## UC San Diego UC San Diego Electronic Theses and Dissertations

## Title

High-Fidelity Lagrangian Coherent Structures Analysis and DNS with Discontinuous-Galerkin Methods

**Permalink** https://escholarship.org/uc/item/2cv4f732

Author Nelson, Daniel Alan Wendell

**Publication Date** 2015

Peer reviewed|Thesis/dissertation

#### UNIVERSITY OF CALIFORNIA, SAN DIEGO

#### SAN DIEGO STATE UNIVERSITY

#### High-Fidelity Lagrangian Coherent Structures Analysis and DNS with Discontinuous-Galerkin Methods

A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy

in

Engineering Sciences (Mechanical and Aerospace Engineering)

by

Daniel Alan Wendell Nelson

Committee in charge:

University of California, San Diego

Professor Sutanu Sarkar, Co-Chair Professor Yuri Bazilevs Professor Daniel M. Tartakovsky

San Diego State University

Professor Gustaaf B. Jacobs, Co-Chair Professor Fletcher J. Miller

2015

Copyright Daniel Alan Wendell Nelson, 2015 All rights reserved. The dissertation of Daniel Alan Wendell Nelson is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Co-Chair

Co-Chair

University of California, San Diego

San Diego State University

2015

## DEDICATION

To Sonia and Isaac.

## EPIGRAPH

Sometimes attaining the deepest familiarity with a question is our best substitute for actually having the answer. — Brian Greene

## TABLE OF CONTENTS

Signature Pag	e		iii
Dedication .			iv
Epigraph			v
Table of Cont	ents		vi
List of Figure	s		ix
List of Tables			xiv
Acknowledge	ments .		xv
Vita			xvi
Abstract of th	e Dissert	ation	xviii
Chapter 1 Chapter 2	Introdu 1.1 M 1.2 H 1.3 L 1.4 O Govern 2.1 T 2. 2.2 L 2.2 L 2.	ction	1 2 6 8 11 11 11 11 14 17 17
	2.	2.2 Finite-Time Lyapunov Exponent Field	18
Chapter 3	The Dis	scontinuous-Galerkin Spectral Element Method	22
Chapter 4	Effect of a Disco 4.1 O 4.2 H 4. 4. 4.	of Boundary Representation on Viscous, Separated Flows in ntinuous-Galerkin Navier-Stokes Solververview and Summaryigh-Order Curved Boundary Representation2.1Airfoil Geometry2.2Boundary Node Adjustment2.3Compatibility of Commercial Mesh Generators with	28 28 29 29 33
	т.	DG	34

	4.2.4 High-Order Boundary Fitting	35
	4.3 Problem Setup and Grid Independence	38
	4.3.1 Mesh Characteristics	41
	4.3.2 Grid Refinement	44
	4.3.3 Comparison with Previous Studies	44
	4.4 Direct Numerical Simulation	47
	4.4.1 Wake Analysis	47
	4.4.2 Force Analysis	51
	4.4.3 Pressure and Skin Friction Distributions	55
	4.4.4 Curvature Analysis	56
	4.4.5 Discussion	59
	4.5 Summary of Results	60
Chapter 5	Lagrangian Coherent Structures with High-Order Discontinuous-	
	Galerkin Methods	63
	5.1 Overview and Summary	63
	5.2 FTLE with Spectral Methods	64
	5.2.1 Fluid Particle Tracking Algorithm	64
	5.2.2 Forward-Time FTLE with Spectral Methods	65
	5.2.3 Multiple Flow Maps From a Single Particle Trace	66
	5.2.4 Algorithm Summary	74
	5.3 Forward-Time FTLE Test Cases	75
	5.3.1 Gyre Flow	76
	5.3.2 Vortex Advected by Uniform Flow	78
	5.3.3 Flow around a square cylinder	80
	5.4 DG-FTLE Test Cases	82
	5.4.1 Gyre Flow	83
	5.4.2 Vortex Advected by Uniform Flow	84
	5.4.3 Flow around a square cylinder	89
	5.5 Summary of Results	93
Chapter 6	High-Order Visualization of Three-Dimensional Lagrangian Coher-	
	ent Structures with DG-FTLE	99
	6.1 Overview and Summary	99
	6.2 Forward-Time FTLE with Spectral Methods	100
	6.3 Two and Three-Dimensional Lagrangian Visualizations With	
	DG-FTLE	107
	6.3.1 Lagrangian Coherent Structures Analysis of a Two-	
	Dimensional Airfoil Flow	108
	6.3.2 Three-Dimensional ABC Flow	117
	6.3.3 Three-Dimensional Airfoil Flow	124
	6.4 Summary of Results	130

Chapter 7	Conclusions	134
	7.1 Discontinuous-Galerkin Methods on Complex Geometries .	134
	7.2 Lagrangian Coherent Structures	135
	7.3 Future Work	137
Appendix A	FTLE Field for an Analytical Inviscid Vortex	138
Bibliography		141

### LIST OF FIGURES

Figure 2.1:	Schematic of a repelling LCS (a) and an attracting LCS (b)	18
Figure 2.2:	Maximal stretch of an infinitesimal spherical fluid parcel as deter-	
	mined by projecting the eigenvectors and eigenvalues of <b>C</b> onto an	
	ellipsoid <i>via</i> the operator norm.	20
Figure 3.1:	Mapping from physical space $(x, y, z)$ to the unit cube in computa-	
-	tional coordinates $(\xi, \eta, \zeta)$ . The mapping is uniquely defined by the	
	edge polynomial representation $\Gamma_i$ , the face polynomials $\Sigma_i$ and the	
	corner coordinates $\mathbf{x}_i$	23
Figure 3.2:	Covariant bases definition.	26
Figure 4.1:	Differences in face ordering between GridPro and DGSEM	35
Figure 4.2:	Comparision of the skin friction (a) and pressure (b) of the small do-	
0	main size and the large domain size using the curved-sided boundary	
	approximation.	39
Figure 4.3:	Comparison of the skin friction (a) and pressure (b) of the small do-	
i iguio iioi	main size and the large domain size using the straight-sided boundary	
	approximation	40
Figure 4.4:	Comparision of the lift spectra between the small domain size and the	10
	large domain size using the curved-sided boundary approximation	
	(a) and the straight-sided boundary approximation (b)	40
Figure 4 5.	Airfoil simulation setun	41
Figure 4.6:	Coarse mesh (a) and fine mesh (b)	42
Figure 4.7	Straight-sided mesh detail (a) and curved-sided mesh detail (b) for	12
I iguie 1.7.	Mesh 1 The polynomial order is $N = 12$	43
Figure 4.8.	Straight-sided mesh detail (a) and curved-sided mesh detail (b) for	15
1 iguie 1.6.	Mesh 2 The polynomial order is $N = 12$	43
Figure 4 9.	Convergence for curved-sided Mesh 1	45
Figure 4 10	Convergence for straight-sided Mesh 1. Note the additional frequen-	15
ingule mo	cies as compared to the curved-sided mesh	45
Figure 4 11.	Convergence for straight-sided Mesh 2	46
Figure 4 12:	Convergence for straight-sided Mesh 2. Note the additional frequen-	10
1 iguie 1.12.	cies as compared to the curved-sided mesh	46
Figure 4 13.	Flow field plots for grid 1. Vorticity for the curved-sided grid (a) and	10
1 iguie 1.15.	the straight-sided grid (h)	48
Figure $4.14$ .	Flow field plots for grid 1 RMS velocity for the curved-sided grid	40
1 iguie 4.14.	(a) and the straight-sided grid (b)	10
Figure 1 15.	Elow field plots for grid 2. Vorticity for the curved-sided grid (2) and	72
11guie 4.15.	the straight sided grid (b)	50
Figure 1 16.	Elow field plots for grid 2 <b>DMS</b> valority for the ourwad sided grid	50
1 iguie 4.10.	(a) and the straight sided grid (b)	51
Figure 4 17.	(a) and the straight-show grid $(0)$	51
1 igule 4.1/:	Time series of the fitt (a) and frequency spectra of the fitt (b) for grid 1 N = 12 for both the straight sided and surred sided and	57
	1. $IV = 12$ for boun the straight-sided and curved-sided grid	33

Figure 4.18:	Time series of the lift (a) and frequency spectra of the lift (b) for grid $2$ . $N = 12$ for both the straight-sided and curved-sided grid.	54
Figure 4.19:	Average pressure coefficient, $C_n$ , (a) and skin friction coefficient, $C_f$ .	01
U	for the upper surface (b) for Mesh 1	55
Figure 4.20:	Average pressure coefficient, $C_p$ , (a) and skin friction coefficient, $C_f$ ,	
-	for the upper surface (b) for Mesh 2	56
Figure 4.21:	Slope of the airfoil spline definition plotted against the <i>x</i> -coordinate	
	(a) and local error in slope (b) plotted against the spline parameter, <i>s</i> .	
	The error in the slope was computed for Mesh 2, $N = 12$	57
Figure 4.22:	Curvature in the airfoil spline definition (a) and local error in the	
	curvature (b), both plotted as a function of the spline parameter, <i>s</i> .	
	The error in the curvature was computed for the curved-sided Mesh	
	$2, N = 12. \dots \dots$	58
Figure 4.23:	Effect of boundary representation near the leading edge. The black	
	vertical lines depict the locations of element boundaries. Curved-	50
Eigung 4 24.	Sided Mesn 2 is used	38
Figure 4.24:	vertical lines deniet the locations of element boundaries. Curved	
	sided Mash 2 is used	50
Figure 5.1.	The nodes that define the flow map subdomain boundaries $\Gamma^{\phi}$ are	59
1 iguie 5.1.	constructed from the final locations of the nodes initialized at the	
	boundaries of the subdomains at time $t = t_0$ such that $\Gamma^{\phi} = \Phi_N _{20}$ .	68
Figure 5.2:	The fluid tracers are initialized at the Chebyshev-Lobatto quadrature	
U	nodes at time $t = t_0$ , designated by the filled circles (a). The particles	
	are then advected to time $t_1$ (filled circles (b)), where they are mapped	
	to the master element with the inverse map $\mathbf{x}(\alpha)^{-1}$ ).	68
Figure 5.3:	With the fluid tracers integrated from time $t_0$ (b) to $t_1$ (a), then to $t_2$	
	(d), we can find the locations of the quadrature nodes at $t_2$ (e) and	
	$t_0$ (c) that map from the master element to the flow map at $t_1$ using	
	interpolation if we invert the interpolation matrix $I_{pq}$	72
Figure 5.4:	Velocity field and FTLE field for the periodic gyre flow	77
Figure 5.5:	Comparison of the convergence rates for the spectral element ap-	-
	proach and the finite difference approach.	/8
Figure 5.6:	velocity field for the periodic vortex now (a) and the FILE field for $T = 0.5$ (b)	70
Figure 5 7.	I = 0.5 (0)	19
Figure 5.7.	are present in the flow map, while the velocities and ETLE field	
	demonstrate spectral convergence	80
Figure 5.8:	Problem setup for square cylinder case.	81
Figure 5.9:	Convergence of the fluid solution can be seen in the vorticity plot	
C	using the most refined mesh, where the defects at the subdomain	
	boundaries are no longer visible. The finer mesh is four times the	
	resolution of the coarse mesh.	82

Figure 5.10:	Improvement in resolution of the FTLE is visible as the mesh be-	
	defacts in the sharp ETLE ridges in the near weke. The gradients in	
	the flow map are generally much greater than the gradients	83
Figure 5 11.	Uniform distribution of subdomains (a) and a cosine distribution	85
Figure 5.11.	where the subdomains are clustered near the ETLE ridges (b)	81
Eigung 5 12.	where the subdomains are clustered hear the FILE fluges (b) Convergence for the ETLE fields commuted from $\Phi_{\rm cluster}$ at time intervals	04
Figure 5.12.	Convergence for the FTLE fields computed from $\Phi_N$ at time intervals $T-2/3$ and $4$	85
Figure 5 13.	$\Gamma = 2, 5, \text{ and } + \dots +$	05
1 iguie 5.15.	and $\Theta_{V}$ (backward-time) at time intervals $T - 2/3$ and $4$	85
Figure 5 14.	Time-dependent gyre ETLE field for a time interval of $T - 4$	86
Figure 5.15:	Convergence for the FTLE fields computed from $\Phi_{y}$ at time intervals	00
1 iguie 5.15.	T = 3 and 4 for the time-dependent gyre flow	87
Figure 5 16.	Subdomain deformation at $T = 0.5$ (a) and $T = 2.5$ (b)	87
Figure 5.17:	The convergence of the RMS error of $\Phi_N$ at two times $T = 0.5$ and	07
1 iguie 5.17.	The convergence of the RWB error of $\Psi_N$ at two times, $T = 0.5$ and $T = 5.0$	88
Figure 5 18.	Dependency of maximum condition number on polynomial order	00
1 iguie 5.10.	and subdomain deformation. The refined case uses a uniform grid of	
	$\Delta r = 0.05$	89
Figure 5 19.	The convergence rates of the RMS error of the backward-time and	07
i iguie 5.17.	forward-time flow maps found using matrix inversion at two time	
	intervals. T	89
Figure 5.20:	The convergence rates of the backward and forward FTLE fields at	
0	time interval $T = 0.5$	90
Figure 5.21:	The error in the backward-time FTLE computed at two time intervals	
C	-T = 1.5 and $T = 2.5$ — using the method in Haller and Sapsis [1]	
	and this work.	90
Figure 5.22:	Problem setup for square cylinder case.	91
Figure 5.23:	Comparison of the forward-time FTLE field (a) and the backward-	
	time FTLE field (b). The time interval is $T = 10. \dots \dots \dots$	92
Figure 5.24:	Comparison of the forward-time FTLE field (a) to the log of the	
	condition number, $\ln(\kappa)$ , (b) and the parameter $\Delta \alpha$ (c). The time	
	interval is $T = 10$	96
Figure 5.25:	Comparison of the backward-time FTLE computed from a particle	
	trace in backward-time (a) and the backward-time FTLE computed	
	from the forward-time flow map using Lagrange interpolation (b).	
	The time interval is $T = 10.$	97
Figure 5.26:	The forward-time FTLE computed with a time interval $T = 10$ at	
	times $t_0 = 0$ (a), 1 (b), and 2 (c) with one particle trace	98
Figure 6.1:	The edges, $\Gamma_i$ , and faces, $\Sigma_i$ uniquely define a transfinite map for a	
	subdomain and are constructed from the Chebyshev-Lobatto nodes	
	with the same order as the subdomain discretization. $\ldots$ .	101
Figure 6.2:	Filter function, $\sigma$ at various filter orders	108

Figure 6.3:	Boundary conditions and geometry of the two-dimensional DNS of	100
	the flow over an airfoil at $Re = 20,000$ .	109
Figure 6.4:	Two-dimensional DNS of the flow over an airfoil at $Re = 20,000.$ ;	110
		110
Figure 6.5:	Attracting (blue) and repelling (red) LCS overlayed on vorticity.	
	Pressure-side vortices are highlighted with solid boxes while suction-	110
	side vortices are highlighted with dashed-boxes.	110
Figure 6.6:	Development of the forward-FILE field over time. The highlighted	
	regions defineate fluid that is eventually entrained in the vortex cores.	
	The dashed lines represent suction-side vortices and the solid lines	110
	represent pressure-side vortices.	112
Figure 6./:	Development of the backward-FILE field over time. The highlighted	110
<b>D</b> ' ( )	regions are the same as those in Figure 6.6.	113
Figure 6.8:	Location of the line from which data is extracted (a). As the time	
	interval T is increased, gradients in the flow map increase, leading	114
	to near discontinuities (b).	114
Figure 6.9:	The exponential filter is applied to the flow map, where it smooths	
	steep gradients (a). Steep gradients lead to Gibbs oscillations in the	
	unfiltered FTLE (b)-top. Application of an exponential filter with	
	$\gamma = 4$ removes the unphysical oscillations (b)-bottom	114
Figure 6.10:	Backward-time FTLE field computed from the forward-time flow	
	map at $T = 0.04$ (a), and the backward-time FTLE field determined	
<b>D</b> ' (11	with direct integration (b).	115
Figure 6.11:	Original coarse mesh (a), and the refined mesh constructed by split-	
	ting each element twice (b). Backward-time FTLE field computed	
	from the forward-time flow map at $T = 0.04$ with the coarse mesh	
<b>D</b>	(c), and the backward-time FILE field	117
Figure 6.12:	Forward-time flow maps for $T = 0.25$ starting at $t_0 = 0$ (a), $t_0 = 0.02$	
	(b), and $t_0 = 0.04$ (c). Each frame is computed from the same particle	
<b>F</b> ' (10)	trace that was initialized at $t_0 = 0$ .	118
Figure 6.13:	Flow map $\Phi^{x}$ , of the steady ABC flow (a) unfiltered, and (b) filtered	100
	with filter power $\gamma = 4$	120
Figure 6.14:	FILE field of the steady ABC flow using different filter powers. (1)	
	Unfiltered, (b) $\gamma = 16$ , (c) $\gamma = 8$ , and (d) $\gamma = 4$ . The single element	
	mesh is used with $N = 240$ .	121
Figure 6.15:	The effect of different filter strengths on the (a) flow map and the (b)	100
	FILE field.	122
Figure 6.16:	Effect of the filter on different polynomial orders for the $10 \times 10 \times 10$	100
	mesh (a) and the single element mesh (b) with filter power $\gamma = 4$	122
Figure 6.17:	Effect of the filter on the flow map for the $10 \times 10 \times 10$ mesh (a) and	100
<b>D</b>	the single element mesh (b) with filter power $\gamma = 4$	123
Figure 6.18:	FILE field of the steady ABC flow for the $10 \times 10 \times 10$ mesh (a)	100
	and the single element mesh (b) with different polynomial orders.	123

Figure 6.19:	Iso-surfaces of the FTLE field for the steady ABC flow	124
Figure 6.20:	Iso-contours of vorticity magnitude, colored by <i>u</i> -velocity of the	
	three-dimensional flow over an airfoil at $Re = 20,000.$	126
Figure 6.21:	Isocontours of vorticity magnitude (a) and FTLE (b)	127
Figure 6.22:	Slice of the magnitude of vorticity (a) and the FTLE field at the same	
	location (b). The FTLE field is computed over a time interval of	
	$T = 0.14$ , with a polynomial order of $N = 36$ and a filter with $\gamma = 4$ .	
	Vortex structures are highlighted.	127
Figure 6.23:	Vorticity magnitude at $x = 0.98$ (a), 1.03 (c), 1.18 (e), 1.33 (g), and	
	FTLE field at the same intervals (b), (d), (f), (h). The gray regions	
	at the bottom of (a) and (b) denote the surface of the airfoil. Vortex	
	structures are highlighted.	128
Figure 6.24:	Slice of the backward-time FTLE interpolated from the forward-time	
	flow map (a) and the backward-time FTLE field computed with the	
	direct method (b)	129
Figure 6.25:	Slice of the backward-time FTLE interpolated from the forward-time	
	flow map (a) and the backward-time FTLE field computed with the	
	direct method (b) for the refined mesh	130
Figure 6.26:	Forward-time flow maps for $T = 0.02$ starting at $t_0 = 0$ (a), $t_0 = 0.01$	
	(b), and $t_0 = 0.02$ (c). Each of these were computed from the same	
	particle trace that was initialized at $t_0 = 0$	131

### LIST OF TABLES

Table 4.1:	Results for the small domain and the large domain, $N = 12$ . CS	
	denotes the curved-sided element mesh and SS denotes the straight-	
	sided element mesh	39
Table 4.2:	Results for Mesh 1 and Mesh 2, $N = 12$	54

#### ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Dr. Gustaaf Jacobs, for all of his patience and wisdom over the course of my academic career, to Dr. David Kopriva for all of his insight and the provision of his DG code, and to my committee members for their contribution: Dr. Yuri Bazilevs, Dr. Daniel Tartakovsky, Dr. Fletcher Miller, and to my co-advisor, Dr. Sutanu Sarkar. I would like to acknowledge Solar Turbines for their generous financial support of this research. Thank you to my parents for their support and encouragement, and my wife and son for making everything worth it in the end.

Chapter 4, in part, has been submitted for publication of the material as it may appear in Theoretical and Computational Fluid Dynamics, 2015, Nelson, Daniel; Jacobs, Gustaaf and Kopriva, David, "Effect of Boundary Representation on Viscous, Separated Flows in a Discontinuous-Galerkin Navier-Stokes Solver". The dissertation author was the primary investigator and author of this paper.

Chapter 5, in part, is a reprint of the material as it appears in Journal of Computational Physics, vol. 295, 2015, Nelson, Daniel; Jacobs, Gustaaf, "DG-FTLE: Lagrangian Coherent Structures with Discontinuous-Galerkin Methods". The dissertation author was the primary investigator and author of this paper.

Section 5.3, in part, is a reprint of the material as it appears in The Proceedings of the ASME 2013 International Mechanical Engineering Congress and Exposition, 2013, Nelson, Daniel; Jacobs, Gustaaf, "Computation of Forward-Time Finite-Time Lyapunov Exponents Using Discontinuous-Galerkin Spectral Element Methods", The dissertation author was the primary investigator and author of this paper.

Chapter 6, in part, has been submitted for publication of the material as it may appear in Computers and Fluids, 2016, Nelson, Daniel; Jacobs, Gustaaf, "High-Order Visualization of Three-Dimensional Lagrangian Coherent Structures with DG-FTLE". The dissertation author was the primary investigator and author of this paper.

#### VITA

2009	Systems Engineering Intern, Space Exploration Technologies, Hawthorn, CA
2010	B. S. Aerospace Engineering, San Diego State University, San Diego, CA
2010	Aerospace Engineering Department Outstanding Graduate Award, San Diego State University
2010-2011	Mechanical Engineering Intern, NASA Jet Propulsion Laboratory, Pasadena, CA
2014	Aerodynamics Intern, Solar Turbines, San Diego, CA
2010-2015	Graduate Teaching Assistant, San Diego State University, San Diego, CA
2015	Ph. D. in Engineering Sciences (Mechanical and Aerospace Engi- neering), Joint Doctoral Program, San Diego State University and University of California San Diego, San Diego, CA

#### PUBLICATIONS

D. Nelson and G. Jacobs, "High-Order Visualization of Three-Dimensional Lagrangian Coherent Structures With DG-FTLE", *Computers and Fluids*, Submitted, 2016.

D. Nelson, G. Jacobs and D. Kopriva, "Effect of Boundary Representation on Viscous Flow Solutions with Discontinuous-Galerkin Methods", *Theoretical and Computational Fluid Dynamics*, In Review, 2015.

D. Nelson and G. Jacobs, "DG-FTLE: Lagrangian Coherent Structures with High-Order Discontinuous-Galerkin Methods", *Journal of Computational Physics*, 295, 2015.

D. Nelson and G. Jacobs, "Computation of Forward-Time Finite-Time Lyapunov Exponents Using Discontinuous-Galerkin Spectral Element Methods", Proceedings of the ASME 2013 International Mechanical Engineering Congress and Exposition, San Diego, CA, 2013.

#### PRESENTATIONS

D. Nelson and G. Jacobs, "DG-FTLE: Lagrangian Coherent Structures with High-Order Discontinuous-Galerkin Methods." Presented at the 13th Annual National Congress on Computational Mechanics. San Diego, CA. July 29, 2015.

D. Nelson, G. Jacobs, M. Sadek and P. Diamessis, "Lagrangian Coherent Structures in an Unstable Bottom Boundary Layer Under a Solitary Wave." Presented at the 67th Annual Meeting of the APS. San Francisco, CA. November, 2014.

D. Nelson and G. Jacobs, "High-order discontinuous-Galerkin simulations of flows over airfoils with curved boundaries." Presented at the 66th Annual Meeting of the APS. Pittsburg, PA. November, 2013.

D. Nelson and G. Jacobs, "Computation of Forward-Time Finite-Time Lyapunov Exponents Using Discontinuous-Galerkin Spectral Element Methods." Presented at the ASME 2013 International Mechanical Engineering Congress and Exposition. San Diego, CA. November 2013.

D. Nelson and G. Jacobs, "Lagrangian Coherent Structure Visualizations in High-Resolution DNS." Presented at the 65th Annual Meeting of the APS. San Diego, CA. November, 2012.

#### ABSTRACT OF THE DISSERTATION

#### High-Fidelity Lagrangian Coherent Structures Analysis and DNS with Discontinuous-Galerkin Methods

by

Daniel Alan Wendell Nelson

Doctor of Philosophy in Engineering Sciences (Mechanical and Aerospace Engineering)

University of California, San Diego, 2015 San Diego State University, 2015

Professor Gustaaf B. Jacobs, Co-Chair Professor Sutanu Sarkar, Co-Chair

High-fidelity numerical tools based on high-order Discontinuous-Galerkin (DG) methods and Lagrangian Coherent Structure (LCS) theory are developed and validated for the study of separated, vortex-dominated flows over complex geometry. The numerical framework couples prediction of separated turbulent flows using DG with time-dependent analysis of the flow through LCS and is intended for the development of separation control strategies for aerodynamic surfaces.

The compressible viscous flow over a NACA 65-(1)412 airfoil is solved with a

DG based Navier-Stokes solver in two and three dimensions. A method is presented in which high-order polynomial element edges adjacent to curved boundaries are matched to boundaries defined by non-smooth splines. Artificial surface roughness introduced by the piecewise-linear boundary approximation of straight-sided meshes results in the simulation of incorrect physics, including wake instabilities and spurious time-dependent modes. Spectral accuracy in the boundary approximation is not achieved for non-analytic boundary functions, particularly in high curvature regions.

An algorithm is developed for the high-order computation of Finite-Time Lyapunov Exponent (FTLE) fields simultaneously and efficiently with two and three dimensional DG-based flow solvers. Fluid tracers are initialized at Gauss-Lobatto quadrature nodes within an element and form the high-order basis for a flow map at later time. Gradients of the flow map and FTLE are evaluated with DG operators. Multiple flow maps are determined from a single particle trace by remapping the flow map to the quadrature nodes on deformed mesh elements. For large integration times, excessive subdomain deformation deteriorates the interpolating conditioning. The conditioning provides information on the fluid deformation and identifies subdomains that contain LCS. An exponential filter smooths the flow map in highly deformed areas. The algorithm is tested on several benchmarks and is shown to have spectral convergence.

The two and three-dimensional LCS field are analyzed for the unsteady flow over a NACA 65-(1)412 airfoil at a free-stream Reynolds number of Re=20,000 based on the chord length and a Mach number of 0.3. In two-dimensions, a Karman vortex street forms at the trailing edge. The three-dimensional vortex street breaks down to turbulence at the trailing edge.

## Chapter 1

## Introduction

## **1.1 Motivation**

Prediction and control of separated flow is an important field of study for the advancement of technologies that rely on fluid transport, such as aircraft and turbomachinery. Large pressure losses associated with separation on aerodynamic surfaces lead to performance losses. With Active Flow Control (AFC) of fluid separation, the operating range can be enhanced, operational lifetime extended and fuel consumption and emissions reduced.

Turbulent, separated flow, and the control thereof, are challenging for several reasons. First, a large separation of scales exists, ranging from small scale instabilities, to large scale vortex-dominated wake structures. Secondly, in these highly non-linear environments, large scales are very sensitive to small perturbations in the flow and modifications in the geometry, such as changes in the airfoil design. Lastly, non-linear instabilities complicate the dynamic and unsteady control of coherent structures and wave phenomena. These challenges pose high demands on prediction and control tools, requiring high-fidelity in most, if not all, stages of a control loop.

Industry standard Computational Fluid Dynamics (CFD) software typically relies on Reynolds-Averaged Navier-Stokes (RANS) solvers. However, RANS methods do not capture the sensitive, time-dependent structures that are critical in understanding flow separation and advancing AFC methods. Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES) are necessary to resolve the temporal evolution of unsteady fluid flow. To obtain, accurate, unsteady Navier-Stokes solutions efficiently, higher-order methods are most often used for DNS. The low dispersion and diffusion characteristics of high-order methods ensure long time accuracy and make them well suited for DNS and LES.

## **1.2 High-Order Methods on Complex Geometries**

Historically, efforts towards the DNS of the Navier-Stokes equations began with finite difference methods. Important early contributions include those of Courant *et al.* [2], Evans and Harlow [3], and Godunov [4]. These examples were restricted to relatively small numbers of degrees of freedom and did not address the simulation of turbulent flows. The DNS of full three-dimensional incompressible isotropic turbulence was first presented by Orszag and Patterson [5]. The simulation was performed using a Fourier spectral method with 32<sup>3</sup> degrees of freedom. As computing power and resources became more available, flows with higher Reynolds numbers and more complex turbulence problems were addressed with DNS. Most early efforts in the DNS of turbulent flows were focused on isotropic or free shear turbulence. For instance, the DNS of compressible turbulence in a shear flow was analyzed with a Fourier spectral method by Sarkar *et al.* [6].

The high-fidelity numerical analysis of turbulent wall bounded flows was begun in the 1980s with the LES of the incompressible plane channel flow [7]. This was followed by the DNS of the incompressible plane channel flow by Kim *et al.* [8]. The first DNS of turbulent flow on curved geometries was studied by Moser and Moin [9], where the incompressible turbulent flow through a curved channel was simulated. The channel flow simulations of [8, 9] used a Fourier spectral method in the streamwise and spanwise directions and a Chebyshev spectral method in the wall normal directions.

As advancements in the DNS of turbulent flows began to include more complex geometrical configurations, spectral methods were used comparatively infrequently since few spectral methods were available that could accommodate complex geometries. For instance, much of the early work on the DNS of flow separation focused on the separated flow over a backward-facing step using finite differences and finite volume methods [10, 11].

Patera [12] introduced the spectral element method, thus combining the superior convergence qualities of spectral methods with the geometric flexibility afforded by finite volume methods. Further developments led to numerous studies on separated flows using spectral element methods. Kaiktsis *et al.* [13] implemented a spectral element approach for the backward-facing step, however Gresho *et al.* [14] showed that the results in [13] were under-resolved. Mittal and Balachandar presented a single domain spectral method on a curvilinear, boundary-fitted mesh for the turbulent separated flow over elliptical cylinders. Bassi and Rebay [15, 16] demonstrated spectral convergence can be achieved on geometries with curved edges using spectral element methods. Jacobs [17] revisited the separated, turbulent flow over a backward-facing step using a spectral element method and found good agreement with the literature. Hughes *et al.* [18] proposed methods for mesh refinement on complex curved geometries. Sun *et al.* [19] demonstrated a multidomain spectral method on the inviscid and viscous flow over a sphere.

Several aspects of high-order Discontinuous-Galerkin (DG) spectral element methods provide a robust approach to the numerical analysis of separated flows in non-trivial boundary configurations. We refer to [20, 21] for comprehensive overviews of spectral element methods. By mapping the solution from an irregular element to a regular master element, high-order orthogonal polynomials are used to construct spectral methods on unstructured meshes. Dubiner [22] introduces a method in which the solution is mapped to triangular elements, making it consistent with triangular mesh generators used frequently in the industry. Sherwin and Karniadakis [23] extended this idea to three-dimensional elements. Persson *et al.* [24] describe a scheme for solving the time dependent Navier-Stokes on deformable meshes with a high-order DG method. Kopriva [25, 26] presents a quadrilateral based scheme that projects nonconforming element boundaries to a mortar space, providing a flexible route for local element refinement. Element boundary fluxes are treated locally so that the solution in each element only depends on adjacent element. In this way, parallelization is easily implemented and efficient. Since high-order polynomials are used to approximate the solution on the reference element, polynomial element boundary representations are implemented naturally with appropriate treatment of transfinite mappings and metric terms [27].

Although DG methods are naturally suited for numerical analysis of flows on curved geometries, the use of curved elements introduces some challenges. At least one of the edges of the elements neighboring physical boundaries are not linear. The Jacobian for curved elements is not constant throughout the element, and the mapping to computational space is non-linear. If the inner products are exactly integrated, the cost of storing and inverting the mass matrix is substantial, leading to higher computational and memory costs for curved-sided elements when compared to straight-sided elements.

The added costs of implementing curved-sided elements in fully integrated DG methods has motivated research into improving the efficiency of solvers using irregular elements. Lou et al. [28] develop a fast multigrid DG method for curved elements. Kirvodonova and Berger [29] present a method for efficiently approximating curved

boundaries in DG methods by using straight-sided meshes and matching the angle of the tangential velocity vectors and surface normals to the exact geometric boundary. Zhang and Tan [30] propose a similar method for problems with curved material interfaces. Toulorge and Desmet [31] show that the boundary approximation of [29] does not significantly improve accuracy of computations in which aeroacoustic effects are important. Salman et al. [32] argue that when the flow region of interest is far from any curved boundaries, it is sufficient to use straight-sided elements at those curved boundaries, thus avoiding the issue.

An additional challenge for DG approximations of complex edges is that engineering geometries are frequently defined by non-smooth parametric functions, such as splines and Bezier curves [33, 34]. Discontinuities in the higher derivatives of non-smooth curves are difficult to accurately approximate with smooth functions (*e.g.* polynomials), as Gibbs-like phenomena near the discontinuities are not trivial to overcome with high-order approximations [35, 36, 37, 38, 39, 40, 41].

Although implementing DG methods on curved bodies involves several challenges, errors introduced by the approximate geometry may significantly affect the large scale characteristics of a flow. Poor quality surface approximations introduce variations in surface height as compared to the smooth curved boundaries that they approximate, numerically emulating surface roughness effects. The sensitivity of separated flows to surface roughness upstream of the separation location is well documented in the literature [42, 43, 44, 45]. Small-scale roughness elements induce perturbations in the boundary layer that destabilize the separated shear layer and lead to the development of turbulence downstream, which has a significant effect on the aerodynamic characteristics of the flow. Low numerical diffusion makes high-order methods particularly susceptible to the perturbations induced by small-scale geometric imperfections since these perturbations are not attenuated by numerical dissipation.

## **1.3 Lagrangian Coherent Structures and Separated Flow**

The control of vortex dominated wakes induced by flow separation starts on the solid boundary where the flow breaks away from the geometry. Flow separation, hence, has been a major fields of study in aerodynamics in general, and for flow control specifically. Dating back 100 years ago, Prandtl [46] defined flow separation by deriving exact Eulerian criteria for steady flow separation in two dimensions. Since then, several authors have developed both Eulerian and Lagrangian criteria for unsteady separation in two and three-dimensions. Rott [47], Moore [48] and Sears and Telionis [49] proposed criteria for moving separation surfaces, assuming a separation speed is known. Cowley et al. [50] numerically analyzed unsteady separation in the Lagrangian frame, showing that fluid ejects from the wall along spikes. Shariff *et al.* [51] proposed a Lagrangian separation theory that defined separation in incompressible periodic flows as a fixed point with an unstable material manifold. More recently, criteria for unsteady separation in two and three dimensions were derived that relates separation lines to unstable manifolds in compressible flows [52, 53, 54]. Numerical validations of this theory [55, 56] has shown that Lagrangian separation surfaces evolve into complex manifolds that align with important transport barriers in the flow known as Lagrangian Coherent Structures (LCS).

To identify LCS in time-dependent fluid flows, Haller and Yuan [57] and Haller [58], introduce a theory which associate LCS with regions of locally maximal repulsion or attraction. They show that the finite-time Lyapunov exponent (FTLE) field provides a quantitative measure of the repulsion rate in forward-time and the attraction rate in backward-time in a fluid. Subsequent work has been devoted to rigorously developing a precise definition of LCS manifolds and the role they play in fluid motion. Shadden *et al.* [59] define LCS as ridges in the FTLE field. Lekien [60] extends this to higher dimensions. A range of applications involving transport phenomena (*e.g.* [61, 62, 63, 64, 65, 66])

have shown this method to be useful in uncovering the physical mechanisms driving momentum and particle transport. More recently, Haller [67, 68] has formulated a variational approach that generalizes the definition of LCS to be co-dimension one manifolds identified as minimal geodesics of the metric induced by the Cauchy-Green strain tensor. A robust method for computing hyperbolic LCS with this variational approach is developed by Farazmand and Haller [69]. Blazevski and Haller [70] have extended application of the variational theory to three-dimensions, developing methods that extract hyperbolic and elliptic LCS from three-dimensional flow fields.

The determination of the FTLE field in vortex-dominated flows and complex geometry is not trivial. In areas of large fluid deformation (*e.g.* near solid walls or in wakes), greater resolution or a more refined approximation is required to accurately determine the FTLE. However, there has been relatively little work towards improving the accuracy of the near-wall FTLE field. Typically, FTLE algorithms use low-order Cartesian grids such that the FTLE near complex boundaries is poorly resolved. Salman *et al.* [32] emphasize the importance of high-fidelity LCS identification in near wall regions, particularly in the context of flow control. They solve the Navier-Stokes equations with a high-order spectral element method on unstructured grids coupled with a low-order boundary-fitted finite element method to compute the FTLE field. Cardwell and Mohseni [71] and Lipinski *et al.* [72] detail the identification of LCS in the two-dimensional flow over an airfoil. Others have introduced boundary-fitted methods with the aim of improving the resolution of the near-wall FTLE field using Adaptive Mesh Refinement (AMR) [73, 74]. However, to the best of the authors' knowledge, the first high-order method for determining the FTLE field on unstructured grids was introduced in [75, 76].

Much of the work with LCS in three-dimensions has focused on flows related to bio-fluid mechanics [77, 63] and geophysical fluid dynamics [78, 79, 80]. In contrast, relatively few studies focus on geometries with industrial application, such as bluff-bodies

and aerodynamic control surfaces. Garth *et al.* [81] extract ridges from the FTLE field of three dimensional flows around high-speed trains and delta wings. Bourgeois *et al.* [82] uses the FTLE field of the three-dimensional phase-averaged wake of a wall-mounted square cylinder to identify high-strain structures and correlates them with incoherent turbulence production. Sadlo and Peikert [83] visualize LCS in three-dimensions in a Pelton turbine and a hydro-powerplant intake using an AMR method. However, studies of the three-dimensional flow over airfoils with LCS methods are absent from the literature, to the best of the authors' knowledge.

A significant issue with the integration of LCS theory into flow control problems is the high computational costs associated with advecting dense grids of fluid tracers over long time intervals. Recent work has focused on improving efficiency and accuracy of LCS identification algorithms (*e.g.* [81, 74, 84]). The requirement for high-fidelity resolution of ridges in the FTLE field is important, particularly in the area of flow control, as pointed out by Salman et al. [32]. However, relatively few authors have explored the application of LCS theory on flow control problems [32, 65].

In this thesis, we focus on the development of high-fidelity CFD methods for simulation of turbulent flows and high-fidelity analysis of Lagrangian separation and coherent structures in separated wakes — all components that are integral to the development and testing of successful AFC methods for separated fluid flows.

## **1.4 Outline and Contributions**

The most important contribution of this dissertation is the development of the numerical tools necessary to apply Lagrangian analysis to flow control analysis, including methods specifically designed to extract LCS with high-order accuracy from turbulent separated flows simultaneously with the DNS or LES of the underlying flow on complex

geometries with curved surfaces. The following two chapters of this work describe the governing fluid conservation laws and the DG spectral element method used to solve them. The next chapter introduces LCS and details the novel numerical algorithm developed in this work to compute FTLE fields with high-order DG operators. Chapters five and six present the DNS of a NACA 65-(1)412 airfoil at low Reynolds number in two an three dimensions.

The major contributions of this work can be summarized as follows:

- An algorithm is developed that computes the Cauchy-Green Strain (CGS) tensor using high-order accurate DG differential operators, thus extending high-order accuracy to Lagrangian quantities based on the CGS tensor, such as FTLE fields and LCS. The algorithm is consistent in implementation and accuracy with DG fluid solvers and can be integrated directly into the fluid solver to determine FTLE fields concurrently with the fluid solution. Furthermore, the algorithm relies on DG elemental discretization, which is boundary fitted and approximates curved boundaries with high-order accuracy.
- The algorithm is extended to determine multiple flow maps, both backward and forward in time, from a single particle trace using high-order interpolation on orthogonal bases constructed on the deformed subdomains formed by flow maps at later times. By doing so, redundant particle integrations are avoided and the efficiency of the algorithm is greatly improved. Additionally, the conditioning of the interpolation matrix constructed from the deformed subdomains is shown to provide additional quantitative information on the deformation of the subdomains and flow field.
- A method for implementing high-order curved boundary approximations on complex geometries is developed that allows the flexibility to match the order of the elemental boundary curves to the order of the fluid solver. It is shown that high-

order boundary representation is necessary in the context of separated flows to simulate the correct physics, and that small variations in the boundary approximation induce measurable, and in some cases significant, variations in the global flow characteristics.

• The DNS of a NACA 65-(1)412 airfoil in two and three dimensions is performed at a Reynolds number of Re = 20,000. Few numerical studies have been performed in this Reynolds number range. It is shown that at Re = 20,000, the wake is an asymmetric Karman vortex street in two dimensions. In three dimensions, the vortex street breaks town into a network of interconnected spanwise and streamwise vortices due to an elliptic instability in the vortex shedding zone at the trailing edge of the airfoil.

# **Chapter 2**

# **Governing Model**

## 2.1 The Navier-Stokes Equations

## 2.1.1 Dimensional Form

We consider three dimensional, Newtonian, compressible flows that are governed by

$$\mathbf{Q}_t^* + \nabla \cdot \mathbf{F}^* = 0, \tag{2.1}$$

where the flux tensor **F** can be separated into advective (*a*) and viscous (*v*) fluxes, such that

$$\nabla \cdot \mathbf{F}^* = \mathbf{F}_x^{a*} + \mathbf{G}_y^{a*} + \mathbf{H}_z^{a*} - (\mathbf{F}_x^{v*} + \mathbf{G}_y^{v*} + \mathbf{H}_z^{v*}).$$
(2.2)

The conserved variables and advected fluxes are given by

$$\mathbf{Q}^{*} = \begin{bmatrix} \rho^{*} \\ \rho^{*} u^{*} \\ \rho^{*} v^{*} \\ \rho^{*} w^{*} \\ \rho^{*} e^{*} \end{bmatrix}, \qquad (2.3)$$

$$\mathbf{F}^{a*} = \begin{bmatrix} \rho^* u^* \\ p^* + \rho^* u^{*2} \\ \rho^* u^* v^* \\ \rho^* u^* w^* \\ u^* (\rho^* e^* + p^*) \end{bmatrix}, \quad \mathbf{G}^{a*} = \begin{bmatrix} \rho^* v^* \\ \rho^* v^* u^* \\ p^* + \rho^* v^{*2} \\ \rho^* v^* w^* \\ v^* (\rho^* e^* + p^*) \end{bmatrix}, \quad \mathbf{H}^{a*} = \begin{bmatrix} \rho^* w^* \\ \rho^* w^* u^* \\ \rho^* w^* v^* \\ p^* + \rho^* w^{*2} \\ w^* (\rho^* e^* + p^*) \end{bmatrix}.$$
(2.4)

The sum of the internal and kinetic energy is  $\rho^* e^* = \rho^* c_v T^* + \rho^* (u^{*2} + v^{*2} + w^{*2})/2$ . The viscous fluxes are

$$\mathbf{F}^{\nu*} = \begin{bmatrix} 0 \\ \tau^*_{xx} \\ \tau^*_{yx} \\ \tau^*_{yx} \\ u^* \tau^*_{xx} + \nu^* \tau^*_{yx} + w^* \tau^*_{zx} + \kappa^* T^*_x \end{bmatrix}, \qquad (2.5)$$

$$\mathbf{G}^{\nu*} = \begin{bmatrix} 0 \\ \tau_{xy}^{*} \\ \tau_{yy}^{*} \\ \tau_{zy}^{*} \\ u^{*}\tau_{xy}^{*} + \nu^{*}\tau_{yy}^{*} + w^{*}\tau_{zy}^{*} + \kappa^{*}T_{y}^{*} \end{bmatrix}, \qquad (2.6)$$
$$\mathbf{H}^{\nu*} = \begin{bmatrix} 0 \\ \tau_{xz}^{*} \\ \tau_{yz}^{*} \\ \tau_{zz}^{*} \\ u^{*}\tau_{xz}^{*} + \nu^{*}\tau_{yz}^{*} + w^{*}\tau_{zz}^{*} + \kappa^{*}T_{z}^{*} \end{bmatrix}. \qquad (2.7)$$

The shear stresses are

$$\begin{aligned} \tau_{xx}^{*} &= 2\mu^{*}(u_{x}^{*} - (u_{x}^{*} + v_{y}^{*} + w_{z}^{*})/3), \\ \tau_{yy}^{*} &= 2\mu^{*}(v_{y}^{*} - (u_{x}^{*} + v_{y}^{*} + w_{z}^{*})/3), \\ \tau_{zz}^{*} &= 2\mu^{*}(w_{z}^{*} - (u_{x}^{*} + v_{y}^{*} + w_{z}^{*})/3), \\ \tau_{yx}^{*} &= \tau_{xy}^{*} &= \mu^{*}(v_{x}^{*} + u_{y}^{*}), \\ \tau_{zx}^{*} &= \tau_{xz}^{*} &= \mu^{*}(w_{x}^{*} + u_{z}^{*}), \\ \tau_{zy}^{*} &= \tau_{yz}^{*} &= \mu^{*}(w_{y}^{*} + v_{z}^{*}). \end{aligned}$$

$$(2.8)$$

An ideal gas is assumed so that the constitutive equation relating the pressure, density and temperature is the ideal gas law,

$$p^* = \rho^* R T^*, \tag{2.9}$$

where R is the particular gas constant.

### 2.1.2 Non-Dimensional Form

The dimensional equations are made non-dimensional by normalizing all of the variables by the reference length  $(L_f)$ , density  $(\rho_f)$ , velocity  $(U_f)$ , temperature  $(T_f)$ , and viscosity  $(\mu_f)$  and thermal diffusivity  $(\kappa_f)$ . The non-dimensional variables become

$$x = \frac{x^{*}}{L_{f}} \qquad u = \frac{u^{*}}{U_{f}}$$

$$y = \frac{y^{*}}{L_{f}} \qquad v = \frac{v^{*}}{U_{f}}$$

$$z = \frac{z^{*}}{L_{f}} \qquad w = \frac{w^{*}}{U_{f}}$$

$$t = \frac{t^{*}U_{f}}{L_{f}} \qquad T = \frac{T^{*}}{T_{f}}$$

$$\mu = \frac{\mu^{*}}{\mu_{f}} \qquad \rho = \frac{\rho^{*}}{\rho_{f}}$$

$$\kappa = \frac{\kappa^{*}}{\kappa_{f}} \qquad p = \frac{p^{*}}{\rho_{f}U_{f}^{2}} \qquad (2.10)$$

The non-dimensional equation of state is  $p = \rho T / \gamma M_f^2$ . The reference Reynolds, Mach and Prandtl numbers are defined by  $Re_f = \rho_f U_f L_f / \mu_f$ ,  $M_f = U_f / (\gamma RT_f)^{1/2}$  and  $Pr = c_p \mu_f / \kappa_f = 0.72$ , respectively. Viscosity,  $\mu$ , and thermal diffusivity,  $\kappa$ , are computed by Sutherland law [85],

$$\mu = \kappa = \frac{(1+R_T)T^{3/2}}{T+R_T},$$
(2.11)

where  $R_T$  is the ratio of the Sutherland constant and the reference temperature,  $R_T = S/T_f = 110/200$ . The ratio of specific heats,  $\gamma$ , is taken to be 1.4. Under this normalization, the Navier-Stokes equations are written

$$\mathbf{Q} + \mathbf{F}_x^a + \mathbf{G}_y^a + \mathbf{H}_z^a - \frac{1}{\operatorname{Re}_f} (\mathbf{F}_x^v + \mathbf{G}_y^v + \mathbf{H}_z^v) = 0.$$
(2.12)

The solution vector is

$$\mathbf{Q} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\rho} \boldsymbol{u} \\ \boldsymbol{\rho} \boldsymbol{v} \\ \boldsymbol{\rho} \boldsymbol{w} \\ \boldsymbol{\rho} \boldsymbol{e} \end{bmatrix}, \qquad (2.13)$$

and the advective fluxes are

$$\mathbf{F}^{a} = \begin{bmatrix} \rho u \\ p + \rho u^{2} \\ \rho uv \\ \rho uw \\ u(\rho e + p) \end{bmatrix}, \quad \mathbf{G}^{a} = \begin{bmatrix} \rho v \\ \rho vu \\ p + \rho v^{2} \\ \rho vw \\ v(\rho e + p) \end{bmatrix}, \quad \mathbf{H}^{a} = \begin{bmatrix} \rho w \\ \rho wu \\ \rho wu \\ \rho wv \\ p + \rho w^{2} \\ w(\rho e + p) \end{bmatrix}. \quad (2.14)$$

The non-dimensional total energy is  $\rho e = p/(\gamma - 1) + \rho(u^2 + v^2 + w^2)/2$ . The viscous fluxes are

$$\mathbf{F}^{\nu} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{yx} \\ \tau_{zx} \\ u\tau_{xx} + \nu\tau_{yx} + w\tau_{zx} + \frac{\kappa}{(\gamma - 1)\operatorname{Pr} \operatorname{M}_{f}^{2}} T_{x} \end{bmatrix}, \qquad (2.15)$$

$$\mathbf{G}^{\nu} = \begin{bmatrix} 0 & & \\ \tau_{xy} & & \\ \tau_{yy} & & \\ \tau_{zy} & & \\ u\tau_{xy} + v\tau_{yy} + w\tau_{zy} + \frac{\kappa}{(\gamma-1)\operatorname{Pr} M_f^2} T_y \end{bmatrix}, \quad (2.16)$$
$$\mathbf{H}^{\nu} = \begin{bmatrix} 0 & & \\ \tau_{xz} & & \\ \tau_{yz} & & \\ \tau_{yz} & & \\ \tau_{zz} & & \\ u\tau_{xz} + v\tau_{yz} + w\tau_{zz} + \frac{\kappa}{(\gamma-1)\operatorname{Pr} M_f^2} T_z \end{bmatrix}, \quad (2.17)$$

where the terms of the non-dimensional stress tensor,  $\boldsymbol{\tau},$  are

$$\tau_{xx} = 2\mu(u_x - (u_x + v_y + w_z)/3),$$
  

$$\tau_{yy} = 2\mu(v_y - (u_x + v_y + w_z)/3),$$
  

$$\tau_{zz} = 2\mu(w_z - (u_x + v_y + w_z)/3),$$
  

$$\tau_{yx} = \tau_{xy} = \mu(v_x + u_y),$$
  

$$\tau_{zx} = \tau_{xz} = \mu(w_x + u_z),$$
  

$$\tau_{zy} = \tau_{yz} = \mu(w_y + v_z).$$
(2.18)
## 2.2 Lagrangian Coherent Structures

#### 2.2.1 Trajectories and Flow Maps

Following Shadden et al. [59], we consider dynamical systems which are solutions to the initial value problem given by

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{v}(\mathbf{x}(t), t) \\ \mathbf{x}_0 = \mathbf{x}(t_0) \end{cases}$$

This linear, first order ordinary differential equation (ODE) defines the path that the particle takes in space, as a function of time, subject to the velocity field given by  $\mathbf{v}(\mathbf{x}(t),t)$ . The solution to (2.19) is

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{t_0}^t \mathbf{v}(\mathbf{x}(\tau), \tau) \mathrm{d}\tau$$
 (2.19)

We can rewrite (2.19), including the parameters  $\mathbf{x}_0$  and  $t_0$  in the function argument, and expressing the time in terms of the time interval  $T \equiv t - t_0$ ,

$$\mathbf{x}(T;\mathbf{x}_0,t_0) = \mathbf{x}_0 + \int_{t_0}^{t_0+T} \mathbf{v}(\mathbf{x}(\tau),\tau) \mathrm{d}\tau$$
(2.20)

The flow map is given by

$$\phi(\mathbf{x}_0, t_0; T) \equiv \mathbf{x}(\mathbf{x}_0, t_0; T).$$
(2.21)

The flow map is a vector field that is a function of the space-time given by  $(\mathbf{x}_0, t_0)$  subject to the parameter *T*, which is the time interval that the flow map is determined over. The space-time  $(\mathbf{x}_0, t_0)$  is known as the *extended phase space*.

We denote the extended phase space as  $\mathcal{X} \subset \{\mathbb{R}^n, t\}$ . We relate the flow map with



Figure 2.1: Schematic of a repelling LCS (a) and an attracting LCS (b).

the equations for particle trajectories by defining the the flow map to be the set of all values of  $\mathbf{x}$  constrained by (2.20) within the extended phase space X, or

$$\phi(\mathbf{x}_0, t_0; T) := \left\{ \mathbf{x} \subset \mathbb{R}^n : \mathbf{x} = \mathbf{x}_0 + \int_{t_0}^{t_0 + T} \mathbf{v}(\mathbf{x}(\tau), \tau) \mathrm{d}\tau \quad \forall \{\mathbf{x}_0, t_0\} \in \mathcal{X} \right\}.$$
(2.22)

#### 2.2.2 Finite-Time Lyapunov Exponent Field

Haller [58] originally defined LCS as material lines with locally the strongest *normal* repulsion, in forward-time and the strongest attraction, in backward-time. Conceptually, LCS are the material surfaces where, in forward-time, a fluid parcel will stretch normal to the surface as it is advected by the flow (Fig. 2.1a). Attracting LCS, on the other hand, tend to attract nearby fluid parcels, which are then stretched along the length of the LCS (Fig. 2.1b).

Ridges in the FTLE field gives locally the strongest stretching and Shadden *et al.* [59] showed that under certain circumstances, these ridges are nearly Lagrangian. Hence, it is proposed in [59] that LCS be defined by ridges in the FTLE field. However, more recently, Haller [67] demonstrated that, in general, ridges in the FTLE field do not always coincide with LCS since there is no guarantee that the stretching is normal to the ridge. In light of this, the FTLE field provides a useful diagnostic for identifying candidate LCS and other important structures (*e.g.* shear layers) in unsteady flows. Furthermore, computation of the FTLE field from the Cauchy-Green strain tensor is efficient as compared with extracting LCS using geodesic theory [69, 68], and thus is well suited for identifying candidate LCS simultaneously with the solution of the Navier-Stokes equations using CFD applications. Therefore, in this thesis, we will limit our analysis to the FTLE field.

Given a dynamical system with a flow map defined by (2.21), we can define the *linearized flow map*,  $\bar{\phi}$ , using the Taylor expansion of (2.21),

$$\phi(\mathbf{x}_0, t_0; T) \approx \phi(\mathbf{y}_0, t_0; T) + \frac{\partial \phi(\mathbf{y}_0, t_0; T)}{\partial \mathbf{x}_0} \left(\mathbf{x}_0 - \mathbf{y}_0\right) + \mathcal{O}\left(\left(\mathbf{x}_0 - \mathbf{y}_0\right)^2\right).$$
(2.23)

The linearized flow map describes the mapping of perturbations in  $\mathbf{x}_0$  subject to the time interval *T*. In the limit that  $\mathbf{y}_0 \to \mathbf{x}_0$ , we define the perturbation

$$\delta \mathbf{x}_0 \equiv \lim_{\mathbf{y}_0 \to \mathbf{x}_0} |\mathbf{x}_0 - \mathbf{y}_0|. \tag{2.24}$$

Similarly, we define

$$\delta \phi \equiv \lim_{\mathbf{y}_0 \to \mathbf{x}_0} |\phi(\mathbf{x}_0, t_0; T) - \phi(\mathbf{y}_0, t_0; T)|.$$
(2.25)

With (2.23), (2.24) and (2.25), the linearized flow map is defined by the first-order Taylor expansion,

$$\bar{\phi}(\mathbf{x}_0, t_0; T) : \delta\bar{\phi} = \frac{\partial\phi(\mathbf{x}_0, t_0; T)}{\partial\mathbf{x}_0} \delta\mathbf{x}_0 = \nabla_{\mathbf{x}_0}\bar{\phi}\,\delta\mathbf{x}_0.$$
(2.26)

Henceforth, we will assume  $\phi(\mathbf{x}_0, t_0; T) = \overline{\phi}(\mathbf{x}_0, t_0; T)$ , and so only use  $\phi$  in the remaining analysis.



**Figure 2.2**: Maximal stretch of an infinitesimal spherical fluid parcel as determined by projecting the eigenvectors and eigenvalues of **C** onto an ellipsoid *via* the operator norm.

Since  $\nabla_{\mathbf{x}_0} \phi$  is a linear operator on  $\delta \mathbf{x}_0$ , we can use the operator norm to find the maximal amount, max  $|\delta \phi|$ , that  $\nabla_{\mathbf{x}_0} \phi$  will stretch an initial line element of length  $|\delta \mathbf{x}_0|$ ,

$$\max |\delta \phi| = \|\nabla_{\mathbf{x}_0} \phi\| |\delta \mathbf{x}_0|. \tag{2.27}$$

It is important to note that

$$\|\nabla_{\mathbf{x}_0}\phi\| = \sqrt{\lambda_{max}\left(\nabla_{\mathbf{x}_0}\phi^*\nabla_{\mathbf{x}_0}\phi\right)} = \sqrt{\lambda_{max}(\mathbf{C})},\tag{2.28}$$

where **C** is the (right) Cauchy-Green strain tensor and  $\lambda_{max}$  (**C**) is the maximal eigenvalue of **C**. The operator norm serves to project the maximal and minimal eigenvalues of **C** to the semi-major and semi-minor axes of an ellipsoid, respectively, as in Figure 2.2. The orientation of the ellipsoid is such that the axes are aligned with the eigenvectors of **C**, **N**<sub>*i*</sub>.

By assuming that max  $|\delta \phi|$  diverges exponentially in time, we can determine the

rate of this divergence, or "stretching", by rewriting (2.27):

$$\max |\delta \phi| = |\delta \mathbf{x}_0| \, e^{\sigma |T|} \tag{2.29}$$

Thus, by construction, we define the FTLE ( $\sigma$ ) as:

$$\sigma(\mathbf{x}_0, t_0; T) = \frac{1}{|T|} \ln \|\nabla_{\mathbf{x}_0} \phi\|$$
(2.30)

Or equivalently:

$$\sigma(\mathbf{x}_0, t_0; T) = \frac{1}{|T|} \ln \sqrt{\lambda_{max}(\mathbf{C})}$$
(2.31)

It is clear from (2.29) and (2.30) that the FTLE is a measure of the maximum rate of stretching in the fluid. From a Lagrangian point of view, neighboring trajectories that diverge exponentially over the time interval *T* lie at the interface of a fluid boundary where, by definition, fluid particles do not cross.

# **Chapter 3**

# The Discontinuous-Galerkin Spectral Element Method

The Navier-Stokes equations are discretized with a dG spectral element method. For detailed descriptions of dG methods, we refer to [25, 26, 86]. We begin by dividing the computational domain,  $\Omega$ , into a set of non-overlapping quadrilateral subdomains,  $D_k$ , such that

$$\Omega = \bigcup_{k=1}^{K} D_k. \tag{3.1}$$



**Figure 3.1**: Mapping from physical space (x, y, z) to the unit cube in computational coordinates  $(\xi, \eta, \zeta)$ . The mapping is uniquely defined by the edge polynomial representation  $\Gamma_i$ , the face polynomials  $\Sigma_i$  and the corner coordinates  $\mathbf{x}_i$ .

The subdomains,  $D_k$ , are mapped to the unit cube as in Fig. 3.1 using a transfinite map given by the linear blending formula

$$\begin{aligned} \mathbf{x}(\xi,\eta,\zeta) &= -(1-\eta)\Sigma_{1}(\xi,\zeta) - \eta\Sigma_{2}(\xi,\zeta) - (1-\zeta)\Sigma_{3}(\xi,\eta) \\ &-\xi\Sigma_{4}(\eta,\zeta) - \zeta\Sigma_{5}(\xi,\eta) - (1-\xi)\Sigma_{6}(\eta,\zeta) \\ &+ (1-\eta)(1-\zeta)\Gamma_{1}(\xi) + \xi(1-\eta)\Gamma_{2}(\zeta) \\ &+ (1-\eta)\zeta\Gamma_{3}(\xi) + (1-\xi)(1-\eta)\Gamma_{4}(\zeta) \\ &+ \eta(1-\zeta)\Gamma_{5}(\xi) + \xi\eta\Gamma_{6}(\zeta) + \eta\zeta\Gamma_{7}(\xi) + (1-\xi)\eta\Gamma_{8}(\zeta) \\ &+ (1-\xi)(1-\zeta)\Gamma_{9}(\eta) + \xi(1-\zeta)\Gamma_{10}(\eta) + \xi\zeta\Gamma_{11}(\eta) + (1-\xi)\zeta\Gamma_{12}(\eta) \\ &- \mathbf{x}_{1}(1-\xi)(1-\eta)(1-\zeta) - \mathbf{x}_{2}\xi(1-\eta)(1-\zeta) \\ &- \mathbf{x}_{3}\xi\eta(1-\zeta) - \mathbf{x}_{4}(1-\xi)\eta(1-\zeta) \\ &- \mathbf{x}_{5}(1-\xi)(1-\eta)\zeta - \mathbf{x}_{6}\xi(1-\eta)\zeta - \mathbf{x}_{7}\xi\eta\zeta - \mathbf{x}_{8}(1-\xi)\eta\zeta. \end{aligned}$$
(3.2)

The functions  $\Gamma_i$  are parametric curves that represent the subdomain edges, while the

functions  $\Sigma_i$  are two-dimensional parametric surfaces representing the subdomain faces. Under this mapping, the governing equation (2.1) becomes

$$\tilde{\mathbf{Q}}_t + \nabla \cdot \tilde{\mathbf{F}} = 0, \tag{3.3}$$

where,

$$\tilde{\mathbf{Q}} = J\mathbf{Q},$$
  
$$\tilde{\mathbf{F}}_i = (\mathbf{a}_j \times \mathbf{a}_k)_i \cdot \mathbf{F}.$$
 (3.4)

In three-dimensions, the vectors  $\mathbf{a}_i$  represent the metric terms and are given by  $\mathbf{a}_i = \partial \mathbf{x}/\partial \xi_i$ . For instance, evaluating  $\mathbf{a}_1 = \partial \mathbf{x}/\partial \xi_1 = \partial \mathbf{x}/\partial \xi$ , we have

$$\begin{aligned} \frac{\partial \mathbf{x}}{\partial \xi} &= -(1-\eta)\partial_{\xi}\Sigma_{1}(\xi,\zeta) - \eta\partial_{\xi}\Sigma_{2}(\xi,\zeta) - (1-\zeta)\partial_{\xi}\Sigma_{3}(\xi,\eta) \\ &- \Sigma_{4}(\eta,\zeta) - \zeta\partial_{\xi}\Sigma_{5}(\xi,\eta) + \Sigma_{6}(\eta,\zeta) \\ &+ (1-\eta)(1-\zeta)\Gamma_{1}'(\xi) + (1-\eta)\Gamma_{2}(\zeta) \\ &+ (1-\eta)\zeta\Gamma_{3}'(\xi) - (1-\eta)\Gamma_{4}(\zeta) \\ &+ \eta(1-\zeta)\Gamma_{5}'(\xi) + \eta\Gamma_{6}(\zeta) + \eta\zeta\Gamma_{7}'(\xi) - \eta\Gamma_{8}(\zeta) \\ &- (1-\zeta)\Gamma_{9}(\eta) + (1-\zeta)\Gamma_{10}(\eta) + \zeta\Gamma_{11}(\eta) - \zeta\Gamma_{12}(\eta) \\ &+ \mathbf{x}_{1}(1-\eta)(1-\zeta) - \mathbf{x}_{2}(1-\eta)(1-\zeta) \\ &- \mathbf{x}_{3}\eta(1-\zeta) + \mathbf{x}_{4}\eta(1-\zeta) \\ &+ \mathbf{x}_{5}(1-\eta)\zeta - \mathbf{x}_{6}(1-\eta)\zeta - \mathbf{x}_{7}\eta\zeta + \mathbf{x}_{8}\eta\zeta. \end{aligned}$$
(3.5)

Equation (3.5) and the analogous equations for  $\eta$  and  $\zeta$  form a set of covariant basis vectors as depicted in Fig. 3.2 for two-dimensions. The *J* in (3.4) is the Jacobian of the

mapping,

$$J = \mathbf{a}_i \cdot (\mathbf{a}_j \times \mathbf{a}_k). \tag{3.6}$$

Gradients under the mapped subdomains are computed using the chain rule such that  $\nabla_{\mathbf{x}} F(\mathbf{x})$  is given by the matrix-vector multiplication

$$\nabla_{\mathbf{x}}F(\mathbf{x}) = (\mathbf{J}^T)^{-1}\nabla_{\boldsymbol{\xi}}F \tag{3.7}$$

where  $(\mathbf{J}^T)^{-1}$  is the inverse of the Jacobian transpose matrix  $\mathbf{J}^T$  and in three-dimensions, (3.7) expands to

$$\begin{pmatrix} \frac{\partial F}{\partial x} \\ \frac{\partial F}{\partial y} \\ \frac{\partial F}{\partial z} \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{pmatrix}^{-1} \begin{pmatrix} \frac{\partial F}{\partial \xi} \\ \frac{\partial F}{\partial \eta} \\ \frac{\partial F}{\partial \eta} \\ \frac{\partial F}{\partial \zeta} \end{pmatrix}.$$
(3.8)

We can recast (3.7) in terms of the covariant bases,  $\mathbf{a}_i$ , such that

$$\nabla_{\mathbf{x}} F(\mathbf{x}) = \frac{1}{J} \sum_{i=1}^{3} \frac{\partial}{\partial \xi^{i}} [(\mathbf{a}_{j} \times \mathbf{a}_{k})F].$$
(3.9)

 $\tilde{\mathbf{Q}}$  is approximated with *N*<sup>th</sup>-order Legendre polynomials in the computational coordinates,  $(\xi, \eta, \zeta)$ ,

$$\tilde{\mathbf{Q}}_{N}(\xi,\eta,t) = \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=0}^{N} \tilde{\mathbf{Q}}_{i,j,k}(t) \ell_{i}(\xi) \ell_{j}(\eta) \ell_{k}(\zeta), \qquad (3.10)$$

where  $\ell_i(\xi)$ ,  $\ell_j(\eta)$  and  $\ell_k(\zeta)$  are the Lagrange polynomials given by

$$\ell_i(\xi) = \prod_{\substack{k=0\\k\neq i}}^{N} \frac{\xi - \xi_k}{\xi_i - \xi_k}.$$
(3.11)



Figure 3.2: Covariant bases definition.

The spatial derivative is determined in one-dimension as follows,

$$f'(\xi_i) \approx \sum_{j=0}^N f_j \ell'_j(\xi_i) = \sum_{j=0}^N D_{ij} f_j,$$
 (3.12)

with  $D_{ij}$  given by

$$D_{ij} = \begin{cases} \frac{1}{\xi_j - \xi_i} \prod_{k=0}^{N} \frac{\xi_i - \xi_k}{\xi_j - \xi_k}, & i \neq j \\ \sum_{l=0}^{N} \frac{1}{\xi_j - \xi_l}, & i = j. \end{cases}$$
(3.13)

In the Galerkin approach, the residual is minimized by taking the inner product with the test function,  $\varphi(\vec{\xi}) = \ell_i(\xi)\ell_j(\eta)\ell_k(\zeta)$ , yielding

$$\int_{D_k} \left( \tilde{\mathbf{Q}}_t + \nabla \cdot \tilde{\mathbf{F}} \right) \varphi(\vec{\xi}) \, \mathrm{d}\vec{\xi} = 0, \qquad (3.14)$$

Integrating by parts leads to

$$\int_{D_k} \frac{\partial \tilde{\mathbf{Q}}_N}{\partial t} \boldsymbol{\varphi}(\vec{\xi}) \, \mathrm{d}\vec{\xi} + \int_{\partial D_k} \tilde{\mathbf{F}}_N \cdot \mathbf{n} \, \boldsymbol{\varphi}(\vec{\xi}) \, \mathrm{d}S - \int_{D_k} \tilde{\mathbf{F}}_N \cdot \nabla \boldsymbol{\varphi}(\vec{\xi}) \, \mathrm{d}\vec{\xi} = 0.$$
(3.15)

To ensure connectivity between each subdomain, we replace  $\mathbf{F}_N \cdot \mathbf{n}$  in the boundary integral with a numerical flux  $\mathbf{F}^*$ , which for conservation laws is typically the solution to an approximate Riemann problem based on  $\mathbf{Q}_N$  at the coincident faces from two subdomains. Simplifying (3.15) with

$$\mathcal{M}_{i,j,k} = \int_{D_k} \ell_i(\xi) \ell_j(\eta) \ell_k(\zeta) \, \mathrm{d}\vec{\xi},$$
  

$$\mathcal{F}_{i,j,k} = \int_{\partial D_k} \ell_i(\xi) \ell_j(\eta) \ell_k(\zeta) \, \mathrm{d}S,$$
  

$$\mathcal{S}_{i,j,k} = \int_{D_k} \nabla(\ell_i(\xi) \ell_j(\eta) \ell_k(\zeta)) \, \mathrm{d}\vec{\xi}.$$
(3.16)

we have

$$\mathcal{M}\frac{\partial \tilde{\mathbf{Q}}_N}{\partial t} + \mathcal{F}\,\tilde{\mathbf{F}}^* - \mathcal{S}\,\tilde{\mathbf{F}}_N = 0, \qquad (3.17)$$

where  $\mathcal M$  is the mass matrix and  $\mathcal F$  is the stiffness matrix.

The remaining system of ordinary differential equations is integrated in time with a 4th-order low-storage, low-dispersion explicit Runge-Kutta method [87].

# **Chapter 4**

# Effect of Boundary Representation on Viscous, Separated Flows in a Discontinuous-Galerkin Navier-Stokes Solver

Sections 4.2 through 4.5 are reprinted from an upcoming manuscript submitted to Theoretical and Computational Fluid Dynamics in 2015 [88].

# 4.1 Overview and Summary

Little, if any, literature has been published on the effect of boundary representation on fluid solutions with high-order methods. Benchmark problems for DG Navier-Stokes solvers are nearly always in simple flow regimes with smooth geometries. Examples of simulations performed on straight-sided meshes and curved-sided meshes exist [15, 16, 28, 29, 31, 32, 35], however, no study has focused on the relationship between the approximation of curved boundaries and the physics of flow separation.

In this chapter, the effect of approximating curved boundaries with high-order curved DG elements on the DNS of the separated flow over an airfoil at low Reynolds number with DG methods is investigated. The flow over a NACA 65(1)-412 airfoil is studied, with emphasis on the impact of the boundary representation on the flow physics. The airfoil boundary is defined by a cubic spline constructed from coordinates published in the classical book on airfoil theory by Abbot et al. [89]. The edge polynomial of the DG element is fit to the spline and the polynomial order of the fluid solver is matched. The effect of *h* and *p* refinement are analyzed for both the curved-sided and straight-sided grids. It is shown that the straight-sided grids induce spurious instabilities in the flow, which lead to a large, chaotic wake and reduced aerodynamic performance when compared with the curved-sided grids.

# 4.2 High-Order Curved Boundary Representation

#### 4.2.1 Airfoil Geometry

The NACA 65(1)-412 airfoil was selected for its frequent use in industrial applications, such as turbomachinery. Geometries in real engineering applications tend to be defined by piecewise continuous functions, such as splines, as is the NACA 6-series airfoils. Hence, the complex boundary definition of the NACA 6-series are ideally suited for studying methods for approximating complex, real world geometries that do not have an analytical representation.

The coordinates for NACA 6-series airfoils are derived from a prescribed velocity profile with a method introduced by Abbot et al. [89]. According to Abbot et al., given a desired velocity profile, two parameters,  $\psi$  and  $\varepsilon$ , are computed from a theory developed by Theodorsen [90]. Theodorsen showed that the velocity distribution over an airfoil of arbitrary shape can be approximated by

$$v = V \frac{[\sin(\alpha + \theta + \varepsilon) + \sin(\alpha + \varepsilon_T)](1 + \varepsilon')e^{\psi_0}}{\sqrt{(\sinh^2 \psi + \sin^2 \theta)(1 + {\psi'}^2)}},$$
(4.1)

where *V* is the free-stream velocity,  $\alpha$  is the angle of attack,  $\varepsilon_T$  is the value of  $\varepsilon$  at the trailing edge, and  $\psi_0$  is the average value of  $\psi$ . The parameters,  $\psi$  and  $\varepsilon$ , are functions of  $\theta$  – the angle around the unit circle – and the derivatives,  $\psi'$  and  $\varepsilon'$ , are taken with respect to  $\theta$ .  $\psi$  and  $\varepsilon$  are related by

$$\cosh \Psi = \frac{x}{2\cos \theta},\tag{4.2}$$

$$\sinh \Psi = \frac{y}{2\sin\theta},\tag{4.3}$$

$$\boldsymbol{\varepsilon} = \boldsymbol{\phi} - \boldsymbol{\theta}. \tag{4.4}$$

Additionally, we have

$$z = e^{\psi_0 + i\phi},\tag{4.5}$$

$$z' = z e^{(\psi - \psi_0) - i\varepsilon},\tag{4.6}$$

$$\zeta = z' + \frac{1}{z'},\tag{4.7}$$

and

$$\zeta = x + iy. \tag{4.8}$$

where x and y are the airfoil coordinates. Hence, x and y are mapped to the z' plane with a conformal map and the parameters  $\psi$  and  $\varepsilon$  are computed as functions of  $\theta$  with Equations (4.2) - (4.8). According to the procedure by Abbot et al., symmetric airfoils with a predetermined velocity profile are produced iteratively by adjusting the coordinates, x and y until the desired velocity is reached as computed with (4.1). Camber is introduced

by computing the desired camber line,  $(x_c, y_c)$ , and projecting the symmetric airfoil coordinates,  $(x_s, y_s)$ , normal to the camber line such that we have

$$x_u = x_c - y_s \sin \delta(x_c) \tag{4.9}$$

$$y_u = y_c + y_s \cos \delta(x_c) \tag{4.10}$$

$$x_l = x_c + y_s \sin \delta(x_c) \tag{4.11}$$

$$y_l = y_c - y_s \cos \delta(x_c), \qquad (4.12)$$

where the subscripts *u* and *l* refer to the upper surface and lower surface, respectively, and  $\delta(x_c)$  is the local inclination of the mean line at  $x_c$ . Airfoils with the same camber line and series number that differ in thickness to cord ratio are generated by scaling the functions  $\psi$  and  $\varepsilon$  by a constant factor. The constant factor is generally not known a priori, hence the scale factor is applied iteratively until the required thickness ratio is achieved.

Several methods exist for computing additional coordinates not published in [89]. The first such method, developed by Ladson et al. [91], uses the coordinates published in [89] to determine the values for  $\phi$  and  $\varepsilon$  for a representative symmetric airfoil from each family in the NACA 6-series. New values of  $\phi$  and  $\varepsilon$  are obtained by interpolating between the stored values of  $\phi$  and  $\varepsilon$  using a cubic spline. The coordinates for an arbitrary thickness ratio are produced by providing an initial guess for the scale factor and iterating until the desired thickness ratio is obtained. Camber is added with the method discussed in [89], as before. This algorithm was later updated to include additional airfoil types and reflect advances in computer technology [92]. Carmichael [93] developed a method in which the scale factor can be determined using a polynomial fit through several precomputed scale factors, eliminating the need to iterate.

The methods in [91, 92, 93] are intended to rapidly produce coordinates for

airfoils that do not have coordinates already published in [89]. When airfoils published in [89] are used, the most common method for generating additional surface points is cubic spline interpolation, as is implemented in rapid airfoil design software such as XFoil, JavaFoil and XFLR5 [94]. This work uses the cubic spline method to conform to common practice. The spline,  $\mathbf{X}(s)$ , is defined on a set of *n* data points,  $\mathbf{X}_i$ , for i = 1, 2, ...n. The data points begin at the trailing edge and run over the upper surface, to the leading edge, then across the lower surface back to the trailing edge. The spline parameter, *s*, is defined such that the nodes, *s<sub>i</sub>*, are determined by the distance between each data point, and then are normalized to lie on the interval [0,1]. Before normalization, we have

$$s'_{i} = \sum_{j=1}^{i-1} \sqrt{(X_{j+1} - X_{j})^{2} + (Y_{j+1} - Y_{j})^{2}},$$
(4.13)

for i = 2, 3, ..., n and  $s'_1 = 0$ . The parameter is then normalized so that

$$s_i = \frac{s'_i - s'_1}{s'_n - s'_1}.\tag{4.14}$$

The curve,  $\mathbf{X}_{j}(s)$  for each interval *j* in which  $s_{i} \leq s_{j} \leq s_{i+1}$  is computed with

$$\mathbf{X}_j(s) = \mathbf{a}_j + \mathbf{b}_j s + \mathbf{c}_j s^2 + \mathbf{d}_j s^3, \qquad (4.15)$$

with the constraints

$$\mathbf{X}_j(s_i) = \mathbf{X}_{j+1}(s_i), \tag{4.16}$$

$$\mathbf{X}'_{j}(s_{i}) = \mathbf{X}'_{j+1}(s_{i}), \tag{4.17}$$

$$\mathbf{X}_{j}''(s_{i}) = \mathbf{X}_{j+1}''(s_{i}). \tag{4.18}$$

A natural spline is constructed by setting the second derivative at the endpoints to zero,

$$\mathbf{X}_{1}^{\prime\prime}(0) = \mathbf{X}_{n-1}^{\prime\prime}(s_{n}) = 0.$$
(4.19)

A tri-diagonal system of equations is constructed to solve for the coefficients in (4.15) and the coefficients are stored for evaluation of  $\mathbf{X}(s)$ .

#### 4.2.2 Boundary Node Adjustment

We use the commercial software GridPro to generate the initial computational grid. GridPro produces straight-sided, quadrilateral, multi-block meshes with an iterative smoothing algorithm that ensures that the orthogonality and mesh smoothness are optimized. The requirement for generated nodes to coincide exactly with boundaries is relaxed in order to maximize smoothness and orthogonality and improve robustness in case of conflicting geometry constraints. In theory, the grid generator should converge to coincide with the boundaries. However, in practice, this does not occur with curved geometries, since smoothness and orthogonality are enforced at the cost of precision [95].

We adjust the nodes in Matlab to coincide exactly on the spline by determining the point on the spline,  $X_s$ , that is closest to the GridPro-generated point,  $X_{GP}$ . First, the closest spine node,  $X_i$ , is determined, by cycling through each spline node and finding the node in which

$$\sqrt{(X_i - X_{GP})^2 + (Y_i - Y_{GP})^2} \tag{4.20}$$

is minimum. The nearest point on the spline is then found by minimizing the function

$$f(s) = (\Delta \mathbf{X}(s))^2 = (X(s) - X_{GP})^2 + (Y(s) - Y_{GP})^2.$$
(4.21)

To find the minimum of f(s), a Newton method is used to find the zeros of f'(s) with the spline nodes in (4.20) as an initial guess. The first and second derivatives of f(s) are given by

$$f'(s) = (X(s) - X_{GP})X' + (Y(s) - Y_{GP})Y',$$
(4.22)

$$f''(s) = (X(s) - X_{GP})X'' + (Y(s) - Y_{GP})Y'' + (X')^2 + (Y')^2,$$
(4.23)

respectively. The parameter, *s*, corresponding to the nearest spline point is then determined by iterating

$$s^{n+1} = s^n - \frac{f'(s^n)}{f''(s^n)},$$
(4.24)

until *s* converges to the desired tolerance. We calculate the location,  $\mathbf{X}_s$ , using the stored spline coefficients in (4.15).

#### 4.2.3 Compatibility of Commercial Mesh Generators with DG

GridPro produces geometry that has two characteristics that are incompatible with the DG method employed in this work, and hence require adjustment in order to produce a mesh that can be used in the fluid solver. First, the face numbering convention in GridPro is different than the convention used in DG, as depicted in Figure 4.1. Second, GridPro produces a multiblock mesh while DG does not.

Geometry produced by GridPro is converted to DG compatible geometry in several steps. First, the face ordering of the GridPro blocks is remapped to be consistent with the DG method. Next, the DG element data is constructed from each element in the GridPro blocks. Within each  $N \times M$  block, the element indexing is such that DG element *n* corresponds to GridPro element (j-1)i for i = 1, 2, ...N and j = 1, 2, ...M. The blocks are then merged by identifying he duplicate nodes on block interfaces and deleting them. Finally, the node data, element data, and boundary condition data are all



Figure 4.1: Differences in face ordering between GridPro and DGSEM.

printed to a mesh format compatible with the DG code.

A summary of the mesh conversion algorithm is given below:

- 1. The boundary nodes are adjusted to coincide with the spline curve.
- 2. Element faces are reordered to be consistent with the DG convention.
- DG elements are constructed from the elements within the blocks generated by GridPro.
- 4. The duplicate nodes on the boundaries of the blocks are eliminated.
- 5. The node, element, and boundary condition data are written in a mesh file that is compatible with the DG solver.

#### 4.2.4 High-Order Boundary Fitting

The straight-sided elemental data generated by GridPro are fit to the spline boundary with high-order polynomial elemental boundaries in the DG solver. The boundary in the DG scheme is piecewise approximated by the parametric curves,

$$\Gamma_k(\xi) = \sum_{j=0}^N \mathbf{X}_j \ell_j(\xi), \qquad (4.25)$$

for  $k = 1, 2, ..., K_b$ , where  $K_b$  is the number of elements along the boundary and  $\xi$  is the parameter. Each boundary curve spans a length of the spline over the interval  $[\mathbf{X}_a, \mathbf{X}_b]$ , with corresponding spline interval  $[s_a, s_b]$ . To fit  $\Gamma_k(\xi)$  to the boundary spline,  $\mathbf{X}$ , we map the parameter,  $\xi$ , to the parameter, s, using an affine map such that

$$s_j = \xi_j (s_b - s_a) + s_a,$$
 (4.26)

for j = 0, 1, ..., N, where *N* is the desired polynomial order. The nodes  $\xi_j$  are taken to be the Chebyshev-Lobatto quadrature nodes. Hence, for the nodes of  $\Gamma_k(\xi)$ , we have  $\mathbf{X}_j = \mathbf{X}(s_j)$  and with  $s_j$  as in (4.26).

When the curved element faces are constructed, the metric terms for the curved elements are computed. By constructing the curved element faces in the pre-processing stage of the solver, the order of the fluid solver is matched ad hoc and a single grid with straight-sided elements is used, eliminating the need to generate multiple grids with different boundary polynomial orders, if needed, and ensuring that the boundary approximation will always be consistent with the order of the fluid solver.

The convergence of the high-order boundary representation depends, in part, on the underlying geometry. Uniform convergence of the boundary approximation assumes that the approximated function is smooth for all *s* over the interval of the approximating polynomial,  $[s_a, s_b]$ . Following Canuto et al. [96], the error in the Legendre polynomial interpolant,  $I_N u$  approximating the function *u* is estimated to be

$$\|u - I_N u\|_{L^2(-1,1)} \le C N^{-m} |u|_{H^{m;N}(-1,1)}, \tag{4.27}$$

assuming  $u \in H^m(-1,1)$ , where  $H^m(-1,1)$  is the  $m^{th}$ -order Sobolev space. When u is analytic, and hence  $u \in C^{\infty}$ , the interpolant will converge to u faster than any power of N, *i.e.* spectrally. Given that the boundary we are approximating is a cubic spline,  $\mathbf{X}(s)$ , we have  $\mathbf{X}(s) \in H^2(s_a, s_b)$ , and

$$\|\mathbf{X} - \Gamma\|_{L^{2}(s_{a}, s_{b})} \sim \mathcal{O}(N^{-2}).$$
(4.28)

Hence, we have second-order convergence, at best.

Algebraic convergence of the approximating polynomial is a consequence of the discontinuous third derivative of the spline at the spline nodes,  $s_i$ , where we have

$$\Delta \mathbf{X}^{\prime\prime\prime\prime} \equiv |\mathbf{X}^{\prime\prime\prime\prime}(s_i^+) - \mathbf{X}^{\prime\prime\prime\prime}(s_i^-)| \neq 0 \,\forall s_i. \tag{4.29}$$

Approximating discontinuous functions with smooth functions results in Gibbs-like oscillations in the approximating function near the discontinuities. When we have one or more  $s_i \in [s_a, s_b]$ , the error in the interpolant will depend, in part, on the size of the discontinuity,  $\Delta \mathbf{X}'''$ , as well as the number of discontinuities in  $[s_a, s_b]$ .

Geometric accuracy of the boundary approximation is measured by the local error in the slope, m, and the curvature,  $\kappa$ , of the DG representation of the boundary. The local error of the slope is given by

$$\varepsilon_m(s) = \left| \frac{(\Gamma_k^y(\xi|_s))'}{(\Gamma_k^x(\xi|_s))'} - \frac{Y'(s)}{X'(s)} \right|.$$
(4.30)

The curvature of the spline curve is defined by

$$\kappa(s) = |X'(s)Y''(s) - Y'(s)X''(s)| \left( \left[ X'(s) \right]^2 + \left[ Y'(s) \right]^2 \right)^{-3/2}.$$
(4.31)

The curvature of the DG boundary representation,  $\kappa_{\Gamma}$ , is similarly determined with the functions  $\Gamma_k^x(\xi|_s)$  and  $\Gamma_k^y(\xi|_s)$ . The local error of the curvature is given by

$$\varepsilon_{\kappa}(s) = |\kappa_{\Gamma} - \kappa| \,. \tag{4.32}$$

The accuracy of the fluid solution near the boundary depends on the degree to which the boundary approximation is affected by the inconsistency in the differentiability, which is directly related to the size of the discontinuity in the third derivative of the boundary spline given by (4.29). Regions with high curvature tend to be associated with large changes in curvature, which often result in large third derivative discontinuities in a cubic spline. Alternatively, regions with low curvature area accompanied by small changes in curvature and hence small third derivative discontinuities. As such, one would expect the region on a spline with the highest curvature to correspond to the region of highest error in the spline boundary approximation.

### 4.3 **Problem Setup and Grid Independence**

The flow over a NACA 65(1)-412 airfoil is computed at a Reynolds number of Re = 20,000 based on the cord length and the freestream velocity. The Mach number is M = 0.3 and the angle of attack is  $\alpha = 4^{\circ}$  (Fig. 4.5).

The size of the domain is based on the results by Jones *et al.* [97], who show that for the two-dimensional DNS of a NACA 0012 airfoil at a low Reynolds number (Re = 50,000), the difference between the solution from a grid with boundaries 5.3 chord-lengths from the airfoil and a solution with a grid 7.3 chord lengths away is not significant. To verify this, we compute the solutions with two domain sizes. One has an inlet surface that is 5 cord-lengths away from the leading edge, and the wake extends 15 cord lengths downstream of the trailing edge. The second, larger domain extends 30 chord-lengths from the airfoil in all directions. We compute the solution using the straight-sided boundary approximation and the curved-sided boundary approximation, both with a polynomial order of 12.

Comparing the average pressure and average skin-friction coefficients (Fig. 4.2)



**Figure 4.2**: Comparision of the skin friction (a) and pressure (b) of the small domain size and the large domain size using the curved-sided boundary approximation.

**Table 4.1**: Results for the small domain and the large domain, N = 12. CS denotes the curved-sided element mesh and SS denotes the straight-sided element mesh.

	Small Domain		Large Domain	
	CS	SS	CS	SS
Total Lift	0.22426	0.20295	0.22188	0.20144
Total Drag	0.01197	0.01297	0.01194	0.01257
Frequency	2.7	2.5	2.7	2.5

between the curved-sided results shows that they are the same to graphical accuracy and the boundaries in the smaller domain do not have a significant effect on the solution. The lift coefficient, drag coefficient, and Strouhal number for the small domain differ by at most 1% from the large domain values, which are tabulated in Table 4.1. Results from the straight-sided mesh are in agreement between the small domain and large domain as well, as shown in Figure 4.3 and Table 4.1. The effect of the domain size on the lift spectra is not significant in both the curved-sided and straight-sided case (Fig. 4.4). Since the solution is independent of domain size for the domains chosen, the remainder of the analysis in this work will use the smaller domain.

The left side of the domain has an inflow condition specified, while the top,



**Figure 4.3**: Comparision of the skin friction (a) and pressure (b) of the small domain size and the large domain size using the straight-sided boundary approximation.



**Figure 4.4**: Comparision of the lift spectra between the small domain size and the large domain size using the curved-sided boundary approximation (a) and the straight-sided boundary approximation (b).



Figure 4.5: Airfoil simulation setup.

bottom and right side specify pressure outflow conditions. The surface of the airfoil is a no-slip, adiabatic wall. Each computation was run for 90 time units. After a ten time unit initial transient stage, the average flow field was computed over 40 time units and the RMS flow field was computed over an additional 40 time units.

#### 4.3.1 Mesh Characteristics

The simulations are performed on C-grids, which are depicted in Fig. 4.6. One grid is coarse and contains 1,044 elements, whereas the refined grid contains 2,256 elements. The ratio of largest element area to smallest element area for the coarse grid is  $A_{max}/A_{min} = 8.46 \times 10^3$ , with  $A_{min} = 2.1 \times 10^{-4}$ . For the fine grid,  $A_{max}/A_{min} = 9.68 \times 10^3$ , with  $A_{min} = 9.9 \times 10^{-5}$ . Polynomial orders range from N = 4 to N = 12. A close-up of the leading edge for each mesh is seen in Figs. 4.7 and 4.8, illustrating the improvement in boundary approximation with the curved-boundary representation.



Figure 4.6: Coarse mesh (a) and fine mesh (b).



Figure 4.7: Straight-sided mesh detail (a) and curved-sided mesh detail (b) for Mesh 1. The polynomial order is N = 12.



**Figure 4.8**: Straight-sided mesh detail (a) and curved-sided mesh detail (b) for Mesh 2. The polynomial order is N = 12.

#### 4.3.2 Grid Refinement

The effect of *h*- and *p*-refinement is assessed by evaluating the frequency spectrum of the lift time history and the skin friction coefficient,  $C_f$ . Each mesh is *p*-refined up to a polynomial order of N = 12. The accuracy of each mesh is estimated with the root-mean-square (RMS) local error in  $C_f$ ,  $\varepsilon_{RMS}$ , with respect to the curve-sided Mesh 2 solution at N = 12,  $C_f^{ref}$ ,

$$\varepsilon_{RMS} = \left(\frac{1}{n}\sum_{i=1}^{n} \left(C_f(x_i) - C_f^{ref}(x_i)\right)^2\right)^{1/2}.$$
(4.33)

As *N* is increased for the straight-sided meshes (Figs. 4.10 and 4.12), there is no significant change in the accuracy. However, the straight-sided approximations converge to different solutions that depend on the mesh. For the straight-sided Meshes 1 and 2,  $\varepsilon_{RMS} = 2.6 \times 10^{-3}$  and  $1.3 \times 10^{-3}$ , respectively. Furthermore, the straight-sided meshes have multiple large peaks in the frequency spectrum (Figs. 4.10a and 4.12a).

In contrast, the curved-sided approximations (Figs. 4.9 and 4.11) converge to the same solution with increasing *N*, regardless of the mesh used, as we have  $\varepsilon_{RMS} = 5.5 \times 10^{-4}$  for the coarse mesh at N = 12. Both curved-sided meshes show a single peak frequency in the frequency spectrum (Figs. 4.9a and 4.11a).

#### 4.3.3 Comparison with Previous Studies

Few numerical studies of airfoils at a Reynolds number near 20,000 are available in the literature, so an adequate benchmark for airfoils at such a Reynolds number has not yet been established. However, Yarusevych *et al.* [98] present an approach to determine a characteristic Strouhal number for bluff-body flows that is largely independent of Reynolds number, thus providing a universal point of comparison for a wide class of



Figure 4.9: Convergence for curved-sided Mesh 1.



Figure 4.10: Convergence for straight-sided Mesh 1. Note the additional frequencies as compared to the curved-sided mesh.



Figure 4.11: Convergence for straight-sided Mesh 2.



Figure 4.12: Convergence for straight-sided Mesh 2. Note the additional frequencies as compared to the curved-sided mesh.

flows. The universal Strouhal number,  $St_s^*$ , is given by

$$St_s^* = \frac{f_s d^*}{U_0},$$
 (4.34)

where  $f_s$  is the vortex shedding frequency,  $U_0$  is the free-stream velocity and  $d^*$  is the distance between the two local maxima in the upper and lower portions of the wake r.m.s velocity profiles obtained at x/c = 1.25. For airfoil flows computed at a range of Reynolds numbers of  $Re \approx 50,000 - 225,000$ , Ref. [98] reports values of  $St_s^* \approx 0.14 - 0.25$  with an average of  $St_s^* \approx 0.17$ . The value of  $St_s^*$  computed in this work is  $St_s^* \approx 0.26$ , which is comparable to the values given in [98].

Work by Dortmann [99] details results at a similar Reynolds number that qualitatively agree with the results presented in this work. Specifically, the flow over the airfoil separates near mid-cord and the wake is characterized by a regular Karman vortex street.

## 4.4 Direct Numerical Simulation

#### 4.4.1 Wake Analysis

The wake is time-periodic with an asymmetric von Karman street of vortices shed from the upper and lower surface of the airfoil (Fig. 4.13). The stronger vortices are shed from the lower surface. The separation region on the suction surface of the airfoil, defined to be the region of fluid beginning at the initial point of boundary-layer separation at  $x/c \approx 0.5$ , and extending to the trailing edge, consists of a series of stable vortices, the furthest downstream of which is the progenitor of the upper surface shed vortex.

Straight-sided and curved-sided elements produce different wakes. With the straight-sided grid, the wake is irregular and upturned, indicating lower circulation, a loss of lift and an increase in drag.



**Figure 4.13**: Flow field plots for grid 1. Vorticity for the curved-sided grid (a) and the straight-sided grid (b).



**Figure 4.14**: Flow field plots for grid 1. RMS velocity for the curved-sided grid (a) and the straight-sided grid (b).

The RMS velocity,  $u_{rms}$ , in Fig. 4.14 shows that the curved-sided mesh (Fig. 4.14a) produces a flow with a very narrow, well defined wake, whereas the straight-sided mesh (Fig. 4.14b) produces a flow with a wide, vertically dispersed wake. Furthermore,  $u_{rms}$  is higher on the suction surface near the trailing edge of the straight-sided grid results as compared to the curved-sided, which indicates a complex time-dependency in the separation region for the straight-sided grid.

When the mesh is *h*-refined, the differences between the curved-sided and straightsided mesh are less pronounced (Fig. 4.15). The curved-sided airfoil has the same stable vortex street as in the flow from Mesh 1. Although the straight-sided airfoil has an unstable vortex street, the wake is stable for a longer time than for the coarse mesh flow (Fig. 4.15b).

The fine straight-sided mesh also shows an increased wake dispersion as compared



**Figure 4.15**: Flow field plots for grid 2. Vorticity for the curved-sided grid (a) and the straight-sided grid (b).



**Figure 4.16**: Flow field plots for grid 2. RMS velocity for the curved-sided grid (a) and the straight-sided grid (b).

to the curved-sided results. Similar to the coarse mesh  $u_{rms}$  profile, there is an increase in the  $u_{rms}$  near the trailing edge of the suction surface, corresponding to increased variation in the fluid motion in the separation region.

### 4.4.2 Force Analysis

The aerodynamic forces on the airfoil are computed using Gauss quadrature to ensure a consistent high-order accuracy of the force calculation. The lift and drag are determined by

$$L = \sum_{k=1}^{K_b} \int_{S} (\tau_{xy} n_x + \tau_{yy} n_y - p n_y) dS$$
(4.35)

$$=\sum_{k=1}^{K_b}\sum_{i}^{N}(\tau_{xy}n_x + \tau_{yy}n_y - pn_y)_i w_i dS_i, \qquad (4.36)$$

$$D = \sum_{k=1}^{K_b} \int_{S} (\tau_{xx} n_x + \tau_{yx} n_y - p n_x) \mathrm{d}S$$
(4.37)

$$=\sum_{k=1}^{K_b}\sum_{i}^{N}(\tau_{xx}n_x+\tau_{yx}n_y-pn_x)_iw_i\mathrm{d}S, \qquad (4.38)$$

from the shear stress tensor,  $\tau_{ij}$ , (given by (2.18)) and the pressure, *p*, using the outward normal, **n**, the quadrature weights,  $w_i$ , and dS is the differential arc-length element given by

$$dS = \sqrt{\left(\frac{\partial\Gamma_k^x(\xi)}{\partial\xi}\right)^2 + \left(\frac{\partial\Gamma_k^y(\xi)}{\partial\xi}\right)^2} d\xi.$$
(4.39)

The lift and drag are computed at each time step and the time series is recorded.

The integration error under Gauss quadrature relative to the Legendre weight, from Canuto *et al.* [96], is given by

$$\left| \int_{-1}^{1} u(x)\phi(x)dx - (u,\phi)_{N} \right| \le CN^{-m} |u|_{H^{m;N}(-1,1)} ||\phi||_{L^{2}(-1,1)},$$
(4.40)

where  $\phi \in \mathbb{P}_N$ . Thus the integrated forces converge spectrally in a manner that is consistent with the spectral accuracy of the solution method.

The pressure and skin friction at the boundary are computed by projecting the pressure and shear stress tensor to the boundary using Lagrange interpolation. The pressure at the boundary is then extracted directly and the skin friction is determined from the stress at the boundary given by

$$C_f = \frac{1}{Re_f} \left\{ - \left( \tau_{xx} n_y + \tau_{yx} n_y \right) n_x + \left( \tau_{xy} n_x + \tau_{yy} n_y \right) n_x \right\},$$
(4.41)

where  $n_x$  and  $n_y$  are the components of the surface normal vector.

The large element sizes in the present study, which are typical for high-order DGSEM grids, cause significant differences in the vortex shedding behavior between the


Figure 4.17: Time series of the lift (a) and frequency spectra of the lift (b) for grid 1. N = 12 for both the straight-sided and curved-sided grid.

straight-sided mesh and the curved-sided mesh. Large amplitude perturbations imposed by the sharp corners in the coarse straight-sided mesh contribute additional frequencies as compared to the curved-sided mesh, which destabilize the wake. In the frequency domain, the amplitude spectrum of the lift and drag (Fig. 4.17b) show a single peak at a frequency of 2.7, *i.e.* the flow is periodic with a single, distinct period. In contrast, the amplitude spectrum of the straight-sided grid exhibits several peaks at lower frequencies, with the primary peak at a different frequency than the curved-sided grid, which is 2.1.

The wide wake dispersion depicted in Fig. 4.14b means that there is poor pressure recovery on the trailing edge of the airfoil, which leads to reduced lift and increased drag (Fig. 4.17). The average lift coefficient,  $C_L$ , for the curved and straight-sided grid is 0.225 and 0.199, respectively. The average drag coefficient,  $C_D$  is 0.0117 and 0.0137 for the curved and straight-sided grid respectively.

Lower lift and higher drag for the straight-sided airfoil as compared to the curvedsided airfoil are observed in the fine grid (Mesh 2) results. However, the fine grid induces smaller amplitude perturbations on the boundary layer, as compared to the coarse grid (Mesh 1). The reduced strength of the corner perturbations has a decreased effect on the



Figure 4.18: Time series of the lift (a) and frequency spectra of the lift (b) for grid 2. N = 12 for both the straight-sided and curved-sided grid.

	Mesh 1		Mesh 2	
	CS	SS	CS	SS
p-Lift	0.224139	0.197644	0.223019	0.20178
f-Lift	0.001245	0.001119	0.001237	0.001171
Total Lift	0.225383	0.198763	0.224256	0.202952
p-Drag	0.00223	0.003853	0.002541	0.003399
f-Drag	0.009439	0.009857	0.009425	0.009571
Total Drag	0.011668	0.01371	0.011966	0.01297
Frequency	2.7	2.1	2.7	2.5

Table 4.2: Results for Mesh 1 and Mesh 2, N = 12.

large scale aerodynamic characteristics. The differences between the straight-sided and curved-sided mesh are less for the refined mesh when compared to the coarse mesh (Fig. 4.17).

The aerodynamic forces are tabulated in Table 4.2. The  $C_L$  for the curved and straight-sided grid is 0.224 and 0.203 respectively. The  $C_D$  for the curved and straight-sided is 0.0120 and 0.0130 respectively. The amplitude spectrum (Fig. 4.18b) also depicts smaller differences, with fewer additional peaks in the straight-sided grid. The peak frequency for the curved and straight-sided grid is 2.7 and 2.5, respectively.



**Figure 4.19**: Average pressure coefficient,  $C_p$ , (a) and skin friction coefficient,  $C_f$ , for the upper surface (b) for Mesh 1.

# 4.4.3 Pressure and Skin Friction Distributions

The straight-sided meshes have sharp corners that the fluid must travel around or separation will occur. The effect the corner has on the fluid is dependent on the angle of the corner, with higher angles producing larger effects. Note that the effect of the corners in viscous flow is highly non-linear. Relatively small angles can produce large scale effects on the flow. As the fluid traverses the corner, a pressure well and a peak in the shear stress develops as a consequence of the large centripetal force associated with turning of the flow. These wells and peaks lead to poor pressure recovery and a corresponding loss in lift, as well as increases in skin friction and an overall increase in drag. An increase in the skin friction in the separation region is seen for the straight-sided mesh, contributing to the drag further. The average aerodynamic profiles for Mesh 1 (Fig. 4.19) illustrate the pressure wells (Fig. 4.19a) and skin friction peaks (Fig. 4.19b) resulting from the straight-sided mesh, as compared to the smooth profiles produced by the curved-sided mesh.

With the *h*-refined boundary, the effect of the sharp corners on the flow decreases



**Figure 4.20**: Average pressure coefficient,  $C_p$ , (a) and skin friction coefficient,  $C_f$ , for the upper surface (b) for Mesh 2.

as the slope discontinuities decrease (Fig. 4.20). The pressure wells (Fig. 4.20a) and skin friction peaks (Fig. 4.20b) remain, similar to the Mesh 1 results. However, with h-refinement, the peaks and wells are smaller with a corresponding smaller loss in lift and increase in drag. The straight-sided skin friction profile in the separation region compares well with the curved-sided results, suggesting that vortex shedding behavior is similar between the refined straight-sided grid and the curved-sided grid.

### 4.4.4 Curvature Analysis

We measure the accuracy of the boundary representation in the curved mesh with the error in slope,  $\varepsilon_{\lambda}$ , and curvature,  $\varepsilon_{\kappa}$  as a function of parameter *s*. Error is analyzed on the refined mesh with the polynomial order, N = 12. Two significant sources of error in the boundary representation are the high gradients near the leading edge and the discontinuities in the second derivative of the boundary spline. The highest errors in the slope are concentrated near the leading edge ( $s \approx 0.5$ ) (Fig. 4.21) where the curvature is highest and the spline discontinuities are the largest. Similar to the slope error, the error in the curvature (Fig. 4.22) is highest near the leading edge ( $s \approx 0.5$ ) where the changes



**Figure 4.21**: Slope of the airfoil spline definition plotted against the *x*-coordinate (a) and local error in slope (b) plotted against the spline parameter, *s*. The error in the slope was computed for Mesh 2, N = 12.

in curvature are greatest.

The effect of the boundary representation on the near-wall flow is examined in two regions. The first is near the leading edge, where the curvature is high. The second is in the suction surface separation region, where the curvature is low.

The error in the boundary curvature is compared to the surface aerodynamic forces near the leading edge in Fig. 4.23. The data from three elements near the leading edge are overlaid, where the vertical black lines depict the element boundaries. The largest spread in the solution data coincides with the element with the largest curvature error. Furthermore, large second derivatives in the skin friction curves are evident at the element boundaries, as compared to the pressure curves, hence the skin friction is more sensitive to the quality of the boundary representation than the pressure.

The error in the boundary curvature as compared to the aerodynamic forces near the separation location is depicted in Fig. 4.24. The curvature errors are roughly two orders of magnitude smaller than the error near the leading edge. Notably, the spread in the solution data is very small and all polynomial orders depicted are in excellent



**Figure 4.22**: Curvature in the airfoil spline definition (a) and local error in the curvature (b), both plotted as a function of the spline parameter, *s*. The error in the curvature was computed for the curved-sided Mesh 2, N = 12.



**Figure 4.23**: Effect of boundary representation near the leading edge. The black vertical lines depict the locations of element boundaries. Curved-sided Mesh 2 is used.



**Figure 4.24**: Effect of boundary representation on separation region. The black vertical lines depict the locations of element boundaries. Curved-sided Mesh 2 is used.

agreement with each other within graphical accuracy.

## 4.4.5 Discussion

Differences in aerodynamic performance between Mesh 1 and Mesh 2 (Table 4.2) for the curved airfoil are small, with a maximum of 2.5% difference in the drag between the two meshes. In contrast, the aerodynamic performance between Mesh 1 and Mesh 2 for the straight-sided airfoil are less consistent, with a 2.1% increase in lift with Mesh 2 over Mesh 1 and a 5.4% decrease in drag for Mesh 2 over Mesh 1. The results from Mesh 2 with the straight sides are in better agreement with the curved-sided results than the straight-sided Mesh 1. This is consistent with the piecewise linear approximation

improving as the number of elements increases. The large observed differences between the flow physics produced by straight-sided meshes and the curved-sided meshes imply that the straight-sided meshes fail to simulate the correct physics.

# 4.5 Summary of Results

A comprehensive analysis of the effect of curved-boundary approximation on the physics of the separated flow over an airfoil is presented. An approach to fitting high-order curved DGSEM elemental edges to the airfoil geometries defined by splines is introduced and the DNS results of the high-order boundary approximation is compared to low-order piecewise-linear boundary representations. The piecewise-linear approximations are shown to result in the simulation of incorrect physics.

The method for fitting high-order DGSEM elements with curved-edges to complex boundaries defined by splines matches the polynomial order of the simulation by fitting a polynomial at the Chebyshev-Lobatto quadrature nodes of the same order as the fluid solver for each element edge along the boundary. Constructing the curved elements as a preprocessing step in the DGSEM solver allows for greater flexibility with the mesh definition, i.e. the high-order curved boundary approximation does not have to be incorporated into the mesh file. A method for improving the accuracy of boundary node location is included in the geometry generation algorithm.

Although the method uses high-order piecewise polynomials to approximate the boundary, spectral accuracy in the boundary representation cannot be achieved for boundary functions that are non-analytic, as demonstrated by the implementation of this algorithm on the cubic spline that defines the profile of the NACA 65(1)-412 airfoil. In regions of the airfoil with high curvature, poor boundary convergence leads to poor convergence of the solution. Solution convergence improves in regions with low curvature.

The DNS of the flow over a NACA 65(1)-412 is performed to demonstrate the effect of using a high-order boundary representation as compared to a piecewise linear approximation. It is established that the solution is grid independent with the comparison of skin friction profiles and the time series amplitude spectrum for each grid.

Artificial surface roughness introduced by the piecewise-linear boundary approximation of the straight-sided meshes results in the simulation of incorrect physics. Visualization of the wake vorticity field, and RMS field reveal the destabilization of the von Karman vortex street by the piecewise-linear boundary approximation, as compared to the high-order boundary approximation. The destabilized wake leads to a wide wake dispersion, indicating loss of aerodynamic performance.

Analysis of the time series of the lift and drag for each grid demonstrates the loss in aerodynamic performance by the piecewise linear boundary approximation as compared with the high-order boundary approximation. Amplitude spectra of the lift and drag time series indicate the introduction of spurious frequencies by the piecewise-linear boundary, whereas the high-order boundary representation results in a single strong peak.

A comparison of the surface profile of the skin friction and the pressure for each mesh identifies pressure wells and skin friction peaks at the corners of the straight-sided mesh, which are not present in the profiles associated with the curved-sided mesh. The boundary effects of the straight-sided mesh are identified as a significant source of the loss in the predicted aerodynamic performance of the piecewise-linear boundary approximation.

The effect of the boundary representation on local flow features is analyzed, including the flow near the leading edge and the flow separation on the suction surface. The effect on the flow is greater near the leading edge, where the boundary error was higher. Significant sources of the boundary error include the second derivative discontinuities in the boundary spline and the large boundary derivatives in the high curvature region of the leading edge.

Chapter 4, in part, has been submitted for publication of the material as it may appear in Theoretical and Computational Fluid Dynamics, 2015, Nelson, Daniel; Jacobs, Gustaaf and Kopriva, David, "Effect of Boundary Representation on Viscous, Separated Flows in a Discontinuous-Galerkin Navier-Stokes Solver". The dissertation author was the primary investigator and author of this paper.

# Chapter 5

# Lagrangian Coherent Structures with High-Order Discontinuous-Galerkin Methods

Sections 5.2, 5.4 and 5.5 are a reprint of a manuscript published in The Journal of Computational Physics in 2015 [75]. The results presented in Section 5.3 are reprinted from a conference publication submitted to the American Society of Mechanical Engineers in 2013 [76].

# 5.1 Overview and Summary

In this Chapter, we present a DG-based algorithm that determines flow maps and corresponding FTLE fields at multiple times from a particle trace at a single initial time, both backward in time and forward in time in two dimensions. Since the algorithm requires only a single particle trace at one initial time to determine the FTLE fields at several (forward and backward) times, it is low-storage and computationally efficient. By initiating particles at quadrature points, flow map approximations are generated that are consistent with the piecewise DG approximation. Hence, DG operators can be directly used to determine the deformation gradients required for the FTLE. Because of its seamless fit with DG Navier-Stokes solvers, the algorithm is not only consistent, but is also suited for the simultaneous computation with large-scale, parallel direct numerical simulations and large eddy simulations based on DG [100, 101, 102, 103], at low computational overhead, while eliminating the need for storage of large data sets.

Although effort has been made to decrease storage requirements and improve efficiency of the presented algorithm as compared to currently available methods, the focus of this chapter is on the high-order accuracy of the algorithm. Computational costs will be examined in detail elsewhere.

In the next section, we summarize the particle tracer, the flow map approximation and FTLE determination with DG. Next, the interpolation of flow maps at multiple times from a single initial time particle trace is discussed, including its conditioning and the relation of conditioning to the deformation field. The algorithm is tested on three benchmark cases, including a spatially periodic gyre flow, an analytical vortex advected by a uniform flow, and the viscous flow around a square cylinder. The final section is reserved for conclusions and future work.

# 5.2 FTLE with Spectral Methods

## 5.2.1 Fluid Particle Tracking Algorithm

The particle tracking algorithm requires three steps [104]: 1) the host domain of the particle is located; 2) the fluid velocity is interpolated from the DG grid to the particle's location; 3) the system (2.19) is integrated in time.

We initialize particles at Chebyshev Gauss-Lobatto quadrature nodes in each

subdomain, and numerically integrate the dynamical system (2.19) over the time interval  $T = t - t_0$ . The particle final locations form the approximation

$$\phi \approx \bigcup_{k=1}^{K} \Phi_N^k. \tag{5.1}$$

where  $\Phi_N^k$  is the *N*<sup>th</sup>-order polynomial approximation of  $\phi$  on the subdomain  $D_k$ . Since in DG, the boundaries of conforming subdomains coincide, we remove duplicate boundary particles to improve the efficiency of the particle tracking algorithm.

The particle's host cell is found by comparing the particle coordinate in the unit square to the bounds of the unit square. The inverse of the blending function (3.2) maps the particle's physical space location to its location,  $\xi_p$ , within the unit square. For curved sided subdomains, an analytical expression for the inverse map does not exist and so a Newton root-finding method is used to invert the map. A fluid particle leaves a subdomain if it's mapped coordinate is outside the bounds of a unit square. If this is the case, the subdomains shared by the nearest corner determine the new host cell. The fluid velocity is interpolated to the particle location using the existing polynomial approximation of the solution vector,  $\mathbf{Q}_N$ . Therefore, the interpolation error in the fluid particle velocity is of the same order as the numerical scheme. Particles are advected by integrating (2.19) in time with an Adam-Bashforth (AB) scheme. The AB is used because it is computationally more efficient as compared to Runge-Kutta schemes used to integrate the Navier-Stokes equations and the linear particle equations are not constrained by numerical stability.

## 5.2.2 Forward-Time FTLE with Spectral Methods

The locations of the fluid particles in each subdomain that are traced over the time interval, *T*, are denoted by  $\Phi_{ij}$ . These locations construct the interpolant for the

flow map given by

$$\Phi_N(\xi, \eta, t) = \sum_{i=0}^N \sum_{j=0}^N \Phi_{i,j}(t) \ell_i(\xi) \ell_j(\eta).$$
(5.2)

The deformation gradient tensor is determined from  $\Phi_N$  using the DG operators (3.12) and (3.13). Taking into account the mapping to computational space, we compute the components of the deformation gradient tensor at the quadrature nodes as follows,

$$\frac{\partial \Phi}{\partial x_0} = \frac{1}{J} \left[ \left( \sum_{k=0}^N D_{ik}^{(\xi)} \Phi_{k,j} \right) \frac{\partial y_0}{\partial \eta} - \left( \sum_{k=0}^N D_{jk}^{(\eta)} \Phi_{i,k} \right) \frac{\partial y_0}{\partial \xi} \right], \\ \frac{\partial \Phi}{\partial y_0} = \frac{1}{J} \left[ \left( \sum_{k=0}^N D_{jk}^{(\eta)} \Phi_{i,k} \right) \frac{\partial x_0}{\partial \xi} - \left( \sum_{k=0}^N D_{ik}^{(\xi)} \Phi_{k,j} \right) \frac{\partial x_0}{\partial \eta} \right].$$
(5.3)

From the deformation gradient tensor, the FTLE is determined with (2.31).

# 5.2.3 Multiple Flow Maps From a Single Particle Trace

We can determine the flow map at time  $t_2 > t_1$ , over time interval  $T_2 = t_2 - t_1$ ,

$$\phi_{t_1}^{t_2} = \mathbf{x}^2(t_2; t_1, \phi_{t_0}^{t_1}), \tag{5.4}$$

with the same particle trace we used to determine the flow map at  $t_1$  over the time interval  $T_1 = t_1 - t_0$ ,

$$\phi_{t_0}^{t_1} = \mathbf{x}^1(t_1; t_0, \mathbf{x}^0), \tag{5.5}$$

if we take the flow map at  $t_1$  as the initial state for the flow map at  $t_2$ . To do so, we map and interpolate the  $N^{th}$ -order polynomial approximation of the flow map on a subdomain at  $t_1$ ,

$$\phi_{t_0}^{t_1} \approx \Phi_N, \tag{5.6}$$

to Gauss quadrature nodes, so that it can be used as the basis for the polynomial approximation of the map at  $t_2$ ,

$$\phi_{t_1}^{t_2} \approx \Psi_N. \tag{5.7}$$

Similarly, we find an approximation to the backward-time flow map at  $t_0$ ,

$$\phi_{t_1}^{t_0} \approx \Theta_N, \tag{5.8}$$

using the same mapped and interpolated flow map at  $t_1$ , (5.6) as a basis.

#### Mapping of Flow Map to Unit Square

To project the approximation of the flow map at  $t_1$ ,  $\Phi_N$ , to the quadrature nodes, the nodes,  $\Phi_{ij}$ , are first mapped to the master element with the inverse of the linear blending function (3.2). We parameterize the boundary of the map,  $\Gamma_i^{\phi}$ , in (3.2) with

$$\Gamma_i^{\phi} = \Phi_N|_{\partial D_k^i},\tag{5.9}$$

i.e. the polynomial approximation of the flow map,  $\Phi_N$ , at the edges of the subdomains,  $D_k$ , as shown in Fig. 5.1.

The particles that are initialized at the quadrature nodes at time  $t_0$  (Fig. 5.2a), and that are advected until time  $t_1$ , form the flow map,  $\Phi_N$  based on the particle nodes  $\Phi_{ij}$ , (filled circles, Fig. 5.2b). Through the inverse transfinite map function (3.2),  $\Phi_{ij}$ , maps to the nodes in the unit square,  $\alpha_{ij}$  (filled circles, Fig. 5.2c), where we invert the non-linear transfinite mapping function (3.2) with an iterative Newton method using the quadrature nodes as an initial guess.

The nodes  $\alpha_{ij}$  do not necessarily coincide with the quadrature nodes,  $\alpha'_{ij}$  (open



**Figure 5.1**: The nodes that define the flow map subdomain boundaries,  $\Gamma_i^{\phi}$  are constructed from the final locations of the nodes initialized at the boundaries of the subdomains at time  $t = t_0$  such that  $\Gamma_i^{\phi} = \Phi_N|_{\partial\Omega_i}$ .



**Figure 5.2**: The fluid tracers are initialized at the Chebyshev-Lobatto quadrature nodes at time  $t = t_0$ , designated by the filled circles (a). The particles are then advected to time  $t_1$  (filled circles (b)), where they are mapped to the master element with the inverse map  $\mathbf{x}(\alpha)^{-1}$  (filled circles (c)). The open circles in (b) and (c) correspond to the quadrature nodes in physical space,  $\Phi'_{ij}$ , and mapped space,  $\alpha'_{ij}$ , respectively.

circles, Fig. 5.2c), i.e.,

$$\alpha_{ij} \neq \alpha'_{ij}, \tag{5.10}$$

except along the parameterized curves that bound the subdomain, and in the trivial case of uniform flow.

#### **Orthogonal Basis for the Integrated Flow Map**

On the unit square, we formulate the following interpolant,

$$\Phi_N(\alpha,\beta) = \sum_{i=0}^N \sum_{j=0}^N \Phi'_{ij} \ell_i(\alpha) \ell_j(\beta).$$
(5.11)

Here,  $\Phi'_{ij}$  are values of the flow map at  $t_1$  on the quadrature nodes. They are obtained using the known values of the flow map,  $\Phi_{lm}$ , at the coordinates  $(\alpha_l, \beta_m)$  in the unit square. So,

$$\Phi_{lm} = \sum_{i=0}^{N} \sum_{j=0}^{N} \Phi'_{ij} \ell_i(\alpha_l) \ell_j(\beta_m),$$
(5.12)

which in matrix-vector form can be expressed as

$$\tilde{\Phi}_q = \tilde{\Phi}'_p I_{pq}, \tag{5.13}$$

where  $I_{pq}$  is the  $(N+1)^2 \times (N+1)^2$  interpolation matrix,

$$I_{pq} = \ell_i(\alpha_l)\ell_j(\beta_m), \tag{5.14}$$

and  $\Phi_q$  and  $\Phi'_p$  are contiguously aligned, so that p = i(N+1) + j and q = l(N+1) + m. Finally, by inverting (5.13) we have

$$\tilde{\Phi}'_p = I_{pq}^{-1} \tilde{\Phi}_q. \tag{5.15}$$

The flow map,  $\tilde{\Phi}'_p$  is based an orthogonal, tensorial, quadrature grid and forms the basis at  $t_1$  for new flow maps in forward and backward time.

#### **Construction of Backward-Time Flow Maps**

We find the backward-time flow map at  $t_0$  by integrating the known flow map at  $t_1$  to  $t_0$ ,

$$\Theta_{ij} = \Phi_{ij} + \int_{t_1}^{t_0} \left(\frac{\mathrm{d}\Phi_{ij}}{\mathrm{d}t}\right) \mathrm{d}t$$
(5.16)

Since (5.16) back-traces the particles that we initiate at  $\mathbf{x}_0$  to obtain  $\Phi_N$  in forward time, it follows that  $\Theta_{ij}$  represent the nodes on the initial DG-mesh,  $\mathbf{x}_{ij}^0$ , (Fig. 5.3b).

With respect to the flow map,  $\Phi'_{ij}$ , on the orthogonal quadrature nodes at  $t_1$ , we find the flow map in backward time at  $t_0$  (Fig. 5.3c) as

$$\tilde{\Theta}'_p = I_{pq}^{-1} \tilde{\Theta}_q, \tag{5.17}$$

according to the definitions in (5.13) and (5.14).

We prove (5.17) by multiplying (5.16), recast in matrix-vector form, with  $I_{pq}^{-1}$ ,

$$I_{pq}^{-1}\tilde{\Theta}_{q} = I_{pq}^{-1}\tilde{\Phi}_{q} + I_{pq}^{-1}\int_{t_{1}}^{t_{0}} \left(\frac{d\tilde{\Phi}_{q}}{dt}\right)dt.$$
 (5.18)

Since the operators are linear we can rewrite this as

$$I_{pq}^{-1}\tilde{\Theta}_q = I_{pq}^{-1}\tilde{\Phi}_q + \int_{t_1}^{t_0} \left(\frac{\mathrm{d}I_{pq}^{-1}\tilde{\Phi}_q}{\mathrm{d}t}\right)\mathrm{d}t.$$
(5.19)

With (5.15) and (5.17) it follows that

$$I_{pq}^{-1}\tilde{\Theta}_q = \tilde{\Phi}'_p + \int_{t_1}^{t_0} \left(\frac{\mathrm{d}\tilde{\Phi}'_p}{\mathrm{d}t}\right) \mathrm{d}t.$$
(5.20)

The right hand-side represents the integration of orthogonal flow map at  $t_1$  to  $t_0$  and thus is equal to  $\tilde{\Theta}'_p$ . This proves (5.17).

Since  $\tilde{\Theta}'_p$  represent an approximation of the backward-time flow map on the

orthogonal quadrature basis, we can use spectral operators like in (5.3) to determine the deformation gradients from the flow map.

Alternatively, the backward-time FTLE can be computed using a method by Haller and Sapsis [1], who derive an analytical relationship between the largest backward-time FTLE and the smallest forward-time FTLE. This enables one to compute the backwardtime FTLE associated with the flow map,  $\phi_{t_1}^{t_0}$ , from the forward-time flow map,  $\phi_{t_0}^{t_1}$ , in the following way:

$$\sigma(\phi, -T) = \frac{1}{|T|} \ln \sqrt{\lambda_{max} \left(\frac{\partial \mathbf{x}_0}{\partial \phi}^* \frac{\partial \mathbf{x}_0}{\partial \phi}\right)}$$
$$= -\frac{1}{|T|} \ln \sqrt{\lambda_{min} \left(\frac{\partial \phi}{\partial \mathbf{x}_0}^* \frac{\partial \phi}{\partial \mathbf{x}_0}\right)}.$$
(5.21)

The backward-time FTLE computed from (5.21) is determined on the same deformed grid at  $t_1$  that we use for our DG-FTLE algorithm, and so suffers the same computational challenge as the present work. Although (5.21) presents an efficient and accurate method for computing the backward-time FTLE field from the forward-time flow map, the method Haller and Sapsis [1] present does not allow for the determination of the backward-time flow map, whereas our DG-based method does. Work by Karrasch et al. [105] implies that determination of the backward-time flow map will reduce the computational costs and increase the numerical stability associated with the identification of repelling hyperbolic LCS according to the geodesic theory by Haller [67]. Hence, extensions of this algorithm to identify LCS from geodesic theory will benefit from the high-order accurate backward-time flow map computed from the forward-time flow map afforded by this algorithm. We compare our DG-based approach with the approach by Haller and Sapsis on a vortex test case in section 5.4.



**Figure 5.3**: With the fluid tracers integrated from time  $t_0$  (b) to  $t_1$  (a), then to  $t_2$  (d), we can find the locations of the quadrature nodes at  $t_2$  (e) and  $t_0$  (c) that map from the master element to the flow map at  $t_1$  using interpolation if we invert the interpolation matrix  $I_{pq}$ .

#### **Multiple Forward-Time Flow Maps From a Single Particle Trace**

Similar to the backward-time flow map, we find new flow maps in forward-time at time  $t_2 > t_1$  with

$$\Psi_{ij} = \Phi_{ij} + \int_{t_1}^{t_2} \left(\frac{\mathrm{d}\Phi_{ij}}{\mathrm{d}t}\right) \mathrm{d}t, \qquad (5.22)$$

from a particle released at the single initial time,  $t_0$ . Following Fig. 5.3a, d and e, the flow map at  $t_2$  is determined with respect to the flow map,  $\Phi'_{ij}$ , on the orthogonal quadrature nodes at  $t_1$ , as

$$\tilde{\Psi}'_p = I_{pq}^{-1} \tilde{\Psi}_q, \tag{5.23}$$

according to the definitions in (5.13) and (5.14).

We can reiterate the projection to an orthogonal basis at  $t_2$  and find as many flow maps as we desire from a single initial time particle trace and continue to use consistent spectral differentiation to find the deformation gradient tensor and its corresponding FTLE.

#### Notes on the DG-FTLE from single particle trace

Pertaining to the use of a single particle trace to determine multiple forward and backward FTLEs, we make the following remarks:

**Remark 1:** As the flow map time interval increases, gradients in  $\Phi_N$  increase and hyperbolicity in the flow field causes the flow map topology to become increasingly tangled. This translates to Gibbs oscillations and aliasing errors in the approximation of the flow map.

**Remark 2:** The distance between the quadrature nodes,  $\alpha'_{ij}$ , and the inverse-mapped nodes,  $\alpha_{ij}$ , increases as the flow map gradients increase with time. Since the Newton method uses  $\alpha'_{ij}$  as an initial guess, increasing *T* will affect the convergence of the inverse map.

**Remark 3:** It is well known that deviations from near-optimal nodal points, such as the Chebyshev Gauss-Lobatto nodes, increase the Lebesgue constant,  $\Lambda_N(I)$  [106], which is connected to poor conditioning of the interpolation. Therefore, since *I* depends on  $\alpha'_{ij}$ as in (5.14), as the time interval is increased,  $\Lambda_N(I)$ , the conditioning and interpolation errors increase. As the polynomial order, *N*, increases,  $\Lambda_N(I)$  further increases [106]. **Remark 4:**  $I_{pq}$  is easily inverted with

$$I_{pq}^{-1} = [\ell_i(\alpha_l)\ell_j(\beta_m)]^{-1} = \ell_l(\alpha_i')\ell_m(\beta_j'),$$
(5.24)

where p and q are the same as in (5.14).

While the conditioning can lead to large interpolation errors, it also offers insights into the deformation of the fluid that compliment the FTLE. So, in addition to the FTLE, we have more measures for the deformation of the subdomains, including: 1) the Lebesgue constant,  $\Lambda_N$ ; 2) the condition number,  $\kappa$ , of *I*, given by

$$\kappa(I) \equiv \frac{\lambda_{max}(I)}{\lambda_{min}(I)};$$
(5.25)

and 3) the separation  $|\alpha_{ij} - \alpha'_{ij}|$ , which we reduce to a single parameter  $\Delta \alpha$  defined by

$$\Delta \alpha \equiv \max |\alpha_{ij} - \alpha'_{ij}|. \tag{5.26}$$

In general, the Lebesgue constant is difficult to determine for an arbitrary two-dimensional set of interpolation nodes. Hence, we do not include  $\Lambda_N$  in our analysis. In the vortex and cylinder test cases, we illustrate the correlation between  $\kappa(I)$  and  $\Delta \alpha$  and the FTLE.

# 5.2.4 Algorithm Summary

The algorithm that determines several FTLEs from a single particle trace is summarized as follows:

- A. Particle tracing algorithm in DG:
  - 1. Initialize fluid tracers on Chebyshev Gauss-Lobatto nodes.
  - 2. Remove double particles at subdomain boundaries.
  - 3. The velocity solution of a DG Navier-Stokes solver is interpolated to each particle's location.
  - 4. The particles are advected one step in time using a 3<sup>rd</sup>-order Adams-Bashforth scheme.
  - 5. Particle search algorithm:
    - i. The particle is mapped to its last known host domain.
    - ii. If the particle leaves the host domain, a new host domain is found from the host domains connected to the nearest node.

- B. Forward-time FTLE using DG at time  $t_1$ :
  - 1. The deformation gradient tensor is determined using spectral operators on the flow map  $\Phi'_{ij}$ .
  - 2. The Cauchy-Green deformation tensor is constructed.
  - 3. The FTLE is computed according to (2.30).
- C. The FTLE at time  $t_2$ :
  - 1. Given the nodes of the flow map approximations,  $\Phi_{ij}$  and  $\Psi_{ij}$ , find the node locations  $\alpha_{ij}$  using the inverse of (3.2).
  - 2. Construct the interpolation operator matrix  $I_{pq}$  from the nodes  $\alpha_{ij}$  and the quadrature nodes  $\alpha'_{ij}$ .
  - 3. Construct the vector  $\tilde{\Psi}_q$  from  $\Psi_{ij}$  and compute  $\tilde{\Psi}'_p$  from (5.23).
  - 4. Once  $\Psi'_{ij}$  is computed, the FTLE can be computed using (5.3) and (2.30).
- D. Backward-time FTLE:
  - 1. Given the interpolation matrix  $I_{pq}$ , construct the vector  $\tilde{\Theta}_q$  from  $\Theta_{ij}$  and compute  $\tilde{\Theta}'_p$  from (5.17).
  - 2. Once  $\Theta'_{ii}$  is computed, the FTLE is computed using (5.3) and (2.30).

# **5.3 Forward-Time FTLE Test Cases**

We test the spectral forward-time FTLE algorithm using three benchmark cases, including a periodic gyre, a vortex advected in an inviscid flow field, and the viscous flow around a square cylinder. Details of the velocity fields for these cases are given in Section 5.4

## 5.3.1 Gyre Flow

The gyre flow is defined by the following velocity field [59]:

$$u(x, y, t) = -\pi A \sin(\pi f(x, t)) \cos(\pi y)$$
$$v(x, y, t) = \pi A \cos(\pi f(x, t)) \sin(\pi y) \frac{\partial f}{\partial x},$$
(5.27)

where we have

$$f(x,t) = a(t)x^{2} + b(t)x,$$
(5.28)

$$a(t) = \varepsilon \sin(\omega t), \tag{5.29}$$

$$b(t) = 1 - 2\varepsilon \sin(\omega t). \tag{5.30}$$

While this is not a solution to the fluid conservation laws, it represents a simplified model of a generic gyre flow, which is a common flow type in geophysical systems. When we set t = 0, we have a simplified, time-independent flow given by

$$u(x,y) = -\pi A \sin(\pi x) \cos(\pi y),$$
  
$$v(x,y) = \pi A \sin(\pi y) \cos(\pi x).$$
 (5.31)

As seen in Fig. 5.4a, the time-independent gyre velocity field consists of spatially periodic rotating square cells that are one unit wide. The FTLE field (Fig. 5.4b) is a spatially periodic arrangement of ridges bounding the rotating cells seen in the vector field. As the time interval is increased, the gradient around the ridges increases and hence the ridges become sharper.

As the time interval is increased, the gradient around the ridges in the FTLE field



Figure 5.4: Velocity field and FTLE field for the periodic gyre flow.

increases and the ridges become sharper. At high polynomial orders, the approximation of this ridge can become oscillatory. However, this can be mitigated with grid refinement near the ridges, as in Fig. 5.11b.

The high-order algorithm converges spectrally, as compared to the algebraic convergence of the second-order finite difference method (Fig. 5.5). Additionally, when we use a grid with a cosine distribution, where subdomains are refined along the ridges, the accuracy and convergence rate increases over a regular grid implementation. Although it is not always possible to know the locations of FTLE ridges *a priori*, it is a general trend for the solution of conservation laws that ridges tend to form in flow regions that require high grid resolution, such as near walls and in turbulent regions.



Figure 5.5: Comparison of the convergence rates for the spectral element approach and the finite difference approach.

## 5.3.2 Vortex Advected by Uniform Flow

We consider the benchmark problem of a circular, isentropic vortex that is advected by a uniform flow. Following [107], the velocity field is given by

$$u(x,y,t) = U_{\infty} - \frac{A}{R}(y - Y_c) \exp\left(-\frac{(x - X_c - U_{\infty}t)^2 + (y - Y_c)^2}{2R^2}\right),$$
  
$$v(x,y,t) = \frac{A}{R}(x - X_c - U_{\infty}t) \exp\left(-\frac{(x - X_c - U_{\infty}t)^2 + (y - Y_c)^2}{2R^2}\right),$$
(5.32)

and the temperature field by

$$T = 1 - \frac{1}{2} (AM_{\infty})^2 \exp\left(-\frac{(x - X_c - U_{\infty}t)^2 + (y - Y_c)^2}{R^2}\right) (\gamma - 1).$$
(5.33)

where  $X_c$  and  $Y_c$  are the initial coordinates of the vortex center. The initial conditions given by (5.32) and (5.33) are solutions to the Euler equations. The velocity field at initial time  $t_0$  with the vortex center located at  $(X_c, Y_c) = (0.5, 0.5)$  is shown in Fig. 5.6a. The uniform free-stream velocity is set to  $U_{\infty} = 1$  and the radius is R = 0.1. The vortex strength *A* is set to 0.2.

Although the exact velocity is known for this flow, the velocity was computed



Figure 5.6: Velocity field for the periodic vortex flow (a) and the FTLE field for T = 0.5 (b).

using the DGSEM method to solve the Euler equations. The FTLE was computed simultaneously during the computation of the velocity field in order to test this algorithm's effectiveness as an "on-the-fly" method for computing the FTLE. The resulting FTLE field is shown in Fig. 5.6b for a polynomial order of 9 and a time interval  $t - t_0 = 0.5$ .

Since we have an analytical solution for the flow field, flow map, and FTLE field, we are able to directly measure the error and compare error sources more rigorously. As shown in Fig. 5.7, the fluid solution, represented by the u-velocity and v-velocity, converge spectrally. The error in the flow map is affected by the high-order interpolation of the velocity field on the fluid particles and the third-order time stepping scheme used to advect the particles. Since the interpolation is high-order and the time stepping is algebraic, we expect the flow map error to be dominated by the time stepping error and, hence, be nearly algebraic. This is shown in Fig. 5.7 by the  $X_p$  and  $Y_p$  curves, which converge more slowly than the fluid solution. The error in the FTLE is a combination of the error in the fluid solution, the velocity interpolation, the fluid particle time integration, and the high-order derivatives used to compute the deformation gradient. Since most



**Figure 5.7**: Error sources in the "on-the-fly" implementation. The largest errors are present in the flow map, while the velocities and FTLE field demonstrate spectral convergence.

of the error sources are from spectral methods, we expect the FTLE error to converge spectrally until the time stepping error becomes dominant at high polynomial orders. In Fig. 5.7, we see that the FTLE does indeed converge spectrally until about P = 6, at which point the flow map error begins to dominate and the convergence rate decreases.

#### **5.3.3** Flow around a square cylinder

In our last case that we considered, the compressible Navier-Stokes equations were solved to simulate the viscous flow around a square cylinder [62, 17]. This test case comprises a more complex geometry to illustrate the application of the algorithm developed in this paper to complex, time-dependent flows. The simulation was performed on a rectangular computational domain shown in Fig. 5.22. The reference length of the flow is the cylinder width, c=1. The computational domain size is  $40c \times 20c$  with the center of the cylinder located 10c from the left boundary and 10c from the top and bottom boundaries. Uniform inflow conditions are set on the left boundary, with uniform outflow conditions set at the top, bottom and right. The cylinder wall is adiabatic. The Reynolds number and Mach number based on the free-stream velocity and cylinder width



Figure 5.8: Problem setup for square cylinder case.

are Re = 150 and M = 0.3 respectively.

This simulation was run at a polynomial order of 6. Figure 5.9 shows the fluid solution using 2 different grid resolutions. The finer mesh is four times the resolution of the course mesh. Figure 5.10 shows the FTLE field and the x-flow map for the 2 grids. There are visible jumps at the subdomain interfaces in the fluid solution using the coarse mesh, which indicate that this solution is not fully resolved. The fluid solution is very smooth using the refined mesh, hence it is more resolved than that of the coarse mesh solution. As shown in Fig. 5.10, the flow map using the refined mesh is much smoother than that using the course mesh. This leads to a much smoother result in the FTLE field on the fine mesh as well. However, the steepest ridges in the near wake and up stream of the cylinder due to the high gradients associated with the steeper ridges. Resolution requirements in numerical studies are strongly linked to and often dictated by gradients in the solution field, hence the high gradients in the FTLE field, which



**Figure 5.9**: Convergence of the fluid solution can be seen in the vorticity plot using the most refined mesh, where the defects at the subdomain boundaries are no longer visible. The finer mesh is four times the resolution of the coarse mesh.

originate in the flow map, are the cause of the errors seen in Fig. 5.10. Since there is no requirement that the polynomial order of the solution must be consistent with the polynomial order of the FTLE computation, a higher polynomial can be used to compute the FTLE by interpolating the fluid solution onto the higher resolution FTLE grid. To improve efficiency in future implementations of this algorithm, an adaptive routine can be implemented that increases the polynomial order in a subdomain when a certain deformation gradient threshold is reached.

# 5.4 DG-FTLE Test Cases

We test the FTLE algorithm for three cases; a steady, spatially periodic gyre flow with a prescribed velocity field that is not a solution to the Navier-Stokes equations; a vortex advected by a uniform, inviscid flow, with a velocity field that is computed with a DG method; and the complex, viscous flow around a square cylinder.



**Figure 5.10**: Improvement in resolution of the FTLE is visible as the mesh becomes more refined. However, the most refined mesh still exhibits defects in the sharp FTLE ridges in the near wake. The gradients in the flow map are generally much greater than the gradients in the corresponding fluid solution. Hence, the FTLE field, in regions of high FTLE, will generally require more resolution than the fluid solution. The same two meshes that were used in Fig. 5.9 are used in this figure.

# 5.4.1 Gyre Flow

We determine the FTLE field on two grids for this case; one uniform grid (Fig. 5.11a) and a grid with increased resolution at the locations of the ridges (Fig. 5.11b) using a cosine distribution.

As a reference, we consider a highly resolved computation based on a cosine grid with  $32 \times 32$  subdomains, a polynomial order of 12, and a time step,  $\Delta t = 0.00125$ . Coarser computations on a  $16 \times 16$  grid, with  $\Delta t = 0.0025$  and polynomial orders from P = 2 to P = 7 are compared to the reference case to determine the errors. The FTLE is determined at T = 2, 3, and 4 for each case.

Fig. 5.12 shows that the flow map  $\Phi_N$  converges spectrally to the fine grid solution at time  $t_1$ , while, as *T* increases, the error increases. The forward-time FTLE at  $t_2 = t_0 + 2T$  and the backward-time FTLE computed from the same trace converge spectrally as well (Fig. 5.13). Since the gradients are larger in  $\Psi_N$  as compared to  $\Theta_N$ , the



**Figure 5.11**: Uniform distribution of subdomains (a) and a cosine distribution where the subdomains are clustered near the FTLE ridges (b).

error in the deformation gradient tensor increases and hence the error in the forward-time FTLE is larger than in the backward-time FTLE.

The time-dependent gyre flow provides a quantitative test of the accuracy of the algorithm on a simple FTLE field with an LCS that is not grid-aligned, in contrast to the FTLE field for the steady gyre case. The unsteady gyre flow is given by u(x, y, t) and v(x, y, t) in (5.27) and is a time-periodic perturbation in the *x*-direction of the steady gyre flow. Time-dependence of the flow field leads to a curved FTLE ridge separating the gyres (Figure 5.14). In this study, we set  $\varepsilon = 0.1$ , A = 0.1, and  $\omega = 2\pi/10$ . The FTLE is computed over the domain  $[0, 2] \times [0, 1]$ . The convergence of the error for the FTLE field computed over time intervals of T = 3 and 4 is exponential as depicted in Figure 5.15.

## 5.4.2 Vortex Advected by Uniform Flow

The flow map and FTLE field for this velocity field can be determined analytically, as mentioned in [76]. We present the solution in A.



Figure 5.12: Convergence for the FTLE fields computed from  $\Phi_N$  at time intervals T = 2, 3, and 4.



**Figure 5.13**: Convergence for the FTLE fields computed from  $\Psi_N$  (forward-time) and  $\Theta_N$  (backward-time) at time intervals T = 2, 3, and 4.

#### **Errors in the Forward-Time Flow Map**

The analytical solution to the vortex case provides a reference for a detailed analysis of the numerical errors. There are a number of sources of error in the numerical algorithm to compute the flow map,  $\Phi_N$ , including the error in the DG computation of the velocity field, and the error in the flow map,  $\Phi_N$ , with contributions from the numerical integration of the fluid tracers, the grid spacing, and the interpolation error, which in turn is affected by high gradients in the flow map. For a detailed discussion of the fluid particle tracking algorithms and accuracy and convergence of DG methods, we refer to



Figure 5.14: Time-dependent gyre FTLE field for a time interval of T = 4.

[104] and [21], respectively. In this work, we ensure that the numerical errors from the DG method and fluid particle integration algorithm are small compared to the flow map error so that the error in the FTLE determination can be examined. In order to quantify the local error of the flow map, we compute the interpolation nodes exactly and compute the difference between the exact flow map,  $\phi$ , and the polynomial approximation,  $\Phi_N$ , given by

$$\varepsilon_{\phi} = |\Phi_N - \phi|. \tag{5.34}$$

As the integration time, T, increases, the flow map topology becomes more distorted and tangled (Fig. 5.16), leading to large gradients. Therefore, the interpolation error must increase with increasing T. The root-mean-square error,  $\varepsilon_{RMS}$ , of  $\Phi_N$  is given by

$$\varepsilon_{RMS} = \sqrt{\frac{1}{M} \left( \sum_{i=1}^{M} |\Phi_N - \phi|^2 \right)},\tag{5.35}$$

where *M* is the number of interpolation points. A comparison of  $\varepsilon_{RMS}$  computed at T = 0.5 and T = 5 is plotted in Fig. 5.17 versus the polynomial order, *P*. Spectral



Figure 5.15: Convergence for the FTLE fields computed from  $\Phi_N$  at time intervals T = 3 and 4 for the time-dependent gyre flow.



Figure 5.16: Subdomain deformation at T = 0.5 (a) and T = 2.5 (b).

accuracy is seen in  $\Phi_N$  for both T and with increasing T,  $\varepsilon_{RMS}$  increases significantly.

#### **Errors in the Flow Map Interpolation**

In addition to the error introduced by the flow map,  $\Phi_N$ , the errors in the new flow maps  $\Psi_N$  and  $\Theta_N$  are affected by the conditioning of the interpolation matrix I. In Fig. 5.18, the maximum condition number,  $\kappa(I)$ , and the maximum value of the parameter  $\Delta \alpha$  are plotted as a function of time for a 4<sup>th</sup>-order and 6<sup>th</sup>-order solution. As Fig. 5.18 illustrates,  $\kappa$  increases as the time interval increases, as does  $\Delta \alpha$ . In both cases,



Figure 5.17: The convergence of the RMS error of  $\Phi_N$  at two times, T = 0.5 and T = 5.0.

a reduction in the magnitude of the parameters is observed when the grid is refined. The relationship between  $\kappa$  and the Lebesgue constant,  $\Lambda_N$  results in an increase in  $\kappa$  when *P* increases. However,  $\Delta \alpha$  exhibits no dependency on *P*. This is because  $\Delta \alpha$  depends only on the relationship between the topology of the flow map,  $\Phi_N$ , and the transfinite mapping function,  $\mathbf{x}(\alpha)$ .

Both the forward and backward-time flow maps are spectrally convergent for two different time intervals (Fig. 5.19). However, the magnitude of the error for the T = 5 flow maps is greater than the error for the T = 0.5 flow maps. Fig. 5.20 depicts the convergence of the FTLE computation from the flow maps  $\Psi_N$  and  $\Theta_N$  with a time interval T = 0.5. Given that  $\Psi_N$  and  $\Theta_N$  converge spectrally, their respective FTLEs converge spectrally since spectral operators are used to compute the deformation gradient tensor.

The backward-time FTLE from time interval  $[t_1, t_0]$  computed using the method presented by Haller and Sapsis [1] is compared to the method presented in this work and the convergence of  $\varepsilon_{RMS}$  is depicted in Fig. 5.21. The error in the method presented in this work compares well with the error in the Haller-Sapsis method. Differences arise from errors due to the poor conditioning of the interpolation problem at high deformation


Figure 5.18: Dependency of maximum condition number on polynomial order and subdomain deformation. The refined case uses a uniform grid of  $\Delta x = 0.05$ .



**Figure 5.19**: The convergence rates of the RMS error of the backward-time and forward-time flow maps found using matrix inversion at two time intervals, *T*.

and high polynomial orders in the method in this work, whereas the Haller-Sapsis method does not require interpolation and, hence, does not suffer from interpolation errors.

### 5.4.3 Flow around a square cylinder

We consider the unsteady flow around a square cylinder [62, 17]. The results presented in this section are based on the refined grid solution. The fluid computation is performed until quasi steady-state is reached [62, 17].



Figure 5.20: The convergence rates of the backward and forward FTLE fields at time interval T = 0.5.



Figure 5.21: The error in the backward-time FTLE computed at two time intervals — T = 1.5 and T = 2.5 — using the method in Haller and Sapsis [1] and this work.

A von Karman vortex street develops in the wake of the cylinder, and provides velocity fields that are well suited for study using LCS methods [32]. The forward-time FTLE field exhibits a strong repelling LCS along the stagnation streamline upstream of the cylinder, while the vortex boundaries can be seen in the wake (Fig. 5.23a). In the backward-time FTLE field, strong attractors form in the near wake, where fluid becomes entrained just behind the cylinder in a near-wake recirculation region (Fig. 5.23b). Additional attractors form at the vortex cores and along material lines connecting the vortices. The time interval is T = 10, and the FTLE fields are computed using the



Figure 5.22: Problem setup for square cylinder case.

algorithm presented in [76].

The effect of the conditioning of *I* and the errors in the inverse transfinite map is most significant in the regions with the highest gradients in the flow map. This generally correlates to regions with high FTLE values, as seen in Fig. 5.24. In Fig. 5.24b, the log of the condition number,  $\ln(\kappa)$ , is plotted. The subdomains with the highest ridges near the cylinder and along the stagnation streamline exhibit the largest  $\kappa$  values, which tends to wash out the more subtle increases in  $\kappa$  around the ridges that form in the wake. Alternatively, the parameter,  $\Delta\alpha$ , in Fig. 5.24c, scales in a way that allows improved visualization of the flow structures in the far wake.

Fig. 5.25 shows the backward-time FTLE determined using the standard approach, in which the fluid tracers are integrated backward in time (Fig. 5.25a), and the backward-time FTLE determined from the forward-time flow map using the algorithm presented in this work. Since the backward-time FTLE in Fig. 5.25b is computed on a highly deformed grid, the values of the FTLE are interpolated onto a uniform grid using cubic



Figure 5.23: Comparison of the forward-time FTLE field (a) and the backward-time FTLE field (b). The time interval is T = 10.

interpolation so that the results can be better visualized. The high FTLE values in the left side of Fig. 5.25b are a result of the lack of data in that region since the tracers that were initialized upstream of the cylinder have advected downstream. Both methods agree very well and depict the same stable manifolds in the wake with minor differences in the near wake resulting from the poor conditioning of the subdomains in those areas for Fig. 5.25b.

The forward-time FTLE field is depicted in Fig. 5.26 for the time interval T = 10and at the start times  $t_0 = 0$  (5.26a),  $t_0 = 1$  (5.26b), and  $t_0 = 2$  (5.26c). The frames in Fig. 5.26 were determined from a single particle integration, initialized at t = 0, illustrating the potential for the numerical method presented in this work to compute FTLE fields at different starting states while only integrating one set of particles.

### 5.5 Summary of Results

An algorithm is developed that computes FTLE fields with DG operators at multiple times, both in backward-time and in forward-time, from a single particle trace. The algorithm is designed to be efficient and consistent in implementation and accuracy with high-order DG methods used to solve the Navier-Stokes equations for the underlying velocity vector fields and can be used simultaneously with the integration of the Navier-Stokes equations, as opposed to post-processing.

By seeding particles on the DG nodal mesh points, the integrated flow maps find their basis in the piecewise representation of the DG mesh, ensuring spectral convergence of deformation gradients and FTLE fields if DG operators are used.

Projection to quadrature on a unit square of integrated flow maps that are initiated on nodal points provides a basis for the determination of forward and backward flow maps at several times from particles that are seeded at a single initial time. Conditioning of the projection is poor when the flow map deforms significantly in time. Quantitative measures for the conditioning also provide additional information regarding the deformation of the flow field.

The algorithm is tested on three benchmarks: a spatially periodic gyre flow; the advection of a vortex in a uniform, inviscid flow; and the unsteady, viscous flow around a square cylinder. Spectral convergence is demonstrated on the gyre case and the vortex case.

The sources for error are examined in detail on the gyre and vortex cases. The most significant source of error is in the interpolation operator, I, where the conditioning for the interpolation depends on the deformation of the flow map. As the time interval, T, increases, the increasing deformation of the subdomains with high flow map gradients leads to poor conditioning for the interpolation to other flow maps, as measured by the

condition number,  $\kappa$ , and the parameter  $\Delta \alpha$ . The parameters  $\kappa$  and  $\Delta \alpha$  are shown to highly correlate with the locations of ridges in the FTLE field for the square cylinder case.

The method for computing the backward-time FTLE from the forward-time flow map using DG operators is compared to the method proposed by Haller and Sapsis. The accuracy and convergence of the DG method compare well with the Haller-Sapsis method. The DG method has the advantage of computing the backward-time flow map directly from the forward-time flow map, providing for the ability to analyze the backwardtime flow map if needed. The Haller-Sapsis method, on the other hand, determines the backward-time FTLE directly from the forward-time flow map, without determining the backward-time flow map at any point. The DG method, has the drawback that it is more computationally intensive than the Haller-Sapsis method.

The present work focuses on implementation of the algorithm in two dimensions. Extension to three dimensions is straightforward. In future work, we aim to present test cases in three dimensions. The algorithm can be extended to compute the eigenvector field of the Cauchy-Green strain tensor with spectral convergence. Therefore, hyperbolic material surfaces can be extracted with high-order accuracy using the LCS variational theory. We aim to examine the application of this algorithm to the work of Farazmand and Haller [69]. Our future efforts will further focus on developing an adaptive mesh algorithm for improving the conditioning of I and extending the time interval over which the algorithm can be applied, as well as parallelization, and improvements in efficiency.

Chapter 5, in part, is a reprint of the material as it appears in Journal of Computational Physics, vol. 295, 2015, Nelson, Daniel; Jacobs, Gustaaf, "DG-FTLE: Lagrangian Coherent Structures with Discontinuous-Galerkin Methods". The dissertation author was the primary investigator and author of this paper.

Section 5.3, in part, is a reprint of the material as it appears in The Proceedings of

the ASME 2013 International Mechanical Engineering Congress and Exposition, 2013, Nelson, Daniel; Jacobs, Gustaaf, "Computation of Forward-Time Finite-Time Lyapunov Exponents Using Discontinuous-Galerkin Spectral Element Methods", The dissertation author was the primary investigator and author of this paper.



**Figure 5.24**: Comparison of the forward-time FTLE field (a) to the log of the condition number,  $\ln(\kappa)$ , (b) and the parameter  $\Delta\alpha$  (c). The time interval is T = 10.







**Figure 5.26**: The forward-time FTLE computed with a time interval T = 10 at times  $t_0 = 0$  (a), 1 (b), and 2 (c) with one particle trace.

# Chapter 6

# High-Order Visualization of Three-Dimensional Lagrangian Coherent Structures with DG-FTLE

Sections 6.2 through 6.4 are reprinted from an upcoming manuscript to be published in Computers and Fluids in 2016 [108].

## 6.1 Overview and Summary

In this chapter, we extend the DG-FTLE algorithm presented in [75] to separated flows over curved geometries and flows in three-dimensions. The algorithm is applied on boundary fitted grids with curved subdomain edges. The FTLE field is used to identify vortex structures in the wake of an airfoil in two and three dimensions. Implementation of the DG-FTLE algorithm in parallel for efficient three-dimensional computation and integration into DG fluid solvers is discussed. An exponential filter is applied to the flow map to smooth nearly discontinuous regions and remove Gibbs oscillations from the FTLE field when the integration time is large. The method for constructing multiple flow maps from a single particle trace using interpolation developed in Chapter 5 is demonstrated in two and three-dimensions, and the conditioning of the interpolation is shown to be improved by *h*-adaptivity of the grid.

In the next section, the FTLE algorithm from [75] is presented in three dimensions, followed by a discussion of the parallel implementation and the exponential filter. Next, the algorithm is thoroughly assessed with three test cases to demonstrate the method on curved geometries in two and three-dimensions. The first is the two-dimensional separated flow over a NACA 65-(1)412 airfoil at low Reynolds number. The second is the canonical three-dimensional ABC flow, followed by the three dimensional flow over the NACA 65-(1)412 airfoil. In the final section, we discuss conclusions and future research.

### 6.2 Forward-Time FTLE with Spectral Methods

To determine the forward FTLE field, the subdomain mesh for the DG-FTLE algorithm is the same as the element mesh used for the DG-based fluid solver. However, the fluid solver element data structures are decoupled from the DG-FTLE subdomain data structures, as two differences exist. First, the FTLE field requires finer resolution than the fluid solver, hence different polynomial orders are used between the two grids. Second, the DG-FTLE grid uses Lobatto quadrature nodes. Particles initialized at the Lobatto node locations are duplicated at the boundaries between adjacent subdomains. These duplicate particles are removed so that fewer particles are integrated.

Metric terms for the DG-FTLE subdomains are determined from the transfinite mapping (3.2). The edges,  $\Gamma_i$ , and faces,  $\Sigma_i$ , are defined by Chebyshev-Lobatto nodes in one-dimension and two-dimensions respectively, to be consistent with the subdomain



**Figure 6.1**: The edges,  $\Gamma_i$ , and faces,  $\Sigma_i$  uniquely define a transfinite map for a subdomain and are constructed from the Chebyshev-Lobatto nodes with the same order as the subdomain discretization.

discretization (Fig. 6.1). In Lagrange form, the faces and edges are

$$\Sigma(\xi, \eta) = \sum_{i=0}^{N} \sum_{j=0}^{N} \mathbf{x}_{i,j} \ell_i(\xi) \ell_j(\eta), \qquad (6.1)$$

and

$$\Gamma(\xi) = \sum_{i=0}^{N} \sum_{j=0}^{N} \mathbf{x}_i \ell_i(\xi), \qquad (6.2)$$

respectively. Subdomain faces that lie on curved boundaries are constructed by fitting the face nodes,  $\mathbf{x}_{i,j}$ , to the boundary, so that the face forms a high-order approximation to the boundary.

Fluid particles are initialized at the Chebyshev-Lobatto quadrature nodes in each subdomain and are integrated in time over the interval *T*. A detailed description of the fluid particle integration algorithm is given in [75] and [104]. The locations of the fluid particles in each subdomain that are traced over the time interval, *T*, are denoted by  $\Phi_{ijk}$ . The values  $\Phi_{ijk}$  are the nodes of the high-order polynomial approximation of the flow

map given by

$$\Phi_N(\xi,\eta,\zeta,t) = \sum_{i=0}^N \sum_{j=0}^N \sum_{k=0}^N \Phi_{i,j,k}(t) \ell_i(\xi) \ell_j(\eta) \ell_k(\zeta).$$
(6.3)

The deformation gradient tensor is determined from  $\Phi_N$  using the DG operators (3.12) and (3.13). From the deformation gradient tensor, the FTLE is determined with (2.31).

### **Multiple Flow Maps from a Single Particle Trace**

We summarize the algorithm for computing the FTLE field at multiple initial times from a single particle trace in three dimensions. Additional details are given for the construction of deformed subdomains from flow maps in three dimensions. For a complete description of the algorithm in two dimensions, we refer to [75].

We denote the flow map approximation from time  $t_0$  to  $t_1 = t_0 + T_1$  by

$$\phi(\mathbf{x}_0, t_0; T_1) \approx \Phi_N. \tag{6.4}$$

Furthermore, we denote another flow map approximation from  $t_1$  to  $t_2 = t_1 + T_2$  by

$$\phi(\mathbf{x}_1, t_1; T_2) \approx \Psi_N. \tag{6.5}$$

 $T_2$  can be positive for a forward-time flow map or negative for a backward-time flow map.

We apply DG differential operators to the flow map,  $\Psi_N$ , by composing an orthogonal basis on  $\Phi_N$ . Thus, we determine the FTLE field over the interval  $T_2$  without the need to re-initialize fluid tracers at time  $t_1$  and eliminate redundant particle integrations.

The faces of the new subdomains based on  $\Phi_N$  are constructed from the locations

of the particles initialized at the subdomain faces of the original grid. Since the fluid particles are initialized on the Lobatto nodes, The faces of the new subdomains are ensured to be conforming. With the new faces, the transfinite mapping for the new subdomains is constructed, which in turn is used to compute the metric terms.

The particle locations,  $\Psi_{lmn}$ , integrated from the initial grid at  $t_0$ , in general are not associated with quadrature nodes on the deformed subdomain given by  $\Phi_N$ . We must determine the quadrature node locations,  $\Psi'_{ijk}$ , by interpolation.

The orthogonal basis on  $\Phi_N$  is established by, first, projecting the fluid particle locations given by the nodes,  $\Phi_{lmn}$  onto the reference element using the inverse of the transfinite map given by (3.2). Since (3.2) does not have an explicit inverse, we find the mapped locations with a Newton method. We denote the reference element coordinates projected from  $\Phi_N$  with  $(\alpha, \beta, \gamma)$ , to distinguish from the reference element coordinates associated with the original grid,  $(\xi, \eta, \zeta)$ .

With the locations,  $(\alpha_{lmn}, \beta_{lmn}, \gamma_{lmn})$ , we build an interpolation operator that interpolates the known values,  $\Psi_{lmn}$ , to the unknown values  $\Psi'_{iik}$ ,

$$\Psi'_{ijk} = \sum_{l=0}^{N} \sum_{m=0}^{N} \sum_{n=0}^{N} \Psi_{lmn} \ell_l(\alpha'_{ijk}) \ell_m(\beta'_{ijk}) \ell_n(\gamma'_{ijk}).$$
(6.6)

Note that the coordinates  $(\alpha'_{ijk}, \beta'_{ijk}, \gamma'_{ijk})$  coincide with the quadrature nodes in the reference element. Hence, (6.6) is equivalent to

$$\bar{\Psi}_p' = I_{pq}^{-1} \bar{\Psi}_q, \tag{6.7}$$

where

$$I_{pq} = \ell_i(\alpha_{lmn})\ell_j(\beta_{lmn})\ell_k(\gamma_{lmn}), \qquad (6.8)$$

and  $p = i(N+1)^2 + j(N+1) + k$  and  $q = l(N+1)^2 + m(N+1) + n$ .

The operator  $I_{pq}^{-1}$  is a linear operator that can operate on any  $\Psi_{lmn}$ , regardless of the time interval,  $T_2$ .

### **Interpolation Conditioning**

As reported in [75], spectral convergence is achieved for interpolated FTLE fields in elements with low levels of deformation. For elements with high levels of deformation, errors in the FTLE field grow for two reasons.

First, as the time interval increases, the interpolation operator *I* becomes increasingly ill-conditioned as the node locations in the reference element move away from the Chebyshev nodes. From polynomial approximation theory, the Lebesgue constant, given by

$$\Lambda_N = \max \sum_{i=0}^N |\ell_i(x)|, \tag{6.9}$$

is known to grow exponentially with increasing polynomial order for uniformly distributed nodes [106]. This growth rate can be faster than exponential for irregular distributions. Hence, the error induced by the poor conditioning of the interpolation operator I can be larger for elements with large N, as compared with elements with smaller N.

Second, the Newton-Raphson method with which the particles are projected to the reference element is not robust for high-order polynomials. As the order of a polynomial increases, the size of the domain of convergence for each root decreases [109]. Thus, the method grows more sensitive to the initial guess with increasing polynomial order. Determining a suitable initial condition for the Newton-Raphson method in large elements with high polynomial orders is made difficult by the highly non-linear motion of the fluid particles.

Numerical experiments in previous work [75] have shown that *h*-refinement of the grid on which the FTLE is computed can reduce the condition number of the interpolation

operator to acceptable levels. The algorithm can be further improved by implementing root finding methods that are less sensitive to the initial conditions as compared with the Newton-Raphson method [110]. Alternatively, the particles may be reinitialized periodically at the Lobatto nodes to minimize the growth of error and the sequence of integrations can be recomposed using interpolation [84]

### **Parallel Implementation**

Integration of the DG-FTLE algorithm with a three-dimensional DG-based fluid solver requires the DG-FTLE algorithm to be parallelized. Several components of the algorithm require special treatment to be implemented in parallel, including the storage of subdomain metric terms and node data, fluid particle data storage, particle integration, and FTLE determination. To handle the transfer of data between partitions on a multiprocessor platform, the Message Passing Interface (MPI) library is used.

Each DG-FTLE subdomain corresponds to a DG element in the fluid solver. Hence, the partitioning of the DG-FTLE mesh is the same as that for the fluid solver so that the metric terms and node data for each subdomain are assigned to the same partition as the corresponding fluid solver element.

As particles are integrated through the domain, some particles may pass from a subdomain located on one partition to a subdomain located on another partition. To determine if a particle should be passed to another processor, each subdomain adjacent to the previous subdomain is searched by mapping the particle location to the corresponding reference element. If the particle is found within a new subdomain that is on another partition, the particle information is passed to that partition. For straight-sided subdomains, only the corner locations are needed to map to the reference element. Therefore, the corner coordinates for subdomains in adjacent partitions that share a face with the previous partition are stored on the previous partition. For curved subdomains, the face data also must be stored.

Computation of the FTLE field requires constructing a flow map from the particle's current location and differentiating the flow map with respect to the particle's initial location. However, since the DG-FTLE metric terms and derivative matrices may be stored on a different processor than the particle is currently located, to compute the FTLE, the particle current location is sent back to the initial partition.

### **Exponential Filter**

The steep gradients in the flow map may lead to unphysical oscillations in the flow map approximation. These oscillations can be reduced with the application of a low-pass exponential filter. Following Vandeven [111] the exponential filter function,  $\sigma(k/N)$ , is  $C^{\infty}[-1,1]$  and satisfies

$$\sigma(0) = 1, \qquad \sigma(\pm 1) = 0,$$
  
 $\sigma^{(j)}(0) = 0, \qquad \sigma^{(j)}(\pm 1) = 0, \qquad j \le \gamma,$ 
(6.10)

where  $\gamma$  is the filter order and  $\sigma^{(j)}$  is the *j*-th derivative of  $\sigma$ . The filter function is given by

$$\sigma(k/N) = \exp\left(-\alpha(k/N)^{\gamma}\right), \qquad 0 \le |k| \le N, \tag{6.11}$$

with  $\alpha = -\ln \varepsilon$ ,  $\varepsilon$  equal to the machine zero, and  $\gamma > 1$  is the filter order.

As discussed by Don [112], the filter is applied to the modal expansion of the function of interest,  $f_N(x)$ 

$$f_N(x) = \sum_{k=0}^{N} a_k T_k(x),$$
(6.12)

where  $a_k$  are the Chebyshev coefficients and  $T_k(x)$  are the Chebyshev polynomials. The

filtered function,  $\tilde{f}_N(x)$  is then

$$\tilde{f}_N(x) = \sum_{k=0}^N \sigma(k/N) a_k T_k(x).$$
(6.13)

In practice, the filtering operation can be performed as a matrix-vector multiplication, where a filtering matrix, **S**, is constructed by inserting the definition of  $a_k$  into (6.13) and rearranging the sum so that we have

$$\tilde{f}_N(x_i) = \sum_{j=0}^N S_{ij} f_j.$$
(6.14)

In matrix-vector notation, we have

$$\tilde{\mathbf{f}}_N = \mathbf{S}\mathbf{f}_N. \tag{6.15}$$

The filter provides a flexible method for reducing the magnitude of high wave number modes occuring near regions with steep gradients without affecting the spectral accuracy of the lower wave number modes. The modes affected by the filter are determined by the strength of the filter, which is easily modified by the parameter  $\gamma$ . Lower values of  $\gamma$  result in stronger filters. The filter function using various values for  $\gamma$  is depicted in Figure 6.2. In this work, the filter is applied directly to the flow map.

# 6.3 Two and Three-Dimensional Lagrangian Visualizations With DG-FTLE

The DG-FTLE algorithm is thoroughly assessed on three test cases. The first is the two-dimensional viscous flow over a NACA 65-(1)412 airfoil. Second, a sensitivity study of the exponential filter with respect to h and p resolution is performed with a



Figure 6.2: Filter function,  $\sigma$  at various filter orders.

steady, three-dimensional ABC flow. Lastly, the DG-FTLE algorithm is applied to the three-dimensional flow over a NACA 65-(1)412 airfoil.

# 6.3.1 Lagrangian Coherent Structures Analysis of a Two-Dimensional Airfoil Flow

To test the algorithm on a complex, curved geometry in two-dimensions, we compute the FTLE field of the wake flow of a NACA 65-(1)412 airfoil at a Reynolds number of Re = 20,000 based on the free-stream velocity and the airfoil chord length. The Karman vortex street in the wake of the airfoil is visualized with the FTLE field. The exponential filter is shown to smooth the flow map and remove Gibbs oscillations in the FTLE field when computed over long time intervals. Multiple FTLE fields, both backward and forward in time, are computed with a single particle trace.

### Direct Numerical Simulation of a NACA 65-(1)412 Airfoil

The compressible two-dimensional Navier-Stokes equations are solved with a Mach number of M = 0.3, so that the fluid is nearly incompressible. A schematic of



Figure 6.3: Boundary conditions and geometry of the two-dimensional DNS of the flow over an airfoil at Re = 20,000.

initial conditions and boundary conditions is depicted in Figure 6.3. The solid boundary of the airfoil is approximated with high-order curved boundary-fitted elements so that the boundary approximation is consistent in order with the fluid solution. The simulation was run until quasi-steady state was reached. Additional details for the flow solution can be found in [88].

The flow over the suction surface of the airfoil separates near x/c = 0.5 and the wake consists of an asymmetric Karman vortex street. Vortices shed from the pressure surface of the airfoil are stronger than vortices shed from the suction surface, as shown in Figure 6.4 by the lower density in the pressure-side vortex cores as compared to the suction-side vortices.

#### Wake Analysis With DG-FTLE

Vortex structures are identified by comparing the *z*-vorticity with the FTLE field. Strong vorticity regions enclosed by curved FTLE ridges delineate a vortex. Such structures are highlighted in Figure 6.5. We note the greater detail in the FTLE in the vortices that are shed from the pressure surface, as compared with the suction surface



**Figure 6.4**: Two-dimensional DNS of the flow over an airfoil at Re = 20,000.; contours of density.



**Figure 6.5**: Attracting (blue) and repelling (red) LCS overlayed on vorticity. Pressure-side vortices are highlighted with solid boxes while suction-side vortices are highlighted with dashed-boxes.

vortices. As vortex strength increases, the fluid rotation rate in the vortex core increases. The faster rotation rate will induce more tangling of the stable and unstable manifolds (LCS) in a strong vortex for a given time interval as compared with a weak vortex. In terms of Eulerian quantities, higher rotation rates in vortex cores are associated with higher vorticity magnitude as compared with lower rotation rates. Thus, within vortex cores, more detail in the FTLE field should correspond with larger vorticity magnitudes, which is consistent with Figure 6.5.

As the flow develops in time, ridges in the forward-time FTLE field partition the flow field into regions where fluid is entrained in the wake vortices (Fig. 6.6). The suction-side vortices draw in fluid from above the shear layer, whereas the pressure-side vortices draw in fluid from the near the lower surface of the airfoil. This is consistent with the results given by Cardwell and Mohseni [71] and Lipinski *et al.* [72], who show that for two-dimensional airfoil flows with laminar separation bubbles, the fluid in the upper surface vortices is entrained from above the separated shear layer.

Ridges in the backward-time FTLE identify the locations in the fluid with the strongest stretching. Hence, backward-time FTLE ridges provide few details on the entrainment of fluid in vortex structures (Figure 6.6). Note the characteristic spiral structure in the core of the pressure-side vortex cores, which is difficult to distinguish in the suction-side vortex cores for the time interval used. As the time interval is increased, more details in the FTLE field emerge, due to the increased stretching and folding of the flow map. This suggests that the amount of detail given by the FTLE field for a particular flow feature is connected to the characteristic time scale of that feature.

### **Effect of Exponential Filter**

Steep gradients form in the flow map in regions with locally strong fluid strain and deformation, such as in vortices and shear layers. Gradients in the flow map grow as the time interval, *T*, over which the flow map is computed increases. In regions of particularly strong fluid deformation, the flow map becomes nearly discontinuous, as depicted in Figure 6.8. The steep flow map gradients induce Gibbs oscillations in the FTLE field, particularly in elements with near discontinuities. To remove these oscillations, an exponential filter is applied to the flow map with an exponential factor  $\gamma = 4$ . The filter reduces steep gradients and discontinuities, as depicted in Figure 6.9, while preserving the features away from the discontinuities. Hence, the filtered FTLE field preserves all of the fine scale structures present in the unfiltered FTLE field, while the Gibbs oscillations are removed.



**Figure 6.6**: Development of the forward-FTLE field over time. The highlighted regions delineate fluid that is eventually entrained in the vortex cores. The dashed lines represent suction-side vortices and the solid lines represent pressure-side vortices.



**Figure 6.7**: Development of the backward-FTLE field over time. The highlighted regions are the same as those in Figure 6.6.



Figure 6.8: Location of the line from which data is extracted (a). As the time interval T is increased, gradients in the flow map increase, leading to near discontinuities (b).



Figure 6.9: The exponential filter is applied to the flow map, where it smooths steep gradients (a). Steep gradients lead to Gibbs oscillations in the unfiltered FTLE (b)-top. Application of an exponential filter with  $\gamma = 4$  removes the unphysical oscillations (b)-bottom.



**Figure 6.10**: Backward-time FTLE field computed from the forward-time flow map at T = 0.04 (a), and the backward-time FTLE field determined with direct integration (b).

### Multiple Flow Maps with a Single Particle Trace

The backward FTLE is computed using two different methods. In the first, the backward-time flow map is constructed from the forward-time flow map by interpolating the fluid tracers onto an orthogonal basis constructed at time T = 0.04. The FTLE field is then computed on this basis (Fig. 6.10a). The second approach is to integrate the fluid particles backward in time from an regular grid and compute the FTLE field in the same manner as for the forward-time FTLE field (Fig. 6.10b). The polynomial order is taken to be N = 24 for both cases.

The strong vortex shedding from the trailing edge deforms the subdomains initialized in the vortex to the extent that the projection of particle locations to the reference element does not converge, thus those elements are colored completely red in Figure 6.10a. The non-optimal conditioning in subdomains with moderate deformation concentrates numerical errors near the edges of the subdomains [106]. This effect is seen as non-physical ridges in the FTLE field near the subdomain boundaries.

The conditioning in the interpolation operator is improved by reducing the polynomial order and *h*-refining the mesh. The mesh is *h*-refined and the polynomial order is decreased so that the same number of degrees of freedom are used in the unrefined and the refined mesh. The lower polynomial order reduces the Lebesgue constant for the interpolation operator, thus reducing the conditioning. Furthermore, by reducing the polynomial order, the domains of convergence for the Newton solver used to project the particle locations to the master element are increased, thus making the Newton solver more robust.

We refine the mesh by two levels, so that there are 16 refined elements per single coarse element. Thus, the polynomial order of each element is reduced by a factor of four to N = 6. The refined mesh is compared to the original mesh in Figure 6.11b and 6.11a, respectively. The refined mesh produces a smooth FTLE field that compares well with the direct method (Fig. 6.11d). A small number of highly deformed subdomains initialized in the core of the trailing edge vortex lead to localized large errors in the FTLE field. However, the local nature of these errors suggests that further refinement of those particular subdomains will minimize the interpolation conditioning.

We compute three FTLE fields, with  $t_0 = 0$ , 0.02 and 0.04, and a time interval of T = 0.25 (Fig. 6.12), on the refined mesh. A shorter time interval than T = 0.37, as is used in the preceding sections, is chosen so that errors due to the conditioning are prevalent, as compared with Gibbs oscillations. The subdomains with significant errors due to poor conditioning are restricted to those that were initialized in the trailing edge vortex, which is consistent with the backward FTLE results (Fig. 6.11d).



**Figure 6.11**: Original coarse mesh (a), and the refined mesh constructed by splitting each element twice (b). Backward-time FTLE field computed from the forward-time flow map at T = 0.04 with the coarse mesh (c), and the backward-time FTLE field determined with the refined mesh (d).

### 6.3.2 Three-Dimensional ABC Flow

To demonstrate the determination of the FTLE field in three-dimensions using DG differential operators, we apply our algorithm to the steady three-dimensional ABC flow. The fluid tracers are integrated using the exact velocity given in (6.16), with no velocity interpolation, such that the determination of the flow with DG operators can be assessed separately from the other components of the algorithm.

A sensitivity study of the exponential filter with respect to *h* and *p*-refinement and filter strength is performed on the ABC flow. The ABC flow is defined by an analytical velocity field and provides a suitable case for studying the effect of the filter applied to the flow map, since the only components of the DG-FTLE algorithm used are the filter and the DG differential operators. In this way, the effects of the filter are isolated.

The performance of the filter is evaluated on several grid configurations. Two meshes were used: one with a single element and another with  $10 \times 10 \times 10$  elements. The effect of the filter is compared for large elements with high polynomial order on the single element mesh and small elements with lower polynomial orders on the multi-



(a)



(b)



**Figure 6.12**: Forward-time flow maps for T = 0.25 starting at  $t_0 = 0$  (a),  $t_0 = 0.02$  (b), and  $t_0 = 0.04$  (c). Each frame is computed from the same particle trace that was initialized at  $t_0 = 0$ .

The three dimensional ABC flow is given by the velocity field

$$\begin{cases} u(\mathbf{x}) = A\sin z + C\cos y, \\ v(\mathbf{x}) = B\sin x + A\cos z, \\ w(\mathbf{x}) = C\sin y + B\cos x, \end{cases}$$

with  $A = \sqrt{3}$ ,  $B = \sqrt{2}$  and C = 1. The ABC flow is a benchmark for three-dimensional FTLE algorithms and has been covered extensively in the literature [58, 84, 113, 114]. The flow field is spatially periodic over the interval  $[0, 2\pi]$  in each direction and forms a criss-crossing network of invariant KAM-type vortex tubes. The regions containing the vortex tubes are partitioned by repelling LCS and align with strong ridges in the FTLE field.

As the time interval increases, the flow map becomes nearly discontinuous in places, as seen in Figure 6.13a. Application of the exponential filter with  $\gamma = 4$  smooths these near-discontinuities (Fig. 6.13b). The steep gradients in the flow map lead to severe Gibbs-phenomena and poor quality visualization of the FTLE field as compared with the filtered flow map (Fig. 6.14).

Different filter strengths are compared to assess the effect of the filter on the flow map and the FTLE field. Using the single element mesh, the polynomial order is taken to be N = 240, and the filter is applied using  $\gamma = 4$ , 8, and 16. The FTLE field for each of these is compared to the unfiltered FTLE field (Fig. 6.14). As the filter strength is increased, the Gibbs oscillations are removed, beginning with the oscillations in the smooth regions using the weakest filter ( $\gamma = 16$ ) and progressing to the ridges for the strongest filter ( $\gamma = 4$ ). Taking data from a line in the *z*-direction at x = y = 0.5, we



**Figure 6.13**: Flow map  $\Phi^x$ , of the steady ABC flow (a) unfiltered, and (b) filtered with filter power  $\gamma = 4$ .

compare the filters directly (6.15). Although all filters resolve the large structures in the flow map (Fig. 6.15a), the weaker filters still retain steep gradients, leading to FTLE values which more closely match the unfiltered case (Fig. 6.15b).

As N is increased on each mesh, more details in the flow map are resolved given the same filter power, while the quality of the FTLE field is preserved (Fig. 6.16). The multi-element mesh leads to more resolved filtered flow maps as compared to the singleelement mesh (Fig. 6.17). As a result of the finer detail retained in the flow map for the multi-element mesh as compared to the single element mesh, more details are resolved in the FTLE as well (Fig. 6.17).

The general trends with the operation of the filter suggest that when the filter is applied to very large polynomial orders, some of the fine details are lost as compared to the results when smaller elements and smaller polynomial orders are used.

In three-dimensions, ridges in the FTLE field identify two-dimensional surfaces embedded in the three-dimensional flow domain. However, it is often challenging to extract these features from the FTLE field, since the ridge height may vary over the ridge [59]. In the case of the ABC flow, a very good approximation to the location of the ridge





**Figure 6.14**: FTLE field of the steady ABC flow using different filter powers. (1) Unfiltered, (b)  $\gamma = 16$ , (c)  $\gamma = 8$ , and (d)  $\gamma = 4$ . The single element mesh is used with N = 240.



Figure 6.15: The effect of different filter strengths on the (a) flow map and the (b) FTLE field.



Figure 6.16: Effect of the filter on different polynomial orders for the  $10 \times 10 \times 10$  mesh (a) and the single element mesh (b) with filter power  $\gamma = 4$ .



**Figure 6.17**: Effect of the filter on the flow map for the  $10 \times 10 \times 10$  mesh (a) and the single element mesh (b) with filter power  $\gamma = 4$ .



Figure 6.18: FTLE field of the steady ABC flow for the  $10 \times 10 \times 10$  mesh (a) and the single element mesh (b) with different polynomial orders.



Figure 6.19: Iso-surfaces of the FTLE field for the steady ABC flow.

can be obtained with iso-contours of the FTLE field, due to the relative uniformity of the ridge heights (Fig. 6.19).

### 6.3.3 Three-Dimensional Airfoil Flow

The DG-FTLE algorithm is assessed on the three-dimensional transitional flow over a NACA 65-(1)412 airfoil. Three-dimensional streamwise and spanwise vortices in the wake are identified. Multiple FTLE fields, both backward and forward in time, are computed with a single particle trace.

### Three-Dimensional DNS of a NACA 65-(1)412 Airfoil

Transitional bluff body flows are characterized by a breakdown of the twodimensional spanwise vortex structures into a three-dimensional network of spanwise and streamwise vortices. This class of flows has been studied extensively in the literature [115, 116, 117, 118, 119]. The development of the streamwise vortices is a result of instabilities in the spanwise vortex cores and the braid region connecting the spanwise vortices, termed elliptic (or mode A) and hyperbolic (or mode B) instabilities, respectively.
Both instability mechanisms lead to streamwise vortex structures aligned along the braid region connecting the spanwise vortices, differentiated by the spanwise separation of the secondary vortices [118].

Although vortex cores are characterized by strong vorticity, vorticity alone cannot unambiguously identify a vortex. Several alternative Eulerian criteria have been developed to identify vortex structures. The work of Green *et al.* [61] and Haller [113] provide comparisons of various Eulerian vortex identification methods with the FTLE field. Since Eulerian criteria, such as Q,  $\lambda_2$ ,  $\Delta$ , are computed from Eulerian quantities such as rate of strain and vorticity, they are easily computed from instantaneous data sets as compared with Lagrangian methods such as the FTLE field. However, it is shown that the Eulerian criteria do not adequately identify vortices in rotating frames since they are not objective [113]. Furthermore, Green *et al.* [61] points out that the finer resolution of the FTLE field provides more information on vortex internal structure than Eulerian criteria, which are restricted to the resolution of the velocity data.

Following Green *et al.* [61], we choose structures that can be identified *a priori* to illustrate the application of the DG-FTLE algorithm on a complex flow of practical interest. The airfoil flow is simulated at a transitional Reynolds number and the streamwise vortices are identified (Fig. 6.20). In this work, we compare the vorticity magnitude to the FTLE field. Streamwise vortex cores identified by the vorticity magnitude are correlated with LCS visualized by the FTLE field.

The three-dimensional mesh for the fluid solution is generated by extruding the domain in the z-direction to a depth of  $L_z = 0.5c$ . The boundary conditions in the z-direction are periodic. To reduce the computation time, the two-dimensional solution is used as the initial condition for the three-dimensional solution. The polynomial-order is N = 8. The Reynolds number, Mach number, Prandtl number and all other flow variables and boundary conditions are the same for the three-dimensional case as compared with



Figure 6.20: Iso-contours of vorticity magnitude, colored by u-velocity of the three-dimensional flow over an airfoil at Re = 20,000.

the two-dimensional case, with the exception of the periodic boundary conditions in the *z*-direction of the three-dimensional case. Additional details of the three-dimensional solution will be presented elsewhere.

#### Identification of Wake Structures With DG-FTLE

The forward-time FTLE field is determined over a time interval of T = 0.14, with a polynomial of degree N = 36 and the flow map is filtered with  $\gamma = 4$ .

Vortex core regions in the wake identified with vorticity magnitude coincide with vortex structures identified by ridges in the FTLE field (Fig. 6.21). The FTLE field shows a greater level of detail than does the vorticity, highlighting the internal structure of the streamwise vortices and the complex fluid folding induced by the interaction between the streamwise vortices and the spanwise vortices.

Slices of the FTLE field showing the wake structures in profile (Fig. 6.22), and at several *x*-locations in the wake (Fig. 6.23) illustrate the fine detail in the complex vortex wake structures. Evidence of the counter-rotating configuration of the streamwise vortices can be seen in Figure 6.23f.



Figure 6.21: Isocontours of vorticity magnitude (a) and FTLE (b).



**Figure 6.22**: Slice of the magnitude of vorticity (a) and the FTLE field at the same location (b). The FTLE field is computed over a time interval of T = 0.14, with a polynomial order of N = 36 and a filter with  $\gamma = 4$ . Vortex structures are highlighted.





(c)

(a)

(d)

(b)



(e)





**Figure 6.23**: Vorticity magnitude at x = 0.98 (a), 1.03 (c), 1.18 (e), 1.33 (g), and FTLE field at the same intervals (b), (d), (f), (h). The gray regions at the bottom of (a) and (b) denote the surface of the airfoil. Vortex structures are highlighted.



**Figure 6.24**: Slice of the backward-time FTLE interpolated from the forward-time flow map (a) and the backward-time FTLE field computed with the direct method (b).

#### **Multiple FTLE Fields From a Single Particle Integration**

Computation of the backward-time FTLE field from the forward-time flow map is made difficult by the large subdomain sizes relative to the characteristic lengths of the flow structures within each subdomain. The small scale structures resolved by the FTLE field require high polynomial orders to adequately capture them. However, the conditioning of the interpolation operator rapidly deteriorates as the polynomial order is increased. Furthermore, the high deformation in the subdomains lead to poor initial estimates for the Newton solver to project the flow map locations to the reference element. Therefore, the FTLE determined in such a way performs poorly as compared with computing the FTLE field directly by integrating the fluid tracers backward in time (Fig. 6.24). The FTLE fields in Figure 6.24 are computed over the time interval T = 0.02 with a polynomial of degree N = 24. The exponential filter is not applied in either case.

By refining the mesh, we improve the interpolation in three-dimensions, just as was done in two dimensions. The three-dimensional FTLE field is computed on a mesh that has four times the resolution as compared with the original mesh. The polynomial order is also decreased by a factor of four to N = 6, such that the same number of degrees of freedom are used and to minimize  $\Lambda_N$ . In Figure 6.25, it is seen that the FTLE field quality is greatly improved.



**Figure 6.25**: Slice of the backward-time FTLE interpolated from the forward-time flow map (a) and the backward-time FTLE field computed with the direct method (b) for the refined mesh.

As with the two-dimensional airfoil case, the algorithm is used to compute FTLE fields at multiple initial times from a particle integration initialized at a single time. The FTLE fields are computed over a time interval of 0.02 for initial times  $t_0 = 0, 0.01$ , and 0.02. As seen in Figure 6.26, the results are consistent with what was seen in the two-dimensional case.

## 6.4 Summary of Results

Extension of the DG-FTLE algorithm developed in [75] to three dimensions and its implementation on complex geometries with curved boundaries is presented.

Passive fluid tracers are initialized on DG nodal mesh points within each subdomain so that a high-order approximation of the flow map is constructed from the tracer final locations. By computing the Cauchy-Green strain tensor with DG differential operators, the FTLE field is high-order accurate.

Steep gradients that form in the flow map over long time integrations are smoothed with an exponential filter. Parallel implementation of the particle tracing algorithm and FTLE determination is presented.

The algorithm is thoroughly assessed on three benchmarks: the two-dimensional



(a)



(b)



**Figure 6.26**: Forward-time flow maps for T = 0.02 starting at  $t_0 = 0$  (a),  $t_0 = 0.01$  (b), and  $t_0 = 0.02$  (c). Each of these were computed from the same particle trace that was initialized at  $t_0 = 0$ .

viscous flow over an airfoil at Re = 20,000, a steady three-dimensional ABC flow, and the three-dimensional transitional flow over an airfoil.

Computation of the two-dimensional FTLE field on a mesh with curved boundaries is demonstrated on the two-dimensional airfoil flow. The FTLE field is determined at multiple initial times, both backward and forward in time. The FTLE field is used to identify vortex structures in the wake.

Implementation of the exponential filter and computation of FTLE fields with DG operators is illustrated with the steady ABC flow for several *h* and *p* resolutions. Smaller elements with lower polynomial orders are shown to produce more accurate results than very large elements with higher polynomial orders, given the same number of degrees of freedom.

Extension of the algorithm in three dimensions is tested on the three-dimensional airfoil flow. Streamwise and spanwise vortex structures are shown in greater detail with the FTLE field than with vorticity.

Application of an exponential filter to the flow map is shown to remove Gibbs oscillations in the FTLE field when computed over long time intervals. Poor conditioning in the interpolation on deformed grids has a negative impact on the quality of FTLE fields constructed from these deformed subdomains, both backward and forward in time. Interpolation conditioning and FTLE quality are shown to significantly improve with *h*-refinement of the FTLE grid.

Future efforts will focus on developing an adaptive mesh algorithm for improving the conditioning of I and extending the time interval over which the algorithm can be applied, as well as improvements in efficiency. Additionally, we aim to examine the application of this algorithm to the work of Farazmand and Haller [69].

Chapter 6, in part, has been submitted for publication of the material as it may appear in Computers and Fluids, 2016, Nelson, Daniel; Jacobs, Gustaaf, "High-Order Visualization of Three-Dimensional Lagrangian Coherent Structures with DG-FTLE". The dissertation author was the primary investigator and author of this paper.

## Chapter 7

## Conclusions

High-order accurate numerical tools based on DG spectral element methods are developed for the Lagrangian analysis of separated, turbulent flows over complex geometries. The compressible Navier-Stokes equations are solved in two and three dimensions with a DG spectral element method which is coupled with a high-order accurate algorithm for identifying and analyzing LCS in the flow. The numerical framework is intended to be applied to the development on AFC strategies for flow separation over airfoils and vanes.

# 7.1 Discontinuous-Galerkin Methods on Complex Geometries

The unsteady, separated flow over a NACA 65-(1)412 airfoil is simulated with a high-order DG spectral element method. The solution is computed at a Reynolds number of 20,000 based on the free stream velocity and airfoil chord length, and a Mach number of 0.3. A procedure for fitting high-order DG elements with curved-edges to boundaries defined by splines is presented. The polynomial order of the boundary approximation is

matched to the order of the fluid solver by fitting a polynomial to the Chebyshev-Lobatto quadrature nodes of each element edge along the boundary.

Solutions using meshes with curved boundary elements are compared with solutions which use straight-sided boundary element meshes. Curved-sided meshes produce regular, Karman vortex street wakes with a single, peak shedding frequency. The piecewise-linear boundary approximation of the straight-sided meshes introduce artificial surface roughness that leads to the simulation of incorrect physics. The surface variations induce spurious time dependent modes in the wake that lead to large-scale instabilities in the vortex shedding behavior. When compared with the curved-sided solutions, the straight-sided mesh solutions under-predict the aerodynamic performance of the airfoil.

The error in the DG boundary approximation as compared to the exact spline boundary definition is analyzed. It is shown that spectral accuracy in the boundary approximation is not achieved for non-analytic boundary definitions. Regions with high curvature produce large errors in the boundary approximation and lead to poor convergence of the solution. Regions with low curvature lead to much better convergence of the solution.

#### 7.2 Lagrangian Coherent Structures

A method for identifying LCS using FTLE fields is developed that is consistent with high-order DG methods in both accuracy and implementation. The method is designed to be efficient and integrated directly into DG fluid solvers so that FTLE fields are determined simultaneously with the fluid solution. Passive fluid particles are initialized at the quadrature nodes in the DG elements so that the flow map is approximated with a spectral polynomial interpolant. The flow map gradient is computed using DG differential operators.

In two dimensions, the algorithm is tested on three benchmarks: an analytical spatially periodic gyre flow, a vortex advected by a uniform inviscid flow, and the viscous flow around a square cylinder. In these cases, the algorithm is shown to have spectral convergence. In three-dimensions, the flow is tested for an ABC flow.

As the time interval over which the FTLE field is computed increases, steep gradients in the flow map form, inducing Gibbs oscillations in the FTLE. An exponential filter is applied to the flow map to remove these oscillations.

An algorithm for computing multiple flow maps from a single particle trace is presented. New initial times are established on flow maps at later times, as opposed to initializing a new particle trace, by projecting the particle locations in the flow maps isoparametrically to the reference element. A new, deformed initial element is then constructed and the nodes at the projected locations are interpolated to the quadrature nodes. The new flow maps can be determined both backward in time and forward in time.

For highly deformed elements, the interpolation is poorly conditioned. However, the conditioning of the interpolation operator is shown to provide quantitative information regarding the deformation of the fluid in each element that complements the FTLE field.

The two-dimensional and three-dimensional LCS field are analyzed for the unsteady flow over a NACA 65-(1)412 airfoil. Strongly attracting and repelling stationary LCS are visualized near the trailing edge of the airfoil. These transport barriers are not advected into the wake, but remained fixed near the trailing edge. In two-dimensions, alternating vortices are shed from the upper surface at the location of the stationary LCS and from the lower surface at the trailing edge, forming a von Karman vortex street in the wake. In three-dimensions, the vortex street breaks down at the location of the stationary LCS and an elliptic instability induces the formation of streamwise vortical structures connecting the spanwise vortices that are shed from the airfoil.

### 7.3 Future Work

Future work should focus on implementing adaptive mesh refinement on the FTLE algorithm to improve efficiency and accuracy. The conditioning of the interpolation operator on deformed subdomains can be used as criteria for mesh refinement. Efforts should also be directed towards extending the algorithm to extract LCS according to the recent geodesic theory. Although LCS and Lagrangian separation manifolds are widely assumed to be equivalent, a rigorous validation of this relationship is needed so that the methods developed in this dissertation can be applied to Lagrangian flow separation.

Numerical studies of airfoils at Re higher than 20,000 (Re = 50,000-60,000), show that the separated shear layer on the suction surface breaks down upstream of the trailing edge to a KH-type instability. This implies that at some Re > 20,000, there exists a critical point in which a bifurcation occurs, thus allowing the growth of the KH instability in the shear layer and dramatically altering the overall characteristics of the flow. A study to determine the nature and transition point of this bifurcation would enhance the understanding of very low Re airfoils flows.

# Appendix A

# FTLE Field for an Analytical Inviscid Vortex

The flow map for the uniformly advected vortex case with velocity field given by (5.32) can be computed if we assume a change of coordinates such that

$$x' = x - X_c - U_{\infty}(t - t_0),$$
  
 $y' = y - Y_c,$  (A.1)

$$u' = u - U_{\infty}.$$

$$v' = v.$$
(A.2)

Then, the velocity field simplifies to

$$u'(x',y') = -\frac{A}{R}y'\exp\left(\frac{-r^2}{2R^2}\right),$$
  

$$v'(x',y') = -\frac{A}{R}x'\exp\left(\frac{-r^2}{2R^2}\right),$$
(A.3)

where  $r = \sqrt{(x')^2 + (y')^2}$ . By transforming to plane polar coordinates, the velocity can be integrated. The velocity becomes

$$u_r = 0,$$
  
$$u_{\theta} = \frac{A}{R} \exp\left(\frac{-r^2}{2R^2}\right) r.$$
 (A.4)

After integrating with respect to  $t - t_0$ , and transforming back to Cartesian coordinates, we have the trajectory equations,

$$x(t;t_{0},\mathbf{x}_{0}) = U_{\infty}(t-t_{0}) + X_{c}$$

$$+ r \cos \left[ \arctan\left(\frac{y_{0} - Y_{c}}{x_{0} - X_{c}}\right) + \frac{A}{R} \exp\left(-\frac{r^{2}}{2R^{2}}\right)(t-t_{0}) \right],$$

$$y(t;t_{0},\mathbf{x}_{0}) = Y_{c} + r \sin\left[ \arctan\left(\frac{y_{0} - Y_{c}}{x_{0} - X_{c}}\right) + \frac{A}{R} \exp\left(-\frac{r^{2}}{2R^{2}}\right)(t-t_{0}) \right].$$
(A.5)
(A.6)

where, in this case, we note that *r* is constant for each fluid particle according to Eqn. (A.4). The components to the deformation gradient,  $\nabla \phi$ , are,

$$\frac{\partial \phi^{x}}{\partial x_{0}} = \cos \left[ \arctan \left( \frac{y_{0} - Y_{c}}{x_{0} - X_{c}} \right) + \frac{A}{R} \exp \left( -\frac{r^{2}}{2R^{2}} \right) t \right] \frac{x_{0} - X_{c}}{r}$$
$$- r \sin \left[ \arctan \left( \frac{y_{0} - Y_{c}}{x_{0} - X_{c}} \right) + \frac{A}{R} \exp \left( -\frac{r^{2}}{2R^{2}} \right) t \right]$$
$$\times \left[ -\frac{y_{0} - Y_{c}}{r^{2}} - \frac{A(x_{0} - X_{c})}{R^{3}} \exp \left( -\frac{r^{2}}{2R^{2}} \right) t \right], \qquad (A.7)$$

$$\frac{\partial \phi^{x}}{\partial y_{0}} = \cos \left[ \arctan \left( \frac{y_{0} - Y_{c}}{x_{0} - X_{c}} \right) + \frac{A}{R} \exp \left( -\frac{r^{2}}{2R^{2}} \right) t \right] \frac{y_{0} - Y_{c}}{r} - r \sin \left[ \arctan \left( \frac{y_{0} - Y_{c}}{x_{0} - X_{c}} \right) + \frac{A}{R} \exp \left( -\frac{r^{2}}{2R^{2}} \right) t \right] \times \left[ \frac{x_{0} - X_{c}}{r^{2}} - \frac{A(y_{0} - Y_{c})}{R^{3}} \exp \left( -\frac{r^{2}}{2R^{2}} \right) t \right],$$
(A.8)

$$\frac{\partial \phi^{y}}{\partial x_{0}} = \sin \left[ \arctan \left( \frac{y_{0} - Y_{c}}{x_{0} - X_{c}} \right) + \frac{A}{R} \exp \left( -\frac{r^{2}}{2R^{2}} \right) t \right] \frac{x_{0} - X_{c}}{r} + r \cos \left[ \arctan \left( \frac{y_{0} - Y_{c}}{x_{0} - X_{c}} \right) + \frac{A}{R} \exp \left( -\frac{r^{2}}{2R^{2}} \right) t \right] \times \left[ -\frac{y_{0} - Y_{c}}{r^{2}} - \frac{A(x_{0} - X_{c})}{R^{3}} \exp \left( -\frac{r^{2}}{2R^{2}} \right) t \right],$$
(A.9)

$$\frac{\partial \phi^{y}}{\partial y_{0}} = \sin \left[ \arctan \left( \frac{y_{0} - Y_{c}}{x_{0} - X_{c}} \right) + \frac{A}{R} \exp \left( -\frac{r^{2}}{2R^{2}} \right) t \right] \frac{y_{0} - Y_{c}}{r} + r \cos \left[ \arctan \left( \frac{y_{0} - Y_{c}}{x_{0} - X_{c}} \right) + \frac{A}{R} \exp \left( -\frac{r^{2}}{2R^{2}} \right) t \right] \times \left[ \frac{x_{0} - X_{c}}{r^{2}} - \frac{A(y_{0} - Y_{c})}{R^{3}} \exp \left( -\frac{r^{2}}{2R^{2}} \right) t \right].$$
(A.10)

Since  $\nabla \phi$  is a 2 × 2 tensor, the maximal eigenvalue is given explicitly by

$$\lambda_{max}(\mathbf{C}) = \frac{1}{2} \left[ \left( \frac{\partial \phi^{x}}{\partial x_{0}}^{2} + \frac{\partial \phi^{y}}{\partial y_{0}}^{2} \right) + \sqrt{4 \left( \frac{\partial \phi^{y}}{\partial x_{0}} \frac{\partial \phi^{x}}{\partial y_{0}} \right) \left( \frac{\partial \phi^{y}}{\partial x_{0}} \frac{\partial \phi^{x}}{\partial y_{0}} \right) + \left( \frac{\partial \phi^{x}}{\partial x_{0}}^{2} - \frac{\partial \phi^{y}}{\partial y_{0}}^{2} \right)^{2}} \right], \quad (A.11)$$

substituting (A.7) - (A.10). The FTLE is then computed using (2.30).

## **Bibliography**

- [1] George Haller and Themistoklis Sapsis. Lagrangian coherent structures and the smallest finite-time Lyapunov exponent. *Chaos*, 21:023115, 2011.
- [2] Richard Courant, Kurt Friedrichs, and Hans Lewy. On the partial difference equations of mathematical physics. *IBM journal of Research and Development*, 11(2):215–234, 1967.
- [3] Martha W Evans, Francis H Harlow, and Eleazer Bromberg. The particle-in-cell method for hydrodynamic calculations. Technical report, DTIC Document, 1957.
- [4] Sergei Konstantinovich Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik*, 89(3):271–306, 1959.
- [5] Steven A Orszag and GS Patterson Jr. Numerical simulation of turbulence. In *Statistical models and Turbulence*, pages 127–147. Springer, 1972.
- [6] Sutanu Sarkar, Gordon Erlebacher, and M Yousuff Hussaini. Direct simulation of compressible turbulence in a shear flow. *Theoretical and Computational Fluid Dynamics*, 2(5-6):291–305, 1991.
- [7] Parviz Moin and John Kim. Numerical investigation of turbulent channel flow. *Journal of fluid mechanics*, 118:341–377, 1982.
- [8] John Kim, Parviz Moin, and Robert Moser. Turbulence statistics in fully developed channel flow at low reynolds number. *Journal of fluid mechanics*, 177:133–166, 1987.
- [9] Robert D Moser and Parviz Moin. The effects of curvature in wall-bounded turbulent flows. *Journal of Fluid Mechanics*, 175:479–510, 1987.
- [10] Hung Le, Parviz Moin, and John Kim. Direct numerical simulation of turbulent flow over a backward-facing step. *Journal of fluid mechanics*, 330:349–374, 1997.

- [11] Hans Wengle, André Huppertz, Günter Bärwolff, and Gerd Janke. The manipulated transitional backward-facing step flow: an experimental and direct numerical simulation investigation. *European Journal of Mechanics-B/Fluids*, 20(1):25–46, 2001.
- [12] Anthony T Patera. A spectral element method for fluid dynamics: laminar flow in a channel expansion. *Journal of computational Physics*, 54(3):468–488, 1984.
- [13] Lambros Kaiktsis, George Em Karniadakis, and Steven A Orszag. Onset of threedimensionality, equilibria, and early transition in flow over a backward-facing step. *Journal of Fluid Mechanics*, 231:501–528, 1991.
- [14] Philip M Gresho, David K Gartling, JR Torczynski, KA Cliffe, KH Winters, TJ Garratt, A Spence, and John W Goodrich. Is the steady viscous incompressible two-dimensional flow over a backward-facing step at re= 800 stable? *International Journal for Numerical Methods in Fluids*, 17(6):501–541, 1993.
- [15] F. Bassi and S. Rebay. High-order accuracte discontinuous finite element solutions of the 2D Euler equations. J. Comp. Phys., 138:251–285, 1997.
- [16] F. Bassi and S. Rebay. High-order accuracte discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. J. Comp. Phys., 131:267–279, 1997.
- [17] Gustaaf B. Jacobs, Farzad Mashayek, and David A. Kopriva. A comparison of outflow boundary conditions for the multidomain staggered-grid spectral method. *Numer. Heat Transfer, B*, 44(3):225–251, 2003.
- [18] Thomas JR Hughes, John A Cottrell, and Yuri Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*, 194(39):4135–4195, 2005.
- [19] Yuzhi Sun, Zhi Jian Wang, and Yen Liu. High-order multidomain spectral difference method for the navier-stokes equations on unstructured hexahedral grids. *Communications in Computational Physics*, 2(2):310–333, 2007.
- [20] Jan S. Hesthaven, Sigal Gottlieb, and David Gottlieb. *Spectral Methods for Time-Dependent Problems*. Cambridge, Cambridge, UK, 2007.
- [21] Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous-Galerkin Methods: Algorithms, Analysis, and Applications.* Springer, New York, 2008.
- [22] Moshe Dubiner. Spectral methods on triangles and other domains. *Journal of Scientific Computing*, 6(4):345–390, 1991.

- [23] SJ Sherwin and G Em Karniadakis. A triangular spectral element method; applications to the incompressible navier-stokes equations. *Computer methods in applied mechanics and engineering*, 123(1):189–229, 1995.
- [24] P-O Persson, J Bonet, and J Peraire. Discontinuous galerkin solution of the navier–stokes equations on deformable domains. *Computer Methods in Applied Mechanics and Engineering*, 198(17):1585–1595, 2009.
- [25] David A. Kopriva. A staggered-grid multidomain spectral method for the compressible Navier-Stokes equations. J. Comp. Phys., 143:125–158, 1998.
- [26] David A. Kopriva, Stephen L. Woodruff, and M. Y. Hussiani. Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method. *Int. J. Numer. Meth. Engng*, 53:105–122, 2002.
- [27] David A Kopriva. Metric identities and the discontinuous spectral element method on curvilinear meshes. *Journal of Scientific Computing*, 26(3):301–327, 2006.
- [28] Hong Lou, Joseph D. Baum, and Rainald L ohner. Fast *p*-multigrid discontinuous Galerkin method for compressible flows at all speeds. *AIAA Journal*, 46(3):635– 652, 2008.
- [29] Lilia Kirvodonova and Marsha Berger. High-order accurate implementation of solid wall boundary conditions in curved geometries. J. Comp. Phys., 211:492–512, 2006.
- [30] Xiangxiong Zhang and Sirui Tan. A simple and accurate discontinuous Galerkin scheme for modeling scalar-wave propagation in media with curved interfaces. *Geophysics*, 80(2):T83–T89, 2015.
- [31] Thomas Toulorge and Wim Desmet. Curved boundary tratments for the discontinuous Galerkin method applied to aeroacoustic propagation. AIAA Journal, 48(2):479–489, 2010.
- [32] Hayder Salman, Jan S. Hesthaven, Tim Warburton, and George Haller. Predicting transport by Lagrangian coherent structures with a high-order method. *Theor. Comput. Fluid Dyn.*, 21:39–58, 2007.
- [33] Gerald E Farin. A history of curves and surfaces in CAGD. In Gerald E Farin, Josef Hoschek, and Myung-Soo Kim, editors, *Handbook of computer aided geometric design*. Elsevier, 2002.
- [34] A. R. Forrest. On Coons and other methods for the representation of curved surfaces. *Comput. Graph. Image Process.*, 1(4):341–359, 1972.

- [35] Florian Hindenlang. Mesh Curving Techniques for High Order Parallel Simulations on Unstructured Meshes. PhD thesis, University of Stuttgart, 2014.
- [36] David Gottlieb and Chi-Wang Shu. On the Gibbs phenomenon IV: recovering exponential accuracy in a subinterval from a Gegenbauer partial sum of a piecewise analytic function. *Math. Comp.*, 64(211):1081–1095, 1995.
- [37] David Gottlieb and Chi-Wang Shu. On the Gibbs phenomenon V: recovering exponential accuracy from collocation point values of a piecewise analytic function. *Numer. Math*, 71:511–526, 1995.
- [38] David Gottlieb and Chi-Wang Shu. On the Gibbs phenomenon III: recovering exponential accuracy in a subinterval from a spectral partial sum of a piecewise analytic function. *J. Numer. Anal.*, 33(1):280–290, 1996.
- [39] David Gottlieb, Chi-Wang Shu, Alex Solomonoff, and Herve Vandeven. On the Gibbs phenomenon I: recovering exponential accuracy from the Fourier partial sum of nonperiodic analytic function. J. Comp. App. Math., 43(1):81–98, 1992.
- [40] Francesc Arandiga, Albert Cohen, Rosa Donat, and Nira Dyn. Interpolation and approximations of piecewise smooth functions. J. Numer. Anal., 43(1):41–57, 2005.
- [41] John P. Boyd. Trouble with Gegenbauer reconstruction for defeating Gibbs' phenomenon: Runge phenomenon in the diagonal limit of Gegenbauer polynomial approximations. *J. Comp. Phys.*, 204(1):253–264, 2005.
- [42] A.V. Boiko, A.V. Dovgal, A.V. Kozlov, and V.A. Shcherbakov. Flow instability in the laminar boundary layer separation zone created by a small roughness element. *Fluid Dyn.*, 25(1):12–17, 1990.
- [43] Elmar Achenbach. Influence of surface roughness on the cross-flow around a circular cylinder. *J. Fluid Mech.*, 46:321–335, 1971.
- [44] Unal M.F. and D. Rockwell. On vortex formation from a cylinder. part 1. the initial instability. *J. Fluid Mech.*, 190:491–512, 1988.
- [45] M.E. Goldstein. Scattering of acoustic waves into Tollmien-Schlichting waves by small streamwise variation in surface geometry. J. Fluid Mech., 154:509–529, 1985.
- [46] Ludwig Prandtl. Über Flüssigkeitsbewegung bei sehr kleiner Reibung. Verhandl. III. Intern. Math. Kongr. Heidelberg, pages 484–491, 1904.
- [47] Nicolas Rott. Unsteady viscous flows in the vicinity of a separation point. Q. Appl.

Maths, (13):444-451, 1956.

- [48] Franklin K Moore. On the separation of the unsteady laminar boundary layer. In *Grenzschichtforschung/Boundary Layer Research*, pages 296–311. Springer, 1958.
- [49] WRj Sears and DP Telionis. Unsteady boundary-layer separation. In *Recent* research on unsteady boundary layers, volume 1, pages 404–447. Laval University Press, 1971.
- [50] Stephen J Cowley, Leon L Van Dommelen, and Shui T Lam. On the use of lagrangian variables in descriptions of unsteady boundary-layer separation. *Philo-sophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 333(1631):343–378, 1990.
- [51] K Shariff, TH Pulliam, and JM Ottino. A dynamical systems analysis of kinematics in the time-periodic wake of a circular cylinder. *Lect. Appl. Math*, 28:613–646, 1991.
- [52] George Haller. Exact theory of unsteady separation for two-dimensional flows. J. *Fluid Mech.*, 512:257–311, 2004.
- [53] Amit Surana, Oliver Grunberg, and George Haller. Exact theory of threedimensional flow separation. Part 1. Steady separation. J. Fluid Mech., 564:57–103, 2006.
- [54] Amit Surana, Oliver Grunberg, George Haller, and Gustaaf Jacobs. An exact theory of three-dimensional fixed separation in unsteady flows. *Phys. Fluids*, 20:107101, 2008.
- [55] Gustaaf B. Jacobs, A. Surana, T. Peacock, and George Haller. Identification of flow separation in three and four dimensions. In 45th AIAA Aerospace Sciences Meeting. American Institute for Aeronautics and Astronautics, January 2007.
- [56] A. Surana, Gustaaf B. Jacobs, and George Haller. Extraction of separation and attachment surfaces from three-dimensional steady shear flows. *AIAA Journal*, 45(6):1290–1302, 2007.
- [57] George Haller and G. Yuan. Lagrangian coherent structures and mixing in twodimensional turbulence. *Physica D*, 147:352–370, 2000.
- [58] George Haller. Determining material surfaces and coherent structures in threedimensional fluid flows. *Physica D*, 149:248–277, 2001.
- [59] Shawn C. Shadden, Francois Lekien, and J. E. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-

dimensional aperiodic flows. Physica D, 212:271-304, 2005.

- [60] Francois Lekien, Shawn C. Shadden, and Jerrold E. Marsden. Lagrangian coherent structures in n-dimansional systems. *J. Math. Phys.*, 48:065404, 2007.
- [61] M. A. Green, Clarence W. Rowley, and George Haller. Detection of Lagrangian coherent structures in three-dimensional turbulence. J. Fluid Mech., 572:111–120, 2007.
- [62] Gustaaf B. Jacobs and Karen Armstrong. Inertial particle dispersion in the Lagrangian wake of a square cylinder. In *47th AIAA Aerospace Sciences Meeting*. AIAA, January 2009.
- [63] Shawn C. Shadden, Matteo Astorino, and Jean-Fredereic Gerbeau. Computational analysis of an aortic valve jet with Lagrangian coherent structures. *Chaos*, 20:017512, 2010.
- [64] Amirhossein Arzani and Shawn C. Shadden. Characterization of the transport topology in patient-specific abdominal aortic aneurysm models. *Phys. Fluids*, 24:081901, 2012.
- [65] Yong Wang, George Haller, Andrzej Banaszuk, and Gilead Tadmor. Closed-loop Lagrangian separation control in a bluff body shear flow model. *Phys. Fluids*, 15(8):2251–2266, 2003.
- [66] Wenbo Tang, Pak Wai Chan, and George Haller. Accurate extraction of Lagrangian coherent structures over finite domains with application to flight data analysis over Hong Kong International Airport. *Chaos*, 20:017502, 2010.
- [67] George Haller. A variational theory of hyperbolic Lagrangian coherent structures. *Physica D*, 240:574–598, 2011.
- [68] George Haller and Francisco J. Beron-Vera. Geodesic theory of transport barriers in two-dimensional flow. *Physica D*, 241:1680–1702, 2012.
- [69] Mohammad Farazmand and George Haller. Computing Lagrangian coherent structures from their variational theory. *Chaos*, 22:013128, 2012.
- [70] Daniel Blazevski and George Haller. Hyperbolic and elliptic transport barriers in three-dimensional unsteady flows. *Physica D: Nonlinear Phenomena*, 273:46–62, 2014.
- [71] Blake M Cardwell and Kamran Mohseni. Vortex shedding over a two-dimensional airfoil: Where the particles come from. *AIAA journal*, 46(3):545–547, 2008.

- [72] Doug Lipinski, Blake Cardwell, and Kamran Mohseni. A lagrangian analysis of a two-dimensional airfoil with vortex shedding. *Journal of Physics A: Mathematical and Theoretical*, 41(34):344011, 2008.
- [73] Filip Sadlo, Alessandro Rigazzi, and Ronald Peikert. Time-dependent visualization of Lagrangian coherent structures by grid advection. In Valerio Pascucci, Xavier Tricoche, Hans Hagen, and Julien Tierny, editors, *Topological Methods in Data Analysis and Visualization*, Mathematics and Visualization, pages 151–165. Springer, Berlin, 2011.
- [74] Francois Lekien and S. D. Ross. The computation of finite-time Lyapunov exponents on unstructured meshes and for non-Euclidean manifolds. *Chaos*, 20:017505, 2010.
- [75] Daniel A. Nelson and Gustaaf B. Jacobs. DG-FTLE: Lagrangian coherent structures with high-order discontinuous-Galerkin methods. *Journal of Computational Physics*, 295:65–86, 2015.
- [76] Daniel A. Nelson and Gustaaf B. Jacobs. Computation of forward-time finitetime Lyapunov exponents using discontinuous-Galerkin spectral element methods. In ASME 2013 International Mechanical Engineering Congress and Exposition. ASME, November 2013.
- [77] Melissa A Green, Clarence W Rowley, and Alexander J Smits. The unsteady three-dimensional wake produced by a trapezoidal pitching panel. *Journal of Fluid Mechanics*, 685:117–145, 2011.
- [78] João H Bettencourt, Cristóbal López, and Emilio Hernández-García. Oceanic three-dimensional lagrangian coherent structures: A study of a mesoscale eddy in the benguela upwelling region. *Ocean Modelling*, 51:73–83, 2012.
- [79] João H Bettencourt, Cristóbal López, and Emilio Hernández-García. Characterization of coherent structures in three-dimensional turbulent flows using the finite-size lyapunov exponent. *Journal of Physics A: Mathematical and Theoretical*, 46(25):254022, 2013.
- [80] Brent Knutson, Wenbo Tang, and Pak Wai Chan. Lagrangian coherent structure analysis of terminal winds: Three-dimensionality, intramodel variations, and flight analyses. *Advances in Meteorology*, 2015, 2015.
- [81] Christoph Garth, Florian Gerhardt, Xavier Tricoche, and Hans Hagen. Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1464–1471, 2007.

- [82] J. A. Bourgeois, P. Sattari, and R. J. Martinuzzi. Coherent vortical and straining structures in the finite wall-mounted square cylinder wake. *Int. J. Heat Fluid Fl.*, 35:130–140, 2012.
- [83] Filip Sadlo and Ronald Peikert. Efficient visualization of lagrangian coherent structures by filtered amr ridge extraction. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1456–1463, 2007.
- [84] Steven L. Brunton and Clarence W. Rowley. Fast computation of finite-time Lyapunov exponent fields for unsteady flows. *Chaos*, 20:017503, 2010.
- [85] Frank M White and Isla Corfield. *Viscous Fluid Flow*. McGraw-Hill New York, 3 edition, 2006.
- [86] David A. Kopriva. Implementing Spectral Methods for Partial Differential Equations. Springer, New York, 2009.
- [87] F. Q. Hu, M. Y. Hussaini, and J. Manthey. Application of low dissipation and dispersion Runge-Kutta schemes to benchmark problems in computational aeroacoustics. In *ICASE/LaRC Workshop on Benchmark Problems in Computational Aeroacoustics (CAA)*, pages 73–98. National Aeronautics and Space Administration, October 1994.
- [88] Daniel A. Nelson, Gustaaf B. Jacobs, and David A. Kopriva. Effect of boundary representation on viscous, separated flows in a discontinuous-Galerkin Navier-Stokes solver. *Theoretical and Computational Fluid Dynamics*, 2015. In Review.
- [89] Ira Herbert Abbot and Albert Edward Von Doenhoff. *Theory of wing sections, including a summary of airfoil data*. Dover Publications, New York, 1959.
- [90] Theodore Theodorsen. Theory of wing sections of arbitrary shape. NACA Report No. 411, 1931.
- [91] Charles L. Ladson and Cuyler W. Brooks Jr. Development of a computer program to obtain ordinates for NACA 4-digit, 4-digit modified, 5-digit, and 16-series airfoils. *NASA TM X-3284*, 1975.
- [92] Charles L. Ladson, Cuyler W. Brooks Jr., Acquilla S. Hill, and Darrell W. Sproles. Computer program to obtain ordinates for NACA airfoils. NASA TM 4741, 1996.
- [93] Ralph L. Carmichael. Algorithm for calculating coordinates of cambered NACA airfoils at specified chord locations. In AIAA Aircraft Technology, Integration and Operations Conference. American Institute for Aeronautics and Astronautics, October 2001.

- [94] Mark Drela. Xfoil user's manual, 2000.
- [95] Program Development Corporation. GridPro v4.1: User's guide and reference manual for TIL, Ggrid and other utilities, 1999.
- [96] Claudio Canuto, M Hussaini, A Quarteroni, and TAJ Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, New York, 1987.
- [97] LE Jones, RD Sandberg, and ND Sandham. Direct numerical simulations of forced and unforced separation bubbles on an airfoil at incidence. *Journal of Fluid Mechanics*, 602:175–207, 2008.
- [98] Serhiy Yarusevych, Pierre E. Sullivan, and John G. Kawall. On vortex shedding from an airfoil in low-Reynolds-number flows. *J. Fluid Mech.*, 632:245–271, 2009.
- [99] K Dortmann. Computation of viscous unsteady compressible flow about airfoils. In 11th International Conference on Numerical Methods in Fluid Dynamics, pages 230–234. Springer, 1989.
- [100] G. B. Jacobs, D. A. Kopriva, and F. Mashayek. Validation study of a multidomain spectral element code for simulation of turbulent flows. *AIAA J.*, 43(6):1256–1264, 2004.
- [101] A. Surana, G.B. Jacobs, and G Haller. Extraction of separation and reattachment surfaces from 3d steady shear flows. *AIAA J.*, 45:1290–1302, 2007.
- [102] K. Sengupta, G.B. Jacobs, and F. Mashayek. Large-eddy simulation of compressible flows using a spectral multi-domain method. *Intl J. Numer. Meth. Fluids*, 61(3), 2008.
- [103] K. Sengupta, B. Shotorban, G.B. Jacobs, and F. Mashayek. Spectral-based simulations of particle-laden, turbulent flow. *Int. J. Multiphase Flow*, 35(9), 2009.
- [104] Gustaaf B. Jacobs, David A. Kopriva, and Farzad Mashayek. Towards efficient tracking of inertial particles with high-order multidomain methods. *J. Comp. Phys.*, 206:392–408, 2007.
- [105] Daniel Karrasch, Mohammad Farazmand, and George Haller. Attraction-based computation of hyperbolic Lagrangian coherent structures. J. Comp. Dyn., 2014. Submitted.
- [106] Theodore J. Rivlin. *An Introduction to the Approximation of Functions*. Dover, New York, 1981.

- [107] Changquing Hu and Chi-Wang Shu. Weighted essentially non-oscillatory schemes on triangular meshes. J. Comp. Phys., 150:97–127, 1999.
- [108] Daniel Nelson and Gustaaf Jacobs. High-order visualization of three-dimensional Lagrangian coherent structures with DG-FTLE. *Computers and Fluids*, 2015. In Preparation.
- [109] Kendall Atkinson and Weimin Han. *Theoretical numerical analysis*, volume 39. Springer, 2005.
- [110] James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30. Siam, 1970.
- [111] Hervé Vandeven. Family of spectral filters for discontinuous problems. *Journal of Scientific Computing*, 6(2):159–192, 1991.
- [112] Wai Sun Don. Numerical study of pseudospectral methods in shock wave applications. *Journal of Computational Physics*, 110(1):103–111, 1994.
- [113] George Haller. An objective definition of a vortex. J. Fluid Mech., 525:1–26, 2005.
- [114] Mohamed HM Sulman, Helga S Huntley, BL Lipphardt, and AD Kirwan. Leaving flatland: diagnostics for lagrangian coherent structures in three-dimensional flows. *Physica D: Nonlinear Phenomena*, 258:77–92, 2013.
- [115] E Meiburg and JC Lasheras. Comparison between experiments and numerical simulations of three-dimensional plane wakes. *Physics of Fluids (1958-1988)*, 30(3):623–625, 1987.
- [116] E Meiburg and JC Lasheras. Experimental and numerical investigation of the threedimensional transition in plane wakes. *Journal of Fluid Mechanics*, 190:1–37, 1988.
- [117] JC Lasheras and E Meiburg. Three-dimensional vorticity modes in the wake of a flat plate. *Physics of Fluids A: Fluid Dynamics (1989-1993)*, 2(3):371–380, 1990.
- [118] Charles HK Williamson. Vortex dynamics in the cylinder wake. *Annual review of fluid mechanics*, 28(1):477–539, 1996.
- [119] Richard R Kerswell. Elliptical instability. *Annual review of fluid mechanics*, 34(1):83–113, 2002.