Fast computation of finite-time Lyapunov exponent fields for unsteady flows

Steven L. Brunton, and Clarence W. Rowley

Citation: Chaos 20, 017503 (2010); doi: 10.1063/1.3270044
View online: https://doi.org/10.1063/1.3270044
View Table of Contents: http://aip.scitation.org/toc/cha/20/1
Published by the American Institute of Physics

Articles you may be interested in

Finite-time Lyapunov exponent-based analysis for compressible flows
Chaos: An Interdisciplinary Journal of Nonlinear Science 26, 083112 (2016); 10.1063/1.4961066

Lagrangian coherent structures and the smallest finite-time Lyapunov exponent
Chaos: An Interdisciplinary Journal of Nonlinear Science 21, 023115 (2011); 10.1063/1.3579597

Lagrangian coherent structures from approximate velocity data
Physics of Fluids 14, 1851 (2002); 10.1063/1.1477449

The computation of finite-time Lyapunov exponents on unstructured meshes and for non-Euclidean manifolds
Chaos: An Interdisciplinary Journal of Nonlinear Science 20, 017505 (2010); 10.1063/1.3278516

Lagrangian based methods for coherent structure detection

Introduction to Focus Issue: Lagrangian Coherent Structures
Chaos: An Interdisciplinary Journal of Nonlinear Science 20, 017501 (2010); 10.1063/1.3278173
Fast computation of finite-time Lyapunov exponent fields for unsteady flows

Steven L. Brunton and Clarence W. Rowley

Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, New Jersey 08544, USA

(Received 1 September 2009; accepted 6 November 2009; published online 5 January 2010)

This paper presents new efficient methods for computing finite-time Lyapunov exponent (FTLE) fields in unsteady flows. The methods approximate the particle flow map, eliminating 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 error which is large in regions where the opposite-time FTLE field is large, rendering it unusable. The algorithms are implemented and compared on three example fluid flows: a double gyre, a low Reynolds number pitching flat plate, and an unsteady ABC flow. © 2010 American Institute of Physics. [doi:10.1063/1.3270044]

Lagrangian coherent structures (LCSs) are hyperbolic material lines or surfaces that provide a useful analogue of invariant manifolds for unsteady flow fields. LCSs are often determined as ridges of the field of finite-time Lyapunov exponents (FTLEs) that satisfy an additional 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 flows. The methods presented here streamline the computation of a sequence of FTLE fields by removing redundant trajectory integrations between neighboring particle flow maps. There are two categories of methods which approximate the particle flow map. The unidirectional methods compose intermediate flow maps of the same time direction, and the bidirectional methods compose intermediate flow maps of opposite time directions. It is shown that the unidirectional methods are the only methods that are both fast and accurate, providing orders of magnitude computational savings over the standard method when computing a sequence of FTLE fields in time for an unsteady flow.

I. INTRODUCTION

Coherent structures are important for understanding and modeling the underlying physical mechanisms of complex fluid flows. In particular, LCSs are defined using particle trajectories and are Galilean invariant, unlike Eulerian criteria. LCSs are hyperbolic material lines or surfaces; ridges of the FTLE field provide candidate material lines. 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. 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. 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.

The theory and computation of FTLEs are a relatively modern development, with extensions to three-dimensional (3D) and n-dimensional flows. FTLE analysis has been widely applied in a number of branches of fluid mechanics, including fluid transport, biopropulsion, flow over airfoils, plasmas, and geophysical flows. 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, often from stored velocity fields, in order to obtain a particle flow map. 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 (AMR) to reduce the number of integrations. Although effective, these methods are difficult to implement in practice, and they do not utilize information from previous flow maps for future calculations.

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 as unidirectional composition, composes intermediate flow maps which are all aligned in the same time direction. The second class, denoted as bidirectional composition, constructs a flow map by composing intermediate flow maps which are aligned in both positive and
negative times. 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 the only fast and accurate method. Orders of magnitude speedup may be achieved over the standard method, and computational improvement scales with the desired time resolution of the FTLE animation.

The bidirectional method, although requiring significantly less memory than the unidirectional method, suffers from significant error and is generally unusable. In particular, the errors in the positive-time FTLE field align with ridges of the negative-time FTLE field and vice versa. To understand this coherent error, we provide an error analysis for both methods, which is consistent with the interpretation of positive-time LCS (pLCS) and negative-time LCS (nLCS) as finite-time stable and unstable manifolds, respectively. Specifically, 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

\[
\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}(t), t).
\]

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 candidate LCS ridges from the Hessian of the FTLE field, \( \mathbf{u} \) must be \( C^2 \) in space. The velocity field may be analytically defined, but it is more often obtained from experiments or direct numerical simulation (DNS) which produce velocity field data at discrete snapshots in time. A method of computing FTLEs on a finite amount of discrete velocity field data has been developed.\(^4\)

Computing the FTLE field typically involves four steps. First, a grid of particles, \( \mathbf{x}_0 \subset \mathbb{R}^n \), 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 \( \Phi_{0, T}^{T, n} \), defined as

\[
\Phi_{0, T}^{T, n}: \mathbb{R}^n \to \mathbb{R}^n; \quad \mathbf{x}(0) \mapsto \mathbf{x}(0) + \int_0^T \mathbf{u}(\mathbf{x}(\tau), \tau) d\tau.
\]

Next, the flow map Jacobian \( \mathbf{D} \Phi_{0, T}^{T, n} \) is computed, usually by finite differencing, to obtain the Cauchy–Green deformation tensor,

\[
\Delta = (\mathbf{D} \Phi_{0, T}^{T, n})^* \mathbf{D} \Phi_{0, T}^{T, n},
\]

where \( * \) denotes the transpose. Finally, the largest eigenvalue \( \lambda_{\text{max}} \) of this symmetric tensor is extracted and synthesized into a FTLE field,

\[
\sigma(\Phi_{0, T}^{T, n}; \mathbf{x}_0) = \frac{1}{|T|} \log \lambda_{\text{max}}(\Delta(\mathbf{x}_0)).
\]

The bottleneck in this procedure is the large number of particle integrations required to obtain the particle flow map \( \Phi_{0, T}^{T, n} \). 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. (Color online) The standard method for computing a sequence of FTLE fields. Flow maps \( \Phi_{k,T}^{k+1,n} \) for \( k \in \{0,1,2,3\} \) are shown (solid black arrow). Essential (blue dotted ovals) and redundant (red dashed ovals) particle integrations are outlined.](image)

III. FLOW MAP APPROXIMATION

As shown in Fig. 1, the standard method of computing a sequence of FTLE fields involves inefficient reintegration of particles. The unidirectional and bidirectional methods outlined below streamline the computation of neighboring FTLE fields by approximating the time-\( T \) flow map \( \Phi_{0,T}^{T,n} \), which can be written as

\[
\Phi_{0,T}^{T,n} = \Phi_{0,T}^{T,n}_{n,T} \circ \cdots \circ \Phi_{0,T}^{T,n}_{1,T} \circ \Phi_{0,T}^{T,n}_{0,T},
\]

where \( t_N = t_0 + T \).

Because the flow maps are obtained numerically on a discrete grid of points, \( \mathbf{x}_0 \subset \mathbb{R}^n \), it is necessary to interpolate the maps at points \( \mathbf{x} \neq \mathbf{x}_0 \). Consider a flow map \( \Phi: \mathbb{R}^n \to \mathbb{R}^n \) and the same flow map restricted to \( \mathbf{x}_0 \), \( \Phi|_{\mathbf{x}_0}: \mathbb{R}^n \to \mathbb{R}^n \). The interpolation operator \( I \) acts on the discrete map \( \Phi|_{\mathbf{x}_0} \) and returns the interpolated map \( I \Phi: \mathbb{R}^n \to \mathbb{R}^n \), which approximates \( \Phi \) on \( \mathbb{R}^n \).

\[
I: \Phi|_{\mathbf{x}_0} \to I \Phi.
\]

Here we use the shorthand \( I \Phi = I(\Phi|_{\mathbf{x}_0}) \). We now obtain an approximation to the flow map in Eq. (5),

\[
\Phi_{0,T}^{T,n}(X_0) = I \Phi_{0,T}^{T,n}_{n,T} \circ \cdots \circ I \Phi_{0,T}^{T,n}_{1,T} \circ \Phi_{0,T}^{T,n}_{0,T}(X_0) = \Phi_{0,T}^{T,n}(X_0).
\]

The unidirectional method approximates the time-\( T \) flow map by composing a number of smaller time-\( h \) flow maps \( \Phi_{k,T}^{k+1,h} \), which all have the same time direction. The bidirectional method approximates the time-\( T \) flow map \( \Phi_{k,T}^{k+1,T} \) by first integrating backward to a reference time, \( t=0 \), then interpolating forward through a previously computed time-\( T \) map \( \Phi_{k,T}^{k+1,T} \), and finally integrating forward to time \( t_0 + T \). Additionally, the chain rule may be applied to each of the methods, resulting in an approximation to the flow map Jacobian \( \mathbf{D} \Phi_{k,T}^{k+1,T} \).
A. 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 \( k \) smaller time-\( h \) flow maps, where \( T=kh \).

\[
\Phi_0^{kh} = \Phi_{h}^{kh} \circ \cdots \circ \Phi_{h}^{kh} \circ \Phi_{0}^{h}.
\]

(8)

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.

The simplest approach is to compute a number of time-\( h \) flow maps and store them in memory. Then, to construct an approximate \( \Phi^{kh+T}_{0} \), it remains only to compose the sequence of interpolated time-\( h \) flow maps. The next iteration involves integrating one additional time-\( h \) flow map and composing the next sequence, as in Fig. 2.

To further improve efficiency by reducing the total number of flow map compositions, it is possible to construct a multitiered 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. 3.

B. Bidirectional composition

Bidirectional approximation eliminates redundancy from neighboring FTLE field computations by using the information from a known flow map at a given time \( \Phi_{0}^{h} \) to calculate an approximation to the flow map at another time \( \Phi^{kh+T}_{0} \). First, \( X_{0} \) is integrated from \( t_{0} \) to the reference time 0. The distorted grid \( \Phi_{0}^{0}(X_{0}) \) is then flowed through the interpolated map \( \Phi_{E}^{T} \) and finally integrated an amount \( t_{0} \) to the desired time \( t_{0}+T \).

\[
\Phi_{0}^{kh+T} = \Phi_{T}^{kh+T} \circ \Phi_{E}^{T} \circ \Phi_{0}^{0},
\]

(9)

The flow map \( \Phi_{0}^{T} \) is stored as a reference solution to compute an approximation to the flow map at neighboring times \( \Phi_{kh}^{kh+T} \) by

\[
\Phi_{kh}^{kh+T} = \Phi_{T}^{kh+T} \circ \Phi_{0}^{0} , \quad k \in \mathbb{Z}.
\]

(10)

This is referred to as bidirectional method 1 (BM1), and is shown in Fig. 4.

Instead of using \( \Phi_{0}^{T} \) as the reference solution for every future flow map computation, it is convenient to use the new approximate flow map \( \Phi_{kh}^{kh+T} \) as the reference solution for the next iteration, \( \Phi_{(k+1)h}^{kh+T} \),

\[
\Phi_{(k+1)h}^{kh+T} = \Phi_{kh}^{kh+T} \circ \Phi_{E}^{T} \circ \Phi_{0}^{0} ,
\]

(11)

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 2 (BM2), and is shown in Fig. 5.

Figures 4 and 5 are schematics of bidirectional methods 1 and 2 for the computation of a sequence of positive-time flow maps. The methods also work for the computation of a sequence of negative-time flow maps. When computing a
sequence of positive-time flow maps, the first reference flow map computed will have the earliest start time, and subsequent iterations of the method will provide flow maps starting at later times. Similarly, when computing a sequence of negative-time flow maps, the first reference map will have the latest start time, and subsequent iterations will produce flow maps where the start time works backward in time.

A variation on the bidirectional methods above, suggested by an anonymous reviewer, works in the opposite direction of BM1 and BM2. Thus, the first reference frame of the new method would be the last frame in the original method, and will work in the opposite direction. Figure 6 illustrates this variation on BM2 for the computation of positive-time flow maps. This variation on BM1 and BM2 will be denoted as bidirectional method 3 (BM3) and bidirectional method 4 (BM4), respectively.

C. Chain rule of compositions

As seen in Eq. (3), once the flow map $\Phi_{t_f}^{t_0+T}$ 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,

$$D(\Phi_{t_f}^{t_0+T})(x) = D(\Phi_{t_f}^{t_{N-1}} \circ \cdots \circ \Phi_{t_1}^{t_0}) D(\Phi_{t_0}^{t_f})(x)$$

$$= D(\Phi_{t_f}^{t_{N-1}}(\Phi_{t_{N-1}}^{t_{N-2}}(x)) \times \cdots \times D(\Phi_{t_0}^{t_f}(x)),$$

Applied to the bidirectional methods, this yields

$$\Phi_{h}^{T+h} = \Phi_{h}^{T+T} \circ \Phi_{0}^{T} \circ \Phi_{h}^{0}$$

$$\Rightarrow D(\Phi_{h}^{T+h}(x)) = D(\Phi_{h}^{T+T}(\Phi_{0}^{T}(\Phi_{h}^{0}(x)))) \times D(\Phi_{h}^{T}(\Phi_{0}^{T}(\Phi_{h}^{0}(x)))) \times D(\Phi_{h}^{0}(x)),$$

and applied to the unidirectional methods, this yields

$$\Phi_{0}^{T} = \Phi_{0}^{T-h} \circ \cdots \circ \Phi_{h}^{T} \circ \Phi_{h}^{h}$$

$$\Rightarrow D(\Phi_{0}^{T}(x)) = D(\Phi_{0}^{T-h}(\Phi_{h}^{T}(\Phi_{h}^{0}(x)))) \times \cdots \times D(\Phi_{h}^{T}(\Phi_{0}^{T}(\Phi_{h}^{0}(x)))) \times D(\Phi_{0}^{0}(x)).$$

IV. COMPARISON OF METHODS

Each method from Sec. III is implemented and tested on three example problems: the periodic double gyre, two-dimensional (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 open, closed, or periodic domains. Further details of the examples may be found in Appendix B.

In each comparison, the unidirectional method is accurate and offers the greatest speedup 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 is generally unusable. Table I summarizes the results comparing each method on the three example flows. In each comparison, the

<table>
<thead>
<tr>
<th>Problem</th>
<th>Resolution</th>
<th>$T/h$</th>
<th>Frames</th>
<th>Method</th>
<th>Memory (Gbytes)</th>
<th>Speedup</th>
<th>Accurate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Double gyre</td>
<td>1024×512</td>
<td>15</td>
<td>30</td>
<td>Standard</td>
<td>0.05</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Unidirectional</td>
<td>0.36</td>
<td>10</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bidirectional</td>
<td>0.14</td>
<td>6.2</td>
<td>No</td>
</tr>
<tr>
<td>Pitching plate</td>
<td>1024×512</td>
<td>15</td>
<td>30</td>
<td>Standard</td>
<td>0.48</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Unidirectional</td>
<td>0.70</td>
<td>8.2</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bidirectional</td>
<td>0.50</td>
<td>5.4</td>
<td>No</td>
</tr>
<tr>
<td>Pitching plate</td>
<td>600×300</td>
<td>150</td>
<td>192</td>
<td>Standard</td>
<td>0.48</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Unidirectional</td>
<td>1.8</td>
<td>67</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bidirectional</td>
<td>0.48</td>
<td>54</td>
<td>No</td>
</tr>
<tr>
<td>ABC flow</td>
<td>128³</td>
<td>20</td>
<td>40</td>
<td>Standard</td>
<td>0.48</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Unidirectional</td>
<td>2.6</td>
<td>68</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bidirectional</td>
<td>0.73</td>
<td>7.3</td>
<td>No</td>
</tr>
</tbody>
</table>
standard, unidirectional, and bidirectional methods are used to compute a sequence of FTLE fields which are used as frames in an unsteady animation. Unless explicitly stated, the unidirectional method refers to the single-tiered method as depicted in Fig. 2, and the bidirectional method refers to BM2 as depicted in Fig. 5. The flow map duration used to compute a 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 Sec. IV B, this is an upper bound on the speedup of the unidirectional method.

Figure 7 compares FTLE fields computed using each method after a number of iterations of the method have been applied. The number of iterations $k$ was chosen so that $kh = T$ to magnify the effect of bidirectional error. The column of FTLE fields calculated using unidirectional composition (C2) agrees well with the exact FTLE fields computed using the standard method (C1). The column of FTLE fields calculated using BM2 (C3) all have significant errors which are aligned with the opposite-time coherent structures. The opposite-time FTLE fields are shown in the right-most 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; please refer to Figs. 14–16 for color bars.

FTLE fields computed using BM2, 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. A graphical comparison of BM4 applied to the double gyre is shown in Fig. 8. Part (b) of Fig. 8 shows the approximate FTLE field of the map $\Phi_{0.15}^T$ obtained after 15 iterations of BM4. The approximation resulting from BM4 does not capture the true FTLE field, shown in part (a), but is more closely aligned with the longer-time FTLE field of the map $\Phi_{0.30}^T$, shown in part (c). An analysis of the coherent error observed in Figs. 7 and 8 is provided in Sec. V.
A. Example: Double gyre

This section provides a detailed comparison of the computational resources and accuracy of each method on the double gyre velocity field. Figure 9 shows the $L_2$ and $L_\infty$ errors of the forward-time FTLE field for the double gyre computed using the standard method with $T=16$, as time-step $\Delta t$ and grid spacing $\Delta x$ 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 $\Delta t \approx 0.001$, 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 10 compares computation time and $L_2$ error versus frame number (iteration number) for a sequence of FTLE fields of the double gyre, computed using the standard, unidirectional, and bidirectional methods. Each iteration produces a 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 $k=1$, and the time step of integration is $\Delta t=0.01$. The multitier unidirectional method uses four tiers.

The computation time of BM1 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. 5. After $T/2h=8$ iterations of BM1, it is advantageous to compute a new reference flow map using the stan-
dard method. This explains the breaks in the solid red curve in part (b) of Fig. 10, as the bidirectional method is exact at these iterations.

As seen in Fig. 10, the unidirectional method is both the fastest and most accurate method in this comparison. However, the $L_2$ error does not always indicate how well the various methods preserve coherent structures. For example, if the $L_2$ error of an approximate FTLE field is large, this could be caused by erroneous structures, but it could also be caused by correct structures that are either misaligned with or have higher peaks than those in the true field. However, if the $L_2$ error is sufficiently small, then this is a good indication that the coherent structures in the field are preserved. Because the end goal is the extraction of coherent structures, it is necessary to use these error norms in conjunction with actual FTLE field plots, as in Figs. 7 and 8.

**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 $\Delta t$ 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 BM2 requires $2h/\Delta t$ integration steps. Assuming $\Delta t \ll h$, the speedup 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 $\Delta t \rightarrow 0$, the computation of $\Phi^{T/h}_0$ 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 $\Phi^{T/h}_0$, while the bidirectional method stores three maps: $\Phi^0_{h'0}$, $\Phi^T_{h'0}$ and $\Phi^{T/h}_0$. The unidirectional method, however, stores every intermediate time-$h$ flow map $\Phi^{h(i-1)}_{h}$ of which there are $T/h$. Therefore, the memory requirement of the unidirectional method scales linearly with the upper bound on its speedup, $T/h$.

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 $T/h$.

$$\text{Memory (Gbytes)} \sim S \times D \times R^D \quad (15)$$

$$= \frac{8 \text{ bytes/double}}{1024^3 \text{ bytes/Gbytes}} \times \frac{T}{h} \times D \times R^D. \quad (16)$$

For example, a series of 2D, high-definition (1920×1080 resolution) FTLE fields may be computed using the unidirectional method with up to 100 times speedup using approximately 3.1 Gbytes of random access memory (RAM). A 3D FTLE field with resolution of $512 \times 256 \times 64$ may be computed with up to 100 times speedup with approximately 19 Gbytes of RAM.

In the double gyre and ABC flow examples, the velocity field is defined analytically according to Eqs. (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 DNS. 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 reloading the same velocity 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. 7?

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

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

$$\Sigma_\alpha(\Phi) = \{x | \sigma(\Phi; x) > \alpha\}, \quad (17)$$

is the collection of points where error will magnify the most through the map $\Phi$. Typically the threshold $\alpha$ is chosen so that the set $\Sigma_\alpha(\Phi)$ contains the dominant ridges of the FTLE field. The flow map approximations above all involve composing intermediate flow maps,

$$\Phi_2 \circ \Phi_1, \quad (18)$$

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

If the flow map $\Phi_2$ is defined on a regular grid $X_\alpha$, it is necessary to pass the trajectories of $\Phi_1$ through the interpolated map $\tilde{Z}(\Phi_2|x_\alpha)$. This is the source of error in the flow
map approximations, and this error is significant when the trajectories of \( \Phi_1 \) pass into the set \( \Sigma_\lambda(\Phi_2) \), where FTLE is large. Using a nearest neighbor interpolation, the interpolation error becomes particularly simple,

\[
\Phi_2(\Phi_1(x)) = \mathcal{I}(\Phi_2|\mathcal{I})(\Phi_1(x)) = \Phi_2(\Phi_1(x) + \epsilon),
\]

where \( x \in X_0 \), and \( \epsilon \) is the difference between \( \Phi_1(x) \) and its nearest neighbor in \( X_0 \). Moreover, 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 compositions is the subject of Sec. V B.

A. Accumulation of particles

Particles near the pLCS flow into particles near the nLCS in forward time, and vice versa. This observation is consistent with the fact that pLCS and nLCS correspond to finite-time unstable and stable manifolds, respectively, and is observed in Figs. 11 and 12 for the pitching plate and double gyre examples.

Figure 11 shows particles in the set \( \Sigma_{0.14}(\Phi_{15}^T) \), defined in Eq. (17), near the pLCS of the pitching plate example. As the particles convect downstream they attract onto the nLCS, seen by comparing the first and last panels of the second row of Fig. 7. Similarly, Fig. 12 shows points in \( \Sigma_{0.1}(\Phi_{15}^T) \) near the nLCS of the double gyre being integrated in negative time until they attract onto the pLCS, seen by comparing with the first row of Fig. 7.

The bottom panel of Fig. 12 is a zoom-in of the tangle of particles near a time-dependent saddle point at \( T = -10 \) in the double gyre example. A point \( x(t) \) 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 \( x(t) \) 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 \( x(t) \) is a time-dependent hyperbolic trajectory. 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 (respectively nLCS) to be the time-dependent stable (respectively unstable) manifold of \( x(t) \).

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 phenomenon, namely, that in the neighborhood of a time-dependent saddle, the particles near the nLCS originate 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 pLCS computed using a longer integration time. This is observed in Fig. 13, where particles in \( \Sigma_{15}(\Phi_{15}^T) \) are flowed backward along \( \Phi_{15}^T \), resulting in a set which accumulates on the pLCS of the longer-time flow \( \Phi_{15}^T \).

B. Propagation of interpolation error

Consider the bidirectional composition of a positive-time flow map \( \Phi^T \) with a negative-time flow map \( \Phi^{-T} \), where error \( \epsilon \) is introduced due to interpolation

\[
\Phi^T(\Phi^{-T}(x) + \epsilon) \approx \Phi^T(\Phi^{-T}(x)) + \text{D} \Phi^T(\Phi^{-T}(x)) \cdot \epsilon
\]

\[
= x + \text{D} \Phi^T(\Phi^{-T}(x)) \cdot \epsilon.
\]

The composition error is largest for points \( x \in X_0 \) where \( \text{D} \Phi^T(\Phi^{-T}(x)) \) is large. From Eqs. (3) and (4), we have the following relationship:
\[
\|D\Phi^T(y)\| \geq e^{\alpha |T|} \quad \text{for} \quad y \in \Sigma_\alpha(\Phi^T),
\]
where \(\|A\| = \max_x(\|Ax\|_2/\|x\|_2)\) is the maximum singular value of \(A\). Thus, composition error is large at points \(x\), where \(y = \Phi^{-T}(x)\) is in the set \(\Sigma_\alpha(\Phi^T)\), for large \(\alpha\) and \(T\).

Moreover, the results of Sec. V A indicate that points \(x\) satisfying \(\Phi^{-T}(x) \in \Sigma_\alpha(\Phi^T)\) originate in 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 \(\epsilon\),
\[
\Phi^T(\Phi^T(x) + \epsilon) = \Phi^{2T}(x) + D\Phi^T(\Phi^T(x)) \cdot \epsilon
\]
Here the error is largest for points \(x \in X_0\) where \(D\Phi^T(\Phi^T(x))\) is large. Again, \(D\Phi^T(y)\) is large when \(y \in \Sigma_\alpha(\Phi^T)\) for sufficiently large \(\alpha\) and \(T\).

In unidirectional composition, because the pLCS is repelling in positive time, points \(x \in X_0\) must be exactly in \(\Phi^{-T}(\Sigma_\alpha(\Phi^T))\), 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.

VI. CONCLUSIONS

Two classes of interpolation methods have been developed for the efficient computation of 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.

All bidirectional methods are generally unusable. Bidirectional methods 1 and 2 (BM1 and BM2) 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 can be 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 LCSs to error analysis of near identity particle maps in general. Similarly, bidirectional methods 3 and 4 (BM3 and BM4) suffer from large errors, which are aligned with longer time flow maps.

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 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 are often only available on a limited domain, which might correspond to the FTLE domain. In this case, the unidirectional and standard methods will produce identical positive-time FTLE fields.

FTLE algorithms lend themselves to parallelization, so it is a feasible goal to develop real-time FTLE visualizations for interesting problems. It is also interesting to extend the above methods to incorporate AMR as well as complex domain geometries. Additionally, it is important to determine more precisely how and when particles near the pLCS flow into particles near the nLCS in positive time.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support for this work from the Air Force Office of Scientific Research (Grant No. FA9550-07-1-0127). We also thank Melissa Green, Phil Holmes, and the anonymous referees for valuable comments and discussion.

APPENDIX A: NOTATION

- $X_0 \in \mathbb{R}^n$—Discrete particle grid.
- $\Phi_t^t—Particle flow map from $t=a$ to time $t=b$. $\Phi$—Approximation to the particle flow map $\Phi$. $\Phi|_{X_0}$—Flow map restricted to a discrete grid $X_0$. $\hat{T}(\Phi|_{X_0})$—Interpolant of the flow map $\Phi|_{X_0}$ defined on a discrete grid $X_0$. $T(\Phi|_{X_0})$—Shorthand for interpolation.
- $\mathbf{D} \Phi$—The Jacobian of the flow map $\Phi$.
- $\Delta$—The Cauchy–Green deformation tensor.
- $\sigma(\Phi, x)$—FTLE for flow $\Phi$ at point $x$.
- pLCS—Positive-time Lagrangian coherent structure or repelling material line. A ridge of the FTLE field $\sigma(\Phi|_{t=T}, X_0)$ 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 $\sigma(\Phi|_{t=0}, X_0)$ is a nLCS if and only if it has nonzero Lagrangian rate of strain,
- $\Sigma_\alpha(\Phi) = \{ x | \sigma(\Phi, x) > \alpha \}$—Set of particles with FTLE above a given threshold $\alpha$; for $\alpha$ sufficiently large, this set represents a neighborhood of the dominant LCS ridges.
- $\text{St}$—Strouhal number defined as $\text{St}=fA/U_\infty$, where $f$ is frequency, $A$ is amplitude, and $U_\infty$ is free stream velocity.

APPENDIX B: EXAMPLE VELOCITY FIELDS

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

1. 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

$$\psi(x, y, t) = A \sin(\pi f(x, t)) \sin(\pi y),$$

$$f(x, t) = \epsilon \sin(\omega t) x^2 + x - 2 \epsilon \sin(\omega t) x,$$

which yields the following vector field:
Fast computation of unsteady FTLE fields

Chaos 20, 017503 (2010)

where $\alpha=0.1$, $\omega=2\pi/10$, $\epsilon=0.25$, $T=15$.

\[ u = -\frac{\partial \psi}{\partial y} = -\pi A \sin(\pi f(x)) \cos(\pi y), \]
\[ v = \frac{\partial \psi}{\partial x} = \pi A \cos(\pi f(x)) \sin(\pi y) \frac{df}{dx}. \]  

The positive-time FTLE field for the double gyre is shown in Fig. 14. The light blue ridges are regions with high FTLE and are candidates for repelling pLCS.

2. 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:

\[ \alpha(t) = \alpha_{\text{max}} \sin(2\pi ft), \]

with maximum angle of attack $\alpha_{\text{max}}=20^\circ$ and frequency $f=0.4$. The Strouhal number $St$ is a dimensionless pitching frequency given by

\[ St = \frac{fA}{U_\infty} = 0.274, \]

where $A=2\sin(20^\circ)$ is the amplitude of the plate’s excursion and $U_\infty$ is the free stream velocity of the uniform flow.

The motion of the plate is enforced using the multidomain immersed boundary method of Taira and Colonius$^{25}$ using a second-order Adams–Bashforth time stepper. The output of the DNS is a time sequence of velocity fields spaced 0.05 apart in nondimensional time units. Each velocity field snapshot is defined on five nested grids. The finest grid has resolution of $200 \times 64$ domain, nondimensionalized by chord length. Each grid has resolution of $200 \times 200$. This nested grid 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.

The negative-time FTLE field for the pitching plate is shown in Fig. 15. The regions with large FTLE are colored red and yellow, indicating that they are candidates for attracting nLCS. In this example, regions of large FTLE clearly outline the wake and separated flow around the plate.

3. Unsteady ABC flow

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

\[ \dot{x} = \left( A + \frac{1}{2} \sin(\pi t) \right) \sin z + C \cos y, \]
\[ \dot{y} = B \sin x + \left( A + \frac{1}{2} \sin(\pi t) \right) \cos z, \]
\[ \dot{z} = C \sin y + B \cos x. \]

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. 16. Ridges of the FTLE field, which are candidates for attracting nLCS, are colored in red and yellow.


