OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

Topology of Vector Fields with Uncertainty

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von: Dipl. Inform. Mathias Otto
geb. am 02. Januar 1983 in Magdeburg

Gutachter: Prof. Dr. Holger Theisel
Prof. Dr. Ronald Peikert
Prof. Dr. Rüdiger Westermann

Magdeburg, den 10. April 2015

Ehrenerklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; verwendete fremde und eigene Quellen sind als solche kenntlich gemacht. Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert,
- fremde Forschungsergebnisse verzerrt wie dergegeben.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht Unterlassungs- und Schadensersatzansprüche des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Magdeburg, 13. Oktober 2014

Abstract

In this work we present methods for the extraction and visualization of important features in vector fields with uncertainty.

In the first part we introduce the first global approaches to extract and visualize topological features in 2D and 3D steady vector fields that contain uncertainty. For this, we extend the concept of vector field topology to uncertain vector fields by considering the vector field as density distribution functions. By generalizing the concepts of stream lines and critical points we obtain a number of density fields representing an uncertain topological segmentation. These features we can visualize in the 2D case as height surfaces and for the 3D case as volume renderings.

In a second part we present an approach that generalizes concepts of region-based and geometry-based vortex extraction, namely the Q-criterion, the λ_2-criterion and the Parallel Vectors Operator applied to the method by Sujudi and Haimes. All these computations are based on the Jacobian of the input field. We give a definition for the Jacobian computation in uncertain vector fields by using correlated distribution functions around the location we want to evaluate. After fitting the distribution functions we compute the vortex measures in a Monte Carlo Simulation. The final results are scalar fields representing the probabilities of vortex structures. They are visualized by isosurfaces and volume renderings.

The last part of this work treats time-dependent vector fields. For such a field topological features are defined as Lagrangian Coherent Structures (LCS) that can be computed by extracting extremal structures of the Finite Time Lyapunov Exponent (FTLE). They are close to material structures, and for an infinite integration time they converge to exact material structures. However, due to the finite integration time in FTLE, they are generally not exact material structures. We introduce a modification of the FTLE method which is guaranteed to produce separating material structures as features of a scalar field. We achieve this by incorporating the complete available integration time both in forward and backward direction, and by choosing an appropriate definition for separating structures.

All of our presented methods are tested with synthetic and real-world examples.

Kurzfassung

In der vorliegenden Arbeit zeigen wir Methoden für die Extraktion und Visualisierung wichtiger Strukturen in unsicherheitsbehafteten Vektorfeldern.

Im ersten Teil führen wir den ersten uns bekannten Ansatz für die Extraktion und Visualisierung von topologischen Strukturen in 2D und 3D unsicherheitsbehafteten Vektorfeldern ein. Hierfür erweitern wir das Konzept der Vektorfeldtopologie für unsicherheitsbehaftete Vektorfelder. Dabei wird das unsichere Vektorfeld als Feld von Verteilungsfunktionen dargestellt. Durch die Generalisierung der Konzepte von Stromlinien und kritischen Punkten bestimmen wir eine Vielzahl von Dichteverteilungen, welche die topologische Segmentierung darstellen. Diese Strukturen werden im 2D-Fall als Höhenfelder und im 3D-Fall als Volumerenderings dargestellt.

Im zweiten Teil zeigen wir einen Ansatz für die Extraktion von regionsbasierten und geometriebasierten Wirbelstrukturen in unsicheren Vektorfeldern, wie das Q-Kriterium, das λ_2-Kriterium und den Parallel Vectors Operator, angewandt auf das Verfahren von Sujudi und Haimes. Alle diese Berechnungen beruhen auf der Extraktion der Jakobimatrix des Vektorfeldes. Aus diesem Grund entwickeln wir eine Definition für die Berechnung der Jakobimatrix beruhend auf einer hochdimensionalen Verteilungsfunktion, welche die lokale Nachbarschaft mit einbezieht. Nach dem Erstellen der Verteilungsfunktionen berechnen wir die Wirbelmaße mit einer Monte Carlo Methode. Die finalen Resultate repräsentieren Wahrscheinlichkeiten für Wirbelstrukturen in Form von Skalarfeldern. Sie werden mit Isoflächen und Volumerenderings dargestellt.

Der letzte Teil dieser Arbeit befasst sich mit zeitabhängigen Vektorfeldern. Topologische Strukturen solcher Felder werden mittels Lagrangian Coherent Structures (LCS) definiert und können mit Hilfe von Extremalstrukturen des Finite Time Lyaponov Exponent (FTLE) berechnet werden. Diese Extremalstrukturen nähern sich Materialstrukturen an, für eine unendlich lange Integrationszeit konvergieren sie sogar zu Materialstrukturen. Bedingt durch eine endliche Integrationszeit des FTLE, sind sie allerdings keine exakten Materialstrukturen. Wir führen eine modifizierte FTLE Methode ein, die garantiert Materialstrukturen zu erzeugen in Form von Isoflächen in Skalarfeldern. Dies wird erreicht indem wir sowohl die Vorwärts- als auch die Rückwärtsintegration über die komplette gegebene Zeit mit einbeziehen.

Alle unsere Methoden werden anhand von synthetischen und realen Daten getestet.

Acknowledgments

This work would not be possible without the help of many people.

At first I want to thank Prof. Dr. Holger Theisel who provided me the possibility to work on this topic. He supported me always with his advice and gave me the right hints. Further I enjoyed working at the Institute of Simulation and Graphics and want to thank all people there making my daily work pleasant. Especially I want to thank my dear colleagues of the visual computing group Dr. Tobias Germer, Dr. Alexander Kuhn, Wito Engelke, Dr. Christian Rössl, Dr. Janick Martinez Esturo, Maik Schulze and Dr. Dirk Lehmann. They supported me every time with beneficial discussions. Working with Tobias, Alex and Wito on paper was a pleasure for me. Winning the IEEE Visualization Contest 2011 together with Alex and Wito was the climax. Next I want to thank Dr. Niklas Röber and PD Dr. Gabor Janiga for supplying me with data sets and discussions from a practical point of view. Also I want to thank my co-workers Prof. Ronald Peikert, Hans-Christian Hege and Mykhaylo Nykolaychuk. Last I thank Dirk Schrader for providing the North Sea data set.

Finally I want to say that all these people contributed to make one of my best times, resulting in this work.

Contents

1 Introduction

The analysis of flow fields plays a vital role in many computational fluid dynamics (CFD) applications. As the amount and complexity of CFD data rapidly increases, this raises a need for efficient and reliable analysis. Additionally, for many applications not only the amount and complexity is increasing, but also the possibilities how data can be acquired and analyzed for a specific case. For example, due to today's computational power, CFD simulations are repeated multiple times with slightly varying parameters. This introduces the need to manage and process multiple flow fields describing one and the same phenomenon. Each field has an associated reliability which can be captured by the concept of uncertainty. The consideration of uncertainty is one of the most relevant problems in visualization [Joh04]. A variety of methods has been introduced to represent uncertainty in scalar, vector, and tensor fields. These methods share a common factor of uncertainty as a local property of the field, i.e., it is obtained by a locally computed or measured process. Vector fields describing flow phenomena may also contain local uncertainty. Examples could include noise, measurement or simulation errors, uncertain simulation parameters, initial and boundary conditions or the inherent randomness due to turbulence. However, in flow fields this *uncertainty is transported* along the flow, yielding a global uncertainty. This is the point where this work starts. We present methods that we developed over the last years for the extraction and visualization of topological features in uncertain vector fields, vortices in uncertain vector fields and Lagrangian Coherent Structures with guaranteed material separation:

- Uncertain 2D Vector Field Topology [OGHT10]
- Uncertain Topology of 3D Vector Fields [OGT11b]
- Closed Stream Lines in Uncertain Vector Fields [OGT11a]
- Vortex Analysis in Uncertain Vector Fields [OT12]
- Lagrangian Coherent Structures with Guaranteed Material Separation [GOPT11]

1.1 Goals

The goal of this work is the visualization of features in steady 2D and 3D uncertain vector fields. Flow fields contain very complex information and the complexity increases if uncertainty occurs. So we choose methods that summarize the information inside the flow field, such as topology and vortex structures. To the best of our knowledge we are not aware of any technique that treats topology or vortices in vector fields with uncertainty.

In this work we present methods for the extraction of topological features in such uncertain fields. The challenges here are the modeling of the uncertain vector field, the efficient extraction of topological structures and an appropriate visualization of the high-dimensional results. In comparison to vector field topology without uncertainty, we cannot extract sharp structures such as points and curves, we compute density distribution functions of the incidence of topological features, namely sinks, sources, saddles, and closed orbits.

Concerning vortex structure we have the same challenges. For this the modeling of the uncertainty for the computation of the Jacobian is most important. We present methods which compute probabilities of region-based and geometry-based vortex structures. The key for these computations is the use of high-dimensional correlated distribution functions for the computation of the Jacobian of the uncertain vector field.

The last goal of this work addresses unsteady vector fields. Topological features in unsteady vector fields are Lagrangian Coherent Structures (LCS) with material separation. Such structures can be approximated with extremal structures of the Finite Time Lyapunov Exponent (FTLE). But this is only an approximation. In general there is a nonzero volumetric flux across these structures. We present a new way to compute LCS with guaranteed material separation.

1.2 Contributions

The main contribution of this work is that we introduce *topological analysis for 2D and 3D vector fields with uncertainty*. We achieved this by the following steps:

- We describe a vector field equipped with local uncertainty as a density distribution function.
- In order to analyze its asymptotic flow behavior, we integrate particle density functions describing the probability that a particle is at a certain location.
- For this setup we define the concepts of stream lines and critical points.
- The resulting uncertain topological segmentation consists of a number of density distribution functions which represent the probability that a particle starting from a particular location will end in an (uncertain) source or sink.
- We visualize the resulting segmentation as well as the uncertain critical points as height fields.

Also for *vortex structures* we extend known techniques to vector fields with uncertainty. In this field we achieved these contributions:

- We generalize the definition of vortex regions and cores to uncertain vector fields.
- We propose a model of uncertainty that uses correlations referring to the probability distributions in the neighborhood. This uncertainty model enables us to compute all vortex extractors by the same Monte Carlo method.

- Finally, we test our method on real-world data sets and discuss the results.

In the context of unsteady vector field visualization we contribute a *new variant of the visualization of separating structures*.

- We analyze the fact that FTLE ridges are not exact material structures and show the relevance of this problem on different examples.
- We introduce a modification of the FTLE method which is guaranteed to produce separating structures that are material structures. *Our Material Separation Field (MSF)* method is novel and produces material structures that separate regions of coherent flow behavior.
- We apply our method to three different test data sets and discuss the differences from FTLE methods.

1.3 Outline

This work is structured as follows:

- Chapter 2 gives an overview about the theoretical background of vector field topology, vortex detection and FTLE computation.
- Chapter 3 describes the related work concerning local and global visualization techniques of vector fields, methods for time dependent vector fields, and uncertainty visualization.
- Chapter 4 gives a definition for uncertain vector fields and shows some simple examples.
- Chapter 5 explains the extraction of topological features in uncertain 2D and 3D vector fields.
- Chapter 6 applies criteria for vortex detection to uncertain vector fields. We do this for region-based and geometry-based vortex criteria.
- Chapter 7 goes in a different direction. Here we develop a technique that extracts structures with guaranteed material separation in time dependent certain vector fields.
- Chapter 8 concludes this work.

2 Theoretical Background

This chapter gives an overview over those aspects of vector fields which we apply to uncertainty in the following chapters.

2.1 Vector Field Topology

Vector field topology of 2D and 3D vector fields considers critical points and separating curves or surfaces (we call them separatrices). This results in a visualization of a segmentation of the flow field into regions of similar flow behavior (Fig. 2.1). A 2D unsteady vector field is given by a map

$$\mathbf{v}(x,y,t) = \begin{pmatrix} u(x,y,t) \\ v(x,y,t) \end{pmatrix}$$

and a 3D unsteady vector field by

$$\mathbf{v}(x,y,z,t) = \begin{pmatrix} u(x,y,z,t) \\ v(x,y,z,t) \\ w(x,y,z,t) \end{pmatrix}.$$

Steady vector fields do not depend on the time parameter t, so we get a 2D steady vector field

$$\mathbf{v}(x,y) = \begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix}$$

and a 3D steady vector field

$$\mathbf{v}(x,y,z) = \begin{pmatrix} u(x,y,z) \\ v(x,y,z) \\ w(x,y,z) \end{pmatrix}.$$

The first derivative of a vector field is called Jacobian matrix $\mathbf{J}$.

Critical points are important to characterize these segments.

Definition 1 *A* critical point of a vector field $\mathbf{v}$ *is an isolated point* $\mathbf{x}$ *in the domain* $\mathbf{D}$ *where the vector* $\mathbf{v}(\mathbf{x})$ *vanishes.*

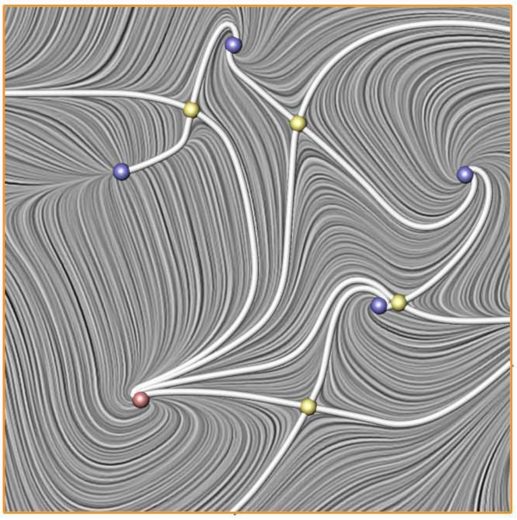


Figure 2.1: Example for topology of a 2D vector field (source: [Wei08]). Red dots are sources, blue dots are sinks, yellow dots are saddle points and white lines are separatrices.

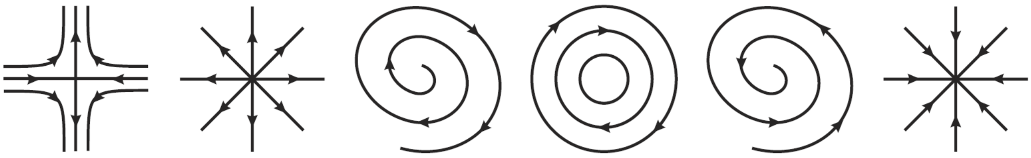


Figure 2.2: Critical points of a 2D vector field (from left to right): saddle, source node, source focus, center, sink focus, sink node. (source: Helman et al. [HH89b])

A 2D critical point with a full rank Jacobian matrix **J** (simple critical point) is classified by the real and imaginary parts of the eigenvalues λ_1 and λ_2 of **J** of the 2D vector field **v** at the location of the critical point:

$$\begin{aligned}
\text{Source Node :} &\quad 0 < Re(\lambda_1) \leq Re(\lambda_2) \text{ and } Im(\lambda_1) = Im(\lambda_2) = 0 \\
\text{Sink Node :} &\quad Re(\lambda_1) \leq Re(\lambda_2) < 0 \text{ and } Im(\lambda_1) = Im(\lambda_2) = 0 \\
\text{Source Focus :} &\quad 0 < Re(\lambda_1) \leq Re(\lambda_2) \text{ and } Im(\lambda_1) = -Im(\lambda_2) \neq 0 \\
\text{Sink Focus :} &\quad Re(\lambda_1) \leq Re(\lambda_2) < 0 \text{ and } Im(\lambda_1) = -Im(\lambda_2) \neq 0 \\
\text{Saddle :} &\quad Re(\lambda_1) < 0 < Re(\lambda_2) \text{ and } Im(\lambda_1) = Im(\lambda_2) = 0 \\
\text{Center :} &\quad Re(\lambda_1) = Re(\lambda_2) = 0 \text{ and } Im(\lambda_1) = -Im(\lambda_2) \neq 0
\end{aligned}$$

The classes above are depicted in Fig. 2.2. In a similar way a 3D critical point is classified by the eigenvalues λ_1, λ_2, and λ_3 of Jacobian matrix **J** of the 3D vector field **v** at the location of the critical point (if **J** has full rank):

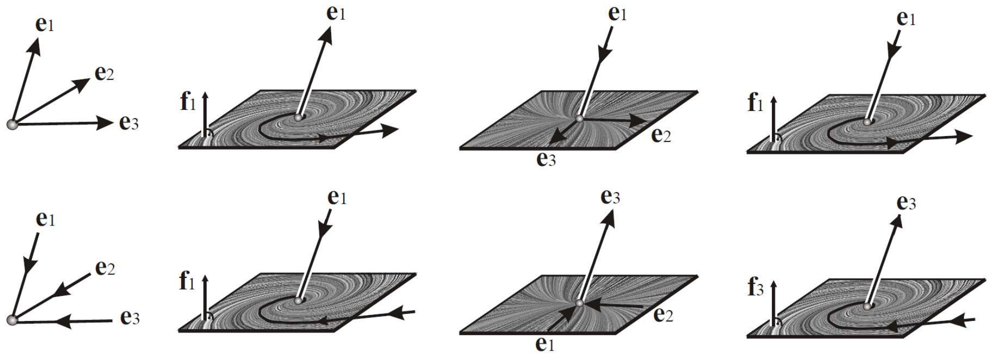


Figure 2.3: Critical points of a 3D vector field (top row from left to right): repelling node, repelling focus, repelling saddle node and repelling saddle focus, (bottom row from left to right): attracting node, attracting focus, attracting saddle node and attracting saddle focus (source: Theisel et al. [TWHS03]).

with three real eigenvalues

Repelling Node :	$0 < Re(\lambda_1) \leq Re(\lambda_2) \leq Re(\lambda_3)$
Repelling Node Saddle :	$Re(\lambda_1) < 0 < Re(\lambda_2) \leq Re(\lambda_3)$
Attracting Node Saddle :	$Re(\lambda_1) < Re(\lambda_2) < 0 < Re(\lambda_3)$
Attracting Node :	$Re(\lambda_1) \leq Re(\lambda_2) \leq Re(\lambda_3) < 0$

with one real eigenvalue λ_1 and two complex conjugated eigenvalues λ_2 and λ_3

Repelling Focus :	all real parts are positive
Repelling Focus Saddle :	one negative real part and two positive real parts
Attracting Focus Saddle :	one positive real part and two negative real parts
Attracting Focus :	all real parts are negative

Attracting nodes and foci are summarized as sinks and repelling nodes and foci as sources. Repelling saddles have one direction of inflow and a 2D plane with outflow behavior, while attracting saddles have one direction of outflow and a 2D plane with inflow behavior [TWHS03]. The 2D plane of node saddles is described by the second and third eigenvectors of **J** and the directional flow by the first eigenvector of **J**. For focus saddles the 2D plane is described by the only real eigenvector of $\mathbf{J}^T$ acting as normal of the 2D plane and the directional flow by the only real eigenvector of **J**.

To visualize the topology of a vector field, we need in addition to critical points special stream lines and stream surfaces called separatrices.

Definition 2 *A* stream line *is a differentiable curve in domain of the vector field* **v** *with the property that the tangent of the curve in every pointis identical to the vector of vector field at that point.*

Considering a velocity field, a stream line can be interpreted by the path of a massless particle within the velocity field. A stream line can be obtained by integration of the vector field.

Definition 3 *A* stream surface *is a 2D manifold in the domain of the vector field* $\mathbf{v}$ *with the property that every point on the surface have a surface normal orthogonal to the vector of the vector field* $\mathbf{v}$ *at this point.*

A stream surface is computed by starting stream lines on a given curve in the domain of the vector field. With these definitions we are able to define separatrices in 2D and 3D. Separatrices act as separators of different flow behavior.

Definition 4 *In 2D vector fields* separatrices *are stream lines converging to a saddle point.*

Practically separatrices of 2D vector fields are extracted by seeding four stream lines in epsilon distance of the saddle. The seeding points are located on the eigenvectors of the Jacobian matrix $\mathbf{J}$ at the location of the saddle point. Using forward integration they end in a sink or another saddle, using backward integration they end in a source or a another saddle. In 3D vector fields *separatrices* appear as 1D manifolds and 2D manifolds.

Definition 5 *A 2D separatrix is a stream surface that includes the 2D plane of a saddle point.*

1D separatrices in 3D vector fields are a special case. In opposite to all other stream lines in its neighborhood a 1D separatrix converges to a saddle point.

Visualizing 2D separatrices in 3D vector fields tend to visual clutter with increasing number of separation surfaces. An approach to reduce the visual complexity is the usage of saddle connectors [TWHS03] and boundary switch connectors [WTHS04].

Definition 6 *Let* $\mathbf{v}$ *be a 3D vector field, and let* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *be two saddle points in* $\mathbf{v}$*. We consider the intersection of the two 2D separatrices starting in the outflow/inflow planes of* $\mathbf{x}_1$ *and* $\mathbf{x}_2$*. If this intersection is a curve, we call it a* saddle connector.

Saddle connectors, as shown in Fig. 2.4, illustrate the essential skeleton of 2D separatrices. Assuming we have a limited domain with outflow and inflow at the boundary.

Definition 7 Boundary switch curves *consist of all points on the boundary where the flow direction is tangential to the boundary surface.*

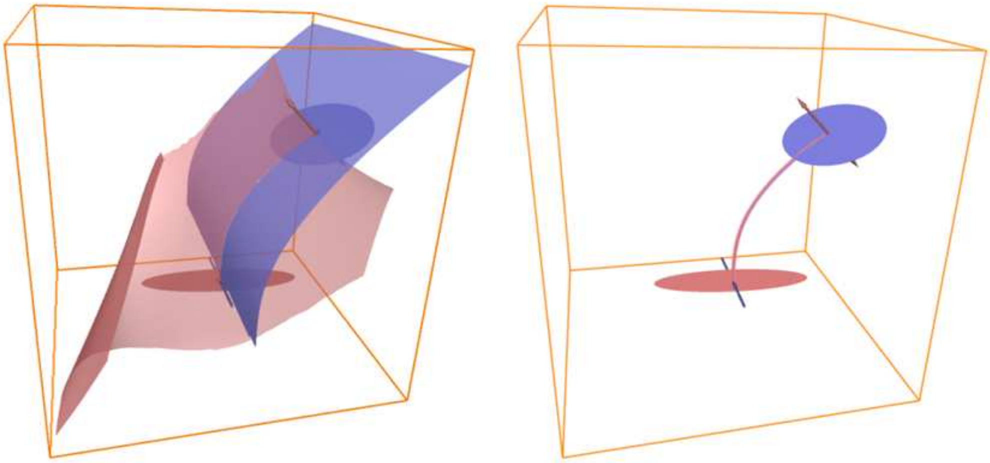


Figure 2.4: Example for a saddle connector. Left image shows two saddle points and their 2D separatrices. Right image shows the intersection curve, representing the saddle connector. (source: Theisel et al. [TWHS03])

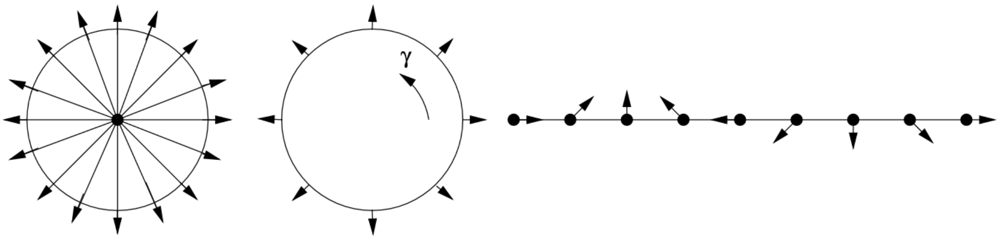


Figure 2.5: Example of index computation. The vector field along γ makes a counterclockwise turn, resulting in an index of +1. (image source [SHK$^+$97])

The inflow and outflow behavior change at boundary switch curves. By starting a stream surface from a boundary switch curve, we get another class of separatrix, dividing in- and outflow behavior. If the integration of such a surface is started at a boundary switch curve it will end in the same boundary switch curve, or in another boundary switch curve, or in a saddle. Intersection curves of these surfaces with 2D separatrices starting from saddles are call *boundary switch connectors*. Saddle connectors and boundary switch connectors are visualized as ribbons of the intersecting 2D separatrices close to the saddle connector or boundary switch connector curve.

Vector field topology does not only considers simple critical points with full rank Jacobian matrix. An approach to classify simple and higher order critical points uses the Poincaré-Hopf index [SHK$^+$97]. The classification incorporates the vectors on a curve γ around the critical point as shown in Fig. 2.5.

$$\mathsf{ind}_{\mathbf{x}}\mathbf{v} = \mathsf{ind}_{\gamma}\mathbf{v} = \frac{1}{2\pi}\int_{\gamma}\frac{\mathbf{v}_1 d\mathbf{v}_2 - \mathbf{v}_2 d\mathbf{v}_1}{\mathbf{v}_1^2 + \mathbf{v}_2^2}$$

The curve must cover a close neighborhood of the critical point so that no other critical point is inside the curve. Having simple critical points with full rank Jacobian matrix no singularity get an index of 0, sinks and sources an index of +1, and saddle points an index of -1.

A special type of stream line is a *closed orbit*. A closed orbit is a stream line that is started at point $\mathbf{x}_0$ and returns to point $\mathbf{x}_0$ after a certain integration time. An isolated closed orbit can act as sink, as source (2D and 3D), or saddle (3D only). In the special case of divergence free vector fields all stream lines that do not intersect with the boundary are closed orbits.

2.2 Vortex Detection

The analysis of vortices in flow data sets is common in many engineering disciplines, e.g. aeronautics and automotive. So far there is no unique definition what a vortex is. However a number of vortex criteria have been defined in literature (see section 3.2). There are two main classes of vortex criteria: region-based criteria and vortex core lines. Here we will explain the criteria we will use in our work.

Two of the most common vortex region detectors are the λ_2 criterion[JH95] and the Q criterion[HWM88]. Both criteria make use of the decomposition of the Jacobian matrix $\mathbf{J}$ in a symmetric part

$$\mathbf{S} = \frac{1}{2}\left(\mathbf{J} + \mathbf{J}^T\right)$$

and an antisymmetric part

$$\Omega = \frac{1}{2}\left(\mathbf{J} - \mathbf{J}^T\right).$$

The λ_2 criterion is defined by the second largest eigenvalue of the matrix

$$S^2 + \Omega^2$$

and the Q criterion by

$$\frac{1}{2}\left(\|\mathbf{S}\|^2 + \|\Omega\|^2\right).$$

One wide spread method to compute vortex core lines is the Parallel Vectors operator [RP98]. The general approach is to find two derived vector fields of the vector field $\mathbf{v}$ that are parallel where a vortex core line exists. The Parallel Vectors operator is able to express a lot of vortex core line criteria [PR99]. We will consider the method proposed by Sujudi and Haimes [SH95]. This method searches points of zero curvature of stream lines and rotary behavior (imaginary eigenvalues of the $\mathbf{J}$). It can be computed by using the Parallel Vectors operator of the vector field $\mathbf{v}$ and its acceleration field $\mathbf{a}$, because zero curvature exists where the acceleration field $\mathbf{a} = \mathbf{J}\mathbf{v}$ is parallel to the original vector field $\mathbf{a} \parallel \mathbf{v}$.

2.3 Finite Time Lyapunov Exponent

The Finite Time Lyapunov Exponent (FTLE) is used in flow visualization to approximate Lagrangian Coherent Structures (LCS) [Hal01]. LCS encode features of unsteady vector fields, such as flow transport barriers, and delineate regions of similar flow behavior [PD10]. We describe the basic computation of FTLE in this section. References to approaches using FTLE or similar approaches are named in section 3.3.

FTLE is based on the gradient of the flow map ϕ. The Flow map ϕ is described by a mapping of the start position of a massless particle in a velocity field $\mathbf{v}$ (vector field) at time t to its end position after the time interval T. Given the gradient

$$\nabla = \nabla(\mathbf{x}, t, T) = \frac{d\phi(\mathbf{x}, t, T)}{d\mathbf{x}}$$

of ϕ, the FTLE is computed as follows:

$$\mathrm{FTLE}(\mathbf{x}, t, T) = \frac{1}{T} \ln \sqrt{\lambda_{max}(\nabla^* \cdot \nabla)}$$

with λ_{max} as largest eigenvalues of $\nabla^* \cdot \nabla$. Computing FTLE on a forward integrated flow map, the resulting scalar values represent the rate of separation of two particles. Height ridges of FTLE fields do not be perfect material separation curves or surfaces, but with increasing integration time of the FTLE field they converge to material separation curves or surfaces.

3 Related Work

In this chapter we give an overview of work related to the topics of global vector field visualization, time dependent vector field visualization, the visualization of vortical structures, and uncertainty visualization. We start with the main topic of our work global visualization of vector fields. The goal is to give an insight of proposed methods that we generalize in our work to vector fields with uncertainty.

3.1 Vector Field Topology

Global approaches for vector field visualization incorporate the transfer in vector fields. This is realized by integration based techniques. Global features include for examples stream lines and stream surfaces. A survey about these topics can be found in McLoughlin et al. [MLP$^+$10]. In this work we will concentrate on topological methods.

Vector field topology has been used as a standard tool for visualizing *2D vector fields*. It was introduced by Helman and Hesselink [HH89a]. After that a significant amount of research has been done in the field [PVH$^+$03, LHZP05]. Its main goal is a simple to understand visualization of a complex flow field.

Hence, immediately after the introduction of 2D topological methods, 3D methods have been proposed as visualization approaches [HH91, GLL91, Asi93, CPC90, PS97]. However, different technical, perceptional and theoretical reasons [TWHS07] hindered 3D topological methods from being as common as visualization tool as 2D methods. In fact, they were restricted to data sets with a rather low topological complexity [HH91, GLL91, LDG98]. A number of technical [Hul92, Gel01, SBM$^+$01, vW93] and conceptual [MBHJ03, MBS$^+$04, TWHS03, WTHS04] improvements were necessary to make 3D topological methods applicable as standard tools. Two of the conceptual discoveries, namely saddle connectors [TWHS03] and boundary switch connectors [WTHS04] play an essential role for our work. Saddle and boundary switch connectors decrease the visual complexity of the visualization. In our work critical points are not comparable to simple critical points. Due to uncertainty they can become higher order critical points. For vector fields without uncertainty an analysis of higher order critical points is given by Weinkauf et al.[WTS$^+$05]. In our work we also treat a special type of stream lines, named *closed orbit*. A closed orbit can be extracted by tracking of stream lines [WS01, WSH01, WS02]. These methods work on the cells of the data grid by finding cell cycles and analyzing stream lines of their boundary vertices. An alternative grid-independent approach to extract 2D closed stream line uses a transformation of the 2D vector field into an appropriated

3D vector field where intersections of certain stream surfaces represent the closed orbit [TWHS04].

3.2 Vortex Visualization

Vortex-related features play a central role in the analysis of flow fields. This is reflected by the amount of literature and approaches existing on this subject. Among CFD applications a set of common standard tools has been settled. Those methods can be grouped into two main categories:

- *Region-based* vortex criteria define a characteristic scalar field in which closed subsets define distinct regions of vortical behavior. Regarding this, there are the Q-criterion defined by Hunt et al. [HWM88] and the λ_2 criterion by Hussian et al.[JH95]. In addition to this, physically inspired definitions are given by analyzing vorticity magnitude (e.g., described by Zabusky et al.[ZBP$^+$91]), and helicity of the underlying flow field (e.g., as done by Degani et al. [DLS90]).
- *Vortex-core-line-based* approaches define binary criteria that describe the location of centers of vortical fluid motion. Such approaches are the Parallel Vectors (PV) operator described by Roth and Peikert [RP98], the Sujudi and Haimes vortex core extractor [SH95], and the local pressure minima method by Banks et al. [BS94]. In fact, most of the methods falling into the second category can be generalized by a Parallel Vectors description as shown by Peikert et al. [PR99]. Among those approaches Weinkauf et al. [WSTH07] presented a detector for centers of swirling particle motion, and Sahner et al. [SWH05] a Galilean invariant detection methodology. A scale-space approach is given by Bauer et al.[BP02].

One subclass of those approaches deals with tracking of core line features in order to describe the temporal evolution of vortex structures over time, e.g., Theisel et al. [TSW$^+$05]. All existing approaches on this topic are out of the scope of this work, we refer to the overview about core line based methods presented by Jiang et al. [JMT04].

Almost all methods considered in this section are developed for steady vector fields. The next section describes methods which are applicable for time dependent vector fields.

3.3 Time Dependent Vector Field Visualization Approaches

The extraction and visualization of Lagrangian Coherent Structures (LCS) has attracted scientists in CFD and visualization for almost a decade. One of the most prominent approaches for this is the computation of ridge structures in Finite Time Lyapunov Exponent (FTLE) fields, as introduced by Haller [Hal01, HY00]. FTLE ridges have been used for a variety of applications [LCM$^+$05, Hal02, SLP$^+$09, WPJ$^+$08]. Shadden et al. [SLM05] have shown that ridges of FTLE are approximate material structures, i.e., they converge to material structures for increasing integration times. This fact was used in [SW09] to extract topology-like structures and in [LM10] to accelerate the FTLE computation in 2D flows. Also in the visualization community, different approaches have been proposed to increase performance, accuracy and usefulness of FTLE as a visualization tool [SP07a, GLT$^+$07, GGTH07, SP07b, SRP09].

A crucial part of FTLE ridge extraction is the choice of a suitable ridge definition. Ridge definition and extraction is an active field of research in computer vision, computer graphics, and visualization. A variety of ridge definitions has been proposed in literature. We mention local conditions obtained by relaxing conditions of extremal structures [EGM$^+$94, Lin98], topological/watershed approaches [SWTH07], second derivative ridges [LM10] or definitions based in extremal curvature structures [OBS04]. [PS08, STS10] focus on the extraction of ridge surfaces in 3D fields. None of the ridge definitions mentioned above is sufficient to guarantee material separation.

Now we mentioned a wide set of vector field visualization methods. To complete our related topics we finally focus on methods that were proposed for the visualization of uncertainty.

3.4 Uncertainty Visualization

In general the notion of uncertainty has become increasingly popular in many computational applications. Uncertainty visualization adds value to understand the reliability of the displayed data better. In comparison to visualizations without uncertainty that show only mean values or most likely values, uncertainty visualization evaluates parameters of the underling error model.

Uncertainty is classified in different types: statistical, error, range [TK94]. In visualization every step of the visualization pipeline adds uncertainty to the resulting visualization [PWL96]:

Acquisition is the first stage to generate data sets by measuring or performing numerical simulation. In both cases an error is introduced. The error of measurements depends on the instruments that are used. With more measurements of the same phenomenon the confidence grows, but there will be a variance. Also for numerical models that are used for simulations an error exists caused by simplifications of the model in comparison to the real-world phenomenon and by parameters that are based on the knowledge of domain experts. Different parameters and numerical methods lead also to statistical variations in the results.

Transformation of raw data is usually done before visualization. During preprocessing the data is for example rescaled, resampled or derived data is generated from multiple data sources. Such methods modify the original data and are potential sources of uncertainty.

Interpolation and integration schemes have a big influence on the quality of the resulting visualization and also the uncertainty. Dependent on the method and its parameters uncertainty is added to the visualization pipeline. For example the impact on the results is immense if a nearest neighbor, a linear or higher order interpolation scheme is used. In our case, for flow visualization (e.g., stream lines and path lines) the integration step width and the seeding strategy are very important factors. Here the error is propagated and the error of interim result influences the error of the final result.

Rendering itself can also introduce uncertainty. For example lighting impacts the perception of the user. Dependent of the used techniques different results are achieved. For example using ray tracing with different ray seeding strategies leads to slightly different looking results. Nearly every parameter or processing during the rendering step leads to uncertainty in the visualization.

For the concept of uncertainty a number of visualization approaches was developed. A general overview of previous uncertain data analysis is presented by Griethe et al.[GS06]. In the following sections we will focus on recent work which inspired our work or is closely related to our work.

3.4.1 Uncertainty in Scalar Fields

In this section we will go into more details on recent work of uncertainty in scalar fields that are related to our approaches for vector fields. These approaches rely on similar basics that we use for our methods presented in this work.

An approach for scalar fields that is methodically similar to this work (chapter 6) extracts isosurfaces of uncertain scalar fields by using correlated Gaussian distributions of scalar values [PH11, PWH11]. This approach was extended to generic distribution functions [PH13].

Information about structural variability of isosurfaces can be shown by a combined visualization using an isosurface of the mean scalar field and showing glyphs of anisotropic correlation structures in the vicinity of the uncertain isosurfaces [PW12].

Visualizing spatial cumulative distribution function (CDF) and the spatial probability density function (PDF) of uncertain iso-contours give information about the topological structure of the data set and the gradient strength at uncertain iso-contours [PW13].

A mathematical formulation of the gradient of discrete uncertain scalar fields was discovered in [PMW13]. It uses multivariate Gaussian distributions to model the uncertainty, resulting in probability distributions for the orientation and the magnitude of the gradient. This formulation can also be applied to uncertain vector fields.

Another topic in uncertain scalar fields is the visualization of critical points. An approach to visualize them uses indicators for the positional stability and the type stability [MW14]. The indicators represent the possibility of critical points in the area of the grid point. The positional stability shows the probability for a critical point in general. The type stability represents the probability that a critical point can be present at one location as different types. Types is defined as a classification of the critical point based on its Jacobian and Hessian matrix (second derivative).

3.4.2 Local Uncertainty in Vector Fields

Local visualization approaches of uncertain vector fields describe uncertainty as an impact that can be evaluated at a point inside the vector field, without considering its long-term integral behavior. One of the first approaches by Wittenbrink et al. [WPL96] uses glyphs mapping the uncertainty to the width of arrows or the color. Visually like LIC approaches, e.g., Sanderson et al. [SJK04] describe patterns of uncertainty using a reaction-diffusion model.

Botchen et al. [BWE05] introduce a semi-Lagrangian texture-based visualization technique representing local reliabilities by cross advection perpendicular to the flow direction and error diffusion. They show a short time flow behavior using streak lines. Uncertainty becomes visible as a blur effect on these streak lines. Additional user-adjusted color coding of uncertainty helps to explore the data set interactively [BWE06].

The work of Botchen et al. was applied to bidirectional vector fields in Zuk et al. [ZDG$^+$08]. Additionally a glyph showing the uncertainty in orientation was introduced.

Another glyph-based approach for unsteady vector fields symbolizes the local vectors over the complete time span [Hla1]. These glyphs are also applied to uncertain vector fields.

A very similar approach to our work was proposed in Petz et al. [PPH12]. This method extracts local features in uncertain vector fields, for example, critical points and vortices. Methodically it is similar to our work in chapter 6, but we focused on vortex extraction in uncertain vector fields.

3.4.3 Global Uncertainty in Vector Fields

In contrast, global approaches propagate the information how reliable a certain velocity information is within the uncertain flow field. The following papers are related to time-dependent Lagrangian features.

An approach similar to FTLE is proposed by Schneider et al. [SFRS11]. It is called finite time variance analysis. This method is based on a stochastic flow map. The difference to FTLE is the usage of the covariance matrix of the stochastic flow map instead of the Cauchy-Green tensor. For the creation of the stochastic flow map Schneider et al. assume a vector field with constant error and independence of each data point. The results of this technique are similar to a FTLE field. Indeed it measures the separation in a vector field over time, too, but it does not show a qualitative result of the reliability of the separation. With higher uncertainty, structures get only less sharp. Schneider et al. state that a model using constant error representing the uncertainty in vector fields is better than a Gaussian distribution, because Gaussian distributions have a nonzero probability for an unbounded movement of a particle. On the other hand in our observations physical phenomena are much more complex than the constant error bound model (see examples in chapter 6.1).

More insight about the stability of Lagrangian transport is visualized by an approach showing common and contradictory trends of Lagrangian flow behavior in CFD ensembles [HOGJ13]. In our work we interpret ensembles as input fields describing uncertainty, while Hummel et al. [HOGJ13] give a comprehensive visualization of the ensembles. Firstly, a Lagrangian neighborhood for ensembles is defined. Path lines of particles in the Lagrangian neighborhood are evaluated using a principal component analysis (PCA) resulting in individual variance given by each ensemble member and the joint variance incorporating all ensemble members. The key idea of this paper is a classification space containing all four variants of high and low average individual and joint variance. Trends in the Lagrangian transport become visible, using linking and brushing between a plot of individual and joint variances, and a visualization of the vector field.

A completely different visualization method uses direct volume rendering of high-resolution 3D visitation maps for large numbers of trajectories [BFMW12]. A visitation map counts for every cell the number of trajectories passing through this cell. Isosurfaces of a visitation map show frequently used paths.

3.4.4 Uncertainty in Tensor Fields

Tensor visualization plays also a role in uncertainty visualization. A big application field is the medical visualization of brain diffusion magnetic resonance imaging (MRI) data that mainly visualizes fiber and nerve bundles. Such fiber bundles connect certain parts in the brain and are important to know before surgical procedures.

Schultz et al.[SVBK14] give an overview about current fiber tractography techniques. They argue for probabilistic fiber tractography techniques. Their main argument is those methods without handling uncertainty, like stream line tractography, underestimate the size of fiber bundles.

In a former work Schultz et al.[STS07] extract and visualize the topological skeleton of brain diffusion MRI data. The goal of this approach is to visualize so called p-faces. These faces enclose regions that connect two critical regions with a given probability. The sum of all p-faces creates the topology.

4 Vector Fields with Uncertainty

In this work we consider steady 2D and 3D vector fields. In classical vector field visualization such a vector field is described as a map from a 2D domain $\mathbf{D}$ into $\mathbb{R}^2$ as

$$\mathbf{v}_c(x,y) = \begin{pmatrix} u_c(x,y) \\ v_c(x,y) \end{pmatrix} \quad , \quad (x,y) \in \mathbf{D}. \tag{4.1}$$

and a 3D domain into $\mathbb{R}^3$ as

$$\mathbf{v}_c(x,y,z) = \begin{pmatrix} u_c(x,y,z) \\ v_c(x,y,z) \\ w_c(x,y,z) \end{pmatrix} \quad , \quad (x,y,z) \in \mathbf{D}. \tag{4.2}$$

In the following, we call this a *certain vector field*. If uncertainty comes into play, there is no unique vector assigned to a point (x,y) but rather a probability distribution of vectors, leading to the following definition:

Definition 8 *A steady 2D* uncertain vector field *over the domain* $\mathbf{D}$ *is a 4D scalar field* $\rho_v(x,y;u,v)$ *with*

- $(x,y) \in \mathbf{D}$ *and* $(u,v) \in \mathbb{R}^2$
- $\rho_v(x,y;u,v) \geq 0 \quad (x,y) \in \mathbf{D}$ *and* $(u,v) \in \mathbb{R}^2$
- $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \rho_v(x,y;u,v)\, du\, dv = 1$ *for all* $(x,y) \in \mathbf{D}$.

The value $\rho_v(x,y;u,v)\, du\, dv$ denotes the probability that at the location (x,y) the vector field has its value in the range $[u,u+du] \times [v,v+dv]$.

This definition is straight forward to extend to 3D uncertain vector fields:

Definition 9 *A steady 3D* uncertain vector field *over the domain* $\mathbf{D}$ *is a 6D scalar field* $\rho_v(x,y,z;u,v,w)$ *with*

- $(x,y,z) \in \mathbf{D}$ *and* $(u,v,w) \in \mathbb{R}^3$
- $\rho_v(x,y,z;u,v,w) \geq 0 \quad (x,y,z) \in \mathbf{D}$ *and* $(u,v,w) \in \mathbb{R}^3$
- $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \rho_v(x,y;u,v)\, du\, dv\, dw = 1$ *for all* $(x,y,z) \in \mathbf{D}$.

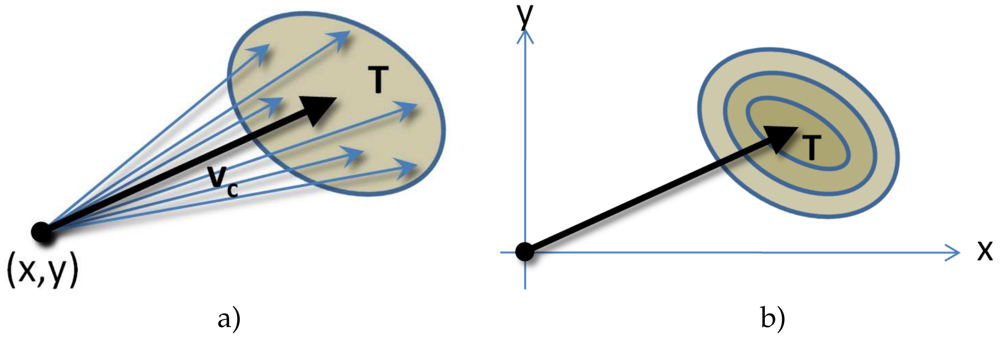


Figure 4.1: Gaussian distribution vector field; a) at a location (x,y) the distribution is defined by a vector $\mathbf{v}_c$ and a 2D tensor $\mathbf{T}$ which is visualized as an ellipse; b) isolines of the corresponding distribution field ρ_v.

4.1 Examples of Uncertain Vector Fields

A certain vector field as described in equation 4.1 is a special case of an uncertain vector field and described by the 2D Dirac delta distribution $\rho_v(x,y;u,v) = \delta(u - u_c(x,y), v - v_c(x,y))$. A *Gaussian distribution 2D vector field* is given by a 2D vector field $\mathbf{v}_c(x,y)$ and a 2×2 symmetric positive definite second order tensor field $\mathbf{T}(x,y)$. For a location (x,y) it assumes a Gaussian distribution of the vectors around the mean $\mathbf{v}_c$ with the standard deviation encoded in the covariance matrix $\mathbf{T}$:

$$\rho_v(x,y;u,v) = \frac{1}{2\pi\sqrt{\det(\mathbf{T})}}\; e^{-\frac{1}{2}(\mathbf{v}-\mathbf{v}_c)^T\,\mathbf{T}^{-1}\,(\mathbf{v}-\mathbf{v}_c)} \tag{4.3}$$

with $\mathbf{T} = \mathbf{T}(x,y)$, $\mathbf{v} = (u,v)^T$. Fig. 4.1 gives an illustration.

4.2 Obtaining Uncertain Vector Fields

We determine uncertainty from given data similarly to the approach proposed in [BWE05] that measures a flow phenomenon multiple times. This way our input are N vector fields over the same domain which vary due to local uncertainty. For these N samples we determine parameters of a bivariate Gaussian distribution, i.e., we obtain a mean vector field $\mathbf{v}_c$ and a tensor field of covariance matrices $\mathbf{T}$ and can hence model ρ_v. The choice of the Gaussian distribution is the most common empirical distribution of errors. It provides a fair balance of good approximation properties and efficient numerical computation. Contrary to [BWE05], we model the local uncertainty not only orthogonal to the flow direction but in every direction. This ensures stability in areas of slow flow.

5 Topology of Uncertain Vector Fields

Based on the definition of an uncertain vector field we explain the extraction of topology. Our novel approach generalizes the known topology of certain vector fields to uncertain vector fields. We start with a fundamental technique for the extraction, the integration of stream line like structures in a probabilistic way. This technique is used to compute 2D and 3D critical distributions which are the equivalent for critical points in certain vector fields. Furthermore the probabilistic stream line integration is needed to find saddle connector like and boundary connector like structures in 3D uncertain vector fields. First we explain our concepts for 2D uncertain vector fields. If needed, we give after each section a separate extension to 3D uncertain vector fields. At the end of this chapter we demonstrate our proposed methods on a number of synthetic and real-world examples.

5.1 2D Stream Line Integration

Classical stream line integration is not defined for uncertain vector fields as there is no unique vector at a location (x,y). To illustrate this, we consider a Gaussian distribution field (4.3) with

$$\mathbf{v}_c(x,y) = -0.4 \begin{pmatrix} x \\ y \end{pmatrix} \;,\; \mathbf{T}(x,y) = \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix} \tag{5.1}$$

and start an "uncertain Euler integration" of a particle at $(x,y) = (-9.5,0)$ in the following way: at the position (x_i,y_i), a vector $\mathbf{v}(x_i,y_i)$ is randomly chosen by evaluating the distribution function $\rho_v(x_i,y_i;u,v)$. Fig. 5.1 shows the integration of 2 different particles. Note that the particles follow different paths but have a similar

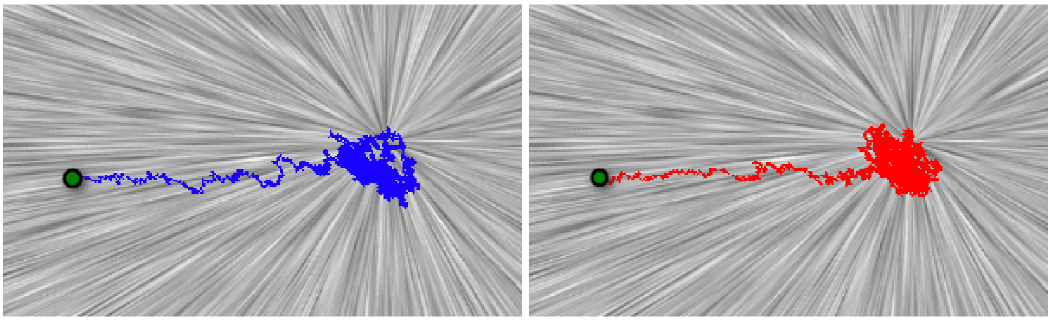


Figure 5.1: Integration of 2 particles in the Gaussian distribution field (5.1) starting from the same location.

global behavior: they first move towards the point $(0,0)$, then (due to the uncertainty) move randomly around it without leaving its neighborhood. This behavior does not depend on the step size of the integration: changing the step size effects the shapes and complexity of the curves but not their global behavior. Also note that the particles produce an ellipse shaped path pattern corresponding to **T**.

This example shows that for uncertain vector fields we have to advect *particle density functions* instead of particular particles.

Definition 10 *A* particle density function *over the domain* **D** *is a 2D time–dependent scalar field $p(x,y;t)$ with $(x,y) \in \mathbf{D}$, $t \in \mathbb{R}^+$ and*

- $p(x,y;t) \geq 0$ *for all* $(x,y) \in \mathbf{D}$ *and* $t \geq 0$
- $\int\int_{\mathbf{D}} p(x,y;t)\, dx\, dy \leq 1$ *for all* $t \geq 0$.
  (We use $\leq$ instead of $=$ because particles may leave the domain during integration.)

The value $p(x,y;t)\, dx\, dy$ denotes the ratio of particles in $[x, x+dx] \times [y, y+dy]$ in relation to the initial number of particles in **D** at $t = 0$.

Now we consider the spatial transport of particle densities by an uncertain vector field ρ_v. In order to avoid boundary effects we consider the infinite domain $\mathbf{D} = \mathbb{R}^2$. The densities are represented by virtual particles without inertia that are transported by the uncertain vector field ρ_v. Let Δt be a time interval that is short enough that the virtual particles can be considered to move on a straight line, i.e., in a constant vector field, during this interval. Then at time $t + \Delta t$ the number of particles in an infinitesimal volume $dx\, dy$ at some location (x,y) is the sum of the numbers of particles in cells $dr\, ds$ at all locations (r,s) times the probabilities that they are transported from (r,s) to (x,y) in time Δt, i.e., they experience a velocity $((x - r)/\Delta t, (y-s)/\Delta t)^T$. These probabilities are given by $\rho_v(r,s; \frac{x-r}{\Delta t}, \frac{y-s}{\Delta t})\, d(\frac{x-r}{\Delta t})\, d(\frac{y-s}{\Delta t})$. After division by the cell volumes we have $dx\, dy = dr\, ds$. This yields the following expression for the transport of particle densities:

$$\begin{aligned} & p(x,y;t+\Delta t) \\ = \; & \int\int_{\mathbf{D}} p(r,s;t)\, \rho_v(r,s; \frac{x-r}{\Delta t}, \frac{y-s}{\Delta t})\, d(\frac{x-r}{\Delta t})\, d(\frac{y-s}{\Delta t}) \\ = \; & \frac{1}{\Delta t^2} \int\int_{\mathbf{D}} p(r,s;t)\, \rho_v(r,s; \frac{x-r}{\Delta t}, \frac{y-s}{\Delta t})\, dr\, ds \end{aligned} \qquad (5.2)$$

A formal definition of a stream line in an uncertain vector field is given by:

Definition 11 *Given an uncertain vector field $\rho_v(x,y\,;\,u,v)$, a* stream line *starting at the particle density function $p_0(x,y)$ is a time-dependent particle density function $p(x,y;t)$ with*

$$\begin{aligned} p(x,y;t_0) &= p_0(x,y) \\ \frac{\partial p(x,y;t)}{\partial t} &= \lim_{\Delta t \to 0} \frac{p(x,y;t+\Delta t) - p(x,y;t)}{\Delta t} \\ p(x,y;t+\Delta t) &= (5.2). \end{aligned}$$

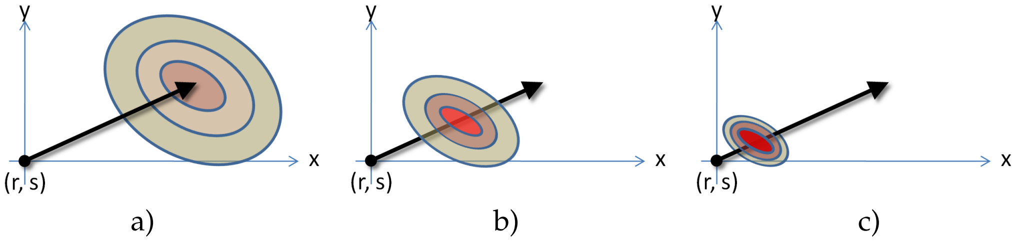


Figure 5.2: One Euler integration step of a particle density function initially concentrated at (r,s) with different step sizes Δt: a) $\Delta t = 1$, b) $\Delta t = 0.5$, c) $\Delta t = 0.25$; shown are the new particle density functions $p(x,y;t_i + \Delta t)$ as isolines.

Note that definition (11) defines a forward integration of p in ρ_v. A backward integration is obtained by a forward integration of p in $\rho_v(x,y;-u,-v)$. Definition (11) also leads to uniqueness of stream lines that are started at the same particle densitiy function $p_0(x,y)$. However, $\frac{\partial p}{\partial t}$ can generally be obtained only by numerical differentiation. As an example for particle density integration, imagine that at a time t_i all particles are at a location $(r,s) \in \mathbf{D}$, i.e., that $p(x,y;t_i) = \delta(x-r, y-s)$. Applying one step of an Euler integration with the step size Δt gives the new particle distribution function $p(x,y;t_{i+1})$ at the time $t_{i+1} = t_i + \Delta t$ as

$$p(x,y;t_{i+1}) = \frac{1}{\Delta t^2}\,\rho_v\left(r,s;\frac{x-r}{\Delta t},\frac{y-s}{\Delta t}\right).$$

Note that definition (10) guarantees that $\int\int_{\mathbf{D}} p(x,y;t_{i+1})\,dx\,dy = 1$ for any positive Δt if $\mathbf{D} = \mathbb{R}$. Fig. 5.2 illustrates one step of this Euler integration if ρ_v is a Gaussian distribution field: the smaller Δt, the closer the peak of $p(x,y;t_{i+1})$ comes to (r,s) and the higher this peak is.

5.2 3D Stream Line Integration

The principle for stream lines in 3D uncertain vector fields is very similar to the 2D case. We also consider particle distributions to track stream lines. For this we modified definition (10) into 3D space:

Definition 12 *A* particle density function *over the domain* $\mathbf{D}$ *is a 3D time–dependent scalar field* $p(x,y,z;t)$ *with* $(x,y,z) \in \mathbf{D}$, $t \in \mathbb{R}^+$ *and*

- $p(x,y,z;t) \geq 0$ *for all* $(x,y,z) \in \mathbf{D}$ *and* $t \geq 0$
- $\int\int\int_{\mathbf{D}} p(x,y,z;t)\,dx\,dy\,dz \leq 1$ *for all* $t \geq 0$.
(We use $\leq$ *instead of* $=$ *because particles may leave the domain during integration.)*

Such a particle distribution is advected within a 3D uncertain vector field. In the 2D case equation (5.2) described that advection. For the 3D case we can modify this equation as follows:

$$\begin{aligned}
& p(x,y,z;t+\Delta t) \\
= & \int\int\int_{\mathbf{D}} p(q,r,s;t)\,\rho_v(q,r,s;\tfrac{x-q}{\Delta t},\tfrac{x-r}{\Delta t},\tfrac{y-s}{\Delta t})\,d(\tfrac{x-q}{\Delta t})\,d(\tfrac{y-r}{\Delta t})\,d(\tfrac{z-s}{\Delta t}) \\
= & \frac{1}{\Delta t^3}\int\int\int_{\mathbf{D}} p(q,r,s;t)\,\rho_v(q,r,s;\tfrac{x-q}{\Delta t},\tfrac{y-r}{\Delta t},\tfrac{z-s}{\Delta t})\,dq\,dr\,ds \qquad (5.3)
\end{aligned}$$

Analogous to definition (11) a formal definition of a stream line in a 3D uncertain vector field is given by:

Definition 13 *Given an uncertain vector field $\rho_v(x,y,z;u,v,w)$, a* stream line *starting at the particle density function $p_0(x,y,z)$ is a time-dependent particle density function $p(x,y,z;t)$ with*

$$\begin{aligned}
p(x,y,z;t_0) &= p_0(x,y,z) \\
\frac{\partial p(x,y,z;t)}{\partial t} &= \lim_{\Delta t \to 0} \frac{p(x,y,z;t+\Delta t) - p(x,y,z;t)}{\Delta t} \\
p(x,y,z;t+\Delta t) &= (5.3).
\end{aligned}$$

$$p(x,y,z;t_{i+1}) = \frac{1}{\Delta t^3}\,\rho_v\left(q,r,s;\frac{x-q}{\Delta t},\frac{y-r}{\Delta t},\frac{z-s}{\Delta t}\right).$$

5.3 Stream Line Integration with Gaussian Distribution Functions

The stream line integration from definition 11 and definition 13 assume a constant step size Δt corresponding to the vector field distribution ρ_v. However, for the implementation we need different step sizes in order to minimize integration steps while maintaining stability. Unfortunately, different step sizes cannot be applied to equation (5.2), because (multiple) convolution of linearly scaled distribution functions does not yield the original distribution function. For example, two integration steps with $\frac{\Delta t}{2}$ would result in a different particle density function than a single integration step with Δt.

For the special case of Gaussian distribution functions we show how to scale the distribution functions for different step sizes appropriately. We assume a target step size of $\Delta t_s = \frac{\Delta t}{s}$, i.e., s integration steps of the Dirac delta should result in the original distribution from ρ_v. If we assume Gaussian distributions, our integration method can also be modeled by a stochastic differential equation describing the Brownian motion

$$\mathbf{X}(t) = x_0 + \int_0^t b(\mathbf{X}(r))dr + \int_0^t \mathbf{B}(\mathbf{X}(r))d\mathbf{W} \qquad (5.4)$$

with x_0 as initial point, $b(\mathbf{X}(r))$ as mean vector field, $\mathbf{B}(\mathbf{X}(r))$ as field of covariance matrices, and $\mathbf{W}$ as standard Wiener process. For the standard Wiener process there exists a heuristic $d\mathbf{W} \approx (dt)^{1/2}$ [Eva09]. In our discrete case this means $(dt)^{1/2} = s^{-1/2}$. Here we can confirm the correctness of this rule by considering the 1D Gaussian distribution function:

$$f(x) = \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{5.5}$$

We want to reproduce this function by convolving a Dirac delta $\delta(x)$ s-times with a scaled distribution function $g(x,s)$. In order to achieve this, we scale the standard deviation with $s^{-1/2}$ and the mean value with s^{-1}:

$$g(x,s) = \frac{1}{\sqrt{\frac{2\pi}{s}}\,\sigma} e^{-\frac{1}{2}\left(\frac{x-\frac{\mu}{s}}{\sqrt{\frac{1}{s}}\,\sigma}\right)^2}. \tag{5.6}$$

Starting with a Dirac delta

$$h_0(x) \quad = \quad \delta(x),$$

the convolution of h with $g(x,s)$ is described by

$$h_{k+1}(x) = \int_{-\infty}^{\infty} h_k(x) g(x-r,s) dr. \tag{5.7}$$

This series is described by the following function

$$r(x,k,s) = \frac{\sqrt{s}}{\sqrt{2k\pi\sigma}} e^{-\frac{1}{2k}\frac{(k\mu - xs)^2}{s\sigma^2}}. \tag{5.8}$$

Integrating $\delta(x)$ s-times corresponds convolving it s-times. Then, $k = s$ and $r(x,k,s) = f(x)$. Therefore, integrating $g(x,s)$ s-times results in the original distribution function $f(x)$. Thus, the mean value has to be scaled by s^{-1} and the standard deviation by s^{-1} in order to properly integrate with step size $s^{-1/2}$. This argumentation also holds for the 3D case, where the covariance matrix T (see equation (5.11)) has to be scaled by $s^{-1/2}$.

In the context of stochastic differential equations, our method can also be regarded as a random dynamical system [HK07]. Our search for uncertain sources and sinks is equivalent to the computation of random attractors [AO03]. In contrast, we also consider non-attracting structures like saddles and separating structures and give efficient methods to compute the uncertain topology.

5.4 2D Sink and Source Detection

Once the concept of stream lines is established, we can make a topological analysis of an uncertain vector field: starting from every point $(r,s) \in \mathbf{D}$ (i.e., from $p(x,y;t_0) = \delta(x-r, y-s)$) we start a stream line integration and observe its asymptotic behavior for $t \to \infty$ which generally converges to a *critical point distribution*:

Definition 14 *The particle density function $p_0(x,y)$ is a* critical point distribution *of ρ_v if for a stream line integration starting at $p(x,y;t_0)$ the condition $\frac{\partial p}{\partial t} = 0$ holds.*

In the following we use the terms critical point distribution and critical point synonymously whenever the context is clear.

Lemma 1 *Given are n critical points $p_1(x,y),...,p_n(x,y)$ of ρ_v. Then any positive linear combination of critical points is a critical point as well: $\sum_{i=1}^{n} \alpha_i\, p_i$ is a critical point for any $0 \leq \alpha_1,...,\alpha_n \leq 1$ and $\sum_{i=1}^{n} \alpha_i \leq 1$.*

Note that the linear combination is usually not convex, i.e., not $\sum_i \alpha_i = 1$, as part of the flow leaves the domain. Lemma 1 states that critical points of uncertain vector fields are not isolated but building a continuum of critical points. The proof follows directly from definition 11. In order to make a topological analysis, we have to find a finite number of linearly independent critical points such that every critical point can be represented as a positive linear combination of them.

Definition 15 *A critical point $p(x,y)$ is a* sink distribution *(or a stable critical point) of ρ_v if the forward integration of any small perturbation of p converges to p. The critical point $p(x,y)$ is a* source distribution *of ρ_v if the backward integration of any small perturbation of p converges to p. The critical point $p(x,y)$ is a* saddle distribution *of ρ_v if both forward and backward integration are unstable, i.e., diverge from p for a small perturbation of p.*

Note that sources and sinks play a different role than saddles for the definition of the topological skeleton of (certain) vector fields: sources and sinks define the number of different areas to be segmented (every segmented area corresponds to a pair of a source and a sink), while saddles are the starting points of the separating stream lines. Similar to this, we also focus on sources and sinks to define the segmentation of uncertain fields.

Definition 16 *Given an uncertain vector field $\rho_v(x,y;u,v)$, a set of sinks $(p_1(x,y),..., p_n(x,y))$ is called* spanning sink set *if $p_1,...,p_n$ are linearly independent and every sink $p(x,y)$ of ρ_v can be uniquely described as $p = \sum_{i=1}^{n} \alpha_i\, p_i$ with $0 \leq \alpha_1,...,\alpha_n \leq 1$ and $\sum_{i=1}^{n} \alpha_i \leq 1$. Similarly, a set of sources $(\tilde{p_1}(x,y),...,\tilde{p_m}(x,y))$ is called* spanning source set *if $\tilde{p_1},...,\tilde{p_m}$ are linearly independent and every source $\tilde{p}(x,y)$ of ρ_v can be uniquely described as $\tilde{p} = \sum_{i=1}^{m} \beta_i\, \tilde{p}_i$ with $0 \leq \beta_1,...,\beta_m \leq 1$ and $\sum_{i=1}^{m} \beta_i \leq 1$.*

Fig. 5.10a gives an illustration: the Gaussian distribution field (4.3) with

$$\begin{aligned} \mathbf{v}_c(x,y) &= \begin{pmatrix} -x\,(1-x)\,(1+x)\,(1-y^2) \,-\, x\,y^2 \\ y\,(1-y)\,(1+y)\,(1-x^2) \,+\, y\,x^2 \end{pmatrix} \\ \mathbf{T}(x,y) &= \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \end{aligned} \tag{5.9}$$

over the domain $[-1.5,1.5]^2$. There, the topological skeleton of $\mathbf{v}_c$ consists of 5 critical points: a saddle at $(0,0)$, two sources at $(-1,0)$ and $(1,0)$, and two sinks at $(0,-1)$ and $(0,1)$ (see the underlying LIC image in Fig. 5.3). For the uncertain

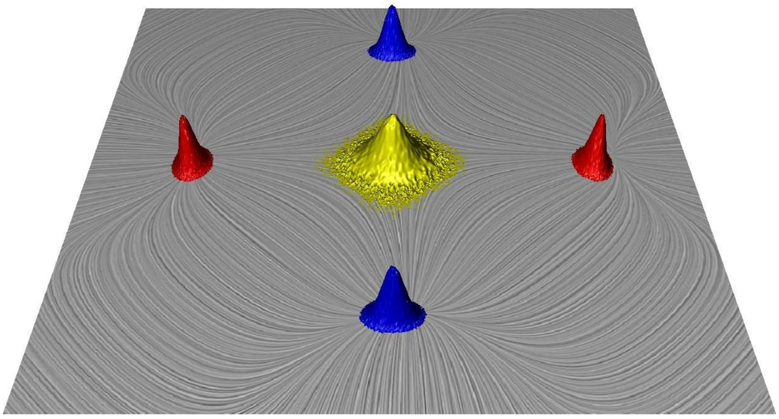


Figure 5.3: Example field (5.9): sink (blue) and source (red) distributions of ρ_v, represented as height maps.

case, the spanning sink set consists of two sinks p_1, p_2 which are shown as blue height fields in Fig. 5.3. The spanning source set consists of two sources $\tilde{p_1}, \tilde{p_2}$, shown as red height fields in Fig. 5.10. Note that $p_1, p_2, \tilde{p_1}, \tilde{p_2}, \hat{p_1}$ are almost zero in most regions except rather narrow peak regions. We draw their height fields only if they exceed a certain $\epsilon > 0$. Here we have chosen $\epsilon = 0.01$. This allows to show multiple height surfaces without visual clutter. Note that there is a one-to-one relation between the critical points of $\mathbf{v}_c$ and the spanning source set of ρ_v, as long as $||\mathbf{T}||_F$ is small enough, i.e., ρ_v is dominated by $\mathbf{v}_c$.

Once a spanning sink set $(p_1, ..., p_n)$ is found, every sink p can be described as $p = \sum_{i=1}^{n} \alpha_i \, p_i$. We call $(\alpha_1, ..., \alpha_n)$ the *coordinates of* p with respect to the spanning sink set $(p_1, ..., p_n)$. In a similar way we define $(\beta_1, ..., \beta_m)$ as the coordinates of a source $\tilde{p}$ with respect to the spanning source set $(\tilde{p_1}, ..., \tilde{p_m})$.

5.5 3D Sink and Source Detection

The concept of 3D sink and source distributions is equivalent to the 2D case. In order to extract sinks, we start a forward stream line integration from all points in the domain until the distribution of particles converges, and for sources we do a back ward integration. As example we generate a simple 3D vector field given by an average field with:

$$\mathbf{v}_{avg}(x,y,z) = \begin{pmatrix} 2\,(-x\,(1-x)(1+x)(1-y^2) - y^2\,x) \\ 2\,(y\,(1-y)(1+y)(1-x^2) + x^2\,y) \\ 2\,(z\,(1-z)(1+z)(1-x^2) + x^2\,z) \end{pmatrix} \tag{5.10}$$

over the domain $[-2,2]^3$. The 3D uncertain vector field is defined as Gaussian distribution functions:

$$\rho_v(x,y,z;u,v,w) = \frac{1}{2\pi\sqrt{\det(T)}} e^{-\frac{1}{2}(\mathbf{v}-\mathbf{v}_{avg}(x,y,z))^T T^{-1}(\mathbf{v}-\mathbf{v}_{avg}(x,y,z))} \tag{5.11}$$

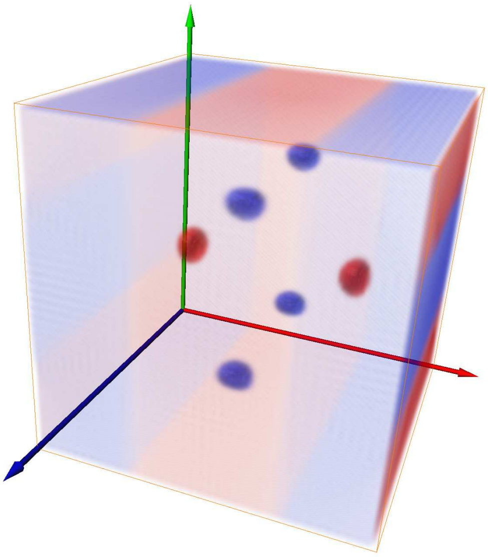


Figure 5.4: Example field (5.10): sink (blue) and source (red) distributions of ρ_v, represented as volume rendering, including inflow and outflow regions that are interpreted as sources and sinks.

with $\mathbf{v} = (u,v,w)^T$ and

$$T = T(x,y,z) = \begin{pmatrix} 0.4 & 0 & 0 \\ 0 & 0.4 & 0 \\ 0 & 0 & 0.4 \end{pmatrix}. \tag{5.12}$$

This result is shown in Fig. 5.4. It consist of two sources, shown in red, and 4 sinks shown in blue. The probability is illustrated by the density used by the volume rendering.

5.6 2D Saddle Points

For a complete topological analysis of the vector field, we have to detect saddle points in addition to the topological skeleton described in section 5.9. However, for uncertain vector fields finding saddle points is difficult, because they are unstable under forward and backward integration (see definition 15). To still find such points, we assume that the magintude of the (uncertain) vector field drops around critical points. To find local magnitude minima, we derive an uncertain vector field ρ_g that corresponds to the gradient of the squared magnitude in ρ_v. Given a sample of $\mathbf{v}_s(x,y) = (u_s,v_s)^T$ of $\rho_v(x,y;u,v)$ at a location (x,y), we can compute the squared magnitude gradient $\mathbf{v}_g$ as

$$\mathbf{v}_g \quad = \quad \nabla(\mathbf{v}_s^2) = (J(\mathbf{v}_s))^T \mathbf{v}_s, \tag{5.13}$$

where $J(\mathbf{v}_s)$ is the Jacobian of $\mathbf{v}_s$. By creating many samples $\mathbf{v}_g = (J(\mathbf{v}_s))^T \mathbf{v}_s$, we can approximate the uncertain gradient field ρ_g. The uncertain sources of ρ_g correspond to local magnitude minima. However, not all minima correspond to saddles. Therefore, we classify the sources of $\mathbf{v}_s$ by computing its (uncertain) Poincaré-Hopf index at ρ_v. See section 5.11.3 for details.

5.7 Uncertain Saddle and Boundary Switch Connectors

For the visualization of separating structures of 3D vector fields only a few approaches exist. The direct visualization of separation surfaces do not work well on 3D vector fields because of visual clutter. One solution to this problem is the concept of saddle and boundary switch connectors [TWHS03, WTHS04]. A saddle connector is the intersection curve of the saddle surfaces from an attracting and a repelling saddle point. Therefore, it is the intersection from two separating surfaces.

Here we adapt the saddle connector approach for 3D uncertain vector fields. In the certain case saddle points are the starting points for the integration of the saddle surfaces. In 3D uncertain vector fields saddle structures are unstable under forward and backward integration of ρ_v, which makes their computation difficult. A solution for 2D uncertain vector fields is proposed in [OGHT10]. This approach computes saddle distributions by the backward integration of the squared magnitude gradient of ρ_v. The result contains all critical distributions, which have to be classified. This is done by computing a modified Poincaré-Hopf index applied on ρ_v. However, such a classification becomes difficult and unstable in the 3D case.

Here we follow a different strategy to find separating structures. We consider the definition of repelling and attracting saddle points for certain 3D vector fields:

- A repelling saddle point has one inflow direction and a 2D manifold with outflow behavior.
- An attracting saddle point has one outflow direction and a 2D manifold with inflow behavior.

In both cases the 2D manifold separates two sources and two sinks, respectively. Based on this observation and the segmentation of 3D uncertain vector fields, we can compute the separating structures without saddle distributions in uncertain vector fields.

Given n sink distributions and m source distributions, a stream line integration started from every location (x,y,z) converges to a sink distribution under forward integration and a source distribution under backward integration. Considering definition 16 these sink and source distributions can be described by linear combinations $\sum_{i=1}^{n} \alpha_i \, p_i(x,y,z)$ and $\sum_{i=1}^{n} \beta_i \, \tilde{p}_i(x,y,z)$ with respect to the spanning sink set $(p_1(x,y,z),\ldots,p_n(x,y,z))$ and the spanning source set $(\tilde{p_1}(x,y,z),\ldots,\tilde{p_m}(x,y,z))$. The corresponding coordinates $\alpha_i(x,y,z)$ and $\beta_j(x,y,z)$ are scalar fields in the domain $\mathbf{D}$. These scalar fields represent the probability that a particle at a location (x,y,z) moves to the i-th sink and comes from the j-th source distribution. Fig. 5.5

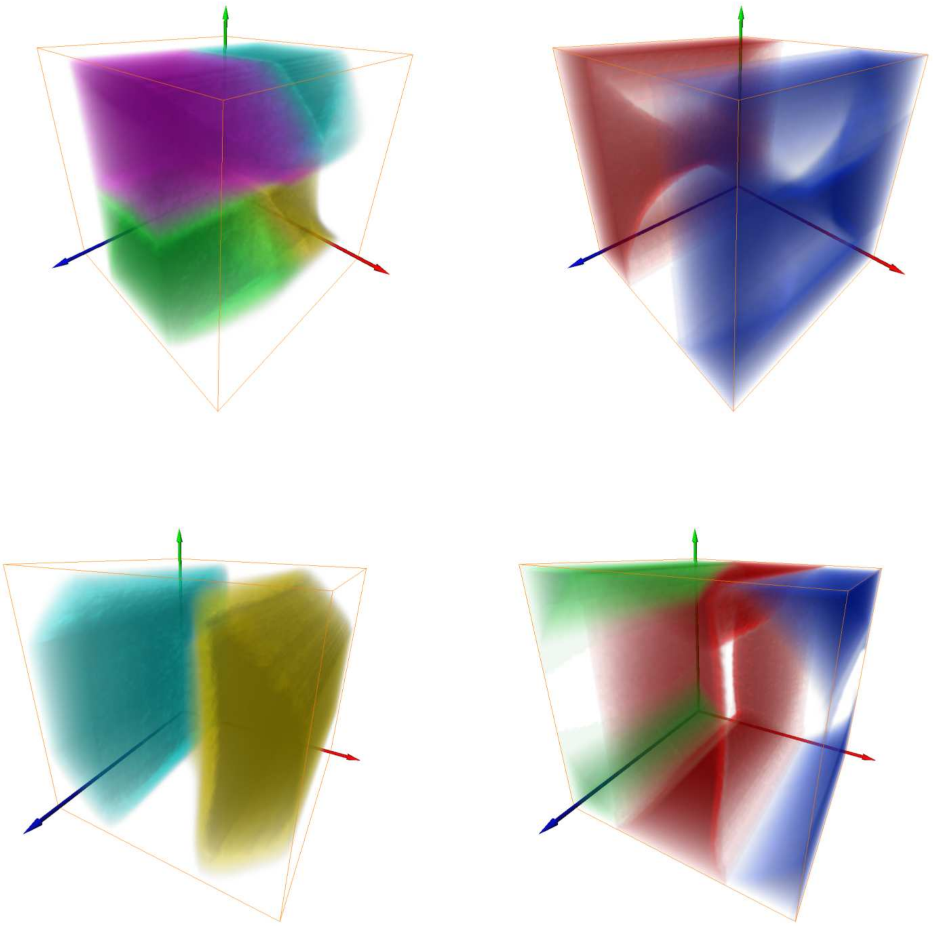


Figure 5.5: Example field of equation (5.10): volume renderings of corresponding coordinates (top, left) of the sink distributions; (top, right) of the outflow distributions; (bottom, left) of the source distributions; (bottom, right) of the inflow distributions.

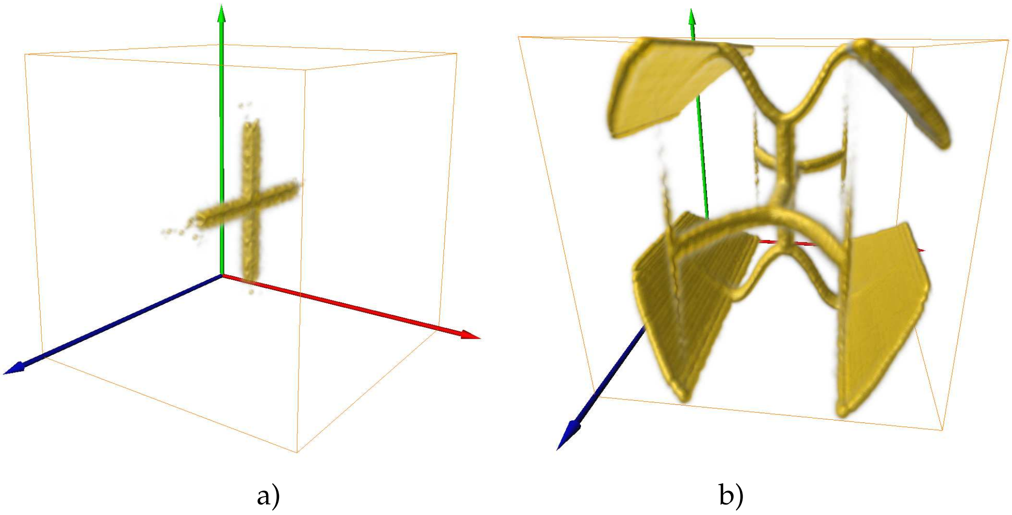


Figure 5.6: Example field of equation (5.10): *(a)* uncertain saddle connectors *(b)* uncertain saddle connectors uncertain saddle and boundary switch connectors.

shows volume renderings of these scalar fields. In most parts of the field the values are either 1 or 0, meaning that either all or no particles converge to the sink resp. source. In contrast, at locations with values $0 < \alpha_i(x,y,z) < 1$ and $0 < \beta_j(x,y,z) < 1$, the particles converge to multiple sinks resp. sources. These volumes represent the separating structures for uncertain vector fields. Now we can define uncertain saddle connectors as overlapping of separating volumes equivalent to the intersection of separating surfaces:

Definition 17 *Given is a 3D uncertain vector field containing the spanning sink set* $(p_1(x,y,z),\dots,p_n(x,y,z$ *and the spanning source set* $(\tilde{p_1}(x,y,z),\dots,\tilde{p_m}(x,y,z))$ *as well as the corresponding probability fields* $(\alpha_1(x,y,z),\dots,\alpha_n(x,y,z)$ *and* $(\beta_1(x,y,z),\dots,\ \beta_m(x,y,z)$. *We select a pair* $(\alpha_a(x,y,z),\alpha_b(x,y,z)) \in (\alpha_1(x,y,z),\dots,\alpha_n(x,y,z))$ *and another pair* $(\beta_c(x,y,z),\beta_d(x,y,z)) \in (\beta_1(x,y,z),\dots,\beta_m(x,y,z))$. *An uncertain saddle connector is the volume where the following scalar field* $s(x,y,z) > 0$*:*

$$s(x,y,z) = (1 - \alpha_m(x,y,z))(1 - \beta_m(x,y,z)) \tag{5.14}$$

with

$$\alpha_m(x,y,z) = \max(\alpha_a(x,y,z),\alpha_b(x,y,z)) \tag{5.15}$$

and

$$\beta_m(x,y,z) = \max(\beta_c(x,y,z),\beta_d(x,y,z)) \tag{5.16}$$

Fig. 5.6a shows an example. We extend this definition to boundary switch connectors by treating outflow and inflow regions as sink and source distributions, as shown in Fig. 5.6b.

Definition 17 gives us a particular uncertain saddle connector. However, not all combinations of corresponding probability fields create saddle connectors. To avoid the unnecessary computation of empty saddle connector fields we compute all uncertain saddle connectors in one scalar field s_{all}, simply by replacing $\alpha_m(x,y,z)$ with

$$\alpha_{max}(x,y,z) = \max(\alpha_1(x,y,z),\ldots,\alpha_n(x,y,z)) \tag{5.17}$$

and $\beta_m(x,y,z)$ with

$$\beta_{max}(x,y,z) = \max(\beta_1(x,y,z),\ldots,\beta_m(x,y,z)) \tag{5.18}$$

resulting in

$$s_{all}(x,y,z) = (1-\alpha_{max})(1-\beta_{max}). \tag{5.19}$$

Note that these saddle and boundary switch connectors already contain the saddle points. They appear at crossings of saddle and boundary switch connectors.

5.8 Closed Stream Lines

In this section we show that the extraction of closed orbits in uncertain vector fields is conceptually the same as extracting uncertain critical points. To illustrate this, we use two analytic examples for the 2D and 3D case. Then we show that the spanning sink and source sets contain all closed orbits which act topologically as sink or source.

5.8.1 Synthetic Examples

All of our examples assume Gaussian distribution functions in order to model the uncertainty. Each uncertain vector field is represented by a mean vector field and a tensor field containing the covariance matrices.

Example 1 *The first example illustrates the 2D case. It defines an uncertain vector field over the domain $\mathbf{D} = [-2,2] \times [-2,2]$. The mean vector field is given as*

$$v_m(x,y) = \begin{bmatrix} y - x\left(\sqrt{x^2+y^2}-1\right) \\ -x - y\left(\sqrt{x^2+y^2}-1\right) \end{bmatrix} \tag{5.20}$$

and the covariance matrix as

$$T(x,y) = \begin{bmatrix} 0.09 & 0 \\ 0 & 0.09 \end{bmatrix}. \tag{5.21}$$

Example 2 *In the 3D case we use a similar uncertain vector field where the z-component is added. It is defined over the domain $\mathbf{D} = [-2,2] \times [-2,2] \times [-2,2]$ with the mean vector field*

$$v_m(x,y,z) = \begin{bmatrix} y - x\left(\sqrt{x^2+y^2}-1\right) \\ -x - y\left(\sqrt{x^2+y^2}-1\right) \\ -z + x \end{bmatrix} \tag{5.22}$$

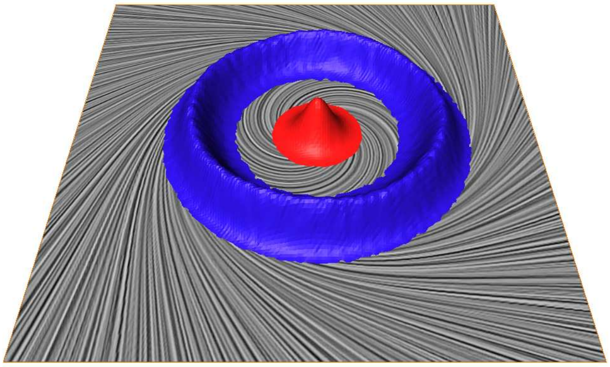


Figure 5.7: Example 1: attracting closed orbit with a source in the middle

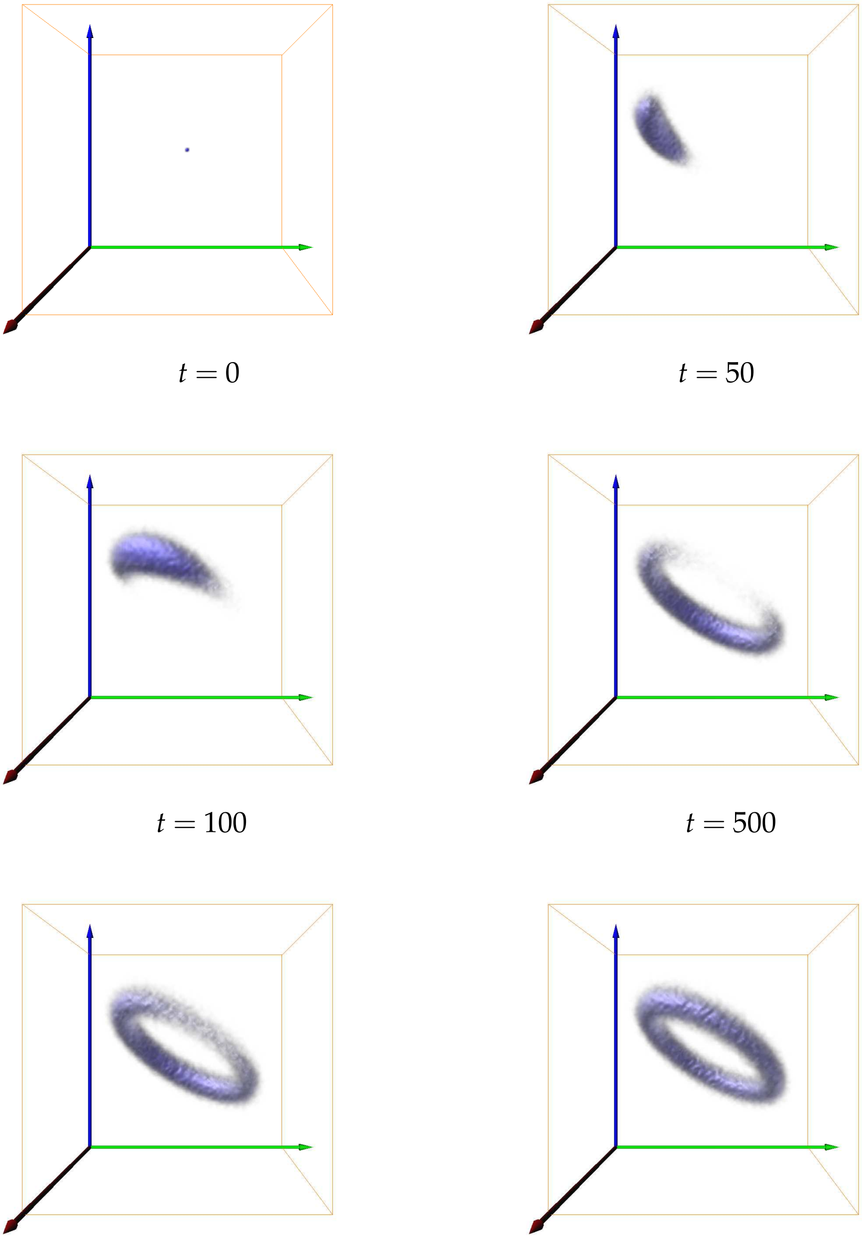
and the covariance matrix

$$T(x,y,z) = \begin{bmatrix} 0.16 & 0 & 0 \\ 0 & 0.16 & 0 \\ 0 & 0 & 0.16 \end{bmatrix}. \tag{5.23}$$

5.8.2 Detection

In section 5.4 and section 5.5 we showed that sinks and sources of uncertain vector fields are detected by integration of particle distributions until they converge. It turns out that attracting and repelling closed stream lines can be found with this scheme as well. In uncertain vector fields, closed stream lines with attracting and repelling character are also represented by unique critical distributions. These distributions act analogously to ordinary sink or source distributions, because they attract and repell neighboring particles. The main difference is the cyclic movement of particles inside the critical distribution. However, this particle movement does not influence the shape of the critical distribution, because the particles converge to an asymptotic cyclic distribution. In Fig. 5.7 such a particle distribution is shown based on the synthetic example 1.

In order to find such a particle distribution, the integration has to be started at some location in a close neighborhood of the closed stream line. The initial particle distribution is not important. Fig. 5.8 illustrates an integration started from a Dirac delta function $p_0 = \delta(1,0,0)$. The integration uses the uncertain vector field defined in example 2. During the first integration steps the particle distribution moves along the path of the closed orbit. With increasing integration time the particle distribution converges to the distribution of the closed orbit. A faster convergence is achieved by a uniform initial particle distribution as shown in Fig. 5.9. The reason



$t = 0$

$t = 50$

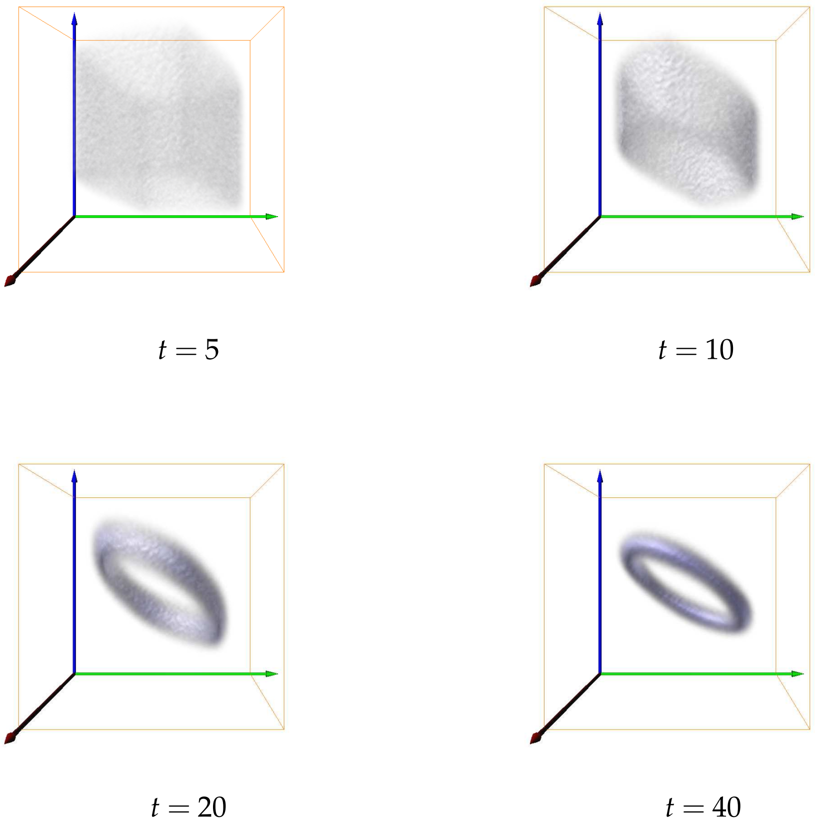
$t = 100$

$t = 500$

$t = 1000$

$t = 3000$

Figure 5.8: Example 2: volume rendering of the particle density during the integration of one uncertain stream line started at location $(1, 0, 0)$.



$t = 5$

$t = 10$

$t = 20$

$t = 40$

Figure 5.9: Example 2: volume rendering of the particle density during the integration of one uncertain stream line started from a uniform particle distribution.

for the faster convergence is that the particle distribution does not need to expand from a single point, and the closed orbit acts attracting to all particles in this field. Therefore, they only move to the closed orbit and stay there.

5.9 Topological Skeleton

Now we can describe our algorithm to extract the uncertain topological skeleton in the following way:
Given an uncertain 2D vector field $\rho_v(x,y;u,v)$ over the domain $\mathbf{D}$:

1. Find a spanning sink set $(p_1,...,p_n)$ of ρ_v.
2. Find a spanning source set $(\tilde{p_1},...,\tilde{p_m})$ of ρ_v.
3. For every location $(r,s) \in \mathbf{D}$:
    a) start a forward stream line integration at $p_0(r,s) = \delta(x-r,y-s)$ until it converges to the sink $p(r,s)$ or completely leaves $\mathbf{D}$
    b) start a backward stream line integration at $p_0(r,s) = \delta(x-r,y-s)$ until it converges to the source $\tilde{p}(r,s)$ or completely leaves $\mathbf{D}$
    c) compute the coordinates $(\alpha_1(r,s),...,\alpha_n(r,s))$ of $p(r,s)$ with respect to $(p_1,...,p_n)$
    d) compute the coordinates $(\beta_1(r,s),...,\beta_m(r,s))$ of $\tilde{p}(r,s)$ with respect to $(\tilde{p_1},...,\tilde{p_m})$

The result of the topological segmentation are $n+m$ scalar fields $(\alpha_1(r,s),...,\alpha_n(r,s))$ and $(\beta_1(r,s),...,\beta_m(r,s))$ with $(r,s) \in \mathbf{D}$. Informally spoken, $\alpha_i(r,s)$ describes the probability that a particle started to move at (r,s) will end in the i-th (uncertain) sink under forward integration. Similarly $\beta_j(r,s)$ gives the probability for a particle starting at (r,s) to converge to the j-th (uncertain) source under backward integration.

Here we described the algorithm for a 2D uncertain vector field. This can be also applied to a 3D uncertain vector field. In this case one more scalar field is computed representing the probability of boundary switch connectors.

5.10 Visualization

For an appropriate visualization of our topological segmentation of *2D uncertain vector fields*, we want to represent both the spanning critical point sets and the topological characterizations in a single visualization. This is possible because the critical points in the spanning critical point set are usually rather separated. Since the probability that a particle started at (r,s) moves into the $i-$th sink under forward and into the $j-$th source under backward integration is described by the product $\alpha_i(r,s) \cdot \beta_j(r,s)$, we represent the $n \times m$ functions $\alpha_i\beta_j$ as randomly colored height fields but cut off areas where $\alpha_i < \epsilon$ for a small user-chosen $\epsilon > 0$ (we are using

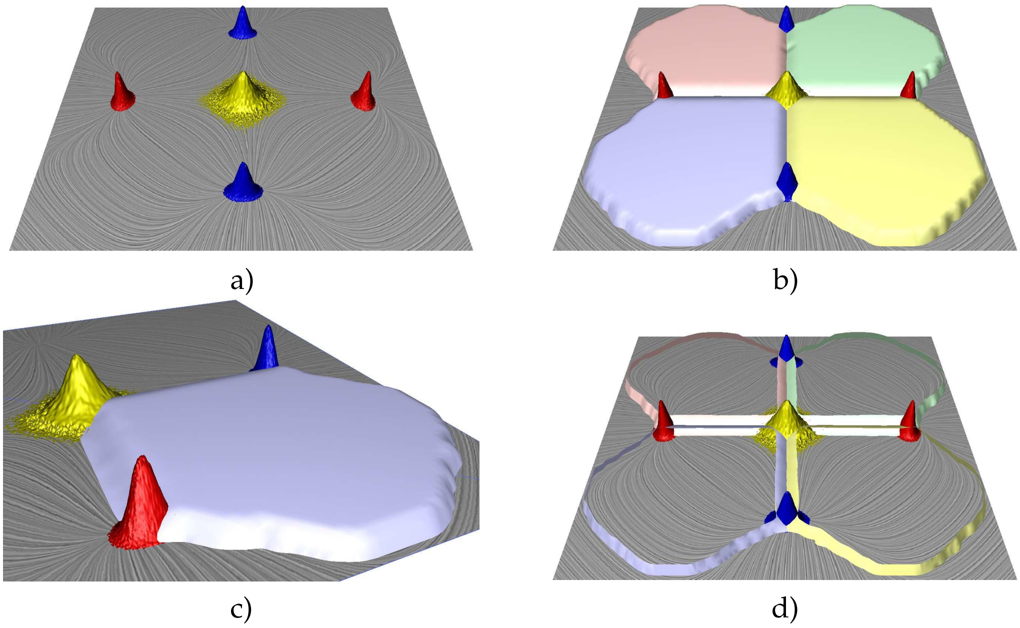
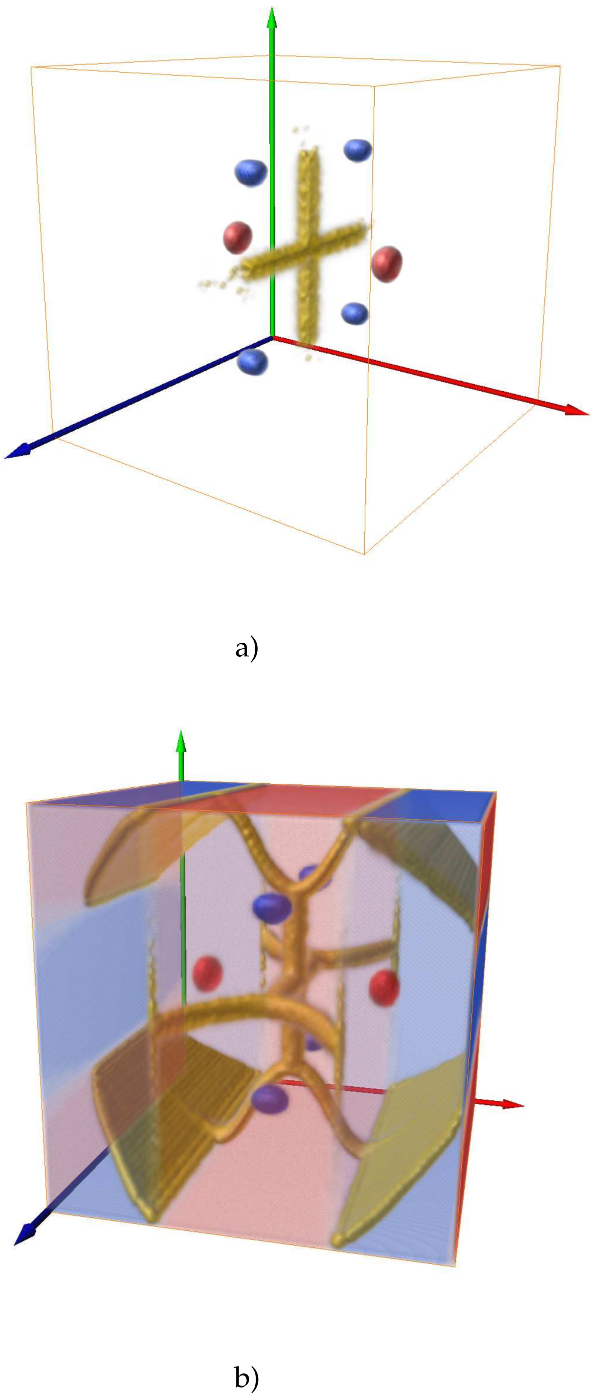


Figure 5.10: Gaussian distributed vector field (6): a) LIC of v_c and uncertain critical points, b) complete segmentation as height fields, c) closeup of one sector depicting the flow from a source (red) to a sink (blue), d) skeleton reduced to uncertain areas.

$\epsilon = 0.01$). We use height fields, because more than two $\alpha_i\beta_j$ functions could overlap. Therefore, other techniques like color maps do not work well for this purpose. Every height surface can be interactively selected/deselected. In addition, we show the critical points as height surfaces as well: sinks in blue, sources in red, and saddles in yellow, all with the same height for visual clarity. Fig. 5.10b-d give an illustration for the vector field (5.9). Since here we have two sources and two sinks, 4 topological regions are represented as height fields (Fig. 5.10b). Fig. 5.10c shows a closeup of a region describing the flow from a source (red) to a particular sink (blue). Note that this height field is almost 1 for a rather large area, indicating a rather certain flow behavior. To focus on the uncertain regions, we cut off the height surfaces in areas of high certainty, i.e., we render them only if $\epsilon \leq \alpha_i\beta_j \leq 1 - \epsilon$. This way, only those areas are rendered where at least two height surfaces are significantly above zero, i.e., where the flow behavior is uncertain. Fig. 5.10d illustrates this. In this figure (as well as in the following visualizations) we use a LIC (line integral convolution) of the average field $\mathbf{v}_c$ as context information for our uncertain visualization. Note that there is no direct correlation between the isolines of our height fields and the stream lines of $\mathbf{v}_c$.

The results of our method in case of *3D uncertain vector fields* are multiple scalar fields, representing probability distributions of sinks, sources, saddle connectors and boundary switch connectors. In order to visualize them we use a volume rendering approach. For this, each type is visualized using a linear transfer function: sinks in transparent to blue, sources in transparent to red and saddle and boundary switch connectors in transparent to yellow. To analyze the whole set of sink and



a)

b)

Figure 5.11: Example field (5.10): *(a)* visualization of the complete topology with saddle connectors *(b)* inflow, ouflow regions and boundary switch connectors added to the visualization.

source distributions the alpha value and range of the transfer functions are user defined, because these features are represented by particle densities, which possibly contain very different maximal values (depending on the size and the global influence of a feature). For a better visual separation of the different distributions we use specular lighting.

With respect to the goal of an adequate visualization we display inflow and outflow regions at the boundaries of the domain also as volume rendering, but more transparent than the real sink and source distributions. This avoids an occlusion of the inner features. We use for inflow regions a transparent to red transfer function similar to source distributions and for outflow a transparent to blue transfer function similar to sink distributions. Additionally we visualize boundary switch connectors with a transparent to yellow transfer function. The final result is shown in Fig. 5.11b. In order to further reduce occlusions we can also hide these boundary regions and only display the sink and source distributions and the saddle connectors. This is shown in Fig. 5.11a.

5.11 Technical Realization

5.11.1 Numerical Stream Line Integration

To integrate stream lines, we chose a Lagrangian Monte-Carlo approach based on probabilistic particle movements. The initial density p_0 is sampled by a high number of particles reflecting the initial distribution. Each particle is integrated by an "uncertain" Euler method: sampling the vector field $(u,v)^T$ at a certain location (x_i,y_i) yields a *random* vector respecting the distribution $\rho_v(x_i,y_i;u,v)$. In practice, we use the Box-Muller method to sample the Gaussian distribution, which limits the sampling region to a finite radius. This sample is multiplied with the time step and added to the particle position to update its state. After a sufficient number of time steps the final distribution is approximated from the discrete spatial distribution of particles. In practice, we monitor changes in the particle density distribution over time and stop the integration if no significant changes are observed. We implement this monitoring by dividing the domain into uniform cells where the particles are counted. In our examples, the cell grid was ten times the size of the input vector field. To check for invariant distributions, we calculate the maximal difference of particle numbers within the cells every 50 time steps. We consider the distribution as stationary if this difference drops below a threshold that depends on the total number of particles and the cell size.

Special care must be taken if the particles leave the domain. At no-slip boundaries (e. g., the step in Fig. 5.14) , we project the particles back into the domain. If the boundary contains inflow or outflow regions, we simply stop the particles and accumulate them. This way, inflow/outflow areas act as topological source/sink and can therefore be considered for our segmentation.

5.11.2 Computing the Spanning Source/Sink Set

To compute the spanning sink set, we start with a Monte Carlo based integration where a high number of initial particles is equally distributed in $\mathbf{D}$. Since this integration includes particles from every location, the critical point p_a to which it converges will contain parts of every element of the spanning sink set. To extract them from p_a, we compute all local maxima of p_a (including the borders of the domain) and assign a region to each of them by a flood fill algorithm. This way we find a critical point p_i for every local maximum of p_a. Before considering all p_i as elements of the spanning sink set, we have to check them for linear independence. If two or more p_i are linearly dependent, they are merged into one component of the spanning sink set. To get a spanning source set, a similar approach is done in backward direction.

In addition to a pure CPU version of the Monte Carlo simulation, we developed a GPU version, including interpolation of the distribution field, pseudo random number generation and path integration. Table 5.1 gives a comparison between the CPU and GPU versions. All timings in the following section were measured on an AMD Opteron 2218 and an NVIDIA GeForce 9800GTX platform with the GPU version.

5.11.3 2D Saddle Points

The extraction of saddle points starts similar to the computation of sources. The key difference is that we use the uncertain vector field ρ_g from section 5.6 to integrate the particles. We implemented two different approaches to achieve this:

Explicit computation: The original field ρ_v transformed by its transposed Jacobian (see equation 5.13) possibly results in uncertainty that is no Gaussian distribution. Therefore, we explicitly compute ρ_g by storing a general density distribution function for each node (x, y) in the vector field on a discrete map. We build this map by randomly sampling the Jacobian and ρ_v at (x, y) a sufficient number of times. Finally, we can advect particles by sampling these maps.

Jacobian on demand: An alternative to the explicit storage of ρ_g is to transform ρ_v in the Euler integration only, when a sample is needed. Here, we compute a probabilistic sample of J^T and ρ_v for every particle at every time step, which is used to advect the particle. The benefit of this method is the reduced memory overhead. Therefore, we also included this in our GPU implementation. Table 5.1 gives a comparison of our implementations.

After finding the source distribution of ρ_g, we segment individual critical points as described in section 5.11.2. Finally, every critical point must be classified w.r.t. ρ_v to find saddle points. For this task, we find the probabilities for the Poincaré-Hopf index by sampling the local region around the critical points and computing its index a sufficient number of times. We consider the critical point only as a saddle, if index -1 has the highest probability.

	CPU	GPU
Euler integration	307ms	100ms
Explicit computation of ρ_g	701ms	-
Jacobian on demand	1006ms	114ms

Table 5.1: Timings for one integration step for 10^6 particles in 2D space.

At this point, we can also extract sink and sources as critical points in ρ_g with index +1 at ρ_v. Therefore, the extraction of critical points in ρ_g serves as a preprocessing step of our algorithm. We use the found sinks and sources as starting points for the particle integration described in section 5.11.2. This resolves the shapes of the sources and sinks in ρ_v.

5.11.4 Corresponding Coordinates, Saddle and Boundary Switch Connectors

For the integration starting at every location (r,s), we use the Monte Carlo integration and release a higher number of particles at (r,s) until the distribution converges to a sink $p_s(r,s)$. The coordinates of $p_s(r,s)$ are the relative number of particles that arrived in the respective sinks $(p_1,...,p_n)$.

In this algorithm, a high number of particles is released at every sampling position in the domain and integrated for a high number of steps. To speed this up, we exploit the spatial coherence of the particles. If the coordinates α_i are already known at a certain particle position, we can stop the particle, because its probabilities to reach sink i are expressed in α_i. Thus, the particle adds to $p_s(r,s)$ with these probabilities. To fully exploit the spatial coherence, we first integrate the particles that are started near the sinks and store the respective coordinates at the starting points. We then successively proceed to the particles that are started in the neighborhood of the just completed region. These particles have a high probability to run into that region, where they can be stopped after only a few integration steps. This way, the total number of necessary integration steps decreases considerably. Depending on the input data we experienced a speed-up of 5× to 10×.

In case of a 3D uncertain vector field, knowing the corresponding coordinates in forward and backward direction at each location in the domain, we are able to compute the probability of saddle and boundary switch connectors as described in section 5.7.

5.12 Results

To test our approach, we apply the extraction of uncertain topology to synthetic, measured, and simulated data sets.

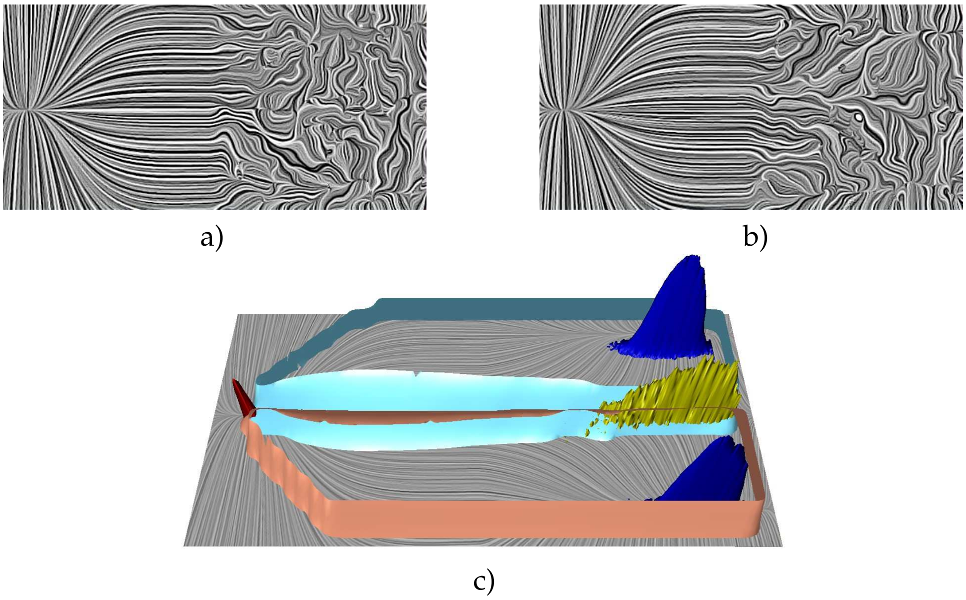


Figure 5.12: Flow from a certain to an uncertain area: a, b) samples of the input fields are identical on the left-hand side but differ significantly in the right-hand region; c) the topological skeleton shows the global transport of uncertainty.

5.12.1 Synthetic Data Sets

Fig. 5.12a and 5.12b show two input vector fields which are obtained by considering the field

$$\mathbf{v} = \begin{cases} \begin{pmatrix} 1 + 1.5x \\ x(y-0.5)\,(5x-1) \end{pmatrix} & \text{if } x < 0 \\ \begin{pmatrix} 1 - x \\ -8xy\,(2y-1)\,(y-1) \end{pmatrix} & \text{if } x \geq 0 \end{cases} \tag{5.24}$$

in the domain $[-0.8, 1.2] \times [-0.2, 1.2]$, where a rather strong Gaussian noise is added to the right-hand region, i.e., the region $x > 0$. From a sufficient number of them we can reconstruct the uncertain distribution field

$$\mathbf{v}_c = \mathbf{v} \ , \ \mathbf{T} = \begin{cases} \mathbf{T}(x,y) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} & \text{if } x < 0 \\ \mathbf{T}(x,y) = \begin{bmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{4} \end{bmatrix} & \text{if } x \geq 0 \end{cases} \tag{5.25}$$

This means that for $x < 0$ we have a zero local uncertainty: all input fields have exactly the same value there. Fig. 5.12c shows our uncertain topological skeleton consisting of one source (red height field), two sinks (blue) and one saddle (yellow). They give two topological sectors where the height fields are shown only in the uncertain areas.

This example shows two properties of our approach: firstly, the right hand side of the input fields have a strong topological complexity which is mainly due to the added Gaussian noise. Our approach can deal with it by revealing the few

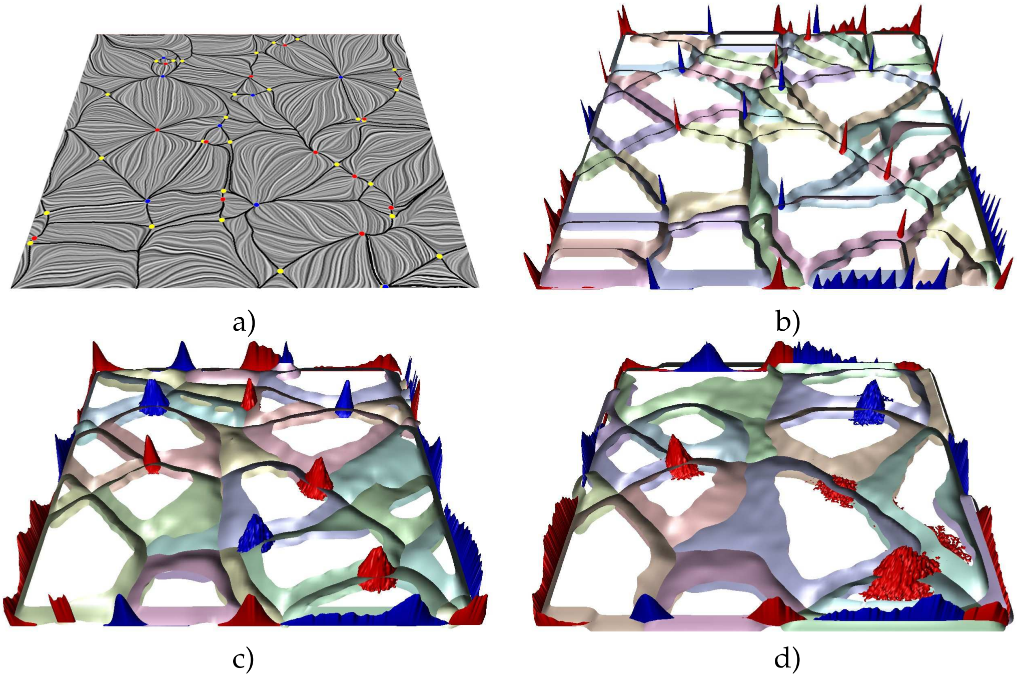


Figure 5.13: Increasing the uncertainty of a random vector field: a) certain topology of mean vector field; b) $\|\mathbf{T}\|_F = 0.2$; c) $\|\mathbf{T}\|_F = 2.0$; d) $\|\mathbf{T}\|_F = 5.0$.

important topological structures. Secondly, although there is no local uncertainty in the left-hand part of the flow, our algorithm finds a rather strong global uncertainty there: for rather large areas it is uncertain to which sink a particle started from there will converge. These are the areas where the two height surfaces intersect and their supports overlap.

Fig. 5.13 shows a synthetic field to study the impact of increasing uncertainty on the topology. We constructed an initial vector field $\mathbf{v}_c$ as the gradient of a Perlin noise scalar field. As shown in Fig. 5.13a, its topology contains many critical points. Then we added different amounts of uncertainty, i.e., an isotropic matrix $\mathbf{T}$ with $\|\mathbf{T}\|_F = 0.2, 2.0, 5.0$, respectively. Increasing $\|\mathbf{T}\|_F$ has two effects: the height surfaces show more overlapping areas and therefore more uncertainty, and the number of uncertain critical points decreases. Compared to Fig. 5.13a, the uncertain topology contains fewer structures that are more robust against variations in flow behavior.

5.12.2 PIV Data Set

Fig. 5.14 shows a PIV (particle image velocimetry) measurement of a real flow around a backward facing step. The PIV measurement was carried out 1024 times, yielding 1024 reconstructed vector fields on a regular 105x103 grid. Fig. 5.14a and 5.14b show two consecutive fields of these which act as input for our approach. They look rather different, indicating a higher local uncertainty. Moreover, no particular order or temporal similarity comes with the data, such that an

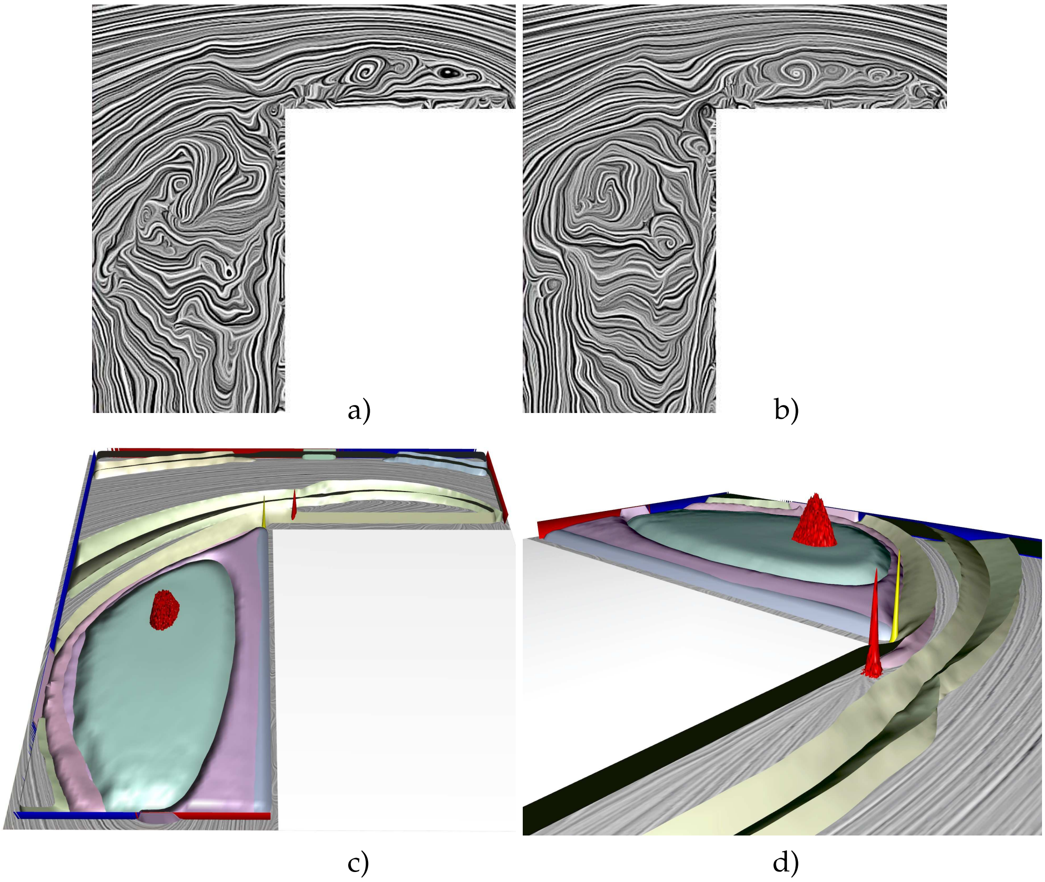


Figure 5.14: PIV data set of a flow around a backward facing step: a, b) two of the 1024 data sets, c) uncertain topological skeleton, d) close-up.

uncertain Gaussian distribution field is the method of choice for reconstruction. Our uncertain topological visualization (Fig. 5.14c) consists of one saddle and two sources. In addition we interpret areas of inflow/outflow at the domain boundaries as sources/sinks as well, yielding 3 more sources and 3 more sinks (marked red and blue at the domain boundaries). The obstacle is modeled with no-slip boundaries. The visualization shows a generally high uncertainty: in rather large areas, none of the height surfaces is 1, meaning that only probabilistic statements about the flow behavior are possible. The only certain region is the upper part of the flow where the underlying LIC images are visible, indicating a rather laminar flow from the right-hand to the left-hand boundary. Fig. 5.14d is a close-up of Fig. 5.14c. The computing time for this data set was 5 minutes and 41 seconds.

5.12.3 Flow in a Tube

This data set is a PIV measurements that consists of 240 snapshots of a slice of a flow in a tube. The measurements are taken from a flow with constant velocity of 4m/s. The timing for the measurements are 190μs and the uptake rate 3,3Hz. We

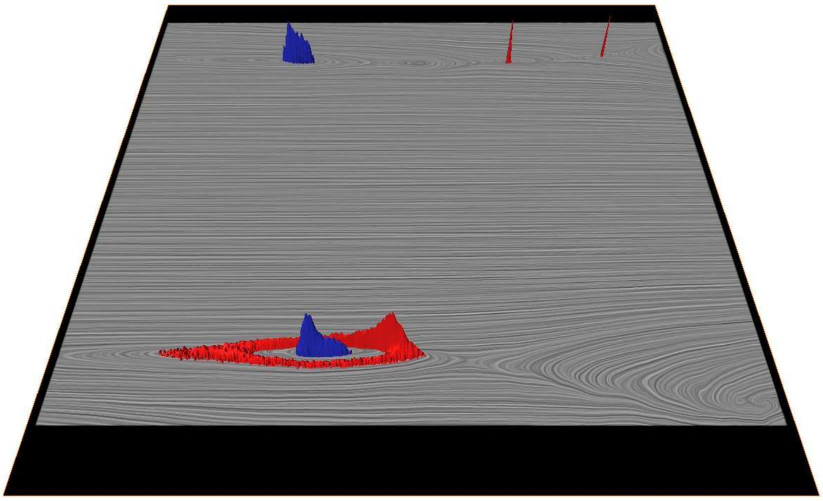


Figure 5.15: Topology of the uncertain 2D vector fields generated by the PIV measurement of a flow in a tube. Attracting features are visualized in blue, repelling features in red.

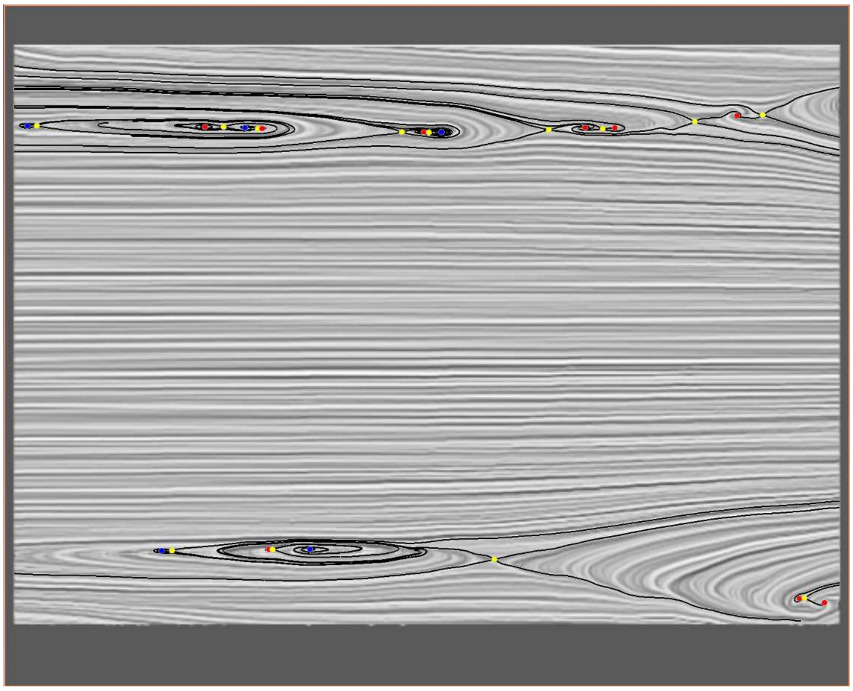


Figure 5.16: Visualization of topology with underlying LIC of the mean vector field of all PIV measurements. Sinks are blue colored, sources red colored and saddles yellow colored.

used the 240 single 2D velocity fields to generate an uncertain 2D vector field. The uniform data grid has a resolution of 83 × 67.

Fig. 5.15 shows the resulting uncertain topology. It contains one repelling closed stream line, two sinks and two sources. Fig. 5.16 illustrates the topology of the mean vector field of all 240 measurements, containing five sinks, nine sources and eleven saddles. In comparison the uncertain topology contains less critical structures.

For the computation of the particle distributions representing attracting and repelling structures, we generate a uniform initial distribution with 100 particles per grid cell, with a total of 556,100 particles. The computation time for this example is about 25 minutes.

5.12.4 Flow in the North Sea

Fig. 5.17 shows the visualization of simulated flow of the North Sea in the German Bight (Deutsche Bucht) between 17/10/2008 and 09/11/2008. Although the flow is 3D, it is dominated by its horizontal components and can therefore be interpreted as a 2D vector field with divergence. The flow is strongly dominated by the tides, leading to a constantly changing domain over time. For a visual analysis, we have selected all time steps of the same relative tide time (i.e., where the sea has flooded approximately the same amount of land), leading to 33 data sets over a regular 160 × 130 grid which describe the flow every 12 hours and 25 minutes (one tidal period). Fig. 5.17a and 5.17b show two of the input data sets. Fig. 5.17c shows the results of our uncertain topological analysis: including boundary inflow/outflow we have 19 sources and 21 sinks, the corresponding uncertain topology reveals both regions of certain and uncertain behavior. The computing time was 8 minutes and 56 seconds.

5.12.5 Flow in a Bay Area

Fig. 5.18 shows a data set describing (the perpendicular of) the flow of a bay area of the Baltic Sea near Greifswald (Germany). The data was given as an incomplete flow data set on a regular 115 × 103 grid at 25 time steps. We use these time steps as the input fields of our method. Fig. 5.18a-c show three of them as LIC images. Our uncertain topology revealed 74 sinks and 78 sources including regions on the boundary that where detected as critical points (see Fig. 5.18d). The flow falls into three classes of main behavior. In the middle part, there are large regions of a certain behavior where the underlying LIC plane is visible. Contrary, in the areas left and right of the center region strong overlap of height surfaces appears, indicating a generally more uncertain behavior. The computing time for this data set was 2 minutes and 13 seconds.

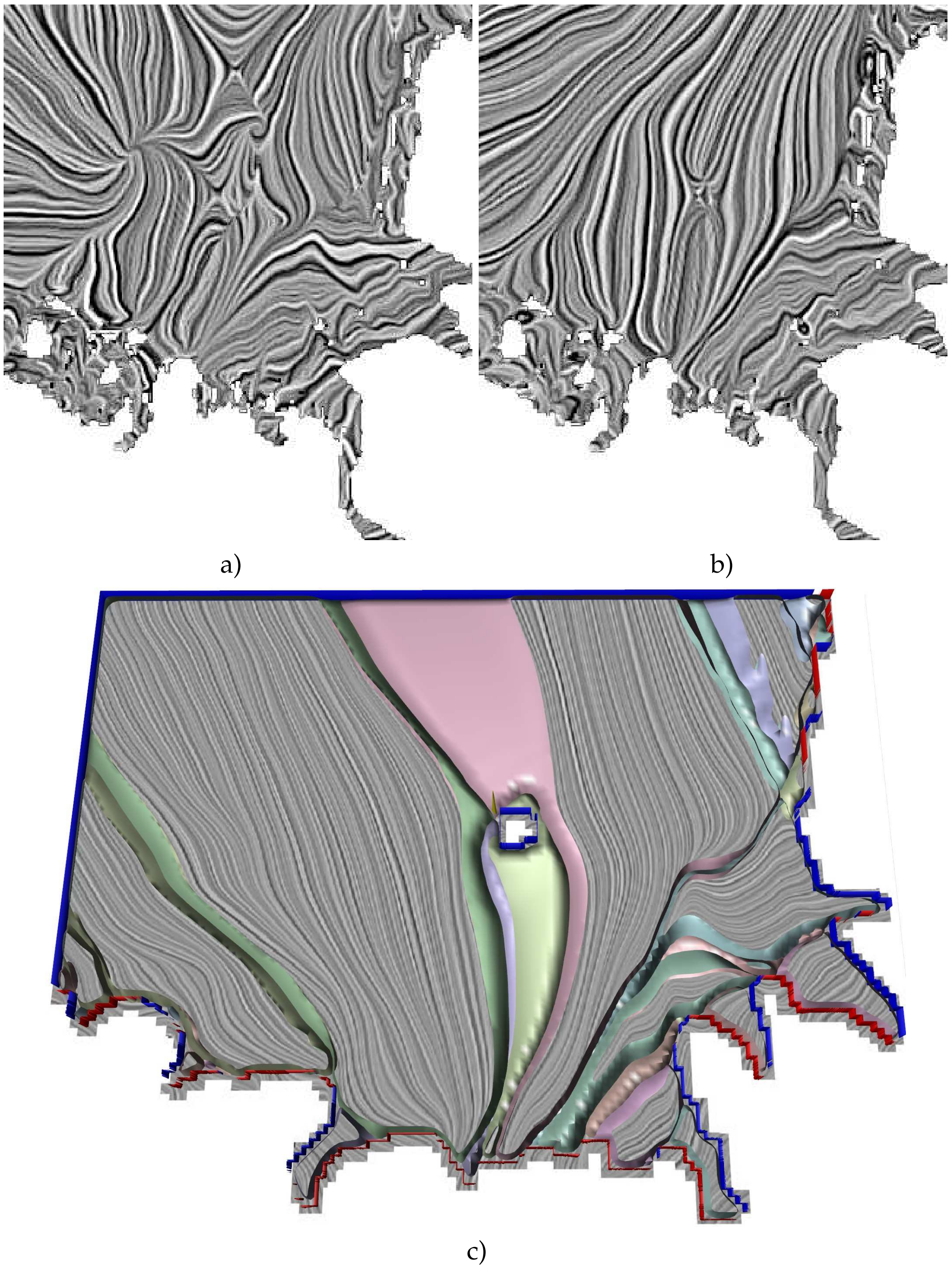


Figure 5.17: Topological skeleton of a simulated flow. a,b) show LIC images from sample vector fields; c) shows the uncertain topology with an underlying LIC of the average vector field.

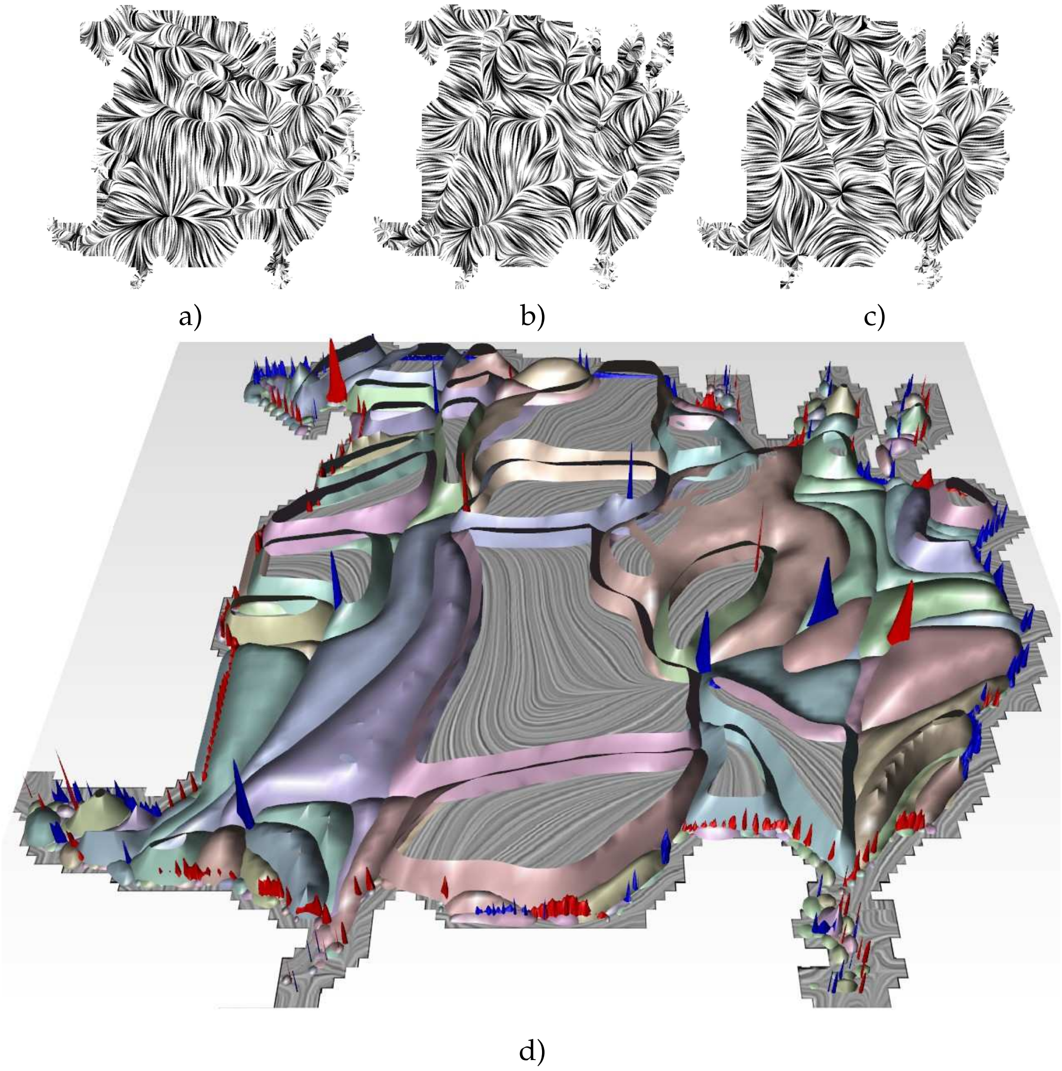


Figure 5.18: Flow in a bay area: a-c) LIC of input fields; d) topological skeleton of the uncertain vector field.

5.12.6 DNS Simulation

Direct Numerical Simulations (DNS) are becoming increasingly useful for turbulent flow applications. They constitute a natural complement to experiments, in particular to investigate in detail complex physical processes in simple geometries. It does not rely on any approximate turbulence models, nevertheless the computational cost is tremendous.

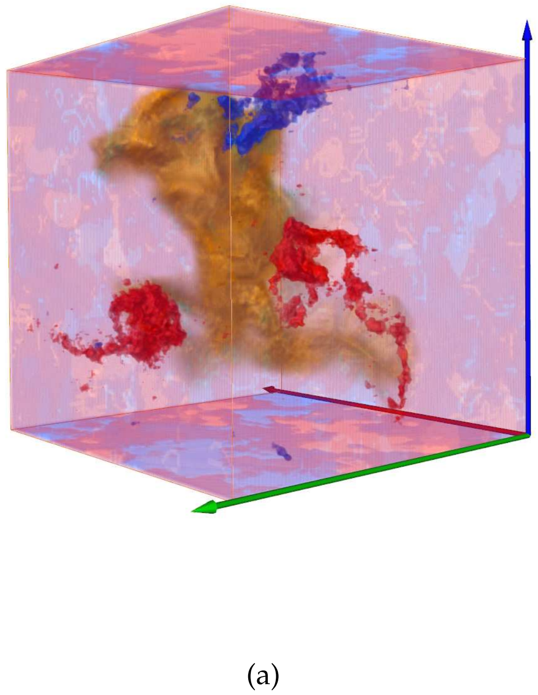
The simulations presented here have been carried out with the DNS code π^3 originally developed by Thévenin and coworkers [FSL$^+$09]. It is a finite-difference three-dimensional code solving the fully compressible Navier-Stokes equations for reacting flows. Derivatives are computed using centered explicit schemes of order six, the temporal integration is realized with a Runge-Kutta algorithm of order four. The code is parallelized through domain decomposition.

In this study a turbulent air flow without reactions is considered in a cubic domain with a size of $0.5 \times 0.5 \times 0.5 cm^3$ with 51 equidistant points in each direction. This leads to a fixed, homogeneous spatial resolution of 100μm, necessary to resolve accurately the fine details of the flow. Periodic boundary conditions are applied on all sides of the domain.

A turbulent flow is considered initially superposed with a field of synthetic homogeneous isotropic turbulence corresponding to a von Kármán spectrum with Pao correction for near-dissipation scale.

An initial turbulence field is generated with a turbulent fluctuation velocity $u' =$ 3m/s and an integral scale $L_t = 4.5$mm. It yields a turbulent Reynolds number of $\mathrm{Re}_t = 250$ and the corresponding Kolmogorov scale is 25mm.

The result of such a simulation was given as a time series of 50 vector fields in an interval of 3.2μs. The simulated turbulent flow changes its characteristic very fast. In the beginning the flow contains strong divergence that decreases over the time. So we can find topological features inside the field only in the beginning, otherwise there are only features generated by boundary effects. For the creation of the uncertain vector field we have chosen four time steps in a rather small time interval from 0.0025¯s to 0.01¯s such that the field does not change too much and contains some features. The resulting visualization in Fig. 5.19a contains 5 sink and 3 source distributions (including inflow and outflow) and one saddle like region in the middle of them. The computation time for this example is about 42 minutes: a few seconds for precomputation of the step size fields, 7 minutes for the computation of the spanning sink and source sets with only 20 particles per data point (1,000 integration steps were needed for each set) and 35 minutes for the computation of the corresponding coordinates with 200 particles per grid point. In comparison Fig. 5.19b shows the common topological skeleton of the average certain vector field. It contains 234 critical points and no clear structure is visible. This data set confirms for 3D what has already been shown for 2D [OGHT10]: the consideration of global uncertainty in flow fields tends to act as a feature reduction. The uncertain skeleton contains only the most important features, while unimportant topological features in the certain flow data (which are mainly due to noise) are removed.



(a)

(b)

Figure 5.19: DNS simulation: *(a)* topology of the turbulent flow phenomenon example, described by an uncertain vector field *(b)* critical points and illuminated stream lines of the turbulent flow phenomenon example.

5.12.7 Flow in the Pacific Ocean

These simulations were carried out at the German Climate Computing Center using the MPI-OM ocean model. The MPI-OM model was developed at the Max-Planck-Institute for Meteorology in Hamburg, and is used to simulate various processes in the different oceanic regions. It is part of the simulations that are carried out for the IPCC assessment reports. The data set has a horizontal resolution of 1 degree (360x180) and consists of 40 depth levels, specified by pressure, and shows the velocity of ocean currents. The data set contains the average vector fields for each month of one year.

We use these velocity fields of this simulation to create an uncertain vector field. It symbolizes the general global flow in the oceans over one year. This data set contains several thousands of features. For this reason we choose only a section of this data set, a part of the Pacific Ocean in front of South America. The original data set is given on a stacked grid. To use our method we resampled the data on an uniform grid. The grid of the section that we analyze is $64 \times 91 \times 100$. The result of our analysis contains 124 uncertain sources and 104 sinks as shown in Fig. 5.20 and Fig. 5.21. We compare our result with the analysis of the average vector field, which contains 527 critical points (shown in Fig. 5.22).

The runtime of the computation of the uncertain topology is about 7 hours: 10 minutes precomputation, 50 minutes computation of the spanning sink and source sets with 10 particles per grid point and the rest of the time for the computation of the corresponding coordinates with 100 particles per grid point. The reason for this long computation time is the strong variation in velocity in the given data. The precomputation of the step size helps to compensate for the very slow motion in the deep sea, but it still stays slow. To compute the spanning source and sink sets we needed 12,000 integration steps for each set until the particles arrive in a classified region.

5.12.8 Segment of the Pacific Ocean

This example is based on a flow simulation of the Pacific Ocean, described above. It contains hundreds of critical structures, therefore we picked only a very small segment of the Pacific Ocean in order to show some examples of closed stream lines. The region of interest has a resolution of $28 \times 22 \times 6$.

Fig. 5.23(left top) shows a volume rendering of the sink and source distributions of this data set. There are two attracting and two repelling closed stream lines. We adjusted the range of the linear transfer functions to the density of particles on closed stream lines, because there the particle density is much smaller than in the region of real sink and source distributions.

Again, we used a uniform initial particle distribution with 100 particles per grid cell (a total of 369,600 particles). For counting particles in buckets we choose a denser grid with a resolution of $140 \times 110 \times 30$. The computation time is about 40 minutes.

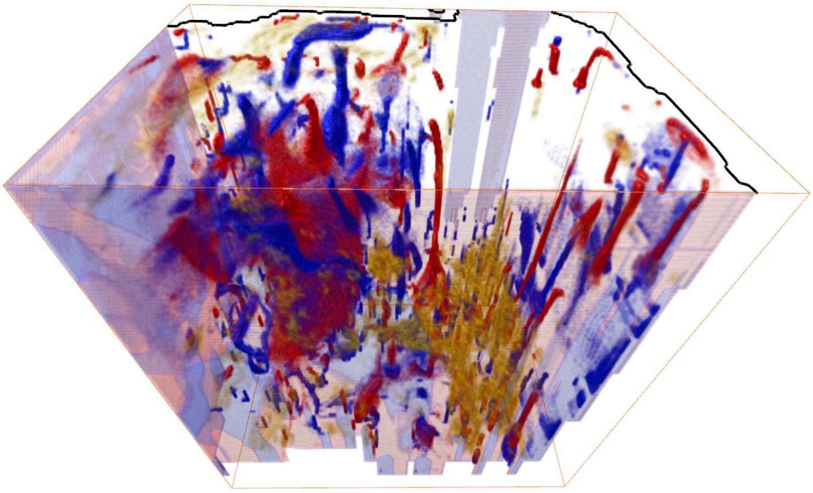


Figure 5.20: Flow in the Pacific Ocean: topology of the uncertain vector field (north top).

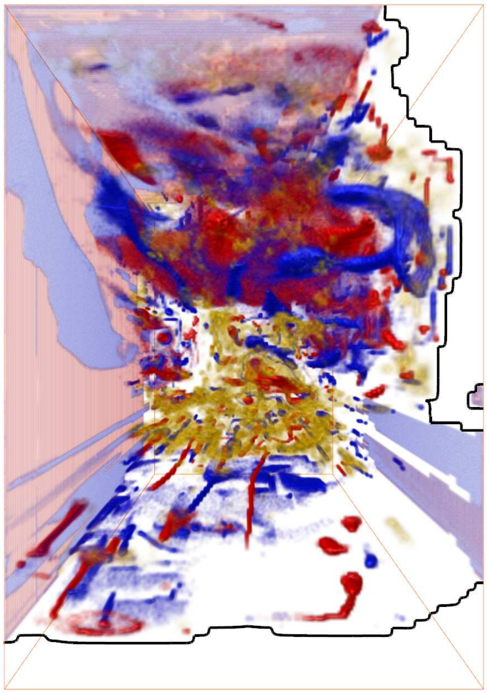


Figure 5.21: Flow in the Pacific Ocean: topology of the uncertain vector field (north left).

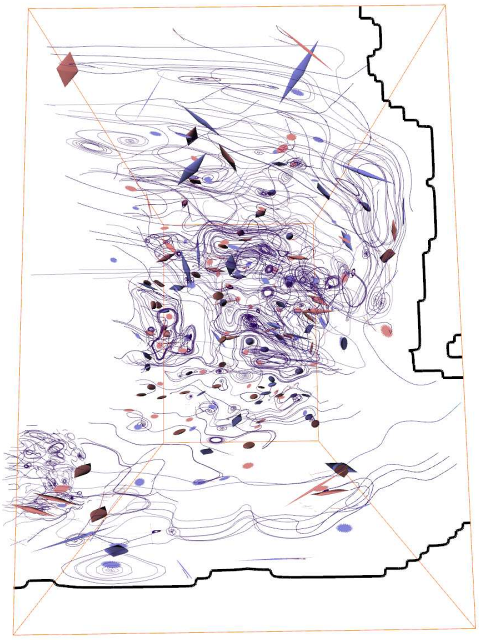


Figure 5.22: Flow in the Pacific Ocean: critical points and illuminated stream lines of the average vector field.

Furthermore, we use this example to analyze the stability of closed orbits in uncertain vector fields. For this we artificially amplify the uncertainty by factors of 2 and 4. Fig. 5.23 shows results for different amplitudes of uncertainty. The left column shows the critical structures and the right column shows volume renderings of a scalar field s with

$$s = \frac{maximal\ eigenvalue\ of\ covariance\ matrix}{length\ of\ mean\ vector}.$$

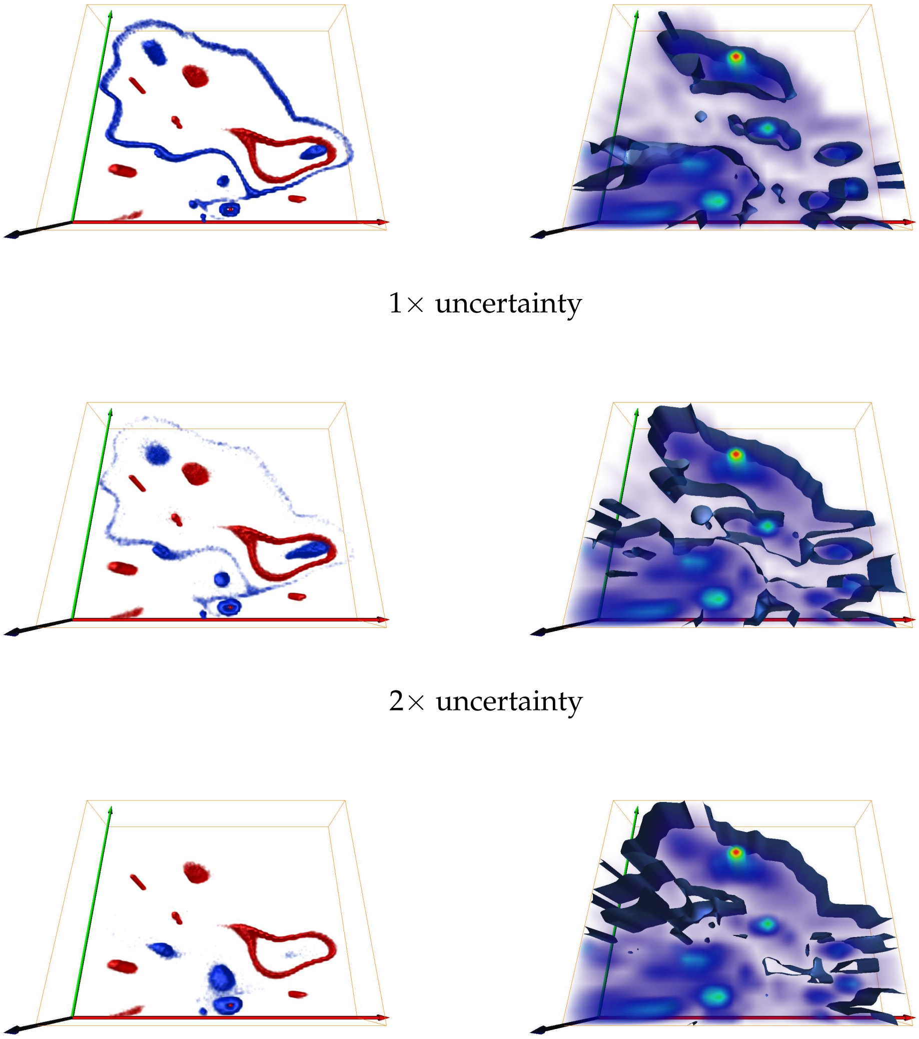
This gives an impression of the impact of the uncertainty to the movement of particles in the uncertain vector field. For isovalues $s > 1$ the uncertainty dominates the particle motion. We illustrate this case with an isosurface of $s = 1$.

While increasing the uncertainty an attracting closed stream line and a sink distribution become weaker and finally disappear. However repelling structures hardly change, even if they exist in regions where uncertainty dominates, like the large repelling closed stream line.

To consider the results the stability of critical structures of any type are mainly affected by global uncertainty and not by the local uncertainty.

5.12.9 Uncertain Lorenz Attractor

The last example is derived from the well-known Lorenz attractor, which has no closed stream line in the strict sense. Here we want to show that our method also



1× uncertainty

2× uncertainty

4× uncertainty

Figure 5.23: Segment of the Pacific Ocean with different amount of uncertainty: *(left column)* Volume renderings of the particle distributions visualize attracting (blue) and repelling (red) structures. While the uncertainty increases, the big attracting closed stream line becomes weaker until it disappears. *(right column)* Volume renderings of a scalar field s and an isosurface with $s = 1$. For the region where $s > 1$ the uncertainty dominates the particle motion in the uncertain vector field.

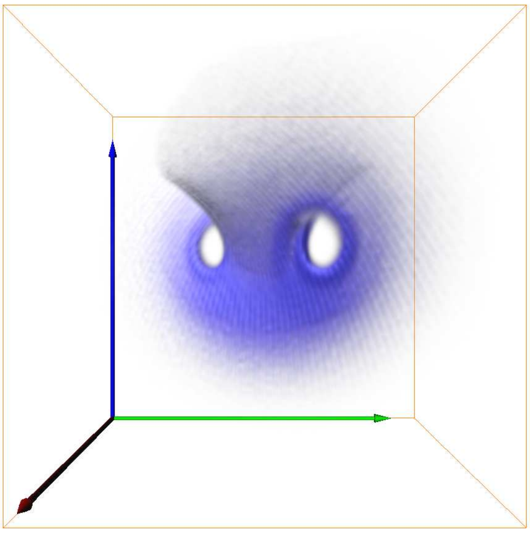


Figure 5.24: Uncertain Lorenz attractor.

works for uncertain strange attractors. We set up an uncertain vector field with the Lorenz attractor as the mean field:

$$v_m(x,y,z) = \begin{bmatrix} \sigma(y-x) \\ rx - y - xz \\ xy - bz \end{bmatrix}.$$

with

$$T(x,y,z) = \begin{bmatrix} 0.16 & 0 & 0 \\ 0 & 0.16 & 0 \\ 0 & 0 & 0.16 \end{bmatrix} \;,\; \sigma = 10 \;,\; r = 28 \text{ and } b = \frac{8}{3}.$$

This field is defined over the domain $\mathbf{D} = [-25,25] \times [-25,25] \times [0,50]$. Fig. 5.24 shows the critical particle distribution which is created by a forward integration of a uniform initial distribution with 30 particles per grid cell (a total of 3,750,000 particles). The algorithm converges after 250 integration steps, with a step size of $\Delta t = 0.02$. The computation time is 42 seconds. Although the mean field includes a strange attractor instead of a closed stream line, our method still finds this structure.

5.13 Discussion

Uncertain vector field topology is clearly a generalization of the classical certain vector field topology. However, even though we provide a GPU-based highly-parallel implementation, the computing time for uncertain topology is still higher than for the certain topology. This is due to the fact that we still do an expensive integration for every point, while for certain topologies only a low number of separatrices have to be integrated. These significantly high computing costs only pay off for data sets where the uncertainty is relevant and of special interest. For data sets where the uncertainty is less relevant, a faster certain topological extraction may give almost the same segmentation than the methods presented in this work. Our approach is therefore not going to replace the certain topology but is a generalization particularly useful for the analysis of global uncertainty in flow fields.

Our method produces usually fewer uncertain critical points than the input fields have, it can therefore be interpreted as a topology simplification method. There is a number of approaches for topological simplification of (certain) vector fields [dLvL99a, dLvL99b, TSH00, TSH01]. However, these methods consider only local conditions for simplification and therefore might give different results than our global method. For example, the results in Fig. 5.12 are hard to achieve with local topology simplification methods.

6 Vortices in Uncertain Vector Fields

Beside topological techniques vortical structures play a vital role in vector field visualization. We gave an overview of region-based and geometry-based vortex criteria without uncertainty in section 3.2. The goal of this chapter is to generalize these concepts to uncertain vector fields. Therefore we have to extend the definition of uncertain vector fields which we introduced in section 4. With this definition we are able to compute derivatives of uncertain vector fields which is the base for computing vortex criteria.

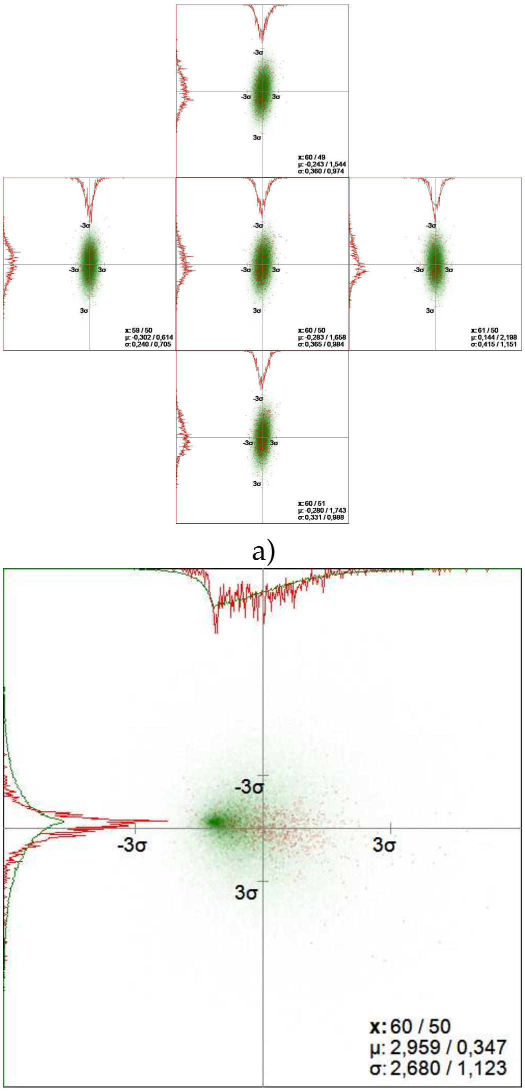
6.1 Jacobian of Uncertain Vector Fields

In order to compute vortex structures, the usual way is to compute derived fields from the velocity field which contain its derivatives. In the uncertain case, the derived fields are uncertain fields as well. To compute them, two problems have to be solved:

1. Even if the original field has a Gaussian distribution, the derived fields are not Gaussian. Moreover, the derived fields generally do not have closed form solutions.
2. When computing the uncertain velocity gradient it has to be considered that the distributions at adjacent grid points are correlated.

We explain and illustrate both problems at an example: the computation of the uncertain acceleration field. In the certain case, acceleration is given as $\mathbf{a} = \mathbf{J}\mathbf{v}$, where $\mathbf{J}$ is the Jacobian matrix. In the uncertain case each component of $\mathbf{J}$ is a 1D probability distribution function for which we assume a Gaussian distribution. However, after the multiplication with the uncertain vector field (which is also a Gaussian distribution), another kind of distribution function results. It is a product distribution which is computable in a closed form for only a few special cases [Cra34]. In fact, only for zero means the product distribution is described by a modified Bessel function of the second kind. In general, the uncertain acceleration cannot be written in a closed form.

To illustrate this, we consider a part of the PIV data set that will be fully introduced later in section 6.4.3. Here, it is sufficient to mention that at each grid point 1024 velocity vectors were measured, i.e., $m = 1024$, and that we consider the grid point $(i,j) = (60,50)$. Fig. 6.1a consists of 5 parts. The one in the middle shows the (end points of the) vectors $\mathbf{v}_{i,j,k}$ for $i = 1,..,1024$ as red dots where the median is moved to the image center. This red point cloud already gives an impression of the distribution of $\mathbf{v}_{i,j,k}$. Using a binning technique, the marginal distributions can be shown



a)

b)

Figure 6.1: Uncertain velocity and acceleration at grid point (60,50) of the PIV data set using uncorrelated Gaussian distributions. a) sample vectors $\mathbf{v}_{i,j,k}$ as red dots and marginal density distribution as red curves; uncorrelated Gaussian reconstruction and marginal density distribution as green point cloud and curve; b) sampled acceleration and marginal distribution as red point cloud and curves; Monte Carlo sampling of acceleration by using uncorrelated Gaussian at grid points as green point cloud and curves: the red and green curves do not coincide.

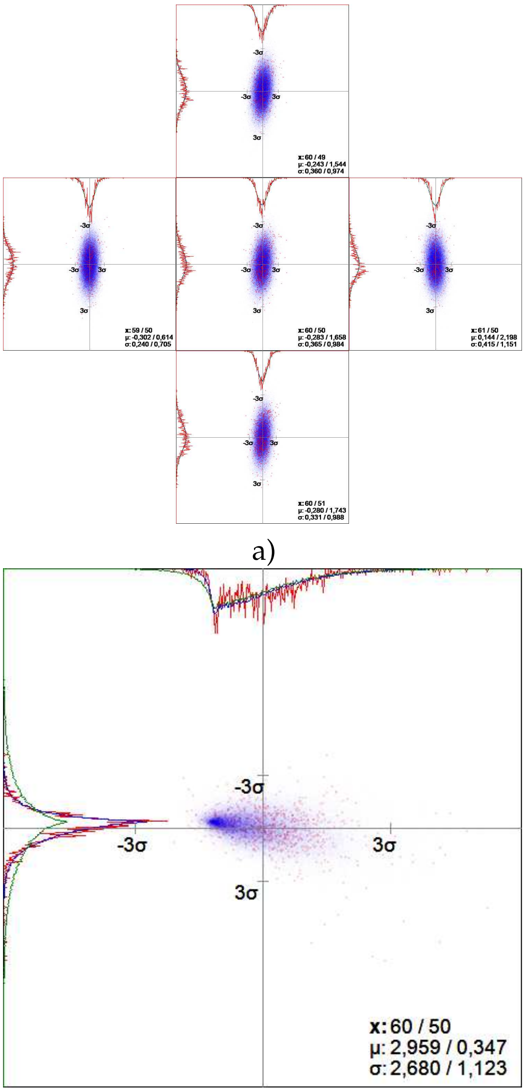
as red curves on the boundaries of Fig. 6.1a (middle). Note that due to the rather low number of samples, the red curves look non-smooth. Nevertheless, their general shape can be observed. By applying Gaussian fitting we have computed the Gaussian distribution field at the grid point. We visualize it by a Monte Carlo approach, i.e. by computing a large number (here 100,000) of green random sample points of the distribution as well as the marginal distributions as green curves. The image clearly shows that a Gaussian distribution is indeed a suitable choice at the considered grid point: the densities of the red and green points are clearly correlated, and the red and green curves are rather similar. The remaining 4 images in Fig. 6.1a show the same for the grid points $(i-1,j),(i+1,j),(i,j-1)$, and $(i,j+1)$ respectively. Fig. 6.1b shows the uncertain acceleration at (i,j) in the following way: for $k=1,...,m$, we compute $\mathbf{a}_{i,j,k}=\mathbf{J}_{i,j,k}\mathbf{v}_{i,j,k}$ where the Jacobian is estimated by central differences $\mathbf{J}_{i,j,k}=\left(\frac{\mathbf{v}_{i+1,j,k}-\mathbf{v}_{i-1,j,k}}{2\,dx},\frac{\mathbf{v}_{i,j+1,k}-\mathbf{v}_{i,j-1,k}}{2\,dy}\right)$ and dx,dy are the grid resolutions. Then $\mathbf{a}_{i,j,k}$ are drawn as red dots in Fig. 6.1b. By binning, the two red curves show the marginal density distributions of $\mathbf{a}_{i,j,k}$. They clearly show that their distributions are *not* Gaussian (problem 1 above). The green point cloud is produced by a Monte Carlo approach in the following way: for $h=1,...,100000$, we consider random sample vectors $\tilde{\mathbf{v}}_{i,j,h}$, $\tilde{\mathbf{v}}_{i-1,j,h}$, $\tilde{\mathbf{v}}_{i+1,j,h}$, $\tilde{\mathbf{v}}_{i,j-1,h}$, $\tilde{\mathbf{v}}_{i,j+1,h}$ obeying the Gaussian distributions at the respective grid points. From them, we compute the random acceleration vectors as $\tilde{\mathbf{a}}_{i,j,h}=\tilde{\mathbf{J}}_{i,j,h}\tilde{\mathbf{v}}_{i,j,h}$ with $\tilde{\mathbf{J}}_{i,j,k}=\left(\frac{\tilde{\mathbf{v}}_{i+1,j,k}-\tilde{\mathbf{v}}_{i-1,j,k}}{2\,dx},\frac{\tilde{\mathbf{v}}_{i,j+1,k}-\tilde{\mathbf{v}}_{i,j-1,k}}{2\,dy}\right)$. Again, every $\tilde{\mathbf{a}}_{i,j,h}$ is shown as a green dot, and the green curves show the marginal distribution of $\tilde{\mathbf{a}}_{i,j,h}$. The green curves again indicate a non-Gaussian distribution. Moreover, the red and green curves do not coincide, meaning that the chosen model of computing the Jacobian is not appropriate (problem 2).

Our approach to overcome the problems 1 and 2 mentioned above is to use a Monte Carlo approach together with a correlated estimation of the Jacobian. Monte Carlo methods [KW86] are a standard approach to solve probabilistic problems. For considering the correlation between adjacent grid points, we do a simultaneous Gaussian fitting at a grid point and its neighbors. Instead of considering the n-dimensional vectors $\mathbf{v}_{i,j,k}$, we consider the $5n$-dimensional vectors $\bar{\mathbf{v}}_{i,j,k}=(\mathbf{v}_{i,j,k},\mathbf{v}_{i-1,j,k},\mathbf{v}_{i+1,j,k},\mathbf{v}_{i,j-1,k},\mathbf{v}_{i,j+1,k})^T$ for $k=1,...,m$. For them, we apply a $5n$-dimensional Gaussian fitting

$$\rho(\mathbf{x}_{i,j},\bar{\mathbf{v}})=\mathcal{N}(\bar{\mathbf{m}}_{i,j},\bar{\mathbf{C}}_{i,j}) \tag{6.1}$$

where $\bar{\mathbf{m}}_{i,j}$ is the $5n$-dimensional median and $\bar{\mathbf{C}}_{i,j}$ is the $5n\times 5n$ covariance matrix. Note that (6.1) contains both the distributions of the velocity and Jacobian at (i,j).

Fig. 6.2 illustrates the application of (6.1) to the same example as in Fig. 6.1. Fig. 6.2a shows the vectors $\bar{\mathbf{v}}_{i,j,k}$ as points distributed over the 5 adjacent grid points; the red curves denote the marginal distributions. The Gaussian distribution is computed by (6.1) and is shown by the blue point cloud and the blue curves. The coincidence of the red and blue curves shows the correctness of the assumption of Gaussian distribution of $\bar{\mathbf{v}}_{i,j,k}$. Fig. 6.2b shows the uncertain acceleration. The red point clouds and curves are identical to Fig. 6.1b. The blue point clouds and curves show the distribution of the acceleration by using (6.1). Here we can clearly see the coincidence of the red and blue curves, which means that our correlated Gaussian model



a)

b)

Figure 6.2: Uncertain velocity and acceleration at grid point $(60, 50)$ of the PIV data set using correlated Gaussian distributions. a) sample vectors $\mathbf{v}_{i,j,k}$ as red dots and marginal density distribution as red curves; correlated Gaussian reconstruction and marginal density distribution as blue point cloud and curve; b) sampled acceleration and marginal distribution as red point cloud and curves; Monte Carlo sampling of acceleration by using correlated Gaussian at grid points as blue point cloud and curves: the red and blue curves coincide.

correctly reproduces the uncertain acceleration (the green curves show the result of the uncorrelated distribution from Fig. 6.1b for comparison.)

So far, we have an approach which can correctly deal with uncertain derived fields containing only the velocity and the Jacobian. Fortunately, many common vortex concepts fall into this category. We are now ready to discuss them in detail.

6.2 Uncertain Vortex Criteria

At first, we want to define a vortex in an uncertain vector field. Even in the certain case several definitions exist. In general, it is described by a swirling motion of a fluid around a core line[RC91, Por97]. In the uncertain case such a well defined core line does not exist. Here we have to deal with density distribution functions that express the movement of particles inside the flow. For the movement of such particle density functions we can only compute probabilities that particles of this distribution move around a core line. The core line itself is not a distinct line structure. We can compute such a line only for one sample of the uncertain vector field. The overall result is a probability for the existence of a vortex core in a small region.

Now we consider the results of vortex detectors. There are two categories: line-based and region-based vortex criteria. In the certain case vortex core line detectors generate a binary decision whether there is a core line at a certain location or not. Vortex region extractors compute scalar fields that describe the strength of the vortical motion. Level sets on this field are defined representing vortical structures. For the computation of vortical structures in uncertain vector fields the types of resulting fields change. Computing vortex cores of such fields results in a probability for the occurrence of a vortical motion at a given location. For uncertain vortex region detectors the result is a 1D density distribution function describing the probabilities of values representing the strength of vortical motion at every location.

We will adapt Q and λ_2 criteria as examples for vortex regions and the method proposed by Sujudi and Haimes using the Parallel Vectors operator, as examples for vortex cores detectors, to uncertain vector fields.

6.2.1 Vortex Cores

We will apply the Parallel Vectors operator, explained in section 2.2, to uncertain vector fields. For this we have to compute the probability that two uncertain vectors are parallel. A second condition of the Parallel Vectors operator is that vortex cores only exist where the Jacobian of the vector field has two imaginary eigenvalues. In section 6.1 we have shown that this is not possible in a closed form. Therefore, we use a Monte Carlo method that calculates the probability of a vortex core line in a cell of a uniform data grid. The support region for discrete vortex core line computation is shown in Fig. 6.3 (left). It contains 32 data points. Thus, every data point of the uncertain vector field consists of a 96D mean vector and a 96×96 dimensional covariance matrix. We generate N samples of the uncertain vector

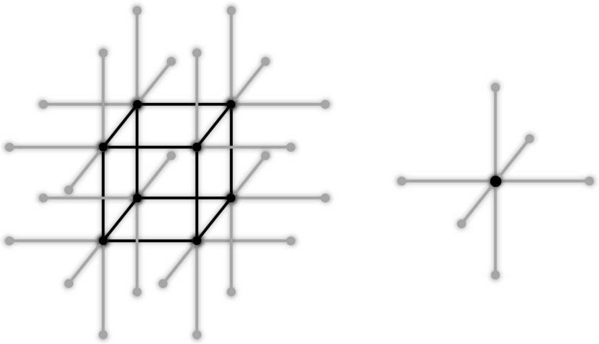


Figure 6.3: (left) Scheme of the support regions for discrete vortex core line computations in 3D space. The cell we want to evaluate is colored black. (right) Scheme of the support region of vortex region.

field. From these sample vectors we compute the acceleration vectors $\mathbf{a}_s$ at the cell nodes. Vortex core lines do not consist of isolated points where the vectors of both fields are parallel, they are continuous lines that cross cells. So we need to compute parallel vectors on the boundary of each cell. As described by Roth and Peikert[RP98], we assume linear interpolation on all triangles of the cell boundary to get an analytic solution for the parallel vectors computation on the boundary. The probability for the occurrence of a vortex core line inside the cell is the relative frequency of the sampled Parallel Vectors operator on its boundary faces. It is given by:

$$P_{SH} = \frac{1}{N} \sum_{s=1}^{N} \begin{cases} 1 & if\, \mathbf{a}_s \parallel \mathbf{v}_s \;\wedge\; \#(\lambda_{im}(\mathbf{J}_s)) = 2 \\ 0 & otherwise \end{cases} \tag{6.2}$$

with N as the number of samples. At least two faces have to contain a vortex core that the cell is marked. The high-dimensional Gaussian distribution described by equation (6.1) is used to generate samples of the uncertain vector field including $\mathbf{v}_s$. Using these samples $\mathbf{J}_s$ and $\mathbf{a}_s$ are computed.

6.2.2 Vortex Regions

In section 2.2 we described the two region based vortex criteria Q and λ_2. To apply these criteria to uncertain vector fields we use a Monte Carlo method that samples the uncertain vector field, computes for each sample the Jacobian and the respective criteria. In this case, we need a support region containing the six direct neighbors of the node we want to evaluate and the node itself (Fig. 6.3(right)). Therefore, the uncertain vector field is defined by 21D mean vectors and 21 × 21 covariance matrices. This results in a 1D histogram of the distribution of the criteria for each data point. The challenge is the visualization. In the certain case vortical structures are enclosed by level sets of the criteria. For the uncertain case we search for a similar visualization. We can visualize the probability that the Q criterion is larger and the λ_2 criterion is smaller than a certain threshold:

$$P(Q > t) = \frac{1}{N} \sum_{s=1}^{N} \begin{cases} 1 & if\, Q_s > t \\ 0 & otherwise \end{cases} \tag{6.3}$$

$$P(\lambda_2 < t) = \frac{1}{N} \sum_{s=1}^{N} \begin{cases} 1 & if\, \lambda_{2_s} < t \\ 0 & otherwise \end{cases} \tag{6.4}$$

with N as number of samples generated from the Gaussian function described in equation (6.1), Q_s and λ_{2_s} are the vortex criteria based on these samples. The visualization shows the probability that at a location with a probability $P > 0$ is enclosed by a level set. If there is no uncertainty, our method returns the same results as the approaches for vector fields without uncertainty.

6.3 Implementation

We implemented our uncertain vortex framework in C# using the Task Parallel Library for parallelization. The input of our method is a number of vector fields (measured or simulated) describing the same flow phenomenon. These vector fields are given on uniform grids. Using only one vector field corresponds to the certain case. The results of our methods are scalar fields that represent probabilities of vortex criteria. All computations are done on a small local region and are repeated multiple times. This makes our method easy to parallelize using parallel for-loops.

For each local operation a support region is defined. For these regions we generate multivariate Gaussian distributions of the input vectors. These distribution functions represent the uncertain vector field. Samples of such distribution functions are created by a pseudo random generator. A uniformly distributed vector is generated (same dimension as the mean vector of the Gaussian of the support region). To this vector we apply a Box Muller filter that transforms it into a Gaussian distribution. After that, we multiply it with the Eigenvector matrix of the covariance matrix (Eigenvectors are scaled by their Eigenvalues) and add the mean vector. Finally, we split the large vector into 3D vectors according to the nodes of the support region.

With this sample vector set we can compute all vortex criteria. We repeat the sampling and the vortex criteria computation a few hundred times. For region-based vortex criteria we count the number of sample sets with a vortex criterion larger or smaller than a given threshold. Concerning vortex core lines we count the number of sample sets for which the Parallel Vectors operator returns true. The final result is the relative frequency of the positive events.

6.4 Results

To test our approach we apply our methods to two real-world examples, a flow around a cylinder and a flow field from a climate simulation. All results are generated with a laptop containing an Intel i7 2820QM with 4 cores and HT and 16GB

RAM. All data sets were provided by domain experts, who gave positive feedback on the results.

6.4.1 Flow around a Cylinder

In this example we have got four simulations of a flow around a cylinder with different Reynolds-numbers. The Reynolds numbers are between 290 and 320 based on the free stream velocity and the diameter of the cylinder. The flow should be laminar at these Reynolds numbers, but should already show three-dimensional structures. Therefore, only three-dimensional configurations are retained. The time-dependent computations have been performed for 100 seconds physical time. Every 50th time step (every second physical time) is stored for further analysis. The numerical computations have been performed using the open-source software package OpenFOAM 1.6 using finite volume discretization. Block-structured grids are applied using 645,120 hexahedral elements refined at the cylinder wall. A constant inlet velocity boundary condition is considered at the inlet, pressure outlet at the outlet and symmetry conditions are chosen for the top, bottom and side boundaries. No-slip boundary condition is employed on the cylinder wall. The size of the domain is selected at least 20 times the cylinder diameter to eliminate the reflection on the boundaries.

We resampled the data set to a uniform grid of the domain $[-1,11] \times [-3,3] \times [-3,3]$ with a resolution of $200 \times 100 \times 100$. Then we applied our techniques to this data set. As a helpful orientation all of our result images show LIC of the corresponding mean field in the background. At first we started with the region-based techniques illustrated in Fig. 6.4. Here, from left to right the time steps 98 to 100 are shown. The first row gives an overview of the uncertainty in the data set. In the second row we illustrate the results of our uncertain λ_2 vortex detector. These images show isosurfaces for $P(\lambda_2 < -0.003) = 0.05$ and 0.95. The next two rows show analogue results of the uncertain Q criterion for $P(Q > 0.003)$. Similar to the certain setting probabilities for Q and λ_2 level sets are correlated. Also regions of strong uncertainty correlate with these level sets. Results of the Parallel Vectors operator are shown in Fig. 6.5 for the same time steps. In the volume renderings and cross-sections we see the probabilities for the occurrence of vortex cores. The extracted volumes where the probability for vortex cores is larger than zero are much smaller than the volumes extracted by the region-based methods. High probabilities only appear at the vortex cores behind the cylinder. The rest of the extracted structures has a rather small probability, but larger volumes.

For time step 100 we make a more detailed analysis. We show a comparison of the λ_2 and Q criteria to the input fields (shown as colored isolines) and the mean field (black isolines) in Fig. 6.6. Regions with high probability correspond to the average level sets and regions with lower probability capture almost all regions of the level sets of the input fields. In Figure 6.7 we compare our vortex core extractor with the vortex cores of the input fields and the mean vector field. Here, also the level set with 10% probability corresponds to the vortex cores of the mean vector field, and the cores of the input fields are almost captured by regions with lower probability.

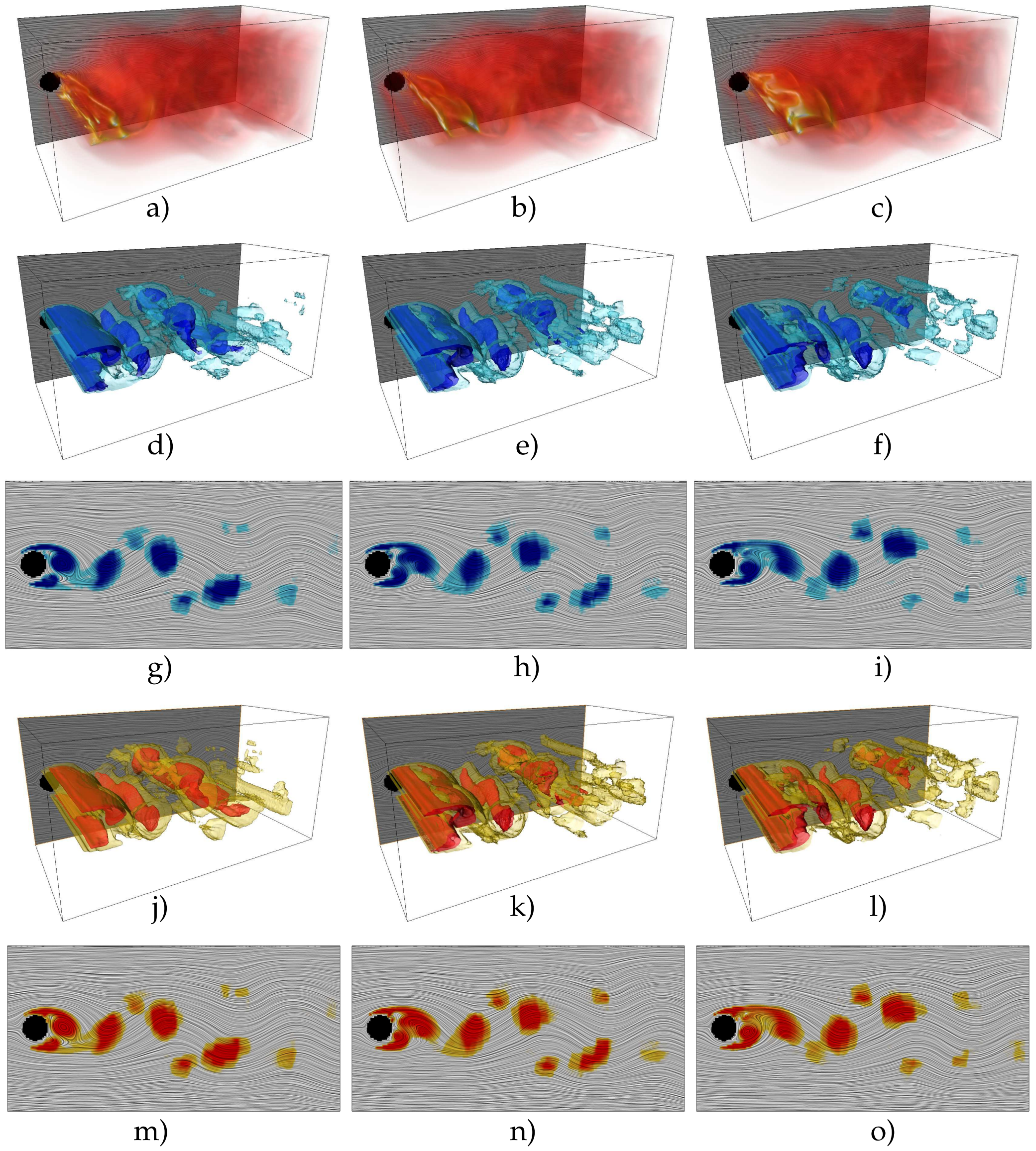


Figure 6.4: Cylinder data set: **(a,d,g,j,m)** time step 98, **(b,e,h,k,n)** time step 99, **(c,f,i,l,o)** time step 100, **(a,b,c)** volume rendering of the maximal standard deviation, **(d,e,f)** isosurfaces of the probability field $P(\lambda_2 < -0.003)$ with iso values 0.05 and 0.95, **(g,h,i)** cross section of the probability field $P(\lambda_2 < -0.003)$ at $z = 0$ **(j,k,l)** isosurfaces of the probability field $P(Q > 0.003)$ with levels 0.05 and 0.95, **(m,n,o)** cross section of the probability field $P(Q > 0.003)$ at $z = 0$.

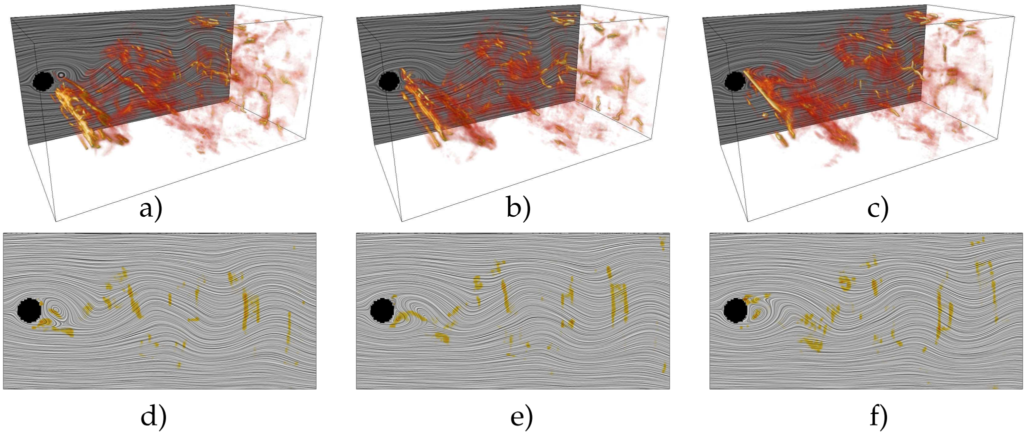


Figure 6.5: Cylinder data set: **(a,d)** time step 98, **(b,e)** time step 99, **(c,f)** time step 100, **(a,b,c)** volume rendering of the of the vortex core probability field, **(d,e,f)** cross section of this field at $z = 0$.

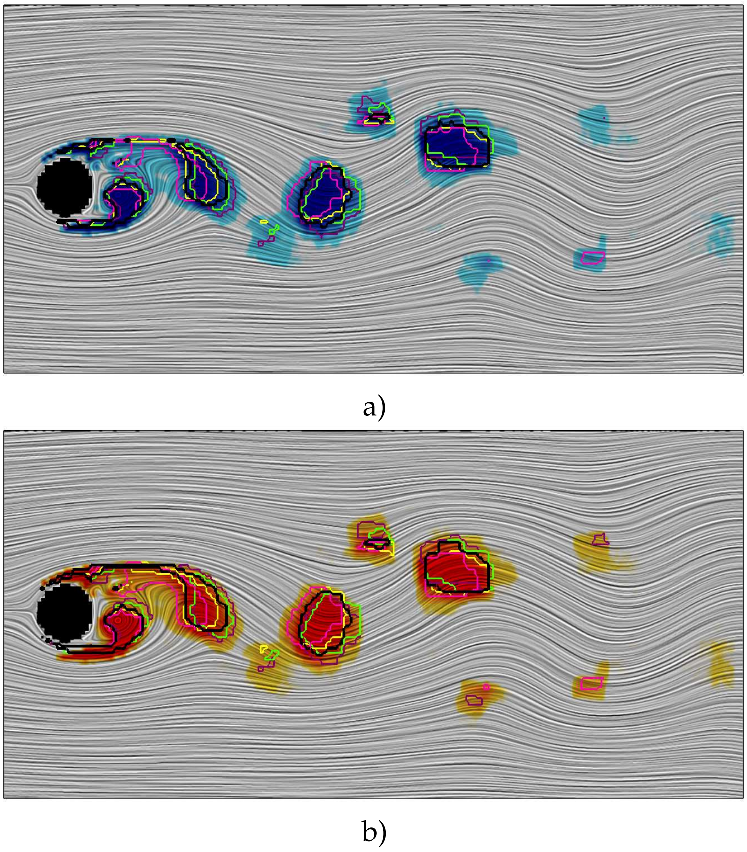
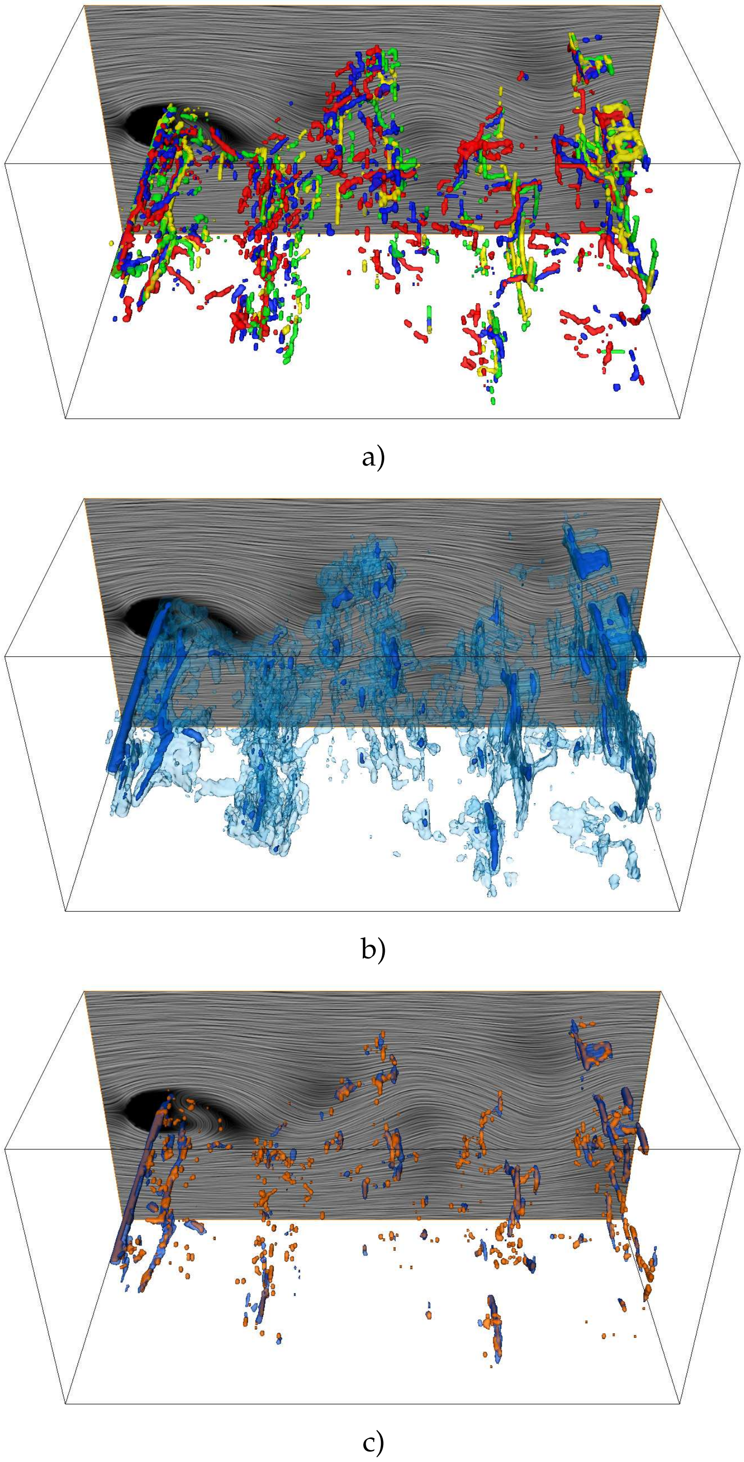


Figure 6.6: Cylinder data set time step 100: **(a)** cross section of the probability field $P(\lambda_2 < -0.003)$ and **(b)** $P(Q > 0.003)$ at $z = 0$ compared with isolines of the input data.



a)

b)

c)

Figure 6.7: Cylinder data set time step 100: **(a)** vortex cores of the input vector fields , **(b)** isosurfaces of $P(\mathbf{a}_\rho \parallel \rho) \geq 0.01$ (light blue) and 0.1 (blue), **(c)** isosuface 0.1 (blue) with vortex cores of the mean vector field (orange).

The overall computation time for λ_2, Q and vortex cores is 3 hour and 40 minutes for each time step, using 200 samples for each data point.

6.4.2 Ocean Ensemble Simulation

The ocean simulation data is part of the CMIP5 runs (Coupled Model Intercomparison Project). The model itself is the MPI-OM ocean model which was developed by the Max-Planck Institute for Meteorology in Hamburg. The simulation was carried out on a tri-polar curvilinear grid with a horizontal resolution of 1.5 degree at 40 height levels. Prior to the analysis, the data was resampled to a rectilinear grid. The data set consists of just three scalar variables (uko, vke, wo) that describe the ocean currents. The simulation was performed using an ensemble run, and all together, 10 ensembles with the monthly mean of the currents were used.

Clearly visible in all results (see Fig. 6.8) is the Antarctic circumpolar current, as well as the equatorial currents in the Pacific ocean and the gulf stream and the north Atlantic current in the Atlantic ocean. Especially the circumpolar current which is rich in vortices is detected very well using all criteria. With a higher resolution simulation, probably also smaller features, such as the Kuroshio in Japan, and the Agulhas current at the south-eastern tip of Africa can be detected. The overall computation time for this data set is 1 hour and 16 minutes using 200 samples per grid point.

6.4.3 Measured PIV Data Set

Here we used the same data set we introduced in section 5.12.2. We applied the region-based vortex criteria to this data set. Fig. 6.9a illustrates the probability that the Q criterion has a larger isovalue than zero, while Fig. 6.9b shows the probability of the λ_2 criterion for isovalues smaller than zero. For comparison Fig. 6.9c shows the Q criterion and Fig. 6.9d the λ_2 criterion of the mean vector field. The probability fields contain only values less than 100%. Also, the results of the mean vector fields do not always correlate with the probability fields. That means there are some configurations that locally differ completely from the average field. For example, the λ_2 criterion of the mean field has some large positive values above the step, while the probability computed by our method shows clearly that there are values smaller than zero. The overall computation time for this data set was 8 minutes and 20 seconds. We used 10,000 samples for each data point.

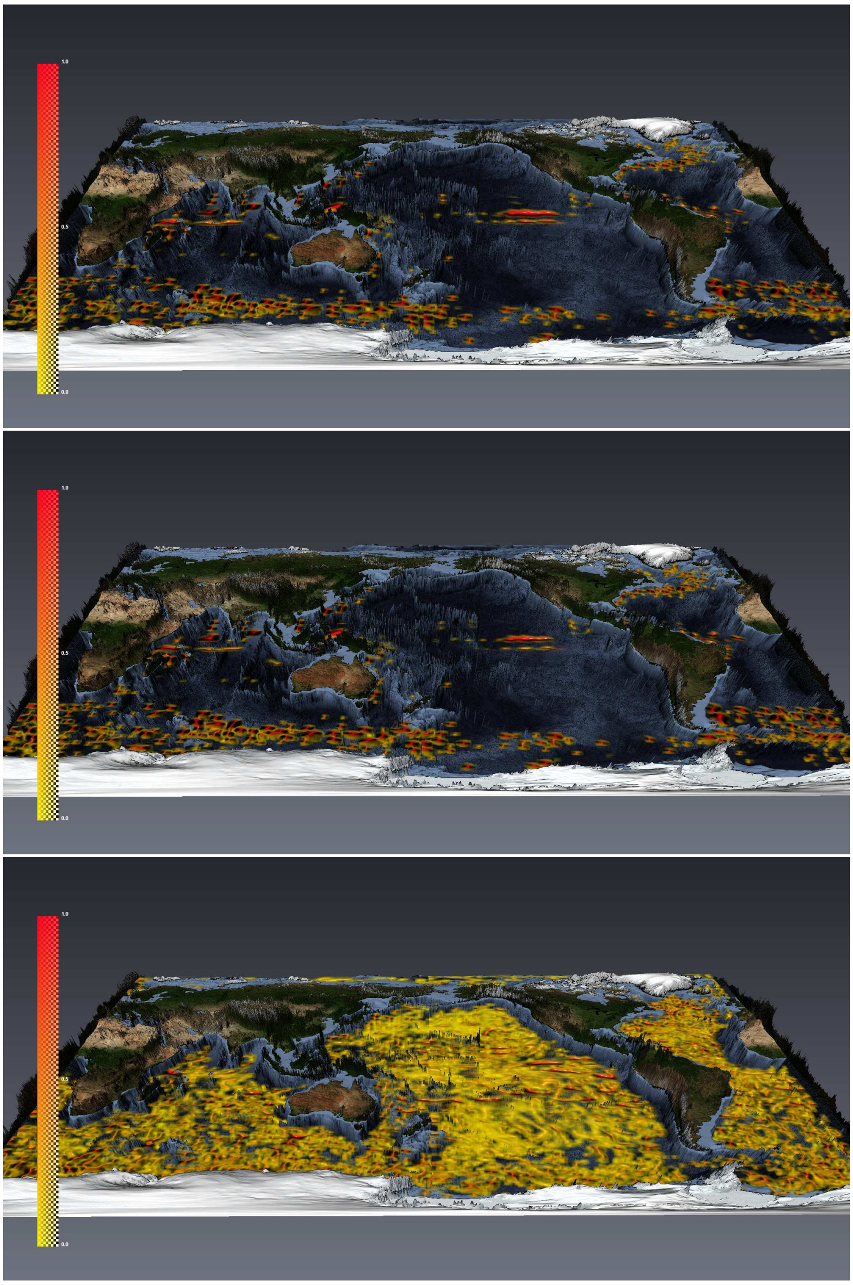


Figure 6.8: CMIP5 data set: **(top)** showing $P(\lambda_2 < -0.003)$, **(middle)** $P(Q > 0.003)$, and **(bottom)** the probability of vortex cores.

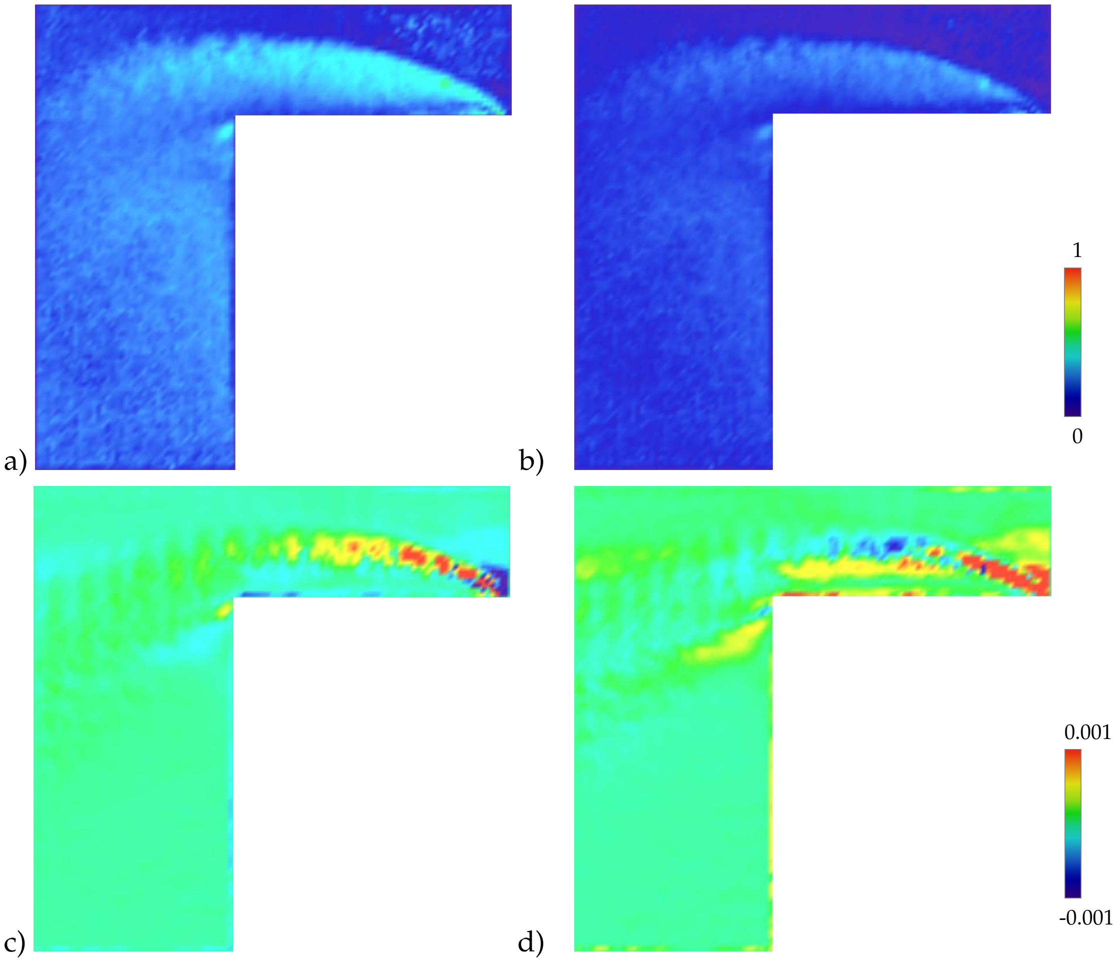


Figure 6.9: Flow around a backward facing step: **(a)** $P(Q > 0)$, **(b)** $P(\lambda_2 < 0)$, **(c)** Q criterion of the mean vector field, **(d)** λ_2 criterion of the mean vector field.

7 Lagrangian Coherent Structures with Guaranteed Material Separation

This chapter deals with features in unsteady vector fields. However, approaches from vector field topology do not apply well to unsteady vector fields because the meaning of stream lines in time-dependent flows is limited. Here, the features are mainly related to the Lagrangian viewpoint. This has lead to the concept of Lagrangian Coherent Structures (LCS) which define regions of coherent flow behavior. As the name implies, these structures are advected in the flow. Thus, there is zero flux across these structures.

A common way to find LCS is to compute the Lyapunov Exponent (LE) and find its ridges [Lia66]. The LE characterizes the rate of separation of infinitesimally close trajectories. While the LE is computed for infinite time, we have to deal with data sets of finite time in practice. This has lead to the notion of Finite Time Lyapunov Exponents (FTLE), where the LE is computed over fixed time intervals [Hal01].

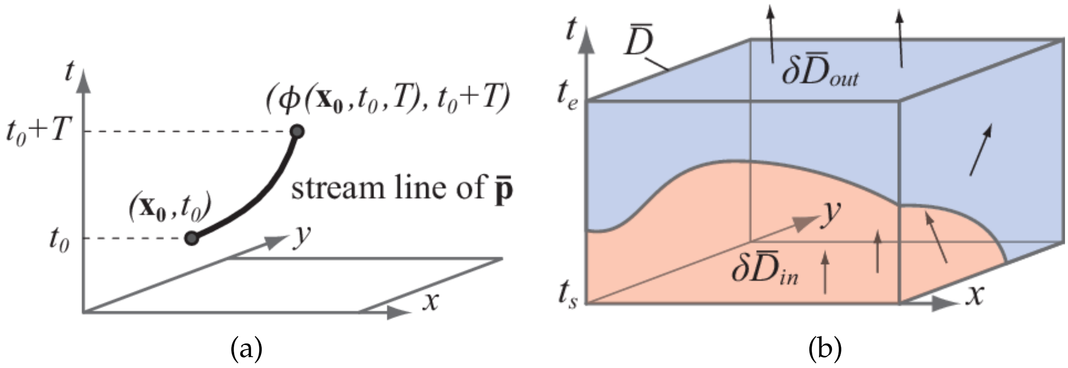
It has been shown that the ridges of FTLE fields approximate LCS [Hal01]. However, FTLE ridges are not exact material structures and therefore deviate from LCS, as we analyze in section 7.2 using a number of counterexamples.

In this chapter, we develop a modification of the FTLE method in order to find separating structures that are guaranteed material structures. Analogously to the FTLE method, we compute a scalar field that describes LCS, which we call Material Separation Field (MSF). Using this scalar field, we introduce an appropriate definition for separating structures, which leads to material structures. We achieve the material separation by incorporating all available time steps to compute the scalar field. With our method, we have zero flux across separating structures and path lines always have constant MSF values. Therefore, separating structures found in one time step can easily be integrated to other time steps.

7.1 Notation

In this section, we want to characterize our input data and clarify the notation used throughout this chapter.

Given is an n-dimensional ($n = 2{,}3$) time dependent vector field $\mathbf{v}(\mathbf{x},t)$ over the space-time domain $\bar{D} = D \times [t_s,t_e]$ where $D \subset \mathrm{I\!R}^n$ is a compact and closed set acting as the spatial domain of $\mathbf{v}$ and $[t_s,t_e]$ is the time interval on which $\mathbf{v}$ is given.



(a) (b)

Figure 7.1: (a) The stream line integration of $\bar{\mathbf{p}}$ starting from $(\mathbf{x}_0, t_0)$ over the integration time T ends at $(\phi(\mathbf{x}_0, t_0, T), t_0 + T)$. (b) Example of domains of a 2D flow: D is rectangular, $\bar{D}$ is a (volumetric) box, $\delta\bar{D}$ is the surface of the box, $\delta\bar{D}_{in}$ denotes regions of inflow, $\delta\bar{D}_{out}$ denotes regions of outflow.

Furthermore, let δD be the boundary of D. Then the boundary of the space-time domain is

$$\delta\bar{D} = D \times \{t_s, t_e\} \;\cup\; \delta D \times [t_s, t_e]. \tag{7.1}$$

We use the notation $\lambda_i(\mathbf{M})$ and $\mathbf{e}_i(\mathbf{M})$ for the i-th eigenvalue and corresponding eigenvector of a matrix $\mathbf{M}$, and $\lambda_{max}(\mathbf{M})$ for its maximal eigenvalue. Furthermore, let $\mathbf{M}^*$ be the transpose of $\mathbf{M}$.

The *flow map* $\phi_t^T(\mathbf{x}) = \phi(\mathbf{x}, t, T)$ of $\mathbf{v}$ is defined as the location of a particle seeded at $(\mathbf{x}, t)$ after a path line integration of $\mathbf{v}$ over a time interval T. Given the spatial gradient

$$\Delta = \Delta(\mathbf{x}, t, T) = \frac{d\phi(\mathbf{x}, t, T)}{d\mathbf{x}} \tag{7.2}$$

of ϕ, the FTLE is computed as follows:

$$\mathrm{FTLE}(\mathbf{x}, t, T) = \frac{1}{T} \ln \sqrt{\lambda_{max}(\Delta^* \cdot \Delta)} \tag{7.3}$$

In order to simplify some concepts, we can interpret the n-dimensional unsteady vector field $\mathbf{v}$ as a steady $(n+1)$-dimensional vector field

$$\bar{\mathbf{p}}(\mathbf{x}, t) = \begin{pmatrix} \mathbf{v}(\mathbf{x}, t) \\ 1 \end{pmatrix}. \tag{7.4}$$

(The bar denotes that $\bar{\mathbf{p}}$ is a $(n+1)$-dimensional vector field in $\bar{D}$.) It is known that path lines of $\mathbf{v}$ correspond to stream lines of $\bar{\mathbf{p}}$ [TWHS05]. For $\bar{\mathbf{p}}$ we define the $(n+1)$-dimensional flow map

$$\bar{\phi}(\mathbf{x}, t, T) = (\phi(\mathbf{x}, t, T)\,,\, t + T) \tag{7.5}$$

Fig. 7.1a gives an illustration for $n = 2$.

The field $\bar{\mathbf{p}}$ segments the boundary of the space-time domain $\delta\bar{D}$ into areas of inflow and outflow:

$$\begin{aligned}\delta\bar{D}_{in} &= \{(\mathbf{x},t) \in \delta\bar{D} : \exists\, \epsilon_0 > 0 \;\; \forall\epsilon \in (0,\epsilon_0) : (\mathbf{x},t) + \epsilon\bar{\mathbf{p}}(\mathbf{x},t) \in \bar{D}\} \\ \delta\bar{D}_{out} &= \{(\mathbf{x},t) \in \delta\bar{D} : \exists\, \epsilon_0 > 0 \;\; \forall\epsilon \in (0,\epsilon_0) : (\mathbf{x},t) + \epsilon\bar{\mathbf{p}}(\mathbf{x},t) \notin \bar{D}\}\end{aligned}$$

Obviously, $D \times \{t_s\} \subseteq \delta\bar{D}_{in}$ and $D \times \{t_e\} \subseteq \delta\bar{D}_{out}$. Fig. 7.1b gives an illustration.

7.2 FTLE Ridges are not Material Lines

Experience has shown that ridges of FTLE coincide in many cases well with material structures. However, in general they are not exact material structures, no matter which ridge definition of a scalar field we use. Unless very long integration times are used, FTLE ridges can deviate considerably from material structures. To show this, we now present three examples where the FTLE ridges significantly differ from material structures. We start with two synthetic data sets, before we analyze the flow in a real data set.

7.2.1 Vanishing Ridges

Let $(\mathbf{x}_0, t_0) \in \bar{D}$ be a point on a ridge of FTLE$(\mathbf{x}, t, T)$. There is only a certain part of $\bar{D}$ which has influence on FTLE$(\mathbf{x}_0, t_0, T)$: only the domain $D \times [t_0 - T, t_0 + T]$ is involved. If the ridge is a material structure, then the complete path line $\phi(\mathbf{x}_0, t_0, T)$ for any T must be on the ridge. We construct a simple counterexample where this is not the case. We define a flow field $\mathbf{w}$ as

$$\mathbf{w}(\mathbf{x},t) = \begin{cases} \mathbf{v}(\mathbf{x},t) & \text{for} \quad t_0 - T \le t < t_0 + T \\ \frac{t_0+T+\epsilon-t}{\epsilon}\, \mathbf{v}(\mathbf{x}, t_0 + T) & \text{for} \quad t_0 + T \le t < t_0 + T + \epsilon \\ \mathbf{0} & \text{for} \quad t_0 + T + \epsilon \le t < t_0 + 3T + \epsilon \end{cases}$$

Note that $\mathbf{w}$ is obtained by a linear blending of an arbitrary vector field $\mathbf{v}$ and the zero vector field $\mathbf{0}$ in the time interval $[t_0 + T, t_0 + T + \epsilon]$ for a certain positive blending time ϵ. Then $(\mathbf{x}_0, t_0)$ is an extremal point of the FTLE of $\mathbf{w}$ as well since $\mathbf{v}$ and $\mathbf{w}$ coincide in a sufficiently large neighborhood of $(\mathbf{x}_0, t_0)$. Let $\bar{\phi}_{\mathbf{w}}$ be the flow map of $\mathbf{w}$ and let $(\mathbf{x}_1, t_1) = \bar{\phi}_{\mathbf{w}}(\mathbf{x}_0, t_0, 2T + \epsilon)$ be a point on the path line of $\mathbf{w}$ starting from $(\mathbf{x}_0, t_0)$. Since $\mathbf{w}(\mathbf{x}, t) = \mathbf{0}$ in a sufficiently large neighborhood of $(\mathbf{x}_1, t_1)$, no measure will detect it as part of a ridge. Hence, the ridge from $\mathbf{w}$ cannot include the path line starting from $(\mathbf{x}_0, t_0)$. Fig. 7.2 gives an illustration. Over time, the FTLE values on the path line decrease, until they reach zero at $t = 2T + \epsilon$. Therefore, the FTLE ridge terminates before the path line leaves $\bar{D}$. In contrast, our method produces separating structures that include the full path lines.

7.2.2 Laminar Speed Change

Even in cases where the ridges appear as distinct line-like features, there can be a high amount of flux crossing them. We show this in the following two data sets.

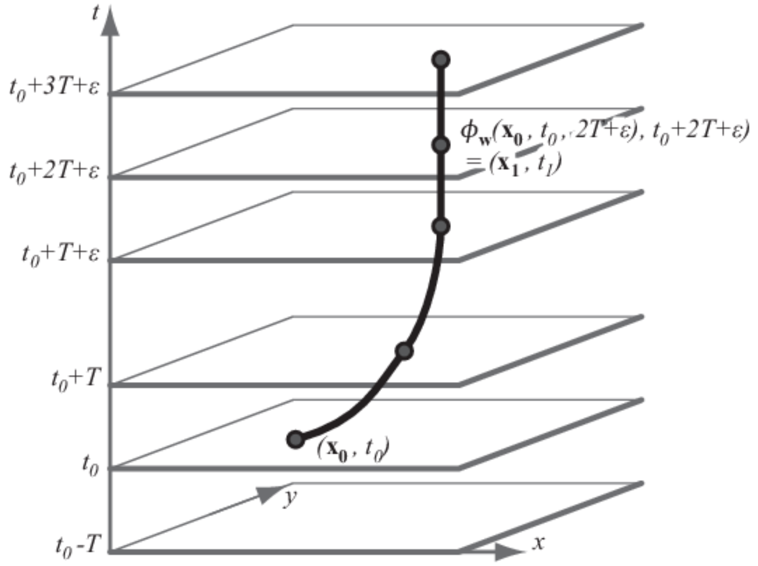


Figure 7.2: Construction of a field $\mathbf{w}$ such that $(\mathbf{x}_0, t_0)$ is on a ridge but $(\mathbf{x}_1, t_1)$ is not. Hence the ridge is not a material structure.

First, we consider a steady field describing a laminar flow in $D = [0,10] \times [0,2]$ which is defined as

$$\mathbf{v}(x,y,t) = \mathbf{v}(x,y) = \begin{pmatrix} 1 + f(x) \\ 0 \end{pmatrix} \tag{7.6}$$

with

$$f(x) = \begin{cases} \int_{4.5}^{x} b(t)dt & \text{for} \quad 4.5 \leq x \leq 5.5 \\ 0 & \text{else} \end{cases} \tag{7.7}$$

and $b(t)$ is a quintic Bézier function over the domain [4.5,5.5] defined as

$$b(t) = \sum_{i=0}^{5} b_i \, B_i^5 \left(\frac{t - 4.5}{5.5 - 4.5} \right) \tag{7.8}$$

with $(b_0, ..., b_5) = (0,0,5,-5,0,0)$ and B_i^5 are the Bernstein polynomials. In $x = (4.5, 5.0)$, the flow first accelerates and in $x = (5.0, 5.5)$, it decelerates. These speed changes result in positive FTLE values at vertical bars across D (see Fig. 7.3a). Clearly, these bars indicate vertical FTLE ridges. Over time, the FTLE field stays constant, because the field is steady and T is constant. Therefore, the positions of these ridges do not change. However, all path lines cross these ridges, resulting in an arbitrary large flux across the ridges.

If we enlarge T, the ridges move and become weaker (see Fig. 7.3b). However, they are still constant in time. Only for $T \to \infty$, the ridges will disappear. With our method, the separating structures move with the flow, so that no path line crosses them (see section 7.2.2).

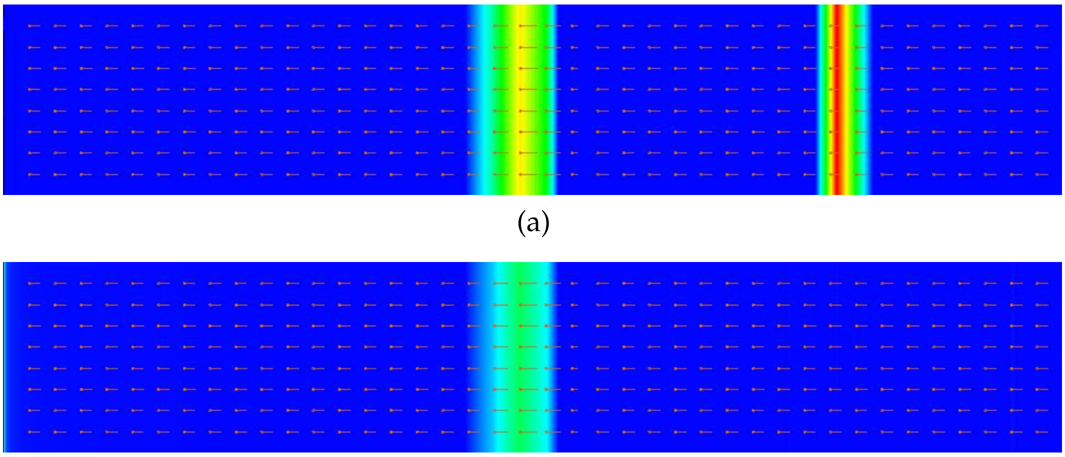


Figure 7.3: Vertical ridges in the laminar speed change example: (a) FTLE field for $T = 10$, (b) FTLE field for $T = 20$.

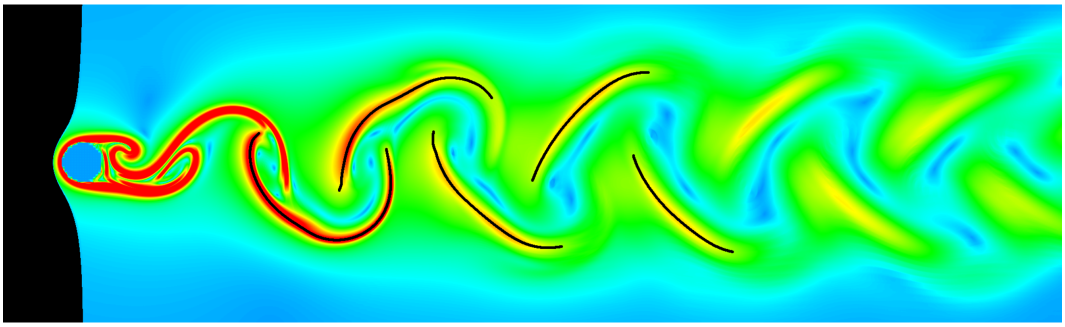


Figure 7.4: FTLE field of the cylinder data set. The black lines indicate the ridges that we analyzed for cross flux.

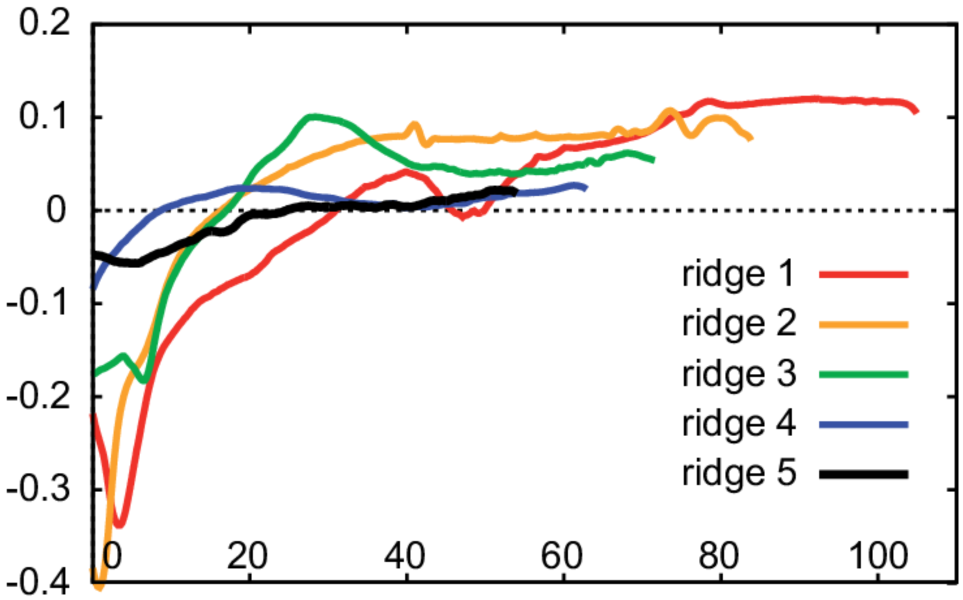


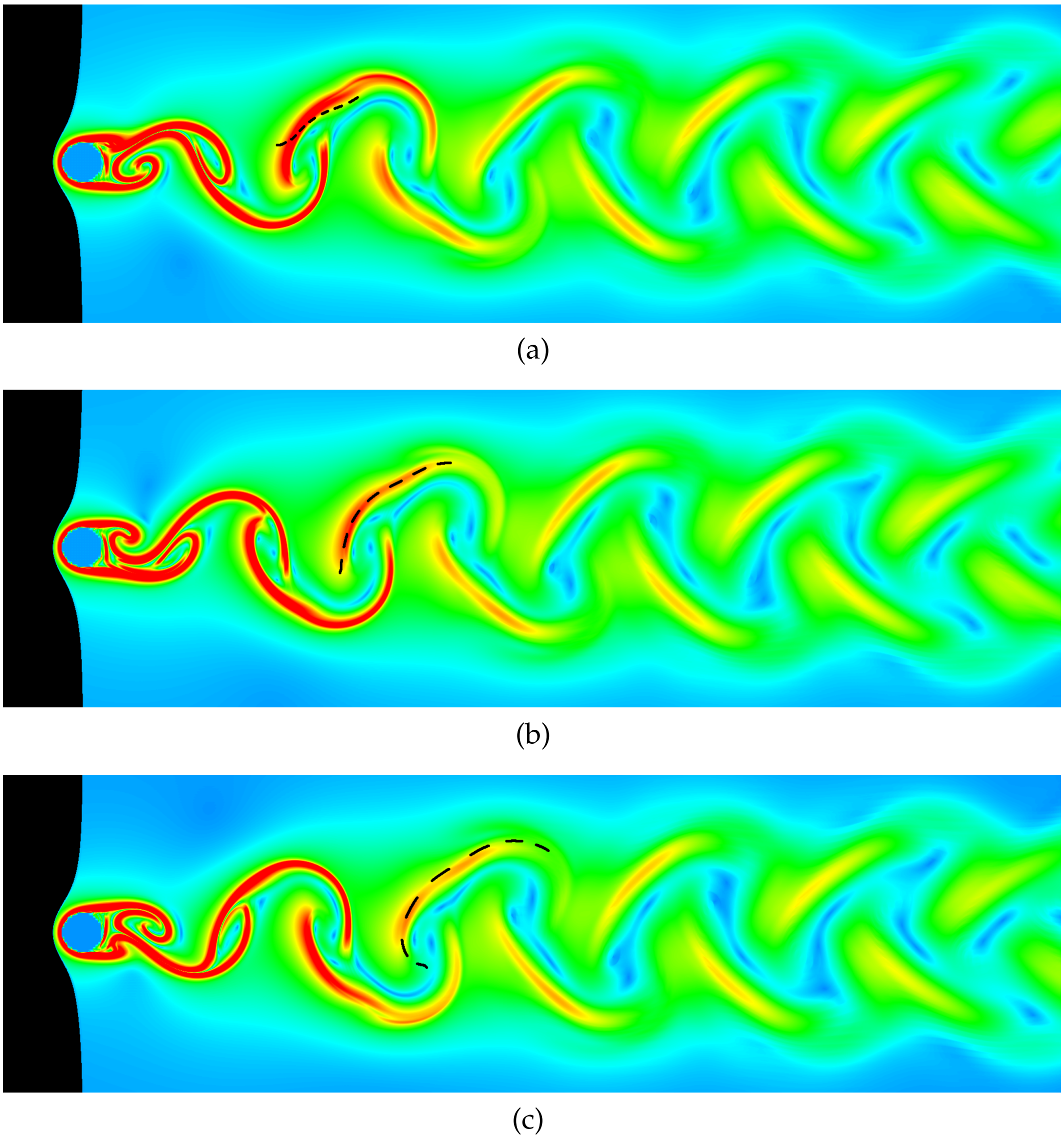
Figure 7.5: Percentage of the flux crossing the five ridges from Fig. 7.4, plotted against ridge arc length.

7.2.3 Cylinder Flow

For the cylinder flow dataset (described in section 7.4.2), we computed the cross flux at some of its FTLE ridges. We selected a few local maxima of FTLE and extracted the associated height ridges using the parallel vectors method [PS08]. The FTLE field and the selected ridges are shown in Fig. 7.4. We use the common heat map technique discussed in section 7.3.3 for the visualization of the fields. For the FTLE computation we used backward integration of $T = 80$, where the data set has a total time span of 480. As we found the ridges to be distinct and clearly visible, we assume that T is long enough.

At a set of sample points per height ridge the flux per unit length was then computed by taking the velocity component orthogonal to the height ridge and subtracting the speed of the moving ridge in this orthogonal direction. The motion of the ridge was estimated by extracting height ridges from FTLE fields computed at a few earlier and later points in time (and verifying independence of the temporal sampling rate). Following Shadden et al. [SLM05], we divided this flux rate by the local velocity magnitude. This results in the percentages plotted in Fig. 7.5, which show relatively high values for the cross flux, ranging from -40 to over 10 percent. In Fig. 7.6 we extracted particles on one ridge and integrated them in the vector field for $T = 32$ and $T = -32$. The figure shows that the particles clearly deviate from the ridge and therefore cross it.

By applying this flux computation also to the "double gyre" field (see section 7.4.1), we were able to confirm Shadden's value of about 0.05% for the (rather long) integration time $T = 30$, while for $T = 3$ we obtained similar cross flux percentages as for the cylinder flow example.



(a)

(b)

(c)

Figure 7.6: Flux across FTLE ridges: the black line consists of particles integrated in the vector field, at (a) $T = -32$, (b) $T = 0$ and (c) $T = 32$.

7.3 Material Separation Fields

We now describe our method, which produces separating material structures. The key idea is to define a modification of FTLE fields called "Material Separation Fields" (MSF) such that every point on a path line has the same MSF value, i.e.,

$$\mathrm{MSF}(\mathbf{x},t) = \mathrm{MSF}(\phi(\mathbf{x},t,T),t+T) \tag{7.9}$$

for any integration time T which does not leave $\bar{D}$. Once we have this, an arbitrary point on a path line can serve as representative for computing MSF, and the MSF for all other points on the path line can simply be obtained by advection along $\bar{\mathbf{p}}$.

7.3.1 Parametrization of Path Lines

Since we need only one point per path line as representative, we search for subsets $\bar{P}$ of $\bar{D}$ with the property that every stream line of $\bar{\mathbf{p}}$ intersects $\bar{P}$ in exactly one point. Then the MSF only has to be computed on $\bar{P}$ and from there simply be advected to every location of $\bar{D}$. We call the definition of $\bar{P}$ the *parametrization of path lines*.

Parametrization is a well-studied concept for curves and surfaces, meaning to find an injective map from a subset of $\mathbb{R}^2/\mathbb{R}^3$ to a curve/surface. This map allows a unique addressing of every curve/surface point. Fortunately, for path lines there is a simple solution for the parametrization. Since the last component of $\bar{\mathbf{p}}$ is 1, it is guaranteed that the integration of $\bar{\mathbf{p}}$ starting from any point $(\mathbf{x},t) \in \bar{D}$ will leave $\bar{D}$ in a unique point on $\delta\bar{D}$ for both forward and backward direction. This gives two simple solutions: both $\delta\bar{D}_{in}$ and $\delta\bar{D}_{out}$ can act as the domain of the path line parametrization. Based on this we can even define a parametrization which is based on an arbitrary time slice $t = const$ as

$$\bar{P}_t = \{D \times \{t\}\} \,\cup\, \{(\mathbf{x},s) \in \delta\bar{D}_{in} | s > t\} \,\cup\, \{(\mathbf{x},s) \in \delta\bar{D}_{out} | s < t\}. \tag{7.10}$$

(7.10) has the special cases $\bar{P}_{t_s} = \delta\bar{D}_{in}$ and $\bar{P}_{t_e} = \delta\bar{D}_{out}$. Fig. 7.7 illustrates the path line parametrization of a 1D vector field $u(x,t)$ for which we consider the 2D path line field $\bar{\mathbf{p}}(x,t) = (u(x,t),1)^T$. Fig. 7.14 shows a parametrization in a practical example.

7.3.2 Defining MSF

Given a point $(\mathbf{x},t) \in \bar{D}$, we integrate $\bar{\mathbf{p}}$ until we leave $\bar{D}$ in a point $(\mathbf{x}_{out},t_{out})$ under forward and $(\mathbf{x}_{in},t_{in})$ under backward integration (in the implementation we also stop if one of the particles needed for the gradient computation leaves $\bar{D}$):

$$\begin{aligned} (\mathbf{x}_{in},t_{in}) &= \bar{\phi}(\mathbf{x},t,t_{in}-t) \\ (\mathbf{x}_{out},t_{out}) &= \bar{\phi}(\mathbf{x},t,t_{out}-t) \end{aligned} \tag{7.11}$$

with $(\mathbf{x}_{in},t_{in}) \in \delta\bar{D}_{in}$ and $(\mathbf{x}_{out},t_{out}) \in \delta\bar{D}_{out}$. Note that $t_s \le t_{in} \le t_{out} \le t_e$. Fig. 7.8a gives an illustration.

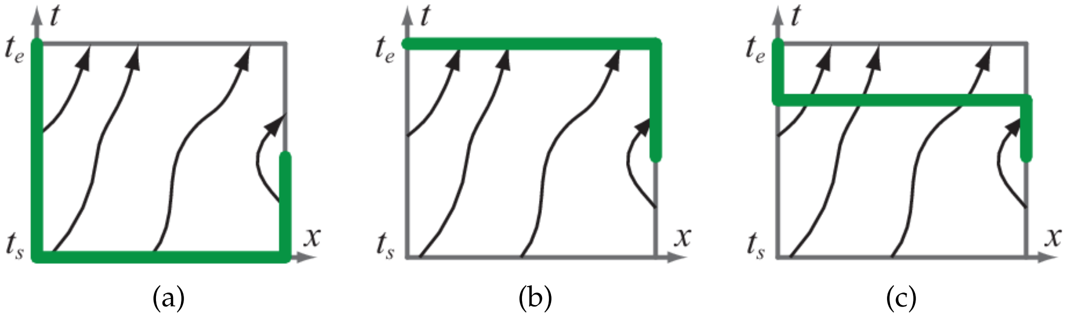
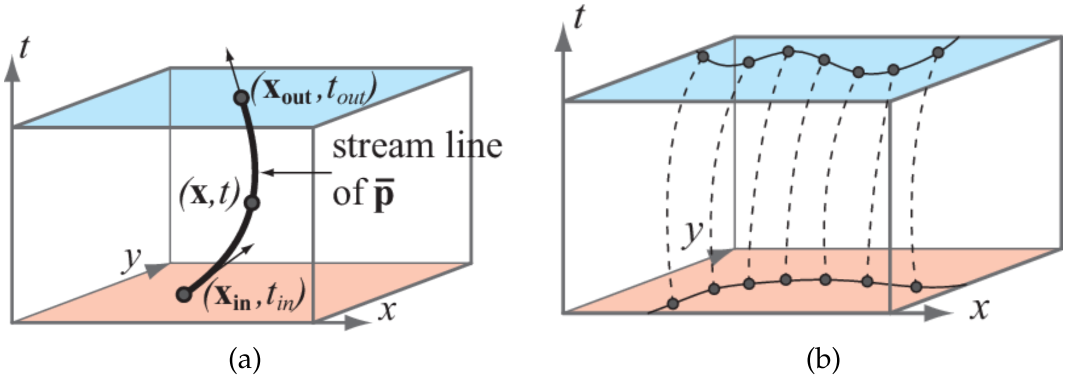


Figure 7.7: Three parametrizations of the path lines of a 1D time-dependent vector field: (a) $\bar{P}_{t_s}$; (b) $\bar{P}_{t_e}$; (c) $\bar{P}_t$ with $t_s < t < t_e$. The green lines are the parametrizations: integrating from them covers the whole domain.



t
$(\mathbf{x}_{\mathbf{out}}, t_{out})$
stream line
of $\bar{\mathbf{p}}$
$(\mathbf{x},t)$
y
$(\mathbf{x}_{\mathbf{in}}, t_{in})$
x
(a)
t
y
x
(b)

Figure 7.8: (a) Integrating $\bar{\mathbf{p}}$ from $(\mathbf{x}, t)$ leaves $\bar{D}$ in $(\mathbf{x}_{in}, t_{in})$ and $(\mathbf{x}_{out}, t_{out})$. (b) Separating structures of MSF on $\delta\bar{D}_{in}$ and $\delta\bar{D}_{out}$ are connected by path lines.

In order to compute $\mathrm{MSF}(\mathbf{x},t)$, we consider the whole path line from $(\mathbf{x}_{in},t_{in})$ to $(\mathbf{x}_{out},t_{out})$. We define

$$\begin{aligned} \Delta_M(\mathbf{x},t) &= \Delta(\mathbf{x}_{in},t_{in},t_{out}-t_{in}) \\ &= (\Delta(\mathbf{x},t,t_{in}-t))^{-1}\cdot\Delta(\mathbf{x},t,t_{out}-t). \end{aligned} \tag{7.12}$$

Therefore, $\Delta_M(\mathbf{x},t)$ expresses the gradient of the flow map from t_{in} to t_{out} at point $(\mathbf{x}_{in},t_{in})$. Note that $\Delta_M(\mathbf{x},t)=\Delta_M(\phi(\mathbf{x},t,T),t+T)$ for any T with $t_{in}\leq t+T\leq t_{out}$, meaning Δ_M stays constant along a given path line.

Now we can compute the forward MSF as

$$\mu_i=\frac{1}{t_{out}-t_{in}}\ln\sqrt{\lambda_i((\Delta_M)^*\cdot\Delta_M)} \tag{7.13}$$

and

$$\mathrm{MSF}_f=\max_{i=1..n}\mu_i. \tag{7.14}$$

In a similar way we can define the backward MSF:

$$\nu_i=\frac{1}{t_{out}-t_{in}}\ln\sqrt{\lambda_i((\Delta_M{}^{-1})^*\cdot\Delta_M{}^{-1})} \tag{7.15}$$

and

$$\mathrm{MSF}_b=\min_{i=1..n}\nu_i. \tag{7.16}$$

There is a simple relation between μ_i and ν_i:

$$\{\mu_i,...,\mu_n\}=\{-\nu_1,...,-\nu_n\}. \tag{7.17}$$

Therefore, $\mathrm{MSF}_f=-\mathrm{MSF}_b$. Because of this equivalence, we only use the forward MSF in the rest of this work.

As can be seen from equations 7.13 and 7.14, the MSF is computed in a similar way to the FTLE (compare to equation 7.3). The FTLE uses a fixed integration time T. In contrast, we incorporate all available time steps from t_{in} to t_{out}. This means, that the forward and backward integration times can differ for different points. However, their sum always equals $t_{out}-t_{in}$. Because Δ_M is constant along a given path line, the MSF is also constant along that path line.

Given the parametrization of path lines as described in section 7.3.1, we now have a simple and efficient way to compute the MSF for all points $(\mathbf{x},t)\in\bar{D}$. We only need to compute the MSF on any parametrization $\bar{P}_t$. The MSF values for other points $(\mathbf{x},t)$ can then be found by looking up the intersection of the corresponding path line with $\bar{P}_t$. Therefore, the MSF of $\bar{D}$ can be regarded as a time-morph of the MSF of $\bar{P}_t$ along the vector field.

7.3.3 Time Period of Interest

Another consequence of constant MSF values along path lines is that our results represent mixtures of structures from forward and backward integration (called unstable and stable manifolds in [SLM05]). Both types of structures can cross and overlap each other. Such crossings are well illustrated in Fig. 7.9a and Fig. 7.11. The corresponding data sets are described in section 7.4. While this effect results from the material separation property, we still need a way to distinguish unstable and stable manifolds for practical use. We also would like to incorporate a way to analyze short term behavior that is decoupled from the temporal extent of the data set.

In order to address these issues, we provide an optional visualization technique. Commonly, heat maps are used for the visualization of FTLE fields. In this work, we normalized the field values and mapped them to the color range shown Fig. 7.10a. For an alternative visualization, we introduce a time period of interest (POI) which represents a certain timeframe the user is interested in. The user provides three time stamps:

- t_c: defines the current time frame, where the MSF should be computed,
- t_s^{poi}: defines the start of the POI,
- t_e^{poi}: defines the end of the POI,

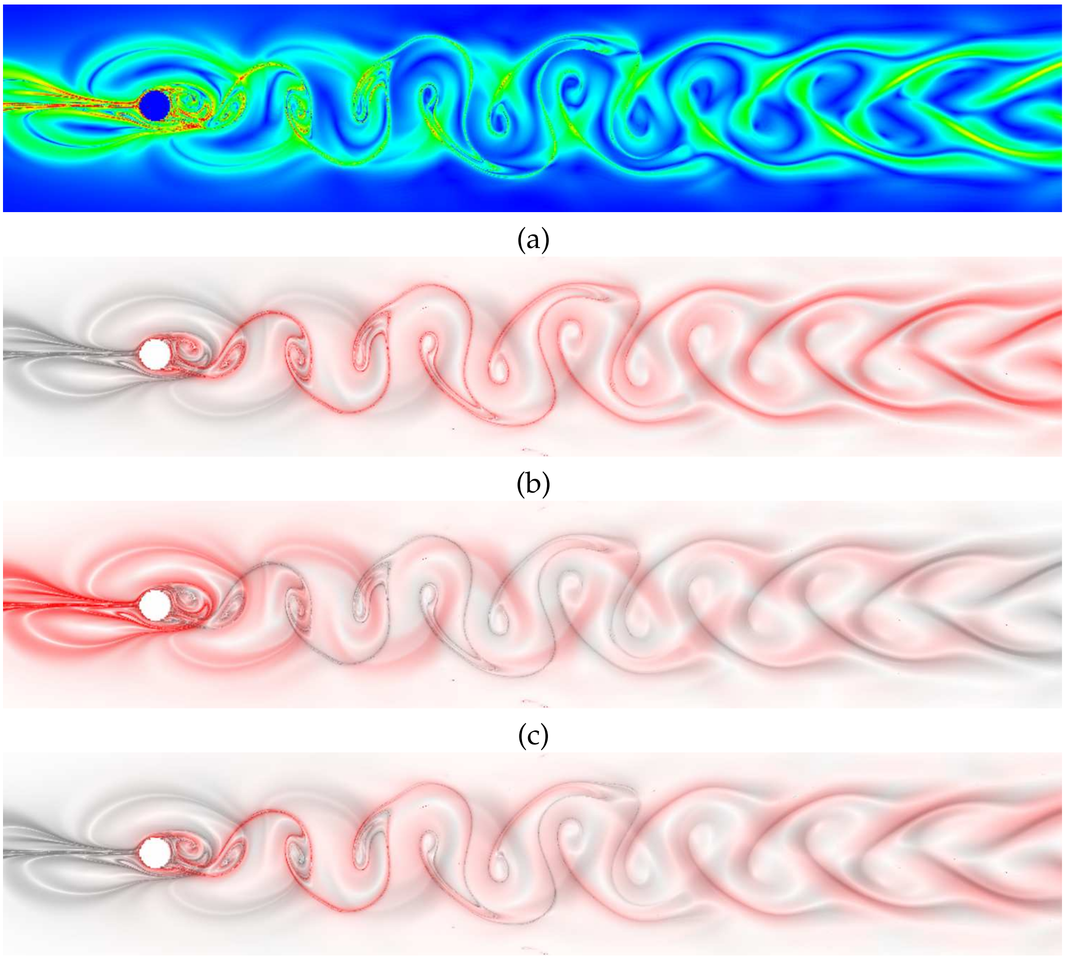
where $t_s^{poi} \leq t_c \leq t_e^{poi}$. In addition to the gradient of the flow map for the whole domain (Δ_M, see section 7.3.2), we similarly compute the gradient Δ_{poi} for the domain $D \times [t_s^{poi}, t_e^{poi}]$, which is restricted by the POI. Using this gradient, we compute the MSF for the POI:

$$\mathrm{MSF}_{poi} = \max_{i=1..n} \frac{1}{t_{out} - t_{in}} \ln \sqrt{\lambda_i((\Delta_{poi})^* \cdot \Delta_{poi})}. \tag{7.18}$$

Finally, we compute the value $\sigma = MSF/MSF_{poi}$, which describes the portion of separation that exists within the POI. Note that we need the term $t_{out} - t_{in}$ (instead of $t_e^{poi} - t_s^{poi}$) in equation 7.18 in order to put these two quantities in relation.

For the alternative visualization, we map MSF values to pixel intensity and σ to the saturation of red (see Fig. 7.10b). Therefore, structures with strong separating behavior within the POI appear red, while structures with separation outside the POI appear gray and desaturated. This way, we can highlight stable manifolds using $t_s^{poi} = t_{in}$ and $t_e^{poi} = t_c$, as can be seen in Fig. 7.9b. Analogously, unstable manifolds are highlighted by setting $t_s^{poi} = t_c$ and $t_e^{poi} = t_{out}$ (see Fig. 7.9c). Short term structures can also be emphasized by using different values for t_s^{poi} and t_e^{poi}, that are near t_c (see Fig. 7.9d).

In the context of MSF, the drawback of this visualization technique is that color values do not represent material structures anymore. Nevertheless, we regard this technique as a compromise, because material structures can still be recognized from the intensity values.



(a)

(b)

(c)

(d)

Figure 7.9: Time periods of interest for the MSF at $t = 240$: (a) heat map for $t = 240$, (b) $t_{poi} = [0, 240]$, (c) $t_{poi} = [240, 480]$, (d) $t_{poi} = [120, 240]$.

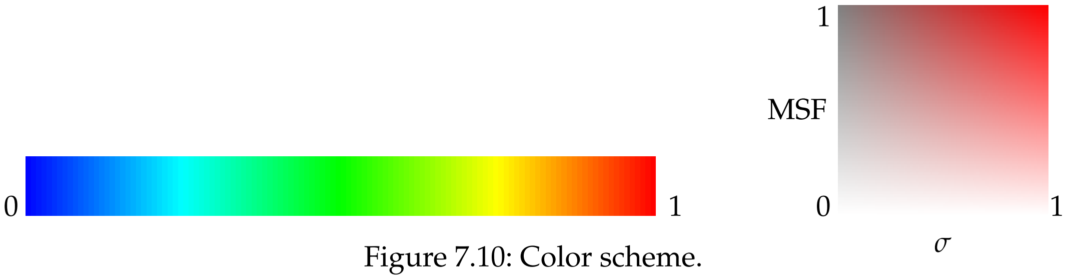


Figure 7.10: Color scheme.

The field values are normalized and mapped to the color range shown Fig. 7.10.

7.3.4 Appropriate Separating Structures

The fact that the MSF is constant along path lines does not guarantee material separation yet. In addition, we have to use an appropriate extraction of separating structures. However, all ridge definitions we are aware of fail to extract material structures as ridges. We propose a simple definition of MSF separating structures which guarantees material separation.

Instead of a central ridge line, we regard level sets of the MSF as separating structures, that is

$$s_c(t) = \{\mathbf{x} : MSF(\mathbf{x}, t) = k\}, \tag{7.19}$$

where k is a user defined constant. Per default, we choose k such that

$$f(k) = \frac{number\ of\ saddle\ points\ inside\ s_c(t)}{\sqrt{area\ enclosed\ by\ s_c(t)}} \tag{7.20}$$

is maximal. This formulation is motivated by the observation that in the discrete setting ridges are composed of a sequence of alternating saddles and maxima. Therefore, these level sets surround MSF maxima tightly and enclose regions with ridge-like behavior. Because high MSF values mark separating flow behavior, the level sets represent volumetric regions (instead of lines) that separate different flow behavior. Note that a similar view point has been taken in the field of vortex extraction, where both line structures (vortex core lines) and volumetric regions are well excepted approaches for vortex characterization. In a similar way, we propose a region approach for the separating structures. In the context of steady-state flow, the approach of representing separation behavior with regions was also described in [CMLZ08].

Because MSF values are constant along path lines and level sets define the subspaces of points with constant MSF values, the level sets stay on path lines. Therefore, our separating structures are material structures.

Alternatively, we could consider the regions inside the level sets

$$s_r(t) = \{\mathbf{x} : MSF(\mathbf{x}, t) > k\} \tag{7.21}$$

as separating structures. Again, these regions stay on path lines and are deformed by the vector field. Computing them can be done by a simple thresholding of the MSF.

Given a suitable threshold k, our separating structures tightly enclose the ridges of the MSF. Therefore, they describe ridge-like regions, in which the flow is mostly parallel to the enclosed ridge. Still, our separating structures are elongated like ridges and separate the flow on both sides of the structure. Our formulation of separating structures as volumetric regions represents an alternative to the notion of ridges that represents flow separation, but also leads to material structures as well as the segmentation of ridge-like flow behavior.

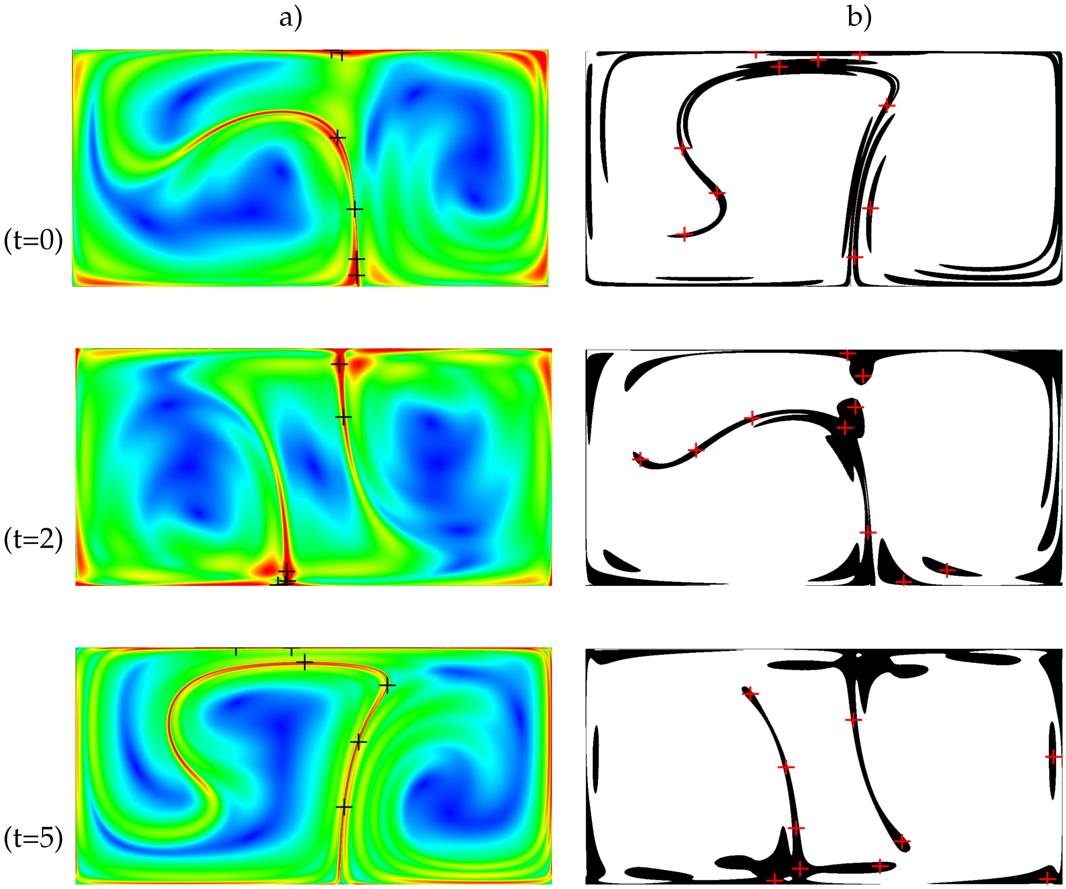


Figure 7.11: Double Gyre example: (a) MSF, (b) MSF separating structures. The black and red crosses show particles advected by the flow.

Our definition of separating structures means that they are connected by path lines. In particular, this holds for separating structures on $\delta\bar{D}_{in}$ and $\delta\bar{D}_{out}$ (see Fig. 7.8b). Thus, if we are interested in the separating structures of $\bar{D}$, we only need to compute the separating structures on any parametrization $\bar{P}_t$ (like $\delta\bar{D}_{in}$ or $\delta\bar{D}_{out}$) and advect them in the vector field like material structures. The parametrization and constant MSF of the path lines guarantees that all separating structures are found.

7.4 Examples

We implemented and tested the MSF method on a 3.2 GHz four core PC. All data sets were sampled on a regular 3D grid representing space-time. We used a fourth-order Runge-Kutta scheme and trilinear interpolation for the integration of path lines. The integration was carried out on the GPU (GeForce GTX 260) using CUDA. We computed the gradient of the flow map using finite differencing of neighboring particles as described in [SLM05]. The computation time mainly depends on the number of output pixels and the integration steps for each path line. Therefore, it also depends on the total time span of the data set. Corresponding to other FTLE visualization papers, we mapped the scalar field values to color hue, where blue means low value and red means high value. In the following, we present our results using three examples.

7.4.1 Double Gyre

Our next example is the double gyre data set, which was introduced in [SLM05]. It is computed using the following functions:

$$\begin{aligned} f(x,y,t) &= a(t)x^2 + b(t)x \\ a(t) &= \epsilon \, sin(\omega t) \\ b(t) &= 1 - 2\epsilon \, sin(\omega t) \end{aligned} \tag{7.22}$$

The vector field at point (x,y) at time t is given by:

$$\begin{aligned} u &= -\pi \, A \, sin(\pi f(x)) \, cos(\pi y) \\ v &= \pi \, A \, cos(\pi f(x)) \, sin(\pi y) \frac{df}{dx} \end{aligned} \tag{7.23}$$

The domain for this data set is $D = [0,2] \times [0,1]$. We used the parameters $A = 0.1, \omega = \frac{2\pi}{10}$ and $\epsilon = 0.25$.

First, we consider the time span $t = [0,10]$, consisting of one full period of the double gyre motion. We sampled space-time with a 200 × 100 × 100 grid. Fig. 7.11a shows the resulting MSF at different points in time. Fig. 7.11b shows the corresponding separating structures. The threshold k of the level set is determined by the maximum of $f(k)$ displayed in Fig. 7.13a. The computation time for one picture was approximately 15 seconds. The black and red crosses mark particles that are seeded by hand and advected in the flow. Note that they do not cross the separating structures at any time. This confirms our statement that MSF separating structures are indeed material structures.

Both separating structures from forward and backward integration are visible. At $t = 0$ only structures from forward integration are extracted. As t grows, separating structures from backward integration develop and cross the forward structures. Particles at these crossings flow into both saddle regions on the top edge under forward integration and on the bottom edge under backward integration. If we enlarge the time span of the data sets, the separating structures become more complex and intersect each other more often. Fig. 7.12 shows an MSF of the double gyre data set with two periods.

7.4.2 Cylinder Flow

In the third example, we analyzed a data set of a flow around a circular cylinder that was simulated on an irregular grid. This grid was uniformly resampled onto a regular grid with dimension 560 × 160 × 61. The time span of the data set is $t = [0,480]$.

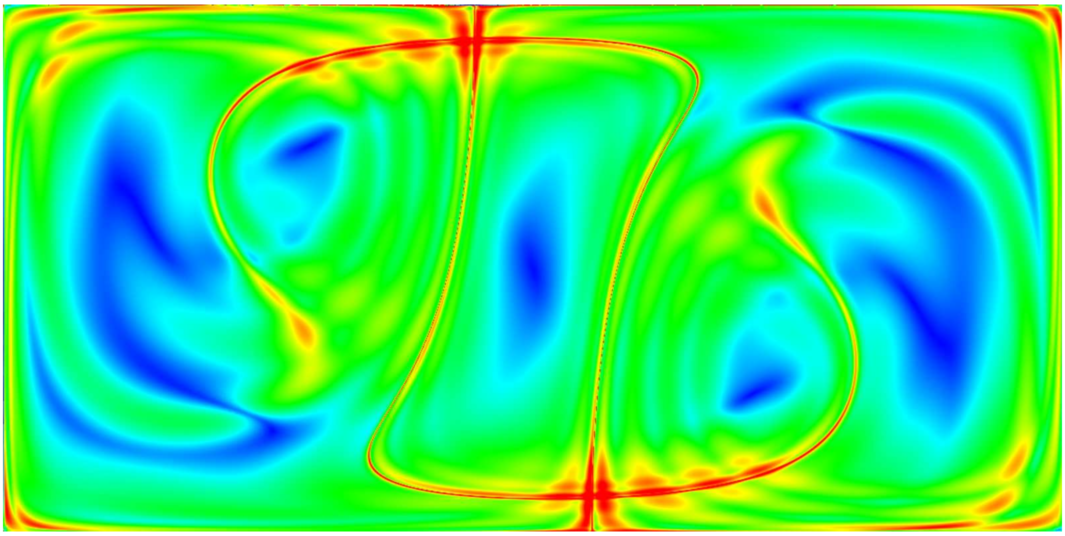


Figure 7.12: Double period of the double gyre (t=[0,20]) at t=10. The more periods are considered, the more the separating structures cross.

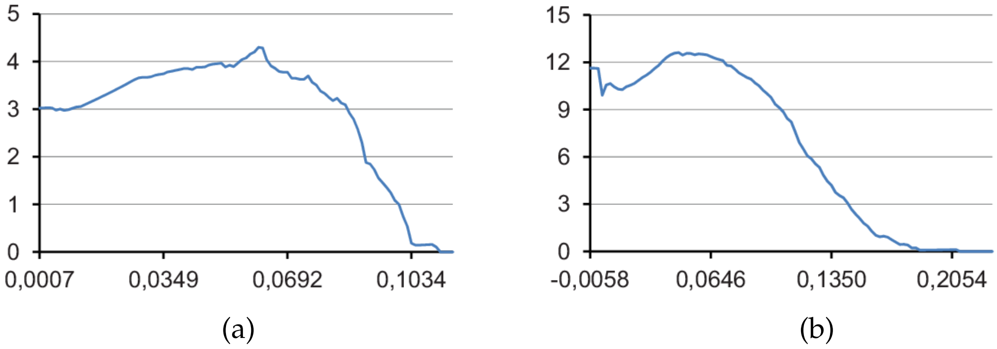


Figure 7.13: Plot of $f(k)$ (equation 7.20) against k: (a) double gyre with a maximum of $f(k)$ at a MSF value of $k = 0.06$, (b) cylinder with a maximum of $f(k)$ at $k = 0.046$.

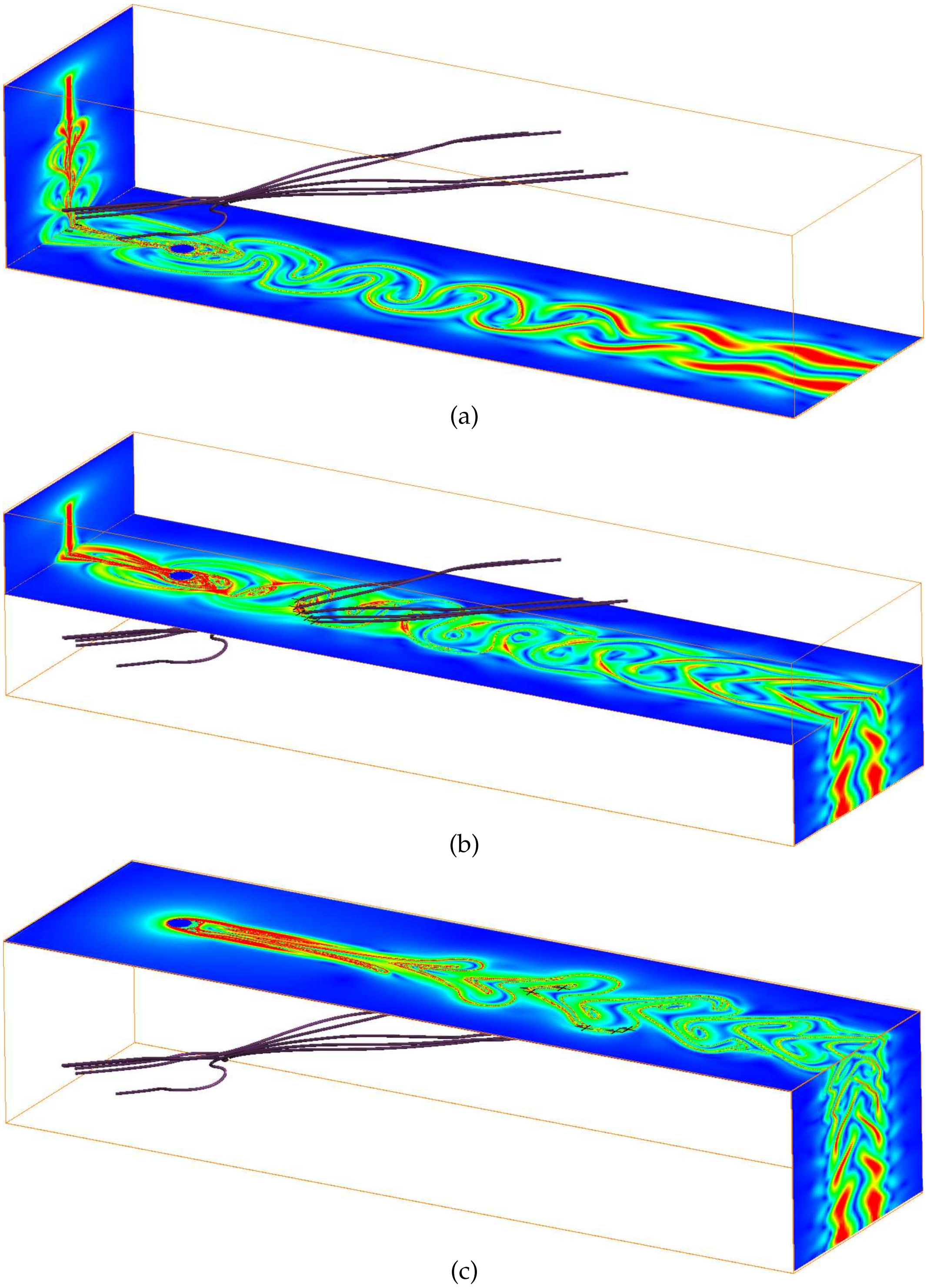
Fig. 7.15 shows the MSF of four time steps (colored images). The computation time was approximately 40 seconds per picture. The corresponding structures in Fig. 7.15 (monochrome) clearly separate regions of different flow behavior. The threshold k of the level set is determined by the maximum of $f(k)$ displayed in Fig. 7.13b. Again, the black and red crosses mark the position of five particles that are advected in the flow. In contrast to the FTLE ridges in Fig. 7.4, the particles do not cross the separating structures and stay on them.

In Fig. 7.14, we show different path line parametrizations. The MSF on the inflow boundary $\delta\bar{D}_{in}$ and outflow boundary $\delta\bar{D}_{in}$ is shown in Fig. 7.14a and Fig. 7.14c. Note that the shape of these boundaries is particularly simple in this data set, because throughout the time span, the left and right boundaries have only inflow and outflow components, respectively. In Fig. 7.14c we build a parametrization $\bar{P}_t$ based on the time slice $t = 240$ and completing it with the appropriate parts of $\delta\bar{D}_{in}$ and $\delta\bar{D}_{out}$ to get a complete path line parametrization, as described in section 7.3.1. All three parametrizations are crossed by all path lines in the domain. Five of them are shown in Fig. 7.14. Note that they cross separating structures in all parametrizations. Therefore, they are material structures.

7.5 Discussion

We have analyzed that, in general, FTLE ridges are not material structures. At first sight, this is a contradiction to the results of Shadden et al. [SLM05], who found FTLE ridges to approximate material structures well. Shadden et al. also have shown that for $T \to \infty$ FTLE ridges become exact material lines, which on the other hand becomes equivalent to the infinite-time LE. In fact, our results for the double gyre with rather long integration time confirm the results of Shadden et al. However, for shorter integration times, the approximation error of FTLE becomes larger. In practice, we have to face the fact that we have data sets of limited time spans. Because FTLE is computed using constant integration times, FTLE methods rely on relatively short integration times in practical applications. With the cylinder data set we have shown that in such cases the approximation error can become substantial and the flux across FTLE ridges cannot be ignored. In contrast, we have shown that particles advected in the flow stay on separating structures of the MSF and these have zero cross flux. In conclusion, we agree with the result of Shadden et al. However, we found that the error for shorter integration times can become more substantial in practice.

Our method represents a modification of the FTLE method. While FTLE analyzes flow separation over a given time span, the MSF method incorporates the whole temporal extent of the data set and extracts true material structures as separating structures. While FTLE can be regarded as a “time-local” method, our goal is to provide a global view onto separation behavior. The consequence is that flow-phenomena from one time step can cause MSF structures at far-off time steps. While this may not be appropriate for every application, it provides an asymptotic view of the flow.

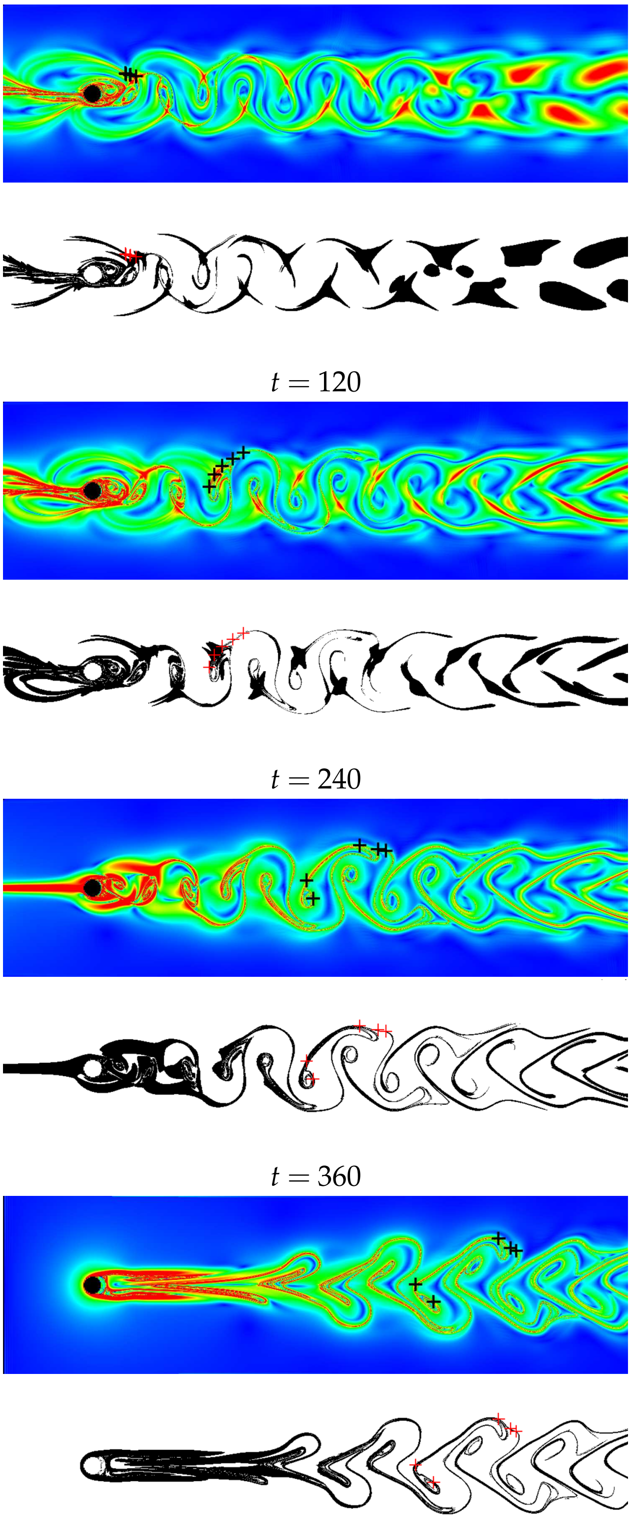


(a)

(b)

(c)

Figure 7.14: MSF on parametrization subdomains of cylinder flow: (a) $\delta\bar{D}_{in}$, (b) $\bar{P}_t$ based time slice $t = 240$, (c) $\delta\bar{D}_{out}$. The dark lines are 5 different path lines.



$t = 120$

$t = 240$

$t = 360$

$t = 480$

Figure 7.15: Flow around a circular cylinder: (colored) MSF, (monochrome) MSF separating structures. The black and red crosses show particles advected by the flow.

Since at any time step t, the MSF is computed both from forward and backward integration of path lines, our results contain mixtures of unstable and stable manifolds. In section 7.3.3 we have presented an optional visualization technique to still differentiate between these structures. At the initial time steps, the separating structures from forward integration are dominant. Over the course of time, these structures compress and corresponding particles converge to the boundary. At the initial time steps, the particles on separating structures from backward integration are very close to the boundary. In subsequent time steps they diverge from the boundary while the structures expand. Note that the technique from section 7.3.3 is also useful for outlining local structures in the context of the global view provided by the MSF.

An advantage of our method is that the topology of the MSF is preserved over time. Hence, MSF separating structures consistently bound regions of coherent flow behavior throughout time. In contrast, FTLE values on a given path line are not constant. Thus, the topology of FTLE fields and ridges vary over time. Furthermore, for divergence-free vector fields the volume of MSF separating structures is invariant. The double gyre example (Fig. 7.11b) demonstrates this behavior.

With our method, the MSF value of any point $(\mathbf{x}, t)$ in $\bar{D}$ can be found by computing the MSF values of a parametrization $\bar{P}$ as introduced in section 7.3.1 in preprocess and integrating a particle from $(\mathbf{x}, t)$ to $\bar{P}$. This technique could be used for the rapid computation of MSF, which could serve for real-time exploration applications. However, the MSF values of $\bar{P}$ are computed at discrete samples. This can lead to artifacts caused by the insufficient resolution of computed MSF values. For instance, the folding of separating structures at the border of the double gyre example is hidden by a limited resolution of the sampled MSF.

8 Conclusion and Future Work

In chapters 4 and 5 of this thesis, pioneering work has been done in the field of vector field topology with uncertainty. Chapter 6 introduced vortex criteria for such uncertain vector fields. In chapter 7 we presented a variation of Lagrangian coherent structures with guaranteed material separation.

We introduced the first approaches to extract and visualize the topology of uncertain 2D and 3D vector fields. For this we started with a definition of an uncertain 2D vector field as a 4D scalar field and an uncertain 3D vector field as a 6D scalar field, respectively. Such fields represent the transport probability from any location in the domain to all other locations in the same domain. Based on these definitions we defined an integration scheme for uncertain vector fields by using a time dependent scalar field that represents a particle distribution. If the integration of the particle distribution converges, we will find sink distributions in forward direction and source distributions in backward direction. Saddle distributions cannot be found using integration. For the extraction of saddle distributions we showed two different ways. For the first way we invented a method using a derived uncertain vector field, the acceleration field. This field is integrated in the same manner as the original uncertain vector field. Sink distributions in this field correlate to all critical distributions in the original field. By using a probabilistic version of the Poincaré-Hopf index, we identify the type of the critical distributions. The disadvantage of this method is that we approximate the acceleration field with Gaussian distribution that is not an appropriate distribution function. In section 6.1 we showed that this leads to inaccurate results. For 3D uncertain vector fields we developed a more precise algorithm to detect saddle and boundary switch connectors including saddle points at crossings. This algorithm does not need any derived vector field. We use the particle distributions in all data points of the domain. If particles integrated from one data point move to more than one sink in forward direction or more than one source in backward direction, a probability for a saddle connector exists. Using the probabilities for sinks, sources, saddle connectors and boundary switch connectors we create colored height maps in 2D and volume visualizations in 3D that give a combined visual summary of critical points, separating structures and their probabilities. Such a visualization of uncertain topology is completely new. Previous approaches visualized only probabilities of local uncertain feature or global features like topology without uncertainty. Within this work we closed this gap.

In chapter 6 we applied vortex criteria to uncertain vector fields. We extended region based (Q , λ_2) and geometry based (Sujudi & Haimes) vortex criteria to uncertain vector fields. The key for this is the computation of the Jacobian by using a correlated support region. This means we create a correlation matrix for all data points which are involved in the computation of such a vortex criterion. We showed that uncorrelated support regions lead to inexact results and wrong distribution functions as results. In chapters 4 and 5 we can visualize the results, representing probabilities for the vortex criteria as height maps and volume visualizations. Our method is very similar to Petz et al. [PPH12] which was published at the same time. In general this method can be applied to all methods that extract local features in uncertain vector fields and work locally on values and derivatives. They showed this on examples for cores of swirling motion and for critical points. We believe that this cannot be done for critical point distributions, because in comparison to the particle integration method, their results depend mainly on the discretization of the input data. As result they get the probability that a particle stays at the location. In uncertain vector fields critical points are no local features, they are global features.

The chapters 4 to 6 present approaches to approximate computations on uncertain vector fields. The chapter 6 gives satisfying results for truly local features and results of chapter 4 and 5 approximate integrations on uncertain vector fields. Theoretically, a perfect result for integrations would be possible if all distributions of all data points in the domain were correlated. So only one extremely large correlation matrix represents the uncertain vector field. This would open the possibility to create complete sample fields to generate smooth trajectories. Our method produces trajectories which follow a Brownian motion (see Fig. 4.1) that does not consider the history of a trajectory. However, currently it is not possible to create such a correlation matrix for uncertain vector fields. The problems start with gathering the required input data to create the correlation matrix and continue with extreme requirements in terms of memory and computational power. Here we see room for future work.

In the last part of this thesis we introduced a novel way to compute material separating structures. We showed that ridges of FTLE are only approximate material separating structure. For this we demonstrated the flux over FTLE ridges. To solve the problem we introduced a new approach that uses the whole available integration time. Further, we do not use lines as ridge structures, instead we use level sets as material separating structures. The level sets are real material separating structures. In future we see the possibility to apply this method to uncertain time dependent vector fields. The key for this is to compute the relative probability of given level sets, representing the MSF ridges. This would be an alternative to the proposed LIC technique in Hummel et al. [HOGJ13] to show the general flow behavior.

Bibliography

[AO03] P. Ashwin and G. Ochs. Convergence to local random attractors. *Dynamical Systems: An International Journal*, 18(2):139–158, 2003.

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

[BFMW12] Kai Bürger, Roland Fraedrich, Dorit Merhof, and Rüdiger Westermann. Instant visitation maps for interactive visualization of uncertain particle trajectories. *Proc. SPIE*, 8294:82940P–82940P–12, 2012.

[BP02] Dirk Bauer and Ronald Peikert. Vortex tracking in scale-space. In *Proceedings of the symposium on Data Visualisation 2002*, VISSYM '02, pages 233–ff, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[BS94] David C. Banks and Bart A. Singer. Vortex tubes in turbulent flows: identification, representation, reconstruction. In *IEEE Visualization*, pages 132–139, 1994.

[BWE05] Ralf P. Botchen, Daniel Weiskopf, and Thomas Ertl. Texture-based visualization of uncertainty in flow fields. In *IEEE Visualization*, pages 647–654, 2005.

[BWE06] Ralf P. Botchen, Daniel Weiskopf, and Thomas Ertl. Interactive visualization of uncertainty in flow fields using texture-based techniques. In *Intl. Symp. on Flow Visualization*, 2006.

[CMLZ08] 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.

[CPC90] M. S. Chong, A. E. Perry, and B. J. Cantwell. A general classification of three-dimensional flow fields. *Physics of Fluids A*, 2(5):765–777, 1990.

[Cra34] Cecil C. Craig. On the frequency function of xy. *Ann. Math. Statist.*, 7(1):1–15, 1934.

[DLS90] David Degani, Yuval Levy, and Arnan Seginer. Graphical visualization of vortical flows by means of helicity. *AIAA Journal*, 28:1347–1352, 1990.

[dLvL99a] W. de Leeuw and R. van Liere. Collapsing flow topology using area metrics. In *IEEE Visualization*, pages 149–354, 1999.

[dLvL99b] W. de Leeuw and R. van Liere. Visualization of global flow structures using multiple levels of topology. In *Proc. Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym '99)*, pages 45–52, 1999.

[EGM$^+$94] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach. Ridges for image analysis. *Journal of Mathematical Imaging and Vision*, 4(4):353–373, 1994.

[Eva09] Lawrence Craig Evans. An introduction to stochastic differential equations. Online Version 1.2, December 2009.

[FSL$^+$09] G. Fru, H. Shalaby, A. Laverdant, C. Zistl, G. Janiga, and D. Thévenin. Direct numerical simulations of turbulent flames to analyze flame/acoustic interactions. In Anna Schwarz and Johannes Janicka, editors, *Combustion Noise*, pages 239–268. Springer Berlin Heidelberg, 2009.

[Gel01] A. Van Gelder. Stream surface generation for fluid flow solutions on curvilinear grids. In *Proc. Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym '01)*, pages 95–106, 2001.

[GGTH07] 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(6):1464–1471, 2007.

[GLL91] A. Globus, C. Levit, and T. Lasinski. A tool for visualizing the topology of three-dimensional vector fields. In *IEEE Visualization*, pages 33–40, 1991.

[GLT$^+$07] C. Garth, G. Li, X. Tricoche, C.D. Hansen, and H. Hagen. Visualization of coherent structures in transient 2d flows. In *Proceedings of TopoInVis*, 2007.

[GOPT11] T. Germer, M. Otto, R. Peikert, and H. Theisel. Lagrangian coherent structures with guaranteed material separation. *Computer Graphics Forum (Proc. EuroVis 2011)*, 30(3):761–770, 2011.

[GS06] Henning Griethe and Heidrun Schumann. The visualization of uncertain data: Methods and problems. In *SimVis*, pages 143–156, 2006.

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

[Hal02] G. Haller. Lagrangian coherent structures from approximate velocity data. *Physics of Fluids*, 14(6):1851–1861, 2002.

[HH89a] J. Helman and L. Hesselink. Representation and display of vector field topology in fluid flow data sets. *IEEE Computer*, 22(8):27–36, August 1989.

[HH89b] James L. Helman and Lambertus Hesselink. Representation and display of vector field topology in fluid flow data sets. *Computer*, 22(8):27–36, August 1989.

[HH91] J. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Comput. Graph. Appl.*, 11:36–46, May 1991.

[HK07] Boris Hasselblatt and Anatole Katok, editors. *Handbook of Dynamical Systems*, volume 1A. North Holland, 2007.

[Hla1] Hlawatsch, Marcel and Leube, Philipp and Nowak, Wolfgang and Weiskopf, Daniel . Flow Radar Glyphs - Static Visualization of Unsteady Flow with Uncertainty . *IEEE TVCG* , 17 (12): inprint , 2011 .

[HOGJ13] Mathias Hummel, Harald Obermaier, Christoph Garth, and Kenneth I. Joy. Comparative visual analysis of lagrangian transport in CFD ensembles. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2743–2752, 2013.

[Hul92] J. Hultquist. Constructing stream surfaces in steady 3D vector fields. In *IEEE Visualization*, pages 171–177, 1992.

[HWM88] J. C. R. Hunt, A. Way, and P. Moin. Eddies, stream, and convergence zones in turbulent flows. Center for Turbulence Research Report CTR-S88, Standford University, 1988.

[HY00] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D*, 147(3-4):352–370, 2000.

[JH95] Jinhee Jeong and Fazle Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285(-1):69–94, 1995.

[JMT04] Ming Jiang, Raghu Machiraju, and David Thompson. Detection and visualization of vortices. *Visualization Handbook*, page 287?301, 2004.

[Joh04] Chris Johnson. Top scientific visualization research problems. *IEEE Comput. Graph. Appl.*, 24(4):13–17, 2004.

[KW86] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo methods. Vol. 1: basics*. Wiley-Interscience, New York, NY, USA, 1986.

[LCM$^+$05] Francois Lekien, Chad Coulliette, Arthur J. Mariano, Edward H. Ryan, Lynn K. Shay, George Haller, and Jerry Marsden. Pollution release tied to invariant manifolds: A case study for the coast of florida. *Physica D*, 210(1), 2005.

[LDG98] H. Löffelmann, H. Doleisch, and E. Gröller. Visualizing dynamical systems near critical points. In *Spring Conference on Computer Graphics and its Applications*, pages 175–184, 1998.

[LHZP05] R. Laramee, H. Hauser, L. Zhao, and Fr. Post. Topology-based flow visualization, the state of the art. In *Proc. Topo-in-Vis*, pages 1–20, 2005.

[Lia66] A. M. Liapunov. *Stability of motion*. Academic Press, 1966.

[Lin98] Tony Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal Computer Vision*, 30(2):117–156, 1998.

[LM10] Doug Lipinski and Kamran Mohseni. A ridge tracking algorithm and error estimate for efficient computation of lagrangian coherent structures. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 20(1):017504, 2010.

[MBHJ03] K. Mahrous, J. Bennett, B. Hamann, and K. Joy. Improving topological segmentation of three-dimensional vector fields. In *Data Visualization 2003. Proc. VisSym 03*, pages 203–212, 2003.

[MBS$^+$04] K. Mahrous, J. Bennett, G. Scheuermann, B. Hamann, and K. Joy. Topological segmentation in three-dimensional vector fields. *IEEE TVCG*, 10(2):198–205, 2004.

[MLP$^+$10] T. McLoughlin, R. S Laramee, R. Peikert, F. H Post, and M. Chen. Over two decades of integration-based, geometric flow visualization. *Computer Graphics Forum*, 29(6):1807–1829, 2010.

[MW14] Mihaela Mihai and Rüdiger Westermann. Visualizing the stability of critical points in uncertain scalar fields. *Computers & Graphics*, 41(0):13 – 25, 2014.

[OBS04] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. Ridge-valley lines on meshes via implicit surface fitting. *ACM Transactions on Graphics*, 23(3):609–612, 2004.

[OGHT10] Mathias Otto, Tobias Germer, Hans-Christian Hege, and Holger Theisel. Uncertain 2D vector field topology. *Computer Graphics Forum (Proc. Eurographics)*, 29(2):347–356, 5 2010.

[OGT11a] M. Otto, T. Germer, and H. Theisel. Closed stream lines in uncertain vector fields. In *Proc. Spring Conference on Computer Graphics (SCCG)*, pages 115–121, 2011.

[OGT11b] M. Otto, T. Germer, and H. Theisel. Uncertain topology of 3d vector fields. In *Proceedings of 4th IEEE Pacific Visualization Symposium (PacificVis 2011)*, pages 67–74, Hong Kong, China, March 2011.

[OT12] M. Otto and H. Theisel. Vortex analysis in uncertain vector fields. *Computer Graphics Forum (In Proc. EuroVis 2012)*, 31:1035–1044, 2012.

[PD10] Thomas Peacock and John Dabiri. Introduction to focus issue: Lagrangian coherent structures. *Chaos*, 20(1):p017501, Mar 2010.

[PH11] Kai Pöthkow and Hans-Christian Hege. Positional uncertainty of isocontours: Condition analysis and probabilistic measures. *IEEE TVCG*, 17(10):1393 – 1406, 2011.

[PH13] Kai Pöthkow and Hans-Christian Hege. Nonparametric models for uncertainty visualization. *Computer Graphics Forum*, 32(3):131 – 140, 2013.

[PMW13] T Pfaffelmoser, M Mihai, and R Westermann. Visualizing the variability of gradients in uncertain 2d scalar fields. *IEEE transactions on visualization and computer graphics*, 2013.

[Por97] L.M. Portela. *On the Identification and Classification of Vortices*. Phd thesis, Stanford University, School of Mechanical Engineering, 1997.

[PPH12] Christoph Petz, Kai Pöthkow, and Hans-Christian Hege. Probabilistic local features in uncertain vector fields with spatial correlation. *Computer Graphics Forum*, 31(3):1045 – 1054, 2012.

[PR99] Ronald Peikert and Martin Roth. The "parallel vectors" operator - a vector field visualization primitive. In *IEEE Visualization*, pages 263–270, 1999.

[PS97] P. A. Philippou and R. N. Strickland. Vector field analysis and synthesis using three dimensional phase portraits. *Graphical Models and Image Processing*, 59:446–462, November 1997.

[PS08] R. Peikert and F. Sadlo. Height Ridge Computation and Filtering for Visualization. In *Proceedings Pacific Vis*, pages 119–126, 2008.

[PVH$^+$03] Fr. Post, B. Vrolijk, H. Hauser, R. Laramee, and H. Doleisch. The state of the art in flow visualization: Feature extraction and tracking. *Computer Graphics Forum*, 22(4):775–792, 2003.

[PW12] T. Pfaffelmoser and R. Westermann. Visualization of global correlation structures in uncertain 2d scalar fields. In *Computer Graphics Forum*, volume 31, pages 1025–1034. Wiley Online Library, 2012.

[PW13] Tobias Pfaffelmoser and Rüdiger Westermann. Visualizing contour distributions in 2d ensemble data. In *EuroVis-Short Papers*, pages 55–59. The Eurographics Association, 2013.

[PWH11] Kai Pöthkow, Britta Weber, and Hans-Christian Hege. Probabilistic marching cubes. *CGF*, 30(3):931 – 940, 2011.

[PWL96] Alex T. Pang, Craig M. Wittenbrink, and Suresh K. Lodh. Approaches to uncertainty visualization. *The Visual Computer*, 13:370–390, 1996.

[RC91] Stephen Kern. Robinson and Ames Research Center. *The kinematics of turbulent boundary layer structure*. National Aeronautics and Space Administration, Ames Research Center, 1991.

[RP98] Martin Roth and Ronald Peikert. A higher-order method for finding vortex core lines. In *IEEE Visualization*, pages 143–150, 1998.

[SBM$^+$01] G. Scheuermann, T. Bobach, H. Hagen K. Mahrous, B. Hamann, K. Joy, and W. Kollmann. A tetrahedra-based stream surface algorithm. In *IEEE Visualization*, pages 151–158, 2001.

[SFRS11] Dominic Schneider, Jan Fuhrmann, Wieland Reich, and Gerik Scheuermann. A variance based ftle like method for unsteady uncertain vector fields. In *TopoInVis*, 2011.

[SH95] David Sujudi and Robert Haimes. Identification of swirling flow in 3-d vector fields. *Journal of Fluid Mechanics*, 285:69–94, 1995.

[SHK$^+$97] Gerik Scheuermann, Hans Hagen, Heinz Krüger, Martin Menzel, and Alyn P. Rockwood. Visualization of higher order singularities in vector fields. In *IEEE Visualization*, pages 67–74, 1997.

[SJK04] Allen R. Sanderson, Chris R. Johnson, and Robert M. Kirby. Display of vector fields using a reaction-diffusion model. In *IEEE Visualization*, pages 115–122, 2004.

[SLM05] Shawn C. Shadden, Francois Lekien, and Jerrold E. Marsden. Definition and properties of lagrangian coherent structures from finite-time lyapunov exponents in two-dimensional aperiodic flows. *Physica D*, 212(7):271–304, 2005.

[SLP$^+$09] Shawn C. Shadden, Francois Lekien, Jeffrey D. Paduan, Francisco P. Chavez, and Jerrold E. Marsden. The correlation between surface drifters and coherent structures based on high-frequency radar data in monterey bay. *Deep Sea Research Part II: Topical Studies in Oceanography*, 56(3-5):161 – 172, 2009.

[SP07a] F. Sadlo and R. Peikert. Visualizing lagrangian coherent structures and comparison to vector field topology. In *Proceedings of the 2007 Workshop on Topology-Based Method in Visualization (TopoInVis)*, 2007.

[SP07b] Filip Sadlo and Ronald Peikert. Efficient visualization of lagrangian coherent structures by filtered amr ridge extraction. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization)*, 13(6):19–26, 2007.

[SRP09] Filip Sadlo, Alessandro Rigazzi, and Ronald Peikert. Time-Dependent Visualization of Lagrangian Coherent Structures by Grid Advection. In *Proceedings of TopoInVis (to appear)*. Springer, 2009.

[STS07] Thomas Schultz, Holger Theisel, and Hans-Peter Seidel. Topological visualization of brain diffusion mri data. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1496–1503, 2007.

[STS10] Thomas Schultz, Holger Theisel, and Hans-Peter Seidel. Crease surfaces: From theory to extraction and application to diffusion tensor MRI. *IEEE Transactions on Visualization and Computer Graphics*, 16(1), 2010.

[SVBK14] Thomas Schultz, Anna Vilanova, Ralph Brecheisen, and Gordon Kindlmann. Fuzzy fibers: Uncertainty in dmri tractography. In *Scientific Visualization: Uncertainty, Multifield, Biomedical, and Scalable Visualization*, Mathematics + Visualization. Springer, 2014. Accepted for publication.

[SW09] Filip Sadlo and Daniel Weiskopf. Time-Dependent 2D Vector Field Topology: An Approach Inspired by Lagrangian Coherent Structures. *Computer Graphics Forum (to appear)*, 2009.

[SWH05] Jan Sahner, Tino Weinkauf, and Hans-Christian Hege. Galilean invariant extraction and iconic representation of vortex core lines. In *EuroVis*, pages 151–160, 2005.

[SWTH07] J. Sahner, T. Weinkauf, N. Teuber, and H.-C. Hege. Vortex and strain skeletons in eulerian and lagrangian frames. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):980–990, September - October 2007.

[TK94] Barry N. Taylor and Chris E. Kuyatt. Guidelines for evaluating and expressing the uncertainty of nist measurement results. Technical report, NIST Tecnical Note 1297, 1994.

[TSH00] X. Tricoche, G. Scheuermann, and H. Hagen. A topology simplification method for 2D vector fields. In *IEEE Visualization*, pages 359–366, 2000.

[TSH01] X. Tricoche, G. Scheuermann, and H. Hagen. Continuous topology simplification of planar vector fields. In *IEEE Visualization*, pages 159–166, 2001.

[TSW+05] H. Theisel, J. Sahner, T. Weinkauf, Hans-C. Hege, and H.-P. Seidel. Extraction of parallel vector surfaces in 3d time-dependent fields and applications to vortex core line tracking. In *IEEE Visualization*, 2005.

[TWHS03] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. Saddle connectors - an approach to visualizing the topological skeleton of complex 3D vector fields. In *IEEE Visualization*, pages 225–232, 2003.

[TWHS04] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. Grid-independent detection of closed stream lines in 2d vector fields. In *Proc. Vision, Modeling and Visualization (VMV) 2004*, pages 421–428, Stanford, USA, 15.-18. November 2004 2004.

[TWHS05] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. Topological methods for 2D time-dependent vector fields based on stream lines and path lines. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):383–394, 2005.

[TWHS07] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel. On the applicability of topological methods for complex flow. In *In: Proc. Topo-In-Vis 2005*, pages 105–120, 2007.

[vW93] J. van Wijk. Implicit stream surfaces. In *IEEE Visualization*, pages 245–252, 1993.

[Wei08] T. Weinkauf. *Extraction of Topological Structures in 2D and 3D Vector Fields*. PhD thesis, University Magdeburg, 2008.

[WPJ+08] M. Weldon, T. Peacock, G. B. Jacobs, M. Helu, and G. Haller. Experimental and numerical investigation of the kinematic theory of unsteady separation. *Journal of Fluid Mechanics*, 611:1–11, 2008.

[WPL96] 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(3):266–279, September 1996.

[WS01] Thomas Wischgoll and Gerik Scheuermann. Detection and visualization of closed streamlines in planar flows. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):165–172, 2001.

[WS02] Thomas Wischgoll and Gerik Scheuermann. Locating closed streamlines in 3d vector fields. In *VISSYM '02: Proceedings of the symposium on Data Visualisation 2002*, pages 227–ff, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.

[WSH01] Thomas Wischgoll, Gerik Scheuermann, and Hans Hagen. Tracking closed streamlines in time-dependent planar flows. In *In Proceedings of the Vision Modeling and Visualization Conference 2001 (VMV 01)*, pages 447–454, 2001.

[WSTH07] T. Weinkauf, J. Sahner, H. Theisel, and H.-C. Hege. Cores of swirling particle motion in unsteady flows. *IEEE TVCG (Proc. IEEE Visualization)*, 13(6):1759–1766, 28. October - 1. November 2007.

[WTHS04] T. Weinkauf, H. Theisel, H.-C. Hege, and H.-P. Seidel. Boundary switch connectors for topological visualization of complex 3D vector fields. In *Proc. Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym '04)*, pages 183–192, 2004.

[WTS$^+$05] T. Weinkauf, H. Theisel, K. Shi, H.-C. Hege, and H.-P. Seidel. Extracting higher order critical points and topological simplification of 3D vector fields. In *Proc. IEEE Visualization 2005*, pages 559–566, Minneapolis, U.S.A., October 2005.

[ZBP$^+$91] N. J. Zabusky, O. N. Boratav, R. B. Pelz, M. Gao, D. Silver, and S. P. Cooper. Emergence of coherent patterns of vortex stretching during reconnection: A scattering paradigm. *Phys. Rev. Lett.*, 67:2469–2472, Oct 1991.

[ZDG$^+$08] Torre Zuk, Jon Downton, David Gray, Sheelagh Carpendale, and J.D. Liang. Exploration of uncertainty in bidirectional vector fields. In *Visualization and Data Analysis*, volume 6809, 2008.