

Diss. ETH No. 19284

# Computational Visualization of Physics and Topology in Unsteady Flow

A dissertation submitted to  
**ETH Zurich**

for the Degree of  
**Doctor of Sciences**

presented by  
**Filip Sadlo**  
Dipl. Informatik-Ing. ETH Zurich  
born 27. February 1972  
citizen of Zurich, ZH, Switzerland

accepted on the recommendation of  
**Prof. Dr. Markus Gross**, ETH Zurich, examiner  
**Prof. Dr. Ronald Peikert**, ETH Zurich, co-examiner  
**Prof. Dr. Gerik Scheuermann**, University of Leipzig, co-examiner

2010

---

An expert is a person who has made all the mistakes that can be made in a very narrow field. – *Niels Bohr*

---

# Abstract

Vector fields play an important role in many areas in research and engineering. This thesis originated in the context of a project with two industry partners active in turbo-machinery and is hence oriented toward this area. Nevertheless, its first part addresses generic fluid flows whereas its second part is even more universal: its subject are vector fields in general.

Design of turbo-machinery looks back on a long history. Although high efficiency is already achieved in many areas, there is still high effort in pushing efficiency some percent further. Although this seems to be a worthless undertaking at first sight, it can substantially account into the return of these machines. Especially in the field of hydropower plants, it pays off due to their typically very long operation duration. The operation of turbines depends on various factors, many of which address to the field of fluid dynamics. For the design, there is a clear trend from experiments toward computational fluid dynamics simulations. This is especially true for these decades since computing power and simulation techniques are coming to a level where experiments are almost dispensable. This leads to the situation that reliable numerical data are easily available but often hard to interpret because of their size and intricacy. This makes computational flow visualization and analysis increasingly important. The main objective of this thesis is to push computational flow visualization further toward the needs and possibilities of modern turbo-machinery design.

This thesis focuses on two fields in computational flow visualization: vortices and global structure of flows. Although vortices are beneficial in many areas of nature, from the birth of solar systems over the flight of dragonflies to the physics of our metabolism, they tend to affect technology. Although there are fields in technology that benefit from vortices such as chemical production, they lower performance and lifetime in many others, especially where relatively high velocities are involved, such as automotive engineering and energy conversion. Two approaches are presented in this thesis for the analysis of vortex creation and vortex dynamics in general. The first one bases solely on the instantaneous structure of the curl of the velocity field, the so-called vorticity, whereas the second one examines its transport over time by advection and diffusion. This can help both, the production and avoidance of vortices as well as their control in each sense, depending on the requirements and possibilities.

Understanding the global and semi-global structure of flows, or vector fields in general, gives deeper insight into many mechanisms such as transport, mixing, and residence time and therefore allows for more control and hence more focused

research and engineering. This thesis addresses this field in two sub-categories: vector field topology and Lagrangian coherent structures. As an introduction, it is shown how the visualization of flow features can benefit from topologically seeded streamlines and stream surfaces. A special vortex type, the so-called ring vortex, is the subject in the remaining research in the field of vector field topology: methods are presented for the visualization and analysis of such flow phenomena. Lagrangian coherent structures represent an alternative to vector field topology which is, in contrast, also easily applicable and interpretable for time-dependent vector fields. First, a method based on adaptive mesh refinement for ridge extraction is presented for the efficient extraction of single time frames of Lagrangian coherent structures. Finally, a method based on the advection of computational grids is presented, which exploits temporal coherence for the extraction of time series of Lagrangian coherent structures.

# Kurzfassung

Vektorfelder spielen eine wichtige Rolle in vielen Bereichen der Forschung und Entwicklung. Diese Arbeit entstand im Kontext eines Projekts mit zwei Industriepartnern, welche im Bereich des Turbomaschinenbaus aktiv sind und ist deshalb auf dieses Gebiet ausgerichtet. Nichtsdestotrotz behandelt ihr erster Teil allgemeine Strömungen von Fluiden während ihr zweiter Teil noch universeller ist: er hat Vektorfelder im Allgemeinen zum Thema.

Das Design von Turbomaschinen blickt auf eine lange Geschichte zurück. Obwohl ein hohes Mass an Effizienz bereits in vielen Bereichen erreicht wurde, wird immer noch viel Aufwand betrieben, die Effizienz einige Prozent weiter zu steigern. Obwohl dies auf den ersten Blick ein wertloses Unterfangen zu sein scheint, kann es sich substantziell auf den Ertrag dieser Maschinen auswirken. Es zahlt sich insbesondere im Gebiet der Wasserkraftwerke aufgrund ihrer langen Betriebszeiten aus. Der Betrieb von Turbinen hängt von vielen Faktoren ab, viele davon zählen zum Bereich der Fluidodynamik. Hinsichtlich des Designs erkennt man einen klaren Trend weg von Experimenten hin zu Strömungssimulationen. Dies gilt besonders für diese Jahrzehnte, da Rechenleistung und Simulationen eine Güte erreichen, welche Experimente fast überflüssig machen. Dies führt zu der Situation, dass zuverlässige numerische Daten einfach verfügbar aber aufgrund ihrer Grösse und Komplexität oft schwer interpretierbar sind. Dies lässt die Bedeutung der numerischen Strömungsvisualisierung und Analyse stetig wachsen. Das Hauptaugenmerk dieser Arbeit ist das Vorwärtsbringen der numerischen Strömungsvisualisierung hinsichtlich der Bedürfnisse und Möglichkeiten des modernen Designs von Turbomaschinen.

Diese Arbeit konzentriert sich auf zwei Gebiete der numerischen Strömungsvisualisierung: Wirbel und die globale Struktur von Strömungen. Obwohl Wirbel in vielen Bereichen der Natur nützlich sind, von der Entstehung von Sonnensystemen über den Flug von Libellen zu der Physik unseres Stoffwechsels, neigen sie dazu den Einsatz von Technologie zu beeinträchtigen. Obwohl es in der Technik Bereiche gibt, welche von Wirbeln profitieren, wie zum Beispiel die chemische Produktion, verringern sie die Leistung und Lebensdauer in vielen anderen, insbesondere wenn vergleichsweise hohe Geschwindigkeiten im Spiel sind, wie in der Fahrzeugtechnik und Energiewandlung. In dieser Arbeit werden zwei Ansätze für die Analyse von Wirbelentwicklung und Wirbeldynamik im Allgemeinen vorgestellt. Der erste basiert ausschliesslich auf der instantanen Struktur der Rotation des Geschwindigkeitsfeldes, der sogenannten Vortizität, während der zweite deren zeitlichen Transport mittels Advektion und Diffusion untersucht. Dies kann

bei beidem, der Erzeugung wie auch der Verhinderung von Wirbeln sowie deren Kontrolle in beider Hinsicht helfen, abhängig von den Anforderungen und Möglichkeiten.

Ein Verständnis der globalen und semi-globalen Struktur von Strömungen, oder Vektorfeldern im Allgemeinen, erlaubt ein tieferes Verständnis vieler Mechanismen wie zum Beispiel Transport, Mischung und Verweildauer und erlaubt deshalb ein höheres Mass an Kontrolle und somit gezieltere Forschung und Entwicklung. Diese Arbeit behandelt dieses Gebiet in zwei Unterkategorien: Vektorfeld-Topologie und Lagrange'sche kohärente Strukturen. Als Einführung wird gezeigt, wie die Strömungsvisualisierung durch topologiebasierte Auswahl von Stromlinien und Stromflächen profitieren kann. Ein spezieller Wirbeltyp, der sogenannte Ringwirbel, ist das Forschungsthema der restlichen Forschung im Bereich der Vektorfeld-Topologie: wir stellen Methoden zur Visualisierung und Analyse solcher Strömungsphänomene vor. Lagrange'sche kohärente Strukturen stellen eine Alternative zur Vektorfeld-Topologie dar, welche, im Gegensatz, auch auf zeitabhängige Vektorfelder einfach anzuwenden und zu interpretieren ist. Zuerst stellen wir eine Methode zur effizienten Extraktion von Lagrange'schen kohärenten Strukturen in isolierten Zeitfenstern vor, welche auf adaptiver Verfeinerung von Gittern zur Extraktion von Graten basiert. Schlussendlich stellen wir eine Methode für die Extraktion von Zeitserien von Lagrange'schen kohärenten Strukturen vor, welche die zeitliche Kohärenz ausnutzt, indem sie das Rechengitter advektiert.

# Acknowledgments

First of all I want to thank my advisor Ronald Peikert and my supervisor Markus Gross. They both gave me highest freedom to do research but were always there for help, discussion, and guidance. With their friendliness, openness, and expertise they made these three years a wonderful place to research and learn.

Special thanks go to Ronald Peikert for all the discussions during lunch break, his great expert knowledge and diligence, and his invaluable criticism. Without him, this thesis would never have been written. I also want to thank Matthias Teschner, the advisor of my diploma thesis, for his support and faith.

Further, I want to thank Christophe Bron. He guided me to the trail of research and encouraged me to study. These ten years of collaboration were very fascinating and motivated me to do my studies and finally to start this thesis.

I also want to thank all the members of the Computer Graphics Laboratory for their support and affection. Last, but not least, I want to thank my family and friends for their support during these years.

This work was funded by KTI/CTI grant 7338.2 ESPP-ES.



# Contents

<b>Abstract</b>	<b>v</b>
<b>Kurzfassung</b>	<b>vii</b>
<b>Acknowledgments</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Contributions . . . . .	2
1.3 Organization . . . . .	4
1.4 Notation . . . . .	4
1.5 Publications . . . . .	4
<b>2 Fundamentals and State of the Art</b>	<b>7</b>
2.1 Basics . . . . .	7
2.1.1 Vector Fields . . . . .	8
2.1.2 Representations . . . . .	8
2.1.3 Flow Physics . . . . .	11
2.1.4 Vortices . . . . .	13
2.1.5 Background on Vortices and Vorticity . . . . .	15
2.1.6 Vortex Criteria . . . . .	17
2.2 Volumetric Approaches . . . . .	20
2.3 Glyphs . . . . .	20
2.3.1 Vector Glyphs . . . . .	20
2.3.2 Tensor Glyphs . . . . .	21
2.4 Integral Curves . . . . .	21
2.4.1 Path Lines . . . . .	21
2.4.2 Streamlines . . . . .	22
2.4.3 Streak Lines . . . . .	23
2.4.4 Time Lines . . . . .	23
2.4.5 Eigenvector Field Lines . . . . .	24
2.4.6 Properties and Issues . . . . .	25
2.4.7 Placement of Integral Curves . . . . .	25
2.4.8 Integral Surfaces . . . . .	27
2.5 Vector Field Topology . . . . .	29

2.5.1	Critical Points . . . . .	30
2.5.2	Periodic Orbits and Poincaré Maps . . . . .	33
2.5.3	Separatrices and Manifolds . . . . .	34
2.5.4	Limit Sets and Limit Cycles . . . . .	37
2.5.5	Applications of Vector Field Topology . . . . .	38
2.5.6	Tensor Field Topology . . . . .	39
2.5.7	Lagrangian Coherent Structures . . . . .	40
2.6	Feature-Based Methods . . . . .	42
2.6.1	Ridges, Valleys, and Watersheds . . . . .	42
2.6.2	Vortex Core Lines . . . . .	43
2.6.3	Manifolds of Attachment and Separation . . . . .	45
<b>I</b>	<b>Flow Physics</b>	<b>47</b>
<b>3</b>	<b>Instantaneous Vorticity</b>	<b>49</b>
3.1	Analysis of Vortex Cores . . . . .	50
3.2	Analysis of Vorticity Magnitude . . . . .	50
3.3	Vorticity Field Line Placement . . . . .	52
3.4	Results . . . . .	55
3.4.1	Vortex Analysis . . . . .	57
3.4.2	Vorticity Field Lines . . . . .	58
3.5	Discussion . . . . .	61
<b>4</b>	<b>Vorticity Transport</b>	<b>63</b>
4.1	Theory . . . . .	65
4.1.1	Vorticity Equation for Uniformly Viscous Flow . . . . .	65
4.1.2	Vorticity Equation for Non-Uniformly Viscous Flow . . . . .	67
4.2	Vorticity Transport Analysis Tools . . . . .	68
4.2.1	Quantities . . . . .	68
4.2.2	Path Line Plots . . . . .	69
4.2.3	Striped Path Lines . . . . .	70
4.2.4	Complementary Visualization Techniques . . . . .	71
4.2.5	Methodology . . . . .	74
4.3	Results . . . . .	75
4.3.1	Separation Vortex . . . . .	75
4.3.2	Recirculation and Vortex . . . . .	76
4.3.3	Vortex at Bifurcation . . . . .	77
4.3.4	Vorticity Stretching at Boundary Layer Separation . . . . .	78
4.3.5	Transient Vortex Rope . . . . .	79
4.4	Discussion . . . . .	80
<b>II</b>	<b>Flow Topology</b>	<b>83</b>
<b>5</b>	<b>Invariant Manifolds in Flow Fields</b>	<b>85</b>

5.1	Topology of Divergence-Free Vector Fields . . . . .	86
5.1.1	Saddle and Twisted Saddle Periodic Orbits . . . . .	86
5.1.2	Center Periodic Orbits . . . . .	87
5.2	Topology near No-Slip Boundaries . . . . .	87
5.2.1	Critical Points on No-Slip Boundaries . . . . .	89
5.3	Results . . . . .	90
5.4	Discussion . . . . .	91
<b>6</b>	<b>Vortex Rings</b>	<b>93</b>
6.1	First Look by Streamline Sampling . . . . .	97
6.2	Unsteadiness and Compressibility . . . . .	99
6.3	Analytical Vortex Ring Model . . . . .	101
6.4	Detection of Vortex Rings . . . . .	104
6.5	Boundary Surface . . . . .	106
6.6	Islands of Stability . . . . .	107
6.7	Divergence in Vortex Rings . . . . .	109
6.8	Results . . . . .	109
6.8.1	Francis Draft Tube . . . . .	110
6.8.2	Intake of River Power Plant . . . . .	111
6.9	Discussion . . . . .	112
<b>7</b>	<b>Filtered AMR Ridge Extraction</b>	<b>115</b>
7.1	Height Ridges . . . . .	116
7.1.1	Ridges in Discrete Data . . . . .	116
7.1.2	Ridge Filtering . . . . .	117
7.2	Filtered AMR Extraction of Ridges . . . . .	118
7.2.1	Algorithm . . . . .	119
7.2.2	Implications for Fields Based on Local Operators . . . . .	123
7.3	Discussion . . . . .	123
<b>8</b>	<b>Lagrangian Coherent Structures</b>	<b>125</b>
8.1	Lyapunov Exponent . . . . .	126
8.1.1	Finite-Time Lyapunov Exponent . . . . .	126
8.1.2	Finite-Size Lyapunov Exponent . . . . .	128
8.1.3	Finite-Time Lyapunov Exponent Maximum . . . . .	129
8.1.4	Separation and Attachment Lines . . . . .	130
8.2	Results . . . . .	130
8.2.1	Pelton Turbine . . . . .	130
8.2.2	Hydropower Plant Intake . . . . .	134
8.2.3	Francis Turbine . . . . .	135
8.3	Discussion . . . . .	135

<b>9</b>	<b>Grid Advection for Efficient FTLE Computation</b>	<b>137</b>
9.1	Motivation . . . . .	138
9.2	Method . . . . .	139
9.2.1	Grid Initialization . . . . .	139
9.2.2	Grid Adaptation . . . . .	142
9.2.3	Grid Advection . . . . .	143
9.3	Results . . . . .	145
9.3.1	Flow around a Cuboid . . . . .	145
9.3.2	Intake of a Power Plant . . . . .	146
9.4	Discussion . . . . .	147
<b>10</b>	<b>Conclusion</b>	<b>149</b>
10.1	Overview of Contributions . . . . .	149
10.2	Discussion . . . . .	150
<b>A</b>	<b>Notation</b>	<b>153</b>
	<b>Bibliography</b>	<b>155</b>
	<b>Index</b>	<b>171</b>
	<b>Curriculum Vitae</b>	<b>177</b>

# Chapter 1

---

## Introduction

This chapter gives an overview of the overall motivation of the thesis and states the resulting contributions to the field of computational flow visualization. It also gives an overview of its topics and the organization. Although the methods presented in this thesis originated in the context of turbo-machinery, they are applicable to a much wider field. Those contained in the first part of the thesis are applicable to general fluid flows, whereas those in its second part address the analysis of vector fields in general.

Turbo-machinery engineering tends to adhere to well-established but often simple computational flow visualization techniques such as color-coded cross sections of quantities of interest, streamlines, and path lines. Reasons for this circumstance are that these techniques are easy to implement and therefore widely available, easy to apply, robust, and allow easy and long-time approved interpretations. However, due to the increasing complexity of *computational fluid dynamics* (CFD) results and the involved phenomena, this is often tedious and delivers only a restricted view which has to be often virtually integrated by the mind of the investigator.

One aim of this thesis is therefore to give engineers tools at hand that enable them to solve their tasks more efficiently and even inspire new questions and new ideas. We also hope that this will encourage engineers to be more interested in the field of computational flow visualization research.

### 1.1 Motivation

The definition, detection, and quantification of vortices and related processes has a long scientific history. Surprisingly, no satisfactory definition of what a vortex exactly is has been come up with until today. Still there exists a multitude of definitions, or criteria, none of which unfortunately turned out to be appropriate in all cases of vortical flow. Moreover, many of these criteria not only disagree on their existence, but also exhibit mutually inconsistent results. One of the most

old-established and still widely accepted quantities involved in the research on vortical flow is *vorticity*. The first part of this thesis focuses on the visualization of vorticity with the goal to improve the understanding of the mechanisms involved in vortex creation and vortex dynamics in general.

*Vector field topology*, on the other hand, also looks back on a long tradition in the context of dynamical systems theory and vector fields in general. It mainly consists in the identification of special *streamlines* and among others, it allows one to partition the domain into regions of qualitatively different behavior. Introduced to the visualization community two decades ago, it has proven useful in many applications and is still subject to research. Probably the main deficiency of this concept is, however, that it is based on streamlines, or trajectories in autonomous dynamical systems, meaning that it does not allow for an in-depth analysis of time-dependent vector fields; these have to be analyzed at isolated “frozen” times. It has therefore been the aim for a long time to come up with a variant of vector field topology that accounts for time-dependence in vector fields. To some extent such an alternative has been found in the concept of *Lagrangian coherent structures* (LCS), present as ridges in the *finite-time Lyapunov exponent* (FTLE). A major drawback with this approach is, however, its almost prohibitive computational cost: the spatiotemporal domain needs to be sampled by a dense set of *path lines* at high resolution. The second part of this thesis focuses firstly on visualization methods utilizing vector field topology with the goal to conveniently obtain expressive visualizations in general and also in the special case of the so-called vortex ring phenomenon. Secondly, it addresses the extraction of Lagrangian coherent structures with the main goal of accelerating their computation.

## 1.2 Contributions

Research on vortex analysis on the one hand, and research on the structure of vector fields in the sense of vector field topology both are established and ongoing. Although the former mostly belongs to the field of fluid dynamics and the latter also to mathematics, both are object of research in the field of visualization too. The focus of this thesis was the development of methods for both better understanding of vortical flow and improved analysis of the global and semi-global space-time structure of vector fields, resulting in the following contributions:

**Vorticity field line placement (Chapter 3, [120]):** A large part of the research of vortical flow is based on vorticity. However, vorticity is present in both real vortices and non-vortical shear flow. An important concern with many vortex criteria is therefore to disambiguate this property. Therefore, vorticity is usually looked at separately at these two extremes. Visualizing the complete vorticity field by field lines similar to those used for magnetic fields shows interrelations of these two

phenomena and can also serve as supporting information if scalar vortex criteria are mapped to the field lines by color.

**Vorticity transport analysis (Chapter 4, [121]):** The above approach of visualizing vorticity field lines has one major drawback: because vorticity is an instantaneous quantity, the resulting visualization fails to visualize temporal processes such as the advection of vorticity in time-dependent flow and makes it even hard to interpret it in steady flow. Furthermore, there are other mechanisms involved such as diffusion of vorticity which cannot be adequately visualized by the former method. Therefore, a method is presented that allows to inspect regions of vortical flow in terms of the origination of vorticity and the involved physics on its way to the vortical region as well as inside the vortical region itself. This allows for a deeper understanding and analysis of vortex dynamics regarding a particular region, and in the focus of this thesis, it, e.g., allows the researcher to develop procedures for vortex control.

**Visualization of vortex rings (Chapter 5, Chapter 6, [104–106]):** It was the vortex breakdown phenomenon that guided us to the concept of Lagrangian coherent structures and in this chapter we present different algorithms for the visualization of vortex breakdown bubbles. They allow for an automatic extraction of these features and hence for a quick analysis of the large-scale behavior of vector fields in that context, e.g., reduced throughput due to their blocking nature. In the smaller scale, we provide techniques that reveal the organization of breakdown bubbles, which allows for distinguished examination, and can serve as a basis for their qualitative and quantitative analysis.

**Filtered AMR Ridge extraction (Chapter 7, [118]):** A contribution for accelerated ridge surface extraction is presented in this chapter: a ridge surface extraction technique that constrains sampling to regions containing ridges by adaptive mesh refinement. This way, when applied for the computation of FTLE ridges, the computationally very expensive procedure of FTLE evaluation can be substantially accelerated and hence making FTLE analysis more applicable in everyday research.

**Extraction of Lagrangian coherent structures (Chapter 8, [118, 119]):** Here we explain the concept of FTLE and LCS, apply and evaluate the filtered AMR ridge extraction in that domain, propose an approach for the computation of the FSLE, a concept related to FTLE, and propose another variant going one step further in making the concept parameter-free, i.e., avoiding the prescription of a time scope for the analysis.

**Grid advection for Lagrangian quantities (Chapter 9, [122]):** Since LCS change over time in time-dependent vector fields, it is often necessary to investi-

gate time series, i.e., animations, of FTLE ridges. To lower the computational cost of this process and hence making it better applicable in research and development, we present a method that exploits temporal coherence by advection of the FTLE sampling grid.

## 1.3 Organization

This thesis covers two different areas of research: the visualization of physics of steady and unsteady flows, covered by the first part, and visualization of topology of steady and unsteady vector fields, addressed by the second part.

In the first part, we concentrate on the physical properties of flows, an aspect often neglected in computational flow visualization. There arise some implications and issues in this field. One example is the need for additional variables such as wall distance and viscosity. It is also often important to know how the fluid was modeled, more precisely, which simulation method and which turbulence model have been used, possibly leading to additional variables such as eddy viscosity and modified pressure. Interpolation functions and residual errors of the CFD simulation also have to be often taken into account.

The second part is more generic: it builds on velocity alone. Therefore these methods are applicable to any vector field, e.g., arising from dynamical systems.

## 1.4 Notation

Most *emphasized* terms together with many other are listed in the index and are represented as hyperlinks. There, bold page numbers indicate the location where they are introduced. Please use this mechanism as a replacement for references.

## 1.5 Publications

This thesis is based on the following publications in peer-reviewed conference proceedings and journals:

Sadlo, F., Rigazzi, A., and Peikert, R. Time-Dependent Visualization of Lagrangian Coherent Structures by Grid Advection. In *Topological Methods in Data Analysis and Visualization (Proceedings of Topology-Based Methods in Visualization 2009)*, to appear. Springer, 2011.

Peikert, R. and Sadlo, F. Topologically Relevant Stream Surfaces for Flow Visualization. In *Proceedings of Spring Conference on Computer Graphics*, pages 43–50, 2009.

Peikert, R. and Sadlo, F. Height Ridge Computation and Filtering for Visualization. In *Proceedings of IEEE VGTC Pacific Visualization Symposium 2008*, pages 119–126, 2008.

Peikert, R. and Sadlo, F. Visualization Methods for Vortex Rings and Vortex Breakdown Bubbles. In *Proceedings of EuroVis 2007*, pages 23–25, 2007.

Peikert, R. and Sadlo, F. Flow Topology Beyond Skeletons: Visualization of Features in Recirculating Flow. In *Topology-Based Methods in Visualization II (Proceedings of Topology-Based Methods in Visualization 2007)*, pages 145–160. Springer, 2008.

Sadlo, F. and Peikert, R. Visualizing Lagrangian Coherent Structures and Comparison to Vector Field Topology. In *Topology-Based Methods in Visualization II (Proceedings of Topology-Based Methods in Visualization 2007)*, pages 15–30. Springer, 2008.

Sadlo, F. and Peikert, R. Efficient Visualization of Lagrangian Coherent Structures by Filtered AMR Ridge Extraction. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1456–1463, 2007.

Sadlo, F., Peikert, R., and Sick, M. Visualization Tools for Vorticity Transport Analysis in Incompressible Flow. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):949–956, 2006.

Peikert, R. and Sadlo, F. Topology-guided Visualization of Constrained Vector Fields. In *Topology-Based Methods in Visualization (Proceedings of Topology-Based Methods in Visualization 2005)*, pages 21–34. Springer, 2007.

Sadlo, F., Peikert, R., and Parkinson, E. Vorticity Based Flow Analysis and Visualization for Pelton Turbine Design Optimization. In *Proceedings of IEEE Visualization*, pages 179–186, 2004.

Related publications:

Fuchs, R., Peikert, R., Sadlo, F., Alsallakh, B., and Gröller, M. E. Delocalized Unsteady Vortex Region Detectors. In *Proceedings VMV 2008*, pages 81–90, 2008.

Fuchs, R., Peikert, R., Hauser, H., Sadlo, F., and Muigg, P. Parallel Vectors Criteria for Unsteady Flow Vertices. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):615–626, 2008.

Thürey, N., Sadlo, F., Schirm, S., Müller-Fischer, M., and Gross, M. H. Real-time simulations of bubbles and foam within a shallow water framework. In *Symposium on Computer Animation*, pages 191–198, 2007.

Mallo, O., Peikert, R., Sigg, C., and Sadlo, F. Illuminated Lines Revisited. In *Proceedings of IEEE Visualization*, pages 19–26, 2005.



## Chapter 2

---

# Fundamentals and State of the Art

There has been a considerable development in the field of computational flow visualization in the last decades. Many flow aspects have been addressed and many new techniques have been developed that do not only allow researchers and engineers to produce results that could be achieved in real flow visualization experiments, but also such that could not, or not easily, be achieved in reality.

## 2.1 Basics

Computational flow visualization is often based solely on the vector field, i.e., on velocity. In this case it typically generalizes to the wider field of vector field visualization. Sometimes however, intrinsic properties of fluids come into account, such as zero divergence in case of incompressible fluids or vanishing curl. Other physical mechanisms in fluids are often reflected by respective laws, which in turn may involve additional constants. One important example in the field of fluid dynamics is viscosity, reducing velocity gradients and responsible for dissipative effects, such as the transformation of turbulence into heat. The first part of this thesis concentrates on the analysis and visualization of vortices. First, we will have a structural look at vorticity. This can be achieved by taking velocity alone into account. Later, we will analyze the dynamics of vorticity. This, in contrast, can only be achieved for many simulated flows if the used turbulence model is taken into account.

## 2.1.1 Vector Fields

A vector field is a vector-valued function defined on a domain  $D \subset \mathbb{R}^n$ . In general, it assigns to each position  $\mathbf{x} = \mathbf{x}_i = (x_1, x_2, \dots)^\top$  a vector  $\mathbf{u} = \mathbf{u}_i = (u_1, u_2, \dots)^\top$ :

$$\mathbf{u}(\mathbf{x}) = \begin{pmatrix} u_1(x_1, x_2, \dots) \\ u_2(x_1, x_2, \dots) \\ \vdots \end{pmatrix}$$

or

$$\mathbf{u}(\mathbf{x}) = u_i(\mathbf{x})$$

using *Einstein notation*. In case of 3-dimensional vector fields, this is often written as  $\mathbf{u}(\mathbf{x}) = (u(x, y, z), v(x, y, z), w(x, y, z))^\top$ . There are *steady-state* (*static*, *stationary*) vector fields of the form

$$\mathbf{u}(\mathbf{x})$$

and *unsteady* (*time-dependent*, *transient*) vector fields

$$\mathbf{u}(\mathbf{x}, t)$$

where the domain includes time  $t$ . The dimensions of  $\mathbf{u}$  and  $\mathbf{x}$  are equal, often 2 or 3. However, vector fields can also be defined on manifolds such as curves  $\mathbf{x}(i)$  and surfaces  $\mathbf{x}(i, j)$ . In this case the vector field is a function of these parameters and, potentially, time:

$$\mathbf{u}(i, j, t) = \mathbf{u}(\mathbf{x}(i, j), t).$$

For simplicity of notation, we often use  $\mathbf{u}$  instead of  $\mathbf{u}(\mathbf{x})$  and assume Cartesian coordinates. Unless mentioned explicitly, we additionally assume at least *Lipschitz continuity* in space and time, i.e.,  $\exists C > 0$  such that  $\|\mathbf{u}(\mathbf{x}, t) - \mathbf{u}(\mathbf{x}_0, t_0)\| \leq C\|(\mathbf{x}^\top, t) - (\mathbf{x}_0^\top, t_0)\|$  for all  $(\mathbf{x}^\top, t)$  inside a neighborhood around  $(\mathbf{x}_0^\top, t_0)$ . Lipschitz continuity is stronger than  $C^0$  continuity but weaker than  $C^1$  continuity (continuously differentiable). In case of discrete data, addressed next, Lipschitz continuity is satisfied in the case of linear tensor product interpolation, such as cell-wise bilinear and trilinear interpolation.

## 2.1.2 Representations

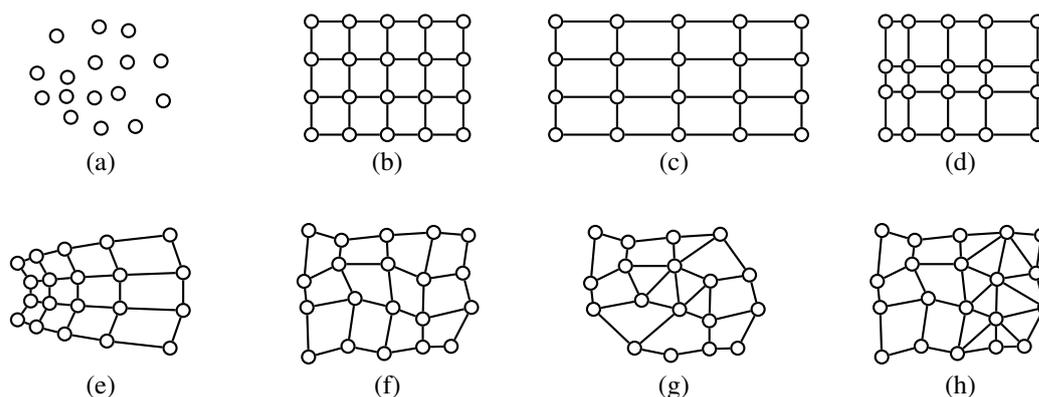
The field concept implies that vector fields are continua, i.e., a value is defined for any point inside the domain, possibly except for isolated point sets. Although this property is naturally fulfilled in many analytic considerations, it requires design choices in many numerical approaches. Analytic representations can be computationally handled, as in the case of software for algebraic computations, but this approach is often expensive, and more importantly, many data are only given in

numerical form. One reason for this circumstance is that it is often simpler to produce numeric results, as in the case of differential equations. Additionally, there are nonlinear problems, such as finding solutions for the Navier-Stokes equations describing the dynamics of fluids, where it is up to now not even clear if closed-form solutions exist in general.

There are different possibilities for the representation of the solution fields. Probably the simplest is to provide a set of positions with corresponding values. This representation is called mesh-less, scattered, or point-based. To render these data functional, a technique has to be chosen for representing the field between the sampled positions. One approach is to place an appropriate basis function at each sample position such that several basis functions cover each position between the samples and contribute to the value at each of these positions, *interpolating* the field. Another approach is to perform meshing, or triangulation, for defining a topological structure of the sampling and to apply methods described next.

A common approach however is to partition the domain during simulation (output) into small regions, the so-called *cells*. The data can then be given at the centers of these cells, or, as a dual approach, at the corners of the cells called *nodes*, leading to *cell-based* or *node-based* representation. One example for a kind of intermediate representation is the so-called *higher-order representation*: here an analytic representation is given inside each of the cells, e.g., by polynomials. Nowadays this approach is increasingly applied for solving problems in mechanics, in the form of the so-called *Discontinuous Galerkin* method. Also applied in CFD, this approach allows to partition the domain into fewer (larger) cells without loss of accuracy at the expense of increased analytical complexity of the representation. Thereby, the cell sizes as well as the analytical complexity can be chosen adaptively, depending on the degree of variation in the result.

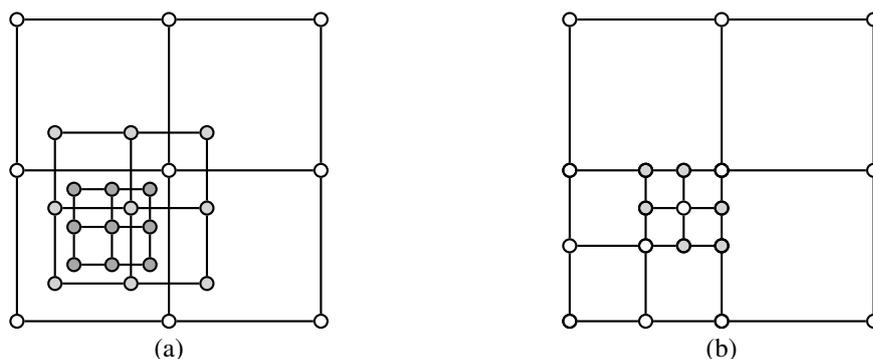
Nevertheless, still most data in the field of CFD are given numerically in a discrete manner: values are given at distinguished positions, namely the cell centers or the nodes. Since many considerations and techniques build on the continuum property of fields, interpolation techniques (or *reconstruction* techniques) have to be applied. Nowadays, linear tensor product interpolation is by far most often utilized in scientific visualization, i.e., linear interpolation in 1D domains, bilinear in 2D, and trilinear in 3D. Often also interpolation schemes of higher order are used, such as cubic or even Taylor series, radial basis functions, or related splines approaches. There is however a caveat: the choice of the reconstruction method is in many cases predetermined by the model/phenomenon that produced the data. In this sense discrete field results are only complete if they are accompanied by the appropriate reconstruction method. In this case these data can typically be interpreted as the aforementioned higher-order representation. Unfortunately, this requirement is nowadays still widely violated; visualization is carried out without knowledge about the correct interpolation scheme. Many researchers and engineers prefer in this case linear tensor-product interpolation because it can be seen as the best trade-off between continuity and conservativeness, meaning that it does not unnecessarily introduce variation, e.g., due to ringing.



**Figure 2.1:** Grid types. (a) Scattered representation (samples without topology). (b) Cartesian grid (isometric). (c) Uniform grid (not isometric). (d) Rectilinear grid. (e) Curvilinear grid (structured grid). (f) Irregular grid (structured grid). (g) Unstructured grid. (h) Hybrid grid (consisting of a quadrilateral and a triangular part).

The collection of cells and nodes is called a *grid*. Grids can be categorized into two major classes: *structured grids* with regular topology and *unstructured grids* where grid topology has to be defined explicitly. In this sense *scattered* data can be classified as a structured grid type (Figure 2.1 (a)). Other structured grid types include *Cartesian*, *uniform*, *rectilinear*, *curvilinear*, and *irregular* (Figures 2.1 (b)-2.1 (f)). An example for an unstructured grid is shown in Figure 2.1 (g) and a combination of a structured and an unstructured grid, called *hybrid*, in Figure 2.1 (h). Besides 2-dimensional cells, unstructured grids nowadays typically consist of tetrahedra, pyramids, prisms, and hexahedra although higher order polyhedra are being increasingly used.

Unstructured grids are widely used in engineering because they provide high flexibility regarding resolution and orientation, i.e., smaller cells can be easily used in regions with detailed phenomena. The cells can even be oriented appropriately, e.g., boundary cells can be constructed consistently with arbitrarily shaped boundaries. In CFD, and in particular in the context of turbo-machinery, this has the advantage that the accuracy of simulation results generally profits from cell alignment with flow direction. There are two further widely used approaches for addressing variable scale in simulations: *multigrid* and *adaptive mesh refinement* (AMR). The multigrid approach simply uses additional meshes of finer resolution at regions of interest inside the coarser grid (Figure 2.2 (a)). Often these meshes are nested, i.e., regions of interest may be covered by several meshes of increasing resolution. The AMR approach follows the same idea but maintains a single grid. Typically the grid is obtained by repeated subdivision of selected cells. A severe problem of AMR grids are however the so-called hanging, or T-, nodes (Figure 2.2 (b)) which may lead to, e.g., inconsistencies (discontinuities) of interpolated values at cell edges because different nodes contribute to the values at both sides of an edge.



**Figure 2.2:** Grid types. (a) Multigrid with three different layers (white, light gray, dark gray). (b) AMR grid with hanging nodes (gray).

To conclude, all mesh-based grids allow for the definition of a *computational* and a *physical space*. Whereas the physical space represents the native parametrization of the domain which is, e.g., used for defining the positions of the grid nodes, the computational space is typically a parametrization of each cell separately by  $[0, 1]^n$  with  $(c_1, c_2, \dots)$ ,  $c_i \in \{0, 1\}$  representing the nodes of the cell. Examples for this parametrization are linear tensor-product interpolation inside quadrilaterals and barycentric interpolation inside triangles. 3D elements exhibiting trilateral as well as quadrilateral faces, such as prisms, are typically parametrized by combination of quadrilateral and barycentric parametrization. Whereas obtaining the physical coordinates of a point given in computational coordinates is straightforward and finding the computational coordinates in uniform and rectilinear grids is simple, it requires *point location* strategies in curvilinear, irregular, and unstructured grids. Computational coordinates are used in many fields such as interpolation, contour extraction, and integral curve extraction.

### 2.1.3 Flow Physics

The probably most basic quantity in fluid dynamics is *velocity*. One derived quantity is the *velocity gradient* (or *Jacobian* of velocity)

$$\nabla \mathbf{u} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{pmatrix} = \partial u_i / \partial x_j.$$

Using also the time derivative of velocity, the *material derivative* (or *substantial derivative*) of velocity represents the *acceleration*:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u},$$

often abbreviated by

$$\frac{D\mathbf{u}}{Dt}.$$

The velocity gradient tensor can be decomposed into its *symmetric part* called *rate of strain tensor*

$$\mathbf{S} = \frac{\nabla \mathbf{u} + (\nabla \mathbf{u})^\top}{2}$$

and its *antisymmetric part* called *vorticity tensor*

$$\mathbf{\Omega} = \frac{\nabla \mathbf{u} - (\nabla \mathbf{u})^\top}{2}.$$

The vorticity tensor is closely related to *vorticity*. Vorticity is the curl of velocity:

$$\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)^\top = \nabla \times \mathbf{u} = \left( \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}, \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}, \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right)^\top$$

and represents a right-handed local rotation around an axis aligned with the vorticity vector and with absolute angular velocity related to vorticity magnitude  $\|\boldsymbol{\omega}\|$ . However, it is worth noticing that the absolute angular velocity of a particle at infinitesimal distance from the reference point of  $\mathbf{\Omega}$  is only  $1/2\|\boldsymbol{\omega}\|$  or in other words, vorticity magnitude is twice the absolute angular velocity. The vorticity tensor  $\mathbf{\Omega}$  is related to vorticity  $\boldsymbol{\omega}$  as follows:

$$\begin{aligned} \mathbf{\Omega} &= \begin{pmatrix} 0 & (\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x})/2 & (\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x})/2 \\ (\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y})/2 & 0 & (\frac{\partial v}{\partial z} - \frac{\partial w}{\partial y})/2 \\ (\frac{\partial w}{\partial x} - \frac{\partial u}{\partial z})/2 & (\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z})/2 & 0 \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}, \end{aligned}$$

which can be also written as  $\Omega_{ij} = -\varepsilon_{ijk}\omega_k/2$  using the permutation symbol  $\varepsilon_{ijk}$ .

In Chapter 3, we will present methods for the visualization of the instantaneous vorticity field structure inspired by common visualizations of magnetic fields. Although this first approach also already gives some insight to vortex dynamics, we present a dedicated method in Chapter 4 for the visualization of vorticity transport in steady and unsteady flow. This second approach can be seen as a complementary technique: whereas the first method mainly visualizes the structure of vorticity, the second method focuses on its advection and diffusion aspects.

We conclude this short overview with a look at some further quantities common in fluid dynamics. We start the tour with a look at the Navier-Stokes momentum equations modeling the dynamics of fluid flows:

$$\frac{D\mathbf{u}}{Dt} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u}. \quad (2.1)$$

The balance of acceleration on the left hand side and the pressure gradient  $\nabla p$  divided by the density  $\rho$  of the fluid on the right hand side is the main mechanism

governing dynamics in typical situations of this model. The second term on the right hand side models the diffusion of momentum according to the kinematic viscosity  $\nu$  related to the dynamic viscosity  $\mu$  by  $\nu = \frac{\mu}{\rho}$ . Viscosity models the internal friction inside fluids and is hence an important cause for the reduction of the velocity gradient across flow direction.

There are many more derived quantities, many of them with a particular meaning and field of application. Some of them are dimensionless and referred as numbers. One important property of flows is their complexity with respect to turbulence. This can be expressed by the *Reynolds number* defined as

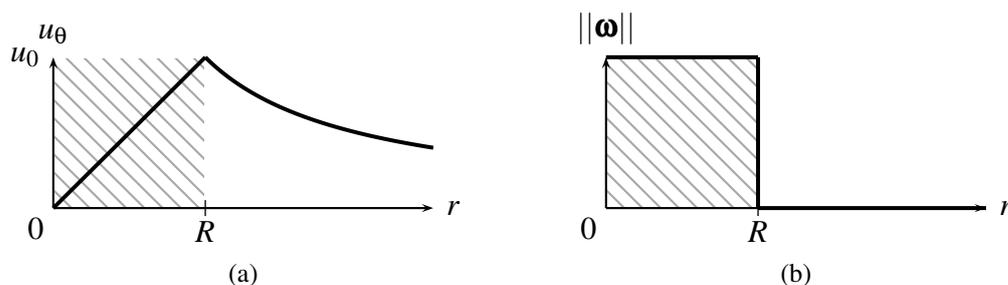
$$Re = \frac{\rho u L}{\mu}. \quad (2.2)$$

It is, too, a dimensionless quantity, describing the “complexity” of the flow, i.e., low numbers indicate laminar flow (flow in parallel layers) whereas high values indicate turbulent flow. The Reynolds number reflects the interesting fact that fluid phenomena typically cannot be simply scaled, e.g., a change of the spatial scale  $L$  necessitates also an adaptation of speed  $u$ , viscosity  $\mu$ , or density  $\rho$ . This is often not easily achieved and a reason why physical special effects in entertainment industry used to lack in plausibility in the days before computer-generated imagery (CGI). Another implication is that wind channels used for flow experiments sometimes require cooling and pressurizing to get into the desired flow regime.

This leads us to the last topic discussed here in the context of flow physics: the *Kolmogorov cascade*. Turbulence typically becomes manifest in a large scale-range of vortices and large vortices contain more turbulent energy than small ones. Kolmogorov identified an energy cascade between these vortices, i.e., a transfer of energy from larger to smaller vortices. Depending on fluid properties, there is a vortex scale at which dissipative effects come into play, i.e., vortical energy is transformed into heat by molecular friction. This scale is called the *Kolmogorov length microscale* and represents the smallest spatial flow scale in this model. There are also corresponding Kolmogorov scales with respect to time and velocity. Because the length scale is typically comparably small, e.g., in the order of a millimeter, a *direct numerical simulation* (DNS) is likely to be out of reach for the next centuries for most practical applications. Therefore, it is nowadays practice to discretize the simulation domains at much lower resolution and to involve *turbulence models* to model the influence of small-scale vortical motion.

## 2.1.4 Vortices

Flow quality mostly determines the operation of turbo-machinery and, generally, it is crucial for many processes in industry and nature. From the rich field of flow physics, we focus on vortex phenomena in the first part of this thesis because vortices tend to affect the operation of turbo-machinery, e.g., by reduced throughput, abrasion, dangerous resonances, and even *cavitation* (abrupt evaporation due to



**Figure 2.3:** Rankine vortex model and vortex core (hatched). (a) Rigid body vortical behavior for  $r \leq R$  and free vortex for  $r > R$ . The vortex core line (vortex axis) is at  $r = 0$ . (b) Vorticity exhibits discontinuity at  $r = R$ .

sudden pressure drop) [11]. On the other hand, vortices can also be beneficial, such as in the case of mixing processes in combustion and production. Either way, in order to diminish or even avoid the creation and intensification of vortices, or to generate beneficial vortices, it is necessary to understand and to identify the processes and mechanisms involved in their creation and dynamics.

A well known difficulty in the discussion of vortices is the lack of a formal definition. Robinson extended Lugt's working definition of a region of closed or spiraling *streamlines* by the requirement that the observer must move with the average velocity measured at the *vortex core* [89, 114]. By its implicit nature, this definition is difficult to apply, and its room for interpretation led to a variety of vortex criteria, even if the scope is narrowed to strictly physical ones. We give an overview in this chapter, and discuss direct visualization of the vorticity field.

An ideal vortex in an inviscid fluid is called a *free vortex* or *irrotational vortex* and exhibits no vorticity at all, except for the singularity at its center where velocity goes to infinity. Its tangential velocity is

$$u_\theta(r) = \frac{\Gamma}{2\pi r}$$

with circulation  $\Gamma$  and radius  $r$ . For real, viscous flows there exists the simple *Rankine vortex* model. Additionally to the free vortex, it contains a *forced vortex* or *rotational vortex* region at its center. This region represents the *vortex core* and contains all the vorticity, which in this case is constant due to the rigid body rotational behavior (Figure 2.3). Its tangential velocity is

$$u_\theta = \begin{cases} \frac{u_0 r}{R}, & (r \leq R) \\ \frac{u_0 R}{r}, & (r > R) \end{cases}$$

with maximum velocity at the peak  $u_0$ , radius  $r$ , and radius of the vortex core  $R$ .

Although there is no precise definition for vortex cores in many real flows, it can be considered a region inside the vortex that exhibits substantial vortical motion, e.g., where vorticity magnitude dominates the shear strain rate in some sense, as in the case of the vortex criterion  $Q$  [62] (see Section 2.1.6).

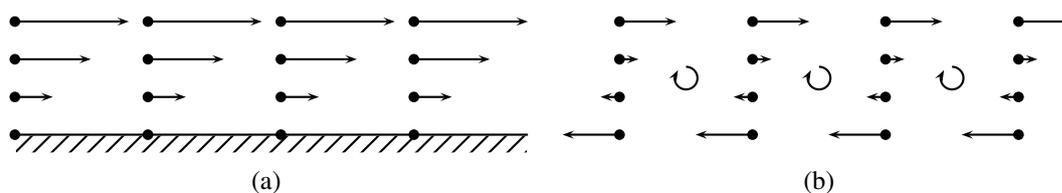
A first and well established step in the analysis of vortices is often the detection of *vortex core lines*. This can be achieved by the methods described in Section 2.6.2. However, due to the local and geometric nature of these methods, they do not give much insight into the dynamics of vortices, subject of Chapter 4.

The extraction of vortices from steady flow data has been investigated by many researchers over the last decades. In a large part of recent literature, vortices are approached via the shape of streamline bundles. The shape of vortices has been described in topological [52] or geometrical terms [80, 145] or by templates [29]. Shape-based methods have been used successfully for various types of, mostly longitudinal, vortices. However, the shape of a streamline pattern is not Galilean invariant, but depends on the frame of reference. Therefore, these methods tend to fail on vortices that need to be observed in a moving frame of reference, i.e., vortices with substantial time dependency. Classical examples are *hairpin vortices* occurring in boundary and mixing layers, and *von Kármán vortex streets*. Methods for their detection should therefore be formulated in a Galilean invariant manner, e.g., not relying on velocity directly. Reportedly [27], the method giving the sharpest extraction results for this type of vortices is the  $\lambda_2$  method [62] described in Section 2.1.6.

Depending on the dominant flow direction through a vortical region, one can classify the corresponding vortex as longitudinal with substantial flow along the core line and transversal, also called tumble vortex, where the dominant direction is perpendicular to it. *Vortex rings* can be seen as a special case of tumble vortices and are the subject of Chapter 6. Basically, these can be understood as vortices that are closed, i.e., that exhibit a closed vortex core line. One prominent example is the so-called *smoke ring*. They are also closely related to the vortex breakdown phenomenon examined in Chapter 6.

## 2.1.5 Background on Vortices and Vorticity

The methods to be presented in the first part of the thesis focus on vorticity. Vorticity relates to the orientation and angular velocity of local rotation and therefore offers a generic and powerful approach for the analysis of the mechanisms involved in vortical motion. Sometimes, it is even used directly as a criterion for the existence of vortices. Depending on the application, the study of vortices can be accomplished by finding regions of high vorticity magnitude. For example, Silver and Wang [137] identified vortices by connected isosurface components of vorticity magnitude and developed an algorithm for tracking them over time. Additionally to vorticity-based methods, there are also vortex core detection methods that are not explicitly based on vorticity, e.g., the  $\lambda_2$  method by Jeong and Hussain [62], described in Section 2.1.6. However, many of these methods are based on the velocity gradient and hence make implicit use of vorticity. A different example is the method presented by Laramée et al. [78]. They use a texture advection method for exploring the relationship between velocity and vorticity.



**Figure 2.4:** Shear flow and vorticity. (a) Shear flow in frame of reference of solid boundary (hatched). (b) Same flow, in frame of reference moving with average flow speed. Differential rotation is apparent.

However, there is a fundamental problem: although vorticity is present in any real vortex in a viscous fluid, it is also present in shear flow which does not exhibit swirling motion at all. One example is a uniformly oriented flow with a non-vanishing velocity gradient (Figure 2.4 (a)). Although it exhibits straight streamlines, it can contain an arbitrary amount of vorticity. This can be seen by subtracting its global average of velocity (Figure 2.4 (b)), an operation not affecting vorticity which is based on spatial derivatives only. In this case, vorticity is oriented into the plane of projection, i.e., perpendicular to velocity, a property indicating shear flow. Especially in ducted viscous flows, most of the overall vorticity is typically contained in the boundary shear layer. This is due to the *no-slip* condition on the boundary that forces the adjacent fluid to adhere. A classical example is the *Hagen-Poiseuille flow* through a cylindrical pipe. It has a quadratic velocity profile and therefore vorticity magnitude increases linearly toward the wall. In more complex flow fields, such as those often encountered in fluid machines, shear flow can separate from the boundary, this way transporting vorticity into the interior of the flow, and possibly developing into a vortex. When a vortex develops from the boundary layer, this can be described by transport and transformation of vorticity. For a better understanding of a flow it is therefore of interest to examine vorticity in all of its different roles. The goal of the first part of this thesis is not primarily the visualization of vortices but of vorticity distribution, orientation and dynamics, and in particular its role in vortex phenomena.

Vorticity has strong analogy with magnetic fields. Because it is divergence-free and exhibits field lines that are closed or of infinite length (or reach the domain boundaries), it can be visualized by *vortex lines* such that their geometric density is proportional to the field magnitude. Therefore we came up with a method that visualizes vorticity using field lines (possibly rendered using the illuminated lines method presented in [93]) where line density represents vorticity magnitude, as commonly used for the visualization of magnetic fields. This approach is described in Chapter 3. As a continuation of this approach, we developed a method for visualizing the transport and transformation of vorticity, presented in Chapter 4. This allows to understand the mechanisms involved in vortex generation and vortex dynamics in general.

## 2.1.6 Vortex Criteria

A number of well known criteria for the presence of a vortex or, more generally, a swirling flow, are derived immediately from the *Navier-Stokes* equations. For incompressible flow, the Navier-Stokes equations are

$$\frac{D\mathbf{u}}{Dt} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u} \quad (2.3)$$

where the left hand side is the material derivative of velocity, i.e., the acceleration,  $\rho$  the constant density,  $p$  is pressure and  $\nu$  is the constant kinematic viscosity.

Some of the criteria discussed in this section relate to eigenvalues and the *tensor invariants* of the velocity gradient  $\nabla \mathbf{u}$ . In 3-dimensional vector fields, the *eigenvalues*  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$ , satisfy the *characteristic equation*

$$\lambda^3 - P\lambda^2 + Q\lambda - R = 0 \quad (2.4)$$

with the tensor invariants  $P = \text{trace} \nabla \mathbf{u} = \nabla \cdot \mathbf{u}$ ,  $Q = \frac{1}{2}(\text{trace}^2 \nabla \mathbf{u} - \text{trace}(\nabla \mathbf{u})^2)$ , and  $R = \det \nabla \mathbf{u}$ .

It is worth noticing that the divergence, curl, and gradient operators, when applied to Eq. 2.3, all yield equations which are relevant for the detection and visualization of vortices.

- The *divergence* of Eq. 2.3 is the scalar equation

$$\nabla \cdot \left( \frac{D\mathbf{u}}{Dt} \right) = -\frac{1}{\rho} \nabla^2 p \quad (2.5)$$

having made use of the (incompressible) continuity equation  $\nabla \cdot \mathbf{u} = 0$ . A positive Laplacian of pressure is a well-known vortex indicator [27]. The pressure Laplacian is up to a constant factor identical to the second invariant  $Q$  of the velocity gradient tensor  $\nabla \mathbf{u}$  which is the basis of Hunt's vortex criterion [57].  $Q$  represents the local balance between the shear strain rate and vorticity magnitude and using the *Frobenius norm*  $\|\mathbf{A}\|_F = \sqrt{\text{trace}(\mathbf{A}^\top \mathbf{A})}$  it can be formulated as follows:

$$Q = \frac{1}{2}(\|\mathbf{\Omega}\|_F^2 - \|\mathbf{S}\|_F^2)$$

with the vorticity tensor  $\mathbf{\Omega}$  and the rate of strain tensor  $\mathbf{S}$  being the antisymmetric and symmetric parts of the velocity gradient  $\nabla \mathbf{u}$ . It is also known as the elliptic version of the Okubo-Weiss criterion by Okubo [102] and Weiss [176]. It has to be noted that it requires the additional condition that pressure needs to be lower than the ambient value. Hunt defines an "eddy" as the region where  $Q$  is positive. In the case of the Rankine vortex, the zero-isosurface of  $Q$  separates the vortex core from the outer region. Eq. 2.5 gives yet another, and quite intuitive, formulation: instead of looking for

low pressure regions, we can equivalently look for regions of high convergence of the acceleration field. This can also be interpreted as regions of high centrifugal force and hence vortical motion. Furthermore, this relates to a very common vortex criterion: the identification of vortices by reduced pressure [99]. Here, the pressure gradient balances the centrifugal force.

- The *curl* of Eq. 2.3 leads to the well-known *vorticity equation*

$$\frac{D\boldsymbol{\omega}}{Dt} = \boldsymbol{\omega} \cdot \nabla \mathbf{u} + \nu \nabla^2 \boldsymbol{\omega}. \quad (2.6)$$

It describes vorticity transport, i.e., the rate of change of a particle's vorticity by *vortex stretching*, *vortex tilting*, and by diffusion of vorticity. The fact that pressure has disappeared makes Eq. 2.6 an attractive alternative to Eq. 2.3 for flow simulations and has led to the so-called vortex methods [20]. In visualization, this equation allows to separately visualize vortex stretching, vortex tilting, and vorticity diffusion, as described in Chapter 4.

- The *gradient* of Eq. 2.3 is the matrix equation

$$\nabla \frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho} \nabla \nabla p + \nu \nabla \nabla^2 \mathbf{u}. \quad (2.7)$$

It is the basis for the  $\lambda_2$  vortex criterion [62] by Jeong and Hussain, which is derived as follows. The starting point for the  $\lambda_2$  method is again the aim to locate regions of reduced pressure. However, since true local pressure minima can be affected by unsteady straining and viscous effects, leading to inconsistency with vortices, the idea is to discard these two influences. As detailed in the context of height ridges (Section 7.1), the pressure Hessian contains the information about local pressure and hence minima, and so this method aims at deriving the Hessian of a “corrected” pressure, liberated from the mentioned influences. The antisymmetric part of Eq. 2.7 is again the vorticity equation Eq. 2.6. In this case it can be discarded because the Hessian is a symmetric matrix. Therefore it suffices to take the symmetric part of Eq. 2.7:

$$\frac{D\mathbf{S}}{Dt} - \nu \nabla \nabla^2 \mathbf{S} + \mathbf{S}^2 + \boldsymbol{\Omega}^2 = -\frac{1}{\rho} \nabla \nabla p$$

where  $\mathbf{S}$  is the symmetric and  $\boldsymbol{\Omega}$  the antisymmetric part of  $\nabla \mathbf{u}$ . After removing terms one (unsteady irrotational straining) and two (viscous effects), the remaining part of the pressure Hessian is, up to a constant factor, the symmetric matrix  $\mathbf{S}^2 + \boldsymbol{\Omega}^2$ . If its three eigenvalues are ordered as  $\lambda_1 \geq \lambda_2 \geq \lambda_3$ , the criterion for  $\lambda_2$  is  $\lambda_2 < 0$  which means that the pressure function graph has positive curvature in at least two orthogonal directions, consistent with Eberly's criterion for codimension 2 ridges (Section 7.1). In the case of the Rankine vortex, the zero-isosurface of  $\lambda_2$  exactly extracts the vortex core.

True local minima would additionally require a zero gradient of the “corrected” pressure. But on the one hand this would restrict the resulting region to the very center of the vortex, i.e., its core line, and on the other hand this additional requirement is often impossible to impose because  $\mathbf{S}^2 + \mathbf{\Omega}^2$  does not fulfill the properties of a Hessian in general: it is only a symmetric matrix and therefore it is usually not possible to derive a consistent gradient or even the “corrected” pressure itself, e.g., by integration. A drawback is however the often encountered inability of separating individual vortices by means of  $\lambda_2$ .

Another vortex criterion called  $\Delta$  [57] is based on the  $Q$  criterion and additionally uses the third tensor invariant called  $R$ . It can be expressed as

$$\Delta = \left(\frac{1}{3}Q\right)^3 + \left(\frac{1}{2}R\right)^2.$$

Again, it indicates a vortex region if  $\Delta > 0$ . The extension of Hua et al. [55] of considering acceleration terms, which includes temporal derivatives and expresses the feature extraction process from the Lagrangian perspective, is another variant related to  $Q$ .

One of the most prominent vortex criteria which is *not* Galilean invariant is *helicity*, the dot product of velocity and vorticity. *Normalized helicity* (also called *helicity density*) is an often used variant obtained by normalizing velocity and vorticity prior to multiplication. Under this aspect, the shear flow sketched in Figure 2.4 exhibits zero helicity and hence a vortex-free region.

There exist several further vortex criteria which are Galilean invariant. A typically required property for a vortex is the presence of complex eigenvalues of the velocity gradient tensor  $\nabla \mathbf{u}$  [18, 46], often referred to as *vortex strength* and formulated as the modulus of the imaginary part of  $\nabla \mathbf{u}$ . Again, this is in practice often not sufficient for isolating distinguished vortices. Other proposed criteria are high vorticity [143], low pressure [99], and the combination of low pressure and high vorticity [6].

One of the recent vortex criteria is the  $Mz$  criterion by Haller [50]. This criterion is not only Galilean invariant, it is even *objective*. This means that it is invariant under any kind of accelerated translation and even accelerated rotation of frame of reference. The invariance to rotation makes it especially appealing for vortex detection in turbo-machinery, where flow often exhibits rotational behavior, e.g., due to a turbine, but does not exhibit vortical motion in a co-rotating frame of reference. Although it might resemble a philosophical question if there is a vortex in such a case, since it depends to some extent on the frame of reference and the questions that the investigator wants to have answered, a vortex criterion that can discriminate between typical vortices and approximate rigid body rotations would be beneficial in some cases. Haller has shown that  $Mz$  does this job at least for analytical vortex examples. Simply spoken,  $Mz$  measures the deviation of the flow from the behavior suggested by the rate of strain. Haller first computes a scalar field and then measures the time a trajectory spends in certain values of that field.

This results in a variant of *hyperbolicity time* and this Lagrangian measure is used to discriminate between hyperbolic (non-vortex) regions and regions that represent vortices. The reader is referred to the cited work for details and to Section 2.6.2 for the extraction of *vortex core lines*.

## 2.2 Volumetric Approaches

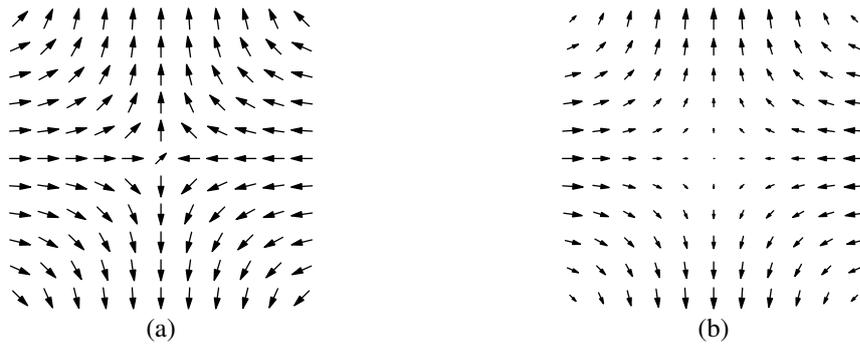
One of the probably simplest ways to visualize vector fields is by extraction of *contours* for 2-dimensional data or *isosurfaces* in the case of 3-dimensional vector fields. These visualizations can be obtained using methods from the family of *marching cubes* [86] algorithms. Similarly, one can use *volume rendering* to visualize these data [96]. Since these methods typically take scalar quantities as input, one needs to derive such from the vector field or use scalar quantities directly available from CFD results. Popular quantities derived from vector fields include velocity magnitude  $\|\mathbf{u}\|$ , vorticity magnitude  $\|\boldsymbol{\omega}\|$ , and the vortex criterion  $\lambda_2$  (see Section 2.1.6). Quantities directly available from CFD solvers and often used for visualization include pressure, temperature, and *turbulent kinetic energy* (TKE). A common approach is also to extract, e.g., isosurfaces from one quantity and to color-code another quantity on the resulting surfaces. There are also approaches for volume rendering of non-scalar quantities such as that for tensors by Kindlmann et al. [69]. Please also see the integration-based approaches in Section 2.4.8.

## 2.3 Glyphs

Glyphs, also called icons, are geometric primitives used for visualizing multiple quantities by means of their orientation, size, shape, and material properties. They are applied in both, *information visualization* and *scientific visualization*. One early example for the former are *Chernoff faces* [17] and for the latter the glyphs proposed by Globus et al. [40] for the characterization of *critical points* in vector fields. A basic overview regarding feature visualization is presented by Post et al. [113].

### 2.3.1 Vector Glyphs

One of the most primitive methods is glyph-based flow visualization. A widely used variant is to plot arrows in the domain of the vector field, called *hedgehog* fields (Figure 2.5). Although the generation and interpretation of this type of visualization is simple, it often suffers from clutter and occlusion, especially if performed on 3D domains. One remedy to these problems is to perform *glyph placement*, a strategy to place glyphs such that clutter is reduced and at the same



**Figure 2.5:** Arrow glyphs visualizing a vector field exhibiting saddle behavior. (a) Normalized field pronounces direction but does not convey magnitude. (b) Arrow sizes scaled by speed show both direction and magnitude. The cosine-shaped velocity profile (in  $x$ - and  $y$ -direction) is apparent.

time glyphs get positioned at positions with higher informational value. See the taxonomy by Ward [170] for further details concerning glyph placement.

## 2.3.2 Tensor Glyphs

Tensor fields are also often visualized using glyphs. A simple example is shown in Figure 2.7 (a). Tensor fields often arise as velocity gradients and from diffusion tensor MRI. De Leeuw et al. [22] presented glyphs for the visualization of velocity gradients. Haber presented tensor glyphs or ellipsoids for tensor visualization [44, 45]. Kindlmann presented regularly placed superquadric tensor glyphs [71] for the visualization of 3-dimensional diffusion tensor MRI data. More recently, Kindlmann et al. presented a glyph placement technique called adaptive glyph packing [72].

## 2.4 Integral Curves

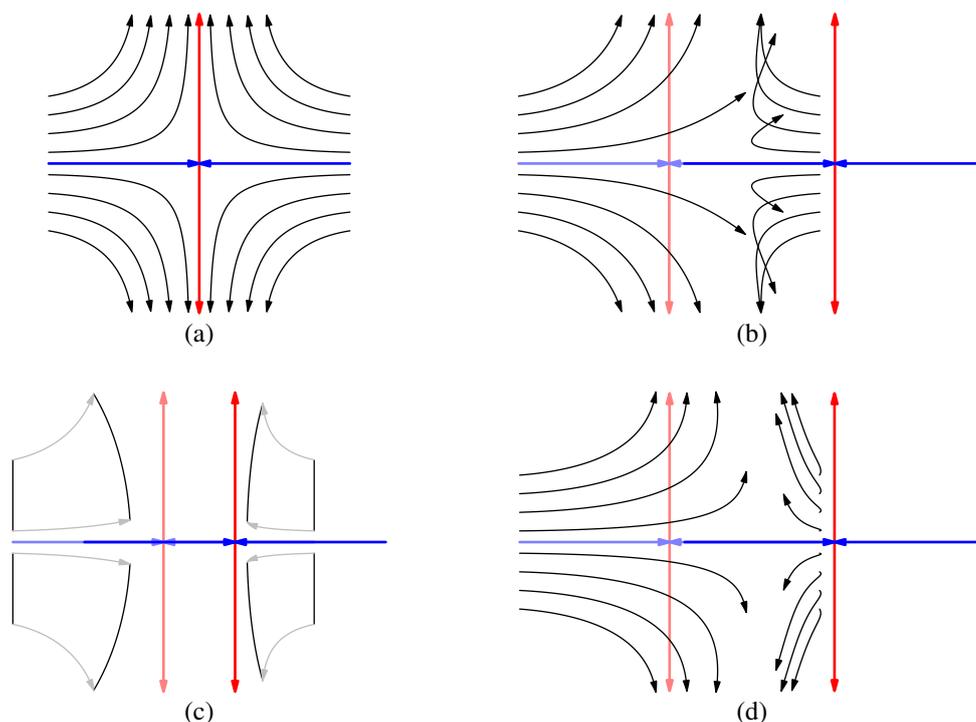
If one interprets the values of a vector field as velocity  $\mathbf{u} = \dot{\mathbf{x}}$ , vector fields can be understood as *ordinary differential equations* (ODE) (and vice versa):

$$\dot{\mathbf{x}}(t) = \mathbf{u}(\mathbf{x}(t), t). \quad (2.8)$$

### 2.4.1 Path Lines

Together with an initial condition  $\mathbf{x}(t_0) = \mathbf{x}_0$  Eq. 2.8 leads to an *initial value problem* (IVP). The solution to this problem is the integral curve

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{t_0}^t \mathbf{u}(\mathbf{x}(\tau), \tau) d\tau.$$



**Figure 2.6:** Vector field from Figure 2.5. (a) Streamlines. (b) Path lines, as field translates from left to right (pale arrows visualize saddle at start). (c) Time lines in same time-varying field but over shorter time interval. (d) Streak lines in field from (b).

Since this curve describes a trajectory (path of a mass-less particle inside a time-dependent flow field), it is called a path line (Figure 2.6 (b)). In a physical experiment, a path line can be obtained by photographic long-time exposure of a marker advected by a flow. In Chapter 4 we present a method for the visualization of vorticity transport based on path lines.

## 2.4.2 Streamlines

Steady vector fields correspond to *autonomous* ODEs:

$$\dot{\mathbf{x}}(t) = \mathbf{u}(\mathbf{x}(t))$$

and the corresponding integral curve

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{t_0}^t \mathbf{u}(\mathbf{x}(\tau)) d\tau \quad (2.9)$$

is a streamline (for velocity fields) or a *field line* (other fields), shown in Figure 2.6 (a). Streamlines (or field lines) can also be computed from unsteady vector fields by taking a “snapshot” at a given time  $t_i$ :

$$\mathbf{u}_{t_i}(\mathbf{x}) = \mathbf{u}(\mathbf{x}(t_i)).$$

With reasonable assumptions on the vector field (*Lipschitz continuity*, assumed throughout this thesis unless explicitly noted), solutions to Eq. 2.9 always exist and cannot cross each other, they are *unique*. Although for unsteady vector fields streamlines are often of lower significance than path lines because they do not describe a true trajectory (it is hard to obtain them in physical flow experiments), they can be appropriate for derived quantities of unsteady vector fields. One example are integral curves of vorticity  $\boldsymbol{\omega}$ , called *vortex lines*, which can be used for the analysis of vortices (Chapter 3 and Chapter 4) and for conservation laws, such as Helmholtz's theorems [77].

### 2.4.3 Streak Lines

Another concept is that of streak lines. Streak lines are generated by continuously releasing particles at a fixed position and taking a snapshot of the generated pattern. One example in physical flow visualization is the release of smoke at distinct points in, e.g., a wind tunnel, see also Figure 2.6 (d). Streak lines starting at position  $\mathbf{y}$  and captured at time  $t_n$  can be generated algorithmically as follows:

1. For each time sample  $t_0, t_1, \dots, t_n$  solve the IVP

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}(\mathbf{x}_i(t), t), \quad \mathbf{x}_i(t_i) = \mathbf{y}.$$

2. Extract from each integral curve  $\mathbf{x}_i(t)$  the point  $\mathbf{x}_i(t_n)$ .
3. Connect these points.

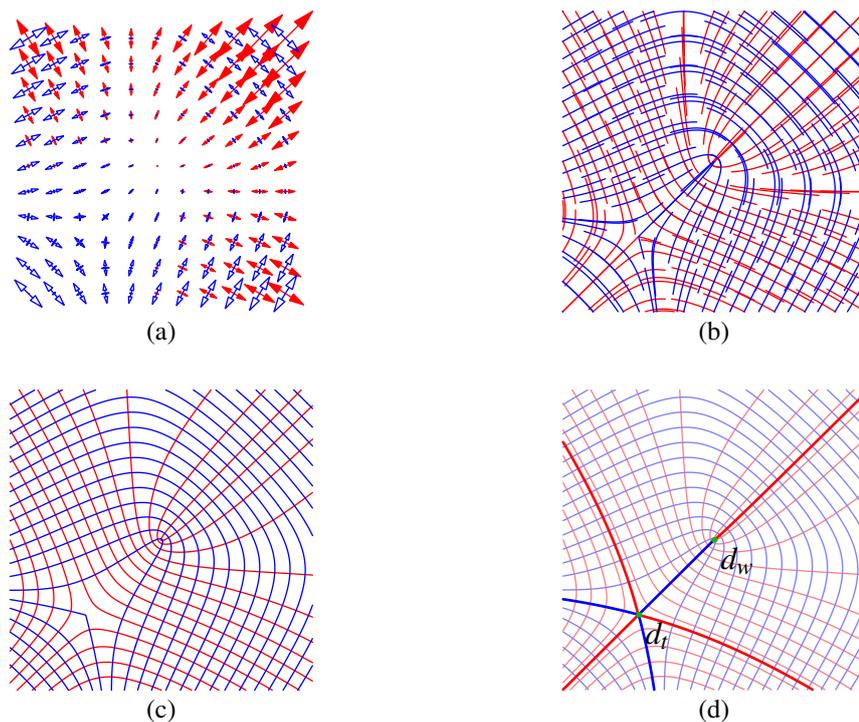
Numerically, the temporal interval between the time samples must be adaptively refined to avoid successive particles diverging too much. Recently, a generalized version of streak lines has been proposed by Wiebel et al. [178]. They abandon the requirement that the source  $\mathbf{y}$  releasing the particles has to be fixed. Interestingly, in the limit case of infinite velocity of the seeding point, generalized streak lines are equivalent to time lines, discussed next.

### 2.4.4 Time Lines

Finally, time lines are obtained by releasing particles placed on a seed curve at distinct times and taking a snapshot of the generated pattern at some time (Figure 2.6 (c)). They can be obtained in physical experiments by placing a thin wire into a fluid and applying current pulses. Each pulse produces fine bubbles by electrolysis which are then advected with the flow. Time lines starting at seeds  $\mathbf{y}_i$  on the curve and captured at time  $t_j$  can be generated algorithmically as follows:

1. For each point sample  $\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_n$  on the seed curve solve the IVP

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}(\mathbf{x}_i(t), t), \quad \mathbf{x}_i(t_0) = \mathbf{y}_i.$$



**Figure 2.7:** 2D tensor field visualization. (a) Simple glyphs for generic tensors (major filled, minor outlined, positive red, and negative blue). (b) Eigenvector field lines (major red and minor blue) seeded on regular grid convey more information but suffer from visual clutter. (c) Line placement by Turk and Banks [164], adapted for tensor fields. (d) Additionally degenerate points of type trisector and wedge (green), with corresponding separatrices (bold red and blue).

2. Extract from the integral curve  $\mathbf{x}_i(t)$  the point  $\mathbf{x}_i(t_j)$ .
3. Connect these points.

The resulting curve is a time line for time  $t_j$ . Similarly to the generation of streak lines, the spatial interval must be adaptively refined to avoid neighboring particles diverging too much.

## 2.4.5 Eigenvector Field Lines

Another example for (instantaneous) field lines from possibly unsteady vector or tensor fields, similar to the vortex lines from Section 2.4.2, are eigenvector field lines (Figure 2.7 (b)–(d)). These are integral curves following one of the eigenvectors (minor, medium, major) of a tensor. Unfortunately there is some confusion of meaning of the term *tensor line*. In some research areas this term is used for integral curves of eigenvector fields whereas in others the definition due to Weinstein et al. [175] is used, where during integration an artificial inertia is modeled to

pass regions of low linear anisotropy. There is, however, the problem that eigenvectors are indistinguishable if the corresponding eigenvalues are equal, leading to infinitely many possible directions. The loci where this can happen are called *degenerate points*, a counterpart to *critical points* (Section 2.5.1), and are subject to *tensor field topology* analysis (Section 2.5.6). In general, eigenvector field lines stop at these points.

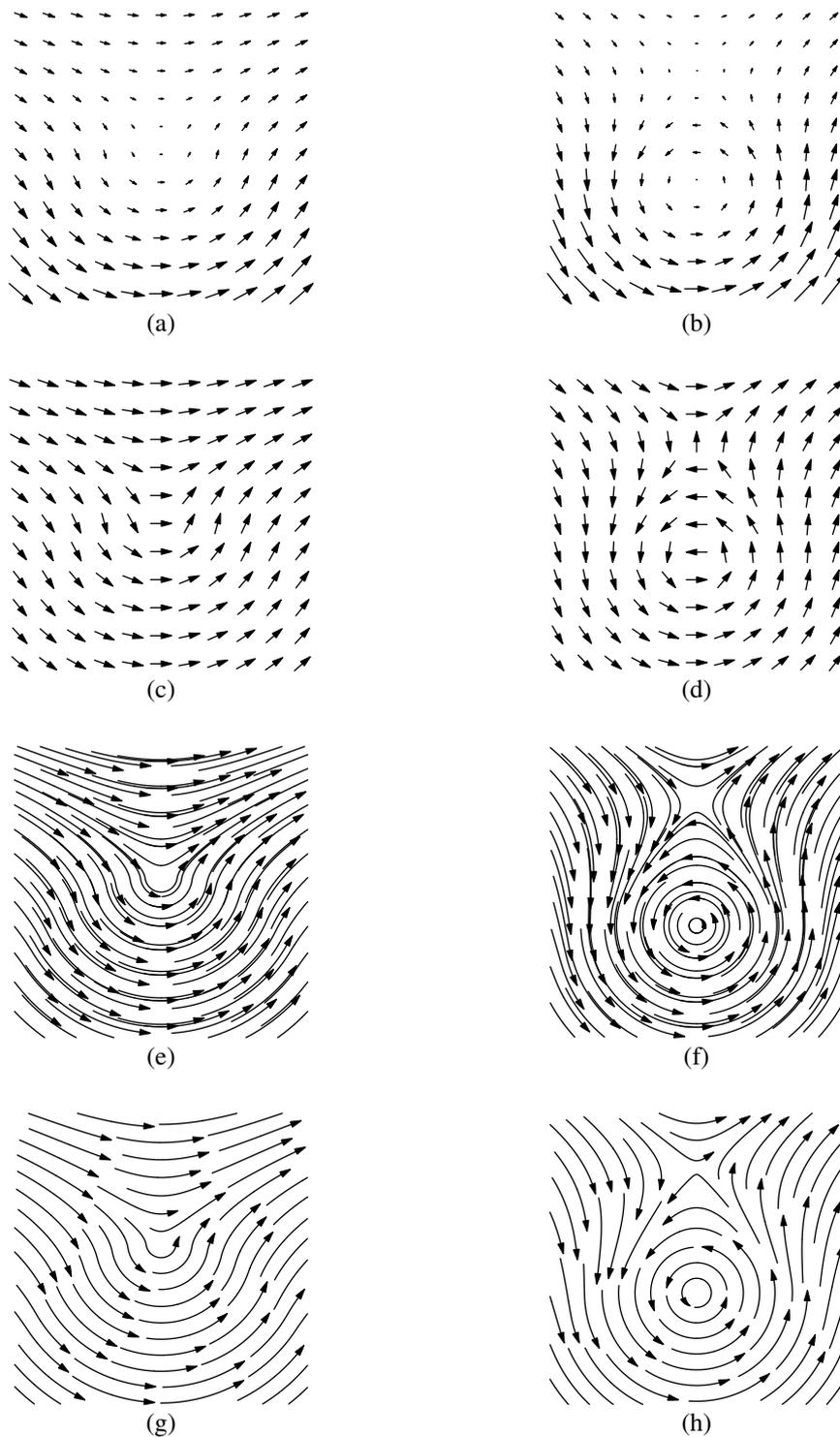
Delmarcelle et al. [24] propose two variants of *hyperstreamlines*: *hyperstreamline tube*, generated by an ellipse sweeping along one of the eigenvector field lines and deforming on its way according to the other eigenvectors and eigenvalues, and *hyperstreamline helices* produced by a cross sweeping along a eigenvector field line and changing according to the other eigenvectors and eigenvalues. Another approach to the visualization of tensor fields is that by Zhang et al. [185] using *stream tubes* and *stream surfaces*.

## 2.4.6 Properties and Issues

As a consequence, computing streak lines and time lines is more expensive than solving a single IVP. Another interesting fact is that streamlines, path lines, and streak lines are identical for steady vector fields. There are several issues when computing streamlines, path lines, or derived curves such as streak lines and time lines. Appropriate integration depends on a correct choice or adaptive control of the step size, the choice of the integration scheme, an efficient point location method (global and local), and the right interpolation scheme. In this thesis, we use fourth-order Runge-Kutta integration scheme and tensor product linear interpolation unless explicitly noted. Due to the overall simplicity of computation, handling, and interpretation of streamlines and path lines, they are widely used in science and engineering and are available in many visualization packages. All four types of lines also give rise to integral surfaces discussed in Section 2.4.8.

## 2.4.7 Placement of Integral Curves

Similarly to the case of glyphs, visualizations using multiple streamlines, such as streamlines seeded on regular grids, can also suffer from clutter and occlusion, especially in 3-dimensional vector fields. One way of handling this problem is to perform *streamline placement*, which tries to place streamlines in a significant manner. The original work by Turk and Banks [164] addresses the placement of streamlines in 2-dimensional vector fields by an iterative image-space method maintaining a density field to achieve a controlled spacing (Figure 3.3, Figure 2.7 (c) and (d), and Figure 2.8). Later single-pass approaches like those by Jobard et al. [64], Fuhrmann et al. [34], Verma et al. [168], and Mattausch et al. [95] and Schlemmer et al. [131] address 2-dimensional vector fields, 3-dimensional vector fields, utilize *vector field topology* (see also the work by Ye et al. [183]), and allow variable line density, respectively.



**Figure 2.8:** Fold bifurcation. Before bifurcation (left column) and after (right column). (a) and (b): arrow glyphs scaled by vector magnitude. (c) and (d): normalized arrow glyphs pronounce orientation but lose speed. (e) and (f): regularly seeded streamlines of normalized field suffer from visual clutter. (g) and (h): streamline placement by Turk and Banks [164]. Separatrices still not apparent (cf. Figure 2.12).

The placement of path lines is less popular, probably because neighboring path lines can represent particle positions at differing times, and possibly more complicated because a path line not only depends on the location but also on the time where it is seeded. One example for path line placement is in the context of path line predicates [126]. To our notion, there is no work concerned with the placement of streak lines. One reason may be that streak lines tend to undergo extensive folding after some time of advection, and therefore, they may be interesting only for short advection times. Therefore, significant visualizations using sets of streak lines can often be achieved by, e.g., seeding them regularly on a predefined curve. Finally, the placement of time lines can be controlled by the location and shape of the seeding curve, and by the temporal release interval. Again, we are not aware of techniques addressing this problem. We refer the reader to the survey by McLoughlin et al. [97] for a further overview. In Chapter 3 we present a method for the placement of vortex lines with line density proportional to vorticity magnitude, resulting in a visualization technique for vortical flow, inspired by the common approach to visualize magnetic fields.

## 2.4.8 Integral Surfaces

One of the most basic but still not widely applied techniques for the visualization of 3D vector fields is that of *stream surface* integration. Stream surfaces and streamlines play an important role in *vector field topology* where they can represent invariant manifolds. Stream surfaces are manifolds that can be created by seeding streamlines along a curve and connecting neighboring streamlines by strips of meshes, resulting in mesh surfaces. Made popular in visualization by Hultquist [56], the reason why it is still not present in many visualization (also called *post-processing*) packages, is that it can fail in some flow configurations, i.e., requires robust algorithms, in particular in the vicinity of critical points (Section 2.5.1).

In the classical stream surface algorithm by Hultquist the stream surface is generated by triangulation between a set of discretized streamlines. Triangle shape is optimized by choosing the shorter of the two possible edges in the process of triangulating between two adjacent streamlines. Triangle size is controlled by seeding new streamlines between existing streamlines or by stopping streamlines. This basic algorithm can be implemented with a depth-first strategy. However, to evaluate the criteria for adding or stopping a streamline, it is more convenient to use a breadth-first strategy where a current “front” is used. Garth et al. [38] addressed some of these problems, e.g., by proposing a refinement criterion based on the angle between adjacent segments of the front, and so did we [108], but there are still few commercial packages that implement these methods. Theisel et al. [150] remarked that Hultquist’s algorithm fails if the tangents of the front are almost in the direction of the vector field, a situation which can arise, e.g., near critical points or periodic orbits. They use as an initial front a line perpendicular to the vector

field. This way, even tightly spiraling streamlines can be handled. However, the choice of the line is critical to avoid cracks or multiple coverings. Also, this approach produces spurious internal boundaries which have to be post-processed for the result to be manifold.

The mentioned techniques can also be based on path lines instead of streamlines, resulting in *path surfaces* that allow a time-dependent view on the vector field. One has to note however, that path surfaces can intersect and can be hard to interpret because they span a time interval. There are also the concepts of *streak surfaces* and *time surfaces* [97] that suffer from similar construction problems and therefore can be addressed by methods inspired by those just mentioned or more advanced ones as that by Krishnan et al. [76]. Sometimes these concepts are visualized without the use of surfaces by simply advecting particles, which however often requires using animations or even stereo rendering in order to reduce visual clutter. A related but static approach is point-based construction of these manifolds, proposed by Schafhitzel et al. [128].

As their constituting curves, stream, streak, and time surfaces are quite simple to interpret. In physical experiments, streak surfaces are often obtained by placing a thin wire into a fluid and applying a voltage to it. This way small bubbles of gases like hydrogen can be generated by electrolysis that advect with the flow. Time surfaces are, however, more difficult to produce in real experiments because placing a fine mesh tends to influence flow behavior.

If only two integral curves are used and kept at constant distance, the resulting surface strips are called (stream) *ribbons*. If, on the other hand, one starts the integral curves on closed seeding curves, one obtains *stream tubes* (made of streamlines) and *vortex tubes* (made of vortex lines), for mentioning those that are most prominent in fluid dynamics due to their relation to conservation principles (see, e.g., the *Helmholtz theorems*).

In some cases, such as the extraction of *separation surfaces* and *attachment surfaces*, the often demanding procedure of stream surface integration can be replaced by the extraction of Lagrangian coherent structures by means of ridge surface extraction from FTLE fields (see Section 2.5.7 and Chapter 8), however usually at higher computational cost. The concept of implicit stream surfaces by van Wijk [166] is to some extent a related approach based on implicit surfaces of scalar fields computed from the flow. The flow volumes approach by Xue et al. [182] renders a closely related field by volume rendering and texture advection techniques. Another related concept is that by Westermann et al. [177] measuring “flow time” and allowing to extract time surfaces by level sets in that field. All three methods in some sense represent a further step in the sequence of streamlines to stream surfaces, i.e., they represent some kind of integral volumes. Finally, in the field of vortex detection, the *hyperbolicity time* and  $M_z$  concepts by Haller [50] (Section 2.1.6) and the delocalized flow criteria by Fuchs et al. [33] are related examples in that field.

## 2.5 Vector Field Topology

Here we give an introduction to selected topics and concepts in the field of vector field topology. Vector field topology reflects the analogy between vector fields, differential equations, and dynamical systems in general, as discussed in Section 2.4. In particular, trajectories represent solutions of initial value problems and are called orbits in this field. In the dynamical systems view, the domain is spanned by parameters describing the state of the system, i.e., the domain represents its degrees of freedom.

Hence, each point represents a parametrization, i.e., a possible state of the system, and the domain is called *phase space*. In physics, the phase space is often spanned by velocity and momentum. In our applications, the space represents true Euclidean space and the notion of state is reflected by the vector field, describing the transition from one state (position) to another. In this context vector fields represent the *extended phase space*. Traditionally, mostly steady vector fields, i.e., autonomous dynamical systems, are examined. Time-dependent setups are often addressed by holding the respective parameter (time) fixed and analyzing the resulting steady vector field. If that parameter is varied, there exist distinguished values where the corresponding topology undergoes substantial qualitative changes. One example is the annihilation or generation of pairs of critical points and corresponding separatrices. These instances are called *bifurcations* and reflect the inherent principle of *index invariance*. The *Poincaré-Hopf index* of a 2D critical point is the count of counterclockwise field rotations as one travels around it in counterclockwise direction. Hence, nodes, foci, and centers have index  $+1$  whereas saddles have  $-1$ . One consequence is that critical points of type saddle and focus can annihilate/originate (Figure 2.12), satisfying the invariance of the total sum of indices.

Vector field topology deals with special types of streamlines and is therefore only capable of giving an instantaneous view to unsteady vector fields. It aims at visualizing the overall structure of vector fields, leading to a condensed representation of the vector field. The current popularity of vector field topology in visualization is to a large extent due to its capability of automatically segmenting 2-dimensional vector fields into regions of qualitatively different flow behavior. Many interesting applications in fluid mechanics can be found in the work by Perry et al. [109]. Vector field topology was introduced to the visualization community by Helman and Hesselink [52] in the context of 2D and 3D [53] flow fields and can be summarized as the use of concepts from the theory of continuous dynamical systems (see, e.g., [43]) in scientific visualization. We give a short overview here, additional information can be found in Chapter 5 and Chapter 6, in the work by Globus et al. [40] and Asimov [3], and more generally, in the book by Abraham and Shaw [2].

## 2.5.1 Critical Points

If a streamline is started at position  $\mathbf{x}_0$  with  $\mathbf{u}(\mathbf{x}_0) = \mathbf{0}$  and consequently  $\mathbf{x}(t) \equiv \mathbf{x}_0$ , ( $t \in \mathbb{R}$ ), the streamline degenerates to a single point. These points are called *stationary points* or *constant orbits*. A stationary point is called *isolated* if it is not adjacent to other stationary points. Furthermore, it is called a *critical point* if the velocity gradient at this point is regular, i.e., has non-zero determinant. Here, we restrict the scope to first-order critical points, i.e., those amenable to analysis by means of linearization. Visualization of higher-order critical points are addressed, e.g., by Scheuermann et al. [129], Tricoche et al. [160], and Weinkauff et al. [173].

A vector field can be linearized around a critical point  $\mathbf{x}_c$ :

$$\mathbf{u}(\mathbf{x}_c + \mathbf{x}) = (\nabla \mathbf{u})\mathbf{x} + O(\mathbf{x}^2).$$

Critical points are *isolated* and the vector field in the vicinity of a critical point takes all possible directions. Critical points can be classified by the eigenvalues of the velocity gradient at that point, more precisely, by the signs of their real parts.

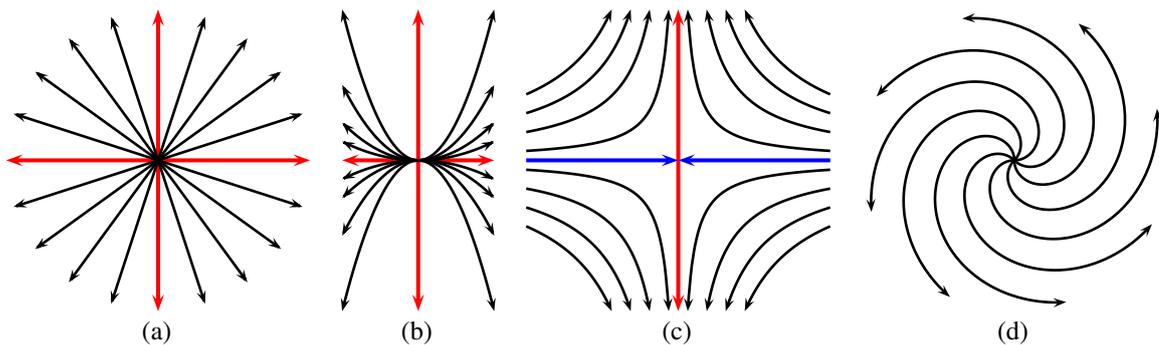
A critical point is called *hyperbolic* if all eigenvalues of the velocity gradient have non-zero real parts. The main implication of the hyperbolicity of a critical point is that it is *locally structurally stable*, i.e., it is stable against small perturbations of the vector field. This means that applying a perturbation does not change the topology of the vector field, or in other words, it does not change the topology of nearby streamlines. For 2-dimensional vector fields, hyperbolic critical points can be classified according to Table 2.1 and Figure 2.9. Conversely, critical points that are locally structurally stable must be hyperbolic. One exception to this rule are critical points with zero real parts (purely imaginary eigenvalues) in 2D divergence-free vector fields. The corresponding flow pattern is called a *center* and is the limit case between a focus sink and a focus source critical point. Although not hyperbolic and not structurally stable in general (they transform to focus sources or focus sinks if a perturbation is applied in general vector fields), these critical points are structurally stable against perturbations in divergence-free fields. Stationary points that are hyperbolic are necessarily isolated. Throughout this thesis, we assume critical points to be hyperbolic, unless stated explicitly.

For 3-dimensional vector fields, hyperbolic critical points can be classified according to Table 2.2 and Figure 2.10. It has been shown by Theisel et al. [149] that tracking of critical points over time in unsteady vector fields can be achieved by solving for integral curves in their so-called *feature flow field*, a vector field with an additional dimension representing time. For a given time step, the respective critical points are simply obtained by intersecting the resulting curves with a hyperplane at constant time. Other approaches for tracking critical points have been presented by Tricoche et al. [163] and by Theisel et al. [152]. The reader is referred to the survey by Post et al. [112] on feature tracking in general.

A first generation of topology-based visualization methods locates, classifies, and displays critical points of the given vector field as point icons. Even if the full set of critical points is used without any type analysis, this strategy was shown to

two real eigenvalues	type
both positive	<i>node source (2 out)</i>
both negative	<i>node sink (2 in)</i>
opposite signs	<i>saddle (1 in, 1 out)</i>
two conjugate complex eigenvalues	type
positive real parts	<i>focus source (2 out)</i>
negative real parts	<i>focus sink (2 in)</i>

**Table 2.1:** Critical point classification in 2-dimensional vector fields, see also Figure 2.9.



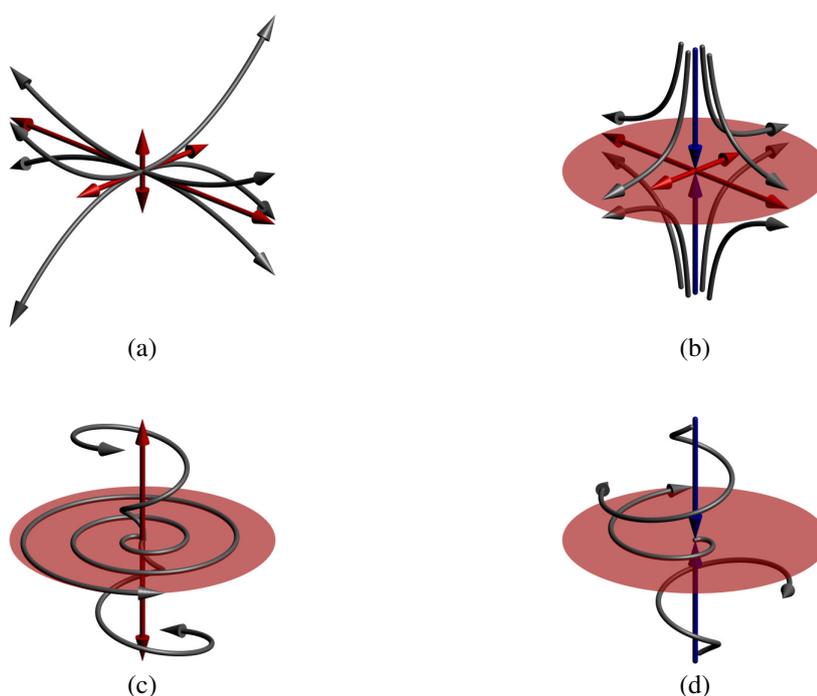
**Figure 2.9:** Critical points in 2D vector fields. Eigenvectors corresponding to positive (red) and negative (blue) eigenvalues. (a) Isotropic source. (b) Node source. (c) Saddle (stable (blue) and unstable (red) manifolds). (d) Focus source. Remaining cases can be obtained by field reversal.

yield effective visualizations by Weinkauff et al. [172]. Alternatively, a visualization of the local topological and geometric flow behavior near critical points can be obtained by displaying icons showing the linearized flow defined by the critical point type and by the eigenvectors of the Jacobian of the vector field [40]. The same information can be used to seed short streamlines near critical points [82,83], giving a slightly more global picture of the flow, possibly rendered as illuminated streamlines [140] for improved perception.

It is interesting to notice that most work done so far in topology-based visualization falls in one of two categories, either giving a global picture of the entire domain or a local picture of neighborhoods of critical points. While global effects are an interesting part of dynamical systems and chaos theory, it can be argued that for flow visualization, they are often less relevant because of issues such as domain boundaries, simulation accuracy, or time-dependence. But also the other extreme, independent visualization of critical points, can be regarded as unsatisfactory, since much of the topological information is left unused. We believe that vector field topology has much to offer for flow structures which fall in between the two extremes. One such structure is the vortex ring, topic of Chapter 6.

three real eigenvalues	type
all positive	<i>source (3 out)</i>
two positive, one negative	<i>1-saddle (1 in, 2 out)</i>
one positive, two negative	<i>2-saddle (2 in, 1 out)</i>
all negative	<i>sink (3 in)</i>
one real and two complex eigenvalues	type
positive real eigenvalue, positive real parts	<i>spiral source (3 out)</i>
positive real eigenvalue, negative real parts	<i>2-spiral saddle (2 in, 1 out)</i>
negative real eigenvalue, positive real parts	<i>1-spiral saddle (1 in, 2 out)</i>
negative real eigenvalue, negative real parts	<i>spiral sink (3 in)</i>

**Table 2.2:** Critical point classification in 3-dimensional vector fields, see also Figure 2.10.



**Figure 2.10:** 3D critical points. The colored arrows represent eigenvectors with length proportional to corresponding positive (red) and negative (blue) eigenvalue. In case of complex eigenpairs, the corresponding eigenplane is represented by a disk, red for positive and blue for negative real parts. (a) Source. (b) Saddle (blue line: 1D stable manifold, red surface: 2D unstable manifold). (c) Spiral source. (d) Spiral saddle (blue line: 1D stable manifold, red surface: 2D unstable manifold). Remaining cases can be obtained by field reversal. See a digital version of the document for the 3D interactive models.

## 2.5.2 Periodic Orbits and Poincaré Maps

Another type of special streamlines, unless the vector field is irrotational, are *periodic orbits* or *closed orbits*. These are streamlines that pass a point  $\mathbf{x}_0$  more than once, in other words:  $\mathbf{u}(\mathbf{x}_0) \neq \mathbf{0}$  and  $\mathbf{x}(t + k\Delta t) = \mathbf{x}_0$ , ( $\Delta t \in \mathbb{R}$ ,  $k \in \mathbb{Z}$ ). Wischgoll et al. [179, 180] presented methods for detecting periodic orbits and Tricocche et al. [162] presented a method for tracking them over time. Periodic orbits in 3-dimensional vector fields are usually analyzed and classified by means of *Poincaré maps*, which is a non-linear method as opposed to linear methods such as the eigenanalysis of the Jacobian at critical points. Although usually applied to 3-dimensional vector fields, it can also be applied to 2-dimensional vector fields. In this case the Poincaré map is 1D instead of 2D. Another use of Poincaré maps is to include them into 3D visualizations for a better understanding of the flow near the periodic orbit [84, 85].

A Poincaré map  $f$  is obtained by intersecting the periodic orbit with a plane, located anywhere on the periodic orbit. More precisely, the plane has to be oriented such that the vector field is nowhere tangent to it, should be disk-shaped with its center  $\mathbf{x}_0$  on the periodic orbit, and with radius  $d$  small enough not to intersect the periodic orbit at other locations, see Figure 2.11. Such a disk is also called a *local section* or *Poincaré section*. The map is sampled on that disk by starting streamlines on it and detecting the location where they intersect the disk after one revolution, this is the reason why Poincaré maps are also called *first recurrence maps* or *first return maps* of the periodic orbit. The position of the resulting intersection point  $\mathbf{y}$  is stored in the map at the position  $\mathbf{x}$  where the streamline was started, defining a mapping from the starting point to the intersection point on the disk  $D$  of radius  $d$ :

$$f : \mathbf{x} \mapsto \mathbf{y}.$$

Because some streamlines started on the disk may diverge too far to intersect the disk after one revolution, and because of the assumed uniqueness of streamlines (Lipschitz continuity of the vector field), a radius  $d_0 < d$  can always be determined such that all streamlines started inside disk  $D_0$  with radius  $d_0$  intersect the disk at first recurrence inside radius  $d$ :

$$f : D_0 \mapsto D.$$

Poincaré maps can be analyzed by methods similar to those used for the analysis of critical points because they exhibit a fixed point at  $\mathbf{x}_0$ : they can be linearized around the point  $\mathbf{x}_0$ :

$$\mathbf{x}_0 + \mathbf{x} \mapsto \mathbf{x}_0 + (\nabla_{\mathbf{x}_0} f)\mathbf{x} + \mathcal{O}(\mathbf{x}^2),$$

$\nabla_{\mathbf{x}_0} f$  being the Jacobian of the map at position  $\mathbf{x}_0$ . One interesting fact is that the analysis by means of Poincaré maps is independent of the selection of  $\mathbf{x}_0$  on the periodic orbit, their orientation, and even the coordinate system of the disks as

well as their radii  $d$  and  $d_0$ , meaning that all these Poincaré maps are topologically equivalent, or in other words, the eigenvalues of the resulting  $\nabla_{\mathbf{x}_0} f$  are identical.

A periodic orbit is *hyperbolic* if all eigenvalues, the so-called Floquet multipliers, lie off the complex unit circle. As in the case of hyperbolic critical points, this means that small perturbations near hyperbolic periodic orbits will lead to topologically equivalent vector fields. And again conversely, any periodic orbit that is locally structurally stable must be hyperbolic. For 3-dimensional vector fields, hyperbolic periodic orbits can be classified according to Table 2.3.

two real eigenvalues	type
$ \lambda_1  > 1$ and $ \lambda_2  > 1$	<i>source periodic orbit</i>
$ \lambda_1  < 1$ and $ \lambda_2  < 1$	<i>sink periodic orbit</i>
$\lambda_1 > 1$ and $0 < \lambda_2 < 1$	<i>saddle periodic orbit</i>
$\lambda_1 < -1$ and $-1 < \lambda_2 < 0$	<i>twisted saddle periodic orbit</i>
two conjugate complex eigenvalues	type
$\lambda_1$ and $\lambda_2$ outside the unit circle	<i>spiral source periodic orbit</i>
$\lambda_1$ and $\lambda_2$ inside the unit circle	<i>spiral sink periodic orbit</i>

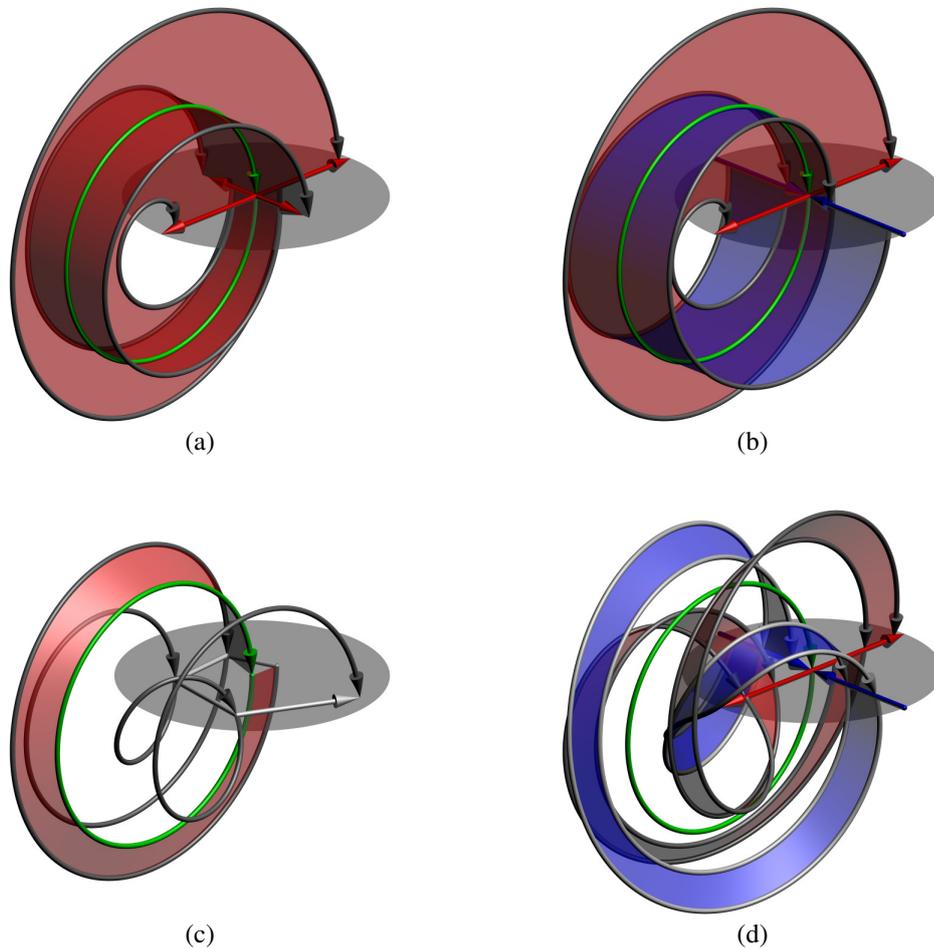
**Table 2.3:** Classification of hyperbolic periodic orbits in 3-dimensional vector fields according to the eigenvalues  $\lambda_1$  and  $\lambda_2$  (numbering is of no significance) of the gradient of the Poincaré map on the periodic orbit. See also Figure 2.11.

The reader is referred to Section 5.1 for further details and to the notes of Asimov [3] for the classification of periodic orbits in 2-dimensional vector fields and many other topics in vector field topology. Furthermore, Chapter 6 addresses vortex ring phenomena that include periodic orbits, and their analysis by means of Poincaré maps.

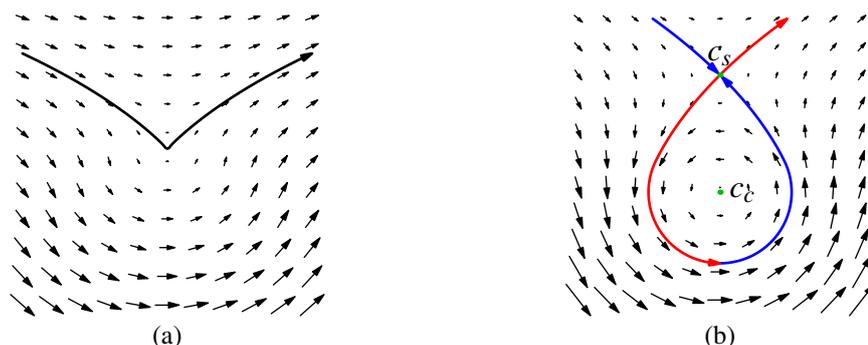
Those streamlines that are neither critical points nor periodic orbits are called *regular streamlines*. Regular streamlines can converge toward stationary points or periodic orbits, but they cannot contain them due to their uniqueness. From a topological view, every point on a regular streamline is surrounded by a neighborhood filled up by streamlines the same way parallel lines do.

### 2.5.3 Separatrices and Manifolds

The additional descriptions in Table 2.1 describe the orientation of the vector field along the corresponding eigenvector directions. For example, (1 in, 1 out) means that the streamlines of the saddle-type flow are converging toward the critical point along one eigenvector from both sides, and diverging from the critical point along the other eigenvector to both sides. In contrast, for the example of (2 in), streamlines converge to the critical point not only along both eigenvectors, but all streamlines in the vicinity of the critical point are converging to it. Consequently, saddle-



**Figure 2.11:** Poincaré map (gray disk) of periodic orbit (green). (a) Source periodic orbit. (b) Saddle periodic orbit: one stable (blue) and one unstable (red) 2D manifold. (c) Spiral source periodic orbit. (d) Twisted saddle periodic orbit with 180 degree twist: two Möbius strips representing one stable (blue) and one unstable (red) 2D manifold. As in (b), the stable and unstable manifolds intersect along the periodic orbit. Remaining cases can be obtained by reversing the vector field. See a digital version of the document for the 3D interactive models.



**Figure 2.12:** Fold bifurcation. (a) Before bifurcation, no critical points. (b) After bifurcation, a saddle ( $c_s$ ) and a center ( $c_c$ ) with corresponding stable (blue) and unstable (red) manifolds. Please also see Figure 2.8.

type critical points play an important role because they are, in contrast, related to distinguished streamlines.

In 2-dimensional vector fields, streamlines that converge to a saddle-type critical point in positive time (or in negative time meaning that they diverge from it) are called *separatrices* of the vector field because they separate regions of qualitatively different behavior of the vector field (e.g., Figure 2.12 (b)). The unity of all critical points, and separatrices is called *topological skeleton*. However, unless the vector field is irrotational, there may also exist periodic orbits that behave like sources or sinks. If this is the case, the topological skeleton computed this way is incomplete. Only if the set of (isolated) periodic orbits is added to the skeleton, the full segmentation is obtained. For irrotational vector fields, the topological skeleton can alternatively be obtained by extraction of all *watersheds* of the corresponding potential field.

In 2-dimensional and 3-dimensional vector fields, the set of all streamlines that converge to a given critical point or periodic orbit of any type in positive or negative time is called an *invariant manifold*. *Stable manifolds* consist of the streamlines that converge to a given critical point or periodic orbit in positive time and *unstable manifold* consist of those converging in negative time. In other words, the stable manifold of a point  $P$  is the set  $W^s(P)$  of points from where a streamline converges to  $P$  with time  $t \rightarrow \infty$ . The unstable manifold  $W^u(P)$  is defined likewise but with  $t \rightarrow -\infty$ .

Taking (2 in, 1 out) as an example from Table 2.2 for 3-dimensional vector fields, “2 in” means that the flow is approaching the saddle-type critical point along two eigenvectors, again, as in the case of a 2D sink, a whole manifold of streamlines is converging to the critical point in this direction, not only the streamlines along the two eigenvectors. More precisely, this manifold of streamlines is planar in the vicinity of the critical point and there it is spanned by the two corresponding eigenvectors, or their real and imaginary parts if they are complex. This is the reason, why it is called a 2D stable manifold of the critical point. The “1 out” in this example means that streamlines converge to the critical point in negative

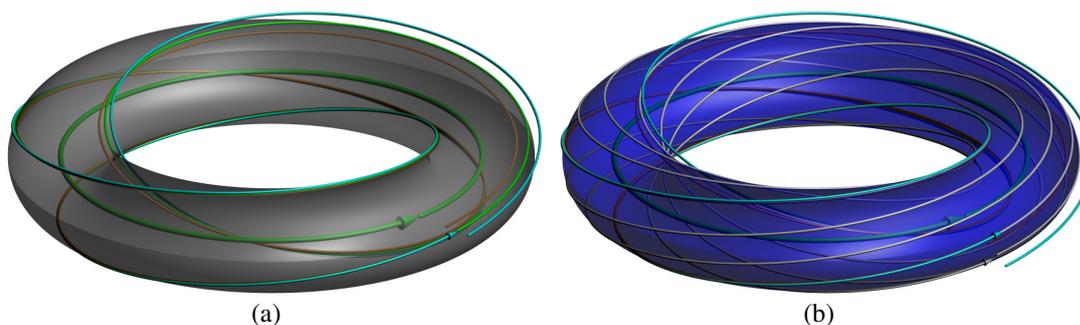
time only along the corresponding eigenvector, as in the case of the 2D saddle, representing a 1D unstable manifold. So all in all, even 3D saddles are related to distinguished streamlines: a 1D manifold and a 2D manifold. In the case of swirl and consequently complex eigenvalues and eigenvectors, the 2D manifold can be determined as the eigenplane of the corresponding eigenvalues.

For 3-dimensional vector fields there additionally exist manifolds of periodic orbits of type saddle (Figure 2.11 (b)) and twisted-saddle (Figure 2.11 (d)). In this case, both manifolds are 2-dimensional. Again, as in the case of 2-dimensional vector fields, the unity of all manifolds respective to critical points and periodic orbits is called the topological skeleton. However, 1D manifolds are obviously not very useful for obtaining a segmentation of a 3-dimensional domain. Nevertheless, in the case of spiral saddles, they have some relevance, as they are sometimes understood as *topological vortex core lines* [40]. The 2D manifolds theoretically provide a segmentation, but in practical flows, this has not yet proved very successful, except perhaps for irrotational vector fields. One reason for this is of course the occlusion problem which forbids to display dozens of stream surfaces. This is the motivation for the *saddle connectors* proposed by Theisel et al. [150]. These are the intersection curves of all 2D manifolds with small remaining bands of the manifolds along the intersection curves, similar to Figure 2.11 (b). This dramatically reduces occlusion but still conveys a large amount of the topological information of the vector field. The intersection curves alone are known from *dynamical systems* theory as *heteroclinic* and *homoclinic orbits*, the former connect two different critical points whereas the latter connect to the same critical point. Due to the assumed uniqueness of streamlines, saddle connectors, i.e., intersections of stream surfaces, are streamlines, or in dynamical systems terminology, *orbits*. It is also known in this field that homoclinic orbits can give rise to intricate flow, e.g., to *Shilnikov chaos*, examined in Chapter 6.

*Invariant tori* represent another case of special streamlines. These are stream surfaces seeded at special closed curves with the property that they pass the seed curve more than once and hence represent invariant manifold, see also Section 5.1.2 and Chapter 6. Regarding the streamlines that constitute the invariant torus, they may exhibit periodic behavior on the torus, meaning that they are closed, or be quasi-periodic, meaning that they never pass a point twice (Figure 2.13). Periodic orbits in divergence-free vector fields can be enclosed by nested invariant tori and can, in this case, be seen themselves as the limit case of the innermost invariant torus with radius zero.

## 2.5.4 Limit Sets and Limit Cycles

The *omega-limit set*  $\omega(\mathbf{x})$  of a streamline  $\mathbf{x}(t)$  is the set of points of the domain that the streamline converges to if  $t$  approaches infinity. For example, if a streamline is spiraling closer and closer around a periodic orbit, the periodic orbit is its omega-limit set (Figure 2.14 (a)). The *alpha-limit set*  $\alpha(\mathbf{x})$  is the counterpart: it



**Figure 2.13:** Torus around a spiral source periodic orbit (thick green tube). (a) Ratio of revolution (around small and large radius of the torus) of a periodic orbit (thin green tube) on the torus is  $\frac{2}{3}$ . An orbit (orange) diverging from the spiral source orbit is converging toward this periodic orbit and another orbit is converging from outside (cyan). The periodic orbit on the torus may be isolated (and hence a sink periodic orbit or a saddle periodic orbit), or non-isolated. (b) If the ratio of revolution is irrational, here  $\pi\frac{2}{9}$ , the orbit on the torus is not closed, i.e., the orbit is quasi-periodic. In this case the torus is a limit torus representing the omega limit set for both orbits converging from outside and inside (Section 2.5.4). As in the 2D case of Figure 2.14 combinations of omega and alpha sets are possible. Please see a digital version of the document for the interactive 3D models.

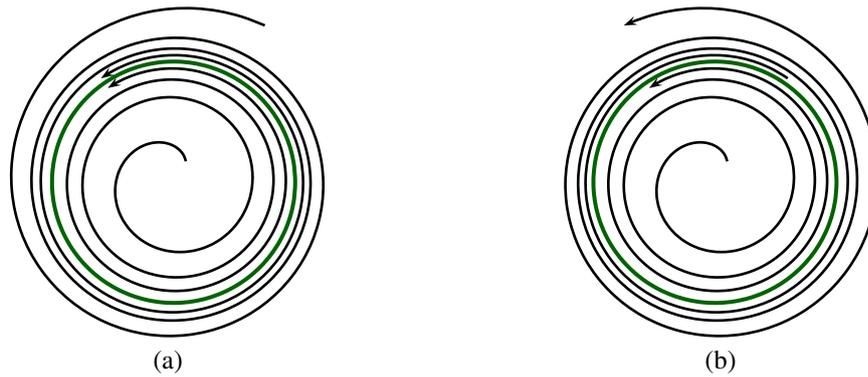
is the omega-limit set of the reversed streamline  $\mathbf{x}(-t)$  (Figure 2.14 (b)). See also Figure 2.13 (b) for the 3D example of a limit torus. If a streamline remains inside a compact region for all  $t \geq t_0$  then the following statements must hold:

- $\omega(\mathbf{x})$  is non-empty,
- $\omega(\mathbf{x})$  is closed,
- $\omega(\mathbf{x})$  is invariant by the flow (it is a union of streamlines),
- $\omega(\mathbf{x})$  is connected.

A *limit cycle* is a periodic orbit that is contained in  $\alpha(\mathbf{x})$  or  $\omega(\mathbf{x})$  of some trajectory other than the periodic orbit itself (Figure 2.14). Limit cycles are also insensitive to small perturbations. An interesting theorem concerned with limit sets is that by *Poincaré and Bendixson*. It states that if  $\omega(\mathbf{x})$  is non-empty, compact, and contains no stationary points, it must be a periodic orbit.

## 2.5.5 Applications of Vector Field Topology

The striking property of the previously mentioned direct topological methods is that they are fully automatic and to a large degree free of tuning parameters. A practical limitation is however that for many kinds of vector field data the topology



**Figure 2.14:** Limit cycle (green). (a) Representing the omega limit set of trajectories both converging from inside and outside. (b) Representing the omega limit set for trajectories converging from inside and the alpha limit set for those converging from outside in reverse time.

is far too rich to be displayed in full detail. This led to concepts such as topological simplification [79, 157, 161].

When considering the use of vector field topology for visualizing CFD data, it has to be kept in mind that topological features are often not the final result an engineer or scientist wants to see. The topological analysis can, however, be a valuable first step to be followed by other visualization techniques. One possible strategy is to use topology for segmenting a vector field into regions of similar flow. This is particularly successful in 2D, while in 3D the notion of segmentation must often be somehow relaxed to a more local property [91].

## 2.5.6 Tensor Field Topology

Tensor field topology was introduced to the visualization community by Delmarcelle et al. [24]. There, tensor field topology was visualized by means of *hyperstreamlines* (Section 2.4.5).

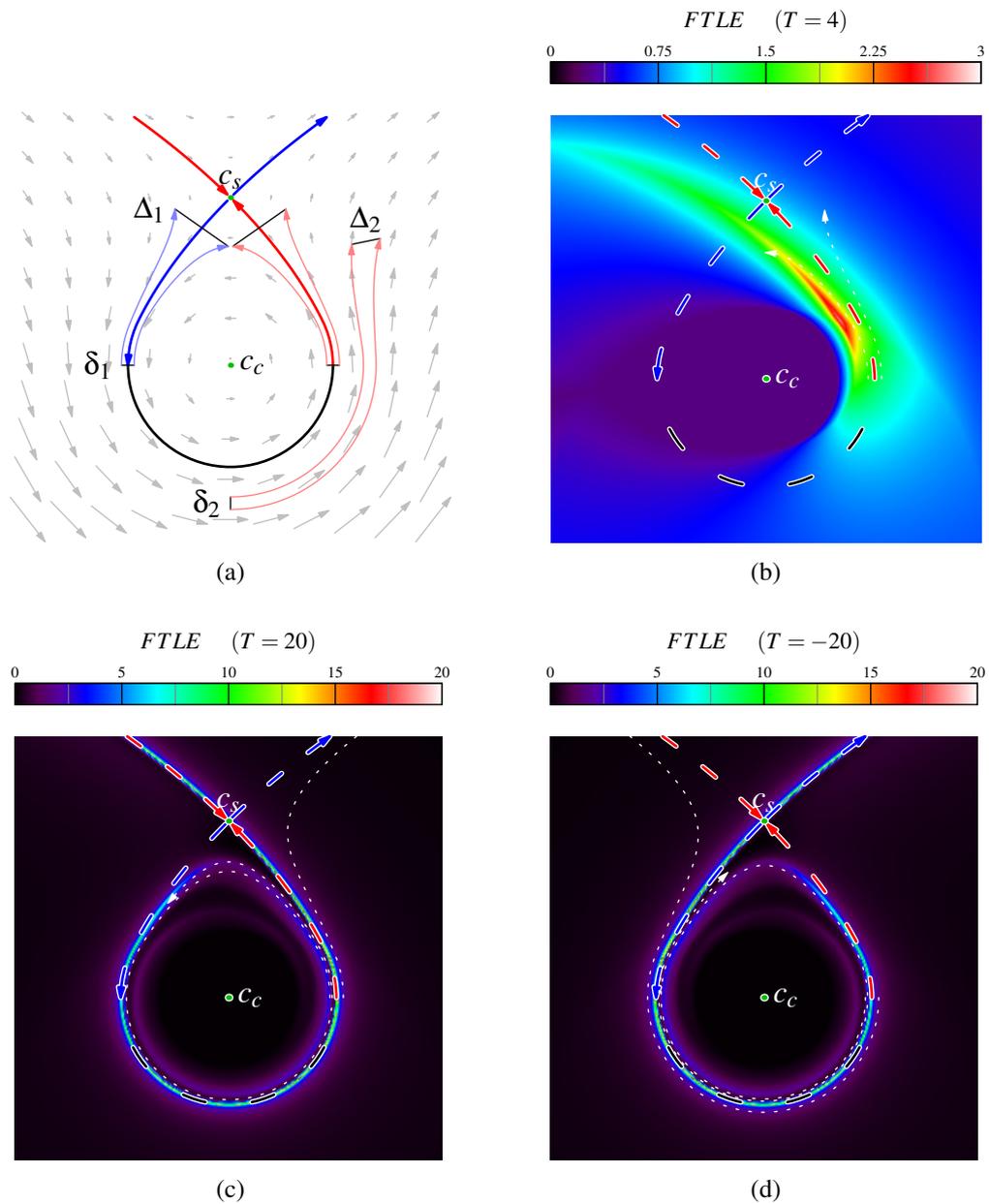
While vector field topology builds on critical points, tensor field topology builds on so-called *degenerate points*. Degenerate points are the loci where the tensor exhibits two or more identical eigenvalues. Again, this represents indefinitely many possible directions, and degenerate points can be classified using the gradient of the field, this time tensor gradients. In 2-dimensional tensor fields, there exist two types of degenerate points: *wedges* and *trisectors* (Figure 2.7). Because tensor field topology is not addressed in this thesis, the reader is referred to [23, 25], [54], [132], and [186, 187] for details regarding tensor field topology of 2-dimensional and 3-dimensional tensor fields.

## 2.5.7 Lagrangian Coherent Structures

Vector field topology has proved useful in the analysis of vector fields in many areas and many applications. However, since it builds on streamlines instead of path lines, it is only able to give an instantaneous view, i.e., it is directly interpretable only for stationary vector fields or isolated time steps. Although it may give some hint about the processes in vector fields with low time dependence [109], it is surely not satisfactory to use it for the analysis of this kind of fields. Shadden et al. [134] demonstrated with their simple 2-dimensional “double gyre” example that the separatrix can be clearly dislocated from the actual flow separation. The concept of *Lagrangian coherent structures* (LCS), on the other hand, represents an appropriate alternative to vector field topology for unsteady vector fields. Haller defined LCS as ridges in the *finite-time Lyapunov exponent* (FTLE) field [48]. The FTLE is computed from path lines and hence it is able to give an appropriate view at the topology of this kind of data with respect to true advection processes.

Chapter 8 gives a detailed introduction and description of the computation of several variants of the FTLE. Here, we would like to introduce the concept in a more informal manner. The Lyapunov exponent was introduced for quantifying the growth of perturbations under the action of time-dependent vector fields, non-autonomous differential equations, or respective dynamical systems. Whereas low values indicate predictable regions of a system or phase-space, high values indicate chaotic evolution. In a simplified manner, the FTLE can be obtained by measuring the growth factor of the distance between two points as they are advected with the flow (Figure 2.15 (a)). The FTLE can be computed from both forward and reverse advection. Under the forward-time influence of the flow, the distance  $\delta_2$  grows to  $\Delta_2$  and the FTLE can be computed from  $\frac{\Delta_2}{\delta_2}$  (Figure 2.15 (b)). The reverse case is illustrated by the growth of  $\delta_1$ . One can see that the ridges in the forward FTLE field tend to be consistent with the stable manifolds. According to Haller’s terminology, ridges in the forward-time FTLE typically represent *repelling material lines* (or surfaces) because a perturbation perpendicular to these manifolds tends to grow exponentially with time, i.e., a particle will move away from such a manifold. Paradoxically, these repelling material lines are the counterpart to stable manifolds (those that flow toward a critical point in the terminology of vector field topology). Conversely, *attracting material lines* (or surfaces) are ridges in the reverse-time FTLE and tend to be consistent with unstable manifolds. Please note that therefore in Figure 2.15 we use coloring inverse to that in Figure 2.12 (b). Increasing the advection time  $T$  for FTLE computation typically leads to growth of the respective ridges (Figure 2.15 (c) and (d)).

Besides using FTLE as a measure for predictability and as a time-dependent variant of vector field topology it has been used for other purposes so far. One example is to use it for controlling the seeding of particles, as proposed by Garth et al. [36] and used in the context of the anchor lines proposed by Bürger et al. [16].



**Figure 2.15:** Vector field topology manifolds, LCS, and FTLE computation, illustrated on vector field from Figure 2.12 (b). (a) FTLE represents growth of perturbations in forward flow ( $\delta_2 \rightarrow \Delta_2$ , (b), and (c)) or reverse flow ( $\delta_1 \rightarrow \Delta_1$  and (d)). Increasing the advection time  $T$  typically leads to growth of FTLE ridges, i.e., the LCS. Please see text for details on relation to the manifolds (red and blue bold curves).

## 2.6 Feature-Based Methods

Many constructs and properties can be denoted as *features*. In this thesis we tend to use this term for geometric representations of distinguished sets of loci. Examples include creases, the axes of vortices, as well as lines of flow separation. An introduction and overview in the field of feature visualization, extraction, and tracking can be found in the surveys by Post et al. [111–113].

### 2.6.1 Ridges, Valleys, and Watersheds

Ridges play an important role in many fields such as computer vision, image processing, and visualization. Commonly used in topography to describe an elevational crest, the term has been widened to general  $n$ -dimensional scalar data. A ridge is a “generalized local maximum”. Instead of requiring the point (or set of points) to be a local maximum in all possible directions, this is required only for selected directions. For the example of a 2-dimensional scalar field, requiring a local maximum in both coordinate directions leads to true local maxima, or points. If this property is required only in one direction, we obtain lines in general. If this direction is kept constant, one obtains *profile ridges*. However, this approach can lead to deviations from the naturally expected shape of a ridge. If, on the other hand, one can manage that this direction is always oriented perpendicularly to the desired ridge, one obtains a *height ridge* [28]. Although this sounds like a “Hen and Egg” problem, it is possible to obtain these directions from the eigenbasis of the corresponding Hessian matrix of the scalar field. Valleys can be obtained by solving for ridges of the negated scalar field (see below).

The generalization of the concept to higher dimensions is straightforward. For example, in the 3D case 1D (codimension 2) ridges and 2D (codimension 1) ridges are possible. Furst et al. [35] presented a cell-based algorithm for 2D ridges in 3D fields, inspired by the marching cubes algorithm. Alternatively, it is also possible to obtain height ridges by streamline integration in a so-called feature flow field [149]. A further alternative based on integral curves is the concept of *watersheds* and their counterpart, the *watercourses*. Watersheds are slope lines (streamlines) of the gradient field, starting at *saddle-type critical points* in that field, in this sense they are part of the field of *scalar field topology*. All in all, many different definitions for ridges have been given so far in the last centuries, but there are typically only minor differences (see also the work by Shadden et al. [134]). One of the earliest respective work in modern science was that by de Saint-Venant [8] and Breton de Champ [15].

Haralick [51], Eberly [28], and Lindeberg [81] proposed closely related definitions for  $d$ -dimensional *height ridges* in  $n$ -space. A further concept are *second derivative ridges* by Majer et al. [92]. The work by Serrat et al. [133] provides a classification of several definitions. There have been disputes on whether watersheds or height ridges are the “correct” ridges. Koenderink and van Doorn used

an example, which they called the “curved gutter” [74], as an argument against height ridges. However, as we show in [107] similar examples can be given that favor height ridges over watersheds instead. Another aspect of these methods is locality. Whereas obtaining watersheds is a global operation and perturbations have remote effects, height ridges are defined locally and hence errors have no remote influence. On the other hand, global methods tend to be superior in smoothness compared to local ones. Furthermore, typically not the entire slope line represents a watershed and there is in general no slope line that is characterized by special properties on its entire length [65] and hence only segments can be attributed to a ridge or a valley. A similar situation is met in the extraction of vortex core lines, there are approaches based on streamline integration such as the *topological vortex core lines* as opposed to local approaches (see Section 2.6.2).

Besides geomorphology [141], ridges have become popular in flow visualization, including the field of vortex core line extraction as ridges in vorticity magnitude [143] and valley lines of pressure [99], and flow separation [134]. More recently, also 2-dimensional ridges, i.e., ridge surfaces, in volumetric data were used by Kindlmann et al. [70] for visualization of diffusion tensor MRI data, by Sahner et al. [125] for visualization of vorticity and strain, and by us for the extraction of Lagrangian coherent structures [119]. Please refer to Chapter 7 for more information on height ridges and ridge extraction in general.

Except for scalar fields, the concept of ridges also exists for surfaces in space. A well known example for this type of ridges are the *maximum curvature ridges* [110], used, e.g., in the field of non-photorealistic rendering for enhancing salient features of a surface [12, 60]. Although both concepts seem to be identical at first glance, a fundamental difference is the invariance of these ridges with respect to rotation, in contrast to ridges in height fields which instead are invariant under height scaling.

## 2.6.2 Vortex Core Lines

A concept related to the formulation of ridges, in particular with respect to its parallel vectors formulation (see below), are vortex core lines. Vortex core lines can be understood as the “axes” of vortices, similar to center lines and medial axes [19]. As vortices are not straight in practical flows, vortex core lines are also curved in general. There are several approaches for the extraction of vortex core lines. They often produce results that differ in the existence, location, and shape of the core lines. The fact that until now there exists no precise definition of what a vortex is (see Section 2.1.4) also reflects in today’s handling of the different methods. It is common practice to apply the different methods and to choose the one that fits best the aim of the investigator, supported by interactive investigation, e.g., by streamlines and isosurfaces of  $\lambda_2$ .

Vortex cores are inherent properties of the velocity field and do, in general, not depend on parameter choices as would be the case for isosurfaces. The strength

of vortex core line algorithms lies in their ability to isolate nearby vortices. Their weakness is that the obtained information is often strictly local. Hence, further steps are often needed to get information such as the spatial extent of a vortex or the relative importance of nearby vortices. Importance can be defined in terms of kinetic energy, of occupied space or other criteria, depending on the aim and the application.

In 1995 Sujudi and Haimes [145] proposed a method that builds on vector field topology. They search for critical points with one real and a pair of complex-conjugate eigenvalues of their so called “reduced velocity”. The reduced velocity results from subtracting the velocity component parallel to the real eigenvector of the velocity gradient from the velocities. In the same year, Banks and Singer [7] proposed a predictor-corrector method based on vorticity and pressure. They predict the direction of the core line segment based on vorticity and correct it by forcing it to a pressure minimum on a plane perpendicular to vorticity at the predicted position, hence strongly related to valley lines of pressure. This method has been adapted by Stegmaier et al. [142] in 2005 by replacing the pressure field by the  $\lambda_2$  field. This makes the method more appropriate for many flows, such as low Reynolds number flows, and avoids the need for pressure information, a quantity often not available in given datasets. The results from this method are closely related to the approach by Sahner et al. [124], where vortex core lines are extracted as *extremum lines* of quantities such as  $\Delta$ ,  $Q$ , or  $\lambda_2$ . In this case, the valley or ridge lines are not extracted by the approach due to Eberly described in Section 7.1, but by extracting integral curves in a *feature flow field* [149]. Finally there is one more related method, the method proposed by Schafhitzel et al. [127]. There, vortex core lines are extracted from  $\lambda_2$  using skeletonization of  $\lambda_2$  isovolumes. The obtained skeletons are forced to local minima of  $\lambda_2$  on planes perpendicular to the skeleton. This way, the obtained core lines are topologically consistent with the isosurfaces of  $\lambda_2$ .

Miura and Kida [99] extract vortex core lines as valley lines of pressure in 1997 and in 1998 Strawn et al. [143] extracted them as *ridge lines* of vorticity magnitude. In the same year Roth and Peikert [116] presented a higher-order method for the extraction of vortex core lines. While the method by Sujudi and Haimes locally requires zero curvature of the core lines, the approach by Roth and Peikert relaxes the requirement to zero torsion. This condition was formulated by parallelism (or anti-parallelism) of two vectors, a concept thoroughly examined and applied to many different features including the extraction of vortex core lines, ridge and valley lines, as well as separation and attachment lines in the *parallel vectors* framework of Peikert and Roth [103] in 1999. Levy et al. [80] define a vortex core as a region where normalized helicity is close to -1 or +1. It was shown by Peikert and Roth in the same paper that from this definition of a core region a core line criterion can be derived. Theisel et al. [149] have shown that the parallel vectors approach can generally be transformed into an initial value problem inside the feature flow field. This allows to extract vortex core lines by solving for integral curves in that field. One of the benefits of this approach is the increased smooth-



**Figure 2.16:** 2D flow separation (a) and attachment (b) represent half saddles in terms of 2D vector field topology (Figure 2.9 (c)). The critical points represent separation (a) and attachment (b) points and the separatrices the respective manifolds. For this illustration, we assume slip boundaries (hatched) or wall shear stress thereon.

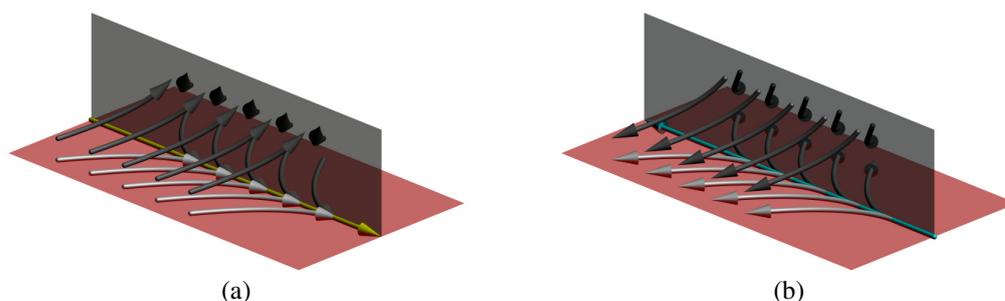
ness of the resulting curves due to interpolation during integration. A drawback of this approach, however, are remote effects, i.e., error accumulation.

In 2002 Bauer and Peikert [10] based the extraction and the tracking of vortex core lines over time on the *scale space*, exploiting temporal coherence. The scale space concept was introduced by Iijima in 1959 [59, 171] in Japan and made popular by Witkin [181] and Koenderink [73]. Another approach for tracking core lines in unsteady flow fields is the “parallel surface” approach by Theisel et al. [148], again basing on the feature flow field.

Finally, we would like to mention three more recent alternatives for vortex core line extraction. Jankun-Kelly et al. [61] presented a method building on additional scalar fields in 2006. They extract core lines again based on the approach by Banks and Singer but use scalar fields other than pressure or  $\lambda_2$  for the correction step. For the detection of the local maximum on the search plane, they use function fitting to avoid the numerical problems in the computation of derivatives. They also cluster candidate cells by k-means clustering in order to segment merging vortices. In 2007 Weinkauff et al. [174] presented a method inspired by the method of Sujudi and Haimes [145], but this time basing on path lines instead of streamlines. As in the case of feature flow fields, they approach the problem by extending the 3-dimensional vector field by treating time as an additional dimension, leading to 4-dimensional vector fields. Vortex core lines are then extracted by an extension of the “parallel vectors” approach, leading to the *coplanar vectors* operator. A very similar approach was developed by us at the same time [32], also basing on path lines and the parallel vectors operator, but leading to a different definition for the core lines. A comparison of the two approaches can be found in [130]. It has to be mentioned however, that the definition by Weinkauff et al. is Galilean invariant as opposed to ours.

### 2.6.3 Manifolds of Attachment and Separation

A feature closely related to vector field topology are *attachment lines* and *separation lines* in 3D vector fields (Figure 2.17), and *attachment points* and *separation*



**Figure 2.17:** 3D flow separation and flow attachment. (a) Separation, streamlines on the solid boundary (red) converge toward the separation line (yellow). (b) Attachment, streamlines on boundary diverge from attachment line (turquoise). See a digital version of the document for the interactive 3D models.

*points*, their counterpart in 2D vector fields (Figure 2.16). These constructs represent the loci along which flow attaches to or separates from boundary walls. A common approach for their analysis is to compute the manifolds originating at these locations, the manifolds of separation and attachment, present as separatrices in the aforementioned analogy. Indeed, Helman and Hesselink [53] already included separation and attachment manifolds in their flow topology. In the 3D case, they propose to extract the flow topology on solid boundaries first, resulting in a 2D topology analysis on the boundaries, i.e., critical points and separatrices (separation lines). Then they propose to generate manifolds from these constructs into the interior of the flow, resulting in 3D cases corresponding to the 2D case illustrated in Figure 2.16, see also Figure 2.17. We show in Chapter 9 that these manifolds can be alternatively obtained by extraction of Lagrangian coherent structures in the case of steady-state vector fields. Furthermore, since Lagrangian coherent structures appropriately depict time-dependent phenomena, this approach is also well suited for the study of time-dependent flow separation and attachment.

Separation lines are classified in this context into two categories: *open separation* and *closed separation*. Closed separation lines can be obtained by means of vector field topology, i.e., by streamline integration along the boundaries from critical points, whereas open separation does not exhibit involved critical points. While the former is amenable by means of vector field topology, the latter is not. However, in 1999 Kenwright presented a method [68] for the extraction of *attachment* and *separation lines*. This concept is defined locally and hence able to extract both cases. A drawback of this technique is, however, that its solutions tend to deviate if the feature lines are curved [115]. Flow separation is an undesirable process in many areas, notably in aviation in the context of flow around an airfoil, known as stall. It is also often a major cause for vortex generation when boundary shear flow separates from the wall and develops into a vortex (Chapter 4).

**Part I**

**Flow Physics**



## Chapter 3

---

# Instantaneous Vorticity

The first part of this thesis starts with an investigation on vortical flow, in particular, its vorticity. We title this chapter instantaneous vorticity to contrast it from the next chapter about vortex dynamics, where the focus is not on the instantaneous structure of vorticity but on phenomena involving time, such as its advection and diffusion. As detailed in Section 2.1.5, vorticity is a valuable quantity for modeling the creation of vortices and their interrelation with shear flow, but fails to disambiguate between the two. However, instead of seeing this as a drawback, this chapter focuses on the overall organization of the vorticity field with the aim of revealing interrelations between vortices and also between vortices and shear flow, in particular, boundary shear flow where vorticity typically originates.

This chapter's underlying industrial case is a design optimization for a Pelton water turbine. An important industrial objective is to improve the quality of the water jets driving the runner. In particular, jet quality is dominantly affected by vortices originating upstream in the distributor ring. For a better understanding of this interrelation, it is crucial to concentrate on the mechanisms of their creation.

We start with two simple tools as an introduction to the topic. The first, described in Section 3.1, allows for the analysis of the flow field in vortical regions whereas the second, detailed in Section 3.2, represents a method for obtaining a special kind of *isosurfaces* tailored at the extraction of vorticity magnitude in ducted flow. These tools were mainly designed for usage in the Cykloop virtual environment [169] installed at VA Tech Hydro's site, therefore we aimed at mostly automatic methods with little user input other than picking. Then, in Section 3.3, we proceed to the main contribution of this chapter, a method for obtaining visualizations of the field lines of vorticity, called *vortex lines*, similar to the common visualization of magnetic field lines, this is, field lines with line density proportional to the field magnitude. We show that this can be achieved by a modification of the seminal work by Turk and Banks [164] on *streamline placement*. For this purpose, the algorithm is changed from an image-guided 2D approach to a data-guided one in 3-dimensional unstructured grids.

## 3.1 Analysis of Vortex Cores

Vortices are important and revealing features in flow fields, especially in turbomachinery, where flow quality is often closely related to vortical motion. Therefore, a good first step to approach a new CFD dataset consists in the extraction of *vortex core lines*. The methods used for core line extraction in this chapter are those by Levy et al. [80] and by Sujudi and Haines [145]. Whereas the former builds on the orientation of vorticity, i.e., identifies sets of points where it is parallel to velocity and hence relates to helicity, the latter makes implicit use of vorticity by requiring a non-vanishing imaginary part of a velocity gradient's eigenvalue. False positives and weak vortices are suppressed as described in [103].

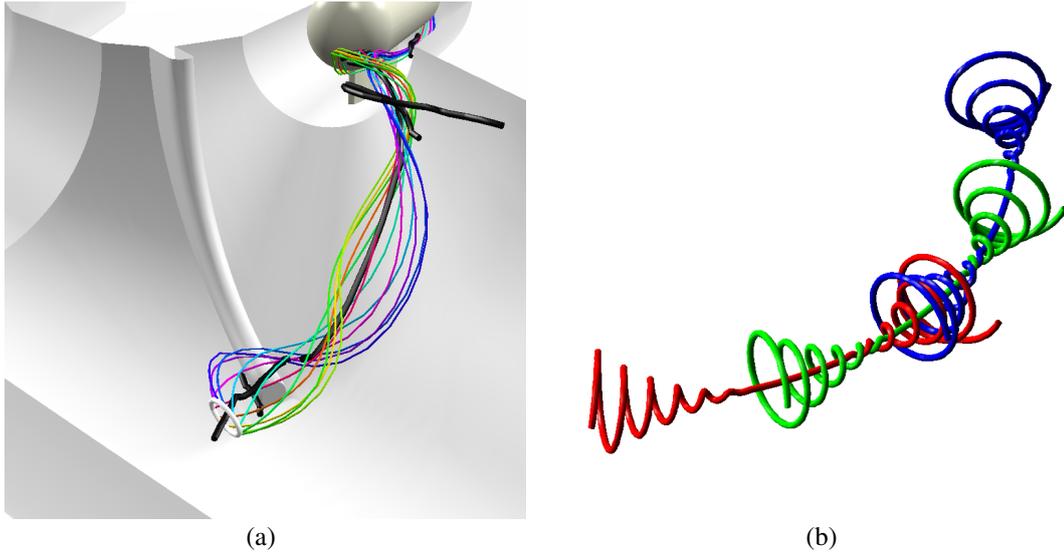
There exist automatic methods for reducing the number of obtained features based on clustering [79, 157], or on scale-space analysis [10]. For a more thorough analysis of groups of *vortex cores*, we developed a simple tool which proved to be useful. The tool allows the user to pick a core line which he or she wants to analyze. A circle of seed points is centered at the selected location such that its axis is aligned with the velocity vector. From this circle, a stream tube according to Hultquist's algorithm [56], or a set of streamlines, is generated in one or both directions along the vector field. The radius of the seed circle can be adjusted manually. Figure 3.1 (a) shows the streamline seeding tool in the study of the main separation vortex at the bifurcation to the first injector of the Pelton turbine.

This semi-automatic analysis step resembles the verification techniques in [63] and [38]. It is important to note that any such verification technique is subject to interpretation. The reason is that it cannot be expected that a *stream tube* follows the vortex core line over an extended time. Even if a streamline coincides with the core line for a while, it may later start to wind around it, and a different streamline takes over the role of best matching the core line. This situation, sketched in Figure 3.1 (b), is often met in practice. Consequently, this is a case where *topological vortex core lines* [40] may not be appropriate. Surprisingly, vortex core lines obtained by the common definitions can neither be expected to be consistent with a vorticity field line. An example is the vortex ring phenomenon subject to research in Chapter 6. It exhibits two vortex core lines, one along the main axis and one along the center of the ring, but the vorticity field exhibits helical field lines in this flow configuration.

## 3.2 Analysis of Vorticity Magnitude

The simplest model of a laminar viscous flow through a circular pipe (of radius  $R$  and length  $L$ ) is the so-called *Hagen-Poiseuille flow*, in cylindrical coordinates:

$$u = -\frac{p_0 - p_1}{4\nu\rho L}(R^2 - r^2) \quad v = w = 0. \quad (3.1)$$



**Figure 3.1:** (a) Vortex core lines (black), seeding circle (white), and streamlines (colored) at the separation vortex. Main flow direction is from right to left, part of the flow is split at the bifurcation toward the injector at the top. (b) Sketch of vortex core approximately followed by successive streamlines.

Its vorticity field has magnitude

$$\omega = \frac{p_0 - p_1}{2\nu\rho L} r \quad (3.2)$$

which is increasing linearly toward the boundary. In practical flow fields the same behavior can be observed in the boundary layer. Often the highest vorticity magnitudes can be found near no-slip boundaries rather than within vortices.

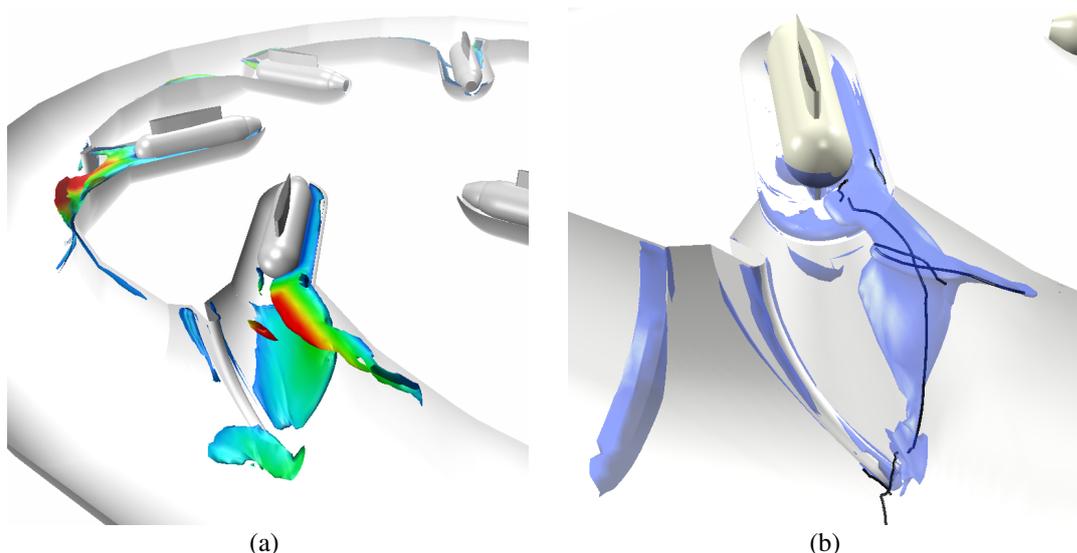
Isosurfaces of vorticity magnitude are often used for the visualization of vortices [137]. However, it is often not the absolute values of vorticity magnitude which are of interest but rather their deviation from the “expected” behavior. This explains why they often give poor results for vortex detection in ducted flows, i.e., they suffer from massive occlusion. On the other hand, isosurfacing is a convenient tool for interactive data visualization. A simple way out of this dilemma is to apply constraints to isosurfaces. Such constraints can be any inequalities for additional quantities, such as the requirement for a minimum *helicity* or a minimum (or maximum) distance from solid boundaries.

A seeming disadvantage of such a “conditional isosurface” is that it is no more a closed surface. This can however be fixed by taking pairs of such surfaces:

$$\{\mathbf{x} | f(\mathbf{x}) = f_0 \wedge g(\mathbf{x}) \geq g_0\} \cup \{\mathbf{x} | g(\mathbf{x}) = g_0 \wedge f(\mathbf{x}) \geq f_0\}. \quad (3.3)$$

Any combination of inequality signs is of course possible.

The “conditional isosurfaces” proved useful for isolating interesting regions of high vorticity. We observed that features obtained by specifying a minimum vorticity magnitude and a minimum wall distance tend to be similar to those obtained



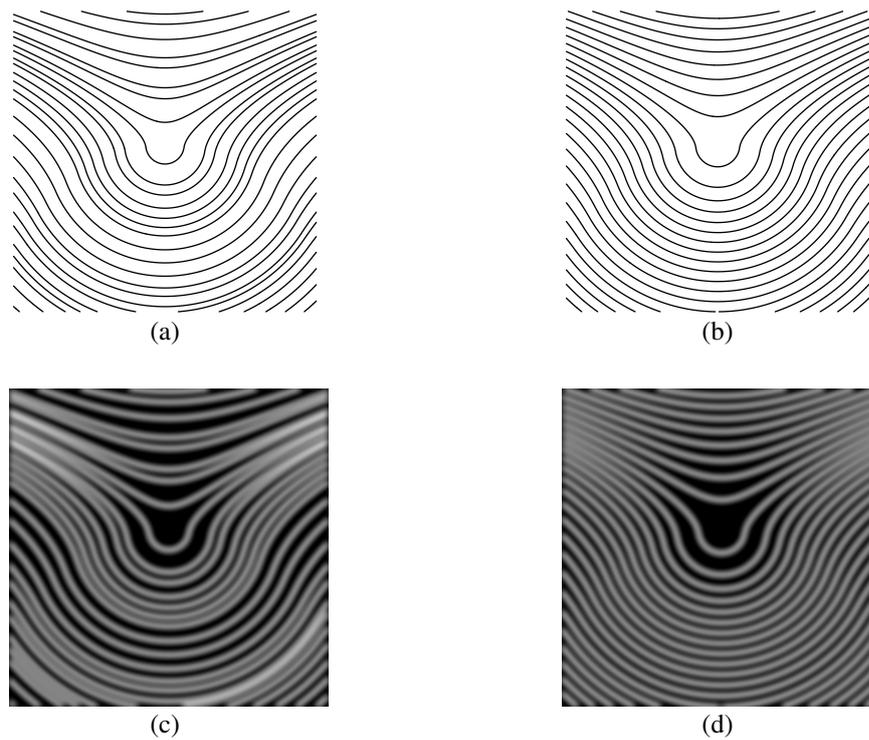
**Figure 3.2:** (a) Vorticity isosurface, wall distance  $d > 5$  mm. (blue = 0 mm, red = 50 mm). (b) Vortex core lines near first injector,  $\lambda_2$  isosurface.

by the  $\lambda_2$  vortex criterion (see Figure 3.2 (a) and Figure 3.2 (b)). However,  $\lambda_2$  tends to be too accepting, i.e., poor in separating vortices if isosurfaces at isolevel zero are used. For that reason, the level of the  $\lambda_2$  isosurfaces is commonly set to some value well below zero. However in a strict sense, this is subject to interpretation as the level motivated by the derivation of  $\lambda_2$  is zero.

For the further analysis of such regions of high vorticity, we adapted the streamline seeding tool of Section 3.1. From the point picked on a given isosurface, a planar intersection curve is computed through the isosurface. The normal of the intersection plane automatically adjusts to the average velocity direction. The seeding curve is restricted to a single connected component. This seeding tool can be used to generate stream tubes and vorticity tubes as shown in Figure 3.4 (a).

### 3.3 Vorticity Field Line Placement

The visualization of vortical flow by means of isosurfaces of vorticity magnitude already gives some insight. However, the direction of vorticity is an important property for the identification and interpretation of vortices, present in the definition of helicity and detailed in Section 2.1.5, but not conveyed by this technique. Another motivation for the method presented here is the fact that a vortex typically has no sharp boundary, i.e., identifying the extent of a vortex by its *core* or its *vortex hull* is not appropriate in many aspects. Therefore, we propose the visualization of the vorticity field by field lines of vorticity, i.e., its *vortex lines* as a complementary method. This allows for both, to account for the fuzziness of



**Figure 3.3:** 2D streamline placement by Turk and Banks [164]. (a) Initial placement with corresponding density image (c). (b) Final placement after energy minimization with corresponding density image (d).

vortices as well as visualizing the role of vorticity in both vortices and shear flow as well as in the states in between.

Because of the analogy between vorticity fields and magnetic fields, i.e., field lines are closed, terminated by a domain boundary or of infinite length, we decided to use a corresponding visualization technique for the vorticity field. Magnetic fields are usually visualized by field lines of maximum length with line density proportional to the magnitude of the field. This familiar property of magnetic fields holds for all divergence-free vector fields and thus for vorticity. The reason is that by Gauss' theorem the total flux through a closed surface is zero for a divergence-free field. Hence for an arbitrary *vortex tube* segment along the vorticity field, the fluxes through the two cross sections are equal which implies that average magnitudes are inversely proportional to projected surface and therefore proportional to the field line density.

From an algorithmic point of view, a set of field lines with this density property can be achieved by using a modification of the streamline placement algorithm of Turk and Banks [164] (see Figures 2.8 and 3.3 for resulting vector field visualizations and Figure 2.7 for an example where the method was adapted for visualizing eigenvector field lines). To support the interpretation of the 3D flow, we color the vorticity field lines by scalar quantities as shown in Section 3.4.2.

Visualization of vorticity fields by field lines with line density proportional to vorticity magnitude can be achieved by an extension of that iterative streamline placement algorithm to 3-dimensional unstructured grids with some additional modifications. We have not chosen to use a single-pass method like [34,64,95,168] but to build on the mentioned iterative method because the requirement of a prescribed spatially varying line density is achieved progressively in this case, providing preview results and more precise results on convergence. First, we describe the original method and later on, the necessary modifications.

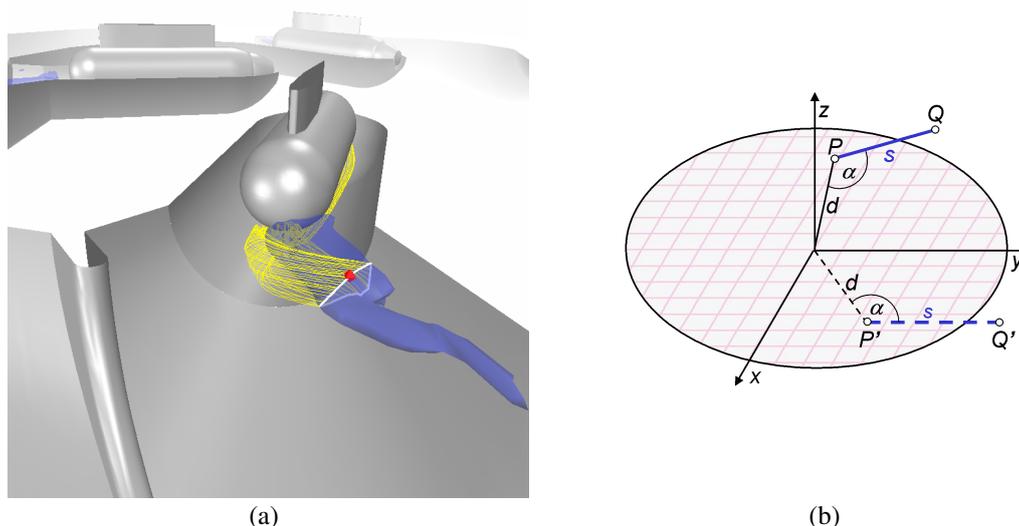
The algorithm by Turk and Banks maintains a low-pass filtered version of the streamline image for representing line density, as shown in Figure 3.3. The placement of the streamlines is steered by minimization of an energy function that is defined as the sum of the squared error between the low-pass image and the target line density. Minimization is done by random descent using a set of operations that only take effect if energy is reduced:

- Insert: generate a streamline and try to add it.
- Remove: try to remove a streamline.
- Move: try to move a streamline.
- Lengthen, shorten, combine: try to lengthen or shorten a streamline, or to combine two streamlines to a single one if their ends are sufficiently close.

The optimization process is accelerated by an oracle that tells which operation on which streamline is supposed to reduce energy most. For the extension to 3-dimensional unstructured grids, only few modifications have to be applied:

First, one has to switch from image-guided placement to data-guided placement because we want the lines to be distributed view-independently and to be guided by an additional field, in our case vorticity magnitude. In the 2D algorithm by Turk and Banks, the low-pass image is generated from the streamlines without accessing any field data, therefore the algorithm is not bound to any grid geometry or topology. We use a 3-dimensional unstructured low-pass field with geometry and topology identical to the vorticity field to avoid sampling issues and because line density has to be steered by vorticity magnitude.

The low-pass filter must also be extended to three dimensions. In the 2D algorithm, evaluation of the radially symmetric filter contribution is simplified by rotating the streamline segments around the filter origin to become parallel to the  $y$ -axis. Following this idea, we first rotate the 3D line segments onto the  $xy$ -plane (Figure 3.4 (b)) and give that as input to the filter look-up of the original 2D algorithm. This way, a radially symmetric 3D filter is applied implicitly. Due to the unstructured grid geometry, we decided not to adapt the energy function as in the work by Mao et al. [94] but to use this filter in physical space. As a consequence, an efficient point location algorithm is required to find the grid nodes that a line segment can contribute to when it is low-pass filtered.



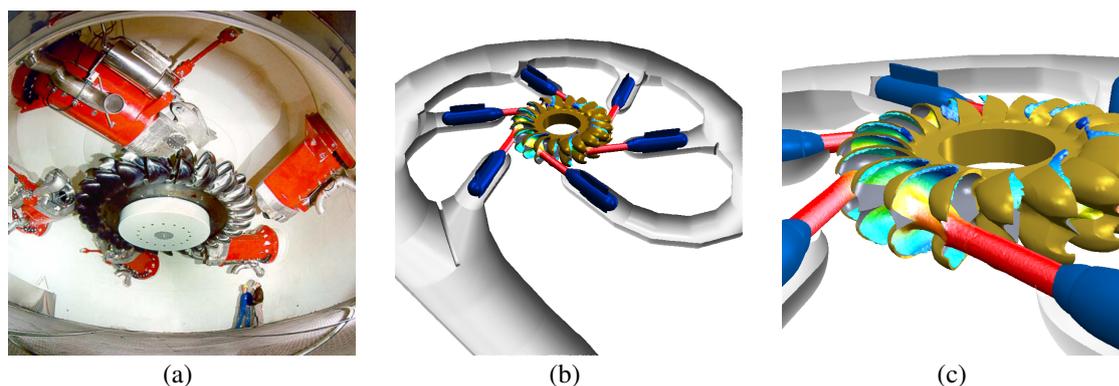
**Figure 3.4:** (a) Isosurface of vorticity, clipped at 20 mm distance from solid boundaries. Picked point (red), planar curve (white), and streamlines. (b) Filter look-up for the 3D line segment  $PQ$  by rotating the triangle  $OPQ$  onto the  $xy$ -plane.

The above two modifications yield a 3D version of the streamline placement algorithm. If the interest is however to visualize divergence-free fields, the vector magnitude can be used as target density for the placement of the field lines. Possible applications include vorticity fields, velocity fields of incompressible flow, and magnetic fields. It has the advantage that field lines of maximal length can be used, resulting in a more consistent visualization and obviating the need for lengthening, shortening, and combine operations. Another advantage is the reduction of occlusion, since the user will usually scale the target density field so that there will be almost no lines in regions of little interest.

For the control of line density by another field, the following modification has been chosen. The algorithm by Turk and Banks is mainly designed for streamline placement with constant density. It uses a target gray level, constant over the image, that has to be approximated by the low-pass image. Line density can be controlled by variation of the filter radius using an image that defines a filter radius for every point. Because we want to approximate line density given by a field, we have chosen to use a constant filter radius and to use the given field as target low-pass level instead. Overall line density can be adjusted by scaling of the density field.

## 3.4 Results

Now we exemplify the presented tools in the context of a design study of a Pelton water turbine. The underlying simulations were carried out by our industry partner

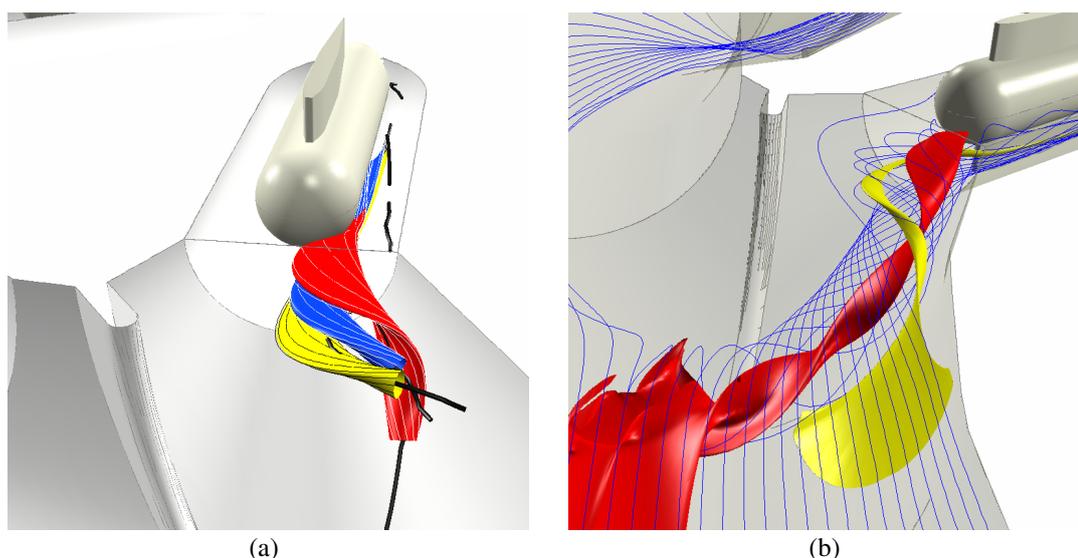


**Figure 3.5:** (a) A photograph of a Pelton turbine with five injectors. (b) Pelton turbine simulation: distributing manifold, 6 injectors with torpedoes (blue), water jets, and the runner with 20 buckets. (c) Detail of jet colored by kinetic energy.

VA Tech Hydro using the CFX-TASCflow solver. The flow inside a Pelton turbine can be partitioned into three main parts (Figure 3.5), the distributing manifold, also called distributor ring, which guides the flow toward the runner and generates the jets by the so-called injectors, the jets themselves, and the flow inside the buckets of the runner. Whereas the flow inside the manifold is modeled as steady-state, time-dependent two-phase simulations represent the jets and the flow inside the buckets. All these simulations were conducted for several different operating points. For the manifold and for the runner, symmetry planes were assumed and used as boundary conditions. In the following, we focus on the vortices occurring at the bifurcations of the manifold in front of the injectors, which are being studied with highest priority due to their role in vortex generation.

As known from experience [30], the characteristics of the water jets have a direct impact on the efficiency of the hydraulic turbine. Indeed, the jet does not necessarily follow the injector's geometrical axis, can show a non-cylindrical shape and has a dispersed mixed air / water zone on its boundary. These hydrodynamic patterns are mostly consequences of the flow characteristics in the injector, which itself results from the main pipe flow after the bifurcation where the flow is deviated toward the injector. This flow deviation, together with the bend in the main pipe, induces complex vortex structures in the injectors.

The characteristics of these vortices in size, intensity, and spatial configuration appear to be key control parameters of the jet "quality". Consequently, a strong interest lies in the identification and detection of these vortices. Their energetic intensity classification is also an important factor as it helps in identifying the "key" vortices with respect to the jet quality, and thus the related design optimization approaches that could be investigated to minimize or even eliminate their negative effects.

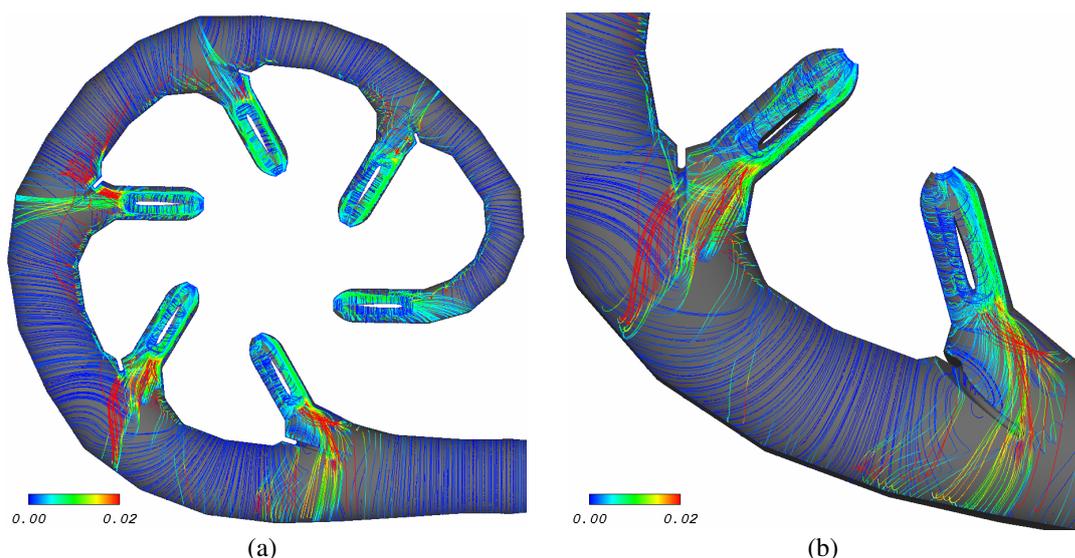


**Figure 3.6:** (a) Local “verification” of three vortex cores by stream tubes. (b) Forward and backward stream tubes reveal one separation vortex and one vortex sheet roll-up.

### 3.4.1 Vortex Analysis

In the performed study, it turns out that the strongest vortices present in the distributing manifold are separation vortices at the bifurcations. Although there is one such vortex per bifurcation, they differ significantly in shape and especially in their interaction with other vortical structures.

At the first bifurcation, the vortex core line extraction yields a set of three lines running almost parallel at some point (Figure 3.2 (b)). Possible interpretations include (a) separate vortices, (b) one main vortex and other vortices either joining or leaving it, and (c) one vortex with noisy core region. Interpretation (c) is often the correct one, as can be verified by a scale-space analysis. Vorticity magnitude, when combined with minimal wall distance, is only able to indicate the interesting region, but fails to isolate vortices (Figure 3.2 (a)). Isosurfaces of  $\lambda_2$ , giving a sharper picture, do not indicate separate vortices, regardless of the chosen level (Figure 3.2 (b)). The hypothesis of separate vortices can be verified by integrating stream tubes seeded on a small circle around each core line (Figure 3.6 (a)). The streamlines of each tube clearly wind around the core line. Such an analysis is however too localized and does not account for the radial extent of vortices. By backward integrating from a seed circle of an appropriate experimentally found radius, we obtain a very thin stream tube hitting the wall and spreading to a wide opening angle of streamlines running along the wall (Figure 3.6 (b)). The most plausible interpretation is therefore that of two separate vortices, namely the expected strong *separation vortex* plus a minor vortex apparently generated by a *vortex sheet rolling up* from boundary shear flow.



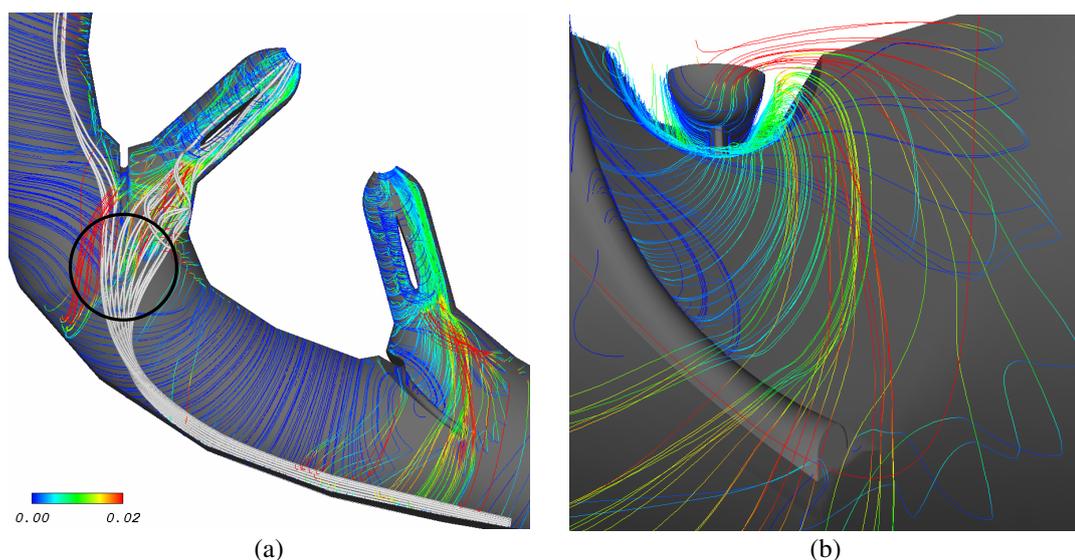
**Figure 3.7:** (a) Top view of vorticity field lines colored with distance to boundary (in meters). (b) Closer look at the first two injectors.

### 3.4.2 Vorticity Field Lines

Although the methods used in the previous section already conveyed important mechanisms of the flow inside the distributing manifold, they nevertheless concentrated on individual vortices, i.e., they did neither reveal the interaction of vortices nor their interrelation with boundary shear flow. In this sense, we will now have a complementary look on these mechanisms, i.e., in terms of field lines of the vorticity field. Hereby, the field lines are placed using the method from Section 3.3. For better visualization and to support their interpretation, the field lines are colored with scalar quantities, for example:

- For the visualization of vortices, the field lines are colored with vortex indicators like helicity,  $\lambda_2$ , or pressure.
- For the discrimination between field lines of the boundary shear flow and field lines of the inner flow, the lines are colored with Euclidean distance to the boundary. This also shows where field lines detach from the boundary.
- For studies of the vorticity field and its role in vortex phenomena, the field lines are colored with vorticity magnitude.

All images of this section show vorticity field lines with line density proportional to vorticity magnitude. As already mentioned, simulation was done only for the lower symmetry half of the turbine. Accordingly, boundaries and field lines are visualized only inside this region. The boundary is rendered in gray and the torpedoes inside the injectors were omitted for better visibility.

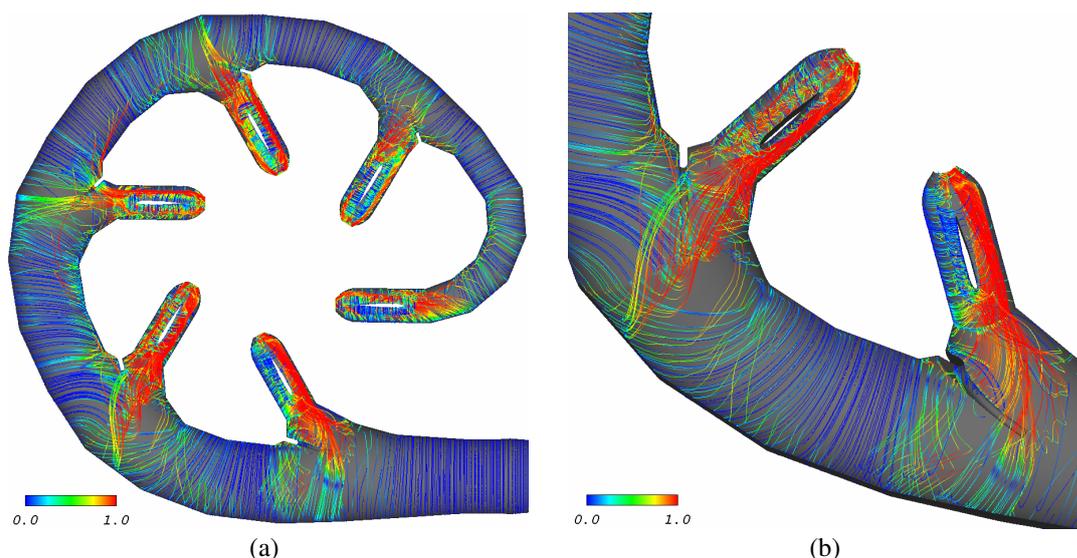


**Figure 3.8:** (a) Closer look at the first two injectors, field lines again colored with distance to boundary. In the circled region, field lines detach roughly in direction of the streamlines (white tubes). (b) Close look at the first injector. Some field lines follow the flow inside the injector while others connect from the boundary of the manifold to the boundary of the torpedo.

In Figures 3.7 (a) to 3.8 (b), the field lines are colored with distance to the boundary to support the distinction between boundary flow and inner flow. Figure 3.7 (a) provides a quick impression of the flow regarding vorticity and hence vortical flow: field lines near the boundary are colored blue whereas the usually more important field lines reaching the inner flow get different colors. Figure 3.7 (b) shows different flow features:

- In accordance with the analysis in Section 3.4.1, we identify the large separation vortex and the smaller vortex caused by vortex sheet roll-up, passing over the separation vortex. Both exhibit bundles of red field lines, meaning that these have detached far away from the wall.
- At the second injector, vorticity lines smoothly detach from the boundary of the manifold ring and join into the right part of the injector. In front of the right part of the injector, we also identify a back-flow region indicated by vorticity lines reaching the symmetry plane (refer to Figure 3.8 (a) for a verification using streamlines). Figure 3.8 (a) also shows that the field lines detach roughly in direction of velocity (toward the injector). This indicates high helicity and hence a vortical region. Such a configuration between velocity and vorticity is likely to exhibit the vortex stretching phenomenon which is subject to research in the next chapter.

Figure 3.8 (b) shows a close view at the first injector. We see that the field lines of the separation vortex follow the flow inside the injector, whereas the field

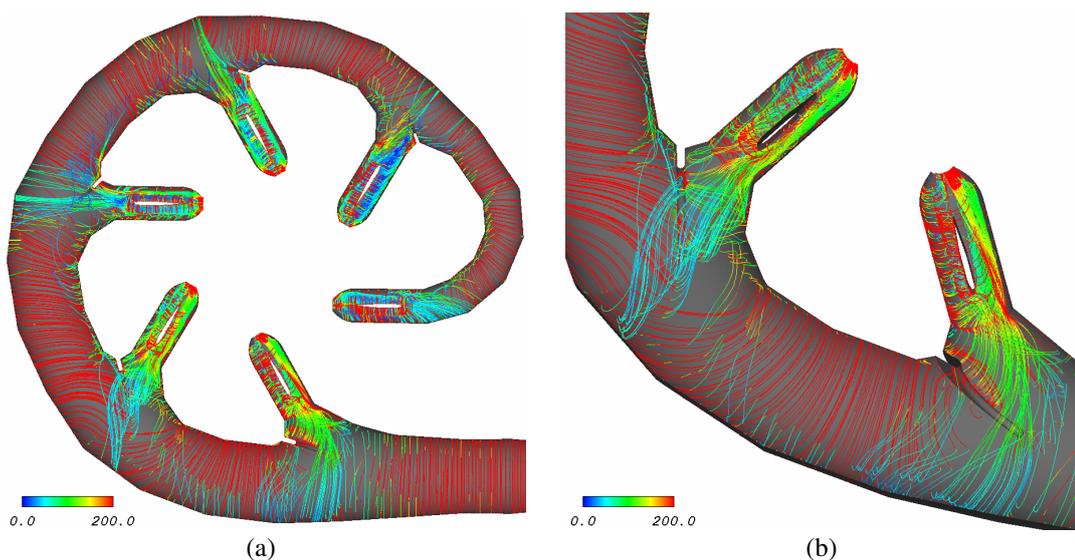


**Figure 3.9:** (a) Top view of vorticity field lines colored with absolute value of normalized helicity. (b) Closer look at the first two injectors.

lines of the vortex sheet roll-up connect to the boundary layer of the distributing manifold on one side and to the boundary layer of the torpedo on the other. A possible interpretation is that this vortex gets its rotational energy from boundary shear flow on both of its ends.

Figures 3.9 (a) and (b) show the same views, but this time the field lines are colored with the modulus of normalized helicity to help the distinction between vortices (mainly red) and shear flow (mainly blue). This again allows to identify vortices together with the related vortex lines of the boundary shear flow. As a last example, Figures 3.10 (a) and (b) show once more the same views, but this time the field lines are colored with vorticity magnitude. This type of visualization gives a more quantitative view of the vorticity field. It reflects the circumstance that although vorticity magnitude is typically highest inside the boundary layer, regions that deviate from the “regular” behavior are often of higher interest.

Table 3.1 contains some statistics of the line placement procedure. It has to be mentioned that, compared to the original algorithm by Turk and Banks, it took more time to adjust the parameters (scaling of the vorticity magnitude field, filter radius, random move radius) to obtain pleasant results. Because many contributions of a given field line contribute to the same node of the low-pass field (due to several field line integration steps per grid cell and due to filter kernel size), multiple contributions to the same low-pass node are combined first. For our dataset and algorithm settings, this resulted in a reduction of memory usage by one order of magnitude. The table shows counts of the already combined contributions.



**Figure 3.10:** (a) Top view of vorticity field lines colored with vorticity magnitude. (b) Closer look at the first two injectors.

## 3.5 Discussion

In this chapter we first presented two visualization tools for the explorative analysis of velocity and vorticity fields. Based on extracted vortex core lines, the user can generate initially orthogonal circular stream tubes, by just a radius selection and a picking operation. Similarly, stream tubes can be generated from closed regions defined by one or two isosurfaces. This allowed for an interactive investigation of individual vortices.

For a more thorough analysis of the vorticity field, we then presented a field line placement algorithm for generating sets of field lines where line density is proportional to the field magnitude. We found that the proposed visualization method is a useful tool for the investigation of vortical flow. In our belief, it visualizes vorticity and vortices in an appropriate manner, revealing interrelations and additional details both with respect to vortices and boundary shear flow. However, this approach mainly gives an instantaneous structural view on vorticity, i.e., its transport processes are not captured adequately. This is the goal of the method presented in the next chapter.

Run	1	2	3
$N_n$ = number of grid nodes		789132	
vorticity magnitude scale	0.003	0.005	0.005
iterations	10000	10000	40000
filter radius [m]	0.02	0.02	0.02
random move radius [m]	0.005	0.005	0.005
$N_l$ = number of lines	1139	1541	2709
$N_v/N_l$ = vertices per line	217.472	220.099	194.882
$N_c/N_l$ = contributions per line	6404.75	6179.37	5668.7
initial energy	4.45526e+08	1.23757e+09	1.23757e+09
final energy	4.0472e+08	1.15074e+09	1.0707e+09
computation time [sec]	2460.43	2274.35	9673.02

**Table 3.1:** Statistics for the field line placement. Memory usage of the low-pass field is  $N_n$  and memory used by the field lines is  $3N_v + N_c$ .

## Chapter 4

---

# Vorticity Transport

The methods from the previous chapter reveal many interesting aspects, especially the interrelation of vorticity inside the boundary shear flow with the vortices in the interior flow. However, the vorticity field alone gives only an instantaneous view and does not sufficiently allow for interpretations of involved transport mechanisms. Furthermore, it is strictly appropriate only for steady flows: an interpretation of time-dependent mechanisms in unsteady flow is difficult. This is similar to the case of vector field topology, which has to be applied to isolated time steps of a transient simulation, making it difficult to come up with an appropriate time-dependent interpretation.

Therefore, we decided to do a further step and to develop a visualization method for the analysis of vorticity and its dynamics in unsteady flow. We first planned to come up with a visualization method that exploits *Helmholtz's theorems*. There are different formulations of the three theorems, we repeat the following version that is appropriate in our context:

- The strength of a *vortex tube* stays constant over time. The strength represents here the flux of vorticity through any cross-section of the tube or the circulation along any closed curve around the tube, according to Stoke's theorem.
- *Vortex lines* advect with the flow, i.e., they are frozen into the fluid. Also, vortex lines and vortex tubes are of infinite length, closed, or start/end at boundaries of the domain. This second property was exploited in Chapter 3.
- Fluid elements cannot obtain vorticity.

An important caveat with these theorems is that they hold only if viscous forces are negligible, i.e., there is no diffusion nor generation of vorticity due to viscous effects. In that context, buoyant effects, another major source of vorticity, have also to be negligible. However, since many flow problems need nowadays to be modeled using viscous fluids, these theorems are substantially violated in typical applications. We therefore base our approach on another finding by Helmholtz

that takes into account viscosity: the *vorticity equation*. Helmholtz started with the Navier-Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u}, \quad (4.1)$$

and derived from it the vorticity equation

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \mathbf{u} + \nu \nabla^2 \boldsymbol{\omega}, \quad (4.2)$$

where  $\mathbf{u}$  is the velocity and  $\boldsymbol{\omega} = \nabla \times \mathbf{u}$  is vorticity. This formulation describes the time-dependent mechanisms of vorticity transport, also sometimes called *vortex dynamics*. The first term on the right hand side of Eq. 4.2 can be decomposed into two components, one parallel to the vorticity vector, and one perpendicular. The first one is called *vortex stretching* and the second one is called *vortex tilting*. We decided to leave out vortex tilting from the analysis because it cannot change the magnitude of vorticity. This allows us to formulate vorticity transport with respect to vorticity magnitude. The scalar property of this approach allowed us to come up with a visualization method that represents the involved quantities and processes by color coding and radii of tubes [121]. Figure 4.3 shows an example.

We find that this approach allows to analyze most vortex phenomena such as the roll-up of vortex sheets from boundary shear to vortices, the involved intensification of vortices due to vortex stretching, and the loss of vortex strength due to diffusion. To our knowledge vorticity has always been visualized as a whole. But we believe that a deeper understanding of the different mechanisms involved in the dynamics of vorticity can be obtained only by visualizing the individual terms of the vorticity equation.

The methods in this chapter aim at analyzing the transport of vorticity inside incompressible flow. The analysis is based on the vorticity equation and is performed along *path lines* which are typically started in upstream direction from vortex regions to visualize the mechanisms that relate to the given vortex at the given time. Different methods for the quantitative and explorative analysis of vorticity transport are presented and applied to CFD simulations of water turbines in Section 4.3. Simulation quality as well as numerical errors involved in the analysis are accounted for by including the errors of meshing, convergence, and gradient estimation into analysis and visualization. The obtained results are discussed and interpretations with respect to engineering questions are given.

The main contribution of this chapter is a set of tools for the exploration of vorticity distribution and vorticity transport. The latter is described by the vorticity equation containing several terms that can be separated by their physical meaning. Additional terms occur if the viscosity is not uniform, e.g., because a two-equation *turbulence model* is used. For the interactive study of such multimodal data it is beneficial to have multiple *linked views* available [26], combining techniques from scientific and information visualization.

Section 4.1 gives some background on vorticity transport and the derivation of the vorticity equation in the case of non-uniform viscosity, e.g., due to two-equation turbulence models. Methods for the visual analysis of vorticity transport in incompressible flow are developed in Section 4.2. Finally, in Section 4.3 these methods are applied to industrial CFD simulations in hydro-mechanical projects and interpretations of the results are given.

## 4.1 Theory

In this section two versions of the vorticity equation are described. The first version is applicable to CFD simulations with known uniform viscosity, or in a reduced manner to simulations with unknown viscosity. The second version is applicable to simulations where a spatially varying viscosity is given and known. This is often the case because turbulence models nowadays typically lead to spatially varying (modified) viscosity even for fluids with uniform viscosity.

Scalar quantities which will be used for the analysis by the methods presented in Section 4.2 are derived for both versions of the vorticity equation.

### 4.1.1 Vorticity Equation for Uniformly Viscous Flow

The flow simulations to be visualized are based on the Navier-Stokes momentum equations for divergence-free flow with uniform density and viscosity:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u}, \quad (4.3)$$

with pressure  $p$ , density  $\rho$ , kinematic viscosity  $\nu$ , and Laplacian  $\nabla^2$ .

Taking the curl of Eq. 4.3 and applying several vector identities leads to the vorticity equation (the reader is referred to, e.g., [9] for a full derivation):

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \nabla \mathbf{u} + \nu \nabla^2 \boldsymbol{\omega}. \quad (4.4)$$

The first term on the right hand side can be decomposed into *vortex stretching* and *vortex tilting* by computing its components parallel and perpendicular to vorticity:

$$\begin{aligned} \frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} &= \underbrace{(\boldsymbol{\omega} \cdot \nabla \mathbf{u})_{\parallel \boldsymbol{\omega}}}_{\text{stretching}} + \underbrace{(\boldsymbol{\omega} \cdot \nabla \mathbf{u})_{\perp \boldsymbol{\omega}}}_{\text{tilting}} \\ &\quad + \underbrace{\nu \nabla^2 \boldsymbol{\omega}}_{\text{diffusion}}. \end{aligned} \quad (4.5)$$

The stretching term represents the rate of stretching of vorticity. Positive stretching in direction of the vorticity vector increases its magnitude because of conservation of the angular momentum of the fluid element. A well-known example for this phenomenon is a pirouetting skater contracting his arms to increase his angular velocity. The tilting term represents the rate of rotation (tilting) of the vorticity vector and hence does not change its magnitude. The diffusion term, too, can be decomposed in a component parallel to vorticity and a component orthogonal to it. Again, only the parallel component changes the magnitude of vorticity and therefore the perpendicular component is omitted from the quantitative analysis described in Section 4.2. This leads to the following equation used for analysis:

$$\begin{aligned} \left(\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega}\right)_{\parallel \boldsymbol{\omega}} &= (\boldsymbol{\omega} \cdot \nabla \mathbf{u})_{\parallel \boldsymbol{\omega}} + (\mathbf{v} \nabla^2 \boldsymbol{\omega})_{\parallel \boldsymbol{\omega}} \\ &= \sigma + \delta, \end{aligned} \quad (4.6)$$

with stretching  $\sigma$  and diffusion  $\delta$ , both parallel to vorticity. Eq. 4.6 contains third-order derivatives because vorticity already incorporates partial derivatives. Computing these quantities from CFD results typically introduces numerical error. In practice, it is usually possible to get first derivatives of velocity directly as additional output from the solver. But even then, second derivatives have to be computed numerically. Theoretically, this can be avoided by computing the diffusion term simply as the difference of the left hand side and the stretching term  $\sigma$ . However, this would require that Eq. 4.6 holds exactly for the given data. This is not the case if, e.g., a two-equation turbulence model has been used to compute them. Turbulence models introduce modifications to Eq. 4.3 which obviously lead to modifications in Eq. 4.6. In the following section, the modifications are studied for a special class of turbulence models.

Other reasons why Eq. 4.6 often cannot be assumed to hold exactly are residual errors in the simulation and also errors in gradient estimation. From a numerical point of view often only moderate simulation accuracy is needed for answering questions in engineering. Because CFD simulations have a high computational cost, mesh refinement is often limited and the simulation is often not given the time to fully converge. Additionally, for steady simulations, it is often not possible to achieve fully-converged solutions because the problem would require to be solved using a time-dependent model.

Therefore, unless information on the residuals or on the viscosity is available, only two of the three terms of Eq. 4.6 are available for visualization: the left hand side describes the change of vorticity magnitude for a particle at a given location and time, while  $\sigma$  gives the portion of this change due to vortex stretching.

If the available information allows to compute the third term  $\delta$ , as is often the case because the modeled fluid is known and because solvers typically output viscosity information, the *discrepancy*  $\Delta$  is computed as the absolute difference between the two sides of Eq. 4.6 and is used as a measure of uncertainty. In this case vortex stretching  $\sigma$ , vorticity diffusion  $\delta$ , and discrepancy  $\Delta$  are available for visualization.

## 4.1.2 Vorticity Equation for Non-Uniformly Viscous Flow

Turbulence models were introduced in CFD to obtain good results with a realistic level of spatial discretization (refinement level of the mesh) as discussed in the context of *DNS*. Two-equation turbulence models (see, e.g., [1]) such as the  $k - \epsilon$ ,  $k - \omega$ , and the SST models [98] are based on turbulent viscosity. These models are incorporated into the Navier-Stokes equations Eq. 4.3 by a *modified pressure* and a *modified viscosity*:

$$p' = p + \frac{2}{3}\rho k, \quad \nu_e = \frac{\mu + \mu_t}{\rho},$$

where  $k$  is the turbulent kinetic energy,  $\mu$  is the dynamic viscosity, and  $\mu_t$  is the turbulent (eddy) viscosity .

The modified Navier-Stokes equations then become:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{\nabla p'}{\rho} + \nu_e \nabla^2 \mathbf{u} + (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) \cdot \nabla \nu_e. \quad (4.7)$$

The last term of Eq. 4.7 accounts for the fact that turbulent viscosity is in general not uniform, even though the medium is homogeneous. This makes our method also applicable to fluid that exhibit inhomogeneous viscosity despite of used turbulence models. Eq. 4.7 can be simplified using the *rate of strain* tensor  $\mathbf{S}$ :

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{\nabla p'}{\rho} + \nu_e \nabla^2 \mathbf{u} + 2\mathbf{S} \cdot \nabla \nu_e. \quad (4.8)$$

Taking the curl of Eq. 4.8, applying the same vector identities as for Eq. 4.5, and applying to the right hand side the additional vector identity  $\nabla \times (f\mathbf{v}) = f \nabla \times \mathbf{v} + \nabla f \times \mathbf{v}$  leads to the modified vorticity equation:

$$\begin{aligned} \frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} &= \underbrace{(\boldsymbol{\omega} \cdot \nabla \mathbf{u})}_{\text{stretching}} + \underbrace{(\boldsymbol{\omega} \cdot \nabla \mathbf{u})_{\perp \boldsymbol{\omega}}}_{\text{tilting}} \\ &+ \underbrace{\nu_e \nabla^2 \boldsymbol{\omega} + \nabla \nu_e \times \nabla^2 \mathbf{u} + \nabla \times (2\mathbf{S} \cdot \nabla \nu_e)}_{\text{diffusion}}. \end{aligned} \quad (4.9)$$

Again, Eq. 4.9 is now restricted to the direction parallel to  $\boldsymbol{\omega}$ , giving

$$\begin{aligned} \left( \frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} \right)_{\parallel \boldsymbol{\omega}} &= (\boldsymbol{\omega} \cdot \nabla \mathbf{u})_{\parallel \boldsymbol{\omega}} \\ &+ (\nu_e \nabla^2 \boldsymbol{\omega} + \nabla \nu_e \times \nabla^2 \mathbf{u} + \nabla \times (2\mathbf{S} \cdot \nabla \nu_e))_{\parallel \boldsymbol{\omega}} \\ &= \boldsymbol{\sigma} + \boldsymbol{\delta}. \end{aligned} \quad (4.10)$$

Having simulation data which solve Eq. 4.7 (up to residual errors), there are again two ways to compute the diffusion part, either by computing its sub-terms

as  $\delta = (\mathbf{v}_e \nabla^2 \boldsymbol{\omega} + \nabla \mathbf{v}_e \times \nabla^2 \mathbf{u} + \nabla \times (2\mathbf{S} \cdot \nabla \mathbf{v}_e))_{\parallel \boldsymbol{\omega}}$  or by subtracting  $\sigma$  from the left hand side as  $\delta' = (\partial \boldsymbol{\omega} / \partial t + \mathbf{u} \cdot \nabla \boldsymbol{\omega})_{\parallel \boldsymbol{\omega}} - \sigma$ . The discrepancy  $\Delta = |\delta - \delta'|$  can again be taken as a measure for the uncertainty of the obtained result. It is equal to the discrepancy between the two sides of Eq. 4.10.

Here, in contrast to Section 4.1.1 without additional information, vortex stretching  $\sigma$ , vorticity diffusion  $\delta$ , and the discrepancy  $\Delta$  are available for visualization.

## 4.2 Vorticity Transport Analysis Tools

Our analysis of vorticity transport is based on path lines because the left hand sides of Eq. 4.5 and Eq. 4.9 are convective derivatives of vorticity with respect to velocity. This way, a path line can be used to visualize the vorticity transport effects that a fluid element undergoes on its way through the flow. Visualizing differential quantities on path lines avoids the need of integration along the transport paths and hence error accumulation due to integration errors and numerical discrepancy in the data. This is an important design decision for our method since practical CFD simulations tend to suffer from residual errors. Computing path lines upstream or downstream from points of interest allows the researcher to analyze the flow behavior with respect to that region and time.

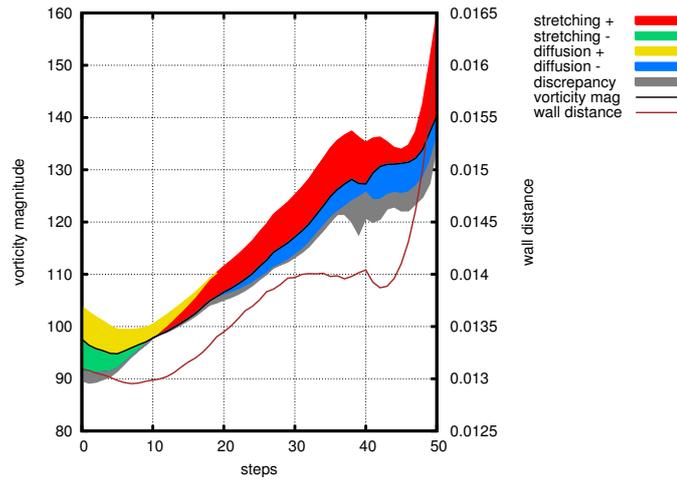
The quantities of interest are sampled along the path line and plotted as described in Section 4.2.2. Selected quantities are also visualized directly on the path line as described in Section 4.2.3 and Section 4.2.4. Vorticity magnitude is mapped to the thickness of the path lines together with parallel projection rendering. Vorticity orientation is visualized by adding vorticity streamlets to the path lines. Where appropriate, wall distance can be visualized using “support columns”. The quantities used for visualization could be computed during line integration. For the presented results, it was, however, decided to precompute them, in order to achieve short response times for the interactive analysis. All line integrations are performed with fourth-order Runge-Kutta integration.

### 4.2.1 Quantities

The quantities selected for the basic analysis described in Section 4.1.1 include vorticity magnitude, its rate of change along the path line, the stretching term  $\sigma$ , and the distance to the boundary. As already mentioned, the focus of the quantitative analysis is on the magnitude of vorticity, not its orientation and hence, vortex tilting is omitted in the visualization.

If the extended analysis is possible, the diffusion term  $\delta$  and the discrepancy  $\Delta$  are used as additional quantities, and derived from them also the relative error

$$\varepsilon = \frac{|\delta - \delta'|}{|\sigma| + |\delta|}. \quad (4.11)$$



**Figure 4.1:** Plot along a single path line, corresponding to Figure 4.2.

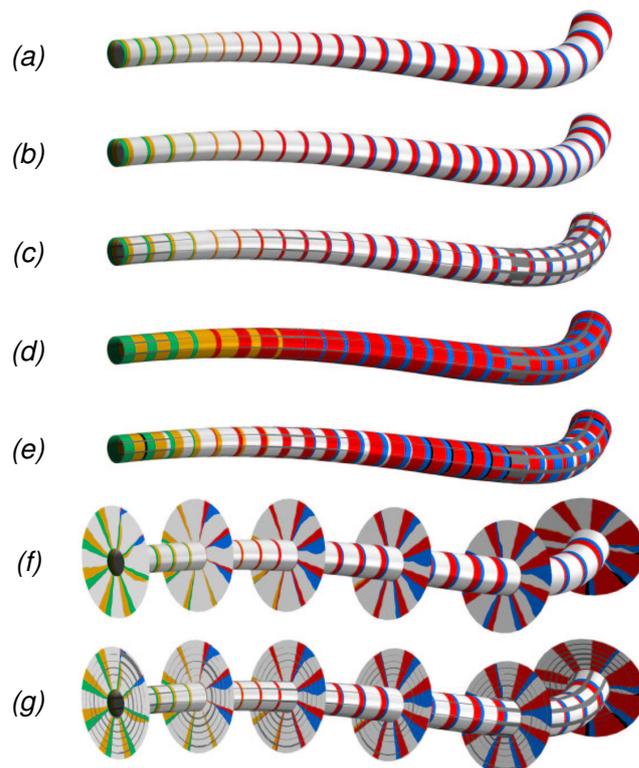
If available, the residuals as a further data channel can be included in the visualization. Also the *Courant number* giving indications on possible convergence problems of the simulation can be taken for comparative visualization.

According to Eq. 4.6,  $\sigma$  and  $\delta$  represent a decomposition of the rate of change of vorticity magnitude into the two mechanisms vortex stretching and vorticity diffusion. The visualization methods should account for this decomposition property.

## 4.2.2 Path Line Plots

As a first type of visualization, a 2D plot is generated for the path line in question. The abscissa of the plot represents either time or arc length along the path line while the quantities are mapped to the ordinate. The purpose of having this basic type of visualization available in our toolkit is to have an undistorted view of the path line and at the same time being able to see more different quantities than would be possible with a condensed 3D representation.

The quantities  $\sigma$  and  $\delta$  are plotted relative to the plot of vorticity magnitude in order to meet their decomposition property. If positive, they are plotted as area above the plot of vorticity magnitude, otherwise they are plotted as area below it. Stretching forms the inner band while diffusion forms the outer. Figure 4.1 shows an example result. The discrepancy  $\Delta$  is added as another band below the existing plot. This visualization shows the proportion of the two terms as well as their total effect on vorticity magnitude. Because  $\sigma$ ,  $\delta$ , and  $\Delta$  are of unit  $s^{-2}$  in contrast to vorticity which is of unit  $s^{-1}$ , the values of  $\sigma$ ,  $\delta$ , and  $\Delta$  can be freely scaled by a user-defined factor for obtaining significant plots. The sum of the unscaled  $\sigma$  and  $\delta$  corresponds to the slope of the vorticity magnitude plot, up to discrepancy  $\Delta$ .



**Figure 4.2:** Striped path line modes using color scheme of Figure 4.1. Evenly-timed segment lengths indicating velocity (a), evenly-spaced segment lengths (b), same as (b) but with error stripes (c), with additional normalization of the stripes (d), scaling instead of normalization (e), same as (b) but with additional striped slices (f), and additional error stripes (g). Tube radius represents vorticity magnitude.

### 4.2.3 Striped Path Lines

Striped path lines are introduced to show several of the computed quantities at their physical location along the path lines. The idea is to provide a condensed visualization on an interactively seeded set of path lines. Any of the path lines can then be selected for closer inspection by a path line plot as a linked view.

A tube is generated along each path line, with radius representing vorticity magnitude, similar to the iconic stream tube [165]. The path line is divided into segments of constant user-defined length (or time for also visualizing velocity) and the corresponding segments of the tube are used for visualization. The quantities for visualization are sampled at the midpoint of the path line segments. Three quantities are visualized by color stripes on the tube segments: stretching  $\sigma$  and diffusion of vorticity  $\delta$ , as well as the error  $\epsilon$ .

**Data Stripes:** The coloring strategy for the data stripes is chosen according to the path line plots of Section 4.2.2. A color pattern proportional to  $\sigma$  and  $\delta$  can be obtained by looking at a column of the plot area band that represents them.

The idea is to map this proportional pattern to the segment of the tube using color stripes. Transverse stripes are chosen for better readability. Since stretching comes first (is the inner band) in the plots, it is used for the first stripe and diffusion is used for the second stripe. This coding also indicates flow direction. For normalized visualization of the proportion only, the two-color pattern is scaled to fit the complete length of the segment. If also the absolute value of stretching and diffusion is to be visualized, a third stripe is used as a filler and colored white. The scaling is determined in this case from  $\max(|\sigma| + |\delta|)$  over the path line or set of path lines. Because of data peaks along the path line, this scaling can lead to unsatisfactory visualization and therefore the user is allowed to scale the stripe width. This will lead to clipping at the peaks and allows the user to inspect the region of interest. Clipping is indicated by a black stripe of predefined width. Figure 4.2 shows some examples.

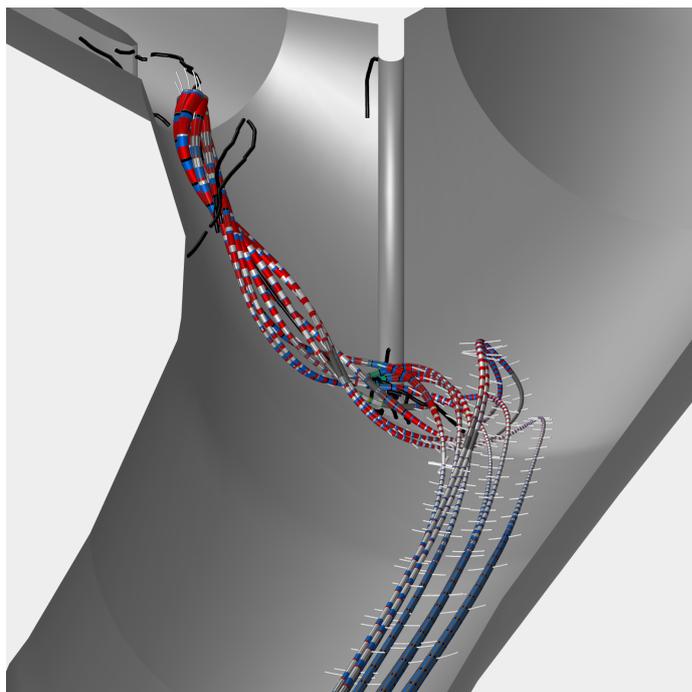
**Error Stripes:** According to the role of error, the error stripes are chosen orthogonal to the data stripes, resulting in longitudinal stripes along the tube. The width of the error stripes is proportional to  $\epsilon$  (Eq. 4.11). At full error ( $\epsilon \geq 1$ ) the complete tube segment is covered by error stripes leaving no data stripes visible. At zero error no error stripe is visible and all is covered by data stripes. More than one error stripe is generated to avoid its occlusion but the number is kept low to preserve resolution. Figure 4.2 shows some examples.

## 4.2.4 Complementary Visualization Techniques

While the two techniques of path line plots and striped path lines already convey a multitude of information, these are all local data, and combining them with some contextual visualization would allow for a more profound analysis. Since we are concerned mainly with data exploration rather than presentation, complementary visualizations can be easily added or removed during inspection of the data. We describe in the following a few techniques that turned out to play well together with the proposed basic technique.

**Slices:** Until now, the visualization was confined to the path lines. A single path line visualizes the advective aspects of vorticity well but does not allow to inspect the local neighborhood of the path line. However, if there is, e.g., gain of vorticity by diffusion on the path line, one would like to know where it diffuses from. This can be analyzed by generating other path lines nearby, but only in the case that the path lines pass the region in question at the same time.

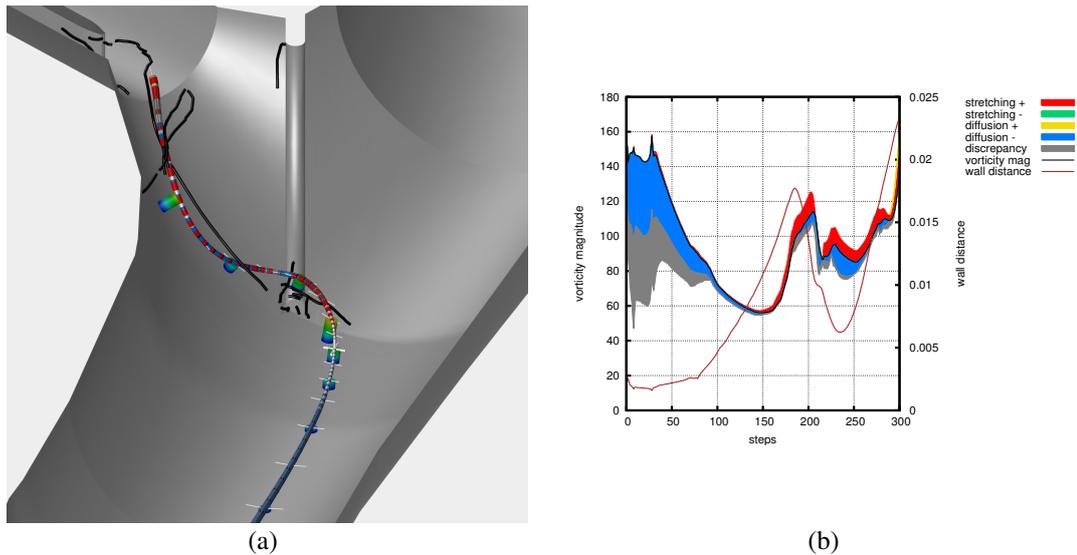
An alternative and more systematic method is to generate perpendicular slices on the path line, sample the quantities on the slices, and to apply the same color coding as for the striped path lines. All modes of the path line stripes are implemented for the slices, too. The circular slices are divided into sectors and this time the data stripes are radial and the error stripes are circumferential. This gives a continuous visualization with increasing radius and contrasts well to the transverse data stripes on the tube. Figure 4.2 shows examples.



**Figure 4.3:** Separation Vortex. Vorticity transport visualized by a set of striped path lines, backward integrated from a seed circle at the downstream end of the vortex core line (black). Visualization is augmented by vorticity streamlets (white).

**Vorticity Streamlets:** So far, only the magnitude of vorticity transport is visualized. This is sufficient for the quantitative analysis, but it does not produce a complete image, i.e., it does not convey the structure of vorticity. Therefore, streamlets of vorticity are added to the path line visualization. The length of the streamlets is defined by a user-defined time interval. This plays well together with the tube radius since the tube radius is visualizing vorticity magnitude. Vorticity streamlets disappearing inside the tube due to partial alignment with the tube indicate high normalized helicity which by Levy's criterion is associated with swirling flow and hence indicate vortices. Two streamlets are computed at each seed, one in positive and one in negative vorticity direction. Due to the analogy between vorticity and magnetism, it is natural to interpret the streamlets as a kind of compass-needles with possibly bent ends that follow the field. Furthermore, this technique also visualizes *vortex tilting*. Figure 4.3 shows an example.

**Wall Distance Indicators:** Wall distance plays an important role in the analysis of vortical motion because of the vorticity present in boundary shear flow and possibly transforming into vortices. Another reason for keeping an eye on wall distance is that the quality of simulation as well as the quality of the derivatives is often low close to the wall. Wall distance could be visualized, e.g., by color coding of the vorticity streamlets. But this would add another type of colored stripes and therefore would interfere with the visualization by striped path lines.



**Figure 4.4:** (a) Selected path line from Figure 4.3 with wall distance indicators (colored cylinders). (b) Corresponding path line plot.

Therefore “support columns” are introduced. These are tubes that indicate the distance from the path line to the wall. The tubes have a top face that is located at the path line and follow to the wall. Instead of using straight tubes visualizing the shortest distance to the wall, we generate them by streamline integration along the negative gradient of wall distance until the boundary is reached. However, without additional information, it would not be possible to tell the length of such a tube because of the projection in the resulting rendering. Therefore, the tubes are continuously colored by distance to the boundary. This allows the user to see the value of wall distance. At the same time, long tubes represent a color legend when seen from the side. If the tube radius is large enough, the top faces of the tubes are well visible even when looked from the top, even together with the path line tube. In this case the color of the top face visualizes wall distance. Figure 4.4 (a) shows an example.

**Velocity-Vorticity Line Rakes:** So far, the described vorticity streamlets are usually chosen shorter than the path line along which they are seeded, because they only add supporting information. However, there are applications where opposite roles are fruitful. This leads to a special case of *rakes*, where a single integral curve is generated interactively and other integral curves are started from that curve in evenly-spaced or evenly-timed distribution. Doing this for the same field would not deliver additional information. But velocity and vorticity are particularly suitable, e.g., because they are sufficiently close to perpendicular in shear flow. The rake with a velocity streamline as primary line and vortex lines starting from it (denoted here as  $(u, \omega)$  rakes) can be used for the visualization of boundary shear flow. The rake with a vortex line as primary line and integral curves of velocity starting from it (denoted here as  $(\omega, u)$  rakes) can be used for visualizing sepa-

ration of boundary shear flow. Figure 4.9 (a) and Figure 4.9 (b) show examples where both types of rakes are used. The described type of rakes has some advantages compared to conventional rakes: the user is freed from the tedious process of rake orientation, a single selection operation already defines the rake. Another advantage is that it can generate curved rakes, e.g., following the boundary, also by a single selection operation. Visualizations by  $(u, \omega)$  rakes represent an alternative to vortex line placement investigated in Chapter 3. They too are only able to visualize the instantaneous structure of vorticity. Extending the concept of  $(u, \omega)$  rakes to time-dependent integral curves of velocity does not seem to be useful due to interpretation difficulties. The extension of  $(\omega, u)$  rakes to time-dependent integral curves of velocity, i.e., path lines, makes sense and can be used as shown in Section 4.3.4 and Figure 4.9 (b).

## 4.2.5 Methodology

For the analysis of vortical regions in terms of vortex dynamics, the striped path lines are typically seeded interactively at significant vortex core lines, on tubes around the core lines, or on isosurfaces of vorticity magnitude or  $\lambda_2$  (at time of interest in the transient case) and integrated in upstream direction. This allows for the analysis of the mechanisms that are involved in the development of the given vortex. In our case, vortex core lines are either computed according to Levy et al. [80] or Sujudi and Haines [145]. Another possibility is to seed the path lines at regions on the boundary where vorticity is advected away and to compute the path lines in downstream direction. This conveys how and where vorticity is transported.

Although the path lines are typically computed in upstream direction to analyze involved mechanisms, it has to be noted that the striped path line visualization has to be interpreted always in downstream direction because  $\sigma$  and  $\delta$  are computed in positive time direction. The different modes presented in Section 4.2.3 and Section 4.2.4 are selected interactively until satisfactory visualization is achieved.

Slices, streamlets, and wall distance indicators (Section 4.2.4) are placed either at evenly timed or evenly spaced positions along the path line. When using wall distance indicators and vorticity streamlets at the same time, it is preferable that the vorticity streamlets get placed at the same position as the wall distance indicators. This improves the perception of streamlet orientation because either both positive and negative parts of the streamlet tend to be visible and hence in the plane of the top face of the tube, or one disappears inside the tube. Another benefit is that the top faces of the support tubes tend to have the same orientation as the wall. This facilitates inspection of vorticity in boundary shear flow because vorticity is parallel to the wall as well in the general boundary shear flow case.

## 4.3 Results

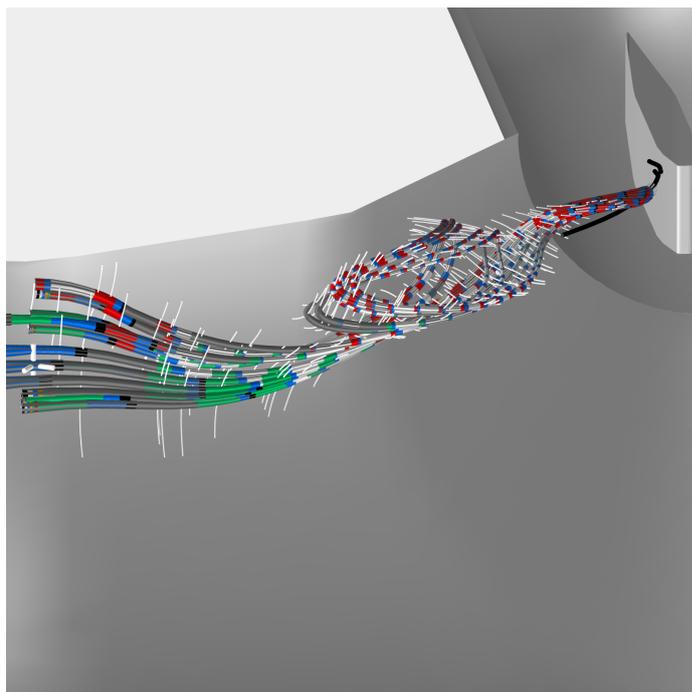
In this section, the proposed methods for vorticity transport analysis are applied to several CFD simulations from industry. The first example is the flow inside the distributor ring of a Pelton water turbine from Chapter 3. The second example is a free-surface simulation of the basin of a river power plant. Finally, the methods are applied to the simulation of a vortex rope in a draft tube of a Francis turbine. The former two are steady-state simulations while the latter is unsteady. Different flow phenomena are examined using the described methods.

### 4.3.1 Separation Vortex

The first region of inspection in the Pelton dataset is the large separation vortex at the entrance of the first injector. Figure 4.3 shows a set of path lines started in upstream direction from the downstream end of its core line. Figure 4.4 (a) shows a selected path line from this set and Figure 4.4 (b) shows the corresponding path line plot. The images show a part of the distributor ring of the Pelton water turbine. The flow comes from the bottom left and follows the distributor ring to the upper right corner. Part of the flow branches off into the injector at the upper left corner of the image.

We start the visual examination at the upstream end of the path line and follow it downstream (Figure 4.4). From the plot or from the distance indicators we know that the path line is close to the wall in the beginning. There is high vorticity magnitude and this indicates together with the perpendicular orientation of the vorticity streamlets that the path line is part of the boundary shear flow there. There is a large discrepancy at the beginning. This can be explained by the fact that simulation accuracy and quality of the computed derivatives are low close to the wall. However, it can be seen from the plot, that the error reduces quickly to a practical level as the path line gains distance to the wall. It can also be seen that vorticity is diffusing away (blue stripes), i.e., from the boundary shear flow in this region. This exemplifies the common fact that in the beginning, vorticity is typically diffusing away from boundary shear flow and only later on it is transported by advection. The advection process can be identified by the subsequent path line region of comparably constant radius and thin data stripes.

Next, the path line starts to deviate into the injector and aligns more and more with the vorticity field, which can be again interpreted as vortex tilting. It can be seen that this is near the upstream end of the separation vortex core line structure, and that the path line swirls around the core line of that vortex for the rest of its way. Here we observe the typical situation that vorticity is mostly concentrated by stretching in the region of vortices with simultaneous loss by diffusion. The positive stretching of vorticity is often observed in vortices that are located at regions of increasing velocity, e.g., the outlets. The velocity component parallel to vorticity typically increases downstream in these regions due to the pressure



**Figure 4.5:** Recirculation region. Path lines computed in upstream direction from a circle at downstream end of vortex core line. Vorticity streamlets allow to identify a recirculation zone (left, vertical streamlets) and a vortex region (right, horizontal streamlets).

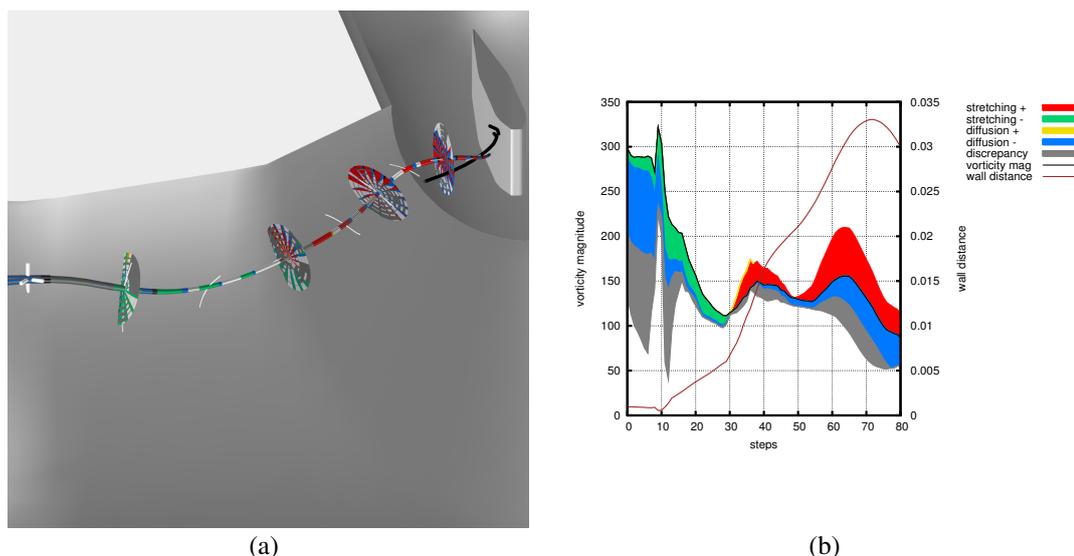
gradient, leading to stretching of the fluid and hence its vorticity. The loss by diffusion of vorticity is explained by the locally high vorticity magnitude typically found in the core region of a vortex, also due to concentration by vortex stretching.

### 4.3.2 Recirculation and Vortex

The entrance of the second injector of the Pelton distributor ring shows a quite different flow pattern. Instead of a clear separation vortex, a short vortex connected to a large recirculation region is observed. Figure 4.5 shows a set of path lines seeded at the downstream end of the vortex, Figure 4.6 (a) a selected path line, and Figure 4.6 (b) the corresponding path line plot. Again, the flow comes from the lower left corner of the image and part of it is branching off into the injector.

Generally, recirculation zones are often directly related to the boundary layer: they often occur when the boundary shear layer separates from the wall. Again, the upstream parts of the path lines are very close to the boundary. Observing the almost constant tube width and thin data stripes, almost pure advection of vorticity from the boundary shear flow can be identified.

The upstream parts of the path lines belong to the recirculation zone, identified by vertical vorticity streamlets. Some of the path lines follow a loop through

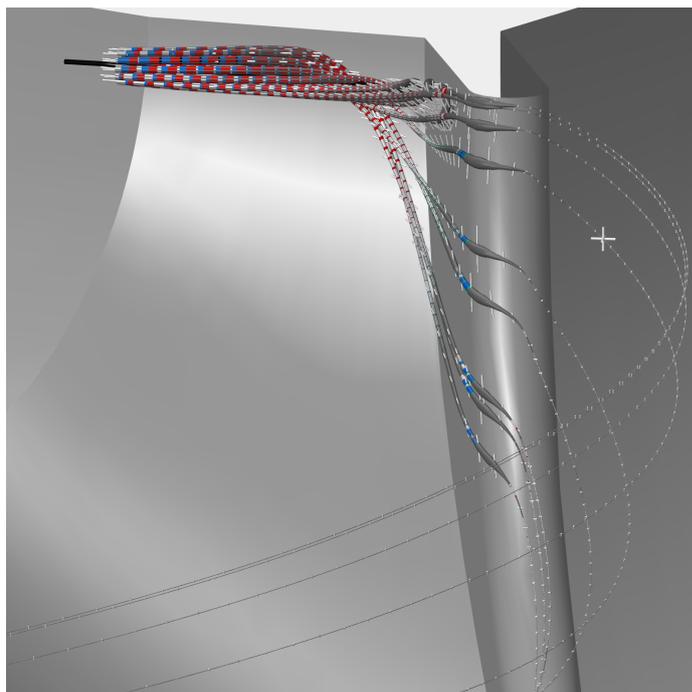


**Figure 4.6:** (a) Selected path line from Figure 4.5 with slices showing asymmetrical distribution of the quantities in the neighborhood of the path line. (b) Corresponding path line plot.

the recirculation zone before they reach the vortex. Part of this loop exhibits vertical vorticity streamlets which are consistent with the recirculation zone. In the other part of the loop, vorticity orientation is consistent with the vortex. The dominant mechanisms in this example are therefore the advection of vorticity from the boundary shear, followed by vortex tilting in the recirculation zone, and finally again positive vortex stretching plus loss of vorticity by diffusion inside the vortical region, as in the case of Section 4.3.1.

### 4.3.3 Vortex at Bifurcation

In this section a vortex near the sickle of the fourth Pelton injector is analyzed. The sickle is a plate at each bifurcation of a Pelton distributor ring which is inserted for mechanical reasons. Figure 4.7 shows a set of upstream path lines, started from the downstream end of the core line. Figure 4.8 (a) shows the plot corresponding to the selected path line. Following the path lines in downstream direction, part of them gets very close to the sickle where vorticity is obtained from the boundary shear flow and then advected into the vortex. Again, vorticity is concentrated by stretching with simultaneous loss by diffusion in the vortex region. Some of the regions where the path lines are close to the wall show high discrepancy  $\Delta$ , visible as gray peak in the plot. The visualization of the Courant number in Figure 4.8 (b) confirms that there are possible problems with simulation numerics in this region, comprised in the outermost layer of grid cells. This is a case where the computed diffusion may not be reliable close to the wall and hence is an example where the basic method (Section 4.1.1) could be appropriate.

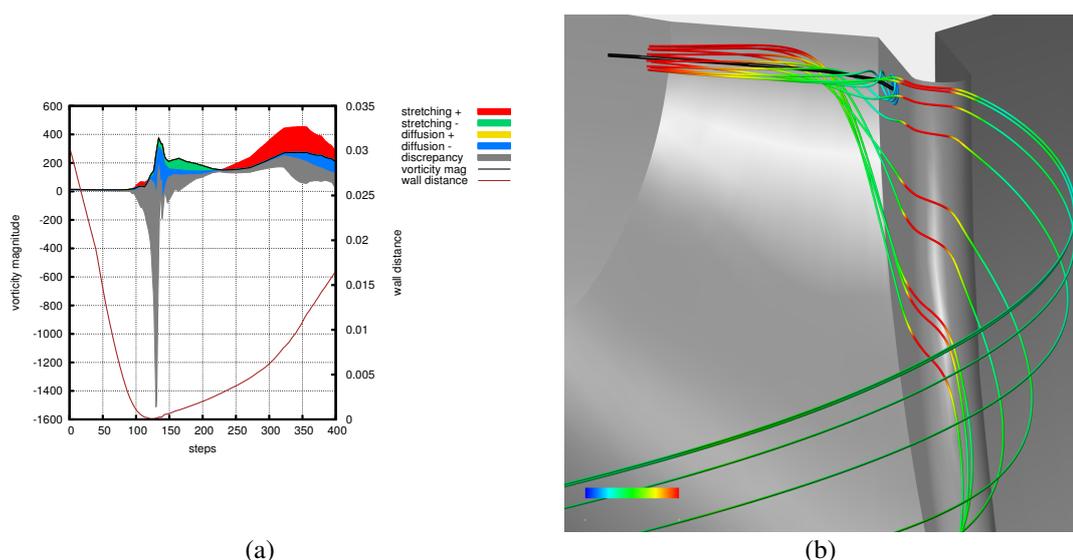


**Figure 4.7:** Set of upstream path lines, started on a circle at the downstream end of the core line (black). Vorticity increases rapidly as path lines pass the sickle in downstream direction.

### 4.3.4 Vorticity Stretching at Boundary Layer Separation

Now we examine a region in the river power plant dataset near a free-surface vortex (see Figure 4.9 (a)). The flow comes from the bottom of the image and passes around the housing of the generator, visible at the right side of the image. A streamline is generated interactively close to the bottom of the basin and used for generating a  $(u, \omega)$  rake of vortex lines for the visualization of the boundary shear flow, as described in Section 4.2.4. Some of the vortex lines detach in the region of the vortex and cover it. In order to analyze the flow in the region of flow detachment, a  $(\omega, u)$  rake is interactively generated close to the wall inside the vortex region. It can be seen that its primary vortex line keeps well inside the vortex region, due to the orientation of near-wall vorticity inside the vortex region. The complementary rake is built using striped path lines as shown in Figure 4.9 (b). This time, evenly-spaced tube segments are used instead of evenly-timed segments.

The main feature in this region is the vorticity gain by vortex stretching after the path lines have left the boundary shear flow. Stretching is due to the increasing velocity component perpendicular to the wall which is roughly the direction of vorticity, visualized by vorticity streamlets in Figure 4.9 (b).



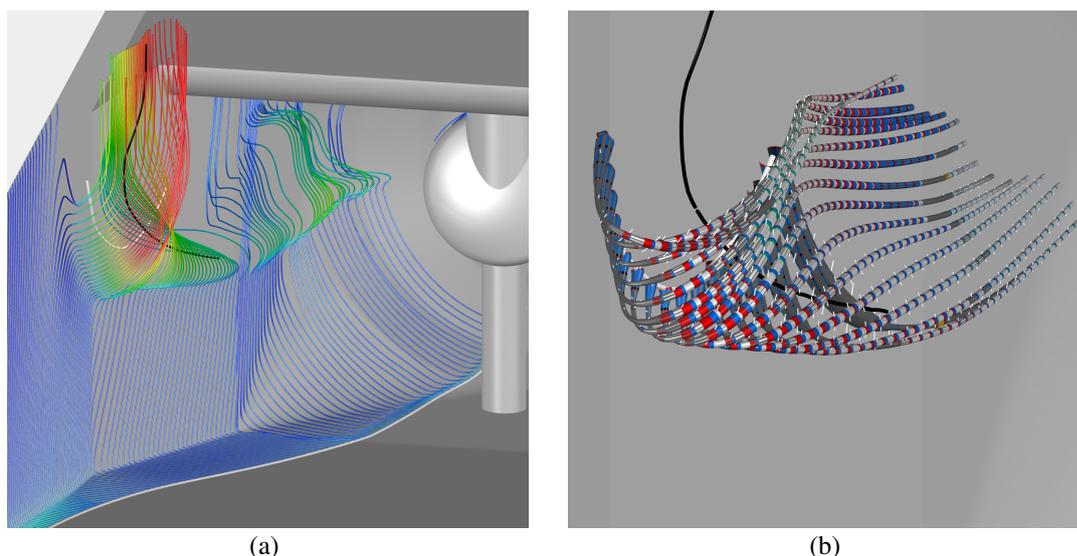
**Figure 4.8:** (a) Cropped path line plot of sickle vortex corresponding to selected path line in Figure 4.7. (b) Visualization of the Courant number. Comparison with Figure 4.7 shows that the error on the striped path lines correlates with high Courant number.

### 4.3.5 Transient Vortex Rope

Finally, we examine a transient simulation of a vortex rope in the draft tube of a water turbine by means of the presented method. The inlet boundary condition for the simulation was determined empirically from measured data on a slice through an already developed vortex rope. To study the vorticity transport into the vortex, its vortex core line is determined for the last time step of the simulation and used for seeding path lines in upstream direction as shown in Figure 4.10 (a). Additionally, the vorticity transport in the boundary layer is visualized by six path lines seeded on a circle at the inlet and computed in downstream direction.

The path lines that lead into the vortex exhibit strong alternating gain and loss of vorticity by stretching. It can be seen from Figure 4.10 (b) that most of the vorticity that is first gained by stretching is lost on the way again by negative stretching, but after all, most of the vorticity at the inlet boundary condition reaches the vortex core line of the last time step. The path line that is seeded at the lower end of the core line shows that vorticity magnitude is low around the vortex. It can be summarized that the vorticity concentrated in the center region of the inlet boundary condition is responsible for the development of the vortex rope.

The six path lines of the boundary layer first show gain of vorticity by diffusion and later loss of vorticity by diffusion. In between, vorticity is also increased by stretching. This can be identified as development of the boundary shear flow of the draft tube, i.e., the path lines enter the boundary shear flow as its front grows from the wall by diffusion. The asymmetry of the boundary shear flow (the path



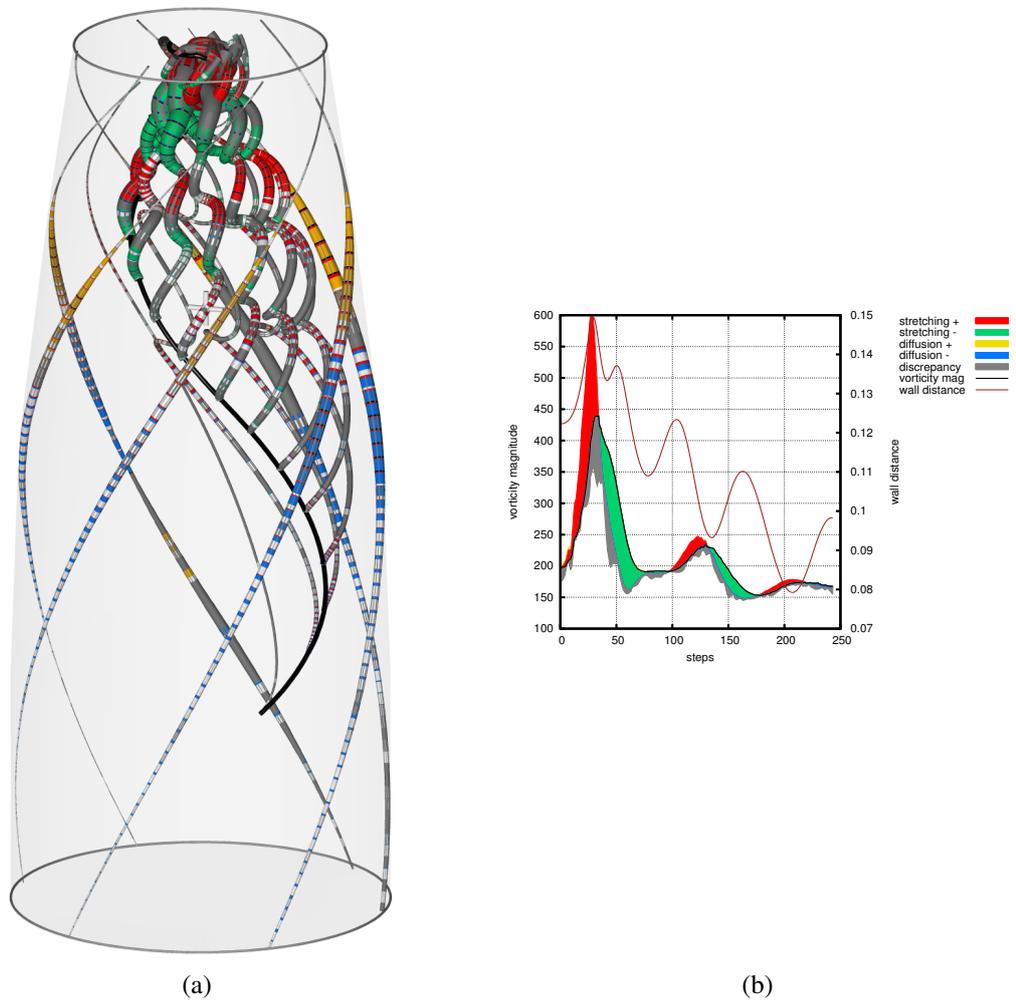
**Figure 4.9:** (a) Separation at large free-surface vortex. Core line of vortex is black. Vortex lines of the boundary shear flow are colored by distance to the boundary. The vortex lines of the boundary shear flow detach at the vortex and cover it. A vortex line has been generated close to the wall inside the vortex region (white tube). (b) Close-up view of vortex region showing a rake of striped path lines computed in downstream direction, started from the vortex line.

line at the back of the tube has larger radius) originates from the helical and hence asymmetric structure of the vortex rope.

## 4.4 Discussion

Different methods for the visualization of vorticity transport in unsteady flow have been presented. Integrating various tools based on scientific and information visualization techniques enables both qualitative and quantitative analysis by combining them flexibly in an interactive environment. The visualization toolkit gave deeper insight into the vortex dynamics of the examined cases, revealing how vorticity and vortices are reinforced or weakened by the main flow field. As discussed at the various examples, this method has the potential to support a more thorough understanding of complex flow patterns and might become beneficial in the development and application of turbulence models.

The strength of this method is the fact that a complex flow field can be visualized in a compact way without the need of switching back and forth between different representations, even for time-dependent fields. As a limitation of the system, one can say that the visualization is quite complex and that it requires some familiarity with the method and the underlying physics.



**Figure 4.10:** (a) Transient vortex rope simulation. Path lines computed in upstream direction from vortex core line of last time step (black tube). Six path lines computed in downstream direction visualize vorticity transport in the boundary shear flow. (b) Plot of selected path line.



**Part II**

# **Flow Topology**



## Chapter 5

---

# Invariant Manifolds in Flow Fields

The second part of this thesis presents research in the field of vector field topology. We start with a short and simple investigation of flow simulation data by means of streamlines and stream surfaces based on vector field topology, i.e., 1D and, in particular, 2D manifolds of saddle-type *critical points* and saddle-type *periodic orbits*. Thereby, we address the special case of divergence-free vector fields which is often met in simulations of incompressible fluids, such as water, and extend the topological analysis to *no-slip* boundaries by treating 3D velocity and 2D *wall shear stress* in a unified way.

Compared to arbitrarily chosen stream surfaces, such 2D manifolds can be more expressive and in many cases also of a simpler shape. In particular, recirculation zones and separation surfaces are well suited for this type of visualization. The underlying idea of visualizing topologically meaningful stream surfaces and their relationship to topological features has previously been used by Garth et al. [38] in their visualization of a *vortex breakdown* in the flow over a delta wing, a phenomenon which is subject of research in Chapter 6.

The concept of vector field topology requires at least Lipschitz continuous 2D or 3D vector fields to assure unique streamlines. However, to allow for the common classification approach by the gradient of velocity, differentiable vector fields are required. Usually, no further preconditions are required for its application. This is appropriate in the context of dynamical systems [43], which was the original application of vector field topology. Vector fields arising in physics, however, often exhibit additional properties, i.e., they may be divergence-free or irrotational or both. In Section 5.1 we will explore some of the implications of zero divergence to vector field topology and its application to the visualization of flow structures. Another property often met in CFD simulations are no-slip boundary conditions imposed on the fluid, i.e., forcing the fluid to adhere to the boundary. Extending vector field topology to fluids with no-slip boundaries is the topic of Section 5.2.

## 5.1 Topology of Divergence-Free Vector Fields

Divergence-free vector fields, also called *solenoidal*, are encountered in many disciplines, and in particular in fluid dynamics. Examples include fields in hydrodynamics, magnetism, and also derived fields such as the vorticity field, exploited in Chapter 3. Additionally, if a vector field does not exhibit neither sources nor sinks, it can be turned into a divergence-free field by multiplication with an appropriate scalar field. This is based on the fact that multiplication with a nonvanishing scalar field cannot change its topology. As an example, the momentum field exhibits the same topology as the underlying velocity field, because they are identical up to a nonvanishing scaling factor, the density. If, for example, the velocity field is a steady-state solution of the compressible continuity equation  $\partial\rho/\partial t + \nabla \cdot (\rho\mathbf{u}) = 0$ , then the corresponding momentum field is divergence-free.

The special case of divergence-free vector fields has implications on the analysis of critical points. Asimov [3] mentions that in 2D and 3D divergence-free vector fields, sources and sinks are not possible, but any types of saddles are. And in the 2D case, there is a new structurally stable type of critical points, namely the center. The center is said to have *constrained structural stability*, and has the property that in a neighborhood, all streamlines are closed.

A similar analysis as for critical points can be done for periodic orbits in divergence-free 3D vector fields. Periodic orbits are of interest as they can indicate recirculation zones. As described in Section 2.5.2, many properties of the periodic orbit can be studied in two dimensions by means of the Poincaré map.

The Poincaré map  $\mathcal{P}$  has a fixed point where it is intersected by the periodic orbit. For the eigenvalue analysis,  $\mathcal{P}$  is linearized in a neighborhood of a fixed point. This linearized map  $\mathbf{P}$  takes an infinitesimal circle centered at the fixed point to an ellipse with the same center. If the velocity field is divergence-free and thus volume preserving, the fluxes through the circle and the ellipse are equal. The flux is the integral of the normal velocity over the area enclosed by the circle or ellipse. The normal velocity can be linearized as well, and due to symmetry, replaced by its average. It follows that  $\mathbf{P}$  is area conserving, i.e., has a determinant of one. The sign is positive because a Poincaré map always conserves orientation.

### 5.1.1 Saddle and Twisted Saddle Periodic Orbits

It can be readily seen that periodic orbits of type source or sink are not possible for a divergence-free vector field. In the case of a source (either node source or spiral source), both eigenvalues lie outside of the complex unit circle. Hence, the determinant of  $\mathbf{P}$  has modulus greater than one, meaning that the area of an infinitesimal circle is not conserved under  $\mathcal{P}$ . The same can be concluded for sinks.

In contrast, periodic orbits of type saddle and twisted saddle are possible in divergence-free vector fields. Such periodic orbits are particularly suitable for visualization because they have a stable and an unstable manifold which are stream surfaces converging to the periodic orbit in positive or negative time. The nice property of these manifolds is that they “return to themselves” when following the periodic orbit for a full turn. This means, if a streamline is seeded on the intersection of the manifold with a Poincaré section and sufficiently close to the periodic orbit, it will return to the same intersection curve. If the seed curve is reduced to an infinitesimal line segment, its behavior is given by the eigenvalues of  $\mathbf{P}$ . As detailed in Section 2.5.2, if both eigenvalues are positive, the generated stream surface band returns untwisted to the Poincaré section or may have done an integer number of full (360 degrees) so-called extrinsic twists. And it can shrink or stretch, depending on the eigenvalue associated to the eigenvector aligned with the seed line. If both eigenvalues are negative, the stream surface band does an additional half twist. In our case of divergence-free vector fields the product of the two eigenvalues equals one because of the above-mentioned conservation of area. Because of their property to return to the seed curve, (un-)stable manifolds are ideal stream surfaces to depict the local behavior near the periodic orbit.

### 5.1.2 Center Periodic Orbits

If the Poincaré map of a periodic orbit in a divergence-free vector field exhibits complex eigenvalues its type can be neither spiral source nor spiral sink. It must be the in-between case with eigenvalues on the complex unit circle. Although the periodic orbit is not *hyperbolic* in this case, it exhibits constrained structural stability similar to that of center critical points in 2D divergence-free fields. By analogy, we call it a *center periodic orbit*.

The linearized Poincaré map  $\mathbf{P}$  of such a periodic orbit has complex eigenvalues and a determinant of one. It can therefore be written as  $\mathbf{P} = \mathbf{T}\mathbf{R}\mathbf{T}^{-1}$  where  $\mathbf{R}$  is a pure rotation. It follows that  $\mathbf{T}$  applied to an infinitesimal circle yields an ellipse which is invariant under  $\mathbf{P}$ . This means that a stream surface seeded at this ellipse returns to the ellipse after following the periodic orbit for a full turn. The same idea can be used for finding finite invariant tori. The goal is here to find a closed seeding curve in the Poincaré section which is invariant under  $\mathcal{P}$ . As an initial guess a scaled version of the infinitesimal ellipse can be used. If starting from this an invariant seeding curve can be found, the problem is solved. However, we found that in practice this is a numerically challenging problem.

## 5.2 Topology near No-Slip Boundaries

As defined in Section 2.5.1, a critical point is an isolated zero of the vector field; vector field topology does not treat extended singularities. However, these occur

in practical flow fields having solid boundaries with associated no-slip boundary conditions. Velocity  $\mathbf{u}(\mathbf{x})$  vanishes on such a boundary, but using the unsigned distance to the boundary, i.e., wall distance,  $s(\mathbf{x})$ , it can be written as a product

$$\mathbf{u}(\mathbf{x}) = s(\mathbf{x})\tilde{\mathbf{u}}(\mathbf{x}), \quad (5.1)$$

where the related vector field  $\tilde{\mathbf{u}}(\mathbf{x})$  can be assumed to be well-defined also on the boundary and to be nondegenerate there.

From the assumption of zero divergence it follows for points on the boundary:

$$0 = \nabla \cdot (s\tilde{\mathbf{u}}) = (\nabla s) \cdot \tilde{\mathbf{u}} + s(\nabla \cdot \tilde{\mathbf{u}}) = (\nabla s) \cdot \tilde{\mathbf{u}} \quad (5.2)$$

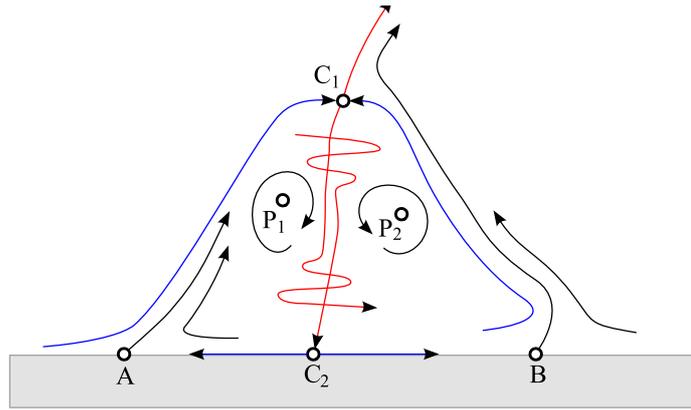
which means that on the boundary the field  $\tilde{\mathbf{u}}$  has no normal component. In terms of vector field topology this means that no streamline of  $\tilde{\mathbf{u}}$  can pass from the solid boundary to the interior or vice versa.

If Eq. 5.2 holds, then on the boundary,  $\tilde{\mathbf{u}}$  is related to the wall shear stress  $\tau_w$  by  $\tau_w = \mu\tilde{\mathbf{u}}$  where  $\mu$  is the dynamic viscosity of the fluid, assuming *Newtonian fluids*, i.e., uniform viscosity. Because of this proportionality  $\tilde{\mathbf{u}}$  on the boundary has the same topology as  $\tau_w$ . At interior points,  $s$  is nonzero and therefore  $\tilde{\mathbf{u}}$  has the same topology as  $\mathbf{u}$ . Hence, the field  $\tilde{\mathbf{u}}$  nicely combines the wall shear stress field with the interior velocity field. However, this relies on the divergence-free property of the vector field. In the general case the field  $\tilde{\mathbf{u}}$  has a normal component on the solid boundary and hence it cannot be used to produce the topology of both the velocity field and the wall shear field in a generalized way. Of course the two vector fields could be blended, but then the topology of  $\mathbf{u}$  may not be conserved.

Returning to the case of a divergence-free field, we saw that no streamline of  $\tilde{\mathbf{u}}$  passes from the interior to the boundary. But there may be convergence toward critical points on the solid boundary, which are 3D saddle points having two of its eigendirections along the boundary surface. Also, convergence toward periodic orbits on the boundary is possible.

The advantage of using the field  $\tilde{\mathbf{u}}$  is that it is no more necessary to extract both 2D and 3D critical points in different fields (with possible consistency issues). Critical points on the boundary are now regular 3D critical points. In the special case of  $\mathbf{u}$  being divergence-free, sources and sinks can be excluded due to structural stability and the fact that  $\tilde{\mathbf{u}}$  has the same topology as  $\mathbf{u}$ . Consequently, such critical points must be saddles or spiral saddles. Furthermore, by Eq. 5.2 their 2-dimensional stable or unstable manifolds lie completely on the boundary. The eigenvalue belonging to the remaining eigenvector is real-valued. Its sign determines whether the point is on a *separation line* (positive sign) or an *attachment line* (negative sign).

In discrete data, dividing by  $s$  has the drawback that topology can change due to interpolation inside the cells. A better strategy is to use the original field  $\mathbf{u}$  for computing and analyzing the critical points in all cells that are not adjacent to no-slip boundaries. Only for computing the topology in the first layer of cells at the boundary, the modified field  $\tilde{\mathbf{u}}$  is actually needed. The following steps are performed for cells adjacent to no-slip boundaries:



**Figure 5.1:** Sketch of typical recirculation zone with two critical points of type spiral saddle ( $C_1, C_2$ ) and one periodic orbit ( $P_1, P_2$ ) involved. 1D manifolds (red curves) nearly meet. 2D manifolds (blue curves) have a strong spiraling component.

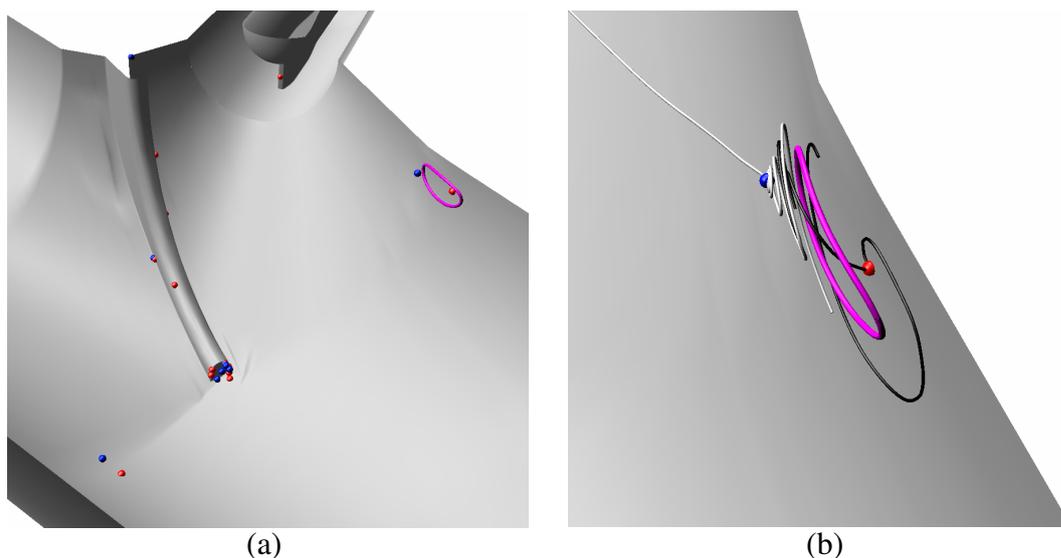
1. On their interior nodes: compute  $\tilde{\mathbf{u}}$  by dividing  $\mathbf{u}$  by the wall distance.
2. On their boundary nodes: interpolate  $\mathbf{u}$  on two points on the boundary normal, compute  $\tilde{\mathbf{u}}$ , and linearly extrapolate  $\tilde{\mathbf{u}}$  to the boundary node.
3. Find critical points inside  $\tilde{\mathbf{u}}$  on the cell faces on the no-slip boundary. Use a 2D algorithm to find the critical points, classify them as 3D critical points.

## 5.2.1 Critical Points on No-Slip Boundaries

Critical points on no-slip boundaries are important features for the study of flow separation and flow attachment. By applying the 3D classification, we will now concentrate on saddles and spiral saddles and ignore sinks and sources. These are of minor interest for the study of these phenomena, and furthermore, they cannot occur in divergence-free vector fields.

Under the given preconditions the stable manifolds of 2-saddles and 2-spiral saddles perfectly follow the boundary and their boundary curves are attachment lines or domain boundaries. Similarly, the unstable manifolds of 1-saddles and 1-spiral saddles are also following the boundary and are confined by separation lines or the domain boundary.

A pattern we encountered often consists of a pair of spiral saddles, one of them in the interior and one on a solid boundary (see Figure 5.1, points  $C_1$  and  $C_2$ , respectively). They are rotating in the same sense and mark a recirculation area. The 1D manifolds nearly meet, while the 2D manifold of  $C_1$  encloses the recirculation zone. This stream surface is not closed, so recirculation is not perfect. Within the recirculation zone there is a periodic orbit ( $P_1$  and  $P_2$ ). Finally, the points  $A$  and  $B$  appearing as saddles in the planar section, indicate separation. However, these points are topologically nothing special, they are just the points where a skin fric-



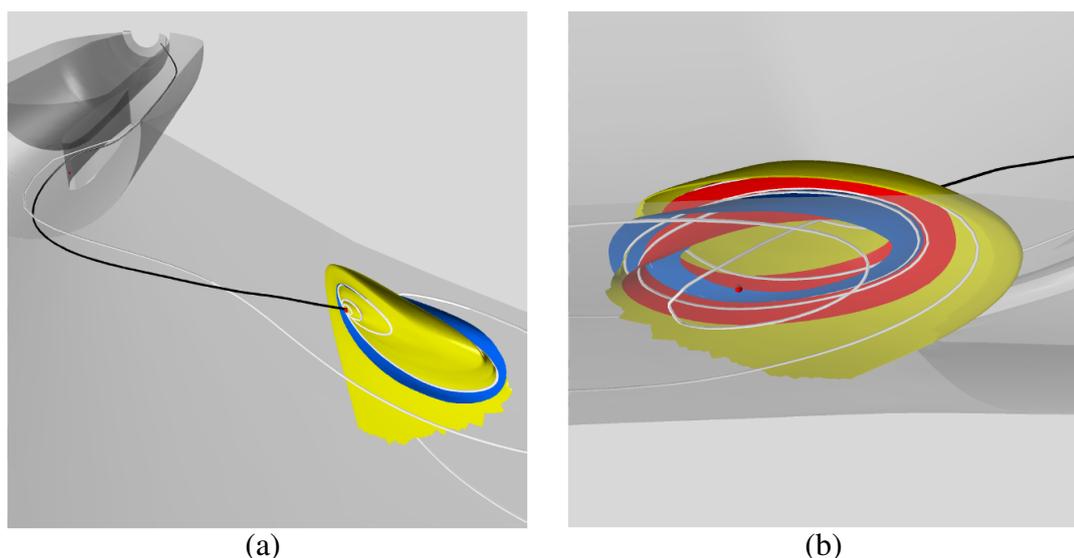
**Figure 5.2:** (a) Extracted interior (blue) and boundary (red) critical points. Periodic orbit (magenta). (b) Streamlines seeded near the boundary critical point (black) and the interior critical point (white).

tion line is intersected orthogonally by the planar section. Surana et al. in their exact theory of flow separation [146] suggest as a criterion "strong hyperbolicity", i.e., large absolute eigenvalues of the saddles in the orthogonal section. An alternative and purely topological definition would be to pick the boundary curve of the unstable manifold of  $C_2$  (which lies entirely on the solid boundary). In general this is composed of separatrices of nearby saddle points on the solid boundary (*closed separation*). Of course, traditional explicit methods for detecting separation and attachment lines can also be used, such as that by Kenwright et al. [68] which is also able to detect *open separation*.

## 5.3 Results

We exemplify the described methods again on the Pelton turbine dataset. We again inspect the region of the first injector (Figure 3.6 (b)), where the red stream surface visualizes the separation vortex and the yellow one a smaller scale tornado-type separation involved in a recirculation zone. Inspecting the nearby critical points reveals that there is a pair of spiral saddles in this region, one of them is on the no-slip boundary (upper right in Figure 5.2 (a)). A quick exploration by integrating a streamline forward and backward from seed points near the critical points gives an idea of the stable and unstable manifolds of the two spiral saddles (Figure 5.2 (b)).

Consistent with the situation sketched in Figure 5.1, the stable manifold of the interior critical point encloses the recirculation zone (Figure 5.3 (a)). The recirculation zone contains a single periodic orbit which is of twisted saddle type (Fig-



**Figure 5.3:** (a) The stable manifold of the interior critical point (yellow) and the unstable manifold of the periodic orbit (blue). A nearby streamline (white) approaches the periodic orbit along the red stream surface, leaves it along the blue stream surface, approaches the critical point along the yellow stream surface and finally leaves it along the 1D manifold opposite the black streamline. (b) Back view showing the stable (red) and unstable (blue) manifolds of the periodic orbit and revealing their Möbius strip topology.

ure 5.3 (b)). In this case, the stable and unstable manifolds of the periodic orbit are classical Möbius strips with a half twist and no further extrinsic twisting.

Similar flow patterns as near the first injector also appear near the third and fifth injector. In all cases, a periodic orbit of type twisted saddle can be observed. However, in the case of the third injector, the eigenvalues are relatively close to  $-1$ , which suggests that instead of the twisted saddle, the center type (with a rotation angle close to 180 degrees) would be possible as well for slightly different data.

## 5.4 Discussion

We have seen flow features in real divergence-free CFD datasets which can be nicely illustrated by 2D manifolds of 3D saddles. However, experience showed that stream surface integration gets particularly challenging for these special cases of stream surfaces. This gave rise to a “topologically aware” stream surface integration technique [108], which is, however, not described in detail in this thesis because it does not fit its core topics. We now proceed to a phenomenon related to periodic orbits: vortex rings and the phenomenon called vortex breakdown.



## Chapter 6

---

# Vortex Rings

In the previous chapter we had a short topology-oriented look on a recirculation region inside a Pelton turbine exhibiting two spiral-saddle critical points and a periodic orbit (Figure 5.1). In this chapter we present visualization methods for the identification, extraction, and analysis of such phenomena. If the periodic orbit is of type *spiral source*, *spiral sink*, or *center*, i.e., exhibits non-negligible swirl around its core line, the vortex is called a *vortex ring*. In other words, vortex rings are simply vortices that are closed, i.e., they exhibit a simply closed vortex core line. The most prominent example of such vortices are *smoke rings*. It can easily be seen from experiments that smoke rings exhibit swirling motion in planes perpendicular to the core line but negligible tangential velocities, i.e., rotation around the principal axis. This is also one of the reasons why they can advance at surprisingly high speeds and stay stable over long time periods. Such vortex rings represent one of the simplest configurations of recirculating flow which is not boundary-related, e.g., not related to flow separation, assuming that the vortex rings are stationary or that they are looked at in a relative frame of reference. The opposite extreme are periodic orbits of type *source*, *sink*, or the limit case in between. There is no swirling motion around the core line in this case, only tangential one. In this chapter we require non-vanishing swirl around the core line, i.e., we constrain the analysis to vortex rings. Surprisingly, such flows still can exhibit two fundamentally different topologies: the intuitively clear and simple configuration from Figure 6.1 (a) and an astonishingly rich one in the case of so-called *vortex breakdown bubbles* exhibiting *Shilnikov chaos* (Figure 6.1 (b)).

Unperturbed vortex breakdown bubbles (the former case) are axisymmetric and consist of nested invariant tori (depicted by the closed manifold around  $P_1$  and  $P_2$ ) on which all streamlines have to lie. In the simplest topological case, the nested tori are bounded on the inner side by a periodic orbit and on the outer side by a sphere-like stream surface. If the tori are allowed to have non-circular cross sections, then such vortex models can be physically valid, i.e., be solutions of Navier-Stokes equations. Examples are Hill's vortex (see [123]) and Norbury's vortex rings [101]. If we further assume that the vortex ring has nonvanishing swirl, (ro-

tational component around the principal axis of the tori), the whole structure has only two critical points. They are of spiral saddle (saddle focus) type, and are located where the axis intersects the outer boundary ( $C_2, C_3$  in Figure 6.1 and  $C_0, C_1$  in Figure 6.7 and Figure 6.6).

However, this ideal topology of a vortex ring is usually not found in practical velocity fields [153, 167]. The reason is that it requires the 2D unstable manifold of the one critical point to exactly coincide with the stable manifold of the other, and likewise for the pair of 1D manifolds, a configuration which is not structurally stable. In a perturbed version of this ideal vortex ring, the 2D manifolds do not exactly match, which means that the boundary is not “water-tight” or that the recirculation is not perfect. The 1D manifolds do not coincide either, so that the axis of the structure is no more well defined in a topological sense. As a side remark, this is an example of a vortex whose core line is not a streamline.

If the vector field is divergence-free, e.g., because of an incompressible fluid, the effect of a perturbation is even more dramatic. In divergence-free 3D vector fields with sufficient swirl, any transversal intersection of the 2D (un-)stable manifolds of two spiral saddles automatically implies the so-called Shilnikov chaos [136]. This is also illustrated in Figure 6.7, where  $C_0$  is a 1-spiral saddle and  $C_1$  is a 2-spiral saddle. In general, the 2D (un-)stable manifolds of such critical points do not coincide, but intersect transversally. In this case they intersect at an even number of saddle connectors (heteroclinic orbits), usually a pair  $\sigma$  and  $\sigma'$  of them. Between the windings of the saddle connectors, the manifolds form two “tubes” that are wrapped around the structure. The tubes have constant flux (i.e., independent of cross sections) because the 2D manifolds are stream surfaces, and the sum of the two fluxes is zero because of the divergence-free condition. This implies that near the critical points, where velocities approach zero, the tubes must either have increasing cross section area or develop folds which extend into regions of higher velocities. These folds, known as lobes, are typical of vortex breakdown bubbles (see, e.g., [139]). Furthermore, this type of vortex rings typically contains regions exhibiting chaotic dynamics with possible islands of stability and KAM tori (impermeable) or cantori (permeable) separating the regions (Figure 6.1 (b)), often exhibiting fractal configurations. We refer the reader to the paper of Sotiropoulos et al. [138] for details. While a vortex breakdown bubble and a smoke ring are two different physical phenomena, from a topological point of view, they can both be treated as vortex rings and are also denoted as such in the literature [75].

Vortex breakdown bubbles got their name from a phenomenon called *vortex breakdown* [88]. Under certain flow configurations which will be looked at below, a vortex can undergo a breakdown, meaning that the vortex abruptly changes its structure and topology. From its common tubular shape, it develops an additional vortex ring surrounding the main vortex core line, with non-negligible tangential velocities along the closed core line. Vortex breakdown bubbles can persist over long time periods and can be seen as a superposition of a vortex ring with a vortex around the central axis of the ring. Accordingly, they exhibit helical *vortex lines*, a superposition of the closed vortex lines of the vortex ring and the straight vortex

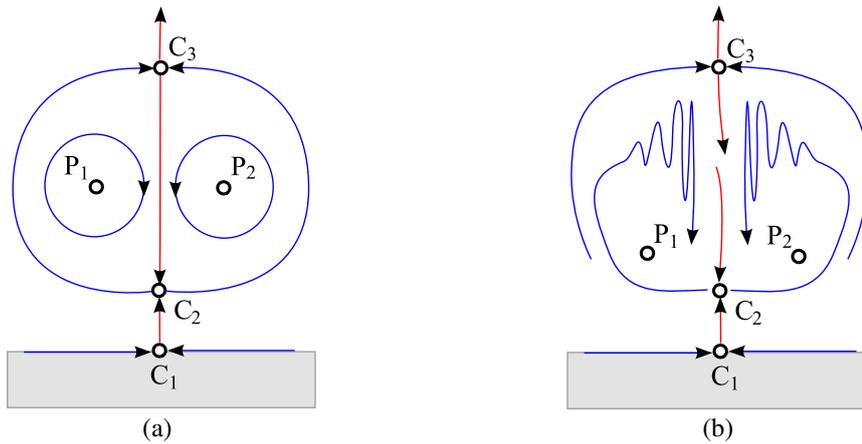
---

lines of the main vortex. Vortices typically exhibit a longitudinal velocity component along their core line. Whenever we identify adjacent pairs of critical points on a vortex core line, this indicates that the longitudinal flow direction is reversed along that core line segment, which often necessitates the presence of a vortex breakdown bubble. Due to the flow reversal, i.e., the recirculation, vortex rings often act as an obstacle for the main flow, leading to, e.g., reduced throughput, as detailed in the example of a Francis turbine simulation below.

Vortex breakdown bubbles are a subject which is of interest in many disciplines. In aeronautics, they mostly cause problems because of the sudden change in flow caused by a breakdown of a vortex. So-called delta wing airplanes often exhibit very flat airfoils that produce comparably low lift per se. They profit from large longitudinal vortices that are present on the top side of the wings, generating a large part of the lift. Breakdown of these vortices leads to an abrupt loss of lift and are feared during approach for a landing. Vortex rings that persist over longer times can also lead to increased drag in locomotion and reduced throughput in turbo-machinery. In chemical and physical production they are often beneficial because they improve mixing. Examples where breakdown bubbles are generated purposely are combustion processes and mixing in bio-reactors. As will be visualized later in this chapter, vortex breakdown bubbles exhibit chaotic advection, a process achieving high mixing performance. Furthermore, they feature fractal flow patterns in space as well as in time, which additionally enhances mixing.

Vortex breakdown bubbles in CFD data have been observed and visualized by Garth et al. [38, 39] using techniques based on stream surfaces, and by tracking the pairs of critical points associated with this structure. Tricoche et al. [158] used direct volume rendering for visualization and vector field topology in planar slices for extracting periodic orbits. Rütten and Chong [117] developed a visualization method based on tensor field color coding. The reader is referred to [88] for details on the different types of vortex breakdown.

Here we present topological approaches for the detection and extraction of vortex rings, which exploit the special topology of vortex rings and which are more closely oriented at the underlying theory of continuous dynamical systems. A vortex ring requires at least two saddle-type critical points, and in fact this minimum number is observed in many synthetic or real-world examples. Based on this observation, we present a visualization technique utilizing a Poincaré section that contains the pair of critical points. The Poincaré section by itself can be taken as a visualization of the vortex ring, especially if streamlines are seeded on the stable and unstable manifolds of the critical points. It is also used for obtaining seed curves for the integration of stream surfaces that convey the hull of the vortex ring, i.e., the stable and unstable 2D manifolds of the two saddles. As discussed at the end of Section 2.5.1, this is an example where constraining the size of the manifolds to the region of interest reduces occlusion and hence improves the visualization. The alternative approach of extracting only the intersection curves of the manifolds, i.e., the saddle connectors, does typically not give enough insight in this case. Therefore, we visualize the manifolds but present a method to limit their

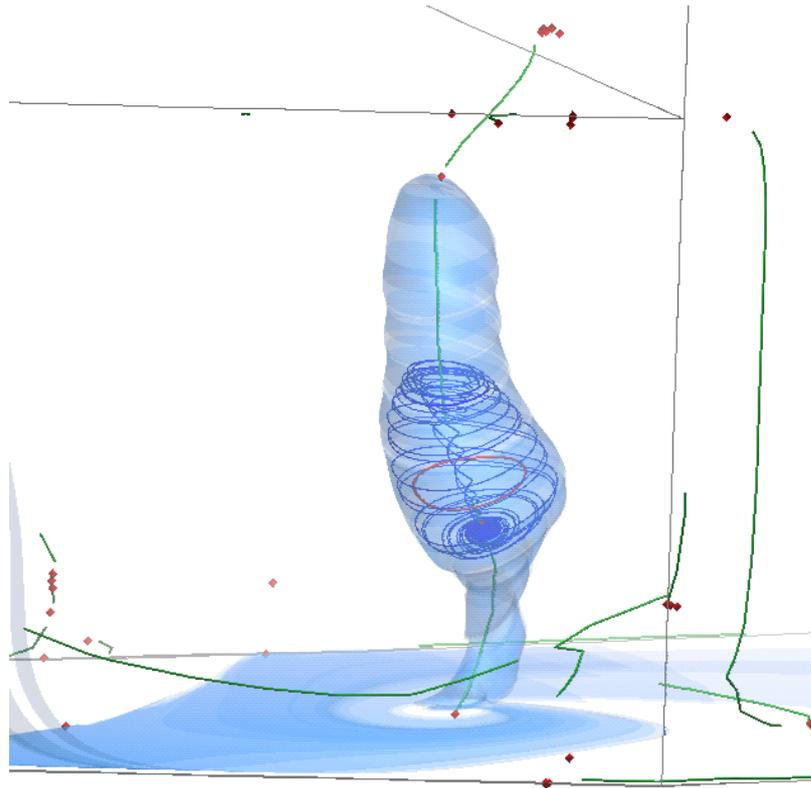


**Figure 6.1:** (a) Sketch of ideal (unperturbed) vortex breakdown bubble. Saddle-type critical points ( $C_1, C_2, C_3$ ), periodic orbit ( $P_1, P_2$ ), and 1D manifolds (red) and 2D manifolds (blue). (b) Sketch of real-world (perturbed) vortex breakdown bubble. No intersection of the two 2D manifolds due to non-vanishing divergence.

sizes such that the overall complexity is reduced. This is an important aspect in this context because the manifolds can be massively convoluted. This is typically due to Shilnikov chaos.

The presented methods can robustly and automatically handle features such as non-isolated periodic orbits within nested tori, periodic orbits of higher periodicities, and stream surfaces with strong folding, which are difficult to treat with general-purpose methods. For incompressible flow, they are able to visualize the characteristic properties of vortex rings such as the boundary, the chaotic dynamics, and possible invariant tori in *islands of stability*. The basic idea for the visualization of invariant tori is to find invariant closed curves in the Poincaré plane, which are then taken as seed curves for stream surfaces. For visualization the two extremes of the set of nested tori are computed. This is on the inner side the periodic orbit toward which the tori converge, and on the outer side, a torus which marks the boundary between ordered and chaotic flow, a distinction which is of importance for the mixing properties of the flow.

For the purpose of testing, we developed a simple analytical model of a perturbed vortex ring based on Hill's spherical vortex [123], which we also used in the field of real-time simulation of bubbles [154]. Since some of the methods require very long streamlines, the effect of numerically introduced divergence has to be considered. From an existing subdivision scheme, a method for divergence conserving interpolation of cuboid cells is derived, and results are compared with those from standard trilinear interpolation. Also a comparison of results obtained with and without divergence cleaning is given. We apply the proposed visualization methods to the synthetic vector field and to two CFD results.



**Figure 6.2:** Tornado-type separation and vortex in the draft tube dataset. Stream surface (transparent blue) starts at saddle and goes upstream enclosing a vortex breakdown bubble (blue streamline) containing a periodic orbit (red). Critical points (red) and vortex core lines (green).

## 6.1 First Look by Streamline Sampling

We start with a simple investigation of a vortex breakdown bubble in the incompressible CFD simulation of a Francis draft tube. The design of the draft tube is such that in its lower part it is split into two channels (Figure 6.13 (a)). As observed in the simulation data, the right channel exhibits significantly stronger vortices. For topological interests, and because we observed surprisingly long residence times of streamlines in this region, we picked one of the strong vortices extending horizontally and almost orthogonally to the primary flow direction. The transient simulation of this vortex consists of 3 interesting and comparably steady phases: first, there is a vortex ring that exhibits a massively folded unstable 2D manifold of a spiral saddle, enclosed in a simple stable 2D manifold of another spiral saddle (Figure 6.1 (b)). Then the vortex ring collapses and, in turn, the stable manifold develops folds around a new vortex ring. Finally this vortex ring collapses too, leaving a common vortex. Interestingly, one of the manifolds exhibits throughout the massive folding typical for vortex breakdown bubbles but

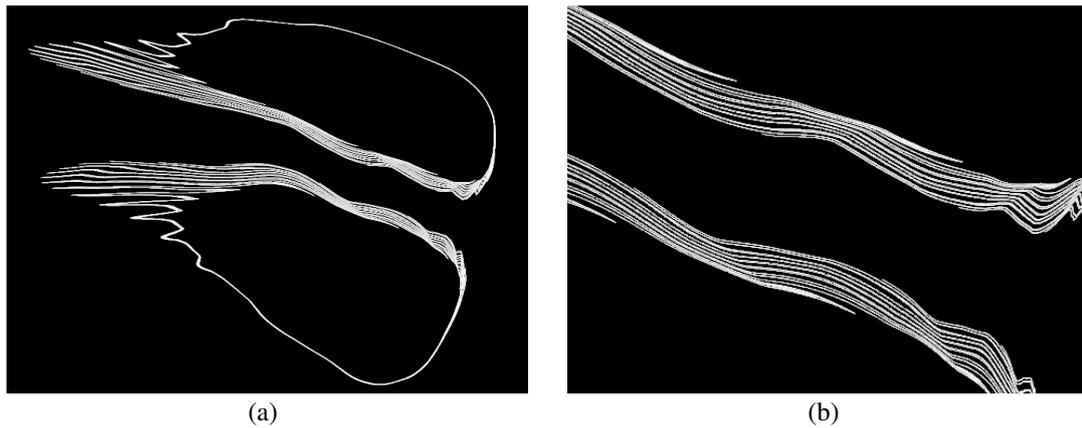
the manifolds do not intersect, see below the discussion of this circumstance. We have chosen a time step of the first phase where the vortex breakdown bubble is relatively steady and hence the examination of its instantaneous topology should reveal some of its dynamics. As reported by Spohn et al. [139], vortex breakdown bubbles indeed often result in steady flow behavior.

Figure 6.2 gives a view from top on the flow going from left to right. It shows (from bottom to top) a tornado-type separation with a critical point on the wall, and a vortex core line that connects to that critical point and spans across the channel into the part where the two channels merge. There is the recirculation region identified as a vortex breakdown bubble with a critical point at its bottom, with its unstable 2D manifold visualized by a streamline, and a periodic orbit inside it. Another critical point resides above the bubble where the detected core line is disrupted. The stable 2D manifold of that saddle is visualized by an upstream surface that encloses the vortex breakdown bubble and approaches the wall.

In our case, a vortex breakdown bubble containing a periodic orbit has been identified (Figure 6.1 (b)). The stable 2D manifold of the upper saddle (transparent stream surface in Figure 6.2) marks the end of the time-dependent recirculation region. However, Spohn et al. [139] report that vortex breakdown bubbles exhibit permanent inflow and outflow at the downstream tail. According to our topological analysis (Figure 6.1 (b)), the stable manifold of  $C_3$  and the unstable manifold of  $C_2$  do not intersect. Hence, there is an inflow through the gap between those two manifolds. By Gauss' theorem, this indicates non-vanishing divergence in the recirculation region. We interpret this circumstance that, although the simulation modeled an incompressible fluid, i.e., water, the simulation did not achieve zero residuals. Hence, we supply two approaches for handling non-vanishing divergence explicitly: divergence cleaning and divergence-preserving interpolation.

We conclude this section by a simple technique for visualizing the observed vortex breakdown bubble phenomenon. Computing the vortex breakdown bubble as a stream surface seeded at the critical point seems impossible with Hultquist-type algorithms due to the complex folding and also due to the quasi-periodicity of the streamlines. Since a single streamline covers the manifold of the vortex breakdown bubble densely due to its quasi-periodic dynamics, it can be seeded near the critical point and sampled on a voxel grid. The resulting field can then be visualized by an isosurface. To reduce aliasing effects and enhance resolution, a voxel value is not set in a binary manner when the streamline passes but accumulated based on coverage. The initial sequence of integration steps was not sampled in order to avoid an isolated spiral from the saddle point to the unstable manifold.

Figure 6.3 (a) shows a slice of the resulting voxel field after tracing the particle for  $10^9$  time steps. Its resolution is  $750 \times 600 \times 600$  and it spans the complete bubble. Figure 6.3 (b) shows a finer sampling of a subregion. This makes the massive folding of the surface visible. One can interpret the reason for the massive folding as follows: on its way from the critical point to the downstream end of the bubble, the manifold starts to exhibit folds by so-called *folding*, i.e., the expanding effect of the downstream saddle perpendicular to the manifold, amplifying perturbations



**Figure 6.3:** (a) Slice of the voxel field that sampled a single streamline. (b) Detail sampled at higher voxel resolution.

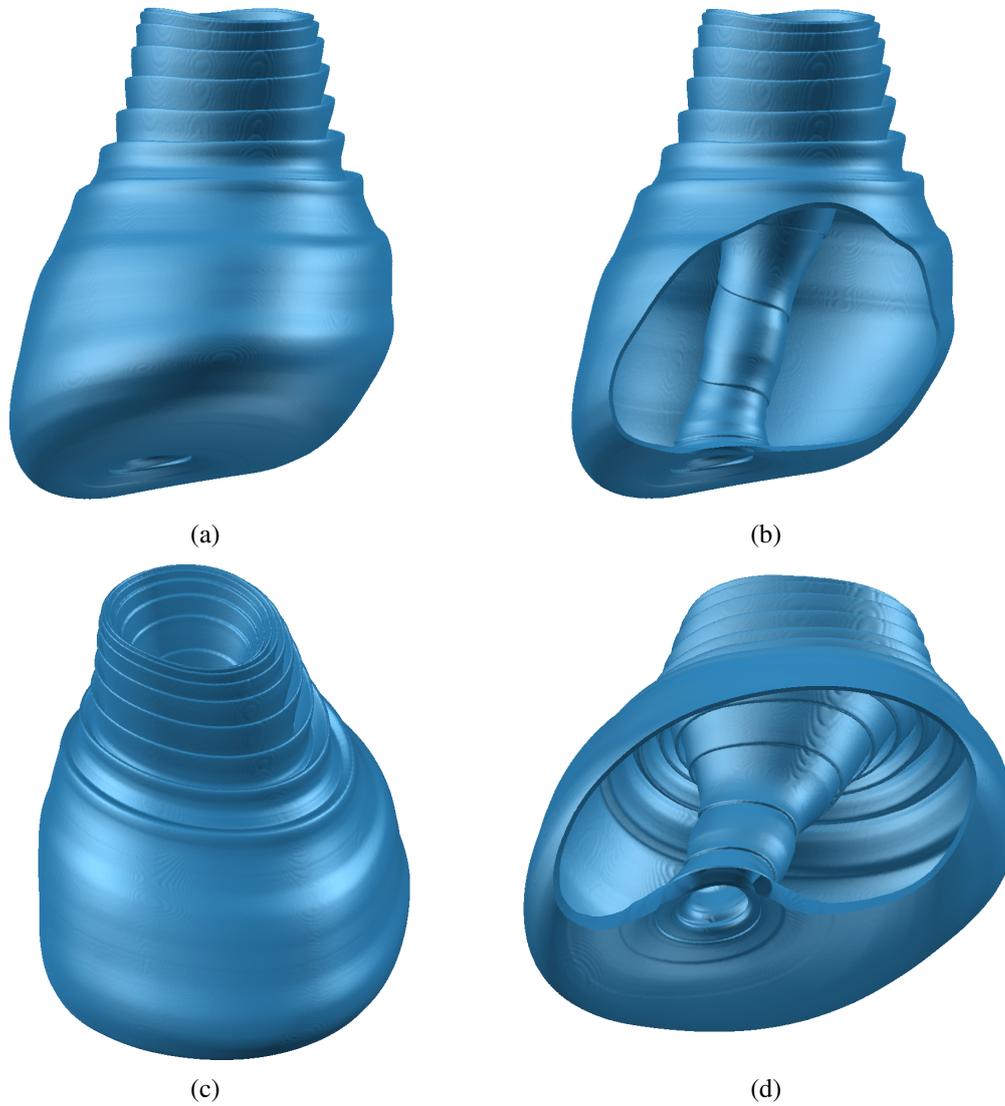
to folds. Then these folds follow the tube-like region at the center of the breakdown bubble toward the upstream critical point and get compressed and elongated by so-called *thinning*. This process repeats ad infinitum and therefore the manifold consists of folds of folds of folds and so forth. The dynamics of thinning and folding is present in many mixing processes, as kneading dough or stirring coffee.

Figure 6.4 shows an isosurface of the voxel field with the complete bubble. The isolevel was chosen to be 5% instead of 50% in order to avoid unmanageably many triangles in the fine folds. By adding a Gaussian smoothing step, we were able to cut down the triangle count to about 20 million.

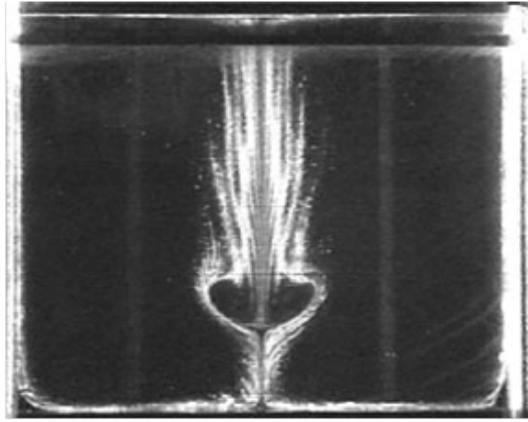
We applied isosurface extraction here although theoretically extracting a ridge surface (Chapter 7) would be more appropriate for the extraction of the sampled manifold. However, isosurface extraction exhibits two advantages in this context: it does not require the estimation of higher derivatives and it allows a robust surface extraction despite of the massive folding. An overall drawback of a regular sampling of the streamline are aliasing artifacts apparent in the resulting visualizations, a trade-off for the simplified surface extraction compared to stream surfaces.

## 6.2 Unsteadiness and Compressibility

When concepts from vector field topology are applied to unsteady vector fields, it is important to be aware of the consequences. If the unsteady vector field is quasi-stationary, vector field topology might give an accurate view and allow straightforward interpretations. However, the more unsteady the underlying vector field is, the harder it is to interpret the results in terms of transport mechanisms. This is especially true for vortex rings, because they often contain relatively long trajectories, and therefore the view based on vector field topology and hence streamlines may diverge substantially from that given by path lines. Haller [47] states that



**Figure 6.4:** Unstable manifold of the spiral saddle at the bottom of the vortex breakdown bubble in the Francis turbine dataset. (a) and (c): isosurface of the voxel-sampled streamline (Figure 6.3). (b) and (d): same isosurface clipped for view to the inside.



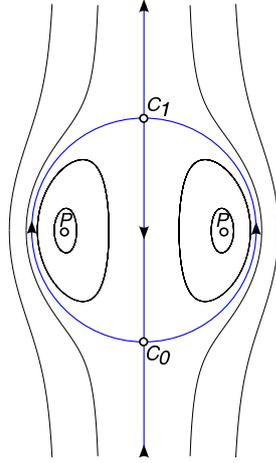
**Figure 6.5:** Photograph of vortex breakdown bubble in experiment. Image copied with permission from [139].

structures such as chaotic tangles or KAM tori (i.e., invariant tori of a Hamiltonian system) do not exist in finite-time turbulent datasets. Nevertheless we believe that it is interesting to search for such structures, first of all in steady flow fields. It can be demonstrated that these topological features exist in practical flow data, meaning that the catalog of features to be studied in vector field topology must include invariant tori, chaotic regions, intersecting stable and unstable manifolds and multiple saddle connectors. Fortunately, practical flow often exhibits small enough time-dependence so that their visualization as steady flow is a good enough approximation in many respects. Vortex rings and, in particular, vortex breakdown bubbles have been photographed in experiments [139] (see Figure 6.5) and their shape has been shown to be consistent with the manifolds of critical points in a steady vector field [138].

Much of dynamical systems theory deals with the special case of Hamiltonian systems, because of their area-conserving maps which are mainly responsible for chaotic behavior. Among the vector fields, the divergence-free ones play a similar role, and in fact they are related to Hamiltonian systems. In two dimensions, divergence-free vector fields (written as ODEs) and Hamiltonian systems are even the same, with the stream function  $\Psi$  (with  $\frac{\partial \Psi}{\partial x} = -\dot{y}$  and  $\frac{\partial \Psi}{\partial y} = \dot{x}$ ) playing the role of the Hamiltonian function. In 3D, a divergence-free vector field is volume preserving, but does not necessarily have area-conserving Poincaré maps. Nevertheless, the Poincaré map is at least flux-conserving, which is the reason for the above mentioned Shilnikov chaos to occur.

## 6.3 Analytical Vortex Ring Model

For testing our algorithms, we developed a simple analytic vortex ring model based on Hill's spherical vortex (see, e.g., [123]). An analytical vector field has the



**Figure 6.6:** Hill's spherical vortex (axial slice).  $C_0, C_1$ : critical points (spiral saddles),  $P$ : periodic orbit.

advantage that artifacts due to discretization and interpolation can be excluded. A second motivation was to demonstrate that a rich topology (e.g., Figure 6.9 (a)) is possible even if the vector field has only two critical points and can be expressed with only quadratic terms (Eq. 6.3).

An instance of Hill's spherical vortex can be described by the two velocity fields

$$\mathbf{u}^i(x, y, z) = \begin{pmatrix} xz \\ yz \\ z^2 + 1 - 2r^2 \end{pmatrix} \quad (6.1)$$

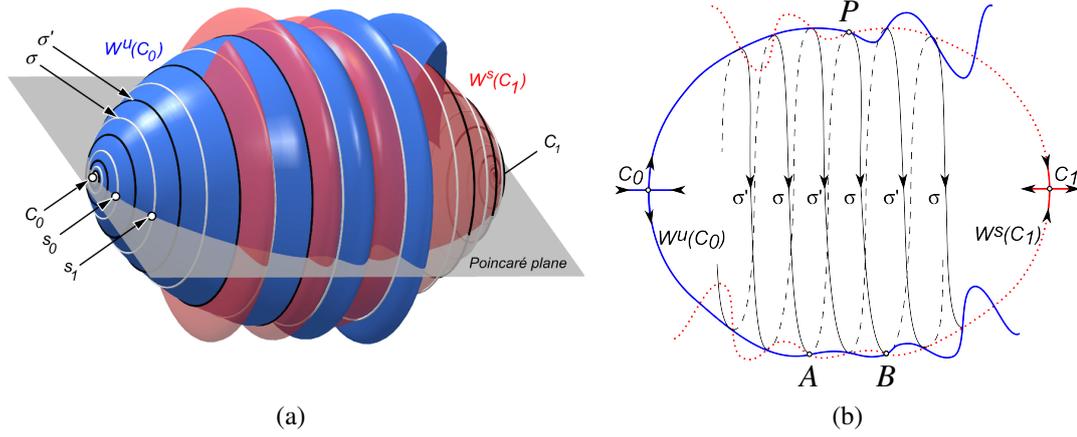
for points inside the unit sphere  $r = \sqrt{x^2 + y^2 + z^2} \leq 1$  and

$$\mathbf{u}^o(x, y, z) = \begin{pmatrix} xzr^{-5} \\ yzr^{-5} \\ z^2r^{-5} - \frac{1}{3}r^{-3} - \frac{2}{3} \end{pmatrix} \quad (6.2)$$

for points outside it ( $r \geq 1$ ). Figure 6.6 shows a sketch of this field. It is divergence-free and it solves the Navier-Stokes equations together with a matching pressure field. Furthermore, the field has zero vorticity outside the unit sphere.

By adding a swirl  $(\omega y, -\omega x, 0)$  around the principal axis, a rotating vortex ring model is obtained. This simple model does not solve the Navier-Stokes equations but is capable of generating the topological phenomena that can be observed in vortex rings. Physically correct variants of Hill's vortex with swirl exist, but they are more expensive to compute since Bessel functions have to be evaluated [123]. A different kind of generalization of Hill's spherical vortex are the Norbury vortex rings [101] where the vorticity is confined to toroidal regions instead of the sphere.

In order to obtain the chaotic behavior of a real vortex ring, the symmetry must be broken. In our model we do this by tilting the x-axis, which is motivated by experimental studies of vortex rings (see [153]). By substituting  $z' = z + \epsilon x$  for  $z$



**Figure 6.7:** Unstable manifold (blue) and stable manifold (red) of spiral saddles  $C_0$  and  $C_1$ , respectively, intersecting in a pair of saddle connectors  $\sigma$  and  $\sigma'$ . (a) In a Poincaré plane through  $C_0$  and  $C_1$ , the seed curve for integration is chosen between two successive intersection points  $s_0$  and  $s_1$  of  $\sigma$ . (b) The boundary surface is obtained by joining the manifolds at  $P$ , resulting in a turnstile between  $A$  and  $B$ .

and  $w' = w + \varepsilon u$  for  $w$  in Eq. 6.1, and by adding the swirl, we get the velocity fields

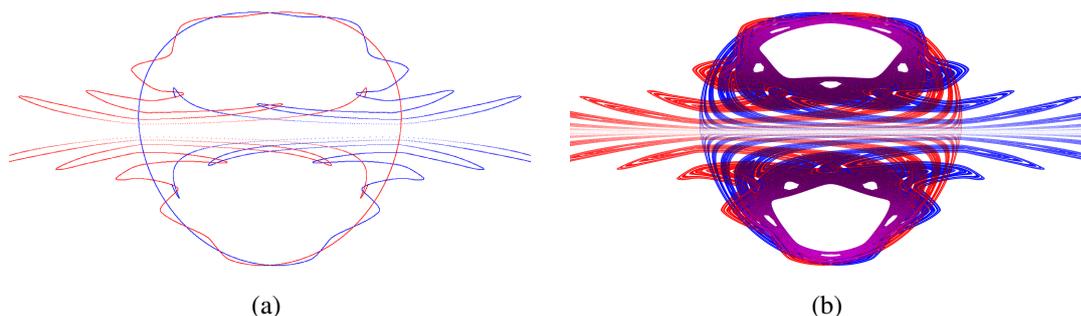
$$\mathbf{u}_{\varepsilon\omega}^i(x, y, z) = \begin{pmatrix} \omega y \\ -\omega x \\ 0 \end{pmatrix} + \begin{pmatrix} xz' \\ yz' \\ zz' + 1 - 2r'^2 \end{pmatrix} \quad (6.3)$$

for points inside the distorted unit sphere  $r' = \sqrt{x^2 + y^2 + z'^2} \leq 1$  and

$$\mathbf{u}_{\varepsilon\omega}^o(x, y, z) = \begin{pmatrix} \omega y \\ -\omega x \\ 0 \end{pmatrix} + \begin{pmatrix} xz' r'^{-5} \\ yz' r'^{-5} \\ zz' r'^{-5} - \frac{1}{3} r'^{-3} - \frac{2}{3} \end{pmatrix} \quad (6.4)$$

for points outside of it. This modified field is still divergence-free. It can be shown that the only critical points are two spiral saddles at  $(0, 0, -1)$  and  $(0, 0, 1)$ .

Figure 6.9 (a) shows a  $x = 0$  slice of the unstable manifold of the critical point at  $(0, 0, 1)$ , computed by seeding 200000 streamlines near the critical point and allowing for a maximum of 200000 intersections with the plane. The coloring of intersection points represents time, expressed in number of intersections with the plane. A rainbow color map is used, starting with violet and ending with red for intersection number 1000 and above. The system of three ODEs was solved with the 4<sup>th</sup> order Runge-Kutta-Fehlberg routine from the Netlib library. If an even simpler model is needed, it is also possible to use just the inner part  $\mathbf{u}_{\varepsilon\omega}^i$  for the entire domain, see Figure 6.9 (b).



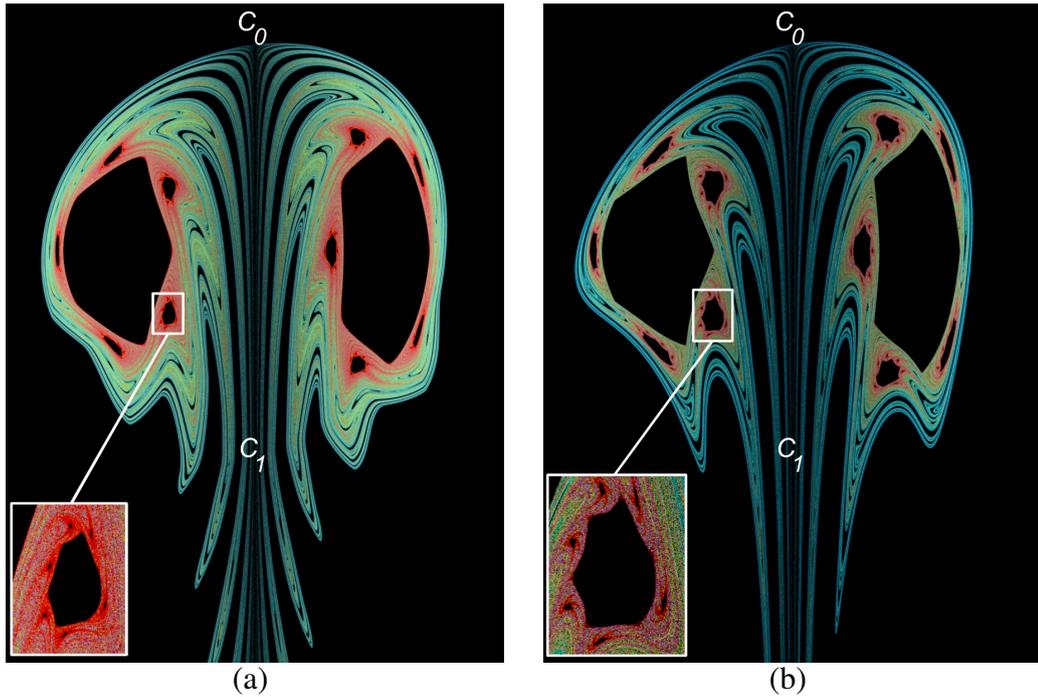
**Figure 6.8:** (a) Planar section of 2D manifolds of a synthetic vortex breakdown bubble. (b) Manifolds at longer integration time.

## 6.4 Detection of Vortex Rings

Our strategy for finding vortex rings is to look for pairs of critical points, one of them being a 1-spiral saddle and one a 2-spiral saddle. The number of such candidate pairs can be reduced by imposing a maximum distance. Additionally, it can be demanded that they lie on a common vortex core line. Another approach would be to extract the set of saddle connectors which gives the correct pairs directly. Each candidate pair of critical points is then checked for defining a vortex ring. For this purpose, a Poincaré section is taken by using a plane passing through the two points and oriented around this axis by fitting it to the real eigenvector directions of the two spiral saddles. The 2D (un-)stable manifold of the spiral saddle is now computed based on a discrete set of seed points and the intersections of the manifold with the Poincaré section are stored as a (texture) image. The extent of the grid is chosen based on the distance  $d$  between the two critical points. We found a square with edge length  $2d$  to be sufficient in most cases.

Seed points for the manifold of, say,  $C_0$  are generated as follows (see Figure 6.7 (a)). A first seed  $s_0$  is chosen at a small offset from  $C_0$  on the Poincaré plane where it intersects the plane spanned by the two complex eigenvectors. From  $s_0$  a streamline is integrated in the time direction where the distance from  $C_0$  increases. Its next iterate (i.e., intersection with the Poincaré plane) is denoted by  $s_1$ . Further seed points are now generated on the straight line segment between  $s_0$  and  $s_1$  by logarithmically interpolating the distance of the seed points to  $C_0$ . Logarithmic interpolation is appropriate because streamlines of the linearized vector field at  $C_0$  are logarithmic spirals, and the error introduced by interpolating along a straight line falls off with the streamlines converging to the 2D manifold.

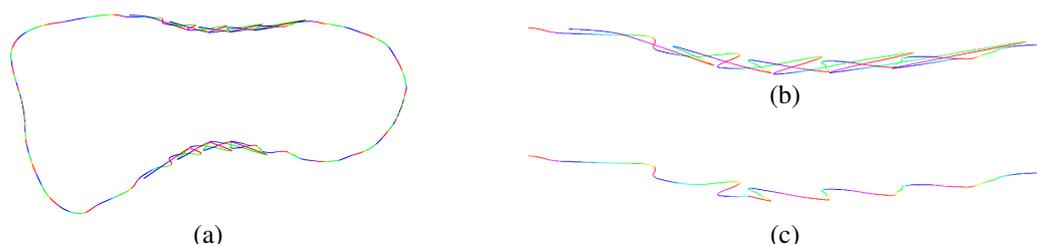
The resulting image (Figure 6.8 (a) is an example) then shows a section of the manifold with its folds, also called lobes, extending toward the second critical point. In the image plane, it is easily possible to check for an intersection of the two curves. If an intersection is detected, the vortex ring is verified.



**Figure 6.9:** (a) Hill's spherical vortex with swirl ( $\omega = 2\pi$ ) and tilt ( $\varepsilon = 0.313$ ). Slice of the stable manifold of the critical point at  $(0,0,1)$ . (b) Inner part  $\mathbf{u}_{\varepsilon\omega}^i$  of Hill's spherical vortex with swirl ( $\omega = 2\pi$ ) and tilt ( $\varepsilon = 0.442$ ).

One way of visualizing the vortex ring is to increase the integration time until the inner structure becomes visible on the Poincaré section, as is depicted in Figure 6.8 (b) and Figure 6.14. The resulting image shows the chaotic region formed by the inward folding lobes, and it typically shows a hierarchy of islands of stability. The islands of stability are toroidal regions around a periodic orbit of minimal period. The inner part of stability islands is typically filled with nested invariant tori with no flux across them (stream surfaces, known as KAM tori in the case of Hamiltonian systems). Farther out, chains of secondary islands can often be seen. These can possibly be separated from the primary island by first regions of chaos. When the chaotic region is reached, so-called cantori [90] can appear. These are porous tori of measure zero, which in some cases (if rotation number is a “noble” irrational number) exhibit negligible cross-flux, and hence act as partial barriers.

Islands of stability can be visualized by showing their outer boundary and/or their center periodic orbit. The periodic orbit can be located as usual using the method by Wischgoll and Scheuermann [179] or by searching for fixed points in the Poincaré map. A faster algorithm exploiting the given structure of nested tori in islands of stability is presented in Section 6.6 where also a method for computing the approximate boundary of the islands is described. These points and curves in the Poincaré section can further be used as integration seeds, giving a 3D visualization of the periodic orbits and invariant tori, respectively.



**Figure 6.10:** (a) Section of the manifolds of the vortex ring in the Francis draft tube. (b) Close-up. (c) Manifolds after trimming at selected intersection point, minimizing the maximal size of the lobes.

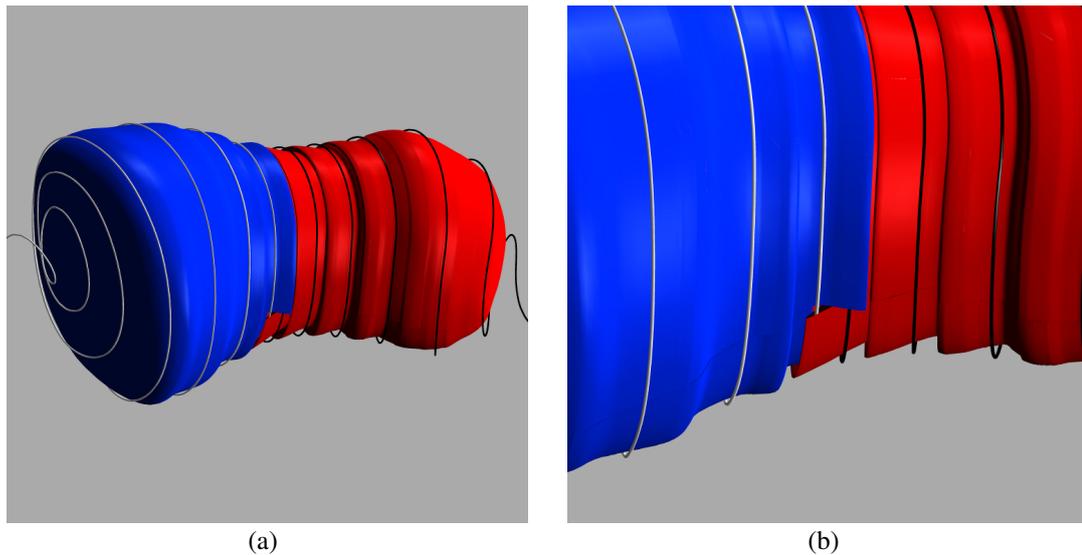
## 6.5 Boundary Surface

A practically useful result for pairs of spiral saddles is that the recirculation zone defined by such a pair can be visualized by a boundary surface constructed from the 2D (un-)stable manifolds. The boundary is made up of two stream surfaces and can be closed by adding a so-called *turnstile* [90] which is usually small and which is the only place where flux through the surface can occur.

In Figure 6.7 (b) it is shown how a boundary of a (perturbed) vortex ring is obtained. First, the intersection of the 2D manifolds  $W^u(C_0)$  and  $W^s(C_1)$  with a Poincaré section are computed (shown as red and blue curves). Then an intersection point  $P$  is found at which the two curves are trimmed and joined into one.

The criterion for the choice of  $P$  is to minimize the total arc length of the curve  $C_0 - P - C_1$ . For an approximate solution we use the following greedy algorithm: starting with the curve segment between the seeds  $s_0$  and  $s_1$  (Figure 6.7 (a)), compute successive iterates of the segment and add it to the curve. The obtained segments have the same frequency as the lobes (corresponding to a cycle of hues in Figure 6.10 (a)), with a possible phase shift. This process of extending a curve is continued in an interleaved way with both curves, such that the arc lengths of the two next segments are compared and the shorter is used.

Having now a seed curve, a stream surface is integrated backward and forward until the Poincaré section is intersected again. Under these “half turn” ( $\pi$ -period) Poincaré maps the curve  $C_0 - P - C_1$  is mapped to  $C_0 - A - C_1$  and  $C_0 - B - C_1$ , respectively. The two curves coincide with the exception of their parts between  $A$  and  $B$ . Therefore the stream surface is closed, up to the figure eight shaped opening lying in the Poincaré section plane between the points  $A$  and  $B$ . This opening, the turnstile, consists of an inflow and an outflow having equal absolute flux if the field is divergence-free. The influx (or outflux) can be easily computed and it tells what fraction of the enclosed fluid is exchanged per time unit. In practical cases, the turnstile is often very small. Figure 6.10 (a) and Figure 6.10 (b) show the two manifolds for the application example of Section 6.8.1 and Figure 6.10 (c) shows the resulting seed curve. When starting from here, stream surface integration yields the boundary of the vortex ring shown in Figure 6.11.



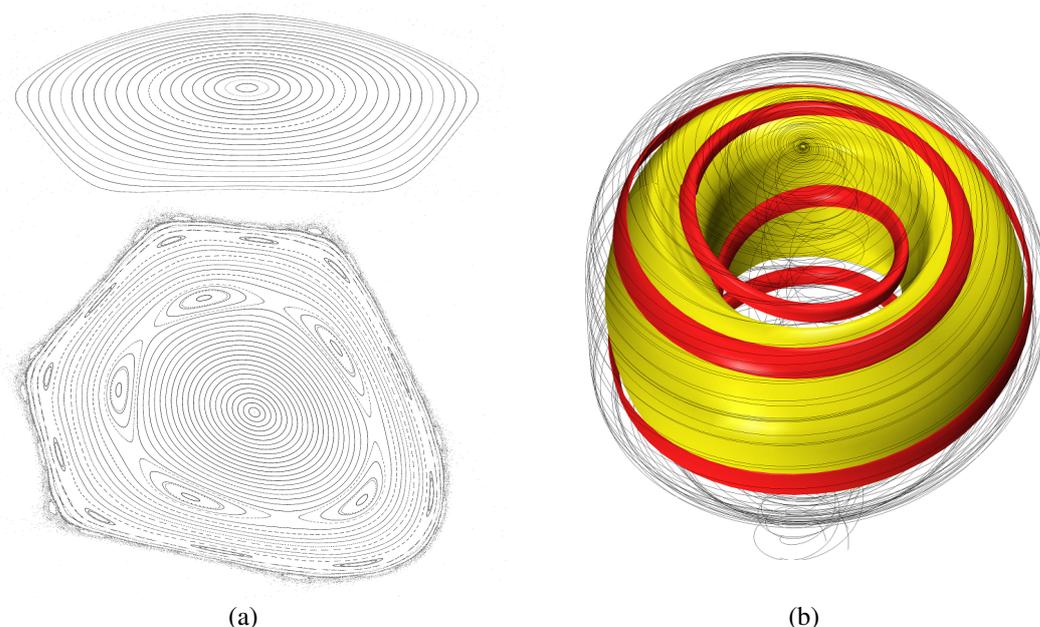
**Figure 6.11:** (a) Unstable manifold (blue) and stable manifold (red) rendered as stream surfaces. Streamlines entering (white) and leaving (black) the recirculation region. (b) Close-up.

## 6.6 Islands of Stability

From the previous step the Poincaré sections of  $W^u(C_0)$  and  $W^s(C_1)$  are now given as scalar fields on a regular 2-dimensional grid where the data values store the integration time for cells that were intersected. The goal is now to segment in the overlay of the two images the islands of stability. First, to clean the boundaries, a morphological closure operation is performed. This is followed by a component labeling step. Any component that does not extend to the image boundary is now checked for being an island of stability. A problem here is to distinguish islands of stability from holes that are formed by inward folding lobes. It can be observed that the latter are reached after much shorter integration time, hence when the average data value on their boundary is computed, this value is small compared to that of stability islands (see Figures 6.9 (a), 6.9 (b), 6.14, 6.16 (b)).

The obtained candidates for islands of stability are now processed in order of decreasing size. First, a streamline is seeded at the center of the island's bounding rectangle and whenever the Poincaré plane is intersected, the labeled component of the intersection point is marked as being part of the same island. If the streamline intersects the Poincaré section at a point outside of a component with a valid label, the test has failed.

Given now an island of stability, we want to visualize its internal structure which is a periodic orbit surrounded by nested invariant tori, with possible island chains interspersed in the outer part. For the Hill's vortex example, the primary and secondary islands are shown in Figure 6.12 (a).



**Figure 6.12:** (a) Internal structure of primary and secondary island of Figure 6.9 (a). (b) Primary (yellow) and secondary (red) islands rendered as stream surfaces.

We will visualize as two characteristic features the periodic orbit in the center and the outermost torus. The streamline seeded at the center of the island's bounding rectangle is integrated for a few "rounds" (detectable by increasing/decreasing  $x$  and  $y$  coordinates in the Poincaré plane). This should produce a set of points lying densely on a closed curve, otherwise it has to be retried from a slightly offset seed point. If a closed curve is obtained, the center of its bounding rectangle can be used for the next iteration of the process until a fixed point is found.

This algorithm exploits the special structure of nested tori and is significantly faster than the general approach of looking for fixed points of the Poincaré map, especially since in the case of secondary islands no fixed points are found and successive powers of the Poincaré map must be computed and searched for fixed points, too.

For finding the boundary of the island of stability, an iterative search is started with a seed curve consisting of the outermost zero pixels. At pixels which are mapped to a pixel outside the boundary, the seed curve is corrected inward by a pixel. This is repeated until all pixels of the seed curve are mapped to pixels inside the island. Finally, on these pixels the map is iterated a few times in order to reach a fixed curve. The obtained curve can be used as a seed curve for a simplified stream surface algorithm which requires only integration until the same component of the Poincaré plane is intersected again. Figure 6.12 (b) shows a pair of stream surfaces obtained this way.

## 6.7 Divergence in Vortex Rings

From a topological aspect, the divergence-free case is particularly interesting, as already mentioned in the introduction. However, when discrete data are used, divergence can be artificially introduced by the interpolation method. Furthermore, the data may exhibit divergence due to time-dependency in incompressible simulations or non-zero residuals during simulation (Section 6.2). And, finally, they may be divergence-free only with respect to the dual grid, as is the case for finite volumes solutions. We mention now two techniques that can be applied to cope with these issues.

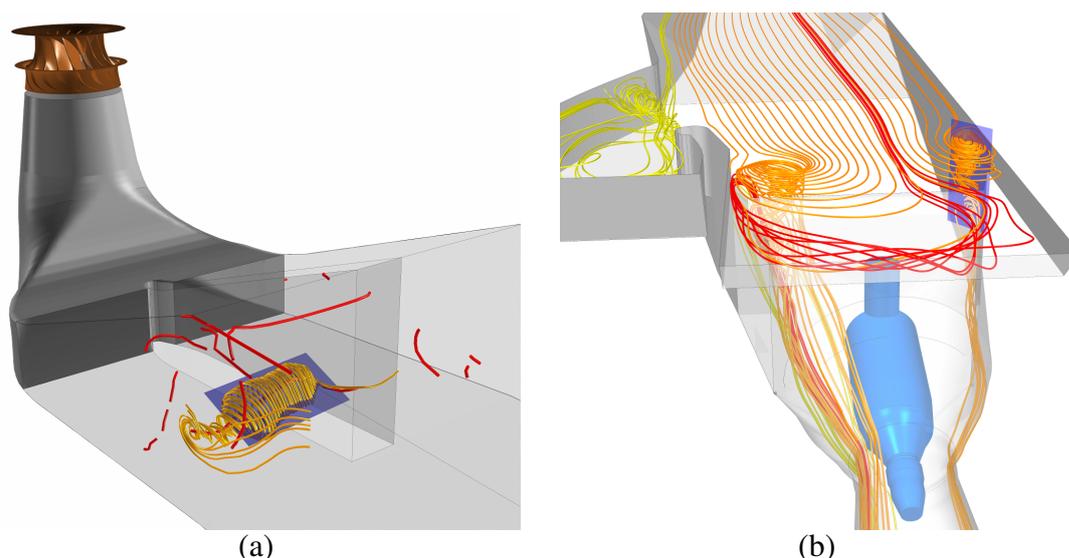
One approach is to modify the data by so-called divergence cleaning. For a divergence-free field, the total flux through the boundary of a volume element is zero by Gauss' theorem. However, if the dataset is a finite volumes solution, this holds for the control volumes, but not necessarily for the grid cells if standard multilinear interpolation is assumed. The control volumes are in general more complex polyhedra than hexahedra or tetrahedra, so it is usually not an option to use the control volumes as the visualization grid. Instead, we keep the original grid, or resample it to a rectilinear (possibly nonuniform) grid if necessary.

Divergence cleaning methods [5] were developed mostly within the field of magnetohydrodynamics. One of them is the Hodge projection [14, 155] which removes divergence based on the decomposition  $\mathbf{u} = \mathbf{u}_0 + \nabla s$  of  $\mathbf{u}$  into a divergence-free part  $\mathbf{u}_0$  and an irrotational part  $\nabla s$ . It follows  $\nabla \cdot \mathbf{u} = \nabla \cdot \nabla s$  which is a Poisson equation for  $s$ . Effectively, the flux of each cell is distributed to its neighbor cells by making small changes to the vector data at the grid nodes. These changes can be minimized either in the absolute or relative sense.

Another approach is to perform divergence-conserving interpolation. We found [105] that Tóth and Roe's technique [156] can be modified to generate instead of a subdivision a tetrahedral decomposition with the property that the piecewise linear interpolant is exactly divergence conserving. However, we show only some results in this thesis and point the reader to that work for details.

## 6.8 Results

We now show some results for the proposed techniques. The underlying cases are two time-dependent CFD simulations both exhibiting vortex breakdown bubbles. The first one is in a simulation of the flow in the draft tube of a Francis turbine and was already used in Section 6.1 and Section 6.5. The second vortex breakdown bubble is present in a recirculation region of the free surface simulation in the intake of a river power plant. We already had a look at this simulation in Section 4.3.4 in the context of vortex lines and vorticity transport.

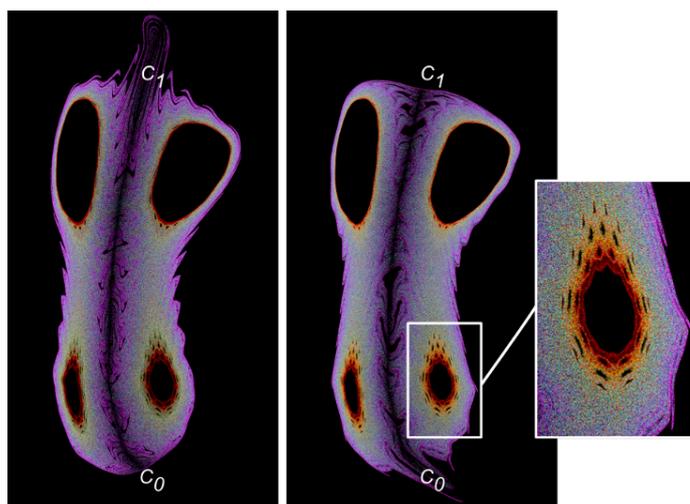


**Figure 6.13:** (a) Overview of the flow in the Francis draft tube with Poincaré section (blue rectangle), streamlines (yellow), and vortex core lines (red). Flow comes from the turbine at the upper left and is temporarily split on its way to the right. (b) Overview of the flow in the river power plant with Poincaré section (blue rectangle) and colored streamlines. The flow comes from the top and is guided to the turbine at the bottom.

## 6.8.1 Francis Draft Tube

Figure 6.13 (a) gives an overview of the flow under investigation. After passing the bend and the divider, the flow forms a large spanwise vortex in the right channel of the draft tube. The recirculation region under examination exhibits a temporally quite stable behavior. Its lifespan extends over 116 of total 301 time steps. At time step 90, a split event occurs where the shape transforms into a figure eight shape (in the section plane) and breaks apart, i.e., a double critical point is created which then separates into two single ones. At time step 94, the smaller of the two structures collapses, and at time step 115 the larger collapses, too. A vortex ring was identified in a selected time step in the phase according to the technique described in Section 6.4. The respective Poincaré section is visible in Figure 6.13 (a) and the resulting direct visualizations of the Poincaré map are shown in Figure 6.14.

We already have shown the boundary of the vortex ring, extracted according to the technique from Section 6.5 (Figure 6.10 and Figure 6.11). Finally, we compare the effect of divergence cleaning and of two choices of the interpolation function. For the numerical integration, the fourth-fifth-order Runge-Kutta-Fehlberg procedure from the Netlib library was used in all cases. In Figure 6.15 (a), the manifolds of the original data (white) and the divergence-cleaned data (colored) are shown. In both cases, standard trilinear interpolation was used. Figure 6.15 (b) shows the same manifolds, but computed using divergence conserving interpolation. It can

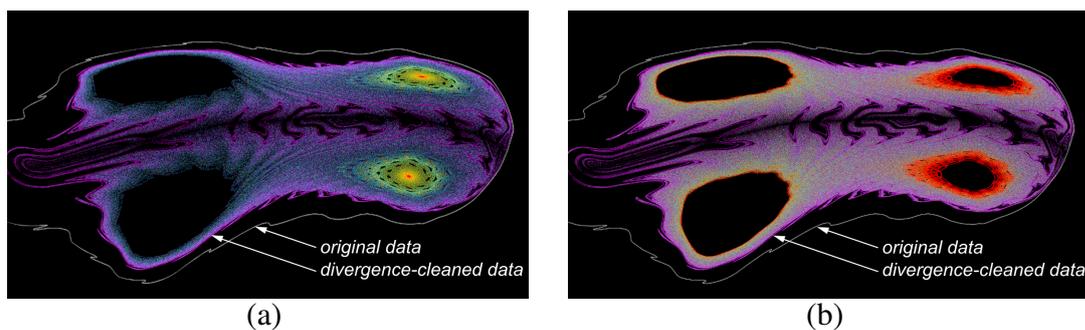


**Figure 6.14:** Stable (left) and unstable (right) manifolds of vortex ring in draft tube dataset exhibiting two primary islands of stability. The abrupt changes of colors near the islands of stability corresponds to jumps in integration time and indicates cantori.

be observed, that the choice of the interpolation scheme has little impact for short integration times. The two white curves are identical up to pixel resolution, and also the outer boundary of the colored manifolds coincide. However, the long-time behavior of the manifold is affected by the interpolation method. With trilinear interpolation, one of the two islands of stability is filled up with streamlines being attracted to a periodic orbit in its center. More important than the interpolation scheme is the use of divergence cleaning. Even though the CFD data represent an incompressible fluid (see also discussion in Section 6.2), without a divergence cleaning as a preprocessing, the vector field has strong enough attracting and repelling behavior to let most of the chaotic structure disappear (Section 6.1).

## 6.8.2 Intake of River Power Plant

The last example in this chapter is the flow in the intake of a river power plant (Figure 6.13 (b)). The flow exhibits two large recirculation zones, or vertical vortices reaching the free surface. We selected the left one for examination, and chose a Poincaré section in the vertical plane through the two critical points. The result is shown in Figure 6.16 (b). This example, however, exhibits a topologically more complex vortex ring. It extends to the (free slip) water surface where one of the two spiral saddles ( $C_1$  in Figure 6.16 (a)) is located. The unstable manifold of the latter coincides with the stable manifold of a periodic orbit  $P$  of saddle type which is also located at the water surface. In order to prevent streamlines from leaving the domain, the vertical velocity component at the water surface had to be set to exactly zero, i.e., residual  $z$ -velocities from the simulation had to be removed. Then, because  $W^u(C_1)$  converges to a periodic orbit, the criterion of intersecting

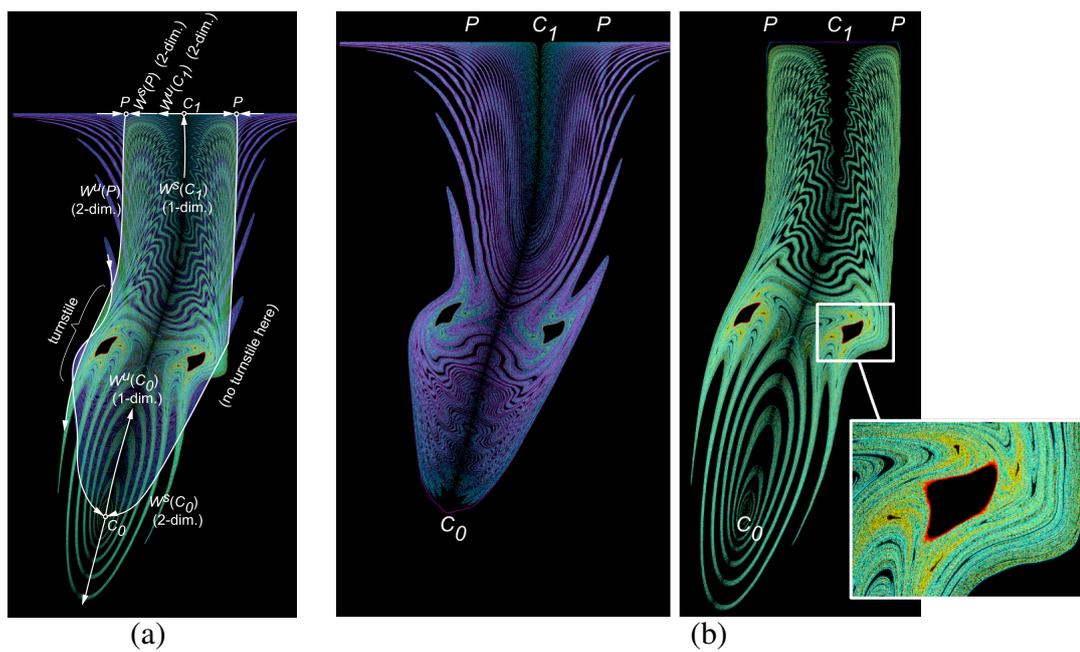


**Figure 6.15:** (a) Unstable manifold computed in trilinearly interpolated field. Comparison of original data (white) and divergence-cleaned data (colored). (b) Unstable manifold computed in divergence-conservingly interpolated field. Comparison of original data (white) and divergence-cleaned data (colored).

2D manifolds is not fulfilled, resulting in a failure of the automatic detection from Section 6.4. We circumvented this problem by giving the seed curve a small downward offset. For a systematic approach, one could in such a case proceed with the unstable manifold of the periodic orbit  $W^u(P)$ , which means, however, to examine two cases, since in general the unstable manifold extends in two directions. The apparent join of pairs of lobes below  $C_0$  is an artifact of the planar section. If a non-planar section following the curved center line of the structure was taken, a picture looking more like Figure 6.8 (b) would have resulted.

## 6.9 Discussion

We presented different methods for detecting and visualizing vortex rings and vortex breakdown bubbles in vector fields. The assumptions made are that the recirculation region is characterized by a pair of critical points of spiral saddle type, and that these are connected by saddle connectors which have a sufficient amount of spiraling. When applied to hydrodynamics CFD results, where such a configuration is likely to produce chaotic dynamics, this is revealed by our method. Once a vortex ring has been detected, a boundary surface is computed by combining parts of the 2D (un-)stable manifolds of the two critical points. The position of the turnstile on the boundary surface is chosen by minimizing the size of its lobes. We also presented methods for obtaining seed curves for stream surfaces that visualize nested invariant tori, i.e., islands of stability as well as a fast method for extracting periodic orbits located inside these tori. Since part of the underlying theory requires divergence-free vector fields, we also analyzed the effect of divergence cleaning and/or a new divergence-conserving interpolation scheme on vortex-rings in industrial CFD data.



**Figure 6.16:** (a) Both manifolds of vortex in river power plant. (b) Left: Stable manifold of spiral saddle  $C_0$  in river power plant dataset. Right: unstable manifold of periodic orbit  $P$ , approximated by seeding just below spiral saddle  $C_1$ , close-up on primary island of stability.



## Chapter 7

---

# Filtered AMR Ridge Extraction

There are many applications in science and industry where visualization by isosurfaces is not feasible, e.g., because the feature of interest is superimposed by a field that decays along its desired isosurface. Other examples include situations where structures are sampled by a scalar field, as in Section 6.1, where a streamline has been sampled inside a regular grid. We extracted the desired surface, i.e., the manifold of a critical point, as an isosurface using the marching cubes algorithm in this case. This way we circumvent the demanding explicit stream surface extraction, similar to van Wijk's implicit stream surface approach [166]. However, this way we obtained two surfaces enclosing the desired manifold and hence resulting in an inefficient and inaccurate representation. Many of these cases are better addressed by ridge extraction instead. In short, ridges are lower-dimensional (elongated) regions of relatively high values (see Section 2.6.1 for an introduction and our work on ridge extraction [107] for further details).

The extraction of 1-dimensional ridges in  $n$ -space is easily accomplished by the *parallel vectors* operator [103]. One of the advantages of this method is that explicit computation of eigenvectors is avoided and therefore the computational costs of the extraction are alleviated [107, 115]. However, the extraction of  $d$ -dimensional ridges with  $d > 1$  cannot be directly addressed by parallel vectors. These are cases where the *marching ridges* method [35] can be appropriate. Extracting ridges by integral curves of the *feature flow field* as done in [124] is an alternative approach but is a global operation and hence not pursued here.

We will present a method based on marching ridges that is applicable in situations where the underlying scalar field can be sampled during ridge extraction. Its strength shows up especially in cases where finely resolved ridges are desired, large regions do not exhibit ridges, or where the sampling of the scalar field is expensive. This is in particular the case when extracting *Lagrangian coherent structures* (LCS) from the *finite-time Lyapunov exponent* (FTLE) field and we developed this method in the context of this application (see Chapter 8).

## 7.1 Height Ridges

Height ridges can be seen as the most natural ridge concept in most applications and therefore it is widely used such as in the case of parallel vectors and marching ridges. This, together with its local definition, is also the cause why it has been chosen for the filtered AMR ridge extraction in this work and used in the context of Lagrangian coherent structures (Chapter 8). An inherent drawback of this definition is, however, that it cannot represent branching ridges [21], a situation sometimes encountered in FTLE fields. As a basis, we now detail Eberly's definition [28] of height ridges.

Height ridges can be seen as local maxima in a relaxed sense. They reside at locations where the scalar field  $s$  exhibits a maximum in at least one direction. Generally, height ridges are  $d$ -dimensional manifolds in  $n$ -dimensional space with  $n > d \geq 0$ . The constituent criteria can be formulated using the gradient and the Hessian of  $s$ . By definition, the eigenvectors  $\mathbf{e}_i$  corresponding to the  $d$  largest eigenvalues  $\lambda_i$  ( $i = 1, \dots, d$ ) of the Hessian point along the intended ridge whereas the eigenvectors  $\mathbf{e}_j$  corresponding to the  $(n - d)$  smallest eigenvalues  $\lambda_j$  ( $j = d + 1, \dots, n$ ) are perpendicular to it. According to the local maximum property, one necessary condition for a ridge is that the directional derivatives in  $\mathbf{e}_j$  directions are zero for  $j = d + 1, \dots, n$ , formulated as

$$\mathbf{e}_j \cdot \nabla s = 0. \quad (7.1)$$

The second condition for a local maximum and hence a height ridge is that the second directional derivatives in  $\mathbf{e}_j$  directions are negative for  $j = d + 1, \dots, n$ , formulated as

$$\lambda_j < 0. \quad (7.2)$$

The same concept can be used to compute *valley lines* which is the opposite of height ridges. They can be obtained by extracting height ridges of the field  $-s$ .

### 7.1.1 Ridges in Discrete Data

As already mentioned, 1-dimensional ridges are preferably extracted from discrete data using the parallel vectors method, whereas  $d$ -dimensional ridges with  $d > 1$  are preferably extracted using the marching ridges algorithm. Marching ridges is similar to the family of *marching cubes* [86] algorithms. Marching cubes is a cell-wise algorithm for generating isosurfaces, it generates a set of triangles for each cell of the grid. The so-called edge intersections are the positions on the edges of the cell where the scalar field has the desired value. The triangles are generated according to the edge intersections using a look-up table for signs of the values at the nodes. Criterion Eq. 7.1 is suited for being addressed by marching cubes. However, the fact that eigenvectors lack an orientation impedes a direct application, meaning that the evaluated directional derivatives can not be assumed to be

consistent. Marching ridges (and the presented algorithm) solves the problem by making the eigenvectors of a cell consistent using *principal component analysis* (PCA). Once the orientations are made consistent, criterion Eq. 7.2 can be applied. If an edge intersection violates it, we do not generate the corresponding triangles. Another issue is the orientation of the resulting triangles. Kindlmann et al. [70] orient the triangles in a post-processing pass. However, we experienced non-orientable manifolds in some cases. This problem is addressed by appropriate rendering techniques such as two-sided normals.

## 7.1.2 Ridge Filtering

Here some filtering criteria for ridges are recapitulated, and discussed in the context of adaptive ridge extraction, that we have presented in [119] for ridge extraction in uniformly sampled FTLE fields. Although we did not apply them here, we would also point the reader to additional filtering criteria we presented in [107].

Because ridges are extracted according to Eberly’s definition in this work, i.e., using the Hessian of the scalar field, noise amplification can become an issue due to estimation of higher derivatives. A common way to handle this, is to apply smoothing prior to the evaluation of derivatives. One has to keep in mind however, that smoothing can deform the ridges. In our case, the gradient at a given node (of the possibly unstructured sampling grid) is computed by fitting a linear field to its neighboring nodes in a *least squares* sense. This allows for incorporating the smoothing into gradient computation by simply increasing the neighborhood range by a user-defined integer value. A value of 2 was used for the results in Section 9.3, which increased perceptibility and only led to negligible deviations.

Even with smoothing, the marching ridges method from Section 7.1.1 often yields more ridge regions than desired. This can be addressed by feature filtering. One natural criterion for filtering ridge regions is to prescribe a minimum height of the ridge:

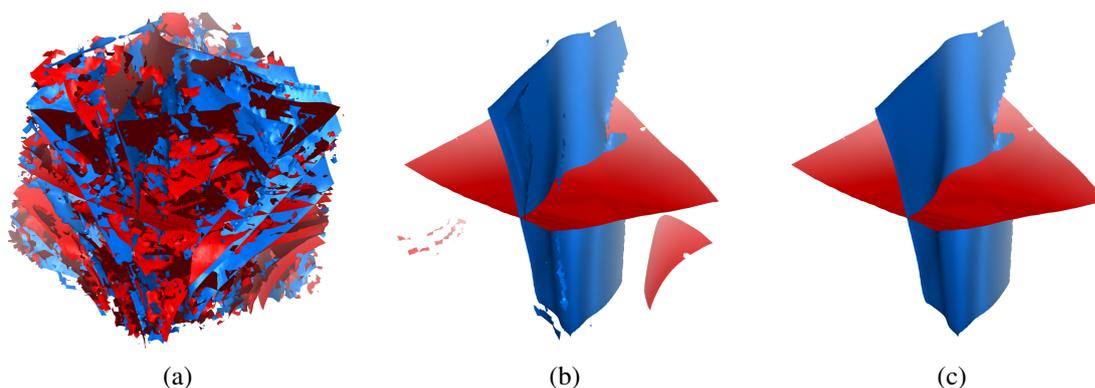
$$s \geq s_{min}. \quad (7.3)$$

In the case of finite-time Lyapunov exponent ridges, this equals to the prescription of a minimum separation and is therefore a straightforward choice (see Figure 7.1). This way, ridges with low separation property are suppressed, leading to significant, consistent, and reliable visualizations. It is therefore our preferred method for filtering finite-time Lyapunov exponent ridges. The reader is referred to [119] for further details on the influence of this filtering criterion.

Another natural criterion for filtering ridge regions is to prescribe a maximum for the second derivative  $\lambda_n$  across the ridge, which results in suppressing regions with too “flat” ridge property:

$$\lambda_n \leq \lambda_{max}. \quad (7.4)$$

In the case of finite-time Lyapunov exponent ridges, this is not as meaningful as criterion Eq. 7.3). Furthermore, it depends on the sampling of the scalar field,



**Figure 7.1:** Ridge filtering at 3D saddle in Francis draft tube. Positive-time FTLE height ridges (blue) and reverse FTLE height ridges (red). (a) No filtering. (b) Minimum FTLE 3.5 (positive-time) and 4.0 (negative-time). (c) Additionally to (b) suppressing components smaller than 1000 (positive-time) and 4000 triangles (reverse).

making it less suitable for guiding the subdivision of Section 7.2, but useful for post-processing, e.g., for further reducing ridges and hence occlusion.

Since ridge extraction often delivers small ridges which might be regarded as noise as long as the other filtering criteria did not disrupt the ridges due to low tolerance, another criterion is to prescribe a minimum size of the connected components of the resulting mesh. However, because this is a global criterion, it can not be used for the subdivision of Section 7.2, only for post-processing, mainly for reducing occlusion (Figure 7.1).

Finally, for the case of scalar fields derived from trajectories, such as Lyapunov exponents, another filtering criterion is to prescribe a minimum integration time for the trajectories that lead to a given ridge region. This allows to suppress ridge parts that are generated due to trajectories reaching the domain boundary. However, it was not needed for the results in Section 9.3.

The point-wise filtering conditions of this section are tested at the vertices of each generated triangle and it is rejected if at least one of the conditions is violated for at least one of its vertices. Triangle trimming was not implemented but would be a way to reduce zigzag ridge borders.

## 7.2 Filtered AMR Extraction of Ridges

Sampling a scalar field at a given resolution uniformly, generating ridge elements from that field, and rejecting many of these elements in the end by filtering wastes a lot of computation time and space, especially if there are large regions that would exhibit no ridges at all even if no filtering was applied. One would like to sample the scalar field only in regions that exhibit ridges in the end, especially if the sampling of the scalar field is as costly as in the case of Lyapunov exponent com-

putation. This would allow to end up with finer resolved ridges in the regions of interest at the same computational cost or even at lower cost. This constitutes the main motivation for incorporating adaptive mesh refinement (AMR) of the scalar field into the ridge extraction procedure.

The main goal of this method and a requirement for successful application in engineering is to obtain results that are identical to those obtained from uniform sampling at the prescribed finest subdivision resolution. This is especially valid for the finite-time Lyapunov exponent since it depends on the sampling resolution (section 2.3 of [48] and Section 7.2.2 of this work). Although complete ridges may get missed in the presented method (because of too coarse initial sampling, because Eq. 7.4 was used instead of Eq. 7.3, or due to reasons mentioned in Section 7.2.2), the obtained ones are identical to those from uniform sampling due to the “ridge growing” procedure starting at line 18 in Algorithm 1. This is the reason why no measurements of accuracy are presented although they have been performed extensively during testing, where ridges were rarely missed. Part of the problem is solved by the “look ahead” procedure presented in the next section. After the algorithm has terminated and the result is obtained, it is possible to continue with the look ahead as a background process to guarantee that all ridges are captured.

Assuming sufficiently fine initial grids, a ridge does not move substantially during refinement, therefore we can start with a relatively coarse mesh and use the ridges therein for the refinement process. The refinement is performed in an iterative manner. At each iteration, all cells that contain, are expected to contain, or adjoin to ridges are subdivided, but not more than to the current subdivision level. The current subdivision level starts with 1 and is increased with each iteration. This procedure makes sure that all cells in the neighborhood of ridges are subdivided to the same level. This has the advantage that the resulting grid has uniform subdivision level in ridge regions and therefore the marching ridges algorithm can be applied without the risk of producing cracks in the meshes, which would usually occur if a ridge extends over regions of different subdivision levels [31]. The next section describes the algorithm in more detail.

## 7.2.1 Algorithm

The following functions will be used in our ridge extraction algorithm, given in pseudo-code below as Algorithm 1:

- `detectRidgeCells(candidateCells, filter)`: Gradient, Hessian, and the eigenvalues and eigenvectors of the Hessian are computed at the nodes of the candidate cells. Then the intersected edges of `candidateCells` are determined in the sense of marching ridges, according to the height ridge criteria Eq. 7.1 and Eq. 7.2. If `filter` is *true*, the point-wise filtering criteria from Section 7.1.2 are also applied to the edge

intersections. A cell is a *ridge cell* and returned in the set of ridge cells if at least one of the edges of the cell remains intersected<sup>1</sup> after filtering.

- `getNeighboringCells(cells, range, level)`: This function returns the neighboring cells that lie at least partially within the node-connected<sup>2</sup> `range` around `cells`, whereas `range` is measured at subdivision `level`. Cells with lower subdivision level  $\sigma$  contribute a range of  $2^{(\text{level}-\sigma)}$ .
- `cellsForLevelDiff1(cells)`: This function returns cells that need to get subdivided additionally to `cells` so that neighboring cells differ at most in one subdivision level.
- `subdivideCells(cells)`: This function subdivides `cells`, adapts their subdivision level  $\sigma$ , and returns newly generated nodes and newly generated cells.

The explanations for Algorithm 1 are as follows:

i) **iterations** (Line 1): The total number of subdivision levels to be performed by the loop at line 12.

ii) **range** (Line 2): The neighborhood range around the ridge cells. Please refer to the discussions of (v) and (vi) below for further details.

iii) **laCellCnt** (Line 3): See look ahead step (viii) explained below.

iv) **laCriterion()** (Line 4): One of the point-wise filtering criteria of Section 7.1.2 has to be chosen for the look ahead procedure (viii). This method returns the value of the quantity tested by the corresponding criterion. If the ridge height criterion Eq. 7.3 is chosen, it returns the maximum of the scalar values at the nodes of the cell. If the second derivative criterion Eq. 7.4 is chosen, it returns the negative of the minimum  $\lambda_n$  at the edge intersections of the cell.

v) **Add ridge cell neighbors** (Line 15): This maintains a uniformly subdivided band of width `range` around the filtered ridge cells which is needed for gradient and Hessian computation that is not affected by AMR (neighboring cells with lower subdivision level). See also Figure 7.2 (a). For results that are identical to those based on a uniformly subdivided grid, `range` has to be set to 2 times the gradient fitting neighborhood range of Section 7.1.2. However, experience has shown that it can often be set to much smaller values, leading to negligible deviations. This allows for a more efficient but approximate ridge extraction because it reduces the number of performed field evaluations. If the scalar field itself requires gradient computation, the corresponding gradient fitting neighborhood range should also be added to `range` (see Section 7.2.2 for details).

<sup>1</sup> Defining a ridge cell by the presence of a generated triangle is not feasible because no triangles are generated if an edge intersection violates, e.g., Eq. 7.2 which is often the case at the fronts of the current ridges, and this would not allow the ridges to grow during the refinement process.

<sup>2</sup> It has been chosen to use node-connected neighborhoods instead of face-connected neighborhoods because it is a better approximation of the support radius of second derivatives (Hessian) even for a range of 1.

**Algorithm 1** Filtered AMR Ridge Extraction

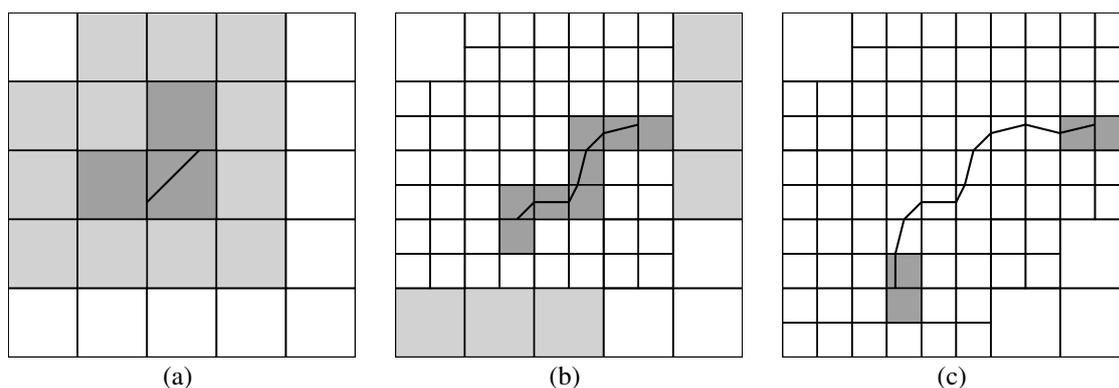
---

```

1: iterations: number of subdivision iterations to perform
2: range: user-defined integer neighborhood range
3: laCellCnt: number of cell-subdivisions to look ahead
4: laCriterion(·): filter criterion used for look ahead
5:
6: Initialization: A coarse sampling grid is supplied by the user and the scalar field is sampled on this grid.
7: for all cells c of sampling grid do
8:    $\sigma_c \leftarrow 0$  //  $\sigma_c$ : subdivision level of cell c
9: end for
10:  $\mathcal{R} \leftarrow \text{detectRidgeCells}(\text{allCells}, \text{true})$ 
11:
12: // iterate over subdivision levels
13: for it = 0 to iterations - 1 do
14:    $\mathcal{S} \leftarrow \mathcal{R}$ 
15:    $\mathcal{S} \leftarrow \mathcal{S} \cup \text{getNeighboringCells}(\mathcal{R}, \text{range}, \text{it} + 1)$ 
16:    $\mathcal{R} \leftarrow \emptyset$ 
17:
18:   // subdivide / let ridges grow
19:   subIter  $\leftarrow 0$ 
20:   while  $\mathcal{S} \neq \emptyset \vee \text{subIter} = 0$  do
21:      $\mathcal{S} \leftarrow \mathcal{S} \cup \text{cellsForLevelDiff1}(\mathcal{S})$ 
22:      $\mathcal{N}, \mathcal{C} \leftarrow \text{subdivideCells}(\mathcal{S})$ 
23:      $\mathcal{S} \leftarrow \emptyset$ 
24:     sample scalar field at new nodes  $\mathcal{N}$ 
25:
26:     // add lower-level neighbors of ridge cells
27:      $\mathcal{P} \leftarrow \text{detectRidgeCells}(\mathcal{C}, \text{true})$ 
28:      $\mathcal{T} \leftarrow \text{cells with } \sigma = \text{it} + 1$ 
29:      $\mathcal{Q} \leftarrow \text{detectRidgeCells}(\mathcal{T} \setminus \mathcal{C}, \text{true})$ 
30:      $\mathcal{M} \leftarrow \text{getNeighboringCells}(\mathcal{P} \cup \mathcal{Q}, \text{range}, \text{it} + 1)$ 
31:      $\mathcal{S} \leftarrow \mathcal{S} \cup \text{cells in } \mathcal{M} \text{ with } \sigma < \text{it} + 1$ 
32:      $\mathcal{R} \leftarrow \mathcal{R} \cup \text{cells in } (\mathcal{P} \cup \mathcal{Q}) \text{ with } \sigma = \text{it} + 1$ 
33:
34:     // test lower-level cells
35:      $\mathcal{T} \leftarrow \text{cells with } \sigma < \text{it} + 1$ 
36:      $\mathcal{S} \leftarrow \mathcal{S} \cup \text{detectRidgeCells}(\mathcal{T}, \text{true})$ 
37:
38:     // look ahead
39:     if  $\text{laCellCnt} > 0 \wedge (\mathcal{P} \neq \emptyset \vee \text{subIter} = 0)$  then
40:        $\mathcal{T} \leftarrow \text{cells with } \sigma < \text{it} + 1$ 
41:        $\mathcal{P} \leftarrow \text{detectRidgeCells}(\mathcal{T}, \text{false})$ 
42:        $R \leftarrow \text{sort cells in } \mathcal{P} \text{ by } \text{laCriterion}()$ 
43:        $\mathcal{S} \leftarrow \mathcal{S} \cup \text{laCellCnt highest cells in } R$ 
44:     end if
45:     subIter  $\leftarrow \text{subIter} + 1$ 
46:   end while
47: end for
48:
49: generate ridge triangles from  $\mathcal{R}$  according to Section 7.1

```

---



**Figure 7.2:** Refinement iteration. (a) 3 ridge cells (dark gray) and their 12 neighbors (light gray) scheduled for subdivision. (b) 8 ridge cells (dark gray) and their 6 lower-level neighbors scheduled for subdivision (light gray). (c) 4 new ridge cells. No new lower-level neighbors, iteration proceeds to next subdivision level (left figure). Ridge is not computed during refinement, only drawn for illustration of edge intersections. Neighborhood *range* is 2.

vi) **Add lower-level neighbors** (Line 26): Schedules lower subdivision-level neighbors of ridge cells for subdivision (see also Figure 7.2 (b)). This maintains the band of uniform subdivision level around the ridges as in (v) above and also allows the ridges to grow. The fact that not only neighbors of the new ridge cells  $\mathcal{P}$  are tested but also neighbors of already existing ridge cells at finest subdivision level  $\mathcal{Q}$ , accounts for the case where *range* has been chosen smaller than required for exact results. In this case subdivided cells may get ridge cells only after subdivision of nearby cells.

vii) **Test lower-level cells** (Line 34): This tests cells that may have become ridge cells because of subdivision of nearby cells.

viii) **Look ahead** (Line 38): We look ahead for the cases where ridge components would get missed completely because they are too small or too faint, or because the initial sampling grid was chosen too coarse.<sup>3</sup> This can happen if no cell of a ridge satisfies the point-wise filtering criteria of Section 7.1.2 at a low subdivision level, but would satisfy them after further subdivision(s). The parameter *laCellCnt* prescribes the number of unfiltered ridge cells with maximum value of *laCriterion()* that are subdivided (looked ahead) at each iteration even if they do not satisfy the filtering criteria. The parameter *laCellCnt* can be set to 0 if the user is only interested in the most prominent ridges.

<sup>3</sup>In theory a minimum of one trajectory per region of different behavior is required, although small regions are often detected during subdivision due to the subdivision band around detected ridges.

## 7.2.2 Implications for Fields Based on Local Operators

Some issues arise if filtered AMR ridge extraction is applied to scalar fields that can not be evaluated in a strictly point-wise manner. One class of such fields are finite-range Lyapunov exponents where the computation is based on gradients of a map (see Section 8.1.1). A natural approach to the computation of such fields is to sample the underlying field (the map) at a grid that is identical to that of the scalar field. In that case, there are several implications. One aspect is that the support radius of that gradient has to be added to the *range* parameter of Section 7.2.1 for obtaining exact results as in the discussion of  $(v)$  in Section 7.2.1.

Another issue is that the scalar value may depend on the sampling, because, e.g., gradients of non-linear fields vary with sampling resolution. It is therefore not possible to estimate the value at a finer resolution from the values at a lower resolution without any assumptions on the field. In the case of finite-time (or finite-size) Lyapunov exponents, no satisfying assumption can be made. The variation of the value during refinement has implications on the ridge height criterion Eq. 7.3. There are two approaches to handle the problem: either using a lower threshold and applying the desired threshold as post-processing, or increasing *laCellCnt*. In the worst case complete ridges may get missed, but thanks to the “ridge growing” procedure (line 18 in Algorithm 1), the obtained ridges will be identical to those from a uniform sampling at finest subdivision level.

A further issue is that there might be nodes in the grid that are invalid. In the case of the map, this can be because the positions are outside the domain. At such nodes no gradient is computed and furthermore they are not used for gradient computation at neighboring nodes. It might also happen that there are not enough neighbors for computing gradients due to the restriction of the node neighborhood. This case has also to be handled appropriately. If the gradient can not be computed at a node for the mentioned reasons, the node of the scalar field is marked accordingly. During application of the filtered AMR ridge extraction method, cells containing such nodes are rejected from the ridge extraction process because they can not be handled by the marching cubes look-up table.

## 7.3 Discussion

We presented filtered AMR ridge extraction as a means of efficient ridge extraction in situations where the underlying scalar field exhibits only few relevant ridges or can be sampled during ridge extraction. Although rarely some ridges may be missed, a drawback typical to many AMR approaches, the obtained ridges are identical to those from a uniform sampling at finest resolution level because a uniformly sampled band of cells is maintained in the neighborhood of the ridges. Chapter 8 presents results in the context of different Lyapunov exponent variants.



## Chapter 8

---

# Lagrangian Coherent Structures

The contribution presented in this chapter is twofold: after an introduction to the field of Lagrangian coherent structures (complementary to the informal introduction in Section 2.5.7), we present an approach for computing the finite-size Lyapunov exponent, and propose a new variant, the finite-time Lyapunov exponent maximum. Finally this section also presents examples of these concepts by means of the filtered AMR ridge extraction presented in Chapter 7.

The concept of Lagrangian coherent structures (LCS) is widely and increasingly used in fluid dynamics and in the analysis of the phase space of dynamical systems. There is no consensus what is meant by coherent structures, which is manifested in different definitions. Some are restricted to vorticity such as that by Hussain [58], others are more general such as the one by Robinson [114] where coherent motion is defined as “a region over which at least one fundamental flow variable exhibits significant correlation with itself or with another variable over a range of space and/or time that is significantly larger than the smallest local scales”. These definitions are referring to 3-dimensional regions. According to Haller [48], attracting and repelling Lagrangian coherent structures in (transient) vector fields are of lower dimension and tend to be the equivalent to unstable and stable manifolds (separatrices) in vector field topology [3, 52]: they separate regions of qualitatively different behavior and are also involved in mixing processes. Opposed to Newtonian coherent criteria such as the  $Q$ -criterion [57], the  $\Delta$ -criterion [18], and the  $\lambda_2$ -criterion [62], which are derived from the velocity gradient and therefore Galilean invariant, Lagrangian coherent structures are based on trajectories and even invariant under rotation of the frame of reference (they are objective). Because of their foundation on trajectories, LCS are insensitive to short-term perturbations or anomalies. In 2001 Haller has shown [42, 48, 49] that LCS can be obtained as ridges in the largest finite-time Lyapunov exponent (FTLE), also called largest direct Lyapunov exponent (DLE). FTLE was defined by Lorenz in 1965 [87] and, e.g., by Goldhirsch et al. in 1987 [41] for measur-

ing predictability, see Yoden et al. [184] for details. For steady vector fields LCS is comparable to vector field topology [48], although it tends to convey more information [119]. However, LCS is still well defined and interpretable for unsteady vector fields due to its Lagrangian definition, whereas classical vector field topology is only able to give an instantaneous view, except for the approaches by Theisel et al. [151] and Shi et al. [135] based on path lines. LCS move and deform over time as the starting time of their trajectories is modified. They behave approximately as material surfaces that get advected with the flow in a fluid dynamics view. LCS have played only a minor role in the field of scientific visualization until this work was presented. Two examples are direct visualization of 2-dimensional FTLE by Garth et al. [37] and the visualization of height ridges in 3-dimensional FTLE by us [119].

## 8.1 Lyapunov Exponent

The Lyapunov exponent (LE), also called Lyapunov characteristic exponent, measures the exponential growth of an infinitesimal perturbation. It is often used to analyze the predictability of continuous dynamical systems or their sensitivity to initial conditions. An  $n$ -dimensional system, or vector field, has  $n$  Lyapunov exponents and the largest Lyapunov exponent  $\sigma_1(\mathbf{x})$  measures the maximum possible divergence of two nearby trajectories, starting in the neighborhood of  $\mathbf{x}$ . If it is positive, the trajectory is part of the unpredictable (chaotic) regime of the system, otherwise it belongs to a predictable region.

The largest Lyapunov exponent at position  $\mathbf{x}$  and time  $t_0$  is defined as

$$\sigma_1(\mathbf{x}) = \lim_{T \rightarrow \infty} \lim_{\|\delta(t_0)\| \rightarrow 0} \frac{1}{|T|} \ln \frac{\|\delta(t_0 + T)\|}{\|\delta(t_0)\|} \quad (8.1)$$

where  $\delta(t)$  is the perturbation at time  $t$  (illustrated in Figure 2.15). The initial perturbation at  $\mathbf{x}$  and time  $t_0$  has to be oriented in direction of maximum expansion.

The subsequent sections describe finite-range LE variants with implications regarding filtered AMR ridge extraction, additional to those mentioned in Section 7.2.2. Section 8.1.1 describes the FTLE variant and its computation according to Haller [48]. In Section 8.1.2 the FSLE variant is described and a method for computing it is presented that builds on the one used for FTLE. Finally, Section 8.1.3 proposes a new variant called FTLE Maximum and describes its computation.

### 8.1.1 Finite-Time Lyapunov Exponent

Eq. 8.1 is an asymptotic measure in time and therefore a global quantity. However, there are several reasons for the need of a more local measure. One reason is that

many vector fields have domain boundaries and therefore do not allow for infinite advection time. Besides computability another reason is that the LE is constant along a trajectory even if the local expansion rate varies along it.

According to Nese [100] the local divergence rate at time  $t_i$  and for time step  $\Delta t$  can be expressed as

$$\frac{1}{|\Delta t|} \ln \frac{\|\boldsymbol{\delta}(t_i + \Delta t)\|}{\|\boldsymbol{\delta}(t_i)\|}, \quad (8.2)$$

and the time average of Eq. 8.2 is the largest Lyapunov exponent

$$\sigma_1(\mathbf{x}) = \lim_{\Delta t \rightarrow 0} \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{k=0}^n \frac{1}{|\Delta t|} \ln \frac{\|\boldsymbol{\delta}(t_0 + (k+1)\Delta t)\|}{\|\boldsymbol{\delta}(t_0 + k\Delta t)\|}, \quad (8.3)$$

provided that  $\boldsymbol{\delta}(t_0)$  is oriented in direction of maximum expansion. The local divergence rate Eq. 8.2 represents the largest finite-time Lyapunov exponent (FTLE) for  $i = 0$  and  $\Delta t = T$ . Nese was concerned with the information theory aspect and therefore used  $\log_2$  instead of  $\ln$ . He also mentioned that in practice, renormalization [13] has to be performed frequently along the trajectories. This addresses the fact that one has to make sure that the trajectories do not separate too much, otherwise they can not measure the expansion rates around either of the trajectories. In the case of renormalization, Eq. 8.3 follows only one of the two trajectories. This is numerically achieved by regular renormalization of the perturbation  $\boldsymbol{\delta}(t_i)$  to the length  $\|\boldsymbol{\delta}(t_0)\|$  but preserving its orientation. Kasten et al. [66] presented a related approach where the velocity gradient is evaluated along a single trajectory. Haller [48] addresses the renormalization issue by stating that only finite-time phenomena are measured and hence requiring a dense enough sampling grid solves the problem. However, there are cases where even arbitrarily fine sampling is not able to produce the same results as with renormalization, e.g., in a flow that splits without shear. On the other hand, the objective of this work is coherent structures and not predictability, and even coarse samplings are able to capture the large-scale behavior of a vector field.

Haller [48] proposed to base FTLE computation on the *flow map*. The flow map  $\boldsymbol{\phi}_{t_0}^{t_0+T}(\mathbf{x})$  maps a sample point  $\mathbf{x}$  to its advected position and is obtained by defining a sampling grid, seeding a particle at each node of the grid at time  $t_0$ , advecting the particles for time  $T$ , and storing the resulting positions at the nodes of the grid. We stop the integration of a trajectory if it reaches a domain boundary. One could determine the direction of maximum expansion as the eigenvector belonging to the largest eigenvalue of the right Cauchy-Green deformation tensor

$$\Delta_{t_0}^T(\mathbf{x}) = (\nabla \boldsymbol{\phi}_{t_0}^{t_0+T}(\mathbf{x}))^\top \cdot \nabla \boldsymbol{\phi}_{t_0}^{t_0+T}(\mathbf{x}).$$

According to Haller [48] the maximum stretching factor can be obtained as the *spectral norm*, i.e., matrix norm induced by the Euclidean norm, of  $\nabla \boldsymbol{\phi}_{t_0}^{t_0+T}(\mathbf{x})$ , defined as the square root of the largest eigenvalue of  $\Delta_{t_0}^T(\mathbf{x})$ :  $(\lambda_{\max}(\Delta_{t_0}^T(\mathbf{x})))^{1/2}$  and the largest *FTLE* is computed therefrom as follows:

$$\sigma_{t_0}^T(\mathbf{x}) = \frac{1}{|T|} \ln \sqrt{\lambda_{\max}(\Delta_{t_0}^T(\mathbf{x}))}. \quad (8.4)$$

The reader is referred to [48] for further information on LCS and FTLE.

Some additional issues arise if the filtered AMR ridge extraction technique from Section 7.2 is to be used for ridge extraction in finite-range LE variants of unsteady vector fields. Because the subdivision procedure is based on the values of these fields, the complete (incremental) procedure of their evaluation has to be performed for all nodes at a given subdivision (and ridge-growth) iteration before being able to proceed to the next level. One implication is that this makes a path line integration method necessary, that is even efficient for depth-first integration of path lines. Instead of loading complete time steps, we reorganize the vector field data in a preprocessing step. A single file is generated that stores for each node the vectors of its time steps consecutively. The access to the data is performed by *mmap()*: the file is mapped to an address space of equal size and the paging subsystem makes sure that for each memory access the corresponding memory page is loaded, with possible read-ahead. This exploits temporal coherence because a trajectory usually traverses multiple time steps before having passed a complete cell, and because several trajectories are likely to pass a given cell at different times when computing flow maps. If the file size exceeds the available address space in case of 32-bit systems, several files are generated instead, each storing only a fixed number of time steps. In this case, the high-level access routine used for integration has to make sure that the file containing the necessary time is mapped before accessing the data. This is easily accomplished and the overall performance does not noticeably degrade compared to the single-file approach because the corresponding file parts are cached by the soft disk cache of the operating system.

## 8.1.2 Finite-Size Lyapunov Exponent

Introduced by Aurell et al. [4], the *finite-size Lyapunov exponent* (FSLE) measures the shortest necessary time it takes for two infinitesimally close particles to separate by a given factor  $s$ . The motivation was to make the measure independent of the advection time  $T$  because different regions of a system often require different choices of  $T$ . Aurell et al. computed the FSLE by advection of differently oriented particle pairs.

Here a formulation of the FSLE  $\sigma_{t_0}^s(\mathbf{x})$  is presented that is based on the FTLE formulation of Haller Eq. 8.4:

$$\sigma_{t_0}^s(\mathbf{x}) = \frac{1}{|T_s|} \ln s,$$

with minimal  $|T_s|$  such that

$$\sqrt{\lambda_{\max}(\Delta_{t_0}^{T_s}(\mathbf{x}))} = s. \quad (8.5)$$

As with the formulation Eq. 8.4 for FTLE, this has the advantages that trajectories have to be integrated only for each node of the grid, and that the computed quantities do not substantially depend on the orientation of the seeding of the trajectories.

However, in practice there is often no simple analytic solution to determine the  $T_s$  that is necessary to achieve separation  $s$ , because  $\Delta_{t_0}^{T_s}(\mathbf{x})$  is based on trajectories computed from discrete fields. Therefore it was chosen to compute FSLE in an incremental manner. This is achieved by increasing  $T_s$  from 0 to a user-defined upper limit  $T_{max}$  by  $n$  time steps  $\Delta t = T_{max}/n$ , and each time computing the left-hand side of Eq. 8.5. The trajectories are computed incrementally. If the value gets larger than  $s$ , the corresponding advection time  $T_s$ , with linear interpolation in the last time step, is used for FSLE computation. The corresponding trajectories are not stopped however, because they are likely to be needed for the gradient computation at nearby trajectories later on.

Computation of the trajectories is the most expensive part in the computation of the left-hand side of Eq. 8.5. The computation of the flow map gradient, the eigenvalues, and the square root is much less expensive. Therefore one can afford a sufficiently high number of steps  $n$  and hence perform these operations  $n$  times.

Another issue shows up if filtered AMR ridge extraction is applied to finite-range LE variants that are computed incrementally such as FSLE and FTLEM (see section below). These variants are based on the incremental computation of the flow map and its gradient. As already mentioned at the end of Section 8.1.1, the procedure has to be repeated for each subdivision (and ridge-growth) iteration. For each step in the incremental computation, the flow map consists of intermediate positions of trajectories computed at previous subdivision levels, and the end points of the newly computed trajectories. Therefore all trajectories and not just the flow map have to be stored during filtered AMR ridge extraction.

### 8.1.3 Finite-Time Lyapunov Exponent Maximum

To further reduce the dependency on parameters, we propose a new variant of FTLE, the *finite-time Lyapunov exponent maximum* (FTLEM):

$$\hat{\sigma}_{t_0}^{T,n}(\mathbf{x}) = \max_{k=1,\dots,n} \frac{1}{|k\Delta t|} \ln \frac{\|\delta(t_0 + k\Delta t)\|}{\|\delta(t_0)\|} \quad (8.6)$$

with  $\Delta t = T/n$ , computed in an incremental way, similarly to the FSLE computation of Section 8.1.2. The FTLE  $\hat{\sigma}_{t_0}^{k\Delta t}(\mathbf{x})$  is computed at each of the  $n$  time steps according to Eq. 8.4 and its maximum is taken. The motivation for doing so is to avoid parameters that may require different choices for different regions of the vector field (which is to some extent still the case for FSLE), and to capture high expansions along the trajectory instead of only analyzing the final flow map. Taking the maximum of the largest FTLE is a quite straightforward decision: it still measures the maximum expansion. This FTLE variant comes in two flavors: with or without normalization. Equation Eq. 8.6 is the case with normalization and represents the true maximum of the FTLE along the trajectory. If the normalization  $1/|k\Delta t|$  is omitted, it measures the (logarithm of) maximum stretching factor along the trajectory, which avoids high values of Eq. 8.6 at the beginning (due to

small  $|k\Delta t|$ ) to dominate the result. However, since the logarithm is a monotonic function, ridges in the FTLEM without this normalization may differ from those in FTLE only if the constituting trajectories converge at some point.

## 8.1.4 Separation and Attachment Lines

Separation lines and attachment lines (Section 2.6.3) can be extracted locally as proposed by Kenwright et al. [68] but also globally using methods from vector field topology. However, both methods are only able to give an instantaneous view to flow separation and attachment processes. Furthermore, both are constricted to near-wall flow. One common way to inspect the interrelations with the interior flow is to seed stream surfaces from the separation lines or lines of attachment.

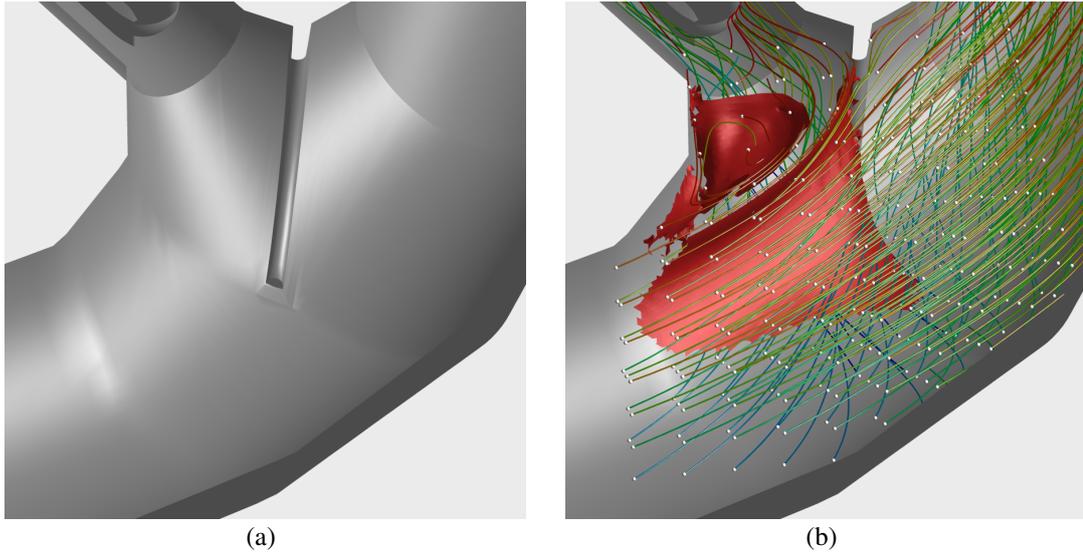
Due to their analogy to vector field topology, Lagrangian coherent structures based on ridges in finite-range Lyapunov exponent variants are suited to visualize both separation and attachment phenomena, and their interrelations with the internal flow in a time-dependent way. Attachment lines can be indicated by the curves where ridges of finite-range positive-time Lyapunov exponents attach to a boundary. Separation lines can be obtained the same way using finite-range negative-time Lyapunov exponents. If such a situation is observed, the ridges in question convey information about the process of attachment or separation.

## 8.2 Results

This section presents some results obtained by filtered AMR ridge extraction from finite-range Lyapunov exponent variants. Several datasets are examined and the different finite-range LE variants are compared. The steady CFD simulation inside the distributor ring of a Pelton water turbine already used in the first part of the thesis is examined and used for the comparison of the finite-range LE variants in Section 8.2.1. An unsteady CFD simulation of the intake of a hydropower plant is the target in Section 8.2.2, and the unsteady CFD simulation inside the diffuser of a Francis water turbine from Chapter 6 is examined in Section 8.2.3. We verified by also doing the non-adaptive ridge extraction that no filtered ridges were missed in the present examples, even without using the “look ahead” mechanism. The obtained ridges deviate only slightly from those of uniform sampling because a *range* parameter of 2 was used instead of 5 (see discussion of choice of *range* in Section 7.2).

### 8.2.1 Pelton Turbine

Filtered AMR ridge extraction was applied to the different finite-range LE variants of the steady flow inside the distributor ring of a Pelton water turbine. The

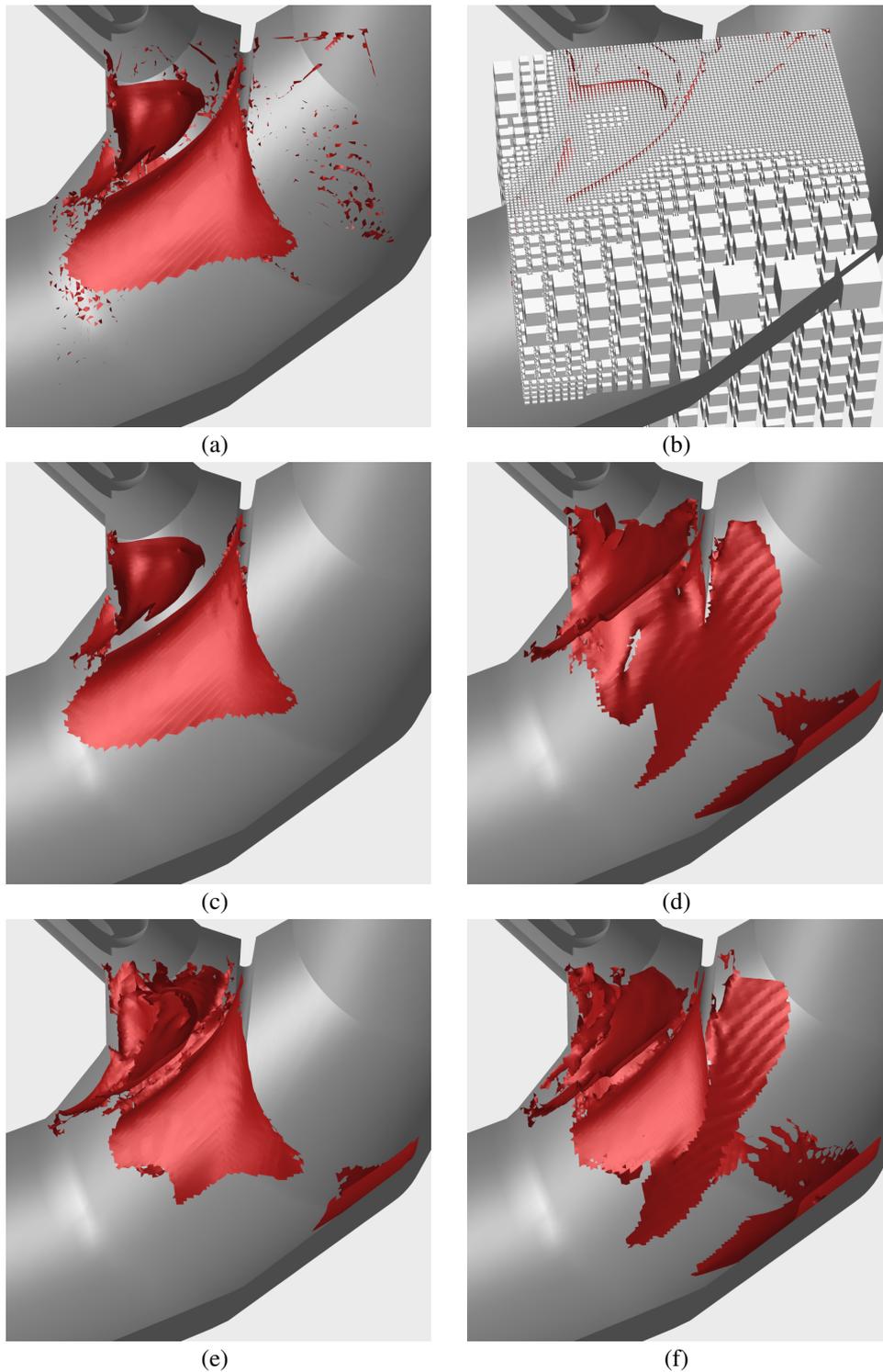


**Figure 8.1:** Pelton dataset at second injector of distributor ring. (a) Geometry. (b) Filtered AMR ridge extraction from FTLE (red, same as Figure 8.2 (c)) with some trajectories (colored lines). Seeding points of trajectories visualized by white spheres.

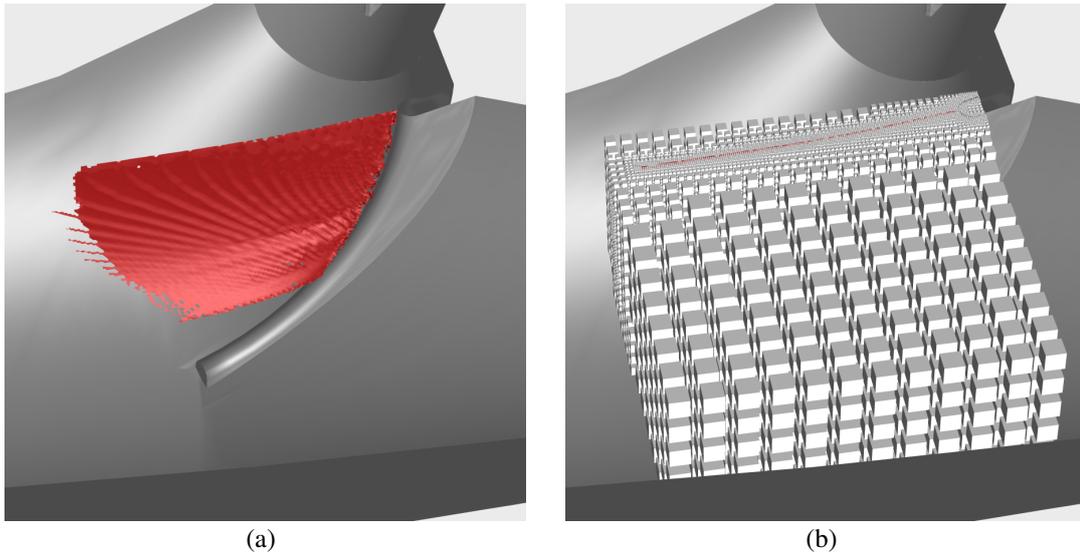
inspected region is in front of one of the constructs called sickle that bifurcate the flow into the injectors. The injectors produce the water jets that impel the runner of the turbine. Figure 8.1 (a) shows the geometry at the second injector and Figure 8.1 (b) shows the ridge resulting from the FTLE with some trajectories. The flow comes from the bottom left and continues to the upper right while part of it is bifurcated to the upper left. Figure 8.2 compares the results of the different positive-time finite-range LE variants and Table 8.1 gives some extraction details.

	FTLE	FSLE	FSLE	FTLEM
initial grid (nodes)	7x7x5	7x7x5	7x7x5	7x7x5
final grid (nodes)	157204	239551	215491	244651
integration time [s]	0.1	0.1	0.1	0.1
sep. factor $s$	–	1.5	4	–
$1/ T $	yes	yes	yes	yes
min. scalar $s_{min}$	23	50	14	22
gradient range	2	2	2	2
ridge range	2	2	2	2
iterations	4	4	4	4
Hess. $\lambda_{max}$	0	-150000	-30000	-50000
min. triangle cnt.	2000	2000	2000	2000
Figure	8.2 (c)	8.2 (d)	8.2 (e)	8.2 (f)

**Table 8.1:** Extraction details for Pelton example. The parameters below the second horizontal line were not used for mesh refinement, only for post-processing.



**Figure 8.2:** Pelton dataset at second injector. Filtered AMR ridge extraction from different finite-range LE variants. See Table 8.1 for extraction details. (a) FTLE ridge without post-processing. (b) same as (a) with adaptive mesh. (c) FTLE ridge with post-processing. (d) FSLE with separation factor 1.5. (e) FSLE with separation factor 4. (f) FTLEM (similar to (d) and (e)).

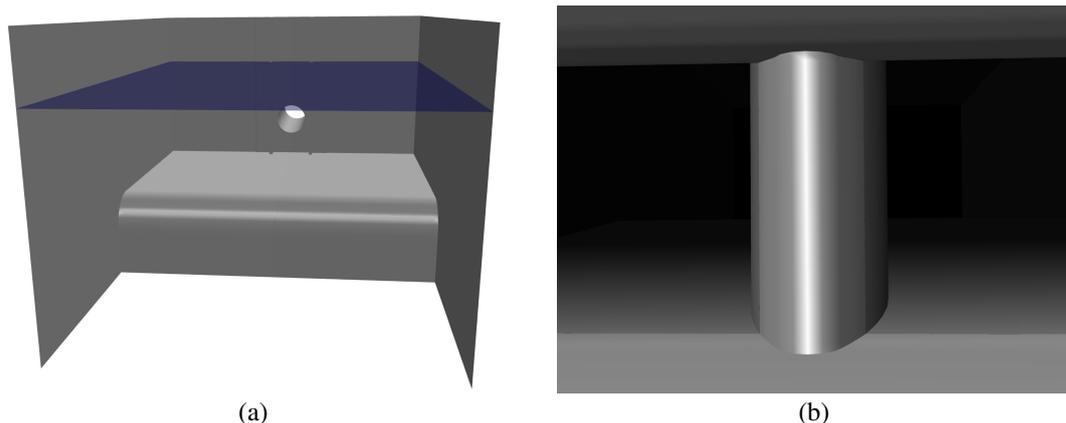


**Figure 8.3:** Pelton dataset at first injector, see Table 8.2 for performance details. (a) Filtered AMR ridge extraction for FTLE. The ripples are due to high FTLE values along the ridge and its approximate alignment with the sampling grid. This is an inherent problem with discretized FTLE computation. (b) Together with adaptive mesh.

Figure 8.2 (a) shows ridges of FTLE without post-processing and Figure 8.2 (b) shows additionally the corresponding mesh (the cells have been shrunk for visualization). The result has been post-processed by suppressing ridges that were smaller than 2000 triangles, visualized in Figure 8.2 (c). The FTLE ridges visualize the bifurcation at the sickle and the recirculation zone at the top (see also Figure 8.1 (b)). Regarding FSLE (Figure 8.2 (d)–(e)), one can see that the results depend on the choice of the prescribed separation factor (compare also Table 8.1). The ridges of the low-separation FSLE in Figure 8.2 (d) visualize the near-wall flow whereas the ridges of the high-separation FSLE in Figure 8.2 (e) mainly show the bifurcation at the sickle. The recirculation region is captured by both FSLE examples, but not as well as by the FTLE example of Figure 8.2 (c). Figure 8.2 (f)

	direct	adaptive
initial grid	193x193x97 (3613153 nodes)	13x13x7 (1183 nodes)
final grid	193x193x97 (3613153 nodes)	298964 nodes
flow map [s]	19953.51	2350.21
FTLE [s]	10.73	30.73
ridge extr. [s]	278.46	2337.16
total [s]	20242.74	4930.72

**Table 8.2:** Performance analysis for Pelton dataset at first injector. Four iterations of filtered AMR ridge extraction from FTLE compared to direct computation on corresponding uniform grid. Achieved speed-up factor is  $> 4$ . See also Figure 8.3.



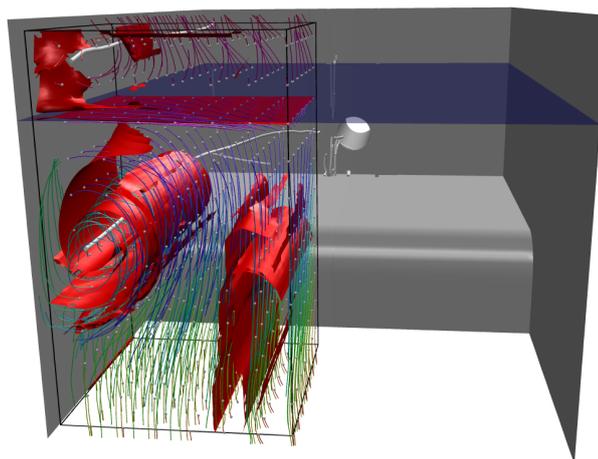
**Figure 8.4:** (a) Hydropower plant intake. Water enters the region from the bottom and flows to the intake of the power plant visible as hole in the back wall. Air enters the region at the top front and leaves the region at the top. Water/air interface by isosurface (transparent blue). (b) Draft tube of Francis turbine in front of divider. The flow is bifurcated at the divider into the two channels.

shows the advantage of FTLEM over FSLE: it is capable of visualizing both the phenomena of Figure 8.2 (d) and Figure 8.2 (e) without the need for finding appropriate FSLE separation factors.

The efficiency gain of filtered AMR ridge extraction over direct ridge extraction from a uniformly sampled field at corresponding resolution was only about 1.4 for the FTLE in the inspected region. This is because the result contains a lot of ridge regions in the region of interest. However, many applications do not exhibit that dense Lagrangian coherent structures. In order to show the efficiency gain, we examined a region in front of the first injector of the distributor ring (see Figure 8.3 and Table 8.2). There is only a single filtered ridge in this region and therefore the efficiency gain is  $> 4$  at the fourth subdivision level. The efficiency gain would further increase with an increase of the number of subdivision iterations or with longer advection times (this test was not performed because the FTLE computation on the uniform grid already took a considerable amount of time).

## 8.2.2 Hydropower Plant Intake

The underlying data of this section is an unsteady CFD simulation in the intake of a hydropower plant [144] (Figure 8.4 (a)). Filtered AMR ridge extraction was applied to negative-time FTLE in a region of interest. Ridge regions with FTLE  $< 0.02$  or Hessian  $\lambda_n > -10$  were suppressed and small ridge components were rejected, see Figure 8.5. One can see FTLE ridges that wind around the vortex core lines, both in water and in the air. Another FTLE ridge is consistent with the water/air interface.



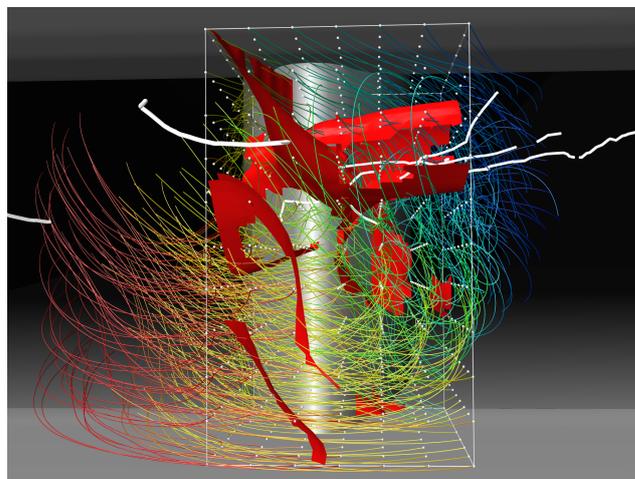
**Figure 8.5:** Hydropower plant intake. FTLE ridges (red) with some upstream trajectories (colored lines) and vortex core lines (white). Seeding points of trajectories visualized by spheres (white), and water/air interface by isosurface (transparent blue).

## 8.2.3 Francis Turbine

This section examines the flow of an unsteady CFD simulation in the draft tube of a Francis water turbine. More precisely, the region in front of a construct named divider, that bifurcates the flow into the two channels (Figure 8.4 (b)). Filtered AMR ridge extraction of positive-time FTLE resulted in several ridges, some of them interacting with the vortices over time. Ridge regions with  $FTLE < 10$  were rejected and only the large ridge components were used for visualization (Figure 8.6). One FTLE ridge winds around the vortex. This is consistent with the notion that vortices are coherent structures. The same ridge also visualizes the bifurcation at the top of the divider.

## 8.3 Discussion

Lagrangian coherent structures present as ridges in the FTLE are a powerful concept for the analysis of time-dependent vector fields. In this chapter we mainly exemplified the filtered AMR ridge extraction method from Chapter 7 in the context of the FTLE. Furthermore, we have shown how the FSLE can be efficiently computed from Haller's FTLE formulation and proposed a new FTLE variant, the FTLEM, further liberating the concept from parameters and making it more flexible with respect to domain regions exhibiting different behavior.



**Figure 8.6:** Filtered AMR ridge extraction from FTLE (red) in draft tube of Francis turbine, some trajectories (colored lines), their seeds (white spheres), and vortex core lines (white tubes). One of the ridges winds around vortex in front of the divider.

## Chapter 9

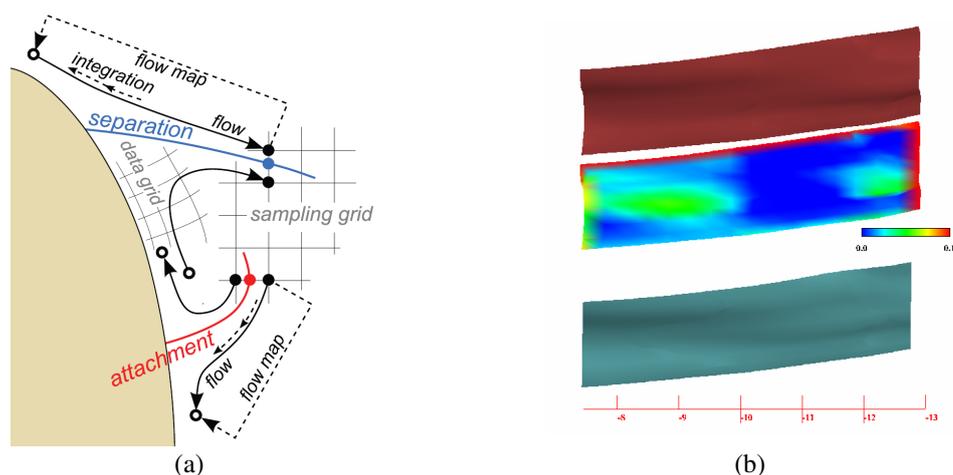
---

# Grid Advection for Efficient FTLE Computation

Lagrangian coherent structures feature two temporal degrees of freedom: the moment of seeding  $t_0$  and the advection time  $T$  (Section 8.1.1). Once the spatial region of interest and the time scope are chosen, it is often desired to vary the seeding time  $t_0$  to obtain a full picture of time-dependent flows. This way, one obtains time series, or animations, of FTLE. While we presented a method for accelerated computation of FTLE at a given seeding time  $t_0$  in Chapter 7, we present here an approach for accelerated computation of time series of FTLE. The main idea is to reuse part of trajectories by means of advection of the sampling grid. A focus of this work is the analysis of LCS related to predefined regions such as boundaries, in particular, flow attachment and separation.

The method maintains a sampling grid that grows and shrinks with the ridges that it contains and that is advected with the flow between the steps of the time series. This is consistent with the fact that ideal LCS, i.e., those derived from FTLE based on sufficiently long trajectories, are material surfaces [48] and hence advect with the flow. The grid is initialized by the user in a region of interest. By initializing the grid near a solid boundary, flow separation and attachment surfaces are obtained. An advantage of this visualization method is that it does not rely mainly on the data next to the boundary, and in particular does not need the computation of derivatives in cells adjacent to the boundary.

As detailed in Section 2.6.3, the topology of the flow on solid boundaries or of the wall shear stress can be used for the analysis of steady separation and attachment. By combining it with the topology of the velocity field in the interior, Surana et al. were able to give exact definitions of separation and attachment surfaces, and they showed that for Navier-Stokes flows, the separation slope and angle formulas depend on on-wall quantities only [147].



**Figure 9.1:** (a) Flow separation and flow attachment. The unstable manifold (blue) attracts the fluid along the boundary and guides it into the interior of the domain whereas the stable manifold (red) guides the fluid in opposite direction. (b) Intake dataset. Comparison of ridge from advected grid (red) and uniform grid (blue) together with ridge from advected grid color-coded by distance to ridge from uniform grid.

## 9.1 Motivation

The adaptation of the uniform sampling grid to the regions containing ridges is motivated by our approach from Chapter 7. The goal is to optimize the computation of time series of FTLE ridges to make the method more applicable in every-day applications in research and engineering. The increase in efficiency is achieved by restricting the computation to regions that contain the LCS of interest, and, in particular, by exploiting the temporal coherence of unsteady vector fields for the computation of time series of FTLE by advection of the sampling grid.

One of the application goals is to offer a method for the analysis of unsteady flow separation and attachment. Separation phenomena are the cause of many undesired effects in engineering, such as development of recirculation zones, reduced throughput, reduced lift, vortex generation, lowered mixing, and reduced flow control in general. Flow separation exhibits diverging trajectories in backward time and flow attachment exhibits diverging trajectories in forward time (see Figure 9.1 (a)). This fact is the reason why separation and attachment lines (or points) are usually accompanied by corresponding LCS and why these processes are amenable to an analysis by FTLE ridges. Shadden et al. [134] have already demonstrated the utility of those ridges for the analysis of unsteady flow separation in their example of flow separation over an airfoil. We also believe that an analysis based on LCS provides a deeper and more precise insight into these unsteady phenomena compared to standard techniques such as stream surface integration or particle tracing.

## 9.2 Method

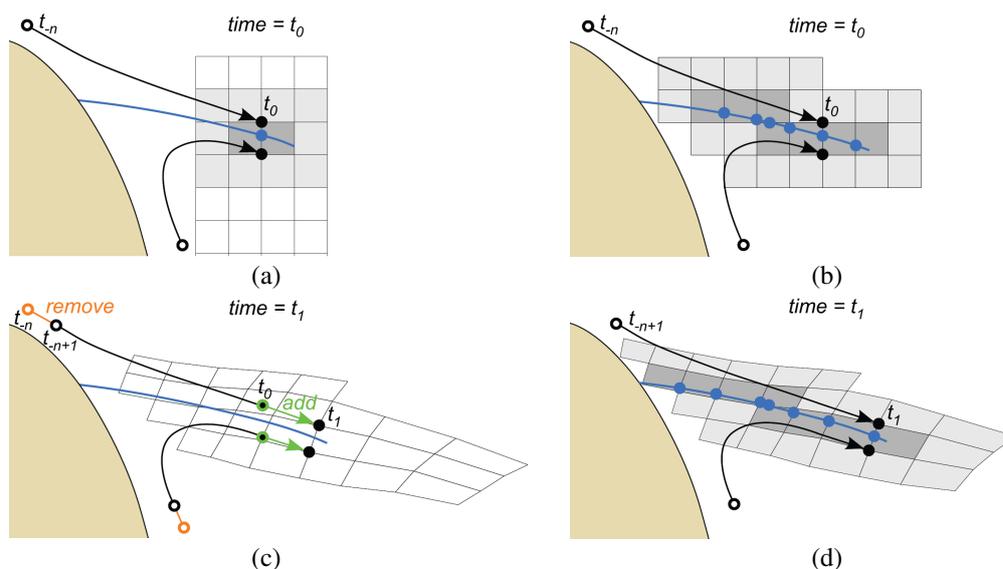
The proposed method can be subdivided into two parts: one that constrains the sampling grid to filtered ridges of interest at a given time step (or the only time step in case of steady vector fields), described in Section 9.2.1 and Section 9.2.2, and one that exploits temporal coherence to speed up the computation of time series of quantities computed from trajectories, such as the FTLE, in the case of unsteady vector fields (Section 9.2.3). The time series are obtained by variation of the  $t_0$  parameter of the FTLE.

Algorithm 2 describes the methods presented in Section 9.2.1 to Section 9.2.3 for the case of FTLE ridge extraction. However, it also can be applied for the extraction of ridges of other quantities based on trajectories and could be easily modified for the scalar quantity itself instead of its ridges. If the quantity is not computed using local operators such as gradients, larger distortions may be acceptable and hence longer advection times could be used, leading to a further increase in speedup.

### 9.2.1 Grid Initialization

In the filtered AMR ridge extraction method (Chapter 7) the complete domain (or region of interest) is sampled at low resolution and the sampling is adaptively refined in regions containing filtered ridges. Although this results in a speedup compared to a uniform sampling at the finest subdivision level, the method suffers from several drawbacks when applied to quantities that can not be evaluated in a point-wise manner but are computed using local operators, such as gradients in the case of FTLE ridges. The main problem here is that the value is inherently sampling dependent because the gradient can be underestimated if the sampling is coarse. Together with a restrictive threshold for the ridge filtering this can result in missed ridges in the adaptive approach, if they are not detected in the coarse sampling and hence the corresponding regions are not refined which would capture them later on. The remedy is either to use a finer initial sampling, a lower threshold for filtering, or to increase the look-ahead count (Section 7.2), all leading to an increased number of samples and hence lowered speedup. See also Section 7.2.2 for further information on the implications for quantities based on local operators.

In the present approach one typically avoids sampling the whole domain (or region of interest). Instead, we require initial sampling regions that already capture part of the ridges (cf. Figure 9.2 (a) and Figure 9.3 (a)) and adapt the sampling regions to the present ridges (Figure 9.2 (b) and Figure 9.3 (b)). This allows one to use initial samplings of sufficient resolution and avoids the need for lowered filtering thresholds. In the case of FTLE analysis of flow separation and attachment, possible ways for choosing the initial regions include:



**Figure 9.2:** Grid advection for flow separation. (a) Initial sampling grid. Ridge cells (dark gray) and their neighboring cells (light gray). Cell edge intersected by negative-time FTLE ridge (blue point). Neighborhood range is 1 for illustration purposes. (b) After grid adaptation. (c) After one step of grid advection. (d) After grid adaptation of advected grid.

- Definition from special regions of the simulation mesh, e.g., the complete boundary of the domain, or a subset thereof such as the blades of a turbine. These regions are often explicitly available from the simulation file formats.
- Automatic definition by quantities such as “surface divergence” or its local maxima as presented by Tricoche et al. [159].
- Automatic definition by features. A possibility is to extract separation and attachment lines according to Kenwright [67] or Tricoche et al. [159] and to place initial sampling regions around (part of) those.
- Manual identification and definition by preceding interactive exploration using standard techniques such as path line integration or (AMR) ridge extraction (Chapter 7) of the FTLE in regions of interest. It might seem cumbersome to extract FTLE ridges in a first step with a standard technique, but this can be afforded if the goal is to compute time series of FTLE (Section 9.2.3).

We require the initial sampling regions and resampled regions (Section 9.2.3) to be parts of a virtual uniform grid that covers the complete domain. This makes sure that separated grids are consistently sampled and hence can merge (even after advection) when cells are added by the procedure described in Section 9.2.2.

**Algorithm 2** Grid Advection for FTLE Ridges

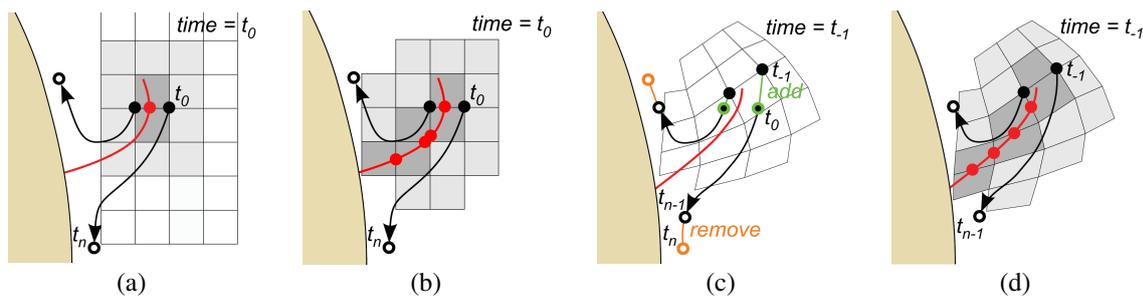
---

```

1: steps: number of steps for the FTLE ridge time series
2: range: topological neighborhood range around ridges
3: tolerance: tolerance for RMS of FTLE
4:
5: place initial sampling grid
6: compute all trajectories and compute FTLE
7:  $\mathcal{R} \leftarrow$  detect ridge cells from FTLE
8:  $\mathcal{N} \leftarrow$  cells in range around  $\mathcal{R}$  //  $\mathcal{N}$  may contain existing cells and cells to add
9:
10: // compute frames of FTLE time series
11:  $r \leftarrow 2$  // number of advection steps
12: lastResampleStep  $\leftarrow 0$ 
13: for step=1 to steps do
14:   // grid growing
15:   while first iteration at step or grid changed do
16:     // add cells in neighborhood range around  $\mathcal{R}$ 
17:     for all cells  $c \in \mathcal{N}$  and not yet in sampling grid do
18:       add  $c$  directly if grid regular, or by advection or meshing
19:     end for
20:     compute (or reuse) trajectories and compute FTLE
21:      $\mathcal{R} \leftarrow$  detect ridge cells from FTLE
22:      $\mathcal{N} \leftarrow$  cells in range around  $\mathcal{R}$ 
23:   end while
24:
25:   // grid shrinking
26:   for all cells  $c$  of sampling grid do
27:     if  $c$  outside domain or  $c \notin (\mathcal{R} \cup \mathcal{N})$  then
28:       remove  $c$ 
29:     end if
30:   end for
31:
32:   // grid advection
33:   if step < steps then
34:     advect grid nodes to next time step
35:     compute (or reuse) trajectories and compute FTLE
36:     // resampling
37:     if step - lastResampleStep >  $r$  then
38:       resample uniformly, recompute all trajectories and compute new FTLE
39:       RMS  $\leftarrow$  measure RMS between old FTLE and FTLE on resampled grid
40:        $r \leftarrow \max(1, \lfloor r * \textit{tolerance} / \textit{RMS} \rfloor)$ 
41:       lastResampleStep  $\leftarrow$  step
42:     end if
43:      $\mathcal{R} \leftarrow$  detect ridge cells from FTLE
44:      $\mathcal{N} \leftarrow$  cells in range around  $\mathcal{R}$ 
45:   end if
46: end for

```

---



**Figure 9.3:** Grid advection for flow attachment. (a) Initial sampling grid. Ridge cells (dark gray) and their neighboring cells (light gray). Cell edge intersected by positive-time FTLE ridge (red point). Neighborhood range is 1 for illustration purposes. (b) After grid adaptation. (c) After one step of grid advection. (d) After grid adaptation of advected grid.

## 9.2.2 Grid Adaptation

We now describe how the initial sampling grid from Section 9.2.1 is adapted to the filtered ridges (cf. Figure 9.2 (b), Figure 9.2 (d), Figure 9.3 (b), and Figure 9.3 (d)). To prevent long extraction times in cases where the ridges extend into regions that are of no interest, a region of interest can be defined which restricts the adaptation, possibly leading to truncated ridges.

**Grid Growing** The first adaptation step is to add new cells to the boundary of the sampling grid where necessary. As defined in Section 7.2.1, a *ridge cell* is a cell that has an edge intersected by a filtered ridge according to (Eq. 7.3) or (Eq. 7.4). Because we aim at results that are identical to those from a uniform sampling as in Chapter 7, the support range of the Hessian, which is needed for the height ridge extraction, has to be taken into account. If the underlying scalar quantity is computed using a local operator (as in the case of FTLE), its support radius has to be added to that of the Hessian as well. Having the total support range, one needs to add all cells to the grid that are within that topological neighborhood of any ridge cell.

In cases of steady vector fields, where the grid advection from Section 9.2.3 does not apply, the sampling grid is uniform and adding cells is a trivial procedure. However, if the grid is advected, adding cells is a challenging problem due to the distortion of the grid. Nevertheless, the initial grid is uniform and the grid gets uniformly resampled from time to time. If we need to add a cell to the distorted grid, we simply go back to the last time step where the grid was uniform, add the nodes of the corresponding cell there and advect the added nodes to the actual time step. This makes the cell fit to the desired position. Additionally, the computed trajectories for the advection of the nodes can be reused for computing the quantity (FTLE), resulting in little overhead.

However, if a node of the cell in the uniform grid is located outside of the domain, there is no vector field that could be used to advect it to the desired time step and position. In this case the cell can be constructed by extrapolation of the grid or any standard meshing technique. The grid growing procedure is iterated until convergence, meaning that each added cell and its neighbors are tested for being a ridge cell and if this is the case, it is attempted to add the cells inside the neighborhood range. This way, the sampling grid grows to the necessary extent, similar to the AMR approach from Chapter 7.

**Grid Shrinking** The next step is to remove unnecessary cells from the grid. These are cells that are neither ridge cells nor in the relevant neighborhood of any ridge cell, or cells where one or more nodes are outside of the domain.

### 9.2.3 Grid Advection

Lagrangian coherent structures are material lines or material surfaces [48], in other words, they advect with the flow, similarly to streak lines (surfaces) and time lines (surfaces). This would allow for exploiting temporal coherence for the generation of time series of FTLE ridges by advection of the extracted ridges. One could compute the FTLE and its ridges only after every  $n$  time steps and advect them with the flow in between. However, this would not account for changes in the FTLE during advection. New ridges can originate and existing ones can grow, shrink, or disappear, especially if the ridges are filtered as in our case (Section 7.1.2). Therefore we propose a different approach: the advection of the sampling grid itself during the advection intervals. This results in a generic method for quantities based on trajectories, not only FTLE.

During advection, a short trajectory has to be computed for each node of the sampling grid to advance it to the next position. The striking advantage is that these short trajectories can be appended to the existing trajectories needed for the computation of the FTLE, making it possible to reuse large parts of the trajectories and hence improving efficiency, see Figure 9.2 (c) and Figure 9.3 (c).

As already mentioned, advection of the sampling grid tends to distort its cells and this in turn tends to affect the computation of derivatives, which are needed for FTLE computation and ridge extraction. Additionally, compared to other quantities, the FTLE tends to be particularly sampling dependent, especially if not computed using renormalization (Section 8.1.1). All in all, this generally leads to artifacts in the extracted FTLE ridges such as deformation, false negatives, and false positives.

To restrict the artifacts to an appropriate level, the FTLE is periodically resampled on a subset of the virtual uniform grid spanning the whole domain: only those cells of the grid are generated (and the corresponding trajectories are computed) which overlap with the advected grid or which are contained in the region of the initial sampling. An additional strategy is to place the sampling grid outside re-

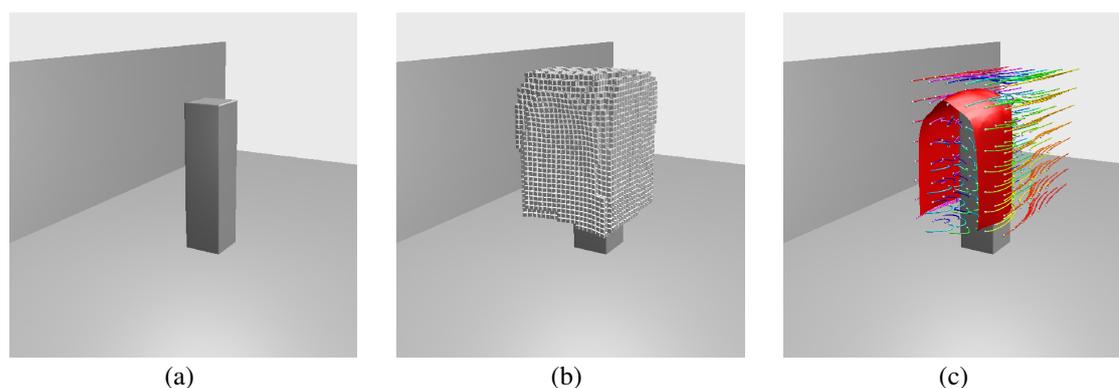
gions producing high distortion such as wakes and vortices. Although this looks like a compromise, it is often a natural choice to analyze LCS away from disturbing phenomena since they would also distort them, even when uniformly sampled, and hence complicate interpretation.

Because the flow map is computed by integrating trajectories in discrete vector fields, and because of the intricacy of gradient computation on unstructured grids, it is generally not possible to provide error bounds with respect of the distortion of the grid. Garth et al. [36] measure the error for their subdivision scheme in the flow map, not based on FTLE or its ridges. Similarly, we propose to measure the error based on the FTLE, not its ridges, and to use it for triggering the resampling.

The grid is uniformly resampled (recomputing the trajectories and the FTLE) after every  $r$  advection steps with an initial value of  $r = 2$ . After the FTLE has been computed on the resampled grid, the FTLE of the advected grid is interpolated at the node positions of the new grid and the root mean square (RMS) of the difference over all nodes is computed. The RMS is then tested against a user-defined tolerance, and a new  $r$  is estimated from the RMS and from the tolerance by linearization of the RMS over the advection steps (line 40 of the algorithm). The algorithm then proceeds to the next advection phase. However, the linearization of the error can be inappropriate in the sense that  $r$  is chosen too large such that after the next  $r$  advection steps the RMS exceeds the prescribed tolerance. One solution to this problem is to enforce the tolerance by reducing  $r$  (and hence undoing advection steps) until the RMS tolerance is fulfilled. However, our experiments have shown that it is usually sufficient not to enforce the tolerance and instead to prescribe a reduced tolerance, e.g., by 15 percent, to achieve the intended precision.

To support the user in an appropriate choice of the RMS tolerance and the sampling region, we provide visualization of both sets of ridges, those before and after resampling, or color-coding the former ones by their distance to the latter as shown in Figure 9.1 (b), serving as uncertainty information. Another approach is to judge the popping artifacts visually when moving from a time step based on an advected grid to a subsequent time step that has been derived using a uniformly resampled computation grid.

Note that for the analysis of flow separation, time series of FTLE ridges have to get generated by advecting the grid in positive time direction (Figure 9.2), whereas for attachment the grid has to get advected in negative-time direction (Figure 9.3). This is necessary since the ridges are captured at the regions of interest (e.g., at the boundaries) and FTLE ridges visualizing attachment approach the boundary in positive time. Hence it is necessary to start with the last time step and to compute them in negative time direction in order to capture all of them at the boundary, even those that separate from the boundary later on. So finally, all LCS that come in contact with the boundary (or other regions of interest) at any time get captured.



**Figure 9.4:** Flow around a cuboid. (a) Geometry. (b) Sampling grid adapted to ridge region and advected. (c) Resulting FTLE ridge with some upstream trajectories (colored) from uniform grid, and their seeds (white spheres).

## 9.3 Results

In this section the presented method is applied to two unsteady CFD simulations. The first one examines the flow around a cuboid, exhibiting flow similar to a von Kármán vortex street, but in this case the vortices become tilted soon after they detach from the cube. This leads to flow separation behavior that differs from the standard von Kármán vortex street. Secondly, the method is applied to a simulation of an intake of a river power plant. The scope there is a construct that prevents salmon from getting into the runner of the turbine.

### 9.3.1 Flow around a Cuboid

This example produces a kind of a von Kármán vortex street. The unsteady flow comes from the right back and follows to the left front (Figure 9.4 (a)). The main difference to a common von Kármán vortex street is that there is also flow over the top face of the cuboid. The flow separation at the cuboid is the subject of analysis in this case. The resulting FTLE ridge (Figure 9.4 (c)) shows that flow separation is in progress on both sides as well as the top of the cuboid. It can be seen that the FTLE ridge separates the vortex street region from the outer flow. However, further downstream the FTLE ridge does not exhibit this property anymore: it crosses the vortices. Time series of FTLE ridges reveal that the separation zones are oscillating consistently with the von Kármán vortex street.

Table 9.1 shows some performance details for this example. The achieved speedup in this case is only about 2.3. This is due to the relatively short trajectories (Figure 9.4 (c)). The prescribed RMS tolerance was 15.0 and at step 33 this was exceeded by 0.88 percent. There have been 12 advection phases, each performed 5 advection steps in average. Because of the shape of the FTLE ridge and

	uniform	grid advection		uniform	direct on adapted grid	grid advection
grid [nodes]	16399	14220 (step 33)	grid [nodes]	8800	5007	3913 (step 39)
flow map [s]	13704.87	2944.88	flow map [s]	15369.17	9062.73	355.62
total [s]	13707.92	5800.31	total [s]	15374.22	9489.96	1026.81
speedup	1	2.36	speedup	1	1.62	14.97
Figure		9.4 (b)	Figure	9.5 (a)		9.5 (b)

**Table 9.1:** Performance analysis for the cuboid dataset. 60 steps of grid advection compared to 61 direct evaluations on uniform grid. See also Figure 9.4 (b).

**Table 9.2:** Performance analysis for the turbine intake dataset. 100 steps of grid advection compared to 101 direct evaluations (on uniform grid and adapted grid). See also Figure 9.5 (a) and Figure 9.5 (b).

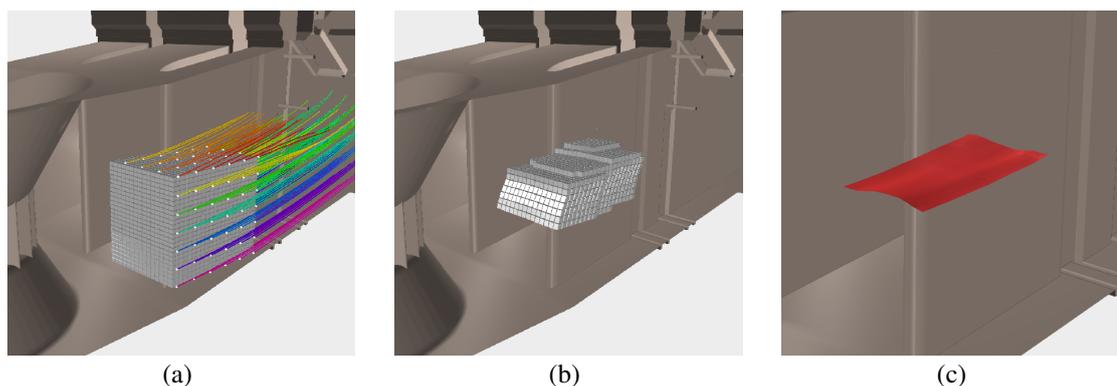
because the initial sampling grid is already well adapted to the FTLE ridge, the expected speedup from grid adaptation is small and was therefore not measured.

## 9.3.2 Intake of a Power Plant

The underlying data of this section is an existing run-of-river plant in the US. All devices shown are installed in the intake of the plant in order to protect juvenile salmon from passing through the runner. The water flow of the unsteady CFD simulation comes from the right back and follows to the left front where it enters the turbine (Figure 9.5).

The horizontal rods at the right hand side of the image lead the salmon into the vertical channels at the top in the installation. However, these rods produce a noticeable wake in the upper part of the main channel (see path lines in Figure 9.5 (a)). Additionally, the back-flow from the salmon channel (the opening at the top downstream from the rods) also is involved in a recirculation zone at the top wall, located above the sampling grid of Figure 9.5 (a). On the one hand, an FTLE ridge was extracted using a regular grid at the confluence of the three main channels (Figure 9.5 (a)), on the other it was extracted using the presented grid advection method (Figure 9.5 (b) and Figure 9.5 (c)). The obtained FTLE ridge separates well the fast flow at the bottom of the channel from the slower flow in the upper half of the channel.

Table 9.2 shows some performance measurements of the presented case. The speedup from grid adaptation is quite low (1.62) because of the relatively low resolution of the sampling grid and because the sampling region was already quite well adapted to the ridge. The speedup from including grid advection is significantly higher (about 15) and would further increase with increasing the integration time for the trajectories. The RMS tolerance was set to 0.012 and at step 39 this was exceeded by 14.2 percent, which was the maximum during the 13 advection



**Figure 9.5:** Intake of a water turbine. (a) Uniform grid with some of the upstream trajectories (colored) used for FTLE computation, and their seeds (white spheres). (b) Sampling grid adapted to ridge region and advected. (c) Resulting FTLE ridge.

phases. Figure 9.1 (b) shows the corresponding distance error of the ridge. In average, 7.7 advection steps were performed per advection phase.

## 9.4 Discussion

We presented a generic method for accelerating the computation of time series of quantities based on trajectories, such as FTLE. On the one hand the efficiency is improved by restricting the sampling grid to the phenomena of interest, on the other hand and more important, the computation is accelerated by reusing part of the trajectories, which is made possible by advection of the sampling grid. In the case of gradient-based visualizations, such as FTLE ridges, the quality tends to suffer if the distortion caused by the advection of the grid is high. Therefore, the obtained quality is inferior to evaluations on regular grids or that from Chapter 7 in terms of quality, but superior in terms of speed if time series are computed. All in all, we propose to use the method at least as a fast preview technique and to use low RMS error thresholds (leading to low acceleration) or even exact methods, such as direct computation on uniform grids or that by Garth et al. [36] or ours from Chapter 7, if exact time series are required.



## Chapter 10

---

# Conclusion

This chapter gives an overview of the methods presented in this thesis, together with a discussion and possible related future work.

### 10.1 Overview of Contributions

The following contributions to the field of visualization of physics and topology in unsteady flow resulted in the context of this thesis:

- A method for the placement and augmentation of vorticity field lines has been presented, providing a more holistic analysis of vortices, i.e., the overall structure of the vorticity field is conveyed. This not only visualizes vortical flow, but also the interrelation with shear flow, an important aspect since vortices typically originate from shear flow. However, although many aspects of vortical flow are amenable to an analysis by this technique, only limited capability is offered for the analysis of transport of vorticity. Hence, as a complementary approach, this is the focus of the next contribution.
- A technique for the analysis of vorticity transport in transient flow has been introduced. Several scalar quantities are derived from the vorticity equation and mapped together with additional values on path lines. This allows to investigate the physical phenomena that lead to a given vortex at a given time. We were able to provide insight into vortex generation to our industry partners in turbo-machinery applications, e.g., we identified vortex stretching and vortex tilting to be major mechanisms of vortex generation in many configurations.
- The vortex breakdown bubble phenomenon represents the link from the first part of the thesis dealing with vortices, to the second part investigating the overall structure of vector fields in terms of topology. Several techniques have been presented for the analysis of the phenomenon, from automatic extraction, to techniques that reveal the internal organization.

- Filtered AMR ridge extraction has been presented as a means for efficient ridge surface extraction. The method is based on adaptive refinement of the sampling grid and allows for high speed-up especially if the underlying scalar field is sampled during ridge extraction. This is the case for the computationally expensive finite-time Lyapunov exponent (FTLE), our primary application, where densely seeded sets of trajectories have to be computed. Nevertheless, the presented technique offers acceleration also on discretized data. Although a drawback is the risk of missing complete ridges, the obtained ridges are exact, i.e., identical to those derived by straightforward ridge extraction.
- An algorithm for the computation of a variant of the FTLE, the finite-size Lyapunov exponent (FSLE) has been presented together with a new variant of FTLE, the FTLE maximum (FTLEM). FSLE was introduced by Aurell et al. in an attempt to make the concept less dependent on the scope parameter, and so has been the FTLEM. We compare the different approaches using the filtered AMR ridge extraction method.
- Whereas the filtered AMR ridge extraction represents an acceleration technique for computing FTLE variants at particular points in time, it does not exploit temporal coherence if several points in time, i.e., time series, are to be analyzed. Therefore a method based on the advection of the computational grid has been presented. Since the grid advects along path lines, large portions of trajectories can be reused, leading to a substantial speed-up. A drawback is, however, that the computational grid tends to distort over time, affecting numerics. Therefore, resampling is triggered by a user-defined error threshold.

Although the two parts of this thesis may seem distinct, there are several links between them. First of all, the axes of vortices, the vortex core lines, are related to the vector field topology in appropriate projections of the flow, and, as shown, vortex breakdown bubbles can directly be analyzed by techniques from 3D vector field topology. Second, vortices also represent coherent structures. One of the first definitions of coherent structures by Robinson even was in terms of vortices, and they are also amenable to an analysis by FTLE. Further, recirculation zones, a common case of vortical motion, and flow separation in general can be analyzed in terms of separation and attachment lines both also related to vector field topology. To some extent, this thesis also reflects and supports the transition trend from steady to transient data and from Eulerian to Lagrangian analysis in visualization.

## 10.2 Discussion

The analysis and understanding of vortical phenomena has a long history but is by far not finished yet. Progresses are made regularly in that field. In the first part

of this thesis methods have been presented that focus on the physics of flows, in particular, the understanding of vortex generation and vortex dynamics in general. These allow a deeper understanding of how and why vortices originate and enable in wide parts of research and engineering a more detailed investigation and more focused design. A possible direction for further research is the analysis of vorticity transport due to diffusion. The presented technique allows for quantitative analysis of that mechanism but the qualitative, i.e., the directional transport by diffusion, is addressed in an indirect manner by neighboring path lines and neighborhood disks. A method that explicitly visualizes the transport due to diffusion could give additional insight since diffusion is often the only cause why vorticity transport deviates from pure advection, i.e., is not “frozen into the flow”.

In the second part of this thesis we have focused on approaches in the field of vector field topology. We enabled efficient and effective visualization of vortex ring phenomena, in particular those known as vortex breakdown bubbles. This phenomenon is present in many fields in industry and nature, sometimes generated by purpose, e.g., for mixing, and sometimes unwanted, as in the case of aviation. Our techniques not only reveal the complex structure and allow for a simplified examination of real-world data with respect to dynamical systems theory, they also allow for their automatic detection and extraction, which is important in many investigations. The development of further techniques for the qualitative and quantitative analysis of the flow inside these features with respect to mixing as well as their throughput could be fruitful. A special concern in this context is accuracy, due to the exponential growth of error inside the chaotic regime.

We then proceeded to the topology of time-dependent vector fields. We introduced the concept of the finite-time Lyapunov exponent, popular in the field of fluid mechanics in the last decade, to the visualization community. Its ridges, the Lagrangian coherent structures, represent a time-dependent counterpart to separatrices in vector field topology and hence represent indeed a solution in the search for a time-dependent vector field topology, a quest ongoing in the visualization community for more than two decades. Lagrangian coherent structures are a promising approach for the analysis but they suffer from very high computational cost, often substantially inhibiting research and development. We proposed two approaches for lowering that cost and hence making the method better applicable in everyday life. One accelerates the computation of the FTLE computation at selected times whereas the other allows for efficient computation of time series of FTLE. However, since both approaches are based on adaptive refinement, one only in space and the other also in time, they suffer from a drawback typical in that field: they can miss features. Furthermore, the latter is typically affected by distortion of the sampling grid, which can nevertheless be controlled by a prescribed error tolerance. A possible research direction could investigate other approaches that are less vulnerable to numerical issues due to advection of the sample points.



# Appendix A

---

## Notation

Throughout this thesis, all scalar variables or functions are set in italics, while vector-valued expressions and functions, including matrices, are set in bold face. Symbols are also defined where they first appear.

### Operators

$\cdot$	dot product
$\times$	cross product
$\nabla$	gradient
$\dot{\mathbf{u}}, \ddot{\mathbf{u}}$	first and second time derivative: $\dot{\mathbf{u}} = \frac{\partial \mathbf{u}}{\partial t}$ , $\ddot{\mathbf{u}} = \frac{\partial^2 \mathbf{u}}{\partial t^2}$
$\nabla \cdot$	divergence
$\nabla \times$	curl
$\nabla^2$	Laplacian = $\nabla \cdot \nabla$
$\nabla \nabla$	Hessian
$\ \cdot\ $	Euclidean norm
$ \cdot $	modulus
det	determinant
$\cdot^T$	matrix transpose

### Physics

$\mathbf{u}$	velocity [ $m \cdot s^{-1}$ ]
$\boldsymbol{\omega}$	vorticity [ $s^{-1}$ ]
$p$	pressure [ $kg \cdot m^{-1} \cdot s^{-2}$ ]
$\rho$	density [ $kg \cdot m^{-3}$ ]

$\nu$	kinematic viscosity [ $m^2 \cdot s^{-1}$ ]
$\mu$	dynamic viscosity [ $kg \cdot m^{-1} \cdot s^{-1}$ ]

# Bibliography

- [1] *CFX-5 User manual*. AEA technology, Harwell, UK, 2002. 67
- [2] Abraham, R. H. and Shaw, C. D. *Dynamics, the Geometry of Behavior*. 2nd ed. Addison-Wesley, 1992. 29
- [3] Asimov, D. Notes on the Topology of Vector Fields and Flows. Technical Report RNR-93-003, NASA Ames Research Center, 1993. 29, 34, 86, 125
- [4] Aurell, E., Boffetta, G., Crisanti, A., Paladin, G., and Vulpiani, A. Predictability in the large: an extension of the concept of Lyapunov exponent. *J. Phys. A: Math. Gen.*, 30:1–26, 1997. 128
- [5] Balsara, D. and Kim, J. An Intercomparison Between Divergence-Cleaning and Staggered Mesh Formulations for Numerical Magnetohydrodynamics. *Eprint Astrophysics astro-ph/0310728*, 2003. 109
- [6] Banks, D. C. and Singer, B. A. Vortex tubes in turbulent flows: identification, representation, reconstruction. In *VIS '94: Proceedings of the conference on Visualization '94*, pages 132–139, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press. 19
- [7] Banks, D. C. and Singer, B. A. A Predictor-Corrector Technique for Visualizing Unsteady Flow. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):151–163, 1995. 44
- [8] Barré de Saint-Venant, A. J. C. Surfaces à plus grande pente constituées sur des lignes courbes. *Bulletin de la soc. philomath. de Paris*, pages 24–32, March 1852. 42
- [9] Batchelor, G. K. *An Introduction to Fluid Mechanics*. Cambridge University Press, 1967. 65
- [10] Bauer, D. and Peikert, R. Vortex tracking in scale-space. In *VISSYM '02: Proceedings of the Symposium on Data Visualisation 2002*, pages 233–ff, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association. 45, 50
- [11] Bauer, D., Peikert, R., Sato, M., and Sick, M. A Case Study in Selective Visualization of Unsteady 3D Flow. In *Proceedings of IEEE Visualization*, 2002. 14

- [12] Belyaev, A. G., Anoshkina, E. V., and Kunii, T. L. Ridges, ravines, and singularities. In Fomenko, A. T. and Kunii, T. L., editors, *Topological Modeling for Visualization*, chapter 18, pages 375–383. Springer, 1997. 43
- [13] Benettin, G., Galgani, L., Giorgilli, A., and Strelcyn, J. M. Lyapunov characteristic exponent for smooth dynamical systems and Hamiltonian systems; a method for computing all of them. *Mechanica*, 15:9–20, 1980. 127
- [14] Brackbill, J. and Barnes, D. The effect of nonzero  $\nabla \cdot \mathbf{B}$  on the numerical solution of the magnetohydrodynamic equations. *J. Comput. Phys.*, 35:426–430, 1980. 109
- [15] Breton (de Champ), P. E. Note sur les lignes de faite et de thalweg. *Comptes rendus hebdomadaires des séances de l'académie des Sciences*, 39:647–648, 1854. 42
- [16] Bürger, K., Kondratieva, P., Krüger, J., and Westermann, R. Importance-Driven Particle Techniques for Flow Visualization. In Fujishiro, I., Li, H., and Ma, K.-L., editors, *Proceedings of IEEE VGTC Pacific Visualization Symposium 2008*, pages 71–78, March 2008. 40
- [17] Chernoff, H. The use of faces to represent points in k-dimensional space graphically. *J. A, Stat Assoc*, 68:361–368, 1973. 20
- [18] Chong, M., Perry, A., and Cantwell, B. A general classification of three-dimensional flow fields. *Phys. Fluids, A* 2(5):765–777, 1990. 19, 125
- [19] Cornea, N. D., Silver, D., and Min, P. Curve-Skeleton Properties, Applications, and Algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):530–548, 2007. 43
- [20] Cottet, G. and Koumoutsakos, P. *Vortex Methods: Theory and Practice*. Cambridge University Press, 2000. 18
- [21] Damon, J. Properties of Ridges and Cores for Two-Dimensional Images. *J. Math. Imaging Vis.*, 10(2):163–174, 1999. 116
- [22] de Leeuw, W. C. and van Wijk, J. J. A Probe for Local Flow Field Visualization. In *IEEE Visualization*, pages 39–45, 1993. 21
- [23] Delmarcelle, T. *The visualization of second-order tensor fields*. PhD thesis, Stanford, CA, USA, 1995. 39
- [24] Delmarcelle, T. and Hesselink, L. Visualizing Second-Order Tensor Fields with Hyperstreamlines. *IEEE Comput. Graph. Appl.*, 13(4):25–33, 1993. 25, 39
- [25] Delmarcelle, T. and Hesselink, L. The Topology of Symmetric, Second-Order Tensor Fields. In *Proceedings of the Conference on Visualization 1994*, pages 140–147, 1994. 39

- 
- [26] Doleisch, H., Gasser, M., and Hauser, H. Interactive feature specification for focus+context visualization of complex simulation data. In *VisSym '03: Symposium on Visualisation*, pages 239–248, 2003. 64
- [27] Dubief, Y. and Delcayre, F. On coherent-vortex identification in turbulence. *Journal of Turbulence*, 1, December 2000. 15, 17
- [28] Eberly, D. *Ridges in Image and Data Analysis. Computational Imaging and Vision*. Kluwer Academic Publishers, 1996. 42, 116
- [29] Ebling, J. and Scheuermann, G. Clifford Convolution And Pattern Matching On Vector Fields. In *Proceedings of IEEE Visualization*, pages 193 – 200, 2003. 15
- [30] E.Parkinson, Garcin, H., Bissel, C., Muggli, F., and Braune, A. Description of Pelton Flow Patterns with Computational Flow Simulations. In *Hydro 2002*, November 2002. 56
- [31] Fang, D. C., Weber, G. H., Childs, H. R., Brugger, E. S., Hamann, B., and Joy, K. I. Extracting Geometrically Continuous Isosurfaces from Adaptive Mesh Refinement Data. 119
- [32] Fuchs, R., Peikert, R., Hauser, H., Sadlo, F., and Muigg, P. Parallel Vectors Criteria for Unsteady Flow Vertices. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):615–626, 2008. 45
- [33] Fuchs, R., Peikert, R., Sadlo, F., Alsallakh, B., and Gröller, M. E. Delocalized Unsteady Vortex Region Detectors. In Oliver Deussen, Daniel Keim, D. S., editor, *Proceedings VMV 2008*, pages 81–90, October 2008. 28
- [34] Fuhrmann, A. L. and Gröller, E. Real-Time Techniques for 3D Flow Visualization. In Ebert, D., Hagen, H., and Rushmeier, H., editors, *Proceedings of IEEE Visualization*, pages 305–312, 1998. 25, 54
- [35] Furst, J. D. and Pizer, S. M. Marching Ridges. In *Proceedings of the IASTED International Conference, Honolulu*, pages 22–26, 2001. 42, 115
- [36] Garth, C., Gerhardt, F., Tricoche, X., and Hagen, H. Efficient Computation and Visualization of Coherent Structures in Fluid Flow Applications. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1464–1471, 2007. 40, 144, 147
- [37] Garth, C., Li, G.-S., Tricoche, X., Hansen, C. D., and Hagen, H. Visualization of Coherent Structures in Transient Flows. In Hege, H.-C., Polthier, K., and Scheuermann, G., editors, *Topology-Based Methods in Visualization II (Proceedings of Topology-Based Methods in Visualization 2007)*, pages 1–14. Springer, 2008. 126
- [38] Garth, C., Tricoche, X., Salzbrunn, T., Bobach, T., and Scheuermann, G. Surface Techniques for Vortex Visualization. In *VisSym*, pages 155–164, 346, 2004. 27, 50, 85, 95

- [39] Garth, C., Tricoche, X., and Scheuermann, G. Tracking of Vector Field Singularities in Unstructured 3D Time-Dependent Datasets. In *Proceedings of IEEE Visualization*, pages 329–336, Los Alamitos, CA, USA, 2004. IEEE Computer Society. 95
- [40] Globus, A., Levit, C., and Lasinski, T. A Tool for Visualizing the Topology of Three-Dimensional Vector Fields. In *Proceedings of IEEE Visualization*, pages 33–41, 1991. 20, 29, 31, 37, 50
- [41] Goldhirsch, I., Sulem, P.-L., and a. Orszag, S. Stability and Lyapunov Stability of Dynamical Systems: A Differential Approach and a Numerical Method. *Physica D*, 27:311–337, 1987. 125
- [42] Green, M., Rowley, C., and Haller, G. Detection of Lagrangian Coherent Structures in 3D Turbulence. *J. Fluid Mech.*, 2007. 125
- [43] Guckenheimer, J. and Holmes, P. *Nonlinear Oscillations, Dynamical Systems and Bifurcations of Vector Fields*. Applied Mathematical Sciences, Vol. 42. Springer, New York, Berlin, Heidelberg, Tokyo, 1983. 29, 85
- [44] Haber, R. B. Visualization techniques for engineering mechanics. *Comput. Syst. Educ.*, 1(1):37–50, 1990. 21
- [45] Haber, R. B. and McNabb, D. A. Visualization Idioms: A Conceptual Model for Scientific Visualization Systems. In *Visualization in Scientific Computing*, pages 74–93. CS Press, 1990. 21
- [46] Haines, R. and Kenwright, D. On the Velocity Gradient Tensor and Fluid Feature Extraction, 1999. 19
- [47] Haller, G. Lagrangian structures and the rate of strain in a partition of two-dimensional turbulence. *Phys. Fluids*, 13:3365–3385, 2001. 99
- [48] Haller, G. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D*, 149:248–277, 2001. 40, 119, 125, 126, 127, 128, 137, 143
- [49] Haller, G. Lagrangian coherent structures from approximate velocity data. *Phys. Fluids, A* 14:1851–1861, 2002. 125
- [50] Haller, G. An objective definition of a vortex. *J. Fluid Mech.*, 252:1–26, 2005. 19, 28
- [51] Haralick, R. Ridges and Valleys on Digital Images. *Computer Vision, Graphics, and Image Processing*, 22(10):28–38, 2001. 42
- [52] Helman, J. and Hesselink, L. Representation and Display of Vector Field Topology in Fluid Flow Data Sets. *IEEE Computer*, 22(8):27–36, 1989. 15, 29, 125

- 
- [53] Helman, J. and Hesselink, L. Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, 1991. 29, 46
- [54] Hesselink, L., Levy, Y., and Lavin, Y. The Topology of Symmetric, Second-Order 3D Tensor Fields. *IEEE Transactions on Visualization and Computer Graphics*, 03(1):1–11, 1997. 39
- [55] Hua, B., McWilliams, J., and Klein, P. Lagrangian accelerations in geostrophic turbulence. *Journal of Fluid Mechanics*, 366:87–108, 1998. 19
- [56] Hultquist, J. P. Constructing Stream Surfaces in Steady 3-D Vector Fields. In *Proceedings of IEEE Visualization*, pages 171–178, 1992. 27, 50
- [57] Hunt, J., Wray, A., and Moin, P. Eddies, stream, and convergence zones in turbulent flows. Technical Report CTR-S88, Center for Turbulence Research, NASA Ames Research Center and Stanford University, 1988. 17, 19, 125
- [58] Hussain, F. Coherent structures and turbulence. *Journal of Fluid Mechanics*, 173:303–356, 1986. 125
- [59] Iijima, T. Basic theory of pattern observation. *Papers of Technical Group on Automata and Automatic Control, IECE*, 1959. 45
- [60] Interrante, V., Fuchs, H., and Pizer, S. M. Enhancing Transparent Skin Surfaces with Ridge and Valley Lines. In *Proceedings of IEEE Visualization*, pages 52–59, 1995. 43
- [61] Jankun-Kelly, M., Jiang, M., Thompson, D., and Machiraju, R. Vortex Visualization for Practical Engineering Applications. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):957–964, 2006. 45
- [62] Jeong, J. and Hussain, F. On the Identification of a Vortex. *Journal of Fluid Mechanics*, 285(69):69–94, 1995. 14, 15, 18, 125
- [63] Jiang, M., Machiraju, R., and Thompson, D. Geometric Verification of Swirling Features in Flow Fields, 2002. 50
- [64] Jobard, B. and Lefer, W. Creating Evenly-Spaced Streamlines of Arbitrary Density. In Lefer, W. and Grave, M., editors, *Visualization in Scientific Computing '97. Proceedings of the Eurographics Workshop in Boulogne-sur-Mer, France*, pages 43–56, Wien, New York, 1997. Springer Verlag. 25, 54
- [65] Jordan, C. Sur les lignes de faite et de thalweg. *Comptes rendus hebdomadaires des séances de l'académie des Sciences*, 74:1457–1459, 1872. 43
- [66] Kasten, J., Petz, C., Hotz, I., Noack, B., and Hege, H.-C. Localized Finite-time Lyapunov Exponent for Unsteady Flow Analysis. In Magnor, M., Rosenhahn, B., and Theisel, H., editors, *Vision Modeling and Visualization*, volume 1, pages 265–274. Universität Magdeburg, Inst. f. Simulation u. Graph., 2009. 127

- [67] Kenwright, D. N. Automatic detection of open and closed separation and attachment lines. In *VIS '98: Proceedings of the conference on Visualization '98*, pages 151–158, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press. 140
- [68] Kenwright, D. N., Henze, C., and Levit, C. Feature Extraction of Separation and Attachment Lines. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):135–144, 1999. 46, 90, 130
- [69] Kindlmann, G., Weinstein, D., and Hart, D. Strategies for Direct Volume Rendering of Diffusion Tensor Fields. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):124–138, 2000. 20
- [70] Kindlmann, G., Tricoche, X., and Westin, C.-F. Anisotropy Creases Delineate White Matter Structure in Diffusion Tensor MRI. In *Ninth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'06)*, Lecture Notes in Computer Science 4190, pages 126–133, October 2006. 43, 117
- [71] Kindlmann, G. L. Superquadric tensor glyphs. In *VisSym*, pages 147–154, 2004. 21
- [72] Kindlmann, G. L. and Westin, C.-F. Diffusion Tensor Visualization with Glyph Packing. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1329–1336, 2006. 21
- [73] Koenderink, J. J. The structure of images. *Biol. Cybern.*, 50:363–370, 1984. 45
- [74] Koenderink, J. J. and van Doorn, A. J. Two-Plus-One-Dimensional Differential Geometry. *Pattern Recognition Letters*, 15(5):439–443, May 1994. 43
- [75] Krasny, R. and Fritsche, M. The onset of chaos in vortex sheet flow. *J. Fluid Mech.*, 454:47–69, 2002. 94
- [76] Krishnan, H., Garth, C., and Joy, K. Time and Streak Surfaces for Flow Visualization in Large Time-Varying Data Sets. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1267–1274, 2009. 28
- [77] Kundu, P. and Cohen, I. *Fluid Mechanics. 2nd ed.* Academic Press, 2002. 23
- [78] Laramee, R. S., Garth, C., Schneider, J., and Hauser, H. Texture Advection on Stream Surfaces: A Novel Hybrid Visualization Applied to CFD Simulation Results. In *EuroVis 2006: Symposium on Visualization*, May 2006. 15
- [79] Leeuw, W. and Liere, R. Collapsing Flow Topology Using Area Metrics. In *Vis99*, pages 349–354, 1999. 39, 50
- [80] Levy, Y., Degani, D., and Seginer, A. Graphical Visualization of Vortical Flows by Means of Helicity. *AIAA*, 28(8):1347–1352, 1990. 15, 44, 50, 74
- [81] Lindeberg, T. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1996. 42

- 
- [82] Löffelmann, H., Doleisch, H., and Gröller, E. Visualizing dynamical systems near critical points. In Kalos, L. S., editor, *14th Spring Conference on Computer Graphics*, pages 175–184, April 1998. 31
- [83] Löffelmann, H. and Gröller, E. Enhancing the visualization of characteristic structures in dynamical systems. In Bartz, D., editor, *Visualization in Scientific Computing '98*, pages 59–68. Springer, 1998. 31
- [84] Löffelmann, H., Kucera, T., and Gröller, E. Visualizing Poincaré Maps Together With the Underlying Flow. In *Proceedings of the International Workshop on Visualization and Mathematics '97*, pages 315–328, New York, NY, USA, 1998. 33
- [85] Löffelmann, H. *Visualizing Local Properties and Characteristic Structures of Dynamical Systems*. PhD thesis, Technische Universität Wien, 1998. 33
- [86] Lorensen, W. E. and Cline, H. E. Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, volume 21, pages 163–169, New York, NY, USA, July 1987. ACM Press. 20, 116
- [87] Lorenz, E. N. A study of the predictability if a 28-variable atmospheric model. *Tellus*, 17:321–333, 1965. 125
- [88] Lucca-Negro, O. and O'Doherty, T. Vortex breakdown: a review. *Progress in Energy and Combustion Science*, 27(4):431 – 481, 2001. 94, 95
- [89] Lugt, H. *Vortex Flow in Nature and Technology*. Wiley, 1972. 14
- [90] MacKay, R. S., Meiss, J. D., and Percival, I. C. Transport in Hamiltonian Systems. *Physica D*, 13D:55–81, 1984. 105, 106
- [91] Mahrous, K., Bennett, J., Scheuermann, G., Hamann, B., and Joy, K. I. Topological Segmentation in Three-Dimensional Vector Fields. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):198–205, 2004. 39
- [92] Majer, P. *A Statistical Approach to Feature Detection and Scale Selection in Images*. PhD thesis, University of Goettingen, 2000. 42
- [93] Mallo, O., Peikert, R., Sigg, C., and Sadlo, F. Illuminated Lines Revisited. In *Proceedings of IEEE Visualization*, pages 19–26, 2005. 16
- [94] Mao, X., Hatanaka, Y., Higashida, H., and Imamiya, A. Image-guided streamline placement on curvilinear grid surfaces. In *Proceedings of IEEE Visualization*, pages 135–142, 1998. 54
- [95] Mattausch, O., Bl, T. T., Hauser, H., and Gröller, E. Strategies for interactive exploration of 3D flow using evenly-spaced illuminated streamlines. In *SCCG '03: Proceedings of the 19th spring conference on Computer graphics*, pages 213–222, 2003. 25, 54

- [96] Max, N., Williams, P., Silva, C., and Cook, R. Volume Rendering for Curvilinear and Unstructured Grids. *CGI*, 0:210, 2003. 20
- [97] McLoughlin, T., Laramée, R. S., Peikert, R., Post, F. H., and Chen, M. Over Two Decades of Integration-Based, Geometric Flow Visualization. In Pauly, M. and Greiner, G., editors, *EG 2009 - State of the Art Reports*, pages 73–92, Munich, Germany, April 2009. Eurographics Association. 27, 28
- [98] Menter, F. R. Zonal two equation k-omega turbulence models for aerodynamic flows. In *24th AIAA Fluid Dynamics Conference*, pages 1–21, 1993. 67
- [99] Miura, H. and Kida, S. Identification of Tubular Vortices in Turbulence. *Journal of the Physical Society of Japan*, 66:1331–1334, May 1997. 18, 19, 43, 44
- [100] Nese, J. M. Quantifying Local Predictability in Phase Space. *Physica D*, 35:237–250, 1989. 127
- [101] Norbury, J. A family of steady vortex rings. *J. Fluid Mech.*, 57, Pt. 3:417–431, 1973. 93, 102
- [102] Okubo, A. Horizontal dispersion of floatable trajectories in the vicinity of velocity singularities such as convergences. *Deep Sea. Res.*, 17:445–454, 1970. 17
- [103] Peikert, R. and Roth, M. The parallel vectors operator: a vector field visualization primitive. In *Proceedings of IEEE Visualization*, pages 263–270, 1999. 44, 50, 115
- [104] Peikert, R. and Sadlo, F. Topology-guided Visualization of Constrained Vector Fields. In Helwig Hauser, H. H. and Theisel, H., editors, *Topology-Based Methods in Visualization (Proceedings of Topology-Based Methods in Visualization 2005)*, pages 21–34. Springer-Verlag, 2007. 3
- [105] Peikert, R. and Sadlo, F. Visualization Methods for Vortex Rings and Vortex Break-down Bubbles. In *Proceedings of EuroVis 2007*, pages 23–25, 2007. 3, 109
- [106] Peikert, R. and Sadlo, F. Flow Topology Beyond Skeletons: Visualization of Features in Recirculating Flow. In Hege, H.-C., Polthier, K., and Scheuermann, G., editors, *Topology-Based Methods in Visualization II (Proceedings of Topology-Based Methods in Visualization 2007)*, pages 145–160. Springer, 2008. 3
- [107] Peikert, R. and Sadlo, F. Height Ridge Computation and Filtering for Visualization. In Fujishiro, I., Li, H., and Ma, K.-L., editors, *Proceedings of IEEE VGTC Pacific Visualization Symposium 2008*, pages 119–126, March 2008. 43, 115, 117
- [108] Peikert, R. and Sadlo, F. Topologically Relevant Stream Surfaces for Flow Visualization. In Hauser, H., editor, *Proceedings of Spring Conference on Computer Graphics*, pages 43–50, April 2009. 27, 91

- 
- [109] Perry, A. E. and Chong, M. S. A description of eddying motions and flow patterns using critical-point concepts. *Annual Review of Fluid Mechanics*, 19:125–155, 1987. 29, 40
- [110] Porteous, I. R. *Geometric Differentiation*. Cambridge Univ. Press, Cambridge, UK, 1994. 43
- [111] Post, F., Vrolijk, B., Hauser, H., Laramée, R., and Doleisch, H. Feature Extraction and Visualization of Flow Fields. In *Eurographics 2002 State-of-the-Art Reports*, pages 69–100, Saarbrücken Germany, 2002. 42
- [112] Post, F., Vrolijk, B., Hauser, H., Laramée, R., and Doleisch, H. The State of the Art in Flow Visualization: Feature Extraction and Tracking. *Computer Graphics Forum*, 22(4)(2):775–792, 2003. 30, 42
- [113] Post, F. H., Post, F. J., Walsum, T. V., and Silver, D. Iconic Techniques for Feature Visualization. In *VIS '95: Proceedings of the 6th conference on Visualization '95*, page 288, Washington, DC, USA, 1995. IEEE Computer Society. 20, 42
- [114] Robinson, S. K. Coherent Motions in the Turbulent Boundary Layer. *Annu. Rev. Fluid Mech.*, 23:601–639, 1991. 14, 125
- [115] Roth, M. *Automatic Extraction of Vortex Core Lines and Other Line-Type Features for Scientific Visualization*. PhD thesis, Swiss Federal Institute of Technology, ETH Zürich, 2000. 46, 115
- [116] Roth, M. and Peikert, R. A Higher-Order Method for Finding Vortex Core Lines. In *VIS '98: Proceedings of the conference on Visualization '98*, pages 143–150, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press. 44
- [117] Ruetten, M. and Chong, M. Analyzing Vortex Breakdown Flow Structures by Assignment of Colors to Tensor Invariants. *IEEE Transactions on Visualization and Computer Graphics*, 12, no. 5:1189–1196, 2006. 95
- [118] Sadlo, F. and Peikert, R. Efficient Visualization of Lagrangian Coherent Structures by Filtered AMR Ridge Extraction. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1456–1463, September-October 2007. 3
- [119] Sadlo, F. and Peikert, R. Visualizing Lagrangian Coherent Structures and Comparison to Vector Field Topology. In Hege, H.-C., Polthier, K., and Scheuermann, G., editors, *Topology-Based Methods in Visualization II (Proceedings of Topology-Based Methods in Visualization 2007)*, pages 15–30. Springer, 2008. 3, 43, 117, 126
- [120] Sadlo, F., Peikert, R., and Parkinson, E. Vorticity Based Flow Analysis and Visualization for Pelton Turbine Design Optimization. In *Proceedings of IEEE Visualization*, pages 179–186, 2004. 2

- [121] Sadlo, F., Peikert, R., and Sick, M. Visualization Tools for Vorticity Transport Analysis in Incompressible Flow. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):949–956, 2006. 3, 64
- [122] Sadlo, F., Rigazzi, A., and Peikert, R. Time-Dependent Visualization of Lagrangian Coherent Structures by Grid Advection. In Pascucci, V., Tricoche, X., Hagen, H., and Tierny, J., editors, *Topological Methods in Data Analysis and Visualization (Proceedings of Topology-Based Methods in Visualization 2009)*. To appear. Springer, 2011. 3
- [123] Saffman, P. G. *Vortex Dynamics*. Cambridge Univ. Press, Cambridge, UK, 1992. 93, 96, 101, 102
- [124] Sahner, J., Weinkauff, T., and Hege, H.-C. Galilean Invariant Extraction and Iconic Representation of Vortex Core Lines. In K. Brodlie, D. Duke, K. J., editor, *Proc. Eurographics / IEEE VGTC Symposium on Visualization (EuroVis '05)*, pages 151–160, Leeds, UK, June 2005. 44, 115
- [125] Sahner, J., Weinkauff, T., Teuber, N., and Hege, H.-C. Vortex and Strain Skeletons in Eulerian and Lagrangian Frames. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):980–990, 2007. 43
- [126] Salzbrunn, T., Garth, C., Scheuermann, G., and JoergMeyer. Pathline predicates and unsteady flow structures. *The Visual Computer*, 2008. 27
- [127] Schafhitzel, T., Vollrath, J., Gois, J., Weiskopf, D., Castelo, A., and Ertl, T. Topology-Preserving  $\lambda^2$ -based Vortex Core Line Detection for Flow Visualization. *Computer Graphics Forum (Eurovis 2008)*, 27(3):1023–1030, 2008. 44
- [128] Schafhitzel, T., Tejada, E., Weiskopf, D., and Ertl, T. Point-based stream surfaces and path surfaces. In *GI '07: Proceedings of Graphics Interface 2007*, pages 289–296, New York, NY, USA, 2007. ACM. 28
- [129] Scheuermann, G., Hagen, H., Krüger, H., Menzel, M., and Rockwood, A. P. Visualization of higher order singularities in vector fields. In *Proceedings of IEEE Visualization*, pages 67–74, 1997. 30
- [130] Schindler, B., Fuchs, R., Biddiscombe, J., and Peikert, R. Predictor-Corrector Schemes for Visualization of Smoothed Particle Hydrodynamics Data. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1243–1250, 2009. 45
- [131] Schlemmer, M., Hotz, I., Hamann, B., Morr, F., and Hagen, H. Priority Streamlines: A context-based Visualization of Flow Fields. In *EG/IEEE VGTC Symposium on Visualization (EuroVis '07)*, pages 227–234, 2007. 25
- [132] Schultz, T., Theisel, H., and Seidel, H.-P. Topological Visualization of Brain Diffusion MRI Data. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1496–1503, 2007. 39

- 
- [133] Serrat, J., Lopez, A., and Lloret, D. On Ridges and Valleys. *IPCR*, 04:4059, 2000. 42
- [134] Shadden, S. C., Lekien, F., and Marsden, J. E. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Physica D Nonlinear Phenomena*, 212:271–304, December 2005. 40, 42, 43, 138
- [135] Shi, K., Theisel, H., Weinkauff, T., Hauser, H., Hege, H.-C., and Seidel, H.-P. Path Line Oriented Topology for Periodic 2D Time-Dependent Vector Fields. In *Proc. Eurographics / IEEE VGTC Symposium on Visualization (EuroVis '06)*, pages 139–146, Lisbon, Portugal, May 2006. 126
- [136] Sil'nikov, L. P. A case of the existence of a denumerable set of periodic motions. *Sov. Math. Dokl.*, 6:163–166, 1965. 94
- [137] Silver, D. and Wang, X. Volume Tracking. In *Proceedings of IEEE Visualization*, pages 157–164, 1996. 15, 51
- [138] Sotiropoulos, F., Ventikos, Y., and Lackey, T. C. Chaotic advection in three-dimensional stationary vortex-breakdown bubbles: Sil'nikov's chaos and the devil's staircase. *J. Fluid Mech.*, 444:257–297, 2001. 94, 101
- [139] Spohn, A., Mory, M., and Hopfinger, E. Experiments on vortex breakdown in a confined flow generated by a rotating disc. *Journal of Fluid Mechanics*, 370:73–99, 1998. 94, 98, 101
- [140] Stalling, D., Zöckler, M., and Hege, H.-C. Fast Display of Illuminated Field Lines. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):118–128, 1997. 31
- [141] Steger, C. Subpixel-Precise Extraction of Watersheds. In *ICCV (2)*, pages 884–890, 1999. 43
- [142] Stegmaier, S., Rist, U., and Ertl, T. Opening the Can of Worms: An Exploration Tool for Vortical Flows. In C. Silva and E. Gröller and H. Rushmeier, editor, *Proceedings of IEEE Visualization '05*, pages 463–470. IEEE, 2005. 44
- [143] Strawn, R. C., Kenwright, D. N., and Ahmad, J. Computer Visualization of Vortex Wake Systems. *American Helicopter Society 54th Annual Forum*, May 1998. 19, 43, 44
- [144] Suerich-Gulick, F., Gaskin, S., Villeneuve, M., and Parkinson, E. Experimental and numerical analysis of free surface vortices at a hydropower intake. In *Proceedings of the 7th International Conference on Hydrosience and Engineering*, pages 1–11, September 2006. 134

- [145] Sujudi, D. and Haimes, R. Identification of swirling flow in 3D vector fields. In *12th AIAA Computational Fluid Dynamics Conference*, pages 95–1715, 1995. 15, 44, 45, 50, 74
- [146] Surana, A., Grunberg, O., and Haller, G. Exact theory of threedimensional flow separation. Part I. Steady separation. *J. Fluid Mech.*, 564:57–103, 2006. 90
- [147] Surana, A., Jacobs, G., and Haller, G. Extraction of Separation and Reattachment Surfaces from 3D Steady Shear Flows. *AIAA Journal*, 45(6):1290–1302, 2007. 137
- [148] Theisel, H., Sahner, J., Weinkauff, T., Hege, H.-C., and Seidel, H.-P. Extraction of Parallel Vector Surfaces in 3D Time-Dependent Fields and Application to Vortex Core Line Tracking. In *Proceedings of IEEE Visualization*, pages 631–638, Minneapolis, U.S.A., October 2005. 45
- [149] Theisel, H. and Seidel, H.-P. Feature flow fields. In *VISSYM '03: Proceedings of the Symposium on Data Visualisation 2003*, pages 141–148, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 30, 42, 44
- [150] Theisel, H., Weinkauff, T., Hege, H.-C., and Seidel, H.-P. Saddle Connectors - An Approach to Visualizing the Topological Skeleton of Complex 3D Vector Fields. In *Proceedings of IEEE Visualization*, pages 225–232, 2003. 27, 37
- [151] Theisel, H., Weinkauff, T., Hege, H.-C., and Seidel, H.-P. Stream Line and Path Line Oriented Topology for 2D Time-Dependent Vector Fields. In *Proceedings of IEEE Visualization*, pages 321–328, 2004. 126
- [152] Theisel, H., Weinkauff, T., Hege, H.-C., and Seidel, H.-P. 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. 30
- [153] Thompson, M. C. and Hourigan, K. The sensitivity of steady vortex breakdown bubbles in confined cylinder flows to rotating lid misalignment. *Journal of Fluid Mechanics*, 496:129–138, December 2003. 94, 102
- [154] Thürey, N., Sadlo, F., Schirm, S., Müller-Fischer, M., and Gross, M. H. Real-time simulations of bubbles and foam within a shallow water framework. In *Symposium on Computer Animation*, pages 191–198, 2007. 96
- [155] Toth, G. The  $\text{div } \mathbf{B}=0$  Constraint in Shock-Capturing Magnetohydrodynamics Codes. *Journal of Computational Physics*, 161:605–652, 2000. 109
- [156] Tóth, G. and Roe, P. L. Divergence- and curl-preserving prolongation and restriction formulas. *J. Comput. Phys.*, 180(2):736–750, 2002. 109
- [157] Trichoche, X., Scheuermann, G., and Hagen, H. Vector and Tensor Field Topology Simplification on Irregular Grids. In *Data Visualization 2001 - Proceedings of the*

- Joint Eurographics - IEEE TCVG Symposium on Visualization in Ascona, Switzerland*, D. Ebert, J. M. Favre, R. Peikert (eds.), Springer, Wien, 2001., pages 107–116, 0 2001. 39, 50
- [158] Tricoche, X., Garth, C., Kindlmann, G., Deines, E., Scheuermann, G., Ruetten, M., and Hansen, C. Visualization of Intricate Flow Structures for Vortex Breakdown Analysis. In *Proceedings of IEEE Visualization*, pages 187–194. IEEE Computer Society, October 2004. 95
- [159] Tricoche, X., Garth, C., and Scheuermann, G. Fast and Robust Extraction of Separation Line Features. In Bonneau, G. P., Ertl, T., and Nielson, G. M., editors, *Scientific Visualization: The Visual Extraction of Knowledge from Data*, Berlin, 2005. Springer. 140
- [160] Tricoche, X., Scheuermann, G., and Hagen, H. Higher Order Singularities in Piecewise Linear Vector Fields. In *IMA Conference on the Mathematics of Surfaces*, pages 99–113, 2000. 30
- [161] Tricoche, X., Scheuermann, G., and Hagen, H. A topology simplification method for 2D vector fields. In *Proceedings of IEEE Visualization*, pages 359–366, 2000. 39
- [162] Tricoche, X., Scheuermann, G., and Hagen, H. Topology-Based Visualization of Time-Dependent 2D Vector Fields. In *Data Visualization 2001, Proc. VisSym 01*, pages 117–126. Springer Verlag, 2001. 33
- [163] Tricoche, X., Wischgoll, T., Scheuermann, G., and Hagen, H. Topology tracking for the visualization of time-dependent two-dimensional flows. *Computers & Graphics*, 26(2):249–257, 2002. 30
- [164] Turk, G. and Banks, D. Image-Guided Streamline Placement. In *SIGGRAPH*, pages 453–460, 1996. 24, 25, 26, 49, 53
- [165] Ueng, S.-K., Sikorski, C., and Ma, K.-L. Efficient Streamline, Streamribbon, and Streamtube Constructions on Unstructured Grids. *IEEE Transactions on Visualization and Computer Graphics*, 2(2):100–110, 1996. 70
- [166] van Wijk, J. J. Implicit stream surfaces. In *VIS '93: Proceedings of the 4th conference on Visualization '93*, pages 245–252, Washington, DC, USA, 1993. IEEE Computer Society. 28, 115
- [167] Ventikos, Y. The effect of imperfections on the emergence of three-dimensionality in stationary vortex breakdown bubbles. *Physics of Fluids*, pages L13–L16, 2002. 94
- [168] Verma, V., Kao, D. T., and Pang, A. A flow-guided streamline seeding strategy. In *Proceedings of IEEE Visualization*, pages 163–170, 2000. 25, 54

- [169] VirCinity. <http://www.vircinity.com>, 2004. 49
- [170] Ward, M. O. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3/4):194–210, 2002. 21
- [171] Weickert, J., Ishikawa, S., and Imiya, A. Linear Scale-Space has First been Proposed in Japan. *J. Math. Imaging Vis.*, 10(3):237–252, 1999. 45
- [172] Weinkauff, T., Hege, H.-C., Noack, B., Schlegel, M., and Dillmann, A. Coherent Structures in a Transitional Flow around a Backward-Facing Step. *Physics of Fluids*, 15(9):S3, September 2003. 31
- [173] Weinkauff, T., Theisel, H., Shi, K., Hege, H.-C., and Seidel, H.-P. Extracting Higher Order Critical Points and Topological Simplification of 3D Vector Fields. In *Proceedings of IEEE Visualization*, pages 559–566, Minneapolis, U.S.A., October 2005. 30
- [174] Weinkauff, T., Sahner, J., and Theisel, H. Cores of Swirling Particle Motion in Unsteady Flows. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1759–1766, 2007. 45
- [175] Weinstein, D. M., Kindlmann, G. L., and Lundberg, E. C. Tensorlines: Advection-Diffusion based Propagation through Diffusion Tensor Fields. In *Proceedings of IEEE Visualization*, pages 249–253, 1999. 24
- [176] Weiss, T. The dynamics of enstrophy transfer in two-dimensional hydrodynamics. *Physica D Nonlinear Phenomena*, 48:273–294, 1991. 17
- [177] Westermann, R., Johnson, C., and Ertl, T. A level-set method for flow visualization. In *Proceedings of IEEE Visualization*, pages 147–154, 2000. 28
- [178] Wiebel, A., Tricoche, X., Schneider, D., Jaenicke, H., and Scheuermann, G. Generalized Streak Lines: Analysis and Visualization of Boundary Induced Vortices. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1735–1742, 2007. 23
- [179] Wischgoll, T. and Scheuermann, G. Locating closed streamlines in 3D vector fields. In *Data Visualisation 2002*, pages 227–232, 2002. 33, 105
- [180] Wischgoll, T. and Scheuermann, G. Detection and Visualization of Closed Streamlines in Planar Flows. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):165–172, 2001. 33
- [181] Witkin, A. Scale-space filtering. In *Proc. Eighth Int. Joint Conf. on Artificial Intelligence*, pages 1019–1022, 1983. 45
- [182] Xue, D., Zhang, C., and Crawfis, R. Rendering Implicit Flow Volumes. In *Proceedings of IEEE Visualization*, pages 99–106, Washington, DC, USA, 2004. IEEE Computer Society. 28

- [183] Ye, X., Kao, D., and Pang, A. Strategy for Scalable Seeding of 3D Streamlines. In *Proceedings of IEEE Visualization*, pages 471–478, 2005. 25
- [184] Yoden, S. and Nomura, M. Finite-Time Lyapunov Stability Analysis and Its Application to Atmospheric Predictability. *Journal of the Atmospheric Sciences*, 50(11):1531–1543, 1993. 126
- [185] Zhang, S., Çagatay Demiralp, and Laidlaw, D. H. Visualizing Diffusion Tensor MR Images Using Streamtubes and Streamsurfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(4):454–462, 2003. 25
- [186] Zheng, X., Parlett, B., and Pang, A. Topological Structures of 3D Tensor Fields. *Proceedings of IEEE Visualization*, pages 551–558, 2005. 39
- [187] Zheng, X. and Pang, A. Topological Lines in 3D Tensor Fields. In *Proceedings of IEEE Visualization*, pages 313–320, Washington, DC, USA, 2004. IEEE Computer Society. 39



# Index

- abrasion, 13
- acceleration, **11**, 12, 17
- alpha-limit set, **37**
- angular velocity, 12
- antisymmetric part, **12**
- attachment
  - line, **45**, **46**, 88, 130
  - point, **45**
  - surface, 28
- bifurcation, **29**
- boundary condition
  - no-slip, **16**, 85, 88
- boundary shear layer, 16
- boundary shear, 64
- cantor, 94
- cavitation, **13**
- cell, **9**
  - alignment, 10
- centrifugal force, 18
- characteristic equation, **17**
- Chernoff faces, 20
- circulation, 14, 63
- closed orbit, **33**
- computational fluid dynamics, **1**
  - CFD, **1**
- computer-generated imagery, 13
  - CGI, 13
- connected components, 15
- constrained structural stability,  
**86**, 87
- continuity
  - $C^0$ , 8
  - $C^1$ , 8
  - differentiable, 85
  - Lipschitz, **8**, 23, 85
- contour, 20
  - extraction, 11
- coplanar vector, 45
- Courant number, 69, 77
- critical point, 20, 25, **30**, 85
  - 2D center, **30**, 86
  - 2D focus sink, **31**
  - 2D focus source, **31**
  - 2D node sink, **31**
  - 2D node source, **31**
  - 2D saddle, **31**, 42, 86
  - 2D sink, 86
  - 2D source, 86
  - 3D 1-saddle, **32**
  - 3D 1-spiral saddle, **32**
  - 3D 2-saddle, **32**, 89
  - 3D 2-spiral saddle, **32**, 89
  - 3D sink, **32**
  - 3D source, **32**
  - 3D spiral sink, **32**
  - 3D spiral source, **32**
  - hyperbolic, **30**
  - locally structurally stable,  
**30**
- cross product, 153
- curl, 7, 12, 18, 153
- data
  - cell-based, **9**
  - higher order, **9**
  - mesh-less, **9**
  - node-based, **9**
  - point-based, **9**
  - reconstruction, **9**

- scattered, **9**
- degenerate point, **25**
- density, 12, 17, 153
- determinant, 86, 153
- diffusion
  - momentum, 13
- direct numerical simulation, **13**
  - DNS, **13**
- Discontinuous Galerkin, **9**
- discrepancy, 66
- dissipation, 7
- divergence, 7, 17, 153
- domain, **8**
- dot product, 153
- dynamical systems, 37
  
- eddy, 17
- eigenvalues, **17**
- eigenvector field line, **24**
- Einstein notation, **8**
- Euclidean norm, 127, 153
- extended phase space, **29**
- extremum line, 44
  
- feature flow field, 30, 42, 44, 45, 115
- feature, **42**
- field line, **22**
- finite-size Lyapunov exponent
  - FSLE, **128**
- finite-time Lyapunov exponent, 2, 115
  - FTLE, **40, 127**
  - maximum, **129**
- first recurrence map, **33**
- first return map, **33**
- flow map, **127**
- flow
  - attachment, 89, 138
  - Hagen-Poiseuille, 16, **50**
  - incompressible, 17
  - laminar, 13
  - separation, 89, 93, 138
  - turbulent, 13
  
- flux, 53, **86**
- folding, **98**
- frame of reference, 19
  
- Gauss' theorem, 53
- glyph placement, **20**
- gradient, 153
- grid, **10**
  - adaptive, **10**
  - advection, 143
  - AMR, **10, 119**
  - Cartesian, **10**
  - curvilinear, **10**
  - hybrid, **10**
  - irregular, **10**
  - multi, **10**
  - rectilinear, **10**
  - scattered, **10**
  - structured, **10**
  - uniform, **10**
  - unstructured, **10, 10**
  
- hedgehog, **20**
- height ridge, **42, 42**
- Helmholtz theorems, 28, **63**
- Hessian, 18, 153
- hyperbolicity time, 20, 28
- hyperstreamline, **25, 39**
  - helix, **25**
  - tube, **25**
  
- index invariance, **29**
- information visualization, 20
- initial value problem, 21
- integral curve
  - computational space, 11
- interpolation
  - barycentric, 11
  - cubic, 9
  - mesh-less, **9**
  - radial basis functions, 9
  - splines, 9
  - Taylor, 9
- invariant torus, 93
- invariant

- manifold, **36**, **37**
  - torus, **37**, **87**
- islands of stability, **96**
- isosurface, **20**, **49**, **51**
- Jacobian, **11**
- KAM torus, **94**
- Kolmogorov
  - cascade, **13**
  - microscale, **13**
- Lagrangian coherent structures,
  - 2**, **115**
  - LCS, **40**
- Laplacian, **17**, **153**
- least squares, **117**
- limit cycle, **38**
- linked views, **64**
- local section, **33**
- magnetic field, **12**, **53**
- manifold
  - stable, **36**
  - unstable, **36**
- marching cubes, **20**, **116**
- marching ridges, **115**
- material derivative, **11**
- material line
  - attracting, **40**
  - repelling, **40**
- matrix norm
  - Frobenius, **17**
  - spectral, **127**
- matrix transpose, **153**
- maximum curvature ridge, **43**
- meshing, **9**
- mmap(), **128**
- modulus, **153**
- Navier-Stokes, **9**, **12**, **17**, **64**
  - DNS, **67**
- Newtonian fluid, **88**
- node, **9**
  - hanging, **10**
- T, **10**
- non-photorealistic rendering, **43**
- objective, **19**
- omega-limit set, **37**
- orbit, **37**
  - constant, **30**
  - heteroclinic, **37**
  - homoclinic, **37**
- ordinary differential equation,
  - 21**
  - autonomous, **22**
- orientation, **86**
- parallel vectors, **44**, **115**
- path line, **2**, **22**, **64**
- path surface, **28**
- periodic orbit, **33**, **37**, **85**, **86**
  - center, **87**, **93**
  - hyperbolic, **34**, **87**
  - saddle, **34**, **87**
  - sink, **34**, **86**, **93**
  - source, **34**, **86**, **93**
  - spiral sink, **34**, **87**, **93**
  - spiral source, **34**, **87**, **93**
  - twisted saddle, **34**, **87**
- permutation symbol, **12**
- phase space, **29**
- Poincaré map, **33**, **86**
- Poincaré section, **33**
- Poincaré-Bendixson theorem, **38**
- Poincaré-Hopf index, **29**
- point location, **11**, **25**
- post-processing, **27**
- pressure minima, **18**
- pressure, **17**, **153**
  - corrected, **18**
  - gradient, **12**
  - modified, **67**
  - reduced, **18**
- principal component analysis,
  - 117**
- profile ridge, **42**
- rake, **73**

- $(\omega, u)$ , **73**
- $(u, \omega)$ , **73**, **78**
- rate of strain, **12**, **17**, **18**, **67**
- S**, **12**
- recirculation zone, **85**
- resonance, **13**
- Reynolds number, **13**
- ribbon, **28**
- ridge cell, **120**
- ridge lines, **44**
- ringing, **9**
- rotation
  - rigid body, **14**
- saddle connector, **95**
- saddle connectors, **37**
- scalar field topology, **42**
- scale space, **45**
- scientific visualization, **20**
- second derivative ridge, **42**
- separation
  - closed, **46**, **90**
  - line, **45**, **46**, **88**, **130**
  - open, **46**, **90**
  - point, **45**
  - surface, **28**, **85**
- separatrix, **36**
- Shilnikov chaos, **37**, **93**
- smoke ring, **15**, **93**
- solenoidal, **86**
- space
  - computational, **11**
  - physical, **11**
- stationary point, **30**
  - isolated, **30**
- streak line, **23**
  - generalized, **23**
- streak surface, **28**
- stream surface, **25**, **27**
- stream tube, **25**, **28**, **50**
- streamline, **2**, **14**, **22**
  - placement, **25**, **49**, **54**
  - regular, **34**
  - unique, **23**, **85**
- subdivision, **10**
- substantial derivative, **11**
- symmetric part, **12**
- tensor field topology, **25**, **39**
  - degenerate point, **39**
  - trisector, **39**
  - wedge, **39**
- tensor invariant, **17**, **17**, **19**
  - P**, **17**
  - Q**, **17**
  - R**, **17**, **19**
- tensor line, **24**
- texture advection, **15**
- thinning, **99**
- time line, **23**
- time surface, **28**
- topological skeleton, **36**
- triangulation, **9**
- turbine
  - Francis, **75**
  - Pelton, **49**
- turbulence model, **7**, **13**, **64**, **67**
- turbulent kinetic energy, **20**
- valley line, **116**
- vector field topology, **2**, **25**, **27**, **29**, **63**
- vector field, **8**
  - static, **8**
  - stationary, **8**
  - steady-state, **8**
  - time-dependent, **8**
  - transient, **8**
  - unsteady, **8**
- velocity, **11**, **153**
  - gradient, **11**, **18**
- viscosity, **7**
  - dynamic, **13**, **88**, **154**
  - eddy, **67**
  - kinematic, **13**, **17**, **154**
  - modified, **67**
  - turbulent, **67**
- volume rendering, **20**

- vortex breakdown bubble, 93  
vortex breakdown, 15, 85, **94**  
vortex core line, 15, 19, 20, **43**,  
50  
topological, **37**, 43, 50  
vortex core, **14**, 14, 44, 50, 52  
detection, 15  
vortex criterion  
 $M_z$ , **19**, 28  
 $\Delta$ , **19**  
 $\lambda_2$ , 15, **18**, 44, 52  
Galilean invariant, 19  
helicity density, **19**  
helicity, **19**, 50–52, 58, 59  
normalized helicity, **19**  
 $Q$ , 14, **17**, 19  
vortex strength, **19**  
vortex dynamics, 12, 64  
vortex hull, 52  
vortex line, 16, **23**, 49, 50, 52, 63  
vortex sheet, 57, 59, 64  
roll-up, 57, 59  
vortex stretching, 64  
vortex tilting, 64, 72  
vortex  
forced, **14**  
free, **14**  
hairpin, **15**  
intensification, 64  
irrotational, **14**  
line, 94  
longitudinal, 15  
Rankine, **14**, 17, 18  
ring, 15, **93**  
rotational, **14**  
separation, 50, 57, 59  
streamline, 50  
stretching, 18, **65**, 75  
tilting, 18, **65**, 75  
transversal, 15  
tube, **28**, 53, 63  
tumble, 15  
von Kármán street, **15**  
vortex line, 50  
weakening, 64  
vorticity equation, 18, **64**, 64  
vorticity, 2, **12**, 15, 49, 153  
 $\Omega$ , **12**  
diffusion, 18  
streamlet, 72  
tensor, **12**, 17, 18  
wall distance, 68, 72, 88  
wall shear stress, 85  
watercourse, **42**  
watershed, 36, **42**



# Curriculum Vitae

## Filip Sadlo

born 27. February 1972 in Zurich, Switzerland  
citizen of Switzerland

## Education

- 2005 - 2008 Ph.D. student at the Computer Graphics Laboratory, ETH Zurich, Switzerland
- 2004 - 2005 Development of a Structured Light Scanner, ETH World Project “3-D Webinhalte”, Computer Graphics Laboratory, ETH Zurich, Switzerland
- Sep. 2003 *Diplom* in Computer Science (Dipl. Inform.)
- 1996 - 2003 Studies in Computer Science, ETH Zurich, Switzerland
- 1993 - 1996 Basic studies in Electrical Engineering (untested), ETH Zurich, Switzerland
- Aug. 1991 *Matura* Typus B
- 1985 - 1991 Realgymnasium Rämibühl, Zurich, Switzerland

## Professional Experience

- 2003 Internship at IBM Zurich Research Laboratory (Pervasive Computing Group), Rüschlikon, Switzerland
- 1997 - 2003 Sadlo Software Development (one-man company), Zurich, Switzerland
- Anatomisches Institut, University of Bern
    - Research and development in Medical Visualization
    - Consulting
  - Bike Tree (parking system for bicycles)
    - Research and development
    - Automation, integration, mechatronics
- 1993 - 1996 Medical Visualization, Nationales Zentrum für Retroviren, Zurich, Switzerland
- 1989 - 1992 Library, Institut für Mikrobiologie und Virologie, University of Zurich, Switzerland

## Publications

- Sadlo, F., Rigazzi, A., and Peikert, R. Time-Dependent Visualization of Lagrangian Coherent Structures by Grid Advection. In *Topological Methods in Data Analysis and Visualization (Proceedings of Topology-Based Methods in Visualization 2009)*, to appear. Springer, 2011.
- Peikert, R. and Sadlo, F. Topologically Relevant Stream Surfaces for Flow Visualization. In *Proceedings of Spring Conference on Computer Graphics*, pages 43–50, 2009.
- Fuchs, R., Peikert, R., Sadlo, F., Alsallakh, B., and Gröller, M. E. Delocalized Unsteady Vortex Region Detectors. In *Proceedings VMV 2008*, pages 81–90, 2008.
- Peikert, R. and Sadlo, F. Height Ridge Computation and Filtering for Visualization. In *Proceedings of IEEE VGTC Pacific Visualization Symposium 2008*, pages 119–126, 2008.
- Fuchs, R., Peikert, R., Hauser, H., Sadlo, F., and Muigg, P. Parallel Vectors Criteria for Unsteady Flow Vertices. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):615–626, 2008.

- 
- Peikert, R. and Sadlo, F. Visualization Methods for Vortex Rings and Vortex Breakdown Bubbles. In *Proceedings of EuroVis 2007*, pages 23–25, 2007.
- Peikert, R. and Sadlo, F. Flow Topology Beyond Skeletons: Visualization of Features in Recirculating Flow. In *Topology-Based Methods in Visualization II (Proceedings of Topology-Based Methods in Visualization 2007)*, pages 145–160. Springer, 2008.
- Sadlo, F. and Peikert, R. Visualizing Lagrangian Coherent Structures and Comparison to Vector Field Topology. In *Topology-Based Methods in Visualization II (Proceedings of Topology-Based Methods in Visualization 2007)*, pages 15–30. Springer, 2008.
- Thürey, N., Sadlo, F., Schirm, S., Müller-Fischer, M., and Gross, M. H. Real-time simulations of bubbles and foam within a shallow water framework. In *Symposium on Computer Animation*, pages 191–198, 2007.
- Sadlo, F. and Peikert, R. Efficient Visualization of Lagrangian Coherent Structures by Filtered AMR Ridge Extraction. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1456–1463, 2007.
- Sadlo, F., Peikert, R., and Sick, M. Visualization Tools for Vorticity Transport Analysis in Incompressible Flow. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):949–956, 2006.
- Waschbüsch, M., Würmlin, S., Cotting, D., Sadlo, F., and Gross, M. H. Scalable 3D video of dynamic scenes. *The Visual Computer*, 21(8-10):629–638, 2005.
- Sadlo, F., Weyrich, T., and Peikert, R. A Practical Low-Cost Scanner for Geometry and Appearance. In *Point-Based Graphics*, Elsevier, pp. 49-66, 2007.
- Peikert, R. and Sadlo, F. Topology-guided Visualization of Constrained Vector Fields. In *Topology-Based Methods in Visualization (Proceedings of Topology-Based Methods in Visualization 2005)*, pages 21–34. Springer, 2007.
- Sadlo, F., Weyrich, T., Peikert, R., and Gross, M. A practical structured light acquisition system for point-based geometry and texture. In *Proceedings of Point-based Graphics*, 0:89–145, 2005.
- Mallo, O., Peikert, R., Sigg, C., and Sadlo, F. Illuminated Lines Revisited. In *Proceedings of IEEE Visualization*, pages 19–26, 2005.
- Sadlo, F., Peikert, R., and Parkinson, E. Vorticity Based Flow Analysis and Visualization for Pelton Turbine Design Optimization. In *Proceedings of IEEE Visualization*, pages 179–186, 2004.
- Filip Sadlo. Three-Dimensional Volume Reconstruction of Rice Roots. *Diploma Thesis*, ETH Zurich, 2003.

Bron, C., Sadlo, F., and Burri, P. Microtomography and 3D skeleton analysis of vascular and airway trees. *Focus on Microscopy*, 2003.

Bron, C., Sadlo, F., Székely, G., W., Neuenschwander, Kübler, O., and Schüpbach, J. Segmentation and visualisation of membranes and intracellular organelles contours in 3D electron microscopy. In *SPIE, Visualization in Biomedical Computing*, volume 2359, pages 706–714, 1994.