Fast Computation of FTLE Fields for Unsteady Flows: A Comparison of Methods

Steven L. Brunton & Clarence W. Rowley

This paper presents new, efficient methods for computing finite-time Lyapunov exponent (FTLE) fields in unsteady flows. The methods approximate the particle flow map to eliminate redundant particle integrations in neighboring flow map calculations. Two classes of flow map approximations are investigated based on composition of intermediate flow maps; unidirectional approximation constructs a time-T map by composing a number of smaller time-h maps, while bidirectional approximation constructs a flow map by composing both positive and negative-time maps. The unidirectional method is shown to be fast and accurate, although it is memory intensive. The bidirectional method is also fast and uses significantly less memory; however, it is prone to large error which is aligned with the opposite-time Lagrangian coherent structures. The algorithms are implemented and compared on three example fluid flows: the double gyre, a low Reynolds number pitching flat plate, and unsteady ABC flow.

Lagrangian coherent structures (LCS) are hyperbolic material lines or surfaces that provide a useful analogue of invariant manifolds for unsteady flow fields. LCS are often determined as ridges of the field of finite-time Lyapunov exponent (FTLE) that satisfy an additioanl hyperbolicity criterion. However, FTLE fields are expensive to compute due to the large number of particle trajectories which must be integrated to construct a particle flow map. Moreover, it is often necessary to compute a sequence of FTLE fields in time to visualize unsteady events. The methods presented here streamline the computation of a sequence of FTLE fields by removing redundant trajectory integrations between neighboring particle flow maps, which are necessary to compute the FTLE field. There are two categories of methods which approximate the particle flow map. The unidirectional method composes intermediate flow maps of the same time direction, and the bidirectional method composes intermediate flow maps of opposite time directions. It is shown that the unidirectional method is both fast and accurate, providing orders of magnitude computational savings over the standard method, when computing a sequence of FTLE fields in time to visualize the coherent structures of an unsteady flow.

I. INTRODUCTION

Coherent structures are important for understanding and modeling the underlying physical mechanisms of complex fluid flows [1]. In particular, Lagrangian coherent structures (LCS) are defined using particle trajectories and are Galilean-invariant, unlike Eulerian criteria [2]. LCS are hyperbolic material lines or surfaces [3], and ridges of the finite-time Lyapunov exponent (FTLE) field provide candidate material lines. Ridges of the FTLE field are LCS if and only if the Lagrangian rate of strain is nonzero along the ridge, distinguishing true hyperbolic material lines from regions of high shear [3]. A ridge of the FTLE field can refer to either a curvature or second derivative ridge, although the latter is more convenient for practical computation [4]. LCS extend the notion of invariant manifolds from dynamical systems theory to unsteady flows. FTLE fields provide a measure of the stretching between nearby particles in a given flow and are important in determining transport mechanisms and separatrices in unsteady flows.

The theory and computation of finite-time Lyapunov exponents (FTLE), also known as direct Lyapunov exponents (DLE), is a relatively modern development [3, 4], with extensions to 3-dimensional [5, 6] and n-dimensional [7] flows. FTLE analysis has been widely applied in a number of branches of fluid mechanics, including fluid transport [8–10], bio-propulsion [11–13], flow over airfoils [14–16], plasmas [17], and geophysical flows [18, 19].

Because FTLE analysis is particularly useful for unsteady flows, it is often necessary to compute a sequence of FTLE fields in time to visualize an unsteady event. As flows become more complex, computations become increasingly expensive. In particular, FTLE calculations are expensive because a large number of particle trajectories must be integrated in order to obtain a particle flow map, often from stored velocity fields. When computing a sequence of FTLE fields in time, it is possible to speed up the computation considerably by eliminating redundant particle integrations. One approach that has been developed uses adaptive mesh refinement to reduce the number of integrations [20–23].

The approach here is to construct an approximate flow map by composing intermediate flow maps from FTLE field calculations at neighboring times. The first class of flow map approximation, denoted bidirectional composition, constructs a flow map by composing intermediate flow maps which are aligned in both positive and negative-time. The second class, denoted unidirectional composition, composes intermediate flow maps which are all aligned in the same time direction. The methods are

compared using analytic estimates for accumulated error and computation time as well as benchmarks on a number of example flows.

A. Main Results

In this paper we demonstrate that the unidirectional method is both fast and accurate, although it requires significantly more memory than the bidirectional method. Orders of magnitude speed-up may be achieved over the standard method, and computational improvement scales with the desired time resolution of the FTLE animation.

The bidirectional method suffers from significant error. In particular, the errors in the positive-time LCS (pLCS) align with the negative-time LCS (nLCS) and vice versa. To understand this coherent error, we provide an error analysis for both methods, and uncover an important relationship between the pLCS and nLCS, which correspond to finite-time unstable and stable manifolds, respectively. In particular, in the neighborhood of a time-dependent saddle, particles near the pLCS flow into particles near the nLCS in positive time.

II. STANDARD COMPUTATION OF FTLE

Consider a time-dependent velocity field $\mathbf{u}$ on $\mathbb{R}^n$ and a particle trajectory $\mathbf{x}(t)$ which satisfies

$$\dot{\mathbf{x}} = \mathbf{u}\left(\mathbf{x}(t), t\right). \tag{1}$$

The velocity field, $\mathbf{u}$, may be an unsteady solution of the Navier-Stokes equation, although it is only assumed that $\mathbf{u}$ is at least C^0 in time and C^1 in space. However, to extract Lagrangian coherent structures from the Hessian of the FTLE field, $\mathbf{u}$ must be C^2 in space [4]. The velocity field may be analytically defined, but is more often obtained from experiments or direct numerical simulation which produce velocity field data at discrete snapshots over a finite range of time. A method of computing finite-time Lyapunov exponents (FTLE) on a finite amount of discrete velocity field data has been developed [3, 4].

Computing an FTLE field typically involves four steps. First, a grid of particles $X_0 \subset \mathbb{R}^n$ is initialized over the domain of interest. The particles are advected (i.e., integrated) with the flow from initial time 0 to final time T, resulting in a time-T particle flow map, Φ_0^T, defined as:

$$\Phi_0^T : \mathbb{R}^n \to \mathbb{R}^n; \quad \mathbf{x}(0) \mapsto \mathbf{x}(0) + \int_0^T \mathbf{u}(\mathbf{x}(\tau), \tau) d\tau. \tag{2}$$

Next, the flow map Jacobian, $\mathbf{D}\Phi_0^T$ is computed, usually by finite-differencing, to obtain the Cauchy-Green deformation tensor,

$$\Delta = \left(\mathbf{D}\Phi_0^T\right)^* \mathbf{D}\Phi_0^T \tag{3}$$

where * denotes transpose. Finally, the largest eigenvalue, $\lambda_{\max}$, of this symmetric tensor is extracted and synthesized into an FTLE field:

$$\sigma(\Phi_0^T; \mathbf{x_0}) = \frac{1}{|T|} \log \sqrt{\lambda_{\max}(\Delta(\mathbf{x_0}))}. \tag{4}$$

The bottleneck in this procedure is the large number of particle integrations required to obtain the particle flow map, Φ_0^T. Moreover, if the velocity field is time-varying, it is necessary to compute a sequence of FTLE fields in time to visualize unsteady events, as shown schematically in Fig. 1.

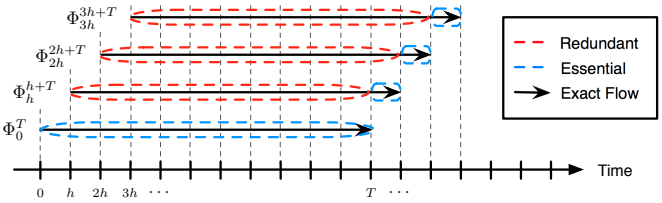


FIG. 1: The standard method for computing a sequence of FTLE fields. Flow maps Φ_{kh}^{kh+T} for $k \in \{0, 1, 2, 3\}$ are shown (solid black arrow). Essential (blue) and redundant (red) particle integrations are outlined in dashed ovals.

III. FLOW MAP APPROXIMATION

As seen in Fig. 1, the standard method of computing a sequence of FTLE fields involves inefficient re-integration of particles. The unidirectional and bidirectional methods outlined below streamline the computation of neighboring FTLE fields by approximating the time-T flow map, $\Phi_{t_0}^{t_0+T}$, which can be written as:

$$\Phi_{t_0}^{t_0+T} = \Phi_{t_{N-1}}^{t_N} \circ \cdots \circ \Phi_{t_1}^{t_2} \circ \Phi_{t_0}^{t_1} \tag{5}$$

where $t_N = t_0 + T$.

Because the flow maps are obtained numerically on a discrete grid of points, $X_0 \subset \mathbb{R}^n$, it is necessary to interpolate the maps at points $\mathbf{x} \notin X_0$. Consider a flow map $\Phi : \mathbb{R}^n \to \mathbb{R}^n$, and the same flow map restricted to X_0, $\Phi|_{X_0} : X_0 \to \mathbb{R}^n$. The interpolation operator $\mathcal{I}$ takes the discrete map $\Phi|_{X_0}$ and returns the interpolated map, $\mathcal{I}\Phi : \mathbb{R}^n \to \mathbb{R}^n$, which approximates Φ on $\mathbb{R}^n$:

$$\mathcal{I} : \Phi|_{X_0} \mapsto \mathcal{I}\Phi. \tag{6}$$

Here we use the shorthand $\mathcal{I}\Phi \triangleq \mathcal{I}\left(\Phi|_{X_0}\right)$. We now obtain an approximation to the flow map in Eq. (5):

$$\begin{aligned} \tilde{\Phi}_{t_0}^{t_0+T}(X_0) &= \mathcal{I}\Phi_{t_{N-1}}^{t_N} \circ \cdots \circ \mathcal{I}\Phi_{t_1}^{t_2} \circ \Phi_{t_0}^{t_1}(X_0) \\ &\approx \Phi_{t_0}^{t_0+T}(X_0) \end{aligned} \tag{7}$$

The bidirectional method approximates the time-T flow map $\Phi_{t_0}^{t_0+T}$ by first integrating backward to a reference

time, $t = 0$, then interpolating forward through a previously computed time-T map, Φ_0^T, and finally integrating forward to time $t_0 + T$. The unidirectional method approximates the time-T flow map by composing a number of smaller time flow maps, $\Phi_{t_i}^{t_i+h}$, which all have the same time direction. Additionally, the chain rule may be applied to each of the methods, resulting in an approximation to the flow map Jacobian, $\mathbf{D}\Phi_{t_0}^{t_0+T}$.

A. Bidirectional Composition

Bidirectional approximation eliminates redundancy from neighboring FTLE field computations by using the information from a known flow map at a given time, Φ_0^T, to calculate an approximation to the flow map at future times, $\Phi_{t_0}^{t_0+T}$. First, X_0 is integrated backward from t_0 to the reference time 0. The distorted grid $\Phi_{t_0}^0(X_0)$ is then flowed forward through the interpolated map, $\mathcal{I}\Phi_0^T$, and finally integrated forward an amount t_0 to the desired time $t_0 + T$:

$$\Phi_{t_0}^{t_0+T} \;=\; \Phi_T^{t_0+T} \circ \mathcal{I}\Phi_0^T \circ \Phi_{t_0}^0. \tag{8}$$

The flow Φ_0^T is stored as a reference solution to compute an approximation to the flow map at later times $\tilde{\Phi}_{kh}^{kh+T} \approx \Phi_{kh}^{kh+T}$ by

$$\tilde{\Phi}_{kh}^{kh+T} = \Phi_T^{kh+T} \circ \mathcal{I}\Phi_0^T \circ \Phi_{kh}^0 \qquad k \in \mathbb{Z} \tag{9}$$

This is referred to as bidirectional method (a), and is shown in Fig. 2:

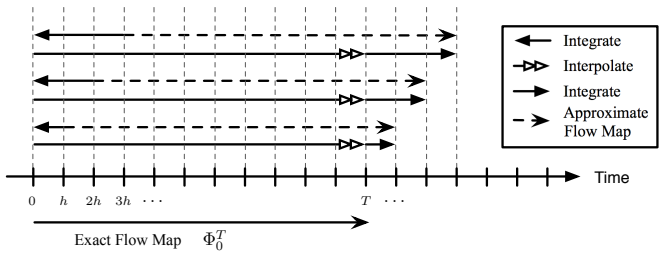


FIG. 2: Schematic for bidirectional method (a). Given a known flow map Φ_0^T (solid black arrow), it is possible to approximate the flow map at later times $\tilde{\Phi}_{kh}^{kh+T}$ (dashed black arrow) by integrating backward in time to $t = 0$ (red arrow), flowing forward through the interpolated map $\mathcal{I}\Phi_0^T$ which was already computed (blue double arrow), and integrating trajectories forward to the correct final time (green arrow).

Instead of using Φ_0^T as the reference solution for every future time, it is convenient to use the new approximate flow map $\tilde{\Phi}_{kh}^{kh+T}$ as the reference solution for the next iteration, $\tilde{\Phi}_{(k+1)h}^{(k+1)h+T}$:

$$\tilde{\Phi}_{(k+1)h}^{(k+1)h+T} = \Phi_{kh+T}^{(k+1)h+T} \circ \mathcal{I}\tilde{\Phi}_{kh}^{kh+T} \circ \Phi_{(k+1)h}^{kh}. \tag{10}$$

Errors will compound more quickly since approximate flow maps are used as the reference solutions for later approximations. However, fewer total integration steps are required, since the reference map advances with every iteration. This is referred to as bidirectional method (b), and is shown in Fig. 3.

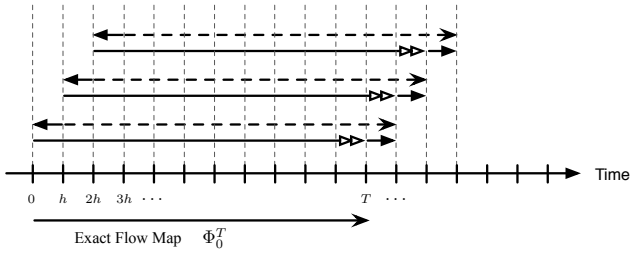


FIG. 3: Schematic for bidirectional method (b). As in Fig. 2, a known flow map (solid black arrow) is used to approximate the flow map at a later time $\tilde{\Phi}_{kh}^{kh+T}$ (dashed black arrow). The approximate flow map is used as the known map for the next step (dashed black arrow).

B. Unidirectional Composition

The basis of the unidirectional method is to eliminate redundant particle integrations by only integrating particle trajectories through a given velocity field a single time. If a sequence of FTLE snapshots is desired at a time spacing of h, for example as frames in an animation, then it is convenient to break up the time-T flow map into smaller time-h flow maps, where $T = kh$:

$$\tilde{\Phi}_0^{kh} \;=\; \mathcal{I}\Phi_{(k-1)h}^{kh} \circ \cdots \circ \mathcal{I}\Phi_h^{2h} \circ \Phi_0^h \tag{11}$$

This method is called unidirectional because particle flow maps of the same time direction are used, as opposed to the bidirectional method which composes both positive-time and negative-time flow maps.

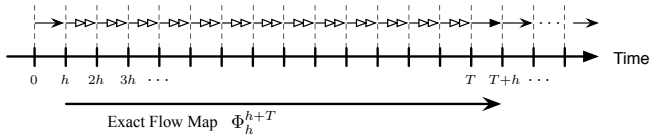


FIG. 4: Schematic for unidirectional method. Time-h flow maps (short blue arrows) are stored and composed to approximate the time-T flow map (long black arrow). The next flow map only requires integrating one new time-h flow map (green arrow).

The simplest approach is to compute a number of time-h flow maps and store them in memory. Then, to construct an approximate $\Phi_{t_0}^{t_0+T}$, it remains only to compose

the sequence of interpolated time-h flow maps. The next iteration involves integrating one more time-h flow map and composing the next sequence, as in Fig. 4.

To further improve efficiency by reducing the total number of flow map compositions, it is possible to construct a multi-tiered hierarchy of flow maps for reuse in neighboring flow map constructions. With enough memory, it is possible to reduce the number of interpolated compositions by increasing the number of tiers of flow maps, each tier being constructed as the composition of two of the flow maps in the next tier lower, as in Fig. 5.

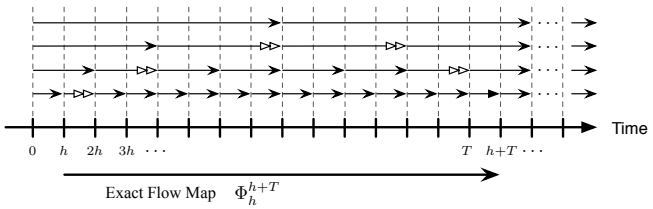


FIG. 5: Schematic for unidirectional method with multiple tiers. The bottom tier of time-h flow maps is computed as in Fig. 4. Pairs are composed to form the next tier of time-$2h$ flow maps, and so on. This method requires more storage, but fewer total compositions when computing a series of FTLE fields for an animation.

C. Chain Rule of Compositions

As seen in Eq. (3), once the flow map $\Phi_{t_0}^{t_0+T}$ is obtained, it is necessary to compute the flow map Jacobian in order to extract the FTLE. Applying the chain rule to Eq. (5), it is possible to express the flow map Jacobian as a product of the Jacobians of intermediate flow maps:

$$\begin{aligned}\mathbf{D}(\Phi_{t_0}^{t_N})(\mathbf{x}) &= \mathbf{D}\left(\Phi_{t_{N-1}}^{t_N}\circ\cdots\circ\Phi_{t_1}^{t_2}\circ\Phi_{t_0}^{t_1}\right)(\mathbf{x}) \\ &= \mathbf{D}\Phi_{t_{N-1}}^{t_N}\left(\Phi_{t_0}^{t_{N-1}}(\mathbf{x})\right)\times\cdots\times\mathbf{D}\Phi_{t_0}^{t_1}(\mathbf{x})\end{aligned} \tag{12}$$

Applied to the bidirectional methods, this yields:

$$\begin{aligned}\Phi_h^{h+T} =& \Phi_T^{h+T}\circ\Phi_0^T\circ\Phi_h^0 \\ \implies \quad \mathbf{D}\Phi_h^{h+T}(\mathbf{x}) =& \mathbf{D}\Phi_T^{h+T}\left(\Phi_0^T\circ\Phi_h^0\right)(\mathbf{x})\times \\ & \times\mathbf{D}\Phi_0^T\left(\Phi_h^0\right)(\mathbf{x})\circ\mathbf{D}\Phi_h^0(\mathbf{x}),\end{aligned} \tag{13}$$

and applied to the unidirectional methods, this yields:

$$\begin{aligned}\Phi_0^T =& \Phi_{T-h}^T\circ\cdots\circ\Phi_h^{2h}\circ\Phi_0^h \\ \implies \quad \mathbf{D}\Phi_0^T(\mathbf{x}) =& \mathbf{D}\Phi_{T-h}^T\left(\Phi_0^{T-h}(\mathbf{x})\right)\times\cdots \\ & \cdots\times\mathbf{D}\Phi_h^{2h}\left(\Phi_0^h(\mathbf{x})\right)\times\mathbf{D}\Phi_0^h(\mathbf{x}).\end{aligned} \tag{14}$$

IV. COMPARISON OF METHODS

Each method from Section III is implemented and tested on three example problems: the periodic double gyre, 2D flow over a pitching flat plate at Reynolds number 100, and 3D unsteady ABC flow. These examples are chosen because they cover a range of features including 2D and 3D vector fields, which are either defined analytically or obtained from data files from DNS on either open, closed, or periodic domains. Each example problem is discussed more in Appendix B, including details such as how the velocity field is defined, and on what domain. In the pitching plate example, velocity field snapshots are all loaded up-front before applying the methods.

Table I summarizes the results comparing each method on the three example fluid flows. In each comparison, the standard, unidirectional and bidirectional methods are used to compute a sequence of FTLE fields which are frames in an unsteady animation. The flow map duration used to compute an FTLE field is T, and the time-spacing between neighboring FTLE fields is h, so the number of animation frames per flow map duration is T/h. As demonstrated in Section IV B, this is an upper bound on the speed-up of the unidirectional method.

In each comparison, the unidirectional method is accurate and offers the greatest speed-up over the standard method. However, it also requires more memory than any other method. The bidirectional method is fast and uses less memory than the unidirectional method, but is prone to large errors in the approximate flow map and does not accurately reproduce the FTLE field.

Contour plots of the FTLE fields computed after a number of iterations of each method are shown in Fig. 6. The FTLE fields computed with the unidirectional method agree with those computed using the standard method, as seen by comparing the first and second columns of Fig. 6. FTLE fields computed using the bidirectional method, shown in the third column, have large errors. It is interesting to note that these errors are aligned with coherent structures found in the opposite-time FTLE field, shown in the fourth column. An analysis of this coherent error is provided in Section V.

A. Example - Double Gyre

Figure 7 shows the L_2 and L_∞ error of the forward-time FTLE field for the double gyre computed using the standard method with $T = 16$, as time-step Δt and grid spacing Δx are varied. At a given grid spacing, a reference FTLE field is computed using a sufficiently small time-step, $\Delta t = 10^{-4}$, so that the FTLE field may be considered exact. For small enough time-step $\Delta t \approx .001$, the FTLE field error converges. All integrations are performed using a fixed time-step, fourth order Runge-Kutta scheme.

The flow map approximation methods are only faster than the standard method when used to compute a sequence of FTLE fields in time, as in the construction of frames for a movie. Figure 8 compares computation time and L_2 error vs. frame number (iteration #) for a sequence of FTLE fields of the double gyre, computed us-

Problem	Resolution	T/h	Frames	Method	Memory (GB)	Speed-up	Accurate
Double Gyre	1024×512	15	30	Standard	.05	1	Yes
				Unidirectional	.36	10	Yes
				Bidirectional	.14	6.2	No
Pitching plate	1024×512	15	30	Standard	.48	1	Yes
				Unidirectional	.70	8.2	Yes
				Bidirectional	.50	5.4	No
Pitching plate	600×300	150	192	Standard	.48	1	Yes
				Unidirectional	1.8	67	Yes
				Bidirectional	.48	54	No
ABC flow	128^3	20	40	Standard	.48	1	Yes
				Unidirectional	2.6	6.8	Yes
				Bidirectional	.73	7.3	No

TABLE I: Comparison of methods on various examples fluid flows. The unidirectional method both fast and accurate, but requires more memory than the other methods, providing one or two orders of magnitude computational improvement over the standard method.

ing the standard, unidirectional, and bidirectional methods. Each iteration produces an FTLE field which is a single frame in an animation of the unsteady FTLE field. In this example, the flow map duration is $T = 16$, the time spacing between each FTLE field is $h = 1$, and the time-step of integration is $\Delta t = .01$. The multi-tier unidirectional method uses four tiers.

The first FTLE field takes roughly the same time to compute using each of the methods. However, for subsequent iterations, the unidirectional and bidirectional methods are significantly faster. The computation time of bidirectional method (a) increases with the number of iterations, k, because integrating back from $t = kh$ to the reference time $t = 0$ becomes more costly as k increases, as seen in Fig. 3. After $T/2h = 8$ iterations of bidirectional method (a), it is advantageous to compute a new reference flow map using the standard method. This explains the breaks in the solid red curve in part (b) of Fig. 8, as the bidirectional method is exact at these iterations. Bidirectional method (b) overcomes this increasing cost vs. iteration by using the flow map from the current iteration as the reference flow map at the next iteration. However, using an approximate flow map to compute the next approximation causes bidirectional method (b) to accumulate error more quickly than method (a). The unidirectional method is both the fastest and most accurate method in this comparison.

B. Computational Resources

Again, consider a sequence of time-T flow maps spaced h apart, as might be required for an unsteady visualization. When there are many integration time-steps of size Δt between each neighboring flow map, i.e. $\Delta t \ll h$, then the added cost of flow map composition becomes relatively small compared with the cost of integrating a time-h flow map.

All methods take about the same amount of time to compute the first FTLE field in the sequence. For subsequent iterations, the standard method involves $(T/h) \times (h/\Delta t)$ integration steps for each new FTLE field, whereas the unidirectional method only requires $h/\Delta t$ integration steps, and bidirectional method (b) requires $2h/\Delta t$ integration steps. Assuming $\Delta t \ll h$, the speed-up of the unidirectional method over the standard method will increase as the number of frames in the animation per flow map duration, T. In other words, as $\Delta t/h \to 0$, the computation of $\Phi_{t_0}^{t_0+T}$ using the unidirectional method is T/h times faster than using the standard method, and twice as fast as the bidirectional method.

In the examples above, all intermediate flow maps were stored in memory until no longer useful for future computations. Regardless of any parameters of the FTLE field animation, the standard and bidirectional methods must store a fixed number of flow maps. The standard method stores the single flow map $\Phi_{t_0}^{t_0+T}$, while the bidirectional method stores three maps: $\Phi_{t_0}^{0}$, Φ_{0}^{T}, and $\tilde{\Phi}_{t_0}^{t_0+T}$. The unidirectional method, however, stores every intermediate time-h flow map $\Phi_{(k-1)h}^{kh}$, of which there are T/h. Therefore, the memory requirement of the unidirectional method scales linearly with the upper-bound on its speed-up, T/h.

The memory usage of the unidirectional method scales with the dimension of the flow D, the spatial resolution R, and the possible computational speed up of the method S, given by T/h:

$$\text{Memory (GB)} \sim S \times D \times R^D \tag{15}$$

$$= \frac{8 \text{ B/double}}{1024^3 \text{ B/GB}} \times \frac{T}{h} \times D \times R^D \tag{16}$$

For example, a series of two dimensional, high-definition (1920×1080 resolution) FTLE fields may be computed

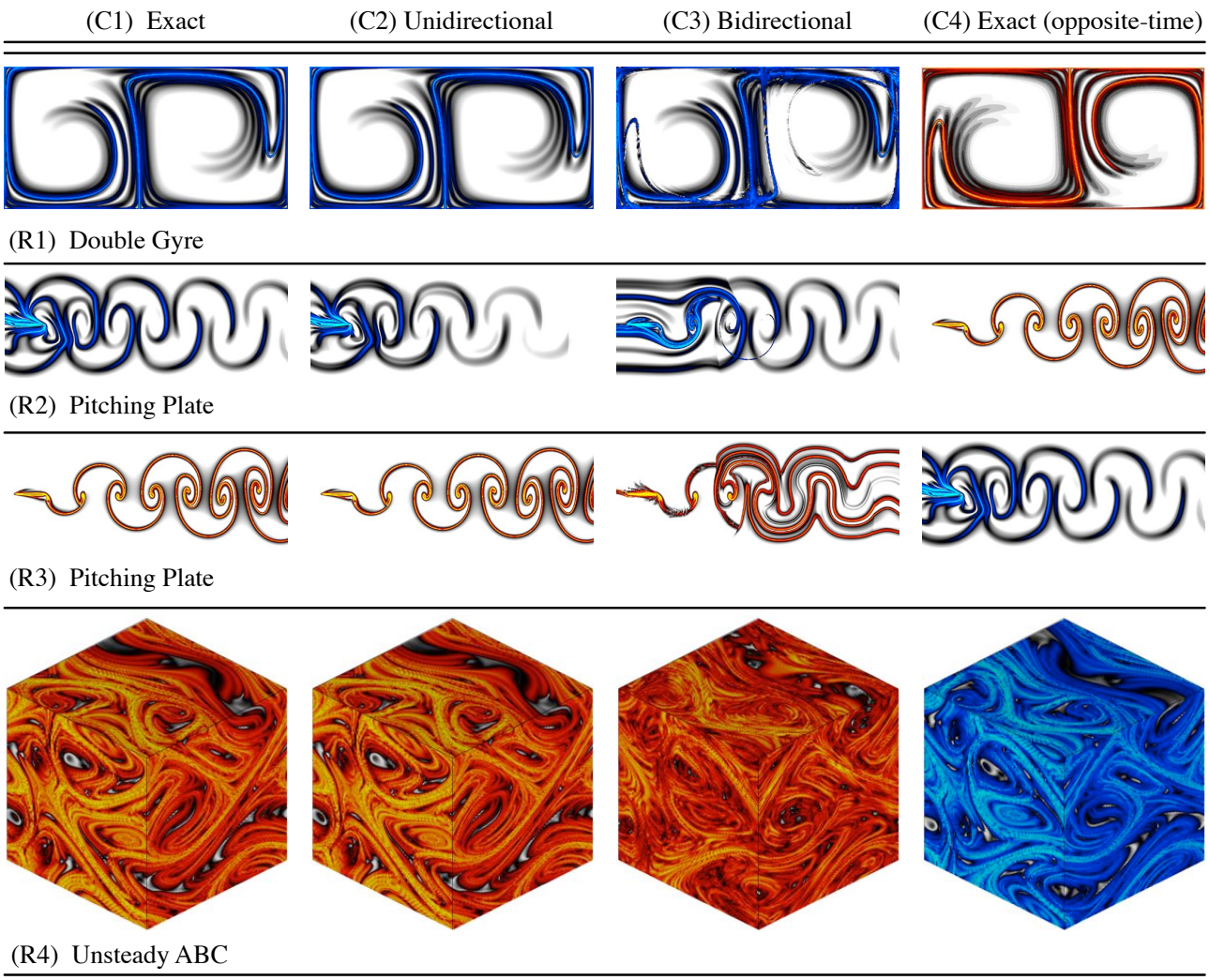


FIG. 6: Graphical comparison of each method on four examples: (top row) positive-time FTLE of double gyre, (second row) positive-time FTLE of 2D pitching plate, (third row) negative-time FTLE of 2D pitching plate, (bottom row) negative-time FTLE of 3D ABC flow. Each figure shows the FTLE field after a number of iterations of the given method. The number of iterations k was chosen so that $kh \approx T$ to magnify the effect of bidirectional error. The column of FTLE fields calculated using unidirectional composition, (C2), agree well with the exact FTLE fields computed using the standard method, (C1). The column of FTLE fields calculated using bidirectional composition, (C3) all have significant error which is aligned with the opposite-time coherent structures. The opposite-time FTLE fields are shown in the rightmost column, (C4), for comparison with the bidirectional method. FTLE fields computed for positive-time flow maps are blue and those computed for negative-time flow maps are red.

using the unidirectional method with up to 100× speed up using approximately 3.1 GB of RAM. A three dimensional FTLE field with resolution $512 \times 256 \times 64$ may be computed with up to 100× speed up with approximately 19 GB of RAM.

In the double gyre and ABC flow examples, the velocity field is defined analytically, according to Eqns. (B2) and (B5). Thus, in these two examples, the velocity field is calculated analytically at every time, and no velocity fields need to be stored in data files. However, in the pitching plate example, velocity fields are obtained from data files which are the output of a direct numerical simulation. Because loading velocity fields which are stored on disk is slow, it is important to minimize the number of file loads. In the pitching plate example, all of the velocity fields are loaded up-front and stored in memory throughout the computation. However, velocity fields are often too large to store them all in memory, for example in large 2D or 3D simulations, so that subsequent iterations of the methods require re-loading the same velocity

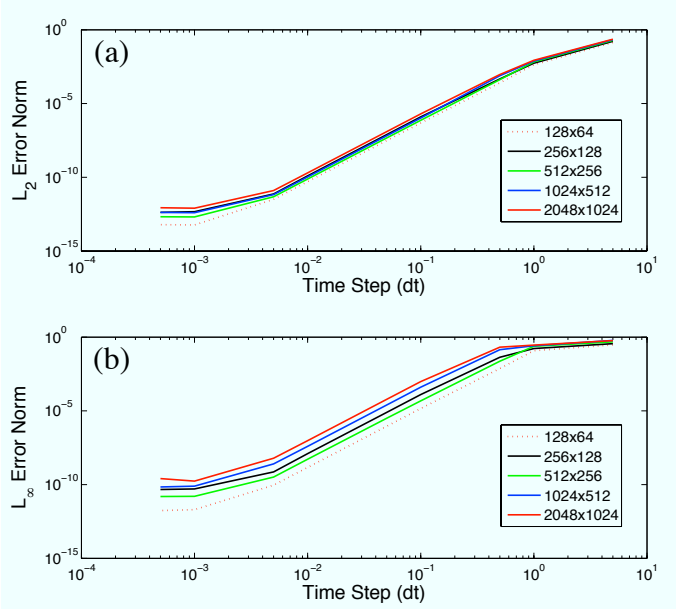


FIG. 7: Convergence tests for L_2 and L_∞ error of the FTLE field vs. integration step and grid spacing on double-gyre.

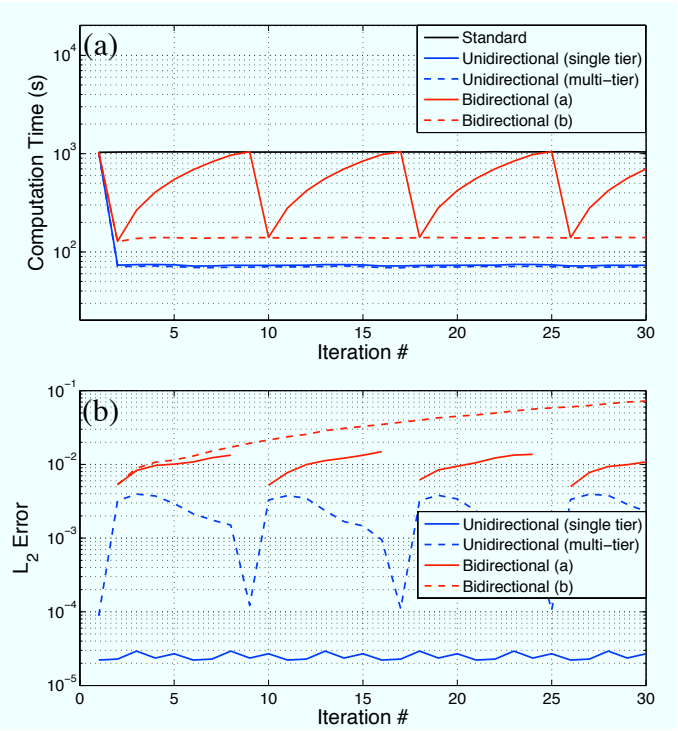


FIG. 8: Comparison of methods for computing the FTLE field of the double gyre with resolution 1024×512. Each method is iterated to compute a sequence of 30 FTLE fields in time. (a) Computational time vs. Iteration, and (b) L_2 error vs. Iteration.

field data from previous iterations. In practice, although loading data files is time consuming, it represents a fraction of the cost of particle integration.

V. ERROR ANALYSIS

This aim of this section is to explain why the method of unidirectional composition is accurate while bidirectional composition is prone to large errors. Moreover, why are the errors in the bidirectional method found in regions of high FTLE of the opposite-time flow map, as illustrated in the third and fourth columns of Fig. 6?

For a given particle in a flow, larger finite-time Lyapunov exponent indicates greater stretching between neighboring particles and more sensitive dependence on initial conditions. Thus, the trajectories of particles with large FTLE are more sensitive to errors in their initial conditions.

The set $\Sigma_\alpha(\Phi)$, defined as the set of points $\mathbf{x}$ with FTLE above a threshold value α,

$$\Sigma_\alpha(\Phi) = \{\mathbf{x} \mid \sigma(\Phi;\mathbf{x}) > \alpha\}, \tag{17}$$

is the collection of points where error will magnify the most through the map Φ. The flow map approximations above all involve composing intermediate flow maps,

$$\Phi_2 \circ \Phi_1, \tag{18}$$

so it is important to know which points flow into $\Sigma_\alpha(\Phi_2)$ through the map Φ_1. In other words, we want to describe the set $\Phi_1^{-1}(\Sigma_\alpha(\Phi_2)) = \{\mathbf{x} \mid \Phi_1(\mathbf{x}) \in \Sigma_\alpha(\Phi_2)\}$, and this is the subject of Section V A.

If the flow map Φ_2 is defined on a regular grid X_0, it is necessary to pass trajectories of Φ_1 through the interpolated map $\mathcal{I}(\Phi_2|_{X_0})$. This is the source of error in the flow map approximations, and this error is significant when the trajectories of Φ_1 pass into the set $\Sigma_\alpha(\Phi_2)$, where FTLE is large. Using a nearest neighbor interpolation the interpolation error becomes particularly simple:

$$\Phi_2(\Phi_1(\mathbf{x})) \approx \mathcal{I}(\Phi_2|_{X_0})(\Phi_1(\mathbf{x})) = \Phi_2(\Phi_1(\mathbf{x}) + \epsilon) \tag{19}$$

where $\mathbf{x} \in X_0$, and ϵ is the difference between $\Phi_1(\mathbf{x})$ and its nearest neighbor in X_0. However, each approximate method has been tested with nearest neighbor, linear and bicubic spline interpolations with no significant qualitative change in results. The propagation of interpolation error using unidirectional and bidirectional composition is the subject of section V B.

A. Accumulation of Particles

The main result of this section is that particles near the positive-time LCS (pLCS) flow into particles near the negative-time LCS (nLCS) in forward time, and vice-versa. This is consistent with the fact that pLCS and nLCS correspond to finite-time unstable and stable man-

ifolds, respectively, and is observed in Figs. 9 and 10 for the pitching plate and double gyre examples.

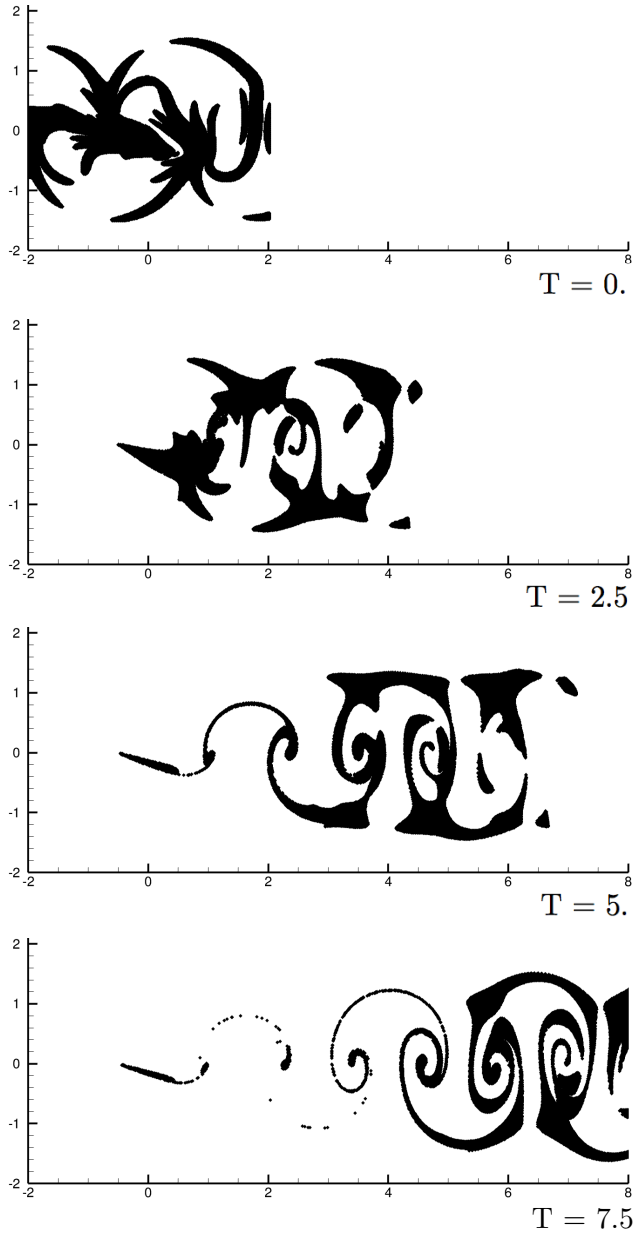


FIG. 9: Particle trajectories of the set $\Sigma_{.14}(\Phi_0^{15})$ for the pitching flat plate. Particles near the pLCS are integrated forward until they attract near the nLCS.

Figure 9 shows particles in the set $\Sigma_{.14}(\Phi_0^T)$, defined in Eq. (17), near the pLCS of the pitching plate example. As the particles convect downstream, they attract onto the nLCS. Compare with the first and last panel of the second row of Figure 6 to see what the pLCS and nLCS look like for this example. Similarly, Fig. 10 shows points in $\Sigma_{.3}(\Phi_0^{-T})$ near the nLCS of the double gyre example being integrated in negative time until they attract onto the pLCS. Compare with the first and last panel of the first row of Fig. 6 to see the pLCS and nLCS of the double gyre.

The bottom panel of Fig. 10 is a zoom-in of the tangle of particles near a time-dependent saddle point at $T = -10$. A point $\mathbf{x}(t)$ is a time-dependent saddle if it is at the transverse intersection of the pLCS and the nLCS. It is numerically observed that these saddles mediate transport of particles near the pLCS into particles near the nLCS in positive time.

Further, suppose that $\mathbf{x}(t)$ persists as a time dependent saddle over a range of time $t \in (t_0-T-\epsilon, t_0+T+\epsilon)$, where $\epsilon > 0$ ensures uniform hyperbolicity. The positive and negative-time FTLE properties of this point establish an exponential dichotomy which implies that $\mathbf{x}(t)$ is a time-dependent hyperbolic trajectory [24]. This trajectory now carries with it all of the regular theory about saddles, including Hartman-Grobman and Stable/Unstable Manifold Theorems. In particular, we may consider the pLCS (resp. nLCS) to be the time-dependent stable (resp. unstable) manifold of $\mathbf{x}(t)$.

Applying the Lambda lemma, it follows that a disk which intersects the pLCS transversely will attract arbitrarily C^1 close to a disk on the nLCS in positive-time, eventually. In the examples above, we observe a similar phenomena, namely, that in the neighborhood of a time-dependent saddle, the particles near the nLCS came from particles near the pLCS at an earlier time.

Similarly, it is possible to flow particles with large positive-time FTLE backward in time, and vice-versa, resulting in a set which resembles a positive-time LCS computed using a longer integration time. This is observed in Fig. 11, where particles in $\Sigma_{.3}\left(\Phi_0^{-15}\right)$ are flowed forward along Φ_0^{15}, resulting in a set which accumulates on the nLCS of the longer-time flow Φ_{15}^{-15}.

B. Propagation of Interpolation Error

Consider the bidirectional composition of a positive-time flow map Φ^T with a negative-time flow map Φ^{-T}, where error ϵ is introduced due to interpolation:

$$\begin{aligned}\Phi^T\left(\Phi^{-T}(\mathbf{x})+\epsilon\right) &\approx \Phi^T\left(\Phi^{-T}(\mathbf{x})\right) + \mathbf{D}\Phi^T\left(\Phi^{-T}(\mathbf{x})\right)\cdot\epsilon \\ &= \mathbf{x} + \mathbf{D}\Phi^T\left(\Phi^{-T}(\mathbf{x})\right)\cdot\epsilon \end{aligned} \tag{20}$$

The composition error is largest for points $\mathbf{x} \in X_0$ where $\mathbf{D}\Phi^T\left(\Phi^{-T}(\mathbf{x})\right)$ is large. From Eqns. (3) and (4), we have the following relationship:

$$\|\mathbf{D}\Phi^T(\mathbf{y})\| \geq e^{\alpha|T|} \quad \text{for} \quad \mathbf{y} \in \Sigma_\alpha(\Phi^T) \tag{21}$$

where $\|A\| = \max_{\mathbf{x}} \dfrac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2}$ is the maximum singular value of A. Thus, composition error is large at points $\mathbf{x}$, where $\mathbf{y} = \Phi^{-T}(\mathbf{x})$ is in the set $\Sigma_\alpha(\Phi^T)$, for large α and T.

Moreover, the results of the previous section indicate that points $\mathbf{x}$ satisfying $\Phi^{-T}(\mathbf{x}) \in \Sigma_\alpha(\Phi^T)$ originate in

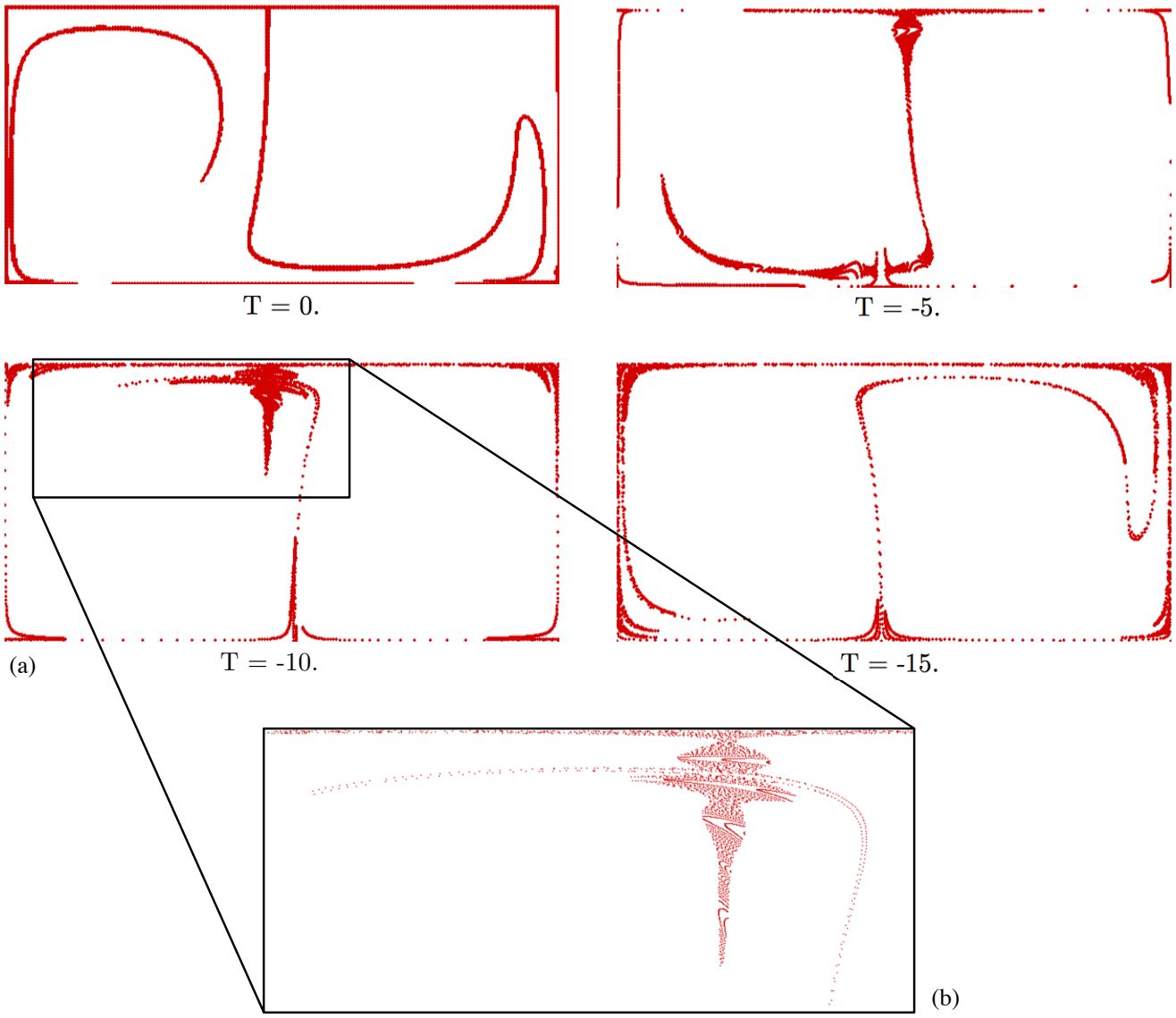


FIG. 10: (top) Particle trajectories of the set $\Sigma_{.3}(\Phi_0^{-15})$ for the double gyre. As particles on the nLCS are integrated backward they begin to adhere to the pLCS. (bottom) The time-dependent saddle (intersection of pLCS and nLCS) at $T = -10$ is blown-up to show the heteroclinic tangle.

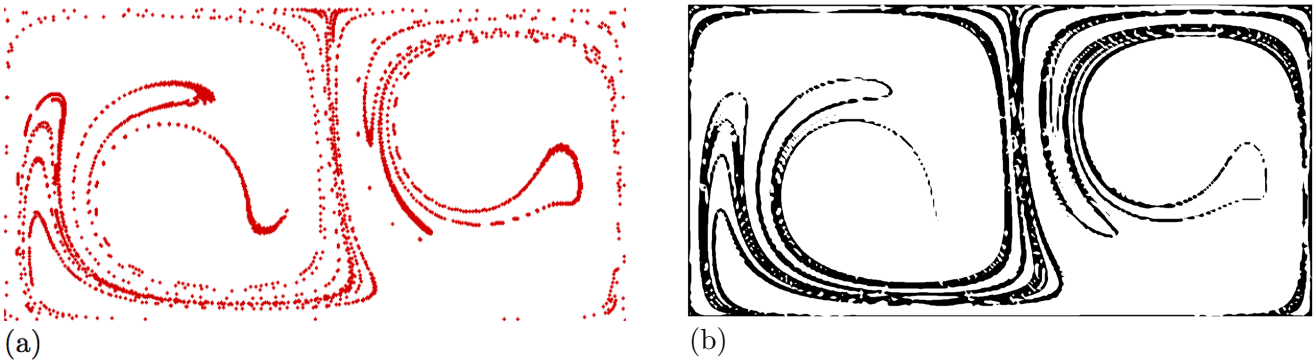
the set $\Sigma_\beta(\Phi^{-T})$ near the nLCS, in a neighborhood of a time-dependent saddle. Therefore, it is seen that the composition error will be largest at points $x \in \Sigma_\beta(\Phi^{-T})$, near the nLCS.

Now, consider the unidirectional composition of two positive-time flow maps, with interpolation error ϵ:

$$\begin{aligned} \Phi^T\left(\Phi^T(\mathbf{x}) + \epsilon\right) &\approx \Phi^T\left(\Phi^T(\mathbf{x})\right) + \mathbf{D}\Phi^T\left(\Phi^T(\mathbf{x})\right) \cdot \epsilon \\ &= \Phi^{2T}(\mathbf{x}) + \mathbf{D}\Phi^T\left(\Phi^T(\mathbf{x})\right) \cdot \epsilon \end{aligned} \quad (22)$$

Here the error is largest for points $\mathbf{x} \in X_0$ where $\mathbf{D}\Phi^T\left(\Phi^T(\mathbf{x})\right)$ is large. Again, $\mathbf{D}\Phi^T(y)$ is large when $y \in \Sigma_\alpha(\Phi^T)$, for sufficiently large α and T.

In unidirectional composition, because the pLCS is repelling in positive time, points $\mathbf{x} \in X_0$ must be exactly in $\Phi^{-T}\left(\Sigma_\alpha(\Phi^T)\right)$, or else they will repel away from the regions where error is magnified. Similarly for bidirectional composition, because the pLCS is attracting in negative time, points will attract toward the regions where error magnifies. For this reason, the unidirectional method is robust to interpolation error, while the bidirectional method amplifies this error.



(a) (b)

FIG. 11: Particle trajectories of the set $\Sigma_{.3}(\Phi_0^{-15})$ for the double gyre. Particles on the nLCS are flowed forward, shown in (a), resulting in a longer time nLCS, shown in (b).

VI. CONCLUSIONS

A number of methods have been developed for the efficient computation of finite-time Lyapunov exponent (FTLE) fields in unsteady flows. In particular, the methods speed up the computation of a sequence of FTLE fields in time, used for frames of a movie, by approximating the particle flow map using information from neighboring times. The methods fall into two categories of flow map approximation based on composition of intermediate flow maps of the same time direction (unidirectional) or of both positive and negative-time directions (bidirectional). The main result is that the unidirectional method is both fast and accurate, and the computational savings over the standard method are proportional to the number of FTLE fields being computed per time T. The unidirectional method provides one or two orders of magnitude computational savings over the standard method on the three example flows, as summarized in Table I.

The bidirectional methods are also fast, and use less memory than the unidirectional methods; however, bidirectional methods suffer from large errors which are concentrated along regions where the opposite-time FTLE field is large, in the vicinity of time-dependent saddle points. This coherent error was unexpected, but is explained by dynamical systems theory, since particles close to the pLCS near a time-dependent saddle will map into particles close to the nLCS in positive time. This result extends the relevance of Lagrangian coherent structure analysis to near identity particle maps in general.

The fast methods are implemented on three example velocity fields, chosen to represent typical fluid flows, and compared on the basis of computation time, accuracy and memory usage. The results of the method comparisons are summarized in Table I and Fig. 6. The unidirectional algorithm works well on 2D and 3D domains with either compact or spatially periodic domains. For open domains, as in the example of the pitching plate in a free stream velocity, the unidirectional method accurately computes the negative-time FTLE fields corresponding to the attracting set, or nLCS. However, error is introduced when computing the positive-time FTLE field, as particle trajectory information is lost downstream of the FTLE domain. This loss of information is not a problem when computing the nLCS because trajectory information upstream of the plate is well approximated using uniform flow. In experiments, however, velocity field data is often only available on a limited domain, which might correspond to the FTLE domain. In this case, the unidirectional and standard methods will produce matching positive-time FTLE fields.

There are a number of future directions which might arise from this work. First, FTLE algorithms lend themselves to parallelization, so it is conceivable that with further optimization, it will be possible to obtain real-time FTLE visualizations for interesting problems. It would also be interesting to extend the above methods to incorporate adaptive mesh refinement (AMR) as well as complex domain geometries. Additionally, it is important to more precisely determine how and when particles near the pLCS flow into particles near the nLCS in positive-time.

Appendix A: Notation

- $X_0 \subset \mathbb{R}^n$ - Discrete particle grid,
- Φ_a^b - Particle flow map from time $t = a$ to time $t = b$,
- $\tilde{\Phi}$ - Approximation to the particle flow map Φ,
- $\Phi|_{X_0}$ - Flow map restricted to a discrete grid X_0,
- $\mathcal{I}(\Phi|_{X_0})$ - Interpolant of the flow map $\Phi|_{X_0}$, defined on a discrete grid X_0,
- $\mathcal{I}\Phi \triangleq \mathcal{I}(\Phi|_{X_0})$ - Shorthand for interpolation,

- $\mathbf{D}\Phi$ - The Jacobian of the flow map Φ,
- Δ - The Cauchy-Green deformation tensor,
- $\sigma(\Phi; \mathbf{x})$ - FTLE for flow Φ at the point $\mathbf{x}$,
- pLCS - Positive-time Lagrangian coherent structure, or repelling material line. A ridge of the FTLE field $\sigma(\Phi_{t_0}^{t_0+T}, X_0)$ is a pLCS if and only if it has nonzero Lagrangian rate of strain,
- nLCS - Negative-time Lagrangian coherent structure or attracting material line. A ridge of the FTLE field $\sigma(\Phi_{t_0}^{t_0-T}, X_0)$ is a nLCS if and only if it has nonzero Lagrangian rate of strain,
- $\Sigma_\alpha(\Phi) = \{\mathbf{x} \mid \sigma(\Phi; \mathbf{x}) > \alpha\}$ - Set of particles with FTLE above a given threshold α; for α sufficiently large, this set represents a neighborhood of the dominant LCS ridges,
- St - Strouhal number, defined as $St = \frac{fA}{U_\infty}$ where f is frequency, A is amplitude and U_∞ is free stream velocity.

Appendix B: Example Velocity Fields

A summary of the attributes of each example velocity field is given in Table 2. Below is a description of how to compute the given velocity fields and an image of each corresponding FTLE field.

Double Gyre

The double gyre is an analytically defined velocity field which is time-periodic on the closed and bounded domain, $[0, 2] \times [0, 1]$. The stream-function is

$$\begin{aligned} \psi(x, y, t) &= A \sin(\pi f(x, t)) \sin(\pi y) \\ f(x, t) &= \epsilon \sin(\omega t) x^2 + x - 2\epsilon \sin(\omega t) x \end{aligned} \tag{B1}$$

which yields the following vector field

$$\begin{aligned} u &= -\frac{\partial \psi}{\partial y} = -\pi A \sin(\pi f(x)) \cos(\pi y) \\ v &= \frac{\partial \psi}{\partial x} = \pi A \cos(\pi f(x)) \sin(\pi y) \frac{df}{dx} \end{aligned} \tag{B2}$$

The positive-time FTLE field for the double gyre is shown in Fig. 12. The light blue ridges are regions with high FTLE, and are candidates for repelling pLCS.

Pitching Flat Plate

The second example is the unsteady velocity field of a flat plate pitching in a uniform flow at low Reynolds number, $Re = 100$. The plate pitches about its leading edge according to the following angle of attack motion:

$$\alpha(t) = \alpha_{\max} \sin(2\pi f t) \tag{B3}$$

with maximum angle of attack, $\alpha_{\max} = 20°$, and frequency $f = .4$. The Strouhal number, St, is a dimensionless pitching frequency given by:

$$St = \frac{fA}{U_\infty} = .274 \tag{B4}$$

where $A = 2\sin(20°)$ is the amplitude of the plate's excursion, and $U_\infty = 1$ is the free stream velocity of the uniform flow.

The motion of the plate is enforced using the multi-domain immersed boundary method of Taira & Colonius [25], using a second-order Adams-Bashforth time-stepper. The output of the direct numerical simulation (DNS) is a time-sequence of velocity fields spaced .05 apart in non-dimensional time units. Each velocity field snapshot is defined on five nested grids. The finest grid covers a 4×4 domain and the coarsest grid covers a 64×64 domain, non-dimensionalized by chord length. Each grid has resolution 200×200. This provides a large computational domain for integrating particle trajectories. Velocity fields from the DNS are stored on disk and are loaded for use in FTLE field computations.

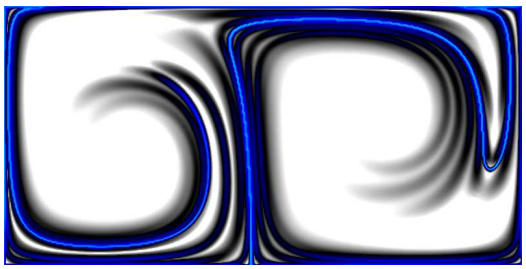


FIG. 12: FTLE field for double gyre, Eq. (B2), with $A = .1, \omega = 2\pi/10, \epsilon = .25, T = 15$.

FIG. 13: FTLE field for a pitching flat plate at $Re = 100$, $St = .274$, and $T = -15$.

The negative-time FTLE field for the pitching plate is shown in Fig. 13. The regions with large FTLE are

Problem	Dimension	Boundary Conditions	Velocity Field	Time Periodic
Double Gyre	2D	Closed	Analytic	Yes
Pitching plate	2D	Open	Data files (DNS)	Yes
ABC flow	3D	Periodic	Analytic	No

TABLE II: Attributes of each example problem.

colored red and yellow to indicate that they are candidates for attracting nLCS. In this example, regions of large FTLE clearly outline the wake and separated flow around the plate.

Unsteady ABC Flow

The unsteady ABC flow is a 3D flow which is aperiodic in time, has spatially periodic boundary conditions, and whose velocity field is defined analytically as follows

$$\begin{aligned} \dot{x} &= (A + \frac{1}{2}t\sin(\pi t))\sin z + C\cos y \\ \dot{y} &= B\sin x + (A + \frac{1}{2}t\sin(\pi t))\cos z \\ \dot{z} &= C\sin y + B\cos x \end{aligned} \tag{B5}$$

All FTLE fields are computed on the periodic cube $X, Y, Z \in [0, 1)$, where $x = 2\pi X$, $y = 2\pi Y$, and $z = 2\pi Z$.

The negative-time FTLE field for the unsteady ABC flow is shown in Fig. 14. Ridges of the FTLE field that are candidates for the attracting nLCS are colored in red and yellow.

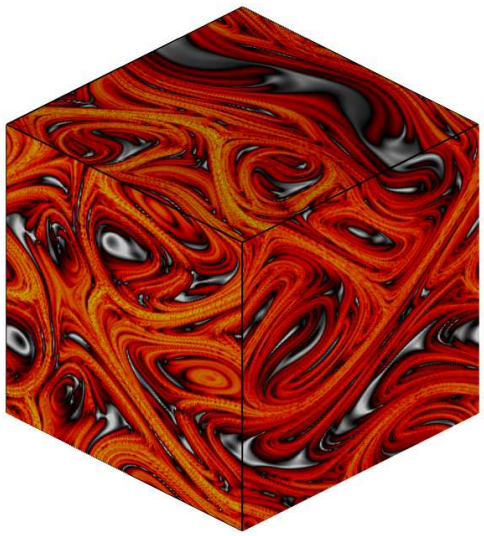


FIG. 14: FTLE field for a unsteady ABC flow with $A = \sqrt{3}$, $B = \sqrt{2}$, $C = 1$ and $T = -8$.

[1] P. Holmes, J. L. Lumley, and G. Berkooz, *Turbulence, coherent structures, dynamical systems and symmetry*, Cambridge Monographs in Mechanics (Cambridge Uni-

versity Press, 1996).
[2] G. Haller, J. Fluid Mech. **525**, 1 (2005).
[3] G. Haller, Physics of Fluids **14**, 1851 (2002).
[4] S. Shadden, F. Lekien, and J. Marsden, Physica D **212**, 271 (2005).
[5] M. Green, C. Rowley, and G. Haller, J. Fluid Mech. **572**, 111 (2007).
[6] G. Haller, Physica D **149**, 248 (2001).
[7] F. Lekien, S. Shadden, and J. Marsden, Journal of Mathematical Physics **48** (2007).
[8] H. Salman, J. S. Hesthaven, T. Warburton, and G. Haller, Theor. Comput. Fluid. Dyn. **21**, 39 (2007).
[9] E. Franco, D. N. Pekarek, J. Peng, and J. O. Dabiri, J. Fluid Mech. **589**, 125 (2007).
[10] S. Shadden, K. Katija, M. Rosenfeld, J. Marsden, and J. O. Dabiri, J. Fluid Mech. **593**, 315 (2007).
[11] J. Peng and J. O. Dabiri, The Journal of Experimental Biology **211**, 2669 (2008).
[12] M. M. Wilson, J. Peng, J. O. Dabiri, and J. D. Eldgredge, J. Phys.: Condens. Matter **21** (2009).
[13] M. Green, Ph.D. thesis, Princeton University (2009).
[14] D. Lipinski, B. Cardwell, and K. Mohseni, J. Phys. A: Math. Theor. **41** (2008).
[15] S. Brunton, C. Rowley, K. Taira, T. Colonius, J. Collins, and D. Williams, 46th AIAA Aerospace Sciences Meeting and Exhibit (2008).
[16] S. Brunton and C. Rowley, 47th AIAA Aerospace Sciences Meeting and Exhibit (2009).
[17] K. Padberg, T. Hauff, F. Jenko, and O. Junge, New Journal of Physics **9** (2007).
[18] F. Lekien, Ph.D. thesis, California Institute of Technology (2003).
[19] F. Lekien, C. Coulliette, A. Mariano, E. Ryan, L. Shay, G. Haller, and J. Marsden, Physica D **210**, 1 (2005).
[20] C. Garth, F. Gerhardt, X. Trichoche, and H. Hagen, IEEE Transactions on Visulization and Computer Graphics **13**, 1464 (2007).
[21] C. Garth, A. Wiebel, X. Trichoche, K. Joy, and G. Scheuermann (IEEE-VGTC, 2008), vol. 27.
[22] F. Sadlo and R. Peikert, IEEE Transactions on Visulization and Computer Graphics **13** (2007).
[23] K. Shi, H.-P. Seidel, H. Theisel, T. Weinkauf, and H.-C. Hege, IEEE Computer Graphics and Applications pp. 24–36 (2008).
[24] S. Wiggins, *Introduction to applied nonlinear dynamical systems and chaos*, Texts in Applied Mathematics (Springer-Verlag, 2000).
[25] T. Colonius and K. Taira, Comput. Methods Appl. Mech. Engrg. **197**, 2131 (2008).