# Time-Dependent Invariant Manifolds Theory and Computation

Cole Lepine

June 1, 2007

**Abstract**

Due to the complex nature of dynamical systems, there are many tools that are used to understand the nature of the flows. One such tool is the invariant manifold, which is relatively easy to calculate for the autonomous system. The time-dependent invariant manifold, or Lagrangian Coherent Structure (LCS), is the extension of this idea to the case where the velocity field is time-dependent. This project introduces the theory behind the idea, and presents sample computations using MATLAB for simple time-dependent systems. Numerical results demonstrate how different paramters controlling the LCS affect the accuracy and efficiency of the computation (e.g., the computation time, refinement of the mesh, etc.)

## Introduction and Motivation

In the theory of time-indepedent dynamical systems, the invariant manifold is very useful in determining the qualitative properties of groups of trajectories. As the name invariant implies, points on the invariant manifold stay on the manifold as they move forward (or backward) in time. An interesting fact is that often we can see distinct types of behaviour on the different sides of the manifold. The ultimate example of this is the separatrix of the simple pendulum. The separatrix of the simple pendulum separates the plane into two sections, each of which contains trajectories that have similar properties. In the inner section the trajectories are closed orbits, and in the outer section the trajectories are 'running modes' (corresponding to when the bob of the pendulum actually swings completely around its fixed point).

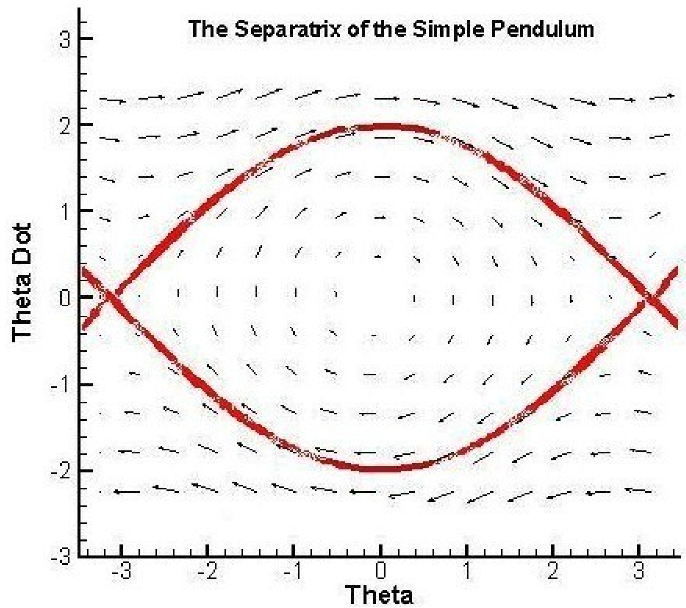


Figure 1: The phase portrait of the simple pendulum.

This example is what motivates the idea behind the extension to time-dependent dynamical systems. We want to find invariant manifolds in the time-dependent system that separate distinct types of behaviour, allowing us to qualitatively see how the system is evolving in time. In the simple pendulum, by looking at points at either side of the separatrix, we see that as we move either forward or backward in time the points move very far apart. Thus, the separatrix is characterized by causing nearby points to have a large stretch forwards or backwards in time. This stretch is the measure that we use to find time-invariant manifolds in the time-dependent case.

# Basic Theory of Lagrangian Coherence Structures

The theory presented here is found in [3]. We want to find this measure of most stretching over some time $T$ ($T$ can be positive or negative). This means

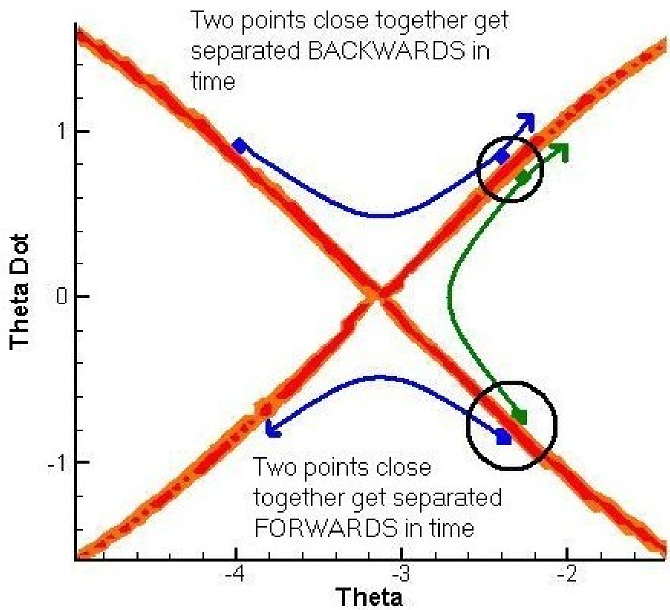


Figure 2: Close-up of the hyperbolic fixed point.

that we want to associate with each point in our domain $D$ a number, thereby creating a scalar field on $D$. We call this field the Finite-Time Lyapunov Exponent (FTLE). To be formal, we list all the ingredients that we have:

- $D \subset \mathbb{R}^n$ is our domain of interest
- $\mathbf{F}(\mathbf{x}, t)$ is the flow field on $D$
- $T$ is the time over which we want to investigate the stretching
- $\mathbf{x} \to \phi_{t_0}^{t_0+T}(\mathbf{x})$ is the map that sends a point $\mathbf{x}$ at time $t_0$ to its image at time $t_0 + T$ after being advected by the flow $\mathbf{F}$

Now we consider a point $\mathbf{x} \in D$, and some point $\mathbf{y}$ close to $\mathbf{x}$. We have:

$$\delta\mathbf{x}(t_0) = \mathbf{y} - \mathbf{x} \tag{1}$$

We want to see the distance between these points after a time $T$, so we want to look at:

$$\delta\mathbf{x}(t_0 + T) = \phi_{t_0}^{t_0+T}(\mathbf{y}) - \phi_{t_0}^{t_0+T}(\mathbf{x}) \tag{2}$$

We want to relate this part to our original distance $\delta\mathbf{x}(t_0)$, so we apply Taylor's theorem to get:

$$\delta\mathbf{x}(t_0+T) = \frac{\mathrm{d}\phi_{t_0}^{t_0+T}(\mathbf{x})}{\mathrm{d}\mathbf{x}}\delta\mathbf{x}(t_0) + \mathcal{O}\left(\|\delta\mathbf{x}(t_0)\|^2\right) \tag{3}$$

Now, depending on the size of $\delta\mathbf{x}(t_0+T)$ we can ignore the last term in (3). Thus, we make sure that we take $\delta\mathbf{x}(t_0)$ small. Now to determine the actual size of the stretching, we apply a simple norm to our distance quantity:

$$\|\delta\mathbf{x}(t_0+T)\| = \sqrt{\left\langle \frac{\mathrm{d}\phi_{t_0}^{t_0+T}(\mathbf{x})}{\mathrm{d}\mathbf{x}}\delta\mathbf{x}(t_0), \frac{\mathrm{d}\phi_{t_0}^{t_0+T}(\mathbf{x})}{\mathrm{d}\mathbf{x}}\delta\mathbf{x}(t_0) \right\rangle} \tag{4}$$

Since the derivative is a linear map between finite-dimensional spaces, we know that it has an adjoint. Thus, we can use the adjoint property to consider the quantity:

$$\Delta = \frac{\mathrm{d}\phi_{t_0}^{t_0+T}(\mathbf{x})}{\mathrm{d}\mathbf{x}}^* \frac{\mathrm{d}\phi_{t_0}^{t_0+T}(\mathbf{x})}{\mathrm{d}\mathbf{x}} \tag{5}$$

We can maximize $\langle \mathbf{x}, \Delta\mathbf{x} \rangle$ by taking $\mathbf{x}$ to be in the direction of the eigenvector of $\Delta$ corresponding to its largest eigenvalue (as proved in most elementary texts in linear algebra, or on CDS qualifying exams). By the construction of $\Delta$ and the singular value theorem, we also have that this maximum eigenvalue must be positive. So if we take our original point $\mathbf{y}$ such that $\delta\mathbf{x}(t_0)$ is parallel to this eigenvector, we can write:

$$\|\delta\mathbf{x}(t_0+T)\| = \sqrt{\delta\mathbf{x}(t_0), \Delta\delta\mathbf{x}(t_0)} = \sqrt{\lambda_{MAX}}\,\|\delta\mathbf{x}(t_0)\| \tag{6}$$

Now we can define the FTLE:

**Definition 1 (FTLE)** *We call* $\sigma_{t_0}^{T}(\boldsymbol{x}) = \frac{1}{|T|}\ln\sqrt{\lambda_{MAX}}$ *where* $\lambda_{MAX}$ *is the maximum eigenvalue of* $\Delta$ *the Finite-Time Lyapunov Exponent.*

The scaling of $\sigma_{t_0}^{T}(\mathbf{x})$ is useful because often the stretching between points can become exponentially large.

Now that we are equipped with our scalar field that measures stretching, it is time to define our invariant manifold. In this project, we restrict to the case of $D \in \mathbb{R}^2$, but it is possible to use the FTLE to find invariant manifolds for general $D$. Motivated by our example of the simple pendulum, we see that each point on the separatrix has associated with it the highest possible stretch number, and that immediately off of the separatrix the stretching drops dramatically. This is similar to the physical idea of a 'ridge'. Thus, we mathematically define a ridge:

**Definition 2** *A **Ridge** is a curve $\gamma(\boldsymbol{s})$ in $D$ that satisfies two properties:*

- $\gamma'(\boldsymbol{s})$ *is parallel to* $\nabla\sigma_{t_0}^{T}(\gamma(\boldsymbol{s}))$
- $\dfrac{\mathrm{d}^2\sigma_{t_0}^{T}(\gamma(\boldsymbol{s}))}{\mathrm{d}\boldsymbol{x}^2}\cdot(n,n) = \min_{\|u\|=1} \dfrac{\mathrm{d}^2\sigma_{t_0}^{T}(\gamma(\boldsymbol{s}))}{\mathrm{d}\boldsymbol{x}^2}\cdot(u,u) < 0$

  *where $n$ is a unit normal vector to the curve $\gamma(\boldsymbol{s})$*

With this in our pockets, we can finally define our invariant manifold:

**Definition 3 (LCS)** *A **Lagrangian Coherence Structure** is a ridge in the FTLE field.*

Of course, we haven't actually proven that this structure is invariant. However, in the two-dimensional case, Shawn Shadden in his 2005 paper proved that the flux across the manifold is controlled by two terms: a term measuring how well defined the ridge is and a term $\mathcal{O}\left(\frac{1}{|T|}\right)$ [3]. The flux goes to zero as we take $|T|$ large and have a very refined velocity field. Thus we have invariance up to whatever numerical accuracy that we desire.

# The Computation

In applications, generally one doesn't use the definition of ridge to find the invariant manifolds. It is much easier to just compute the FTLE field and extract the ridges by 'eye'. In this project, the computation was performed in MATLAB and the results were plotted in Tecplot 360. To perform this calculation on the computer you need to choose a few parameters:

- $\mathbf{F}(\mathbf{x}, t)$ a the time-depedent flow field on $D \in \mathbb{R}^2$
- Choice of spatial grid resolution
- Choice of temporal resolution (the number of slides you want to see the LCS evolve over)

- Choice of the T (which can be positive or negative)

The spatial grid (or mesh) is each point of $D$ for which we want to compute the FTLE. Since the LCS can be computed from experimental data, $\mathbf{F}(\mathbf{x}, t)$ does not have to be defined for every point in $D$ or even for all time. However, if it is not fine enough then an interpolation algorithm can be used to fill in the missing values, such as the algorithm proposed in [2]. In the computation of the derivative, a simple finite-difference approach is taken. The results can then be put into a plot-able format, such as Tecplot, and then outputted into a movie or slide.

In this project, the two velocity fields considered are the time-depedent double gyre (which is periodic in time) and the time-dependent pendulum (which has non-periodic forcing). These velocity fields are analytic and thus are defined for all time and space, which means no interpolation algorithm is needed. The equations describing the double gyre are:

$$\begin{aligned}
\dot{x} &= -\pi A \sin\left(\pi f\left(x,t\right)\right) \cos\left(\pi y\right) \\
\dot{y} &= \pi A \cos\left(\pi f\left(x,t\right)\right) \sin\left(\pi y\right) \frac{\partial f}{\partial x} \\
f\left(x,t\right) &= \epsilon \sin\left(\omega t\right) x^2 + x - 2\epsilon x \sin\left(\omega t\right)
\end{aligned}$$

This equation was studied in [3] and in this project was used as a test of the algorithm used. In this project $A = 0.1, \omega = \frac{2\pi}{10}, \epsilon = 0.25$. The following figure shows the LCS generated for the double-gyre.

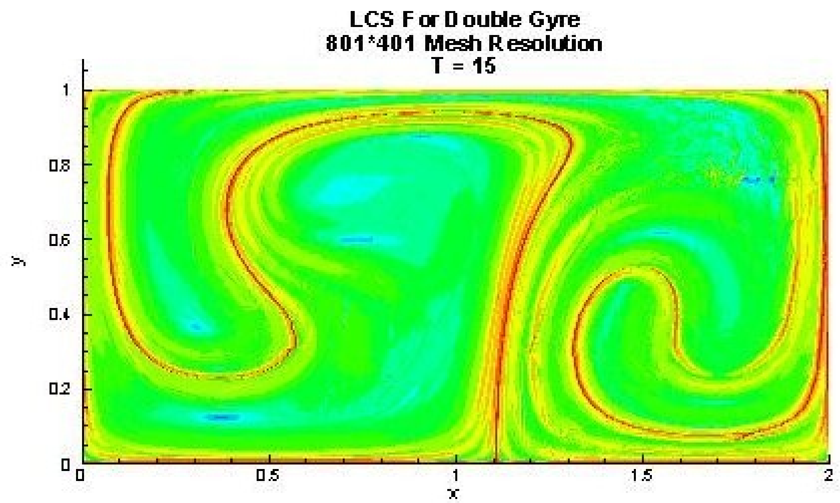


Figure 3: The LCS of the double gyre.

The picture here agree with those found in [3]. The invariantness of the LCS can also be tested by following the evolution of a tracer that was placed 'on' the LCS (this tracer was chosen by 'eye' to be on the manifold, and thus is just very close to the manifold). The following figures show the first and last slide of the movie generated by following the evolution of this tracer. It demonstrates that up to numerical accuracy our LCS is invariant.

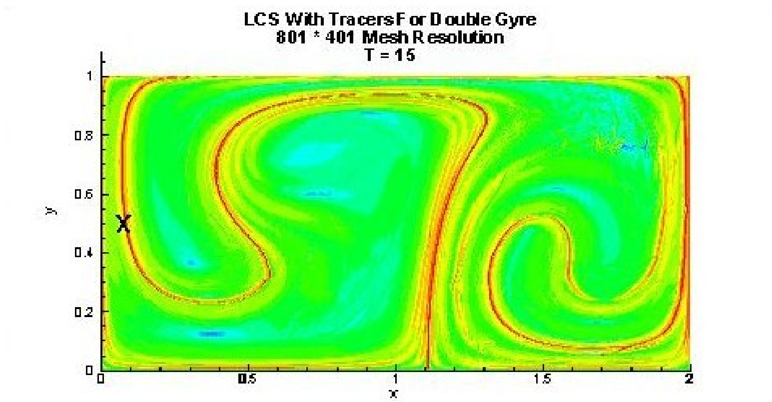


Figure 4: Following a tracer on the LCS.

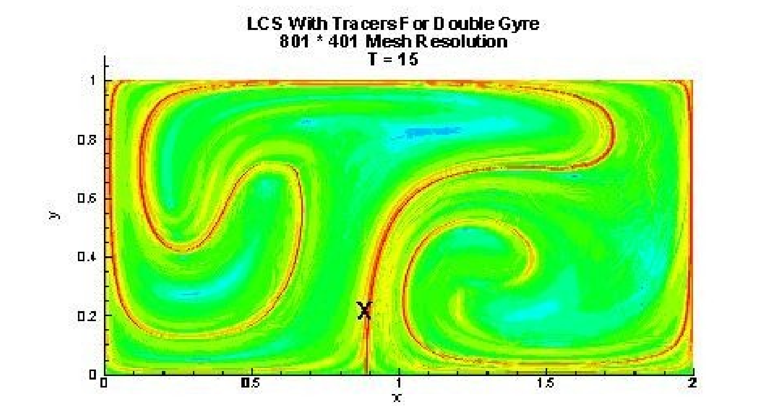


Figure 5: The end result of following the tracer.

The second time-dependent field looked at was the time-depedent pendulum (the simple pendulum with a time-dependent forcing applied). It is described by the following equations:

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -\sin(x) + f(t) \\ f(t) &= A \exp\left(\frac{t}{8}\right) \sin\left(\frac{2\pi t}{3}\right) \end{aligned}$$

In this project, $A = 1$. The following figure show the LCS generated for the time-dependent pendulum.

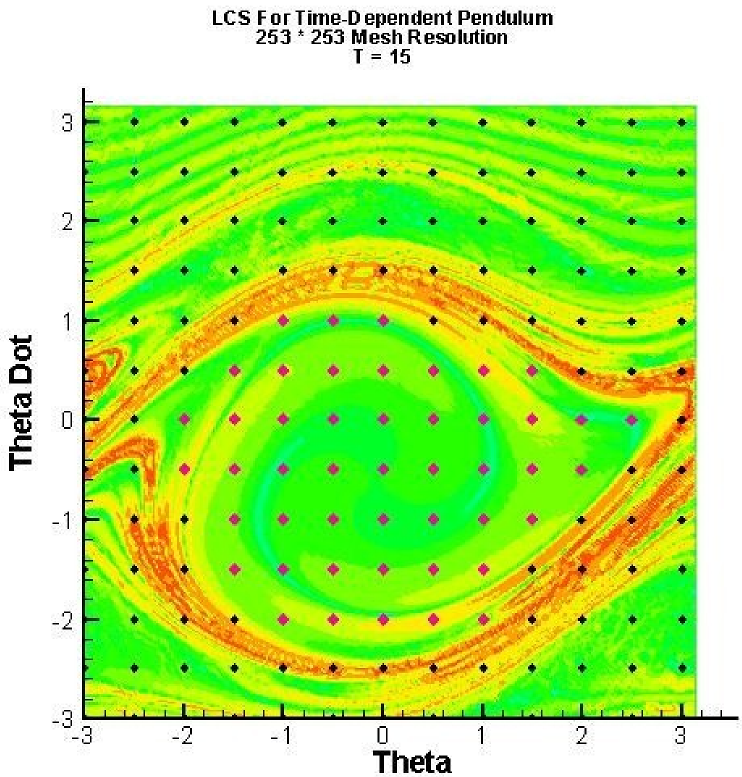


Figure 6: The LCS for the time-dependent pendulum with tracers.

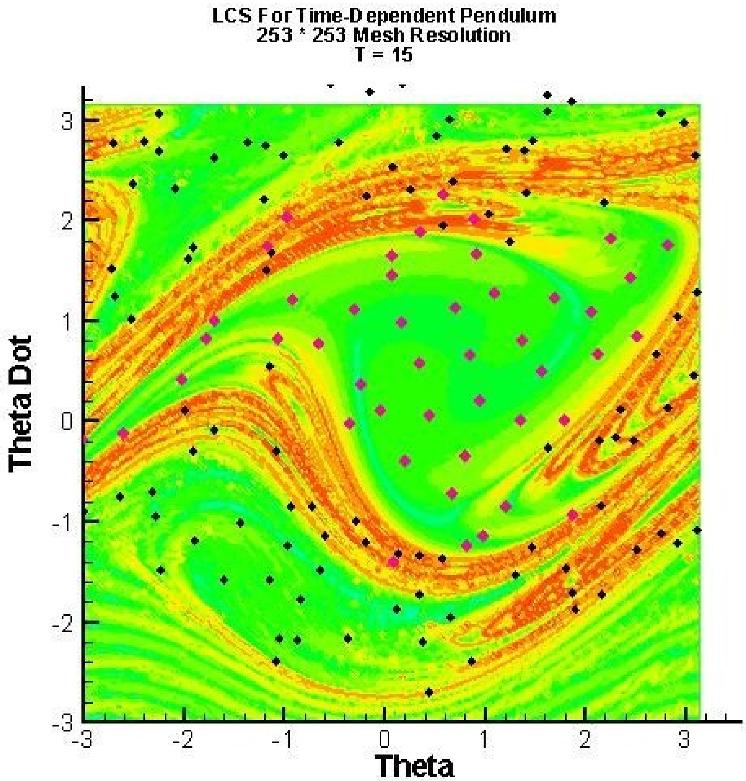


Figure 7: The end result of following the tracers for the time-dependent pendulum.

Following the evolution of tracers in this domain, one can see how the LCS controls the flow of particles. This leads to a subject called 'lobe dynamics', which is interesting in its own right, but not studied here. For more references to this subject, please consult [1].

# Changing the Parameters

There are two parameters that directly influence the actual calculation of the LCS in this project. These are the mesh resolution and the parameter T (the temporal resolution is not important in the calculation of the LCS, but in the calculation of the flow, which for this project is irrelevant since the flow is analytical).

The mesh resolution is clearly a major factor in the determination of the LCS. Starting at the use of finite-differences and continuing to the actual determination of where the LCS is, the higher the mesh resolution the better the accuracy. However, the higher the mesh resolution the more computational time it takes to compute the LCS. So the question becomes: how robust is the algorithm to a coarse mesh resolution? This question was explored using the double-gyre, by considering multiple mesh resolutions while holding all other paramters fixed.

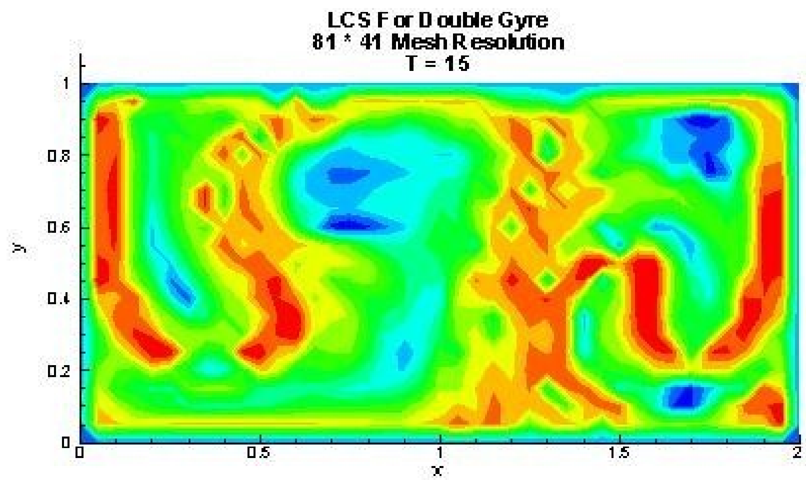


Figure 8: Low mesh resolution.

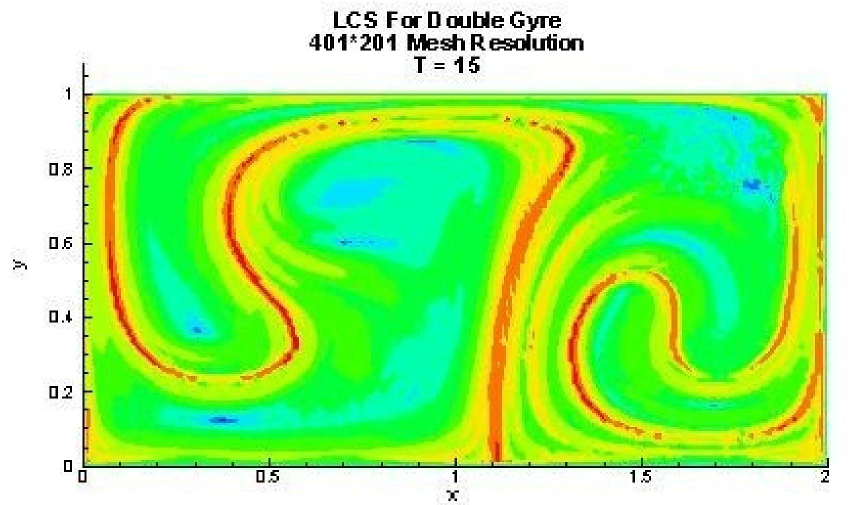


Figure 9: Medium mesh resolution.

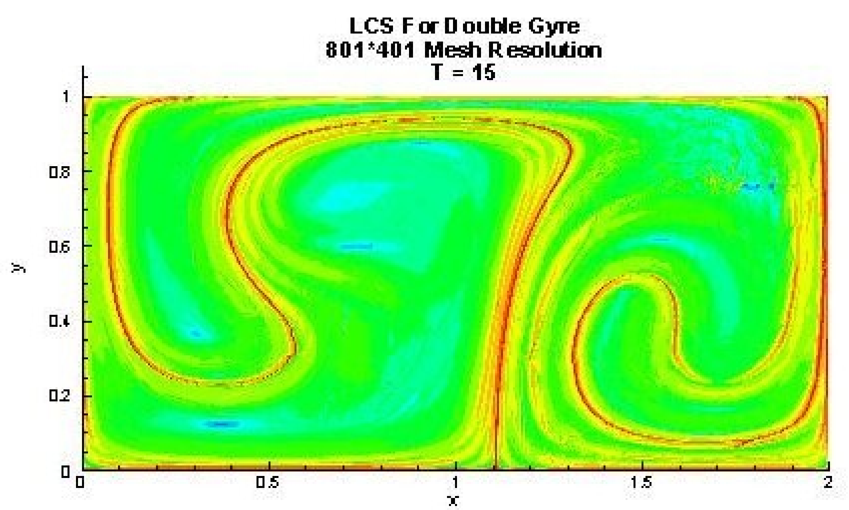


Figure 10: High mesh resolution.

The surprising thing is that even for very coarse resolutions, the general outline of the structures are visible. This suggests that better algorithms could reduce computational time by calculating a coarse resolution LCS and then focusing in on those areas that have a high FTLE and recomputing the LCS using a higher mesh resolution.

The time parameter is perhaps the most important aspect of the FTLE. There are many factors that determine how it should be chosen, but the first and foremost consideration is how the velocity data changes. If there is any temporal event that turns off (or on) after a certain finite time, the computation of the FTLE shouldn't use this data at a time when the event is not occuring. Another consideration, $T$ should be large enough for tracers in the field to have time to escape from the area that we are considering (if the tracers are going to escape). Finally, $T$ should be chosen large enough so that the LCS is indeed invariant (according to [3]). Now, in this project, $T$ can be chosen as large as we like since the velocity field is either periodic or has a single time-dependent event. However, again we are limited by the computational power of the computer. The higher the magnitude of $T$, the longer the flow has to be calculated which bogs down the calculation time. Also, if the magnitude of $T$ is too great and the spatial resolution is too coarse, then the derivation of the FTLE breaks down since the $\mathcal{O}\left(\|\delta \mathbf{x}\left(t_0\right)\|^2\right)$ is no longer negligible in this case [3]. So in this case the question becomes how robust is the algorithm to a small magnitude $T$? Two cases are considered for the double gyre: a low and high spatial resolution with different values of $T$. The high resolution case:

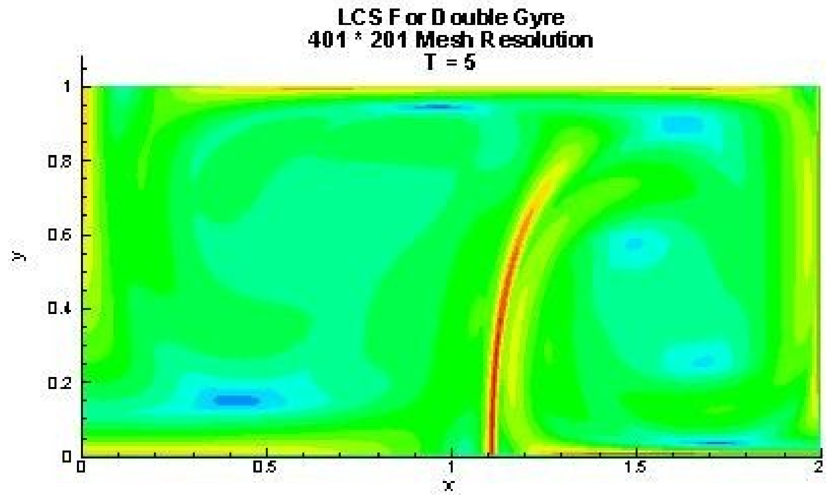


Figure 11: Small time.

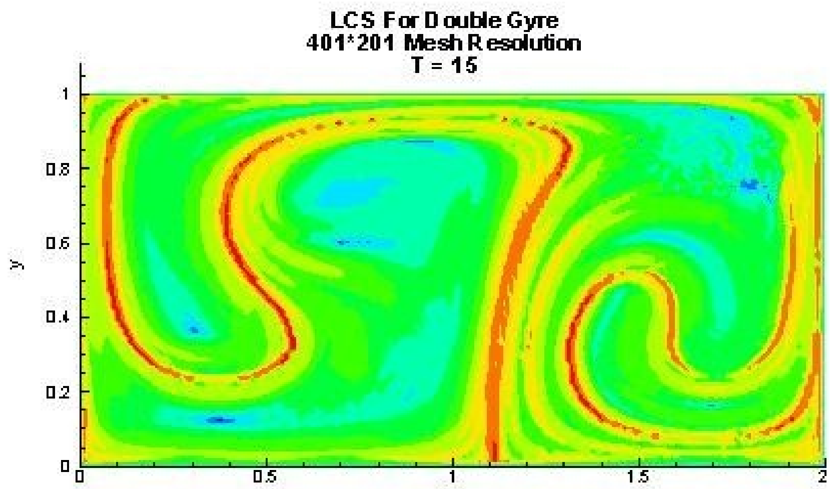


Figure 12: Medium time.

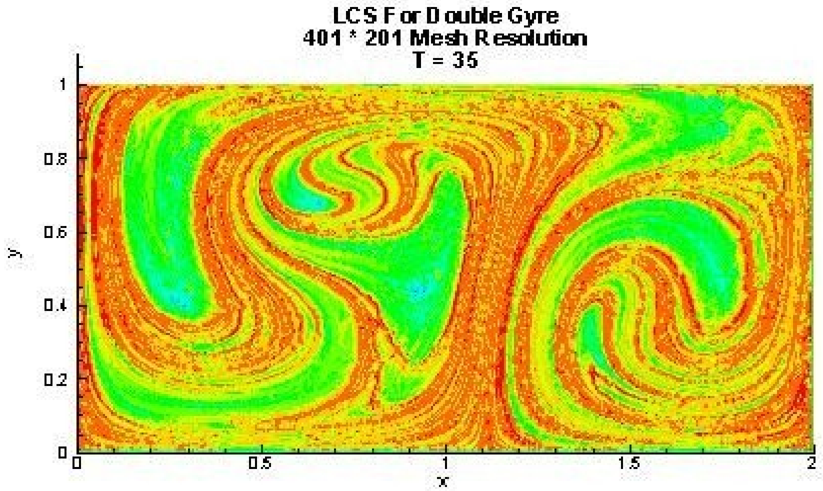


Figure 13: Large time.

For a high resolution mesh with a high magnitude $T$, there are many more structures visible. However, the drawback is the computational time. The $T = 5$ case took about 1 hour to run while the $T = 35$ case took about two full days to calculate. It then becomes important to see how low we can set the mesh resolution for a certain $T$ to still get significant results.

For the case of a low mesh resolution, the case where there is a low mesh resolution with a high magnitude $T$ results in disaster. However, for a low spatial and time resolution, the results seem to indicate where the fastest forming LCS is. This suggests that when determining the mesh resolution and magnitude $T$, it is important to have them matching to get the best results.

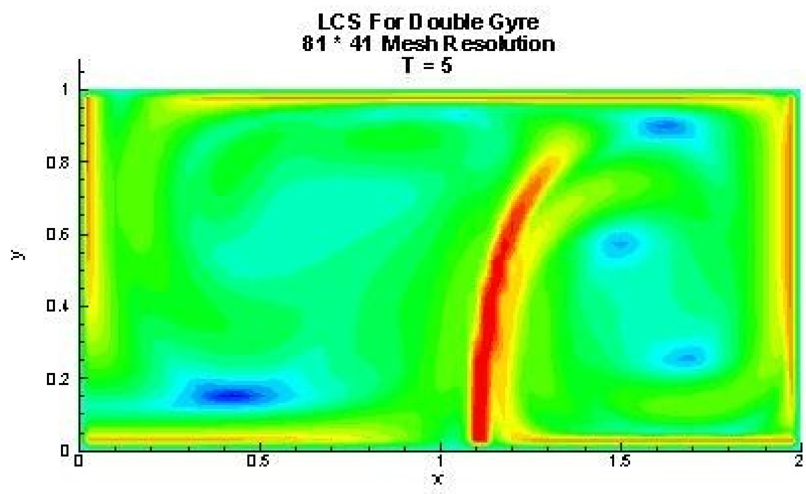


Figure 14: Small time.

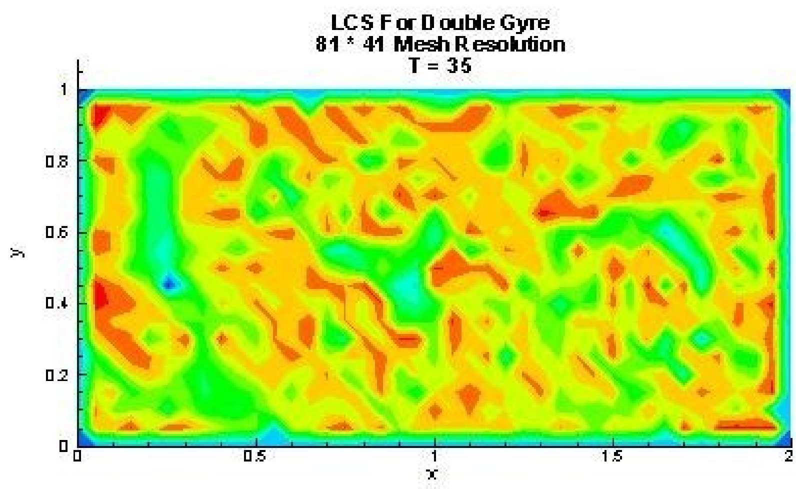


Figure 15: Large time.

## Future Directions

This project just worked on the basic theory of LCS as well as implementing a basic calculation in MATLAB. However, working through these examples shows that there is room for improvements. The first step is to create an adaptive algorithm that computes a coarse LCS and zooms in on interesting areas. This will save computational time as well as increase the accuracy of the actual structures. The second step is to consider averaging methods in conjunction with the LCS calculation. The idea here is that the LCS depends on the average of the velocity field (since it depends on the final locations of particles after being advected by the flow) so there might be computational savings by (for each slide) averaging the velocity field by $T$ and using this averaged equation to determine the flow function. Even if this approach does not yield computational savings, it might be useful in more theoretical applications as it reduces the time-dependence of the velocity field. This idea actually motivates another, since in this case we have removed the time-dependence from the velocity field, meaning that methods of finding the invariant manifolds for autonomous systems might be applicable here. These ideas will be explored in future research.

## References

[1] Michael Dellnitz, Oliver Junge, Wang Sang Koon, Francois Lekien, Martin W. Lo, Jerrold E. Marsden, Kathrin Padberg, Robert Preis, Shane D. Ross, and Bianca Thiere, *Transport in dynamical astronomy and multibody problems*, Internat. J. Bifur. Chaos Appl. Sci. Engrg. **15** (2005), no. 3, 699–727. MR MR2136742 (2006a:70038)

[2] F. Lekien and J. Marsden, *Tricubic interpolation in three dimensions*, Internat. J. Numer. Methods Engrg. **63** (2005), no. 3, 455–471. MR MR2144677 (2006b:86003)

[3] 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*, Phys. D **212** (2005), no. 3-4, 271–304. MR MR2187513