

# A Benchmark for Evaluating FTLE Computations

Alexander Kuhn, Christian Rössl, Tino Weinkauff, and Holger Theisel

## ABSTRACT

The Finite Time Lyapunov Exponent (FTLE) has become a widespread tool for analyzing unsteady flow behavior. For its computation, several numerical methods have been introduced, which provide trade-offs between performance and accuracy. In order to decide which methods and parameter settings are suitable for a particular application, an evaluation of the different FTLE methods is necessary. We propose a general benchmark for FTLE computation, which consists of a number of 2D time-dependent flow fields and error measures. Evaluating the accuracy of a numerically computed FTLE field requires a ground truth, which is not available for realistic flow data sets, since such fields can generally not be described in a closed form. To overcome this, we introduce approaches to create non-trivial vector fields with a closed-form formulation of the FTLE field. Using this, we introduce a set of benchmark flow data sets that resemble relevant geometric aspects of Lagrangian structures, but have an analytic solution for FTLE. Based on this ground truth, we perform a comparative evaluation of three standard FTLE concepts. We suggest error measures based on the variance of both, the fields and the extracted ridge structures.

## 1 INTRODUCTION

Understanding the complex flow behavior in unsteady flows is still a challenging task with a large number of ongoing research activities. An established tool for analyzing flow data is the Finite Time Lyapunov Exponent (FTLE). It measures the rate of convergence or divergence between neighboring flow particles over a finite time interval. Features in the resulting scalar field, most importantly its ridges, have been shown to be in close relation to the definition of Lagrangian Coherent Structures (LCS) described by Haller [8]. Thus, they are a good candidate for compact topological representation of time-dependent flow fields.

Due to intensive research activities, a variety of numerical FTLE computation schemes have emerged: the classic FTLE method described by Haller et al. [6, 7], methods using frequent renormalization by relocating sampling particles after specified reseeding criteria [15], or a localized method relying on the Jacobian to compute a FTLE value using a single path line [10], to mention a few. A more detailed overview is given in Section 2. All approaches share the common goal of computing the same quantity, but differ fundamentally in computational aspects.

The increasing number of computational methods raises the need for finding a common basis to compare different approaches. As FTLE has been suggested to be a relative measure [7], the absolute and exact values are argued to play a secondary role during primary evaluation and visual examination of the resulting scalar fields. However, as soon as additional (automated) post-processing is applied (such as feature extraction and subsequent flow segmentation), the accuracy of obtained values and their derivatives becomes increasingly important. Especially ridge extraction procedures have to rely on geometric properties and exact positions of potentially delicate FTLE features.

It is the goal of this paper to propose a benchmark for the evaluation and comparison of different methods for the numerical com-

putation of FTLE. The obvious first choice for this would be an estimation of the error order of different FTLE schemes. Such an approach is very common for estimating the accuracy of numerical line integrators and has been applied to modern integration schemes [9]. However, deviations in the FTLE computation appear not to be dominated by the error of numerical line integration, but rather by the *gradient estimation* of the flow map.

We are not aware of an analytical estimation of approximation order for methods computing the flow map gradient. Instead, we propose an experimental evaluation, i.e., the comparison of different FTLE schemes on a set of well-chosen example fields. The fundamental problem with this (and perhaps the main reason that such an evaluation has not been carried out before) is the problem of *missing ground truth*: the definition of FTLE is based on the concept of the flow map, which in general cannot be computed exactly, but only approximated by a number of numerical integrations. Hence, different FTLE computation schemes (i.e., numerical approximation methods) can only be compared against each other, but not against a non-approximative ground truth. To overcome this, we make the following contributions:

- We present a general approach to create non-trivial vector fields with a closed-form flow map using the concepts of *mirroring* and *deformation* (Section 3).
- We propose basic error measures to compare different numerical FTLE computation schemes (Section 4.2). All error measures are based on closed-form flow maps and evaluate the scalar FTLE values, as well as the resulting ridge structures.
- We introduce a set of 2D unsteady test vector fields with closed-form flow maps, covering a set of crucial geometric ridge effects, that occur with the evaluation of FTLE fields for real-world flows (Section 4).
- The application of our error measures to these test vector fields constitutes our benchmark. We conduct an extensive parameter study for three FTLE schemes: classic FTLE [6], FTLE with reseeding [4, 15], and Localized FTLE [10] (Section 5).

## 2 RELATED WORK

In practical settings, FTLE has become a standard tool to observe the separating behavior within unsteady flows. It has been used to describe flow behavior on planetary surfaces [3, 14], the movement of jelly fish [11, 26], and turbine separation analysis [21], to name a few (see Peacock et al. [17] for further applications). To describe FTLE in a formal way, we consider an unsteady vector field  $\mathbf{v}(\mathbf{x}, t)$ . Its *flow map*  $\phi^\tau(\mathbf{x}, t) = \phi(\mathbf{x}, t, \tau)$  is defined as the location of a particle seeded at  $(\mathbf{x}, t)$  after a path line integration of  $\mathbf{v}$  over a time interval  $\tau$ . In other words,  $\phi$  maps the start point of a path line integration to its end point. The computation of FTLE is based on the spatial gradient of the flow map  $\phi$ :

$$\nabla\phi(\mathbf{x}, t, \tau) = \frac{\partial\phi(\mathbf{x}, t, \tau)}{\partial\mathbf{x}}. \quad (1)$$

From this, we compute the values

$$\mu_i = \ln \sqrt{\lambda_i(\nabla^T \nabla)}, \quad (2)$$

with  $\nabla = \nabla\phi$  and  $\lambda_i$  denoting the  $i$ -th eigenvalue of the symmetric matrix  $\nabla^T \nabla$ . We finally obtain the FTLE value as

$$\text{FTLE}(\mathbf{x}, t, \tau) = \frac{1}{\tau} \max\{\mu_1, \mu_2\}. \quad (3)$$

All examined methods use the formulas (1)–(3) to compute the FTLE value. The particular differences consist in how these methods approximate the flow map  $\phi$  and its gradient  $\nabla\phi$ . In the following, we discuss three common FTLE computation methods, which will later be evaluated using our benchmark. At the end of this Section, we give a short overview of further FTLE schemes.

### 2.1 Classic Method

#### C-FTLE

The classic FTLE method [7] uses uniformly spaced particles to approximate the flow map: a path line integration is started from every grid point, while the flow map gradient is computed using central differences on the mapped points w.r.t. the original grid. Figure 1 (left) shows a conceptual sketch of this method. Obviously, the grid resolution  $h$  is a crucial parameter for this scheme, since the flow map gradient is approximated using adjacent grid points. Furthermore, the particles of adjacent grid nodes might diverge strongly during integration, which may also lead to less accurate approximations of the local separation rate. As stated by Haller [7], this method is designed to allow for qualitative statements about the separation behavior in one cell rather than obtaining specifically accurate values at the grid vertices. This leads to the assumption, that higher sampling rates generally reduce deviations resulting from the finite difference gradient approximation scheme. In our parameter study (Section 5), we examine the influence of the following parameters:

- $h$  - the size of a uniform grid cell in  $x$ ,  $y$ , and  $z$ -direction, where small values indicate a high sampling resolution. **H**
- $\tau$  - the length of the integration **TAU**

### 2.2 Reseeding Method

#### R-FTLE

Since the problem of approximating the flow map gradient by finite differences is well known, in practice, renormalization strategies are applied during the FTLE computation. In general, these methods aim at a better local approximation of the flow map gradient while maintaining the sampling resolution. In addition to starting a tracer path line for each grid point, four auxiliary particles are seeded in close vicinity to this point, as illustrated in Figure 1 (middle). They are used to approximate the flow map gradient. If the distance between the tracer and its auxiliary particles exceeds a certain threshold, the auxiliary particles are reseeded close to the current position of the particle from the grid point as described, e.g., by Nese et al. [15]. For every reseeding step, the current flow map gradient has to be evaluated and accumulated using the previously computed matrices. Note that in the 2D unsteady case R-FTLE requires five times as many path line integrations as C-FTLE.

In addition to  $h$  and  $\tau$  from above, we consider the following parameters in our study for R-FTLE:

- $\varepsilon_1$  - the distance between grid vertex and auxiliary particles for both: initial seeding and reseeding **HEPS1**
- $\varepsilon_2$  - the threshold that triggers reseeding ( $\varepsilon_1 < \varepsilon_2$ ) **HEPS2**

For  $\varepsilon_1 = h$  and  $\varepsilon_2 = \infty$  C-FTLE and R-FTLE yield the same result. Further, we consider the auxiliary points to be reseeded along the main principle directions ( $x, y$ ) of the original grid. Although the mathematical background of such renormalization strategies has been well defined (see, e.g., [15]), we are not aware of detailed investigations regarding the practical impact of this procedure to the resulting FTLE fields.

### 2.3 Localized Method

#### L-FTLE

In theory, the highest accuracy in terms of approximation of the flow map gradient can be achieved by reseeding after every integration step. This concept leads to an approach named localized FTLE (L-FTLE) proposed by Kasten et al. [10]. This method uses a single path line for each grid point and evaluates the Jacobian (first order derivative of the flow) after every integration step or with a fixed sampling distance  $d$  along the path line. The Jacobian matrices are accumulated along the particle trajectory resulting in a close approximation of the flow map gradient around the grid vertex. This procedure uses just as many path line integrations as C-FTLE, but additionally requires the evaluation of  $J$ . This may be a potential drawback due to extra computational effort or numerical errors in the evaluation of derivatives, especially in applications using non-uniform grids (e.g. as described by Lekien et al. [12]). Kasten et al. further propose an acceleration technique exploiting the spatial coherence of separation along path lines. In our analysis we will focus on the first-mentioned approach without additional speed-up optimizations. Figure 1 (right) illustrates the concept. In addition to  $h$  and  $\tau$ , we consider the following parameter in our study:

- $d$  - fixed sampling distance of the Jacobian matrices are accumulated along path line **HSAMP**

### 2.4 Further Extensions to FTLE Computation

The above-mentioned methods are explicitly analyzed and compared as part of our study. Additionally, this Section will give an overview of advances to the FTLE evaluation with respect to its computational aspects. Adaptive methods subdivide the sampling grid near interesting features such as FTLE ridges. A set of suitable criteria has been presented by Sadlo and Peikert [20] and Lekien et al. [12]. In order to speed up computation, Garth et al. [5] consider the computation of FTLE fields of 3D time-dependent flows on intersection planes. The approximation of FTLE deformation tensors can also be extended to general unstructured grids as shown by Lekien et al. [12]. Fuchs et al. [4] present an approach to interpret integration time  $\tau$  as a scale space parameter, while Sadlo et al. [21] determine a minimal integration time in order to ensure material properties. Both methods allow for selecting suitable intervals for  $\tau$  considering the strength of the resulting ridge structures. With increasing integration time, the number and complexity of the resulting FTLE structures increases as well. Thus, filtering extracted ridge structures becomes important, as shown by Pobitzer et al. [19]. Hierarchical methods take a different approach: they consider shorter integral path segments rather than the complete  $\tau$ -domain to obtain approximations of FTLE. One such method is presented by Hlawatsch et al. [9] replacing the costly particle integration by a less accurate but faster segment interpolation. This allows shifting the computationally expensive integration into a pre-processing phase, but comes at the cost of an inherent error in the final scalar field due to the interpolation scheme. Recently, Leung et al. [13] proposed an Eulerian approach to compute FTLE. Again, the expensive integration within the field is substituted by solving a system of non-linear equations for every grid point.

### 2.5 Comparing and Evaluating FTLE Methods

Considering the variety of methodologies, it is a non-trivial task to compare and assess results of numerical methods for the computation of FTLE: until now, there has been no ground truth data, so only a comparison between approximate results is possible. Our work fills this gap through the generation of non-trivial vector fields with a closed-form representation of FTLE. There exists of course a number of well-known benchmark data sets in the literature. Examples are the *Double Gyre* (*Quad Gyre*) flows proposed by Shadden [24], the forced pendulum (e.g. as used by Kent [11]), or the Arnold-Beltrami-Childress (ABC) flows (see, e.g., [25]). Although these flows are given analytically, neither their flow map nor the

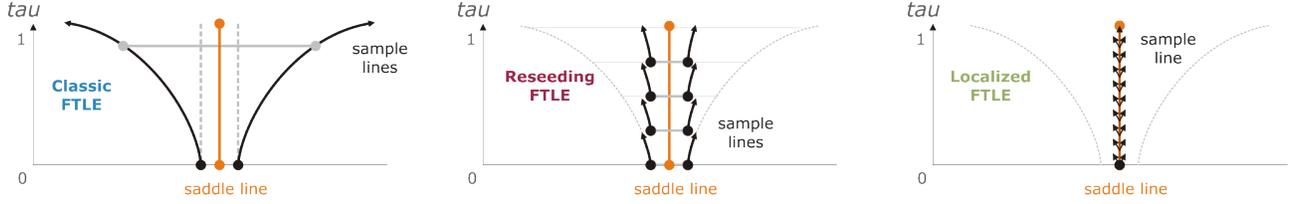


Figure 1: Conceptual representation of three different FTLE computation schemes: C-FTLE, R-FTLE, and L-FTLE (from left to right).

flow map gradient are available in a closed form, which limits their use as ground truth. In fact, the flow map and its gradient can only be formulated in a closed form for very simple analytic flows. In the work of Olcay et al. [16] a ridge-based comparison is presented including an error scheme that is created by PIV measurements. However, only the most prominent ridge structures have been considered by geometric approximation and were used for comparison. Hlawatsch et al. [9] present a profound error order analysis between their hierarchical approach and classic FTLE considering a direct comparison of the resulting ridge structures.

### 3 VECTOR FIELDS WITH CLOSED-FORM FLOW MAPS

Following the challenges outlined in the previous Section, we describe two novel approaches to construct non-trivial vector fields with closed-form flow maps: *domain deformation* and *mirroring*.

#### 3.1 Domain Deformation

Given is a time-dependent vector field  $\mathbf{v}(\mathbf{x}, t)$  in the spatial domain  $D$  and the time domain  $T$ . The flow map of  $\mathbf{v}$  is written as  $\phi_{\mathbf{v}}(\mathbf{x}, t, \tau)$ . Formulating this in space-time, we get

$$\bar{\mathbf{p}}(\mathbf{x}, t) = \begin{pmatrix} \mathbf{v}(\mathbf{x}, t) \\ 1 \end{pmatrix}, \quad \bar{\phi}_{\mathbf{p}}(\mathbf{x}, t, \tau) = \begin{pmatrix} \phi_{\mathbf{v}}(\mathbf{x}, t, \tau) \\ t + \tau \end{pmatrix}.$$

We assume  $\mathbf{v}$  of such a simple structure that we know  $\phi_{\mathbf{v}}$  in a closed form (e.g.,  $\mathbf{v}$  is linear). From this, we want to construct a more complex vector field  $\mathbf{w}$  for which a closed-form flow map exists as well. To get it, we define a differentiable map

$$\alpha : D \times T \rightarrow D$$

which is a diffeomorphism in its reduction to any  $t \in T$ . This means that  $\alpha$  is bijective and there is a unique inverse map  $\beta : D \times T \rightarrow D \times T$  with

$$\beta(\alpha(\mathbf{x}, t), t) = \alpha(\beta(\mathbf{x}, t), t) = \mathbf{x} \quad (4)$$

for any  $\mathbf{x} \in D$  and  $t \in T$ . From this, we obtain the bijective maps (as illustrated in Figure 2) in space-time:

$$\bar{\alpha}, \bar{\beta} : D \times T \rightarrow D \times T$$

$$\bar{\alpha}(\mathbf{x}, t) = \begin{pmatrix} \alpha(\mathbf{x}, t) \\ t \end{pmatrix}, \quad \bar{\beta}(\mathbf{x}, t) = \begin{pmatrix} \beta(\mathbf{x}, t) \\ t \end{pmatrix}.$$

This gives for the gradients:

$$\nabla \bar{\alpha} = \begin{pmatrix} \nabla \alpha & \frac{\partial \alpha}{\partial t} \\ 0 \dots 0 & 1 \end{pmatrix}, \quad \nabla \bar{\beta} = \begin{pmatrix} \nabla \beta & \frac{d\beta}{dt} \\ 0 \dots 0 & 1 \end{pmatrix}$$

where  $\nabla \alpha, \nabla \beta$  are the spatial gradients and  $\frac{\partial \alpha}{\partial t}, \frac{\partial \beta}{\partial t}$  are the time derivatives. Then (4) gives

$$\nabla \bar{\alpha}(\bar{\beta}(\mathbf{x}, t)) = \nabla \bar{\beta}^{-1}(\mathbf{x}, t)$$

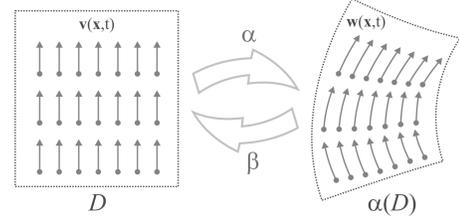


Figure 2: Creating a new vector field  $\mathbf{w}$  by mapping the domain of  $\mathbf{v}$  by  $\alpha$ .

and

$$\nabla \alpha(\beta(\mathbf{x}, t)) = \nabla \beta^{-1}(\mathbf{x}, t)$$

$$\frac{\partial \beta}{\partial t}(\mathbf{x}, t) = -\nabla \beta(\mathbf{x}, t) \cdot \frac{\partial \alpha}{\partial t}(\mathbf{x}, t).$$

Now we can construct  $\mathbf{w}$  out of  $\mathbf{v}, \alpha, \beta$  by applying the domain deformation  $\alpha$  to  $D$ . We start by describing  $\mathbf{w}$  in its space-time formulation  $\bar{\mathbf{q}}(\mathbf{x}, t) = \begin{pmatrix} \mathbf{w}(\mathbf{x}, t) \\ 1 \end{pmatrix}$ . We get

$$\bar{\mathbf{q}}(\mathbf{x}, t) = \nabla \bar{\alpha}(\bar{\beta}(\mathbf{x}, t)) \cdot \bar{\mathbf{p}}(\bar{\beta}(\mathbf{x}, t))$$

$$= \nabla \bar{\beta}^{-1}(\mathbf{x}, t) \cdot \bar{\mathbf{p}}(\bar{\beta}(\mathbf{x}, t)). \quad (5)$$

This gives for  $\mathbf{w}$ :

$$\mathbf{w}(\mathbf{x}, t) = (\nabla \beta)^{-1}(\mathbf{x}, t) \cdot \left( \mathbf{v}(\beta(\mathbf{x}, t), t) - \frac{d\beta}{dt}(\mathbf{x}, t) \right). \quad (6)$$

**Lemma 1** Let  $\phi_{\mathbf{w}}$  and  $\bar{\phi}_{\bar{\mathbf{q}}}$  be the flow maps of  $\mathbf{w}$  and  $\bar{\mathbf{q}}$  respectively. They can be computed as

$$\bar{\phi}_{\bar{\mathbf{q}}}(\mathbf{x}, t, \tau) = \bar{\alpha}(\bar{\phi}_{\bar{\mathbf{p}}}(\bar{\beta}(\mathbf{x}, t), \tau)) \quad (7)$$

$$\phi_{\mathbf{w}}(\mathbf{x}, t, \tau) = \alpha(\phi_{\mathbf{v}}(\beta(\mathbf{x}, t), t, \tau), t + \tau) \quad (8)$$

The proof of Lemma 1 follows directly from applying elementary differentiation rules to (7),(8) to  $\tau$  which gives (5) and (6) respectively. Lemma 1 has the following meaning: if the flow map of a (simple) vector field is known in a closed form, and if the maps  $\alpha, \beta$  are known in a closed form as well, the flow map of the (more complex) flow field  $\mathbf{w}$  is also known in a closed form.

#### 3.2 Mirroring Concept

As already pointed out, in most cases it is not possible to define a ground truth in a closed form. Thus, the main idea of this approach is to construct a non-trivial vector field from an arbitrary input field, which has a trivial flow map for a particular integration time: *the identity*. We start with a complex vector field  $\mathbf{v}(\mathbf{x}, t)$  in the domain  $D \times [t_0, t_1]$ , i.e., a closed form is not available. From this we construct a new vector field  $\mathbf{w}$  over the same domain as:

$$\mathbf{w}(\mathbf{x}, t) = \frac{t_1 - t}{t_1 - t_0} \mathbf{v}(\mathbf{x}, t) - \frac{t - t_0}{t_1 - t_0} \mathbf{v}(\mathbf{x}, t_0 + t_1 - t) \quad (9)$$

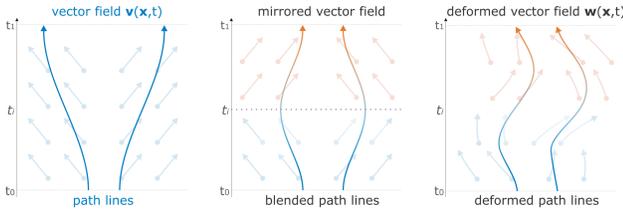


Figure 3: Steps to create a deformed ground truth field, where all path lines end up at their originating position. In the ideal case the FTLE value of every point equates to zero.

which is designed in such a way that its flow for an integration over the time interval  $t_1 - t_0$  is the identity (see Figure 3 for an illustration).

**Lemma 2**  $\phi_{\mathbf{w}}(\mathbf{x}, t_0, t_1 - t_0) = \mathbf{x}$  for any  $\mathbf{x} \in D$  if the integration does not leave  $D$ .

Proof: (9) gives  $\mathbf{w}(\mathbf{x}, \frac{t_0+t_1}{2} + \Delta t) = -\mathbf{w}(\mathbf{x}, \frac{t_0+t_1}{2} - \Delta t)$  for  $0 \leq \Delta t \leq \frac{t_1-t_0}{2}$ . From this it follows  $\phi_{\mathbf{w}}(\mathbf{y}, \frac{t_0+t_1}{2}, -\tau) = \phi_{\mathbf{w}}(\mathbf{y}, \frac{t_0+t_1}{2}, \tau)$  for any  $\mathbf{y} \in D$  and  $0 \leq \tau \leq \frac{t_1-t_0}{2}$ . Let  $\mathbf{y} = \phi_{\mathbf{w}}(\mathbf{x}, t_0, \frac{t_1-t_0}{2})$ . Then it holds:

$$\begin{aligned} \phi_{\mathbf{w}}(\mathbf{x}, t_0, t_1 - t_0) &= \phi_{\mathbf{w}}\left(\mathbf{x}, t_0, \frac{t_1 - t_0}{2}\right) + \phi_{\mathbf{w}}\left(\mathbf{y}, \frac{t_0 + t_1}{2}, \frac{t_1 - t_0}{2}\right) \\ &= \mathbf{y} + \phi_{\mathbf{w}}\left(\mathbf{y}, \frac{t_0 + t_1}{2}, \frac{t_1 - t_0}{2}\right) = \mathbf{x}. \end{aligned} \quad (10)$$

Note that although  $\phi_{\mathbf{w}}(\mathbf{x}, t_0, t_1 - t_0) = \mathbf{x}$ , the value of  $\phi_{\mathbf{w}}(\mathbf{x}, t_0, \tau)$  can attain extreme values for  $0 < \tau < t_1 - t_0$ . By construction path lines of the resulting vector field are symmetrical around  $\frac{t_0+t_1}{2}$ . To increase complexity and create additional structural variance within the resulting vector field, we can apply an additional domain deformation. This deformation has to guarantee to end up at the initial position (e.g., a  $360^\circ$  rotation over the integration domain  $[t_0, t_1]$ ). The resulting flow field is asymmetric, but still guaranteed to yield the identity flow map.

#### 4 BENCHMARK: DATA SETS AND ERROR MEASURES

In the analysis of real flows, one is mainly interested in the behavior of *ridges* in the FTLE fields [6, 16, 20, 21]. Hence, our goal is to design test vector fields that resemble crucial properties of such structures. For real flows, the following effects can be observed for FTLE ridges:

- FTLE ridges tend to be rather sharp and get sharper with increasing integration time. Also, an increasing integration time leads to less flux across the ridges [23].
- FTLE ridges can move towards each other and can in fact come very close to each other with increasing integration time. A good FTLE method should be able to cope with this effect.

In practice, many ridge structures are of far more complex nature, as their properties result from turbulent flow interaction and superposition of such effects. Our benchmark is designed to provide a reasonable abstraction from real flows, which is of course much simpler but parametrizes and imitates relevant geometric ridge effects. Focusing on such properties allows for clear statements for every FTLE method about the possible ridge extraction quality.

#### 4.1 Data Sets

Considering the above mentioned effects and methods described in Section 3, we propose three 2D time-dependent data sets for our benchmark. Every data set is associated with a field-specific shape parameter  $p_0$  which is used to create families of vector fields for every proposed example field. We propose a benchmark consisting of three data sets: **P0**

**Deformed double gyre.** We start with the well-known double gyre presented in [24], which has a simple analytic form but no closed-form flow map. Setting  $t_0 = 0$  and  $p_0$  as specific parameter, applying (9) at first gives a new field  $\mathbf{w}$  with the identity flow map for  $\tau = p_0 - 0$ . On top of this we apply a domain deformation with:

$$\begin{aligned} \alpha(\mathbf{x}, t) &= \begin{pmatrix} (x-1) \cdot \cos(\gamma) - (y-0.5) \cdot \sin(\gamma) + 1 \\ (x-1) \cdot \sin(\gamma) + (y-0.5) \cdot \cos(\gamma) + 0.5 \end{pmatrix} \\ \beta(\mathbf{x}, t) &= \begin{pmatrix} (x-1) \cdot \cos(-\gamma) - (y-0.5) \cdot \sin(-\gamma) + 1 \\ (x-1) \cdot \sin(-\gamma) + (y-0.5) \cdot \cos(-\gamma) + 0.5 \end{pmatrix} \end{aligned}$$

with  $\gamma = 2\pi \frac{t-t_0}{p_0-t_0}$  creating a new field  $\mathbf{w}_{gyre}$  which still has the identity flow map for any integration from  $t_0$  to  $p_0$ . Thus, during our evaluation  $p_0$  is always set to the same value as  $\tau$ . The superimposed deformation additionally rotates the vector field around the origin, while the angle is 0 at  $t_0$  and  $2\pi$  at  $t_1$ . The resulting field is illustrated in Figure 4. The additional deformation is intended to decrease error reduction due to symmetry effects along the  $\tau$  axis as described in Section 3.2.

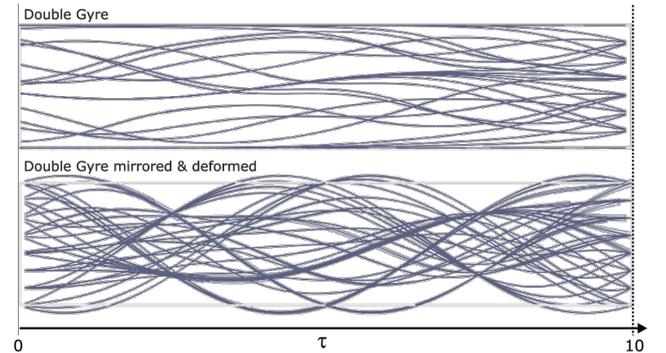


Figure 4: The top image shows path lines of the original field in the  $x, \tau$  plane and the bottom plot for the mirrored and deformed field. Note that path lines of the latter field are non-symmetric and guaranteed to end up at their original position in the  $x, y$  plane.

**Sine ridge data set.** This 2D unsteady field describes a sine-shaped ridge of arbitrary sharpness and is created by path lines diverging from the  $x = 0$  axis. We start with a trivial field

$$\mathbf{v}_{\text{sine}}(\mathbf{x}, t) = \begin{pmatrix} x(x-1)(x+1) \\ 1 \end{pmatrix} \text{ with}$$

$$\phi_{\mathbf{v}_{\text{sine}}}(\mathbf{x}, t, \tau) = \begin{pmatrix} x \\ \sqrt{x^2 + (1-x^2)e^{-2\tau p_0}} \\ y + \tau \end{pmatrix} \text{ as closed-form flow map.}$$

Using the domain deformation with:

$$\alpha(\mathbf{x}, t) = \begin{pmatrix} x + \sin(y) \\ y \end{pmatrix}, \beta(\mathbf{x}, t) = \begin{pmatrix} x - \sin(y) \\ y \end{pmatrix},$$

(6) gives the desired vector field  $\mathbf{w}$  with the closed-form flow map (8). In this case, the parameter  $p_0$  controls the rate of separation over  $\tau$  in terms of a scaling of the resulting vectors (or ‘‘sharpness’’ of the ridge) as shown in Figure 5.

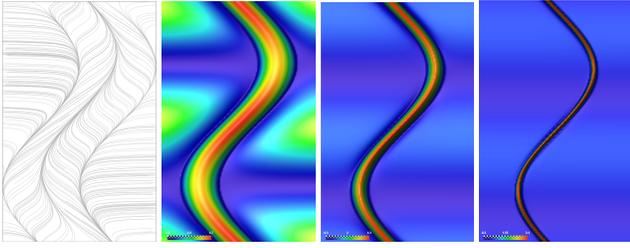


Figure 5: **Sine ridge**: Path line plot and resulting FTLE fields. The last two images show the effect of increasing the integration time and convergence towards the sine ridge structure (from left to right:  $\tau = 0.1, 0.5, 1.0$ ).

**Spiral focus ridge**: This data set is designed to evaluate the effect of approaching ridges. We start out with the trivial field

$$\mathbf{v}_{\text{spiral}}(\mathbf{x}, t) = \begin{pmatrix} 0 \\ x \end{pmatrix} \text{ and } \phi_{\mathbf{v}_{\text{spiral}}}(\mathbf{x}, t, \tau) = \begin{pmatrix} x \\ y + x \cdot \tau \end{pmatrix}.$$

As domain deformation we use

$$\alpha(\mathbf{x}, t) = \begin{pmatrix} x \cdot \cos(\gamma) - y \cdot \sin(\gamma) \\ x \cdot \sin(\gamma) + y \cdot \cos(\gamma) \end{pmatrix}$$

$$\beta(\mathbf{x}, t) = \begin{pmatrix} x \cdot \cos(-\gamma) - y \cdot \sin(-\gamma) \\ x \cdot \sin(-\gamma) + y \cdot \cos(-\gamma) \end{pmatrix}$$

with  $\gamma = \frac{p_0}{1+|x^2+y^2|}$ . The angle of the deformation depends on  $p_0$  and controls the winding number of the resulting ridge shaping a Fermat's spiral. The resulting ridges are created by a shear motion along the deformed main axis as illustrated in Figure 6.

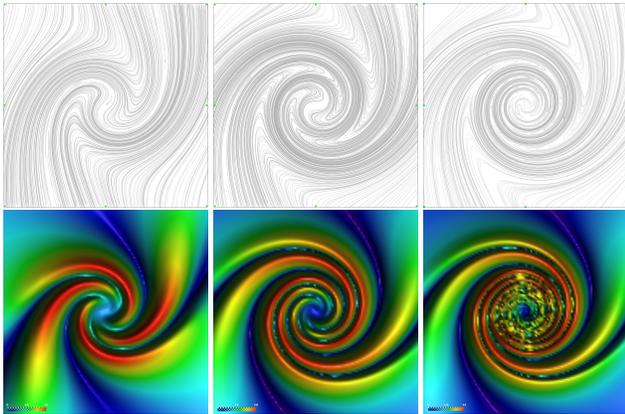


Figure 6: **Spiral focus**: Influence of the parameter  $p_0$  (Upper row: path line plot, Bottom row: resulting FTLE field for  $p_0 = 4, 8, 12$ ).

## 4.2 Error Measures

In order to evaluate the results of our study we introduce two error measures:

First, we define a *field error* measure which reflects the quality of the actual FTLE values. This is achieved by directly comparing given FTLE values against ground truth. We define the distance between scalar fields as the  $L_2$  error

$$L_2^{GT} = \sqrt{\frac{1}{n} \sum_{k=1}^n (\mathbf{f}^{(k)} - \mathbf{f}_{gt}^{(k)})^2},$$

where  $\mathbf{f}$  denotes the computed field and  $\mathbf{f}_{gt}$  denotes the given ground truth (which equals zero when using the mirroring technique described in Section 3.2).

Second, we introduce a *ridge-based error*. To measure the accuracy or suitability of a FTLE computation scheme for a subsequent ridge extraction, we compare the ridges extracted from the FTLE ground truth with those of the numerically obtained FTLE field. For our comparison we use ridges from a high-resolution sampling of the ground truth field (including ground truth partials). Note that we discretized only the domain, while ground truth values, FTLE, and its derivatives are independent of the sampling resolution.

In our study we used the *height ridge* definition by Eberly [2]. Thus, we indirectly consider the quality of first and second derivatives of the computed FTLE fields. We refer to Peikert and Sadlo [18] for a concise treatment regarding the extraction and its parameters; alternative methods are discussed in Schindler et al. [22].

The comparison of two sets of ridge lines is accomplished by considering the respective point sets  $\mathbf{s}_1$  and  $\mathbf{s}_{gt}$  and evaluating a suitable geometric distance:

Given two point sets as sample locations on a ridge, we compute their Hausdorff distance [1]

$$d_h(\mathbf{s}_1, \mathbf{s}_{gt}) = \max \{d_h(\mathbf{s}_1, \mathbf{s}_{gt}), d_h(\mathbf{s}_{gt}, \mathbf{s}_1)\}$$

with the one-directional distance

$$d_1(\mathbf{s}_1, \mathbf{s}_{gt}) = \sup \{d_h(\mathbf{x}, \mathbf{s}_1) | \mathbf{x} \in \mathbf{s}_{gt}\}. \quad (11)$$

For our results presented in Section 5 we used the 1-sided distance  $d_1$  from the ground truth to the extracted ridge. This way we explicitly penalize gaps in the extracted ridge structures, but potentially accept lower error values if the extraction result contains outliers (e.g., due to numerical noise or discontinuities in the resulting derivative fields). In addition to those error measures we measured the required computation time for every method. Although, timings heavily depend on computational resources and efficient implementation, they become of special interest when optimizing the performance of specific methods or evaluating the qualitative impact of speed-up techniques mentioned in Section 2.4.

Besides method-specific parameters and timings we considered the differences between a direct evaluation of the analytic equations and prior resampling of the vector fields (analytic and discrete setting). Note that we also used closed formulas for evaluating first order derivatives in the analytic setting.

## 5 BENCHMARK RESULTS

In this Section we present our evaluation study comparing all methods and measures on the proposed vector fields. In our study we focused on the examination of the following aspects of the FTLE computation: Behavior with increasing resolutions ( $H$ ), integration times ( $TAU$ ), field complexity ( $P0$ ), differences between analytical and discrete computations ( $FLAG$ ) and a relative comparison between the considered methods ( $C - FTLE, L - FTLE, R - FTLE$ ). Our computations have been parallelized using eight QuadCore CPUs with 2,7GHz and 6GB RAM each. Overall, we computed approximately 28,400 FTLE fields on varying resolution ( $50^2$  up to  $2,000^2$  pixels) in about 4,200 hours pure computation time. All FTLE computations were done using the Amira<sup>1</sup> framework, with CPU implementations of classic FTLE and FTLE with reseeded. For the localized FTLE we used the original implementation by Kasten et al. [10]. After computing the FTLE field we obtained a set of measures for each field, such as computation time ( $TIMING$ ), FTLE field error as  $L_2$  norm ( $GT\_L2\_N0$ ) from eq. 11, ridge-based error ( $RE\_HAUSD\_D1$ ) by eq. 11, as described in Section 4.2. The computation of the analytical ground truth fields was done using the Maple computer algebra system. The respective scripts are included in the accompanying material.

<sup>1</sup>www.amira.com

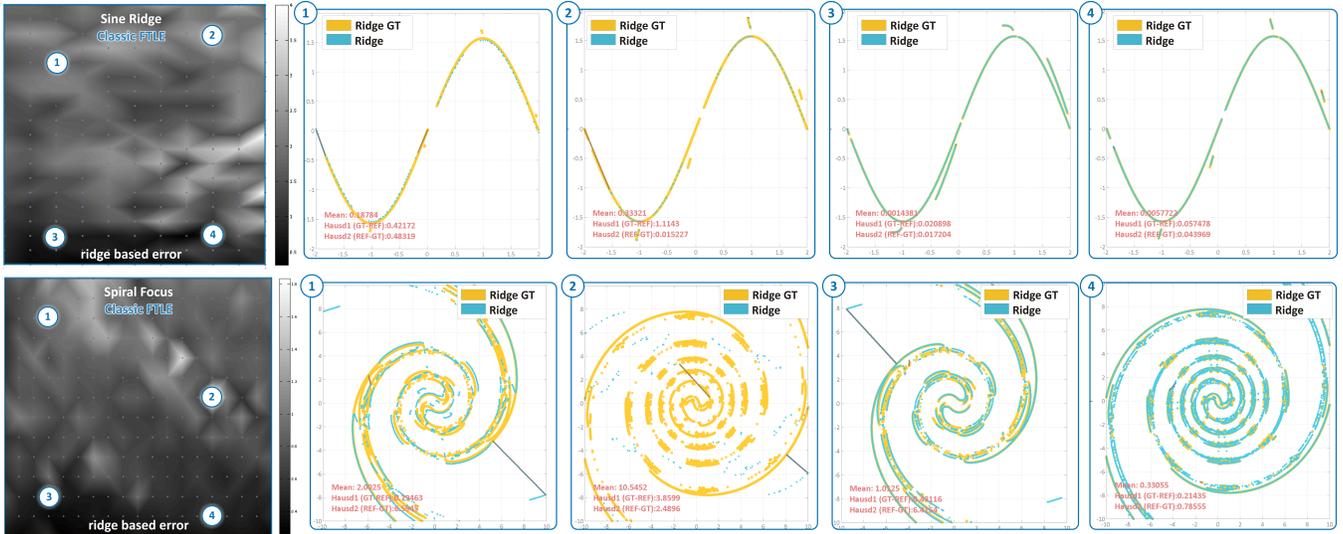


Figure 7: Ridge extraction samples for C-FTLE on the **sine ridge** (top) and **spiral focus** (bottom) test field. Yellow ridges have been extracted from the ground truth (GT) fields at the highest resolution available. The leftmost image shows the ridge error plots of Figure 10 and 12.

### 5.1 Deformed double gyre

To analyze the properties of the vector field constructed by the methods described in Section 3.2 we correlated the integration time  $TAU$  and the resolution  $H$  and plotted it against the  $L_2$  norm error in a sampling subspace (shown in Figure 8). The resulting error surfaces are apparently smooth, indicating that the error is not dominated by numerical artifacts but rather reflects method-inherent properties. Looking at the sampled subspace, R-FTLE and L-FTLE show a similar behavior, as with increasing sampling resolution and small  $\tau$  the resulting error is reduced. In this setting, C-FTLE performs better than the other methods, but the field error *increases* with higher  $TAU$  and high resolutions.

If a smaller subspace of the sampling space is considered, C-FTLE and R-FTLE show the same error behavior, which is consistent with the computation methodology, as after an  $EPS2$  renormalization is triggered (for  $TAU \in [8, 11]$ ).

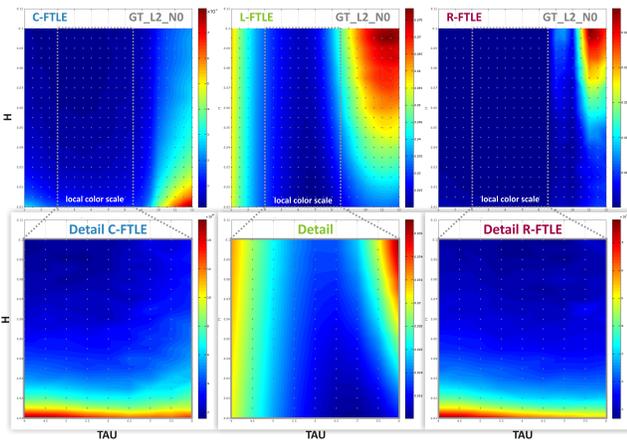


Figure 8: **Double gyre deformed:** Plots for the field error in *local* color scale. The top picture shows the resulting field error in dependence of resolution  $H$  and integration time  $TAU$ .

A detailed plot of the method-specific parameters of L-FTLE and R-FTLE against the  $GT\_L2\_N0$  error in Figure 9 shows a clear

correlation between these parameters and the resulting error to the ground truth. Both graphs indicate suitable settings for both parameters for this field type and emphasize that choosing correct parameter ranges is crucial to obtain reliable flow map gradient approximations.

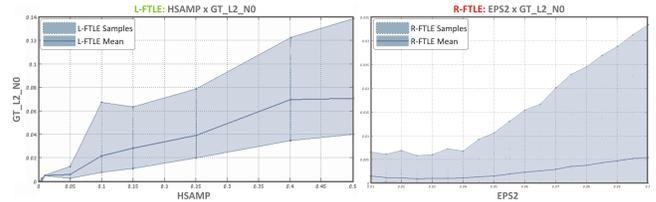


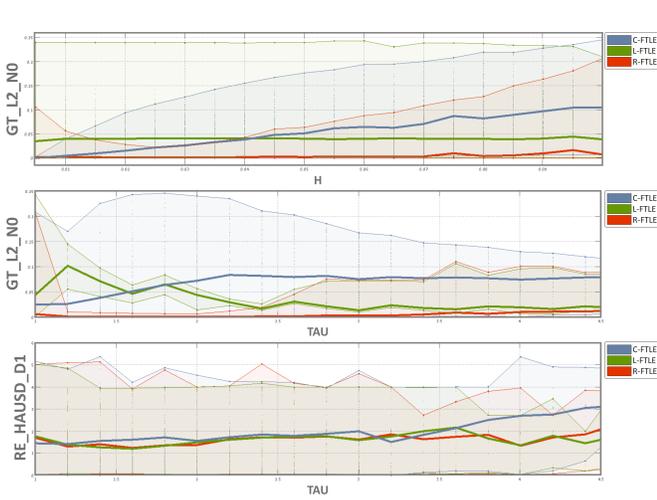
Figure 9: **Double gyre deformed:** Influence of the reseeding sampling distance  $HSAMP$  and reseeding trigger distance  $EPS2$  on the resulting field errors.

### 5.2 Sine Ridge

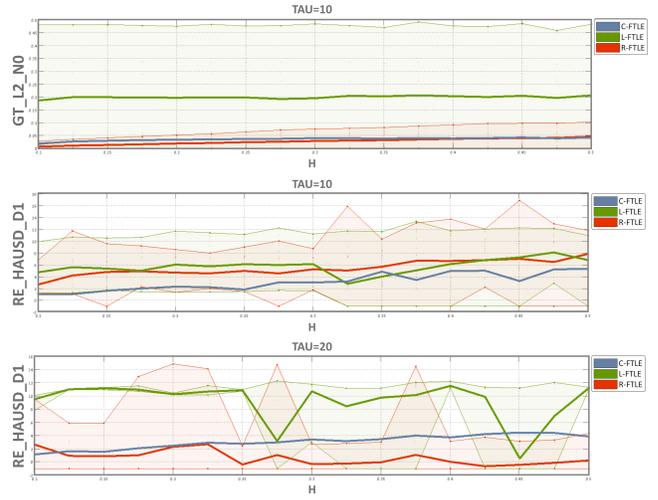
For the sine ridge we plotted the ridge error against integration time and resolution, while color coding represents the 1-sided Hausdorff distance from Equation 11. Due to the point-based discrete comparison the resulting error field is not necessarily smooth anymore as shown in Figure 10. All methods give better results with increasing sampling resolution, while C-FTLE performs better on lower resolutions. R-FTLE and L-FTLE give better results on high resolutions and long integration times, while the average field error is less dependent of  $H$ . Figure 10 also shows a direct comparison between the results obtained for vector fields discretized before evaluation and using a purely analytical description for the field and its derivatives. Except additional computation time due to derivative approximations, for the considered settings the error behavior remains similar in both cases.

### 5.3 Spiral Focus

Our last example represents the most elaborate FTLE field in our study (see Figure 6) in terms the computation time, approximation of derivatives and ridge complexity. Again, we plotted relative resolution against integration time and evaluated the resulting ridge



(a) **Sine ridge**: C-FTLE shows increasing field quality with higher resolution, while ridge quality decreases with higher  $\tau$ . Especially R-FTLE is less affected by  $\tau$ .



(b) **Spiral focus**: Comparison for resolution  $H$ . L-FTLE performs worse for long integration times, due to complex derivatives of the spiral focus example.

Figure 11: Averaged curves for both error measures among all samples. Thick lines indicates the mean values obtained by all samples, while the transparent area shows the range of values. Note, that *smaller* relative resolution  $H$  values denote higher sampling densities.

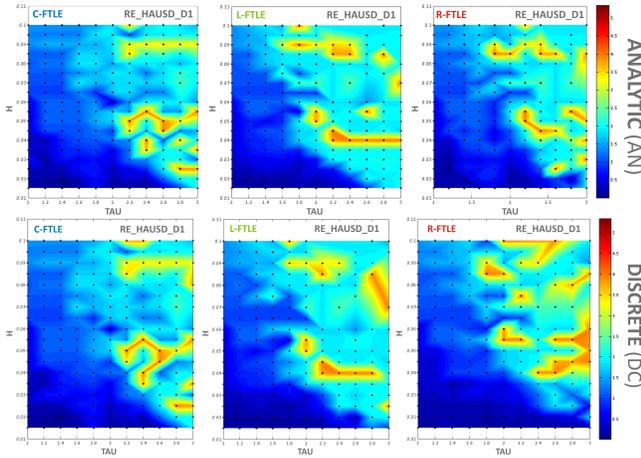


Figure 10: **Sine ridge**:  $H$  plotted against  $\tau$ , while color encodes the resulting ridge error for analytic and discrete field evaluation. C-FTLE performs better on lower resolutions, while the main characteristic is similar for all methods.

error shown in Figure 12. Although similar error characteristics as in the previous example becomes apparent, especially the localized method performs worse. The reason for this are the more complex derivatives of the FTLE field, while L-FTLE relies on frequent evaluation and accumulation of those. This becomes problematic for longer integration times (compare Figure 11b for  $\tau = 10$  and  $\tau = 20$ ). (Additional figures are provided with the accompanying materials for this paper.)

#### 5.4 Benchmark Summary

In summary, classic FTLE (C-FTLE) performed well for our test cases for shorter integration times, has no additional parameters and is well suited to obtain overviews about separating structures. However, it only offers an inaccurate approximation of the flow map gradient, thus especially for longer integration times the FTLE field-based error ( $GT\_L2\_NO$ ) as well as the ridge-based error

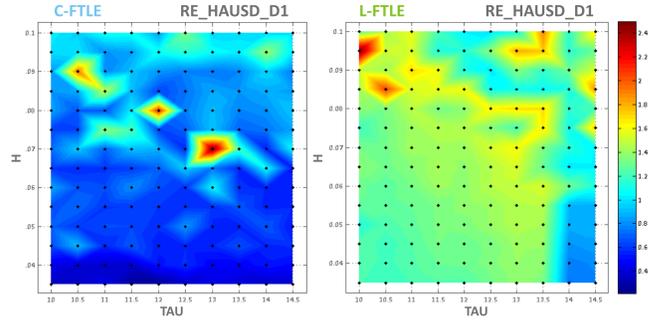


Figure 12: **Spiral focus**: Resolution  $H$  plotted against  $\tau$ , color coding the resulting ridge error. Especially L-FTLE performs worse than in the sine ridge example in figure 10.

( $RE\_HAUSD\_D1$ ) grows, while the rate of this growth depends on the properties of the underlying field (Figure 11). We found that just increasing the resolution shows only slow convergence against an exact solution for the given ground truth examples (Figure 8). In comparison the localized FTLE (L-FTLE) requires an additional parameter ( $HSAMP$ ) during computation and a reliable approximation of local first order derivatives. We showed in our examples, that for suitable parameter settings (see Figure 9) and reasonable estimation of derivatives, L-FTLE performs better than C-FTLE, especially for long integration times and high resolutions, as shown in Figure 11. This is especially important if exact spatial FTLE ridges for high  $\tau$  values are required. FTLE with reseeding (R-FTLE) represents a trade-off between both above-mentioned methods with respect to our proposed error measures. Main drawbacks are the higher computational effort and the additional parameters ( $EPS1, EPS2$ ).

## 6 CONCLUSIONS

We presented a benchmark framework to evaluate FTLE computations and provided a detailed evaluation study for three selected FTLE Methods: classic FTLE, FTLE with renormalization/reseeding and localized FTLE in Section 2. This includes the

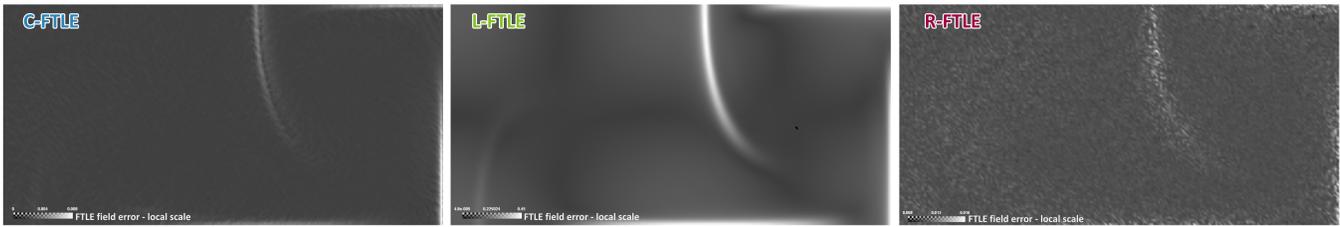


Figure 13: **Double gyre deformed**: FTLE field error plots for  $TAU = 10, H = 0.01$  and  $P_0 = 10$ . Note that the field error has a clear peak at the original ridge location, thus largest FTLE approximation deviations occur close to regions of high separation.

mirroring and deformation concept to evaluate FTLE field deviations on arbitrary vector fields and a general approach to construct non-trivial vector fields with closed-form flow maps. This allows for the exact computation of ground truth for FTLE and FTLE ridges. Based on this, we conducted extensive comparison of three standard FTLE methods using our benchmarking framework.

Using the above-mentioned techniques, we provide a common basis for future FTLE methods, their parameters and extensions, to be compared, tested, and evaluated against each other. This allows detailed analysis, even for fundamentally different approaches, such as Hierarchical FTLE [9] or Eulerian approaches [13], in terms of error behavior, optimal parameter settings, and practical applicability. Furthermore, the proposed benchmark workflow can be used to optimize and/or debug existing FTLE modules, determine and quantify the impact of novel extensions or filtering modules. In our study we mainly concentrated on 2D time-dependent flows. The proposed general method for constructing benchmarks can be similarly applied to 3D time-dependent data.

#### ACKNOWLEDGEMENTS

The project SemSeg acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number 226042. Furthermore, we like to thank Jens Kasten for providing the source code for the localized FTLE method.

#### REFERENCES

- [1] H. Alt, P. Brass, M. Godau, C. Knauer, and C. Wenk. Computing the hausdorff distance of geometric patterns and shapes. *Discrete and Computational Geometry - The Goodman-Pollack-Festschrift*, 2002.
- [2] D. Eberly. *Ridges in Image and Data Analysis*. Kluwer Academic Publishers, Dordrecht, 1996.
- [3] C. Farnetani and H. Samuel. Lagrangian structures and stirring in the Earth's mantle. *Earth and Planetary Science Letters*, 206(3-4):335–348, 2003.
- [4] R. Fuchs, B. Schindler, and R. Peikert. Scale-Space Approaches to FTLE Ridges. *Proceedings of TopoInVis 2011*, pages 1–14, 2011.
- [5] C. Garth, F. Gerhardt, X. Tricoche, and H. Hagen. Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1464–1471, 2007.
- [6] G. Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D*, 149:248–277, 2001.
- [7] G. Haller. Lagrangian structures and the rate of strain in a partition of two-dimensional turbulence. *Physics of Fluids*, 13(11), 2001.
- [8] G. Haller. A variational theory of hyperbolic Lagrangian Coherent Structures. *Physica D*, 240:574–598, Dec. 2010.
- [9] M. Hlawatsch and F. Sadlo. Hierarchical Line Integration. *IEEE Transactions on Visualization and Computer Graphics*, 17(8):1148–1163, 2011.
- [10] J. Kasten, C. Petz, I. Hotz, B. Noack, and H.-c. Hege. Localized finite-time Lyapunov exponent for unsteady flow analysis. In M. Magnor, B. Rosenhahn, and H. Theisel, editors, *Vision Modeling and Visual-*

- ization*, volume 1, pages 265–274. Universit{\ "a)t Magdeburg, Inst. f. Simulation u. Graph., 2009.
- [11] S. Kent. Lagrangian coherent structures: Generalizing stable and unstable manifolds to non-autonomous dynamical systems. Technical report, U. of Arizona, 2008.
- [12] F. Lekien and S. D. Ross. The computation of finite-time Lyapunov exponents on unstructured meshes and for non-Euclidean manifolds. *Chaos (Woodbury, N.Y.)*, 20(1):017505, Mar. 2010.
- [13] S. Leung. An eulerian approach for computing the finite time lyapunov exponent. *J. Comput. Phys.*, 230:3500–3524, May 2011.
- [14] H. B. Man. Titan Wind Analysis using Lagrangian Coherent Structures. *cds.caltech.edu*, pages 1–22, 2008.
- [15] J. Nese. Quantifying local predictability in phase space. *Physica D: Nonlinear Phenomena*, 35(1-2):237–250, 1989.
- [16] A. B. Olcay, T. S. Pottebaum, and P. S. Krueger. Sensitivity of Lagrangian coherent structure identification to flow field resolution and random errors. *Chaos (Woodbury, N.Y.)*, 20(1):017506, Mar. 2010.
- [17] T. Peacock and J. Dabiri. Introduction to focus issue: Lagrangian coherent structures. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(1):017501, 2010.
- [18] R. Peikert and F. Sadlo. Height Ridge Computation and Filtering for Visualization. In *Proc. Pacific Vis 2008*, pages 119 – 126, 2008.
- [19] A. Pobitzer, R. Peikert, R. Fuchs, H. Theisel, and H. Hauser. Filtering of FTLE for Visualizing Spatial Separation in Unsteady 3D Flow. In *Proc. of TopoInVis*, Zürich, Switzerland, 2011. Springer.
- [20] F. Sadlo and R. Peikert. Efficient visualization of lagrangian coherent structures by filtered AMR ridge extraction. *IEEE transactions on visualization and computer graphics*, 13(6):1456–63, 2007.
- [21] F. Sadlo, M. Ueffinger, T. Ertl, and D. Weiskopf. On the finite-time scope for computing lagrangian coherent structures from lyapunov exponent. In *Proc. of TopoInVis 2011*. Springer, 2011.
- [22] B. Schindler, R. Peikert, R. Fuchs, and H. Theisel. Ridge Concepts for the Visualization of Lagrangian Coherent Structures. *Proceedings of TopoInVis*, pages 1–14, 2011.
- [23] S. Shadden, F. Lekien, and J. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena*, 212(3-4):271–304, 2005.
- [24] S. C. Shadden. Lagrangian coherent structures. <http://mmae.iit.edu/shadden/LCS-tutorial/>, 2005.
- [25] G. M. Zaslavskii, R. Z. Sagdeev, D. A. Usikov, and A. A. Chernikov. *Weak Chaos and Quasi-Regular Patterns*. Cambridge University Press, 1991.
- [26] Z. Zhang. Identification of lagrangian coherent structures around swimming jellyfish from experimental time-series data. ., June 2008.