# Lagrangian Coherent Structures: A Climatological Look



Andrew Sven McCall Jr.

Advisors: Dr. Chris Danforth and Dr. Lesley-Ann Dupigny-Giroux

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Arts

in the College of Arts and Sciences Faculty of Mathematics and Geography

March 28th, 2018

#### Abstract

A relatively new area at the crossroads of fluid and nonlinear dynamics are objects known as Lagrangian Coherent Structures (LCSs). LCSs are mathematically classified to differentiate parts of fluid flows. They, themselves, are the most influential parts of fluids. These objects have the most influence on the fluids around them and they allow for a sense of hierarchy in an otherwise busy environment of endless variables and trajectories. While all particles of fluids have the same dynamics on an individual basis, areas of fluid are not created equal and to be able to detect which parts will be the most important to look at allows for easier, bust just as accurate, prediction of fluid movement. Recent applications include cleanup operations during the BP Deepwater Horizon oil spill, pollutant transfer in oceanic basins, and the analysis of polar storm activity. This thesis explores LCSs from the discrete mathematics to the future climatological impacts using virtual fluid simulations and LCS detection tools to facilitate analysis.

# Contents

	Ack	nowledg	gements	. 4		
1	Introduction					
	1.1	What are Lagrangian Coherent Structures?				
		1.1.1	Name Breakdown	. 5		
		1.1.2	Basics of Lagrangian Dynamics: Material Lines and Manifolds .	. 7		
		1.1.3	Lyapunov, FTLEs, and Strainlines	. 8		
		1.1.4	Where Did They Come From?	. 9		
	1.2	Litera	ture Review	. 10		
	1.3	Motiv	ation - The "So What? Who Cares?"	. 13		
2	Lag	grangi	an Coherent Structure Detection and MATLAB	14		
	2.1	1 The Code				
	2.2	8	. 16			
		2.2.1	Double-Gyre	. 16		
		2.2.2	Bickley Jet	. 18		
		2.2.3	Ocean Simulation	. 20		
3	The	The Mathematics				
	3.1	The Four Step Process				
		3.1.1	Flow Map Set Up	. 22		
		3.1.2	Flow Map Gradient	. 23		
		3.1.3	Hypersurfaces and Strain Tensors	. 23		
		3.1.4	Cauchy-Green Strain Tensor Results and Finding LCSs	. 25		
	3.2	Worked Example				
4	Climatology and LCSs					
	4.1	A Brief Background of Weather Prediction				
	4.2	4.2 Fukushima: An LCS Case Study				
		4.2.1	The Process	. 33		
		4.2.2	The Results	. 35		
	4.3	Discu	ssion: LCSs and Climatology	. 38		

To my father

# Acknowledgements

Often times, subjects seem to daunting to undertake. Whether they require too much background knowledge or they are simply hard concepts, some subjects are just inherently left in the dark. The main aim of this project was to examine a topic that was complex, young, and full of potential and dissect it until it was as simple as anything else; to look at LCSs with a fine-tooth comb and have a piece of literature that put things into laymen terms as best as possible.

I would like to extend my most sincere thank-you to my two thesis advisors Chris and Lesley-Ann for I would not be where I am today without them. Far beyond my thesis, they have gotten me to the place I am today and have pushed me to learn that no project is too daunting to jump into. Also, like anything in my life, I have my family to thank; my mom and sisters are my drive for everything I do.

# Chapter 1

## Introduction



**Figure 1.1** Examples of fluid structures whose dynamics might benefit from LCS detection. (Left) Deep water Horizon spill in Gulf of Mexico (NASA), (Middle) Jupiter's Great Red Spot seen from the Voyager 1 mission in February 1979 (NASA/JPL), (Right) Water spout in the Florida Keys (Joseph Golden/NOAA). [1].

## 1.1 What are Lagrangian Coherent Structures?

George Haller, the scientist most responsible for the growing interest in LCSs, says that LCSs are "the hidden skeleton of fluid flows" [2] and this is the best way to start to understand the idea behind LCSs because they are truly an idea. While LCSs can manifest in structures and we can mathematically detect them, it is important to remember that at their core, they are a way of categorizing fluids with specific characteristics. Mathematics is a framework for translating the behavior of our natural world into a universal language; writing the cookbook for the cake that is already made.

### 1.1.1 Name Breakdown

Where we need to start with LCSs is actually their name: Lagrangian Coherent Structures. When the word 'Lagrangian' is put first, all that is really being said is that whatever we are talking about is a function that is describing the dynamics of a system. The functions are interacting with space and time and contain information about how the system is functioning. However, in fluid dynamics, there are two ways of looking at fluids and that is to use Eulerian or Lagrangian dynamics. Eulerian dynamics looks at a specific frame of reference and examines fluids that pass by this frame at each time; these dynamics are more focused on a specific area than specific parcels of fluid. Lagrangian dynamics do the opposite: focus on specific parcels of fluid and follow them through space [2]. Put simply, Eulerian dynamics would be like setting a video camera up over a certain part of the ocean and never moving that camera, whereas Lagrangian dynamics would be like having a camera follow a specific vortex in the ocean. Both dynamics have their pros and cons, and the most important thing to consider about LCSs is that we are concerned with specific structures and not places, which means Lagrangian dynamics are the way to go.

Coherent structures are sometimes difficult and controversial to define [3]. The one thing that everyone can agree on is that a coherent structure has significant correlation in space and time [4]. A structure is considered coherent in our case if it lasts a long time in the flow of the observed fluid. In other words, it is not transient and the structure does not change fast enough to be disregarded. This is a very subjective thing and all depends on the amount of time a person is observing a system. Overall we think of coherent structures as a lasting mathematical object in the system.

The best example of coherence versus non-coherence is actually with a couple of structures everyone is familiar with: eddies and vortices. Imagine a river with tons of flowing water. Eddies are the swirls of water that eventually join back up with the rest of the flow of water. We would consider eddies a non-coherent structure because they form and decay within a few dynamic times. If we take a vortex in comparison, this is a structure that consistently swirls water and often in one location due to the constant dynamics and interaction with the surrounding landscape. Since the vortex lasts a considerable amount of time, we would say it is a coherent structure.

Now that the name is broken down, LCSs already seem to be a lot simpler. In any regard, finding structures in fluids is a way to create a sort of hierarchy. It is a way of stating that in fact, all particles of fluid are equal, but the dynamics within a fluid flow are not. Some areas of fluid have the ability to influence the particles around them more than others; these are what LCSs are. To truly get a grasp on them however, we need to talk about certain structures in fluids that are called material lines.

### 1.1.2 Basics of Lagrangian Dynamics: Material Lines and Manifolds





Haller brings up a key point in his paper which allows us to recognize the upbringing of LCSs but ultimately notes that "the mathematical methods used to identify key material lines in steady, periodic, and quasiperiodic flows rely on knowing the flow field for all time. But the flows that most need to be understood are typically aperiodic" [2]. So, while it is possible to find examples of unstable manifolds in nature, like those in Figure 1.3, these are not the flows which we are usually needing to solve prediction problems for.

Nonetheless, Figure 1.2 is a great graphical start to understanding LCSs. As a parcel of fluid approaches a saddle point - a point of stagnation where the stable and unstable manifold meet - on the stable manifold, which acts as a repulsive barrier, the parcel will deform to take the shape of the unstable manifold. As stated before, these unstable manifolds can be seen in nature sometimes, like that in Figure 1.3, but they are rare and usually have trivial solutions.



**Figure 1.3** Tristan da Cunha island in the Southern Atlantic creating von-Karman vortices behind it with consistent airflow. The surrounding clouds take the shape of this airflow and reveal the unstable manifolds [5].

#### 1.1.3 Lyapunov, FTLEs, and Strainlines

In dynamical systems (sometimes referred to as chaos theory) the Lyapunov number and Lyapunov exponent are introduced as a way to quantify the rate of separation between two points in a flow. The Lyapunov exponent is simply the natural logarithm of the Lyapunov number. If we had a Lyapunov number of 3 or Lyapunov exponent of  $\ln 3$ , this would mean the distance between two points we are observing is tripling each time step. Alternatively, if we had a Lyapunov number of  $\frac{1}{3}$ , the distance would be cutting by a third each time-step [6]. Lyapunov numbers become very important to us when we look into the mathematics in chapter 3 and start to look at which pieces in our math stand out and allow us to find LCSs. For our purposes, these numbers allow us to quantify the amount of deformation occurring in a fluid flow. The term we use to describe this is the Finite-Time Lyapunov-Exponent (FTLE).

FTLEs are important because as a velocity field is analyzed in forward time, regions of fluid which have high separation have high FTLE values. The same thing can be done in reverse time to determine which regions have strong divergence in backward time (which reveals the regions with strong convergence) and correspond to areas with small FTLE values [2]. However, LCSs have proven to have some problems arise when they are being

explicitly used such as false positives and false negatives which become problematic when we need to reliable look for LCSs each and every time. If we remember, material lines are at the basics of Lagrangian transport. If we look back to Figure 1.2, these are the lines the material travels along or moves against (both the stable and unstable manifolds). Strainlines introduce a very similar yet different type of line. As material lines deal with deformation, strain is looked at on an infinitesimal level. This idea is explained more indepth in chapter 3 but we mention it here as a solution. Farazmand and Haller [7] have "shown that repelling LCSs are, in fact, material lines whose initial positions are locally the most repelling strainlines for the time window in question" [2].

#### **1.1.4** Where Did They Come From?

The last piece to fully understanding LCSs is actually where this idea came from. As stated before, the original person to come up with the modern idea of LCSs is a man by the name of George Haller. In 1994, Haller joined the Department of Applied Mathematics at Brown University as an associate professor and it is here that the first inklings of LCSs came about. Haller was in search of a problem to pursue and with his studies in chaos theory and with large amounts of atmospheric and ocean data at hand, the Honk Kong project was born.

While Haller was at Brown, he started working on a project with Wenbo Tang of Arizona State University and Pak Wai Chan of the Hong Kong Observatory. The Hong Kong airport is known for its rocky landings and turbulent air. One thing that does not help the airport, as seen in Figure 1.4, is that it is not only on an island but it is on a man-made island off the east coast of Hong Kong. When Haller and the team came together, the airport was having a lot of trouble with wake turbulence. Just like water, air is a fluid, and when a massive object moves through the air, this causes a wake (just like that of a boat in water) [8].



**Figure 1.4** Hong Kong International Airport (Chek Lap Kok Airport) opened July 6, 1998 on an artificial island after realizing that the current airport, Kai Tak Airport, was at capacity for take-offs and landings [9].

Haller and the team installed lasers on the runway to track aerosols - which allow us to see the effects of air as opposed to the air itself. By looking at the movement of aerosols, Haller used Lagrange's methods (which Lagrange couldn't advance himself due to lack of electronic computers) and found the coherent structures- the most influential areas in the fluid. With real time data crunching, the hope was that these areas could be marked on computers so air traffic controllers could judge the safety of planes departing and landing [8]. It was from here that LCS theory was modernized and born and it took off (pun intended).

## **1.2 Literature Review**

This is an unconventional thesis and with it comes an unconventional literature review. There have been a number of scientific papers written using Haller's "pioneering work" [10] from 2005 and most of them are extremely technical. This brief literature review introduces the main ideas that people have been using LCS detection for. Largely, they have been for climatological events and these will be discussed in detail, but LCSs are new and it needs to be understood that the possibilities are still growing.

The paper that got me excited about the subject of LCS detection was published in 2017, Serra et al. [11] *Uncovering the Edge of the Polar Vortex*. The results of the paper show that

Elliptic LCS theory accurately predicted where the edge of the polar vortex was by doing a backwards time analysis. Knowing the position of the edge of polar vortex has massive implications. It allows you to know so much about the climate, especially the stratospheric ozone hole, and the movement of this edge has major influence on the surface weather.



**Figure 1.5** 3D Reconstruction of the edge of the polar vortex on (a) Dec. 28th, 2013 and (b) Jan. 7th, 2014 as produced by Serra et al. [11]. A video of the animation of these structures can be seen here.

Similarly, Tang et al. [12] looked for coherent structures in the subtropical Jetstream. This process has been done before, but a common theme among many of these papers is the idea that Haller puts forward in his 2005 paper when he talks about how structures could be evident to one frame a not another. Since LCSs use a particle-based definition of structures, this ensures that frame independence is present as Tang et al. say that "any newly proposed constitutive law or flow quantity must be fully frame independent to be considered intrinsic to the properties of the moving continuum" [12]. It is key that in order to validate the properties of a structure in fluid, Lagrangian dynamics must be used.

This is a common theme through most of the papers. Most of the papers use LCSs for a specific reason, not because they are new and exciting. Looking to Gough et al. [13], Harrison et al. [14], and Beron-Vera et al. [15] which take LCSs to the ocean but with slightly different approaches. Gough et al. and Harrison et al. look at coastal upwelling off the coast of California, and use radar data to track fluid movement. Their goal was to find LCSs because "attracting FTLE field maxima can identify confluence and shear in flows which can be useful in mapping dynamics associated with fronts" [13]. Mapping ocean fronts was key to learning all sorts of things about the cold current that travels off the coast of California. Beron-Vera et al. used hydrography and altimetry to categorize mesoscale ocean eddies from surface ocean currents. If we go back to the example of the vortex and the eddy, this paper used that very example on a massive scale. Put simply, they were trying to use LCSs and other dynamical systems tools to figure out which flows in the ocean were most important (ocean currents) and which were structures that were byproducts of other

flows (eddies).

One of the biggest issues that people are trying to investigate today is pollutant transport. Pollutant transport is becoming an ever-encroaching problem as the planet seems to become smaller and smaller. Volcanic eruptions, oil spills, nuclear power plant meltdowns, and the like all have the ability to scare us straight. The most well known of these papers is where Olascoaga and Haller [16] took the BP Deep Water Horizon oil spill and hindcasted it to show that LCSs could aid in showing where wind was going to take the oil. These results were again "based on new mathematical results on the objective (frame-independent) identification of key material surfaces that drive tracer mixing in unsteady, finite-time flow data" [16].



**Figure 1.6** Figure 5 from Olascoaga and Haller [16] where the orange area shows the observed oil spread and the positions of the simulated oil patches in blue that shows very similar results.

While there are many more papers on atmospheric, oceanic, or pollutant studies, many focus on unconventional uses for LCSs. Papers like Qi and Huang [17] and Qi and Xu [18] focused more on the astrophysics side of things looking into *n*-body problems and planetary orbits. Others look at the heart of LCSs: their computation. Sun et al. [10] were able to look at the "trajectory of each fluid particle [which was] explicitly tracked over the whole simulation" [10]. No matter what the problem was though, fluid transport was involved and the investigations have only just begun.

# 1.3 Motivation - The "So What? Who Cares?"

Now that we know what LCSs are and what scientists are doing with them, it seems fitting to ask ourselves a question we, as curious human beings, should always ask - so what? The implications for using LCS detection are far reaching, especially in the world of climate. We focus in on applications of LCSs to the climate in section 4.3 but is there anything else? Are there other areas that are important to mention?

Often, for the layman, the word 'theory' makes things intangible. Outside of academia, applicable ideas and writings are taken to mean they are not tangible - they do not have any meaning in the real world; that could not be further from the truth. Academic studies and papers give rise to the most useful things in the universe. From the phone in your pocket and satellites guiding you to that restaurant to the stocks you trade at light-speed on a daily basis, theories are at work every second.

Yes, LCSs are just another theory and worse than that, they aren't even physical objects; they are mathematically characterized. More important than any of that though, they are a way of understanding something very complex: fluids. Fluids are all around us and we interact with them everyday. From the blood that courses through your veins to the bumpy turbulence you have on your flight, the truth is: we are in a world of fluids and there is no avoiding them. The better we can understand how fluids move and the more steps ahead of fluid we can be, the better off the world will be. Being able to predict the exact location of a hurricane a week in advance, having five days notice of which areas are going to be covered in frost, how much moisture is coming to the mountains before your ski vacation, the amount of lipid transport in your blood, the movement of super storms on the surface of Jupiter and Saturn, the movement of the polar vortex and how harsh the next cold snap will be; these are all possibilities in a world with LCS detection.

More important than anything, LCSs are not going to allow us to see the weather a month in advance, they are not going to be able to stop natural disasters, and they won't be able to stop global warming. Nevertheless, they are objects that allow a lot of those terrible problems and seemingly unsolvable issues to become that much more tangible and solvable.

# **Chapter 2**

### Lagrangian Coherent Structure Detection and MATLAB

I would first like to point out that this project would have been a lot less likely to happen if it were not for Geroge Haller and his group allowing anyone to download the code I used for this project. The code, simply called LCS Tool, is a package that is downloadable by anyone and provides an amazing amount of detail on coding LCS detection [19]. While the mathematics are explained and drawn out in section 3.1 building this code from scratch would have taken way too long to complete in an undergraduate setting. This code, which comes with demos, has allowed me to learn about LCSs and LCS detection visually, mathematically, and through the power of a computer.

## 2.1 The Code

The code comes from George Haller's website (http://georgehaller.com) where their GitHub is likned. It has a suite of very interesting tools, but for my purposes, the LCS Tool was the one I found to be most useful. It comes zipped and ready for matlab directly. It is well documented and contains demo files which we will look at in section 2.2. For this section, we are going to look at which key pieces of code I used the most, break them down further and explain which pieces of code are controlling what, and finally explore what this code might hold for the future.

The very first piece to play with in the code is to actually see flows in action and that was done by a couple of functions working together while using some demo data. The *animate-flow* file and the *flow-animation* file were used to pull gridded data and plot it through time as an animation. The section of *animate-flow* that was manipulated the most were the lines below in lines 17 to 21 where I could manipulate how the fluid looked but not how it behaved. These two files of code were my gate into understanding how a computer was going to try and evaluate such a complex object like fluid flow using nothing but data, interpolation, and clever mathematics.

```
1 % Set default values for flow structure
p = inputParser;
3 p.KeepUnmatched = true;
```

```
addParamValue(p, 'coupledIntegration', true, @islogical);
  parse(p, flow);
  coupledIntegration = p. Results. coupledIntegration;
  if coupledIntegration
      position = deval(flow.solution,flow.timespan(1));
      position = [position (1:2:end-1), position (2:2:end)];
      position = transpose(position);
11
  else
      position = arrayfun (@(iSolution) deval(iSolution, flow.timespan(1)),
     flow.solution, 'UniformOutput', false);
      position = cell2mat(position);
  end
p_1 = plot(mainAxes, position(1, :), position(2, :));
  set(p1, 'LineStyle', 'none')
 set(p1, 'Marker', 'o')
19
  set(p1, 'MarkerFaceColor', 'm')
21 set (p1, 'MarkerSize', 3)
```

However, once playing with animations got old and I had figured out all of the naming issues that prevented animations from running and doing ordinary directory housekeeping, I wanted to figure out exactly how the LCS detection was going to work. The entire .zip file of code comes with around 30 functions that are needed in the three or four files that run different variations of LCS detection. The one I used for this project was a file called *elliptic-hyperbolic-lcs*. This is the file the Haller Group uses as an example to run in their *read-me* on the double gyre demo.

In short, this piece of code does exactly what is discussed in chapter 3 but I am going to summarize it here. The code starts with a .mat file containing the data and variables it needs in order to run. The most important of these is a 3D matrix that is  $t \times m \times n$  where t is the time step, and m and n are the gridded data of latitude and longitudinal velocities during each time interval. For example, if we had a  $52 \times 334 \times 311$  matrix, it would represent a gridded set of vector data (velocities at each point) over a size 52 time interval. The code takes in velocity from the latitudes and the longitudes and uses them to do a gridded interpolation with a built in function to MATLAB. Essentially it takes a bunch of gridded data over time and interpolates it between steps to fill in the gaps. Once the data is in and interpolated, the mathematics take over. Simply put, the strainlines for the entire region are computed and the code looks for areas with Lyapunov exponents that are closest to 0. In chaos theory, we must remember that Lyapunov exponents are what define the amount of stretching and shrinking that is occurring at a point in our system. The importance of Lyapunov exponents in our math is explained in section 3.1.4 as well as why they are significant to find in section 1.1.3.

## 2.2 Demos

While the code itself it very important and taught me a lot about how the Haller Group thought about coding LCS detection into MATLAB to help find regions of interest in certain data, the majority of my learning came from the examples themselves. The ability to look at an image or simulation and reference back to the code to see which pieces of code controlled which output was the most valuable piece of this project. Haller's code came with three demo files representing three distinct applications of LCS detection for fluid flow. The first two analyze computational fluid dynamics simulations whereas the third uses a data set which I have touched on briefly. I will be showing three figures with each demo. The first will be the fluid simulation, the second will be the repelling LCSs, and the third will be the attracting LCSs. Additionally, there will be links to video animations if you are electronically viewing this file.

An additional step I took in this process was to break the code up into smaller time intervals while it evaluated LCSs. Originally, the code evaluated the entire time series and outputs one image for repelling LCSs and one image for attracting LCSs. I figured it would be more useful if you could track these changes in LCSs over time as well and so I broke the intervals into roughly five sections. When these are overlaid with the animations, we can really get a good sense of how these things are moving in time and how they may help us predict what a fluid is going to do in the future.

## 2.2.1 Double-Gyre

A gyre is a term used in the study of oceanic flow and NOAA describes it as "refer[ing] simply to large, rotating ocean currents" [20]. Abarbanel and Young start out their entire book on the basics of ocean circulation talking about early data used on world maps which use abbreviations 'SG' which stand for 'Subtropical Gyre' [21]. The term double-gyre does relate to this but the double-gyre refers to a simulated experiment that is common in fluid dynamics papers. It looks at interactions between two counter-flowing currents and usually watches the transport of fluid around these and this demo is no exception to that. The double-gyre is a standard demo for many tools like this.

```
epsilon = .1;
amplitude = .1;
omega = pi/5;
flow.imposeIncompressibility = true;
flow.derivative = @(t,x,useEoV) derivative(t,x,useEoV, epsilon, amplitude,
omega);
flow.domain = [0,2;0,1];
flow.timespan = [0,10]; %speed of event
flow.resolution = [1,1]*10;
flow = animate_flow(flow);
```



**Caption:** Small section of code required to simulate the double-gyre demo

**Figure 2.1** Output figures from the LCS detection of the double-gyre demo with (top) attracting LCSs in blue, (bottom) repelling LCSs in red and in both images the elliptic LCSs are in green.

### 2.2.2 Bickley Jet

Named after W.C. Bickley who first wrote about this structure in 1937 [22], the basic idea behind this flow is it is "a meandering zonal jet flanked above and below by counter rotating vortices" [23]. The flow is suppose to mimic natural flows that could happen but in a more idealized environment. As del-Castillo-Negrete and Morrison [24] point out, the action of Rossby Waves - fast-moving upper atmospheric jetstreams responsible for the index cycle - will be captured simplistically by using the Bickley jet model.



**Figure 2.2** A view of what del-Castillo-Negrete and Morrison [24] is discussing in his paper but with real data and not an idealized form. This image is from the online tool EarthWindMap which you can find here in this exact orientation. This view is at a 250mb level.

```
u = 62.66;
  lengthX = pi*earthRadius;
  lengthY = 1.77e6;
  epsilon = [.075, .4, .3];
  flow.imposeIncompressibility = true;
  flow.periodicBc = [true, false];
  perturbationCase = 3;
  phiTimespan = [0, 25];
  phiInitial = [0,0];
  phiSol = ode45(@d_phi, phiTimespan, phiInitial);
13 timeResolution = 1e5;
  phi1 = deval(phiSol, linspace(phiTimespan(1), phiTimespan(2),
      timeResolution),1);
15 phi1Max = max(phi1);
  flow.derivative = @(t, x, \tilde{}) derivative (t, x, false, u, length X, length Y, epsilon
      , perturbationCase , phiSol , philMax );
17
  flow.domain = [0, lengthX; [-1, 1] * 2.2599 * lengthY];
19 flow.timespan = [0, 4 * \text{length} X / u];
  % Make grid Cartesian
_{21} resolution X = 100;
  gridSpace = diff(flow.domain(1,:))/(double(resolutionX)-1);
23 resolutionY = round(diff(flow.domain(2,:))/gridSpace);
  flow.resolution = [resolutionX, resolutionY];
25
  flow = animate_flow(flow);
```

Caption: Small section of code required to simulate the bickley jet demo





#### 2.2.3 Ocean Simulation

The third and final demo is different than the other two in that it uses real, approximated data. The data comes from a section of the South Atlantic Ocean when Beron-Vera and Haller [25] gathered "data derived from satellite-observed sea-surface heights under the geostrophic approximation" [23]. It is crucial to note that this data uses geostrophic approximation because it is most likely the main reason for a number of errors in the case study of this project which is talked about in section 4.2. The data is given at a spatial resolution of  $1/4^{\circ}$  and a temporal resolution of 7 days.



**Figure 2.4** Output figures from the LCS detection of the ocean dataset demo with (top) attracting LCSs in blue, (bottom) repelling LCSs in red and in both images the elliptic LCSs are in green. You can click here to see the video animation overlay with the LCS and repelling strain lines that are segmented into 5 subintervals.

# **Chapter 3**

### The Mathematics

George Haller describes LCSs in-depth in many of his papers [1]. Here, I will break down the mathematics again, but in 4 stages that allow us to see exactly what process we go through to gain a perspective of a fluid's structure. Then, we will use these discrete steps and see exactly how these LCSs and strainlines are being computed [2].

## 3.1 The Four Step Process

When Haller released a paper in 2013 about LCSs and their detection, he included a small box that broke down LCS detection into four simple steps [2]. I paraphrase those four steps as follows:

1.) Given a velocity field, define a flow map which shows how particles move from their initial position to any time t in the fluid.

2.) Take the gradient of the flow map

3.) Transpose that gradient and multiply by itself to get the Cauchy-Green Strain Tensor

4.) Evaluate your computation and look for strainlines and LCSs using eigenvalues and eigenvectors of the strain tensor.

I will explicitly walk through each of these steps below and set up a generalized form of finding strainlines and LCSs in a fluid. Some vocabulary mentioned in these steps will be addressed in the following subsections for clarity.

## 3.1.1 Flow Map Set Up

The first thing we do is we start with a velocity field  $\vec{v}(\vec{x},t)$  over the time interval  $[t_0, t_1]$  in a bounded flow domain U which generates trajectories through the following differential equation:

 $\dot{x} = \vec{v}(\vec{x}, t), \quad \vec{x} \in U, \quad t \in [t_0, t_1], \quad U \subset \mathbb{R}^2$  (1)

The next thing we have to do is define our points and how they function in different dimensions. When we have an initial position in a fluid,  $\vec{x}$ , in general, we see:

$$2D : \vec{x} = (x^1, x^2), \quad U \in \mathbb{R}^2$$
$$3D : \vec{x} = (x^1, x^2, x^3) \quad U \in \mathbb{R}^3$$
$$\vdots$$
$$nD : \vec{x} = (x^1, x^2, \cdots, x^n) \quad U \in \mathbb{R}^n$$

where the superscript is an integer representing the spatial dimension of our particle position.

Solutions to (1) are  $\vec{x}(t; t_0, \vec{x_0})$  where  $\vec{x_0}$  is our initial position of  $\vec{x}$  at  $t_0$ . From here on, we will explicitly work in two dimensions but the same process works for any dimension. All of this brings us to our definition of the flow map,  $\vec{F}$ .  $\vec{F}$  is a function which defines the movement of a particle  $\vec{x}$  in our fluid. This particle which we have defined  $\vec{x_0}$  at  $t_0$  and  $\vec{x_1}$  at  $t_1$  gets mapped to  $t_1$  via  $\vec{F}$ . We define  $\vec{F}$  as follows:

$$\vec{F}_{t_0}^{t_1}(\vec{x}_0) = \vec{x}(t; t_0, \vec{x}_0), \quad \vec{x} \in U, \quad t \in [t_0, t_1]$$
 (2)

#### 3.1.2 Flow Map Gradient

Next, we also must remember from our differential equation (1), follows the equation of variations:

$$\dot{y} = \nabla \vec{v}(\vec{x}(t;t_0,\vec{x}_0))y \qquad (3)$$

We can obtain this generic equation from any differential equation. When we take the gradient of F, we get an invertible matrix that also serves as the the fundamental matrix solution to (3). As a reminder, we define  $\nabla F$  as:

$$\nabla \vec{F}_{t_0}^{t_1}(\vec{x}_0) = \begin{bmatrix} \frac{\partial x_1}{\partial x_0} & \frac{\partial x_1}{\partial y_0} \\ \frac{\partial y_1}{\partial x_0} & \frac{\partial y_1}{\partial y_0} \end{bmatrix}$$
(4)

#### **3.1.3** Hypersurfaces and Strain Tensors

Next, we need to consider a hypersurface,  $\vec{\mathcal{M}}(t)$ , of initial fluid positions. Recall that this hypersurface is smooth and has dimension equal to  $\dim U - 1$ . We then define a material surface as

$$\vec{\mathcal{M}}(t) = \vec{F}_{t_0}^{t_1}(\mathcal{M}(t_0)) \qquad (5)$$



Figure 3.1 A visual representation of a hypersurface under the flow map  $\vec{F}$ . (a) represents  $\vec{\mathcal{M}}(t_0)$  and (b) is where  $\vec{\mathcal{M}}(t)$  ends at time t which would make it our material surface of interest.

Why do we care about these hypersurfaces? When we are dealing with two-dimensional fluid flow, our hypersurfaces and material surfaces are actually just lines because they are one-dimensional. However, it helps to talk about hypersurfaces for the generalization of our mathematics since in any dimension higher, they would need to be explained anyway.

We are interested in finding material surfaces that have a lot of influence on the points around them. The perturbations that show us if a material surface does in fact have a large influence on the points around them come from equation (3). Since we know that the gradient of  $\vec{F}$  is the fundamental matrix solution to equation (3), we can write the following:

$$\dot{y} = \nabla \vec{v}(\vec{x}(t;t_0,\vec{x}_0))y \Rightarrow y(t) = \nabla \vec{F}_{t_0}^t(\vec{x}_0)y(t_0)$$

and since we can write this, we can define the squared magnitude at final time  $t_1$  as follows

$$|y(t_1)|^2 = \left\langle \nabla \vec{F}_{t_0}^{t_1}(\vec{x}_0) y(t_0), \nabla \vec{F}_{t_0}^{t_1}(\vec{x}_0) y(t_0) \right\rangle$$

Our final mathematical step in this whole process is right here but it takes a little bit of explaining. We can rewrite the above equation in terms of a variable C which is our Cauchy-Green Strain Tensor. Using the definition of C, below in (6), we can write our squared magnitude in terms of it. This allows us to calculate everything using matrix multiplication, eigenvectors and eigenvalues help us determine where strainlines will be, and which eventually leads us to where LCSs will be as well. We define C as follows

$$\vec{C}(\vec{x}_0) = [\nabla \vec{F}_{t_0}^{t_1}(\vec{x}_0)]^T [\nabla \vec{F}_{t_0}^{t_1}(\vec{x}_0)]$$
(6)

If we build from the very start, we need to define a tensor. A tensor is similar to a vector except it is more general and has an array of components to describe an object in space and which directions it is moving in. This piece allows us to use the idea of deformation and strain.

When we set up something like the deformation gradient of our flow map  $\nabla \vec{F}$ , it allows us to have a matrix representation of what type of transition a set of points will undergo under

the flow map  $\vec{F}$ , but in an objective description. Strain is different from deformation. We need to think of a strain on an infinitesimal level. Strain tensors aim to look at infinitesimal perturbations around an area and often infinitely small spheres or circles are used to show how strain is visualized. Figure 3.2 is a good example of what sort of changes we expect to see. Now, in this visual we are only looking at the horizontal change in motion but effectively, the strain tensor is allowing us to see where this instantaneous change of directionality is.



Figure 3.2 A visual representation of strain in a fluid setting. As two parcels of fluid are mapped via the flow map  $\vec{F}$  from time  $t_0$  to t, their deformation is one thing but where the change of directionality occurs at an infinitesimal level points to a strain line (whether it be attracting or repelling). Note that in this example there is also vertical stretching going on that is not addressed, as our focus is on the horizontal direction.

Now that we understand how strain presents, it will become clearer how the Cauchy-Green Strain tensor is used. The Cauchy-Green strain tensor  $\vec{C}$  is the main matrix we use for the rest of our calculations. The worked example in section 3.2 might solidify these concepts as well. In the next section, I talk about how to look at the matrix  $\vec{C}$  and use its values to guide us in finding these strainlines and LCSs.

A final thing to note here is that from the nature of how  $\vec{C}$  is constructed, it is a symmetric matrix, so all of its eigenvalues  $\lambda_i \ge 0$  and eigenvectors  $\vec{\varepsilon_i}$  are orthogonal to each other. These characteristics become increasingly important as we talk about them in section 3.1.4.

#### 3.1.4 Cauchy-Green Strain Tensor Results and Finding LCSs

Once we have the Cauchy-Green strain tensor  $\vec{C}$ , we use it to try and find strainlines that exist in our frame. We calculate  $\vec{C}$  at every point in our frame. For each point, we get a

 $n \times n$  matrix when we are in *n*-dimensional space. The eigenvalues and eigenvectors become very important to us after this. We will not define the procedure to find these, as they should be known from linear algebra. If we assume that the eigenvalues of  $\vec{C}$  in ascending order are  $\lambda_1, \lambda_2$ , and  $\lambda_3$ , our interest lies with  $\lambda_1$  (aka the smallest eigenvalues of  $\vec{C}$ ). From Haller, we are looking for the eigenvector of the smallest eigenvalue of  $\vec{C}$ .

For definition purposes, we define the eigenvectors  $\vec{\varepsilon}_n$  that correspond to the eigenvalue  $\lambda_n$ . Now that we have named everything, we are always looking for  $\vec{\varepsilon}_1$  in every scenario which will be a length *n* vector. We need to keep in mind that these values are for a single point of interest we are looking at. If we compute these values for the whole frame of interest then we can find our strainlines. Haller says that strainlines will be tangent to  $\vec{\varepsilon}_1$  and each point we move in time will start to build a line that we can define as a strainline. There will be many strainlines throughout the fluid but there is still another step we must take to get from these strainlines to LCSs. If we look at Figure 3.3, we can see an example of these repelling LCSs in red from Haller's code using the Ocean dataset demo. The green lines are elliptic LCSs and we will talk about those next.



**Figure 3.3** An example dataset from Haller and his team shows a simulated ocean segment after an LCS analysis with repelling LCSs in red and elliptic LCSs in green.

Finding LCSs require the entire field of points to be computed for our fluid before we can point them out. Since LCSs are strainlines that act as boundaries and structures, once all strainlines are computed, we are looking for the locally highest averaged values of Cs largest eigenvalue, which is also known as  $\lambda_n$  if we still have them in ascending order. That is all. The repelling and attracting LCSs will be discontinuous and orthogonal to each other at all points of intersection, and elliptic LCSs are continuous structures. Once the entirety of the strainlines are computed, all that is left to do is find the ones with the highest aver-

aged  $\lambda_n$ . Why is this eigenvalue so important?

If we remember back to Figure 1.1 in section 1.1.2, we can start to see why these large values indicate that fluid nearby is deforming (stretching and shrinking) at fast rates which is what makes these hidden structures so significant. As fluid parcels approach these objects, they tend to take the shape of these formation because of how significant they are. This is the backbone of LCS detection: finding areas that influence surrounding fluid parcels the most.

## **3.2 Worked Example**

Here, I work through the steps of finding a strain line for one point in our domain, and then we discuss how we would go about finding the LCSs. These steps were originally explained by Haller, and the  $3 \times 3$  matrix we use here was pulled from from the Green Strain page at *continuummechanics.org*.

#### (1) Defining the Flow Map

Our flow map is not defined explicitly here but instead we start with the matrix of the gradient of  $\vec{F}$ . We do note some key aspects below however:

$$\vec{x} \in U \in \mathbb{R}^3$$
  
 $t \in [t_0, t_1]$ 

# (2) Defining the Deformation Gradient

We start with an example deformation gradient matrix:

	1	0.495	0.5
$\nabla \vec{F} =$	-0.333	1	-0.247
	0.959	0	1.5

(3) Computing the Cauchy-Green Strain Tensor  $(\vec{C}(x_0))$ 

Recall that we compute  $\vec{C}(\vec{x}_0)$  as follows:

$$\vec{C}(\vec{x}_0) = [\nabla \vec{F}(\vec{x}_0)]^T [\nabla \vec{F}(\vec{x}_0)]$$

where T indicates a matrix transpose:

$$\vec{C}(\vec{x}_0) = \begin{bmatrix} 2.0306 & 1.1210 & 2.0208 \\ 1.1210 & 2.2450 & 1.5005 \\ 2.0208 & 1.5005 & 2.5610 \end{bmatrix}$$

#### (4) Eigenvalues, Eigenvectors, and Results

Now that we have our Cauchy-Green strain tensor, which is our most vital piece to this

puzzle, we can start to look at the results. Again, we will not show how to obtain eigenvalues and eigenvectors here since it should be known. First, we will get the column vector of eigenvalues( $\lambda$ ), from  $\vec{C}(\vec{x}_0)$ :

$$\lambda = \begin{bmatrix} 0.2390\\ 1.1676\\ 5.4300 \end{bmatrix}$$
$$\Rightarrow \lambda_1 = 0.2390, \qquad \lambda_2 = 1.1676, \qquad \lambda_3 = 5.4300$$

In MATLAB we can do the following to obtain the matrices V and D, where V is a  $3 \times 3$  matrix with eigenvectors columns as eigenvalues ( $\lambda$ ), D is a matrix with eigenvalues on the diagonal also put into a  $3 \times 3$  for matrix multiplication purposes. We will denote these eigenvectors  $\vec{\varepsilon}_n$  which correspond to  $\lambda_n$ :

$$[V,D] = eig(\vec{C}(\vec{x}_0))$$

$$\Rightarrow V = \begin{bmatrix} 0.7066 & 0.4354 & 0.5579 \\ 0.1260 & -0.8532 & 0.5062 \\ -0.6968 & 0.2874 & 0.6572 \end{bmatrix} D = \begin{bmatrix} 2.0306 & 0 & 0 \\ 0 & 1.1676 & 0 \\ 0 & 0 & 5.4300 \end{bmatrix}$$
$$\therefore \vec{\varepsilon_1} = \begin{bmatrix} 0.7066 \\ 0.1260 \\ -0.6968 \end{bmatrix}, \quad \vec{\varepsilon_2} = \begin{bmatrix} 0.4354 \\ -0.8532 \\ 0.2874 \end{bmatrix}, \quad \vec{\varepsilon_3} = \begin{bmatrix} 0.5579 \\ 0.5062 \\ 0.6572 \end{bmatrix}$$

as we know from Haller's paper, strain lines are tangent to  $\vec{\varepsilon}_1$  for this point in space. So, we have found a strain line for this point in space. If we were to compute every point in the space we are looking at, we would get an image similar to Figure 3.3 but without the green LCSs.

To obtain the LCSs, there is a final step that must be taken and it can only happen once all of the strain lines are computed for the region of interest. In Haller's paper, he states that LCS positions at  $t_0$  are given by strain lines with the locally highest averaged values of Cs largest  $\lambda$  which is  $\lambda_3$  in this example. Were we to compute the rest of the region of interest, we would simply look for where these  $\lambda_3$ s are high in value and have the computer flag them. This tells us which points are on the LCSs and then we can watch how these points get mapped under the flow map  $\vec{F}$  for a finite time and we have predictable fluid movement.

# **Chapter 4**

### **Climatology and LCSs**

In this chapter, I will bring everything I have learned together to run LCS detection on real data from ocean surface movement after the nuclear power plant disaster in Fukushima, Japan. First I will walk through the background of numerical weather prediction and give a basis for what the future of LCSs holds for fluid prediction in general. Next, I will walk through the processes of coding the Fukushima data collection and formatting it for Haller's code to accept. Then I will preform LCS detection and describe the results. Finally, I will do a discussion on LCSs and how we can think about them in the context of Earth's climate. What significance do we see with LCSs? What type of advancement could we see in the world of weather prediction? What makes LCSs different from other advancements in the field of climatology and prediction, such as the ability to measure radiant heat and albedo from clouds more accurately?

## 4.1 A Brief Background of Weather Prediction

Figure 4.1 is the best place to start when looking at weather and climate modeling over the years. If there is one solid answer to the question of "why do we need faster computers?" it is weather forecasts. People have been trying to predict the weather since the beginning of time but it wasn't really until the 1950s when it took off.

Some people did truly have a passion for wanting to use computers to forecast the weather before computers really had the power to do it. In 1922, Lewis Fry Richardson suggested that you could in fact calculate the movement of parcels of air and thus the movement of weather by just using the primitive equations of motion and integrating it forwards [26]. Richardson also made it apparent that one day computational devices would be able to help in this manor, and move faster than the weather itself to produce forecasts [27]. However, it was the shear daunting demeanor of the task at hand that turned people away. It wasn't that people didn't think that Richardson was right, they simply thought the approach was not applicable or useful because of the amount of computing power it would take.



**Figure 4.1** NCEP error results through time using the S1 score as laid out by Teweles and Wobus which measures the relative error in the horizontal gradient at a 500 hPa. This figure is a great representation of accuracy over time from Kalnay's book on Atmospheric Modeling [27].

Fast forward to the 1950s and computers are now fast enough and the idea is now a real possibility, so people get to work. The most important aspect to all of this is that as soon as computers start to get a go at forecasting, things go from fully observational to using observations as a tool to help computers predict what is coming next and this is a big deal for every model after this. One of the most foundational comments about modern weather modeling was said by J.G. Charney in his book *Dynamic Forecasting by Numerical Process* [28] where he states:

By starting with models incorporating only what it is thought to be the most important of the atmospheric influences, and by gradually bringing in others, one is able to proceed inductively and thereby to avoid the pitfalls inevitably encountered when a great many poorly understood factors are introduced all at once.

The key to this idea is that it set the stage of forecasting from then on out. The trade-off that everyone would understand from this point on is that you have to choose the most important parts of the model to calculate to keep the runtime reasonable enough to use the output in time. The history from here on is almost entirely dependent upon the speed of computers, coupled with the accurate collection of data and knowledge of weather phenomenon.

As opposed to going through that entire history step-by-step, I refer back to Kalnay and her book where she explains the trends in Figure 4.1 in four simple bullet points. She states

that "The improvement in skill of numerical weather prediction over the last 40 years...is due to four factors:

• the increased power of supercomputers, allowing much finer numerical resolution and fewer approximations in the operational atmospheric models;

• the improved representation of small-scale physical processes (clouds, precipitation, turbulent transfers of heat, moisture, momentum, and radiation) within the models;

- the use of more accurate methods of data assimilation, which result in improved initial conditions for the models; and
- the increased availability of data, especially satellite and aircraft data over the oceans and the Southern Hemisphere" [27]

It is exactly these areas and this history that will allow us to examine the true power behind LCSs and climate and weather modeling in section 4.3. Before we do that, we need to take a look at real data that this code can run and see what types of results and predictions we can make from using these LCSs on a real, tangible event: Fukushima.

## 4.2 Fukushima: An LCS Case Study

We have taken a look at LCSs, how they work, why they are important and even some simulated data to see how they are detected. Can we take a look at something real; something tangible and put this theory to work on a subject that had significance? We have seen papers using LCSs to look at oceanic flow, atmospheric flow, and pollutant transport and we have even seen Haller tackle the BP Deep Water Horizon oil spill to show how revealing LCSs are.

At 14:46 on Friday March 11, 2011 a 9.0 magnitude earthquake was recorded off the coast of Japan. The earthquake was centered 130 kilometers off shore from the city of Sendai on the main island of Japan, and while the earthquake did considerable damage, the country was hit with a devastating 15-meter tsunami caused by the earthquake as well. To get a feel of the devastation of the earthquake alone, which was a rare double quake that lasted about 3 minutes, the entire country of "Japan moved a few metres east and the local coastline subsided half a metre" [29]. After the tsunami, the death-toll was around 19,000 and over a million buildings were destroyed or partly collapsed.

Several of those one million buildings were located at the Fukushima Daiichi nuclear power plant. As designed, the running reactors (1-3) all shut down once the earthquake started and

none of the reactors were really effected. All six power supplies went offline because of the earthquake, and the backup diesel generators kicked in to keep the reactor cooling process going. The facility performed as it was designed to do, until 41 minutes later at 15:42 when the first tsunami smashed into the coastal facility and wreaked havoc, followed by a second wave 8 minutes later. This caused all sorts of problems with the seawater intakes for the cooling and more power loss and at 19:01 on Friday March 11 a Nuclear Emergency was declared. Over the next 24 hours, evacuations for the surrounding areas went from 2km all the way to 20km on March 12th once the Prime Minister had visited the plant [29]. Figure 4.2 shows a satellite shot of the power plant two years before the disaster and again in 2011 after the disaster.



**Figure 4.2** Satellite images from the Fukushima Daiichi plant in (left) 2009 and (right) 2011 right after the tsunami. These images come from the LA Times and I highly recommend looking the the rest of the before-after images to get an idea of the damage that was done [30].

With such a huge disaster, people started to scramble to get estimates done of the damage that was yet to come; the nuclear waste. Radioactive material getting into the air and the water were the two areas people were trying to predict. Where was this going to spread? How accurately could this be done? A lot of data started to come out surrounding the area so that studies could be done on having a protocol and tool to do this for future events.

Prants et al. [31] explicitly look at LCSs near Japan, but they were looking at ocean current in the Japan Basin which is on the west side of Japan. This paper has references to people in Haller's group and mentions LCSs in the writing. However, in an earlier paper, Prants, Budyansky, and Uleysky [32] look at radionucleotide movement specifically derived from Fukushima using Lagrangian methods, but not specifically LCS theory. In this paper, a different and less complex route was taken to find the value of LCSs to this scenario.

### 4.2.1 The Process

Haller's code contains demos, and each demo has data which was simulated, neatly packaged, and behaviorally complacent. The next step with using this code was to see if real data from real locations could be put into the LCS detector and output an image with neatly placed repelling, attracting, and elliptical LCSs. This entire process proved very difficult because the code was not written with this application in mind, the code authors were not responsive to questions, and the ability to debug required tremendous knowledge of the mathematics and project itself.

The first step the the process was to find an event to focus on. Fukushima was not only a perfect candidate for its obvious implications on problem solving, but it proved to be easy to find data in a format that very closely mimicked the data from the ocean demo. The backbone to this data was actually found as a demo in itself from a tool called NC-Toolbox [33]. NCToolbox was a solution to a problem we knew we were going to run into. A vast majority of oceanic and atmospheric data come in the form of gridded binary (GRIB) and GRIB2 formats. This tool allows MATLAB to deal with these easily. Inside of this NCToolbox was a contributed demo file called *fukushima.m*. This file went out to the Ecosystem Data Assembly Center (EDAC) [34] and obtained surface ocean velocities off the eastern coast of Japan near Fukushima.

The simulated data for the Ocean Dataset demo had a very specific format, and in order to change the code as little as possible, the *fukushima* code needed to be concatenated and outputted in that exact format. This took some nested for-loops and other checks to make sure there were no NaNs or Infs in the data where these values were found, we replaced them with zeros. An example of the data is show in figure 4.3 which represents one of the 105 total velocity fields looked at in this process.

The data analyzed is from March 11, 2011 (the day of the disaster) to March 21, 2011 and every time interval reflects 3-hour increments. One thing we need to make clear is that these times are in Zulu time which is 9 hours behind local Japan time which means that the earthquake occurred at 05:46 on March 11, 2011 Zulu time and the first wave occurred at 06:42 Zulu time.



**Figure 4.3** Output image from the code where the white is the main island of Japan and the colors indicate the velocities and red vectors are visible on the top of this imagery. The animated gif can be seen here. This timestep is at midnight Zulu time on March 11, 2011 - the day of the earthquake but before it happened.

Once the data had been gathered, concatenated, and structured exactly like its simulated counterpart, it was ready to be put into Haller's code. With just a few changes to variable names and other housekeeping items like latitude and longitude shifting and where to start the simulation, we ran the real data through and voilà; errors galore! The data was structured exactly like the input data, it was just larger and was actually taken from the real world and to this day I still am not sure why the initial errors I got were showing up. Through the process, it was evident that interpolating gridded data could cause problems if from one timestep to another, there was a drastic change in velocity. With enough debugging and sidestepping, we were able to get some results which we will look at in section 4.2.2.

For documentation purposes and usefulness to someone in the future, I am going to talk about the warnings and errors we ran into and how we were able to work around these problems in the long run. One difference between the approximated and real ocean data is that the original ocean dataset was run on a timespan of 100 to 130 which in theory is days of the year. Each time step in the data was 7 days however. That means that over the 30 days span, the code was only using 5 data inputs from the original code and was using the interpolant for everything else. In the data we collected, each timestep was 3 hours. The original runs of the code were doing timespans on the order of a couple of days and every time it would fail. In the end, the way to work around this was doing runs of timespans in the form of [t, t + 0.5]. This short of a time was able to run through without having issues with NaNs in the "eig" function built into MATLAB. This function was paramount in building the Cauchy-Green Strain Tensor; without it there was no way anything was going to be outputted. After that, a for-loop was just put in to run through each timestep one by one and output 105 images, one for each 3 hour period during the Fukushima event data.

#### 4.2.2 The Results

From what we can tell, the shortened timespans cut down the amount of information that could be analyzed by the detector, but the five most significant results of repelling LCSs have been shown above. The specific times of the events are at the top of each image. One of the most important things to show first in in figure 4.4 where we see the significant ocean currents that surround Japan. Now, even though the current is slightly south of our area of reference, we can assume the Kuroshio current is the activity we are seeing in our data in the yellow. It is important to know this because no matter what, this current has a large influence on the fluid around it.



**Figure 4.4** From top to bottom and left to right, these images are from (1) 3-11-11 18:00z, (2) 3-13-11 00:00z, (3) 3-14-11 18:00z, (4) 3-18-11 15:00z, and (5) 3-21-11 18:00z. To see these most significant LCSs in action with the animation you can click here to view it on YouTube.



Figure 4.5 A map of the significant ocean currents that surround Japan. The currents correspond to the following names: 1. Kuroshio 2. Kuroshio extension 3. Kuroshio countercurrent 4. Tsushima Current 5. Tsugaru Current 6. Sōya Current 7. Oyashio 8. Liman Current. The most significant to us is number 1, the Kuroshio currents. This is the current we see highlighted in yellow in all of our data and it has the most movement [35]. The red bounding box in this figure is the extent of the data that was collected and the red dot is approximately where the Daiichi plant is.

What do we see from this data? It is clear that the Fukushima data did not produce as clear of an outcome as the original simulated data did, but it still gave us information we did not have before about the surrounding fluid flows. Another observation we can make is how often these LCS change. We see complete change in just a single timestep (3 hours in our case) and that is not surprising. The ocean has a plethora of interacting parts and just watching the surface can only give us a hint as to what is going on. The big thing with these results is that they worked; that what the biggest thing we wanted to happen. We were able to take a free tool that is designed for learning and run real oceanic data through it. Data that is messy, unorganized, limited and full of surprises, but most importantly real.

It is foolish to think we would be able to do any sort of pollutant transport analysis from this small amount of data but the point still remains: with a small amount of effort, we can run real data through these tools and get real results to help solve a problem. LCSs are nothing

more than another tool in our belt to aid in question answering and this takes us into the discussion of how LCSs can really help the future of climate and weather modeling.

## 4.3 Discussion: LCSs and Climatology

At this point, we have beaten LCSs to a pulp. Almost every aspect of LCSs has been touched upon, and we have already talked about the motivation behind using these things, but there is one specific side of LCSs that allow them to have a profound impact on climate studies. From most of the paper and references looked at in this thesis, it is easy to see that LCSs are already used in projects involving fluid data and a lot of those projects have to do with climate and that is no surprise. We know that the atmosphere is the most studied fluid (computationally). We are always trying to go that extra mile and get shove that extra variable in the equation to further this computation. Also, as stated in section 4.1, one of the main reasons for the advancement in weather modeling is computational power. So this begs the question of why LCSs seem to be different? Why are they not like any other advancement in the model?

Model advancements, besides shear computational power, come from more complex modeling. The more variables we can input into the model, the more accurate our model should be (in theory). It brings us back to J.G. Charney's quote on starting with the most influential parts of the model and then adding on as information becomes available. However, LCSs are not another piece of information to add to model. LCSs allow for a different way of categorizing fluid flow. They fundamentally change the way a model is analyzed. Far different than added knowledge of atmospheric conditions such as how clouds affect longwave radiation or how supercells change once they come into contact with certain types of terrain, LCSs allow an approach to modeling based on fundamentals and not additions.

LCSs change everything about how we model climate. Even without changing current weather and climate models, if you ran LCS detections at the same time, they could be used by meteorologists and climatologists as another tool for prediction. Plain and simple, LCSs describe which parts of a fluid are most influential to particles around them. For someone studying things like the El Nino Southern Oscillation (ENSO) or the polar vortex, using LCSs seems like a must for all analyses now. The very fact that LCS theory has been introduced and proven to be able to predict fluid movement accurately means it is a subject that has to be implemented and the first model that does so on a large scale will be dominant.

The world of weather research is very different than a lot of other fields because it has an underlying culture of sharing. Yes it is true that models often "compete" with each other for the most accurate results, but the whole idea behind having multiple models is to get an idea of the uncertainty associated with the path and strength of an oncoming storm. The same principle applies to hurricane tracking. The cone of uncertainty is a visual represen-

tation of where the eye of a hurricane could go and decreasing the size of that cone saves hundreds of lives and millions of dollars in time and resources. That cone is put together by using all of the model predictions. The shape is a cone because all of the models say pretty much the same thing for short lead times.

Implementation of LCS detection into fluid models would allow for a world of change in every discipline of climatology imaginable; from predicting the Index Cycle change on a mesoscale to being able to tell how much precipitation will end up on your front lawn on a microscale. LCSs are the way of the future for fluid prediction and implementing them in every realm possible will allow us to be that much further ahead of the problems that stare us in the face everyday. Imagine being able to know exactly where the oil from BP Deep Water Horizon was going to go the moment it happened or being able to know a deep freeze was coming to kill your crops two weeks before it happened. LCSs are doing to climate and weather modeling what railroads did for the shipping industry in the late 19th century, what TVs did for the advertising world, and what the iPhone did for the phone industry; they are revolutionizing the way we look at fluid.

# References

- [1] George Haller. Lagrangian Coherent Structures. *Annual Review of Fluid Mechanics*, 47(1):137–162, January 2015.
- [2] Thomas Peacock and George Haller. Lagrangian coherent structures: The hidden skeleton of fluid flows. *Physics Today*, 66(2):41–47, February 2013.
- [3] A. S. Sharma and B. J. McKeon. On coherent structure in wall turbulence. *Journal of Fluid Mechanics*, 728:196–238, August 2013.
- [4] S K Robinson. Coherent Motions in the Turbulent Boundary Layer. *Annual Review* of *Fluid Mechanics*, 23(1):601–639, January 1991.
- [5] Landsat Image Gallery Two Views of Von Kármán Vortices.
- [6] Kathleen T. Alligood, Tim D. Sauer, and James A. Yorke. *Chaos: an Introduction to Dynamical Systems*. Textbooks in mathematical sciences. Springer, New York, NY, corr. 3. print edition, 2000. OCLC: 248606567.
- [7] Mohammad Farazmand and George Haller. Computing Lagrangian coherent structures from their variational theory. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(1):013128, March 2012.
- [8] The skeleton of water. *The Economist*, November 2009.
- [9] Hong Kong with Forests of Skyscrapers | 2008 | JAXA Earth Observation Research Center (EORC).
- [10] P.N. Sun, A. Colagrossi, S. Marrone, and A.M. Zhang. Detection of Lagrangian Coherent Structures in the SPH framework. *Computer Methods in Applied Mechanics and Engineering*, 305:849–868, June 2016.
- [11] Mattia Serra, Pratik Sathe, Francisco Beron-Vera, and George Haller. Uncovering the Edge of the Polar Vortex. *Journal of the Atmospheric Sciences*, 74(11):3871–3885, November 2017.

- [12] Wenbo Tang, Manikandan Mathur, George Haller, Douglas C. Hahn, and Frank H. Ruggiero. Lagrangian Coherent Structures near a Subtropical Jet Stream. *Journal of the Atmospheric Sciences*, 67(7):2307–2319, July 2010.
- [13] Matt K Gough, Ad Reniers, M. Josefina Olascoaga, Brian K Haus, Jamie MacMahan, Jeff Paduan, and Chris Halle. Lagrangian Coherent Structures in a coastal upwelling environment. *Continental Shelf Research*, 128:36–50, October 2016.
- [14] Cheryl S. Harrison and Gary A. Glatzmaier. Lagrangian coherent structures in the California Current System – sensitivities and limitations. *Geophysical & Astrophysi*cal Fluid Dynamics, 106(1):22–44, February 2012.
- [15] F. J. Beron-Vera, M. J. Olascoaga, and G. J. Goni. Oceanic mesoscale eddies as revealed by Lagrangian coherent structures: OCEANIC MESOSCALE EDDIES AS REVEALED BY LCSS. *Geophysical Research Letters*, 35(12):n/a–n/a, June 2008.
- [16] M. J. Olascoaga and G. Haller. Forecasting sudden changes in environmental pollution patterns. *Proceedings of the National Academy of Sciences*, 109(13):4738–4743, March 2012.
- [17] Rui Qi and Biao Huang. Numerical investigation of transport mechanism in fourbody problem using Lagrangian coherent structure. *Astrophysics and Space Science*, 361(1), January 2016.
- [18] Rui Qi and Shi Jie Xu. Applications of lagrangian coherent structures to expression of invariant manifolds in astrodynamics. *Astrophysics and Space Science*, 351(1):125– 133, May 2014.
- [19] K. Onu, F. Huhn, and G. Haller. LCS Tool: A computational platform for Lagrangian coherent structures. *Journal of Computational Science*, 7:26–36, March 2015.
- [20] National Oceanic and Atmospheric Administration US Department of Commerce. What is a gyre?
- [21] H. D. I. Abarbanel and W. R. Young, editors. *General circulation of the ocean*. Topics in atmospheric and oceanic sciences. Springer-Verlag, New York, 1987.
- [22] W. G. Bickley D.Sc. LXXIII. The plane jet. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 23(156):727–731, April 1937.
- [23] K. Onu, F. Huhn, and G. Haller. LCS Tool: A computational platform for Lagrangian coherent structures. *Journal of Computational Science*, 7:26–36, March 2015.
- [24] Diego del-Castillo-Negrete and P. J. Morrison. Chaotic transport by Rossby waves in shear flow. *Physics of Fluids A: Fluid Dynamics*, 5(4):948–965, April 1993.

- [25] G. Haller and F. J. Beron-Vera. Coherent Lagrangian vortices: the black holes of turbulence. *Journal of Fluid Mechanics*, 731, September 2013.
- [26] Lewis Fry Richardson. Weather prediction by numerical process. By Lewis F. Richardson. Cambridge (University Press), 1922. 4. Pp. xii + 236. 30s.net. *Quarterly Journal of the Royal Meteorological Society*, 48(203):282–284, August 2007.
- [27] Eugenia Kalnay. *Atmospheric modeling, data assimilation, and predictability*. Cambridge University Press, New York, 2003.
- [28] J. G. Charney. Dynamic Forecasting by Numerical Process. In Thomas F. Malone, editor, *Compendium of Meteorology*, pages 470–482. American Meteorological Society, Boston, MA, 1951.
- [29] Fukushima Accident World Nuclear Association.
- [30] Satellite images of Japan, before and after the earthquake.
- [31] S.V. Prants, V.I. Ponomarev, M.V. Budyansky, M.Yu. Uleysky, and P.A. Fayman. Lagrangian analysis of the vertical structure of eddies simulated in the Japan Basin of the Japan/East Sea. *Ocean Modelling*, 86:128–140, February 2015.
- [32] S. V. Prants, M. V. Budyansky, and M. Yu. Uleysky. Lagrangian study of surface transport in the Kuroshio Extension area based on simulation of propagation of Fukushimaderived radionuclides. *Nonlinear Processes in Geophysics*, 21(1):279–289, February 2014.
- [33] NOAA's National Weather Service. Data Analysis using MATLAB and NCTOOL-BOX.
- [34] Ecosystem Data Assembly Center.
- [35] Wikipedia- Japan's Ocean Currents.