A Variance Based FTLE-like Method for Unsteady Uncertain Vector Fields

Dominic Schneider, Jan Fuhrmann, Wieland Reich, and Gerik Scheuermann

Abstract We present a novel approach to obtain FTLE-like structures and visualizations for uncertain vector fields. For this, we compute the flow map for a certain time interval now mapping to particle distributions rather than single positions. Our method comprises a variance-based analysis to measure the maximal stretching of the distributions thereby obtaining FTLE-like structures for uncertain vector fields. In the case of ordinary vector fields, the results are visually almost identical to FTLE. We analyze our method in the presence of errors by applying it to a number of synthetic and real world datasets.

1 Introduction

Uncertainty in vector field data poses a major challenge for visualization in general [18] but especially for the identification of coherent structures. For deterministic steady data *vector field topology* (VFT) reveals the overall structure in a condensed abstract view. However, VFT as such, is directly applicable only to steady or quasi stationary vector fields. This is due to the fact, that the theoretical foundation of VFT is build on stream lines. A concept build around path lines rather than streamline is the *finite-time Lyapunov exponent* (FTLE) which has its roots in dynamical systems theory and was introduced by Haller [15]. *Lagrangian coherent structures*

Dominic Schneider
University of Leipzig, e-mail: schneider@informatik.uni-leipzig.de

Jan Fuhrmann
University of Heidelberg, Institute for Applied Mathematics e-mail: jan.fuhrmann@uni-hd.de

Wieland Reich
University of Leipzig, e-mail: reich@informatik.uni-leipzig.de

Gerik Scheuermann
University of Leipzig, e-mail: scheuermann@informatik.uni-leipzig.de

(LCS), which can be extracted as ridge lines in the FTLE field, act as material lines or surfaces in the flow [2]. Hence they are either attracting if nearby particles converge towards them or repelling if they diverge from the respective LCS. They are the time-dependent analog to stable and unstable manifolds in steady vector fields. FTLE is one of the most important techniques to analyze flow structures in vector fields, however, it can not be applied to uncertain vector fields directly. Haller [16] provided a formal analysis of FTLE behavior in the presence of error and found that Lagrangian coherent structures are very robust in the presence of these errors. In contrast, we propose an alternative, practical, variance-based approach directly applicable to stochastic flow maps computed from uncertain vector fields. We compute a FTLE-like measure called FTVA (*finite time variance analysis*) yielding structures similar to FTLE for uncertain vector fields. In terms of visualization and in the case of *ordinary vector fields*, FTVA reproduces the same structures as FTLE (see Sec. 5). More precisely, FTVA does not give the same results as FTLE numerically but qualitatively. In this sense FTVA represents a generalization of FTLE to uncertain vector fields. We examine our new technique in the presence of errors and find that the resulting structures are very robust.

2 Related Work

In this section we give an overview of relevant work on vector field topology, FTLE and work in the context of visualizing uncertainty in various types of data.

Vector field topology (VFT) has been introduced to visualization by Helman and Hesselink [17] and Globus et al. [12]. It aims at extracting the so-called invariant sets. An invariant set is a special set of streamlines, most importantly isolated zeros (critical points) which are degenerate stream lines. Löffelmann et al. [23] visualized periodic orbits and Wischgoll et al. [38] and Chen et al. [6] presented an algorithm locating them in the presence of data uncertainty. Invariant sets can segment the vector field in regions of similar flow, hence they are termed *separatrices*. Displaying all separatrices, however, would lead to occlusion problems. One solution to this problem is the display of their intersection curves, the so-called saddle connectors [36].

For the topological analysis of time-dependent vector fields Sadlo et al. [32] used FTLE and generalized vector field topology, where degenerate streak lines take on the role of critical points. Weinkauf et al. [37] integrate a streak line field to facilitate time-dependent topology extraction. In contrast to the integration based perspective, Fuchs et al. [8] provide a differentiation based perspective to find critical points for time-dependent vector fields.

In visualization LCS have been increasingly subject of interest in the last decade [28]. Garth et al. visualized the FTLE field for 2D flows [10] using color and height maps and for 3D flows [9] using direct volume rendering. In subsequent work [11] FTLE has been used to identify attachment and separation on the surface of obstacles. Sadlo et al. compared VFT and LCS visualizations [35] and proposed a scheme for an accelerated computation [31].

Griethe at al. [13], Johnson et al. [19] and Pang et al. [27] give an overview of different uncertainty concepts and the different techniques used in visualization.

Uncertainty in vector fields has been visualized using additional geometry such as glyphs to represent the uncertainty at certain positions [39, 22, 21, 40, 24] to convey the amount of uncertainty explicitly. Other researches used the concept of fuzziness or blurring to convey uncertainty in volumetric data [7, 30] or isosurfaces [14]. Lundstrom et al. [25] and Brown [5] used animation of the different possibilities to convey uncertainty information. Sanderson et al. [33] utilized reaction-diffusion systems to visualize vector fields and show that it is possible to incorporate uncertainty into their model. Botchen et al. [3, 4] proposed texture based flow visualization techniques and convey uncertainty by blurring streak lines using cross advection and diffusion.

Otto et al. [26] considered global uncertainty by the transport of local uncertainty in steady 2D flow fields. The domain is super-sampled by a high number of particles. Each particle is integrated and the final distribution is interpreted as a discrete particle density function. In contrast, we consider the original grid and analyze the transport of uncertainty by computing a stochastic flow map for unsteady flows. The stochastic flow map is analyzed using a Principal Component Analysis yielding a scalar field showing FTLE-like structures for uncertain vector fields.

3 Uncertain Vector Fields

In this work we consider steady and unsteady 2D vector fields. In the following these are called *ordinary vector fields*. In contrast, *uncertain vector fields* no longer map a position to a single unique vector but rather to a probability distribution of vectors. We adopt the definition for a steady uncertain 2D vector field given in [26]:

We follow the approach of [16] and examine a stochastic differential equation describing the vector field and a chosen error model. Since we have full control of the amount and type of error, this provides us with a method to analyze the vector field in the presence of uncertainty, i.e. error. The stochastic differential equation is solved by stochastic integration which is described in the next section.

3.1 Stochastic Integration

In the following we develop a model for stochastic integration in a vector field $\mathbf{v}$ defined over a domain D. We start with solving the following classical ordinary differential equation (ODE):

$$d\phi \quad = \mathbf{v}(\phi(t),t)dt \tag{1}$$

$$\phi(t_0) = x_0 \tag{2}$$

where ϕ is a map $D \to D$ Since we want to analyze vector fields in the presence of errors, it seems reasonable to modify equation 1 to include random effects disturbing the system, thus turning it into a *stochastic differential equation* (SDE):

$$\begin{aligned} d\Phi &= \mathbf{v}(\Phi(t),t)dt + B(\Phi(t))d\xi_t \qquad (3) \\ \Phi(t_0) &= x_0 \qquad (4) \end{aligned}$$

The first term of the right hand side resembles the classical formulation (see Eq. 1) and the second term represents the disturbance with $d\xi_t$ being a so-called *continuous-time stochastic process*, where ξ_t is indexed by real numbers $t \geq 0$ and $B(.)$ characterizing the disturbance.

The most popular example of a stochastic process that is ubiquitous in physics, chemistry, finance and mathematics is the *Wiener process* W_t named after Norbert Wiener with the following three properties (see [34]):

Property 1. For each t, the random variable W_t is normally distributed with mean 0 and variance t.

Property 2. For each $t_1 \leq t_2$, the normal random variable $W_{t_2} - W_{t_1}$ is independent of the random variable W_{t_1}, and in fact independent of all W_t, $0 \leq t \leq t_1$.

Property 3. The Wiener process W_t can be represented by continuous paths.

In order to simplify the matter we set $B(\Phi(t)) = \varepsilon$ constant. This leads to the following stochastic differential equation:

$$\begin{aligned} d\Phi &= \mathbf{v}(\Phi(t),t)dt + \varepsilon dW_t \qquad (5) \\ \Phi(t_0) &= x_0 \qquad (6) \end{aligned}$$

Now we can solve equation 5 in the following way:

$$\Phi(t) = x_0 + \int_0^t \mathbf{v}(\Phi(\tau),\tau)d\tau + \int_0^t \varepsilon dW_\tau \tag{7}$$

This resembles a so-called *drift and diffusion* model where $\mathbf{v}$ is referred to as the drift coefficient, while ε is called the diffusion coefficient. A helpful interpretation of the stochastic integral in equation 7 is that in a time interval of length 1 the stochastic process changes its value by an amount that is normally distributed with expectation $\mathbf{v}$ and variance ε. This change is independent of the processes past behavior because the increments of the Wiener process are independent and normally distributed (see property 1–3). The integral of the Wiener process, in particular, yields a diffusion term and as necessary a contribution to the drift term. For an in depth discussion of stochastic differential equations and integrals we refer the reader to [29].

3.2 Error Model

Despite the popularity and importance of the Wiener process in other fields of research we will use a different random process. We deem this necessary due to the fact that the Wiener process has normally (Gaussian) distributed increments. This implies that the increments are unbounded meaning an arbitrary large error could occur. This means the numerical stochastic integration could perform steps of arbitrary length. This does not account for the situation arising with CFD data, because it would mean that the simulation contains arbitrary large errors. Instead an error bound is provided describing the maximal error of the calculation. Moreover we do not want to make any assumption about the error distribution within this bound which leads us to an equal distribution with zero mean, bounded by a n-dimensional ball with radius ε (a circle in 2D). Furthermore, we assume independence for this stochastic process (see property 2 of the Wiener process). On the contrary, the error in a CFD simulation at one grid point is likely not to be independent from neighboring grid points. However, the modeling of this dependence would require exact knowledge of the underlying CFD solver and it would be a highly complex task and a research paper in its own right. Hence, the presented error model seems to be the best one to be assumed.

Furthermore, we assume the same error bound ε for the whole domain. This has the effect of a homogeneous error distribution and the error beeing independent of the location in space and time. Formally this is achieved by setting $B(\Phi(t)) = \varepsilon$ constant (like in Sec. 3.1) which leads to the following SDE:

$$d\Phi = \mathbf{v}(\Phi(t),t)dt + \varepsilon dR_t \tag{8}$$

$$\Phi(t_0) = x_0 \tag{9}$$

with dR_t denoting the random process complying with the error model described above.

If the error distribution, however, turns out to be inhomogeneous over time, ε is no longer a constant but will depend on the location in space and time $\varepsilon(\Phi(t),t)$. Moreover, the error ε does not need to be a scalar quantity but can provide directional information, hence becoming biased or anisotropic. This further shows the similarity between error and diffusion. In these cases the underlying SDE needs to be modified to fit the chosen error model resulting in a different numerical approximation (see Sec. 3.3). In general the error model can of course be tuned to model any kind of distribution including a Gaussian one.

3.3 Numerical Approximation

The classical numerical approximation schemes (e.g. Euler or Runge-Kutta) cannot be applied to a stochastic differential equation as such, but needs to be modified. In order to approximate the stochastic integration process numerically we need to discretize the stochastic process dR_t in equation 8. This is accomplished by the Euler-Maruyama method (see [20]) which provides an approximate numerical solution of a stochastic differential equation. Application to the stochastic process dR_t yields:

$$\Delta R_n = \varepsilon\ \Delta t\ RW \tag{10}$$

where RW denotes an increment of a so-called *random walk*. A random walk is the mathematical formalization of a trajectory consisting of successive random steps discretizing the considered stochastic process. Our error model (see Sec. 3.2) requires RW to be an undirected random walk with a bounded symmetric uniform probability distribution with bound ε.

The discretization of 8 then reads as follows:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t\ \mathbf{v}(\mathbf{y}_n, t) + \Delta t\ \varepsilon\ RW \tag{11}$$

As one can see, the classical Euler method is a special case of the Euler-Maruyama integration for $\varepsilon = 0$. The successive application of equation 11 yields a stochastic stream- or path-line respectively. However, the obtained trajectory is an approximate realization of the solution stochastic process Φ in Equation 8 and each one will be different.

In the case that the vector field consists of measured data and a stochastic modeling of the process is not possible then the uncertain vector field $\mathbf{v}_u$, consisting of the measured data, can of course be evaluated directly yielding:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t\ \mathbf{v}_u(\mathbf{y}_n, t) \tag{12}$$

where $\mathbf{v}_u$ is automatically respecting the probability distribution.

3.4 Stochastic Flow Maps

A classical discrete flow map $\phi_{t0}^{t_0+t}(\mathbf{x})$ maps from a sample position $\mathbf{x} \in D$ to the position of a particle started at $\mathbf{x}$ at time t_0 advected by the flow for the time t. In other words $\phi_{t0}^{t_0+t}(\mathbf{x})$ maps $\mathbf{x}$ to its advected position. In case the integration reaches the boundary we store the position on the boundary in the flow map. Classical flow maps are computed by integrating one particle per sample position. In contrast, *stochastic flow maps* Φ store a whole distribution per sample position, yet the principle is the same. We approximate the distributions in the stochastic flow map by sampling: we start multiple stochastic integrations at each grid point $\mathbf{x}$. The number of integrations per position is prescribed by the parameter N.

In Sec. 4 we evaluate the stochastic flow map at a certain grid point r and at the neighboring grid points for our calculations. In the following we call these grid points the stencil of the grid point r. Since we are interested in the endpoints of trajectories for the grid points of a stencil they are part of one random experiment.

Therefore, N of these random experiments would have to be run for every grid point in the mesh.

Assigning two or more particles to the same random experiment means they are dependent in a certain way. This dependency manifests in the way that if two particles meet at exactly the same point in space and time they experience the same fluctuation, i.e. RW in Eq. 8 is for both particles the same. However, if one of these two particles is only a small distance off that position the random variables are independent, i.e. RW evaluates for each particle to a different value.

An algorithm taking this into account would create N ordinary vector fields V_d, each disturbed according to the error model. For each of these vector fields particle integration is carried out deterministically. This way we obtain a distribution consisting of advected particles for each start point as well, at the expense of additional memory consumption for the creation of an additional vector field and the computational cost for its construction.

Now we argue that the probability of any two particles which are part of the same random experiment seeded at different positions in space meet at exactly the same position in space and time is zero in the continous setting. Since we are in a discrete setting the probability of this event is not zero anymore. However, if this still very unlikely event is happening we argue that in a continous setting both particles would not have met at exactly the same position in space and time. Hence we argue further that this event is due to the discretisation of the vector field and especially due to the discrete approximation of the integration process. The conclusion of this argumentation is that we calculate trajectories for particles of the same random experiment simply as they were independent random experiments since the possible gain is neglectible. After all we are only interested in the expected value and the variance of the particle distribution. However, this argumentation is only valid for an error model assuming independence of errors.

4 FTLE-like Variance Based Analysis of Stochastic Flow Maps

The classical FTLE method [15] measures the maximal separation or expansion rate of two closely seeded particles when advected by the flow for a finite time t. FTLE can be computed by utilizing the above mentioned flow map. More precisely it is the spectral norm of the (right) Cauchy-Green deformation tensor $\Delta(\mathbf{x})$ [1]:

$$\Delta(\mathbf{x},t,t_0) = (\nabla\phi_{t0}^{t_0+t}(\mathbf{x}))^T(\nabla\phi_{t0}^{t_0+t}(\mathbf{x})) \tag{13}$$

The function Δ is a symmetric matrix and measures the square of the distance change due to deformation. Now FTLE is defined as the logarithm of the Cauchy-Green deformation tensor's maximum eigenvalue λ_{max} normalized by the absolute advection time t. More formally this reads as follows:

$$FTLE_{t_0}^{t}(\mathbf{x}) = \frac{1}{|t|}\ln\sqrt{\lambda_{max}(\Delta(\mathbf{x},t,t_0))} \tag{14}$$

For an in depth discussion about FTLE the reader is referred to the work of Haller [15].

The above description provides us with an algorithm for the computation of FTLE in ordinary vector fields. However, it cannot be directly applied to uncertain vector fields since we are dealing with probability distributions. On the other hand, the idea of FTLE is to find the maximal stretching a virtual particle experiences during its lifetime. We think this idea can be carried over to the realm of uncertain vector fields by replacing the stretching with variance. Therefore, we propose a new geometry driven technique to compute a FTLE-like field for uncertain vector fields based on variance. The FTLE principle then translates to finding the maximal variance of the advected distribution. The best linear approximation for this problem is provided by the *principal component analysis* (PCA). PCA is hence used to measure the deformation of the seeding points and the according probability distributions comprising of the advected particles. Therefore, we evaluate the stochastic flow map at the current and neighboring positions wrt. the mesh. This is necessary mainly for two reasons: First, we need a reference value for the variance to measure the stretching. Moreover this simulates the discrete derivation process in the FTLE computation, since the direct neighbors are involved in the numerical approximation of the derivative. Second, if we would consider only the current position and the error ε is rather small we would obtain rather an analysis of the error (diffusion) process than the stretching caused by the underlying vector field. This is due to the discrete nature of the vector field.

The distributions gained by evaluating the stochastic flow map are approximated by stochastic trajectory endpoints. All these endpoints are merged to one single set or distribution respectively. From this set we compute the covariance matrix C which is a symmetric matrix like the Cauchy-Green deformation tensor. The matrix C is a linear model measuring the square of the standard deviation (variance) in every direction of space. We are interested in the maximal variance of C which is represented by the maximal eigenvalue. Fig. 1 summarizes the algorithm visually.

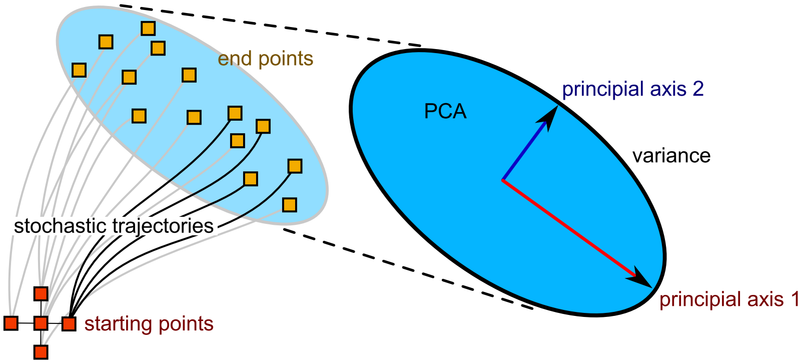

Fig. 1 Stochastic integration from a starting point gives a distribution of end points due to uncertainty. A principal component analysis of the start and end point distribution provides information about the maximum amount of stretching.

The relative maximal standard deviation consists of the maximal standard deviation of the matrix C divided by the maximal standard deviation of the seed points. If we interpret the relative maximal standard deviation as a measure for the maximal stretching of a distribution after integration in the uncertain vector field and recall that variance is the squared standard deviation and account for the similarities between the Cauchy-Green deformation tensor and the covariance matrix C we can simply rewrite equation 14 by changing Δ and C:

$$FTVA_{t_0}^{t}(\mathbf{x}) = \frac{1}{|t|} \ln \sqrt{\lambda_{max}(C(\mathbf{x},t,t_0))} \tag{15}$$

where FTVA stands for *finite time variance analysis*.

5 Results

In this section we apply our method to a 2D steady and unsteady vector field to demonstrate the utility and robustness of our method. The quality of the results naturally depends on certain parameters, which are integration time, the amount of error and the number of particles per position. The fist one is relevant for FTLE as well, whereas the last one is an additional parameter for the stochastic integration. The amount of error is only a parameter for our numerical experiment because it is usually determined by the application.

5.1 Tilted Bar

Our first dataset is a 2D unsteady vector field comprising of 100 time steps each with 79200 positions and 78421 quad cells. It consists of a Karman vortex street behind a tilted bar. A direct visual comparison between FTLE and FTVA in the steady case of a selected time step with integration time 1 is depicted in Fig. 2 and shows only slight visible differences. We used the color mapping proposed in [10], where red structures indicate high divergence in positive time, blue high divergence in negative time, black high divergence in both time directions and white no divergence.

In order to show the robustness of our method and the computed LCS, we increased the error in the computations successively. Figure 3 shows LCS for integration time 0.4 but with different amounts of error. An expected result is that the relative strength of LCS weakens as the error grows (see Fig. 3 bottom). However, despite the weakening, the LCS are very robust. As can be seen the LCS remain visible for a surprisingly large error which, according to our model, grows linearly with the integration time.

Another parameter to be studied is the amount of particles per position which is a crucial one in terms of quality. For the visualizations in figure 4 the number of

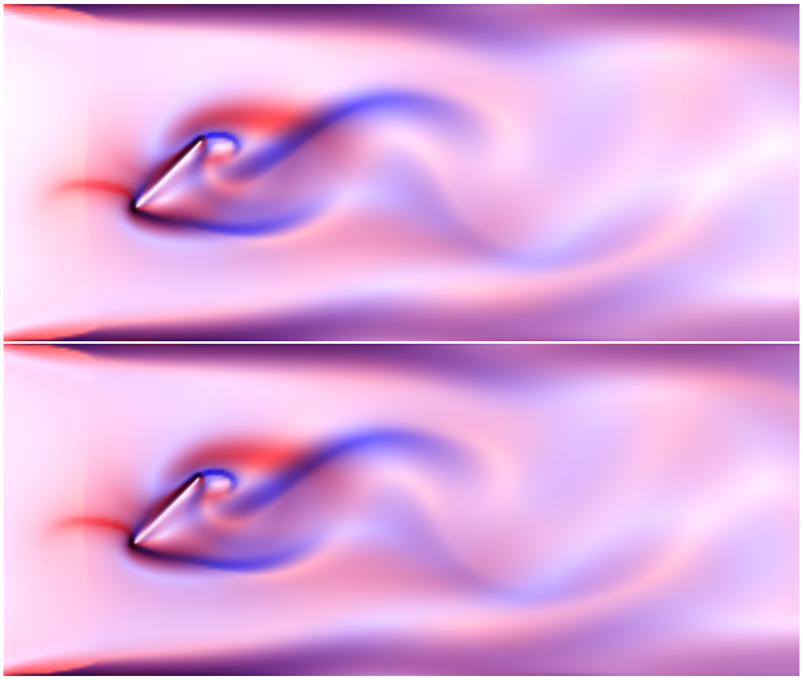

Fig. 2 Visual comparison between FTLE and FTVA computed for a time slice of the tilted bar dataset with integration time 1.0. **Top:** FTLE field **Bottom:** FTVA field with 1 particle per position and $\varepsilon = 0$

particles per position has been set to 100 and the error has been varied. As a result, if too few particles are chosen the distribution cannot be approximated correctly. Hence the FTVA fields become disturbed, which manifests in an un-smooth color map. As expected, this effect unfolds with increasing error.

5.2 *Turbulent Jet Flow*

Our second dataset is a swirling jet flow entering the domain, containing resting fluid, to the left. The dataset is steady, consist of 124×101 quad cells and is highly turbulent. Again, there is almost no visible difference between a visualization of the FTLE and the FTVA fields (see Fig. 6). As was the case in the former example dataset the relative strength of the LCS in the FTVA fields weakens and are less sharp. This means the error does have a diffusion like effect on the LCS. However, even in the presence of a large error the LCS remain visible clearly (see Fig. 7).

The lion's share of the computation time is spent integrating particles for the stochastic flow map, which is about 99% of the timings given in Fig. 5. These tim-

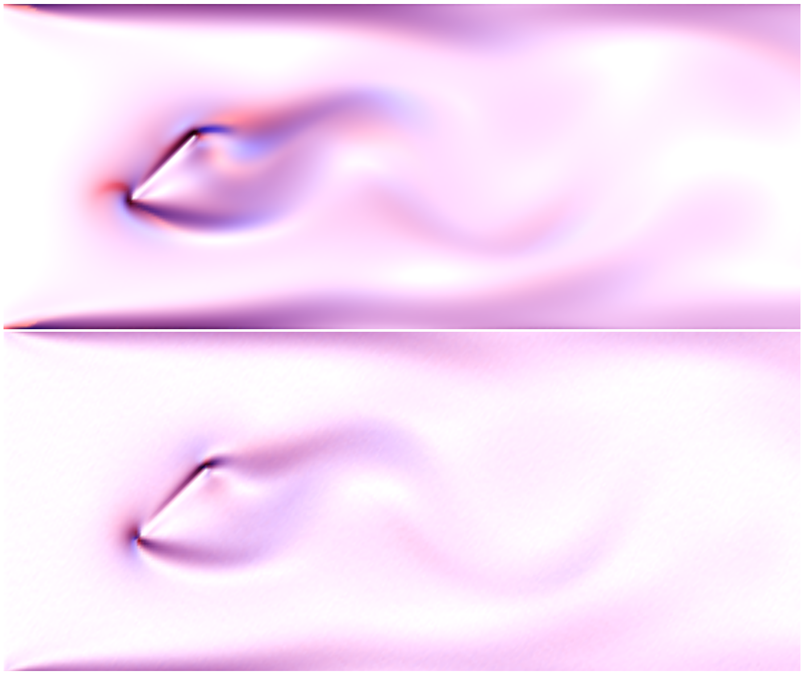

Fig. 3 Sequence of images for the tilted bar dataset, showing the relative weakening of LCS for growing error, obtained for an integration time of 0.4 with different errors: **Top:** 10 Particles and $\varepsilon = 0.01$ **Bottom:** 500 particles and $\varepsilon = 1.0$

ings are given for a non-parallelized version of the algorithm executed on an Intel Xeon CPU E5620 CPU with 2.4 GHz.

6 Conclusion and Future Work

In this paper we have proposed a method to compute an FTLE-like measure called FTVA (finite time variance analysis) to find regions of converging and diverging flow in uncertain flow fields. We have produced promising visualization results, however, we would like to study application examples with naturally arising uncertain vector fields. Despite a successful implementation of the stochastic flow map further research is needed to limit the high computational cost (see Tab. 5). Therefore, we will look into an adaptive approach for the automatic determination of the number of stochastic trajectories necessary per position, which would be highly desirable. Furthermore, we want to research deeper into the differences and similarities of the covariance matrix and the Cauchy-Green stress tensor and exploit the results to improve our method. Also a parallelized version should be much faster and additionally acceleration techniques [31, 9] can be implemented as well.

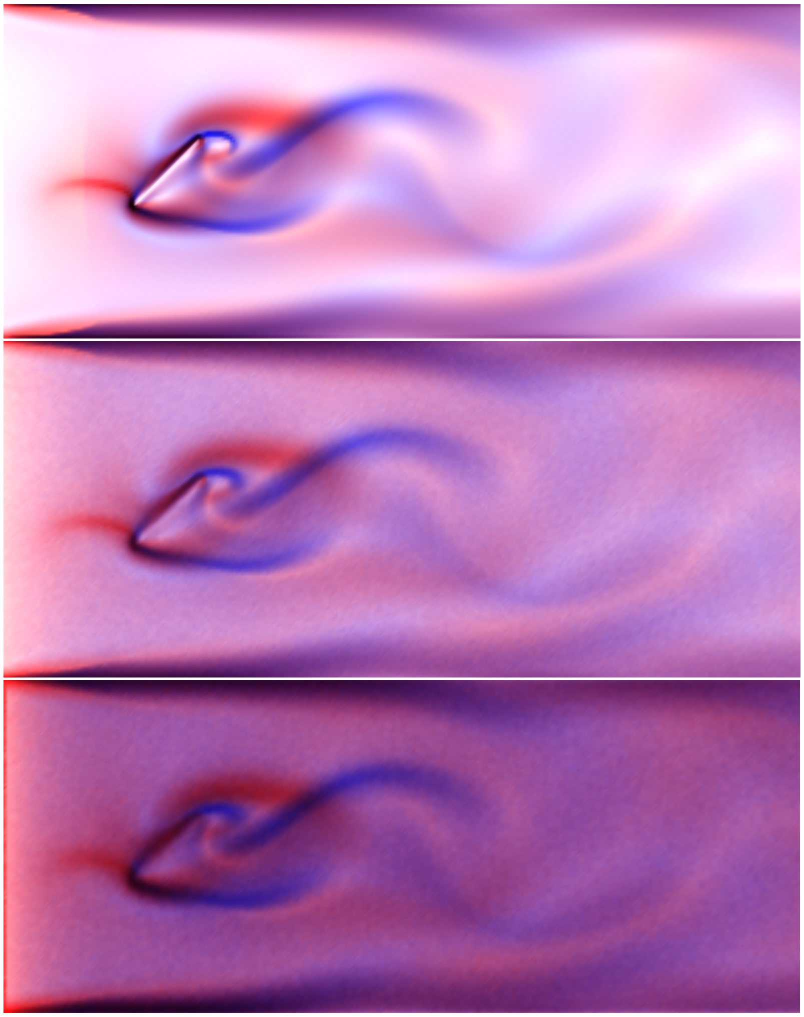

Fig. 4 Comparison between FTVA results for the tilted bar dataset for a constant number of particles (100 particles per position) to show the influence of error on the visualizations: **Top:** integration length 1.0 and $\varepsilon = 0.1$ **Middle:** integration length 1.0 and $\varepsilon = 1.0$ **Bottom:** integration length 1.0 and $\varepsilon = 2.0$

No. particles per position	No. particles total	Computation time
10	$7.92\ 10^5$	7min
100	$7.92\ 10^6$	1h 15min
500	$3.96\ 10^7$	6h 18min
1000	$7.92\ 10^7$	12h 41min

Fig. 5 Computation times for the tilted bar dataset for different number of particles per position. Timings are given for non-parallel integration.

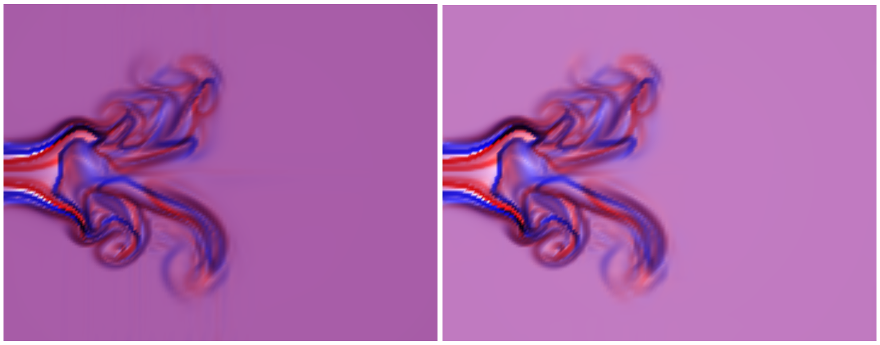

Fig. 6 Comparison between FTLE (**left**) and FTVA (**right**) with 1 particle per position and $\varepsilon = 0$ for the jet flow dataset

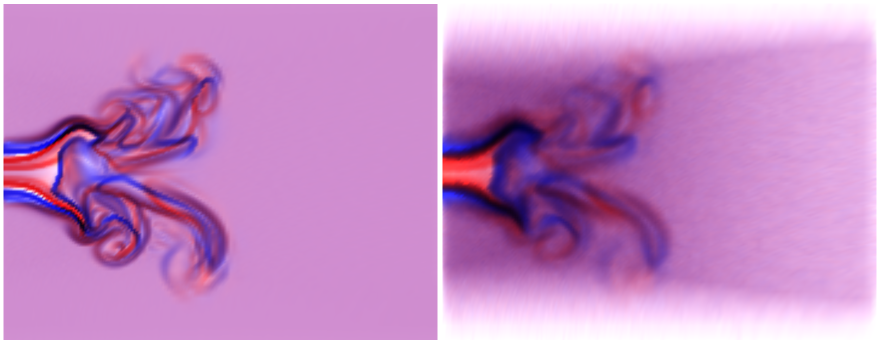

Fig. 7 Visualizations for the jet flow dataset showing the blurring of the FTVA structures for integration length 1, 100 particles per position and varying error **Left:** $\varepsilon = 1$ **Right:** $\varepsilon = 5$

The extention of the presented concepts to 3D is straight forward: First, the stochastic flow map computation needs to be extended to 3D. Second, the stochastic flow map needs to be computed for every grid point. Third, The PCA needs to be computed for all neighboring grid points and the advected distribution from these particles as given by the stochastic flow map. Neither of which poses a major hurdle.

References

1.
2. Francois Lekien A, Chad Coulliette A, Arthur J. Mariano D, Edward H. Ryan D, Lynn K. Shay D, George Haller E, Jerry Marsden A, and Communicated U. Frisch. Pollution release tied to invariant manifolds: A case study for the coast of florida. *Physica D*, 1.
3. Ralf P. Botchen and Daniel Weiskopf. Texture-based visualization of uncertainty in flow fields. *Visualization Conference, IEEE*, 0:82, 2005.
4. Ralf P. Botchen, Daniel Weiskopf, and Thomas Ertl. Interactive visualization of uncertainty in flow fields using texture-based techniques. 2005.

5. Ross Brown. Animated visual vibrations as an uncertainty visualisation technique. In *Proceedings of GRAPHITE*, GRAPHITE ’04, pages 84–89, New York, NY, USA, 2004. ACM.
6. Guoning Chen, Konstantin Mischaikow, Robert S. Laramee, and Eugene Zhang. Efficient morse decompositions of vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 14:848–862, July 2008.
7. Suzana Djurcilov, Kwansik Kim, Pierre Lermusiaux, and Alex Pang. Visualizing scalar volumetric data with uncertainty. *Computers and Graphics*, 26(2):239 – 248, 2002.
8. Raphael Fuchs, Jan Kemmler, Benjamin Schindler, Filip Sadlo, Helwig Hauser, and Ronald Peikert. Toward a lagrangian vector field topology. *Computer Graphics Forum*, 29(3):1163–1172, 2010.
9. Christoph Garth, Florian Gerhardt, Xavier Tricoche, and Hans Hagen. Efficient computation and visualization of coherent structures in fluid flow applications. *IEEE Transactions on Visualization and Computer Graphics*, 13:1464–1471, 2007.
10. Christoph Garth, Guo-Shi Li, Xavier Tricoche, Charles D. Hansen, and Hans Hagen. Visualization of coherent structures in transient 2d flows. In *Topology-Based Methods in Visualization II*. Springer.
11. Christoph Garth, Alexander Wiebel, Xavier Tricoche, Ken Joy, and Gerik Scheuermann. Lagrangian visualization of flow-embedded surface structures. *Computer Graphics Forum*, 27(3):1007–1014, 2008.
12. Al Globus, Creon Levit, and Tom Lasinski. A tool for visualizing the topology of three-dimensional vector fields. In *Proceedings of IEEE Visualization ’91*, pages 33 – 40, October 1991.
13. Henning Griethe and Heidrun Schumann. The visualization of uncertain data: Methods and problems. In *SimVis*, pages 143–156, 2006.
14. Gevorg Grigoryan and Penny Rheingans. Point-based probabilistic surfaces to show surface uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 10:564–573, 2004.
15. G. Haller. Distinguished material surfaces and coherent structures in three-dimensional flows. *Physica D*, 149:248 – 277, 2001.
16. G. Haller. Lagrangian coherent structures from approximate velocity data. *Phys. Fluids*, A14:1851 – 1861, 2002.
17. James L. Helman and Lambertus Hesselink. Visualizing Vector Field Topology in Fluid Flows. *IEEE Computer Graphics and Applications*, 11(3):36 – 46, May 1991.
18. Chris Johnson. Top scientific visualization research problems. *IEEE Comput. Graph. Appl.*, 24:13–17, July 2004.
19. Chris R. Johnson and Allen R. Sanderson. A next step: Visualizing errors and uncertainty. *IEEE Computer Graphics and Applications*, 23:6–10, 2003.
20. Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, 1992.
21. Hongwei Li, Chi-Wing Fu, Yinggang Li, and Andrew Hanson. Visualizing large-scale uncertainty in astrophysical data. *IEEE Transactions on Visualization and Computer Graphics*, 13:1640–1647, 2007.
22. Suresh K. Lodha, Alex Pang, Robert E. Sheehan, and Craig M. Wittenbrink. Uflow: visualizing uncertainty in fluid flow. In *Proceedings of the 7th conference on Visualization ’96*, VIS ’96, pages 249–ff., Los Alamitos, CA, USA, 1996. IEEE Computer Society Press.
23. Helwig Löffelmann, Thomas Kucera, and Eduard Gröller. Visualizing poincar maps together with the underlying flow. In *In International Workshop on Visualization and Mathematics’97 Proceedings*, pages 315–328. Springer, 1997.
24. Adriano Lopes and Ken Brodlie. Accuracy in 3d particle tracing. In *Mathematical Visualization*, pages 329–341. SpringerVerlag, 1998.
25. Claes Lundstrom, Patric Ljung, Anders Persson, and Anders Ynnerman. Uncertainty visualization in medical volume rendering using probabilistic animation. *IEEE Transactions on Visualization and Computer Graphics*, 13:1648–1655, 2007.
26. M. Otto, T. Germer, H.-C. Hege, and H. Theisel. Uncertain 2d vector field topology. In *Computer Graphics Forum (Proceedings of Eurographics 2010)*, Norrköping, Sweden, 5 2010.

27. Alex T. Pang, Craig M. Wittenbrink, and Suresh K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13:370–390, 1997.
28. Armin Pobitzer, Ronald Peikert, Raphael Fuchs, Benjamin Schindler, Alexander Kuhn, Holger Theisel, Kresimir Matkovic, and Helwig Hauser. On the way towards topology-based visualization of unsteady flow - the state of the art. In *EuroGraphics 2010 State of the Art Reports (STARs)*, pages 137–154, 2010.
29. Philip E. Protter. *Stochastic Integration and Differential Equations*. Springer, 2003.
30. Philip J. Rhodes, Robert S. Laramee, R. Daniel Bergeron, and Ted M. Sparr. Uncertainty visualization methods in isosurface volume rendering. In *Eurographics 2003, Short Papers*, pages 83–88, 2003.
31. Filip Sadlo, Alessandro Rigazzi, and Ronald Peikert. Time-dependent visualization of lagrangian coherent structures by grid advection. In *Topological Methods in Data Analysis and Visualization*. Springer.
32. Filip Sadlo and Daniel Weiskopf. Time-dependent 2-d vector field topology: An approach inspired by lagrangian coherent structures. *Computer Graphics Forum*, 29(1):88–100, 2010.
33. Allen R. Sanderson, Chris R. Johnson, and Robert M. Kirby. Display of vector fields using a reaction-diffusion model. In *Proceedings of the conference on Visualization '04*, VIS '04, pages 115–122, Washington, DC, USA, 2004. IEEE Computer Society.
34. Timothy Sauer. Numerical solution of stochastic differential equations in finance. Springer, to appear.
35. Visualizing Lagrangian Coherent Structures and Comparison to Vector Field Topology. Filip sadlo and ronald peikert. In *Topology-Based Methods in Visualization II*. Springer.
36. Holger Theisel, Tino Weinkauf, Hans-Christian Hege, and Hans-Peter Seidel. Saddle connectors - an approach to visualizing the topological skeleton of complex 3d vector fields. *Visualization Conference, IEEE*, 0:30, 2003.
37. Tino Weinkauf and Holger Theisel. Streak lines as tangent curves of a derived vector field. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization 2010)*, 16(6):1225–1234, November - December 2010.
38. Thomas Wischgoll and Gerik Scheuermann. Locating closed streamlines in 3d vector fields. In *Proceedings of the symposium on Data Visualisation 2002*, VISSYM '02, pages 227–ff, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.
39. Craig M. Wittenbrink, Alex T. Pang, and Suresh K. Lodha. Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 2:266–279, September 1996.
40. Björn Zehner, Norihiro Watanabe, and Olaf Kolditz. Visualization of gridded scalar data with uncertainty in geosciences. *Comput. Geosci.*, 36:1268–1275, October 2010.

Acknowledgements We want to thank the anonymous reviewer for his or her valuable suggestion about the modification of the algorithm in case of stochastic dependency. Furthermore, we want to thank Wolfgang Kollman for the turbulent jet stream dataset and Christian Heine for fruitful discussions and valuable hints. This work was partially funded by DFG grant SCHE 663/3-8.