was increased, the vortices in the wake realigned in the transverse direction so that eventually the induced flow between the vortex pairs was in the downstream direction, adding momentum to the streamwise flow and indicating that thrust was being produced by the pitching and heaving airfoil. The thrust-producing 2S wake is also referred to as a “reverse von Kármán vortex street.” A schematic of the differences between a drag-producing vortex wake (2P) and a thrust-producing vortex wake (2S) is shown in figure 1.6.

In the work of Oshima & Natsume (1980) and Freymuth (1988) vorticity was also observed to be shed from the leading edges of the pitching and heaving airfoils. Depending on the pitching frequency and amplitude, this leading-edge vortex (LEV) was sometimes seen to combine with the like-sign vortex being shed from the trailing edge. The combination of like-sign vorticity produced a wake pattern that indicated a stronger thrust production. Anderson *et al.* (1998) performed thrust measurements and DPIV on pitching and heaving foils and found that large propulsive efficiencies were possible (up to 87%), and were associated with the amalgamation of the LEV amalgamated and the vorticity shed from the trailing edge.

Triantafyllou *et al.* (1993) used a linear stability analysis on an experimentally measured wake of a two-dimensional oscillating foil to determine the frequency of optimal propulsive efficiency. They nondimensionalized the frequency of optimal efficiency in terms of the Strouhal number,

$$St = \frac{fA}{U}. \quad (1.1)$$

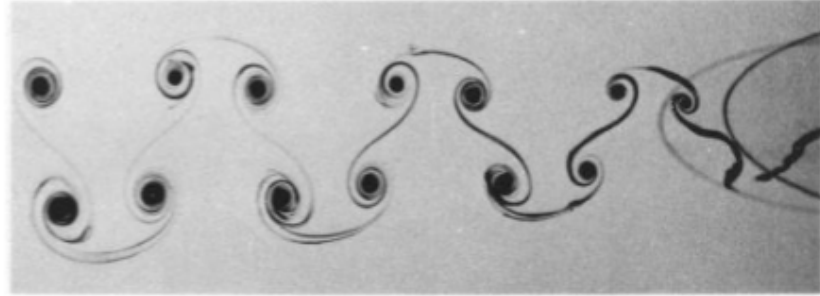
Here,  $f$  is the frequency of oscillation,  $A$  is the width of the wake, and  $U$  is the freestream velocity. The peak-to-peak amplitude of the trailing edge is commonly used as an approximation for  $A$ . The authors predicted that optimal efficiency for flapping foils occurs in the range  $0.25 \leq St \leq 0.35$ , and this was confirmed with a two-dimensional flapping foil experiment. Furthermore, analysis of the locomotion of a variety of fish species showed that many marine animals swim in this Strouhal number range.

Experiments on pitching and flapping bodies of finite aspect ratio have revealed that the wake structure is considerably more complex than the two-dimensional case, and that it is a strong function of Strouhal number and aspect ratio  $AR$  ( $= S/C$ , where  $S$  is the span and  $C$  is the chord). von Ellenrieder *et al.* (2003) and Parker *et al.* (2007) used dye flow visualization and DPIV to study the flowfields of finite aspect ratio pitching airfoils, and proposed the three-dimensional wake structure seen in figure 1.7. Drucker & Lauder (1999) and Tytell (2006) also used DPIV to investigate the flow around oscillating sunfish fins, and described the wakes in terms of staggered vortex rings.

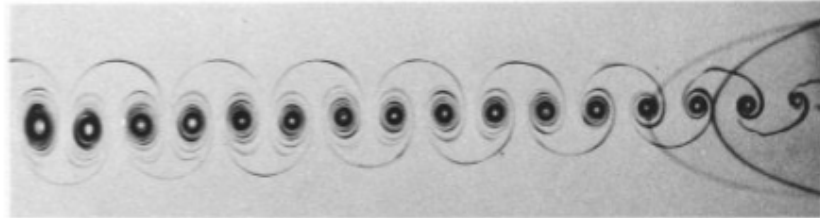
Anguilliform swimming has also been investigated by Tytell & Lauder (2004), who performed flowfield measurements around an American eel, and Jiménez *et al.* (2003)



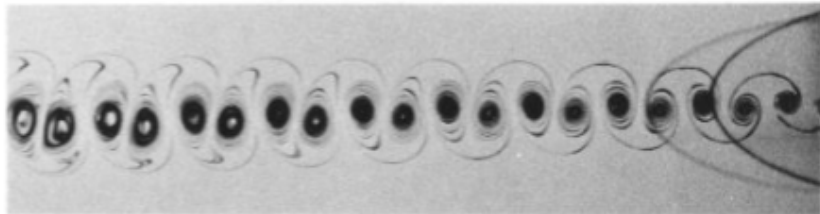
(a)  $A = 4 \text{ deg.}, f = 0.5 \text{ Hz}$



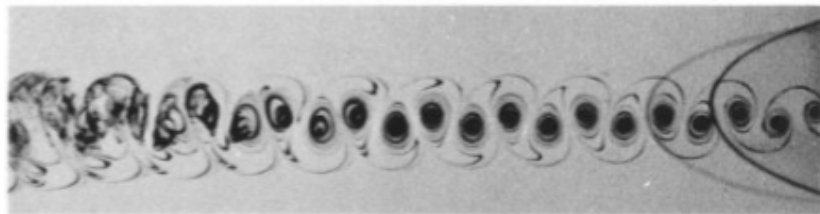
(b)  $A = 4 \text{ deg.}, f = 1.85 \text{ Hz}$



(c)  $A = 2 \text{ deg.}, f = 4.0 \text{ Hz}$



(d)  $A = 2 \text{ deg.}, f = 5.0 \text{ Hz}$



(e)  $A = 2 \text{ deg.}, f = 6.0 \text{ Hz}$

Figure 1.5: Wake of a NACA 0012 airfoil pitching sinusoidally about the quarter-chord point. Flow is from right to left. From Koochesfahani (1989)

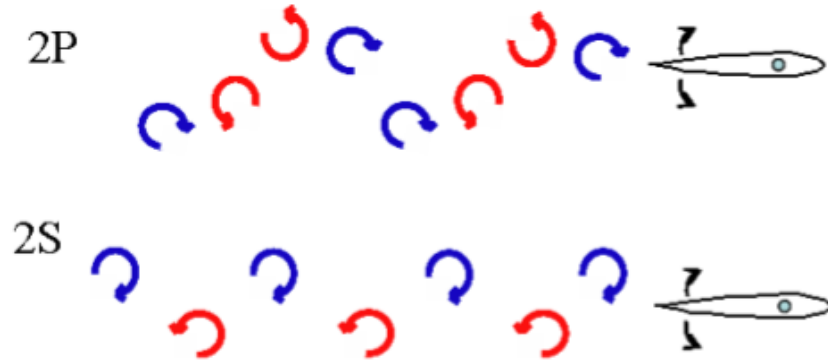


Figure 1.6: Schematic comparing (top) a drag producing 2P wake structure and (bottom) a thrust-producing 2S wake structure (reverse von Kármán vortex street). Freestream flow is from right to left.

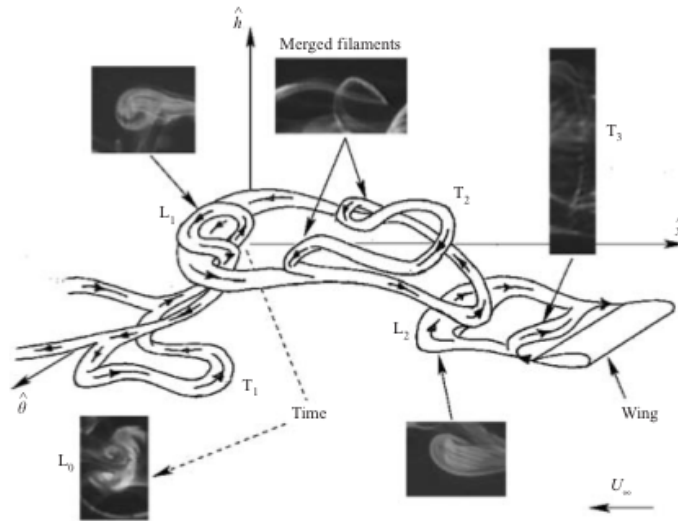


Figure 1.7: Proposed sketch of flow structure around a finite aspect ratio airfoil pitching with  $St = 0.35$ ,  $Re_c = 164$ , based on dye flow visualization. From von Ellenrieder *et al.* (2003).

and Hultmark *et al.* (2007) performed experiments around a buckling panel and a robotic lamprey, respectively. 2P wake configurations were observed for anguilliform swimmers at all Strouhal numbers investigated.

Several computational studies of oscillating bodies have also been done, notably that of Guglielmini (2004) who simulated the experiments of von Ellenrieder *et al.* (2003) and observed a simple chain of vortex rings at  $St = 0.175$  and a more complicated double chain of vortical structures at  $St = 0.35$ . Dong *et al.* (2005) and Dong *et al.* (2006), who used an immersed boundary method, computed thrust and efficiency of a pitching and heaving ellipsoidal wing and found them to depend on Strouhal number, aspect ratio, and Reynolds number.

Zhu *et al.* (2002) used a nonlinear inviscid numerical method to investigate flow structures around a fish-like body (giant danio) and showed that a constructive interaction of the vorticity shed by the body and the vorticity shed by the caudal fin resulted in high thrust, and a opposite destructive interaction resulted in high efficiency. Zhu & Shoele (2008) numerically simulated the flow around a flexible trapezoidal fin, as shown in figure 1.8, and showed that flexibility of the caudal fin can increase propulsive efficiency. Borazjani & Sotiropoulos (2008) numerically simulated the hydrodynamics of an anguilliform swimmer and showed that they produce thrust more smoothly than carangiform swimmers.

Of particular interest to the current work are the experimental results of Buchholz & Smits (2006) and Buchholz & Smits (2008), who performed both wake studies and time-averaged thrust measurements on a series of low-aspect ratio rectangular pitching panel. As seen in figure 1.9, wake studies showed that at low Strouhal number the 2S structure was present in the wake, but as the Strouhal number increased the wake bifurcated into a 2P structure. Additionally, a spanwise compression of the wake was observed at all Strouhal numbers. The model of the vortex skeleton of these wakes is shown in figure 1.10. Thrust measurements showed that at the same  $St$ , thrust increased with increasing aspect ratio, and also exhibited a weaker inverse relationship with pitching amplitude. Buchholz proposed that the appropriate additional scaling parameter for the effect of three-dimensionality is the ratio of amplitude to span  $A/S$ .

To study the vortical structure generation and evolution in the wakes of the rigid pitching panels, we compute the Direct Lyapunov Exponent (DLE) from time-resolved data and identify the Lagrangian coherent structures. This method was introduced by Haller & Yuan (2000) and has proven to be useful in characterizing aperiodic flows. A more complete review of the method and previous implementations is given in chapter 2.

Also in the current work, we investigate the effects of three-dimensionality by studying the unsteady pressure distributions on low-aspect ratio rectangular pitching panels in order to describe the forces on the panel surface. Such measurements do not seem to be presently available, although Hilaire & Carta (1983) performed a survey of pressure experiments on an oscillating NACA 0012 airfoil to explore the effects of sweep angle and mean angle of attack. The focus of their work was the separation at the leading edge of the airfoil, and not the unsteady forces on the airfoil. Usherwood *et al.* (2005) performed pressure measurements on flapping pigeon wings by attaching transducers to the wings of flying pigeons. Other pressure measurements

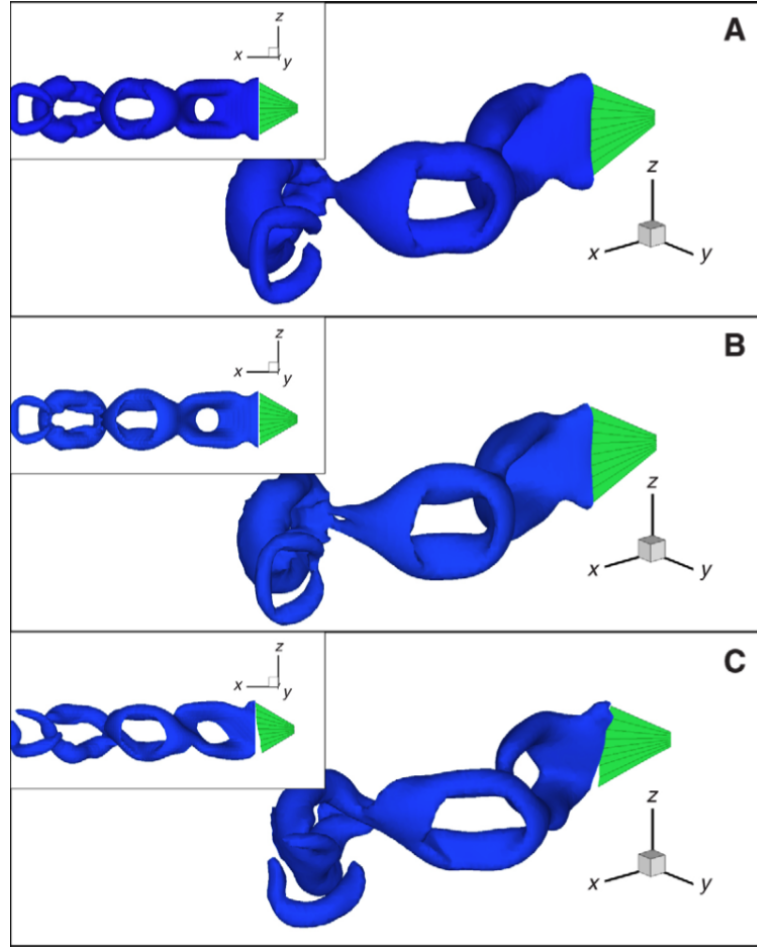


Figure 1.8: Isosurfaces of vorticity magnitude for a trapezoidal fin at  $St = 0.3$ . (a) Rigid trapezoidal panel, (b) spanwise symmetric flexibility, and (c) spanwise asymmetric flexibility. From Zhu & Shoele (2008).

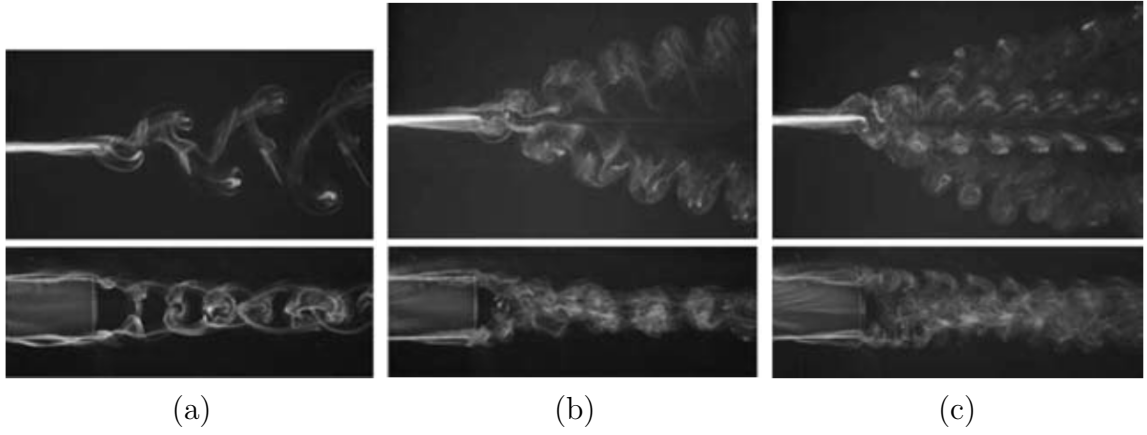


Figure 1.9: Top and side views of dye flow visualization of low-aspect-ratio rectangular panel wakes at different Strouhal numbers,  $Re_c = 640$ . (a)  $St = 0.23$ , (b)  $St = 0.43$ , and (c)  $St = 0.64$ . From Buchholz & Smits (2005)

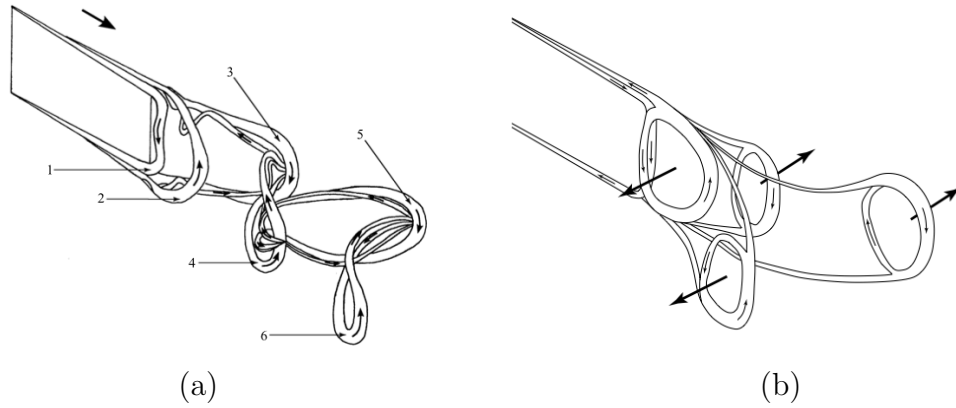


Figure 1.10: Vortex skeletons of the wake behind rigid rectangular pitching panels. (a)  $St = 0.23$  and (b)  $St = 0.43$ . From Buchholz & Smits (2008)



and calculations have been conducted on airfoils to investigate the effects of flutter (Singh *et al.*, 1989; Ardonneau, 1989), but the behavior of the unsteady pressure and its relationship to thrust production does not seem to have received much attention.

## 1.4 Overview and resulting publications

- In **Chapter 2**, we review the concept of the Direct Lyapunov Exponent (DLE) and Lagrangian Coherent Structures (LCS). The merits of a Lagrangian vortex investigation, as opposed to Eulerian, are considered as both are performed on established numerical solutions of wall-bounded flows: an isolated hairpin vortex (§ 2.3) and a fully turbulent channel flow (§ 2.4).

**Resulting publication:** GREEN, M. A., ROWLEY, C. W., & HALLER, G. Detection of Lagrangian coherent structures in three-dimensional turbulence. *J. Fluid. Mech.* **572**, 111–120.

- **Chapter 3** contains details of the experimental setup and techniques used in the course of the current work.
- In **Chapter 4**, we report the results from a series of pressure measurements on rigid rectangular panels. Interpretation of this data led us to propose a new scaling parameter in § 4.1.3 that, when plotted against Strouhal number, neatly collapses both pressure measurements and previously published time-averaged thrust and circulation measurements.

**Resulting publication:** GREEN, M. A. AND SMITS, A. J. Effects of three-dimensionality on thrust production by a pitching panel. *J. Fluid. Mech.* **615**, 211–220.

- In **Chapter 5** we use dye flow visualization and Digital Particle Image Velocimetry (DPIV) to investigate the fluid flow around rigid trapezoidal panels. We describe an observed phenomenon in § 5.1 in which coherent structures generated at the swept edges remain “trapped” on one side of the panel. This occurs on trapezoidal panels at specific Strouhal numbers, dependent on the panel geometry. In § 5.2, temporally- and spatially-resolved 2D DPIV data sets were acquired for the panel pitching at two different Strouhal numbers, one above and one below the critical Strouhal number for this panel. To analyze the DPIV results, both Eulerian (§ 5.2.1) and Lagrangian (§ 5.2.2) methods are used.
- A discussion of the results, concluding remarks, and plans for future work are given in **Chapter 6**.

## Chapter 2

# Lagrangian coherent structure analysis

In this chapter, we use Direct Lyapunov Exponents (DLE) to identify Lagrangian coherent structures in three-dimensional unsteady flows. Previous work on flow structure identification has been primarily Eulerian, i.e., it has been concerned with the spatial structure of quantities derived from the instantaneous velocity field and its gradient. The resulting Eulerian coherent structure criteria have been broadly used in flow structure identification, although none has emerged as a definitive tool of choice. A representative sample of these criteria is given in § 2.1.

By contrast, Lagrangian methods identify flow structures based on the properties of fluid particle trajectories. An immediate advantage of these methods is their objectivity: they remain invariant with respect to rotation of the reference frame, whereas Eulerian criteria are only invariant for translational coordinate transforms (Galilean invariant.) A further advantage of Lagrangian methods is their insensitivity to short-term anomalies in the velocity field. Computing Lagrangian quantities, however, can be computationally expensive. This chapter will explore the additional information that an LCS analysis provides, making it worth the added computational cost.

A majority of the results and analysis presented in this chapter was published in Green *et al.* (2007).

### 2.1 Eulerian methods

Eulerian coherent structure criteria are typically formulated in terms of the invariants of the velocity gradient tensor  $\nabla \mathbf{u}$ . Two example criteria to be discussed here include the  $Q$ -criterion and the swirling strength criterion.

The  $Q$ -criterion, developed by Hunt *et al.* (1988), locates regions where rotation dominates strain in the flow. Letting  $\mathbf{S}$  and  $\mathbf{\Omega}$  denote the symmetric and antisymmetric parts of  $\nabla \mathbf{u}$ , one defines  $Q$  as the second invariant of  $\nabla \mathbf{u}$ , given for incompressible flow by

$$Q = \frac{1}{2}(\|\mathbf{\Omega}\|^2 - \|\mathbf{S}\|^2), \quad (2.1)$$

where  $\|\cdot\|$  is the Euclidean (or Frobenius) matrix norm. A coherent vortex is defined as a region where  $Q > 0$ , although vortices identified by this criterion are usually visualized as isosurfaces of the scalar quantity at some level  $> 0$ .

The swirling strength criterion, employed by Zhou *et al.* (1999), seeks flow structures in regions where  $\nabla \mathbf{u}$  has a complex pair of eigenvalues, which indicates locally spiraling streamlines. The swirling strength  $\lambda_{ci}^2$  is then defined as the squared magnitude of the imaginary part of the complex eigenvalues. Coherent vortices are defined as areas where the swirling strength is greater than some positive threshold.

Other Eulerian criteria have also been used for structure identification, and some of these have been compared to Lagrangian criteria in Haller (2005). These include the  $\Delta$ -criterion (Chong *et al.*, 1990), which defines the vortex as those regions where  $\nabla \mathbf{u}$  has complex eigenvalues, and the  $\lambda_2$ -criterion (Jeong & Hussein, 1995), which identifies pressure minima within two-dimensional subspaces. Additionally, Chakraborty *et al.* (2005) proposed using the ratio of the real and imaginary parts of the complex eigenvalues of  $\nabla \mathbf{u}$  to refine the definition of a vortex core.

These criteria identify similar structures in most flows, but they share several disadvantages. In particular, as Haller (2005) points out, though they are invariant with respect to Galilean transformations, they are not invariant to time-dependent rotations, and thus are not objective (frame-independent). Furthermore, in practice all the above Eulerian criteria require a user-defined threshold to indicate the regions where a structure exists. For example, 3D plots of the  $Q$  criterion are often plotted as isosurfaces defined at percentage of  $Q_{max}$ . The boundaries of the structures depend on the selected threshold, lending subjectivity to the definition of the size or boundary of structures.

## 2.2 Lagrangian criterion

The Lagrangian criterion used in this study is the Direct Lyapunov Exponents (DLE) method (Haller, 2001), also referred to in the literature as Finite Time Lyapunov Exponents (FTLE), which differs only in the scaling of the calculated value. At each point in space, this scalar is a measure of the maximum rate of separation of neighboring particle trajectories initialized near that point. More precisely, if  $\mathbf{x}(t, \mathbf{x}_0, t_0)$  denotes the position of a particle at time  $t$  that began at position  $\mathbf{x}_0$  at time  $t_0$ , one defines a coefficient of expansion  $\sigma_T$  as the square of the largest singular value of the deformation gradient  $\partial \mathbf{x}(t_0 + T, \mathbf{x}_0, t_0) / \partial \mathbf{x}_0$ :

$$\sigma_T(\mathbf{x}_0, t_0, T) = \lambda_{max} \left( \left[ \frac{\partial \mathbf{x}(t_0 + T, \mathbf{x}_0, t_0)}{\partial \mathbf{x}_0} \right]^T \left[ \frac{\partial \mathbf{x}(t_0 + T, \mathbf{x}_0, t_0)}{\partial \mathbf{x}_0} \right] \right). \quad (2.2)$$

The DLE field is then defined as

$$\text{DLE}_T(\mathbf{x}_0, t_0, T) = \frac{1}{2T} \log \sigma_T(\mathbf{x}_0, t_0). \quad (2.3)$$

Since the maximum eigenvalue is used in the definition of  $\sigma_T$ , direction information is not retained in  $\text{DLE}_T(\mathbf{x}_0, t_0)$ . A point  $\mathbf{x}_0$  will have a high DLE value if there is a great amount of expansion in one direction, even if there is compression in all other directions. For incompressible flows, in which pure compression is not possible,  $\text{DLE}_T \geq 0$ .

Regions of maximum material stretching generate local maximizing curves (ridges) for the DLE field. The converse is not true: local maxima of the DLE field may indicate either locally maximal stretching or locally maximal shear. Trajectories that stretch relative to each other when advected in negative time converge in forward time. This is analogous to passive scalars in a fluid flow collecting in coherent structures observed in flow visualization. Therefore, if the DLE is calculated by integrating trajectories in backward time ( $T < 0$ ), ridges in the DLE field may indicate attracting material lines, or attracting Lagrangian coherent structures (attracting LCS) (Haller & Yuan, 2000). Integrating trajectories in forward time ( $T > 0$ ) may produce DLE ridges that mark the location of repelling LCS. The ridges are confirmed as hyperbolic LCS, instead of regions of maximal shear, by calculating the strain normal to the ridge. This is discussed further in § 2.3.1. Quantitative criteria for defining ridges in 2D DLE fields, and proof that they are indeed material lines that are advected with the flow, are given by Shadden *et al.* (2005).

### 2.2.1 Example: Hill's spherical vortex

These positive-time and negative-time LCS delineate the boundary between qualitatively different regions in the flow (Shadden *et al.*, 2006). To illustrate this, we look at a classic vortex flow: the Hill's spherical vortex. This case is an analytic solution of the Euler equations that yields an axisymmetric spherical vortex in a uniform flow. The streamfunction in cylindrical coordinates  $(r, \varphi, z)$  is given in two parts, for inside and outside the sphere of radius  $a$ , by Newton (2001),

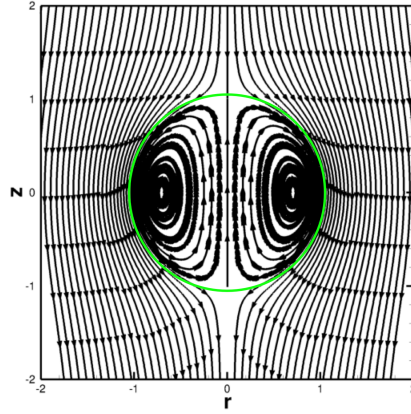
$$\begin{aligned}\Psi_{in} &= \frac{\alpha}{10}(a^2 - z^2 - r^2) & z^2 + r^2 < a^2 \\ \Psi_{out} &= -\frac{\alpha}{15}a^2r^2 \left(1 - \frac{a^3}{(z^2+r^2)^{(3/2)}}\right) & z^2 + r^2 < a^2.\end{aligned}\tag{2.4}$$

The velocity field is given by,

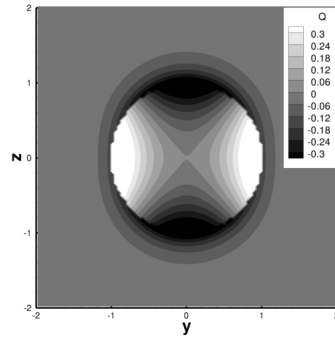
$$u_r = -\frac{1}{r} \frac{\partial \Psi}{\partial z} \quad u_\varphi = 0 \quad u_z = \frac{1}{r} \frac{\partial \Psi}{\partial r}.\tag{2.5}$$

For reference, the streamlines in a two-dimensional cut through the middle of the spherical vortex are shown in figure 2.1(a). The contour plots of two Eulerian criteria, the Q criterion and swirling strength, are shown in figure 2.1(b-c). While the Eulerian criteria highlight the core of the vortex ring, the circular boundary between the inner flow and that which convects with the outer flow is not present in the plot of the swirling strength, and does not exist for a positive threshold of the Q criterion.

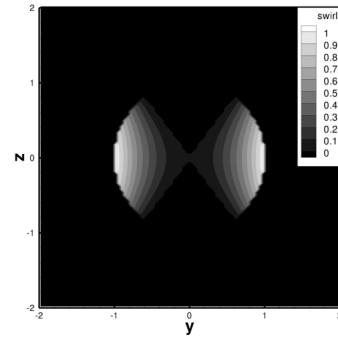
Figure 2.2 shows the negative-time DLE field for a cross section of Hill's spherical vortex, for two different integration times. The boundary of the spherical vortex is



(a)



(b)



(c)

Figure 2.1: Two-dimensional cut through Hill's spherical vortex with radius  $z^2 + y^2 = 1$ . (a) Streamlines with vortex radius indicated by a green circle, (b) Q criterion, and (c) swirling strength.

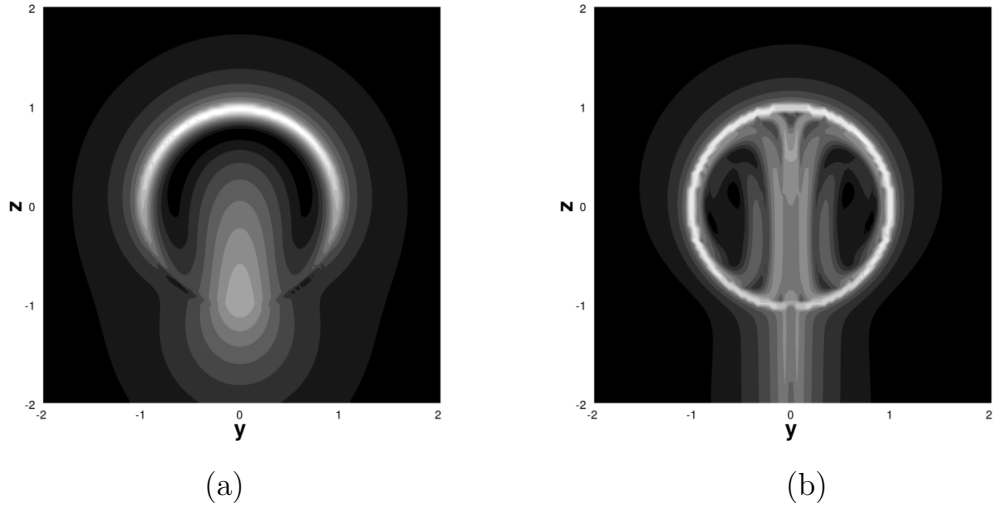


Figure 2.2: Negative-time DLE plots of Hill's spherical vortex, with white showing maximum values. (a) short integration time, and (b) long integration time.

clear. In Figure 2.2(a), the complete boundary is not seen, but as the integration time is increased, the whole boundary emerges and the LCS becomes sharper and clearer. The integration time can be increased or decreased depending on the amount of detail desired from the calculation, but the location of the ridge indicating the boundary of the vortex does not change.

The complete structure of the spherical vortex can also be obtained by plotting both the positive- and negative-time DLE fields. This is done in figure 2.3. The contour for both data sets are only plotted in those regions where  $DLE > 50\%$  of its maximum. The full circular boundary of this vortex is present, and both the positive- and negative-time LCS would sharpen with increased integration time.

Past applications of the DLE as a structure identification tool in two dimensions include LCS in two-dimensional quasi-geostrophic turbulence (Haller & Yuan, 2000), LCS near the stratospheric polar vortex (Koh & Legras, 2002), LCS in freely decaying two-dimensional turbulence (Lapeyre, 2002), and LCS in a magnetically forced two-dimensional conducting fluid experiment (Voth *et al.*, 2002). Lekien & Leonard (2004) used the DLE method to find coherent structures in the currents of Monterey Bay in California based on radar data, and Shadden *et al.* (2006) employed DLE to identify the structure of a piston-generated vortex ring and also captured the vortex ring wake structure of a jellyfish from two-dimensional DPIV data. More recently, Lipinski *et al.* (2008) studied vortex shedding on a two-dimensional airfoil using LCS, and Shadden *et al.* (2007) used LCS to develop a framework for the characterization of the transport and mixing around a laminar vortex ring.

The DLE from discrete data has been shown to be robust and relatively insensitive to imperfect velocity data as long as the errors remain small in a special time-weighted norm (Haller, 2002). Three dimensional DLE has been computed by Haller (2005) on two established analytic flow solutions and Shadden & Taylor (2008) performed an

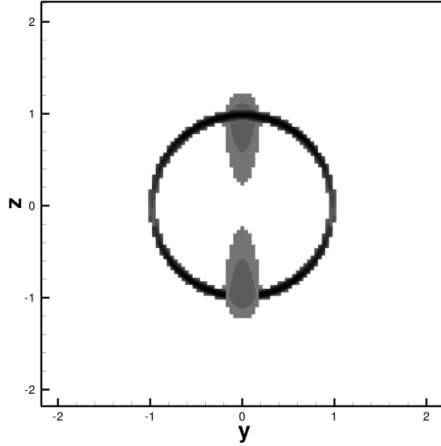


Figure 2.3: Positive- and negative-time DLE field for Hill’s spherical vortex. Contour is only shown for  $DLE > 50\% DLE_{max}$

LCS analysis on three-dimensional computational solutions of the blood flow through abdominal aortic aneurysms. Here, we apply DLE to physically relevant turbulent flows in the interest of detailing structures until now investigated using only Eulerian methods.

In this study, much of the focus is on finding attracting material structures, which correspond to structures seen using flow visualization experiments. Accordingly, the majority of the DLE calculations shown here use a negative integration time (nDLE.) A more thorough understanding of the dynamics of these structures is obtained by including consideration of the positive-time DLE (pDLE), and this is addressed in § 2.3.2. To differentiate DLE fields and LCS calculated using positive and negative integration times, the convention of  $pLCS$  and  $nLCS$  will be used in the remainder of this document.

## 2.3 Isolated hairpin vortex

We begin by studying a single hairpin vortex, a structure commonly found in turbulent wall-bounded flows (Theodorsen, 1955). For this case, the Eulerian and Lagrangian criteria may be compared in an unsteady flow in which the structure location and qualitative shape are known a priori from previous numerical results and experiments, such as those of Head & Bandyopadhyay (1981) and Smith & Walker (1991).

The method we use to extract a single hairpin vortex was introduced by Zhou *et al.* (1999). First, a Direct Numerical Simulation (DNS) of a fully developed turbulent channel flow was performed in a domain periodic in the streamwise and spanwise directions, using the method of Kim *et al.* (1987). The calculation used a spectral collocation method with Fourier modes in the streamwise and spanwise directions and with Chebyshev modes in the wall-normal direction, and a second order Adams-

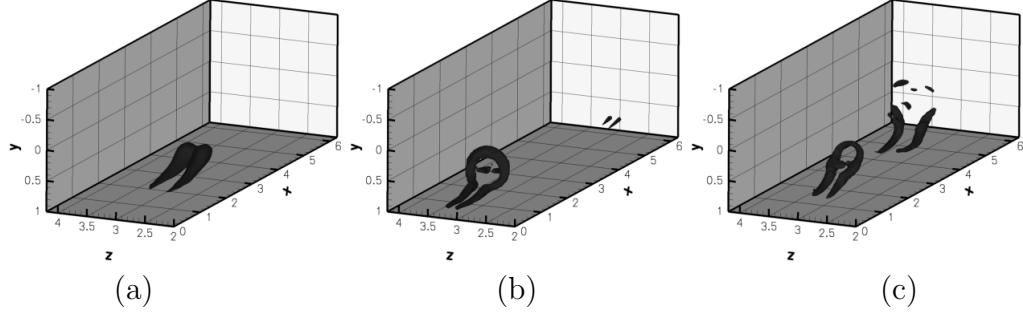


Figure 2.4: Evolved isolated hairpin vortex generated by linear stochastic estimation, plotted using 10% max  $\lambda_{ci}^2$ . (a) Initial condition, (b)  $t^+ = 63$ , and (c)  $t^+ = 171$

Bashforth time march. Our data was validated by comparing the mean profiles, Reynolds stresses, log law, and rms velocity fluctuations, against the original calculation of Kim *et al.* (1987). The Reynolds number based on wall friction velocity and channel half width  $\delta$  was  $Re_\tau = 180$ , with a grid resolution of  $128 \times 129 \times 128$  points and a domain of length  $2\pi\delta$  in both streamwise and spanwise directions.

Next, statistics from this simulation are used to extract a single hairpin vortex. One signature of a hairpin vortex is a fluctuation velocity vector in the second quadrant ( $u' < 0, v' > 0, w' = 0$ ) at the location of the vortex. As in Zhou *et al.* (1999), linear stochastic estimation is used to identify the statistically most probable flowfield that has a specified velocity, here  $(u', v', w') = (-8.16, 3.45, 0)$ , at a prescribed point in the flow (here, a wall-normal location of  $y^+ = 49$ ). The resulting most probable flowfield is used as an initial condition for the DNS solver to study the evolution of the structure.

Figure 2.4(a) shows the iso-surface of the swirl criterion (10% max value) for the initial condition generated by the procedure described above. This structure was evolved in time, and the structure based on 10% max swirl is shown in figures 2.4(b–c) at two later time instants, showing the formation of a hairpin vortex, and the subsequent development of a secondary hairpin. The threshold for these plots (i.e., the value of the level set of  $\lambda_{ci}$ ) was chosen to correspond to Zhou *et al.* (1999). As this hairpin vortex develops into a packet, there are only small differences in structure for different threshold values, but the size of the structure varies. Swirl is the only Eulerian criteria plotted here because for appropriate thresholds, there was little distinction among Eulerian criteria plots.

Figures 2.5 (a–d) show three two-dimensional plots of the negative-time DLE field evaluated at  $t^+ = 63$ , as well as the location of these planes in the three dimensional volume. In this work, two-dimensional planes are often used to study the 3D Lagrangian structure because, while we expect the LCS to depict the boundaries of the structure, these are not constant value surfaces. Therefore, we reconstruct the Lagrangian structure skeleton using data from a small number of two-dimensional planes. Isosurfaces of DLE are also useful to illustrate fully 3D structures boundaries, as seen in figure 2.7, but they can also obscure interesting detail within the structures. The resolution of the DLE data is higher than that of the Eulerian, as the



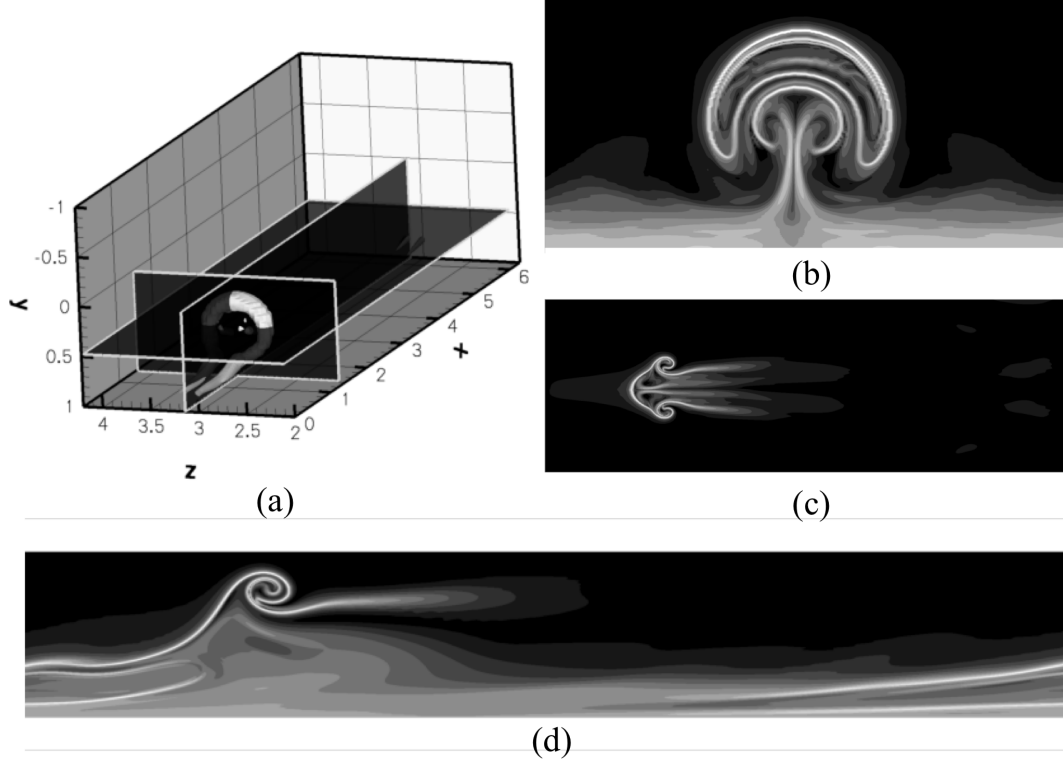


Figure 2.5: Two dimensional nDLE plots of the isolated hairpin evolved to  $t^+ = 63$ . (a) 10% max  $\lambda_{ci}^2$  superimposed with location of the three planes, (b) constant-streamwise ( $y$ - $z$ ) plane, (c) constant wall-normal ( $x$ - $z$ ) plane ( $y^+ = 98$ ), and (d) constant-spanwise ( $x$ - $y$ ) plane

grid of trajectories to be integrated is not restricted to be the same size as the DNS grid. For the plots of the isolated hairpin, the DLE plot resolution is greater by a factor of 6 in both dimensions of the calculated plane. Only a portion of the domain is plotted, including the entire streamwise domain span, half the channel width, and the middle third of the spanwise extent. All the calculations presented used on the order of 500–1000 instantaneous data sets for each plot.

The plots in Figure 2.5 show the results from a nDLE calculation using an integration time of  $t^+ = 45$ . A distinct nLCS is seen as the boundary of the hairpin vortex. As with the Hill’s spherical vortex, using a larger integration time introduces detailed informations about the dynamics inside the structure boundaries, but the size and outer shape of the structure do not change. Here, the (backward) integration time may be at most  $t^+ = 63$ , as of course we cannot evolve backward in time earlier than the initial condition. At this moment in the hairpin evolution, not many differences exist between the nDLE results using longest possible integration time and those using an intermediate integration time.

In figure 2.5(c) and (d) we see the nDLE field in a plane normal to the vorticity in the core of the hairpin vortex. The outer boundary of the structure is apparent, as

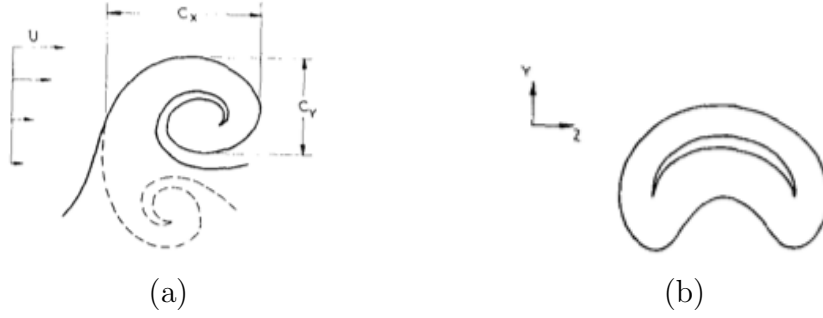


Figure 2.6: Sketches from Falco (1977) of the “typical eddy” in planes (a) parallel to the streamwise flow and normal to the wall, and (b) normal to the streamwise flow and normal to the wall.

is internal structure due to rotational shearing. This structure in figure 2.5(c) bears a strong resemblance to the sketched structure of a “typical eddy” proposed by Falco (1977) based on oil droplet flow visualization in a two-dimensional plane of the same orientation through a turbulent boundary layer. This sketch is shown in figure 2.6(a). These “typical eddies” were described as “highly coherent three-dimensional” eddies, and were often located on the upstream side of large scale motions. Similarly, the hairpin vortex studied here evolves into a larger “packet,” or large scale collection of multiple individual hairpin structures.

Additionally in figure 2.5(c), there is a clear nLCS that extends upstream (to the left) of the hairpin head. This marks the boundary between the fluid that is convecting with the outer flow, and that which has been entrained in the counter-rotating legs of the hairpin vortex. Figure 2.5(b) shows the nDLE in the plane normal to the streamwise flow as the hairpin head cuts through it. Again, the sketch from Falco (1977) based on flow visualization matches closely.

Figure 2.7 illustrates a comparison between the nLCS and the Eulerian criteria. In figure 2.7(a), a “skeleton” of points where the nDLE is greater than 60% of its maximum is shown over level sets of swirling strength  $\lambda_{ci}$ . The extent of the Eulerian structure depends on the threshold used, but the boundary indicated by nDLE is independent of integration time or any other parameters. Although some Eulerian criteria approximate these structures better than others, none capture interior detail like nDLE does. Comparable level sets of 60% max nDLE are plotted in figure 2.7(b). The three-dimensional nDLE field was calculated on one fifth the channel domain in the streamwise and spanwise directions with four times the resolution of the DNS. In the wall-normal direction, nDLE was calculated on one half the domain with six times the resolution. The three-dimensional structure marks the line towards which fluid that will be entrained into the hairpin structure is attracted. The two-dimensional “skeleton” is again plotted to illustrate the internal nLCS detail under the outer shell. As previously noted, the iso-surface is not an accurate depiction of the nLCS, but at 60% maximum nDLE value, the level-set approximates the nLCS surface.

As the single hairpin evolves further, it induces the formation of a secondary

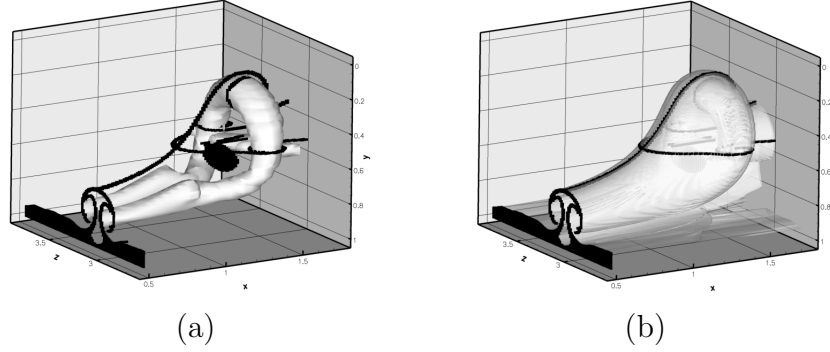


Figure 2.7: (a) Isosurfaces of 10% max  $\lambda_{ci}^2$  (black) superimposed on three planes of nDLE  $\geq 60\%$  max (white) (b) Translucent isosurfaces of 60% max nDLE superimposed on three planes of nDLE (black)  $\geq 60\%$  max (white)

hairpin vortex upstream of the primary hairpin. Eventually, this structure evolves into a family or “packet” of structures. The nDLE fields at time  $t^+ = 171$  (calculated for an integration time of  $t^+ = 45$ ) are shown in Figure 2.8. Here, the 10% max swirl iso-surface is plotted for reference, and Figures 2.8(b–d) show three two-dimensional plots of negative-time nDLE, demonstrating the ability of this Lagrangian method to capture the whole packet of hairpin vortices in great detail.

A study of the LCS dependence on integration time was performed on the constant-spanwise ( $x$ - $y$ ) plane, and those results are shown in Figure 2.9. The increasing detail in the plot gives insight into the behaviour of the particles within the vortex head. For longer integration times, a spiral curve of maxima develops as a result of the shearing rotational fluid flow in that region, and the nLCS indicate the layers of fluid that are sheared apart by the rotation in the hairpin head.

### 2.3.1 Formation of secondary hairpin

In addition to structure identification and characterization, an analysis of hyperbolicity in the nDLE field in this flow yields detailed information about how the hairpin vortex evolves into a packet, a phenomenon described by Zhou *et al.* (1999). In particular, it is observed that the birth of a secondary hairpin structure corresponds to a loss of hyperbolicity (or a bifurcation) along the nLCS.

To investigate this phenomenon, we use Theorem 3 of Haller (2002) to calculate the hyperbolicity (i.e., exponential repulsion or attraction) of the nLCS. Specifically, we compute the rate of strain normal to the surface of the nLCS, given by  $\langle \mathbf{n}, \mathbf{S}\mathbf{n} \rangle$ , where  $\mathbf{n}$  is the unit normal to the LCS. We confirm this ridge to be an attracting material line if this strain rate is negative along its length.

To compute the normal rate of strain, we follow the procedure employed in Mathur *et al.* (2006). Specifically, we first find the locus of points on the LCS surface, using a two-dimensional gradient climb in regions near the local maxima of DLE. The direction of the normal vector is then approximated by calculating the Hessian of the

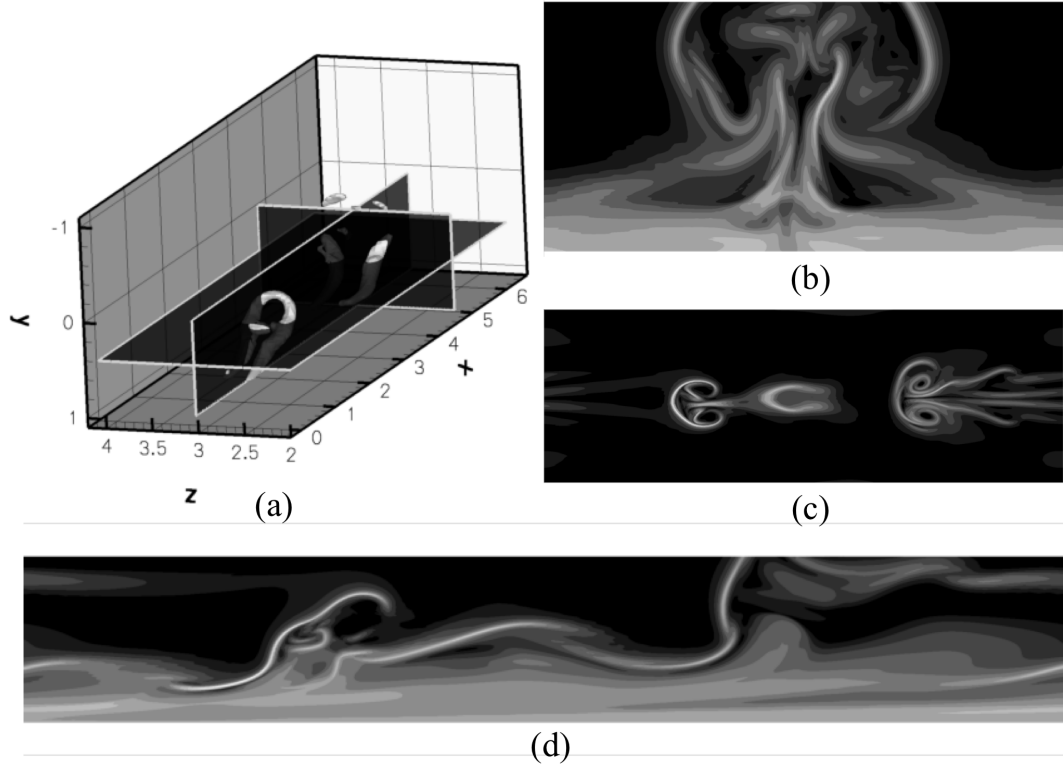


Figure 2.8: Two dimensional nDLE plots of the isolated hairpin evolved to  $t^+ = 171$ . (a) 10% max  $\lambda_{ci}^2$  superimposed with location of the three planes, (b) constant-streamwise ( $y-z$ ) plane, (c) constant wall-normal ( $x-z$ ) plane ( $y^+ = 95$ ), and (d) constant-spanwise ( $x-y$ ) plane

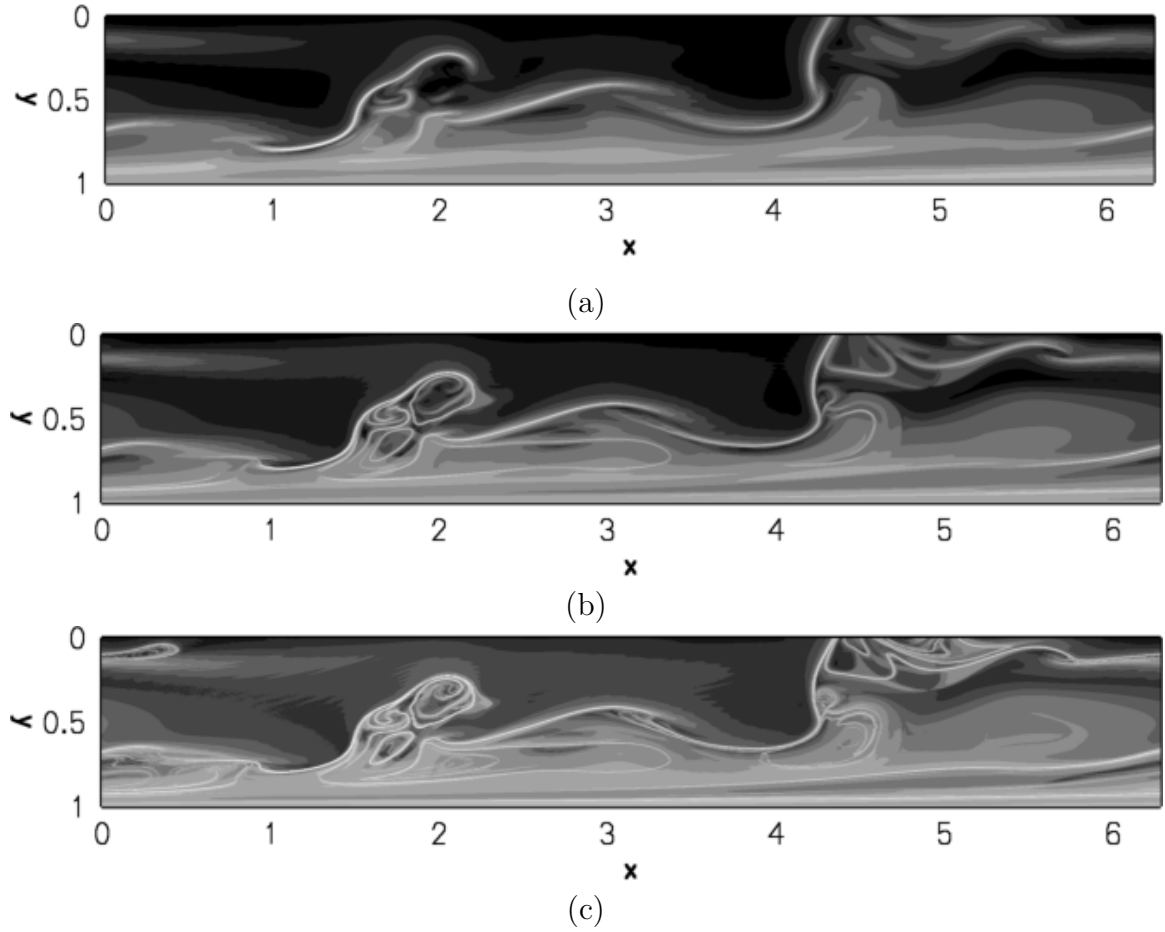


Figure 2.9: nDLE constant-spanwise ( $x$ - $y$ ) planes for three integration times of the isolated hairpin evolved to  $t^+ = 171$ . (a)  $t^+ = 45$ , (b)  $t^+ = 90$ , and (c)  $t^+ = 135$

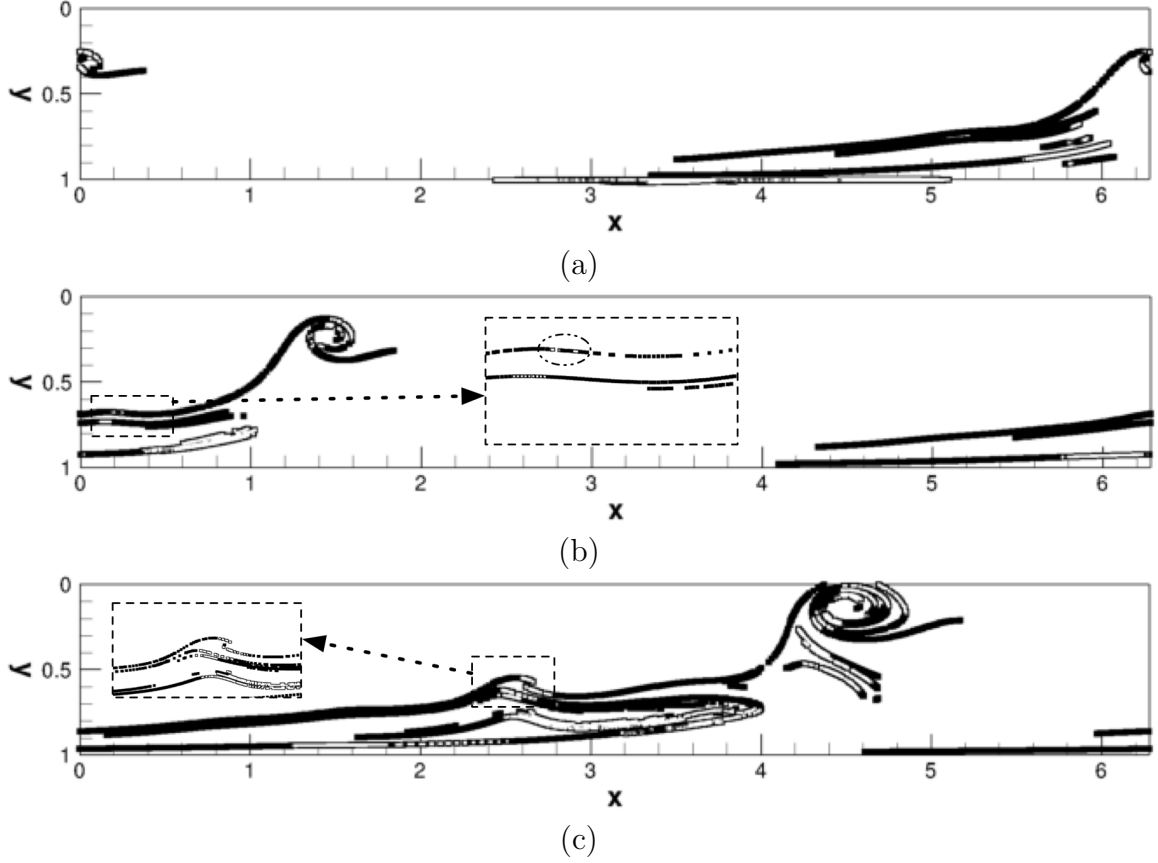


Figure 2.10: Magnitude of  $\langle \mathbf{n}, \mathbf{S}\mathbf{n} \rangle$  along constant-spanwise (x-y) cross sections of nLCS surfaces at times (a)  $t^+ = 45$ , (b)  $t^+ = 63$ , and (c)  $t^+ = 99$ . Negative strain rate (compression normal to the surface) is black; positive strain rate (expansion) is white with black outline.

(three-dimensional) DLE field and using the eigenvector associated with its eigenvalue of largest magnitude.

A plot of the nLCS shaded by the sign of the rate of strain  $\langle \mathbf{n}, \mathbf{S}\mathbf{n} \rangle$  is shown in Figure 2.10 at three time instants in the development of the secondary hairpin vortex. Figure 2.10(a) shows the structure of the hairpin in the mid-span plane at time  $t^+ = 45$ , calculated from an nDLE field which used an integration time of  $t^+ = 45$ . Here, the strain rates normal to the nLCS are negative both upstream and downstream of the vortex head, indicating that this structure is indeed a hyperbolic repelling line.

In Figure 2.10(b), calculated at  $t^+ = 63$  from an nDLE field obtained using an integration time of  $t^+ = 63$ , a hump in the nLCS upstream of the hairpin head legs is seen and a magnified picture is shown in the inset. On the downstream slope of this hump exist small white regions (outlined in black) of non-negative strain rate, highlighted by the dashed oval. The positive rate of strain in these regions corresponds to a loss of hyperbolicity, and this bifurcation of the nLCS indicates the beginning of

the formation of the secondary hairpin structure.

Lastly, Figure 2.10(c) is calculated at time  $t^+ = 99$  from an nDLE field using integration time  $t^+ = 99$ . The secondary hairpin is clearly evolving and its structure begins to resemble that of the primary hairpin, as the nLCS begins to fold and roll up on itself. This is particularly clear in the magnified inset. Additionally, the sign of strain rate along the nLCS alternates in a similar pattern in both hairpin heads, indicative of the shearing rotational flow within.

Zhou *et al.* (1999) discuss at length the physical mechanism that results in the development of this second and subsequent hairpin structures. The Lagrangian criterion, when used in this way, offers a quantitative method for recognizing their formation and interpreting their generation as a loss of hyperbolicity along the Lagrangian coherent structures. For the same initial condition as used in this study, Zhou *et al.* (1999) observe characteristic indications of hairpin development at  $t^+ = 72$ , whereas using nLCS, the bifurcation occurs and can be detected at least as early as  $t^+ = 63$ .

### 2.3.2 Positive-time LCS

While much information about the development of these structures is obtained through the use of the negative-time DLE plots, more information can still be revealed when the positive-time LCS are included in the analysis. Figure 2.11 is a three-dimensional plot of an isosurface in both the positive- and negative-time DLE field which satisfy the corresponding hyperbolicity conditions ( $\langle \mathbf{n}, \mathbf{S}\mathbf{n} \rangle > 0$  for pLCS and  $\langle \mathbf{n}, \mathbf{S}\mathbf{n} \rangle < 0$  for nLCS.) As shown more clearly in figures 2.12(a–b), the two LCS meet and intersect along the outer boundary of this structure, but they do not overlap, as the hyperbolicity criterion can only be satisfied for both the pLCS and nLCS that overlap, but not both. At intersections of the two LCS, highlighted in figure 2.12, the unit normal vector is not the same, and therefore the hyperbolic material lines can coexist.

By plotting both positive- and negative-time LCS together, a more complete boundary of the structure is obtained. One can clearly see not only the lines along which particles collect, but also the separatrices around which particles entrained into the hairpin separate from those which continue to convect with the outer flow. Additionally, the intersections of the pLCS and the nLCS just upstream and just downstream of the hairpin head are significant. If this is plotted in a frame that moves with the upstream intersection point, the flowfield around it resembles that of a saddle point. In figure 2.12(b), the upstream saddle points of the vortex cores are inferred, but unresolved in this calculation.

The vector field of the flow near the hairpin head in this frame is shown in figure 2.13(a), with a black circle marking the approximate location of the saddle point. This resembles what Falco (1977) theorized as the dynamical flowfield structure on the upstream side of large scale motions in a boundary layer. The sketch of his model is shown in figure 2.13(b). In the current work, we refer to these points as time-dependent saddle points.

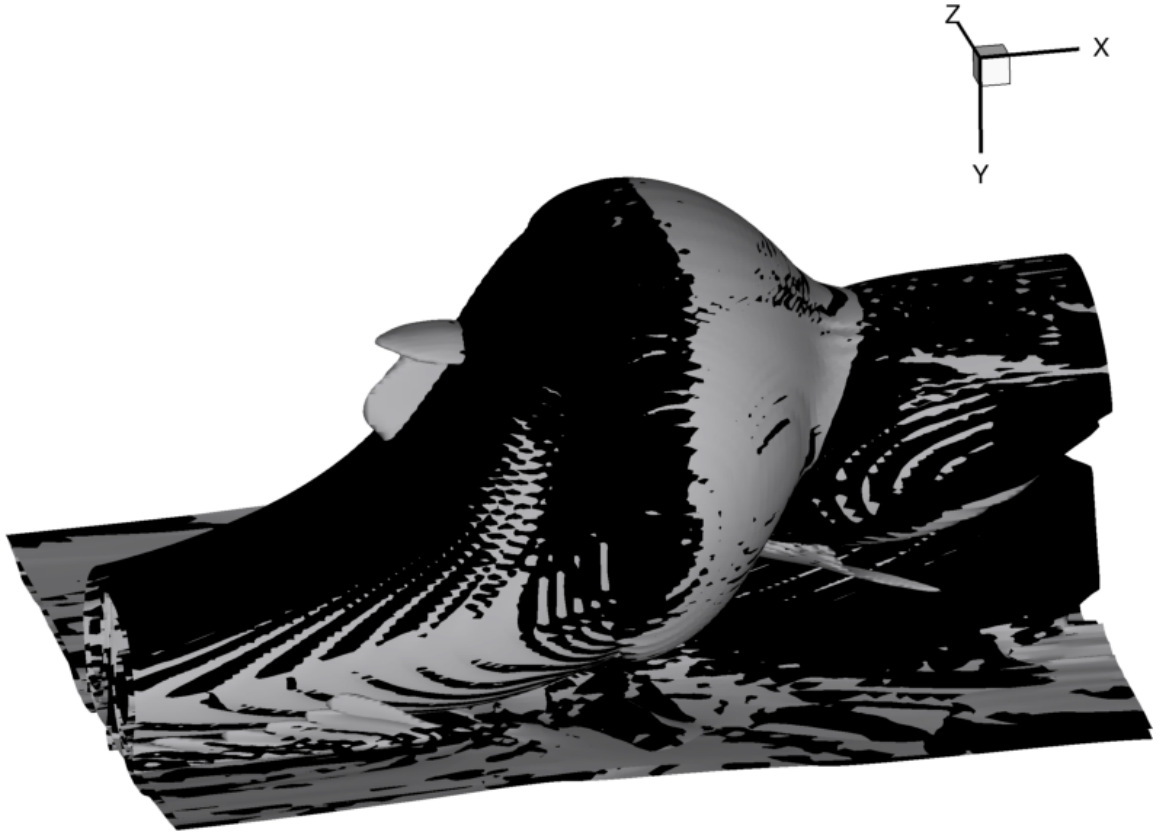
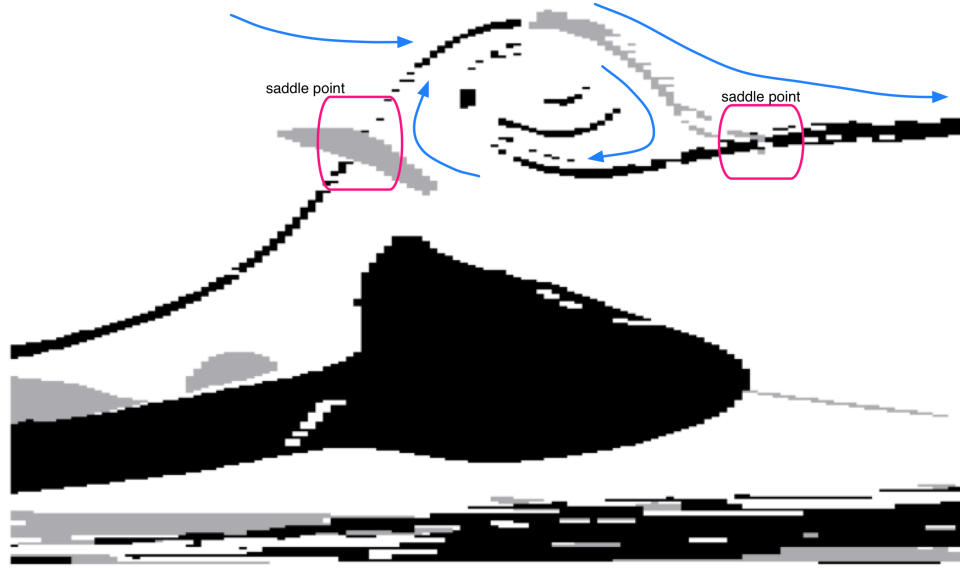
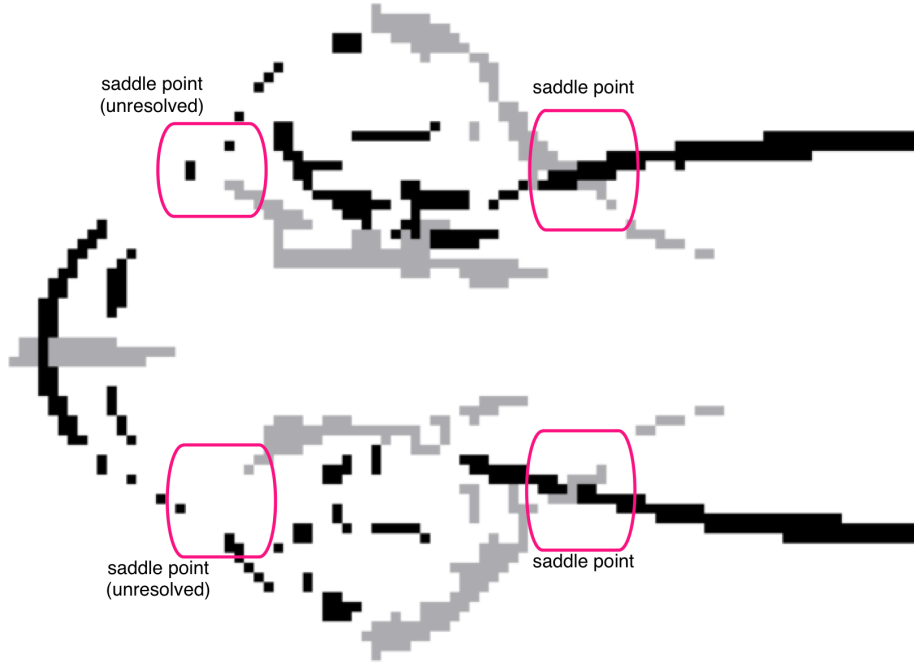


Figure 2.11: Three-dimensional hyperbolic pLCS (gray) and nLCS (black) of the isolated hairpin head, plotted by using isosurfaces of 50% max DLE that satisfy the corresponding hyperbolicity criteria.





(a)



(b)

Figure 2.12: Hyperbolic pLCS (gray) and nLCS (black) of the isolated hairpin head in a two dimensional slice in (a) a constant-spanwise ( $x$ - $y$ ) plane and (b) a constant wall-normal ( $x$ - $z$ ) plane, plotted for regions of  $DLE > 50\%$  maximum value that satisfy the corresponding hyperbolicity criteria. Time-dependent saddle points are highlighted using a pink box. Blue arrows indicate fluid behavior in figure (a).

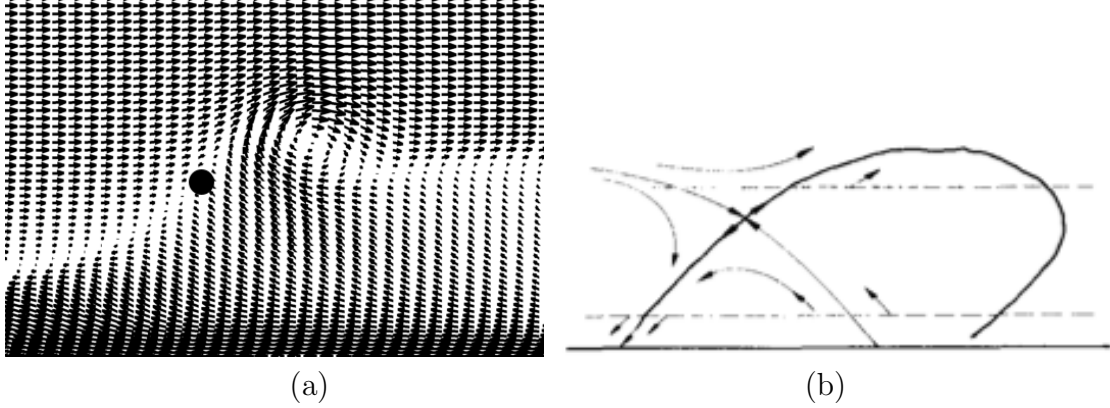


Figure 2.13: (a) Vector field around the hairpin head in a frame that is convecting with the speed of the pLCS and nLCS intersection point. (b) Flowfield sketch of a large scale motion by Falco (1977), with the observer moving with the velocity of the saddle point.

## 2.4 Fully turbulent channel

Finally, the LCS analysis was applied to the fully turbulent channel data. In Figure 2.14(a), the  $Q$  criterion structures are shown as level sets of 1% maximum value. Also shown in this plot are the locations of three two-dimensional planes on which nDLE was calculated. The results are shown in Figure 2.14(b–d). As expected, the nLCS curves are clean and sharp. Several of these curves have the same shape as those of the isolated hairpin in the respective plane, and support the notion that the fully turbulent channel is populated with similar structures. These locations are highlighted by a white box, and can be compared with Figures 2.5–2.8.

If the hyperbolic positive-time LCS are plotted along with the hyperbolic negative-time LCS, patterns similar to that of the isolated hairpin head are noticeable. In figure 2.15, one such structure is highlighted with a white box. This structure is bounded by alternating pLCS and nLCS, with intersections both upstream and downstream of a vortex core piercing through the plane. It is postulated that this is part of the head of a hairpin vortex in this fully turbulent flow. The locations of these intersections are easy to locate in a quantitative sense, and may be useful for future structure identification and tracking in complicated flows.

### 2.4.1 Comparison with Eulerian criteria

In Figure 2.16(a–b), the nDLE and  $Q$  criterion are plotted at the same constant-streamwise ( $y$ - $z$ ) location. This comparison highlights the fact that the nLCS clearly depicts structures in locations where the  $Q$  criterion would not if plotted with a large threshold. Two such structures are marked with white boxes. Also, the finer resolution of the nDLE plot yields more detail than possible with the Eulerian criteria, which require derivatives of the velocity field and are thus restricted to the resolution

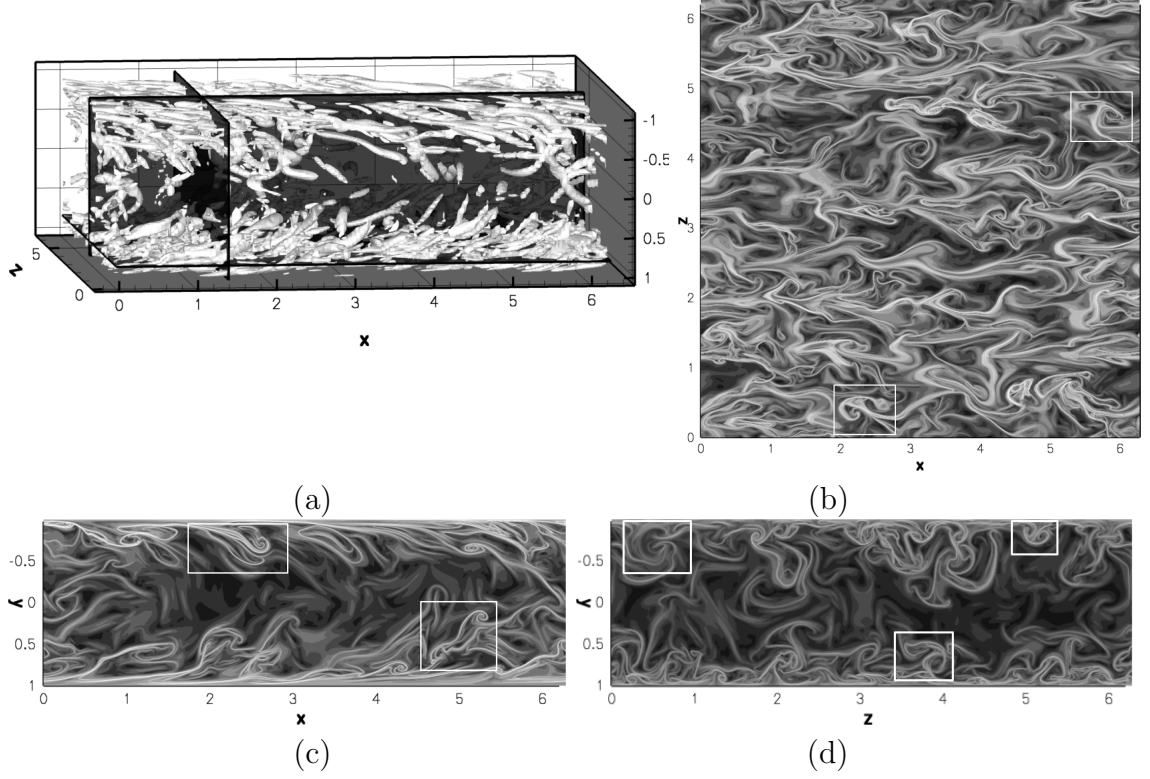


Figure 2.14: Two dimensional nDLE plots of the fully turbulent channel. (a) 1% max  $Q$  superimposed with location of the three planes, (b) constant wall-normal ( $x$ - $z$ ) plane, (c) constant-spanwise ( $x$ - $y$ ) plane ( $y^+ = 33$ ), and (d) constant-streamwise ( $y$ - $z$ ) plane. White boxes highlight structures that resembles corresponding cuts through the isolated hairpin.



Figure 2.15: Hyperbolic pLCS (gray) and nLCS (black) of the isolated hairpin head in a two dimensional slice in a constant-spanwise ( $x$ - $y$ ) plane through the fully turbulent channel, plotted for regions of DLE  $> 50\%$  maximum value that satisfy the corresponding hyperbolicity criteria. A possible hairpin vortex is highlighted by a white box.

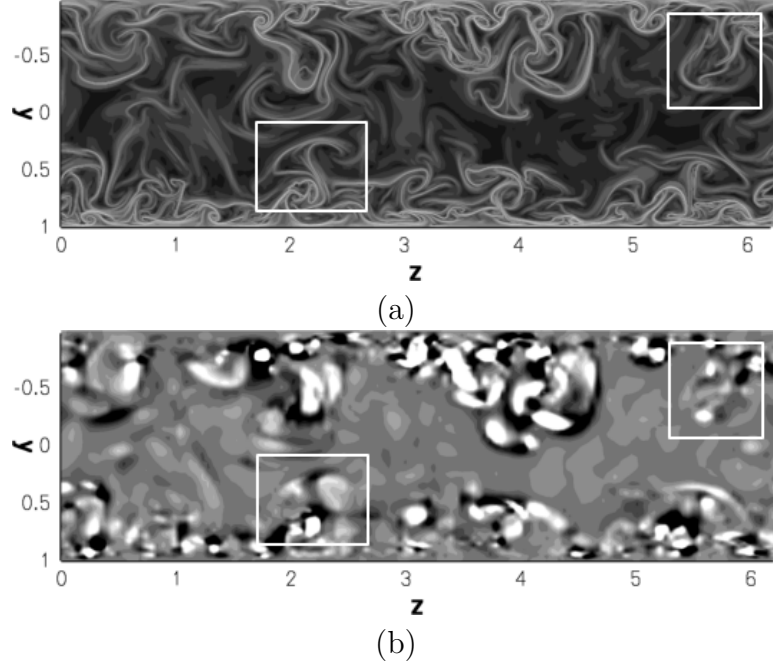


Figure 2.16: Constant-streamwise ( $y$ - $z$ ) planes of (a) nDLE and (b)  $Q$  criterion

of the original data.

## 2.5 Summary

Lagrangian coherent structures (LCS) are identified for various flows in a plane channel, including an isolated hairpin vortex and a fully-developed turbulent flow, by calculating the Direct Lyapunov Exponent (DLE). This Lagrangian method, which yields candidate material lines, captures features of the flow that are familiar from flow visualization experiments, and are also described by various Eulerian criteria currently in use, but the DLE field yields greater detail than existing Eulerian criteria. This is partially because, unlike Eulerian criteria, the DLE may be evaluated on a finer grid than the original velocity data. Additionally, using the Lagrangian criteria, one may quantify the boundary of a vortex as a local maximum of the DLE field. Whereas the size and shape of the Eulerian structures depend on a user-defined threshold, the locations of the LCS are independent of such parameters. Increased integration time yields greater detail, but outer boundaries of the structures do not vary. Lastly, the DLE is truly independent of coordinate frame and would yield the same results for a non-Galilean invariant coordinate transformation.

The development of an isolated hairpin vortex is studied, and it is shown that the birth of a secondary hairpin corresponds to a loss of hyperbolicity along the nLCS. Thus, the Lagrangian criteria can provide a quantitative way of determining when these structures are generated. Previously, such events have been identified using qualitative, visual methods (Zhou *et al.*, 1999), but the Lagrangian method allows

one to detect these events at their earliest stages.

Consideration of the positive-time LCS also proves to be informative, as hyperbolic pLCS, when plotted along with hyperbolic nLCS, yields a more complete structure boundary. Additionally, the characteristic pattern around the core of the hairpin includes two time-dependent saddle points observed as an intersection of the two LCS, and this fact could be used in the future as part of a quantitative method to track vortex cores in more complicated fluid flows.

It is important to point out that while LCS has distinct advantages over commonly used Eulerian criteria, there are a number of factors that suggest that LCS is best used in concert with other methods of analysis. Eulerian criteria, which are calculated using spatial derivatives of the velocity field, are quickly computed and can be used to guide the implementation of the LCS analysis. Furthermore, while the strength of the vortical structures may be inferred from the dynamics of the LCS, the magnitude of Eulerian criteria provide an immediate measure of the relative strength of vortices.

The benefits of the Lagrangian method also come at an expense: Lagrangian calculations are more computationally intensive than any of the Eulerian criteria, as they involve integration of particle trajectories from each point at which the DLE value is desired. Each DLE data set presented in this paper took on the order of 1–2 computational hours, whereas calculation of the Eulerian criteria can take less than a minute. However, as postprocessing represents a relatively small part of the computational time devoted to most flow calculations, this drawback is not severe. Also, large amounts of time-resolved data are necessary for the trajectory integration. For the work presented in this paper, 500–1000 data sets were used for each plot and the database on which the calculations are based used approximately 150GB of hard disk storage. Fluid flows of higher Reynolds number would naturally necessitate a greater cost for the same calculations.

As will be shown in § 5.2.2, these methods may be especially useful for time-resolved experimental data such as Digital Particle Image Velocimetry (DPIV), and may also prove helpful when data sets are too noisy to compute derivatives necessary for the Eulerian criteria. DLE may also be useful as a tool to validate the more easily-implemented Eulerian schemes, and to calibrate appropriate thresholds analysis of different fluid flow systems.

# Chapter 3

## Experimental methods

Water channel experiments were conducted to obtain measurements of the fluid flow around pitching rigid panels in a uniform flow. Pressure measurements were conducted on the surface of rectangular panels of finite aspect ratio in order to examine force generation on the panel surface. Subsequent experiments used rigid panels of trapezoidal planform in order to investigate the effects of the swept edge commonly seen in aquatic animal caudal fins. Flow visualization was first used to determine the large scale flow characteristics, and Digital Particle Image Velocimetry (DPIV) was used to quantify the observed effects. As this flow is three-dimensional, spatially resolved data throughout the volume surrounding the panel was desired and reconstructed from a collection of two-dimensional planes.

### 3.1 Experimental facility

Experiments were conducted in a water channel at Princeton University. The test section of this water channel has a width of 0.46 m and a maximum depth of 0.29 m. An acrylic plate, 12 mm thick and 1.22 m long, was used to cover the free surface and prevent the formation of surface waves which would both influence the flow physics in the test section and distort flow visualization from above. The water channel was capable of speeds up to 0.4 m/s. In this work, experiments were conducted with flow velocities in the range of  $0.012 \text{ m/s} < U < 0.3 \text{ m/s}$

Upstream of the test section, a honeycomb flow straightener, two screens, and a 5:1 contraction were used to condition the flow. During the course of the experiments, the honeycomb and screens were routinely cleared of bubbles and particles using a water jet. Periodically, the screens were removed entirely to clean particles and other sediment that had settled in the upstream reservoir.

As part of his thesis work, Buchholz (2006) characterized the flow quality in the channel. The velocity profile was imaged using a hydrogen bubble wire technique described below, and at the lowest flow velocities ( $\approx 5 \text{ mm/s}$ ), some large inconsistencies were observed. It was proposed that these were associated with thermal plumes caused by heat transfer at the walls of the channel. Using DPIV, turbulence intensities between 12% and 19.5% were calculated from the measured flow field with

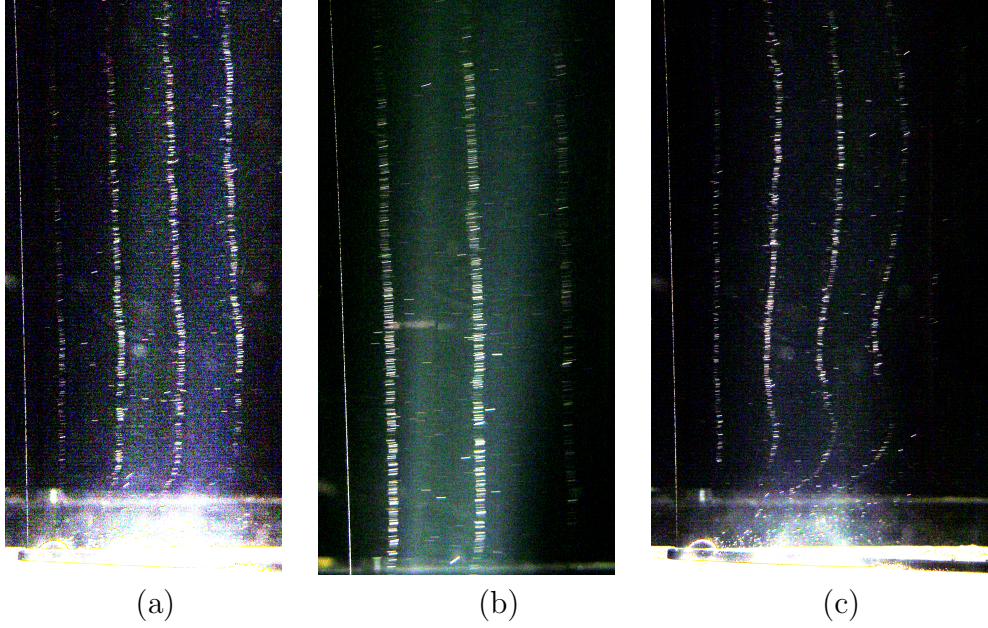


Figure 3.1: Hydrogen bubble wire experiments, flow is from left to right. (a)  $U = 0.012$  m/s, (b)  $U = 0.018$  m/s, and (c)  $U = 0.012$  m/s, showing disruption from thermal plumes.

the panel and fairing removed. For velocities in the moderate Reynolds number range ( $0.057 \text{ m/s} < U < 0.285 \text{ m/s}$ ), turbulence intensities varied between 4.5% and 7.5%.

In the course of the current work, the bubble wire visualizations were repeated to check the quality of the freestream flow. Short duration voltage pulses were applied every 1.25 s to a  $13 \text{ }\mu\text{m}$  diameter tungsten wire installed approximately 0.13 m upstream of the airfoil fairing used to support the pitching panel (see § 3.2). At each voltage pulse, a line of bubbles were released from the wire, and their trajectories depict the flow timelines. Images of the bubbles were acquired using a consumer grade 5.0 megapixel digital camera. Figure 3.1 shows three examples of the images acquired. In figure 3.1(a–b) it is clear that the velocity profiles are clean, straight lines outside of the boundary layer, indicating a low level of turbulence intensity.

The thermal plumes reported by Buchholz (2006) were also observed during the course of experiments, and one example is shown in figure 3.1(c). However, these variations caused by heat transfer from the flashlight used to illuminate the bubbles located at the bottom wall of the channel.

As a quantitative check of the flow quality, the turbulence intensity was calculated using calibration DPIV (§ 3.3.2) image pairs acquired when the panel and fairing were installed in the water channel but stationary and aligned with the flow. The

turbulence intensity is defined as

$$\frac{\sqrt{\overline{u'^2}}}{\overline{U}} = \left( \frac{\overline{(u - \overline{U})^2}}{\overline{U}^2} \right)^{\frac{1}{2}}, \quad (3.1)$$

where  $u$  is the velocity at each point and  $\overline{U}$  is the mean freestream velocity (bars indicate time averages). Calibration data was available for two freestream velocities, and at 2-3 spanwise planes at each velocity. For the turbulence intensity calculation, velocity information was taken from approximately 1200 grid points in each of the 2-3 spanwise planes. At the lower speed at which DPIV was conducted ( $U = 0.036$  m/s), the average turbulence intensity was approximately 1%, with a maximum as high as 7%. At the higher speed ( $U = 0.06$  m/s), the average turbulence intensity was approximately 0.8%, with a maximum of 5%.

In the current work, all quantitative measurements are taken for freestream velocities greater than 0.03 m/s (Pressure measurements:  $Re_c = 3500$ , DPIV:  $Re_c = 2100$ ), at the lower end of the moderate Reynolds number range. Some flow visualization experiments were conducted at flow speeds as low as 0.015 m/s ( $Re_c = 1350$ ).

## 3.2 Panel geometry and kinematics

The rigid panels were made of clear acrylic with a thickness of 2 mm. They were attached at the leading edge to a 4.76 mm diameter pitching shaft. The angular position of the panel was given by a 2048 count-per-revolution quadrature encoder (US Digital model E3-2048-187-IH) mounted to the pitching shaft.

To support the pitching apparatus, the shaft was attached to the trailing edge of a symmetric fairing based on a NACA 0012-64 airfoil, as described by Buchholz & Smits (2008), and shown in figure 3.2. The fairing had a chord length of 50.8 mm, and the trailing edge was truncated to allow the attachment of the pitching shaft. Tapered NACA 0012-64 airfoil trailing edge segments were mounted behind the pitching shaft when the finite aspect ratio panels were in use. The symmetric fairing contained hollow cavities, and a series of 0.79 mm diameter holes were uniformly distributed along 130 mm of the span of the fairing, which enabled dye flow injection upstream of the panel. The panel and the fairing were mounted vertically in the water channel test section, as shown in figure 3.3.

The actuation of the panel used the same mechanism as in Buchholz (2006), which achieved the pitching motion using a four-bar linkage as shown in figure 3.4. In order to adjust the trailing edge pitching amplitude, the motor and crank were mounted on a linear traverse. The amplitude was adjusted by changing the distance between the crank shaft and the pitching shaft along the linear traverse.



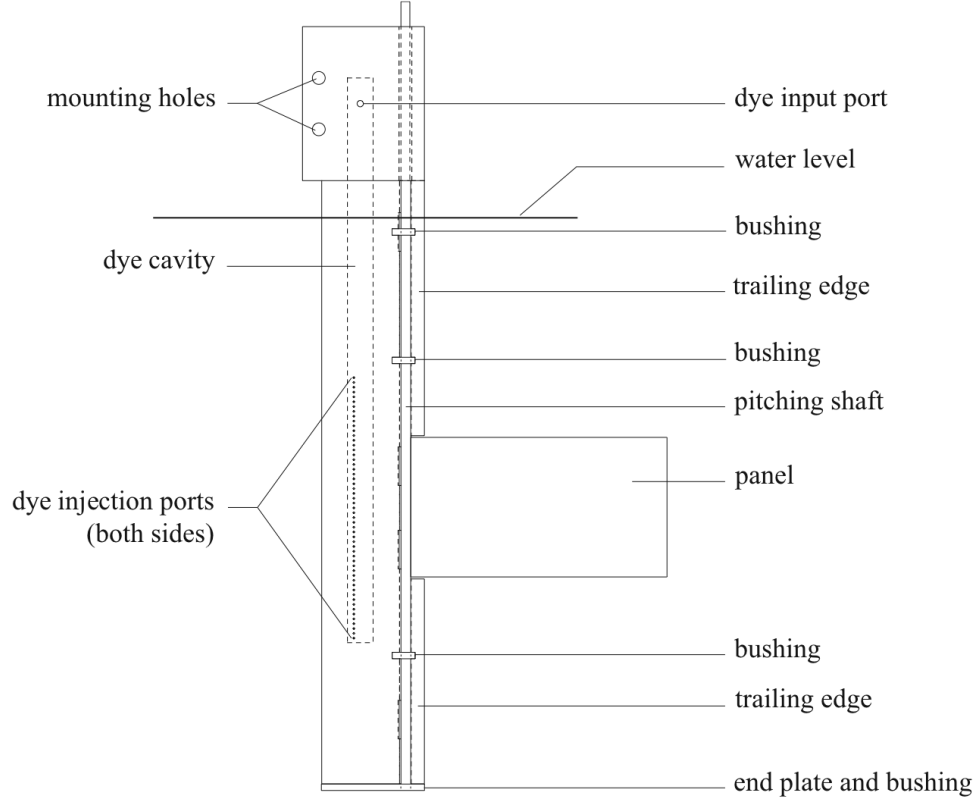


Figure 3.2: Fairing and panel assembly. From Buchholz (2006)

### 3.3 Time-resolved pressure measurements: rigid rectangular panel

Previous measurements of the flow field and time-averaged thrust of a rectangular pitching panel indicated that quantitative and qualitative flow characteristics depend on the Strouhal number and on the aspect ratio of the panel and its trailing edge peak-to-peak pitching amplitude (Buchholz & Smits, 2008). In order to investigate the force distribution on the panel surface, time-resolved pressure measurements on similar pitching panels were desired.

In the previous work, Buchholz & Smits (2008) used four rigid rectangular panels of aspect ratios ( $AR$ ) ranging from 0.54 to 2.38. In the current work, we used panels of two aspect ratios at the extremes of this range (see table 3.1), with the higher aspect ratio panel spanning the water depth in a quasi-two-dimensional configuration. For this panel, any effect of the gaps between the edge of the panel and the top surface plate and bottom wall were evaluated by varying the gap size. Decreasing the gap size from 10 mm to 5 mm at the top and bottom edges showed no change in the pressure at mid-span.

A series of time-resolved pressure measurements were conducted on the rigid rectangular panels. The pressure was measured using a Validyne DP-15 differential pres-

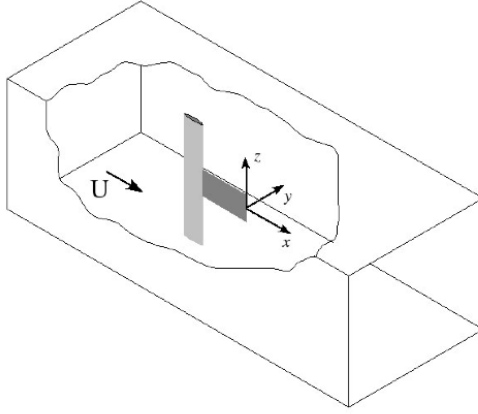


Figure 3.3: Panel and fairing position in water channel test section. From Buchholz (2006)

Panel	chord (c)	span (S)	(AR=S/C)
1	120 mm	6 mm	0.5
2	120 mm	270 mm	2.25

Table 3.1: Rectangular panel dimensions

sure transducer and CD379 carrier demodulator. The pressure transducer had a range of 3.5-5.5 inches of water, with an accuracy of  $\pm 0.25\%$  of full range. Five 3.2 mm diameter pressure ports were placed along the mid-span of both panels. For the low-aspect ratio panel (Panel 1), an additional three pressure ports were placed off the mid-span, a quarter-span length away from the edge of the panel. The chordwise locations of these three ports coincided with the three centerline ports closest to the trailing edge, as shown in figure 3.5. The transducer was connected to the pressure ports by tubing approximately 1.15 m long.

Experiments were conducted for a range of Strouhal numbers at moderate Reynolds numbers, as given in table 3.2. For this range of  $St$  and  $Re_c$ , the velocity was varied from 0.03 m/s to 0.3 m/s and the pitching frequency was varied from 0.5 Hz to 2.67 Hz. Measurements were obtained for the low-aspect ratio panel (Panel 1) pitching at two trailing edge peak-to-peak amplitudes:  $A = 20$  mm and 40 mm. Measurements

Panel	A (mm)	St	$Re_c = Uc/\nu$
1	20	0.133 – 1.33	3500 – 35,000
	40	0.222 – 2.67	3500 – 43,200
2	20	0.167 – 1.33	3500 – 28,800

Table 3.2: Summary of pressure experiment parameters

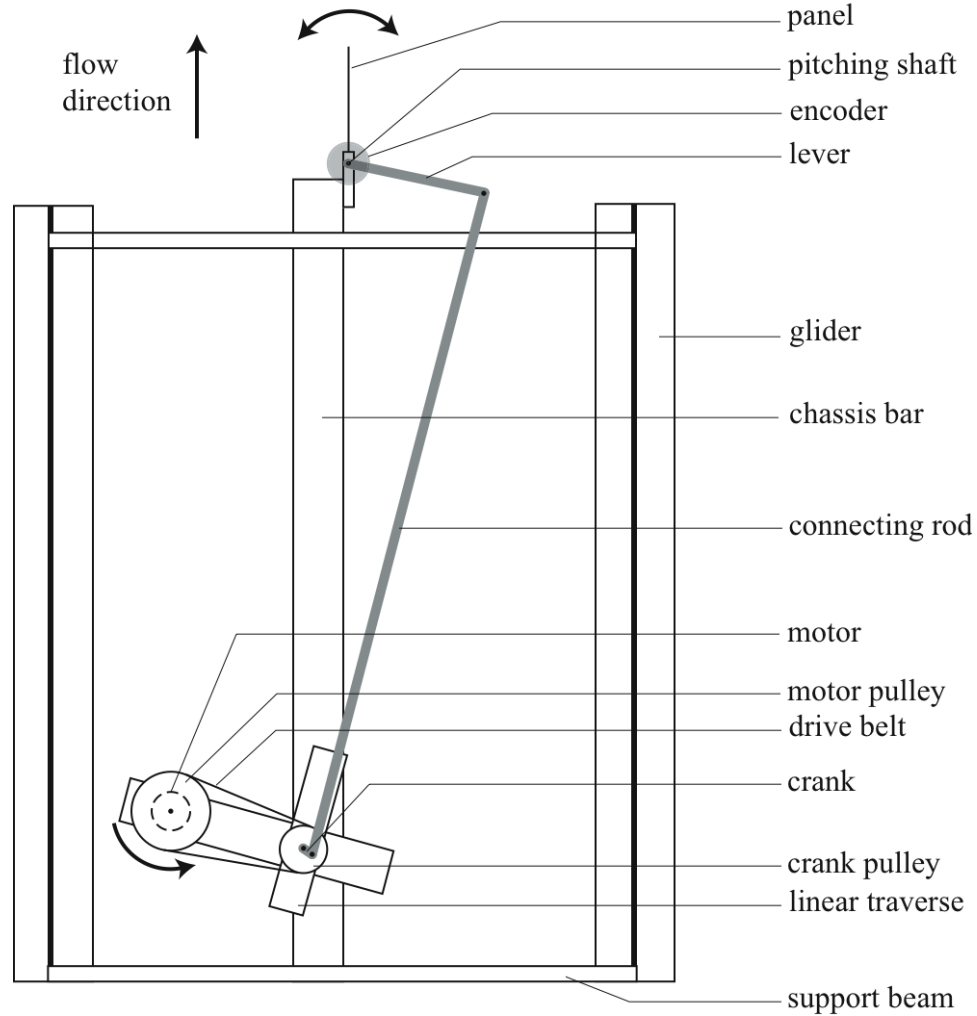


Figure 3.4: Four bar linkage mechanism used to actuate the pitching panel. From Buchholz (2006)

for the high-aspect ratio panel (Panel 2) were acquired for  $A = 20$  mm.

The time-resolved pressure was measured using one port location at a time until the entire distribution was obtained. The phase of the panel motion was monitored using the optical encoder mounted on the pitching shaft, and the pressure distributions were phase averaged to smooth the data.

The error bars presented in plots of pressure shown in chapter 4 represent the composite errors (added in quadrature) arising from the limited frequency response of the system and the effects of phase-averaging. To determine the frequency response of the system, a dynamic calibration of the transducer was performed using a linear solenoid to add a step input to sealed, water-filled tubing of the same length as used in the experiment. The transducer and pressure tubing were modeled as a combination of second order systems, and an unconstrained nonlinear minimization method (MATLAB function `fminsearch`, which uses a Nelder-Mead algorithm) was

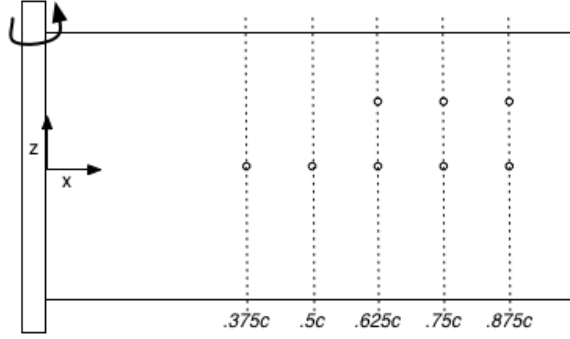


Figure 3.5: Pressure port placement on the low-aspect ratio ( $AR = S/C = 0.54$ ) panel (Panel 1).

used to fit the step response of the system model to the experimental step response. Figure 3.6 shows the step response of the transducer, and the step response of the optimized transfer function.

Once the optimized transfer function was obtained, the dynamic response of this model was used as an estimate of the dynamic response of the pressure transducer and tubing. Figure 3.7 shows the magnitude and phase of the transfer function plotted against frequency. It was found that the resonant frequency of the system was well above the pitching frequencies investigated. The error of the dynamic response was less than 5% for frequencies up to 3.9 Hz. At a pitching frequency of 2.67 Hz, the highest used for the present results, the error of the dynamic response was less than 2.3%. Also, at the frequencies used in the pressure experiments, a maximum phase lag of  $4^\circ$  was possible. The majority of the experiments were conducted at a frequency of 2 Hz, with a dynamic response error of 1.25% and a phase error of  $3^\circ$ . In reporting current results, the small amount of phase lag was ignored, and only the magnitude was taken into account in the calculation of the error bars. As to the uncertainties due to phase-averaging, the error at each point of the cycle was estimated to be two standard deviations based on the spread of points being averaged.

### 3.3.1 Flow visualization and DPIV: rigid trapezoidal panel

Subsequently, a series of experiments to visualize and measure the flow field was conducted on rigid panels of trapezoidal planform. These panels had swept edges that were set at an angle ( $\theta$ ) from the streamwise direction. A schematic of the trapezoidal panel dimensional parameters is shown in figure 3.8 and a summary of the panel dimensions used for the flow visualization and DPIV experiments is given in table 3.3.

### 3.3.2 Dye flow visualization

Two forms of dye flow visualization were used to investigate the large scale structures of the flow around the rigid trapezoidal panels. White light visualization was used to

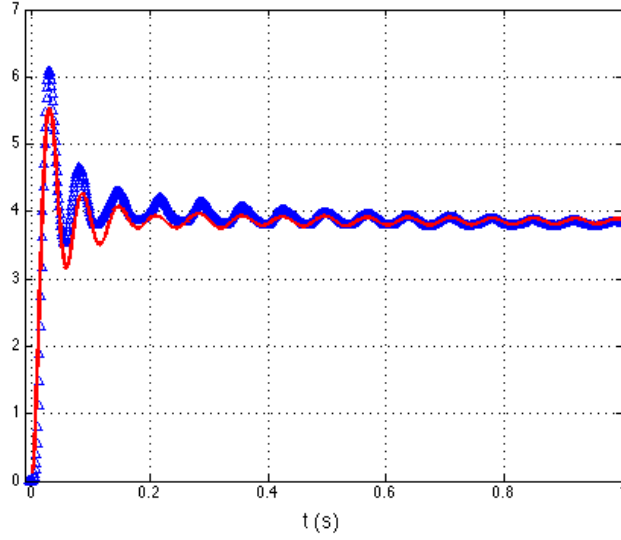


Figure 3.6: Step response of the pressure transducer and tubing.  $\triangle$ : experimental response; — optimized transfer function model.

Experiment	c	S	$\theta$
Flow Visualization	100 mm	168 mm	$30^\circ$
	90 mm	213 mm	$45^\circ$
DPIV	70 mm	175 mm	$45^\circ$

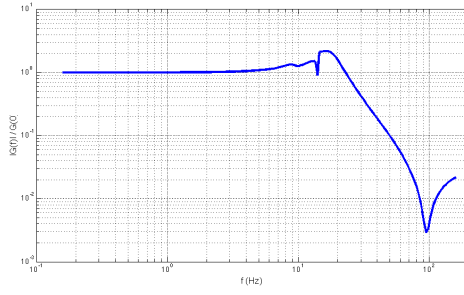
Table 3.3: Trapezoidal panel geometries

fully observe the three-dimensional structures, while planar laser-induced fluorescence (PLIF) was used to investigate the structures as they moved through two-dimensional planes.

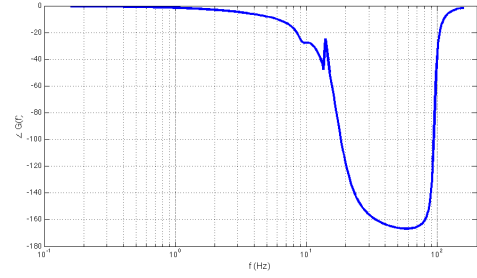
For the white light visualization images, fluorescein (Sigma-Aldrich 166308) and sulforhodamine B (Sigma-Aldrich 230162) dyes were injected into the hollow cavities of the airfoil fairing. The fluorescein (green) was injected from the cavity on the right of the fairing ( $y > 0$ ) and the sulforhodamine B (pink) was injected from the left-half cavity ( $y < 0$ ). Images were acquired using a commercial digital camcorder at flow speeds ranging from 0.015 m/s to 0.042 m/s with the panel pitching at 0.5 Hz with  $A = 20$  mm. This represented Strouhal and Reynolds number ranges of ( $.24 < St < .667$ ) and ( $1800 < Re_c < 5040$ ).

Dye flow visualization films of these panels were taken from two angles, shown in figure 3.9. Both a top view and an angled, upstream-facing side view were used to observe both the trailing edge wake and the structure generation and organization around the swept edges.

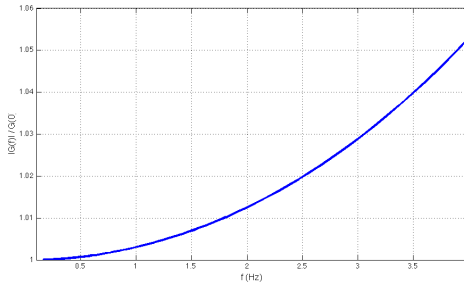
To obtain a more detailed image of the dye flow, PLIF images were taken in a plane normal to both the streamwise flow and the panel surface at four locations along the panel chord:  $25\%c$ ,  $50\%c$ ,  $75\%c$ , and just downstream of the trailing edge.



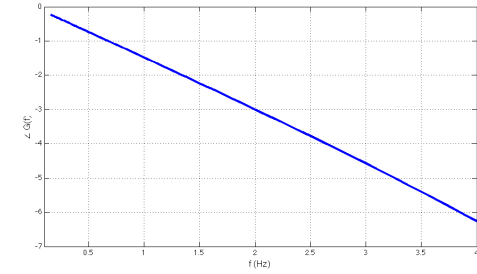
(a)



(b)



(c)



(c)

Figure 3.7: Bode plots of the model transfer function. (a) Bode magnitude plot with log-log axes, (b) Bode phase plot with log-log axes, (c) Bode magnitude plot with linear axes zoomed in on frequencies of interest in the current work, and (d) Bode phase plot with linear axes zoomed in on frequencies of interest.

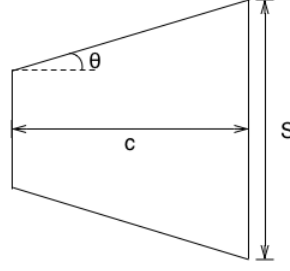


Figure 3.8: Diagram of trapezoid dimensions

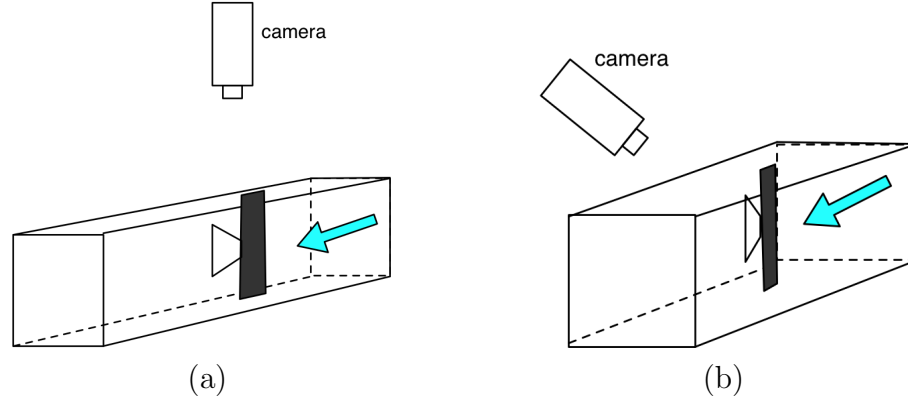


Figure 3.9: White light flow visualization orientations. Blue arrow indicates freestream flow direction. (a) top view, and (b) angles view.

For these experiments, only the fluorescein dye was injected into the left half ( $y > 0$ ) of the airfoil fairing. The dye was illuminated using a Spectra Physics Series 2000 argon-ion continuous-wave laser. Positioning of the laser beam and formation of the laser sheet was accomplished by use of a fiber optic cable, collimating optics and Powell lens (Oz Optics). The Powell lens is a line generator that transforms the collimated beam into an approximately uniform intensity 2 mm thick laser sheet. Images were acquired using a Redlake HG-LE CCD camera. Camera parameters, such as frame rate, exposure time, and number of frames acquired were controlled using the proprietary Motion Central software.

In order to image the plane normal to the streamwise direction, a mirror was inserted downstream of the test section at a  $45^\circ$  angle to the freestream flow. Rigid, closed-cell foam was formed into a trailing edge for the angled mirror in order to streamline the shape and reduce disturbances caused by the mirror. The camera was then positioned at the side wall of the water channel, as shown in figure 3.10. White light flow visualization was recorded both before and after the insertion of the mirror, and no appreciable difference in the structure of wake was observed.

### 3.3.3 Digital Particle Image Velocimetry

To obtain three-dimensional, quantitative velocity information throughout the fluid volume, two-dimensional digital particle image velocimetry (2D DPIV) was acquired

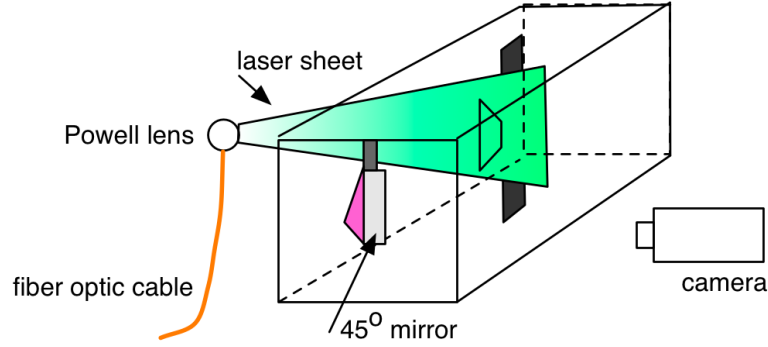


Figure 3.10: Schematic of PLIF orientation and setup

in 121 planes uniformly spaced 2 mm apart, which scanned through the three dimensional volume in the test section. These experiments were conducted with trailing edge amplitudes ( $A$ ) of 20 mm, and at 1 Hz pitching frequency.

The two-dimensional planes were oriented parallel to the streamwise flow and normal to the panel surface, as shown in figure 3.11. The particles used to seed the flow were  $13\mu\text{m}$  hollow silvered spheres made by Potters Industries Inc. (CONDUCT-O-FIL® SH400S33). They were illuminated using the Spectra Physics argon-ion laser, and the laser beam was again manipulated using a fiber optic cable, collimator, and Powell lens which emitted a 2 mm thick laser sheet. The lens was held by an optical clamp and mounted on a Velmex X-Slide traverse, which was controlled by a Velmex VXM stepper motor controller. The traverse was mounted on a table that sat at one of the spanwise walls of the test section.

In addition, the Redlake HG-LE camera that was used to acquire the images was mounted vertically above the water channel on a Velmex Unislide traverse, as shown in figure 3.11. This traverse was controlled by the same Velmex VXM stepper motor controller. The LabVIEW driver software for both the VXM and the camera software (Motion Studio) were used such that a single LabVIEW virtual instrument controlled both the positioning of the laser sheet and camera and the image acquisition at each plane.

Each two dimensional velocity field was obtained at twenty-five discrete phases in the pitching cycle. The images were acquired using the Redlake HG-LE camera, which was externally triggered by a Stanford Research Systems four channel digital delay/pulse generator (Model DG535). The Stanford box controlled the timing of the experiment. The encoder mounted on the pitching shaft sent a 5V pulse each time the pitching panel passed through zero angle of attack (twice per pitching cycle). Each pair of pulses was converted to the rising and falling edges of a square wave signal, which served as the input to the Stanford box. At a user-defined delay from each rising edge of the square wave, the Stanford box sent a finite-width pulse to the camera trigger input. Using the Motion Studio software the exposure time, frame rate, and number of images to acquire were set.

The camera was operated in the Burst Record on Command (BROC) mode, in which it acquires a set number of images at each external trigger. In each plane, 20



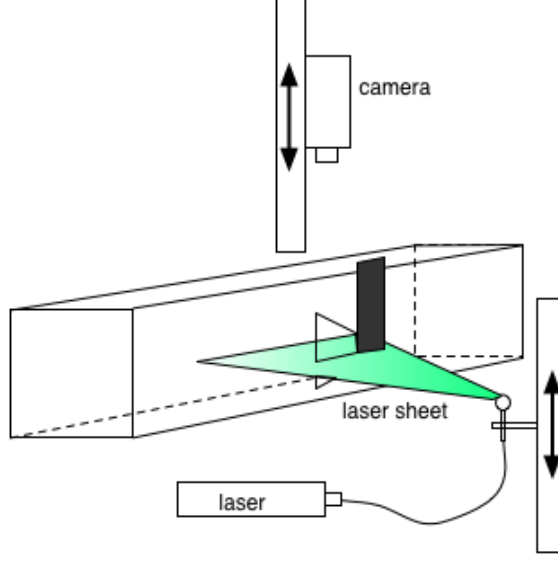


Figure 3.11: DPIV acquisition setup.

image pairs ( $\Delta t$  between each image: 0.02s) were acquired at each of the 25 phases in the panel motion, and the resulting 20 velocity fields at each phase were phase-averaged. This yielded 25 phase-averaged velocity data sets per pitching cycle with a time resolution of 0.04 s.

After the images were acquired and downloaded from the camera, both Velmex traverses were moved 2 mm in tandem and acquisition began in the next plane. Data were taken at 121 planes through the test section volume, uniformly distributed over a spanwise depth of 0.24 m. The extent of each DPIV data set spans 0.10 m ( $1.4c$ ) in the streamwise direction and 0.15 m ( $2.1c$ ) in the transverse direction. Full data sets were taken at two overlapping streamwise locations for a total streamwise data length of 0.19 m ( $2.7c$ ), and the upstream edge of the data is located at  $x = 0.45c$ . To accomplish this, the pitching apparatus was moved further upstream while the camera and laser remained in place. All 121 planes at one streamwise location are shown in figure 3.12(a), and the resulting overlapping grids from both the upstream and downstream locations are shown in figure 3.12(b).

The Redlake camera was operated at its full resolution of  $1128 \times 752$  pixels. With the camera and laser sheet aligned to image the deepest spanwise plane, there was an image resolution of 0.136 mm/pixel. However, due to the change in refractive index as light travels from water to air, this calibration changed with depth. The refractive index of water ( $n_w = 1.33$ ) is greater than that of air ( $n_a = 1.00029$ ) and according to Snell's law,

$$n_w \sin \beta_w = n_a \sin \beta_a, \quad (3.2)$$

Therefore, as light travels from water and across the surface into air, the refractive angle ( $\beta$ ) will increase. As shown in figure 3.13, the increase in refractive index means that for those cases in which the distance between the camera and the image plane is in air is longer than in water, i.e. when the camera is higher above the water surface,

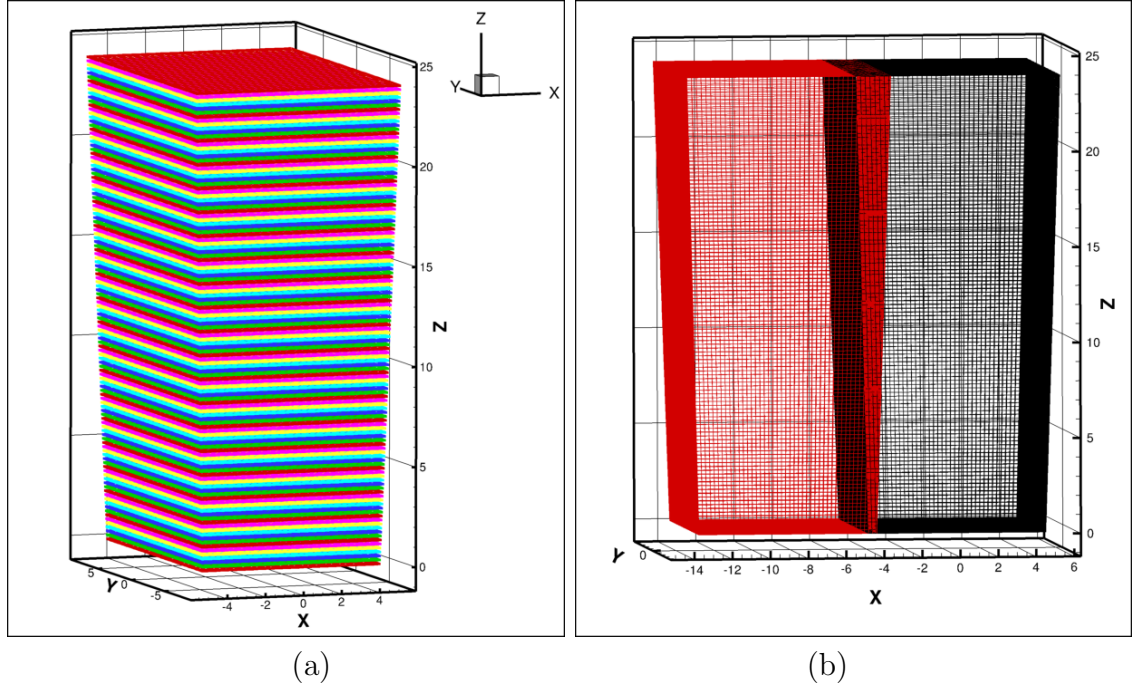


Figure 3.12: DPIV 2D planes and resulting 3D grids

both the height and the width of the image plane are larger. This change in pixel size calibration is linear with depth, and hence the full velocity grids in figure 3.12(b) are trapezoidal prisms. In order to facilitate post-processing, the velocities at these grid points were interpolated to a regular rectangular grid with streamwise and spanwise widths equal to those of the smallest plane at the bottom of the data volume. This was done using a two-dimensional bi-cubic spline interpolation.

The DPIV analysis in each plane was performed using software developed by Jiménez (2002). A multi-pass cross-correlation algorithm was used, with first  $64 \times 64$  and subsequently  $32 \times 32$  pixel window sizes with 50% overlap. In each of these windows, a two-dimensional Gaussian function was fitted to the correlation peak using five points in the  $x$ - and  $y$ -directions, providing sub-pixel accuracy in the calculation of pixel displacement. In general, mean-bias and RMS errors from cross-correlation DPIV analyses are on the order of 0.1 pixels (Huang *et al.*, 1997). Mean-bias error can occur when the number of particles that remain in the interrogation window is small, which can be caused by insufficient particle seeding or a large ratio of particle displacement to pixel window size. Sources of RMS error can include improper particle seeding, strong velocity gradients, and three-dimensional flow causing particles to leave the image plane. Other important factors include non-uniformity in the laser sheet or particle light reflection and electronic noise in the camera or cables.

In the current work, the three-dimensional flow, particularly the out-of-plane velocities near the edges of the panel, are assumed to be the largest source of error in the DPIV experiments. If the out-of-plane velocity (spanwise direction) is assumed to be on the order of the transverse velocity, it will have a maximum value of approximately 50 mm/s. A particle with this spanwise velocity will travel 1 mm in 0.02 s, the time

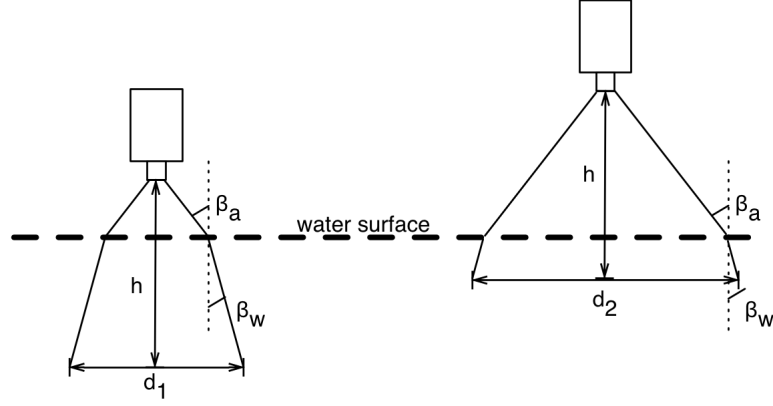


Figure 3.13: Illustration of optical effects. The distance between the camera and the image plane ( $h$ ) is the same in both cases. The width of the image plane when the camera is closer to the water surface is smaller than the width of the image plane when the camera is farther away from the water surface ( $d_2 > d_1$ .)

between each image. Since the laser plane is approximately 2 mm thick, this means that in the regions of the flow where 3D effects are greatest, half the particles in the laser sheet will leave the plane between the first and second images of a pair. The effect on the in-plane velocity calculations will be a decreased magnitude correlation peak which increases the RMS error. However, phase-averaging the resulting velocity fields mitigates these effects.

Each velocity data set consisted of  $88 \times 67 \times 121 = 713416$  points, with 5896 points in each spanwise plane. The velocity data grid spacing was 2.2 mm in the ( $x$ - $y$ ) planes, and as stated previously, 2 mm in the spanwise ( $z$ ) direction.

# Chapter 4

## Three-dimensional effects

The mean thrust measurements on pitching rigid rectangular panels by Buchholz & Smits (2008) revealed that the thrust and efficiency depend on the flow Strouhal number, the aspect ratio of panel ( $S/c$ ), and also weakly on the ratio of pitching amplitude to panel chord ( $A/c$ ). In this chapter, a series of pressure measurements were obtained to pursue a better understanding of the force distribution on the panel surface and to give insight to this dependency. The time-resolved pressure results not only illustrate the necessity of including three-dimensional considerations when characterizing the panel performance, but also add insight into the panel wake organization and evolution which will be discussed in chapter 5.

Most of the results and analysis presented in this chapter was published in Green & Smits (2008).

### 4.1 Pressure measurements

#### 4.1.1 Temporal analysis

As a reference for the vocabulary used to describe the time-resolved pressure measurements relative to the panel motion, a schematic of the panel in advancing motion (toward the measurement surface) and its associated motion curve is shown in figure 3.5.

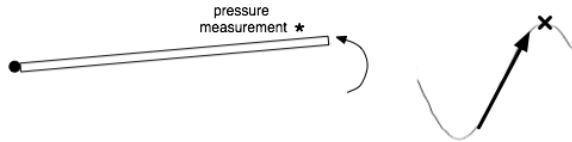


Figure 4.1: Schematic of the panel advancing, and its associated trailing edge amplitude curve.

As published in Buchholz (2006) and shown in figures 4.2 and 4.3, optimal efficiency of the rigid pitching panels varies with Strouhal number and aspect ratio. For

Panel	chord (c)	span (S)	(AR=S/C)
1	120 mm	6 mm	0.5
2	120 mm	270 mm	2.25

Table 4.1: Rectangular panel dimensions

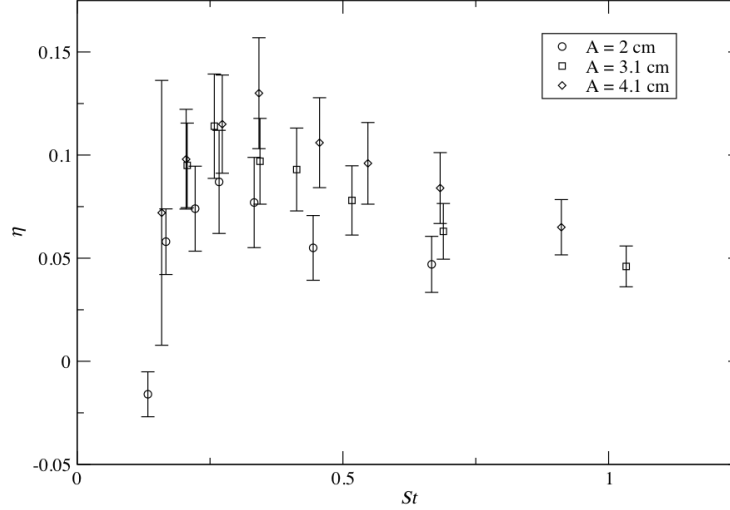


Figure 4.2: Propulsive efficiency variation with Strouhal number for Panel 1. From Buchholz (2006)

the low-aspect-ratio panel (Panel 1, see table 4.1),  $St = 0.27$  corresponded closely to that of the most efficient level of thrust production. The phase-averaged pressure trace obtained on the centerline at  $0.875c$  on Panel 1 at this Strouhal number is shown in figure 4.4.

The pressure reaches its minimum at a phase of  $\phi \approx 100^\circ$ , shortly after the panel stops advancing ( $\phi = 90^\circ$ ). Similarly, the pressure maximum occurs at a phase of  $\phi \approx 300^\circ$ , approximately one half cycle later. The pressure extrema are approximately associated with the maximum and minimum acceleration of the panel. The pressure minimum coincides with the maximum panel acceleration as it begins to retreat, and while the pressure maximum is more delayed, it still coincides closely with the maximum panel acceleration in advancing motion. Among the Strouhal numbers investigated in this study, there was no appreciable change in phase of the pressure extrema.

As the panel retreats, the pressure rises quickly, then levels off for a short period before continuing to increase. This quick rise is attributed to spanwise flow from the edges facilitating the relief of low pressure along the midspan. As will be explained in section 4.1.2, a strong favorable pressure gradient acts toward the midspan during this time ( $\phi \approx 135^\circ$ ), which causes the subsequent loss of low pressure. Similar pressure time-traces were observed on the wings of pigeons during flight by Usherwood *et al.*

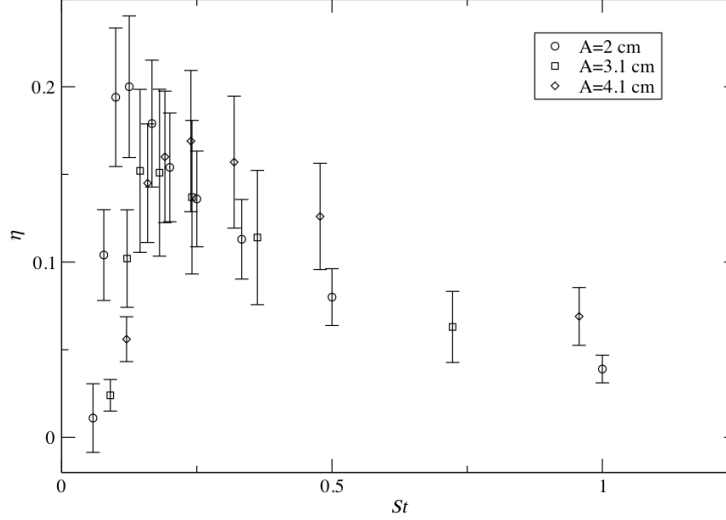


Figure 4.3: Propulsive efficiency variation with Strouhal number for Panel 2. From Buchholz (2006)

(2005), who referred to a “consistent deviation,” and associated it with the “clap” of the wing, a “period of high acceleration and relatively low differential pressure.”

Figure 4.5(a) shows the pressure variation at the same location and amplitude of motion on Panel 2 (see table 4.1) at  $St=0.33$  (the Strouhal number corresponding to the highest efficiency for these conditions according to Buchholz & Smits (2008)). For this higher-aspect-ratio panel, the pressure signal displays a larger trough that begins as the panel starts to retreat and ends as the panel begins to decelerate as it approaches its opposite extremum. The period of low pressure is more extended because the larger aspect ratio inhibits inflow from the spanwise edges which would tend to equalize the pressures.

Figure 4.5(b) shows the pressure variation at the same location on Panel 1, with  $A = 40$  mm and  $St = 0.33$  (again, this Strouhal number corresponds to the most efficient motion for this amplitude according to Buchholz & Smits (2008)). There is a loss of low pressure as the panel retreats, as before, but an additional period of pressure loss appears as the panel motion reaches its maximum pitch angle ( $\phi = 90^\circ$ ) and begins to retreat. The increased pitching amplitude increases three-dimensional effects, which lead to additional low pressure losses.

#### 4.1.2 Spatial analysis

The streamwise distribution of pressure on the advancing surface of Panel 1 pitching at  $St = 0.27$  is shown in figure 4.6 for four phases of the panel motion, with the corresponding panel position and port locations shown superimposed. As the panel advances through zero angle of attack ( $\phi = 0^\circ$ ), an adverse pressure gradient is present along the midspan of the panel. When the panel reaches the extremum ( $\phi = 90^\circ$ ), the

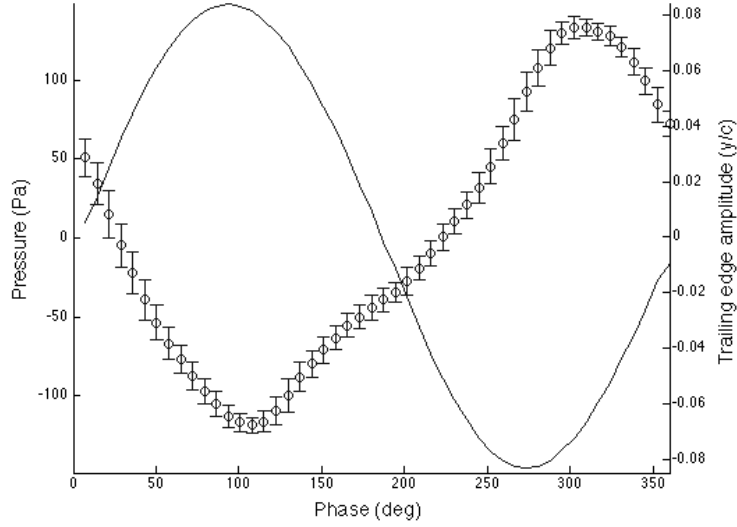


Figure 4.4: Unsteady pressure variation at  $0.875c$  on Panel 1,  $A = 20$  mm,  $St = 0.27$ . —, trailing edge amplitude;  $\circ$ , pressure deviation from the time-averaged mean. Panel is “retreating” for  $90^\circ < \phi < 270^\circ$ .

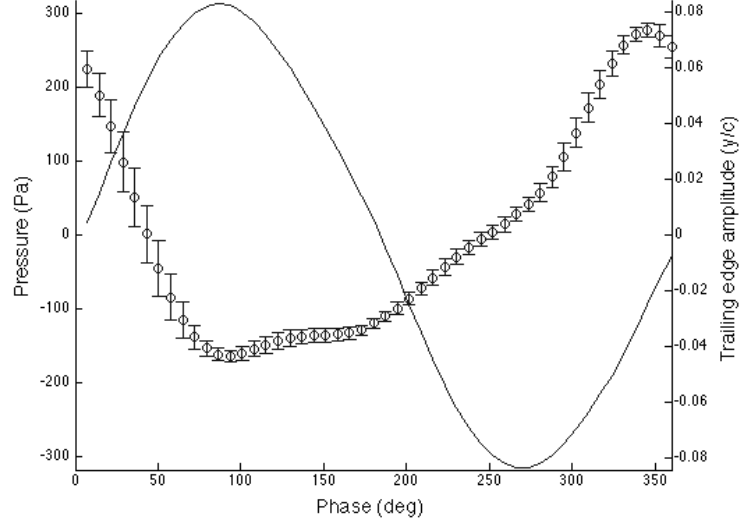
streamwise gradient on the top surface has changed, with strong favorable gradients near  $x = 0.5c$  and  $0.75c$ , where  $x$  is measured from the leading edge of the panel. A slight favorable pressure gradient persists as the panel retreats through the zero angle of attack ( $\phi = 180^\circ$ ). The pressure along the midspan then increases until the panel reaches the opposite extremum ( $\phi = 270^\circ$ ) with relatively strong adverse gradients again present near  $x = 0.5c$  and  $0.75c$ . Similar streamwise distributions were observed for the higher aspect ratio and higher amplitude configurations.

The spanwise distribution of pressure on Panel 1 was measured at  $0.875c$ . A strong favorable gradient acting toward the midspan is observed when the panel is retreating ( $\phi \approx 135^\circ$ ): the pressure magnitude at  $z = 0.25S$  is 47% higher than that at the midspan ( $z = 0$ ). As a result, flow is induced from the advancing surface over the spanwise edges toward the midspan of the retreating surface. It is this spanwise pressure gradient that relieves the period of low pressure on Panel 1. Half a cycle later, a favorable gradient is directed toward the spanwise edges of the panel on the now advancing surface, and the pressure at  $z = 0.25S$  is only 60% of its value at the midspan. At this phase, fluid tends to flow away from the midspan upstream of the trailing edge.

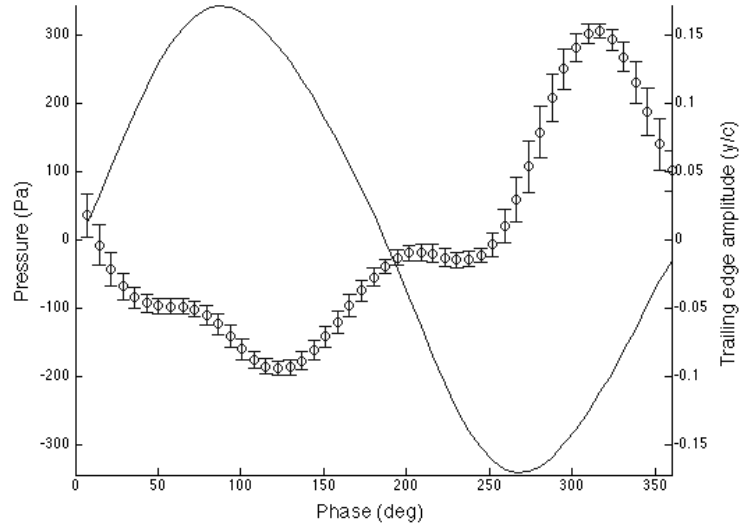
### 4.1.3 Scaling

The coefficient of pressure  $C_p$  is shown in figure 4.7(a) at  $x = 0.875c$ , where

$$C_p = \frac{\Delta p}{\frac{1}{2}\rho U^2}, \quad (4.1)$$



(a)



(b)

Figure 4.5: Unsteady pressure variation at  $0.875c$  on (a) Panel 2,  $A = 20$  mm,  $St = 0.33$ , and (b) Panel 1,  $A = 40$  mm,  $St = 0.33$ . —, trailing edge amplitude;  $\circ$ , pressure deviation from the time-averaged mean.



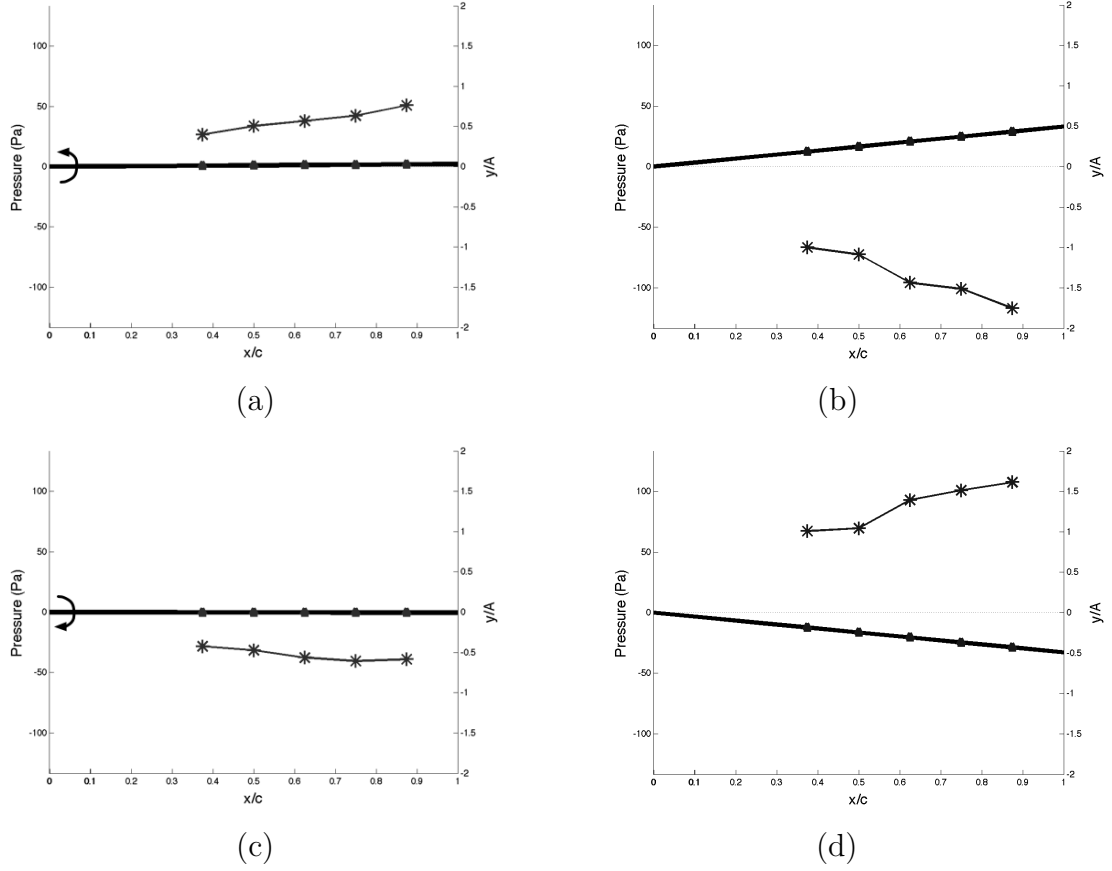


Figure 4.6: Streamwise distribution of pressure along the midspan for Panel 1 pitching at  $St = 0.27$  and  $A = 20$  mm. (a)  $\phi = 0^\circ$ ; (b)  $\phi = 90^\circ$ ; (c)  $\phi = 180^\circ$ ; (d)  $\phi = 270^\circ$ . In each figure, the upper part shows the pressure distribution and the lower part shows the position of the panel. \*, pressure deviation from the time-averaged mean;  $\Delta$ , location of pressure port.

and  $\Delta p$  is the peak-to-peak amplitude of the pressure variation. The pressure coefficient follows distinct trends for each of the three experimental configurations studied here. Buchholz & Smits (2008) showed that the thrust produced by pitching rigid rectangular panels depends on the Strouhal number and the aspect ratio  $S/c$ . In addition, they showed a weak inverse relationship with the amplitude of oscillation  $A$ . Here we find that the peak-to-peak amplitude of the oscillating pressure signal varies with aspect ratio and the ratio of pitching amplitude to chord,  $A/c$ . This suggests a new scaling, such that

$$C_p^* = C_p \left( 1 + f \left( \frac{S}{c}, \frac{A}{c} \right) \right). \quad (4.2)$$

A simple form of  $f$  taken as  $f(x, y) \propto y/x$  leads to

$$C_p^* = C_p \left( 1 + C_1 \frac{A}{S} \right). \quad (4.3)$$

The form of the scaling was determined empirically, but was also defined to ensure  $C_p^* = C_p$  as the aspect ratio goes to infinity, as required. The scaling suggested here is similar to the scaling used to account for the effects of aspect ratio on the lift coefficient in finite wing theory. As shown in figure 4.7(b), the new scaling neatly collapses the pressure coefficient data ( $C_1 = 7$ ).

## 4.2 Application of scaling

The pressure measurements along the midspan showed that the maximum and minimum pressures were associated with the extrema of the panel acceleration. When the panel motion reached an extremum, a positive streamwise pressure gradient existed along the surface that is beginning to advance, which is in contrast to steady flow at the same angle of incidence where a negative pressure gradient would be expected.

Buchholz (2006) used flow visualization and vorticity contours to infer the direction and relative magnitude of the pressure gradient on the advancing surface of the panel, as shown in figure 4.8. The measurements presented here confirm that a strong favorable streamwise gradient occurs over the region close to the trailing edge ( $0.8 < x/c < 1$ ), but this pressure gradient began to exert its influence as the panel stopped and began to advance a quarter cycle earlier.

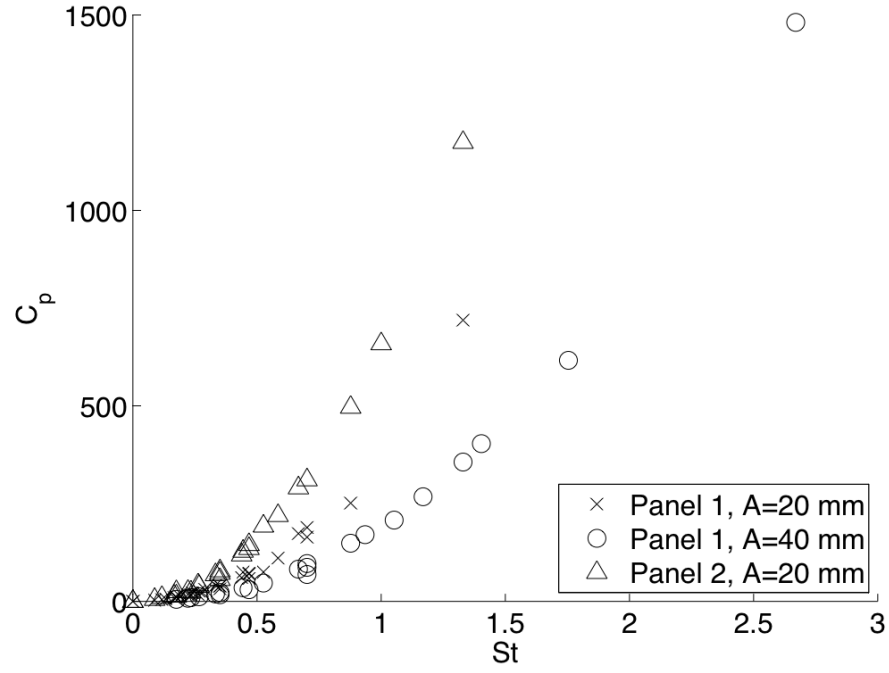
For the experiments reported here, the pressure signal deviated from a smooth sinusoid during periods of low pressure, suggesting the presence of strong three-dimensional effects. Flow visualizations by Buchholz & Smits (2008), shown in figure 4.9, showed spanwise ejections of fluid, and this was seen to occur at the extrema of motion, when a strong favorable pressure gradient acts towards the spanwise edges. A strong spanwise adverse gradient was shown to exist a half cycle later, indicating flow toward the midspan of the panel. This low-aspect-ratio effect also explains the loss of low pressure along the midspan ports.

It was shown that a new pressure coefficient  $C_p^*$  that included the effects of aspect ratio and oscillation amplitude successfully collapsed the pressure results onto a single curve. Since the pressure distribution is directly related to the thrust  $T$  produced by the panel, we suggest a similar scaling for the thrust, and define  $C_T^*$ ,

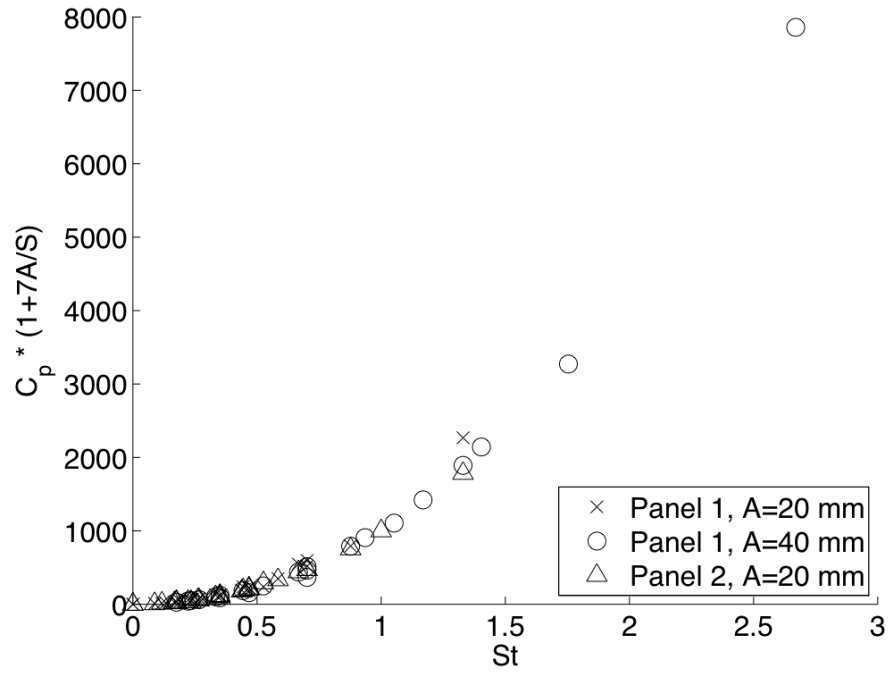
$$C_T^* = C_T \left( 1 + C_2 \frac{A}{S} \right), \quad (4.4)$$

where  $C_T$  is the common coefficient of thrust,

$$C_T = \frac{T}{\frac{1}{2}\rho U^2 S c}, \quad (4.5)$$



(a)



(b)

Figure 4.7: Peak-to-peak pressure amplitude at  $x = 0.875c$ : (a)  $C_p$ ; (b)  $C_p^*$  with  $C_1 = 7$ .

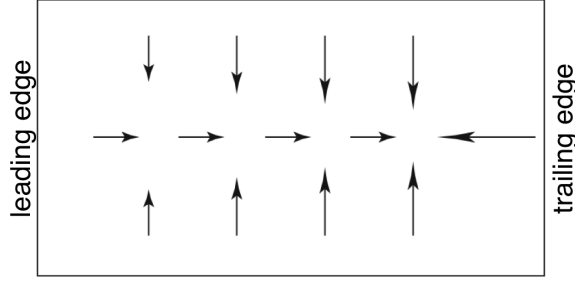


Figure 4.8: Pressure gradient on advancing surface of panel inferred by Buchholz (2006). Magnitude indicated by arrow length. Flow is from left to right.

Here,  $T$  is the resultant force in the streamwise direction. Mean thrust measurements on these panels of different aspect ratio and pitching amplitude were reported by Buchholz & Smits (2008). The results shown in figures 4.10(a–b) demonstrate that this scaling indeed collapses the thrust results within the uncertainty limits. As it turns out, the constant used to scale both the pressure and thrust data is the same, with  $C_1 = C_2 = 7$ . Additional experiments with panels of varying chord are needed to generalize the proposed scaling.

Additionally, in the course of his thesis work, Buchholz (2008-2009) calculated the spanwise circulation in the wake of the pitching rectangular panels. The circulation is defined as the line integral of the fluid velocity around a closed curve, and by Stokes Theorem may also be calculated as the area integral of vorticity,

$$\Gamma = \oint_C \mathbf{V} \cdot d\mathbf{s} = \iint \omega dA. \quad (4.6)$$

Buchholz (2008-2009) calculated the circulation of the spanwise vortex shed after one half-cycle of the pitching panel. This was done for two low-aspect ratio panels, pitching with two different trailing edge amplitudes. Buchholz showed that if these data are non-dimensionalized using the freestream velocity and pitching frequency as the relevant length- and time-scales, such that,

$$C_{\Gamma_{Uf}} = \frac{\Gamma}{U^2(1/f)}, \quad (4.7)$$

the circulation varies with  $St$  as shown in figure 4.11(a). Buchholz then proposed that if the scaling proposed in the current work is applied to the circulation coefficient defined in this way, such that

$$C_{\Gamma_{Uf}}^* = C_{\Gamma_{Uf}} \left( 1 + C_3 \frac{A}{S} \right), \quad (4.8)$$

the results seem to collapse, as shown in figure 4.11(b), with  $C_3 = C_1 = 7$ . Moreover, if instead the circulation is non-dimensionalized using the pitching amplitude and

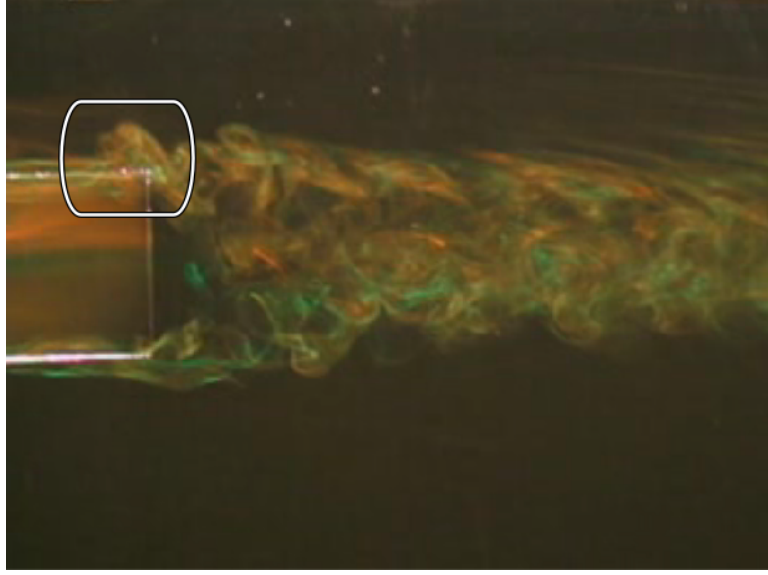


Figure 4.9: Flow visualization of panel 1 pitching at a  $St=0.6$ , which the spanwise ejection of fluid highlighted with a white box. Image from Buchholz (2006).

frequency as the relevant length- and time-scales, such that,

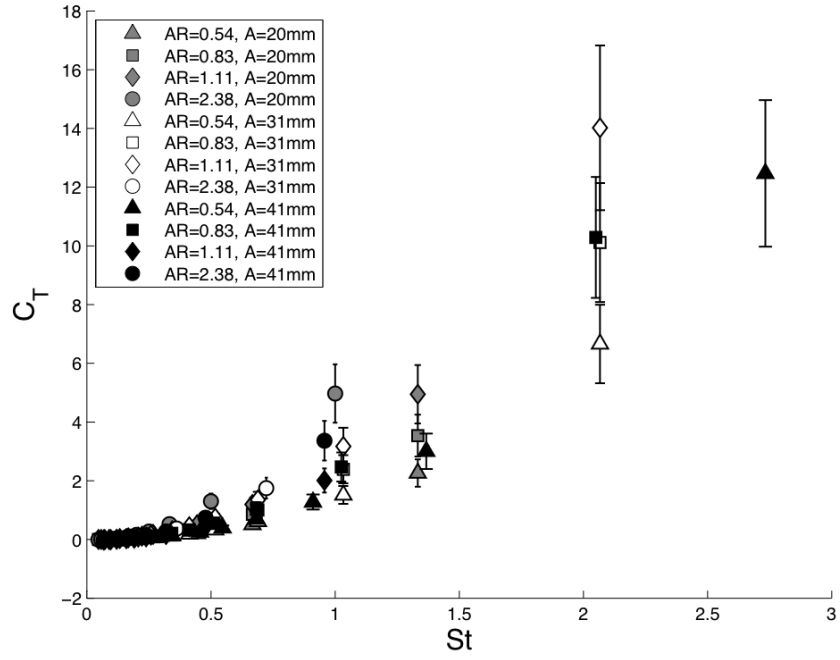
$$C_{\Gamma_{Af}} = \frac{\Gamma}{A^2 f}, \quad (4.9)$$

the variation of circulation with  $St$  is as shown in figure 4.12(a). Using this non-dimensionalization, the dependence of the circulation on the Stouhal number is significantly weaker. If the scaling proposed in the current work is again applied to the circulation coefficient as suggested by Buchholz, such that

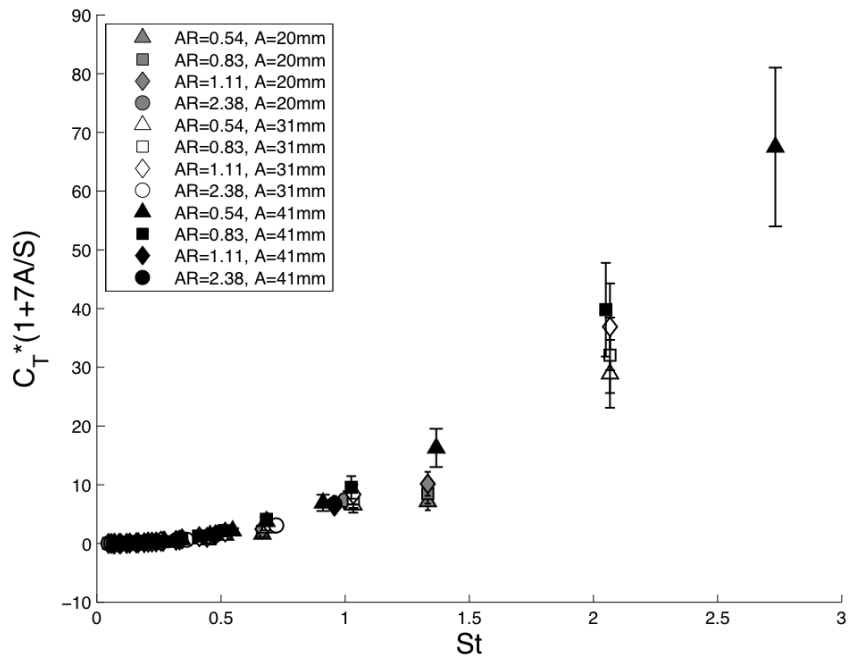
$$C_{\Gamma_{Af}}^* = C_{\Gamma_{Af}} \left( 1 + C_4 \frac{A}{S} \right), \quad (4.10)$$

the results appear to collapse even better, as shown in figure 4.12(b), with  $C_4 = C_1 = 7$ .

By continuing to consider the effects of varying aspect ratio and pitching amplitude, a new interpretation can also be given to the Digital Particle Image Velocimetry (DPIV) results of Buchholz (2006). Figure 4.13 shows isocontours of spanwise vorticity in the wakes of Panels 1 and 2 pitching with the same trailing edge amplitude. In figure 4.13(a), Panel 2 is pitching at  $St = 0.36$  and the wake has a 2S structure, with two single vortices being shed each flapping cycle. Panel 1, pitching at  $St = 0.26$ , is shown in figure 4.13(b), and exhibits a 2P structure, with two pairs of vortices shed each cycle. Previous work has shown that the transition from a 2S structure to a 2P structure is associated with an increase in Strouhal number, but in this case the transition is observed to occur at lower  $St$ , and may be a consequence of the increase

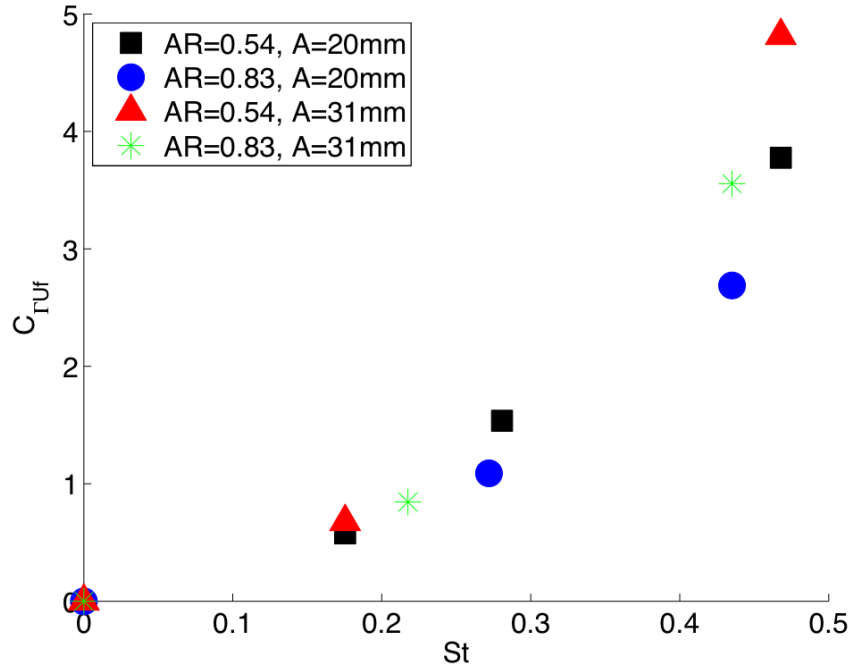


(a)

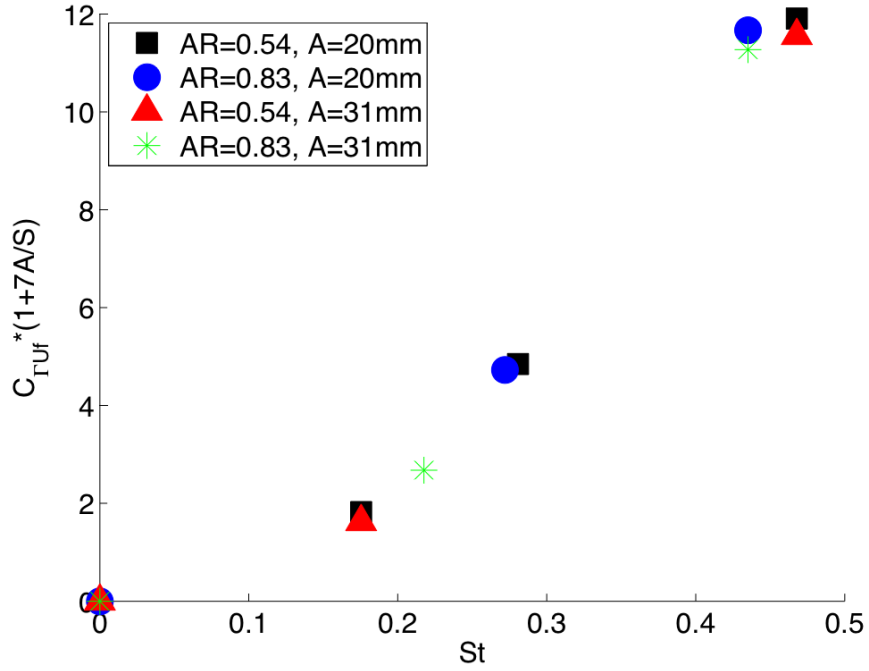


(b)

Figure 4.10: Thrust coefficient measured by Buchholz & Smits (2008) . (a)  $C_T$ ; (b)  $C_T^*$  with  $C_2 = 7$ .

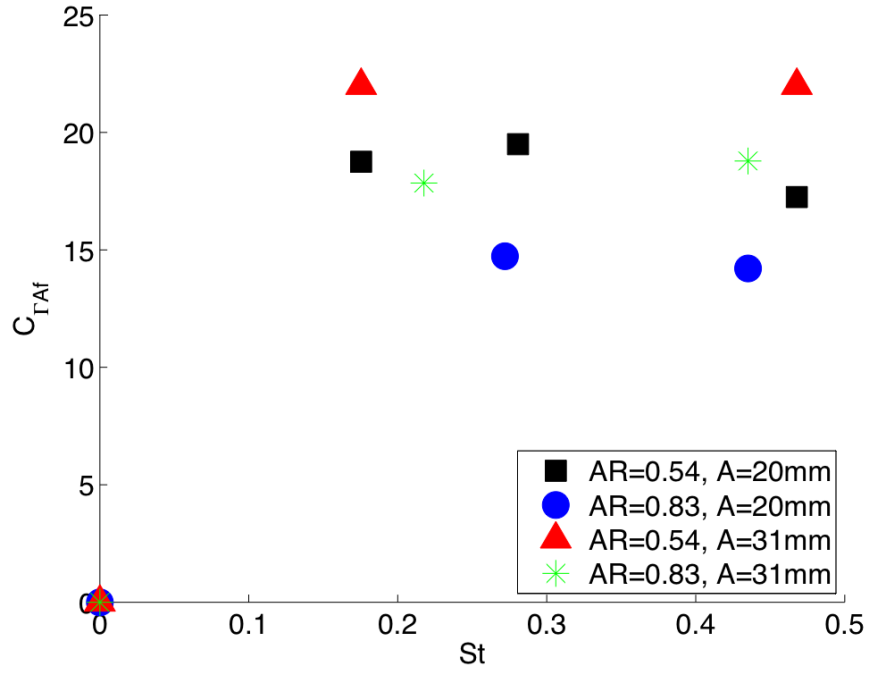


(a)

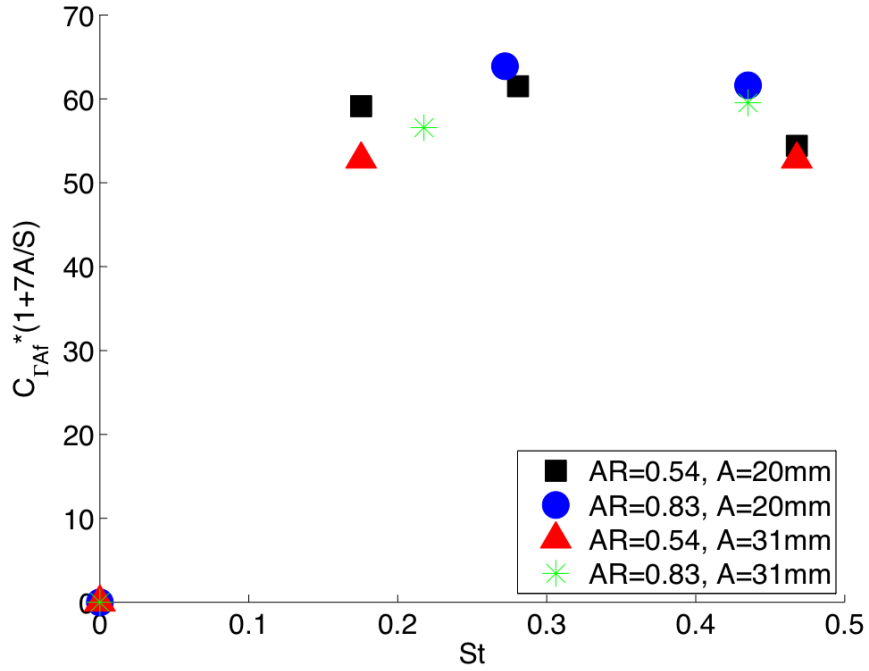


(b)

Figure 4.11: Circulation coefficient calculated by Buchholz & Smits (2008) non-dimensionalized using panel pitching frequency and freestream velocity ( $C_{\Gamma_{Uf}}$ ); (b) Circulation coefficient with additional scaling ( $C_{\Gamma_{Uf}}^*$  with  $C_3 = 7$ ). From Buchholz (2008-2009).



(a)



(b)

Figure 4.12: Circulation coefficient calculated by Buchholz & Smits (2008) non-dimensionalized using panel pitching amplitude and frequency ( $C_{\Gamma Af}$ ); (b) Circulation coefficient with additional scaling ( $C_{\Gamma Af}^*$  with  $C_3 = 7$ ). From Buchholz (2008-2009).



of three-dimensionality (decreased aspect ratio).

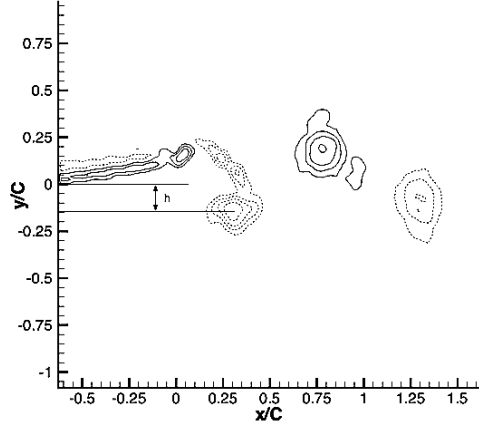
The phase averaged pressure traces at  $0.875c$  shown in figure 4.14 indicate that as the panel retreats from the measurement surface, the pressure on Panel 2 stays at or near its minimum value for a larger fraction of the pitching period than in the low aspect ratio or high amplitude cases. The vorticity contours in figure 4.13(b) show that downstream of the trailing edge of Panel 1, pitching with an amplitude of 31mm, the spanwise structure is at a lateral distance  $y/c \approx -0.30$ . However, downstream of the trailing edge of Panel 2, pitching with the same amplitude, at the same phase of motion it is closer to the edge at  $y/c \approx -0.15$ . The prolonged low pressure on Panel 2 may cause the spanwise structure at the trailing edge to keep rolling up as the trailing edge moves across the wake, and hence transverse wake expansion is inhibited. Hultmark *et al.* (2007), Tytell & Lauder (2004), and Buchholz *et al.* (2003) observed a 2P configuration in the wakes of a robotic lamprey, american eel, and flexible flapping membrane, respectively, and they described the formation of the 2P wake structure as a consequence of the segmentation of the shear layer being shed from the trailing edge. If, as in the higher aspect ratio case presented here, the trailing edge structure is pulled along with the panel surface, it will not spread into a shear layer and be susceptible to segmentation and transition into a 2P structure.

Similarly, an increase in pitching amplitude will enable the elongation and segmentation of the spanwise structure as it is shed from the trailing edge sweeping across the wake. Figure 4.13(c) shows the wake of Panel 1 pitching at  $A = 20$  mm and  $St = 0.27$ . At  $\phi = 90^\circ$ , the structure has travelled a shorter lateral distance than that seen for the same panel with a larger pitching amplitude ( $A = 31$  mm), and is located at  $y/c \approx -0.10$ , compared to  $y/c \approx -0.30$  for the higher amplitude case. The pressure traces shown in figure 4.14 indicate that when pitching with a higher amplitude, the panel experiences additional losses of low pressure. This loss of low pressure decreases the suction force, allowing the trailing edge structure to spread across the wake and break into two segments, forming the pairs that comprise the 2P structure.

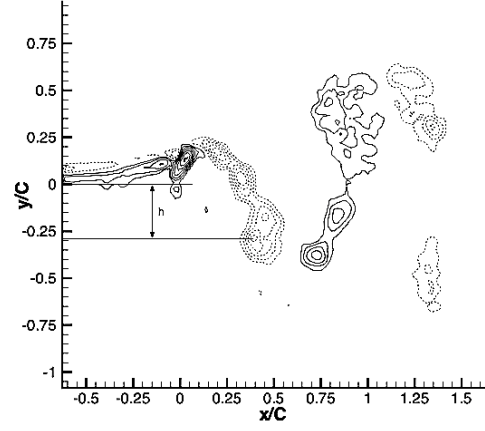
### 4.2.1 Summary

Previous work showed that an increase in unsteadiness (an increase in Strouhal number), affects the propulsive performance of pitching panels and causes the wake to transition from a 2S structure to the more complicated 2P structure. Here, we propose a new scaling law for pressure and thrust results that includes a consideration of three-dimensional effects. The new scaling nicely collapses both current pressure results and previously published mean thrust measurements and circulation calculations.

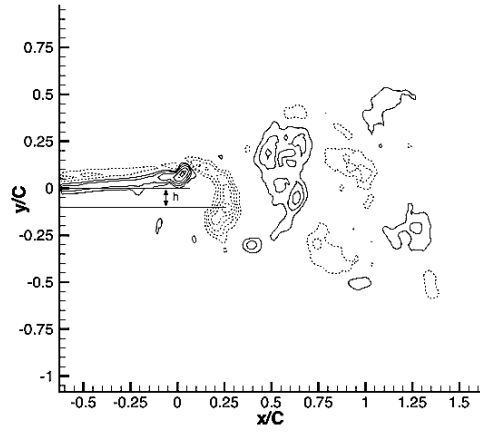
Improved understanding of the influence of three-dimensionality also led to a new interpretation of the wake transition. Three-dimensionality is increased when the aspect ratio of the panel is decreased or when the pitching amplitude is increased, and it is believed to be related to the time the shed vorticity dwells near the trailing edge. It is also believed that this interaction between the shed vorticity and the panel trailing edge affects the organization of the vortex wake, and can inhibit or facilitate



(a)



(b)



(c)

Figure 4.13: PIV taken by Buchholz & Smits (2008) at the midspan of a panel at  $\phi = 90^\circ$ : (a) Panel 2,  $A = 31$  mm,  $St = 0.36$ , (b) Panel 1,  $A = 31$  mm,  $St = 0.26$ , and (c) Panel 1,  $A = 20$  mm and  $St = 0.27$ . Contour levels are  $\pm n^2 s^{-1}$ ,  $n = 2, 3, 4, \dots, 10$ .

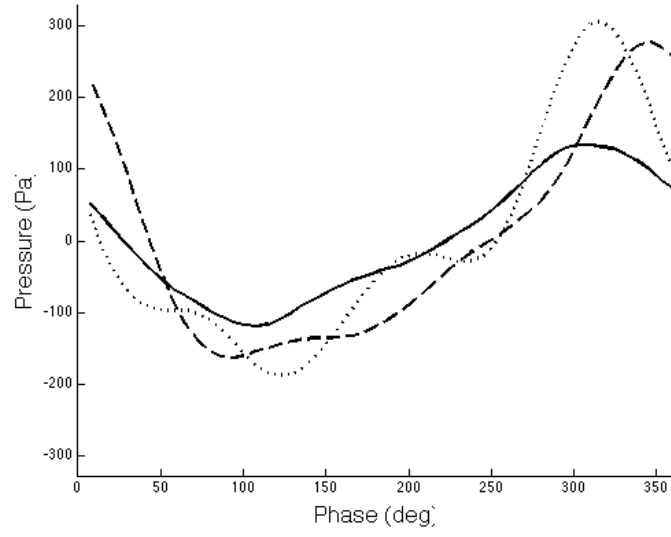


Figure 4.14: Unsteady pressure variation on the centerline at  $0.875c$  with  $A = 20$  mm and  $St = 0.27$ : (—) Panel 1,  $A = 20$  mm; ( $\cdots$ ) Panel 1,  $A = 40$  mm; and (- - -) Panel 2,  $A = 20$  mm.

wake transition. In this way, the qualitative wake characteristics are also consistent with the new scaling.

# Chapter 5

## Three-dimensional panel wake

In this chapter, we continue the study of caudal fin-inspired unsteady propulsors with a series of experiments conducted on pitching rigid trapezoidal panels. This work builds on the results of chapter 4 but adds complexity to the problem by introducing another parameter: edge sweep angle ( $\theta$ ). Flow visualizations were acquired for two trapezoidal panels with different sweep angles:  $\theta = 30^\circ$  and  $\theta = 45^\circ$ . Digital particle image velocimetry (DPIV) was acquired for the panel with  $45^\circ$  edges at two Strouhal numbers, where the large scale structure generation and evolution was found to display some interesting qualitative differences when analyzed using a Lagrangian Coherent Structure (LCS) framework.

### 5.1 Dye flow visualization

Dye flow visualization was used to observe the large-scale structures produced by the trapezoidal panel pitching in a uniform flow. The two panel geometries, given in table 5.1, were chosen to vary sweep angle but to keep surface area constant. For all experiments with the trapezoidal panels, the trailing edge pitching amplitude ( $A$ ) was 20 mm, and the Strouhal number was varied by changing the freestream velocity. The dye was injected into the flow through small holes in the airfoil fairing. These dye injection ports spanned a 130 mm section of the fairing, and therefore, the width of the dye sheet was 61% of the trailing edge span of panel B, and 77% of the trailing edge span of panel C. For this reason, interactions of the flow close to the spanwise tips of the trailing edge were not captured with the flow visualization.

Panel	c	S	A/S	$\theta$
B	100mm	168mm	0.012	$30^\circ$
C	90mm	213mm	0.009	$45^\circ$

Table 5.1: Trapezoidal panel geometries

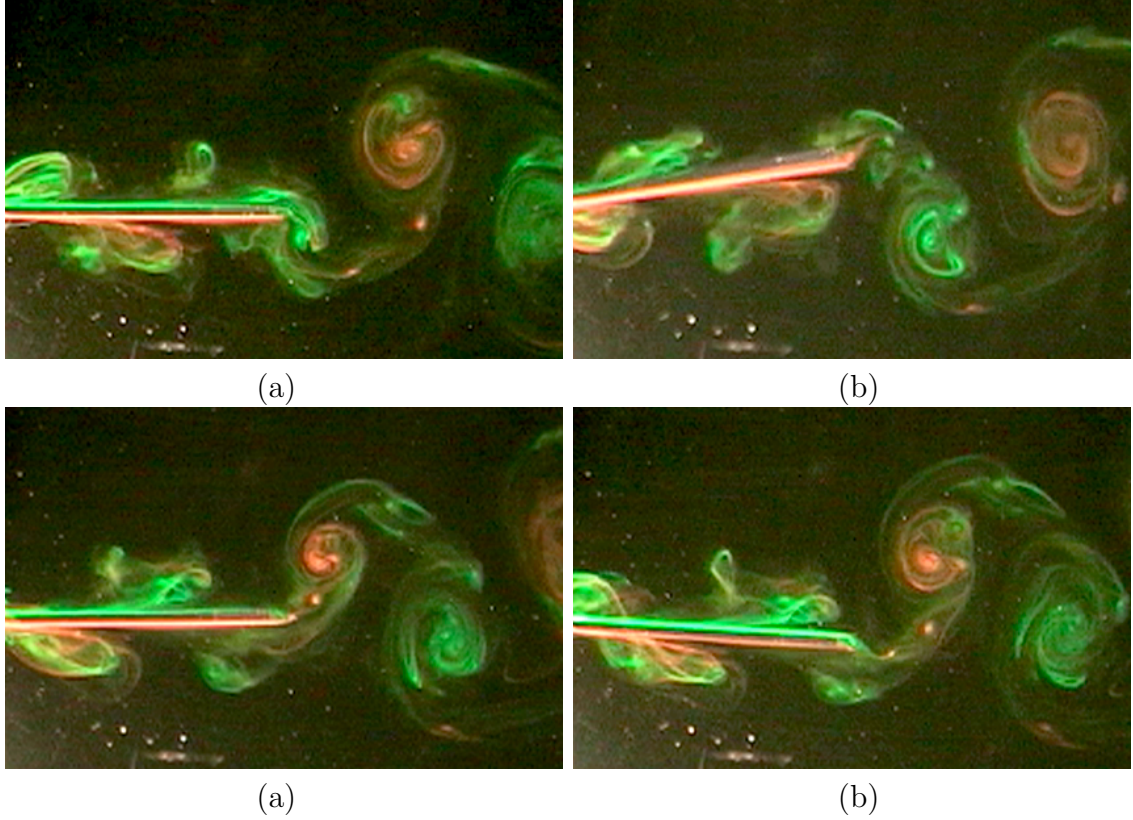


Figure 5.1: White light flow visualization of panel C pitching at  $St = 0.33$ ,  $Re_c = 2690$ , and  $A/S = 0.009$ , viewed from above. (a)  $\phi \approx 0^\circ$ , (b)  $\phi \approx 90^\circ$ , (c)  $\phi \approx 180^\circ$ , and (d)  $\phi \approx 270^\circ$ .

### 5.1.1 White light illumination

Figure 5.1 shows a view parallel to the pitching axis of white light illuminated dye in the flowfield around panel C with  $St = 0.33$ . There is a clear organization of the dye on the top and bottom surfaces of the panel. In particular, in figure 5.1(b) one organized structure is observed on the top surface, and two are observed on the bottom surface. Each structure is identified by a collection of dye, including both the green fluorescein and pink rhodamine, injected from opposite sides of the fairing. As we will discuss in this section, these structures are rolling up around the swept edges of the trapezoidal panel, and therefore entrain both dyes injected upstream of the swept edge.

As the Strouhal number is increased by decreasing the outer flow velocity, this organization of the dye does not persist. In figure 5.2(a–b), the Strouhal number has been increased to 0.42, and while there is still some coherence in the dye pattern around panel C, the packets are closer together and boundaries become less clear. When the Strouhal number is increased to 0.55, and as shown in figure 5.2(c–d), all coherence is lost.

A closer look at the dye flow dynamics at  $St = 0.33$  is shown in figure 5.3. The

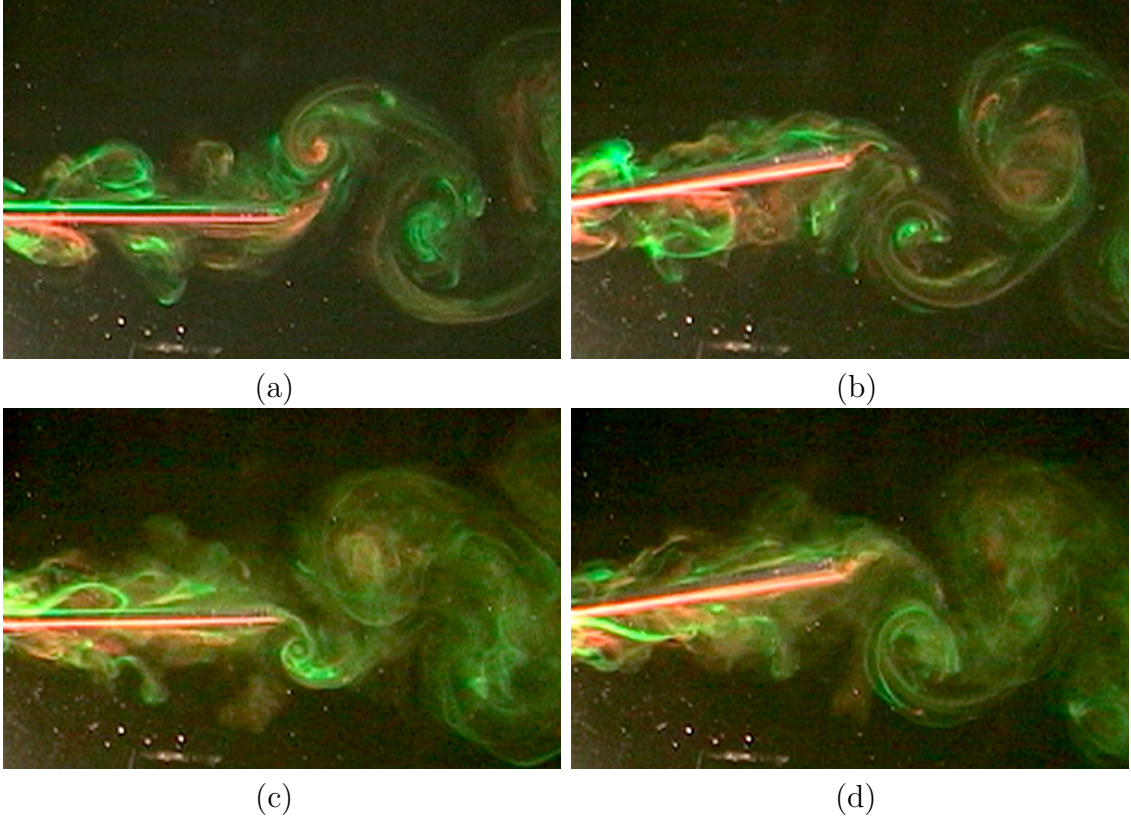


Figure 5.2: White light flow visualization of panel C pitching at two different Strouhal numbers, viewed from above. (a)  $St = 0.42, Re_c = 2150, A/S = 0.009, \phi \approx 0^\circ$ , (b)  $St = 0.42, Re_c = 2150, A/S = 0.009, \phi \approx 90^\circ$ , (c)  $St = 0.55, Re_c = 1615, A/S = 0.009, \phi \approx 0^\circ$ , and (d)  $St = 0.55, Re_c = 1615, A/S = 0.009, \phi \approx 90^\circ$ .

panel is shown at three phases of motion as it sweeps into the page. At the near extremum (figure 5.3(a)), a line of fluorescein is visible that extends onto the far side of the panel surface. As the panel sweeps through zero angle of attack (5.3(b)), the dye stretches across the swept edge. When the panel reaches the far extremum of its motion (5.3(c)), the dye has detached from the swept edge, leaving a packet on the far side of the panel as it begins to collect on the near side. This packet of dye will detach from the panel in the next half-cycle.

If the same three phases are examined when the panel is pitching at the higher Strouhal number ( $St = 0.55$ ), the detachment of the dye-marked structure is no longer apparent, as shown in figure 5.4. One particularly bright collection of dye is highlighted at each phase. When the panel is at its near extremum (5.4(a)), the dye has rolled up on the far surface. As the panel pitches into the page (5.4(b)), no stretching or detachment occurs, and the dye begins to flow over the swept edge. When the panel reaches the far extremum (5.4(c)), the dye has rolled up on the near side of the panel.

### 5.1.2 Planar laser-induced fluorescence

The structure generated by the motion of the swept edge is more clearly seen using Planar Laser-Induced Fluorescence (PLIF). Figure 5.5 shows PLIF images acquired at  $x = 0.25c$  at four phases of the panel motion with  $St = 0.33$ . This view is from downstream of the panel looking upstream, and the image plane is normal to both the streamwise flow. In this view, swirling dye flow marks the streamwise component of vorticity. In figure 5.5(a), the panel is at the right-most extremum, and the dye clearly marks the creation of a counterrotating vortex pair on the left side of the panel surface. This vortex pair has a component in the streamwise direction, but it is also expected to have a component in the spanwise direction, since it is aligned with the swept edge. We shall refer to this structure as a “quasi-streamwise structure” or as a “swept edge structure.” It is created as the pressure difference between the two surfaces of the panel draws fluid from the advancing surface (right) to the retreating surface (left). The fluid rolls around the swept edge, and the vortex pair is created as this flow impinges on the panel surface.

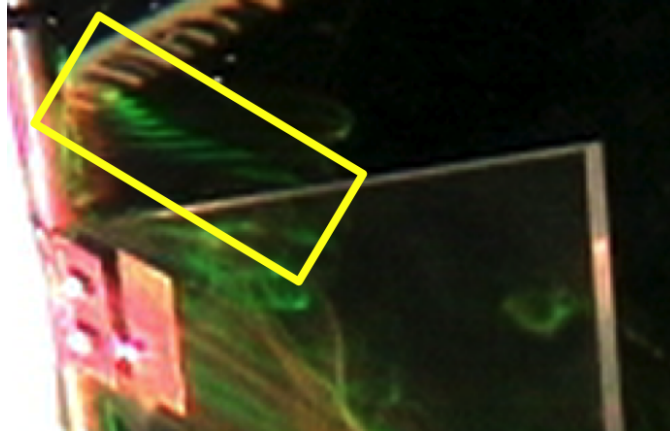
Figure 5.5(b) shows that this vortex pair persists as the panel crosses through the zero angle of attack, and by the time it has reached the left-most extremum (5.5(c)), a similar pair has been created on the now advancing surface (right) of the panel. This structure also persists as the panel sweeps to the right (5.5(d)).

The swept edge structure retains its coherence as it convects along the panel surface, as shown in figure 5.6. In this figure, PLIF images are shown at four locations along the panel chord, at phases chosen so that the swept edge structure generated at  $x \approx .025c$  passes through the laser plane. Similar vortex roll-ups at the swept edge are not observed downstream of  $x = 0.25c$  due to the limited spanwise extent of the dye injection, as indicated earlier. At  $x = 0.5c$  (5.6(b)), the counter-rotating vortices generated upstream are observed one quarter pitching period later. At this location, they have spread apart in the spanwise direction, but as they move downstream, their relative distance does not continue to increase.

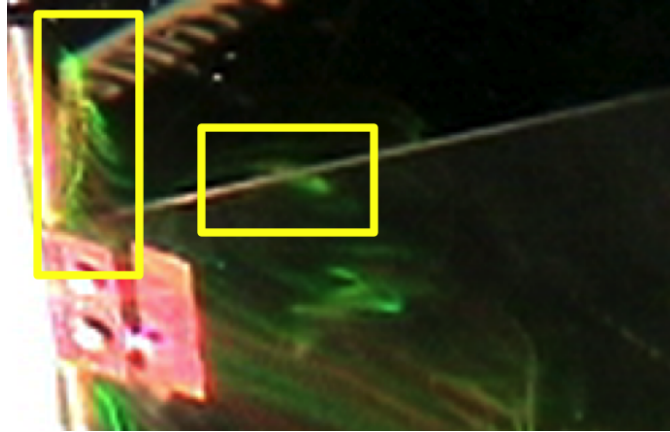




(a)



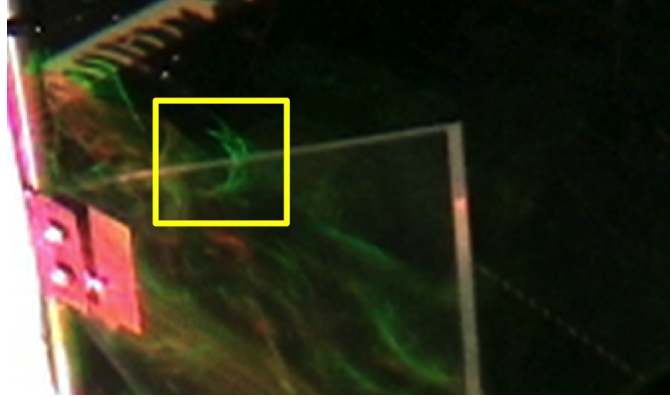
(b)



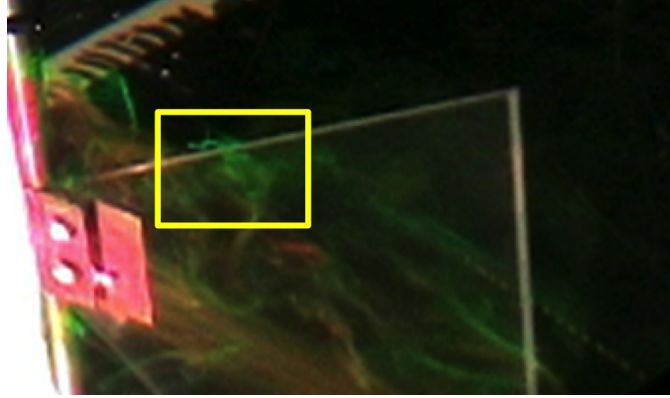
(c)

Figure 5.3: Angled view white light flow visualization of panel C pitching at  $St = 0.33$ ,  $Re_c = 2690$ , and  $A/S = 0.009$ . Yellow boxes highlight the line of dye stretching as fluid that was pulled onto the far surface in the previous half cycles remains trapped there, and does not wash back on the near surface as the panel pitches into the page. (a)  $\phi \approx 90^\circ$ , (b)  $\phi \approx 180^\circ$ , and (c)  $\phi \approx 270^\circ$ .





(a)



(b)



(c)

Figure 5.4: Angled view white light flow visualization of panel C pitching at  $St = 0.55$ ,  $Re_c = 1615$ ,  $A/S = 0.009$ . Yellow box highlights fluid pulled from the far surface to the near surface as the panel pitches into the page. (a)  $\phi \approx 330^\circ$ , (b)  $\phi \approx 30^\circ$ , and (c)  $\phi \approx 90^\circ$ .

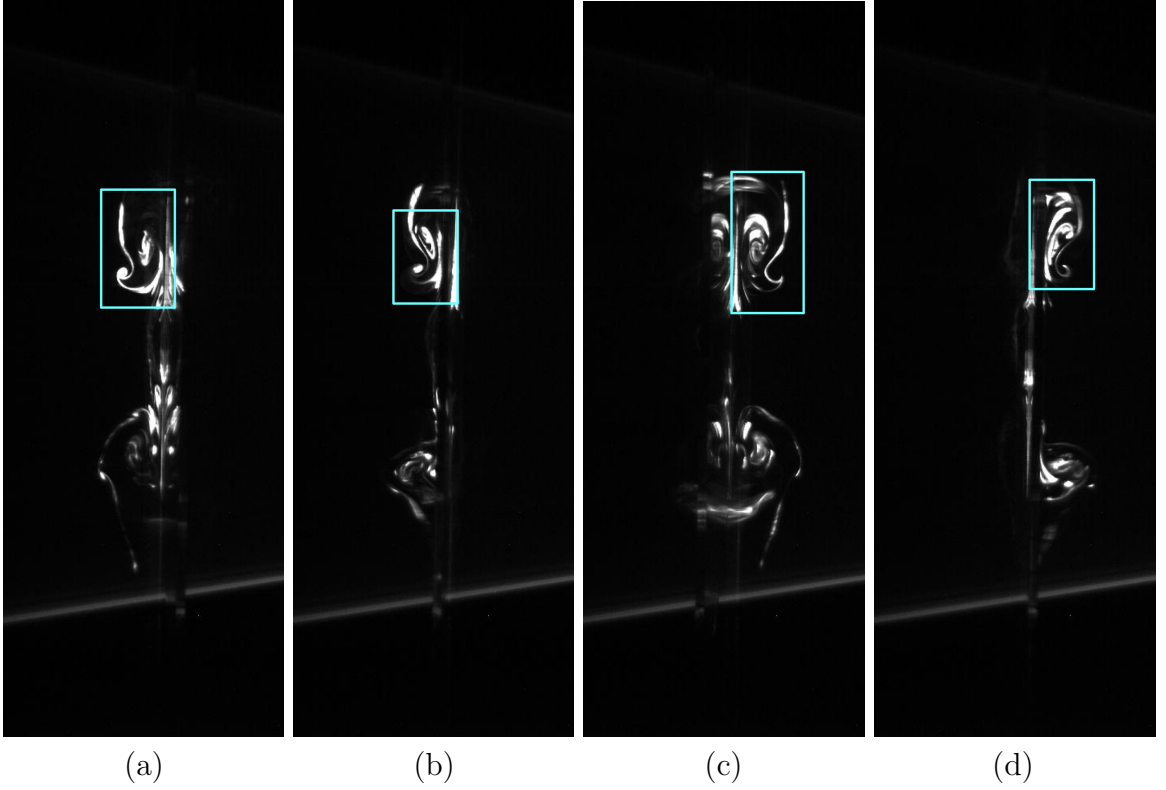


Figure 5.5: PLIF images on panel C pitching at  $St = 0.33$ ,  $Re_c = 2690$ , and  $A/S = 0.009$ , acquired at  $x = 0.25c$ . Blue boxes highlight counterrotating vortex pairs rolling up at the swept edge. (a)  $\phi \approx 90^\circ$ , (b)  $\phi \approx 180^\circ$ , (c)  $\phi \approx 270^\circ$ , and (d)  $\phi \approx 360^\circ$ .

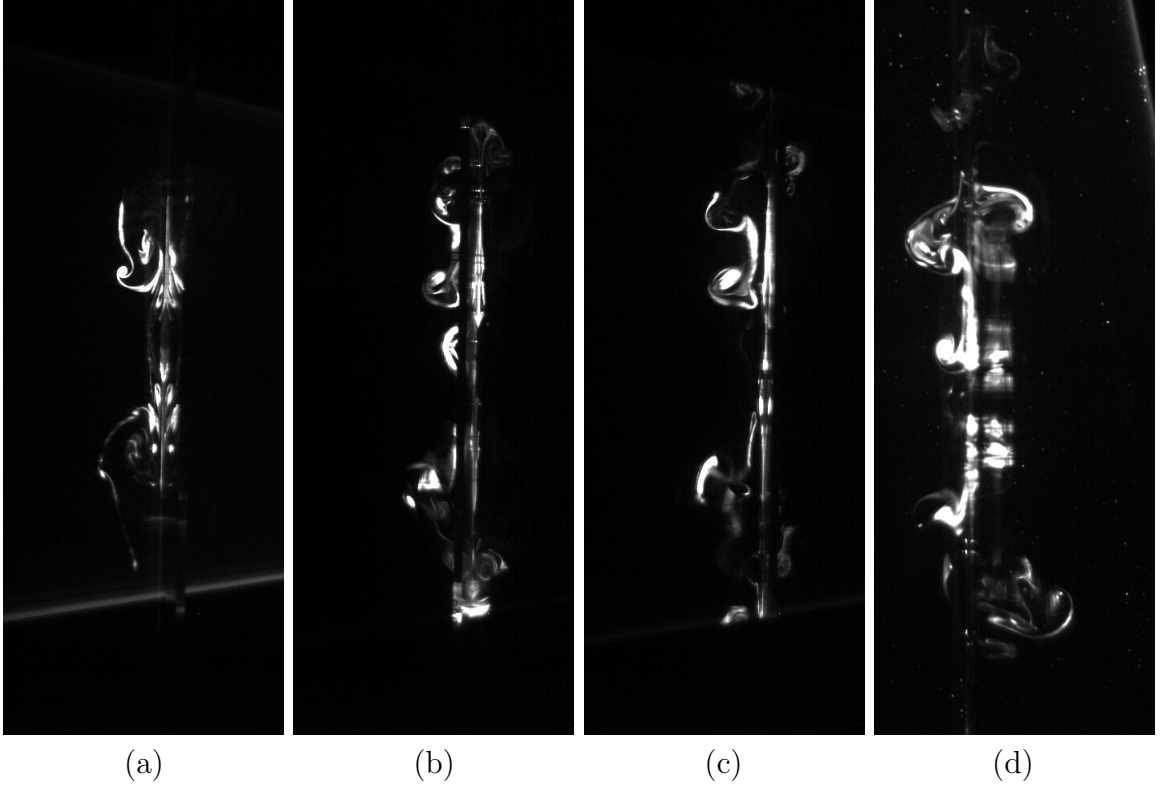


Figure 5.6: PLIF images on panel C pitching at  $St = 0.33$ ,  $Re_c = 2690$ , and  $A/S = 0.009$ , acquired at different streamwise locations along the chord. Phases are chosen to correspond to the crossing of the swept edge structure through the laser plane. (a)  $x = 0.25c$ ,  $\phi \approx 90^\circ$ , (b)  $x = 0.5c$ ,  $\phi \approx 180^\circ$ , (c)  $x = 0.75c$ ,  $\phi \approx 270^\circ$ , and (d)  $x \approx c$ ,  $\phi \approx 360^\circ$ .

Just downstream of the trailing edge, the structure generated at the swept edge at  $0.25c$  retains its shape, and is being shed by the trailing edge (figure 5.6(d)). It is expected that this quasi-streamwise structure interacts with the spanwise vortex shed by the trailing edge, but the nature of the interaction cannot be described using the flow visualization.

PLIF images were also acquired with the panel pitching at  $St = 0.55$ , and these are shown in figures 5.7 and 5.8. In figure 5.7, it is clear that a vortex pair rolls up along the swept edge as the panel moves from left to right. However, as the advancing surface decelerates, this pair does not continue to develop, and instead interacts with the swept edge. As shown in chapter 4, at this point in the phase the pressure on the advancing surface is high, with a favorable gradient toward the spanwise edges. This favorable gradient will induce the flow toward the edges, and break up the swept edge structures. In figure 5.7(b), it can be seen that this structure is already being pulled over the swept edge and losing coherence.

Figure 5.8 shows the PLIF visualization just downstream of the trailing edge with the panel pitching at  $St = 0.55$ . The vortex pair clearly seen at  $St = 0.33$  (figure 5.6(d)) is not observed at the same phase of motion, or indeed at any phase of motion.

We refer to the behavior of the swept edge structure at lower Strouhal numbers, especially with regard to its continued coherence down the panel and in the wake, as “structure trapping.” This phenomenon was observed to depend not only on Strouhal number, which was varied by changing the freestream velocity, but on the panel geometry as well. For example, flow visualization was also acquired for a trapezoidal panel with swept edges at a smaller angle to the freestream (panel B). As shown in figures 5.9(a–b), the swept edge structure generated at  $0.25c$  on panel C pitching at  $St = 0.42$  retains coherence and is “trapped” on the left surface as the panel sweeps from right to left. Images at  $x = 0.5c$  on panel B pitching at the same Strouhal number are shown in figures 5.9(c–d). Here, we see a structure roll up, but as the panel sweeps from right to left, the quasi-streamwise structure is pulled over the swept edge, and no trapping occurs. However, trapping was observed to occur on panel B for  $St < 0.37$ . Trapping was observed on panel C for  $St < 0.47$ . The swept edge structure trapping occurs when either freestream velocity is increased, or the swept edge angle is increased. To explain this, a schematic of the swept edge structure dynamics is shown in figure 5.10.

The swept edge structures outlined by the blue ovals were created in the previous half-cycle, as the panel pitched into the page. In this half-cycle, as it is pitching out of the page, they have convected downstream along the panel surface. It is expected that structure trapping occurs when the vortex has travelled far enough away from the swept edge such that it does not interact with the edge in the subsequent half-cycle. The “edge distance” in this half-cycle, indicated by a red line in figure 5.10, can be increased in two ways: by decreasing the Strouhal number by either decreasing the frequency or increasing the freestream velocity, or by increasing the sweep angle of panel. Both of these have proven successful in the current work. Increasing the freestream velocity increases the downstream distance the structure travels from the edge. Increasing the sweep angle of the panel geometry increases the streamwise

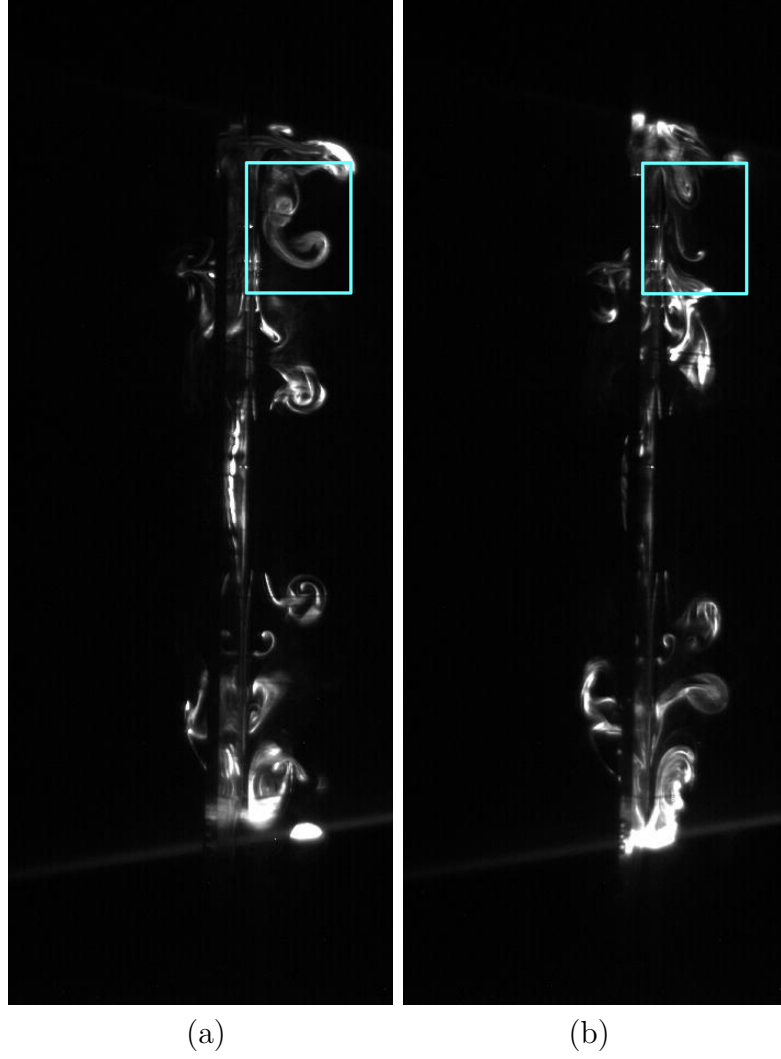


Figure 5.7: PLIF images on panel C pitching at  $St = 0.55$ ,  $Re_c = 1615$ , and  $A/S = 0.009$ , acquired at  $x = 0.5c$ . Blue boxes highlight the quasi-streamwise structure as it is created and then pulled over the swept edge. (a)  $\phi \approx 270^\circ$  and (b)  $\phi \approx 315^\circ$ .

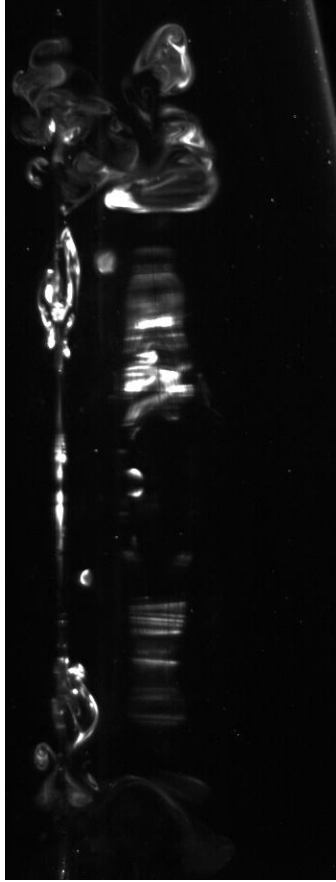


Figure 5.8: PLIF image of panel C pitching at  $St = 0.55$ ,  $Re_c = 1615$ , and  $A/S = 0.009$ , acquired  $x \approx c$  and  $\phi \approx 180^\circ$

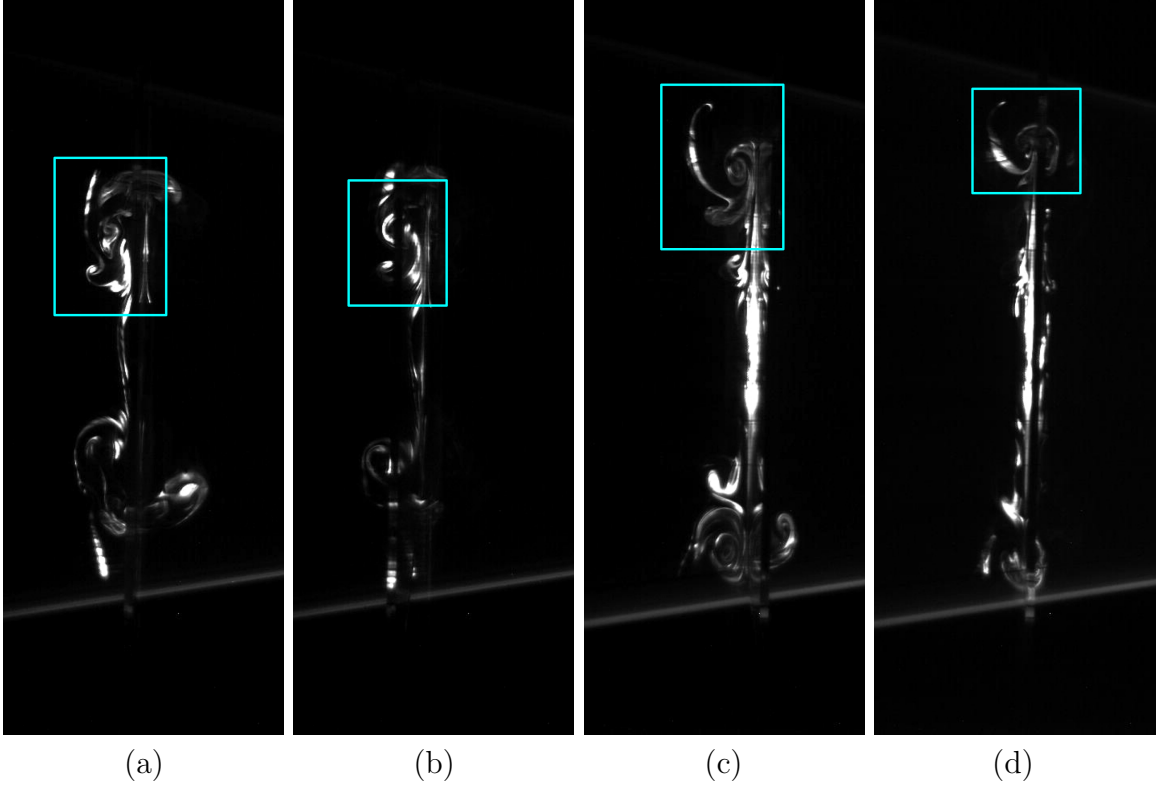


Figure 5.9: PLIF images of swept edge structure roll-up on panels B (at  $x = 0.25c$ ,  $St = 0.42$ ,  $Re_c = 2390$ , and  $A/S = 0.012$ ) and C (at  $x = 0.5c$ ,  $St = 0.42$ ,  $Re_c = 2150$ , and  $A/S = 0.009$ ). Blue box highlights the structure of interest. (a) Panel C,  $\phi \approx 180^\circ$ , (b) panel C,  $\phi \approx 270^\circ$ , (c) panel B,  $\phi \approx 90^\circ$ , and (d) panel B,  $\phi \approx 180^\circ$ .

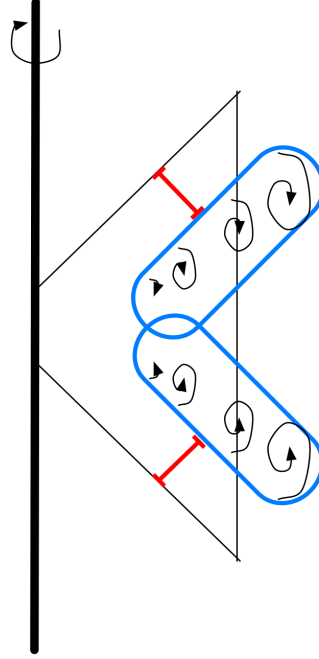


Figure 5.10: Schematic of quasi-streamwise structure dynamics. Structure outlined in blue, distance from the swept edge indicated in red.

component of the edge distance.

Between panels B and C at the same freestream velocity, swept edge structures achieve a larger edge distance on panel C due to the larger sweep angle. Therefore, a higher freestream velocity (lower  $St$ ) was needed for the same phenomenon to occur on panel B.

## 5.2 Digital particle image velocimetry

To quantitatively investigate the three-dimensional wake around the rigid trapezoidal panel, spatially- and temporally-resolved two-dimensional digital particle image velocimetry (2D DPIV) was acquired, as described in chapter 3. The trapezoidal panel used for these experiments had a sweep angle  $\theta = 45^\circ$ , chord  $c = 70$  mm, and trailing edge span  $S = 175$  mm. Eulerian and Lagrangian methods were used to analyze the resulting data sets.

### 5.2.1 Eulerian analysis

At each point in the volume, velocities in the streamwise ( $u$ ) and transverse ( $v$ ) directions are acquired. A second-order accurate central differencing scheme is used to calculate the spatial derivatives of these quantities, and these were in turn used to



calculate spanwise vorticity,

$$\omega_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}. \quad (5.1)$$

In figure 5.11, the spanwise vorticity distribution in four planes along the span of the trapezoidal pitching panel at  $St = 0.33$  are plotted. The contour levels are the same in all planes. At both Strouhal numbers,  $\omega_{z,max/min} = \pm 20 \text{ s}^{-1}$ . An appreciable decrease in spanwise vorticity magnitude is apparent away from the midspan of the panel. In addition, near the midspan (plane 1), the vortices are slightly staggered along the transverse centerline. Induced flow between each consecutive vortex pair is directed slightly downstream, adding a small amount of streamwise momentum, as shown by the black arrows in figure 5.11. However, as we move away from the midspan, the alignment of the vortices shifts, and near the trailing edge tip (plane 4) the induced jets between pairs of vortices is directed upstream, indicating momentum loss (drag production).

The spanwise vorticity is also plotted for the trapezoidal panel pitching at  $St = 0.55$  in figure 5.12. Again, at the midspan the vortices are aligned to induce velocity downstream, but this changes towards the trailing edge tip. In addition, at the midspan the 2S vortex configuration is observed to break down approximately one chord length downstream of the trailing edge.

For a clearer illustration of the organization of spanwise vorticity, isosurfaces of  $\omega_z$  are shown in figure 5.13 at two phases of motion with  $St = 0.33$ . The spanwise extent of the wake is contained within the span of the trailing edge, and decreases slightly as the structures move downstream. Isosurfaces of  $\omega_z$  with the panel pitching at  $St = 0.55$  are shown in figure 5.14. Here, the spanwise compression of the wake is more exaggerated, and there is also a marked decrease in vorticity magnitude downstream, consistent with a breakdown of the 2S wake structure.

A view of the edge of both wakes, from downstream and above the panel, is shown in figure 5.15. For both Strouhal numbers, the change in vortex alignment away from the midspan is observed. Structures of positive spanwise vorticity (red) bend to the left, and those of negative spanwise vorticity (blue) bend to the right.

### 5.2.2 LCS analysis

To more thoroughly describe the vortex interactions caused or induced by three-dimensional effects and wake breakdown, an LCS analysis like that of chapter 2 was conducted using the 2D velocity fields obtained by the DPIV experiments. The spanwise velocity ( $w$ ) was assumed to be zero, which is not a good assumption in these highly three-dimensional flows, but the large-scale structures in this wake are expected to be associated primarily with the spanwise vorticity. A series of two-dimensional planes is used to present the LCS results, as the boundary of the wake as indicated by the LCS would obscure the inner detail.

In figure 5.16, positive- and negative-time LCS, pLCS and nLCS, respectively, are shown at three spanwise locations in the wake of the panel pitching at  $St = 0.33$ . At this Strouhal number, all DLE calculations were done using an integration time of two

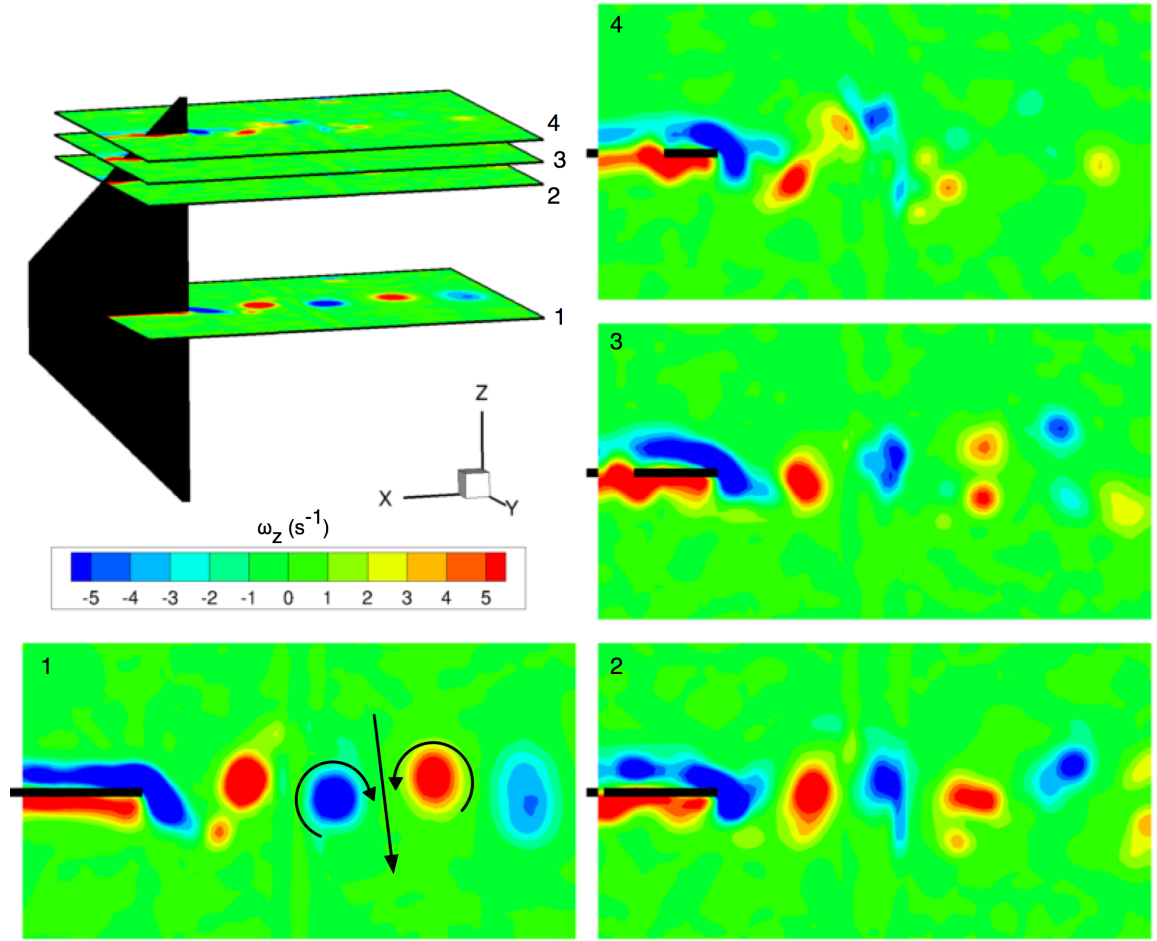


Figure 5.11: Two-dimensional planes of spanwise vorticity in four planes along the span of the trailing edge.  $St = 0.33$ ,  $Re_c = 2390$ , and  $A/S = 0.012$ . Plane 1:  $0.5S$ , plane 2:  $8.67S$ , plane 3:  $0.79S$ , and plane 4:  $0.9S$ .

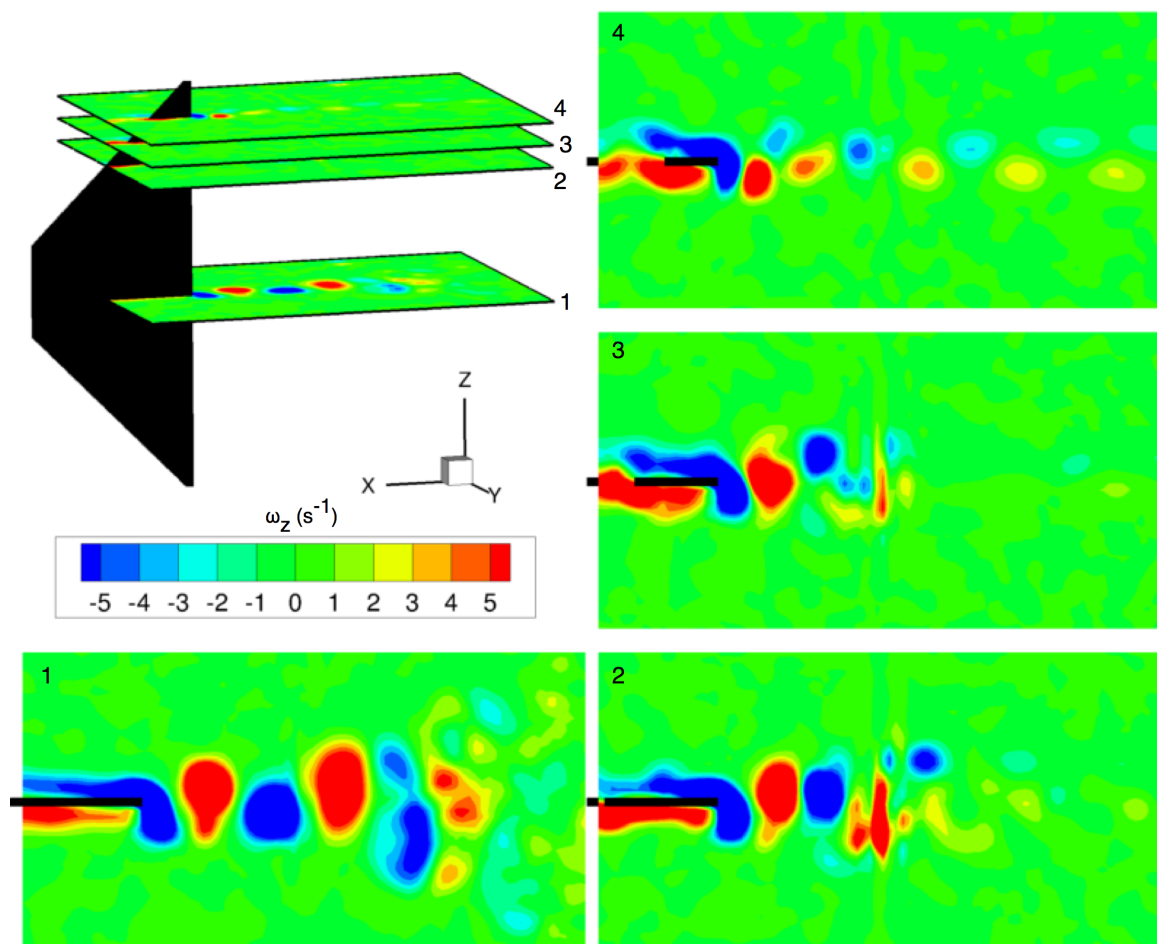


Figure 5.12: Two-dimensional planes of spanwise vorticity in four planes along the span of the trailing edge.  $St = 0.55$ . Plane 1:  $0.5S$ , plane 2:  $0.79S$ , plane 3:  $0.84S$ , and plane 4:  $0.95S$ .

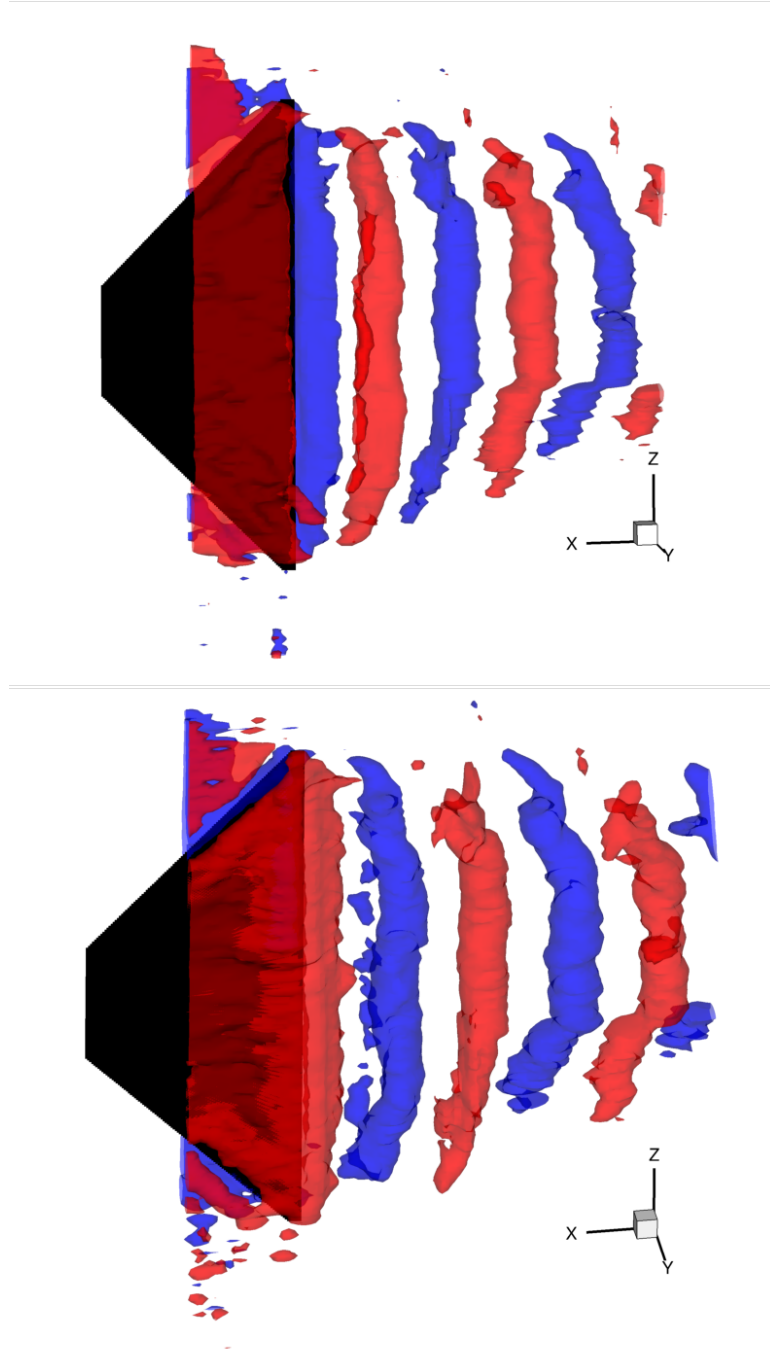


Figure 5.13: Spanwise vorticity for the panel pitching at  $St = 0.33$ . Red and blue surfaces are 18% maximum and minimum  $\omega_z$ , respectively. (a)  $\phi = 0^\circ$  and (b)  $\phi = 180^\circ$ .

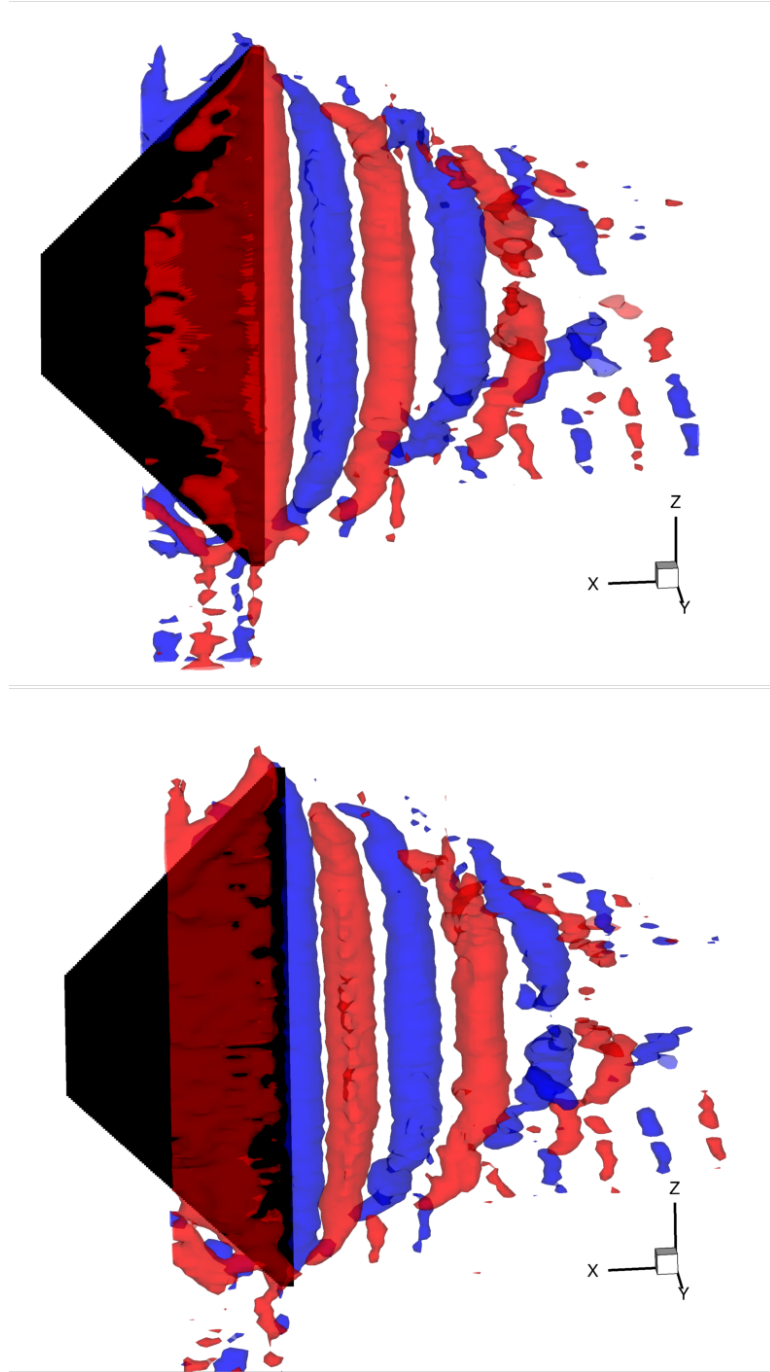


Figure 5.14: Spanwise vorticity for the panel pitching at  $St = 0.55$ . Red and blue surfaces are 17% maximum and minimum  $\omega_z$ , respectively. (a)  $\phi = 0^\circ$  and (b)  $\phi = 180^\circ$ .

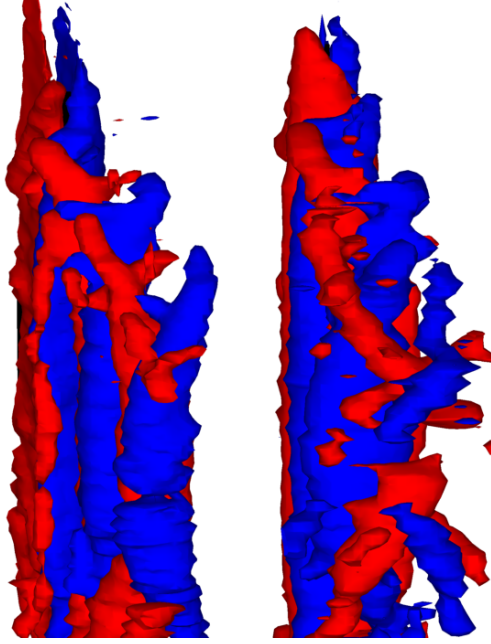


Figure 5.15: Angled view of the isosurfaces of  $\omega_z$  at both Stouhal numbers. Red and blue surfaces are 18% maximum and minimum  $\omega_z$ , respectively. (a)  $St = 0.33, \phi = 0^\circ$  and (b)  $St = 0.55, \phi = 180^\circ$ .

pitching periods for both positive- and negative-time calculations. The integration scheme in the DLE calculation was accelerated by exploiting the periodicity of the phase-averaged velocity fields. A description of this algorithm is given in Appendix A.

Particles that leave the domain were assumed to travel in the streamwise direction with the freestream velocity. For those trajectories that are advected upstream of the data domain, a uniform freestream assumption seems reasonable, as this is the boundary condition upstream of the pitching apparatus. Downstream of the data, however, the loss of information inherent in this assumption causes a lack of sharp ridges in the positive-time DLE field. In many of the figures in this chapter, the pLCS often are not revealed in the downstream half of the data domain for this reason.

At the midspan (figure 5.16(a)), two spanwise vortices downstream of the trailing edge are clearly indicated by both the pLCS and nLCS. Away from the midspan (5.16(b–c)), the qualitative structure of the LCS changes. There is little or no detail in the interior of the vortex cores, and the alignment has changed. Close to the spanwise tip of the trailing edge (5.16(c)), there are no clear pLCS, and therefore no clear repelling material lines in the wake structure. At this spanwise location, three-dimensional effects are assumed to be relatively large (compared to the midspan) and so the loss of information could be the result of not accurately following trajectories as they are leaving the plane.

To more closely describe the dynamical structure of the wake in each 2D plane, the hyperbolicity criteria were applied to the pLCS and nLCS, and the resulting structures are shown in figure 5.17. At the midspan (5.17(a)), the familiar pattern appears

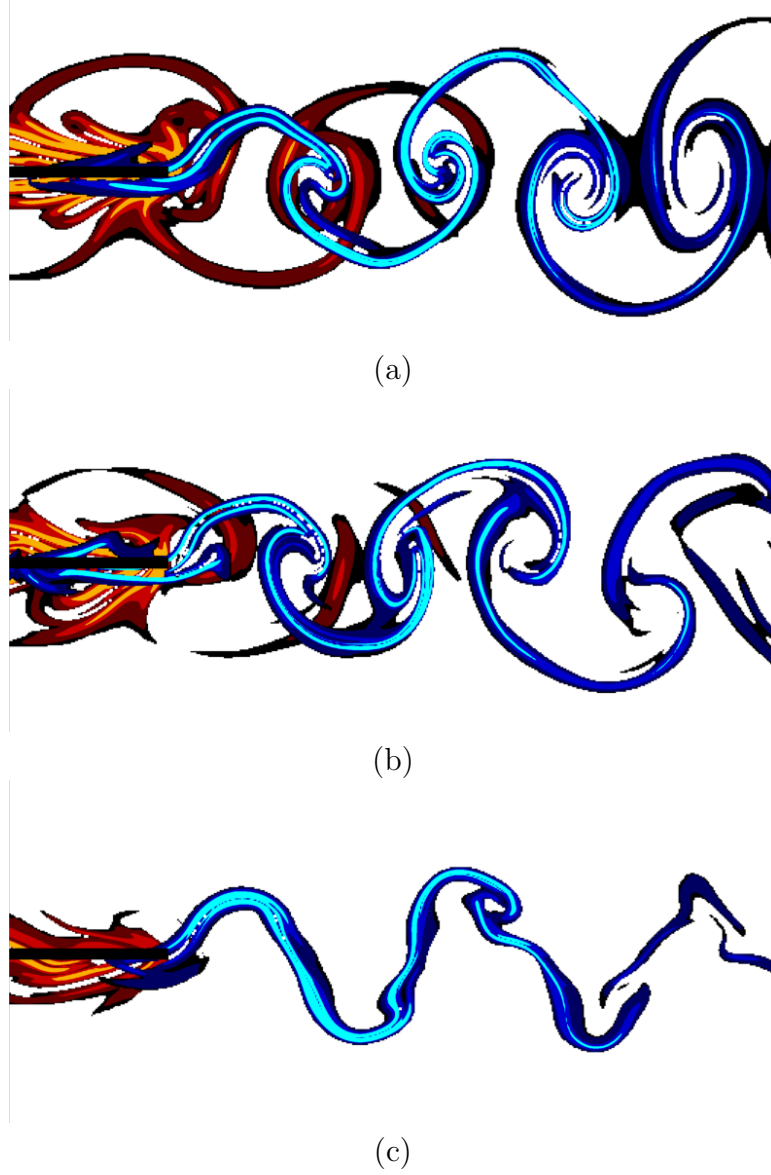


Figure 5.16: pLCS (red) and nLCS (blue) in three two-dimensional planes along the span of the panel pitching at  $St = 0.33$ . LCS are represented by DLE fields that are blank for values less than 33% maximum of the field. (a)  $0.5S$ , (b)  $0.79S$ , and (c)  $0.95S$ .

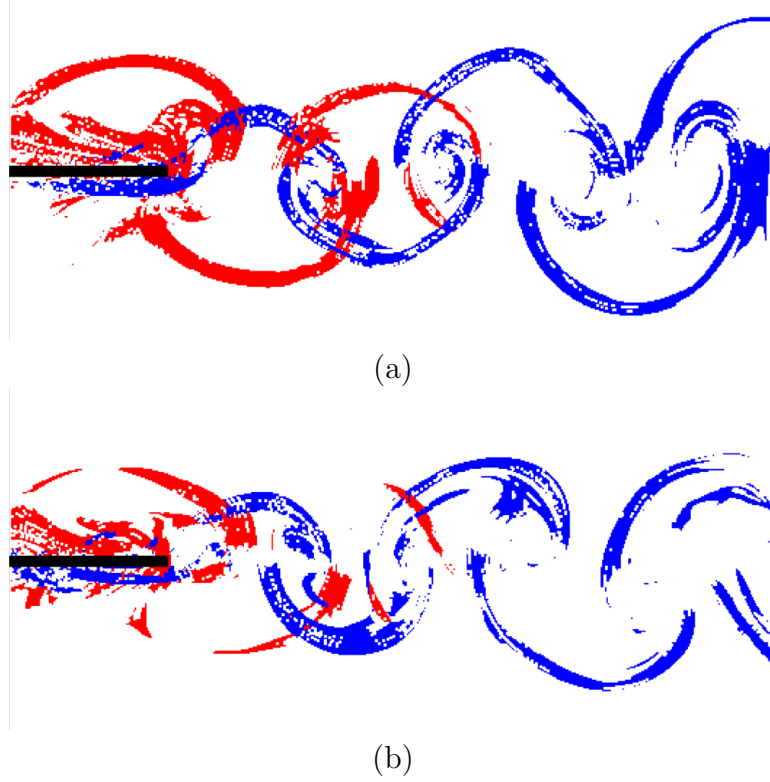


Figure 5.17: Hyperbolic pLCS and nLCS in three two-dimensional planes along the span of the panel pitching at  $St = 0.33$ . Hyperbolic LCS are represented by DLE fields that are blank for values less than 33% maximum of the field, and if the applicable strain criterion is not satisfied. (a)  $0.5S$  and (b)  $0.79S$ .

around the spanwise vortex cores. The boundary of these structures is indicated by a time-dependent saddle point at each transverse end of each structure. Within each vortex core, an alternating pattern of pLCS and nLCS is apparent, similar to that of the vortex cores seen in the wall-bounded turbulence of chapter 2. Again, this pattern is only obvious for approximately one half of the streamwise domain, because of the loss of information during the pDLE calculation.

The spanwise vorticity isosurfaces are plotted simultaneously with the LCS in figure 5.18. It is clear that both the Eulerian and Lagrangian techniques are revealing the same large scale structures in the pitching panel wake. It is clear that the spanwise structures are losing strength away from the midspan, and realigning. It is unclear whether the loss of magnitude in the vorticity calculation and loss of detail in the LCS calculation are caused by a strict weakening of the structures, or if the vortices are bending such that there is a component of the vorticity in the transverse and streamwise directions, and larger out of plane velocities.

In figure 5.19,  $\omega_z$  and LCS are compared more directly in 2D planes along the span of the wake. Both methods track the locations of the spanwise vortices well, especially as the alignment changes, as seen in (5.19(b-c)) at distances from the lower



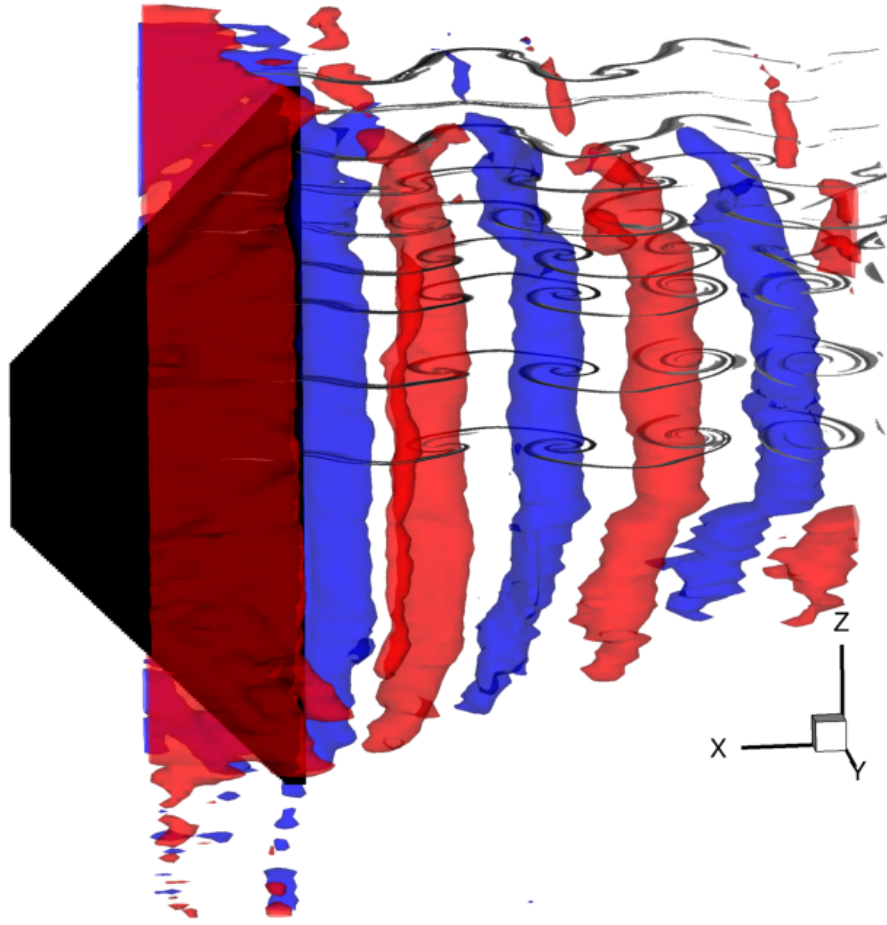


Figure 5.18: Isosurfaces of spanwise vorticity and two-dimensional slices of nLCS around the panel pitching at  $St = 0.33$ . Red and blue surfaces are 18% maximum and minimum  $\omega_z$ , respectively. LCS are represented by regions where  $nDLE > 43\%$  maximum value.

trailing edge tip of  $0.79S$  and  $0.84S$ . One detail apparent in the Eulerian results that does not appear in the Lagrangian structures is the roll-up of the vortex sheet that occurs just downstream of the trailing edge, which is highlighted in figure 5.19(a). This structure is a coherent concentration of vorticity, but seems to have relatively small dynamical influence compared to the larger scale spanwise vortices.

LCS of the panel wake at  $St = 0.55$  is shown in figure 5.20. For this Strouhal number, an integration time of four pitching cycles was used. Immediately downstream of the trailing edge, the qualitative structure of this wake is similar to that at the lower Strouhal number with the familiar scroll pattern bounding the spanwise vortex cores. As the vortices move downstream, however, they become more complicated. As was shown in figure 5.14, three spanwise structures are arranged in a 2S street upstream of the spanwise vorticity break down. Here, we see a transition in the LCS pattern at the same location. Also, there is a similar change in pattern and vortex pair alignment observed in the 2D LCS away from the midspan, and these changes are occurring even closer to the midspan than observed at  $St = 0.33$ .

The hyperbolicity criteria were also applied to the pLCS and nLCS of the higher Strouhal number wake, and the results are shown in figure 5.21. At the midspan (5.21(a)), the characteristic pattern of vortex cores bounded by two time-dependent saddles and an alternating pLCS/nLCS boundary is apparent in the near wake. Away from the midspan, the distance downstream of the trailing edge over which the LCS retain their coherence decreases.

The hyperbolic LCS following the evolution of the wake at the midspan are shown in figure 5.22 at four phases. A bifurcation in the structure of the LCS is seen as the structures move downstream. Black circles highlight two time-dependent saddle points as they convect downstream. The distance between them decreases until they merge into each other after one pitching cycle. More information is needed, particularly pLCS further downstream, to understand the dynamics of this progression. It is unclear whether the two saddle points are converging or if a more complicated bifurcation is occurring.

The isosurfaces of spanwise vorticity are superimposed on planes of nLCS in figure 5.23. In two planes, a yellow box indicates on the nLCS the location of the saddle-point merging, confirming that the dynamical system bifurcation and the qualitative wake transition are occurring at the same location.

This feature is further illustrated in figure 5.24, which shows the spanwise vorticity and nLCS in 2D planes. Again, both the Eulerian and Lagrangian techniques capture the same structures, and both the saddle-point merging and the spanwise vorticity breakdown are shown to occur at the same streamwise location.

The existence of the trapped structure observed in § 5.1.2 was not obvious when plotting spanwise vorticity in § 5.2.1. In the DLE calculation, the trajectories were assumed to have no spanwise velocity and to stay in the plane in which they were initialized. Therefore, in order to investigate the quasi-streamwise structure observed in the flow visualizations, nDLE was calculated in a plane normal to the streamwise flow and normal to the panel surface.

The results are shown in figure 5.25. As the panel pitches from right to left, a strong nLCS, associated with the intersection of the nDLE field with the finite-

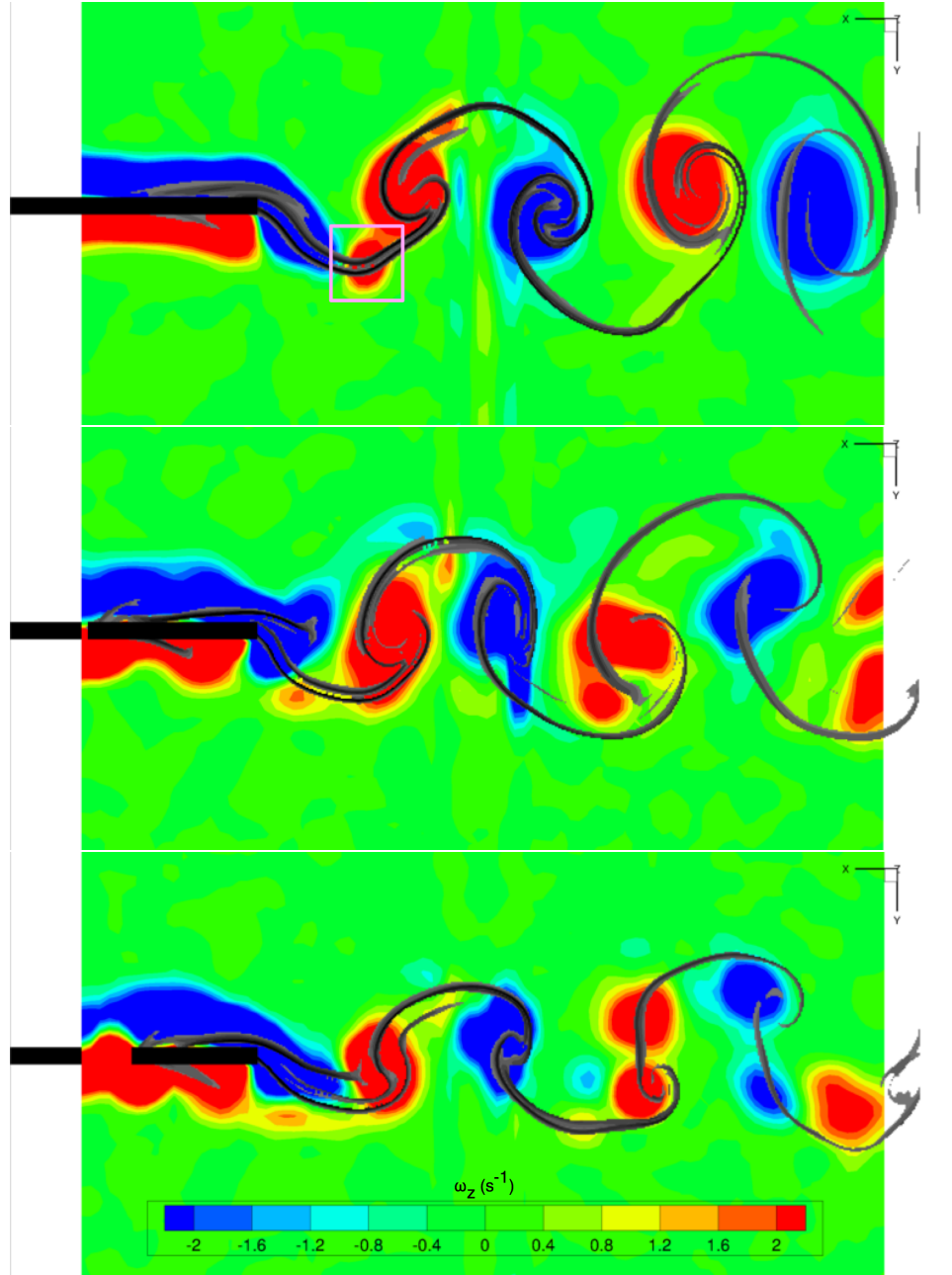


Figure 5.19: nLCS superimposed on two-dimensional planes of spanwise vorticity. nLCS are represented by regions where nDLE  $> 33\%$  maximum value. (a)  $0.5S$  (pink box highlights secondary vorticity sheet roll-up), (b)  $0.79S$ , and (c)  $0.84S$ .



Figure 5.20: pLCS and nLCS in three two-dimensional planes along the span of the panel pitching at  $St = 0.33$ . LCS are represented by DLE fields that are blank for values less than 33% maximum of the field. (a)  $0.5S$ , (b)  $0.67S$ , and (c)  $0.79S$ .

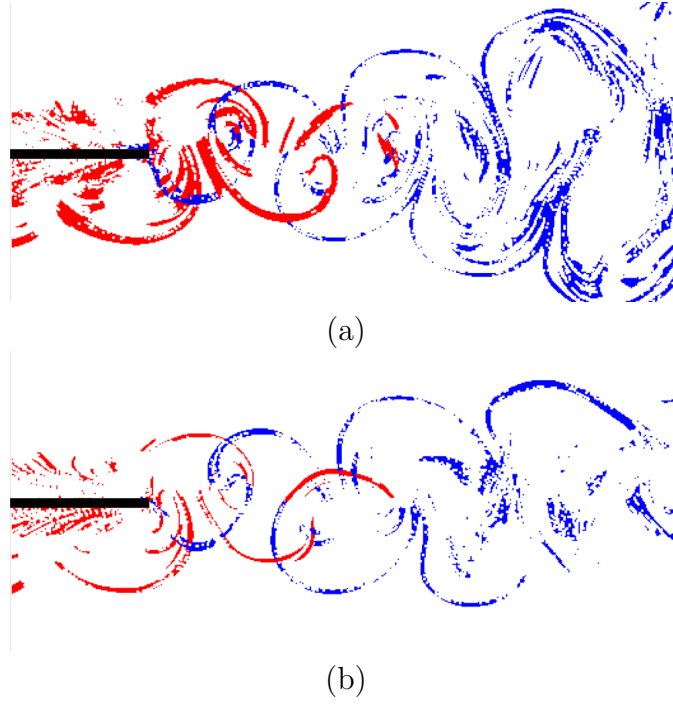


Figure 5.21: Hyperbolic pLCS and nLCS in three two-dimensional planes along the span of the panel pitching at  $St = 0.55$ . Hyperbolic LCS are represented by DLE fields that are blank for values less than 32% maximum of the field, and if the applicable strain criterion is not satisfied. (a)  $0.5S$  and (b)  $0.67S$ .

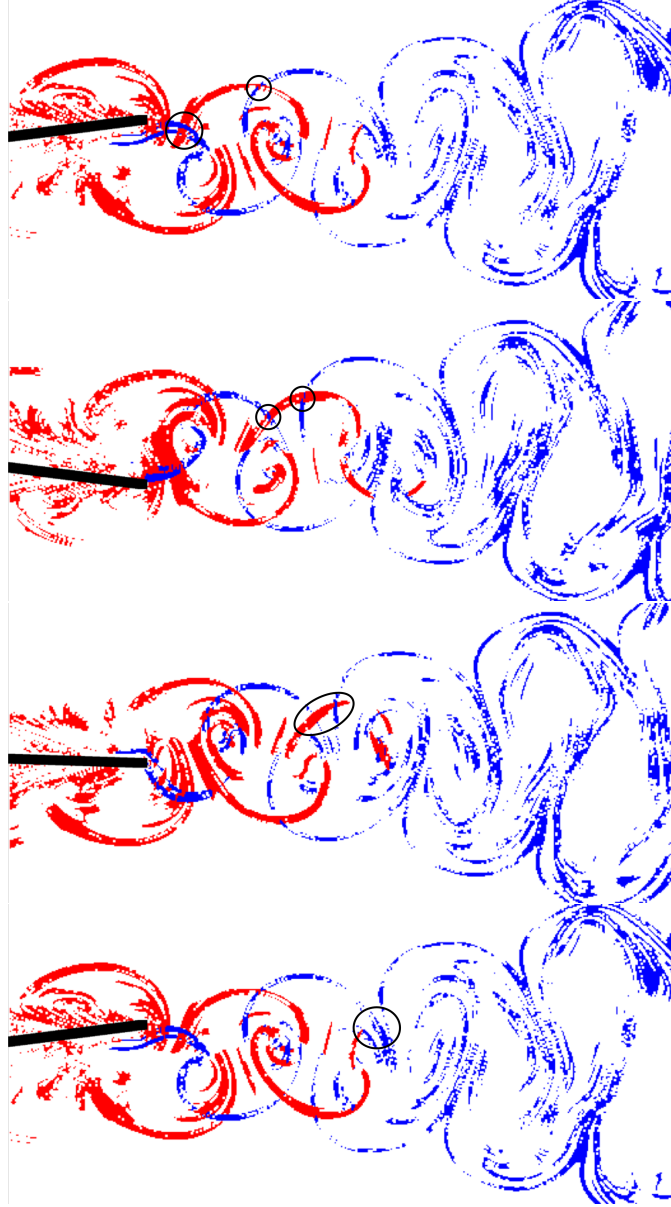


Figure 5.22: Hyperbolic pLCS and nLCS at the midspan of the panel pitching at  $St = 0.55$ . Hyperbolic LCS are represented by DLE fields that are blank for values less than 32% maximum of the field, and if the applicable strain criterion is not satisfied. (a)  $\phi = 100^\circ$ , (b)  $\phi = 260^\circ$ , (c)  $\phi = 345^\circ$ , and (d)  $\phi = 100^\circ$ .

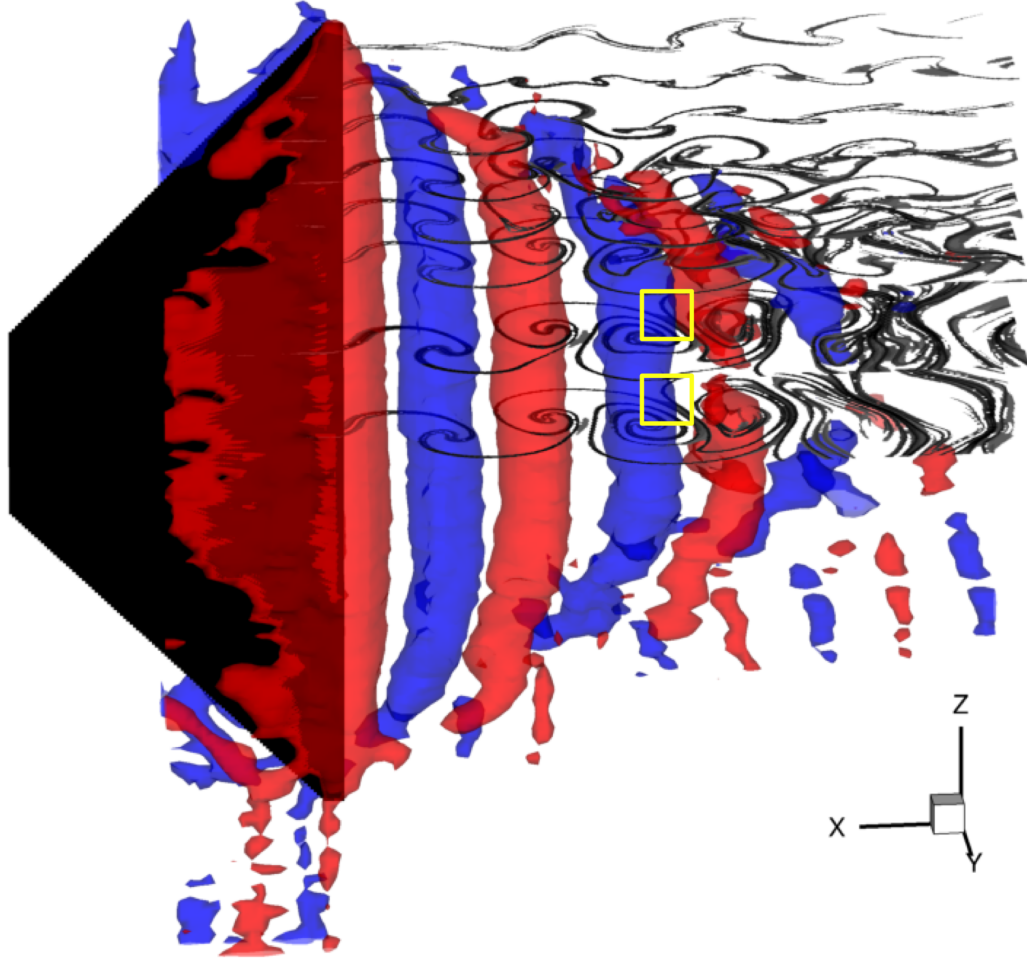


Figure 5.23: Isosurfaces of spanwise vorticity and two-dimensional slices of nLCS around the panel pitching at  $St = 0.33$ . Red and blue surfaces are 18% maximum and minimum  $\omega_z$ , respectively. LCS are represented by regions where nDLE  $> 43\%$  maximum value. Yellow boxes indicate locations of time-dependent saddle merges in the two planes closest to the midspan.

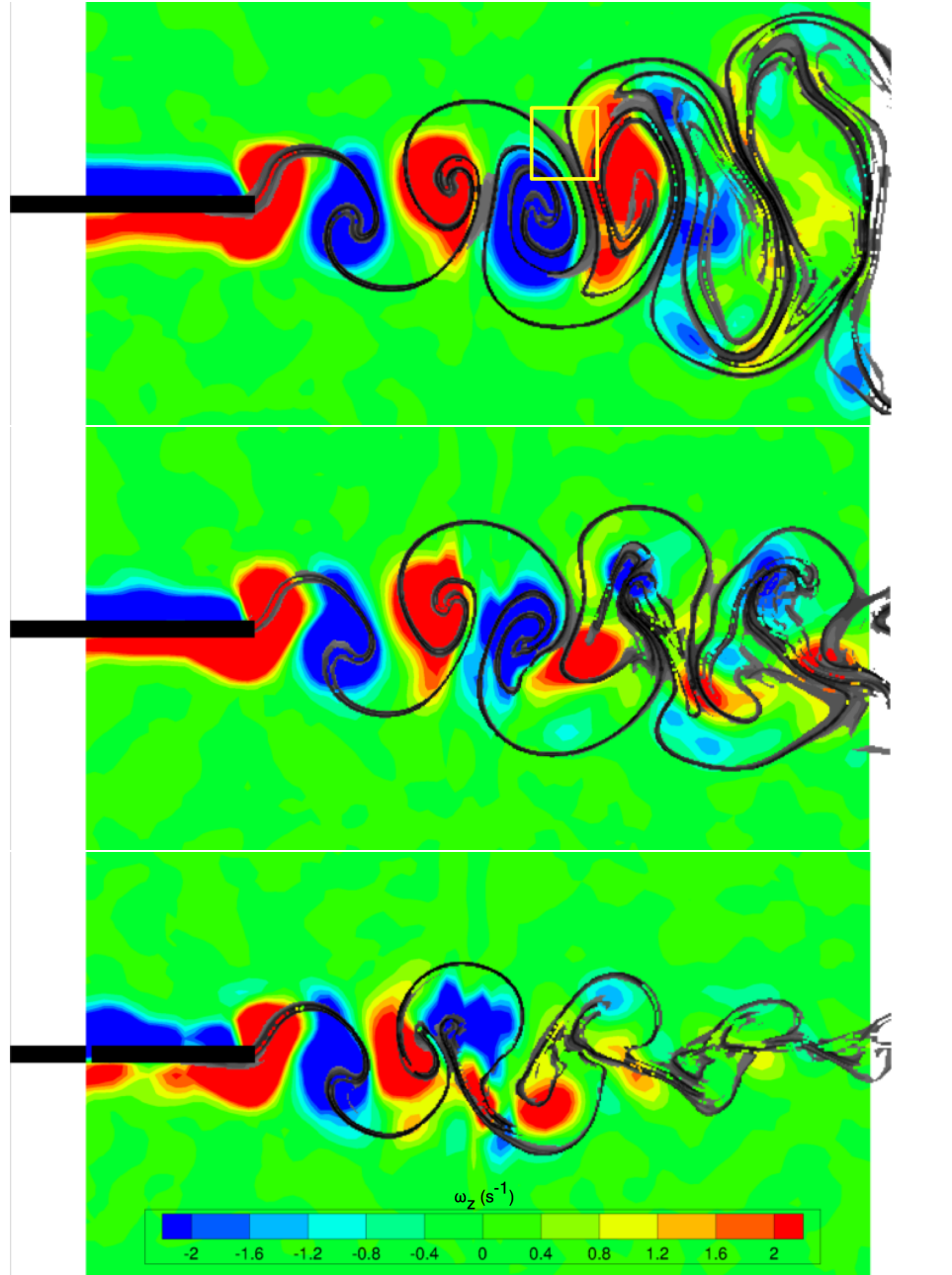


Figure 5.24: nLCS superimposed on two-dimensional planes of spanwise vorticity. nLCS are represented by regions where nDLE  $> 32\%$  maximum value. (a)  $0.5S$  (time-dependent saddle merge highlighted by yellow box), (b)  $0.67S$ , and (c)  $0.79S$ .



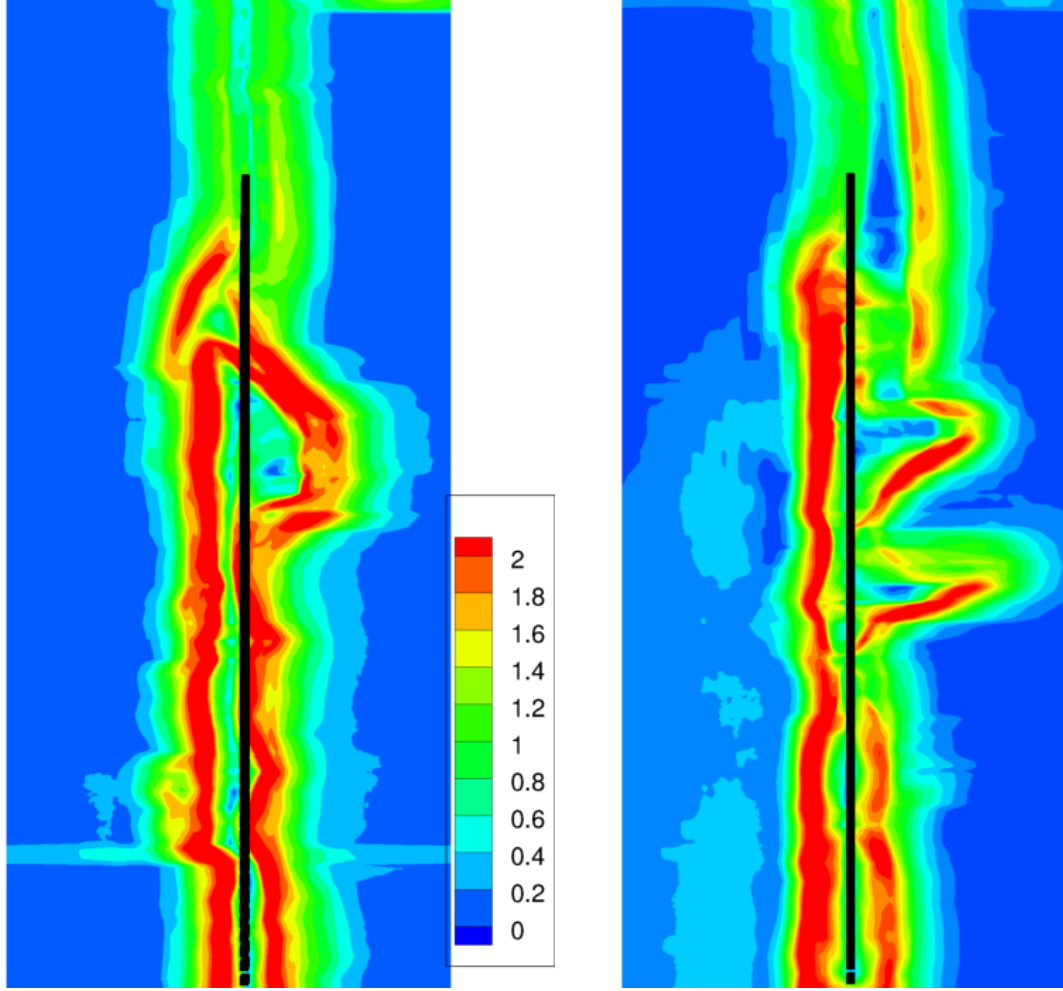


Figure 5.25: Two-dimensional nDLE field at  $x = 0.83c$  on the panel pitching at both Strouhal numbers. (a)  $St = 0.33$  and (b)  $St = 0.55$ .

thickness panel, is observed at both Strouhal numbers. For the lower Strouhal number (5.25(a)), an additional nLCS demarcates a boundary between section of fluid close to the panel surface and the outer flow. This is the case for which flow visualization showed that structures generated by the swept edge were trapped. At the higher Strouhal number (5.25(b)), at which no structure trapping was observed in the flow visualization, no clear boundary is apparent in the nDLE field that would separate flow near the panel from the freestream.

### 5.3 Summary

A series of flow visualization and DPIV experiments were conducted to investigate the three-dimensional wake structure of the rigid trapezoidal pitching panel. Flow visualization indicated that at lower Strouhal numbers, quasi-streamwise structures

were being generated at the swept edges of the panel and convecting along the panel surface while maintaining their coherence. This phenomenon was called “vortex trapping.” At higher Strouhal numbers, these structures continually interacted with the swept edge and did not retain coherence, and the vortices were not “trapped.” The trapped structures were not conclusively identified using Eulerian methods, but a negative-time LCS analyses confirmed that for  $St = 0.33$ , a region of fluid is separated by material lines from the outer flow during one half-cycle. A clear boundary was not apparent at  $St = 0.55$ . The fact that this phenomenon occurs at a Strouhal number closer to that at which real fish swim, and that it is not observed on panels of rectangular planform, might indicate an advantage of trapezoidal geometry.

The LCS analysis was employed, in addition to spanwise vorticity, to form a more complete description of the full three-dimensional wake downstream of the panel. At two Strouhal numbers, the spanwise extent of the wake was observed to decrease as the structures convected downstream. Also, the alignment of the spanwise vortices relative to each other was observed to change away from the midspan. In planes further from the midspan, the alignment was such that induced velocities between consecutive pairs of vortices pointed upstream, subtracting momentum from the streamwise flow and indicating drag production. This reinforces the conclusions of chapter 4, that three-dimensionality affects both the forces on the panel and wake structure and evolution.

A relationship between the qualitative structure of the wake and an underlying quantitative dynamical systems structure was found by using the LCS analysis. In those regions in which the wake consisted of a street of vortices, a characteristic pattern was seen in the hyperbolic nLCS and pLCS plots, which was similar to that described in chapter 2 around the core of a hairpin vortex. At  $St = 0.55$ , when the wake was observed to break down and possibly transition into a 2P structure, the dynamical structure of the flow, as represented by the LCS, exhibited a bifurcation in which two time-dependent saddle points merged. Because the hyperbolic saddle points can be quantitatively identified, the presence of the bifurcation in the Lagrangian results can provide a more accurate notification and description of the wake transition.

# Chapter 6

## Conclusions and Future Work

### 6.1 Discussion and conclusions

#### 6.1.1 Coherent structure identification

Lagrangian Coherent Structures (LCS) are employed as a vortex identification and analysis tool in chapters 2 and 5. One of the greatest benefits of using LCS, instead of previously published Eulerian criteria, is shown to be the ability of the method to capture clear boundaries between qualitatively different regions of the flow. This provides a way to measure the size of the coherent structures that is not dependent on arbitrarily applied thresholds.

It was noted that using both the positive-time and the negative-time LCS, as well as applying conditions of hyperbolicity, are important to provide a full picture of the vortex boundaries and dynamics. Additional information about the dynamical systems structure becomes evident, such as the loss of hyperbolicity along the LCS upstream of the isolated hairpin head indicating the birth of a secondary hairpin vortex. Also, the presence of time-dependent saddle points on opposite sides of vortex cores was shown to be an LCS structure that appears in the analysis of both wall-bounded turbulent structures (§ 2.3 and 2.4) and the pitching panel wake (§ 5.2.2). This result suggests that it may be possible to track individual coherent structures in more complicated flow fields using tracking algorithms that follow these time-dependent saddles.

The capability to follow the quantitatively defined saddle points is particularly useful in the analysis of the low-aspect-ratio pitching panel. In § 5.2.1, Eulerian results of the wake of the trapezoidal panels indicates a transition of the wake structure from 2S to 2P as Strouhal number increases. This transition is represented by a lateral spreading of the spanwise vorticity and a loss of spanwise vorticity magnitude. In the LCS analysis, it is clear that this transition of the wake coincides in space and time with a bifurcation of the dynamical systems structure as two nearby time-dependent saddle points seemingly merge. A more exact description of this event and the associated flow physics will be facilitated by the solution of the three-dimensional velocity fields, additional downstream data which will improve the calculation of the pLCS, and implementation of methods to extract LCS curves from DLE fields.

It is important to point out that while LCS has distinct advantages over commonly used Eulerian criteria, there are a number of factors that suggest that LCS is best used in concert with other methods of analysis. DLE calculations involve a greater computational cost and require additional data upstream and downstream of the region of interest, as well as forward and backward in time. It would be sensible to focus the DLE calculation on those regions of the flow in which structures of interest are known to exist. Eulerian criteria, which are calculated using spatial derivatives of the velocity field, are quickly computed and can be used to guide the implementation of the LCS analysis. Furthermore, while the strength of the vortical structures may be inferred from the dynamics of the LCS, the magnitude of Eulerian criteria provide an immediate measure of the relative strength of vortices.

### 6.1.2 Three-dimensionality

The flow fields around simple, unsteady rigid propulsors are shown to be highly three-dimensional, and the three-dimensional effects can hurt performance. In § 4.1.3 measured quantities such as pressure, time-averaged thrust, and spanwise circulation of vortices in the wake of a rigid rectangular panel are shown to vary not only with Strouhal number, which takes unsteady and inertial effects into account, but also with pitching amplitude and aspect ratio. An additional nondimensional scaling based on the ratio of amplitude to span, which are parameters associated with the three-dimensionality of the flow, neatly collapses these quantities.

Visualizations of the wake using isocontours of spanwise vorticity show a spanwise compression of the wake as the vortical structures move downstream from the panel trailing edge. Also, within the wake, the transverse alignment of the vortices changes away from the midspan. Specifically, in the case of the trapezoidal panel pitching at  $St = 0.55$ , near the midspan vortices are aligned such that induced flow between consecutive vortex pairs is directed downstream, adding momentum to the wake and indicating production of thrust. Away from the midspan, vortices drift in the transverse direction. Close to the tips, this results in the induced velocity from vortex pairs directed upstream, indicating a momentum deficit and drag on the panel. These observations are consistent with those in § 4.1.3 where it was shown that at one Strouhal number, an increase in three-dimensionality caused by a decrease in aspect ratio decreases the thrust production.

### 6.1.3 Trapezoidal planform geometry

Rigid pitching panels of trapezoidal planform geometry are studied to explore the effects of the angled spanwise edge commonly seen in fish caudal fins used in thunniform or ostraciiform locomotion. In flow visualization results, coherent structure generation is observed along the swept edge. For large sweep angles or large freestream velocities (low  $St$ ), the swept edge structures generated in one half-cycle convect away from the edge and remain trapped on one side of the panel, eventually interacting with the spanwise vorticity from the trailing edge. An LCS analysis provided preliminary evidence to support to the existence of these “trapped” structures.

At higher Strouhal numbers or lower sweep angles, the edge distance of the structure in the subsequent half-cycle is relatively small, and the structure continually interacts with the swept edge and not retain coherence.

Because the trapped structures are rolling up along the swept edge, they contain components of both streamwise and spanwise vorticity. Transverse vorticity is assumed to be small for the low pitching amplitudes studied here. In figure 6.1, we provide a sketch of a possible benefit of the trapped structure phenomenon. Close to the spanwise tip of the panel, the trapped structure created in one half-cycle is expected to interact with the vorticity being shed by the trailing edge in the next half-cycle. If this is true, this is a like-sign interaction which may strengthen the spanwise vortical structures near the tip, and possibly buffer the wake from three-dimensional effects. Oshima & Natsume (1980), Freymuth (1988), and Anderson *et al.* (1998) observed this amalgamation of vorticity from both the leading and trailing edges of pitching and heaving airfoils, and proposed that it led to higher thrust production and efficiency. In the case of pitching swept edge panels, it could possibly explain why the spanwise compression of the wake observed in § 5.2.1 is exaggerated at  $St = 0.55$ , when trapping was not observed.

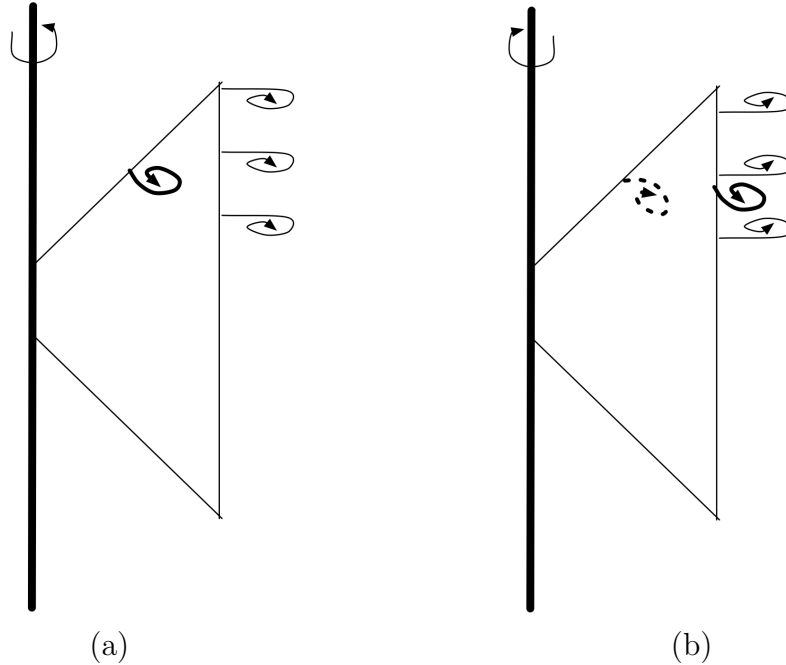


Figure 6.1: Proposed schematic of swept edge structure dynamics in the trapping situation. (a) As panel pitches into the page, a negative spanwise vortex is shed from the trailing edge, while a swept edge structure (bold) with components in the positive streamwise direction and positive spanwise direction. (b) In the next half-cycle as the panel pitches out of the page, the swept edge structure undergoes a like-sign interaction with the positive spanwise vortex being shed from the trailing edge. The swept edge structure forming on the far side of the panel is indicated by a dotted line.

Further work is needed to confirm the trapped structures' existence and significance, as described in the next section. The phenomenon could provide an explanation to why fish caudal fins have evolved to this shape, and an incentive to use similar shapes in engineering design. In addition, a trapezoidal planform geometry with a trailing edge notch, as shown in figure 6.2 may enhance this effect, and could be studied in future work as well.

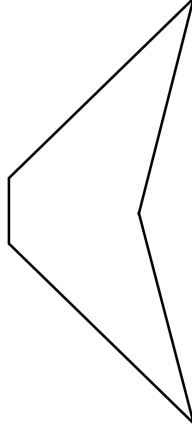


Figure 6.2: Schematic of trapezoidal panel geometry with trailing edge notch.

## 6.2 Future work

- **3D velocity** By using the principle of continuity in an incompressible flow ( $\nabla \cdot \mathbf{u} = 0$ ), we can solve for the spanwise derivative of the out of plane velocity,  $\frac{\partial w}{\partial z}$ . The spanwise extent of the DPIV data around the trapezoidal panel extends into the freestream, where an assumption of  $w = 0$  at the spanwise boundary is acceptable. Therefore, it should be possible to solve for the out of plane velocity by numerically integrating  $\frac{\partial w}{\partial z}$ . Once the fully three dimensional velocity field is acquired the LCS analysis will be recomputed to more precisely investigate the three-dimensional effect and the phenomenon of trapped swept edge structures. Alternatively, stereoscopic or tomographic PIV could be used in future experiments to measure all three components of velocity.
- **Downstream data** As mentioned in § 5.2.2 and § 6.1.1, plots of positive-time DLE do not yield sharp LCS ridges in the downstream half of the field, due to the lack of velocity information downstream of the DPIV data set. Unfortunately, this effect on the pLCS coincides with the dynamically significant saddle-point bifurcation at  $St = 0.55$ . An additional volume of data acquired downstream is desired in order to conduct a more complete LCS analysis of the transition/bifurcation, and could be acquired with the current experimental setup by positioning the pitching apparatus further upstream.

- **LCS extraction** LCS are plotted in this work as though regions of the DLE field which a relatively high threshold of DLE is exceeded. In Lekien *et al.* (2007), a method to mathematically extract these regions (“ridges”) is provided. Future work that looks to use LCS as a quantitative tool to identify and track coherent structures, will involve the implementation of this method to facilitate visualization and location of LCS in complicated fluid flows.
- **Performance measurements** Direct thrust and efficiency measurements are desired to directly associate wake characteristics with propulsive performance.
- **Future experiments** In the course of describing the full flow physics of these simple rigid propulsors, we are creating a framework in which to compare similar unsteady flow fields of increasing complexity. Relative benefits and disadvantages of different planform geometries (such as a notched trailing edge), large pitching amplitudes, flexibility, and multiple fin interactions may be assessed more readily. The effects of incoming flow fields, such as those from the wakes of other fish, can also be explored, and may lead to insights into phenomena such as fish schooling.
- **Design and control implications** A clear understanding of the three-dimensional effects in these flows could provide ways to improve design of both underwater vehicles and their controls systems. Body shapes and control inputs that affect the flow by mitigating three-dimensional effects may be especially useful.

# Bibliography

- ANDERSON, J. M., STREITLIEN, K., BARRETT, D. S. & TRIANTAFYLLOU, M. S. 1998 Oscillating foils of high propulsive efficiency. *J. Fluid Mech.* **360**, 41–72.
- ARDONCEAU, P. L. 1989 Unsteady pressure distribution over a pitching airfoil. *AIAA Journal* **27** (5), 660–662.
- BORAZJANI, I. & SOTIROPOULOS, F. 2008 Numerical investigation of the hydrodynamics of anguilliform swimming in the transitional and inertial flow regimes. *Journal of Experimental Biology* **212**, 576–592.
- BRUNTON, S. & ROWLEY, C. W. 2009 A comparison of methods for fast computation of time-varying FTLE fields. *pre-print*.
- BUCHHOLZ, J. H. J. 2006 The flowfield and performance of a low aspect ratio unsteady propulsor. PhD thesis, Princeton University.
- BUCHHOLZ, J. H. J. 2008–2009 Private Correspondence.
- BUCHHOLZ, J. H. J., JIMÉNEZ, J. M., ALLEN, J. J. & SMITS, A. J. 2003 Hydrodynamics of thrust production in a fish-like flapping membrane. In *13th International Symposium on Unmanned, Untethered Submersible Technology*. University of New Hampshire, Durham, NH.
- BUCHHOLZ, J. H. J. & SMITS, A. J. 2005 The wake of a low aspect ratio pitching plate. *Phys. Fluids* **17** (9).
- BUCHHOLZ, J. H. J. & SMITS, A. J. 2006 On the evolution of the wake structure produced by a low-aspect-ratio pitching panel. *J. Fluid Mech.* **546**, 433–443.
- BUCHHOLZ, J. H. J. & SMITS, A. J. 2008 The wake structure and thrust performance of a rigid low-aspect-ratio pitching panel. *Journal of Fluid Mechanics* **603**, 331–365.
- CHAKRABORTY, P., BALACHANDAR, S. & ADRIAN, R. J. 2005 On the relationships between local vortex identification schemes. *J. Fluid Mech.* (submitted).
- CHONG, M. S., PERRY, A. E. & CANTWELL, B. J. 1990 A general classification of three-dimensional flow fields. *Phys. Fluids A* **2** (5), 765–777.



- DONG, H., MITTAL, R., BOZHURTTAS, M. & NAJJAR, F. 2005 Wake structure and performance of finite aspect-ratio flapping foils. *AIAA Journal* (2005-0081).
- DONG, H., MITTAL, R. & NAJJAR, F. M. 2006 Wake topology and hydrodynamic performance of low-aspect-ratio flapping foils. *J. Fluid Mech.* **566**, 309–343.
- DRUCKER, E. G. & LAUDER, G. V. 1999 Locomotor forces on a swimming fish: three-dimensional vortex wake dynamics quantified using digital particle image velocimetry. *Journal of Experimental Biology* **202**, 2393–2412.
- VON ELLENRIEDER, K. D., PARKER, K. & SORIA, J. 2003 Flow structures behind a heaving and pitching finite-span wing. *Journal of Fluid Mechanics* **490**, 129–138.
- FALCO, R. E. 1977 Coherent motions in the outer region of turbulent boundary layers. *The Physics of Fluids* **20** (10), S124–S132.
- FREYMUTH, P. 1988 Propulsive vortical signature of plunging and pitching airfoils. *AIAA Journal* **27** (9), 1200–1205.
- GREEN, M. A., ROWLEY, C. W. & HALLER, G. 2007 Detection of lagrangian coherent structures in three-dimensional turbulence. *J. Fluid Mech.* **572**, 111–120.
- GREEN, M. A. & SMITS, A. J. 2008 Effects of three-dimensionality on thrust production by a pitching panel. *J. Fluid Mech.* **615** (211-220).
- GUGLIELMINI, L. 2004 Modeling of thrust generating foils. PhD thesis, University of Genoa.
- HALLER, G. 2001 Distinguished material surfaces and coherent structures in 3d fluid flows. *Physica D* **149**, 248–277.
- HALLER, G. 2002 Lagrangian coherent structures from approximate velocity data. *Physics of Fluids* **14** (6), 1851–1861.
- HALLER, G. 2005 An objective definition of a vortex. *J. Fluid Mech.* **525**, 1–26.
- HALLER, G. & YUAN, G. 2000 Lagrangian coherent structures and mixing in two dimensional turbulence. *Physica D* **147**, 352–370.
- HEAD, M. & BANDYOPADHYAY, P. 1981 New aspects of turbulent boundary-layer structure. *J. Fluid Mech.* **107**, 297–338.
- HILAIRE, A. S. & CARTA, F. 1983 Analysis of unswept and swept wing chordwise pressure data from an oscillating naca 0012 airfoil experiment. *NASA CR-3567*.
- HUANG, H., DABIRI, D. & GHARIB, M. 1997 On errors of digital particle image velocimetry. *Meas. Sci. Technol.* **8**, 1427–1440.
- HULTMARK, M., LEFTWICH, M. & SMITS, A. J. 2007 Flowfield measurements in the wake of a robotic lamprey. *Experiments in Fluids* **43**, 683–690.

- HUNT, J. C. R., WRAY, A. A. & MOIN, P. 1988 Eddies, stream, and convergence zones in turbulent flows. *Center for Turbulence Research Report CTR-S88*.
- JEONG, J. & HUSSEIN, F. 1995 On the identification of a vortex. *J. Fluid Mech.* **285**, 69–94.
- JIMÉNEZ, J. M. 2002 Low reynolds number studies in the wake of a submarine model using particle image velocimetry. Master’s thesis, Princeton University.
- JIMÉNEZ, J. M., BUCHHOLZ, J. H. J., STAPLES, A. E., ALLEN, J. J. & SMITS, A. J. 2003 Flapping membranes for thrust production. In *IUTAM Symposium on Integrated Modeling of Fully Coupled Fluid-Structure Interactions Using Analysis, Computations, and Experiments* (ed. H. Benaroya & T. Wei), pp. 115–124. Rutgers University, New Brunswick, NJ: Kluwer Academic Publishers.
- KIM, J., MOIN, P. & MOSER, R. 1987 Turbulence statistics in fully developed channel flow at low reynolds number. *J. Fluid Mech.* **177**, 133–166.
- KOH, T. Y. & LEGRAS, B. 2002 Hyperbolic lines and the stratospheric polar vortex. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **12** (2), 382–394.
- KOOCHESFAHANI, M. M. 1989 Vortical patterns in the wake of an oscillating airfoil. *AIAA Journal* **27** (9), 1200–1205.
- LAPEYRE, G. 2002 Characterization of finite-time lyapunov exponents and vectors in two-dimensional turbulence. *Chaos* **12** (3), 688–698.
- LAUDER, G. V. 2000 Function of the caudal fin during locomotion in fishes: kinematics, flow visualization, and evolutionary patterns. *Amer. Zool.* **40**, 101–122.
- LEKIEN, F. & LEONARD, N. 2004 Dynamically consistent lagrangian coherent structures. In *American Inst. of Physics: 8th Experimental Chaos Conference, CP 742*, pp. 132–139.
- LEKIEN, F., SHADDEN, S. C. & MARSDEN, J. E. 2007 Lagrangian coherent structures in  $n$ -dimensional systems. *Journal of Mathematical Physics* **48**.
- LINDSEY, C. C. 1978 Form, function and locomotory habits in fish. In *Fish Physiology Vol. VII Locomotion* (ed. W. S. Hoar & D. J. Randall), pp. 1–100. Academic Press.
- LIPINSKI, D., CARDWELL, B. & MOHSENI, K. 2008 A lagrangian analysis of a two-dimensional airfoil with vortex shedding. *J. Phys. A: Math. Theor.* **41**, 1–22.
- MATHUR, M., HALLER, G., PEACOCK, T., RUPPERT-FELSOT, J. E. & SWINNEY, H. L. 2006 Uncovering the lagrangian skeleton of turbulence. Preprint.
- NEWTON, P. K. 2001 *The N-Vortex problem: analytical techniques, Applied Mathematical Sciences*, vol. 145. Springer.

- OSHIMA, Y. & NATSUME, A. 1980 Flow field around and oscillating airfoil. In *Flow visualization II. Proceedings of the Second International Symposium on Flow Visualization* (ed. W. Merzkirch). Bochum, W. Germany: Hemisphere Publishing Co.
- PARKER, K., VON ELLENRIEDER, K. D. & SORIA, J. 2007 Morphology of the forced oscillatory flow past a finite-span wing at low reynolds number. *J. Fluid Mech.* **571**, 327–357.
- SFAKIOTAKIS, M., LANE, D. M. & DAVIES, J. B. C. 1999 Review of fish swimming modes for aquatic locomotion. *IEEE Journal of Oceanic Engineering* **24** (2), 237–252.
- SHADDEN, S., DABIRI, J. & MARSDEN, J. 2006 Lagrangian analysis of fluid transport in empirical vortex ring flows. *Physics of Fluids* **18**, 047105–1–047105–11.
- SHADDEN, S., LEKIEN, F. & MARSDEN, J. 2005 Definition and properties of lagrangian coherent structures from finite-time lyapunov exponents in two-dimensional aperiodic flows. *Physica D* **212**, 271–304.
- SHADDEN, S. C., KATIJA, K., ROSENFELD, M., MARSDEN, J. E. & DABIRI, J. O. 2007 Transport and stirring induced by vortex formation. *J. Fluid Mech.* **593**, 315–331.
- SHADDEN, S. C. & TAYLOR, C. A. 2008 Characterization of coherent structures in the cardiovascular system. *Annals of Biomedical Engineering* **36** (7), 1152–1162.
- SINGH, N., AIKAT, S. & BASU, B. C. 1989 Potential flow calculation for three-dimensional wings and wing-body combination in oscillatory motion. *AIAA Journal* **27** (12), 1665–1666.
- SMITH, C. & WALKER, J. 1991 On the dynamics of near-wall turbulence. In *Turbulent Flow Structure Near Walls* (ed. J. Walker). The Royal Society, First published in *Phil. Trans. R. Soc. London A*. 336, 1991.
- THEODORSEN, T. 1955 The structure of turbulence. In *50 Jahre Grenzschichtforschung* (ed. H. Görtler & W. Tollmien). Friedr. Vieweg and Sohn.
- TRIANTAFYLLOU, G., TRIANTAFYLLOU, M. & GROSENBAUGH, M. 1993 Optimal thrust development in oscillating foils with application to fish propulsion. *Journal of Fluids and Structures* .
- TYTELL, E. D. 2006 Median fin function in bluegill sunfish *Lepomis macrochirus*: streamwise vortex structure during steady swimming. *Journal of Experimental Biology* **209**, 1516–1534.
- TYTELL, E. D. & LAUDER, G. V. 2004 The hydrodynamics of eel swimming i. wake structure. *The Journal of Experimental Biology* **207**, 1825–1841.

- USHERWOOD, J., HEDRICK, T., MCGOWAN, C. & BIEWENER, A. 2005 Dynamic pressure maps for wings and tails of pigeons in slow, flapping flight, and their energetic implications. *Journal of Experimental Biology* .
- VOTH, G. A., HALLER, G. & GOLLUB, J. P. 2002 Experimental measurements of stretching fields in fluid mixing. *Phys. Rev. Lett.* **88** (254501).
- WILLIAMSON, C. H. K. & ROSHKO, A. 1988 Vortex formation in the wake of an oscillating cylinder. *Journal of Fluids and Structure* **2**, 355–381.
- ZHOU, J., ADRIAN, R. J., BALACHANDAR, S. & KENDALL, T. M. 1999 Mechanisms for generating coherent packets of hairpin vortices in channel flow. *J. Fluid Mech.* **387**, 353–396.
- ZHU, Q. & SHOELE, K. 2008 Propulsion performance of a skeleton-strengthened fin. *Journal of Experimental Biology* **211**, 2087–2100.
- ZHU, Q., WOLFGANG, M. J., YUE, D. K. P. & TRIANTAFYLLOU, M. S. 2002 Three-dimensional flow structures and vorticity control in fish-like swimming. *J. Fluid Mech.* **468**, 1–28.

# Appendix A

## Accelerated DLE calculation

The periodicity of the DPIV velocity fields was exploited to reduce the computation time of the DLE calculations. Typically, DLE fields are computed by initializing particle trajectories on a user defined grid. The trajectories are then advected in time using a Runge-Kutta fourth-order integration scheme (RK4), which numerically solves the ordinary differential equation,

$$\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}(t), t), \quad (\text{A.1})$$

where  $\mathbf{u}$  is the Eulerian velocity field obtained either numerically, as in § 2.3, or experimentally, as in § 5.2.2. The stability of the RK4 scheme was established by decreasing step size until the results converged.

Integrated trajectories define a flow map, which takes a location  $\mathbf{x}$  at time  $t$  and maps its location after an integration time  $T$ ,

$$\Phi_t^{t+T} : \mathbf{x}(t) \mapsto \mathbf{x}(t+T) \quad (\text{A.2})$$

The flow map can also be written as a composition of  $k$  smaller consecutive flow maps of uniform integration time  $T/k$ ,

$$\Phi_t^{t+T} = \Phi_{t+\frac{(k-1)T}{k}}^{t+T} \circ \Phi_{t+\frac{(k-1)T}{k}}^{t+\frac{(k-2)T}{k}} \circ \dots \circ \Phi_{t+\frac{T}{k}}^{t+\frac{2T}{k}} \circ \Phi_t^{t+\frac{T}{k}}. \quad (\text{A.3})$$

The method to speed up the DLE calculation was taken from Brunton & Rowley (2009). Using the twenty-five phase-averaged DPIV velocity data sets that span the pitching period, we compute a series of twenty-five flow maps ( $\Phi_{t_i}^{t_{i+1}}$ ,  $i = 0, 1, \dots, 24$ ) on the uniform grids on which we are computing DLE. Each flow map is defined at a time  $t_i$  that coincides with one of the twenty-five phase-averaged data sets. Therefore, the integration time of each flow map,  $t_{i+1} - t_i = 1/(25f)$ , where  $1/f$  is the pitching period. For the DLE field at  $t = t_0$ , we integrate the field of trajectories by,

$$\Phi_{t_0}^{2/f} = \Phi_{t_{23}}^{t_{24}} \circ \Phi_{t_{22}}^{t_{23}} \dots \Phi_{t_0}^{t_1} \circ \Phi_{t_{24}}^{t_0} \dots \Phi_{t_1}^{t_2} \circ \Phi_{t_0}^{t_1} \quad (\text{A.4})$$

Consider a trajectory initialized at point  $(x_0, y_0, z_0)$  at time  $t_0$ . Its location in space at time  $t_1$  is given by  $(x_1, y_1, z_1) = \Phi_{t_0}^{t_1}(x_0, y_0, z_0)$ . To calculate its location at time  $t_2$ ,

we use the flow map  $\Phi_{t_1}^{t_2}$ . We calculate all the intermediate flow maps ahead of time, so we already have  $\Phi_{t_1}^{t_2}$  defined on the grid on which DLE is being computed. Using the eight nearest neighbors on that grid to the point  $(x_1, y_1, z_1)$ , we use a three-dimensional linear interpolation scheme to compute  $\Phi_{t_1}^{t_2}(x_1, y_1, z_1)$ . This process is repeated until the desired integration time is reached.

For the DLE results reported in § 5.2.2, 2000 integration steps were used for an integration time of two pitching periods ( $2/f$ ). To implement the new algorithm, an initial computational investments of 1000 integration steps is necessary, but each subsequent DLE calculation involves of only 50 interpolation steps. The interpolation schemes consists of fewer computations than the integration schemes, so the new algorithm offers at least a 40 fold computational time savings after the initial investment.

One drawback to the new method is an increase in error. The error of the interpolation is on the order of  $(\Delta x_{DLE})^2$ . However, it should be noted for the results reported in this work, the DLE spatial resolution is greater than that of the velocity data, such that  $\Delta x_{DLE} = \Delta x_{DPIV}/6$ .

# Appendix B

## DLE program

### B.1 Fortran90 code to compute DLE

#### B.1.1 Main program to compute DLE in turbulent channel

```
program dleprog
  use grid
  use intsubs
  use trajsubs
  use inout
  implicit none
  integer :: n,m,i,j,k,t,tjump
  real, dimension(:,:,:,:), allocatable :: vel, traj
  real, dimension(:,:,:), allocatable :: dle

  ! Read input variables
  call initialize
  ! Allocate data velocity variables
  call velsmem
  ! Set up data and DLE grids
  call setup_grid

  allocate(dle(ox,oy,oz))
  allocate(vel(ox,oy,oz,3))
  allocate(traj(ox,oy,oz,3))

  do tstart = 0,tend

    tjump = 0

    ! Initialize DLE variables on the DLE grid
    call setup_traj(traj)

    forall(i=1:ox, j=1:oy, k=1:oz)
```

```

        dle(i,j,k) = 0.
    end forall

    ! Initialize two sets of data velocity arrays between
    ! which to interpolate
    call readdata(nstart+(tjump*irestart))

    forall(i=1:nx,j=1:ny,k=1:nz)
        u1(i,j,k) = u(i,j,k)
        v1(i,j,k) = v(i,j,k)
        w1(i,j,k) = w(i,j,k)
    end forall

    call readdata(nstart+((tjump+(dir*1))*irestart))

    ! Start DLE integration
    do t=0,int((inttime/(delta)))

        print *, t, int((inttime/(delta))), abs((dir*t)*delta), &
            &abs(irestart*(tjump+(dir*1))*delta)

        ! If current timestep is no longer between two velocity arrays,
        ! must update arrays
        if(abs((dir*t)*delta).gt.abs(irestart*(tjump+(dir*1))*delta))then

            tjump = tjump+(dir*1)

            forall(i=1:nx,j=1:ny,k=1:nz)
                u1(i,j,k) = u(i,j,k)
                v1(i,j,k) = v(i,j,k)
                w1(i,j,k) = w(i,j,k)
            end forall

            call readdata(nstart+((tjump+(dir*1))*irestart))

        endif

        ! Interpolate in time to a velocity grid
        call velst(dir*t, tjump)
        ! Interpolate velocity in space from data grid to current location
        ! of DLE trajectories
        call step_vels_dis(dir*t,vel,traj)
        ! Use updated velocity on at DLE points to step forward in time
        call step_traj_dis(dir*t,tjump,vel,traj)

        ! Output routines

```



```

        if(t.eq.0) then
            call writemesh
        elseif(mod(t,iplt).eq.0) then
            dle = dle_calc(t, traj)
            call writedle(dle, vel,t)
        end if
    end do

    ! If cinematic loop being used
    nstart = nstart + int(tinc/delt)
    if(mod(nstart,2).gt.0) nstart = nstart+1

end do

print *, 'program finished!'

deallocate(vel,traj,dle,u,v,w,u1,v1,w1,velt,x,y,z,x1,y1,z1)

call cleanup_grid

end program dleprog

```

## B.1.2 Supporting modules

```

module grid
    real, dimension(:), allocatable :: x,y,z,x1,y1,z1
    integer :: ox, oy, oz
    integer :: oxm, oym, ozm
    integer :: oxs=1,oys=1,ozs=1
    integer :: nx, ny, nz, nxa, nya, nza
    integer :: nstart,irestart, tstart, tend, dir
    integer :: wherecut
    real :: xrat, yrat, zrat
    real :: tinc, delta, inttime, pi
    real :: xscale, yscale, zscale
    integer :: xstart, ystart, zstart
    character(64) :: outname
    integer :: re,nsteps,iplt
    real :: lx,lz,dex,dely, delt
    character(125) :: idir, odir

    namelist /inputdata/ nx, ny, nz, lx, lz, re, delt, nstart, nsteps, &
        irestart, iplt, idir, odir, &
        tend, tinc, delta, inttime, outname, xscale, yscale, zscale, &
        oxm, oym, ozm, wherecut, xrat, yrat, zrat, xstart, ystart, zstart, &
        dir

```

contains

```
subroutine setup_grid
  implicit none
  integer :: j, status
  character(1) :: hi
  real :: dx, dy, dz
  real :: dx1, dy1, dz1
  character (125) fname
  integer*4 :: hi1, hi2, hi3
  real*4, dimension(nx,0:ny-1,nz) :: xtemp, ytemp, ztemp
  integer, parameter :: fp=67
  integer :: i,k

  allocate(x(nx),y(ny),z(nz))
  allocate(x1(ox),y1(oy),z1(oz))

  pi = 4*atan(1.0)

  ! velocity grid -- can get from dataset

  fname=trim(idir)//'/grid.dat'
  print *,'= Reading mesh file: '// trim(fname)
  open(unit=fp, file=trim(fname), status="old", form="unformatted", &
        iostat=status)
  ! print *, 'file opened'
  if (status .ne. 0) then
    print *,"Error: could not open "//trim(fname)//" for reading.", status
    stop
  end if

  read(fp) hi, hi2, hi3
  read(fp) (((xtemp(i,j,k),i=1,nx),j=0,ny-1),k=1,nz), &
           (((ytemp(i,j,k),i=1,nx),j=0,ny-1),k=1,nz), &
           (((ztemp(i,j,k),i=1,nx),j=0,ny-1),k=1,nz)
  close(fp)

  do j=1,nx
    x(j) = real(xtemp(j,1,1), 8)
  end do
  do j=1,ny
    y(j) = real(ytemp(1,j-1,1), 8)
  end do
  do j=1,nz
    z(j) = real(ztemp(1,1,j), 8)
  end do
```

```

dx1=abs((lx*2*pi)/xrat/((float(nx)*xscale)))
dy1=abs((ly*2*pi)/yrat/((float(ny)*yscale)))
dz1=abs((lz*2*pi)/zrat/((float(nz)*zscale)))

! trajectory grid -- base on factor of resolution,
! startpoints, and fraction of domain

if(x(nx).gt.x(1))then
  do j=1,ox
    x1(j) = x(1) + ((j-1)*dx1) + x(xstart)
  end do
else
  do j=1,ox
    x1(j) = x(nx) + ((j-1)*dx1) + x(xstart)
  end do
end if

if(y(ny).gt.y(1)) then
  do j=1,oy
    y1(j) = y(1) + ((j-1)*dy1) + y(ystart)
  end do
else
  do j=1,oy
    y1(j) = y(ny) + ((j-1)*dy1) + y(ystart)
  end do
end if

if(z(nz).gt.z(1)) then
  do j=1,oz
    z1(j) = z(1) + ((j-1)*dz1) + z(zstart)
  end do
else
  do j=1,oz
    z1(j) = z(nz) + ((j-1)*dz1) + z(zstart)
  end do
end if

! 2D cuts
if(oxm.eq.1.and.ox.eq.5)then
  do j=1,ox
    x1(j) = x(wherecut) + (dx1*(j-3))
  end do
  print *, 'constant streamwise plane centered at x of ', x1(3)
  print *, 'requested plane at i=', wherecut, ', where x=', x(wherecut)
elseif(oxm.eq.1.and.wherecut.eq.nx)then
  do j=1,ox
    x1(j) = x(wherecut) + (dx1*(j-3))
  end do

```

```

        end do
        print *, 'constant streamwise plane centered at x of ', x1(3)
        print *, 'requested plane at i=', wherecut, ', where x=', x(wherecut)
elseif(oxm.eq.1.and.wherecut.eq.1)then
        do j=1,ox
                x1(j) = x(wherecut) + (dx1*(j-1))
        end do
        print *, 'constant streamwise plane centered at x of ', x1(1)
        print *, 'requested plane at i=', wherecut, ', where x=', x(wherecut)
endif

if(oym.eq.1.and.oy.eq.5)then
        do j=1,oy
                y1(j) = y(wherecut) + (dy1*(j-3))
        end do
        print *, 'constant spanwise plane centered at y of ', y1(3)
        print *, 'requested plane at j=', wherecut, ', where y=', y(wherecut)
elseif(oym.eq.1.and.wherecut.eq.ny)then
        do j=1,oy
                y1(j) = y(wherecut) + (dy1*(j-3))
        end do
        print *, 'constant spanwise plane centered at y of ', y1(3)
        print *, 'requested plane at j=', wherecut, ', where y=', y(wherecut)
elseif(oym.eq.1.and.wherecut.eq.1)then
        do j=1,oy
                y1(j) = y(wherecut) + (dy1*(j-1))
        end do
        print *, 'constant streamwise plane centered at y of ', y1(1)
        print *, 'requested plane at j=', wherecut, ', where y=', y(wherecut)
endif

if(ozm.eq.1.and.oz.eq.5)then
        do j=1,oz
                z1(j) = z(wherecut) + (dz1*(j-3))
        end do
        print *, 'constant spanwise plane centered at z of ', z1(3)
        print *, 'requested plane at k=', wherecut, ', where z=', z(wherecut)
elseif(ozm.eq.1.and.wherecut.eq.nz)then
        do j=1,oz
                z1(j) = z(wherecut) + (dz1*(j-3))
        end do
        print *, 'constant spanwise plane centered at z of ', z1(3)
        print *, 'requested plane at k=', wherecut, ', where z=', z(wherecut)
elseif(ozm.eq.1.and.wherecut.eq.1)then
        do j=1,oz
                z1(j) = z(wherecut) + (dz1*(j-1))
        end do

```

```

        print *, 'constant streamwise plane centered at z of ', z1(1)
        print *, 'requested plane at k=', wherecut, ', where z=', z(wherecut)
    endif

end subroutine setup_grid

subroutine initialize
    implicit none
    integer :: status
    integer :: fp=310

    open(unit=fp, file="channel.inp", status="old", iostat=status)

    print *, "Reading channel.inp..."
    read(unit=fp, nml=inputdata)

    close(fp)

    ! Set ox, oy, oz in the cases that there is a 2D cut
    if(oxm.eq.1)then
        oxs = int(wherecut*int(xscale*nx))
        oy = int(yscale*ny)
        oz = int(zscale*nz)
        oym=oy
        ozm=oz

        if(oxs.eq.1 .or. oxs.eq.xscale*nx)then
            ox=3
        else
            ox=5
        endif

    elseif(oym.eq.1)then
        oys = int(wherecut*int(yscale*ny))
        ox = int(xscale*nx)
        oz = int(zscale*nz)
        oxm=ox
        ozm=oz

        if(oys.eq.1 .or. oys.eq.yscale*ny)then
            oy=3
        else
            oy=5
        endif

    elseif(ozm.eq.1)then

```

```

        ozs = int(wherect*int(zscale*nz))
        ox = int(xscale*nx)
        oy = int(yscale*ny)
        oxm=ox
        oym=oy

        if(ozs.eq.1 .or. ozs.eq.zscale*nz)then
            oz=3
        else
            oz=5
        endif

    else

        ox = int(xscale*nx)
        oy = int(yscale*ny)
        oz = int(zscale*nz)
        oxm = ox
        oym = oy
        ozm = oz

    endif

    print *, "nx = ", nx, "ox = ", ox
    print *, "ny = ", ny, "oy = ", oy
    print *, "nz = ", nz, "oz = ", oz

end subroutine initialize

subroutine cleanup_grid
    implicit none
    deallocate(x,y,z)
end subroutine cleanup_grid

end module grid

module intsubs
    use grid
    use inout
    implicit none
    real, dimension(:,:,:), allocatable :: velt
contains

    subroutine velsmem
        implicit none

```

```

allocate(u(nx,ny,nz), v(ny,ny,nz), w(nx,ny,nz), u1(nx,ny,nz), &
        v1(nx,ny,nz), w1(nx,ny,nz), velt(nx,ny,nz,3))

end subroutine velsmem

subroutine step_vels_dis(t,vel,traj)
  implicit none
  integer, intent(in) :: t
  real, dimension(ox,oy,oz,3), intent(inout) :: vel
  real, dimension(ox,oy,oz,3), intent(in) :: traj
  integer, dimension(3) :: spos
  integer :: i,j,k,m

  do i=1,ox
    do j=1,oy
      do k=1,oz

        spos = getbound(traj(i,j,k,:))

        vel(i,j,k,1) = interp3d(spos, traj(i,j,k,:), getcube(spos, velt(:,:,:,1)))
        vel(i,j,k,2) = interp3d(spos, traj(i,j,k,:), getcube(spos, velt(:,:,:,2)))
        vel(i,j,k,3) = interp3d(spos, traj(i,j,k,:), getcube(spos, velt(:,:,:,3)))

      end do
    end do
  end do

end subroutine step_vels_dis

subroutine step_traj_dis(t,tjump,vel,traj)
  implicit none
  integer, intent(in) :: t,tjump
  real, dimension(ox,oy,oz,3), intent(in) :: vel
  real, dimension(ox,oy,oz,3), intent(inout) :: traj
  real, dimension(nx,ny,nz,3) :: midv,endv
  real, dimension(3) :: loc
  integer, dimension(3) :: spos
  integer :: i,j,k,m
  real :: k1x,k1y,k1z,k2x,k2y,k2z,k3x,k3y,k3z,k4x,k4y,k4z

  ! Set up midpoint and endpoint velocity values needed for RK4 integration
  do i=1,nx
    do j=1,ny
      do k=1,nz

```

```

midv(i,j,k,1) = interp(tjump*irestart*delt,(tjump+1)*irestart*delt, &
& u1(i,j,k), u(i,j,k), (t+(dir*.5))*delta)
midv(i,j,k,2) = interp(tjump*irestart*delt,(tjump+1)*irestart*delt, &
& v1(i,j,k), v(i,j,k), (t+(dir*.5))*delta)
midv(i,j,k,3) = interp(tjump*irestart*delt,(tjump+1)*irestart*delt, &
& w1(i,j,k), w(i,j,k), (t+(dir*.5))*delta)

endv(i,j,k,1) = interp(tjump*irestart*delt,(tjump+1)*irestart*delt, &
& u1(i,j,k), u(i,j,k), (t+(dir*1))*delta)
endv(i,j,k,2) = interp(tjump*irestart*delt,(tjump+1)*irestart*delt, &
& v1(i,j,k), v(i,j,k), (t+(dir*1))*delta)
endv(i,j,k,3) = interp(tjump*irestart*delt,(tjump+1)*irestart*delt, &
& w1(i,j,k), w(i,j,k), (t+(dir*1))*delta)

end do
end do
end do

! RK 4
do i=1,ox
do j=1,oy
do k=1,oz

k1x = dir*vel(i,j,k,1)*delta
k1y = dir*vel(i,j,k,2)*delta
k1z = dir*vel(i,j,k,3)*delta

loc(1) = traj(i,j,k,1)+(k1x/2)
loc(2) = traj(i,j,k,2)+(k1y/2)
loc(3) = traj(i,j,k,3)+(k1z/2)

spos = getbound(loc)

k2x = dir*delta*interp3d(spos, loc, getcube(spos, midv(:,:,:,1)))
k2y = dir*delta*interp3d(spos, loc, getcube(spos, midv(:,:,:,2)))
k2z = dir*delta*interp3d(spos, loc, getcube(spos, midv(:,:,:,3)))

loc(1) = traj(i,j,k,1)+(k2x/2)
loc(2) = traj(i,j,k,2)+(k2y/2)
loc(3) = traj(i,j,k,3)+(k2z/2)

spos = getbound(loc)

k3x = dir*delta*interp3d(spos, loc, getcube(spos, midv(:,:,:,1)))
k3y = dir*delta*interp3d(spos, loc, getcube(spos, midv(:,:,:,2)))

```



```

        k3z = dir*delta*interp3d(spos, loc, getcube(spos, midv(:,:,:,3)))

        loc(1) = traj(i,j,k,1)+(k3x)
        loc(2) = traj(i,j,k,2)+(k3y)
        loc(3) = traj(i,j,k,3)+(k3z)

        spos = getbound(loc)

        k4x = dir*delta*interp3d(spos, loc, getcube(spos, endv(:,:,:,1)))
        k4y = dir*delta*interp3d(spos, loc, getcube(spos, endv(:,:,:,2)))
        k4z = dir*delta*interp3d(spos, loc, getcube(spos, endv(:,:,:,3)))

        traj(i,j,k,1) = traj(i,j,k,1) + k1x/6 + k2x/3 + k3x/3 + k4x/6
        traj(i,j,k,2) = traj(i,j,k,2) + k1y/6 + k2y/3 + k3y/3 + k4y/6
        traj(i,j,k,3) = traj(i,j,k,3) + k1z/6 + k2z/3 + k3z/3 + k4z/6

    end do
end do
end do

end subroutine step_traj_dis

function interp(a1,a2,b1,b2,am) result(bm)
    implicit none
    real, intent(in) :: a1,a2,am
    real, intent(in) :: b1,b2
    real :: bm

    bm = b2 + ((b1-b2)*(am-a2)/(a1-a2))

end function interp

function interp3d(spos,loc,cube3d) result(yea)
    implicit none
    integer, dimension(3), intent(in) :: spos
    real, dimension(3), intent(in) :: loc
    real, dimension(2,2,2), intent(in) :: cube3d
    real :: yea
    real :: xp, yp, zp
    real :: fx1y1,fx1y2,fx2y1,fx2y2,fx1,fx2

    xp = loc(1)
    yp = loc(2)
    zp = loc(3)

    ! If the current location is out of the domain, brings it back
    ! into the domain using the spanwise and streamwise

```

```

! periodicity.  If it has gone through the walls, just
! puts the points on the wall exactly.
if(x(nx).gt.x(1)) then
15   if(xp.gt.x(nx)) then
       xp = xp - abs(x(nx)-x(1))
       goto 15
   endif
25   if(xp.lt.x(1)) then
       xp = xp + abs(x(nx)-x(1))
       goto 25
   endif
else
75   if(xp.gt.x(1)) then
       xp = xp - abs(x(nx)-x(1))
       goto 75
   endif
85   if(xp.lt.x(nx)) then
       xp = xp + abs(x(nx)-x(1))
       goto 85
   endif
endif

if(y(ny).gt.y(1)) then
    if(yp.ge.y(ny)) then
        yp = y(ny)
    endif
    if(yp.le.y(1)) then
        yp = y(1)
    endif
else
    if(yp.ge.y(1)) then
        yp = y(1)
    endif
    if(yp.le.y(ny)) then
        yp = y(ny)
    endif
endif

if(z(nz).gt.z(1)) then
55   if(zp.gt.z(nz)) then
       zp = zp - abs(z(nz)-z(1))
       goto 55
   endif
65   if(zp.lt.z(1)) then
       zp = zp + abs(z(nz)-z(1))
       goto 65

```

```

        endif
    else
115      if(zp.gt.z(1)) then
            zp = zp - abs(z(nz)-z(1))
            goto 115
        endif
125      if(zp.lt.z(nz)) then
            zp = zp + abs(z(nz)-z(1))
            goto 125
        endif
    endif
endif

!interpolate along corner lines of 3d cube

fx1y1 = interp(z(spos(3)), z(spos(3)+1), cube3d(1,1,1), cube3d(1,1,2), zp)
fx1y2 = interp(z(spos(3)), z(spos(3)+1), cube3d(1,2,1), cube3d(1,2,2), zp)
fx2y1 = interp(z(spos(3)), z(spos(3)+1), cube3d(2,1,1), cube3d(2,1,2), zp)
fx2y2 = interp(z(spos(3)), z(spos(3)+1), cube3d(2,2,1), cube3d(2,2,2), zp)

!interpolate the x = x1 and the x = x2 faces

fx1 = interp(y(spos(2)), y(spos(2)+1), fx1y1, fx1y2, yp)

fx2 = interp(y(spos(2)), y(spos(2)+1), fx2y1, fx2y2, yp)

!interpolate between the two faces to get the final data point

yea = interp(x(spos(1)), x(spos(1)+1), fx1, fx2, xp)

end function interp3d

function getcube(spos,thing) result(cube)
    implicit none
    integer, dimension(3), intent(in) :: spos
    real, dimension(nx,ny,nz), intent(in) :: thing
    real, dimension(2,2,2) :: cube

    !obtain cube of 8 data points to pass to interp3d
    cube(1,1,1) = thing(spos(1),spos(2),spos(3))
    cube(1,2,1) = thing(spos(1),spos(2)+1,spos(3))
    cube(2,1,1) = thing(spos(1)+1,spos(2),spos(3))
    cube(2,2,1) = thing(spos(1)+1,spos(2)+1,spos(3))
    cube(1,1,2) = thing(spos(1),spos(2),spos(3)+1)
    cube(1,2,2) = thing(spos(1),spos(2)+1,spos(3)+1)
    cube(2,1,2) = thing(spos(1)+1,spos(2),spos(3)+1)

```

```

cube(2,2,2) = thing(spos(1)+1,spos(2)+1,spos(3)+1)

end function getcube

function getbound(loc) result(spos)
  implicit none
  real, dimension(3), intent(in) :: loc
  integer, dimension(3) :: spos
  real :: xp, yp, zp
  integer :: j

  spos(1) = nx-1
  spos(2) = ny-1
  spos(3) = nz-1

  xp = loc(1)
  yp = loc(2)
  zp = loc(3)

  ! Need to make sure that the points are still in the domain, otherwise
  ! take care of them (here, by imposing periodic conditions in x- and
  ! z- directions, and just stick to the wall in y

  if(x(nx).ge.x(1)) then
15    if(xp.gt.x(nx)) then
        xp = xp - abs(x(nx)-x(1))
        goto 15
      endif
25    if(xp.lt.x(1)) then
        xp = xp + abs(x(nx)-x(1))
        goto 25
      endif
  else
75    if(xp.gt.x(1)) then
        xp = xp - abs(x(nx)-x(1))
        goto 75
      endif
85    if(xp.lt.x(nx)) then
        xp = xp + abs(x(nx)-x(1))
        goto 85
      endif
  endif

  if(y(ny).gt.y(1)) then
    if(yp.ge.y(ny)) then

```

```

        yp = y(ny)
    endif
    if(yp.le.y(1)) then
        yp = y(1)
    endif
else
    if(yp.ge.y(1)) then
        yp = y(1)
    endif
    if(yp.le.y(ny)) then
        yp = y(ny)
    endif
endif

if(z(nz).gt.z(1)) then
55    if(zp.gt.z(nz)) then
        zp = zp - abs(z(nz)-z(1))
        goto 55
    endif
65    if(zp.lt.z(1)) then
        zp = zp + abs(z(nz)-z(1))
        goto 65
    endif
else
115    if(zp.gt.z(1)) then
        zp = zp - abs(z(nz)-z(1))
        goto 115
    endif
125    if(zp.lt.z(nz)) then
        zp = zp + abs(z(nz)-z(1))
        goto 125
    endif
endif

```

! Go through data arrays to find indices corresponding to a bounding  
! cube around current location of DLE trajectory

```

if(x(nx).gt.x(1))then
    do j=1,nx
        if(xp.lt.x(j)) then
            spos(1) = j-1
            exit
        end if
    end do
    if(spos(1).eq.0) spos(1)=1
else
    do j=1,nx

```

```

        if(xp.gt.x(j)) then
            spos(1) = j-1
            exit
        end if
    end do
    if(spos(1).eq.0) spos(1)=1
end if

if(y(ny).gt.y(1))then
    do j=1,ny
        if(yp.lt.y(j)) then
            spos(2) = j-1
            exit
        end if
    end do
    if(spos(2).eq.0) spos(2)=1
else
    do j=1,ny
        if(yp.gt.y(j)) then
            spos(2) = j-1
            exit
        end if
    end do
    if(spos(2).eq.0) spos(2)=1
end if

if(z(nz).gt.z(1))then
    do j=1,nz
        if(zp.lt.z(j)) then
            spos(3) = j-1
            exit
        end if
    end do
    if(spos(3).eq.0) spos(3)=1
else
    do j=1,nz
        if(zp.gt.z(j)) then
            spos(3) = j-1
            exit
        end if
    end do
    if(spos(3).eq.0) spos(3)=1
end if

end function getbound

```

```

subroutine velst(t,tjump)
  implicit none
  integer, intent(in) :: t,tjump
  integer :: i,j,k

  do i=1,nx
    do j=1,ny
      do k=1,nz

        velt(i,j,k,1) = interp(tjump*irestart*delt,(tjump+(dir*1)) &
          & *irestart*delt, u1(i,j,k), u(i,j,k), t*delta)
        velt(i,j,k,2) = interp(tjump*irestart*delt,(tjump+(dir*1)) &
          & *irestart*delt, v1(i,j,k), v(i,j,k), t*delta)
        velt(i,j,k,3) = interp(tjump*irestart*delt,(tjump+(dir*1)) &
          & *irestart*delt, w1(i,j,k), w(i,j,k), t*delta)

      end do
    end do
  end do

end subroutine velst

end module intsubs

module trajsubs
  use grid
  implicit none

contains

subroutine setup_traj(traj)
  implicit none
  real, dimension(ox,oy,oz,3), intent(inout) :: traj
  integer :: i,j,k

  do i=1,ox
    do j=1,oy
      do k=1,oz

        traj(i,j,k,1) = x1(i)
        traj(i,j,k,2) = y1(j)
        traj(i,j,k,3) = z1(k)

      end do
    end do
  end do
end do

```

```

end subroutine setup_traj

function dle_calc(num,traj) result(dle)
  implicit none
  integer :: num
  real, dimension(ox,oy,oz,3) :: traj
  real, dimension(ox,oy,oz) :: dle
  real, dimension(3) :: xa,xb,xc,ya,yb,yc,za,zb,zc, st9
  real :: xoa,xob,xoc,yoa,yob,yoc,zoa,zob,zoc
  integer :: i,j,k

  ! Setup finite difference scheme -- don't actually use xb, yb, zb
  ! anymore, since use a central different scheme.
  do i=1,ox
    do j=1,oy
      do k=1,oz

        if(i.eq.1)then
          xa = traj(i,j,k,:)
          xb = traj(i,j,k,:)
          xc = traj(i+1,j,k,:)
          xoa = x1(i)
          xob = x1(i)
          xoc = x1(i+1)
        elseif(i.eq.ox)then
          xa = traj(i-1,j,k,:)
          xb = traj(i,j,k,:)
          xc = traj(i,j,k,:)
          xoa = x1(i-1)
          xob = x1(i)
          xoc = x1(i)
        else
          xa = traj(i-1,j,k,:)
          xb = traj(i,j,k,:)
          xc = traj(i+1,j,k,:)
          xoa = x1(i-1)
          xob = x1(i)
          xoc = x1(i+1)
        end if

        if(j.eq.1)then
          ya = traj(i,j,k,:)
          yb = traj(i,j,k,:)
          yc = traj(i,j+1,k,:)

```



```

        yoa = y1(j)
        yob = y1(j)
        yoc = y1(j+1)
elseif(j.eq.oy)then
    ya = traj(i,j-1,k,:)
    yb = traj(i,j,k,:)
    yc = traj(i,j,k,:)
    yoa = y1(j-1)
    yob = y1(j)
    yoc = y1(j)
else
    ya = traj(i,j-1,k,:)
    yb = traj(i,j,k,:)
    yc = traj(i,j+1,k,:)
    yoa = y1(j-1)
    yob = y1(j)
    yoc = y1(j+1)
end if

if(k.eq.1)then
    za = traj(i,j,k,:)
    zb = traj(i,j,k,:)
    zc = traj(i,j,k+1,:)
    zoa = z1(k)
    zob = z1(k)
    zoc = z1(k+1)
elseif(k.eq.oz)then
    za = traj(i,j,k-1,:)
    zb = traj(i,j,k,:)
    zc = traj(i,j,k,:)
    zoa = z1(k-1)
    zob = z1(k)
    zoc = z1(k)
else
    za = traj(i,j,k-1,:)
    zb = traj(i,j,k,:)
    zc = traj(i,j,k+1,:)
    zoa = z1(k-1)
    zob = z1(k)
    zoc = z1(k+1)
end if

st9 = stretcher(xc, xa, yc, ya, zc, za, xoc, xoa, yoc, yoa, zoc, zoa)

dle(i,j,k) = Log(max(st9(1), st9(2), st9(3)))/(2*num*delta)

end do

```

```

        end do
    end do

end function dle_calc

function stretcher(xd1,xd2,yd1,yd2,zd1,zd2,x01,x02,y01,y02,z01,z02)result(eigs)
    implicit none
    real, dimension(3), intent(in) :: xd1,xd2,yd1,yd2,zd1,zd2
    real, intent(in) :: x01,x02,y01,y02,z01,z02
    real, dimension(3,3) :: dlemat, mult
    real, dimension(3) :: eigs
    real :: orgp, orgq, tilq, orgr, tilr, discr, theta

    ! Set up Cauchy-Green strain tensor
    dlemat(1,1) = (xd1(1)-xd2(1))/(x01-x02)
    dlemat(2,1) = (xd1(2)-xd2(2))/(x01-x02)
    dlemat(3,1) = (xd1(3)-xd2(3))/(x01-x02)

    dlemat(1,2) = (yd1(1)-yd2(1))/(y01-y02)
    dlemat(2,2) = (yd1(2)-yd2(2))/(y01-y02)
    dlemat(3,2) = (yd1(3)-yd2(3))/(y01-y02)

    dlemat(1,3) = (zd1(1)-zd2(1))/(z01-z02)
    dlemat(2,3) = (zd1(2)-zd2(2))/(z01-z02)
    dlemat(3,3) = (zd1(3)-zd2(3))/(z01-z02)

    dlemat = matmul(transpose(dlemat), dlemat)

    ! Since dlemat is symmetric, all eigenvalues will be real
    ! Use algorithm from Ellen Taylor to calculate eigenvalues.
    orgp = -( dlemat(1,1)+dlemat(2,2)+dlemat(3,3) )
    mult = matmul(dlemat,dlemat)
    orgq = 0.5*( orgp**2 - ( mult(1,1)+mult(2,2)+mult(3,3) ) )
    tilq = orgq - orgp**2/3.
    mult = matmul(mult,dlemat)
    orgr = ( -orgp**3 + 3.*orgp*orgq - ( mult(1,1)+mult(2,2)+mult(3,3) ) )/3.
    tilr = orgr + 2.*orgp**3/27. - orgp*orgq/3.

    theta = acos( (-.5*tilr) / ((-1./3.)*tilq)**(1.5) )
    eigs(1) = 2 * ((-1./3.)*tilq)**.5 * cos(theta/3.)
    eigs(2) = 2 * ((-1./3.)*tilq)**.5 * cos((theta+(2*pi))/3.)
    eigs(3) = 2 * ((-1./3.)*tilq)**.5 * cos((theta+(4*pi))/3.)

    eigs(1) = eigs(1) - (1./3.)*orgp
    eigs(2) = eigs(2) - (1./3.)*orgp
    eigs(3) = eigs(3) - (1./3.)*orgp

```

```

end function stretcher

end module trajsubs

module inout
  use grid
  implicit none
  real, dimension(:,:,:), allocatable :: u,v,w
  real, dimension(:,:,:), allocatable :: u1,v1,w1

contains

  subroutine writedle(dle,vel, t)
    ! write a tecplot file
    implicit none
    integer :: ierr, npts
    integer, parameter :: bufsize = 128, fp=10
    real, dimension(ox,oy,oz), intent(in) :: dle
    real, dimension(ox,oy,oz,3), intent(in) :: vel
    integer, intent(in) :: t
    character(40) :: fname
    integer :: i,j,k

    if(dir.lt.0) then
      write(fname,'(a,i6.6,a,i6.6,a)') trim(odir) // trim(outname) &
        & // 'neg_',nstart,"_",t,".dat"
    else
      write(fname,'(a,i6.6,a,i6.6,a)') trim(odir) // trim(outname) &
        & // 'pos_',nstart,"_",t,".dat"
    end if

    print *, 'Writing data file ', fname
    open(24,file=trim(fname),form="unformatted")
    write(24) ox,oy,oz,1
    write(24) (((real(dle(i,j,k),4),i=1,ox),j=1,oy),k=1,oz)
    close(24)

  end subroutine writedle

  subroutine writemesh

    integer :: i,j,k

    print *, 'Writing mesh file'

```

```

open(31,file=trim(odir)//'/grid.dat',form='UNFORMATTED')
write(31) ox,oy,oz
write(31)(((real(x1(i),4),i=1,ox),j=1,oy),k=1,oz),&
          & (((real(y1(j),4),i=1,ox),j=1,oy),k=1,oz),&
          & (((real(z1(k),4),i=1,ox),j=1,oy),k=1,oz)
close(31)

end subroutine writemesh

subroutine readdata(itime)
  implicit none
  integer, intent(in) :: itime
  character(150) :: fname
  integer, parameter :: fp=65
  integer :: status, i, j, k
  integer*4 :: hi, hi2, hi3, hi4
  real*4, dimension(nx, ny, nz) :: u4, v4, w4

  write(fname,'(a,i6.6,a)') trim(idir) // "/chan", itime, ".dat"
  print *,"Reading data file ",trim(fname)
  open(unit=fp, file=trim(fname), status="old", form="unformatted", &
        iostat=status)
  ! print *, 'file opened'
  if (status .ne. 0) then
    print *,"Error: could not open "//trim(fname)//" for reading.", status
    stop
  end if

  read(fp) hi, hi2, hi3, hi4
  read(fp) u4(:,:,:), v4(:,:,:), w4(:,:,:)
  close(fp)

  do i=1,nx
    do j=1,ny
      do k=1,nz
        u(i,j,k) = real(u4(i,j,k), 8)
        v(i,j,k) = real(v4(i,j,k), 8)
        w(i,j,k) = real(w4(i,j,k), 8)
      end do
    end do
  end do

end subroutine readdata

end module inout

```

### B.1.3 Main program to compute the intermediary flow maps for the panel data

```
program intprog
  use grid
  use inout
  use intsubs
  implicit none
  integer :: n,m,i,j,k,t,tjump,filenum
  real, dimension(:,:,:), allocatable :: vel, traj
  integer, dimension(:,:,:), allocatable :: timetraj

  ! Read input variables
  call initialize
  ! Allocate data velocity variables
  call velsmem
  ! Set up data and DLE grids
  call setup_grid

  allocate(timetraj(ox,oy,oz))
  allocate(vel(ox,oy,oz,3))
  allocate(traj(ox,oy,oz,3))

  call writemesh

  do tstart = 0,tend

    tjump = 0

    ! Initialize DLE variables on the DLE grid
    call setup_traj(traj)

    forall(i=1:ox, j=1:oy, k=1:oz)
      timetraj(i,j,k) = 0.
    end forall

    ! Initialize two sets of data velocity arrays between
    ! which to interpolate
    call readdata(nstart+(tjump*irestart))

    forall(i=1:nx,j=1:ny,k=1:nz)
      u1(i,j,k) = u(i,j,k)
      v1(i,j,k) = v(i,j,k)
      w1(i,j,k) = w(i,j,k)
    end forall

    filenum = nstart+((tjump+(dir*1))*irestart)
```

```

if(filenum.lt.0) filenum = filenum+25
call readdata(filenum)

! Start DLE integration
do t=0,int((inttime/(delta)))-1

    !print *, t, (dir*t)*delta, irestart*(tjump+(dir*1))*delt
    ! If current timestep is no longer between two velocity arrays,
    ! must update arrays
    if(abs((dir*t)*delta).gt.abs(irestart*(tjump+(dir*1))*delt))then

        tjump = tjump+(dir*1)

        forall(i=1:nx,j=1:ny,k=1:nz)
            u1(i,j,k) = u(i,j,k)
            v1(i,j,k) = v(i,j,k)
            w1(i,j,k) = w(i,j,k)
        end forall

        call readdata(nstart+((tjump+(dir*1))*irestart))

    endif

    ! Interpolate in time to a velocity grid
    call velst(dir*t, tjump)

    ! Interpolate velocity in space from data grid to current location
    ! of DLE trajectories
    call step_vels_dis(dir*t,timetraj,vel,traj)

    ! Use updated velocity on at DLE points to step forward in time
    call step_traj_dis(dir*t,tjump,vel,traj, timetraj)

end do

call writetraj(vel, timetraj, traj, t)

! If cinematic loop being used
nstart = nstart + ceiling(tinc/delt)
print *, 'nstart', nstart, int(tinc/delt)

end do

print *, 'program finished!'

deallocate(vel,traj,u,v,w,u1,v1,w1,velst,x,y,z,x1,y1,z1)

```

```

    call cleanup_grid

end program intprog

```

#### B.1.4 Main program to compute DLE using a composition of the intermediary flow maps

```

program dleprog
  use grid
  use inout
  use intsubs
  use trajsubs
  implicit none
  integer :: n,m,t
  real, dimension(:,:,:,:), allocatable :: traj
  real, dimension(:,:,:), allocatable :: dle
  real, dimension(:,:,:), allocatable :: timetraj

  ! Read input variables
  call initialize
  ! Allocate data velocity variables
  ! Set up data and DLE grids
  call setup_grid

  allocate(dle(ox,oy,oz))
  allocate(trajin(ox,oy,oz,3), traj(ox,oy,oz,3))
  allocate(timetraj(ox,oy,oz), timetrajin(ox,oy,oz))

  call writemesh

  do tstart = nstart,tend

    forall(i=1:ox, j=1:oy, k=1:oz)
      dle(i,j,k) = 0.
      timetraj(i,j,k) = 0.
      traj(i,j,k,1) = x1(i)
      traj(i,j,k,2) = y1(j)
      traj(i,j,k,3) = z1(k)
    end forall

    ! Start DLE integration
    do t=0,int(numper*inttime) -1

      call read_traj((dir*t)+nstart, trajin, timetrajin)

      call step_traj_dis(t, dir, traj, timetraj)
    end do
  end do
end program dleprog

```

```

        ! Output routines

        if(mod(t+1,numper).eq.0) then
            dle = dle_calc(t+1, timetraj, traj)
            call writedle(dle, timetraj, traj, t)
        end if
    end do

    ! If cinematic loop being used
    nstart = nstart + 1
    print *, 'nstart', nstart

end do

print *, 'program finished!'

deallocate(traj,trajin, timetraj, timetrajin, dle,x1,y1,z1)

end program dleprog

```

## B.2 Fortran90 code to compute hyperbolicity criteria

### B.2.1 Main program

```

program strain
    use grid
    use strainsubs
    use velgradsubs
    use inout
    implicit none
    real, dimension(:,:,:), allocatable :: dle
    real, dimension(:,:,:), allocatable :: dlegrad
    real, dimension(:,:,:), allocatable :: dlehess
    real, dimension(:,:,:), allocatable :: dlenorm
    real, dimension(:,:,:), allocatable :: gvel
    real, dimension(:,:,:), allocatable :: strainprod
    integer :: t, ios
    character(150) :: fname

    call setup_grid

    allocate(dle(ox,oy,oz))

```



```

allocate(dlegrad(ox,oy,oz,3))
allocate(dlehess(ox,oy,oz,3,3))
allocate(dlenorm(ox,oy,oz,3))
allocate(gvel(ox,oy,oz,3,3))
allocate(strainprod(ox,oy,oz))

do t=nstart,tend

    print *, 'time to read data'
    call readdle(dle, t)
    print *, 'time for Hessian calculation'
    call hesscalc(dle, dlehess)
    print *, 'time for norm calculation'
    call normcalc(dlehess, dlenorm)

    print *, 'hold breath, time for velocity gradient'
    call velst(gvel, t)
    print *, 'time for strain calculation'
    strainprod = strainprodcalc(dlenorm, gvel)

    call writestrain(strainprod, dle, t)

end do

deallocate(dle, dlegrad, dlehess, dlenorm, gvel, strainprod)

end program strain

```

## B.2.2 Supporting modules

```

module grid
    real, dimension(:), allocatable :: x,y,z,x1,y1,z1
    integer :: ox, oy, oz, i, j, k
    integer :: nx, ny, nz
    integer*4 :: ox4,oy4,oz4,nx4,ny4,nz4
    integer :: nstart, dlesteps, tend
    character(100) :: odir, chandir, dledir, dlename

    namelist /inputdata/ nstart, tend, odir, chandir, dledir, &
        dlename, dlesteps

contains

    subroutine setup_grid
        implicit none
        integer :: status
        character (125) fname, fname2
        real*4, dimension(:,:,:), allocatable :: xtemp, ytemp, ztemp

```

```

real*4, dimension(:,:,:), allocatable :: x1temp, y1temp, z1temp
integer, parameter :: fp=67, fp2=68
integer :: fp3=310

open(unit=fp3, file="strain.inp", status="old", iostat=status)

print *, "Reading strain.inp..."
read(unit=fp3, nml=inputdata)

close(fp3)

! velocity grid -- can get from dataset

fname=trim(chandir)//'/gridint.dat'
print *, '== Reading mesh file: '// trim(fname)
open(unit=fp, file=trim(fname), status="old", form="unformatted", &
      iostat=status)
! print *, 'file opened'
if (status .ne. 0) then
  print *, "Error: could not open "//trim(fname)//" for reading.", status
  stop
end if

read(fp) nx,ny,nz
allocate(x(nx),y(ny),z(nz))
allocate(xtemp(nx,ny,nz),ytemp(nx,ny,nz),ztemp(nx,ny,nz))
read(fp) (((xtemp(i,j,k),i=1,nx),j=1,ny),k=1,nz), &
          (((ytemp(i,j,k),i=1,nx),j=1,ny),k=1,nz), &
          (((ztemp(i,j,k),i=1,nx),j=1,ny),k=1,nz)
close(fp)

do j=1,nx
  x(j) = real(xtemp(j,1,1), 8)
end do
do j=1,ny
  y(j) = real(ytemp(1,j,1), 8)
end do
do j=1,nz
  z(j) = real(ztemp(1,1,j), 8)
end do

fname2=trim(dledir)//'/grid5.dat'
print *, '== Reading mesh file: '// trim(fname2)
open(unit=fp2, file=trim(fname2), status="old", form="unformatted", &
      iostat=status)
! print *, 'file opened'
if (status .ne. 0) then

```

```

        print *, "Error: could not open "//trim(fname2)//" for reading.", status
        stop
    end if

    read(fp2) ox,oy,oz
    allocate(x1(ox),y1(oy),z1(oz))
    allocate(x1temp(ox,oy,oz),y1temp(ox,oy,oz),z1temp(ox,oy,oz))
    read(fp2) (((x1temp(i,j,k),i=1,ox),j=1,oy),k=1,oz), &
               (((y1temp(i,j,k),i=1,ox),j=1,oy),k=1,oz), &
               (((z1temp(i,j,k),i=1,ox),j=1,oy),k=1,oz)
    close(fp2)

    do j=1,ox
        x1(j) = real(x1temp(j,1,1), 8)
    end do
    do j=0,oy
        y1(j) = real(y1temp(1,j,1), 8)
    end do
    do j=1,oz
        z1(j) = real(z1temp(1,1,j), 8)
    end do

    print *, "nx = ", nx, "ox = ", ox
    print *, "ny = ", ny, "oy = ", oy
    print *, "nz = ", nz, "oz = ", oz

end subroutine setup_grid

subroutine cleanup_grid
    implicit none
    deallocate(x,y,z,x1,y1,z1)
end subroutine cleanup_grid

end module grid

module strainsubs
    use grid
    implicit none

contains

    subroutine hesscalc(dle, dlehess)
        implicit none
        real, dimension(ox,oy,oz), intent(in) :: dle
        real, dimension(ox,oy,oz,3,3), intent(inout) :: dlehess

```

```

integer, dimension(3) :: spos
real :: dx, dy, dz
integer :: i,j,k

dx = x1(2)-x1(1)
dy = y1(2)-y1(1)
dz = z1(2)-z1(1)

do i=2,ox-1
  do j=2,oy-1
    do k=2,oz-1

      dlehess(i,j,k,1,1) = (dle(i+1,j,k) - (2*dle(i,j,k)) + &
        & dle(i-1,j,k))/(dx**2)
      dlehess(i,j,k,2,2) = (dle(i,j+1,k) - (2*dle(i,j,k)) + &
        & dle(i,j-1,k))/(dy**2)
      dlehess(i,j,k,3,3) = (dle(i,j,k+1) - (2*dle(i,j,k)) + &
        & dle(i,j,k-1))/(dz**2)

      dlehess(i,j,k,2,1) = ((dle(i+1,j+1,k)-dle(i+1,j-1,k)) - &
        & (dle(i-1,j+1,k)-dle(i-1,j-1,k)))/(4*dx*dy)
      dlehess(i,j,k,1,2) = dlehess(i,j,k,2,1)

      dlehess(i,j,k,3,1) = ((dle(i+1,j,k+1)-dle(i+1,j,k-1)) - &
        & (dle(i-1,j,k+1)-dle(i-1,j,k-1)))/(4*dx*dz)
      dlehess(i,j,k,1,3) = dlehess(i,j,k,3,1)

      dlehess(i,j,k,2,3) = ((dle(i,j+1,k+1)-dle(i,j-1,k+1)) - &
        & (dle(i,j+1,k-1)-dle(i,j-1,k-1)))/(4*dy*dz)
      dlehess(i,j,k,3,2) = dlehess(i,j,k,2,3)

    end do
  end do
end do

end subroutine hesscalc

subroutine normcalc(dlehess,dlenorm)
  implicit none
  real, dimension(ox,oy,oz, 3, 3), intent(in) :: dlehess
  real, dimension(ox,oy,oz, 3), intent(inout) :: dlenorm
  real, dimension(3,3) :: mat, mult
  real, dimension(3) :: eigs
  real :: orgp, orgq, tilq, orgr, tilr, theta, pi
  integer :: i, j, k, m, n
  real :: a, b, c, d, e, hold

```

```

pi = 4.0*atan(1.0)

do i=1,ox
  do j=1,oy
    do k=1,oz

      do m=1,3
        do n=1,3
          mat(m,n) = dlehess(i,j,k,m,n)
        end do
      end do

      orgp = -( mat(1,1)+mat(2,2)+mat(3,3) )

      mult = matmul(mat,mat)

      orgq = 0.5*( orgp**2 - ( mult(1,1)+mult(2,2)+mult(3,3) ) )
      tilq = orgq - orgp**2/3.

      mult = matmul(mult,mat)

      orgr = ( -orgp**3 + 3.*orgp*orgq - ( mult(1,1)+mult(2,2) + &
        & mult(3,3) ) )/3.
      tilr = orgr + 2.*orgp**3/27. - orgp*orgq/3.

      theta = acos( (-.5*tilr) / ((-1./3.)*tilq)**(1.5) )

      eigs(1) = 2 * ((-1./3.)*tilq)**.5 * cos(theta/3.)
      eigs(2) = 2 * ((-1./3.)*tilq)**.5 * cos((theta+(2*pi))/3.)
      eigs(3) = 2 * ((-1./3.)*tilq)**.5 * cos((theta+(4*pi))/3.)

      eigs(1) = eigs(1) - (1./3.)*orgp
      eigs(2) = eigs(2) - (1./3.)*orgp
      eigs(3) = eigs(3) - (1./3.)*orgp

      if(abs(eigs(1)).ge.abs(eigs(2))) then
        hold = eigs(1)
      else
        hold = eigs(2)
      end if

      if(abs(eigs(3)).gt.abs(hold))) hold = eigs(3)

      a = mat(1,1)
      b = mat(1,2)
      c = mat(1,3)
      d = mat(2,2)

```

```

e = mat(2,3)

dlenorm(i,j,k,1) = (c*b) - (e*(a-hold)) - (((c*b*b) - &
& (c*(a-hold)*(d-hold))))/b
dlenorm(i,j,k,2) = ((-b*c*(a-hold)) + (e*(a-hold)*(a-hold)) )/ b
dlenorm(i,j,k,3) = ((b*b*(a-hold)) - ((a-hold)*(a-hold)*(d-hold)))/b

hold = sqrt((dlenorm(i,j,k,1)**2) + (dlenorm(i,j,k,2)**2) + &
& (dlenorm(i,j,k,3)**2))

dlenorm(i,j,k,1) = dlenorm(i,j,k,1) / hold
dlenorm(i,j,k,2) = dlenorm(i,j,k,2) / hold
dlenorm(i,j,k,3) = dlenorm(i,j,k,3) / hold

end do
end do
end do

end subroutine normcalc

function strainprodcalc(dlenorm, gvel) result(strainprod)
implicit none
real, dimension(ox,oy,oz, 3), intent(in) :: dlenorm
real, dimension(ox,oy,oz, 3, 3), intent(in) :: gvel
real, dimension(ox,oy,oz) :: strainprod
integer :: i, j, k, m, n
real, dimension(3,3) :: strain
real, dimension(3) :: Sn

do i=1,ox
do j=1,oy
do k=1,oz

do m=1,3
do n=1,3
strain(m,n) = .5*(gvel(i,j,k,m,n) + gvel(i,j,k,n,m))
end do
end do

Sn(1) = (strain(1,1)* dlenorm(i,j,k,1)) + (strain(1,2))* &
& (dlenorm(i,j,k,2)) + (strain(1,3)*dlenorm(i,j,k,3))
Sn(2) = (strain(2,1)* dlenorm(i,j,k,1)) + (strain(2,2))* &
& (dlenorm(i,j,k,2)) + (strain(2,3)*dlenorm(i,j,k,3))
Sn(3) = (strain(3,1)* dlenorm(i,j,k,1)) + (strain(3,2))* &
& (dlenorm(i,j,k,2)) + (strain(3,3)*dlenorm(i,j,k,3))

if(i.eq.1.or.i.eq.ox.or.j.eq.1.or.j.eq.oy.or.k.eq.1.or.k.eq.oz) then

```

```

        strainprod(i,j,k)=0
    else
        strainprod(i,j,k) = (dlenorm(i,j,k,1)*Sn(1)) + &
            & (dlenorm(i,j,k,2)*Sn(2)) + (dlenorm(i,j,k,3)*Sn(3))
    end if

    end do
end do
end do

end function strainprodcalc

function interp(a1,a2,b1,b2,am) result(bm)
    implicit none
    real, intent(in) :: a1,a2,am
    real, intent(in) :: b1,b2
    real :: bm

    bm = b2 + ((b1-b2)*(am-a2)/(a1-a2))

end function interp

function interp3d(spos,loc,cube3d) result(yea)
    implicit none
    integer, dimension(3), intent(in) :: spos
    real, dimension(3), intent(in) :: loc
    real, dimension(2,2,2), intent(in) :: cube3d
    real :: yea
    real :: xp, yp, zp
    real :: fx1y1,fx1y2,fx2y1,fx2y2,fx1,fx2

    xp = loc(1)
    yp = loc(2)
    zp = loc(3)

    fx1y1 = interp(z1(spos(3)), z1(spos(3)+1), cube3d(1,1,1), cube3d(1,1,2), zp)
    fx1y2 = interp(z1(spos(3)), z1(spos(3)+1), cube3d(1,2,1), cube3d(1,2,2), zp)
    fx2y1 = interp(z1(spos(3)), z1(spos(3)+1), cube3d(2,1,1), cube3d(2,1,2), zp)
    fx2y2 = interp(z1(spos(3)), z1(spos(3)+1), cube3d(2,2,1), cube3d(2,2,2), zp)

    !interpolate the x = x1 and the x = x2 faces

    fx1 = interp(y1(spos(2)), y1(spos(2)+1), fx1y1, fx1y2, yp)

    fx2 = interp(y1(spos(2)), y1(spos(2)+1), fx2y1, fx2y2, yp)

    !interpolate between the two faces to get the final data point

```

```

    yea = interp(x1(spos(1)), x1(spos(1)+1), fx1, fx2, xp)

end function interp3d

function getcube(spos,thing) result(cube)
    implicit none
    integer, dimension(3), intent(in) :: spos
    real, dimension(ox,oy,oz), intent(in) :: thing
    real, dimension(2,2,2) :: cube

    !obtain cube of 8 data points to pass to interp3d

    cube(1,1,1) = thing(spos(1),spos(2),spos(3))
    cube(1,2,1) = thing(spos(1),spos(2)+1,spos(3))
    cube(2,1,1) = thing(spos(1)+1,spos(2),spos(3))
    cube(2,2,1) = thing(spos(1)+1,spos(2)+1,spos(3))
    cube(1,1,2) = thing(spos(1),spos(2),spos(3)+1)
    cube(1,2,2) = thing(spos(1),spos(2)+1,spos(3)+1)
    cube(2,1,2) = thing(spos(1)+1,spos(2),spos(3)+1)
    cube(2,2,2) = thing(spos(1)+1,spos(2)+1,spos(3)+1)

end function getcube

function getbound(pos) result(spos)
    implicit none
    real, dimension(3), intent(in) :: pos
    integer :: posx1, posy1, posz1
    integer, dimension(3) :: spos
    integer :: j

    spos(1) = ox-2
    spos(2) = oy-2
    spos(3) = oz-2

    do j=1,ox-1
        if(pos(1).lt.x1(j)) then
            spos(1) = j-1
            exit
        end if
    end do

    if(spos(1).eq.0) spos(1)=1

```



```

do j=1,oy-1
  if(pos(2).lt.y1(j)) then
    spos(2) = j-1
    exit
  end if
end do

if(spos(2).eq.0) spos(2)=1

do j=1,oz-1
  if(pos(3).lt.z1(j)) then
    spos(3) = j-1
    exit
  end if
end do

if(spos(3).eq.0) spos(3)=1

end function getbound

end module strainsubs

module velgradsubs
  use grid
  use inout
  use derivs
  use strainsubs
  implicit none

contains

subroutine velst(gvel, t)
  implicit none
  real, dimension(ox,oy,oz, 3, 3) :: gvel
  real, dimension(:,:,:), allocatable :: u,v,w
  real, dimension(:,:,:), allocatable :: ux, uy, uz, vx, vy, vz, wx, wy, wz
  real, dimension(3) :: hold
  integer, dimension(3) :: spos
  integer :: status, t
  character(150) :: fname
  integer, parameter :: fp=65

  allocate(u(nx,ny,nz), v(nx,ny,nz), w(nx,ny,nz))
  allocate(ux(nx,ny,nz), vx(nx,ny,nz), wx(nx,ny,nz))
  allocate(uy(nx,ny,nz), vy(nx,ny,nz), wy(nx,ny,nz))
  allocate(uz(nx,ny,nz), vz(nx,ny,nz), wz(nx,ny,nz))

```

```

call readvel(u,v,w, t)

ux = ddx(u)
uy = ddy(u)
uz = ddz(u)
vx = ddx(v)
vy = ddy(v)
vz = ddz(v)
wx = ddx(w)
wy = ddy(w)
wz = ddz(w)

do i=1,ox
  do j=1,oy
    do k=1,oz
      hold(1) = x1(i)
      hold(2) = y1(j)
      hold(3) = z1(k)

      spos = getboundvel(hold)

      gvel(i,j,k,1,1) = interp3dvel(spos,hold,getcubevel(spos,ux(:,:)))
      gvel(i,j,k,1,2) = interp3dvel(spos,hold,getcubevel(spos,uy(:,:)))
      gvel(i,j,k,1,3) = interp3dvel(spos,hold,getcubevel(spos,uz(:,:)))
      gvel(i,j,k,2,1) = interp3dvel(spos,hold,getcubevel(spos,vx(:,:)))
      gvel(i,j,k,2,2) = interp3dvel(spos,hold,getcubevel(spos,vy(:,:)))
      gvel(i,j,k,2,3) = interp3dvel(spos,hold,getcubevel(spos,vz(:,:)))
      gvel(i,j,k,3,1) = interp3dvel(spos,hold,getcubevel(spos,wx(:,:)))
      gvel(i,j,k,3,2) = interp3dvel(spos,hold,getcubevel(spos,wy(:,:)))
      gvel(i,j,k,3,3) = interp3dvel(spos,hold,getcubevel(spos,wz(:,:)))
    end do
  end do
end do

end subroutine velst

subroutine readvel(u,v,w, t)
  implicit none
  real, dimension(nx, ny, nz), intent(inout) :: u,v,w
  character(150) :: fname
  integer, parameter :: fp=65
  integer :: status, t
  integer :: hi, hi2, hi3, hi4
  real*4, dimension(:,:,:),allocatable :: u4, v4, w4

```

```

allocate(u4(nx,ny,nz), v4(nx,ny,nz), w4(nx,ny,nz))

write(fname,'(a,i2.2,a)') trim(chandir) // "/Velavgint_phase_",t, ".dat"
print *, "Reading data file ",trim(fname)
open(unit=fp, file=trim(fname), status="old", form="unformatted", &
      iostat=status)
! print *, 'file opened'
if (status .ne. 0) then
    print *, "Error: could not open "//trim(fname)//" for reading.", status
    stop
end if

read(fp) hi, hi2, hi3, hi4
read(fp) u4(:,:,:), v4(:,:,:), w4(:,:,:)
close(fp)

do i=1,nx
    do j=1,ny
        do k=1,nz
            u(i,j,k) = real(u4(i,j,k), 8)
            v(i,j,k) = real(v4(i,j,k), 8)
            w(i,j,k) = real(w4(i,j,k), 8)
        end do
    end do
end do

end subroutine readvel

function interp3dvel(spos,loc,cube3d) result(yea)
    implicit none
    integer, dimension(3), intent(in) :: spos
    real, dimension(3), intent(in) :: loc
    real, dimension(2,2,2), intent(in) :: cube3d
    real :: yea
    real :: xp, yp, zp
    real :: fx1y1,fx1y2,fx2y1,fx2y2,fx1,fx2

    xp = loc(1)
    yp = loc(2)
    zp = loc(3)

    if(x(nx).gt.x(1)) then
15      if(xp.gt.x(nx)) then
            xp = x(nx)
            goto 15
        endif
25      if(xp.lt.x(1)) then

```

```

        xp = x(1)
        goto 25
    endif
else
75    if(xp.gt.x(1)) then
        xp = x(1)
        goto 75
    endif
85    if(xp.lt.x(nx)) then
        xp = x(nx)
        goto 85
    endif
endif

if(y(ny).gt.y(1)) then
    if(yp.ge.y(ny)) then
        yp = y(ny)
    endif
    if(yp.le.y(1)) then
        yp = y(1)
    endif
else
    if(yp.ge.y(1)) then
        yp = y(1)
    endif
    if(yp.le.y(ny)) then
        yp = y(ny)
    endif
endif

if(z(nz).gt.z(1)) then
55    if(zp.gt.z(nz)) then
        zp = z(nz)
        goto 55
    endif

65    if(zp.lt.z(1)) then
        zp = z(1)
        goto 65
    endif
else
115    if(zp.gt.z(1)) then
        zp = z(1)
        goto 115
    endif
125    if(zp.lt.z(nz)) then
        zp = z(nz)

```

```

        goto 125
    endif
endif

!interpolate along corner lines of 3d cube

fx1y1 = interp(z(spos(3)), z(spos(3)+1), cube3d(1,1,1), cube3d(1,1,2), zp)
fx1y2 = interp(z(spos(3)), z(spos(3)+1), cube3d(1,2,1), cube3d(1,2,2), zp)
fx2y1 = interp(z(spos(3)), z(spos(3)+1), cube3d(2,1,1), cube3d(2,1,2), zp)
fx2y2 = interp(z(spos(3)), z(spos(3)+1), cube3d(2,2,1), cube3d(2,2,2), zp)

!interpolate the x = x1 and the x = x2 faces

fx1 = interp(y(spos(2)), y(spos(2)+1), fx1y1, fx1y2, yp)

fx2 = interp(y(spos(2)), y(spos(2)+1), fx2y1, fx2y2, yp)

!interpolate between the two faces to get the final data point

yea = interp(x(spos(1)), x(spos(1)+1), fx1, fx2, xp)

end function interp3dvel

function getcubevel(spos,thing) result(cube)
    implicit none
    integer, dimension(3), intent(in) :: spos
    real, dimension(nx,ny,nz), intent(in) :: thing
    real, dimension(2,2,2) :: cube

    !obtain cube of 8 data points to pass to interp3d
    cube(1,1,1) = thing(spos(1),spos(2)-1,spos(3))
    cube(1,2,1) = thing(spos(1),spos(2)+1-1,spos(3))
    cube(2,1,1) = thing(spos(1)+1,spos(2)-1,spos(3))
    cube(2,2,1) = thing(spos(1)+1,spos(2)+1-1,spos(3))
    cube(1,1,2) = thing(spos(1),spos(2)-1,spos(3)+1)
    cube(1,2,2) = thing(spos(1),spos(2)+1-1,spos(3)+1)
    cube(2,1,2) = thing(spos(1)+1,spos(2)-1,spos(3)+1)
    cube(2,2,2) = thing(spos(1)+1,spos(2)+1-1,spos(3)+1)

end function getcubevel

function getboundvel(loc) result(spos)

```

```

implicit none
real, dimension(3), intent(in) :: loc
integer, dimension(3) :: spos
real :: xp, yp, zp
integer :: j

spos(1) = nx-1
spos(2) = ny-1
spos(3) = nz-1

xp = loc(1)
yp = loc(2)
zp = loc(3)

! Need to make sure that the points are still in the domain,
! otherwise take care of them (here, by imposing periodic
! conditions in x- and z- directions, and just stick to the wall in y

      if(x(nx).ge.x(1)) then
15      if(xp.gt.x(nx)) then
            xp = x(nx)
            goto 15
        endif
25      if(xp.lt.x(1)) then
            xp = x(1)
            goto 25
        endif
    else
75      if(xp.gt.x(1)) then
            xp = x(1)
            goto 75
        endif
85      if(xp.lt.x(nx)) then
            xp = x(nx)
            goto 85
        endif
    endif

      if(y(ny).gt.y(1)) then
            if(yp.ge.y(ny)) then
                yp = y(ny)
            endif
            if(yp.le.y(1)) then
                yp = y(1)
            endif
    else

```

```

        if(yp.ge.y(1)) then
            yp = y(1)
        endif
        if(yp.le.y(ny)) then
            yp = y(ny)
        endif
    endif

    if(z(nz).gt.z(1)) then
55      if(zp.gt.z(nz)) then
            zp = z(nz)
            goto 55
        endif
65      if(zp.lt.z(1)) then
            zp = z(1)
            goto 65
        endif
    else
115     if(zp.gt.z(1)) then
            zp = z(1)
            goto 115
        endif
125     if(zp.lt.z(nz)) then
            zp = z(nz)
            goto 125
        endif
    endif

    if(x(nx).gt.x(1))then
        do j=1,nx
            if(xp.lt.x(j)) then
                spos(1) = j-1
                exit
            end if
        end do
        if(spos(1).eq.0) spos(1)=1
    else
        do j=1,nx
            if(xp.gt.x(j)) then
                spos(1) = j-1
                exit
            end if
        end do
        if(spos(1).eq.0) spos(1)=1
    end if

```

```

    if(y(ny).gt.y(1))then
      do j=1,ny
        if(yp.lt.y(j)) then
          spos(2) = j-1
          exit
        end if
      end do
      if(spos(2).eq.0) spos(2)=1
    else
      do j=1,ny
        if(yp.gt.y(j)) then
          spos(2) = j-1
          exit
        end if
      end do
      if(spos(2).eq.0) spos(2)=1
    end if

    if(z(nz).gt.z(1))then
      do j=1,nz
        if(zp.lt.z(j)) then
          spos(3) = j-1
          exit
        end if
      end do
      if(spos(3).eq.0) spos(3)=1
    else
      do j=1,nz
        if(zp.gt.z(j)) then
          spos(3) = j-1
          exit
        end if
      end do
      if(spos(3).eq.0) spos(3)=1
    end if

    end function getboundvel

end module velgradsubs

module derivs
  use grid
  use inout
  implicit none

```



contains

```
function ddx(u) result(diffx)
  real, dimension(nx,ny,nz) :: u
  real, dimension(nx,ny,nz) :: diffx
  real :: xa, xc, ua, uc

  do i=1,nx
    do j=1,ny
      do k=1,nz

        if(i.eq.1)then
          xa = x(i)
          xc = x(i+1)
          ua = u(i,j,k)
          uc = u(i+1,j,k)
        elseif(i.eq.nx)then
          xa = x(i-1)
          xc = x(i)
          ua = u(i-1,j,k)
          uc = u(i,j,k)
        else
          xa = x(i-1)
          xc = x(i+1)
          ua = u(i-1,j,k)
          uc = u(i+1,j,k)
        end if

        diffx(i,j,k) = (ua - uc) / (xa - xc)

      end do
    end do
  end do

end function ddx

function ddy(u) result(diffy)
  real, dimension(nx,ny,nz) :: u
  real, dimension(nx,ny,nz) :: diffy
  real :: ya, yc, ua, uc

  do i=1,nx
    do j=1,ny
      do k=1,nz
```

```

        if(j.eq.1)then
            ya = y(j)
            yc = y(j+1)
            ua = u(i,j,k)
            uc = u(i,j+1,k)
        elseif(j.eq.ny)then
            ya = y(j-1)
            yc = y(j)
            ua = u(i,j-1,k)
            uc = u(i,j,k)
        else
            ya = y(j-1)
            yc = y(j+1)
            ua = u(i,j-1,k)
            uc = u(i,j+1,k)
        end if

        diffy(i,j,k) = (ua - uc) / (ya - yc)

    end do
end do
end do

end function ddy

function ddz(u) result(diffz)
    real, dimension(nx,ny,nz) :: u
    real, dimension(nx,ny,nz) :: diffz
    real :: za, zc, ua, uc

    do i=1,nx
        do j=1,ny
            do k=1,nz

                if(k.eq.0)then
                    za = z(k)
                    zc = z(k+1)
                    ua = u(i,j,k)
                    uc = u(i,j,k+1)
                elseif(k.eq.nz)then
                    za = z(k-1)
                    zc = z(k)
                    ua = u(i,j,k-1)
                    uc = u(i,j,k)
                else
                    za = z(k-1)

```

```

        zc = z(k+1)
        ua = u(i,j,k-1)
        uc = u(i,j,k+1)
    end if

    diffz(i,j,k) = (ua - uc) / (za - zc)

    end do
end do
end do

end function ddz

end module derivs

module fourier
! Routines for transforming to Fourier coefs, and computing derivs
! Note: uses FFTW library for fast Fourier transforms
!
! Clancy Rowley
! 2 November 2001

implicit none
include "fftw_f77.i"

type fourier_type
    integer*8 :: plan, planinv
    integer :: size
    logical :: not2pi
    real :: lengthfac ! 2pi/L
end type fourier_type

type fourier2_type
    integer*8 :: plan, planinv
    integer :: n1, n2, size
    real :: len1, len2 ! length = 2*pi * (len1, len2)
end type fourier2_type

contains

function create_fourier(n, length) result(p)
    implicit none
    type(fourier_type) :: p
    real, optional :: length
    integer :: n

    if (present(length)) then
        p%not2pi = .true.

```

```

        p%lengthfac = (8. * atan(1.)) / length ! 2pi/L
    else
        p%not2pi = .false.
        p%lengthfac = 1
    end if
    p%size = n
    call rfftw_f77_create_plan(p%plan, n, FFTW_REAL_TO_COMPLEX, FFTW_MEASURE)
    call rfftw_f77_create_plan(p%planinv, n, FFTW_COMPLEX_TO_REAL, FFTW_MEASURE)
end function create_fourier

function create_fourier2(n1, n2, length1, length2) result(p)
    implicit none
    type(fourier2_type) :: p
    real, optional :: length1, length2
    integer :: n1, n2

    p%n1 = n1
    p%n2 = n2
    p%size = n1 * n2
    p%len1 = 1.
    p%len2 = 1.
    if (present(length1)) then
        p%len1 = length1
    end if
    if (present(length2)) then
        p%len2 = length2
    end if

    call rfftw2d_f77_create_plan(p%plan, n1, n2, &
        FFTW_FORWARD, FFTW_MEASURE + FFTW_IN_PLACE)
    call rfftw2d_f77_create_plan(p%planinv, n1, n2, &
        FFTW_BACKWARD, FFTW_MEASURE + FFTW_IN_PLACE)
end function create_fourier2

subroutine destroy_fourier(p)
    implicit none
    type(fourier_type) :: p
    call rfftw_f77_destroy_plan(p%plan)
    call rfftw_f77_destroy_plan(p%planinv)
end subroutine destroy_fourier

subroutine destroy_fourier2(p)
    implicit none
    type(fourier2_type) :: p
    call rfftwnd_f77_destroy_plan(p%plan)
    call rfftwnd_f77_destroy_plan(p%planinv)
end subroutine destroy_fourier2

```

```

function phys2fourier(p, u) result(a)
  ! transform physical variable u to Fourier coefs a
  ! Fourier coefs are stored in the following order: (n even)
  !   r(0), r(1), r(2), ..., r(n/2), i(n/2 - 1), ..., i(2), i(1)
  implicit none
  type(fourier_type) :: p
  real, dimension(:) :: u
  real, dimension(size(u)) :: a

  call rfftw_f77_one(p%plan, u, a)
  a = a / p%size
end function phys2fourier

function fourier2phys(p, a) result(u)
  ! transform Fourier coefs a to physical space u
  ! NOTE: destroys the array 'a' (the Fourier coefs)
  implicit none
  type(fourier_type) :: p
  real, dimension(:) :: a
  real, dimension(size(a)) :: u

  call rfftw_f77_one(p%planinv, a, u)

end function fourier2phys

subroutine phys2fourier2(p, u)
  ! transform physical variable to Fourier coefs, in place
  implicit none
  type(fourier2_type) :: p
  real, dimension(p%n1+2, p%n2) :: u

  call rfftwnd_f77_one_real_to_complex(p%plan, u, 0)
  u = u / p%size

end subroutine phys2fourier2

subroutine fourier2phys2(p, a)
  ! transform Fourier coefs to physical space, in place
  implicit none
  type(fourier2_type) :: p
  real, dimension(p%n1+2, p%n2) :: a

  call rfftwnd_f77_one_complex_to_real(p%planinv, a, 0)
end subroutine fourier2phys2

function fourier_d1(p, a) result(da)

```

```

! compute first derivative of Fourier coefficients a
implicit none
type(fourier_type) :: p
real, dimension(:) :: a
real, dimension(size(a)) :: da
! real :: tmp
integer :: k, n

! multiply by ik
n = p%size
da(1) = 0
da(n/2 + 1) = 0
do k=1,n/2-1
    da(k+1) = -k * a(n+1-k)
    da(n+1-k) = k * a(k+1)
end do
if (p%not2pi) da = da * p%lengthfac
end function fourier_d1

function fourier_d2(p, a) result(da)
! compute second derivative of Fourier coefficients a
implicit none
type(fourier_type) :: p
real, dimension(:) :: a
real, dimension(size(a)) :: da
integer :: k, n2

! multiply by -k^2
da(1) = 0
da(p%size/2 + 1) = -a(p%size/2 + 1) * (p%size/2)**2
n2 = p%size + 2
do k=2,p%size/2
    da(k) = -(k-1)**2 * a(k)
    da(n2 - k) = -(k-1)**2 * a(n2 - k)
end do
if (p%not2pi) da = da * (p%lengthfac)**2
end function fourier_d2

function fourier2_d1(p, a) result(da)
! compute first derivative of Fourier coefficients a, w.r.t. first coord
implicit none
type(fourier2_type) :: p
real, dimension(p%n1+2, p%n2) :: a, da
integer :: k1

! multiply by i*k1
do k1 = 0,p%n1/2

```

```

        da(2*k1+1, :) = -k1 * a(2*k1+2, :) / p%len1
        da(2*k1+2, :) = k1 * a(2*k1+1, :) / p%len1
    end do
end function fourier2_d1

function fourier2_d2(p, a) result(da)
    ! compute first derivative of Fourier coefficients a, w.r.t. second coord
    implicit none
    type(fourier2_type) :: p
    real, dimension(p%n1+2, p%n2) :: a, da
    integer :: k1, k2

    ! multiply by i*k2
    do k1 = 0, p%n1/2
        do k2 = 0, p%n2/2-1
            da(2*k1+1, k2+1) = -k2 * a(2*k1+2, k2+1) / p%len2 ! real part
            da(2*k1+2, k2+1) = k2 * a(2*k1+1, k2+1) / p%len2 ! imag part
        end do
        do k2 = -p%n2/2, -1
            da(2*k1+1, k2+1+p%n2) = -k2 * a(2*k1+2, k2+1+p%n2) / p%len2 ! real part
            da(2*k1+2, k2+1+p%n2) = k2 * a(2*k1+1, k2+1+p%n2) / p%len2 ! imag part
        end do
    end do
end function fourier2_d2

function fourier2_lap(p, a) result(da)
    ! compute Laplacian of Fourier coefficients a
    implicit none
    type(fourier2_type) :: p
    real, dimension(p%n1+2, p%n2) :: a, da
    integer :: k1, k2

    ! multiply by -(k1^2 + k2^2)
    ! do j = 1, p%n2
    !     k2 = mod(j + p%n2/2 - 1, p%n2) - p%n2/2
    !     do k1 = 0, p%n1/2
    !         da(2*k1+1, j) = -(k1**2 / p%len1**2 + k2**2 / p%len2**2) * a(2*k1+1, j)
    !         da(2*k1+2, j) = -(k1**2 / p%len1**2 + k2**2 / p%len2**2) * a(2*k1+2, j)
    !     end do
    ! end do
    do k1 = 0, p%n1/2
        do k2 = 0, p%n2/2-1
            da(2*k1+1, k2+1) = -((k1 / p%len1)**2 + (k2 / p%len2)**2) * &
                a(2*k1+1, k2+1) ! real part
            da(2*k1+2, k2+1) = -((k1 / p%len1)**2 + (k2 / p%len2)**2) * &
                a(2*k1+2, k2+1) ! imag part
        end do
    end do
end function fourier2_lap

```

```

        do k2 = -p%n2/2,-1
            da(2*k1+1, k2+1+p%n2) = -((k1 / p%len1)**2 + (k2 / p%len2)**2) * &
                a(2*k1+1, k2+1+p%n2) ! real part
            da(2*k1+2, k2+1+p%n2) = -((k1 / p%len1)**2 + (k2 / p%len2)**2) * &
                a(2*k1+2, k2+1+p%n2) ! imag part
        end do
    end do
end function fourier2_lap

end module fourier

module cheby
    ! Routines for transforming to Chebyshev coefs, and computing derivs
    ! Note: uses FFTW library for fast Fourier transforms
    !
    ! Clancy Rowley
    ! 1 November 2001

    implicit none
    include "fftw_f77.i"

    type cheby_type ! real data
        integer*8 :: plan
        integer :: size
        real, dimension(:), pointer :: tmp
    end type cheby_type

    type chebyc_type ! complex data
        integer*8 :: plan
        integer :: size
        complex, dimension(:), pointer :: tmp
    end type chebyc_type

    private :: cosft, cosftc

contains
    function create_cheby(n)
        implicit none
        integer :: n
        type(cheby_type) :: create_cheby

        create_cheby%size = n
        call rfftw_f77_create_plan(create_cheby%plan, n, &
            FFTW_REAL_TO_COMPLEX, FFTW_MEASURE)
        allocate(create_cheby%tmp(0:n))
    end function create_cheby

```



```

function create_chebyc(n) result(p)
    implicit none
    integer :: n
    type(chebyc_type) :: p

    p%size = n
    call fftw_f77_create_plan(p%plan, n, FFTW_FORWARD, FFTW_MEASURE)
    allocate(p%tmp(0:n))
end function create_chebyc

subroutine destroy_cheby(p)
    implicit none
    type(cheby_type) :: p
    call rfftw_f77_destroy_plan(p%plan)
    deallocate(p%tmp)
end subroutine destroy_cheby

subroutine destroy_chebyc(p)
    implicit none
    type(chebyc_type) :: p
    call fftw_f77_destroy_plan(p%plan)
    deallocate(p%tmp)
end subroutine destroy_chebyc

function phys2cheby(p, u) result(a)
    ! transform physical variable u to Chebyshev coeffs
    implicit none
    type(cheby_type) :: p
    real, dimension(0:) :: u
    real, dimension(0:ubound(u,1)) :: a

    call cosft(p, u, a)

    a = a * 2 / p%size
    a(0) = a(0) / 2
    a(p%size) = a(p%size) / 2

end function phys2cheby

function cheby2phys(p, a)
    ! transform Chebyshev coeffs a to physical space
    implicit none
    type(cheby_type) :: p
    real, dimension(0:) :: a
    real, dimension(0:ubound(a,1)) :: cheby2phys

    a(0) = a(0) * 2

```

```

    a(p%size) = a(p%size) * 2
    call cosft(p, a, cheby2phys)
    ! restore a
    a(0) = a(0) / 2
    a(p%size) = a(p%size) / 2
end function cheby2phys

function phys2chebyc(p, u) result(a)
    ! transform physical variable u to Chebyshev coeffs
    implicit none
    type(chebyc_type) :: p
    complex, dimension(0:) :: u
    complex, dimension(0:ubound(u,1)) :: a

    call cosftc(p, u, a)
    a = a / p%size
end function phys2chebyc

function cheby2physc(p, a) result(u)
    ! transform Chebyshev coeffs a to physical space
    implicit none
    type(chebyc_type) :: p
    complex, dimension(0:) :: a
    complex, dimension(0:ubound(a,1)) :: u

    a(0) = a(0) * 2
    a(p%size) = a(p%size) * 2
    call cosftc(p, a, u)
    u(0) = u(0) * 2
    u(p%size) = u(p%size) * 2
    u = u / 2
    ! restore a
    a(0) = a(0) / 2
    a(p%size) = a(p%size) / 2
end function cheby2physc

function cheby_d1(p, a)
    ! compute first derivative of Chebyshev coefficients a
    implicit none
    type(cheby_type) :: p
    real, dimension(0:) :: a
    real, dimension(0:ubound(a,1)) :: cheby_d1
    integer :: k

    cheby_d1(p%size) = 0
    cheby_d1(p%size-1) = 2 * p%size * a(p%size)
    do k = p%size-1, 2, -1

```

```

        cheby_d1(k-1) = 2*k*a(k) + cheby_d1(k+1)
    end do
    cheby_d1(0) = a(1) + cheby_d1(2)/2
end function cheby_d1

function cheby_d2(p, a)
    ! compute second derivative of Chebyshev coefficients a
    ! (not implemented yet)
    implicit none
    type(cheby_type) :: p
    real, dimension(0:) :: a
    real, dimension(0:ubound(a,1)) :: cheby_d2

    cheby_d2 = cheby_d1(p,a)
    cheby_d2 = cheby_d1(p,cheby_d2)
end function cheby_d2

function cheby_d1c(p, a) result(da)
    ! compute first derivative of complex Chebyshev coefficients a
    implicit none
    type(chebyc_type) :: p
    complex, dimension(0:) :: a
    complex, dimension(0:ubound(a,1)) :: da
    integer :: k

    da(p%size) = 0
    da(p%size-1) = 2 * p%size * a(p%size)
    do k = p%size-1, 2, -1
        da(k-1) = 2*k*a(k) + da(k+1)
    end do
    da(0) = a(1) + da(2)/2
end function cheby_d1c

function cheby_d2c(p, a) result(da)
    ! compute second derivative of Chebyshev coefficients a
    ! (not implemented yet)
    implicit none
    type(chebyc_type) :: p
    complex, dimension(0:) :: a
    complex, dimension(0:ubound(a,1)) :: da

    da = cheby_d1c(p,a)
    da = cheby_d1c(p,da)
end function cheby_d2c

subroutine cosft(p, in, out)
    ! compute cosine transform of data, from Numerical Recipes

```

```

implicit none
type(cheby_type) :: p
real, dimension(0:) :: in, out
integer :: j,n
real :: sum, y1, y2, theta, wi, wpi, wpr, wr, wtemp

n = p%size
theta = 4. * atan(1.) / n
wpr = -2. * sin(0.5 * theta)**2
wpi = sin(theta)
sum = 0.5 * (in(0) - in(n))
out(0) = 0.5 * (in(0) + in(n))
out(n/2) = in(n/2)
! out(n) = in(n)
wi = 0
wr = 1
do j=1,n/2-1
    wtemp = wr
    wr = wr * wpr - wi * wpi + wr
    wi = wi * wpr + wtemp * wpi + wi
    y1 = 0.5 * (in(j) + in(n-j))
    y2 = (in(j) - in(n-j))
    out(j) = y1 - wi * y2
    out(n-j) = y1 + wi * y2
    sum = sum + wr * y2 ! carry along for later use in unfolding xform
end do
call rfftw_f77_one(p%plan, out, p%tmp)

out(0) = p%tmp(0)
out(1) = sum
do j=1,n/2-1
    out(2*j) = p%tmp(j)
    sum = sum - p%tmp(n-j)
    out(2*j+1) = sum
end do
out(n) = p%tmp(n/2)
end subroutine cosft

subroutine cosftc(p, in, out)
! compute cosine transform of complex data, from Numerical Recipes
implicit none
type(chebyc_type) :: p
complex, dimension(0:) :: in, out
integer :: j,n
complex :: sum, y1, y2, im
real :: theta, wi, wpi, wpr, wr, wtemp

```

```

    im = (0.,1.)
    n = p%size
    theta = 4. * atan(1.) / n
    wpr = -2. * sin(0.5 * theta)**2
    wpi = sin(theta)
    sum = 0.5 * (in(0) - in(n))
    out(0) = 0.5 * (in(0) + in(n))
    out(n/2) = in(n/2)
    ! out(n) = in(n)
    wi = 0
    wr = 1
    do j=1,n/2-1
        wtemp = wr
        wr = wr * wpr - wi * wpi + wr
        wi = wi * wpr + wtemp * wpi + wi
        y1 = 0.5 * (in(j) + in(n-j))
        y2 = (in(j) - in(n-j))
        out(j) = y1 - wi * y2
        out(n-j) = y1 + wi * y2
        sum = sum + wr * y2 ! carry along for later use in unfolding xform
    end do
    call fftw_f77_one(p%plan, out, p%tmp)

    out(0) = p%tmp(0)
    out(1) = 2 * sum
    do j=1,n/2-1
        out(2*j) = p%tmp(j) + p%tmp(n-j)
        out(2*j+1) = out(2*j-1) + im * (p%tmp(j) - p%tmp(n-j))
    end do
    out(n) = p%tmp(n/2)
end subroutine cosftc

end module cheby

module inout
    use grid
    implicit none

contains

    subroutine readdle(dle, t)
        implicit none
        real, dimension(ox,oy,oz), intent(inout) :: dle
        integer :: ios, t
        integer :: hi1, hi2, hi3, hi4
        character(125) :: fname
        real*4, dimension(ox,oy,oz) :: temp

```

```

integer, parameter :: fp=42

write(fname, '(a,i6.6, a, i6.6, a)') trim(dledir) // trim(dlename),t, &
  & "_",dlesteps,".dat"
print *, 'Reading dle data file ', fname
open(unit=fp, file=trim(fname), status="old", form="unformatted", &
  iostat=ios)

if (ios .ne. 0) then
  print *, "Error: could not open "//trim(fname)//" for reading.", ios
  stop
end if

read(fp) hi1, hi2, hi3, hi4
read(fp) (((temp(i,j,k),i=1,ox),j=1,oy),k=1,oz)
close(fp)

do i=1,ox
  do j=1,oy
    do k=1,oz
      dle(i,j,k) = real(temp(i,j,k),8)
    end do
  end do
end do

end subroutine readdle

subroutine writestrain(strainprod,dle, t)
  implicit none
  real, dimension(ox,oy,oz), intent(in) :: strainprod, dle
  character(100) :: fname
  integer :: t

  write(fname,'(a,i6.6,a,i6.6,a)') trim(odir) // trim(dlename) // &
    & "strain_",t,"_",dlesteps,".dat"
  print *, 'Writing data file ', fname
  open(24,file=trim(fname),form="unformatted")
  write(24) ox,oy,oz,2
  write(24) (((real(strainprod(i,j,k),4),i=1,ox),j=1,oy),k=1,oz), &
    (((real(strainprod(i,j,k),4),i=1,ox),j=1,oy),k=1,oz)
  close(24)

end subroutine writestrain

end module inout

```

