Analysis of Streamline Separation at Infinity Using Time-Discrete Markov Chains

Wieland Reich and Gerik Scheuermann, *Member, IEEE*

Abstract—Existing methods for analyzing separation of streamlines are often restricted to a finite time or a local area. In our paper we introduce a new method that complements them by allowing an infinite-time-evaluation of steady planar vector fields. Our algorithm unifies combinatorial and probabilistic methods and introduces the concept of separation in time-discrete Markov-Chains. We compute particle distributions instead of the streamlines of single particles. We encode the flow into a map and then into a transition matrix for each time direction. Finally, we compare the results of our grid-independent algorithm to the popular Finite-Time-Lyapunov-Exponents and discuss the discrepancies.

Index Terms—Vector field topology, flow visualization, feature extraction, uncertainty.

1 Introduction

Vector fields are traditional objects of major interest for visualization. The significance of these objects is due to their key role in description of many notions in physics and engineering sciences. This is especially true for fluid mechanics, vehicle and aircraft design, and weather predictions. From the theoretical point of view, vector fields have received much attention from mathematicians, leading to a precise and rigorous framework that greatly facilitates their practical study.

In particular, separation and attachment features are of essential interest in many practical studies due to their adverse effects on the object motion and their implication in vortex genesis. These structures are used to gain insights about essential flow properties for particular applications.

Analysis and visualization of the topology poses several challenges. The phases from generating data (PIV-measurements and CFD-simulations) to processing (interpolation and numerical approximation) are affected by inherent errors. According to [11], the visualization of these errors has often been underrepresented in the past.

Many visualization algorithms rely on the extraction of individual streamlines of the flow associated with a vector field, which are sensitive to a given initial value.

Due to the fact that the streamlines associated with a vector field are dense, it is impossible to deeply study each single representative. As a remedy, one can assign a state to each particle that is not determined by its position but the cell that contains the particle. The movement between those cells, having been successfully integrated in different combinatorial models [22, 2], can also be seen as stochastic process, i.e., the state of a particle changes after a certain time passes. This leads to the theory of time-discrete Markov processes, which are a well explored domain of probabilistic theory and provide answers to a central question one can pose: How likely is it, that multiple particles that are placed at the same region in the flow came from and will end up in the same limit set?

The mathematical foundations for this purpose are found in section 3 and 4, our proposed algorithm in 5 and 6.

2 Related Work

Since Helman and Hesselink [8] introduced the visualization of topological skeletons of flow fields, much research has been done in this area. A significant amount of this research was and still is the extraction of invariants and separating structures in steady and unsteady vector fields.

The original method in [8] was based on a linearization of the flow in a neighborhood of special singularities, called saddle-points. The particle traces started from the stable and unstable manifolds of the saddle-points, called separatrices, divide the domain into classes of equivalent flow behaviour. Their method was later extended by Wischgoll et al. [31], who created an algorithm to find closed streamlines in planar vector fields by searching for cells that get repeatedly crossed by particles. The complexity of the latter method benefits strongly from the Poincaré-Bendixson-Theorem, which ensures that a closed streamline in a 2D vector field must contain at least one singularity.

Scheuermann et al. [24] included the boundary of the domain to generate more separating streamlines and eventually a finer topology. For this method it is necessary that one computes all points on the boundary where the vector field is tangential and traces additional particles from there.

Kenwright et al. [14] used a parallel vector operator [20] to extract local separation lines in piecewise linear flows. The parallel vector operator detects whether the velocity is parallel to an eigenvector of the Jacobian on the boundary of a triangle cell and links these points to separation and attachment lines. Due to fact that the Jacobian in piecewise linear vector fields is not continuous, the resulting structures are not necessarily connected.

Many techniques that visualize Lagrangian coherent structures, such as [6, 5, 23, 1, 13], are based on the Finite-Time-Lyapunov-Exponents (FTLE) [7]. They use a linear approximation, called finite time Jacobian, of a map that is generated by the integration of neighboring initial points for a finite time. The finite time Jacobian is then multiplied with its transposed to eliminate the rotational movement. The resulting matrix is called the Cauchy-Green-deformation-tensor, which describes the deformation of a small volume by the flow. The largest eigenvalue of the Cauchy-Green-deformation-tensor is the visualized parameter here. The method is applicable to steady and time-dependent vector field data.

Reininghaus and Hotz developed a purely combinatorial approach to extract the topological skeleton of 2D vector fields [22]. They construct a simplicial graph from a vector field on a triangular mesh by linking the simplices with nearby simplices of lower dimension, i.e., edges and points. The theoretical basis is given by Forman's work on discrete Morse theory [4].

Another combinatorial approach was made by Chen et al. [2]. Their technique of encoding the vector field into a graph is to compute integration images of triangular cells. The graph is then processed by computing strongly connected components and the vector field invari-

- *Wieland Reich is with University of Leipzig, E-mail: reich@informatik.uni-leipzig.de.*
- *Gerik Scheuermann is with University of Leipzig, E-mail: scheuermann@informatik.uni-leipzig.de.*

Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

ants are classified by the relative homology of their isolating neighborhood (Conley-Index).

Surveys covering more topics of flow visualization are, e.g., [30, 21].

There has also been a growing interest in developing methods to research uncertainty in vector field topology over the past years. A frequently used idea for these purposes is exploring particle distributions instead of single particles. Otto et al. [17] formulated convergence criteria for Gaussian distributed density functions by Euler-integration. Their method also uses the uncertain Poincaré-Index to distinguish between critical distributions.

Schneider et al. [25] considered numerical integration inflicted by bounded uniformly distributed errors and replaced the Finite-Time-Lyapunov-Exponent by a principal component analysis of particle destinations, named finite time variance analysis (FTVA). In contrast to our approach, which is based on uncertain cell mappings, both techniques use an uncertain integration scheme. Furthermore, our considerations will not be restricted to a finite time like in [25].

Readers interested in detecting and tracking more features of flow data are also referred to [29].

While there is a comprehensive selection of tools to visualize uncertain data [19], we will use the classic dual color map.

Stochastic processes, in particular random walks, are not only useful to describe the flow of uncertain particles, but they have also been used to smooth and denoise vector field data [18].

3 Vector Fields, Invariants, and Lyapunov exponents

Let $X' = F(X)$ be a differential equation defined on $\mathbb{R}^2$, then the associated flow is a continuous function $\Phi : \mathbb{R} \times \mathbb{R}^2 \to \mathbb{R}^2$, satisfying

$$\Phi(0,x) = x,$$

$$\Phi(t_1, \Phi(t_2,x)) = \Phi(t_1 + t_2, x).$$

We have

$$\frac{d}{dt}\Phi(t,x)_{|x_0} = F(x_0).$$

A set $S \subset \mathbb{R}^2$ is an **invariant set** if $\Phi(t,S) = S$ for all $t \in \mathbb{R}$. For example, the streamline of any point $x \in \mathbb{R}^2$ is an invariant set.

A set S is an isolated invariant set if there exists an neighborhood, so that S is the maximal invariant set contained in the neighborhood.

Hyperbolic fixed points and periodic orbits are examples of isolated invariant sets, but also the space of their connecting streamlines, e.g. saddle connectors. While fixed points can be found by interpolation, an algorithm for extracting invariants of more complicated behaviour can be found in [2] and [3].

The **alpha-** and **omega limit sets** of $x \in \mathbb{R}^2$ are

$$\alpha(x) = \bigcap_{t \in]-\infty, 0[} \overline{\Phi(t,x)}$$

$$\omega(x) = \bigcap_{t \in]0,\infty[} \overline{\Phi(t,x)}$$

where the overline denotes the closure of a set. Equivalence classes generated by the limit sets of each x induce a topology in $\mathbb{R}^2$.

In mathematics the **Lyapunov exponent** of a dynamical system is a quantify that characterizes the rate of separation of infinitesimally close streamlines Φ_1 and Φ_2:

$$\gamma(x) = \lim_{t \to \infty} \lim_{x_1 \to x_2} ln \frac{||\Phi_1(x_1,t) - \Phi_2(x_2,t)||}{||x_1 - x_2||}.$$

To determine the expansion rate for particles advected by the flow for a finite time t, the **Finite-Time-Lyapunov-Exponent** is a established method:

$$FTLE^t_{x_0}(x) = \frac{1}{|t|} ln \sqrt{\lambda_{max}(\Delta(x,t,t_0))},$$

where

$$\Delta(x,t,t_0) = (\Delta\Phi^{t_0+t}_{t_0} x)^T (\Delta\Phi^{t_0+t}_{t_0} x)$$

is the so-called Cauchy-Green deformation tensor, the finite time Jacobian multiplied with its transposed. λ_{max} denotes the largest eigenvalue of the matrix. It can be interpreted as the square of the largest rate, that an object of small volume will be stretched by integration time t. The eigenvector to the largest eigenvalue λ_{max} points in the direction of largest stretch.

Further explanations can be found in and [7] and [10].

4 Discrete Markov Chains

Stochastic processes, e.g. Brownian motions or random walks, have a far-reaching domain of applications in almost all scientific research areas. For the sake of simplicity, we restrict our theoretical considerations to a special state of stochastic processes, those which have discrete states and discrete time steps. These models lead to time-discrete Markov chains, which are random processes being considered memoryless, i.e., the next state of the system only depends on the current state and not to the sequence of events before. Is the process time-homogeneous, it can be expressed by a single matrix. For a time-inhomogeneous process, the matrix might change from one step to another.

The meaning of these matrices can be easily followed with basic knowledge of linear algebra and probability theory. We give an example.

Imagine two neighboring countries with the same amount of population, i.e. the 2d probability distribution vector only contains entries of 0.5. Each year 10 per cent of the population of country A migrate to country B and 20 per cent of country B migrate to country A. The resulting matrix, that encodes the process, is

$$T = \begin{pmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{pmatrix}.$$

If we are interested in the population distribution after one year, one has to apply the matrix once to the transposed of the probability distribution vector from the right

$$\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}^T \cdot \begin{pmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{pmatrix} = \begin{pmatrix} 0.55 \\ 0.45 \end{pmatrix},$$

so the population of A is, of course, growing, but how does the system behave after a very large number of years? Applying the matrix T many times results in the vector

$$\begin{pmatrix} 0.\overline{6} \\ 0.\overline{3} \end{pmatrix},$$

with any more multiplications having no effect, so we found a (left-)eigenvector of T to the eigenvalue of 1, which is the stationary state of the population.

We now generalize the concept of transition matrices and state the most important properties and theorems, for detailed proofs we refer to [27]. However, there are a lot of books which explain the topic in an adequate way.

A vector v is a probability distribution vector if all entries v_i are greater or equal to 0 and

$$\sum_i v_i = 1.$$

A real matrix T is called (row-)stochastic if all entries t_{ij} are greater or equal to 0 and

$$\sum_j t_{ij} = 1.$$

We will refer to it as **transition matrix**.

The entry t_{ij} of the transition matrix describes the probability of the system from going from state i to state j. A transition matrix is a linear operator, that maps probability distribution vector to another probability distribution vector. The exploration of the long time

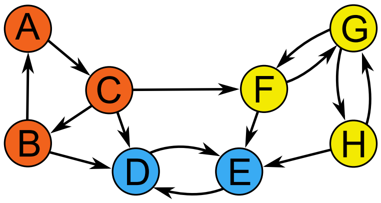


Fig. 1. A drawing of strongly connected components. Different colors describe pairwise disjoint sets of the graph's nodes. In each set exists a path from each node to all others.

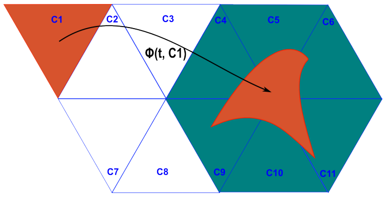


Fig. 2. An illustration of an outer approximation of a cell. The image of the cell C_1 under a fixed integration time t is approximated by all cells C_i, for which $\Phi(t, C_1) \cap C_i$ is not empty.

behaviour of these vectors by repeated multiplication with T is of particular interest in the theory of stochastic processes.
T is bounded in its norm by 1 and so so are its eigenvalues. Although all entries of T are positive and real, negative and complex eigenvalues are common.
We call a eigenvalue λ **dominant** if $|\lambda| = 1$. If a eigenvalue is not dominant, its influence vanishes with each potentiation and all vectors of the corresponding eigenspace will be mapped to zero after a very large number of time steps.
Moreover, the existence of at least one eigenvalue $\lambda = 1$ is guaranteed for every transition matrix T. Its eigenspace is called **stationary state** of the system. A system may have many stationary states and so $\lambda = 1$ may be a repeated eigenvalue. In this case, the stationary distribution is depending on the initial distribution vector, that was presumed. A condition that ensures a unique stationary distribution is
Theorem(Perron-Frobenius):
Let T be a transition matrix and all entries t_{ij} are greater than 0. Then there exists an eigenvalue $\lambda = 1$, that is of multiplicity 1 and the corresponding eigenvector only has positive entries. Further, all distribution vectors converge against the same unique stationary distribution vector.

Even under weaker conditions one can assert a unique stationary distribution. For this purpose we need another Definition:
A transition matrix T is called **irreducible**, when each state can be reached from any other state. Else it is reducible.
The indices of a reducible matrix can be reordered, so that it is of the block-form

$$\begin{pmatrix} T_{MM} & T_{MN} \\ 0 & T_{NN} \end{pmatrix}.$$

Theorem:
Is a transition matrix T irreducible and all entries t_{ij} are greater or equal to 0, then the eigenvalue $\lambda = 1$ is of multiplicity 1.

This weaker formulation does not exclude the possibility, that there might be other eigenvalues fulfilling $|\lambda| = 1$.
There is a relation between the matrix T and an directed Graph G_T. If G_T has exactly n vertices, where n is the size of T, and there is an edge from vertex i to vertex j precisely when $t_{ij} > 0$. Then T is irreducible if and only if G_T is strongly connected. Strongly connected components are an essential processing stage of the Morse-decomposition of Chen et al. [2]. An example of strongly connected components is illustrated in Figure 1.

One has to bring in mind that there still exist transition matrices that do not converge. Just consider the so-called Ehrenfest-Matrix

$$\begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{pmatrix},$$

which is periodic by satisfying $T^n = T^{n+2}$. However, these types of matrices form a compact subset of all transition matrices and small errors will turn them into convergent ones, so many numerically sensitive processes will lead to convergent transition matrices. Some of them might converge very slowly. The numerical instability of non-convergent matrices explains why we were never able to construct a divergent Markov chain by the flow induced by a planar vector field as explained in the next section.

5 Construction of a Time-Discrete Markov Chain generated by a Planar Vector Field

Let us look back to the population model that we introduced as an example in the last section. One could simply assume that, if we replace the countries by the cells in our dataset and the population by the arbitrarily placed particles in the flow, the job is already done. This is wrong.
We cannot only trace a few streamlines and assume that the image of a cell lies entirely in the obtained cells. There may be, depending how expansive the flow is, many holes in the integration image. On the other hand, a very dense sampling of the cell could lead to very high computational costs. A stronger mathematical foundation is needed.

5.1 The Outer Approximation

Chen et al. developed a rigorous and efficient algorithm which can reconstruct the image of a triangle under a fixed integration time. The coarse idea is to adaptively integrate all edges and fill the resulting hole. For a detailed description of the algorithm see chapter 4 of [2]. Our chosen example contains a triangulated mesh (Figure 2), although this method can be easily applied to any other types of cells, too. An enclosing technique of flows sampled on cubical grids can be found in the publication of Mrozek et al. [16].
A set of cells B_i is called an outer approximation of a cell A, if the image of A under a fixed integration time t is completely contained in $\bigcup_i B_i$. It has been proven in [12] that these maps deliver a valid approximation of the flow. Chen et al. encode this mapping into a graph, use the extraction of strongly connected components and apply the computation of a powerful, though theoretically demanding, topological invariant, the Conley Index, to obtain a Morse Decomposition of the vector field.
Our approach in processing is to construct a high-dimensional sparse transition matrix from the particle movement in the field, because there are no generalized algorithms, that compute strongly connected components of a graph with weights on edges.
A sparse matrix is a matrix populated primarily with zeros. Specialized data structures can take advantage of that and do not need to store the zero elements, which allows for fast computations and requires less memory. In particular, the multiplication of a sparse matrix with a vector is reduced to linear complexity.
As mentioned in the related work, our algorithm does not use an uncertain integration scheme like in [17, 25], but an uncertain destination cell. We now reformulate the definition of the outer approximation in the context of probability:
A set of cells B_i is called an **outer approximation** of a cell A, if all particles traced from A for a fixed time t are reaching $\bigcup_i B_i$ with the probability of 1.
We will use the model of a uniform distribution, i.e. our particles reach all cells of the outer approximation with equal probability. The entry t_{ij} is non-zero, if and only if cell j is included in the outer approximation by integration from cell i by one time step and its value is the

reciprocal of the total number of cells contained in the outer approximation. For example, if the image of cell 1 is cell 2,3,7 and 15, then the entries of $t_{1,2}$, $t_{1,3}$, $t_{1,7}$ and $t_{1,15}$ will be set to 0.25. Other models are discussed in the future work section.

We cannot assume a Gaussian distribution as a particle destination, because if we allow particles to move to each cell of the data with probability greater zero, our transition matrix will become dense and efficient computations for big datasets will become impossible. However, such a consideration might be interesting for small datasets, because a completely dense matrix would ensure that the conditions of the Perron-Frobenius-Theorem are fulfilled. Furthermore, a Gaussian probability density function would rather fit to a particle in quantum mechanics, but not fluid mechanics, which is one of the most common sources of discrete vector field data.

After we calculated the outer approximation for each cell, we can fill the transition matrix. It is sparse, because the number of image cells is usually much smaller than the total amount of cells in the dataset.

To determine the state of the system in the next time step, not the time range of the integration is raised, but the distribution vector is multiplied with the transition matrix another time:

$$v_{n+1} = v_n^T \cdot T.$$

One cannot derive the backward particle movement by the inversion of T, because we have no guarantee that its eigenvalues are all different from zero, and even if this is the case, the inversion of such a high dimensional matrix is a numerically unstable operation. As a solution we construct two transition matrices, one for the forward integration (T_+) and one for the backward integration (T_-). The stationary probability vectors of T_+ are cells, that contain the attracting invariant sets, those of T_- are containing the repelling ones. Dellnitz et al. already formulated a method to find these invariants by eigenvector computation [3]. There are several public available tools that can solve eigenvalue problems of high-dimensional sparse matrices, e.g. [9].

We have chosen some examples of invariants to illustrate in Figure 3, where it visible, that there is always a duality between a transition matrix and a probabilistic graph. The matrices have to be considered as a block or multiple blocks of a matrix of much higher dimension. The matrices concerning these examples are

- (a) $\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$,
  which is convergent by potentiation.
- (b) $\begin{pmatrix} 0 & a & 0 & 0 & 0 & 1-a \\ 0 & 0 & b & 0 & 0 & 1-b \\ 0 & 0 & 0 & c & 0 & 1-c \\ 0 & 0 & 0 & 0 & d & 1-d \\ e & 0 & 0 & 0 & 0 & 1-e \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$,
  which is convergent, if one of the probability parameters is not exactly 1.0.
- (c) $\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$,
  which has only eigenvalues on the unit circle in the plane of complex numbers and is divergent.
- (d) $\begin{pmatrix} 0 & a & 1-a & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & b & 1-b \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$,
  which is convergent, if one of the probability parameters is not exactly 1 or exactly 0.

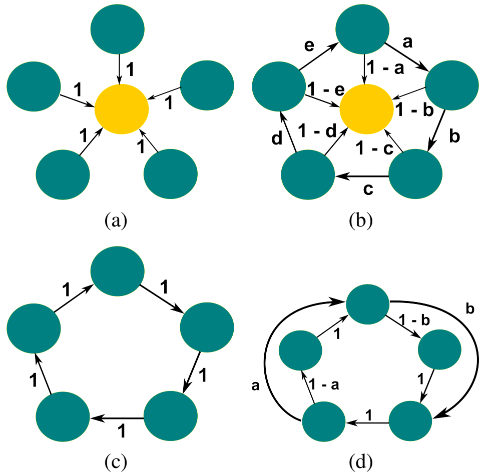


Fig. 3. Examples of different invariants in the context of probabilistic mappings: (a) attracting node, (b) attracting focus, (c) ideal closed orbit, (d) realistic closed orbit. The balls may represent cells or unions of cells.

As a consequence, a time-discrete Markov chain is not necessarily convergent, when it is created by the flow of a single particle. This could lead to an ideal periodic orbit (c). Due to the fact, that we use particle distributions, some particles might reach cells some time steps earlier or later than the majority in the context of cell mappings - the distribution will get blurred alongside the closed streamline, leading to (d).

The reader might already noticed, that saddles and separatrices are missing in these examples. The reason is, that these structures in general cannot be found be eigenvector computation, neither of T_+, nor T_-, because particle distributions won't converge against these invariants. This issue that was also mentioned in [17] . This was the actual motivation for us to write this paper. In Section 6 we will introduce the degree of uncertainty, which removes this shortcoming and is a measure for separation of convergent transition matrices and eventually separating structures in the vector field.

5.2 Boundary Topology

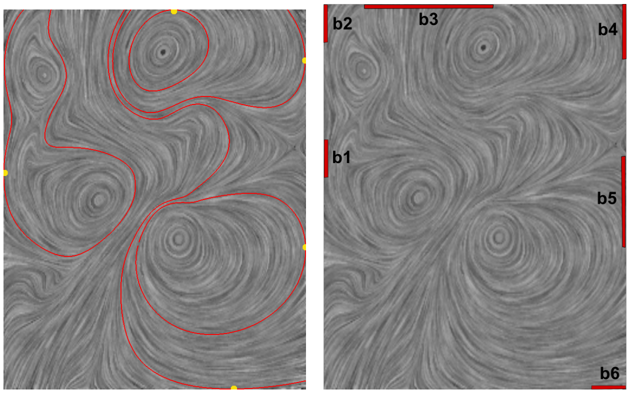
We cannot claim to have a complete algorithm that determines streamline separation at infinity without including the boundary topology. An existing algorithm by Scheuermann et al. [24] computes all inner tangential points on the boundary of the dataset and traces generalized separatrices from there, which divide the vector field into additional equivalence classes of flow behaviour. Our method is similar, except that we extract connected regions of outflow on the boundary, which are often bordered by such tangential points. Both methods are illustrated in Figure 4.

These computed exit sets will be treated as additional cells, i.e., if a particle crosses the boundary, it will be automatically mapped to the corresponding cell which represents the exit set. Further, these cells will be mapped with probability 1 to themselves, so, loosely speaking, particles reaching a certain exit set will stick there forever. The transition matrix has to be extended in size by the number of exit sets, which is normally much smaller than the number of cells in dataset. This has to be done for T_+ and T_-. In the end, the algorithm is able to decide, whether particle distributions are likely to flow into two or more different outflow regions.

A similar method has also been used by Mahrous et al. to improve topological segmentation of 3d vector fields [15].

6 Processing of a Time-Discrete Markov Chain Generated by a Planar Vector Field

The reason, that 2D vector fields generate transition matrices, that are convergent by potentiation, makes it appealing to compute T_+^∞ or T_-^∞.



(a) (b)

Fig. 4. Processing of the boundary: (a) Method by Scheuermann et al., who compute and trace tangential points (yellow) on the boundary. (b) In our method, connected regions of outflow (exit sets) $b1-b6$ are denoted as additional invariants.

Even though we do have a lot acceleration methods, like parallelizing the process or using binary exponentiation, i.e., computing T^{2^n}, this is a quite bad idea. If a matrix T is sparse, it does not automatically hold for its powers. We experienced a heavy rise of computation time with each multiplication step we executed.

The solution for this problem lies in an iteration scheme. We do not need the calculate the stationary distribution for every initial distribution that is possible, but only for unit distributions, so called Dirac-impulses. That means, all entries of the probability distribution vector are zero except at the i-th position, which is 1, representing a distribution located only in the i-th cell. We will refer to it as e_i. The unit vectors e_i will be the initial values for the following iteration process. We apply T to e_i until the result does not change anymore, i.e., given a small $\varepsilon > 0$, we set

$$(s_i)_0 = e_i$$

and compute

$$(s_i)_{n+1} = (s_i)_n^T \cdot T,$$

until the condition

$$||(s_i)_{n+1} - (s_i)_n|| < \varepsilon$$

is fulfilled. That technique has to be executed for all e_i twice, once using T_+ and T_-, and s_{i+} and s_{i-} will be the obtained stationary distribution vectors.

In contrast to computing T_+^∞ or T_-^∞, our matrix will always stay the same and just the values of our vector change.

We experienced the best performance doing the iteration-scheme with dense vectors and converting them back to sparse vectors immediately after they reached the stationary state. An overview over multiple matrix-vector-iteration-schemes can be found in [28].

While all images of e_i under this procedure are contained in the eigenspace of eigenvalue 1 of either T_+ or T_-, it would be hard to visualize the stationary vector for every cell. Instead we analyze how these stationary vectors change in a neighborhood around each cell. This can be done by the ordinary l_1-metric, which measures the distance of the stationary distributions. Defining the neighborhood of a cell i by a number of N cells that have common vertices with i, we introduce the **degrees of uncertainty** by

$$d_{unc+} = \frac{1}{2N} \sum_{j=0}^{N-1} ||s_{i+} - s_{j+}||,$$

$$d_{unc-} = \frac{1}{2N} \sum_{j=0}^{N-1} ||s_{i-} - s_{j-}||.$$

All values inside the metrics had already been calculated by our iteration process. As interpretation, the degree of uncertainty describes how different the stationary distribution obtained by cell i is compared to the stationary distributions in its neighborhood in average. d_{unc} is 0 if all stationary distributions are the same. d_{unc} is 1 if all neighboring cells of cell i lead to stationary states consisting of cells, which are disjoint to those represented by s_i, which is a very exceptional event.

Let us summarize the whole algorithm:

1. Convert the flow field into a collection of cells.
2. Determine the destination probabilities of the particles traced from each cell and fill the transition matrix entries of T_+ and T_-.
3. Iterate all unit distributions, which represent single cells, with the transition matrices to compute the stationary distributions.
4. Determine the measure of flow separation with our propagated formula and use a dual color map to visualize it.

7 Results

We applied our algorithm to artificially generated data (Figure 6 and Figure 5) and CFD-simulations (Figure 7 and Figure 8). For the construction of the outer approximation we used a Runge-Kutta-integration of 4th order. Sparse matrix operations were done by our own implementations. The much bigger part of the computational time was always the matrix iteration process.

We experienced enormous discrepancies in the computational time of the algorithm, which was independent of the number of cells in the data. Vector fields of vanishing rotational behaviour (Figure 6) were handled in less than one minute, while highly rotational fields (Figure 7) took up to several hours. The reason for that is that transition matrices generated by gradient fields often have a less amount of dominating eigenvalues or are even reducible with only one dominating eigenvalue in each block matrix, so that a convergence in linear time of many matrix-vector-iteration processes is ensured [28]. This does not hold for highly rotational fields, where discrete particle distributions take much more time steps to converge.

In Figure 5, the closed streamline could not be detected by using [14], instead 4 lines of local separation were extracted. The FTLE as able to find the closed orbit (assuming, one chooses a sufficient integration time), but still includes the separation lines as features. The time-discrete Markov chain only highlights the attracting closed streamline, because it is the only uncertainty-generating feature in the data concerning backward time integration. Some particles flow to boundary, others to the critical point in the interior, two disjoint stationary distributions, leading to an alternative approach in finding closed streamlines, without checking for cell cycles of single particle movement like in [31].

Figure 6 is the only example, where the results of FTLE and d_{unc} are almost identical.

Due to the fact, that the time-discrete Markov chains only determines the long time behaviour of particle movements, local separation features are completely ignored. Also, the neighborhood of center points in Figure 7, which is containing infinitesimally many closed streamlines, is treated as a one stationary distribution. In these regions, the values of λ_{max} of the FTLE are likely to oscillate by growing integration time, there might be misleading informations of separation features. Also, the separatrices of the saddle on the right were never discovered by any time step we tried in the FTLE. The time-discrete Markov chain had no problems with that feature and also sharply extracted many uncertain destinations of particles on the lower and right boundary of the data (Figure 7(f)). It is remarkable, that the saddle point on the left, unlike to an "ordinary" saddle point, but joining 2 homoclinic periodic orbits, leads to a crossing of two uncertainty lines of the same(!) time direction of integration.

Figure 8 shows, that it is not always easy to guess the ideal time step when using FTLE. The extracted structures may be loose and disconnected for a small integration time (Figure 8(b)) or blurred for a too large integration time (Figure 8(c)). The d_{unc} (Figure 8(d)) is able to reduce these occurrences.

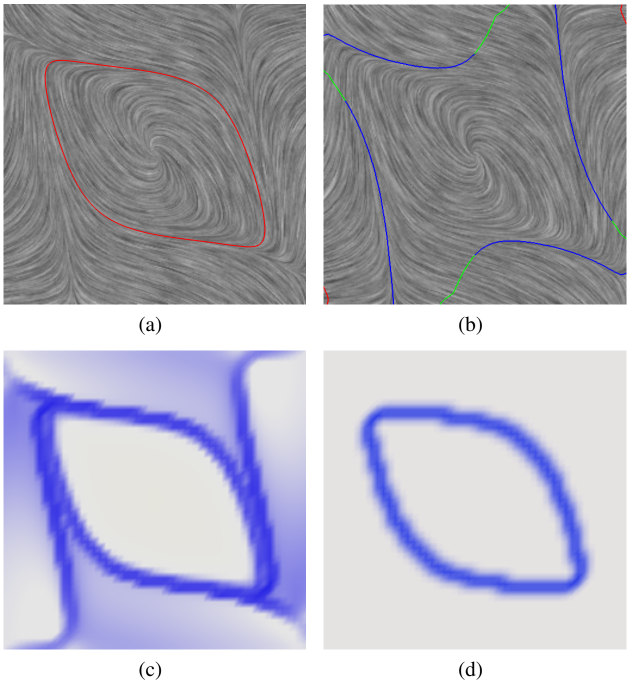


Fig. 5. An attracting closed streamline: (a) a LIC [26] and its exact position (red), (b) separation lines found by the algorithm of Kenwright [14], (c) color map of the FTLE in negative time direction, (d) color map of our method (d_{unc-}) for the same time step like in (c).

Table 1. Comparison of FTLE, FTVA and time-discrete Markov chains.

	FTLE	FTVA	Markov chain
local separation	yes	yes	no
separation at infinite times	no	no	yes
linear complexity	yes	yes	no (future work)
grid-independent	yes	yes	yes
time-dependent data	yes	yes	no (future work)
handles uncertain data	no	yes	yes
includes boundary topology	no	no	yes
free of differential operators	no	yes	yes

Compared to the topological skeleton of a vector field (Figure 7(g)), the d_{unc} is able to detect more structures, in particular features related to the boundary topology. Its separation lines are broader because of the averaging effect that occurs when we include all neighboring cells into the calculation, which also depends on the size of the cells. One has to explore in future work, whether this can be avoided by taking the maximum instead of the average in the calculation of d_{unc}, so that weaker separation features are better visible and do not suffer another low pass filter effect.

8 Conclusion

We have presented an alternative approach to detect separation in steady 2d vector fields by computing the uncertainty that initial particle distributions generate by being transported to their stationary distribution. A brief comparison to FTLE and FTVA can be found in table 1 and the computational times are listed in table 2. Common post-processing methods like the extractions of ridge- and valley-lines are applicable to all of them.

It became clear that increasing the integration time of FTLE does not necessarily lead to better results. The discrete-time Markov chain can fill the gap by allowing infinite-time-evaluation inclusively respecting the boundary topology and eventually being able to detect separation features that were hidden before. It completes existing techniques by getting closer to the real Lyapunov exponents, which pose a very high challenge to be calculated numerically. In the end, the time-discrete Markov chain leads to sharper, less cluttered structures than simply increasing the time for FTLE would do, and, by ignoring local distentions of the flow, to easier interpretations. For the computation of d_{unc+} and d_{unc-} the whole domain of the data is influential, not only values in a local neighborhood. Not every separation feature does automatically lead to higher uncertainty in particle destinations. Sometimes visualizing less is more.

Table 2. Computational times.

dataset	FTLE	Markov chain
figure 6	<1 min	<1 min
figure 7	<1 min	390 min
figure 8	<1 min	172 min

9 Future Work

We admit that the high computational costs have to be considered as the most important issue to solve. One possible solution would be to extend the method with graph analysis, so closed streamlines can be detected earlier and the corresponding cells, which cause the highest computational costs, can be excluded from the process.

As mentioned in the results section, it can be preferable not to average the results over multiple cells.

It is possible that using other models than the uniform distribution of particles over the outer approximation could lead to even better representations, but it would be necessary to sample also the interior of the cell when creating the flow map, which makes this stage of the algorithm much more costly. As a compromise, one could avoid iterating each single cell through the process, but grouping cells by reasonable equivalence relations and iterate initial distributions over these groups instead.

The extension of the method to time-dependent data is in progress. Time-dependent vector fields cannot be expressed by a time-homogeneous Markov chain, i.e. a single matrix, we would need a sparse transition matrix for every time slice. Since there is always a finite number of them, the iteration to ”infinity“ is reduced to a finite product

$$s_i = e_i^T \cdot T_1 \cdot T_2 \cdot \ldots \cdot T_n.$$

That product is surprisingly less costly than our method for steady vector fields. On the other hand, more time is needed to construct the flow maps and one has to expect more influences from the size of the cells. A distribution over many cells, that are very small compared to the rest of the dataset, does not necessarily indicate much separation, neither is guaranteed that s_i contains an invariant set here.

An interesting challenge is to combine Markov processes with integration-free approaches from discrete Morse theory. Due to the fact that simplicial cells in that method are only linked to the neighboring simplices of lower dimension, it could cause the transition matrix to be even less populated and eventually increases the performance.

Also, an extension to 3D is possible, which requires an outer approximation for the image of a 3d cell by integration. The rest of the algorithm would be completely the same.

In case that there will ever occur a divergent time-discrete Markov chain, it is possible that it is periodic instead, like the Ehrenfest matrix, so doing Fourier analysis with the obtained discrete distributions is also an interesting aspect.

Transition matrices also allow a more in-depth exploration of the topology suffering from errors made by integration or interpolation. In the stage of the algorithm, where we process the boundary, we already exploited the fact that we can declare an arbitrarily chosen cell i as an invariant set by setting $t_{i,i}$ to 1. We are able do this practically anywhere and study the changes that it creates. Transition matrices are easy to manipulate.

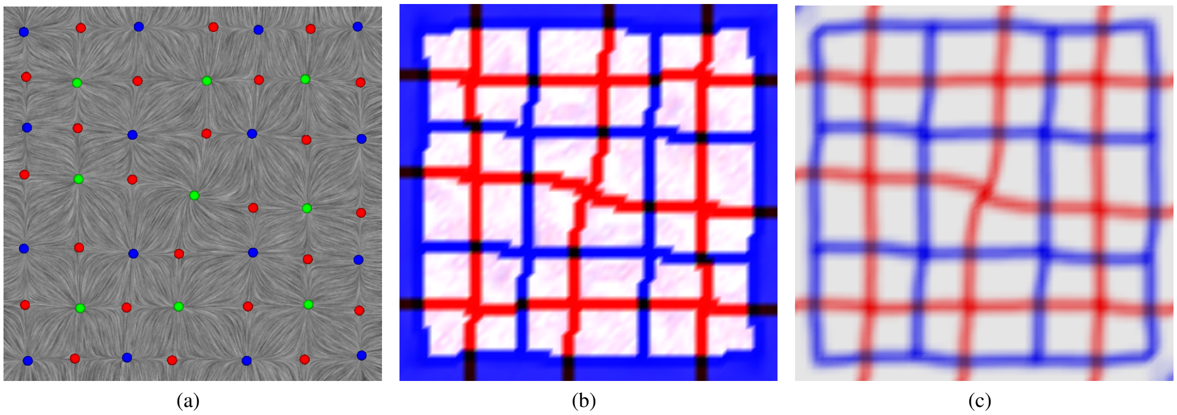


Fig. 6. A gradient vector field containing 49 stationary points: (a) the exact position of the points denoted by spheres (red: saddle, blue: sink, green: source), (b) FTLE (red: forward separation, blue: backward separation), (c) d_{unc+} (red) and d_{unc-} (blue).

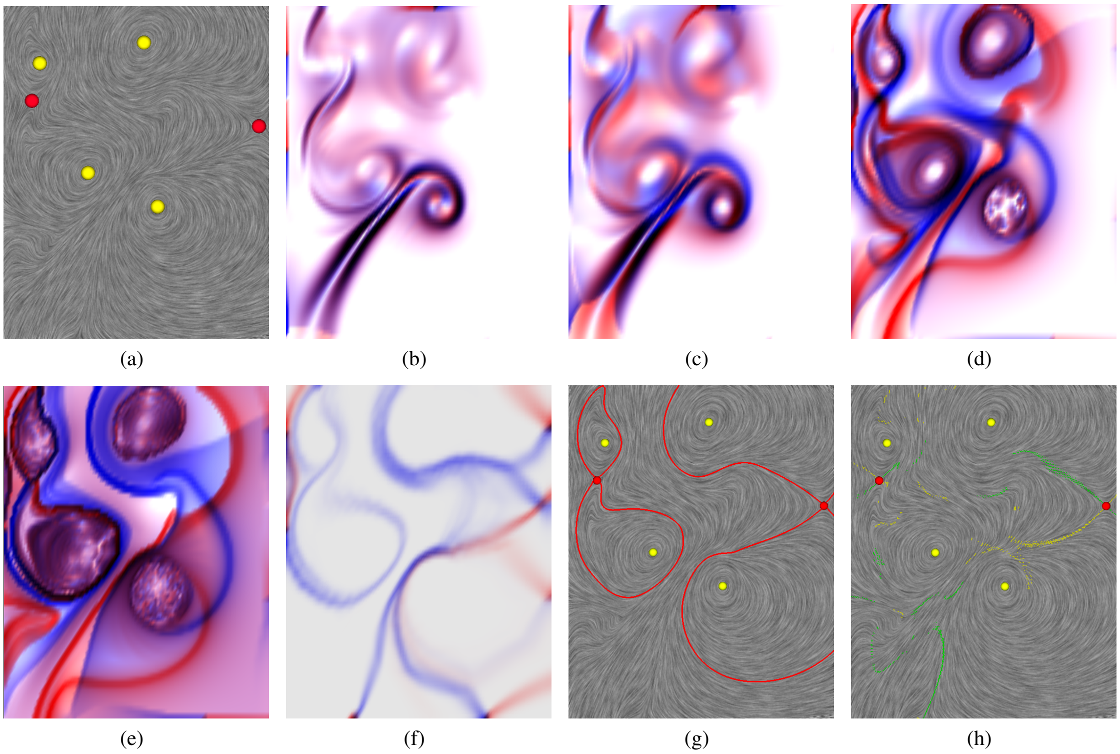
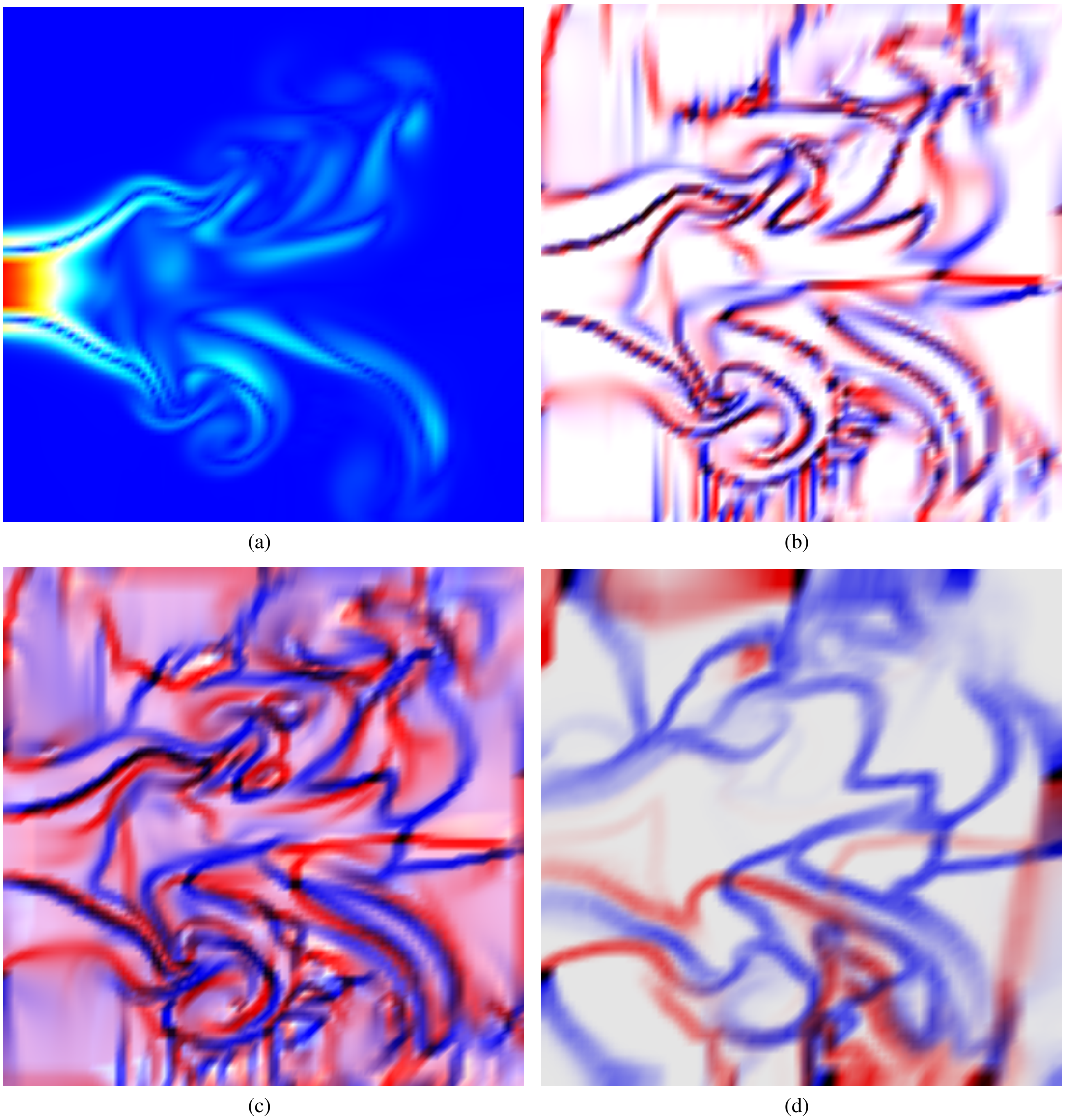


Fig. 7. A CFD-dataset simulating a fluid entering at the left bottom: (a) 4 center points (yellow) and 2 saddles (red), the left saddle joins two homoclinic orbits, (b)-(e) FTLE with increasing time step, (f) d_{unc+} (red) and d_{unc-} (blue), (g) toplogical skeleton (red) extracted by the method of Helman [8] (h) separation lines computed by the method of Kenwright [14].



(a)

(b)

(c)

(d)

Fig. 8. A swirling jet entering a fluid at rest from the left side: (a) the magnitude of velocity from zero (blue) to red (max), (b) FTLE for $t = 0.01$, (c) FTLE for $t = 0.3$, (d) d_{unc+} (red) and d_{unc-} (blue).

Acknowledgments

We like to thank Christian Heine for his help in optimizing our source code. We also thank the reviewers for their valuable hints.

References

[1] S. L. Brunton and C. W. Rowley. Fast computation of finite-time lyapunov exponent fields for unsteady flows. In *Chaos*, volume 20, 2010.

[2] G. Chen, K. Mischakow, R.S.Laramee, and E. Zhang. Efficient morse decompositions of vector fields. In *IEEE Transactions on Visualization and Computer Graphics*, volume 14, pages 848–862, 2008.

[3] M. Dellnitz and O. Junge. On the approximation of complicated dynamical behavior. In *SIAM Journal on Numerical Analysis*, volume 36, pages 491–515, 1999.

[4] R. Forman. Morse theory for cell complexes. In *Advances in Mathematics*, volume 134, pages 90–145, 1998.

[5] C. Garth, G.-S. Li, X. Tricoche, C.-D. Hansen, and H. Hagen. Visualization of coherent structures in transient 2d flows. In *Topology-Based Methods in Visualization II*, 2009.

[6] C. Garth, A. Wiebel, X. Tricoche, K. Joy, and G. Scheuermann. Lagrangian visualization of flow-embedded surface structures. In *Computer Graphics Forum*, volume 27, pages 767–774, 2008.

[7] G. Haller. Distinguished material surfaces and coherent structures in three-dimensional flows. In *Physica D*, volume 149, pages 248–277, 2001.

[8] J. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. In *IEEE Computer Graphics and Applications*, volume 11, pages 36–46, 1991.

[9] V. Hernandez, J.-E. Roman, and V. Vidal. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Transactions on Mathematical Software*, 31:351–362, 2005.

[10] M. Hirsch, S. Smale, and R. Devaney. *Differential Equations, Dynamical Systems and An Introduction to Chaos*. Elsevier, second edition, 2004.

[11] C. Johnson. Top scientific visualization research problems. In *IEEE Computer Graphics and Applications*, volume 24, pages 13–17, 2004.

[12] W. Kalies and H. Ban. A computational approach to conley's decomposition theorem. In *J. Computational and Non-Linear Dynamics*, volume 1, pages 312–319, 2006.

[13] J. Kasten, C. Petz, I. Hotz, B. Noack, and H.-C. Hege. In *Proceedings Vision, Modeling and Visualization 2008*, pages 265–274.

[14] D. N. Kenwright, C. Henze, and C. Levit. Feature extraction of separation and attachment lines. *IEEE Trans. Vis. Comput. Graph.*, 5:135–144, 1999.

[15] K. M. Mahrous, B. H. J. C. Bennett, and K. I. Joy. Improving topological segmentation of three-dimensional vector fields. In *IEEE TCVG Symposium on Visualization*, 2003.

[16] M. Mrozek and P. Zgliczynski. Set arithmetic and the enclosing problem in dynamics. In *Annales Polonici Mathematici*, pages 237–259, 2000.

[17] M. Otto, T. Germer, H.-C. Hege, and H. Theisel. Uncertain 2d vector field topology. In *Computer Graphics Forum*, volume 29, pages 347–356, 2010.

[18] J. Paixao, M. Lage, F. Petronetto, A. Laier, S. Pesco, G. Tavares, T. Lewiner, and H. Lopes. Random walks for vector field denoising. In *Computer Graphics and Image Processing (SIBGRAPI), 2009 XXII Brazilian Symposium on*, pages 112–119, 2009.

[19] A.-T. Pang, C.-M. Wittenbrink, and S.-K. Lodh. Approaches to uncertainty visualization. *The Visual Computer*, 13:370–390, 1996.

[20] R. Peikert and M. Roth. The parallel vector operator - a vector field visualization primitive. In *Proc. IEEE Visualization Conf.*, pages 263–270, 1999.

[21] F. Post, B. Vrolijk, H. Hauser, R. Laramee, and H. Doleisch. The state of art in flow visualization: Feature extraction and tracking. In *Computer Graphics Forum*, volume 22, pages 775–792, 2003.

[22] J. Reininghaus and I. Hotz. Combinatorial 2d vector field topology extraction and simplification. In *Topology in Visualization*, 2010.

[23] F. Sadlo and R. Peikert. Visualizing lagrangian coherent structures and comparison to vector field topology. In *Topology-Based Methods in Visualization II*, 2009.

[24] G. Scheuermann, B. Hamann, K. Joy, and W. Kollmann. Visualizing Local Vector Field Topology. *Journal of Electronic Imaging*, 9(4):356–367, 2000.

[25] D. Schneider, J. Fuhrmann, W. Reich, and G. Scheuermann. A variance based ftle-like method for unsteady uncertain vector fields. In *Topological Methods in Data Analysis and Visualization II*, pages 255–268, 2012.

[26] D. Stalling and H.-C. Hege. Fast and resolution independent line integral convolution. In *ACM SIGGRAPH*, pages 249–256, 1995.

[27] D.-W. Stroock. *An Introduction to Markov Processes*. Springer, 2005.

[28] L.-N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997.

[29] T. Weinkauf. *Extraction of Topological Structures in 2D and 3D Vector Fields*. PhD thesis, University Magdeburg, 2008.

[30] D. Weiskopf and B. Erlebacher. Overview of flow visualization. In *The Visualization Handbook*, pages 261–278, 2005.

[31] T. Wischgoll and G. Scheuermann. Detection and visualization of planar closed streamlines. In *IEEE Trans. Visualization and CG*, volume 7, pages 165–172, 2001.