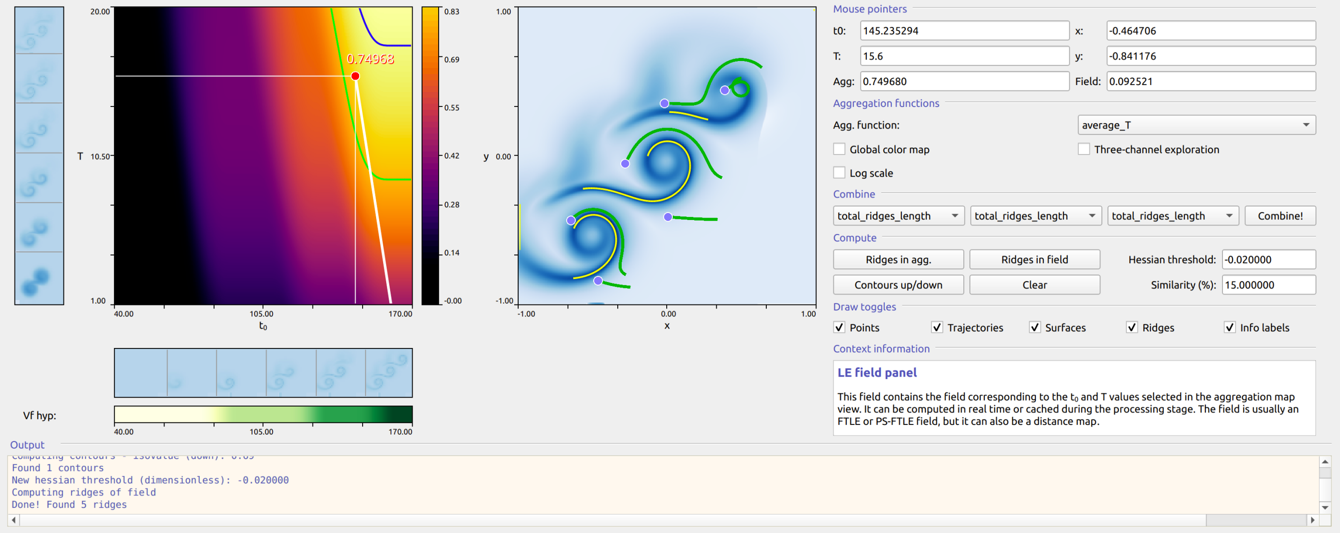
Eurographics Conference on Visualization (EuroVis) 2020
M. Gleicher, T. Landesberger von Antburg, and I. Viola
(Guest Editors)

Volume 39 (2020), Number 3

Visual Analysis of the Finite-Time Lyapunov Exponent

Antoni Sagristà, Stefan Jordan and Filip Sadlo

Heidelberg University, Germany



t0: 145.235294
x: -0.464706
T: 15.6
y: -0.841176
Agg: 0.749680
Field: 0.092521

Agg. function: average_T

□ Global color map □ Three-channel exploration

□ Log scale

total_ridges_length total_ridges_length total_ridges_length

Hessian threshold: -0.020000

Similarity (%): 15.000000

☑ Points ☑ Trajectories ☑ Surfaces ☑ Ridges ☑ Info labels

LE field panel

This field contains the field corresponding to the t_0 and T values selected in the aggregation map view. It can be computed in real time or cached during the processing stage. The field is usually an FTLE or PS-FTLE field, but it can also be a distance map.

```
Computing contours - isovalue (down): 0.69
Found 1 contours
New hessian threshold (dimensionless): -0.020000
Computing ridges of field
Done! Found 5 ridges
```

Figure 1: *Interactive analysis of the Three Gaussian Vortices dataset. In this session, we investigate the average aggregation function (left panel), and the corresponding finite-time Lyapunov exponent (FTLE) field (central panel) for time $t_0 = 145.23$ s and advection $T = 15.6$ s. The aggregation function captures the appearance of each vortex in the $[t_0, T]$ parameter space (purple, orange and yellow bands). Similarity contours in this panel (shown with a range of $\pm 7.5\%$ as blue and green lines) aid the visual analysis by delimiting areas with similar aggregation values. In the FTLE field panel, the six seed points (pale blue) spawn trajectories (green lines) in the underlying vector field. The height ridges extracted from the FTLE field are shown in yellow. To the right and bottom are the UI controls and the output console.*

Abstract

In this paper, we present an integrated visual analytics approach to support the parametrization and exploration of flow visualization based on the finite-time Lyapunov exponent. Such visualization of time-dependent flow faces various challenges, including the choice of appropriate advection times, temporal regions of interest, and spatial resolution. Our approach eases these challenges by providing the user with context by means of parametric aggregations, with support and guidance for a more directed exploration, and with a set of derived measures for better qualitative assessment. We demonstrate the utility of our approach with examples from computation fluid dynamics and time-dependent dynamical systems.

CCS Concepts

• ***Human-centered computing*** → ***Scientific visualization; Visual analytics;***

1. Introduction

A broad range of physical phenomena involves passive transport by some sort of time-dependent flow. Although such advection is typically superimposed with processes such as diffusion or inertia to some extent, these additional effects are often negligible, making advection-based flow analysis an important field of research.

In time-dependent flow, such transport does not only depend on *where* a tracer material is injected into the flow and for *what duration* it is transported, but also *at what time* it is released. Essentially, for an n-dimensional time-dependent vector field $\mathbf{u}(\mathbf{x},t)$, which assigns a vector $\mathbf{u} \in \mathbb{R}^n$ to each position $\mathbf{x} \in \Omega \subset \mathbb{R}^n$ within the spatial domain Ω at time t, advection is defined by the seeding position $\mathbf{x}$, the chosen seeding time t_0, and the advection duration T. Thus, it has $n+2$ degrees of freedom, or in other words, $n+2$ dimensions.

Since exploring the entire (n+2)-dimensional space would be prohibitively tedious, summarization and reduction strategies have proven very useful. A very successful approach in this field is vector field topology, which, instead of depicting all possible transport paths, shows only the essential ones, which separate regions of the domain with qualitatively similar flow behavior. In steady (time-independent) vector fields, this is achieved with traditional vector field topology [HH91, HH89], which is based on instantaneous advection curves, known as streamlines. Those streamlines that converge in forward or reverse time to saddle-type critical points, which are isolated zeros of $\mathbf{u}(\mathbf{x})$, separate such regions, and are thus named separatrices. In time-dependent flow, Lagrangian coherent structures (LCS) take over the role of separatrices. As motivated by Haller [Hal01] and shown by Shadden et al. [SLM05], LCS can be obtained as ridges, i.e., typically codimension-one manifolds with locally highest value, in the finite-time Lyapunov exponent (FTLE) field. The FTLE field $\sigma_{t_0}^{T}(\mathbf{x})$ is a scalar field in the n-dimensional domain Ω, and the $(n-1)$-dimensional LCS provide such a summarization and reduction—however, only for the seeding time t_0 and advection time T used to compute the FTLE field.

Although the FTLE is a very powerful tool for understanding advection in time-dependent flow, it has not reached wide application yet. Its comparably high computational cost (a trajectory has to be computed for each of its sample points in space and time) might not be the main reason for that, since several acceleration strategies have been proposed [GGTH07, SP07, SRP11]. It is much more its highly intricate parametrization and difficult interpretation that impede successful application by the non-expert. For example, T has to be chosen large enough to capture the phenomenon of interest, but small enough to prevent aliasing at the chosen spatial resolution of $\sigma_{t_0}^{T}(\mathbf{x})$. Also, analysis of the two-dimensional space spanned by t_0 and T for relevant structure is not amenable by direct interactive exploration without supporting context. It is the goal of this paper to provide an approach that overcomes such issues and helps FTLE-based flow analysis reach its deserved applications by making it easier to apply and interpret. Although large parts of our technique generalize to any dimension, we focus on two-dimensional vector fields and leave the extension to 3D for future work.

The contributions of this paper include:

- The concept of aggregation functions in the t_0–T space, and
- aggregation function definitions to analyze: basic properties,
- height ridge configurations,
- aliasing and resolution issues, and
- the overall "connectedness" of FTLE fields.
- Finally, an integrated framework to aid in the exploration and analysis of these aggregation functions and the FTLE.

2. Related Work

To the best of our knowledge, this is the first work that applies visual analytics methods to aid in the application and interpretation of FTLE-based flow analysis. A good introduction to time-dependent vector field topology is provided by Pobitzer et al. [PPF*11], and an overall survey on topology-based methods by Heine et al. [HLH*16]. For steady vector field topology, Asimov [Asi93] can provide a starting point to the literature. The seminal paper by Haller [Hal01] presented a computation of the FTLE based on the flow map, and proposed LCS to be ridges in the FTLE field, which has later been confirmed by Shadden et al. [SLM05], with quantitative guarantees. Eberly [Ebe96] introduces the concept of height ridges, and Peikert et al. [PS] explore and assess different methods for their computation and filtering. The current work makes extensive use of these techniques, and presents a framework for interactive exploration and analysis of both the FTLE field and the LCS as height ridges. Combining FTLE and flow visualization with integrated interactive systems has also been addressed before, for example, as a framework to aid in the topological analysis of inertial particles [SJJ*17] but also in the realm fluid dynamics in general [SRBE]. Our work also borrows from Weiskopf's [Wei07] in his exploration of GPU-based interactive visualization techniques of scalar and vector fields. In the field of visualization systems, the Topology ToolKit [TFL*18] implements most of the techniques mentioned above and offers an open-source platform for topological analysis in visualization, while VisTrails [BCC*05] offers a provenance infrastructure which maintains a detailed history of data and workflows in the visualization process. The latter would certainly represent a fine companion to our system in order to ensure the persistence and reproducibility of results.

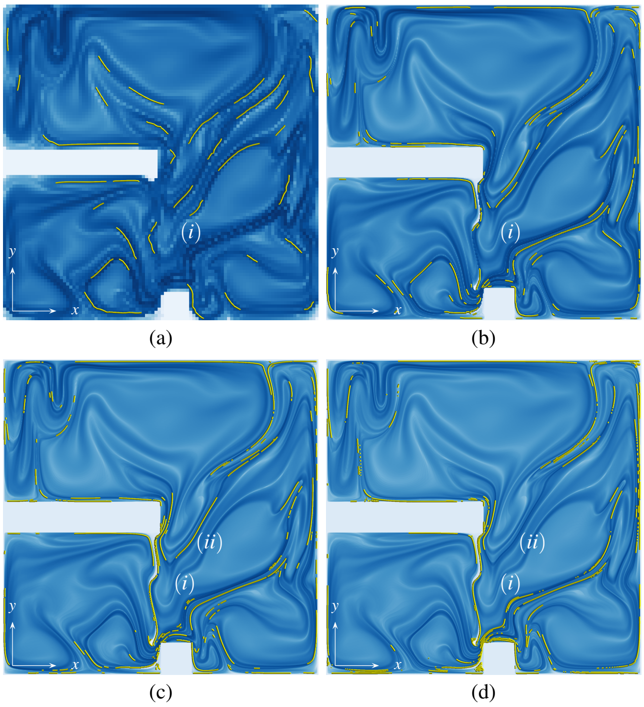
Additional relevant work also addresses the visual analysis of time-dependent data and flow fields. In this direction, Aigner et al. [ABM*07] study the introduction of time as an additional dimension in visual analytics, Bürger et al. [BMI*] integrate local feature detectors in the visual analysis of time-dependent flow simulations, Shi et al. [STH*07] present an approach to visually analyze time-dependent flow fields through the behavior of the pathlines, and Doleisch, Hauser and co-authors [DGH, DMG*] study the visual analysis of complex time-dependent flow simulations and real data. In this context, the current work provides an interactive analysis of trajectories, and introduces the concept of FTLE aggregation fields, which attempt to capture global trends and features in the seeding and advection time dimensions.

3. Fundamentals

For completeness, we first provide some additional basics on the FTLE (Section 3.1) and height ridges (Section 3.2), followed by descriptions of individual challenges of FTLE-based flow analysis together with our respective solutions (Section 4).

3.1. FTLE

Height ridges (Section 3.2) in the FTLE field represent the counterpart to separatrices, i.e., LCS separate spatial regions of similar time-dependent advection. Haller [Hal04] proposes to base the computation of the FTLE field on the flow map $\boldsymbol{\phi}_{t_0}^{T}(\mathbf{x})$, which maps particles seeded at position $\mathbf{x}$ and time t_0 to their final position after advection for time T, i.e., for each pathline (or trajectory), it maps from the start point to its end point. In this approach, the FTLE



(a) (b) (c) (d)

Figure 2: *Height ridge extraction from the Buoyant Flow dataset with four different FTLE field resolutions of 100 × 100 (a), 400 × 400 (b), 800 × 800 (c), and 1600 × 1600 (d), with identical t_0 = 0.09 s and T = 0.088 s. In each case, the ridge filter threshold τ_λ has been adjusted according to an expert's choice to −0.07 (a), −0.06 (b), −0.04 (c), and −0.03 (d). The resolutions (a) and (b) are not able to capture the two aligned, very close ridges in the bottom right quadrant (i) in (a) and (b). However, the ridges are already present, albeit broken, when we increase the resolution (c), and they are much better captured at (d), although even higher resolution would be required. The ridge (ii) is present in (c) but disappears when we double the resolution (d) due to its insufficient sharpness with respect to the increased resolution.*

computes

$$\sigma_{t_0}^{T}(\mathbf{x}) := \frac{1}{|T|} \ln \left\| \nabla \boldsymbol{\phi}_{t_0}^{T}(\mathbf{x}) \right\|_2 , \tag{1}$$

with $\|\cdot\|_2$ representing the spectral norm (the square root of the largest eigenvalue of $A^\top A$ for a matrix A). Example FTLE fields are shown in Figure 2. Notice that increasing T typically leads to a decrease of the overall FTLE values, since the particles would need to diverge exponentially over the entire duration T to keep the overall FTLE values constant. We counteract this effect, which can hinder visual analysis of FTLE behavior, by optional multiplication of the FTLE values with $|T|$, prior to, e.g., applying aggregation functions (see Section 4). Notice also that the sampling of the FTLE grid does not need to (and typically does not) coincide with the grid nodes of the vector field $\mathbf{u}(\mathbf{x},\mathbf{t})$. In fact, the spatial resolution of the FTLE grid is an important parameter, which needs to be chosen sufficiently high to prevent aliasing, or in other words, to appropriately capture the structure of the FTLE. Furthermore, notice that we do not employ acceleration strategies for FTLE computation (except for parallel computation on the GPU), and that we need to sample it for a region of interest of the space spanned by t_0 and T, including derived measures, which leads to very expensive computation. Thus, we employ preprocessing, which, however, is carried out as a batch process since it requires no interaction.

3.2. Height Ridges

A codimension-one ($(n-1)$-dimensional) height ridge in a scalar field $s(\mathbf{x})$, with $\mathbf{x} \in \Omega$, can be obtained according to Eberly [Ebe96] by extracting the zero-isocontour from a derived scalar field:

$$\nabla s(\mathbf{x}) \cdot \boldsymbol{\varepsilon}_{\min}(\mathbf{x}) = 0 , \tag{2}$$

where $\boldsymbol{\varepsilon}_{\min}(\mathbf{x})$ is the minor eigenvector of the Hessian $\nabla\nabla s(\mathbf{x})$, with the additional requirement of the minor eigenvalue $\lambda_{\min}$ of the Hessian being negative. In discrete (grid-based) settings, this isocontour extraction can be accomplished based on the marching squares algorithm [LC87], with an additional step that makes eigenvector orientation consistent [FP01]. Additionally, it is common to suppress spurious solutions (noise) by rejecting those parts of the solutions where $\lambda_{\min}$ is not sufficiently small, i.e., one requires $\lambda_{\min} < \tau_\lambda$ with a user-defined threshold $\tau_\lambda \leqslant 0$. We also employ such filtering in our examples, and document our choice for τ_λ accordingly. As demonstrated in Figure 2, the main issue with height ridge extraction from FTLE fields is typically not (numerical) noise due to the involved second derivatives, but aliasing and ridges that are closer together than the support size of the discrete second-derivative operator, affecting estimation of the Hessian. We address both issues in Sections 4.3 and 4.4. Other approaches to ridge extraction attempt to solve the numerical noise problem, including the filtered AMR ridge extraction by Sadlo et al. [SP07], which is based on the determination of the gradient by least squares and also works for unstructured grids, and the feature extraction method by Kindlmann et al. [KCH*18], implemented in Diderot [KCS*16], which solves the numerical noise issue by using ray casting and advanced interpolation schemes.

4. Approach

We now describe our visual analytics approach in terms of derived measures and their visual representation. In Section 5, we then focus on interaction and implementation aspects, followed by an evaluation (Section 6). Performance and optimization considerations are discussed in detail in Section 7.

4.1. Aggregation Fields

The most basic challenge with FTLE-based flow visualization is the selection of a seeding time t_0 together with an appropriate advection time T. In addition, the temporal structure of vector fields can be very rich, making it difficult to determine relevant instants of time t_0. Notice that it is common practice in FTLE-based visualization to choose a T and then use it to compute the FTLE field $\sigma_{t_0}^{T}(\mathbf{x})$ for a finely resolved sequence of t_0, leading to respective animations. Whereas such animated visualizations might work quite well once the spatiotemporal region of interest, as well as the

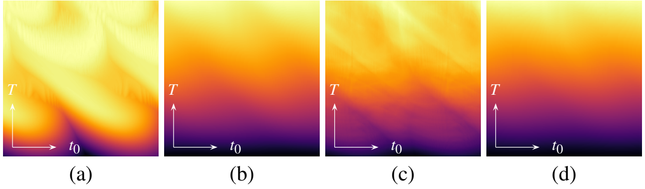


Figure 3: *Basic aggregation functions for the Quad-Gyre dataset with aggregation field resolution $G_{\text{agg}} = 200 \times 200$, FTLE resolution $G = 200 \times 200$, $t_0 \in [0,8]$, and $T \in [1,8]$. $f_{\sigma_{\text{avg}}}$ (b) displays a very soft structure apart from the direct proportionality with T. By contrast, $f_{\sigma_{\text{max}}}$ (a) captures the periodicity of the dataset surprisingly well. We also provide, for comparison, the 95th percentile (c), and the sum of squares (d) aggregation fields.*

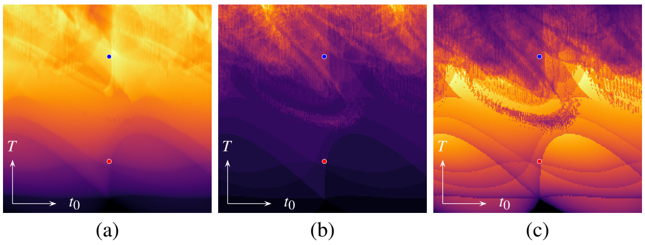


Figure 4: *$f_{\rho_{\text{len}}}$ (a), $f_{\rho_{\text{cnt}}}$ (b), and $f_{\hat{\rho}_{\text{len}(k)}}$ (c) aggregation fields, applied to the Quad-Gyre dataset, with $\tau_\lambda = -0.05$ and the same parameters as for Figure 3. Note that (a) shows a rather continuous field, while (b) contains only integer numbers of ridges, and (c) displays a similar structure due to its dependency on the number of ridges. Additionally, while ρ_{len} increases with T due to longer, sharper ridges, $\hat{\rho}_{\text{len}(k)}$ exhibits in (c), toward higher T values, a sharp drop in the quality of extracted ridges caused by aliasing. The rather noisy top part in all fields shows that the chosen spatial resolution is unable to capture the ridges at high T values well, so they break up in an unpredictable manner causing these patterns.*

appropriate advection time, has been determined, they do not provide temporal context, nor do they support finding such relevant regions of interest. Additionally, the exploration of t_0 is complicated by the fact that t_0 and T are intertwined—their selections impact each other. And furthermore, processes at different locations and different times t_0 in a given vector field typically require different choices of T, imposing hard challenges in choosing appropriate combinations of t_0 and T.

We support the exploration of regions of interest in the space spanned by t_0 and T (denoted $\Omega_{t_0,T}$) by introducing *aggregation fields* $f_\alpha(\mathbf{y})$:

$$f_\alpha : \Omega_{t_0,T} \to \mathbb{R} \,, \tag{3}$$

which map each point $\mathbf{y} \in \Omega_{t_0,T} \subset \mathbb{R}^2$ to the scalar result of an aggregation function α. That is, each point $\mathbf{y}$ corresponds to an FTLE field $\sigma_{t_0}^T(\mathbf{x})$, and the aggregation function

$$\alpha : (\Omega \to \mathbb{R}) \to \mathbb{R} \tag{4}$$

takes the *field* $\sigma_{t_0}^T(\mathbf{x})$ as input and outputs a single scalar value. Aggregation fields are displayed in the aggregation panel (Figure 11) of our system, providing insight into $\Omega_{t_0,T}$ in FTLE-based flow analysis. Notice that we map t_0 to the abscissa and T to the ordinate in the visual analytics framework.

We found the following aggregation functions particularly useful for summarizing properties of FTLE fields and LCS in a single scalar value. We divide them into four distinct groups: basic aggregation (Section 4.2), ridge aggregation (Section 4.3), aliasing aggregation (Section 4.4), and region aggregation (Section 4.5), according to their use.

4.2. Basic Aggregation Functions

As motivated above, a basic need in FTLE-based flow visualization is to support the exploration of combinations of t_0 and T. We evaluated the summarization of an FTLE field for each $\mathbf{y} \in \Omega_{t_0,T}$ by aggregation functions computing its minimum, maximum, average, median, sum of squares, root mean square, and 95th percentile—and identified the maximum and average as the generally most useful ones. The maximum aggregation function is defined as follows:

$$\sigma_{\max}(\sigma_{t_0}^T(\mathbf{x})) := \max_{\hat{\mathbf{x}} \in \hat{\Omega}} \sigma_{t_0}^T(\hat{\mathbf{x}}) \,, \tag{5}$$

with $\hat{\Omega}$ being the discrete domain of the FTLE field, i.e., the set of sampling grid nodes of our node-based representation of $\sigma_{t_0}^T(\mathbf{x})$, and $\hat{\mathbf{x}}$ representing such a node. An example of the resulting aggregation field $f_{\sigma_{\max}}(\mathbf{y})$ is shown in Figure 3a.

The average aggregation function is defined as:

$$\sigma_{\text{avg}}(\sigma_{t_0}^T(\mathbf{x})) := \frac{1}{|\hat{\Omega}|} \sum_{\hat{\mathbf{x}} \in \hat{\Omega}} \sigma_{t_0}^T(\hat{\mathbf{x}}) \,, \tag{6}$$

with $|\hat{\Omega}|$ being the cardinality of $\hat{\Omega}$, i.e., the number of nodes in the FTLE sampling grid. An example for the aggregation field $f_{\sigma_{\text{avg}}}(\mathbf{y})$ resulting from the average aggregation function $\sigma_{\text{avg}}(\cdot)$ is shown in Figure 3b.

4.3. Ridge Aggregation Functions

Since FTLE ridges represent LCS, which in turn are the topological features of time-dependent vector fields, the total amount of ridges is a basic measure for the topological structure of a time-dependent vector field. Thus, our first ridge aggregation function

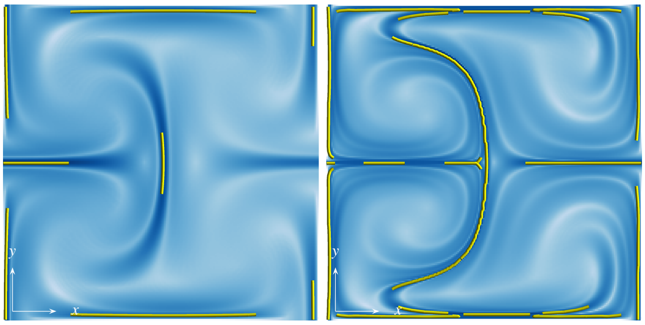
$$\rho_{\text{len}}(\sigma_{t_0}^T(\mathbf{x})) := \sum_{r \in \mathcal{R}} \mu(r) \,, \tag{7}$$

simply measures the total length of all height ridges extracted from the FTLE field $\sigma_{t_0}^T(\mathbf{x})$, with $\mathcal{R}$ being the set of all ridges (in polyline representation), and $\mu(r)$ measuring the length of ridge r. Figure 5 shows an example result of the ridge extraction process and is annotated with the respective ρ_{len} values. Figure 4a shows the aggregation field $f_{\rho_{\text{len}}}(\mathbf{y})$, resulting from the ridge length aggregation function $\rho_{\text{len}}(\cdot)$.

The number of ridges, on the other hand, can capture if ridges are disrupted, e.g., due to insufficient resolution of the FTLE field. This is a major issue in FTLE-based visualization (Figure 2). Thus, our next ridge aggregation function

$$\rho_{\text{cnt}}(\sigma_{t_0}^T(\mathbf{x})) := |\mathcal{R}| \,, \tag{8}$$

counts the number of height ridges extracted from the FTLE



(a) (b)

Figure 5: *Same dataset as Figures 3 and 4, showing the result (yellow) of the ridge extraction process with seeding time $t_0 = 4.0\,s$ and advection time $T = 2.75\,s$ (a), corresponding to the red dots in Figure 4, and $T = 6.25\,s$ (b), corresponding to the blue dots in Figure 4. The value of ρ_{len} in (a) is 5.09, and 12.18 in (b). The aggregation fields show trends which would otherwise be impossible to observe just by looking at the single FTLE fields (higher values by higher saturation).*

field $\sigma_{t_0}^T(\mathbf{x})$, with $|\mathcal{R}|$ being the number of elements in $\mathcal{R}$. Figure 4b shows the respective aggregation field $f_{\rho_{\text{cnt}}}(\mathbf{y})$.

Since LCS, i.e., FTLE ridges, separate regions of qualitatively different time-dependent advection, their individual length is also important—a long ridge represents a larger barrier in the domain and thus also represents a more significant topological structure (Figure 5). This motivates our last ridge aggregation function

$$\hat{\rho}_{\text{len}(k)}(\sigma_{t_0}^T(\mathbf{x})) := \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \mu(r)^k \,, \tag{9}$$

which measures the averaged kth power of the ridge lengths. It provides a good measure of the overall quality of the ridge extraction stage, and, implicitly, discretization quality (appropriateness of the resolution) of the FTLE field. Figure 4c shows the respective aggregation field $f_{\hat{\rho}_{\text{len}(k)}}(\mathbf{y})$ for $k = 1$.

4.4. Aliasing Aggregation Function

The ridge count $\rho_{\text{cnt}}(\cdot)$ and ridge length power $\hat{\rho}_{\text{len}(k)}(\cdot)$ aggregation functions are already able to indicate aliasing issues in the FTLE field $\sigma_{t_0}^T(\mathbf{x})$. However, they do that only indirectly via the properties of the ridges extracted from $\sigma_{t_0}^T(\mathbf{x})$. Since these ridge aggregation functions also capture the structure of the LCS and are thus superimposed by LCS properties, we complement them with a new aggregation function that quantifies aliasing in a direct way.

Aliasing in FTLE fields, as apparent in Figure 2a, is caused by a too low FTLE field resolution compared to the gradients (frequencies) of the FTLE field. It is in particular the very low width of FTLE ridges that makes the FTLE field so hard to sample. This is all the more challenging as this width is related inversely to the advection duration T. That is, the larger T, the longer typically the LCS, and the sharper the FTLE ridges become. In confined domains, the flow has to turn at some point and will cause the ridges to fold, leading to closely adjacent ridges by the so-called stretching and folding mechanism [Sma67]. To the best of our knowledge, there is no analytic result on the relation between FTLE advection time and ridge width, but due to the underlying assumption of exponential divergence, and motivated by our observations, we assume that ridge width decreases exponentially, too, which imposes a very hard challenge on appropriate FTLE sampling. That is, increasing T typically requires much stronger increase of the spatial sampling resolution, otherwise aliasing will appear.

Motivated by the Nyquist–Shannon sampling theorem, we indicate aliasing by analyzing the frequency spectrum of the FTLE field. That is, we use the Fourier transform to obtain the 2D spectrum of the field, and measure the amplitude of the highest frequencies in that spectrum. Our motivation to do so is that if the original signal (in our case the underlying true FTLE field that we are discretizing using the discrete flow map) had frequencies higher than the Nyquist frequency, it would be very likely that the highest frequencies in the discretized signal (our sampled FTLE field) have non-negligible amplitudes. Conversely, negligible amplitudes in the highest frequencies in the FTLE can indicate appropriate sampling.

We realize such a quantification of the amplitude of the highest frequencies of an FTLE field $\sigma_{t_0}^T(\mathbf{x})$ by first transforming it to the 2D frequency domain $\tilde{\Omega} \subset \mathbb{R}^2$ using the fast Fourier transform [CT65], resulting in the spectrum $\tilde{\sigma}_{t_0}^T(\boldsymbol{\xi})$ for $\boldsymbol{\xi} \in \tilde{\Omega}$. Notice that in this representation, zero frequency is at the origin $\boldsymbol{\xi}_0 = \mathbf{0}$ and $\tilde{\Omega}$ is symmetric about the origin. As a consequence, points $\boldsymbol{\xi}$ at equal distance ν from the origin, i.e., $\|\boldsymbol{\xi}\| = \nu$, represent equal frequencies, or in other words, circles in $\tilde{\Omega} \subset \mathbb{R}^2$ (about the origin with radius ν) represent all amplitudes belonging to the frequency ν. Thus, to quantify the amplitude of the m highest percent of frequencies in the FTLE field $\sigma_{t_0}^T(\mathbf{x})$, we define the following aliasing aggregation function, which integrates the spectrum $\tilde{\sigma}_{t_0}^T(\boldsymbol{\xi})$ along these circles:

$$\nu_{\text{sum}(m)} := \iint_{\boldsymbol{\xi} \in \mathcal{X}} \tilde{\sigma}_{t_0}^T(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi} \,, \tag{10}$$

with

$$\mathcal{X} := \left\{ \boldsymbol{\xi}' \;\middle|\; \|\boldsymbol{\xi}'\| \geq \tfrac{100-m}{100} \hat{\nu} \right\} , \tag{11}$$

and with $\boldsymbol{\xi}' \in \tilde{\Omega}$, and $\hat{\nu}$ being the Nyquist frequency, i.e., the spatial resolution of our discretized $\sigma_{t_0}^T(\mathbf{x})$.

Figure 6 shows the resulting aliasing aggregation field $f_{\nu_{\text{sum}(m)}}(\mathbf{y})$ for choices of increasing FTLE resolution. One can see that as the FTLE resolution increases, the area within $\Omega_{t_0,T}$ with negligible highest frequencies (low aliasing) extends to the top, i.e., our aggregation indicates that longer advection times T can be used.

Typically, a correlation between the amount of aliasing and the ridge length aggregation $\rho_{\text{len}}(\cdot)$ can be observed. The main cause for this is that aliasing is typically caused by very sharp (thin) ridges, that cannot be resolved with the chosen spatial resolution. As discussed above, such long and sharp ridges appear in particular with longer advection times—and if the resolution is chosen too low, ridge extraction typically produces disrupted lines, which reflects in low $\hat{\rho}_{\text{len}(k)}$ values (top in Figure 4c). Our system provides the option to map aggregation fields to different color channels for combined analysis for such purposes, as demonstrated in Figure 7.

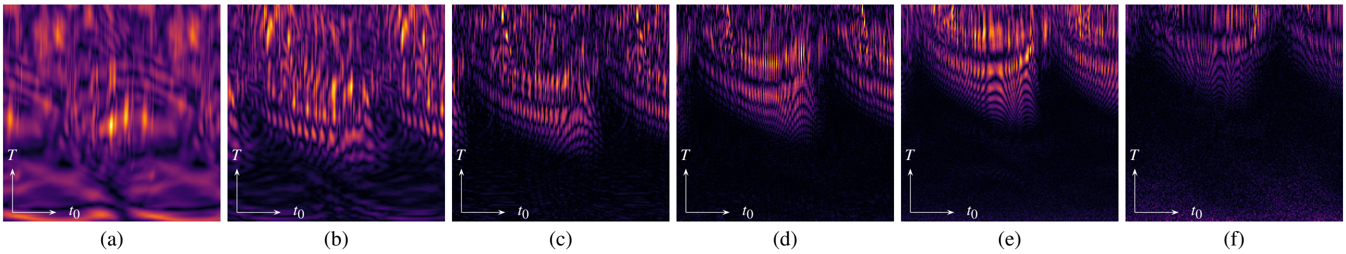


Figure 6: *Aliasing aggregation field $f_{\mathrm{v}_{\mathrm{sum}(2)}}$ for the Quad-Gyre dataset with resolution $G = 25 \times 25$ (a), 50×50 (b), 100×100 (c), 200×200 (d), 400×400 (e), and 800×800 (f). G_{agg} is, in all cases, 200, $T \in [1,8]$, and $t_0 \in [0,8]$. Note the dependency of the aliasing aggregation field on advection time T, and how the aliasing pattern moves to larger T (top) as the field resolution G increases.*

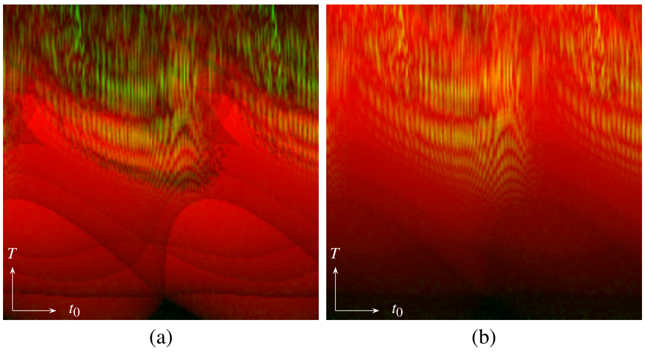


Figure 7: *Aliasing aggregation field $f_{\mathrm{v}_{\mathrm{sum}(2)}}$ mapped to the green channel, and $f_{\hat{\rho}_{\mathrm{len}(k)}}$ (a), and $f_{\rho_{\mathrm{len}}}$ (b) mapped to the red channel for the dataset from Figures 4a, 4c, and 6d. Note that in (a), the fields are somewhat complementary because the ridge quality decreases as the aliasing problems increase, whereas in (b), they both show the same increasing trend with T.*

4.5. Region Aggregation Functions

Our final aggregation function goes one step further and focuses on the interpretation of FTLE fields. As discussed, FTLE ridges, the LCS, separate regions of qualitatively different time-dependent flow behavior. However, whereas separatrices in steady vector field topology indeed separate such regions entirely (assuming integration time of the streamlines goes to infinity), LCS typically do not. That is, LCS do generally not partition the domain—they typically leave gaps (see Figure 5b). The reason for that is that topology of aperiodic time-dependent flow does usually not have the opportunity to consider the limit case $T \to \infty$. Instead, finite-time considerations (with respect to finite advection time T) have to be employed, leaving gaps between LCS due to the time-local nature of the topological processes.

We account for these circumstances by introducing a measure to quantify the connectedness of the domain Ω with respect to "obstacles", which in our case are the FTLE ridges. Due to the high computational complexity of this measure, we define a separate (typically lower-resolution) sampling grid in the FTLE domain Ω, whose grid nodes we denote $\hat{\mathbf{y}} \in \mathcal{D}$ (Figure 8 contains a comparison of different $\hat{\mathbf{y}}$-resolutions). We then discretize the FTLE ridge set $\mathcal{R}$ at the resolution of the FTLE (at nodes $\hat{\mathbf{z}}$), by setting ridge nodes as "background" (white in Figure 8) and non-ridge nodes to "foreground". In other words, we binarize the ridge lines into the grid $\hat{\mathbf{z}}$, and collect its foreground nodes in a set $\mathcal{F}$ and its background nodes in a set $\mathcal{B}$. Based on that, we define our connectedness field $\gamma(\hat{\mathbf{z}})$ at a given node $\hat{\mathbf{z}}$ as follows:

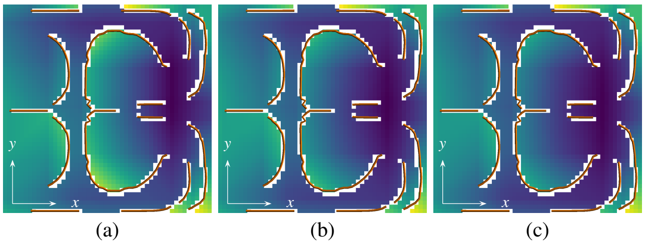


Figure 8: *The same connectedness field of the Quad-Gyre dataset with FTLE resolution $G = 50 \times 50$ and different $\hat{\mathbf{y}}$-resolutions $G_{\mathrm{s}} = 5 \times 5$ (a), $G_{\mathrm{s}} = 10 \times 10$ (b), and the same as G, $G_{\mathrm{s}} = 50 \times 50$ (c). Observe the small difference between the version where $G_{\mathrm{s}} = G$ (c) and the version using a hundred times less samples (a). This example shows that the connectedness of the dataset is equally well captured by all sampling resolutions.*

$$\gamma(\hat{\mathbf{z}}) := \frac{1}{|\mathcal{F}|} \sum_{\hat{\mathbf{y}} \in \mathcal{D}} \mu\big(A^*(\hat{\mathbf{z}}, \hat{\mathbf{y}})\big) \ , \tag{12}$$

with $|\mathcal{F}|$ being the number of foreground, i.e., non-ridge nodes, $A^*(\hat{\mathbf{z}}, \hat{\mathbf{y}})$ being the shortest path between nodes $\hat{\mathbf{z}}$ and $\hat{\mathbf{y}}$ according to the A* algorithm (avoiding the background, i.e., ridge obstacles), and $\mu(\cdot)$ being the length of such a path. If such a shortest path between two foreground nodes $\hat{\mathbf{z}}$ and $\hat{\mathbf{y}}$ does not exist (e.g., because it is completely blocked by LCS and/or boundaries), the distance is set to the perimeter of the domain Ω. If one of the nodes (or both) are contained in $\mathcal{B}$, the respective length is set to zero. Figures 9c and 9d show an example of the connectedness field resulting from the respective ridge configurations (orange lines).

To obtain an aggregation field, we need to combine all values of the connectedness field $\gamma(\hat{\mathbf{z}})$ into a single scalar value. We accomplish this by employing the basic aggregation functions (Section 4.2) to $\gamma(\hat{\mathbf{z}})$ instead of $\sigma_{t_0}^{T}(\mathbf{x})$, resulting in the region aggregation fields $f_{\gamma_{\max}}(\mathbf{y})$ (Figure 9a) and $f_{\gamma_{\mathrm{avg}}}(\mathbf{y})$ (Figure 9b).

Even though there is a correlation between the total ridge length aggregation field $f_{\rho_{\mathrm{len}}}(\mathbf{y})$ and the connectedness aggregation field $f_{\gamma}(\mathbf{y})$ (Figure 10), the latter is able to account for almost isolated or even closed regions, in contrast. Additionally, $f_{\rho_{\mathrm{len}}}(\mathbf{y})$ does

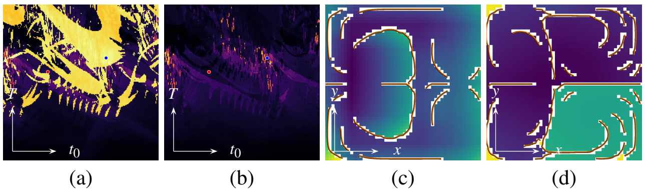


Figure 9: *Region aggregation fields (a)–(b) and connectedness fields (c)–(d) for the Quad-Gyre dataset. Resolutions are $G_{\text{agg}} = 200 \times 200$ and $G = 50 \times 50$. $f_{\gamma_{\text{avg}}}$ (b) shows a strong dependency on T, but also some changes along t_0, indicating different degrees of connectedness for different time settings. By contrast, $f_{\gamma_{\max}}$ (a) clearly displays the fields which have closed areas in yellow. The red marker corresponds to the connectedness field (c), at $t_0 = 2.3$ s and $T = 5.0$ s, which exhibits no closed regions. On the other hand, the blue marker in (a) corresponds to (d), at $t_0 = 5.3$ s and $T = 5.6$ s, which has four isolated regions. In the connectedness fields, the ridge lines (orange) are rasterized in white.*

not take into account the configuration and layout of the ridges in terms of topological structure.

5. System Description

The system is composed of a modular, interactive and customizable front-end and a versatile computing back-end, which enable the user to analyze, examine, and understand the temporal and spatial properties of the system under study. The software can be set up using command line arguments, to compute a user-selected list of aggregation fields, or it can load precomputed runs from files. Additional options include starting an interactive session or only compute in the background, and persisting the results to disk. The software will be open sourced and published at a later date. In the following, we describe its capabilities and features in detail.

Main layout. Our system consists of two panels laid out side-by-side (Figure 11), plus a set of UI elements and controls in charge of receiving and presenting information to the user. The left panel ① contains the aggregation field displaying one or more of the aggregation functions, optionally with logarithmic scale. The panel right next to it ② displays the FTLE field for a selected location in the aggregation space. The FTLE field panel is automatically updated as the user hovers the cursor over the aggregation field.

Aggregation panel. The aggregation panel ① contains the currently selected aggregation field with the user-selected resolution $G_{\text{agg}} = N_{t_0} \times N_T$. Two timelines are available to the left and bottom of the aggregation panel ③. These contain six "field icons" at different positions in t_0 (bottom) and T (left) space. The timelines are also automatically updated with the active mouse position in the aggregation field. Additionally, we display integrated hyperbolicity $\det(\nabla \mathbf{u}(\mathbf{x}))$ of the vector field (which tends to quantify the separating dynamics causing LCS) aligned with the seeding time on the horizontal axis, below the timeline ④. A color legend is provided to the right of the aggregation panel ⑤, and it is updated automatically when the aggregation field changes.

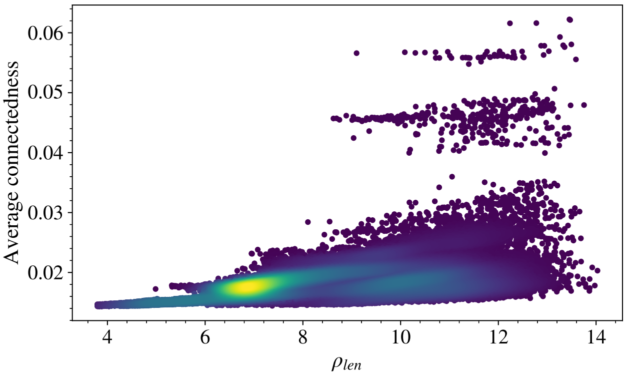


Figure 10: *Scatterplot with density coloring of ρ_{len} (yellow = high), with respect to the average region aggregation field $f_{\gamma_{\text{avg}}}$, for the Quad-Gyre dataset. Note that for shorter ridge lengths, the relation is quasi-linear, but as the total ridge length increases, the values of the average region aggregation field spread out, giving rise to very different connectedness configurations for fields with the same total ridge length. The clusters centered around different average connectedness values (0.02, 0.045, 0.055) correspond to different closed-region configurations. For instance, the topmost cluster contains all configurations where the map is vertically divided into two similarly-sized disconnected regions. The second cluster corresponds to configurations where roughly a quarter of the map is a closed region (see Figure 9d). The rest corresponds to different configurations of smaller-sized closed regions. Note that within each cluster, there is a considerable deviation in average connectedness, proving that this aggregation function is actually able to measure the connectedness quite well, even when closed regions are present.*

FTLE panel. The FTLE panel ② displays the finite-time Lyapunov exponent field for the active coordinate in the aggregation panel, with the user-selected resolution $G = N \times M$. It is recomputed and updated automatically whenever this coordinate changes. Moving the pointer within the FTLE panel, while pressing the control key, spawns a trajectory seed, which follows the pointer movement. Clicking on the field while pressing the control key creates permanent trajectory seeds. All used colormap schemes can be changed at runtime using the *View* menu. The system allows for global and local scaling of colormaps. All images in this paper use local scaling. The colormaps used by default include the *blues* colormap (white to blue, used in regular FTLE fields), the *inferno* colormap (black to purple to orange to yellow, used in the aggregation fields) and the *viridis* colormap (purple to blue to green to yellow, used in the region aggregation fields).

Contours and ridges. Similarity contours can be computed in the aggregation panel in order to aid in the visual exploration of the aggregation field. Ridge extraction can be applied to either the aggregation or the FTLE fields. The ridge extraction threshold (τ_λ) defaults to the one used for the computation of the ridge aggregation fields. The system also supports interactive exploration of different values for τ_λ.

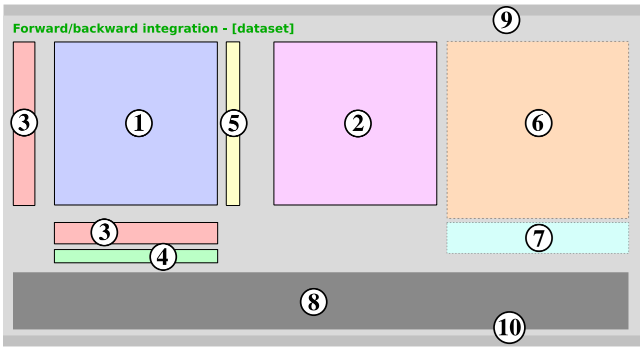


Figure 11: *Illustration of the components in our system. Aggregation panel ①, FTLE panel ②, vertical and horizontal timelines ③, vector field hyperbolicity panel ④, aggregation field color legend ⑤, UI controls ⑥, contextual information pane ⑦, output console ⑧, main menu ⑨, and status bar ⑩.*

Three-way exploration. The system supports a three-way simultaneous exploration mode (Figure 7) for aggregation fields. Any two or three aggregation fields may be mapped to the red, green, and blue channels, in order to explore their combined properties. More interestingly, a three-way exploration mechanism is also offered in the FTLE field panel. Once enabled, the user can successively click on positions in the aggregation panel to map the corresponding fields to the red, green, and blue channels.

Input/output controls. The controls are positioned to the right of the FTLE panel ⑥, and contain the necessary elements to manage the behavior and settings of the current session. In order to guide the user through the visual analysis process and improve understanding, the system offers a contextual information box ⑦, which shows comprehensive information of the currently selected/highlighted elements. The current output log is shown in the console output panel ⑧. The top menu bar ⑨ contains shortcuts to many functions, such as saving the current session, or pausing the update thread. Finally, a summary of the state of the current session is displayed at the bottom of the window, in the status bar ⑩. We refer the reader to the accompanying video, which contains extensive demonstrations of all these features and more.

6. Results

Our recommended exploration flow, as gathered by the experts' experience with the tool, starts by analyzing the basic aggregation in order to get a sense of the general trends that may be captured by the fast statistical functions, followed by the ridge and aliasing aggregation fields, which help narrow down the most interesting parameter subspaces, to end with the region aggregation to gain insight into the connectedness properties of the dataset. The system allows, if required, interactive exploration of the FTLE fields in context with the aggregation space, as well as interactive fine-tuning of key parameters such as the ridge extraction threshold. Additionally, recomputation of interesting sub-spaces at different resolutions and with different parameters may also be required. With these guidelines in mind, we analyze an analytic example (Section 6.1), a 2D fluid dynamics simulation (Section 6.2), a constructed dataset based on appearing Gaussian vortices (Section 6.3), an inertial n-body system (Section 6.4), and an atmospheric wind dataset based on real measurements (Section 6.5).

6.1. Quad-Gyre

This example was introduced by Shadden et al. [SLM05] for the analysis of Lagrangian coherent structures. The vector field is defined by the following analytic representation:

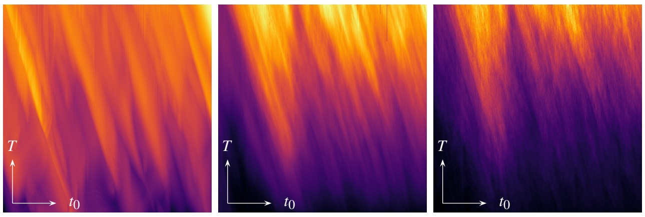
$$\mathbf{a}(\mathbf{x},t) = \begin{pmatrix} -\pi A\sin(\pi f(x,t))\cos(\pi y) \\ \pi A\cos(\pi f(x,t))\sin(\pi y)\frac{\mathrm{d}f}{\mathrm{d}x} \end{pmatrix}, \tag{13}$$

with $f(x,t) = a(t)x^2 + b(t)x$, $a(t) = \varepsilon\sin(\omega t)$, $b(t) = 1 - 2\varepsilon\sin(\omega t)$, $\mathbf{x} = (x,y)^\top$, $\varepsilon = 0.25$, $\omega = \pi/5$, and $A = 0.1$. We sampled the field at a resolution of 50×50 and 100 time steps in the spatial range $\mathbf{x} \in [-1,1] \times [-1,1]$ and temporal range $[0,20]$. Note that we use this y-range instead of the often employed $[0,1]$, leading to a y-symmetric field with four time-dependent vortices instead of two.

We compute all basic, ridge, aliasing, and region aggregation fields for $T \in [1,8]$ and $t_0 \in [0,8]$. Starting with the basic aggregation fields in Figure 3, we can see in $f_{\sigma_{\mathrm{avg}}}$ (Figure 3b) very vague recurring features in t_0 and a strong dependency on T. This periodicity is captured much better by $f_{\sigma_{\max}}$ (Figure 3a). A particular feature of this dataset is that when the main vertical LCS crosses from right to left and vice-versa, it is less sharp, resulting in lower FTLE values. This is captured by $f_{\sigma_{\max}}$ and shown as a valley. $f_{\rho_{\mathrm{len}}}$ shows an obvious proportionality with T (Figure 4a), and some extra periodic structure that shows variations in the total length of ridges. Figure 5 shows the actual FTLE field and ridges with two different advection times for this dataset. $f_{\hat{\rho}_{\mathrm{len}(k)}}$ (Figure 4c) shows that the central values of T (between 3 s and 5 s) are the best candidates for high-quality ridges. An interactive exploration with the system reveals that at higher T values, the ridges break up, and the current resolution is not able to capture them well. The aliasing aggregation field supports this analysis. Figure 6a shows that the field resolution of 25×25 is unable to properly sample the field anywhere in the aggregation space, whereas Figure 6e shows that the resolution of 400×400 does not present problems when $T < 4$s. In particular, looking at the aliasing field for the resolution of 200×200 (Figure 6d) for which $f_{\hat{\rho}_{\mathrm{len}(k)}}$ has been computed (Figure 4c), we discover that areas with lower $f_{\hat{\rho}_{\mathrm{len}(k)}}$ values (i.e., low-quality ridges) coincide with regions with high aliasing indication. Figure 7a shows both aggregation fields in the same image for comparison. We determine that we would need a higher resolution to effectively capture the ridges in that area. Finally, the maximum region aggregation field (Figure 9a) is able to separate the configurations which have disconnected areas from the rest and provides a clear picture of the degree of connectedness of the fields in the aggregation space.

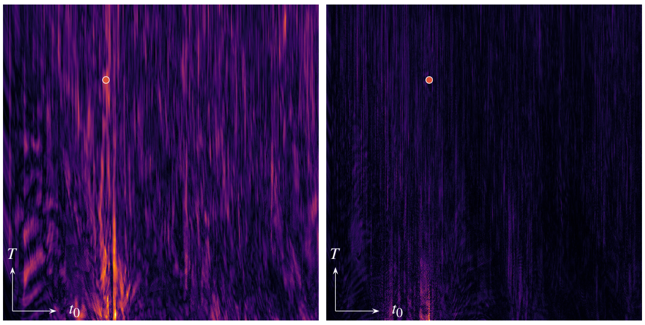
6.2. Buoyant Flow

The Buoyant Flow dataset is a fluid dynamics simulation of air current moving around in a container with a heated bottom wall and a cooled top wall. Figure 12 shows basic and ridge aggregation fields for this dataset. We can see in $f_{\rho_{\mathrm{len}}}$ (Figure 12b) that most



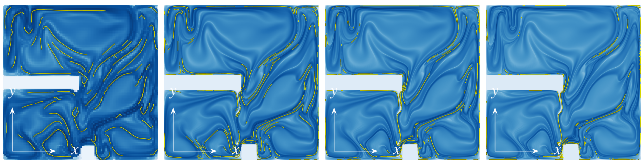
(a) (b) (c)

Figure 12: *Buoyant Flow dataset with $G_{\text{agg}} = 500 \times 500$ and $G = 400 \times 400$, $t_0 \in [0, 0.3]$, and $T \in [0.01, 0.1]$. All fields $f_{\sigma_{\max}}$ (a), $f_{\rho_{\text{len}}}$ (b), and $f_{\rho_{\text{cnt}}}$ (c) show a similar structure with this dataset. The ridge aggregation fields, however, show that the LCS are only sharp enough to be detected as ridges at high T values.*



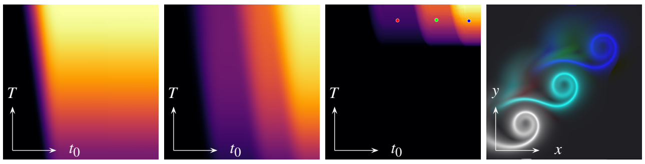
(a) (b)

Figure 13: *Aliasing aggregation fields ($G_{\text{agg}} = 500 \times 500$) for the Buoyant Flow dataset with $G = 100 \times 100$ (a) and $G = 400 \times 400$ (b). Observe how the aliasing values decrease significantly with the increased resolution.*



(a) (b) (c) (d)

Figure 14: *FTLE fields extracted from the Buoyant Flow dataset with $G = 100 \times 100$ (a), $G = 400 \times 400$ (b), $G = 800 \times 800$ (c), and $G = 1600 \times 1600$ (d) with identical $t_0 = 0.09\,s$, $T = 0.088\,s$ (red marker in Figure 13), and $\tau_\lambda = -0.04$ in all cases. Observe how the same τ_λ value leads to very different ridge configurations, depending on the resolution.*



(a) (b) (c) (d)

Figure 15: *Three Gaussian Vortices dataset with $G_{\text{agg}} = G = 400 \times 400$, $t_0 \in [40, 170]$, $T \in [1, 20]$, and $\tau_\lambda = -0.04$ $f_{\sigma_{\max}}$ (a) captures the maximum value in the FTLE field, only showing the boundary where the first vortex appears, while $f_{\sigma_{\text{avg}}}$ (b) captures the average of all values, clearly displaying additional structure as each of the three vortices appears. $f_{\rho_{\text{len}}}$ (c) shows the contribution of each of the three vortices, as they appear, in the form of ridge length. The red, green, and blue markers correspond to the same $T = 18\,s$ and different t_0 of $100\,s$, $133\,s$, and $160\,s$, respectively. (d) shows the three FTLE fields corresponding to the three points in (c), mapped to the red, green, and blue channels. The bottom vortex appears at around $t_0 = 75\,s$ and stays there until the end, thus showing white (red + green + blue). The middle vortex is still present when the top-most one appears, thus showing in cyan (green + blue).*

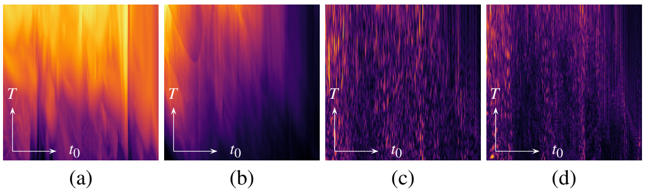
of the structure is concentrated away from low t_0 and T values, but once it shows up, it exhibits little variation. We can infer that the Lagrangian coherent structures, represented by the ridges, move around with the field but the total number stays more or less constant, which indicates their high quality. Figure 13 contains a comparative analysis of the aliasing aggregation field with two different resolutions, 100×100 (Figure 13a) and 400×400 (Figure 13b). Observe how the aliasing problems diminish considerably with this factor of four increase in field resolution. We can be quite confident that the field is sampled correctly across the whole aggregation space, except for a very small problematic zone close to the bottom left. Figure 14 shows the Buoyant Flow FTLE with varying resolutions and identical τ_λ. Our system, as demonstrated in the accompanying video, supports interactive exploration of such parameters.

6.3. Three Gaussian Vortices

The Three Gaussian Vortices dataset is a constructed vector field based on a blend of simpler fields generated using a mixture of programmatic rules and time-dependent analytical functions. It consists of three vortices, each spatially masked with a Gaussian function, that appear at different times and positions on top of a uniform velocity field to the right. The basic aggregation fields $f_{\sigma_{\max}}$ (Figure 15a) and $f_{\sigma_{\text{avg}}}$ (Figure 15b) capture the general trend quite well. $f_{\sigma_{\text{avg}}}$ captures the moment when each vortex appears. Figure 15c shows that $f_{\rho_{\text{len}}}$ additionally captures the location of the vortices in the aggregation space in terms of ridges, providing a very good picture of interesting parameter combinations for this dataset. To that effect, the illuminated areas represent the subspaces of (t_0, T) where one, two, and three vortices are in the field and cause sufficient sharpness in the FTLE field so that the ridge extraction algorithm succeeds.

6.4. 5-Body

We now demonstrate the utility of our approach to handle fields other than traditional FTLE. In this example, we use an inertial n-body system with five bodies. The simulated bodies start more or less evenly distributed in space and interact freely over time (Figure 17). We use the phase-space finite-time Lyapunov exponent field (PS-FTLE) [SJJ*17], which extends the regular FTLE to the dynamics of inertial systems, with initial velocity constrained to $\mathbf{0}$ (PS-FTLE-P). The aggregation fields (Figure 16) exhibit more structure than most previous datasets. $f_{\sigma_{\max}}$ and $f_{\rho_{\text{len}}}$ clearly show the dispersion of the bodies toward higher t_0, i.e., the bodies exit



(a) (b) (c) (d)

Figure 16: *Aggregation fields for the 5-Body dataset. $f_{\sigma_{\max}}$ (a), $f_{\rho_{\text{len}}}$ (b) with $G = 300 \times 300$, and aliasing aggregation fields with $G = 100$ (c) and $G = 300 \times 300$ (d). $G_{\text{agg}} = 300 \times 300$, $t_0 \in [0, 6.5]$, and $T \in [1, 3]$ in all cases. The aliasing aggregation fields suggest that the chosen resolution is too low for this dataset, irrespective of (reasonable) advection times.*

the region of study toward the end of the time range. Figure 17 provides a PS-FTLE-P sequence for this dataset. The aliasing aggregation fields for $G = 100 \times 100$ (Figure 16c) and $G = 300 \times 300$ (Figure 16d) show that, even though the aliasing decreases considerably in some regions, a resolution of 300 is still insufficient. The PS-FTLE-P folds over and over close to the bodies, which the shown resolution is not able to capture appropriately. The region aggregation fields for this dataset (Figure 18) hint at configurations at which the ridges isolate disconnected regions (yellow zones in Figure 18b), and configurations where the ridges do not quite close at the given resolution and thus produce lower connectedness values.

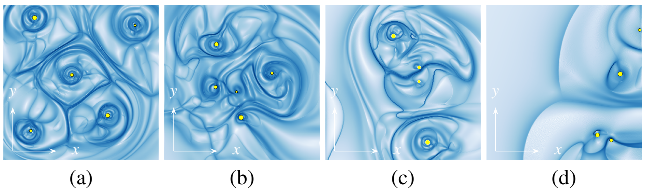
Stefan Jordan, one of our coauthors and an expert astrophysicist, has evaluated the tool with the 5-Body dataset: The system is able to guide to some interesting events like close-body interactions around $t_0 = 1.6$, showing up as a valley in the maximum and ridge length aggregation fields. Even though, to the best of our knowledge, FTLE has never been used in astronomy, the identification of relevant events and features with such a tool could aid in the interpretation and understanding of theoretical astronomical data like n-body and hydrodynamic simulations, the merging of galaxies, the dissolution of star clusters, star and planetary system formation and, convection in stars.

6.5. Atmospheric Wind

This dataset was generated by the Copernicus Climate Change Service (2020) and contains observations of wind velocity in the area of longitudes between 30° and 60° east and latitudes between −15° and 15°, corresponding to the region around the horn of Africa, for December 1 to 3, 2019. This is a rather complex dataset, but it is easy to identify the day–night cycles by investigating the aggregation fields in Figure 19. The maximum aggregation field Figure 19a proves to be the poorest at capturing it, but it shows some interesting additional structure. However, both $f_{\sigma_{\text{avg}}}$ (Figure 19b) and $f_{\hat{\rho}_{\text{len}(k)}}$ (Figure 19c) show that periodic trend, with the latter revealing the nights as times of higher activity with more pronounced ridges.

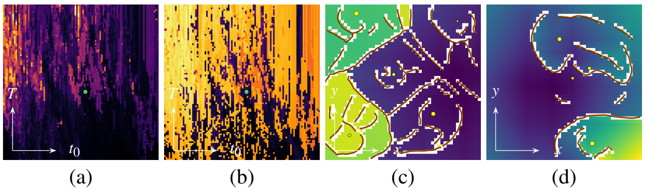
7. Implementation and Performance

The reference application is implemented using Python and the Qt5 GUI library via PyQt5. Much of the computationally intensive operations ((PS-)FTLE fields, trajectories, connectedness, ridges, reverse integration, etc.) are implemented in highly parallel CUDA kernels and managed via PyCuda [KPC*13]. These are laid out and structured using a tailor-made templating system that is able to construct and compile the needed kernel on-demand from a large set of templates. The templating system produces 67 different single kernels, amounting to 20 thousand lines of CUDA code in total. The graphical representation in the panels is handled by the OpenGL abstraction layer, provided by the library VisPy [Vis], and extra care was put into implementing efficient and speedy updating and streaming solutions for all graphical components. The project has 11540 lines of code in total, 8421 of which are in Python, 2580 in CUDA, 179 in Bash scripts, 220 in GLSL, and 140 in HTML.



(a) (b) (c) (d)

Figure 17: *PS-FTLE-P fields for the 5-Body dataset with $G = 300 \times 300$, $T = 2.8\,s$, and $t_0 = 0.25\,s$ (a), $t_0 = 2.25\,s$ (b), $t_0 = 4.25\,s$ (c), and $t_0 = 6.26\,s$ (d). The radius of the bodies (yellow dots) is proportional to the mass. The five bodies start evenly spread, with different initial velocities, then interact with each other and disappear to the right.*



(a) (b) (c) (d)

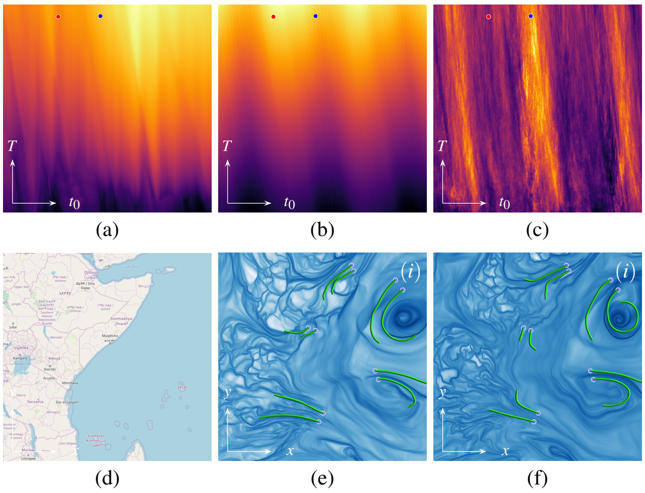
Figure 18: *Average (a) and maximum (b) region aggregation fields for the 5-Body dataset with $G_{\text{agg}} = 100 \times 100$ and $G = 50 \times 50$, together with the actual connectedness fields at $t_0 = 0.057\,s$, $T = 2.68\,s$ (c), and $t_0 = 3.44\,s$, $T = 1.87\,s$ (d), corresponding to the red and green dots in the aggregation fields, respectively.*

7.1. Performance Analysis

The computational complexity of this project resides mainly in the computation of the field matrix in the aggregation space. The

Table 1: *Interactivity performance results with the Quad-Gyre dataset in average frame time, and sample time for different G values, and fixed $\Delta t = 0.001\,s$ and $T = 4.0\,s$. The 'Steps' column refers to the integration steps per seed (i.e., $T/\Delta t$). Timings in seconds.*

G	Steps	Samples	Frame time	Sample time
50×50	4 000	2 500	0.038	$0.152 \cdot 10^{-4}$
100×100		10 000	0.108	$0.108 \cdot 10^{-4}$
200×200		40 000	0.364	$0.091 \cdot 10^{-4}$
400×400		160 000	1.281	$0.080 \cdot 10^{-4}$
800×800		640 000	4.988	$0.078 \cdot 10^{-4}$



(a) (b) (c)

(d) (e) (f)

Figure 19: *The Atmospheric Wind dataset represents wind data of a region close to the eastern coast of Africa (d). We apply our system with $G_{\text{agg}} = 300 \times 300$ and $G = 400 \times 400$, $t_0 \in [0,60]$, $T \in [1,10]$, and $\tau_\lambda = -0.08$ $f_{\sigma_{\max}}$ (a) shows maximum FTLE values increasing with t_0 and T, with some additional substructure caused by the ever-changing wind patterns, while $f_{\sigma_{\text{avg}}}$ (b) reveals periodic structure. $f_{\hat{\rho}_{\text{len}(k)}}$ (c) exhibits three vertical regions with high quality ridges. The FTLE fields are captured at the same $T = 9.5$ h and different $t_0 = 15.8$ h (e), and $t_0$0 28 h (f), corresponding to the red and blue markers in the aggregation fields, respectively. We seed pairs of trajectories at close locations to aid exploration. For instance, the ridge (i), not present in (e), appears in (f) (see the lower seed spiraling into the vortex causing a larger separation). Other trajectories also reveal greater separation in (f). Interactive exploration reveals that this trend is maintained along the bright bands in (c).*

grid resolution of both the aggregation field (G_{agg}) and the FTLE fields (G), plus the current advection duration T_i and integration step Δt_i, define the number of operations to compute to obtain the flow map as $N_{t_0} \cdot N_T \cdot N \cdot M \cdot (T_i/\Delta t_i)$. The interactivity performance analysis (Table 1) shows that the FTLE field frame time is proportional to the number of samples, but the sample time actually decreases slightly as the resolution increases. This may hint at a better GPU occupancy rate with higher resolutions, thus leading to lower sample times.

Table 2: *Precomputation performance results with the Quad-Gyre dataset in total compute time for different aggregation field resolutions, and using only the basic and aliasing aggregation functions (B), additionally the ridge aggregation functions (R), and additionally the region aggregation functions (D). The fixed parameters for this analysis are $G = 50 \times 50$, $G_s = 10 \times 10$, $t_0 \in [0,8]$, and $T \in [1,8]$. All timings are in seconds.*

G_{agg}	B	B+R	B+R+D
5 × 5	0.5	0.7 (+47%)	348.6 (+45651%)
50 × 50	11.9	40.1 (+236%)	29 990.2 (+74508%)
100 × 100	39.6	149.5 (+276%)	135 627.0 (+90610%)
200 × 200	142.3	590.3 (+314%)	480 310.5 (+81261%)

The precomputation performance analysis (Table 2) measures the computation of the aggregation fields. We run different aggregation field resolutions of the same dataset with three different cumulative aggregation function sets: the basic and aliasing aggregation functions (B), the ridge aggregation functions (R), and the region aggregation functions (D). Each set contains the previous, as the ridge extraction is a prerequisite for computing the connectedness, and the basic aggregation fields are always computed. The ridge extraction and connectedness operations are done per field, without batching (see Section 7.2), and that reflects in their cost, with increments of a few hundreds percent for ridge extraction, and tens of thousands percent for region aggregation.

7.2. Optimization

Optimizations have been implemented in order to maximize parallelism on the GPU, improve core occupancy, and minimize CPU synchronization points. In order to do so, we batch the T dimension of the aggregation space into the main kernels, so that each computes N_T FTLE fields. We found that batching T is the most beneficial, because we can significantly shorten the computation time of the flow map by choosing an integration step Δt, N_T, and T extent $[T_0, T_1]$, so that $(T_1 - T_0)/N_T$ is divisible by Δt. This allows us to compute the flow map only once for the highest T setting (i.e., the top of the aggregation field) and sample it for the rest.

Other, more straightforward, optimization strategies include full GPU implementations of A* and ridge extraction. The resolution at which the connectedness fields can be computed is limited by per-thread memory constraints. We found that connectedness fields more than $G_s = 50 \times 50$ could not be computed on the GPU with the available hardware (Nvidia GTX 970 and 1070).

8. Conclusion

In this paper, we presented a novel approach to the visual analysis of finite-time Lyapunov exponent based flow visualization, and we have demonstrated its usefulness with a variety of simulated and analytical datasets. We have introduced a set of aggregation functions that are able to capture different aspects of the underlying fields. We have found that the basic aggregation captures general trends very well, while the aliasing aggregation helps determine whether the discretization is sufficient, especially when the LCS are sharp. The ridge aggregation is able to identify areas with high-quality ridges and the region aggregation helps assess the topological "connectedness" of the dataset under study.

We leave as future work the determination of an approximate optimal resolution by means of heuristics that use information from the aliasing and ridge aggregation functions. Also, we plan to add more aggregation functions and extend the current ones, possibly with extra dimensions. A three-dimensional ridge aggregation field with τ_λ as an additional dimension could prove useful at analyzing the ridge extraction threshold subspace. Another obvious candidate for an additional dimension is the FTLE resolution G, but that would imply deep changes in the current precomputation engine. We also plan to support 3D datasets, since all aggregation functions should be directly applicable in 3D.

References

[ABM*07] Aigner W., Bertone A., Miksch S., Tominski C., Schumann H.: Towards a conceptual framework for visual analytics of time and time-oriented data. In *Proceedings of 2007 Winter Simulation Conference* (2007), pp. 721–729. 2

[Asi93] Asimov D.: Notes on the topology of vector fields and flows. *Technical report, NASA Ames Research Center. RNR-93-003* (1993). 2

[BCC*05] Bavoil L., Callahan S. P., Crossno P. J., Freire J., Scheidegger C. E., Silva C. T., Vo H. T.: VisTrails: enabling interactive multiple-view visualizations. In *Proceedings of IEEE Visualization 2005.* (2005), pp. 135–142. 2

[BMI*] Bürger R., Muigg P., Ilcík M., Doleisch H., Hauser H.: Integrating local feature detectors in the interactive visual analysis of flow simulation data. In *Proceedings of Eurographics / IEEE VGTC Conference on Visualization 2007*, pp. 171–178. 2

[CT65] Cooley J., Tukey J.: An algorithm for the machine computation of complex Fourier series. *Mathematics of Computation 19* (1965), 297–301. 5

[DGH] Doleisch H., Gasser M., Hauser H.: Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of the Symposium on Data Visualisation 2003*, pp. 239–248. 2

[DMG*] Doleisch H., Mayer M., Gasser M., Wanker R., Hauser H.: Case study: Visual analysis of complex, time-dependent simulation results of a diesel exhaust system. In *Proceedings of Eurographics / IEEE VGTC Symposium on Visualization 2004*, pp. 91–96. 2

[Ebe96] Eberly D.: *Ridges in Image and Data Analysis. Computational Imaging and Vision.*, vol. 7. Kluwer Academic Publishers, 1996. 2, 3

[FP01] Furst J., Pizer S.: Marching ridges. In *Proceedings of 2001 IASTED International Conference on Signal and Image Processing* (2001), pp. 22–26. 3

[GGTH07] Garth C., Gerhardt F., Tricoche X., Hagen H.: Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1464–71. 2

[Hal01] Haller G.: Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D 149* (2001), 248–277. 2

[Hal04] Haller G.: Exact theory of unsteady separation for two-dimensional flows. *Journal of Fluid Mechanics 512* (2004), 257–311. 2

[HH89] Helman J., Hesselink L.: Representation and display of vector field topology in fluid flow data sets. *Computer 22* (1989), 27–36. 2

[HH91] Helman J., Hesselink L.: Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications 11* (1991), 36–46. 2

[HLH*16] Heine C., Leitte H., Hlawitschka M., Iuricich F., De Floriani L., Scheuermann G., Hagen H., Garth C.: A survey of topology-based methods in visualization. *Computer Graphics Forum 35*, 3 (2016), 643–667. 2

[KCH*18] Kindlmann G., Chiw C., Huynh T., Gyulassy A., Reppy J., Bremer P.-T.: Rendering and extracting extremal features in 3D fields. *Computer Graphics Forum 37*, 3 (2018), 525–536. 3

[KCS*16] Kindlmann G., Chiw C., Seltzer N., Samuels L., Reppy J.: Diderot: a domain-specific language for portable parallel scientific visualization and image analysis. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2016), 867–876. 3

[KPC*13] Klöckner A., Pinto N., Catanzaro B., Lee Y., Ivanov P., Fasih A.: PyCUDA and PyOpenCL: A scripting-based approach to GPU run-time code generation. *Parallel Computing 38*, 3 (2013), 157–174. 10

[LC87] Lorensen W., Cline H.: Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Computer Graphics 21*, 4 (1987), 163–169. 3

[PPF*11] Pobitzer A., Peikert R., Fuchs R., Schindler B., Kuhn A., Theisel H., Krešimir M., Hauser H.: The state of the art in topology-based visualization of unsteady flow. *Computer Graphics Forum 30*, 6 (2011), 1789–1811. 2

[PS] Peikert R., Sadlo F.: Height ridge computation and filtering for visualization. In *Proceedings of IEEE VGTC Pacific Visualization Symposium 2008*, pp. 119–126. 2

[SJJ*17] Sagristà A., Jordan S., Just A., Dias F., Nonato L. G., Sadlo F.: Topological analysis of inertial dynamics. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 950–959. 2, 9

[SLM05] Shadden S., Lekien F., Marsden J.: Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D: Nonlinear Phenomena 212*, 3–4 (2005), 271–304. 2, 8

[Sma67] Smale S.: Differntiable dynamical systems. *Bulletin of the American Mathematical Society 73* (1967), 747–817. 5

[SP07] Sadlo F., Peikert R.: Efficient visualization of Lagrangian coherent structures by filtered AMR ridge extraction. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1456–1463. 2, 3

[SRBE] Schulz M., Reck F., Bartelheimer W., Ertl T.: Interactive visualization of fluid dynamics simulations in locally refined Cartesian grids. In *Proceedings IEEE Visualization 1999*, pp. 413–553. 2

[SRP11] Sadlo F., Rigazzi A., Peikert R.: Time-dependent visualization of Lagrangian coherent structures by grid advection. In *Topology-Based Methods in Visualization II*. Springer Berlin Heidelberg, 2011, pp. 151–165. 2

[STH*07] Shi K., Theisel H., Hauser H., Weinkauf T., Matkovic K., Hege H. C., Seidel H.: Path line attributes - an information visualization approach to analyzing the dynamic behavior of 3D time-dependent flow fields. In *Topology-Based Methods in Visualization II*. 2007, pp. 75–88. 2

[TFL*18] Tierny J., Favelier G., Levine J. A., Gueunet C., Michaux M.: The topology toolkit. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 832–842. 2

[Vis] VisPy Project Team: Vispy. `http://vispy.org`. [Online; accessed 06-April-2020]. 10

[Wei07] Weiskopf D.: *GPU Based Interactive Visualization Techniques*. Springer Verlag, 2007. 2