Abstract

For general aperiodic fluid flows, coherent structures help organize the dynamics. The prevalence of such flows in nature and industry has motivated many successful techniques for defining and detecting coherent structures. However, these approaches often require very fine trajectory data to reconstruct velocity fields. Instead, we use topological techniques to detect coherent trajectory sets in relatively sparse two-dimensional fluid advection problems. More specifically, we use a homotopy-based algorithm, the ensemble-based topological entropy calculation (E-tec), which evolves fluid material curves forward in time as minimal length bands stretched about the moving data points. These bands are represented as the weighted edges of a triangulation, which allows us to analyze flows using graph theory. In this way, highly connected components of appropriately constructed graphs can be used to partition the fluid particles into coherent trajectory sets.

Coherent Structure Detection using Topological Tools and a Graph Theoretic Approach

Caledonia Wilson

Supervised by Spencer Smith, Assistant Professor of Physics

Submitted to the Department of Physics,

Mount Holyoke College

in partial fulfillment of the requirements for the degree of

Bachelor of Arts in Mathematics

May 9th, 2019

Acknowledgements

I originally began this research as an excuse to avoid taking statistics. Over the past two years, I have learned so much not only about fluid dynamics, but also what it means to do research motivated by curiosity.

I am grateful for the support of Professor Spencer Smith as he tirelessly listened to my half-formed babblings about graphs and helped me formulate that goop into a mathematically rigorous method of problem solving. Whether he was providing conceptual insights into complex physical systems or helping me debug code, his excitement for this research was infectious. I am also grateful to Professor Mark Peterson for his enthusiasm in serving on my committee.

I want to thank Professor Jessica Sidman for acting as my academic advisor for the past four years. Her support buoyed me when I needed it, including her assurance that it was alright if I didn't finish this thesis. I thank her for being in my corner throughout my entire time at Mount Holyoke (and for telling me it was okay to drop my statistics class).

Thanks to my friends, for listening to me complain and gush about this research in (fairly) equal measure. I thank my mother for pretending to care about fluid dynamics, and listening to me talk about it anyways.

And I thank my dad, who had the uncanny ability to read books on countless topics while he watched his 'stories' on television. He showed me that learning can be a lifelong habit. From awful sci-fi original movies to Civil War trivia to the poetry of W.H. Auden, I carry so many random bits of his curiosity with me.

Finally, thanks to you, reader, for muddling through these concepts alongside me.

Contents

List of Figures

1 Introduction

Coherent structures are certain features of a fluid flow, and by studying them, we can better understand the mixing of elements of the flow, as well as mechanisms for scalar transport within the fluid. We begin by giving a qualitative description of these structures and examples of their significance in real-world flows, then situate our research in terms of the Eulerian and the Lagrangian viewpoints. We give a general context for a few prevalent coherent structure detection methods in the literature, such as the Finite-Time Lyapunov Exponent (FTLE) field and braid groups. Then we introduce a method for using the Ensemble-Based Topological Entropy Calculation (E-tec) algorithm to detect coherent structures via graph theoretic methods, with precise definitions for ideas that figured crucially in the work. Finally, we take the reader through examples using E-tec with randomly generated sets of particles within the time-dependent double gyre system, and reflect on the strengths and shortcomings of this method of detection.

1.1 Why Coherent Structures?

Fluid flows are ubiquitous and mysterious. In studying their behavior, the field of fluid dynamics has widespread applications as well as a long legacy of theoretical knowledge. A fair amount of the research in the field seeks to identify dynamic structures in a fluid in order to better understand the fluid's observed behavior over a set amount of time, whether by investigat-

ing at the scale of the overall bulk flow or at the scale of molecular diffusion. Detecting coherent structures is one such method for understanding fluid flows.

Informally, coherent structures are regions of a fluid that do not mix considerably with the rest of the fluid over a fixed time. It follows that the boundaries of these structures prevent particles from other regions from passing into a coherent structure, such that the structures also act as barriers to transport. These regions of fluid are deemed coherent due to their relative stability when compared to other regions of fluid over the same time frame [17]. That is, the structures last over significantly longer time-scales than it takes for nearby fluid elements to diverge.

Coherent structure detection in fluid flows has widespread applications to real-world problems, as they evolve in time. One example prevalent in the literature is the Deepwater Horizon oil spill. A BP oil rig of that name was drilling in the Gulf of Mexico when it exploded on April 20th, 2010, releasing four million barrels of oil into the gulf and leading to an estimated 8.8 billion dollars in damages to the natural resources of the surrounding ecosystem [4].

In seeking to mitigate damage from the 87-day spill, it was difficult to predict the behavior of the oil on the surface. By May 17th [12], a long tendril of oil, dubbed the 'tiger tail', was observed trailing southeastward from the southern portion of the spill (see Figure 1). This tiger tail had markedly different qualitative behavior than the rest of the oil spreading in the gulf. By analyzing the ways Lagrangian coherent structures act as barriers to transport, Haller and Peacock attributed this behavior to underlying currents in the gulf [12].

Another example of perplexing structures detectable on the oceans' surfaces are the Agulhas Rings. Produced by the Agulhas current off the

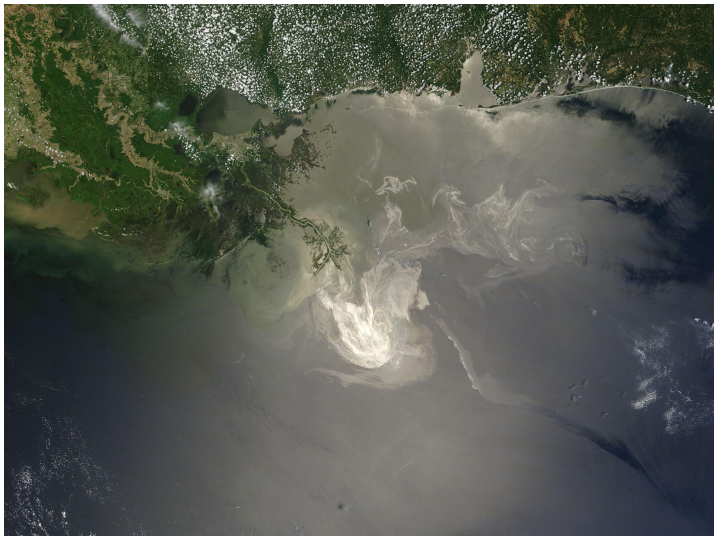


Figure 1: A photograph of the Deepwater Horizon oil spill, taken May 24th, 2010. The 'tiger tail' can be seen as a grey sheen trailing towards the lower right-hand corner of the image. [3]

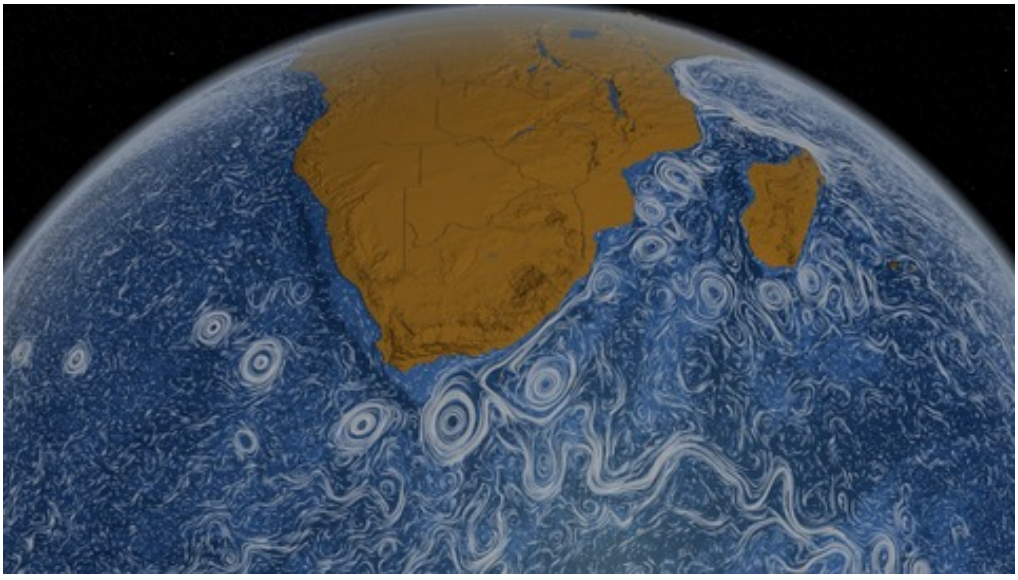


Figure 2: The Agulhas Current and subsequent vortices are tracked as they form of the southern coast of Africa. [9]

southernmost tip of Africa, the Agulhas Rings refer to the retroflection eddies that are shed as the current flows westward [13]. These eddies form rings that are larger than rings typical of such a flow, and contribute to global thermohaline circulation: the movement of seawater based on its temperature and salt content. Analyzing these rings through the lens of coherent structures as barriers to transport could explain why such large rings often remain intact for such a long time (often over two years) compared with nearby regions of fluid [21].

1.1.1 Eulerian and Lagrangian Viewpoints

There are two basic overarching frameworks for analyzing fluid flows. The Eulerian framework considers a domain (which for our purposes, we can define as fixed and two-dimensional). At every fixed point in this domain, we assign a velocity vector (which could be time-dependent) to form a velocity field over the domain. The knowledge of this velocity field constitutes the Eulerian viewpoint. Being able to successfully capture the velocity field of a flow over a time interval amounts to having perfect knowledge of the system, in that the trajectories of individual particles can be extrapolated from this velocity field (as follows in the Lagrangian viewpoint).

The Eulerian framework follows naturally from first principles of physics. Namely, the Navier-Stokes equation that governs fluid flows has solutions in the form of velocity vector fields:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \nu \nabla^2 \mathbf{u} - \nabla P + \mathbf{f}$$

This gives the Navier-Stokes equation (for incompressible flow). This is a partial differential equation, the solution to which is the Eulerian velocity field. We can think of this equation as a continuous version of Newton's

Second Law, $F = ma$. The terms on the left hand side of this equation are acceleration terms, whereas the right hand side includes the force terms.

One major drawback of the Eulerian framework is that it lacks frame-invariance. This means that a change in reference frame could lead to a fundamentally different interpretation of a fluid's structure over time. If we develop methods for analyzing fluid flows within the Eulerian framework, such methods necessarily involve interpreting the behavior of an underlying velocity field, but such velocity fields are not generally frame-invariant.

A change in reference frame has consequences for coherent structure detection. We could have a velocity field with closed streamlines in one reference frame, but those streamlines might not be closed in another reference frame. This means that a barrier to transport in one reference frame might not appear as such in another reference frame, so that the detection of coherent structures would depend on the reference frame we used to observe the flow.

To avoid the potential pitfalls of different reference frames, a Lagrangian viewpoint considers the motion of fluid particles relative to each other. Rather than considering a velocity field fixed over some domain, the Lagrangian viewpoint traces the trajectories of points within the domain as they unfold in time. This is a natural framework to use for coherent structure detection because most experimental data is collected using tracer particles. Experimentalists then analyze the trajectories of these tracers as indicators of the fluid's behavior. The Lagrangian framework is also frame-invariant, such that the observed structures do not depend on the reference frame of data collection, for example. We use the Lagrangian framework to analyze the flows of passively-advected particles, or particles that, as a property of their densities, move with the local velocity field of the fluid. The idea of coherent structures as barriers to transport is rigorously de-

fined within the Lagrangian framework, as exemplified by the Finite-Time Lyapunov Exponent (FTLE) field in the next section.

1.1.2 The Finite-Time Lyapunov Exponent Field

The Finite-Time Lyapunov Exponent (FTLE) field is a geometric approach and one of the primary methods for coherent structure detection. The FTLE field is a scalar field generated by considering a sufficiently small radius around a particle, p, and tracking the trajectory of p, as well as the trajectories of all other particles within that radius, over a set time interval [16].

The FTLE value at p's initial location is given by the maximum exponential stretching rate between p and the other points in the radius. Note that the FTLE value depends on the length of the time interval that we choose. If two fluid particles start very close together and become separated after a set time interval, the FTLE values will be high near the initial location of these particles. If the FTLE values are high along a material line of fluid, very few other fluid particles can cross this line, making it a barrier to transport [12].

As follows from the definitions below, the FTLE field relies on Eulerian computational techniques to measure a Lagrangian conception of stretching in the fluid. We begin by bridging the gap between the Eulerian and Lagrangian frameworks by defining a dynamical system where the velocities of particle trajectories can be generated by finding the value of the velocity field at a specific time and position [16]:

$$\dot{\vec{x}}(t; t_0, x_0) = \vec{v}(\vec{x}(t; t_0, \vec{x_0}), t),$$

$$\vec{x}(t_0; t_0, \vec{x_0}) = \vec{x_0}$$

The vector map $\vec{v}$ need not be continuous, but if $\vec{v}$ is continuous it produces smooth solutions. By specifying a final time, t, we can define a flow map for a finite-time dynamical system over this time interval,

$$\phi_{t_0}^{t} : D \to D \text{ where } \vec{x_0} \mapsto \phi_{t_0}^{t}(\vec{x_0}) = \vec{x}(t; t_0, \vec{x_0})$$

This map takes a point (expressed as a vector quantity) in the domain and maps it to the location of that particle at time t. Such a flow map satisfies the properties:

$$\phi_{t_0}^{t_0}(\vec{x}) = \vec{x}$$

$$\phi_{t_0}^{t+s}(\vec{x}) = \phi_{s}^{t+s}(\phi_{t_0}^{s}(\vec{x})) = \phi_{t}^{t+s}(\phi_{t_0}^{t}(\vec{x}))$$

The first property entails that a point is mapped to itself at the initial time, while the second property entails that such a map is well-defined with respect to different time intervals. This map ensures that given an initial point, we can map it forward in time.

Using this map, we can take a point and map it forward by a time T:

$$\vec{x} \mapsto \phi_{t_0}^{t_0+T}(\vec{x})$$

Now consider a point $\vec{y}$ arbitrarily close to $\vec{x}$, such that $\vec{y} = \vec{x} + \delta\vec{x}(t_0)$ for some arbitrarily small, positive value δ.

Then we can apply the flow map to $\vec{y}$:

$$\delta\vec{x}(t_0 + T) = \phi_{t_0}^{t_0+T}(\vec{y}) - \phi_{t_0}^{t_0+T}(\vec{x}) = \frac{d\phi_{t_0}^{t_0+T}(\vec{x})}{d\vec{x}}\delta\vec{x}(t_0) + O(||\delta\vec{x}(t_0)||^2)$$

This equation describes the vector difference between the final position

of $\vec{x}$ and $\vec{y}$ as a function of the rate of change of the final position of $\vec{x}$ with respect to displacement of $\vec{x}$ initially (which gives us a 2×2 matrix), multiplied by a vector representing a small displacement of $\vec{x}$, denoted $\delta\vec{x}(t_0)$. The final term in this sum refers to higher order terms in δ that can be ignored by picking δ sufficiently small. By picking δ sufficiently small, then, we can condense the information about the flow in a small radius down to a linear approximation: the product of a matrix and the vector displacement we began with.

Within this set up, we can manipulate this matrix to produce the Cauchy-Green deformation tensor, and then take the square root of the maximum eigenvalues of this tensor. The log of this value divided by the time interval is the FTLE value.

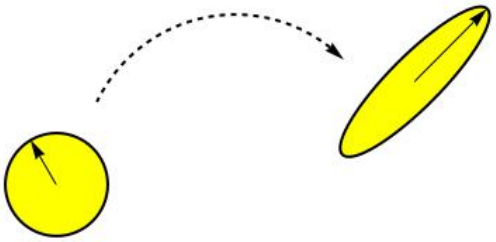


Figure 3: A cartoon of the way an initial disk of radius δ stretches out describes the FTLE value of a particle for a given initial time and position, as well as a given integration time.

The FTLE field that results is a scalar field. Lagrangian coherent structures are defined as ridges of this scalar field, along which the FTLE has high values [18]. Since stretching occurs along these lines, there can be little to no stretching across a ridge-line, making them barriers to transport and hence Lagrangian coherent structures. In this way, the FTLE field picks out the "regions of greatest relative stretching of material elements" [1].

One drawback of this method is that its computations require extensive data in order to interpolate the velocity field and create a smooth flow

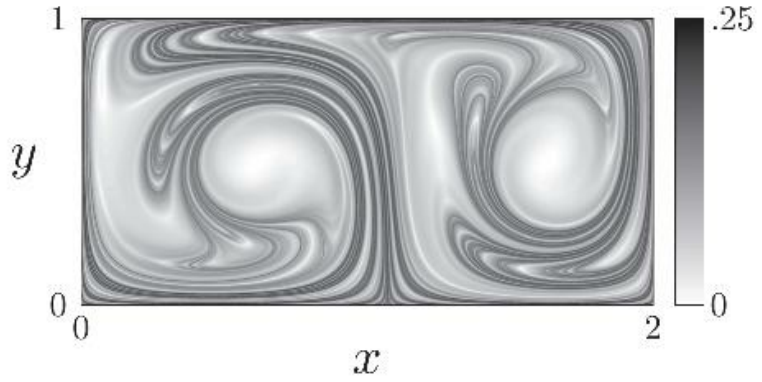


Figure 4: The FTLE field for the time-dependent double gyre (described in Section 1.2.2) with parameters A=.1, ϵ= 0.1, and $\omega = \pi/5$ [1].

map. Due to this dependence on dense data, "the fine-scale structure of a system may not appear without a high point density" while using the FTLE method [15]. Other approaches, such as probabilistic methods, also rely on large amounts of data [5].

1.1.3 A case for sparse data

A phenomenon like an oil spill occurs unexpectedly, usually in the open ocean, where scientists might have to rely on sparsely distributed indicators of fluid flows, such as floating buoys. Even fluid dynamics research in a lab setting often relies on tracer particles to analyze local behavior of a fluid in order to understand its larger structure. Since most experimental data collected is sparse data, a sparse data approach to coherent structure detection could greatly enhance our predictive capabilities.

However, many current approaches to coherent structure detection, like the FTLE field, require large amounts of data, or essentially 'perfect knowledge' of the systems being analyzed, by attempting to 'back out' the underlying vector field from collected trajectory data. The collection of such data is often impractical and could hinder the predictive capabilities of coherent structure detection methods. Some experimental methods to de-

tect structures rely on infusing the collected data with 'ghost' particles, whose trajectories are integrated from the inferred underlying vector field, in order to have a dense enough set of data to create an FTLE field. Not surprisingly, this method only produces an accurate FTLE field for particularly dense sets of passively advected points. The notion of the density of a data set depends on the flow under consideration. For instance, in a laminar flow, the density of particles needed to capture the structure of the underlying flow is lower than it would be for turbulent flow.

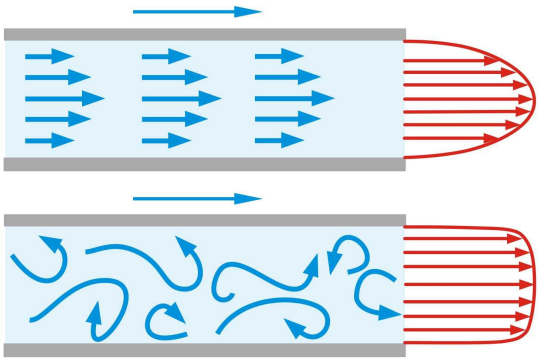


Figure 5: A comparison of laminar (top) and turbulent (bottom) flow. If we seeded tracer particles in each flow, we could capture the behavior of laminar flow more accurately with fewer tracers than we would need to capture the behavior of the turbulent flow [7].

A sparse data approach to coherent structure detection allows us to harness the benefits of the Lagrangian approach by sidestepping concerns of local "smooth structure of the flow," and instead focusing "directly on the global information" [20]. This way, we can build on methods like the FTLE field without needing to rely on extensive data collection, focusing on searching for global topological properties of the flow instead of the local behavior of the velocity field at a given point in the domain.

Methods for coherent structure detection in a sparse data framework can be split into two camps: clustering analysis and topological methods. Clustering analysis is a sparse data approach that relies on defining

a distance metric on trajectories over a set time interval and finding for each particle a membership value "describing the likelihood that a trajectory belongs to a cluster"[6]. This type of analysis is often referred to as 'fuzzy' because the membership values depend on a parameter governing the amount of overlap that is acceptable between individual clusters [8].

Within a sparse data framework, our research is situated in topological methods. Existing topological methods in coherent structure literature explore braid theory and finite-time braiding exponents (FTBE) [2]. In such research, particle trajectories are conceptualized as strands of a braid in space time. More formally, two-dimensional trajectories in space are mapped to three dimensions by considering time as the third axis, often depicted as increasing vertically upwards.

These physical braids mapped out in time are representations of the ways particles intertwine on the surface of a fluid, but are also isomorphic to algebraic braids. An algebraic braid group on a set of n strands, B_n, is defined on a set of generators σ_1, .., σ_{n-1}, and the relations between them, where each generator σ_i represents "the clockwise interchange of the ith and $(i+1)th$ strands" (see Figure 6) [20].

Once a braid is defined, and the specific procedures for mapping experimental trajectory data to algebraic braids are established, the fluid flow is analyzed using **topological entropy**, where a positive value of entropy corresponds to chaotic mixing. In this way, the topological entropy of a system gives a global, quantifiable measure of the amount of chaos in a system. A formal definition of topological entropy is given in Section 2.1.

Since this algebraic braid encodes the relative motion of the particles, and braid theory is a well-studied area of mathematics, coherent structure detection using braid theory is a popular sparse data technique. This method wraps taut loops around a braid and measures the stretching of

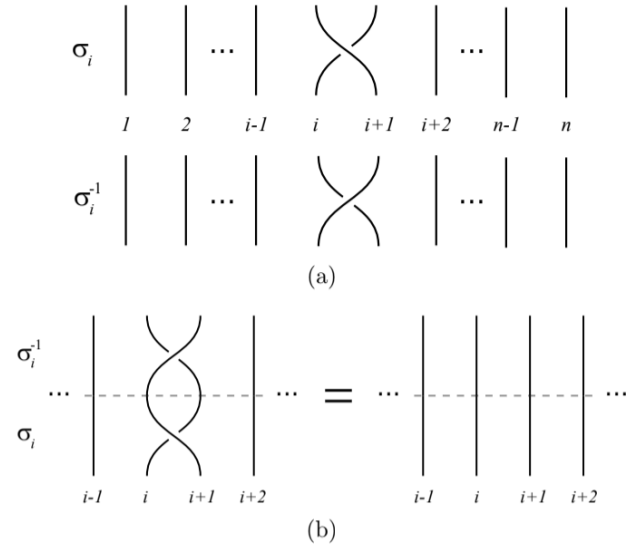


Figure 6: (a) shows the crossing of strands i and $i+1$ due to the generators σ_i and σ_i^{-1}, while (b) shows that a braid can be untangled by applying the inverse generator of the preceding generator that acted on the braid [20].

this loop over time. If particles are in a coherent structure, and therefore not mixing with other regions, then such loops do not stretch exponentially. Measuring the stretching of these loops is known as the Finite-Time-Braiding-Exponent (FTBE) method [2].

One drawback of the braid theory approach to coherent structure detection is that it scales quadratically: if we have N particles in our system, we would need on the order of N^2 algebraic generators. We seek to find a sparse data, topological method to detect these structures that builds on the FTBE method, but is less computationally expensive [15].

In this paper, we focus on detecting coherent structures in generic two-dimensional time-dependent advection problems. We define a coherent structure as a region of fluid that does not significantly mix with the rest of the fluid as it is advected. We seek a partition of the trajectories into subsets that consist of coherent structures and subsets that represent the chaotic mixing region. In order to find these sets, we build on topics from

topology, such as topological entropy and persistent homology, in conjunction with graph theory. We use an algorithm called Ensemble-based topological entropy calculation (E-tec) to analyze sparsely distributed trajectories of the time-dependent double gyre system [15].

Though we have thus far considered examples of coherent structures as entities present on the two-dimensional surface of a fluid (or flows at the interface of two fluids), coherent structure detection methods should not be limited to two dimensions. A three-dimensional approach to coherent structure detection could yield valuable information on the behavior of fluid flows. By finding a sparse data approach to coherent structure detection in two-dimensional flows, a rigorous framework for three-dimensional analysis can be developed which could be significantly less computationally expensive than methods relying on dense data sets of three dimensional flows. Such an expansion into three dimensions is not possible using a braid-theory approach.

1.2 The Double Gyre

1.2.1 The Time-Independent Double Gyre

The double gyre is a popular example in fluid dynamics literature, particularly in coherent structure detection, and so it serves as a useful model for developing a new technique. The double gyre is a periodic system that consists of two counter-rotating vortices.

We first consider the time-independent double gyre. Within this system, the underlying velocity field remains constant for all time [16]. The flow is described by the stream function,

$$\Psi(x, y) = sin(\pi x)sin(\pi y), \text{ where}$$

$$\dot{x} = -\frac{\partial \Psi}{\partial y}$$

$$\dot{y} = \frac{\partial \Psi}{\partial x}$$

give the x and y components of the velocity field used to advect particles.

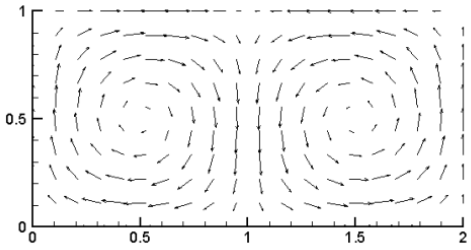
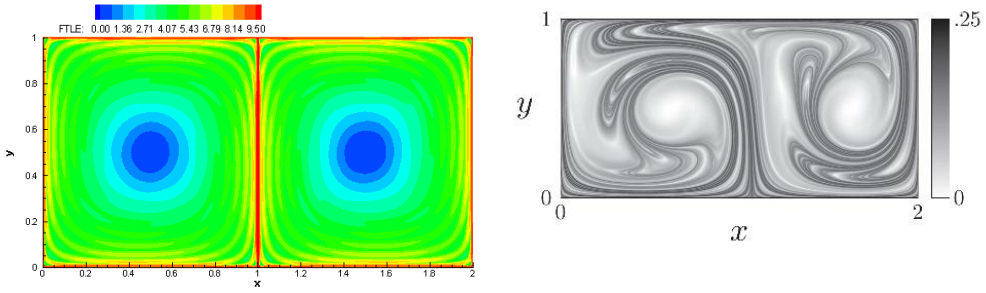


Figure 7: Velocity field for the time-independent double gyre [16]



(a) (b)

Figure 8: (a) gives the FTLE field for the time-independent double gyre (with an integration time of T=17). The color key at the top of the image indicates that the FTLE values are lowest in the two blue, round regions, indicating these regions could be coherent structure candidates [16]. In (b), we have the FTLE field for the time-dependent double gyre with the parameters given in Figure 4 [1].

Shadden uses the FTLE method to analyze this time-independent double gyre (see Figure 8a). It is important to note that since the velocity field is time-independent, the FTLE scalar field does not vary with a given initial time, but does vary with integration time [16]. This means that if we calculate the FTLE field for trajectories that are only "running" for k seconds, we may get a different FTLE field than if we let the trajectories

run for k+t seconds. The fact that the velocity field is time-independent means that regardless of the initial time we choose to start the integration, if we integrate over the same length of time, we produce the same FTLE field.

This is example shows that even in time-independent systems, the detection of coherent structures is still inherently time-dependent as a measure of the relative stability of different fluidic regions. If we choose a time frame that is too short for the system at hand, two particles could appear to stay close together, even though their trajectories may diverge over a longer time interval. Choosing a time interval too long could lead to the collapse of the coherent structures altogether, in that the FTLE field would be measuring global (instead of local) behavior [17].

1.2.2 Introducing Chaos: the Time-Dependent Double Gyre

Building on this time-independent system of two rotating vortices, we add a parameter to introduce time-dependence into the system. This parameter, ϵ, is approximately the amplitude of motion for a vertical line oscillating from left to right, where $2\pi/\omega$ gives the period of this oscillation. This oscillation causes the vortices to expand and contract. The equations governing the flow of the particles are given below. We consider the stream function:

$$\Psi(x, y, t) = Asin(\pi f(x, t))sin(\pi y), \text{ where}$$

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

$$a(t) = \epsilon sin(\omega t),$$

$$b(t) = 1 - 2\epsilon sin(\omega t)$$

This yields the velocity field equations:

$$u = -\frac{\partial \Psi}{\partial y} = -\pi A sin(\pi f(x)) cos(\pi y)$$

$$v = \frac{\partial \Psi}{\partial x} = \pi A cos(\pi f(x)) sin(\pi y) * (df/dx)$$

where A determines the magnitude of these velocity vectors, and ϵ is approximately the amplitude of motion for the vertical line oscillating from left to right, where $2\pi/\omega$ gives the period of this oscillation (as described above).

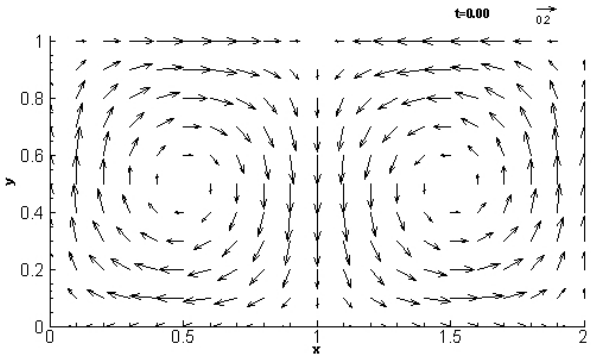


Figure 9: Velocity field for the time-dependent double gyre, for values A=1, ω= $\pi/5$, and ϵ= .25 at time t=0. This field is initially practically identical to the time-independent vector fields in Figure 7 [16].

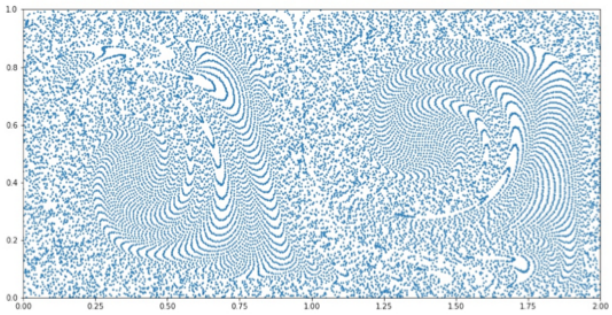


Figure 10: The time-dependent double gyre advects particles in a $[0, 2] \times [0, 1]$ domain. This is a still from a movie in which a dense grid of particles were advected; areas where lines of particles are visible (appearing almost like contour lines in this image) indicate regions of fluid that mix less than areas with particles that appear fuzzier in this image.

The ϵ parameter is significant because it introduces chaos into the system. We can conceptualize chaos as an acute sensitivity to initial conditions of a system. Suppose we have two identical, periodic velocity fields, $\vec{F}1$ and $\vec{F}2$. We drop particle A at some point on $\vec{F}1$, and then carefully place particle B on $\vec{F}2$ an infinitesimal distance from A's initial coordinates on $\vec{F}1$. If the system is chaotic, after some time interval, the trajectories of particles A and B will diverge exponentially, even though they began on identical velocity fields infinitesimally close. Therefore, a consequence of chaos is that given the initial position of a particle, chaos acts as a barrier to predicting its future position.

We can experience this chaos in a fluid flow by noting that if two particles (with initial positions arbitrarily close together) are in a chaotic regime, the distance between their trajectories stretches exponentially as the particles are advected. Note that even though the double gyre system with the ϵ parameter is periodic, meaning that its underlying velocity field changes periodically over time, this velocity field still gives rise to chaotic mixing.

The double gyre appears in the literature as a standard system for testing the viability of different coherent structure detection methods [1]. We analyze a sparse set of points seeded randomly in a two-dimensional $[0, 2] \times [0, 1]$ domain, and find the trajectories of these points for the double gyre system. We use the same parameters listed in the paper to enable easy comparison of our results with those given. These parameters are $A = 0.1$, $\omega = 2\pi/10$, $\epsilon = 0.1$ and $t_0 = 2.5$ and $t_f = 42.5$ [1]. Initially, we seed 450 points in the domain and take 1000 time steps to calculate trajectory data at each step.

To calculate the trajectories for the double gyre system, we use the scipy Python module 'odeint,' a differential equations solver. The entirety of this research was conducted in Python, building off of pre-existing Ensemble-

based topological entropy calculation (E-tec) algorithm modules, as well as creating new code specific to studying the graph theoretic techniques outlined in this paper.

1.2.3 Building Intuition for Coherent Structure Detection

In order to find coherent structures using a sparse-data approach, we seek to partition a set of fluid particles into sets that mix considerably with the rest of the fluid, and sets that do not mix significantly. One way to conceptualize these sets is to imagine wrapping a taut rubber band around a set of fluid particles, and then advecting the points contained in the interior of the band according to the trajectories of the double gyre system. If this band stretches out exponentially, this indicates that the points contained inside the band are mixing considerably, forming a sort of 'chaotic sea.' This high rate of stretching of the band indicates that these points do not constitute a coherent structure.

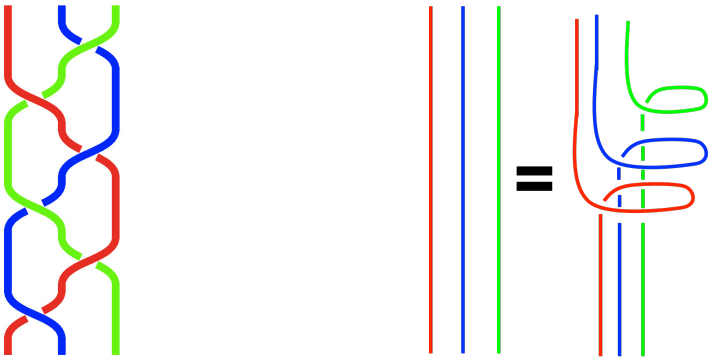


Figure 11: Two braids lead to different stretching: If we put a loop around any two of the strands in the braids above, we can measure the stretching of this loop as the strands are braided. On the left is the golden braid, which gives rise to exponential stretching of the loop, whereas on the right we have rotating strands that amount to no stretching of the loop and a trivial braid

On the other hand, if we wrap a rubber band to include a set of points in its interior region and the band stretches on an algebraic order of magni-

tude, we would consider this stretching to be insignificant (when compared with exponential stretching). Algebraic stretching of the band indicates that the points enclosed by and contained within the band do not mix considerably with the points outside of the band, and so the set of points is a good coherent structure candidate. In order to apply this conceptual framework, we need a more rigorous definition of these rubber bands, as well as what constitutes a "high level" of stretching. We explore such classifications using the E-tec algorithm in the next section.

2 A Sparse Data Approach to Coherent Structure Detection

2.1 Ensemble-based Topological Entropy Calculation (E-tec)

A sparse data approach to fluid flows could enable faster, more efficient prediction of the behavior of real-world, time-dependent flows. The Ensemble-based Topological Entropy Calculation (E-tec) is a sparse data method that analyzes fluid flows by giving a lower bound on the topological entropy of a system. We use E-tec to search for coherent structures using a topological and graph-theoretic approach.

Topological entropy is a global measure of the rate of mixing in a dynamical system that quantifies the amount of chaos in the system. Technically, topological entropy is restricted to fluid flows with periodic flow maps. To find the topological entropy of a fluid flow over a fixed time interval, T, we start with some positive, arbitrarily small ϵ. We consider particles in this flow to be in separate equivalence classes if at any time in the interval, T, the distance between these particles' trajectories is greater than ϵ. As T tends toward infinity and as ϵ tends towards 0, the number of equivalence classes increases exponentially. Topological entropy is the growth rate of distinct equivalence classes. We can also conceptualize the topological entropy as the exponential stretching rate of any material line of fluid in the domain, maximized over all possible material lines. This

conception of topological entropy is more feasible to tackle computationally.

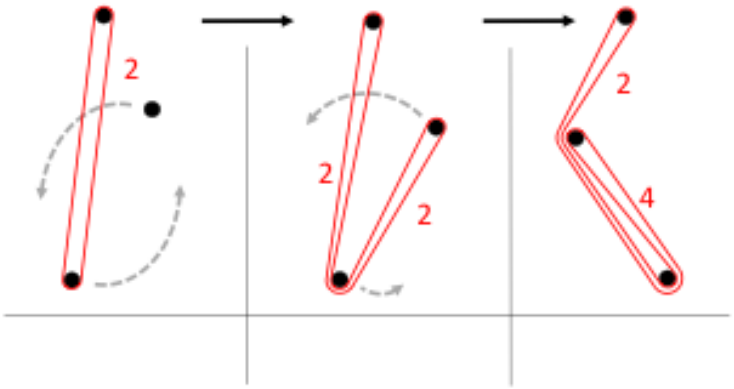


Figure 12: Deforming a band in E-tec. As the band folds over onto itself, the weight of the edge between two vertices represents this folding [15].

E-tec provides a lower bound for the topological entropy of a fluid flow. We begin with a randomly selected set of particles that are sparsely distributed in the domain. The trajectories for these double gyre particles are generated using a differential equations solver, but in the future would ideally be experimental data trajectories. E-tec triangulates these points using a Delauney triangulation, and evolves the triangulation forward according to the particles' trajectories over a user-specified time interval.

For a planar point set, a Delauney triangulation is a triangulation of points such that the circumcircle of every triangle contains no points of the set in its interior. Some of the benefits of the Delauney triangulation are that it maximizes the minimum angle of its triangles, and it is unique up to the position of points in the domain (with a few minor exceptions) [14]. This uniqueness means that if points are in the same configuration at different times and we initialize a triangulation on them at these two distinct times, we will still get the same triangulation in both cases.

Within this set-up, E-tec allows the user to place a rubber band around a collection of points (technically, a "closed, piecewise linear, non-self-

intersecting," two-dimensional band [15]). When fluid particles strike the band, it stretches to accommodate these particles. E-tec records the weights of each edge of the band by monitoring the number of times a section of the band stretches across an edge of the triangulation (see Figure 13).

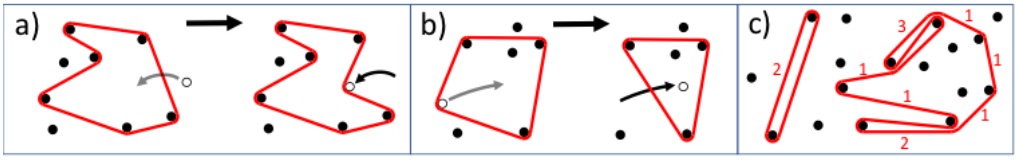


Figure 13: A band stretching in E-tec. In (a), we see that a particle from outside the band strikes and deforms the band. In (b), we see that the band updates to remain taut around the particles contained within it. In (c), we see that edge weights of the band represent the folding of sections of the band back onto itself [15].

The band also remains taut around the outer boundary of the particles it encloses, so that, as the fluid is advected, the band stretches out exponentially. The stretching of this band is precisely what E-tec measures: the user inputs discretized trajectories and specifies the band, and the output is "the number of edge segments in the band as a function of time," which is calculated by tracking the triangulation of particles during advection and updating this triangulation as particles deform the band. The rate of exponential increase of total edge weights in time provides a lower-bound to the topological entropy of the system [15].

By finding the rate of stretching of these bands, E-tec is a valuable tool for evaluating whether a set of points constitutes a coherent structure. More explicitly, if we find a set of points that we believe constitutes a coherent structure, we can put a band around them and run this band through E-tec. If the band stretches exponentially, this rules out this set of points as a coherent structure. If the band stretches only algebraically, then this set of points is a good candidate for a coherent structure (up to

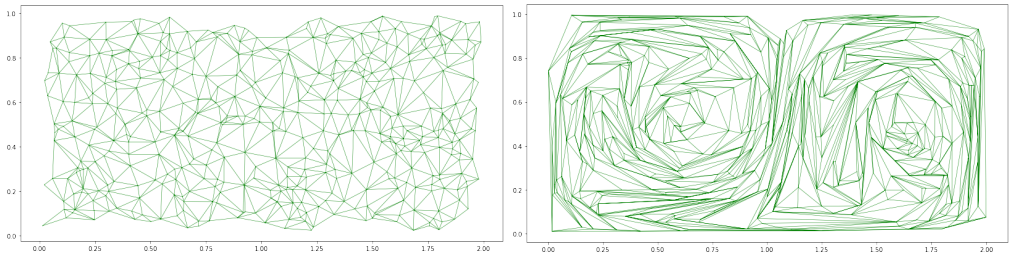


Figure 14: On the left, we have an initial triangulation of points, and on the right, we have the final triangulation that results after the trajectories are run through E-tec. We note that though the initial triangulation is Delauney, the final triangulation (which has been updated to account for the advection of the particles) is not Delauney.

the addition of more points in the interior region of the band).

However, E-tec does not specify which sets of points we should check as possible coherent structures. Even in a sparse data scenario, checking every possible partition of points using E-tec would be a combinatorial nightmare. For instance, (by the Stirling Number of the second kind) even if we were only trying to partition twenty particles into 2 sets, we would have S(20, 2) = 524,287 different possible partitions [11].

Instead, we need to develop a method for partitioning the points into coherent sets, and then use E-tec to evaluate these partitions. Fortunately, we can use E-tec in a different way to search for such partitions.

2.2 Building on E-tec: Defining the Overlap Matrix

Checking candidates for coherent structures is a very useful application of E-tec, but only if we have generated a set of points to check. In order to find such a partition of the particles of a fluid flow, we need to use a different functionality of E-tec by harnessing the power of graph theory.

Within the E-tec algorithm, we are essentially analyzing a changing graph over a set time period. We refer to the graph produced by the triangulation (described in the previous section) as the **Triangulation**

Graph. A **graph** is a collection of nodes and the edges which connect them. An **undirected graph** is a graph in which the edge between nodes A and B can be traversed in either direction. A **weighted graph** is a graph in which each of the edges is assigned a weight (for our purposes, these weights are integers). E-tec produces an undirected, weighted graph, wherein the fluid particles are the nodes of the graph, and the initial edges between them are given by the triangulation. We call a graph **complete** if there exists an edge between every pair of vertices in the graph (see Figure 15).

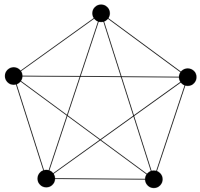


Figure 15: The complete graph on five vertices, K_5

Rather than using E-tec to advect a single 'rubber band' around a set of points, we can apply this idea to each edge in our triangulation. We can think about these edges of the triangulation as forming a mesh of bands, where each edge can be thought of as a rubber band anchored between its two adjacent points. Initially, all of the edges begin with an edge weight of one (except for superfluous boundary edges which have weights of zero). We refer to the edges from this initial triangulation as the initial edges. As E-tec advects the fluid particles according to their trajectories, the algorithm updates the triangulation on the particles. In any given time step, the triangulation consists of the same number of edges. The advection of the particles produces a final triangulation that contains a set of edges, known as the final edges. As the algorithm runs, edges fold into each other and other nearby edges, which causes the edge weights to increase exponentially.

Given an initial edge, we want to know the number of final edges it stretches out to.

We can map every initial edge forward to find what final edges they map onto. This information is contained in the **Weight-Transfer Matrix**, W, where $W_{i,j}$ is equal to the number of times initial edge j is stretched across some final edge i. Summing along each row i in the matrix gives the number of initial edges that have folded onto the ith final edge (where an initial edge that folds k times into final edge i will contribute k to this sum). Summing along a column j gives the total weight that the jth initial edge stretches out to.

$$Column\ Sum = \sum_{i=1}^{n} W_{i,j}$$

$$Row\ Sum = \sum_{j=1}^{n} W_{i,j}$$

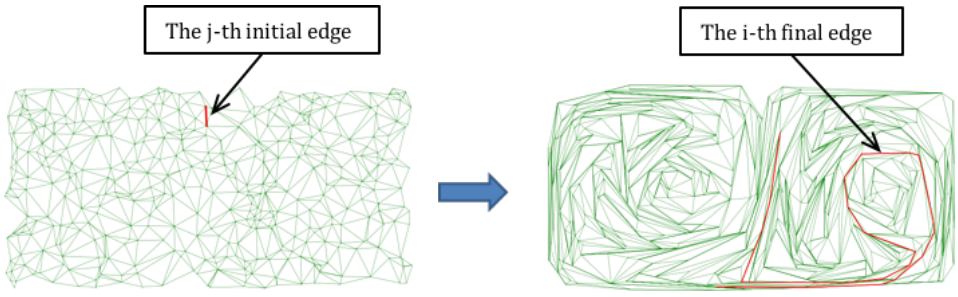


Figure 16: An example of what information the Weight transfer matrix contains, where the jth initial edge stretches across the ith final edge, such that $W_{i,j}$ has a positive value.

We can use the Weight-Transfer Matrix to produce a graph similar to the FTLE. We seed a high density of initial points, take the column sum for each initial edge, and then take the logarithm of each of these sums. Once we have these values, we use a colormap on the domain where light yellow indicates low stretching, and red indicates highly stretching edges.

Since coherent structures are made up of particles that stay close together over time, regions of initial edges in the graph with relatively low final edge weights are good indicators of where coherent structures could be located, in both the FTLE plots and the colormap plots produced by E-tec (see Figure 17). Note that in this figure, we use a high point density, but one of the advantages of E-tec is that it retains very similar information to the FTLE at lower point densities as well.

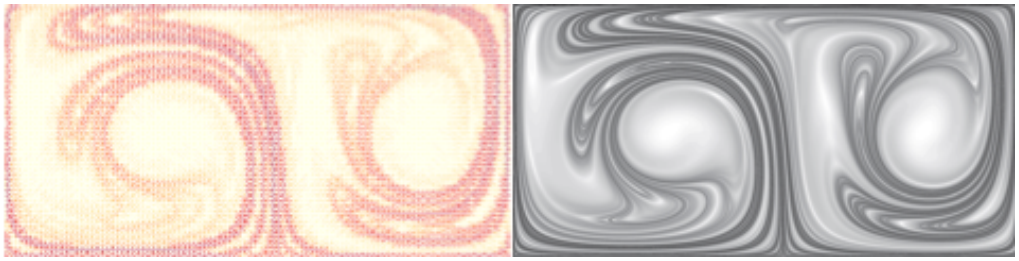


Figure 17: E-tec at a high point density (left) compared with the FTLE for a flow with the same parameters (right) [1].

Now suppose we want to consider two initial edges and analyze their overlap as the fluid is advected. We do this via the **Overlap Matrix**, S, where $S = W^T W$.

Suppose we have n initial edges. Given two initial edges i and j, we map them forward in E-tec to see what final edges they map onto. We can think about the image of these initial edges, i_f and j_f, as n-dimensional vectors: i_f corresponds to the ith column of W and j_f corresponds to the jth column of W. Then if initial edge i contributes to some final edge k, the kth entry of i_f is equal to the number of times initial edge i folds into final edge k, such that the kth entry of i_f is equal to $W_{k,i}$. If initial edge i does not fold into a final edge k, then we assign the kth entry of i_f a value of zero. We follow the same procedure with respect to initial edge j to generate j_f, such that the kth entry of j_f is equal to $W_{k,j}$. In this way, the entry of $S_{i,j}$ is equal to the dot product of i_f and j_f. This gives us a

measure of the overlap that initial edges i and j map forward to.

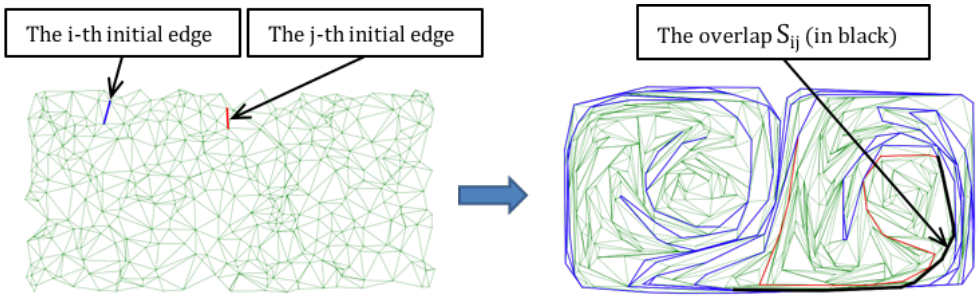


Figure 18: An example of what information the Overlap Matrix contains, where we consider the overlap of two initial edges i and j by finding the final edges that both i_f and j_f map onto.

We can now construct a graph from the Overlap Matrix, known as the **Overlap Graph**. In this graph, initial edges i, j of the Triangulation Graph form the nodes of the Overlap Graph, and the edge between these i and j nodes is weighted by $S_{i,j}$. Hence, the more i and j overlapped during advection, the higher the edge is weighted between them (see Figure 19). If $S_{i,j} = 0$, then there exists no edge between the i and j nodes of the Overlap Graph.

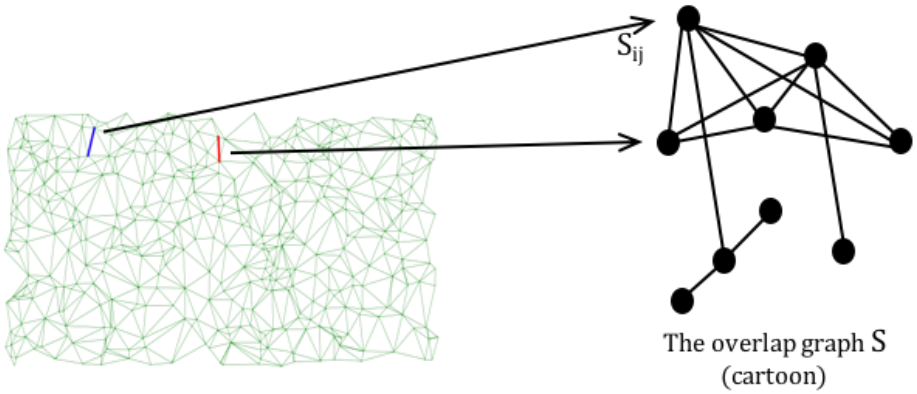


Figure 19: A visual for Overlap Graph

This Overlap Graph is a densely connected, nonplanar graph. We note that, by properties of a triangulation, if we begin with n fluid particles

as nodes of the Triangulation Graph, there are roughly $3n$ edges in that graph, which corresponds to $3n$ vertices of the Overlap Graph. If the Overlap Graph were complete, then we would expect $\frac{3n(3n-1)}{2}$ edges in that graph, from the entries of the (roughly) $3n * 3n$ Overlap Matrix. This is to say that the Overlap Graph is a very large graph with a more complicated structure than the Triangulation Graph.

In order to make sense of the Overlap Graph in terms of coherent structures, we use the following principle:

Principle: the set of initial edges exterior to coherent sets are highly interconnected through large overlap values. This follows from the idea that the fluid particles outside of coherent structures are often highly mixing.

Using this principle, we analyze the Overlap Graph with the goal of finding densely connected components. A graph is connected if every vertex can be reached from a path along edges from every other vertex. A connected component of a graph, G, is a subgraph of G that is connected (see Figure 20).

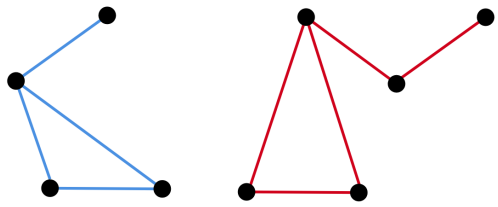


Figure 20: An example of a disconnected graph with its connected components in blue and red.

The densely connected components of the Overlap Graph correspond to initial edges in the Triangulation Graph that have a significant amount of mutual overlap, leading us to believe that these edges are part of the same mixing region in the fluid. Since we have defined coherent structures as regions of fluid that do not significantly mix with the rest of the fluid,

by taking the complement of large connected components in the Overlap Graph, this allows us to find nodes in the Overlap Graph that correspond to edges of the Triangulation Graph that are contained in coherent structures.

3 Implementing Graph Theoretic Measures of Connectivity

3.1 The Edgecut Parameter

In the Overlap Graph, we expect that areas of densely connected nodes correspond to areas of highly mixing fluid. That is, dense connectivity in the Overlap Graph indicates areas devoid of coherent structures. By finding these densely connected areas, we can consider the complement as regions containing potential coherent structures.

One way to search for connected components with high edge weights in the Overlap Graph is by instituting an edge cut parameter, in order to prune the Overlap Graph. At each edge cut, we remove the edges weighted below that edge cut value. This produces a subgraph of the Overlap Graph for each edge cut, weeding out the edges in the Overlap graph with low edge weights. As we let the edge cut value increase, the subgraph it produces will contain fewer edges, and these edges will have increasingly higher weights, indicating that they correspond to the chaotic mixing region of the fluid (see Figure 21). For each subgraph, we use an iterative method to find its largest connected component (as detailed in Section 3.4).

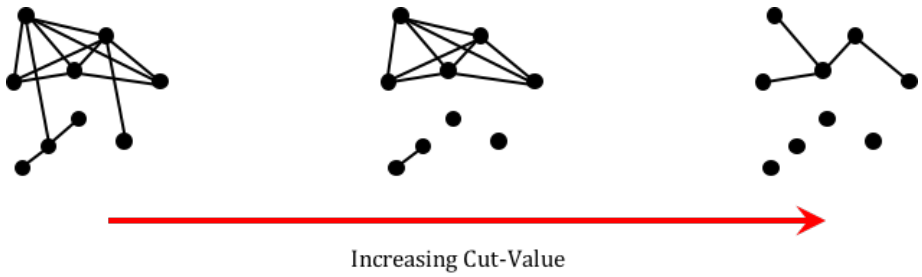


Figure 21: An example of how increasing the cut value prunes a graph

Allowing the edge cut parameter to vary is a similar process to persistent homology, a method of topological data analysis. Given a set of data points (with a distance metric defined on it), we pick an arbitrarily small, positive value, ϵ, and sweep out a circle with radius of ϵ around each node. If two nodes have circles that intersect, we assign an edge between them. If three nodes have circles that intersect, a triangle is formed, which is filled in if all three circles are mutually intersecting. Similar strategies follow for larger subsets of points. In this way, a simplicial complex is built up for each given value of ϵ, consisting of edges, triangles, and potentially higher dimensional geometric objects, like tetrahedra.

For each value of ϵ, we find the Betti numbers of the simplicial complexes. The Betti numbers describe the topological qualities of these complexes in terms of the number of holes these complexes have in different dimensions (formally, the nth Betti number gives the rank of the nth homology group of a topological space [10]). We use these Betti numbers as signatures for a given ϵ, and let the ϵ value vary to see which Betti signatures persist over a range of (increasing) ϵ values. This allows us to analyze the topological properties of the data.

Within our system, we use the edge cut as a parameter similar to that of ϵ. Rather than finding the topological signature of each subgraph, instead we search for graph theoretical signatures of the largest connected com-

ponent for a subgraph produced by a given edge cut value. These graph theoretic measures can, in turn, give us information about the topology of each subgraph.

By reasoning that chaotic mixing regions will give rise to densely connected components in the Overlap Graph, we need some graph theoretic means of finding these components. We use algebraic connectivity and the number of spanning trees as such measures of connectivity. Before implementing these measures, we must prepare the data accordingly.

For each subgraph, S_c (for some edge cut value c), we find the Laplacian, L_c. This matrix is defined as follows:

Definition 3.1. Laplacian **L=D-A**

$$L_{i,j} = \begin{cases} deg(v_i) & \text{if i=j} \\ -1 & if\ v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise} \end{cases}$$

where $deg(v_i)$ is the degree of the vertex i. D refers to the degree matrix of S_c, which has entries equal to the degree of each vertex along the diagonal and zeroes elsewhere. A is the adjacency matrix of S_c, which has zeros along the diagonal and the number of edges connecting v_i and v_j (where $i \neq j$) in its remaining entries.

The Laplacian matrix and its eigenvalues are fundamental to spectral graph theory. One such application is spectral clustering, where the elements of a data set are treated as points of a graph, and the eigenvalues of the Laplacian are used to partition the graph into sets. We use the Laplacian in a similar way, although we have the advantage of starting out with data (the Overlap Graph) that is already in the form of a graph.

One potential drawback of using the Laplacian is that it flattens the

data from the S matrix. From one perspective, considering the Laplacian as a snapshot for a given edge cut value, we could argue we are losing information about the Overlap Graph. The numerical indicator of overlap is not reflected at a given edge cut value, in that the Laplacian reflects whether or not two edges had an overlap that was above the given cut-value, but not the magnitude of such overlap. However, the fact that we take a range of edge cut values and find the Laplacian at each value means that we are not losing the information about the edge weights when we consider the whole range. If we have the Laplacian for every edge cut value (up to the maximum edge weight in the Overlap Graph), we could reconstruct the original edge weights of the entire Overlap Graph.

After finding the Laplacian, we calculate its eigenvalues, and find the algebraic connectivity and number of spanning trees for the largest connected component of the subgraph produced by each edge cut value, as detailed in the next sections.

3.2 Algebraic Connectivity

Algebraic connectivity is a concept from spectral graph theory, and is calculated using the Laplacian and its eigenvalues.

In order to understand the algebraic connectivity, we explore some properties of a graph and its Laplacian. In the Laplacian of a graph, the number of zero-valued eigenvalues indicate the number of connected components in the graph. Recall that a complete graph on n vertices is a simple graph where every pair of vertices are connected by an edge. If a graph is complete, then each of its eigenvalues is equal to n, the number of vertices in the graph, (aside from the first eigenvalue, which is equal to zero, indicating that the complete graph contains one connected component).

The second smallest eigenvalue is known as the **algebraic connectivity**. This number is only greater than zero if the graph is connected. The magnitude of the algebraic connectivity is a measure of the connectivity of a graph. As we increase the number of edges in a graph, the algebraic connectivity increases. Furthermore, the algebraic connectivity can be conceptualized as a measure of how "difficult" it is to split a graph apart into a partition.

The eigenvector associated with the algebraic connectivity is known as the Fiedler vector, and can be used to find a spectral partition of a graph. The vector partitions the sets of nodes based on the sign of their corresponding entry in the Fiedler vector. This partition produces two connected subgraphs, on the conditions that the number of edges between the two graphs is minimized and the subgraphs have roughly the same number of vertices [19].

Recall that we are trying to analyze the connectivity of the largest connected component of S_c, a subgraph of the Overlap Graph (for some edge cut value c). We believe this connected component corresponds to the chaotic mixing region of the fluid flow, and so we seek a densely connected component. Partitioning this connected component into subgraphs of the same size would not aid us in finding coherent structures because such a partition would be meaningless in the context of the problem. If we mapped such a partition on Overlap Graph edges back to the vertices of the Triangulation Graph, neither of the two subgraphs produced are necessarily connected in the Triangulation Graph. For this reason, we do not use the Fiedler vector to partition the Overlap Graph. Instead, we focus on the algebraic connectivity value alone as a measure of connectivity of the largest connected component of S_c.

Algebraic connectivity is sensitive to the number of vertices and the

number of edges in the graph. Since the largest connected component of S_c varies in size with different edge cut values (c), we need a way to normalize the algebraic connectivity values we get for each edge cut value. To normalize the algebraic connectivity, we divide by n (since this is the algebraic connectivity of a complete graph on n vertices). Then for a connected graph, the normalized algebraic connectivity (**NAC**) is a positive value bounded below by 0 and above by 1.

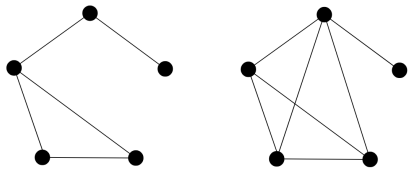


Figure 22: The algebraic connectivity for the graph on the left is .519, whereas the algebraic connectivity for the graph on the right is 1. This increase corresponds to the fact that the graph on the right contains more edges than the graph on the left.

3.3 Spanning Trees

Aside from spectral methods, we use a different graph theoretic measure of connectivity. Consider the number of possible routes we can traverse on the edges of a graph, if we start on one vertex, want to reach every other vertex, and are not allowed to travel in circles. The total number of possible routes we could take gives us an idea of how densely connected the graph is.

We can pose this idea more formally in terms of graph theory. A **circuit** is a subset of the edges of a graph such that a closed path exists that begins and ends at the same vertex (see Figure 23).

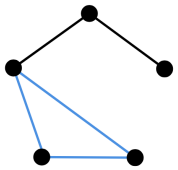


Figure 23: A circuit in blue

A tree is defined as a connected graph that contains no circuits. A **spanning tree** is a tree that has an edge incident to every vertex (see Figure 24). Note that there can be more than one spanning tree on a graph, as shown in Figure 25. In this way, the more densely connected a graph is on n vertices, the greater the number of trees that span the graph. Kirchoff's theorem gives the number of spanning trees as follows, based on the nonzero eigenvalues of the Laplacian, λ_i, for $i \geq 2$:

Kirchoff's Theorem:

$$t(G) = (1/n)(\lambda_2 \ldots . . \lambda_n)$$

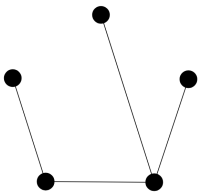


Figure 24: A spanning tree on five vertices

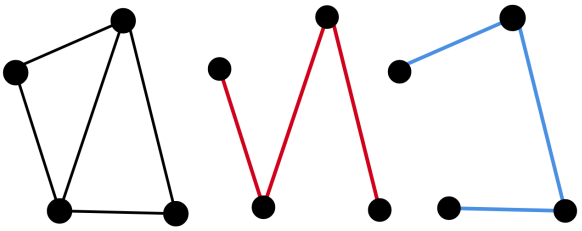


Figure 25: A graph on four vertices in black, with two distinct spanning trees of this graph given in red and blue.

For a complete graph on n vertices, every eigenvalue (aside from the first eigenvalue, λ_1 0) has a value of n. We plug these values into Kirchoff's Theorem to see that a complete graph has n^{n-2} spanning trees (also known as Cayley's Formula). Then for a connected (but not necessarily complete) graph on n vertices, we can normalize the number of spanning trees by dividing by this value. This normalized spanning trees value is positive for a connected graph and bounded above by one (for a complete graph). However, the number of spanning trees is too large for the algorithm to compute, even on our sparse data set. Instead, we consider the logarithm of the spanning trees, and normalize that, called the **LNST**, which takes values between 0 and 1:

$$LNST = \frac{\log t(S_c)}{(n-2)log(n)}$$

where $t(S_c)$ is the number of spanning trees of the largest connected component of S_c, and n is the number of vertices contained in that connected component. The denominator here represents the logarithm of n^{n-2}, the number of spanning trees in K_n.

3.4 Analyzing these Values

We have discussed graph theoretic measures of connectivity for the largest connected component of S_c, but have not yet detailed how to find this connected component. In this section, we describe the iterative method we use to find the connected components for each S_c, followed by organizing these connected components by size from biggest to smallest.

This method consists of starting with an n-dimensional vector, $\vec{v}$, with a value of zero in each entry except the kth entry, where k is randomly chosen. This kth entry is assigned a value of one. We multiply $\vec{v}$ by

the matrix produced at a given edge cut, S_c. By multiplying $S_c\vec{v}$, this method picks out the vertices in S_c that are adjacent to k, since if k and s are adjacent nodes, the product of S_c and $\vec{v}$ will be an n-dimensional vector with a non-zero entry in the kth and sth position. This new vector (with positive entries in the kth and sth entries) is then normalized and multiplied by S_c again. This is repeated iteratively until the number of non-zero components in the vector stops increasing, which implies we have captured all the vertices of a single connected component of S_c. The vertices in this connected component correspond to the positions of the non-zero entries of the vector produced after the final iteration.

We repeat this method to find every connected component of S_c, and then we order the connected components by the number of vertices they contain. Typically, the largest connected component for a cut-value contains considerably more vertices than the other connected components of S_c, since the largest connected component corresponds to the main, highly mixing region of the double gyre system. At this step, we are looking for an edge cut that produces a subgraph S_c that has high values for both the NAC and LNST, but also an edge cut value for which these values correspond to a relatively large connected component. We select an edge cut value based on the NAC and LNST values of S_c for all possible edge cut values, c.

After selecting an edge cut value c, we consider the largest connected component, call it C_c, of the subgraph S_c. We can now reconsider C_c in the context of our Triangulation Graph. Since every vertex in C_c corresponds to an initial edge of the Triangulation Graph, then C_c can be mapped back onto the Triangulation Graph. The result of this map is a subgraph of the Triangulation Graph, T_c, where every edge i in this set corresponds to a vertex v in the Overlap Graph (and we include the vertices in the

Triangulation Graph if such an edge i is incident to them). Note that even though C_c was connected in the Overlap Graph, T_c need not be connected.

Since we operate on the assumption that C_c corresponds to the mixing regions of the flow, when we map C_c to T_c, we assume that T_c also captures the mixing regions of the flow by virtue of the way we constructed T_c. Since we are searching for coherent structures that do not mix considerably with the rest of the fluid, we take the complement of T_c in the Triangulation Graph.

We find the connected components of this complement, and consider the largest four (which we assume correspond to our coherent structure candidates). Recall that when we seek to find coherent structures, we want a partition of the vertices of the Triangulation Graph, not a partition of its edges. Then we have found coherent structure candidates, which consist of connected subgraphs of the Triangulation Graph, via a partition of the edges of the Triangulation Graph. However, recall that the vertices (not the edges) of these connected components constitute our coherent structure candidates, since these vertices correspond to the fluid particles in our double gyre system.

Once we have found a partition of points on the Triangulation Graph, we can check if these points constitute a coherent structure by using E-tec to measure the stretching of a band around each set. Note that constructing these bands is not trivial. For each coherent structure candidate, we have a set of vertices, V, and seek to wrap a band around them such that every vertex v in V is either in the region enclosed by the band or is part of the band (such that the edges incident to v are selected as part of the band's edges). This band should only enclose and contain vertices that are part of our coherent structure candidates. In practice, this band can be difficult to construct, as explained in the next section.

4 Results

Now that we have outlined the methods, we take the reader through a few examples using this process.

First we initialize a triangulation on randomly seeded points in our domain. We evolve this triangulation forward using E-tec and extract the Weight-transfer matrix, W, and then find the Overlap Matrix, S, where $S = W^T W$. Using S, we find S_c for each edge cut, c. Then for each S_c, we find its Laplacian, L_c. Using the Laplacian, we calculate the the normalized algebraic connectivity (NAC), as well as the normalized logarithm for the number of spanning trees (LNST), for the largest connected component of S_c.

This produces data in Figure 26. From this data, we see that near the edge cut value of 11, the NAC levels off after this value, and the LNST value is near a maximum. Now we choose edge cut 11 and get the following result, Figure 27a where we map the edges of the largest connected component of S_c back to the Triangulation Graph.

Then we take the complement of this connected component in the Triangulation Graph (Figure 27b). We find the connected components of the complement using the same iterative method detailed in the previous section, and put bands around them (Figure 27c).

Finally, once we have found these bands, we can use E-tec to check if they stretch out. Figure 27d gives the initial bands on this triangulation. Then we find the final stretching. We refer to structures outlined in

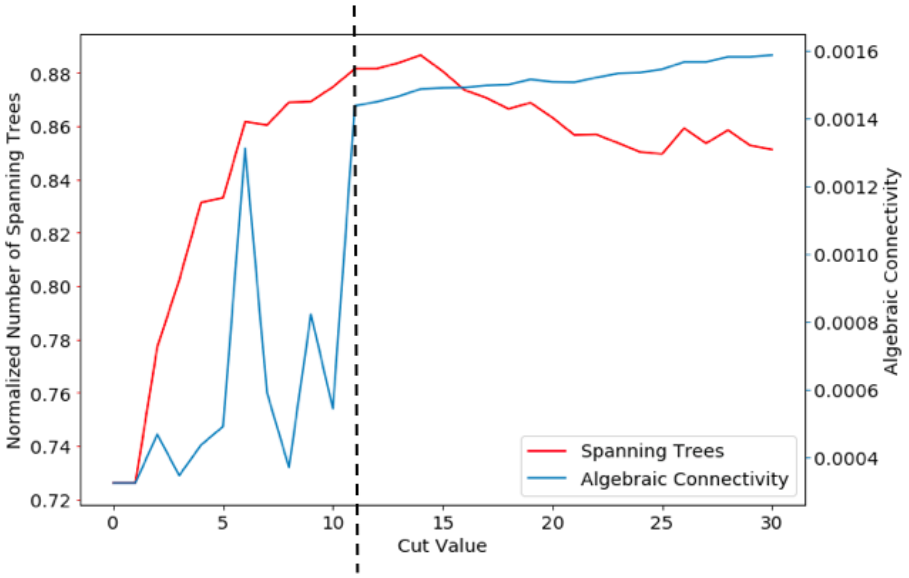


Figure 26: We choose 11 (marked with the dashed vertical line) as the edge cut value from this data.

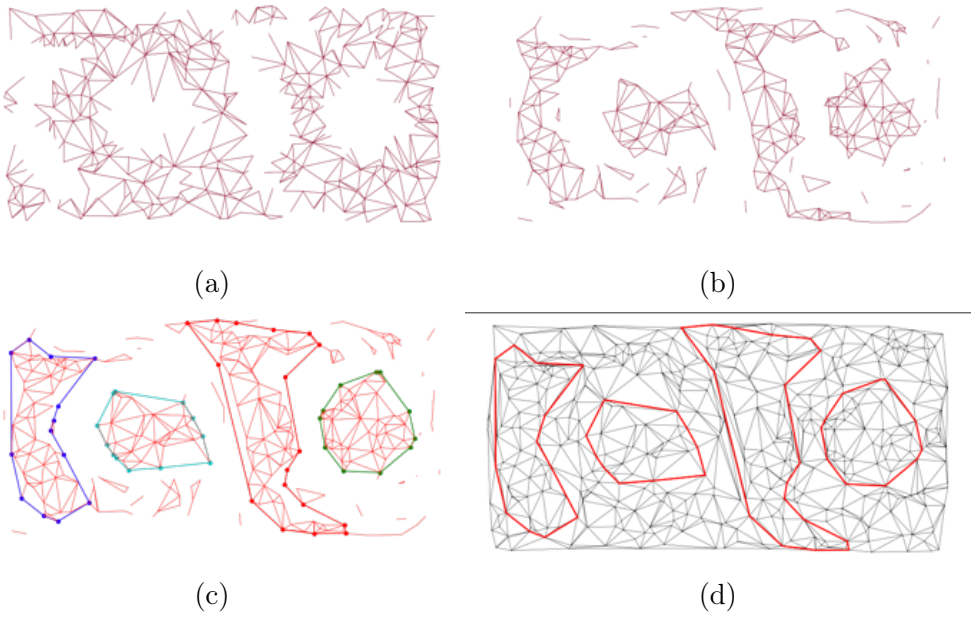
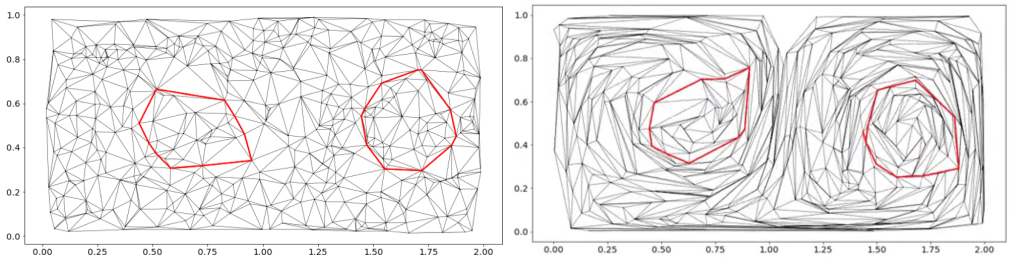
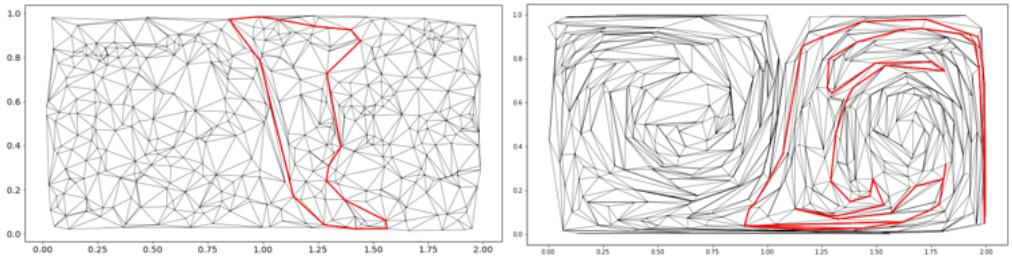


Figure 27: We take the edge cut of 11 in the Overlap Graph, producing S_{11}. In (a), we have the largest connected component of S_{11} mapped back to the Triangulation Graph to form T_{11}, a subgraph of the Triangulation Graph (as pictured). In (b), we have the complement of T_{11} in the Triangulation Graph. In (c), we wrap bands around the connected components of the complement and prepare these bands for E-tec in (d).



(a) Initial (b) Final

Figure 28: The stretching of the band for the round structures at an edge cut of 11



(a) Initial (b) Final

Figure 29: The stretching of the band for the right oblong structure at an edge cut of 11

green and cyan in Figure 27c as the **round structures**, and the remaining structures (on the left and between the round structures) as the **oblong structures**. Figure 28 shows that the round structures do not stretch significantly, while Figure 29 shows the way the band around the oblong structure in the middle of the domain stretches considerably for edge cut value 11 (outlined in red in Figure 27c). Similarly, the band for the oblong structure on the left also stretches considerably.

Now note that using a different triangulation, the results change for the edge cut graph behavior. For instance, we can initialize a different triangulation that produces the new NAC-LNST-edge cut graph shown in Figure 30.

If we take an edge cut of 11 for this new triangulation, we get a connected component of size 740 edges. For comparison, the second smallest

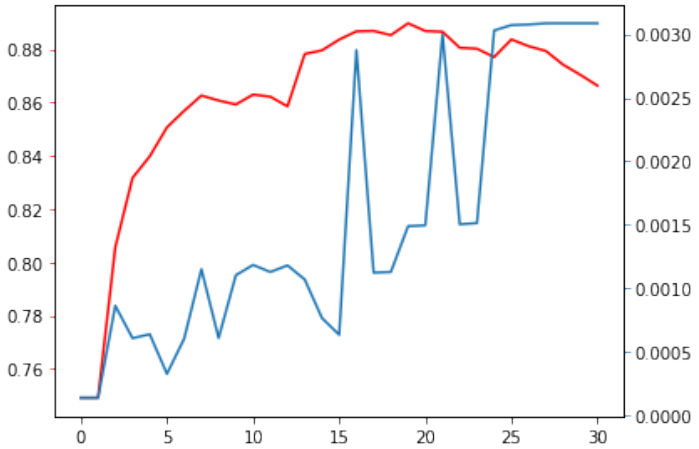
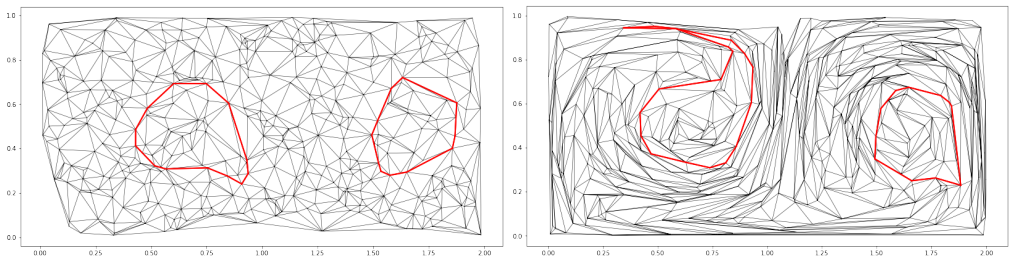
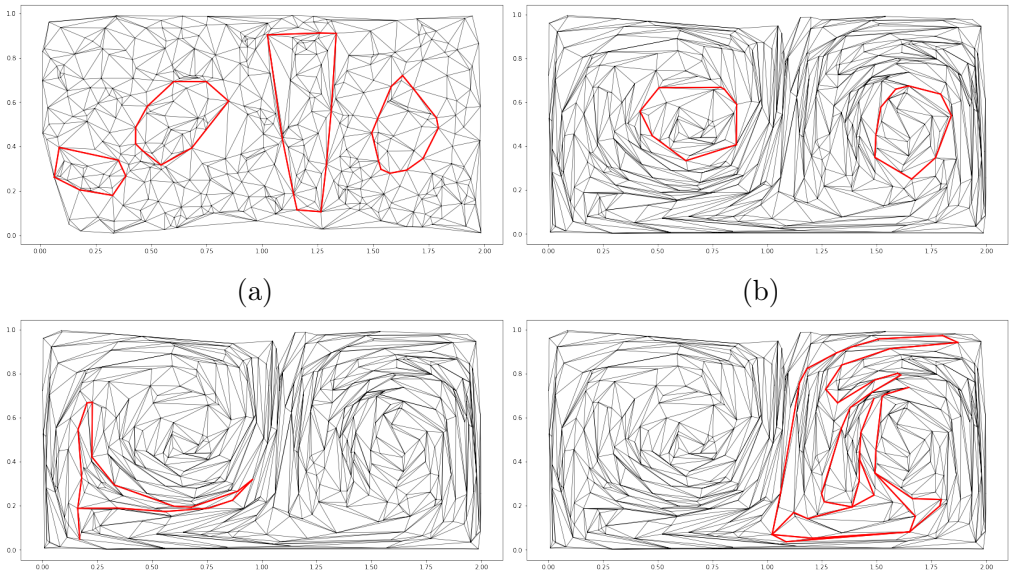


Figure 30: A graph for a new triangulation, showing the LNST in red (with its values on the left axis) and the NAC in blue (with its values on the right axis). Note that the NAC does not level off in this graph, unlike the graph for the first triangulation in Figure 26.



(a) Initial (b) Final

Figure 31: For an edge cut of 11 in our second triangulation, we see that the left round structure band stretches considerably.



(a) (b)

(c) (d)

Figure 32: For an edge cut of 6 in our second triangulation, (a) shows the initial bands of the four structures, (b) shows the final stretching of the round structures, (c) shows the final stretching for the leftmost structure in (a), and (d) shows the stretching for the oblong structure between the round structures in (a).

component has a size of 1. However, if we choose edge cut 11 for this new triangulation, the two round structures have initial bands in Figure 31a stretching to the bands in Figure 31b. We note that the left band stretches significantly for this edge cut value. Therefore, an edge cut value that produces coherent structures in one triangulation may not be successful for a different triangulation.

We need to reduce the edge cut value to 6 to ensure that these round structures do not stretch out (see Figures 32a and 32b. Figure 32c shows the stretching of the left band in Figure 32a, where Figure 32d shows the trapezoidal band stretching from Figure 32a). Even for the edge cut value of 6, the oblong structures still stretch considerably.

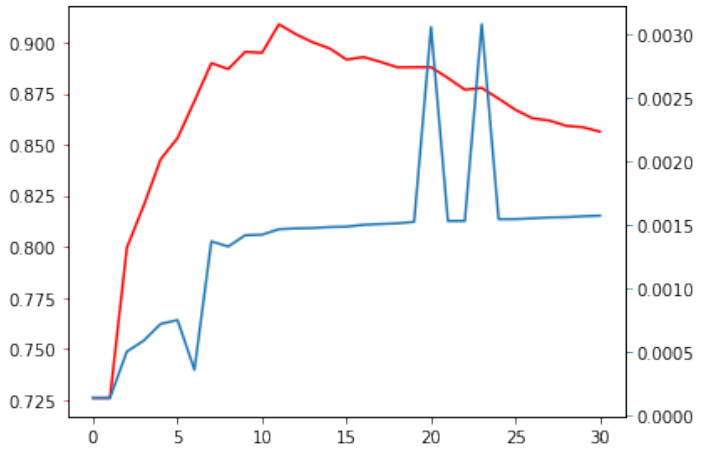
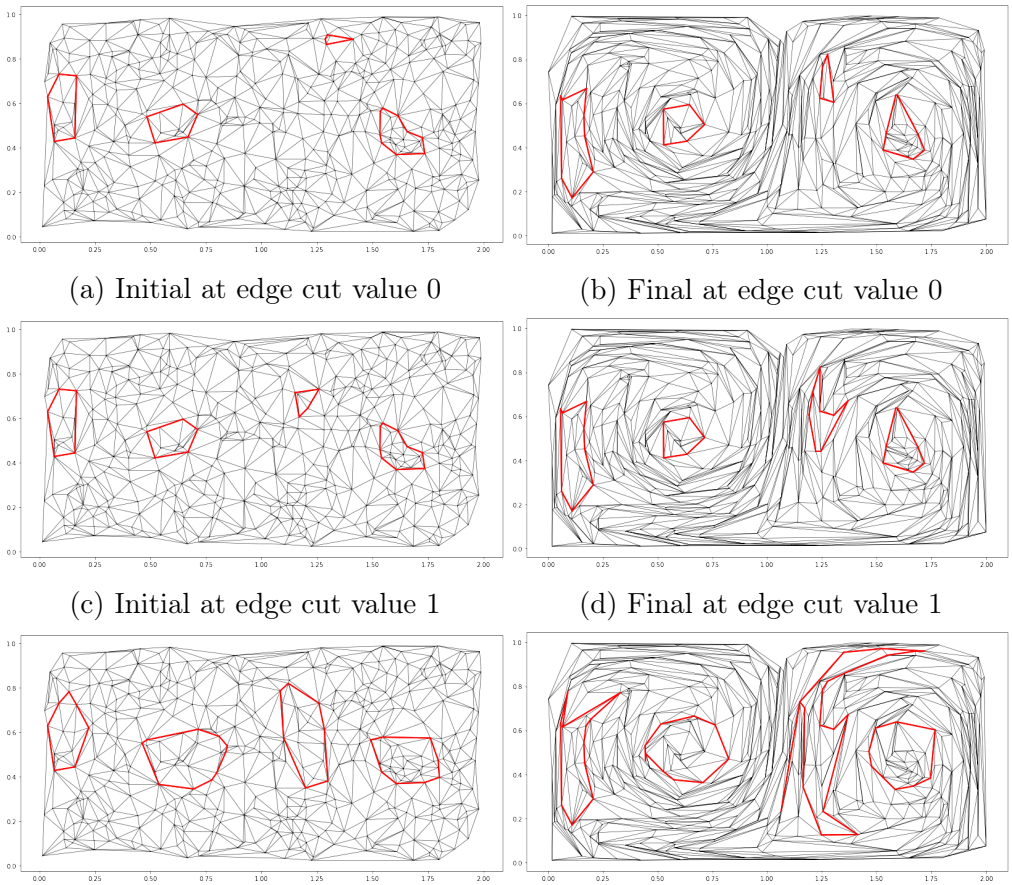


Figure 33: A graph for a third triangulation, showing the LNST in red (with its values on the left axis) and the NAC in blue (with its values on the right axis). Note that the NAC features two prominent spikes, similar to the graph for the second triangulation in Figure 30.

We consider one more triangulation with the edge cut graph in Figure 33, which we refer to as the third triangulation. Rather than picking an edge cut from the graph, we explore what happens when we vary the edge cut from 0 to 1 to 2 (see Figure 34). Note that even with a low edge cut value of 2, we still see stretching in the band for the right oblong structure (see Figure 34f). This highlights that the complement of the Overlap Graph for an edge cut of 2 is too large, causing us to include edges in the coherent structure band that should not be included in the structures.

Finally, there appears to be an issue in the way we wrap bands around the connected components that constitute our coherent structure candidates. For an edge cut of 11 in this third triangulation, we produce Figure 35a, showing the bands we put around these connected components. However, in Figure 35b, we can see that these bands in E-tec are excluding some of the edges that are part of the connected component we picked out. This is particularly noticeable for the oblong structures, where edges that are connected to the rest of the graph at only one vertex are not enclosed by the band. The fact that these portions of connected components are

being excluded makes it difficult to fully vet the edge cut method.



(a) Initial at edge cut value 0

(b) Final at edge cut value 0

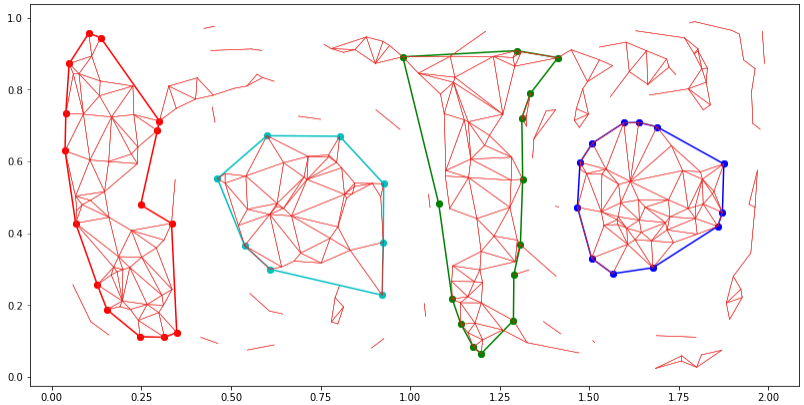
(c) Initial at edge cut value 1

(d) Final at edge cut value 1

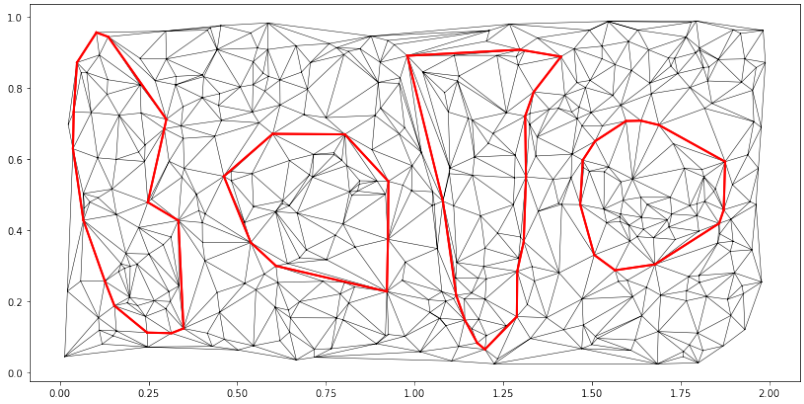
(e) Initial at edge cut value 2

(f) Final at edge cut value 2

Figure 34: As we increase the edge cut value, we see that the round structures appear more robust to this increase (they do not stretch considerably at any stage), whereas both of the oblong structures are very sensitive to the increase from an edge cut value of 1 to 2.



(a)



(b)

Figure 35: Comparing the connected components we picked out with the bands that are wrapped around them, we find edges that are that are part of the connected component we've isolated, but not wrapped in the band in E-tec (particularly on the upper corners of the oblong structures).

5 Conclusion

Overall, this method for detecting coherent structures has promise, but needs refinement. While the edge cut parameter could yield valuable information, it is important to note that it may be dependent on the random triangulation of points we begin with. That is, the fact that the trajectories generated at the beginning of each run are always randomly generated means the edge cut values differ significantly from run to run. This means that going from one triangulation to another, the user can't simply guess an edge cut that is in the ballpark from a previous triangulation, but must have some criteria based on the LNST and NAC values. Such a criteria has yet to be developed. Perhaps by collecting statistical data from different triangulations and their edge cuts, a more rigorous selection process for an edge cut value can be formalized.

Recall that when we introduced the time-independent gyre, we saw that the FTLE field isolated two coherent structures (see Figure 8a). We note that these two time-independent coherent structures are also present in the case where the double gyre is time-dependent (known as the round structures), and that these are the structures our method picks out with reasonable accuracy. These round structures seem fairly time-invariant.

We note that the remaining two oblong coherent structures are heavily time-dependent. If we integrated the trajectories (or specify that E-tec integrate) over a longer time interval, we would see these structures collapse. Recall that our method of picking an edge cut falls apart with respect

to the oblong structures: in some cases, there is no need for an edge cut with regards to these structures: the Overlap Graph's dense connectivity (with no edge cut) has a complement that usually constitutes coherent structures, such that the bands around the connected components of the complement stretch algebraically. In this case, taking an edge cut means including points in the bands of the oblong structures that cause the bands to stretch out. Essentially, the problem is that the edge cut appears to find oblong structures that are too large, enclosing points within the band around these oblong structures that cause the band to stretch considerably.

The problems with the oblong structures indicate that their mixing behavior is not captured as easily with the Overlap Matrix as for the round structures. That is, there is not necessarily a high enough level of mixing around the boundaries of the oblong structures that can be easily picked out by the Overlap Graph.

It is helpful to conceptualize this partitioning problem of fluid particles in the following way. We are concerned with classifying the stretching behavior of three different categories of edges in the Triangulation Graph. The first category is those edges within the chaotic mixing region. These correspond to high edge weight values in the Overlap Matrix. The second category are those edges within the coherent structures, which correspond to low mixing regions and the complement of the largest connected component in the Overlap Graph at a given edge cut value. So far, our methods have worked fairly well in tackling these two categories of edges. However, the third category appears to be the trickiest: classifying the properties of edges in the Triangulation Graph that span the boundaries of a coherent structure, such that they connect a point within a structure to a point within the chaotic mixing region (see Figure 36). The stretching of these edges is difficult to classify with the Overlap Matrix method. I believe

it is these boundary edges that lead to trouble for the oblong structure detection.

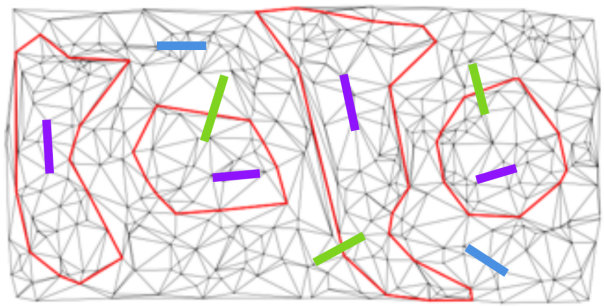


Figure 36: Mixing edges are in blue, coherent structure edges are in purple, and boundary edges are in green.

There are different routes to proceed. One possibility is to institute a different way of using an edge cut parameter and attempt to isolate edges in the Triangulation Graph with low stretching. This method seeks to build up coherent structures from their constituent parts, rather than isolating areas of high stretching and analyzing what's left over. If we wanted to take an edge cut from below, we choose a cut value of k and eliminate all edges whose weights are above k.

We run into a couple of issues with this process. If we have two chaotic edges in different initial regions, there is no guarantee that these two edges will have any overlap, and we would not want these low overlap values to be indicative of coherent structure behavior. We would need to institute a measure of local connectivity on the Triangulation Graph before finding the Overlap Matrix. One such method could be multiplying the Weight-Transfer Matrix component wise with the adjacency matrix of the Triangulation Graph. This would produce a graph that excludes non-adjacent initial edges with no overlap, and therefore such edges would not be present in the subgraphs produced by an edge cut from below. This guarantees that the low overlap edges that are found by taking an edge cut from below

would actually be reasonable coherent structure candidates.

However, implementing notions of local connectivity renders the utility of connected components obsolete. Coherent structures can no longer be picked out by eliminating one massive connected component, but instead must be assembled via a collection of many small connected components with low stretching. This begs the question of what size a connected component would need to be in order to be significant enough to be included in a coherent structure candidate.

Lastly, one of the roadblocks in our method for detection is inherent to the way we have conceptualized the problem. We are seeking to partition a set of nodes in the Triangulation Graph, but instead our methods rely on partitioning the edges of that graph instead (via an admittedly convoluted method of partitioning edges in a higher-dimensional graph). This presents a problem for several reasons. Firstly, there is not a unique way to map from a partition on the set of edges in the Triangulation Graph to a partition on the nodes of that graph. For instance, how do we partition vertices adjacent to the boundary edges? Do we require that vertices in a coherent structure are those that have no mixing edges incident to them, or are vertices that have both mixing and non-mixing edges incident to them acceptable? Secondly, it is difficult to define which of our edges are boundary edges because we are seeking to define the boundary itself. If these boundary edges leave some signature that could help us define the boundary, we have yet to rigorously define it. If instead we had a method for partitioning the vertices of the Triangulation Graph, a partition of edges into our three categories (coherent structures, chaotic mixing, boundary) would follow naturally (although in that case, such classifications would probably be obsolete).

The methods detailed in this paper have attempted to categorize the

behavior of coherent structures, but require more rigorous trial and error to measure their worth over triangulations of randomly distributed particles. Additionally, these methods have not been analyzed rigorously for different point densities or over different time intervals. These problems are not insurmountable, but would require time and care in order to properly vet this work, and decide whether it is a fruitful path of inquiry.

References

[1] Michael R. Allshouse and Thomas Peacock. Lagrangian based methods for coherent structure detection. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(9):097617, 2015.

[2] Michael R Allshouse and Jean-Luc Thiffeault. Detecting coherent structures using braids. *Physica D: Nonlinear Phenomena*, 241(2):95–105, 2012.

[3] Brian Dunbar. NASA Imagery of Oil Spill, 2010. `https://www.nasa.gov/topics/earth/features/oilspill/index.html`.

[4] Environmental Protection Agency. Deepwater Horizon: BP Gulf of Mexico Oil Spill, Apr 2017. `https://www.epa.gov/enforcement/deepwater-horizon-bp-gulf-mexico-oil-spill`.

[5] Gary Froyland. An analytic framework for identifying finite-time coherent sets in time-dependent dynamical systems. *Physica D: Nonlinear Phenomena*, 250:1–19, 2013.

[6] Gary Froyland and Kathrin Padberg-Gehle. A rough-and-ready cluster-based approach for extracting finite-time coherent sets from sparse and incomplete trajectory data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(8):087406, 2015.

[7] G.U.N.T Equipment for engineering education. `http://www.usdidactic.com/teaching-lab-equipment/i4320_Zeichnung_Einzelheit_1.htm`.

[8] Alireza Hadjighasem, Mohammad Farazmand, Daniel Blazevski, Gary Froyland, and George Haller. A critical comparison of Lagrangian methods for coherent structure detection. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(5):053104, 2017.

[9] International Ocean Discovery Program. South African Climates, Jun 2017. https://joidesresolution.org/expedition/361/.

[10] Wolfram MathWorld. Betti number.

[11] Austin Mohr. Stirling number calculator. http://austinmohr.com/home/?page_id=431.

[12] Thomas Peacock and George Haller. Lagrangian coherent structures: The hidden skeleton of fluid flows. *Physics Today*, 66(2):41–47, 2013.

[13] Thierry Pichevin, Doron Nof, and Johann Lutjeharms. Why are there Agulhas Rings? *Journal of Physical Oceanography*, 29(4):693–707, 1999.

[14] VT Rajan. Optimality of the Delaunay triangulation in $\mathbb{R}$. *Discrete & Computational Geometry*, 12(2):189–202, 1994.

[15] Eric Roberts, Suzanne Sindi, Spencer A. Smith, and Kevin A. Mitchell. Ensemble-based topological entropy calculation (E-tec). *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(1):013124, 2019.

[16] Shawn C. Shadden. Lagrangian coherent structure tutorial, Apr 2005. https://shaddenlab.berkeley.edu/uploads/LCS-tutorial/FTLE-interp.html.

[17] Shawn C Shadden. Lagrangian coherent structures. *Transport and Mixing in Laminar Flows: From Microfluidics to Oceanic Currents*, pages 59–89, 2011.

[18] Shawn C Shadden, Francois Lekien, and Jerrold E Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena*, 212(3-4):271–304, 2005.

[19] Brian Slininger. Fiedler's theory of spectral graph partitioning, 2013.

[20] Jean-Luc Thiffeault. Braids of entangled particle trajectories. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(1):017516, 2010.

[21] Y. Wang, F. J. Beron-Vera, and M. J. Olascoaga. The life cycle of a coherent Lagrangian Agulhas ring. *Journal of Geophysical Research: Oceans*, 121(6):3944–3954, 2016.