Extraction of Dynamical Coherent Structures from Time-Varying Data Sets

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES OF ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

Alireza Hadjighasem
M.Eng., McGill University
born on 25.05.1988
citizen of IRAN

accepted on the recommendation of

Prof. George Haller
Prof. Markus Holzner
Prof. Konstantinos Boulouchos

2016
Dedicated with love to my parents

Without their support and sacrifices none of this would be possible
Acknowledgement

This document, and my progress as a doctoral student, have been supported and enriched by many mentors, friends, and family, to whom I am extremely grateful. I would like to begin and express my innermost gratitude to my supervisor, Prof. George Haller. I am indebted to him for his patience, kind words, thoughtful comments, and the compassionate and ethical manner which he embodies and models to his students. His positive influence on my professional and personal development will continue to benefit me throughout life. I am also grateful to Prof. Markus Holzner for kindly accepting to be the co-supervisor of my thesis.

I would also like to thank my friends, Mattia Serra, Mohammad Farazmand, David Oettinger, and other friends for their kindness and friendship over this journey. Your friendship, encouragement and support have been invaluable to me.

Last and foremost, I am grateful to my parents who have supported me through all the stages of my life so far and continue to believe in my abilities and my perseverance.

Zürich, May 2016

Alireza Hadjighasem
Abstract

Most real-world time-dependent dynamical systems have no classic phase space dividing barriers, such as periodic and quasiperiodic orbits, or homoclinic and heteroclinic trajectories. Nevertheless, the absence of mass exchange and distinct dynamical interactions across different regions of the phase space suggest the existence of transport barriers. In a Lagrangian (i.e., material) sense, these barriers can be classified as hyperbolic, elliptic and parabolic barriers depending on the type of influence they exert on nearby trajectories. For example, hyperbolic transport barriers are attracting or repelling material surfaces that act as the centerpieces of mixing patterns. In contrast, elliptic and parabolic transport barriers obstruct mixing by restraining coherent regions of the flow that resist stretching. These structures have collectively become known as Lagrangian Coherent Structures (LCS).

There has been a growing interest in a systematic and accurate detection of LCS in real-life flow data. A large number of available LCS detection methods, however, can only uncover transport barriers when the velocity vector field is known in detail for all times or has a well-defined temporal recurrence (e.g. periodic, quasi-periodic). These methods are of limited use for studying geophysical flows, in which the available data is temporally aperiodic and often spatio-temporally sparse. Moreover, LCS methods can be computationally expensive or rely on extensive user inputs, which makes them impractical for real-time analysis or automatic structure detection in large-scale data sets.

My Ph.D. research has focused on addressing these challenges by extending LCS methods to scenarios where direct measurement of the velocity field is not feasible. I have also undertaken the development new mathematical methods for automated LCS extraction.
Sommario

La maggior parte problemi reale tempo dipendente e di conseguenza lo sono anche i sistemi dinamici che modellano tali fenomeni. I sistemi dinamici tempo varianti non hanno barriere di trasporto nel senso classico come orbite periodiche o quasi periodiche, traiettorie omocliniche o eterocliniche. Si osserva per anche in questi sistemi assenza di trasporto di massa tra differenti regioni dello spazio delle fasi, suggerendo l’esistenza di tali barriere. Da un punto di vista Lagrangiano, queste barriere possono essere divise in iperboliche, ellittiche e paraboliche in base all’effetto che hanno sulle particelle vicine. Ad esempio, barriere di trasporto iperboliche sono superfici materiali attrattive o repulsive responsabili dei processi di mescolamento. Al contrario, barriere di trasporto ellittiche e paraboliche inibiscono il mescolamento e rimangono strutture coerenti nel fluido che resistono al campo di deformazione imposto dallo stesso. Queste strutture sono ora conosciute come Strutture Lagrangiane Coerenti o Lagrangian Coherent Structures (LCS). Si sviluppato un interesse crescente per localizzare in modo sistematico ed accurato tali strutture in applicazioni reali. La maggior parte dei metodi per l’identificazione di LCS, in grado di identificare tali strutture solo quando il campo di velocità del sistema in esame conosciuto per tempo infinito o ha particolari ricorrenze temporali (i.e., in sistemi periodici o quasi periodici). Tali metodi, per, sono di scarsa utilit per lo studio di flussi geofisici, in cui i dati disponibili sono non periodici nel tempo e sparsi nello spazio. Infine, i metodi per l’identificazione di LCS sono computazionalmente costosi o basati su abbondanti input da parte dell’utilizzatore rendendoli, di conseguenza, inadatti per analisi in tempo reale o automatizzate per grandi domini. La ricerca che ho condotto durante il mio dottorato si focalizzata nel risolvere questi svantaggi estendendo l’applicabilit dei metodi per la localizzazione di LCS in condizioni in cui misure dirette dei campi di velocità non sono praticabili e sviluppando nuovi metodi matematici per l’identificazione di LCS in modo automatico. slatio
# Contents

## 1 Introduction
1.1 Introduction ................................................................. 1
1.2 Summary ................................................................. 5

## 2 Detecting Invariant Manifolds, Attractors, and Generalized KAM Tori in Aperiodically Forced Mechanical Systems 
2.1 Set-up ................................................................. 9
  2.1.1 Geodesic transport barriers in phase space .................. 11
  2.1.2 Computation of invariant sets as transport barriers .......... 14
2.2 Results ................................................................. 17
  2.2.1 Proof of concept: Periodically forced Duffing oscillator .... 17
  2.2.2 The aperiodically forced Duffing oscillator ................. 23
2.3 Summary and conclusions .................................................. 27

## 3 Geodesic Transport Barriers in Jupiter's Atmosphere: A Video-Based Analysis 
3.1 Introduction ................................................................. 30
3.2 Set-up ................................................................. 32
3.3 Geodesic LCS theory ...................................................... 33
  3.3.1 Parabolic LCSs ...................................................... 33
  3.3.2 Elliptic LCSs ...................................................... 37
3.4 Unsteady transport barriers in the atmosphere of Jupiter .......... 39
  3.4.1 Prior work and present objectives .............................. 39
  3.4.2 Video footage ...................................................... 39
  3.4.3 Optical velocity field reconstruction ......................... 40
  3.4.4 Validation of the reconstructed velocity field ............... 45
  3.4.5 Lagrangian advection ............................................. 46
  3.4.6 Unsteady zonal jet cores as parabolic LCSs ................. 46
  3.4.7 The Great Red Spot as a generalized KAM region .......... 48
3.5 Summary and conclusions ........................................ 50

4 Spectral-Clustering Approach to Lagrangian Vortex Detection .... 52
  4.1 Introduction .................................................. 52
  4.2 Method ...................................................... 55
    4.2.1 Input: A trajectory data set .......................... 55
    4.2.2 Similarity graph construction ......................... 55
    4.2.3 Graph sparsification .................................. 57
    4.2.4 Spectral clustering .................................... 57
    4.2.5 Estimating the number of clusters by eigenspace analysis ... 60
    4.2.6 Retrieving clusters from matrix $U$ by K-means clustering ... 60
    4.2.7 Large-scale spectral clustering ....................... 61
  4.3 Related previous work ....................................... 63
    4.3.1 The transfer-operator approach ....................... 63
    4.3.2 Hierarchical partitioning of the transfer-operator ...... 66
    4.3.3 Application of Infomap to the transfer operator ....... 66
    4.3.4 Application of fuzzy clustering to a trajectory data set ... 67
  4.4 Results .................................................... 68
    4.4.1 The periodically forced pendulum ...................... 69
    4.4.2 Quasiperiodic Bickley jet ............................. 72
    4.4.3 An ocean surface data set ............................. 73
    4.4.4 The ABC flow .......................................... 77
  4.5 Summary and conclusions ................................... 80
  4.A Approximating Ncut ......................................... 81
  4.B Bipartite spectral graph partitioning ....................... 82

5 Level Set Formulation of Two-dimensional Lagrangian Vortex Detection Methods .... 84
  5.1 Background ................................................ 84
    5.1.1 Implicit boundary representation ..................... 84
    5.1.2 Front evolution and level set theory .................. 85
  5.2 Variational level-set-based vortex detection ................ 86
    5.2.1 Stretching-based formulation .......................... 87
    5.2.2 Graph-based formulation ............................... 88
  5.3 Numerical results ........................................... 90
    5.3.1 Periodically forced pendulum ......................... 92
    5.3.2 Jupiter's wind-velocity field ......................... 94
    5.3.3 An ocean surface data set ............................. 96
5.4 Conclusion ................................................................. 98
5.A Numerical aspects ..................................................... 98
5.B Sparse Field Method .................................................. 100

6 A Comparison of Lagrangian Methods for Coherent Structure Detection 102
6.1 Diagnostics for Lagrangian coherence ............................... 102
   6.1.1 Finite-time Lyapunov exponent (FTLE) ......................... 103
   6.1.2 Finite-Size Lyapunov Exponent (FSLE) ......................... 104
   6.1.3 Mesochronic analysis ............................................. 104
   6.1.4 Trajectory length ................................................ 105
   6.1.5 Trajectory complexity .......................................... 106
   6.1.6 Shape coherence ................................................. 106
6.2 Mathematical approaches to Lagrangian coherence .................. 107
   6.2.1 Geodesic theory of LCS ....................................... 107
   6.2.2 Rotationally coherent Lagrangian vortices ..................... 109
   6.2.3 Coherent pairs of sets ......................................... 112
   6.2.4 Spectral clustering of trajectories .............................. 113
6.3 Method comparisons on three examples ............................. 115
   6.3.1 Bickley jet ...................................................... 116
   6.3.2 Two-dimensional turbulence ................................... 123
   6.3.3 Wind field from Jupiter’s atmosphere .......................... 129
6.4 Summary and conclusions ............................................ 131

7 Bibliography .................................................................. 133
Chapter 1

Introduction

1.1 Introduction

Lagrangian Coherent structures (LCSs) are ubiquitous in oceanic and atmospheric flows. These coherent structures are time-evolving material surfaces along which passive tracers (e.g. dye, temperature, or any material with negligible effect on the flow) develop distinct patterns over times exceeding typical time scales in the flow. These surfaces split the phase-space into regions with dynamically distinct properties, and provide a strong tool for understanding different dynamical behaviors, such as mixing and transport.

Lagrangian coherent structures emerge in nature frequently, and play an important role in various physical processes, such as massive transport of salinity and temperature from the Indian Ocean to the Atlantic Ocean [16, 102, 22]; flourishment of harmful algae in the ocean [147]; swirling flow patterns in blood vessels [159]; volcanic ash transport in the atmosphere; formation of long-lived cloud patterns on giant planets (see fig. 1.1 for examples). These examples in turn have triggered the development of novel Lagrangian mathematical approaches that can identify transport barriers within complex unsteady flow fields, in addition to assessing the role that such structures play in the overall mixing and transport. Moreover, Lagrangian coherent structure detection methods provide predictive capabilities, identifying robust structures that are too strong to be destroyed by short-term fluctuations in the flow field [155]. The predictive value of LCS tools can be applied in predicting regions that pollutants, including oil and radioactive material, most likely contaminate after environmental disasters [142, 149, 155]; identifying the so-called great garbage patches in the ocean [21]; controlling flow separation [188, 191, 177]; predicting wind shear for aviation safety [180]; enhancing search-and-rescue operations in the ocean by predicting the probable path of missing persons or vessels at sea [4].

Transport and mixing in fluid flows are governed by two fundamental physical processes:
advection and molecular diffusion. The advection process is responsible for the stretching and folding of a passive scalar field, while the diffusion smooths out the spatial inhomogeneity of the field. The diffusion process, however, only acts over relatively long periods compared with the advection timescale, making its effect negligible over short time intervals. Hence, the mixing and transport of passive tracers can be described on the basis of advection alone over a short-to-intermediate time intervals \([t_0, t_1]\). In that case, trajectories of passive tracers coincide with the fluid particle trajectories, which are solutions of the differential equation

\[
\dot{x} = v(x, t), \quad x \in \mathbb{R}^n, \quad t \in [t_0, t_1],
\]

where \(v(x, t)\) is the velocity of the fluid defined in two- or three-dimensional space.

Equation (1.1) defines a dynamical system, which consequently makes a large class of techniques in this area applicable to the study of mixing and transport phenomena in the context of fluid flows. For instance, for simple time periodic velocity-fields, transport barriers can be uncovered through appropriate Poincaré maps, which reveal them as stable and unstable manifolds or Kolmogorov–Arnold–Moser (KAM) tori that either enhance or inhibit mixing. The major difficulty, however, is that the velocity field (1.1) often has arbitrary time-dependence and may only be available for a finite-time interval. In this case, classic phase space barriers such as stable and unstable manifolds or KAM tori become undefined, because classic dynamical systems tools, such as Poincaré maps, become inapplicable for exploring such structures. Instead, one has to rely on the state-of-the-art Lagrangian methods developed for velocity fields \(v(x, t)\) that are temporally aperiodic and available only for finite times. In this thesis, we develop these latter methods further, and try to address some outstanding challenges in the finite-time coherent structure detection.

To apply dynamical systems tools to the study of transport phenomena, we invariably rely on
a velocity field $v(x, t)$ to obtain particle trajectories through eq. (1.1). The velocity field may be given analytically, or produced by numerical simulations (e.g., of the Navier–Stokes equations), or observed through in-situ measurements or remote sensing. Whether obtained through numerical simulation or observations, the resulting velocity field $v(x, t)$ is spatio-temporally sparse. The majority of classic dynamical systems methods, however, can only uncover transport barriers when the velocity vector field has a high spatio-temporal resolution. These methods, therefore, have limited application to geophysical flows, in which only a scarce knowledge of the velocity field is available. Moreover, these methods can be computationally expensive or rely on extensive user inputs, which makes them impractical for the purpose of real-time or automatic structure detection in large-scale data sets.

A number of different Lagrangian methods have been developed for the detection of coherence in the fluid flows over a finite-time interval. These methods can roughly be divided into three major categories:

**Geometric approaches** seek distinguished surfaces that divide the fluid domain into regions with qualitatively distinct behaviors. These dividing surfaces depending on the type of influence they exert on nearby fluid particles can be put into one of the three basic categories: Hyperbolic, elliptic, and parabolic transport barriers. Out of these barriers, hyperbolic barriers are time-evolving material surfaces that act as generalized stable and unstable manifolds, repelling or attracting neighboring material elements with locally the highest rate over a finite-time interval [98, 66]. Elliptic barriers (i.e., material vortex boundaries) are closed material surfaces that serve as KAM-type transport barrier, prevailing coherence by inhibiting particle mixing between interior and exterior fluid regions (cf. [101, 102, 146]). Finally, parabolic barriers (i.e., generalized jet stream cores) are material surfaces along which Lagrangian shear is minimal [65]. A comprehensive review of geometric approaches is provided in [99, 92].

**Set-based approaches** target the interiors of coherent flow regions, as opposed to the boundaries encompassing these regions. These approaches are probabilistic in nature and often rely on some concepts of dimension reduction which aims to transform the high-dimensional data space to a low dimensional representation that reveals coherent structures. Examples of set-based approaches include transfer operator-based methods for detecting almost-invariant sets [49, 53, 75] in time-periodic flows, and coherent sets [80, 78] in aperiodic flows; Koopman operator-theoretic methods [32, 31] for time-periodic flows, and trajectory clustering approaches [79, 93] for aperiodic flows.

**diagnostic tools** offer simple commutable scalar fields whose spatial distributions arguably enhance the visualization of coherent structures (see [9, 104, 142], for example). These
methods serve as practical tools for an initial investigation, before applying more rigorous approaches from the first two categories mentioned above.

In the present work, we apply and develop methods that fall in the first two categories. Specifically, in chapters 2 and 3 we adopt geometric approaches to detect coherent structures in the context of mechanical systems and planetary atmospheres, respectively. In chapter 4, we develop a new set-based approach for detecting coherent structures in sparse geophysical data sets. In chapter 4 we introduce a new numerical technique that enables a large class of geometric approaches to detect an a priori unknown number of coherent structures in a fluid domain automatically at a low computational cost. Finally, in chapter 6 we compare different Lagrangian methods, from all the three categories surveyed above. In section 1.2 we review the content of each chapter in more detail.
1.2 Summary

Chapter 2 demonstrates how the concept of Lagrangian coherent structures, originally developed for fluid flows, can be used to study mechanical systems. Specifically, we adapt the theory of geodesic transport barriers [101] to reveal finite-time invariant sets in one-degree-of-freedom mechanical systems that externally perturbed with time-dependent forcing. Invariant sets in such systems turn out to be shadowed by minimal geodesics of the Cauchy-Green (CG) strain tensor, which can be computed from the CG tensor invariants. This approach enables the detection of finite-time invariant sets such as stable and unstable manifolds, attractors and generalized KAM curves under general forcing conditions, when Poincaré maps cannot be constructed anymore (see fig. 1.2). This chapter contains slight modifications compared to the study published in [91].

In chapter 3, we uncover, for the first time, time-dependent transport barriers that form the cores of jet streams and the boundary of the Great Red Spot (GRS), i.e., the largest and longest-living known atmospheric vortex, in Jupiter’s atmosphere. To uncover these coherent structures, we first reconstruct a two-dimensional velocity field from a video footage captured by the NASA Cassini spacecraft. Next, we apply the theory of geodesic transport barriers [102, 65] to uncover transport barriers in Jupiter’s atmosphere. In a broader perspective, our approach provides a numerical platform for the systematic extraction of finite-time coherent structures from time-varying remote observations of unsteady continua (see fig. 1.3). This chapter contains slight modifications compared to the study published in [92].
In chapter 4 we develop an approach that identifies coherent structures by clustering trajectories into coherent and incoherent families. To identify coherent structures, our approach first constructs an undirected weighted graph describing the proximity of Lagrangian particles in the spatio-temporal domain. It then identifies coherent structures by partitioning the graph into sets of distinguished Lagrangian particles that are tightly packed among themselves, yet stay remote from the rest of the particles in the computational domain. An advantage of this approach is that it requires a relatively low number of Lagrangian particles as input, making it suitable for sparse geophysical data sets. Moreover, it locates all coherent vortices in the flow simultaneously, which enables it as a potential tool for automated vortex census and tracking (see fig. 1.4).

In chapter 5 we develop a new numerical technique based on the variational level set methodology to detect coherent Lagrangian vortices in two-dimensional unsteady fluid flows. Specifically, we identify coherent vortices by optimizing appropriate energy functionals whose local minima mark the boundaries of Lagrangian vortices. We carry out the optimization step using a combination of variational calculus and a gradient-descent method. We demonstrate the performance of this technique for two different variational formulations that utilize different notions of coherence. The first formulation uses an energy functional that characterizes the coherent vortices as closed material lines whose pieces have small variations in terms of stretching, a view motivated by [102]. The second formulation is a graph-based approach, which seeks
coherent vortices based on the proximity of Lagrangian particles in the spatio-temporal domain. The latter approach is an extension of the method presented in chapter 4. The proposed technique, more broadly, can reformulate any variational LCS method that seeks coherent structures as a stationary solution of a variational problem (see fig. 1.5).

In chapter 6 we examine a number of different approaches that have been developed in recent years for the detection of coherence in the Lagrangian framework. These methods can be put into one of the two basic categories: diagnostic tools and mathematical approaches. The diagnostic tools usually propose a scalar quantity whose features is expected to indicate transport barriers of the flow. These approaches, however, do not offer physical definitions for a coherent structure, or a procedure for structure extraction from the generated scalar field. In contrast, mathematical approaches identify coherent structures precisely based on physical coherence principles that can be formulated analytically. These mathematical approaches may, nevertheless, provide different solutions to the same coherent structure detection, as they rely on different or even discrepant notions of coherence. Here, we hope to resolve these discrepancies by comparing various methods with each other. Specifically, we compare the performance of eleven methods, consisting of both diagnostic and mathematical approaches, for several two-dimensional flows that range from analytic velocity fields to time-dependent observational data set.

A list of publications on which the upcoming chapters are based is as follows:

Chapter 2:


My contributions to this chapter were helping to design and implement the numerical algorithms; carrying out the LCS computations; and helping to write the paper.

Chapter 3:


Chapter 4:


My main contributions to this paper were initiating and developing the research; providing all the numerical results; and writing the draft. All authors discussed the results and implications and commented on the manuscript at all stages.


My main contributions to this draft were participating in developing the ideas, and discussing the material a number of times with the authors.

Chapter 5:


Chapter 6:


My main contributions to this chapter include coding all the methods, preparing the draft and running the computations to produce the results of all LCS methods, with the following exceptions:

(i) The second author (D. Blazevski) has carried out the computation of the geodesic method for the Bickley jet flow using his code.

(ii) The third author (M. Farazmand) has performed the computation of the geodesic method for his turbulence data set using his code.

All individuals listed as authors have contributed to the design of numerical examples.

Supplemental materials related to this thesis can be found at the following address:
http://www.zfm.ethz.ch/~hadjighasem/.


My main contributions to this paper include helping to develop the LAVD boundary extraction algorithms; publishing the LAVD toolbox [GitHub Link]; producing the results of all LAVD computations, except for the ABC flow example; and helping to edit the paper.
Chapter 2

Detecting Invariant Manifolds, Attractors, and Generalized KAM Tori in Aperiodically Forced Mechanical Systems

Chapter Abstract

In this chapter, we show how the theory of geodesic transport barriers for fluid flows can be used to uncover key invariant manifolds in externally forced, one-degree-of-freedom mechanical systems. Specifically, invariant sets in such systems turn out to be shadowed by least-stretching geodesics of the Cauchy-Green strain tensor computed from the flow map of the forced mechanical system. This approach enables the finite-time visualization of generalized stable and unstable manifolds, attractors and generalized KAM curves under arbitrary forcing, when Poincaré maps are not available. We illustrate these results by detailed visualizations of the key finite-time invariant sets of conservatively and dissipatively forced Duffing oscillators.

2.1 Set-up

The key invariant sets of autonomous and time-periodic dynamical systems—such as fixed points, periodic and quasiperiodic motions, their stable and unstable manifolds, and attractors—are typically distinguished by their asymptotic properties. In contrast, invariant sets in finite-time, aperiodic dynamical systems solely distinguish themselves by their observed impact on trajectory patterns over the finite time interval of their definition. This observed impact is a pronounced lack of trajectory exchange (or transport) across the invariant set, which remains coherent in time, i.e., only undergoes minor deformation. Well-understood, classic examples of such transport barriers include local stable manifolds of saddles, parallel shear jets, and KAM tori of time-periodic conservative systems. Until recently, a common dynamical feature of these
barriers has not been identified, hindering the unified detection of transport barriers in general non-autonomous dynamical systems.

As noted recently in [101], however, a common feature of all canonical transport barriers in two dimensions is that they stretch less under the flow than neighboring curves of initial conditions do. This observation leads to a nonstandard calculus of variations problem with unknown endpoints and a singular Lagrangian. Below we recall the solution of this problem from [101], with a notation and terminology adapted to one-degree-of-freedom mechanical oscillators.

A one-degree-of-freedom forced nonlinear oscillator can generally be written as a two-dimensional dynamical system

$$\dot{x} = v(x, t), \quad x \in U \subset \mathbb{R}^2, \quad t \in [t_0, t_1],$$

(2.1)

with $U$ denoting an open set in the state space, where the vector $x$ labels tuples of positions and velocities. The vector $v(x, t)$, assumed twice continuously differentiable, contains the velocity and acceleration of the system at state $x$ and at time $t$.

Let $x(t_1; t_0, x_0)$ denote the final state of system ((2.1)) at time $t_1$, given its state $x_0$ at an initial time $t_0$. The flow map associated with ((2.1)) over this time interval is defined as

$$F_{t_0}^{t_1}: x_0 \mapsto x(t_1; t_0, x_0),$$

(2.2)

which maps initial states to final states at $t_1$. The Cauchy–Green (CG) strain tensor associated with the flow map ((2.2)) is defined as

$$C_{t_0}^{t_1}(x_0) = [DF_{t_0}^{t_1}(x_0)]^\top DF_{t_0}^{t_1}(x_0),$$

(2.3)

where $DF_{t_0}^{t_1}$ denotes the gradient of the flow map ((2.2)), and the symbol $\top$ refers to matrix transposition.

Note that the CG tensor is symmetric and positive definite. As a result, it has two positive eigenvalues $0 < \lambda_1 \leq \lambda_2$ and an orthonormal eigenbasis $\{\xi_1, \xi_2\}$. We fix this eigenbasis so that

$$C_{t_0}^{t_1}(x_0)\xi_i(x_0) = \lambda_i(x_0)\xi_i(x_0), \quad |\xi_i(x_0)| = 1, \quad i \in \{1, 2\},$$

$$\xi_2(x_0) = \Omega \xi_1(x_0), \quad \Omega = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$  

(2.4)

We suppress the dependence of $\lambda_i$ and $\xi_i$ on $t_0$ and $t_1$ for notational simplicity.
2.1.1 Geodesic transport barriers in phase space

A material line $\gamma_t = F^t_{t_0}(\gamma_{t_0})$ is an evolving curve of initial conditions $\gamma_{t_0}$ under the flow map $F^t_{t_0}$. As shown in [101], for such a material line to be a locally least-stretching curve over $[t_0, t_1]$, it must be a hyperbolic, a parabolic or an elliptic line (see figure [2.1]).

The initial position $\gamma_{t_0}$ of a hyperbolic material line is tangent to the vector field $\xi_1$ at all its points. Such material lines are compressed by the flow by locally the largest rate, while repelling all nearby material lines at an exponential-in-time rate. The classic example of a hyperbolic material lines is the unstable manifold of a saddle-type fixed point.

A parabolic material line is an open material curve whose initial position $\gamma_{t_0}$ is tangent to one of the directions of locally largest shear. At each point of the phase space, the two directions of locally largest shear are given by

$$\eta_{\pm} = \sqrt{\frac{\lambda_2}{\lambda_1 + \lambda_2}} \xi_1 \pm \sqrt{\frac{\lambda_1}{\lambda_1 + \lambda_2}} \xi_2,$$

as derived in [101]. Parabolic material lines still repel most nearby material lines (except for those parallel to them), but only at a rate that is linear in time. Classic examples of parabolic material lines in fluid mechanics are the parallel trajectories of a steady shear flow.

Finally, an elliptic material line is a closed curve whose initial position $\gamma_{t_0}$ is tangent to one of the two directions of locally largest shear given in (2.5). As a result, elliptic lines also repel nearby, nonparallel material lines at a linear rate, but they also enclose a connected region. Classic examples of elliptic material lines are closed trajectories of a steady, circular shear flow, such as a vortex.

Initial positions of hyperbolic material lines are, by definition, strainlines, i.e., trajectories of the autonomous differential equation

$$r' = \xi_1(r), \quad r \in U \subset \mathbb{R}^2,$$

where $r : [0, \ell] \mapsto U$ is the parametrization of the strainline by arc-length. A hyperbolic barrier is then a strainline that is locally the closest to least-stretching geodesics of the CG tensor, with the latter viewed as a metric tensor on the domain $U$ of the phase space. The pointwise closeness of strainlines to least-stretching geodesics can be computed in terms of the invariants of the CG strain tensor. Specifically, the $C^2$ distance (difference of tangents plus difference of curvatures) of a strainline from the least-stretching geodesic of $C^t_{t_0}$ through a point $x_0$ is given by the geodesic strain deviation.
Chapter 2. Manifolds, Attractors and KAM Surfaces in Aperiodic Systems

(a) A repelling hyperbolic barrier (red curve) repels nearby trajectories (gray blob) exponentially fast in time.

(b) A parabolic barrier (red curve) is an open curve that has the locally largest rate of Lagrangian shear along its tangent.

(c) An elliptic barrier (red curve) is a closed curve with the same dynamical property as a parabolic barrier.

Figure 2.1: The three types of transport barriers in two-dimensional flows.
Chapter 2. Manifolds, Attractors and KAM Surfaces in Aperiodic Systems

\[ d_{g_{1}}^{\xi_{1}}(x_{0}) = \frac{|(\nabla \lambda_{2}, \xi_{2}) + 2\lambda_{2}\kappa_{1}|}{2\lambda_{2}^3}, \quad (2.7) \]

with \( \kappa_{1}(x_{0}) \) denoting the curvature of the strainline through \( x_{0} \) [101]. A hyperbolic barrier is a compact strainline segment on which \( d_{g_{1}}^{\xi_{1}} \) is pointwise below a small threshold value, and whose averaged \( d_{g_{1}}^{\xi_{1}} \) value is locally minimal relative to all neighboring strainlines.

Similarly, initial positions of parabolic and elliptic material lines are, by definition, shearlines, i.e., trajectories of the autonomous differential equation

\[ r' = \eta_{\pm}(r), \quad r \in U \subset \mathbb{R}^2. \quad (2.8) \]

A \textit{parabolic barrier} is an open shearline that is close to least-stretching geodesics of the CG tensor. The pointwise \( C^2 \)-closeness of shearlines to least-stretching geodesics is given by the \textit{geodesic shear deviation}

\[ d_{g_{1}}^{\eta_{\pm}}(x_{0}) = \frac{\sqrt{1 + \lambda_{2}} - \sqrt{\lambda_{1}}}{\sqrt{1 + \lambda_{2}}} + \frac{(\nabla \lambda_{2}, \xi_{1})}{2\lambda_{2}\sqrt{1 + \lambda_{2}}} \geq -\frac{\langle \nabla \lambda_{2}, \xi_{2} \rangle (\sqrt{1 + \lambda_{2}^3} - \sqrt{\lambda_{2}^5})}{2\lambda_{2}^3\sqrt{1 + \lambda_{2}^3}} \]

\[ + \frac{\kappa_{1} \left[ \sqrt{\lambda_{2}^5} + (1 - \lambda_{2}^2) \sqrt{1 + \lambda_{2}} \right]}{\lambda_{2}^2\sqrt{1 + \lambda_{2}}} + \frac{\kappa_{2}}{\sqrt{1 + \lambda_{2}}}, \quad (2.9) \]

with \( \kappa_{2}(x_{0}) \) denoting the curvature of the \( \xi_{2} \) vector field at the point \( x_{0} \) [101]. The geodesic shear deviation should pointwise be below a small threshold level for an open shearline to qualify as a parabolic barrier. Similarly, a closed shearline is an \textit{elliptic barrier} if its pointwise geodesic shear deviation is smaller than small threshold level.

For the purposes of the present discussion, we call a mechanical system of the form \( (2.1) \) \textit{conservative} if it has vanishing divergence, i.e., \( \nabla \cdot v(x, t) = 0 \), with \( \nabla \) referring to differentiation with respect to \( x \). This property implies that flow map of \( (2.1) \) conserves phase-space area for all times [7].

While a typical material line in such a conservative system will still stretch and deform significantly over time, the length of a shearline will always be preserved under the area-preserving flow map \( F_{t_{1}}^{t_{0}} \) (cf. [101]). An elliptic barrier in a conservative system will, therefore, have the same enclosed area and arclength at the initial time \( t_{0} \) and at the final time \( t_{1} \). These two conservation properties imply that an elliptic barrier in a non-autonomous conservative system may only undergo translation, rotation and some slight deformation, but will otherwise preserve its overall shape. As a result, the interior of an elliptic barrier will not mix with the rest of the phase-space, making elliptic barriers the ideal generalized KAM curves in aperiodically forced
conservative mechanical systems.

2.1.2 Computation of invariant sets as transport barriers

In this section, we describe numerical algorithms for the extraction of hyperbolic and elliptic barriers in a one-degree-of-freedom mechanical system with general time dependence. Parabolic barriers can in principle also exist in mechanical systems, but they do not arise in the simple examples we study below. In contrast, parabolic barriers are more common in geophysical fluid mechanics where they typically represent unsteady shear jets.

Our numerical algorithms require a careful computation of the CG tensor. In most mechanical systems, trajectories separate rapidly, resulting in an exponential growth in the entries of the CG tensor. This growth necessitates the use of a well-resolved grid, as well as the deployment of high-end integrators in solving for the trajectories of (2.1) starting from this grid. Further computational challenges arise from the handling of the unavoidable orientational discontinuities and isolated singularities of the eigenvector fields $\xi_1$ and $\xi_2$. The reader is referred to Farazmand & Haller [66] for a detailed treatment of these computational aspects.

As a zeroth step, we fix a sufficiently dense grid $G_0$ of initial conditions in the phase-space $U$, then advect the grid points from time $t_0$ to time $t_1$ under system (2.1). This gives a numerical representation of the flow map $F_{t_0}^{t_1}$ over the grid $G_0$. The CG tensor field $C_{t_0}^{t_1}$ is then obtained by definition (2.3)) from $F_{t_0}^{t_1}$. In computing the gradient $DF_{t_0}^{t_1}$, we use careful finite differencing over an auxiliary grid, as described in [66].

Since, at each point $x_0 \in G_0$, the tensor $C_{t_0}^{t_1}(x_0)$ is a two-by-two matrix, computing its eigenvalues $\{\lambda_1, \lambda_2\}$ and eigenvectors $\{\xi_1, \xi_2\}$ is straightforward. With the CG eigenvalues and eigenvectors at hand, we locate the hyperbolic barriers using the following algorithm.
Algorithm 1 Locating hyperbolic barriers

1. Fix a small positive parameter $\epsilon_{\xi_1}$ as the admissible upper bound for the point-wise geodesic strain deviation of hyperbolic transport barriers.

2. Calculate strainlines by solving the ODE (2.6) numerically, with linear interpolation of the strain vector field between grid points. Truncate strainlines to compact segments whose pointwise geodesic strain deviation is below $\epsilon_{\xi_1}$.

3. Locate hyperbolic barriers as strainline segments $\gamma_{t_0}$ with locally minimal relative stretching, i.e., strainline segments that locally minimize the function

$$q(\gamma_{t_0}) = \frac{l(\gamma_{t_1})}{l(\gamma_{t_0})}, \quad (2.10)$$

Here $l(\gamma_{t_0})$ and $l(\gamma_{t_1})$ denote the length of the strainline $\gamma_{t_0}$ and the length of its advected image $\gamma_{t_1}$, respectively.

Computing the relative stretching (2.10) of a strainline $\gamma_{t_0}$, in principle, requires advecting the strainline to time $t_1$. However, as shown in [101], the length of the advected image satisfies $l(\gamma_{t_1}) = \int_{\gamma_{t_0}} \sqrt{\lambda_1} \, ds$, where the integration is carried out along the strainline $\gamma_{t_0}$. This renders the strainline advection unnecessary.

Numerical experiments have shown that a direct computation of $\xi_1$ is usually less accurate than that of $\xi_2$ due to the attracting nature of strongest eigenvector of the CG tensor [66]. For this reason, computing $\xi_1$ as an orthogonal rotation of $\xi_2$ is preferable. Moreover, it has been shown [67] that strainlines can be computed more accurately as advected images of stretchlines, i.e., curves that are everywhere tangent to the second eigenvector of the backward-time CG tensor $C_{t_0}^{t_1}$. In the present chapter, this approach is taken for computing the strainlines.

Computing elliptic barriers amounts to finding limit cycles of the ODE (2.8). To this end, we follow the approach used in [101, 67] by first identifying candidate regions for shear limit cycles visually, then calculating the Poincaré map on a one-dimensional section transverse to the flow within the candidate region (see figure 2.2). Hyperbolic fixed points of this map can be located by iteration, marking limit cycles of the shear vector field (see [67] for more detail).

This process is used in the following algorithm to locate elliptic barriers.

In the next section, we use the above algorithms for locating invariant sets in simple forced and damped nonlinear oscillators.
Algorithm 2 Locating elliptic barriers

1. Fix a small positive parameter $\varepsilon_{\eta\pm}$ as the admissible upper bound for the average geodesic shear deviation of elliptic transport barriers.

2. Visually locate the regions where closed shearlines may exist. Construct a sufficiently dense Poincaré map, as discussed above. Locate the fixed points of the Poincaré map by iteration.

3. Compute the full closed shearlines emanating from the fixed points of the Poincaré map.

4. Locate elliptic barriers as closed shearlines whose average geodesic deviation $\langle d_{g}^{\eta\pm} \rangle$ satisfies $\langle d_{g}^{\eta\pm} \rangle < \varepsilon_{\eta\pm}$.

Figure 2.2: Locating closed shearlines using a Poincaré section of the shear vector field. Closed shearlines pass through the fixed points of the corresponding Poincaré map.
2.2 Results

We demonstrate the implementation of the geodesic theory of transport barriers on four Duffing-type oscillators. As a proof of concept, in the first two examples (section 2.2.1), we consider \textit{periodically} forced Duffing oscillators for which we can explicitly verify our results using an appropriately defined Poincaré map.

The next two examples deal with \textit{aperiodically} forced Duffing oscillators (section 2.2.2). In these examples, despite the absence of a Poincaré map, we still obtain the key invariant sets as hyperbolic and elliptic barriers.

To implement algorithms 1 and 2 in the forthcoming examples, the CG tensor is computed over a uniform grid $G_0$ of $1000 \times 1000$ points. A fourth order Runge-Kutta method with variable step-size (ODE45 in MATLAB) is used to solve the first-order ODEs ((2.1)), ((2.6)) and ((2.8)) numerically. The absolute and relative tolerances of the ODE solver are set equal to $10^{-4}$ and $10^{-6}$, respectively. Off the grid points, the strain and shear vector fields are obtained by bilinear interpolation.

In each case, the Poincaré map of algorithm 2 is approximated by 500 points along the Poincaré section. The zeros of the map are located by a standard secant method.

2.2.1 Proof of concept: Periodically forced Duffing oscillator

Case 1: Pure periodic forcing, no damping

Consider the periodically forced Duffing oscillator

\[
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= x_1 - x_1^3 + \epsilon \cos(t).
\end{align*}
\]

For $\epsilon = 0$, the system is integrable with one hyperbolic fixed point at $(0,0)$, and two elliptic fixed points $(1,0)$ and $(-1,0)$, respectively. As is well known, there are two homoclinic orbits connected to the hyperbolic fixed point, each enclosing an elliptic fixed point, which is in turn surrounded by periodic orbits. These periodic orbits appear as closed invariant curves for the Poincaré map $P := F_0^{2\pi}$. The fixed points of the flow are also fixed points of $P$.

For $0 < \epsilon \ll 1$, the Kolmogorov–Arnold–Moser (KAM) theory \cite{7} guarantees the survival of most closed invariant sets for $P$. Figure 2.3 shows these surviving invariant sets (KAM curves) of $P$ obtained for $\epsilon = 0.08$. For the KAM curves to appear continuous-looking, nearly 500 iterations of $P$ were needed, requiring the advection of initial conditions up to time $t = 1000\pi$. The stochastic region surrounding the KAM curves is due to chaotic dynamics arising from the
transverse intersections of the stable and unstable manifold of the perturbed hyperbolic fixed point of \( P \).

The surviving KAM curves are well-known, classic examples of transport barriers. We would like to capture as many of them as possible as elliptic barriers using the geodesic transport theory described in previous sections. Note that not all KAM curves are expected to prevail as locally least-stretching curves for a given choice of the observational time interval \([t_0, t_1]\); some of these curves may take longer to prevail due to their shape and shearing properties.

We use the elliptic barrier extraction algorithm of section 2.1.2 with \( \varepsilon \eta_{\pm} = 0.7 \). Figure 2.4 shows the resulting shearlines in the KAM regions, with the closed ones marked by red. Note that these shearlines were obtained from the CG tensor computed over the time interval \([0, 8\pi]\), spanning just four iterations of the Poincare map. Despite this low number of iterations, the highlighted elliptic barriers are practically indistinguishable from the KAM curves obtained from five hundred iterations.

Figure 2.5 shows the convergence of an elliptic barrier to a KAM curve as the integration time \( T = t_1 - t_0 \) increases. Note how the average geodesic deviation \( \langle d_{\eta_{\pm}} \rangle \) decreases with increasing \( T \), indicating decreasing deviation from nearby Cauchy–Green geodesics.

Remarkably, constructing these elliptic barriers requires significantly shorter integration time (only four forcing periods) in comparison to visualization through the Poincaré map, which required 500 forcing periods to reveal KAM curves as continuous objects. Clearly, the overall computational cost for constructing elliptic barriers still comes out to be higher, since the CG tensor needs to be constructed on a relatively dense grid \( G_0 \), as discussed in section 2.1.2.
high computational cost will be justified, however, in the case of aperiodic forcing (section 2.2.2), where no Poincaré map is available.

Figure 2.5: Convergence of an elliptic barrier (red) to a KAM curve (black) as the integration time $T = t_1 - t_0$ increases. The gradually decreasing average geodesic deviation $\langle d_{\eta}^{\pm} \rangle$ confirms the convergence to Cauchy–Green geodesics that closely shadow the underlying KAM torus.

In the context of one-degree-of-freedom mechanical systems, the outermost elliptic barrier marks the boundary between regions of chaotic dynamics and regions of oscillations that are regular on a macroscopic scale. To demonstrate this sharp dividing property of elliptic barriers, we show the evolution of system (2.12) from three initial states, two of which are inside the elliptic region and one of which is outside (figure 2.6a). The system exhibits rapid changes in its state when started from outside the elliptic region. In contrast, more regular behavior is observed for trajectories starting inside the elliptic region. This behavior is further depicted in figure 2.6c, which shows the evolution of the $x_1$-coordinate of the trajectories as a function of
Figure 2.6: (a) The outermost elliptic barrier (black curve) and three initial conditions: Two inside the elliptic barrier (blue and green) and one outside the elliptic barrier (red). (b) The corresponding trajectories are shown in the extended phase space of \((x_1, x_2, t)\). The closed black curves mark the elliptic barrier at \(t_0 = 0\) and \(t_1 = 16\pi\). (c) The \(x_1\)-coordinate of the trajectories of figure 2.6.

time.

**Case 2: Periodic forcing and damping**

Consider now the damped-forced Duffing oscillator

\[
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= x_1 - x_1^3 - \delta x_2 + \epsilon \cos(t),
\end{align*}
\]

(2.11)

with \(\delta = 0.15\) and \(\epsilon = 0.3\). This system is known to have a chaotic attractor that appears as an invariant set of the the Poincaré map \(P = F_0^{2\pi}\) (see, e.g., [88]). Here, we show that the attractor can be very closely approximated by hyperbolic barriers computed via algorithm [1].

Figure 2.7a shows strainlines computed backward in time with \(t_0 = 0\) and integration time \(T = t_1 - t_0 = -8\pi\). The strainline with globally minimal relative stretching (2.10) is shown in figure 2.7b. Black dots mark the points where the geodesic deviation \(d_{\xi_{i1}}\) exceeds the admissible upper bound \(\epsilon_{\xi_{i1}} = 10^{-3}\). At its tail (covered by black dots), the strainline persistently deviates from CG geodesics, and hence should be truncated. The resulting hyperbolic barrier, as a finite-time approximation to the chaotic attractor, is shown in figure 2.7c.

The approximate location of the attractor can also be revealed by applying the Poincaré map to a few initial conditions (tracers) released from the basin of attraction. For long enough advection time, the initial conditions converge to the attractor highlighting its position (see figure 2.8a and 2.8b). In figure 2.8c, the hyperbolic barrier is superimposed on the advected tracers showing close agreement between the two. Figure 2.8d shows the tracers advected for
Figure 2.7: Construction of the attractor of the damped-forced Duffing oscillator as a hyperbolic transport barrier. (a) Strainlines computed for the damped-forced Duffing oscillator (2.11) at time $t_0 = 0$, with the integration time $T = -8\pi$. (b) The strainline (red) with globally minimum relative stretching. Points with $d_{\xi} > 10^{-3}$ are highlighted as black dots. (c) Final approximation of the chaotic attractor by a single, continuous strainline with minimal geodesic deviation.
Figure 2.8: (a) Attractor of system (2.11) obtained from four iterates of the Poincaré map. (b) Attractor obtained from 20 iterates of the Poincaré map. (c) Attractor computed as a hyperbolic barrier (red), compared with the Poincaré map (blue) computed for the same integration time (four iterates). (d) Comparison of attractor computed as a hyperbolic barrier (red) with the one obtained from 20 iteration of the Poincaré map (blue). The integration time for locating the hyperbolic barrier is $T = t_1 - t_0 = -8\pi$. 
a longer time \((T = 40\pi)\) together with the hyperbolic barrier; the two virtually coincide. Note that the hyperbolic barrier is a smooth, parametrized curve (computed as a trajectory of (2.6)), while the tracers form a set of scattered points.

### 2.2.2 The aperiodically forced Duffing oscillator

In the next two examples, we study aperiodically forced Duffing oscillators. In the presence of aperiodic forcing, the Poincaré map \(P\) is no longer defined as the system lacks any recurrent behavior. However, KAM-type curves (i.e., closed curves, resisting significant deformation) and generalized stable and unstable manifolds (i.e., most repelling and attracting material lines) exist in the phase-space and determine the overall dynamics of the system.

**Case 1: Purely aperiodic forcing, no damping**

Consider the Duffing oscillator

\[
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= x_1 - x_1^3 + f(t),
\end{align*}
\]

(2.12)

where \(f(t)\) is an aperiodic forcing function obtained from a chaotic one-dimensional map (see figure 2.9).

While, KAM theory is no longer applicable, one may still expect KAM-type barriers to survive for small forcing amplitudes. Such barriers would no longer be repeating themselves periodically in the extended phase space. Instead, a generalized KAM barrier is expected to be an invariant cylinder, with cross sections showing only minor deformation. The existence of such structures can, however, be no longer studied via Poincaré maps.

Figure 2.10 confirms that generalized KAM-type curves, obtained as elliptic barriers, do exist in this problem. These barriers are computed over the time interval \([0, 4\pi]\) (i.e. \(t_0 = 0\) and \(t_1 = t_0 + T = 4\pi\)). As discussed in section 2.1.1, the arclength of an elliptic barrier at the initial time \(t_0\) is equal to the arclength of its advected image under the flow map \(F_{t_0}^{t_1}\) at the final time \(t_1\). This arclength preservation is illustrated numerically in figure 2.11, which shows the relative stretching,

\[
\delta \ell(t) = \frac{\ell(\gamma_t) - \ell(\gamma_0)}{\ell(\gamma_0)}
\]

(2.13)

of the time-\(t\) image \(\gamma_t\) of an elliptic barrier \(\gamma_0\), with \(\ell\) referring to the arclength of the curve. Ideally, the relative stretching of each elliptic barrier should be zero at time \(t_1 = 4\pi\), i.e. \(\delta \ell(4\pi) = 0\). Instead, we find that the relative stretching \(\delta \ell(4\pi)\) of the computed elliptic barriers is at most 1.5%. This deviation from zero arises from numerical errors in the computation of the CG strain.
Chapter 2. Manifolds, Attractors and KAM Surfaces in Aperiodic Systems

Figure 2.9: Chaotic forcing function $f(t)$ for equation (2.12).

Figure 2.10: Closed shearlines for equation (2.12) computed in two elliptic regions. The figure shows the shearlines at time $t_0 = 0$. The integration time is $T = 4\pi$.

Figure 2.11: The relative stretching $\delta\ell(t) \times 100$ of closed shearlines of figure 2.10. The colors correspond to those of figure 2.10. By their arc-length preservation property, the advected elliptic barriers must theoretically have the same arclength at times $t_0 = 0$ and $t_1 = 4\pi$. The numerical error in arclength conservation is small overall, but more noticeable for oscillations with large amplitudes (green and red curves of the right panel).
tensor $C_{t_0}^{t_1}$, which in turn causes small inaccuracies in the computation of closed shearlines.

As noted earlier, the small relative stretching and the conservation of enclosed area for an elliptic barrier in incompressible flow only allows for small deformations when the barrier is advected in time. This is illustrated in figure 2.12, which shows the blue elliptic barrier of figure 2.10b in the extended phase-space. Each constant-time slice of the figure is the advected image of the barrier.

Finally, we point out that the stability of the trajectories inside elliptic barriers show a similar trend as in the case of the periodically forced Duffing equation (figures 2.6 and 2.6c). Namely, perturbations inside the elliptic regions remain small while they grow significantly inside the hyperbolic regions.

**Case 2: Aperiodic forcing with damping**

In this final example, we consider the aperiodically forced, damped Duffing oscillator

\[
\begin{align*}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= x_1 - x_1^3 - \delta x_2 + f(t),
\end{align*}
\]  

(2.14)

with damping coefficient $\delta = 0.15$. The forcing function $f(t)$ is similar to that of Case I above, but with an amplitude twice as large. As a result, none of the elliptic barriers survive even in the absence of damping.

Again, because of the aperiodic forcing, the behavior of this system is a priori unknown and cannot be explored using Poincaré maps. In order to investigate the existence of an attractor, strainlines (figure 2.13a) are computed from the backward-time CG strain tensor $C_{t_0}^{t_1}$ with $t_0 = 30$ and $t_1 = 10$. The strainline with minimum relative stretching ((2.10)) is then extracted. The
Figure 2.13: (a) Strainlines computed in backward time from $t_0 = 30$ to $t_1 = 10$. (b) The resulting hyperbolic barrier extracted with maximum admissible geodesic deviation of $\epsilon_{t_1} = 10^{-5}$.

Figure 2.14: (a) Tracers advected over the time interval from $t_1 = 10$ to $t_0 = 30$. (b) Tracers advected over a longer time interval from $t_1 = 0$ to $t_0 = 30$. (c) The hyperbolic barrier (red) superimposed on the tracers advected for the same time interval (d) Comparison of the hyperbolic barrier (red) with the tracers advected for the longer time interval.
part of this strainline satisfying $d_{g}^{\xi_{1}} < \epsilon_{\xi_{1}}$ is considered as the most influential hyperbolic barrier (figure 2.13b). The admissible upper bound $\epsilon_{\xi_{1}}$ for the geodesic deviation is fixed as $10^{-5}$.

In order to confirm the existence of the extracted attractor, we advect tracer particles in forward time, first from time $t_{1} = 10$ to time $t_{0} = 30$, then from $t_{1} = 0$ to time $t_{0} = 30$. Because of the fast-varying dynamics and weak dissipation, a relatively long advection time is required for the tracers to converge to the attractor. Figure 2.14 shows the evolution of tracers over $[t_{1}, t_{0}]$. Note that the attractor inferred from the tracers is less well pronounced than the hyperbolic barrier extracted over the same length of time. This shows a clear advantage for geodesic transport theory over simple numerical experiments with tracer advection. For a longer integration time from $t_{0} = 0$ to $t = 30$, the tracers eventually converge to the hyperbolic barrier.

Repelling hyperbolic barriers can be computed similarly using forward-time computations. Figure 2.15 shows both hyperbolic barriers (stable and unstable manifolds) at time $t_{0} = 30$. The repelling barrier is computed from the CG strain tensor $C_{t_{0}}^{t_{1}}$ with $t_{0} = 30$ and $t_{1} = 50$.

### 2.3 Summary and conclusions

We have shown how the recently developed geodesic theory of transport barriers \[101\] in fluid flows can be adapted to compute finite-time invariant sets in one-degree-of-freedom mechanical systems with general forcing. Specifically, in the presence of general time dependence, temporally aperiodic stable- and unstable manifolds, attractors, as well as generalized KAM tori can be located as hyperbolic and elliptic barriers, respectively. The hyperbolic barriers are computed
as distinguished *strainlines*, i.e. material lines along which the Lagrangian strain is locally maximized. The elliptic barriers, on the other hand, appear as distinguished *shearlines*, i.e. material lines along which the Lagrangian shear is locally maximized. The barriers are finally identified as strainlines and shearlines that are most closely approximated by least-stretching geodesics of the metric induced by Cauchy–Green strain tensor.

We have used four simple examples for illustration. First, as benchmarks, we considered periodically forced Duffing equations for which stable and unstable manifolds, attractors and KAM curves can also be obtained as invariant sets of an appropriately defined Poincaré map. We have shown that elliptic barriers, computed as closed shearlines, coincide with the KAM curves. Also, stable and unstable manifolds, as well as attractors, can be recovered as hyperbolic barriers. More precisely, as the integration time \( T = t_1 - t_0 \) of the Cauchy–Green strain tensor \( C_{t_0}^{t_1} \) increases, the elliptic barriers in the periodically forced Duffing equations converge to KAM curves. Similarly, the chaotic attractor of the periodically forced and damped Duffing equation is more and more closely delineated by a hyperbolic barrier computed from the backward-time Cauchy–Green strain tensor \( C_{t_1}^{t_0} \) for increasing \( T = t_0 - t_1 \) where \( t_0 > t_1 \).

In the second set of examples, we have computed similar structures for an aperiodically forced Duffing oscillator with and without damping. In this case, Poincaré maps are no longer well-defined for the system, and hence we had to advect tracer particles to verify the predictions of the geodesic theory. Notably, tracer advection takes longer time to reveal the structures in full detail than the geodesic theory does. Also, tracer advection is only affective as a visualization tool if it relies on a small number of particles, which in turn assumes that one already roughly knows the location of the invariant set to be visualized. Finally, unlike scattered tracer points, geodesic barriers are recovered as parametrized smooth curves that provide a solid foundation for further analysis or highly accurate advection.

In our examples, elliptic barriers have shown themselves as borders of subsets of the phase-space that barely deform over time. In fact, as illustrated in figure 2.6, outermost elliptic barriers define the boundary between chaotic and regular dynamics. Trajectories initiated inside elliptic barriers remain confined and robust with respect to small perturbations. We believe that this property could be exploited for stabilizing mechanical systems with general time dependence. For instance, formulating an optimal control problem for generating elliptic behavior in a desired part of the phase-space is a possible approach.

Undoubtedly, the efficient and accurate computation of invariant sets as geodesic transport barriers requires dedicated computational resources. Smart algorithms reducing the computational cost are clearly of interest. Parallel programming (both at CPU and GPU levels) has previously been employed for Lagrangian coherent structure calculations and should be useful in the present setting as well (see e.g. [85]). Other adaptive techniques are also available to lower the
numerical cost by reducing the computations to regions of interest (see e.g. [15, 127]).

In principle, invariant sets in higher-degree-of-freedom mechanical systems could also be captured by similar techniques as locally least-stretching surfaces. The development of the underlying multi-dimensional theory and computational platform, however, is still underway.
Chapter 3

Geodesic Transport Barriers in Jupiter’s Atmosphere: A Video-Based Analysis

Chapter Abstract

Jupiter’s zonal jets and Great Red Spot are well known from still images. Yet the planet’s atmosphere is highly unsteady, which suggests that the actual material transport barriers delineating its main features should be time-dependent. Rare video footages of Jupiter’s clouds provide an opportunity to verify this expectation from optically reconstructed velocity fields. Available videos, however, provide short-time and temporally aperiodic velocity fields that defy classical dynamical systems analyses focused on asymptotic features. To this end, we use here the recent theory of geodesic transport barriers to uncover finite-time mixing barriers in the wind field extracted from a video captured by NASA’s Cassini space mission. More broadly, the approach described here provides a systematic and frame-invariant way to extract dynamic coherent structures from time-resolved remote observations of unsteady continua.

3.1 Introduction

Jupiter’s size is 1,300 times that of the Earth. Its mass is more than twice the mass of all planets in our solar system combined. Jupiter’s fast rotation — one in 10 hours — creates strong jet streams that smear its clouds into bands of zones and belts of almost constant latitude. Another frequently discussed feature of Jupiter is its Great Red Spot (GRS), the largest and longest-living known atmospheric vortex. The GRS is a nearly two-dimensional feature that is apparently unrelated to the topography of the planet [86]. Such vortices abound in nature, but GRS’s size, long-term persistence, and temporal longitudinal oscillations make it unique.

Jupiter’s atmospheric features are generally inferred from still images, but should clearly be time-dependent objects in the planet’s turbulent atmosphere. In fluid dynamics, evolving fea-
tutes in a complex flow are often referred to as transport barriers, which in turn are described as objects that cannot be crossed by other fluid trajectories. However intuitive this characterization of barriers might sound, it actually labels all points in a moving continuum as part of a barrier. This is because any material surface (i.e., a connected, time-evolving set of fluid trajectories) is impenetrable to other trajectories in a flow with unique trajectories. Indeed, a transport barrier is more than just an impenetrable material object: it is a material surface that remains coherent by withstanding stretching and filamentation [99].

A recent approach in dynamical systems theory seeks transport barriers in unsteady flows as key material surfaces with exceptionally coherent features in their deformation. These exceptional material surfaces, Lagrangian coherent structures (or LCSs), were initially defined as most attracting, repelling or shearing material surfaces [104]. Turning this definition into computable mathematical results has proven challenging, prompting instead a widespread use of intuitive diagnostics in LCS detection (see [154] and [169] for reviews). Alternative approaches have also been developed in the meantime to target regions enclosed by LCSs (see [80, 142, 135]).

More recent advances have re-addressed the unsolved mathematical challenges by seeking an LCS as a stationary curve of the Lagrangian strain or shear functional computed along material lines [102, 65]. These variational methods (here collectively referred to as geodesic LCS theory) uncover LCSs as null-geodesics of appropriate strain tensor fields computed from the deformation field. In contrast to the visual assessment of features in intuitive diagnostic fields, geodesic LCS theory renders transport barriers as smooth, parametrized curves that are exact solutions of well-defined stationarity principles. These solutions depend only on frame-invariant tensor fields, and hence remain the same in translating and rotating frames. Given these advantages, we use geodesic LCS detection in the present work to uncover unsteady transport barriers in Jupiter's atmosphere.

Locating geodesic LCSs requires a time-resolved velocity field. For Jupiter, a representative two-dimensional wind-velocity field can be obtained via image-correlation analysis of available cloud videos. In this work, we apply the Advection Corrected Correlation Image Velocimetry (ACCIV) method [12] to obtain a high-density, time-resolved representation of Jupiter's wind field from an enhanced version of a video taken by the Cassini mission of NASA in 2000.

Our main objective in this chapter is twofold. First, we would like to provide a technical review of geodesic LCS theory, summarizing various aspects of the theory from different sources in a unified notation. Second, we wish to show how this theory reveals details of objectively (i.e., frame-invariantly) defined coherent structures in an unsteady flow known only from remote optical sensing. This flow, the wind field of Jupiter reconstructed from the Cassini video, embodies all the major challenges to practical transport barrier detection. First, the data covers a relatively short time period; second, it is temporally aperiodic; and third, it was captured in a
non-inertial frame of reference. These complications necessitate the correct handling of finite-time (as opposed to asymptotic) dynamical systems structures; an abandonment of recurrence and temporal convergence assumptions; and the use of objective (i.e., frame-invariant) methods (cf. [99] for details on these features).

In mathematical terms, our analysis uncovers elliptic and parabolic invariant manifolds in a non-autonomous, temporally aperiodic, finite-time dynamical system. Until recently, a precise definition and extraction of such manifolds has been an unsolved problem even for analytically defined velocity models. As we discuss in detail below, these newly identified dynamical structures support earlier physics-based conclusions obtained by others for Jupiter’s atmosphere. Specifically, we confirm model-based transport predictions by Beron-Vera et al. [18] for zonal jet cores, and find consistency with a geometric circulation model around the GRS proposed by Conrath et al. [42] and de Pater et al. [46]. In addition, we uncover the Lagrangian signature of chevron-type atmospheric features discovered recently in Jupiter’s atmosphere by Simon-Miller et al. [173].

### 3.2 Set-up

Consider a two-dimensional unsteady velocity field

\[ \dot{x} = v(x, t), \quad x \in U \subset \mathbb{R}^2, \quad t \in [t_0, t_1], \quad (3.1) \]

whose trajectories \( x(t; t, x_0) \) define a finite-time flow map \( F^t_{t_0}(x_0) : x_0 \mapsto x(t; t, x_0) \) for times \( t \in [t_0, t_1] \) over the spatial domain \( U \). A material line \( l(t) \) is a smooth curve of initial conditions under the flow, satisfying

\[ l(t) = F^t_{t_0}(l(t_0)). \quad (3.2) \]

Any material line spans a two-dimensional invariant manifold in the three-dimensional extended phase space of the \((x, t)\) variables. Lagrangian Coherent Structures (LCSs) in two-dimensions can loosely be defined as exceptional material curves that end up shaping trajectory patterns. This definition will be made more precise in the next section. Here we only observe that an LCS, just as any material line, is an invariant manifold in the extended phase space \( U \times [t_0, t_1] \), but generally not in the phase space \( U \).

To assess the influence of specific material lines on trajectories, we will need a classic measure of flow deformation, the right Cauchy–Green strain tensor, as defined in (2.3). The tensor \( C^t_{t_0} \) is symmetric and positive definite; it has two positive eigenvalues \( 0 < \lambda_1 \leq \lambda_2 \) and an orthonormal eigenbasis \( \{\xi_1, \xi_2\} \) satisfying (2.4). The Cauchy–Green strain tensor is objective in the sense of continuum mechanics: its invariants remain unchanged in rotating and translating frames [89].
We will also need to use the symmetric part of the tensor $C^t_{t_0}(x_0)\Omega$, defined as
\[
D^t_{t_0}(x_0) = \frac{1}{2} \left[ C^t_{t_0}(x_0)\Omega - \Omega C^t_{t_0}(x_0) \right].
\] (3.3)

### 3.3 Geodesic LCS theory

A general material line of system (3.1) experiences shear and strain in its deformation. Both shear and strain depend continuously on initial conditions owing to the continuity of the map $F^t_{t_0}$. The averaged strain and shear within a strip of $\epsilon$-close material lines, therefore, generically vary by an $O(\epsilon)$ amount within the strip.

The geodesic theory of Lagrangian Coherent Structures (LCSs) seeks exceptionally coherent locations where this general trend breaks down [99]. Specifically, the theory searches for LCSs as special material lines around which $O(\epsilon)$ material belts show no $O(\epsilon)$ variation either in the material shear or in the material strain, both accumulated over $[t_0,t]$ and averaged over material lines.

These variational principles identify the time $t_0$ positions of LCSs as stationary curves of the material-line-averaged Lagrangian shear or Lagrangian strain functionals. Both principles reveal that the initial positions of shearless (hyperbolic and parabolic) and strainless (elliptic) LCSs are null-geodesics of appropriate tensor fields [102, 65]. Later positions of these LCSs can be found by advecting the null-geodesics under the flow map, as described in (3.2). Recent results [115] eliminate numerical instabilities arising in the advection of hyperbolic LCSs. For the elliptic and parabolic LCS considered here, the advection process (3.2) is stable.

We summarize below the main results from [65] for parabolic LCSs (or generalized jet cores) and from [102] for elliptic LCSs (or generalized KAM curves). Parabolic LCSs are expected to identify the unsteady cores of Jupiter’s zonal jets. The largest member of a nested family of elliptic LCSs is expected to mark the Lagrangian boundary of the Great Red Spot. The differential equations rendering these geodesic LCSs only depend on the invariants of the tensor field $C^t_{t_0}(x_0)$, and hence give frame-invariant results. This objectivity of geodesic LCSs is especially important when the underlying velocity field (3.1) is reconstructed in a moving frame, such as the frame of the Cassini spacecraft flying by Jupiter.

#### 3.3.1 Parabolic LCSs

We consider an initial material line $\gamma := l(t_0)$, parametrized as $r(s)$ with $s \in [s_1, s_2]$. The tangent vectors along $\gamma$ are then given by $r'(s)$, and a smoothly varying unit normal along $\gamma$ is given by $n(s) = \Omega r'(s)/|r'(s)|$. The flow map $F^t_{t_0}$ maps $\gamma$ to its time $t$ position, as shown in Fig. 3.1. As
in [65], we define the Lagrangian shear \( p(r(s), n(s)) \) along \( \gamma \) as the tangential projection of the linearly advected normal vector \( \nabla F_{t_0}^t (r(s)) n(s) \).

The averaged Lagrangian shear experienced by \( \gamma \) over the time interval \([t_0, t]\) is given by [65]

\[
P(\gamma) = \int_{\gamma} p(r(s), r'(s))ds = \int_{\gamma} \sqrt{\frac{\langle r'(s), D_{t_0}^t (r(s))r'(s) \rangle}{\langle r'(s), C_{t_0}^t (r(s))r'(s) \rangle}} \langle r'(s), r'(s) \rangle ds.
\]

We seek shearless LCSs as material lines with no leading order variation in their averaged Lagrangian shear. On the time \( t_0 \) position of such LCSs, the first variation of \( P \) must necessarily vanish, i.e.,

\[
\delta P(\gamma) = 0
\]

The most readily observed solutions of (3.4) are those obtained under the largest possible set of admissible variations, including changes to the endpoints of \( \gamma \). As shown in [65], the variational problem (3.4) posed with free-endpoint boundary conditions is equivalent to finding null-geodesics connecting singularities of the Lorentzian metric

\[
g(u, v) = \langle u, D_{t_0}^t (x_0) v \rangle,
\]

where \( \langle ., . \rangle \) denotes the Euclidean inner product. The singularities of \( g(u, v) \) are points where \( \det D_{t_0}^t (x_0) = 0 \). These are precisely the points where the Cauchy–Green tensor field \( C_{t_0}^t (x_0) \) has repeated eigenvalues. Following the convention in the tensor-line literature [51], we refer to such points as the singularities of the Cauchy–Green strain tensor (Fig. 3.2).

All null-geodesics of the metric \( g \) are found to be solutions of one of the two ODEs

\[
r'(s) = \xi_j(r(s)), \quad j = 1, 2.
\]

We refer to trajectories of (3.5) with \( j = 1 \) as shrink lines, as they are compressed by the flow.
Similarly, we call trajectories of (3.5) with $j = 2$ stretch lines, because they are stretched by the flow. Shrink and stretch lines are special cases of tensor lines used in the scientific visualization literature to illustrate features of two-dimensional tensor fields [51]. Smooth null-geodesics connecting metric singularities of $g(u, v)$ are, therefore, smooth heteroclinic chains formed by shrink lines and stretch lines among the singularities. For observable impact on mixing, we focus on such null-geodesic chains that are structurally stable and locally unique. This requirement restricts the shrink-stretch chains of interest to those along which wedge- and trisector-type singularities of $C^t_{t_0}(x_0)$ alternate (cf. Fig. 3.2 and [65] for details).

The time $t_0$ positions of parabolic LCSs (Lagrangian jet cores) are defined as tensor-line chains of the type in Fig. 3.2 that are also weak minimizers of a quantity that measures the closeness of the chain to being neutrally stable under advection by $F_{t_0}$ [65]. This quantity, the neutrality of a tensorline at a point $r$, is defined as

$$
\mathcal{N}_{\epsilon_j}(r) = \left( \sqrt{\lambda_k(r)} - 1 \right)^2, \quad j \neq k.
$$

A weak minimizer of $\mathcal{N}_{\epsilon_j}$ is then defined as a trajectory of (3.5) that lies, together with the nearest trench of $\mathcal{N}_{\epsilon_j}(r)$, in the same connected component of the set of points defined by the

Figure 3.2: (a) A parabolic LCS is a structurally stable, alternating chain of shrink-stretch curve segments that connect Cauchy–Green singularities. A single shrink line (red) in such a chain is superimposed on the local stretch-line geometry (blue) near the LCS. (b) Topology of stretch lines around wedge-type and trisector-type Cauchy–Green singularities.
relation \( \langle \nabla^2 N, \xi_j(r) \xi_k(r) \rangle > 0 \), with \( j \neq k \). (cf. [65] for more detail).

Figure 3.3 illustrates the construction of the time \( t_0 \) position of a parabolic LCS. The position of such an LCS at an arbitrary time \( t \) can be found by advecting its \( t_0 \) position under the flow map \( F_{t_0}^t \). Details on the numerical procedures involved can be found in [65].

The resilience of parabolic LCSs (serving as solutions to (3.4) even under variations to their endpoints) is in agreement with several numerical studies pointing out the robustness and easy observability of Lagrangian jet cores in unsteady zonal flows [160, 162, 18]. For a general discussion of importance of jet cores and their impact on global weather patterns, see [72].

Finally, time \( t_0 \) positions of hyperbolic LCSs are defined as Cauchy-Green tensorlines starting from local extrema of the strain-eigenvalue fields \( \lambda_i(x_0) \) [65, 99]. Hyperbolic LCSs, therefore, are also solutions of the stationarity principle (3.4) for shearless LCSs, but only with respect to variations leaving their endpoints fixed. This constraint on boundary conditions implies a lower degree of observability for hyperbolic LCSs relative to parabolic LCSs. Furthermore, in the short-time, shear-dominated context of the Jupiter video studied here, we have only found weak normal repulsion and attraction along material lines. We therefore omit a discussion of hyperbolic LCSs in this study.
3.3.2 Elliptic LCSs

By Fig. 3.1, the averaged Lagrangian repulsion experienced by a closed material curve $\gamma$ over the time interval $[t_0, t]$ is given by

$$Q(\gamma) = \oint_{\gamma} q(r(s), r'(s)) ds = \oint_{\gamma} \frac{1}{\sqrt{\langle r'(s), [C_{t_0}^t(r(s))]^{-1} r'(s) \rangle \langle r'(s), r'(s) \rangle}} ds. \quad (3.7)$$

To generalize the concept of a Kolmogorov–Arnold–Moser-type (KAM-type) transport barrier from time-periodic to finite-time aperiodic flows, we seek closed material curves $\gamma$ along which the averaged repulsion $Q$ shows no leading-order variation. Satisfying

$$\delta Q(\gamma) = 0, \quad (3.8)$$

such a closed curve $\gamma$ has a thin annular neighborhood in which no material filamentation occurs over the time interval $[t_0, t]$, just as in neighborhoods of KAM curves in the time-periodic case. As a result, the interior of $\gamma$ exhibits no advective mixing with its exterior. The observability of solutions of (3.8) is on par with those of (3.4). Indeed, they prevail under the largest possibly set of admissible variations, as long as those variations are also closed curves.

For incompressible flows, stationarity of the averaged normal repulsion is equivalent to the stationarity of the averaged tangential stretching defined along the material line $\gamma$. As shown in [102], the latter problem is solved by closed null-geodesics of the Lorentzian metric family

$$g_{\lambda}(u, v) = \langle u, E_{\lambda} v \rangle, \quad \lambda > 0,$$

with the generalized Green–Lagrange strain tensor $E_{\lambda}(x_0)$ defined as

$$E_{\lambda}(x_0) = \frac{1}{2} \left[ C_{t_0}^t(x_0) - \lambda^2 I \right].$$

The metric $g_{\lambda}$ is Lorentzian (i.e., indefinite) on the set $U_{\lambda} = \{ x_0 \in U : \lambda_1(x_0) < \lambda^2 < \lambda_2(x_0) \}$. In this set, all closed null-geodesics of $g_{\lambda}$ are trajectories of one of the two families of ODEs

$$r'(s) = \eta_{\lambda}^\pm(r(s)), \quad \lambda \in \mathbb{R}^+, \quad (3.9)$$
Chapter 3. Geodesic Transport Barriers in Jupiter’s Atmosphere

Figure 3.4: The construction of a Lagrangian vortex boundary for the periodically forced Duffing oscillator \( \eta \). Elliptic LCSs are identified as limit cycles of the \( \eta \pm \) direction fields via a Poincare section. The outermost elliptic LCS is highlighted with green as the Lagrangian vortex boundary.

where

\[ \eta_{\pm}(r) = \sqrt{\frac{\lambda_2(r) - \lambda^2}{\lambda_2(r) - \lambda_1(r)}} \xi_1(r) \pm \sqrt{\frac{\lambda^2 - \lambda_1(r)}{\lambda_2(r) - \lambda_1(r)}} \xi_2(r). \]

Again, for reasons of observability, we focus on structurally stable closed trajectories (i.e., limit cycles) of (3.9).

Time \( t_0 \) positions of elliptic LCSs are, therefore, limit cycles of the line fields (3.9). Such limit cycles turn out to encircle at least two wedge-type singularities of the Cauchy–Green strain tensor field (Fig. 3.4). This fact enables the automated numerical detection of elliptic LCS even in complex flow fields [117].

The position of an elliptic LCS at an arbitrary time \( t \) can be found by advecting its \( t_0 \) position under the flow map \( F^t_{t_0} \). Any limit cycle \( \gamma \) of (3.9) turns out to be uniformly stretching under such advection [102]. This means that the arclength of any subset of \( \gamma \) increases exactly by the factor \( \lambda \) under the flow map \( F^t_{t_0} \). Limit cycles of (3.9) only tend to exist for \( \lambda \approx 1 \), guaranteeing a high degree of material coherence for a coherent Lagrangian vortex boundary, defined in [102] as the outermost member of a nested family of limit cycles of (3.9).

Figure 3.4 illustrates the construction of the time \( t_0 \) slice of an elliptic LCSs and coherent Lagrangian vortex boundaries. Details on the numerical procedures involved can be found in [102].
3.4 Unsteady transport barriers in the atmosphere of Jupiter

3.4.1 Prior work and present objectives

Notable earlier attempts to identify transport barriers in Jupiter’s atmosphere from images started in [60], with velocities inferred from a manually assisted image-correlation analysis of 10 pairs of photos. The resulting velocities were then all viewed as part of an underlying steady wind field. This steady vector field produced spiraling streamlines near the GRS without a clear indication of material barriers.

Later work [40] used three high-resolution snapshots from the 2000 Galileo Mission to construct a steady velocity field from automated cloud-tracking via image correlation velocimetry. Again, trajectories spiraling into the GRS emerged from this time-averaged analysis, indicating no particular closed material barrier around the GRS. More recently, a potential-vorticity-conserving flow (steady in the frame co-moving with the GRS) was constructed in [170] as a best fit to cloud motion inferred from different space missions. By construction, this averaged approach renders all streamlines closed near a vortical feature (such as the GRS). The approach, however, does not address the question of actual material transport via the unsteady winds of Jupiter.

Extracting an unsteady velocity field and analyzing its finite-time transport barriers has not been attempted in prior publications. One reason for this is a clear focus of the planetary science community on long-term evolution in Jupiter’s climate. Comparing velocity snapshots and averages taken from different missions and different years, rather than studying a video footage from a single mission, is clearly more appropriate for a study of climate evolution. The unavoidable time-dependence of velocities extracted from temporally close video frames has, in fact, been viewed as undesirable uncertainty to an envisioned steady mean velocity field (see, e.g., Asay-Davis et al. [12]). Another reason for the lack of unsteady transport barrier studies for Jupiter has been the unavailability of precise mathematical tools (such as those surveyed here in Section 3.3) for LCS extraction from finite-time, aperiodic velocity data.

3.4.2 Video footage

The raw footage acquired by the Cassini Orbiter comprises 14 cylindrical maps of Jupiter, covering the 10 days ranging from October 31 to November 9 in the year 2000. We use an enhanced version of this video, which NASA created by interpolation and by addition of information from previous Jupiter missions [203, 204]. The pixels of the enhanced footage extend to 360 degrees of longitude and 180 degrees of latitude, with a resolution of $360^\circ \times 180^\circ$. The time step between the frames is 1.1 hours.
To verify the feasibility of the enhanced cloud movie for Lagrangian advection studies, we used the extracted velocity field (to be described below) to advect the first image of the raw footage up to the final time of the same footage. We show representative portions of these two images in Fig. 3.5 for comparison. We computed an offset error as the $l^2$-distance between the pixels of the advected initial raw image and the raw image at the final time, normalized by the $l^2$-norm of the final raw image. In this fashion, we obtained an offset error of 5.6%, which is in the order otherwise expected from numerical noise, processing errors, and diffusive cloud mixing.

3.4.3 Optical velocity field reconstruction

Image-correlation analysis applied to available cloud videos provides a two-dimensional representation of Jupiter’s winds. Early wind-field reconstruction studies required a human operator to identify the same cloud feature in subsequent frames [143]. The seminal paper by Limaye [126] introduced the first automated image-correlation method using one-dimensional correlations along latitudinal circles. Later improvements involved two-dimensional extensions and coarse-to-fine iteration schemes [30]. Recent approaches add an advection equation to the procedure [12], or apply the idea of an optical flow [130]. Similar methods have also been applied to image sequences of other planets such as Saturn [164], Uranus and Neptune [118], and the Earth [123].

Here we use the Advection Corrected Correlation Image Velocimetry (ACCIV) algorithm developed in [12] to extract a time-resolved atmospheric velocity field from the enhanced Cassini footage described in section 3.4.2. ACCIV uses the idea of the two-pass Image Correlation Velocimetry (CIV) developed by [181, 70, 69] for experimental fluid velocity measurement. In
ACCIV is an iterative technique for reconstructing a velocity field that is assumed to advect an observed scalar field passively. As a first step, ACCIV recursively splits two successive images $I$ and $J$ into sub-images or correlation boxes. Then, for a correlation box $C_i \subset I$, ACCIV finds a matching box $C_j \subset J$ of the same size. The process of matching correlation boxes is performed by maximizing the cross-correlation between intensity patterns. The velocity at the center of $C_i$ is then the distance between the centers of $C_i$ and $C_j$ divided by the time elapsed. The algorithm repeats this process for all the correlation boxes in the image $I$, yielding a crude velocity field approximation under the assumption that a correlation box moves from image $I$ to the subsequent image $J$ without any distortion. The crude initial velocity approximation is then used to advect the images to some intermediate time when no real data are available. The difference
between the synthetic images at the intermediate time is iteratively used to generate correction vector fields, producing increasingly accurate velocity vectors. In a second step, ACCIV makes use of the first-step results and looks for a correlation between a box of pixels in the image $I$ and a box of pixels transformed in the subsequent image $J$. The possible transformations are a combination of translation, rotation, shear and distortion. Figure 3.6 illustrates how considering a deformed correlation box can lead to a better approximation of the displacement vector at the center of the correlation box $C_i$. Similar to the first step, ACCIV iteratively improves the accuracy of the velocity vectors by building synthetic images and producing correction vectors. The next steps consist of further refinement of the velocity field using smaller correlation boxes.

ACCIV repeats each of these steps iteratively until the velocity correlation uncertainty shows
Chapter 3. Geodesic Transport Barriers in Jupiter’s Atmosphere

Table 3.1: ACCIV parameters used to produce the time-resolved velocity field. The box size, search range, and stride are in units of pixels. The box size is the size of the correlation box for the relevant CIV pass. The search range is the range of correlation box displacements used in each dimension. The stride is the number of pixels by which the correlation box is shifted between each measurement. It controls the output resolution of the velocity vector field. The number of image pairs is the total number of pairing of the set of images. The Number of Smooth Fit Control Point Neighbors controls the smoothness of the velocity field on the grid.

<table>
<thead>
<tr>
<th>First pass</th>
<th>Second pass</th>
<th>Number of image pairs</th>
<th>Number of indep. vectors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation box size</td>
<td>Search range</td>
<td>Stride</td>
<td># Smooth Fit Control Point Neighbors</td>
</tr>
<tr>
<td>64 × 64</td>
<td>(−6,6)</td>
<td>14</td>
<td>110</td>
</tr>
</tbody>
</table>

no further decrease. First, ACCIV defines the offset between advected and raw images at time \( t_2 \) as local correlation location uncertainty. The correlation velocity uncertainty is then the correlation location uncertainty divided by the time elapsed between the images. The velocity fields we have extracted with ACCIV have tens to hundreds of thousands of independent velocity vectors with uncertainties on the order of 2-5 m/s.

Figure 3.7 shows the measure of correlation velocity uncertainty as a percentage of the spatial mean velocity for each video frame. The near-periodicity of the uncertainty history arises from the mostly even sampling frequency (roughly two jovian days) of the original raw footage from Cassini. Two time intervals (around days 5-7 and 16-18) with shorter sampling times create an impression of approximate symmetry in the uncertainty distribution with respect to day 12, but this is accidental.

On average, the extracted velocity vectors have about 6.4% uncertainty with respect to the mean of the reconstructed velocities. Figure 3.8 shows the spatial distribution of velocity uncertainty as a percentage of the velocity norm at each grid point. The velocity uncertainty is higher in regions such as the GRS where we have complicated dynamics and presumably cloud mixing.

We recall that we integrate the reconstructed velocity field to find structurally stable structures: limit cycles for elliptic LCSs and robust heteroclinic cycles for parabolic LCSs (cf. Section 3). These structures persist in the (unknown) true velocity field, as long as the imaging and reconstruction errors represent an overall moderate perturbation to the flow map, such as the perturbation we infer from Fig. 3.5. This persistence result holds even if the velocity errors are pointwise large at times (cf. [96]).

Some input parameters for ACCIV, such as the times between frames and the threshold for removing outliers, are straightforward to specify. Other parameters, such as correlation box size, search range, and stride, must be optimized iteratively. Table 3.1 summarizes the input parameters and results for each pass. For more information on setting the input parameters of ACCIV, we refer the reader to the project webpage [206].

Figure 3.9 shows a representative snapshot of the reconstructed unsteady velocity field,
which is available over the domain ranging from $-180^\circ$W to $180^\circ$E in longitudes and from $-35^\circ$S to $20^\circ$N in latitudes, with a grid resolution of $7202 \times 1102$. This supersedes the resolution of earlier velocity fields reconstructed for Jupiter from manual cloud-tracking approaches [143, 60, 173, 124, 184].

In principle, an alternative to the ACCIV method used here would be Digital Particle Image Velocimetry (DPIV), which reproduces fluid velocities from highly resolved tracks of luminescent particles in well-illuminated laboratory flows [106]. Under such conditions, high-speed imaging can reliably detect small particle displacements amidst minimal illumination changes, a basic requirement for the optical flow methods underlying DPIV. In observations of planetary atmospheres, however, these ideal imaging conditions are generally not met. For instance, the sunlight scattered from Jupiter’s cloud cover into the camera of a spacecraft tends to vary considerably between two subsequent images.

As a consequence, optical flow methods have not gained popularity in velocity field reconstruction from planetary observations. A rare exception is Ref. [130], which works with a single pair of well-lit, high-resolution images from NASA’s Galileo mission, separated by one hour. From this data, [130] extracts a single high-quality, steady velocity snapshot. Unfortunately, the data available from the Galileo mission is insufficient for the extraction of a reasonably long unsteady velocity field via the approach of [130].

In comparison to the two high-resolution images used in [130], the more modestly resolved but temporally extended Cassini data set used here offers a clear advantage for unsteady LCS detection. As is now well-established for satellite altimetry maps of the ocean, even without capturing smaller (sub-mesoscale and lower) features of a velocity field, one can accurately capture
its mesoscale LCSs, which in turn agree with in situ float observations [148].

3.4.4 Validation of the reconstructed velocity field

Available observational records of Jupiter go back to the late 19th century, indicating that Jupiter’s atmosphere is highly stable in the latitudinal direction. The average zonal velocity profile as a function of latitudinal degree is, therefore, an important benchmark in assessing the quality of the reconstructed velocity field.

In Fig. 3.10 we compare the temporally averaged zonal velocity profile obtained from ACCIV with the classic profile reported by Limaye [126]. The Limaye profile has been used and confirmed by several other studies on different data sets from different missions (see, e.g., [108, 184, 83]). These studies all support the conclusion that the averaged zonal wind field constructed by Limaye [126] has remained fundamentally unaltered. Limaye’s velocity profile is based on Voyager I and Voyager II images, covering a total of 142 Jovian days in 1979. In contrast, our ACCIV-based velocity profile is based on the Cassini mission, and covers a total of 24 Jovian days in 2000-2001 [203]. Overall, the two profiles shown in Fig. 3.10 match fairly closely, showing only appreciable discrepancies near velocity extrema. These discrepancies arise because ACCIV, as other image-correlation methods, systematically underestimates the
magnitude of the velocity vectors near peaks of the velocity field [12].

We finally note that we have used images taken at the visible wavelength to extract the velocities of Jupiter’s clouds. Asay-Davis et al., [13] show that velocities extracted from images taken with different visible wavelengths (at different times) produce similar zonal velocities. This observation and other empirical studies support the expectation that Jupiter’s cloud velocities can be correctly inferred from images taken at visible wavelengths.

### 3.4.5 Lagrangian advection

For the parabolic LCS computation described in section 3.3.1, we calculate the Cauchy–Green strain tensor field $C_{t_0}$ defined in (2.3) with $t_0 = 0$ and $t = 24$ days, over a uniform grid $G_1$ of $5600 \times 2200$ points. The spatial domain $U$ ranges from $-145^\circ$W to $95^\circ$E in longitudes and from $-35^\circ$S to $20^\circ$N in latitudes. For the elliptic LCS computations described in section 3.3.2, a smaller grid $G_2$ of $900 \times 600$ points suffices, because the accurate identification of Cauchy–Green singularities is not crucial. In all computations, we use a variable-order Adams–Bashforth–Moulton solver (ODE113 in MATLAB) to solve the differential equations (3.1) and (3.9). The absolute and relative tolerances of the ODE solver are chosen as $10^{-6}$. Off the grid points, we obtain the $\xi_j(x_0)$ and $\eta_\pm(x_0)$ line fields from bilinear interpolation.

### 3.4.6 Unsteady zonal jet cores as parabolic LCSs

Influential work on model flows has indicated that high potential-vorticity (PV) gradients occurring along the cores of the eastward jets of Jupiter create effective barriers to material transport [113]. More recent work on perturbed PV-staircase flow models revealed that westward jet cores also act as material barrier cores, even when the PV has vanishing gradients along these lines [18]. Averaged meridional velocities support this conclusion [18], but time-resolved studies using observed winds have not been carried out to ascertain the existence of actual material transport barriers along eastward and westward jet cores. Here we examine the validity of the above model-based conclusions on the unsteady wind field inferred from the time-resolved Cassini footage. Applying the theory surveyed in section 3.3.1, we compute the Lagrangian cores of jet streams located between latitudes $-35^\circ$S and $20^\circ$N (see Fig. 3.11a). In line with the model-based conclusions of [18], we find coherent Lagrangian jet cores both for eastward and westward jets. Unlike the straight lines suggested by individual snapshots, however, the actual unsteady jet cores exhibit small-amplitude north-south oscillations, as shown in Fig. 3.11b.
Figure 3.11: (a) Instantaneous positions of parabolic LCSs at the initial time as shearless transport barriers that form the core of jet streams in the atmosphere of Jupiter. (b) The spatial profile of Jupiter’s southern equatorial jet. The average meandering width is about $0.33^\circ$ latitudinal degree.

Figure 3.12: Impact of Jupiter’s southern equatorial shearless transport barrier on tracer disks over 11 jovian days. The deformed tracer disks resemble the shape of the recently discovered chevrons [173], i.e., dark v-shaped clouds in the background. The contrast of images is improved for better visualization. The complete advection sequence over 24 Jovian days is illustrated in the on-line supplemental movie M1.
Figure 3.12 shows the evolution of initially circular blobs of tracers, centered on the shearless core of the southern equatorial jet, after 11 Jovian days. The shape of tracer blobs resembles the shape of chevrons, serving as Lagrangian footprints of Rossby waves identified recently from an Eulerian perspective on the southern hemisphere of Jupiter [173].

The advected jet core, as a material line, does not allow mixing between the two wings of any chevron. Specifically, the evolving parabolic LCS does act as a transport barrier, keeping its coherence, and showing no wave-breaking or fingering-type deformation. This coherence effectively blocks the advective excursion of material between the upper and lower half of the material jet.

### 3.4.7 The Great Red Spot as a generalized KAM region

Observational evidence suggests the existence of coherent rings around all jovian vortices, including the GRS. The accepted explanation is that these rings are signs of vertically moving air parcels in the three-dimensional atmosphere of the planet [42, 46]. A ring of air parcels is a material transport barrier that is expected to have a two-dimensional footprint in the horizontal wind field around the GRS. Such a signature, however, has not been identified in available advection studies (cf. section 3.4.1).

Here we seek an annular material transport barrier region around the GRS by applying the geodesic LCS theory described in section 3.3.2. This necessitates the computation of limit cycles for the family of autonomous dynamical systems defined in eq. (3.9). We discard limit cycles obtained for the same \( \lambda \) value, if they are within two velocity grid steps from each other. This is to focus on robust enough limit cycles that are far enough from undergoing a saddle-node bifurcation.

This computation yields 73 elliptic LCSs, computed as robust limit cycles of the differential equation family (3.9) (Fig. 3.13a). This set of closed curves forms a generalized KAM region, filled with material loops that resist filamentation and act as coherent transport barriers through the entire duration of the underlying video footage.

Figure 3.13b shows separately the elliptic LCS with perfect coherence (\( \lambda = 1 \)) in black, as well as the outermost elliptic LCS in blue that forms the outer boundary of the coherent Lagrangian vortex associated with the GRS. This vortex boundary is marked by the parameter value \( \lambda = 1.0063 \), which forecasts a roughly 0.6% increase in arclength. The two closed curves enclose a highly coherent annular barrier, the Lagrangian counterpart of the outer ring identified within a collar of the GRS described in [130]. This outer ring was constructed as the annulus outside the perceived core of the collar, a closed curve of velocity maxima.

Our geometric construction of an annular Lagrangian transport barrier is also motivated by the visual observations in [46]. These often indicate a sharper inner boundary and a more dif-
Figure 3.13: (a) Elliptic LCSs defining the Lagrangian footprint of the Great Red Spot at time $t = 0$. The color-bar refers to values of the stretching parameter $\lambda$ arising in the construction of the elliptic LCS family. (b) Elliptic LCS with perfect coherence (black), as well as the Lagrangian vortex boundary (blue) of the Great Red Spot at time $t = 0$. The boundary is extracted from velocity data covering 24 Jovian days. The advection sequence is illustrated in the online supplemental movie M2. (c) Elliptic LCSs defining the Lagrangian footprint of the Great Red Spot at time $t = 24$. (d) Elliptic LCS with perfect coherence (black), as well as the Lagrangian vortex boundary (blue) of the Great Red Spot at time $t = 24$. (e) Relative stretching of the perfectly coherent ($\lambda = 1$) inner-core boundary and the slightly expanding outer boundary of the GRS over 24 Jovian days. As predicted by geodesic theory, the arc-length of the outer boundary changes about 0.6% in agreement with the theoretical stretching value ($\lambda = 1.0063$) of the extracted outer boundary. (f) Plot of aspect ratio ($length^2/area$) as a function of time.
fusive outer boundary for coherent jovian rings. The sharp observational boundary suggests an elliptic LCS of the highest possible coherence (zero stretching), while a diffusive outer boundary is expected near an elliptic LCS of the lowest possible coherence (highest stretching in a family of elliptic LCSs).

The exact location of these oval barriers is expected to change under varying data resolution. The structural stability in the construction of elliptic LCSs, however, guarantees that the barriers move only by a small amount under small enough variations in resolution.

Advected images of the extracted GRS boundary confirm its sustained coherence over the finite time of extraction (see Figures 3.13c to 3.13e). Our computation shows that the predicted coherent core of the GRS indeed regains its arclength after 24 Jovian days. At the same time, the coherent outer boundary of the GRS indeed grows in arclength by about 0.6%, while its longitudinal extent decreases by about 5%. This suggests that the coherent boundary of the GRS is becoming rounder, which is generally consistent with the available observational records taken over much longer periods [172] (see Fig. 3.13f). Clearly, any firm conclusion or prediction about the long-term behavior of the arclength of material GRS boundary would require the analysis of a substantially longer data set.

3.5 Summary and conclusions

We have applied the recently developed geodesic theory of transport barriers [102, 65] to an enhanced video from NASA’s Cassini mission to Jupiter. First, we obtained a representative two-dimensional wind field from this video via the Advection Corrected Correlation Image Velocimetry (ACCIV) algorithm of Asay–Davis et al. [12]. Next, we identified, for the first time, unsteady material transport barriers in the wind field that form the cores of zonal jets and the boundary of the Great Red Spot (GRS) in Jupiter’s atmosphere.

The parabolic LCSs (Lagrangian jet cores) we have found show that both easterly and westerly jet cores provide strong material transport barriers. This latter finding confirms the conclusion of Beron-Vera et al. [18] based on a numerical study of a perturbed potential-vorticity-staircase model relevant for Jupiter. Deforming material blobs placed near the parabolic LCS also reveal the Lagrangian footprint of the recently discovered chevron-type atmospheric features [173].

The elliptic LCSs we identify provide a foliation of the GRS into highly coherent, uniformly stretching layers. This supports the existence of a proposed two-dimensional, cylindrical material transport barrier around the GRS [42, 46]. According to our results, this cylindrical region has finite width, represented by an annulus in our two dimensional analysis. The annulus has a perfectly coherent (λ = 1) inner boundary and a nearly perfectly coherent outer boundary, as
shown in Fig. 3.13b. While the outer boundary shrinks in longitudinal extent over the observed 24 Jovian days, its total arc-length shows a slight increase of about 0.6%. This suggests a modest evolution of this Lagrangian boundary towards more perfect circularity, which is in line with longer observational records [172].

The time-resolved image reconstruction technique employed here is purely kinematic, and does not incorporate a fit to dynamic equations believed to govern Jupiter’s wind fields. Considerable effort has been devoted to fitting dynamically consistent reduced models to optically reconstructed velocities (see, e.g., [170]). These models, however, are steady in a moving frame, and incorporate velocity measurements from different sources and times. Here, instead of securing dynamical consistency for an averaged, steady velocity field, we have constructed an unsteady velocity field that is kinematically consistent with a specific observational period. Imposing some degree of dynamical consistency on the optically reconstructed velocity field and comparing it with steady models in a moving frame remains a viable future research direction. A clear challenge is that the cloud distribution over the GRS does not align with the location of its associated potential vorticity anomaly or with any other of the GRS’s known dynamical features [12].

Arriving at Jupiter in 2016, the Juno mission of NASA will explore some of the material movement deep beneath the planet’s clouds for the first time [205]. Using this future information, we expect to be able to extend some aspects of our present analysis to three-dimensions using recently developed 3D variational LCS methods [27].

We envision further applications of the methodology developed here to remotely observed patterns in meteorology [156], oceanography [20], environmental monitoring [189, 137] and crowd surveillance [3, 178].
Chapter 4

Spectral-Clustering Approach to Lagrangian Vortex Detection

Chapter Abstract

In this chapter, we show how Lagrangian coherent vortices can be extracted as clusters of Lagrangian trajectories. We carry out the clustering on a weighted graph, with the weights measuring pairwise distances of fluid trajectories in the extended phase space of positions and time. We then extract coherent vortices from the graph using tools from spectral graph theory. Our method locates all coherent vortices in the flow simultaneously, thereby showing high potential for automated vortex tracking. We illustrate the performance of this technique by identifying coherent Lagrangian vortices in several two- and three-dimensional flows.

4.1 Introduction

It has long been recognized that even unsteady flows with aperiodic time dependence admit persistent patterns that govern the transport of passive tracers [140, 158, 99]. Generally referred to as coherent structures, these patterns are often vortex-type spatial features that remain recognizable over times exceeding typical time scales in the flow. Our goal here is to systematically decompose trajectories in such a general flow into coherent and incoherent families, providing a conceptual simplification of the underlying dynamical system.

The majority of coherent structure identification methods used in fluid dynamics continues to be Eulerian (see, e.g., [200, 58, 193, 150] for recent examples), concerned with features of the instantaneous velocity field driving the flow [107, 201]. The resulting Eulerian coherent structure criteria have been broadly used in flow structure identification, although none has emerged as a definitive tool of choice. By their focus on the velocity field, these Eulerian criteria
inherently depend on the reference frame in which they are applied [97].

By contrast, Lagrangian methods identify vortical flow structures based on the properties of fluid particle trajectories [158, 28, 190, 154, 99]. Several of these methods are frame-invariant and hence the structures they locate (or miss) are the same in all frames that translate and rotate relative to each other. This invariance is especially important for geophysical flows which are invariably defined in the rotating frame of the earth. In such flows, long lived coherent vortices may transport fluid over great distances, surrounded by strongly mixing background turbulence [158, 102].

Lagrangian vortex detection approaches either seek a coherent material boundary to the vortex, or aim to identify a coherent interior of a vortex. Coherent material vortex boundaries are special cases of Lagrangian coherent structures (LCSs), the most influential material surfaces in the flow [99]. Within this class, Lagrangian vortex boundaries can either be defined as outermost non-filamenting, closed material surfaces (elliptic LCSs [102, 27]), or as outermost, closed material surfaces of equal material rotation [68, 103]. Other approaches target Lagrangian vortex boundaries as locations of minimal curvature change [135] or as curves that maximize the volume to boundary size ratio throughout advection [76].

Approaches seeking the interior of Lagrangian vortices have mostly been probabilistic in nature. Early techniques relied on the diagnostic use of relative and absolute dispersion [158]. Later mathematical approaches offer a bipartition of phase space into minimally diffusive regions by delineating the density evolution that can be characterized by the Perron-Frobenius or transfer operator [80, 74, 78]. Further diagnostic approaches have also been influenced by techniques for ergodic dynamical systems, such as trajectory complexity and long-term averages along trajectories [142, 163, 32].

The clustering approach developed here falls in the second category, focusing on the identification of the interiors of coherent Lagrangian vortices. Our method is unconcerned with the deformation of the boundary, requiring only a bulk coherence for the interior of the material vortex instead. We build on techniques developed over the past few decades in computer science for data clustering [64]. While clustering methods have already been used in coherent structure detection in fluid flows [166, 79], here we apply spectral clustering to a graph describing the spatio-temporal evolution of a fluid. This approach identifies coherent vortices as clusters of Lagrangian trajectories remaining close over a finite-time interval. As we show, our proposed method detects coherent vortices in two- and three-dimensional flows, and can be extended to higher dimensional problems as well. Its main advantage is that it requires a relatively low number of Lagrangian trajectories as an input, making it suitable for the analysis of low-resolution trajectory data sets (see also, [79, 194, 76] for methods with a similar capability).

Prior definitions of coherence are tied to specific geometrical requirements such as convexity
lack of filamentation [102], or shape coherence [135] of the vortex boundary. In contrast, our approach does not pose any geometrical constraint on the vortex boundary, which helps us to identify coherent vortices that may have non-convex or deformable boundaries. Unlike most other Lagrangian methods [102, 135, 142, 80], which rely only on initial and final positions of particles, our method makes use of intermediate particle location information (as does [79]). Another important feature is the ability to extract the a priori unknown number of coherent structures from the trajectory data set together with their simultaneous detection. This is an important prerequisite for automatic vortex tracking in large-scale data sets (see also [117]).

Our approach is based on three basic principles:

**Principle 1.** [Coherence indicator] The dynamical distance between two Lagrangian particles is the distance between their corresponding trajectories in space-time over a finite time interval \([t_0, T]\) of interest.

**Principle 2.** [Coherent structure] A coherent structure is a distinguished set of Lagrangian particles which have mutually short dynamical distances relative to the distances to particles from its complement.

This definition adopts the notion of coherence from spatio-temporal clustering algorithms [119] to coherence in fluid flows, in a fashion similar to [79]. A typical unsteady fluid, however, is not a union of coherent structures. Rather, it is composed of coherent sets and their surrounding incoherent background turbulence [140, 158]. Our third principle makes this explicit as follows.

**Principle 3.** [Coherence vs. incoherence] Coherent structures are surrounded by an incoherent background of particles.

Our Principle 3 underlines the impossibility of a simple clustering of a general fluid flow into coherent structures. Instead, we formulate the following main objective.

**Problem 1.** Given a fluid domain, possibly sampled discretely, and a finite time interval \([t_0, T]\) of interest, find a partition of the fluid domain into coherent structures surrounded by an incoherent background.

The rest of the chapter is organized as follows. Section 4.2 presents our method for identifying coherent vortices. Section 4.3 describes the relationship of our method with previous methods, namely the transfer operator approach [80, 74], its hierarchical application [134], the application of the community detection method Infomap to the transfer operator [166], and the direct application of the fuzzy C-means algorithm to trajectory data sets [79]. We demonstrate the applicability and effectiveness of our method through four examples in Section 4.4.
4.2 Method

The general outline of our method is as follows. To solve the physical Problem 1, we start with a discrete sample of the fluid flow and generate an abstract weighted graph, whose nodes correspond to Lagrangian particles and whose edge weights are determined according to Principle 1. Next, we apply spectral clustering to this graph, which is particularly suited to detect clusters in the graph according to Principle 2 together with the incoherent background, consistently with Principle 3.

4.2.1 Input: A trajectory data set

The essential input for our algorithm is a spatio-temporal trajectory data set, such as particle tracks from a flow experiment, drifter data from the ocean, or from numerical integration of a differential equation. The trajectory data set may be sparse or spatially non-uniform at the initial time. Specifically, we only assume that in a \( d \)-dimensional configuration space, \( n \) trajectory positions \( \{ \mathbf{x}^i(t) \}_{i=1}^n \in \mathbb{R}^d \) are available at \( m \) discrete times \( t_0 < t_1 < \ldots < t_k < \ldots < t_{m-1} = T \). This information can be stored in an \( n \times m \times d \)-dimensional numerical array, with elements \( \mathbf{x}^i_k := \mathbf{x}^i(t_k) \in \mathbb{R}^d \).

From this trajectory data, we define the \textit{dynamical distance} \( r_{ij} \) between Lagrangian particles \( \mathbf{x}^i \) and \( \mathbf{x}^j \) as

\[
 \begin{align*}
 r_{ij} &:= \frac{1}{t_{m-1} - t_0} \sum_{k=0}^{m-2} \frac{t_{k+1} - t_k}{2} \left( | \mathbf{x}^i_{k+1} - \mathbf{x}^j_{k+1} | + | \mathbf{x}^i_k - \mathbf{x}^j_k | \right) \\
 &\approx \frac{1}{t_{m-1} - t_0} \int_{t_0}^{t_{m-1}} \left| \mathbf{x}^i(t) - \mathbf{x}^j(t) \right| \, dt.
\end{align*}
\]

Here \( | \cdot | \) denotes the spatial Euclidean norm, and hence \( r_{ij} \) approximates the \( L^1 \)-norm of pairwise trajectory distances. Since Euclidean coordinate transformations leave Euclidean distances unchanged, one readily sees that the pairwise distances are \textit{objective}, i.e., they remain unchanged in coordinate systems rotating and translating relative to each other [182]. Moreover, it is noteworthy that the pairwise distances remain unchanged under refinements of the spatial resolution.

4.2.2 Similarity graph construction

Next, we convert the spatio-temporal data set with the pairwise distances \( r_{ij} \) into a similarity graph \( G = (V, E, W) \), which is specified by the set of its nodes \( V = \{ v_1, \ldots, v_n \} \), the set of edges \( E \subseteq V \times V \) between nodes, and a similarity matrix \( W \in \mathbb{R}^{n \times n} \) which associates weights \( w_{ij} \) to...
the edge $e_{ij}$ between the nodes $v_i$ and $v_j$.

Specifically, the nodes of $G$ are defined as the Lagrangian particles, i.e., $v_i = x^i$. The edges between these nodes have the associated weights

$$w_{ij} = 1 / r_{ij} \quad i \neq j,$$

for $w_{ij} = 1 / r_{ij}$ for $i \neq j$, expressing pairwise similarities between distinct Lagrangian particles. Other definitions of similarity are also possible. In general, converting distance to similarity can be done via any monotonically decreasing function, as long as the distance function $r_{ij}$ is a metric, i.e., it satisfies for all points in the space the metric axioms of identity, non-negativity, symmetry, and triangle inequality. This is according to the intuition that the ordering of graph nodes from most dissimilar to least dissimilar should be preserved through the similarity conversion.

Extending the present similarity definition (4.1) to the diagonal of $W$ would yield infinitely large quantities. To regularize $W$, we set the diagonal elements to a large constant $w_{ii} = K \gg 1$, $i = 1, \ldots, n$. As we shall see later, the actual value of $K$ is immaterial in our algorithm.

The entries of $W$ characterize the likelihood of nodes $v_i$ and $v_j$ to be in the same coherence cluster. By construction, $W$ is nonnegative ($w_{ij} \geq 0$) and symmetric ($W = W^\top$, with the superscript $\top$ referring to matrix transposition).

The degree of a node $v_i \in V$ is defined as

$$\text{deg}(v_i) := \sum_{j=1}^{n} w_{ij}.$$

Subsequently, the degree matrix $D$ is defined as the diagonal matrix with the degrees $\text{deg}(v_i)$ on the diagonal. For a subset $A \subset V$ of nodes, we denote its complement in $V$ by $\overline{A}$. We measure the size of $A$ by two different quantities:

$$|A| := \{i; v_i \in A\},$$

$$\text{vol}(A) := \sum_{i \in A} \text{deg}(v_i).$$

Here, $|A|$ measures the size of $A$ by its number of nodes, while $\text{vol}(A)$ measures the size of $A$ by summing over the weights of all edges attached to nodes in $A$. 
4.2.3 Graph sparsification

For large data sets, storing all entries of the similarity matrix $W$ is prohibitive. For instance, storing $n = 10^6$ elements with double precision requires 8 Terabytes of memory, which clearly exceeds the capacity of today’s typical personal computers [39].

To address this issue, techniques have been developed to sparsify $W$ by retaining only elements describing strong enough similarity. Two widely-used approaches are the $k$-nearest neighbors and the $\epsilon$-neighborhood approaches [185]. In the former, $w_{ij}$ is retained if $v_j$ (or $v_i$) is among the $k$ nearest neighbors of $v_i$ (or $v_j$), $k \ll n$. In the latter, $w_{ij}$ is retained if it exceeds a specified threshold $\epsilon$. All other $w_{ij}$ entries are set to zero and hence require no storage. Other advanced sparsification approaches include random sampling [114], sampling in proportion to edge connectivities [17], sampling in proportion to the effective resistance of an edge [175], and sampling using relative neighborhood graphs [110, 2, 82].

Here we select the $\epsilon$-neighborhood approach because of its low computational cost. For the practical determination of nearest neighbors, a number of efficient packages are available [84, 144].

4.2.4 Spectral clustering

With the notation developed so far, our original Problem 1 can be re-formulated as follows.

**Problem 2.** [Similarity graph clustering] Given a similarity graph, find a partition of the set of its nodes into clusters such that both of the following hold:

1. Nodes in the same cluster are similar to each other, which aims to maximize the within-cluster similarities.

2. Nodes in a cluster are dissimilar from those located in other clusters or those not included in any cluster (incoherent background), which aims to minimize the between-cluster similarities.

These two requirements for clusters implement Principle 2 and Principle 3, respectively. A particularly efficient method to identify clusters is spectral clustering, which we discuss below (see also [185] for a review).

Spectral clustering and optimal graph cuts

Given a similarity graph $G = (V, E, W)$, a graph cut is a partition of the set of nodes $V$ into two (or possibly more) subsets $A$ and $B$. To such a partition, we assign a weight cut $W(A, B)$ defined
as the sum of the edge weights between two sets $A$ and $B$, i.e.,

$$W(A, B) := \sum_{i \in A, j \in B} w_{ij}. $$

Now, consider a subset of graph nodes with very high within-group similarity and with weak connections to its complement, such as the orange set in fig. 4.1. A graph cut separating this subset from the rest of the graph (such as the cut indicated by the red dashed line) then yields a much smaller weight cut $W(A, \overline{A})$ than another graph cut through $A$, which would necessarily cut some of the strong connections within $A$.

This suggests the following minimization problem, also known as the mincut problem, as a solution of Problem 2: For a given number $k$ of subsets, the mincut problem is to find a partition $A_1, ..., A_k$ of $V$ which minimizes

$$\text{cut}(A_1, ..., A_k) = \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A_i}).$$

(4.2)

For $k = 2$, the mincut problem can be solved very efficiently (see, e.g., [176]). In practice, however, the solution of the mincut problem often just separates one individual node (the one with weakest connections) from the rest of the graph. One way to circumvent this problem is to penalize the smallness of sets in candidate partitions. The most commonly applied objective functions that implement this idea are the normalized cut [171], or $NCut$ for short, RatioCut [94], MinMaxCut [56] and Cheeger ratio cut [37]. Notably, not all of these graph cut objective functions have solutions which satisfy both conditions in Problem 2 (cf. [185] for more details).

In this chapter, we use the $NCut$ objective function, whose (approximate) solutions maximize
the within-cluster similarity and minimize the between-cluster similarity:

$$\text{NCut}(A_i, ..., A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A}_i)}{\text{vol}(A_i)},$$

Introducing the penalizing balancing conditions, however, turns the originally simple mincut problem into an NP hard problem [187]. Spectral clustering is a way to solve relaxed versions of balanced graph cut problems.

**Graph Laplacian**

Shi & Malik [171] showed that the solution of the Ncut problem can be approximated by solutions of the generalized eigenproblem associated with the (unnormalized) graph Laplacian $L = D - W$, where $D$ is the diagonal degree matrix of node degrees and $W$ is the similarity matrix defined earlier.

The *generalized eigenvalue problem* for the graph Laplacian is then defined as

$$Lu = \lambda Du. \quad (4.3)$$

We refer to its solutions as generalized eigenvectors for short. Generalized eigenvectors $u$ then offer an alternative representation of the weighted graph data. As we will see in the next sections, this change of representation enhances the cluster-properties in the data, so that clusters can be easily detected in the new representation. In particular, the simple K-means clustering algorithm has no difficulties to detect the clusters in this new representation (see Section 4.2.6 regarding K-means clustering).

It is known from Spectral Graph Theory [41] that the eigenvalues solving (4.3) satisfy $0 = \lambda_1 \leq ... \leq \lambda_n$. If the underlying graph consists of $k$ disconnected components (clusters with zero between-cluster similarity), then $\lambda = 0$ is a generalized eigenvalue of multiplicity $k$. In that case, the eigenspace corresponding to this eigenvalue is spanned by the indicator vectors of the individual connected components. A perturbation argument implies that if the between-cluster similarities remain small, then the eigenvectors of the first $k$ eigenvalues remain close to indicator type [185]. This enables reconstructing the clusters from the first $k$ eigenvectors obtained from (4.3). The main challenge, therefore, is to extract a meaningful number of clusters directly from the data, as opposed to postulating its value beforehand.
4.2.5 Estimating the number of clusters by eigenspace analysis

For a predetermined number $k$, the spectral clustering algorithm of Shi & Malik [171] collects the $k$ dominant generalized eigenvectors as cluster indicators in a matrix $U = (u_1, \ldots, u_k) \in \mathbb{R}^{n \times k}$. To retrieve $k$ from the graph data, we adopt here the eigengap heuristic [25] by which

$$k = \arg \min \max_i (g_i),$$

where $g_i = \lambda_{i+1} - \lambda_i$ for $i = 1, \ldots, n$. In other words, $k$ is simply determined as the number of eigenvalues preceding the largest gap in the eigenvalue sequence. The presence of such a gap enables us to invoke the perturbation argument of the previous section, and argue that our graph $G = (V, E, W)$ is a perturbation of one with $k$ disconnected components.

Expression (4.4) determines the number of coherent clusters satisfying the definition given in Section 4.2.4. Ultimately, however, we need to partition the graph $G = (V, E, W)$ into $k + 1$ clusters to also account for the incoherent cluster surrounding the coherent clusters, as codified in our Principle 3. We refer to the last, $(k + 1)$st cluster arising in this process as the noise cluster or incoherent cluster since it includes nodes that do not belong to any coherent cluster.

Spectral gap arguments were used before in the context of dynamical systems (see [52, 50, 75, 76] for examples). However, the number of cluster indicators (leading singular- and eigenvectors) in these works does not necessarily coincide with the number of clusters need to be extracted (see [165] and Section 4.3.1 for more details).

Remark 1. As discussed, we identify the number of vortices present in a given domain by locating the largest gap in the eigenvalue sequence. This implies that the number of eigenvalues and eigenvectors to be computed should be greater than the maximum number of vortices expected to be present in a domain. In the absence of intuition for the maximum number of vortices, one needs to conduct a full matrix decomposition instead of a partial decomposition. The computational cost of such a decomposition, however, increases dramatically with respect to the number of eigenvalues to be computed (see [34] for more information).

4.2.6 Retrieving clusters from matrix $U$ by K-means clustering

As a last step, we employ K-means clustering to convert relaxed continuous spectral vectors, corresponding to $U$’s $k$ columns, into a discrete cluster indicator vector containing the cluster assignment for each node $x_i$.

Given the spectral vectors $U \in \mathbb{R}^{n \times k}$ and integer $K$, K-means clustering aims to determine $K$ points in $\mathbb{R}^k$, called centers, so as to minimize the mean squared distance from each node to its nearest center. In 1957 Stuart Lloyd [132] suggested a simple iterative algorithm which
efficiently finds a local minimum for this problem. Given any set of $K$ centers, the algorithm proceeds by alternating between the following two steps:

**Assignment:** find each node’s nearest center and assigns it to the corresponding cluster.

**Update:** recalculate cluster centers by measuring the mean of all nodes included in each cluster.

These steps repeat until no node is reassigned. Readers not familiar with K-means can read about this algorithm in numerous text books, for example see [64]. Throughout the chapter, we choose the number of cluster centers $K$ equal to $k + 1$, where the last, $(k+1)$st cluster corresponds to the incoherent or noise cluster discussed in Section 4.2.5. Note that the K-means algorithm or its probabilistic counterpart fuzzy C-means have been used in the context of dynamical systems (see [79, 32, 75] for examples).

We summarize our numerical procedure in Algorithm 3.

**Algorithm 3**

**Input:** Similarity matrix $W \in \mathbb{R}^{n \times n}$ (cf. Section 4.2.2)

1. Sparsify $W$ by using the NCut algorithm (cf. Section 4.2.3). Remove isolated nodes, i.e., nodes with degree zero, from $G = (V, E, W)$.

2. Compute the graph Laplacian $L$, and solve the generalized eigenvalue problem $Lu = \lambda Du$.

3. Identify the number $k$ of coherent clusters as the number of eigenvalues preceding the largest gap among the increasingly ordered eigenvalues. Select the first $k$ generalized eigenvectors $u_1, \ldots, u_k$ as coherent cluster indicators.

4. Assemble the matrix $U = (u_1, \ldots, u_k)$. Each row of $U$ corresponds to a graph node (excluding the isolated nodes). Apply K-means to the first $k$ eigenvectors and extract $k + 1$ clusters. The last cluster is the incoherent cluster and corresponds to the mixing region filling the space between coherent clusters.

**Output:** Clusters $C_1, \ldots, C_{k+1}$.

**4.2.7 Large-scale spectral clustering**

For large data sets, considerable time and memory is required to compute and store the similarity matrix $W$ and the graph Laplacian $L$. The most commonly used approach to address this issue is graph sparsification, as discussed earlier in Section 4.2.3. From the sparse similarity matrix $W$ so obtained, one determines the corresponding Laplacian matrix $L$, and calls a sparse eigenvalue solver.

Even after the sparsification of $W$, however, calculating the generalized eigenvectors of the graph Laplacian $L$ remains challenging with $O(n^3)$ worst-case complexity [39]. Several authors
Figure 4.2: Partitioning of a bipartite graph $G_B = (V_B, E_B, W_B)$ whose set of nodes $V_B$ is divided into two disjoint sets $A$ and $B$ such that $V_B = A \cup B$. The dashed line shows the solution of normalized graph cut yielding a simultaneous decomposition of $A$ and $B$.

\[ 39 \] \[ 174 \] tried to alleviate the problem by adapting standard eigenvalue solvers to distributed architecture. Other approaches are designed to achieve efficiency by finding numerical approximations to eigenfunction problems \[ 71 \] \[ 38 \] \[ 131 \].

Here, we adopt a low-rank matrix approximation approach. The main idea is to coarse-grain the similarity graph $G = (V, E, W)$, while keeping as much information as possible from the original graph and its weights. To this end, we construct a bipartite graph $G_B = (V_B, E_B, W_B)$ from the original similarity graph by uniform spatial sampling of $q$ graph nodes, called supernodes, from $n$ graph nodes, where $q \ll n$ \[ 33 \] \[ 129 \]. A bipartite graph is a graph whose set of nodes $V_B$ admits a partition into two disjoint sets, $A$ and $B$, such that each edge connects a node in $A$ to one in $B$. As a result, no two nodes within $A$ and within $B$ are connected by an edge. Here, we set $A$ as the set of all $n$ original graph nodes, and $B$ as its subset of $q$ supernodes, considered as independent copies. The weights are now defined as before, such that the square $(n + q) \times (n + q)$ similarity matrix $W_B$ of the bipartite graph can be written as

$$W_B = \begin{pmatrix} 0 & Z^\top \\ Z & 0 \end{pmatrix}$$

where $Z \in \mathbb{R}^{q \times n}$ is a tight similarity matrix containing the edge weights between all nodes and supernodes, i.e., between $A$ and $B$. Now, one can pose the Ncut problem to the bipartite graph whose similarity matrix enjoys a simple block-structure. As shown by Dhillon \[ 55 \], this block-structure breaks the associated Ncut problem into two parts such that the dominant right singular vectors of the normalized $q \times n$ tight similarity matrix $\hat{Z} = D_2^{-1/2}ZD_1^{-1/2}$ play the role of the generalized eigenvectors of the graph Laplacian in Section \[ 4.2.4 \]. Here, $D_1$ is an $n \times n$
diagonal matrix whose entries are column sums of $Z$ and $D_2$ is a $q \times q$ diagonal matrix whose entries are row sums of $Z$ (see Section 4.B for more details).

We now summarize our algorithm for large-scale trajectory data sets.

**Algorithm 4**

1. Select uniformly $q$ supernodes from $n$ graph nodes.
2. Construct a tight similarity matrix $Z \in \mathbb{R}^{q \times n}$ between all original graph nodes and the supernodes.
3. Given $Z$, form $\hat{Z} = D_2^{-1/2}ZD_1^{-1/2}$. Compute the singular values and vectors of $\hat{Z}$. Select the first $k$ right singular vectors $u_1, \ldots, u_k$ as cluster indicators for the original graph.
4. Assemble the matrix $U = (u_1, \ldots, u_k)$. Each row of $U$ corresponds to a graph node. Apply K-means to the first $k$ right singular vectors and extract $k + 1$ clusters. The last cluster is the incoherent cluster and corresponds to the mixing region filling the space between coherent clusters.

Output: Clusters $C_1, \ldots, C_{k+1}$.

### 4.3 Related previous work

#### 4.3.1 The transfer-operator approach

In the transfer operator-based approach [80, 74, 78] finite-time coherent sets are defined as regions in phase space that minimally diffuse with the surrounding phase space during a finite time interval. The method builds on the Perron-Frobenius operator or transfer operator, which describes the evolution of material densities under the flow map.

In practice, the infinite-dimensional transfer operator needs to be approximated by a finite-dimensional matrix, the transition matrix $P$, which is most commonly obtained from a partition of the flow domain $(B_i)_i$ and the flow image $(C_j)_j$ into distinct boxes, and subsequent computation of discrete transition probabilities: the transition matrix entry $P_{ij}$ is computed as the number of particles transported from $B_i$ to $C_j$, normalized by the total number of particles released from $B_i$ (see fig. 4.3). This box partitioning is also referred to as Ulam’s method, and introduces (numerical) diffusion at the implementation level [80].

In our context, the transition matrix $P$ can be interpreted as the tight similarity matrix $Z$ of a bipartite graph $G_B$ as follows: define the first set of nodes $A$ as the collection of initial boxes $B_i$, the second set of nodes $B$ as the collection of final boxes $C_j$, and the edge weights as $Z_{ij} = P_{ij}$.
see fig. 4.2. Note that the connection to graphs has been observed earlier in [29], but has been interpreted differently as a directed graph instead of a bipartite graph.

**Remark 2.** The size and sparsity of the resulting weight matrix depend on the size of the $B_i$’s and $C_j$’s, as well as on the underlying dynamics of the system. For instance, in the presence of chaotic dynamics, particles released at the initial time can scatter in a large domain. This, in return, may require a large number of boxes $C_j$ to cover the final domain, and results in a large number of columns in the subsequent transition matrix. Moreover, it requires a large number of initial particles released from each box $B_i$ to accurately compute the transition probabilities, which likely decreases the sparseness of the transition matrix $P$ (cf. Section 4.4.1 for an example). In contrast, the size of the weight matrix of Algorithms 3 and 4 depends on the number of tracked particles in a controllable fashion.

With this bipartite graph construction, the optimization problem which is underlying the definition of a coherent set in the transfer-operator setting can be reformulated as a clustering problem. In a (bipartite) graph cut, such as the one shown in fig. 4.2, the weight of the cut can be interpreted as the mass leakage of one set with its complement.

As discussed in Section 4.A, minimizing the normalized cut for a binary cluster indicator is NP-hard. Relaxation of the binary cluster indicator in the real value domain yields the eigenvector corresponding to the second smallest eigenvalue of $L$ as an approximate cluster indicator [171]. However, in order to obtain a partition of the graph, we need to re-transform the real-valued cluster indicator vector of the relaxed problem into a discrete indicator vector. The simplest way to do this is to use the sign of the eigenvector as a discrete cluster indicator function [171]. Alternatively, one can search for a splitting point such that the resulting partition has the best $\text{NCut}(A, \overline{A})$ value [171], or apply the line-search algorithm of [80]. Viewing the transfer operator approach [80] as a bipartite spectral graph partitioning [55], one can similarly recover discrete cluster indicator vectors from real-valued singular vectors. Figure 4.4a shows the second largest singular vector of the normalized transition matrix for the Bickley jet model discussed in Section 4.4.2. We obtain the corresponding binary cluster indicator by searching through all possible $\text{Ncuts}$ [171]. As shown in fig. 4.4b, the binary cluster indicator so obtained highlights two coherent sets in which coherent vortices still remain hidden. Therefore, we step in hierarchy of increasingly ordered singular vectors, and search for these vortices in the third singular vector, shown in fig. 4.4c. Similarly, we extract the corresponding discrete-valued indicator vector by examining all possible Ncuts (see fig. 4.4d). Figure 4.4d reveals that the yellow set, as a single entity, is composed of two vortices. The two vortices forming the yellow set have overall small mass exchange with the blue set, implying that the objective function is minimized. The spatial connectedness, however, appears to be missing in this solution as the yellow
Chapter 4. Spectral-Clustering Approach to Lagrangian Vortex Detection

Figure 4.3: Interpreting transition matrix constructed from tracer advection as tight similarity matrix $Z$ of a bipartite graph.

Figure 4.4: (a,c) Second and third largest (left) singular vectors of the normalized transition matrix for the Bickley jet flow. (b,d) Corresponding coherent sets which are obtained by searching through all possible cuts \[171\]. To compute the transition matrix, we subdivided the domain into a grid of $400 \times 120$ identical boxes, and released 400 particles in each box. We then advected particles from $t_0 = 0$ to $t = 40$ days.

set is composed of two distant vortices. In other words, a search for sets with minimal mass exchange, without enforcing spatial connectedness, can lead to a union of coherent structures as a solution. In this case, applying K-means clustering to the collection of leading singular vectors may resolve the issue. However, the number of coherent structures that needs to be extracted, may not be detectable anymore with the eigengap heuristic (see Section 4.2.5), as each singular vector highlights a combination of coherent structures. In fact, the number of coherent structures generally does not coincide with the number of singular/eigenvalues preceding the largest eigengap (see \[75\], pp. 1852-1853), and therefore needs to be guessed or to be known a priori.
The graph cut approach, in general, does not guarantee that the resulting cluster will form a connected region in the physical space. The connectedness constraint, however, can be enforced indirectly during the construction of the similarity graph. Our method specifically enforces this constraint by measuring and penalizing distances in the spatio-temporal domain. As a result, unlike for the transfer operator method, a union of coherent structures is not a solution for our method. In Section 4.4.2, we will apply our Algorithm 3 to the same Bickley jet flow considered earlier in fig. 4.4.

4.3.2 Hierarchical partitioning of the transfer-operator

In the spectral clustering community, one distinguishes between two approaches to detect a specified number of clusters in a given similarity graph using the graph cut procedure [171, 55, 185]: two-way clustering and multi-way clustering. Our methodology presented in Section 4.2 follows (up to the introduction of the incoherent cluster) the multi-way clustering approach, in which \( k \) clusters are retrieved from the \( k \) dominant eigenvectors at once.

In two-way clustering, the following procedure is applied recursively to generate multiple clusters: (i) compute the top generalized eigenvector of the unnormalized graph Laplacian, and (ii) bisect the graph into two sub-graphs. In the transfer-operator context, this procedure has been put forward in [134] and is stopped when the obtained partitions no longer satisfy a pre-specified coherence ratio (cf. [134] for details). In the clustering analysis community, two-way clustering is also found to be inefficient due to the fact that separate eigenvalue problems need to be solved repeatedly [35, 145, 171].

4.3.3 Application of Infomap to the transfer operator

As pointed out in [166] and mentioned in Sections 4.3.1 and 4.3.2, one cannot determine the number of existing coherent structures in a domain within the transfer-operator approach. This fact and the need to address within-cluster similarities, led Ser-Giacomi et al. [166] to devise a network tool called Infomap to detect coherent structures as communities in the graph defined by the transition matrix \( P \). Specifically, Ser-Giacomi et al. [166] construct the transition matrix \( P \) for a special case when the fluid domain is invariant and \( C_j \) can be chosen equal to \( B_i \). In this case, the subsequent square transition matrix \( P \) is viewed as a similarity matrix of a directed graph, whose edges retain both directions and weights.

Viewed as a directed graph, Infomap detects communities by taking random walks on graph edges connecting initial and final times reciprocally, while the transition matrix \( P \) is fixed. However, moving from one state to another without changing the transition matrix \( P \) is equivalent to approximating an unsteady flow with a time-periodic one, which may not have similar coherent
structures.

In the absence of an incoherent cluster/community, the Infomap algorithm seeks to find a partition of the domain into communities, each of which is subject to the same coherence or optimality principle. It is intuitively clear, however, that the incoherent fluid background as a whole does not satisfy the same coherence principles as the coherent regions, which is reflected by partially low coherence ratios in [166, Fig. 10]. This also poses significant challenges in a direct application of classic clustering algorithms to trajectory data sets.

### 4.3.4 Application of fuzzy clustering to a trajectory data set

Recently, Froyland & Padberg-Gehle [79] proposed a method based on traditional fuzzy C-means clustering [24, 61] to identify regions of phase space that remain compact over a finite time interval. Specifically, they first build a trajectory data set $X \in \mathbb{R}^{n \times dm}$ whose rows are vectors $(X_i)_{i=1,...,n}$ containing concatenated positions of Lagrangian particles in time. Second, they apply the C-means algorithm, with a prespecified number of clusters $K$ and a set of $K$ initial starting points in $\mathbb{R}^{dm}$, to the trajectory data set. The result is a membership value describing the likelihood that a trajectory belongs to a cluster. Thus, each trajectory carries $K$ membership values, showing the degree of belonging to each of the $K$ clusters. Finally, each trajectory is assigned to only one cluster based on the maximum membership value it carries. Those trajectories carrying low membership values for all clusters are occasionally considered to be non-coherent (see [79] for more details).

Compared with the fuzzy C-mean clustering used in [79], the spectral clustering technique considers the connectedness of the data, whereas the C-means clustering method considers the compactness of the data. Fuzzy C-means algorithm optimizes cluster compactness by assessing the proximity between the uncertain data points assigned to the cluster and the corresponding cluster center. We note that cluster centers are not true trajectories of a dynamical system although they are in the trajectory space [79]. In contrast, our spectral clustering technique maximizes connectedness inside clusters and disconnectedness between clusters at the same time by measuring pairwise distances between trajectories.

As opposed to centroid-based clustering algorithms such as K-means or C-means, where the resulting clusters tend to be convex sets [79, 138, 109], spectral clustering can find any cluster shape, because it has no preference for the shape of the cluster. This is important as we will show in Section 4.4.3 that vortices with non-convex shapes are the rule rather than the exception considering the known vortex stirring in geophysical flows [5].

Most clustering methods including centroid-based methods are plagued with the problem of noisy data, i.e., identifying good clusters amongst noise points that just do not belong to any cluster [45]. In some cases, even a few noisy points or outliers may bias the final output of the
algorithm [45]. In our specific context, the noise corresponds to the incoherent or turbulence region itself, where particles do not remain compact. This implies that the turbulence region is not residing in a hypersphere, and consequently cannot be captured by adding an extra cluster to C-means or K-means algorithms (see [45] for more details).

On the other hand, the high dimensionality of the trajectory dataset poses a considerable challenge to K-means or C-means clustering approaches. First, the curse of dimensionality can cause slow convergence for these traditional algorithms, and, second, the existence of redundant subspaces may not allow for the identification of the underlying structure in the data (cf. [10] and [43], p. 10).

Similar to many clustering methods, the K-means or C-means algorithms assume that the number of clusters $K$ in the dataset is known beforehand which is not necessarily true in real-world applications. In contrast, the spectral clustering can detect the right number of clusters automatically using techniques such as the eigengap heuristic (cf. Section 4.2.5).

Finally, the result of K-means or C-means clustering, depends on the initial guess for the cluster centers [138, 109], and can reach a local minimum of the objective function instead of the desired global minimum [109, 195]. Often one restarts the procedure a number of times to mitigate the problem. However, when the number of clusters $K$ is large, the number of times to restart K-means or C-means to reach an optimum can be prohibitively high and lead to a substantial increase in runtime (cf. [195]).

### 4.4 Results

We demonstrate the implementation of Algorithms 3 and 4 on four examples to detect coherent Lagrangian vortices. In the first example, we consider a periodically forced pendulum for which we can explicitly confirm our results using an appropriately defined Poincaré map. Our second example is one whose temporal complexity is one level higher: the Bickley jet with quasi-periodic time dependence [161, 47]. In the third example, we detect coherent Lagrangian vortices in a quasigeostrophic ocean surface flow derived from satellite-based sea-surface height observations [81]. Our last example is a three-dimensional velocity field, the Arnold-Beltrami-Childress (ABC) flow, which is an exact solution of Euler’s equation [6]. This is our computationally most demanding example, where we deploy Algorithm 4 to reduce the graph size and the associated computational cost. For the rest of the examples, we use Algorithm 3 with the $\epsilon$-neighborhood graph sparsification approach described in Section 4.2.3.

To implement Algorithms 3 and 4 in the forthcoming examples, we use a variable-order Adams-Bashforth-Moulton solver (ODE113 in MATLAB) to solve the differential equations. The absolute and relative tolerances of the ODE solver are chosen as $10^{-6}$. In Section 4.4.3 we
obtain the velocity field at any given point by interpolating the velocity data set using bilinear interpolation.

The dynamic distances $r_{ij}$ can be computed using two approaches that differ in terms of memory consumption, suitability for parallel computation and accuracy. In the first approach, one builds a spatio-temporal trajectory data set by saving trajectory positions over $m$ intermediate times. One then measures pairwise distances using the trapezoidal rule and sparsifies them simultaneously. This can be done effectively using the ExhaustiveSearcher model object in MATLAB or other packages, such as [84, 144]. This approach is memory consuming but highly parallelizable.

In the second approach, one constructs the similarity matrix without building any spatio-temporal trajectory data set. To this end, one measures pairwise distances concurrent with the advection of particles. Specifically, one defines an extra output argument inside the ODE function which measures and cumulates the pairwise distances over a given time interval.

Compared with the first approach, the second approach is more accurate and more memory efficient. However, its parallel implementation requires communication between processors, which may make the computation prohibitively slow. For this reason, we only employ the second approach in our last example, the Arnold-Beltrami-Childress (ABC) flow, and use the first approach otherwise.

4.4.1 The periodically forced pendulum

Consider the periodically forced pendulum

$$
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\sin(x_1) + \varepsilon \cos(t).
\end{align*}
$$

For $\varepsilon = 0$, the system is integrable with hyperbolic fixed points at $(0, (2m - 1) \pi)$, and elliptic fixed points at $(0, 2m\pi)$, where $m \in \mathbb{Z}$. As is well known, there are two heteroclinic orbits connecting each successive pair of hyperbolic fixed points, enclosing an elliptic fixed point, which is in turn surrounded by periodic orbits. These periodic orbits appear as closed invariant curves for the Poincaré map $P := F^2_0$. The fixed points of the flow are also fixed points of $P$.

Kolmogorov-Arnold-Moser (KAM) theory [7] guarantees the survival of most closed invariant sets for $P$ and $0 < \varepsilon \ll 1$. Increasing the perturbation strength $\varepsilon$ further leads to the appearance of resonance islands [8, 26] and to the coexistence of regular and chaotic particle trajectories, as one would expect in a turbulent fluid flow containing coherent structures. Figure 4.7b shows these surviving invariant sets (KAM tori and resonance islands) of the Poincaré map $P$ obtained for $\varepsilon = 0.4$, obtained from 800 iterations of $P$. This many iterations are required to obtain
Chapter 4. Spectral-Clustering Approach to Lagrangian Vortex Detection

Figure 4.5: Comparison of three different diagnostic fields for the periodically forced pendulum. The scalar fields are constructed for the same integration time $T = 800 \times 2\pi$. (a) Forward-time connectivity field. (b) Forward-time FTLE field. (c) Forward-time FSLE field.

Figure 4.6: (a) Sorted generalized eigenvalues for graph Laplacian $L$ for the periodically forced pendulum. (b-c) The first and ninth generalized eigenvectors of graph Laplacian $L$. Isolated points resulting from the graph sparsification are shown in white.

Figure 4.7: (a) Ten clusters extracted by K-means clustering ($k = 10$) from the first nine generalized eigenvectors of graph Laplacian $L$ for the periodically forced pendulum. The tenth cluster corresponds to the chaotic sea filling the space between elliptic regions. (b) 800 iterations of the Poincaré map for the periodically forced pendulum. (c) Computed clusters, compared with the Poincaré map computed for the same integration time (eight hundred iterates).
continuous-looking boundaries of the various coherent regions. We would like to capture the surviving KAM regions as coherent clusters using Algorithm 3.

To construct the pairwise dynamic distances $r_{ij}$ and subsequent similarity matrix $W$, we advect 90,000 particles, distributed initially over a uniform grid $G^0$ of $300 \times 300$ points, from $t_0 = 0$ to $t_1 = 800 \times 2\pi$. The spatial domain ranges from $-2.6$ to $-0.3$ in $x_1$ direction and from $-1.2$ to $1.2$ in $x_2$ direction. We output the trajectory data with 3600 intermediate points, evenly spaced in time. Moreover, we sparsify edges from the complete graph representing a distance greater than $\epsilon = 0.45$.

Figure 4.5a shows the degree of connectivity of graph nodes, $\deg(v_i)$, as a scalar field. We refer to this scalar field here and in our later examples as connectivity field. This field looks generally smoother than other diagnostic fields, such as the finite-time Lyapunov exponent [104, 95] or finite-size Lyapunov exponent [14, 9] fields (see fig. 4.5). The smoothness of the connectivity field is the result of two averaging processes which attenuate computational and in-situ measurement noises. The first averaging process happens as we integrate Euclidean distances between graph nodes over time. The second averaging takes place once we compute $d_i$, i.e., when summing the edge weights connected to a node $v_i$.

Figure 4.6a shows the first 20 generalized eigenvalues as a function of their indices. We can see that the first nine eigenvalues are very close to 1, while the tenth has an appreciable difference, creating the largest gap in the eigenvalue plot. This eigengap implies that the first nine eigenvectors are cluster indicators from which coherent structures should be extracted. For example, figs. 4.6b and 4.6c show the first and ninth generalized eigenvector of the graph Laplacian $L$.

Finally, fig. 4.7a shows the ten clusters extracted by the K-means algorithm from the first nine generalized eigenvectors of graph Laplacian $L$. The tenth cluster corresponds to the chaotic background filling the space between the coherent clusters. In fig. 4.7c the extracted clusters are superimposed on the Poincaré map, showing close agreement with the Lagrangian vortices of this example, i.e., the elliptic islands. Figure 4.8a shows the execution times for three major steps of Algorithm 3 as a function of increasing spatial resolution of the graph nodes. The main computational bottleneck, as shown in the figure, is computing the pairwise distances and subsequently the similarity matrix $W$. For this purpose, we utilized parallel computing techniques with 300 CPUs, with each processor just computing a few rows/columns of the sparse similarity matrix. Figure 4.8a shows the averaged CPU-times spent on each processor on carrying out the particle advection, sparse similarity matrix construction and eigen-decomposition.

Figure 4.8b shows the sensitivity of the clustering results to the choice of the neighborhood radius used to sparsify the pairwise distances $r_{ij}$. In particular, the figure shows how the averaged within-class similarities of coherent sets change with respect to the choice of neighborhood radius.
radius. Figure 4.8b suggests the existence of a critical radius below which the size and shape of clusters can change. This critical radius simply corresponds to a distance where even strong edges within coherent sets are affected by graph sparsification. It is important to choose the sparsification radius such that strong edges will be maintained. As a rule of thumb, we set the sparsification radius such that only 5%-10% of the elements in the similarity matrix \( W \) will be kept. To estimate such a radius, one can compute the pairwise distances for a subsample of the original graph (e.g., 40 nodes) and choose the sparsification radius accordingly.

### 4.4.2 Quasiperiodic Bickley jet

Next, we consider the Bickley jet, an idealized model of a meandering zonal jet flanked above and below by counter rotating vortices [47, 161]. This model consists of a steady background flow subject to a time-dependent perturbation. The time-dependent Hamiltonian for this model reads as

\[
\psi(x, y, t) = \psi_0(y) + \psi_1(x, y, t),
\]

\[
\psi_0(y) = -U_0 L_0 \tanh \left( \frac{y}{L_0} \right),
\]

\[
\psi_1(x, y, t) = U_0 L_0 \text{sech}^2 \left( \frac{y}{L_0} \right) \Re \left[ \sum_{n=1}^{3} f_n(t) \exp(ik_n x) \right],
\]

where \( \psi_0 \) is the steady background flow and \( \psi_1 \) is the perturbation. The constants \( U_0 \) and \( L_0 \) are characteristic velocity and characteristic length scale, respectively. For the following analysis,
we apply the set of parameters used in [161]:

\[ U_0 = 62.66 \text{ ms}^{-1}, \quad L_0 = 1770 \text{ km}, \quad k_n = 2n/r_0, \]

where \( r_0 = 6371 \text{ km} \) is the mean radius of the earth.

For \( f_n(t) = \varepsilon_n \exp(-ik_n c_n t) \), the time-dependent part of the Hamiltonian consists of three Rossby waves with wave numbers \( k_n \) traveling at speeds \( c_n \). The amplitude of each Rossby wave is determined by the parameters \( \varepsilon_n \). Specifically, the parameter values used are:

\[ c_1 = 0.1446 U_0, \quad c_2 = 0.205 U_0, \quad c_3 = 0.461 U_0, \quad l_y = 1.77 \times 10^6, \quad \varepsilon_1 = 0.0075, \quad \varepsilon_2 = 0.15, \quad \varepsilon_3 = 0.3, \quad l_x = 6.371 \times 10^6 \pi, \]

\[ k_n = 2n\pi/l_x. \]

To construct the dynamic distances \( r_{ij} \) and the similarity matrix \( W \), we advect 48000 particles, distributed initially over a uniform grid of \( 400 \times 120 \) points, from \( t_0 = 0 \) to \( t = 40 \) days. The spatial domain \( U \) ranges from 0 to 20 in \( x \) direction and from \(-3 \) to \( 3 \) in \( y \) direction. We output the trajectory data with 600 intermediate points, evenly spaced in time. Moreover, we sparsify edges from the complete graph representing a distance greater than \( \epsilon = 3 \).

In fig. 4.9, we show the first 20 generalized eigenvalues of the graph Laplacian \( L \) with respect to their indices. We can observe that the largest eigengap is between the sixth and seventh generalized eigenvalues, signaling the presence of six coherent clusters in the domain. Hence, we extract seven clusters from the first six generalized eigenvectors shown in figs. 4.10a to 4.10f. The last cluster, as described earlier in Section 4.2.5, corresponds to the incoherent region filling the space between the coherent vortices. The observed fuzziness of the vortex boundary region is due to the fact that coherent and incoherent motion is, on the chosen time interval, not as distinguished as in the forced pendulum example considered in the previous section. After all, this distinction is retrieved from the trajectory data, as opposed to being imposed externally through some threshold, for instance. Interestingly, this dynamic distinction is very clear in the ocean example considered in the next section, which results in very pronounced cluster indicators.

Figure 4.11a shows the identified clusters at the initial time, and fig. 4.11b shows them at the final time, confirming the coherence of extracted vortices over the 40-day period. The complete advection sequence over 40 days is available in the online supplemental movie M1.

### 4.4.3 An ocean surface data set

Next, we apply Algorithm 3 to a two-dimensional unsteady velocity data set obtained from AVISO satellite altimetry measurements [122]. The domain of the data set is the Agulhas leakage in the Southern Ocean, characterized by large coherent eddies that pinch off from the Agulhas current of the Indian Ocean.

Here, we show how our coherent Lagrangian vortex detection principle uncovers the ma-
Chapter 4. Spectral-Clustering Approach to Lagrangian Vortex Detection

*Figure 4.9:* Sorted generalized eigenvalues for the graph Laplacian $L$ for the quasiperiodic Bickley jet flow.

*Figure 4.10:* The leading generalized eigenvectors of the graph Laplacian $L$ for the Bickley jet flow.

*Figure 4.11:* (a) Seven clusters extracted by K-means clustering from the first six generalized eigenvectors of graph Laplacian $L$ at initial time, $t_0 = 0$. The seventh cluster corresponds to the mixing region filling the space between the coherent clusters. (b) The same clusters advected passively to the final time, $t = 40$ day. The complete advection sequence over 40 days is illustrated in the online supplemental movie M1.
terial eddies over integration time of 168 days, ranging from $t_0 = 11$ January 2006 to $t = 28$ June 2006. The South Atlantic ocean region in question is bounded by longitudes $[8.5^\circ E, 12^\circ E]$ and latitudes $[45^\circ S, 39^\circ S]$. The region in question is chosen away from the coast so that particle positions will be available for the entire integration time. Otherwise, one has to discard those particles hitting obstacles or the coast at some intermediate times from the computation. We compute the pairwise accumulative distances over a uniform grid of $120 \times 180$ points using a trajectory data set composed of 600 evenly spaced intermediate times. We sparsify edges from the complete graph representing a distance greater than $\epsilon = 1$.

Figure 4.12 compares the connectivity field with the FTLE and FSLE fields. Note that we view the connectivity field as a simple visualization tool from which one may diagnose the existence of coherent structures before taking the eigendecomposition step.

In fig. 4.13a, we show the first 20 generalized eigenvalues of the graph Laplacian $L$. We can observe that the largest eigengap exists between the second and third generalized eigenvalues, signaling the presence of two coherent clusters in the domain, which are indicated by the corresponding generalized eigenvectors (see figs. 4.13b and 4.13c).

Figure 4.14a show the coherent vortices extracted from the first two generalized eigenvectors of graph Laplacian $L$ at initial time $t_0 = 11$ January 2006 and final time $t = 28$ June 2006 respectively. In fig. 4.14b, we confirm the coherence of extracted vortices by advecting them to the final time $t = 28$ June 2006.

Interestingly, the coherent cluster shown in blue contains isolated points located far away from the cluster core (see fig. 4.14c). The presence of isolated points in a given cluster, however, seems to be unphysical due to the continuity of fluid flows. To investigate the true nature of these isolated points, we repeat our computation with a higher resolution, over a uniform grid of $300 \times 300$ points, ranging from $[8.5^\circ E, 12^\circ E]$ in longitudes and from $[45^\circ S, 39^\circ S]$ in latitudes (see fig. 4.14d). The higher resolution computation reveals that the previously detected isolated points are part of a narrow fingering emanating from the core of the blue cluster. This is in line with the known vortex stirring reported by several authors (see [5], for example). [htbp]

Despite the strange fingering-type appearance, the cluster remains highly coherent over the extraction period of 168 days. The complete advection sequence over 168 days is illustrated in the online supplemental movies M2 and M3.

This example underlines that a Lagrangian vortical region can have an instantaneously non-convex geometry. It may also, over time, absorb an initial finger-type protrusion and form a convex circular boundary in the end. This illustrates that while requiring convexity [157, 103], lack of filamentation [102], or shape coherence [135] of the vortex boundary may yield boundaries meeting high coherence requirements, they will not necessarily identify the largest set of trajectories forming a coherent cluster.
Chapter 4. Spectral-Clustering Approach to Lagrangian Vortex Detection

Figure 4.12: Comparison of three different diagnostic fields for the ocean data set. The scalar fields are constructed for the same integration time $T = 168$ days. (a) Forward-time connectivity field. (b) Forward-time FTLE field. (c) Forward-time FSLE field.

Figure 4.13: (a) Sorted generalized eigenvalues for the graph Laplacian $L$ for the ocean data set. (b-c) The first two generalized eigenvectors.

Figure 4.14: (a) Coherent vortices at initial time $t_0 = 11$ January 2006. (b) Advected image of the vortices at the final time $t = 28$ June 2006. (c) Magnification of the blue cluster shown in the first panel. The figure shows some isolated points located far from the cluster core. (d) Corresponding cluster obtained from a higher resolution computation, revealing that the previously detected isolated points are part of a narrow fingering emanating from the cluster core. The complete advection sequence over 168 days is illustrated in the online supplemental movies M2 and M3.
Finally, we repeat our computation with a sparse trajectory data set, composed of 57 particles distributed non-uniformly on an unstructured grid. Here, we select the number of intermediate times $m$, and sparsification distance $\epsilon$ similar to our earlier computation. Figure 4.15 shows the clustering result, with fig. 4.14a shown in the background for comparison.

4.4.4 The ABC flow

As a last example, we consider the steady Arnold-Beltrami-Childress (ABC) flow [6]

\[
\begin{align*}
\dot{x} &= A \sin z + C \cos y, \\
\dot{y} &= B \sin x + A \cos z, \\
\dot{z} &= C \sin y + B \cos x,
\end{align*}
\]

an exact solution of Euler’s equation. We select the parameter values $A = \sqrt{3}$, $B = \sqrt{2}$, and $C = 1$. This well-studied set of parameter values [27, 57, 77, 32] yields six coherent vortices.

We construct a high resolution graph by selecting a uniform grid of $120 \times 120 \times 120$ points over the spatial domain ranging from 0 to $2\pi$ in $x$, $y$, and $z$ directions. Next, we subsample the phase space uniformly on a coarser grid by selecting $q = 1000$ supernodes out of the $120^3$ nodes of the original graph, and construct the tight similarity matrix $Z \in \mathbb{R}^{q \times n}$, expressing similarity between the $q$ supernodes and the $n$ nodes of the original graph. To construct the tight similarity matrix $Z$, we measure the dynamic distances in the lifted system, where trajectories can flow out of the $2\pi$ cube. Having the similarity matrix $Z$ in hand, we compute the dominant singular values and singular vectors of $\hat{Z} = D_2^{-1/2} Z D_1^{-1/2}$. The left singular eigenvectors are cluster indicators for the reduced graph built upon $q$ supernodes, while the right singular vectors are cluster indicators for the original graph.
Figure 4.16: (a) Seven clusters extracted by K-means clustering ($k = 7$) from the first six eigenvectors of $L$. The first six clusters correspond to six coherent vortices that were identified earlier in [57]. The chaotic sea between coherent vortices is the seventh cluster and appears as the void between them. (b) The seventh cluster that appears as the chaotic sea between coherent vortices. (c)-(d) 3D vortices are reconstructed by putting together the coherent cluster pieces.

As the last step, we retrieve seven clusters from six cluster indicators using the K-means algorithm. The last cluster, as before, shows the incoherent region filling the space between the coherent clusters or vortices. Figure 4.16a shows the six coherent clusters which are separated by the incoherent cluster. The six clusters capture the six known coherent structures of the ABC flow identified earlier in [57].
Figure 4.17: Coherent vortices extracted by Algorithm 4 are compared with the Poincaré map constructed for integration time $T = 3000$. 
Due to the existence of the spatial periodic boundary condition, the coherent vortices are broken into pieces in the initial cubic domain. However, our algorithm can detect all these pieces as connected entities without any extra effort (see fig. 4.16a). This separates our method from most other methods that rely on having the entire vortex fully contained in the flow domain (see [90] for examples). In figs. 4.16c and 4.16d we put together the pieces of six coherent vortices, and show their full cylindrical geometry. The colors used in figs. 4.16c and 4.16d are consistent with those in fig. 4.16a. In fig. 4.17 the clusters are superimposed on the Poincaré map showing close agreement between the results of the two approaches.

4.5 Summary and conclusions

We have developed here an approach to locate coherent structures based on spectral graph theory. To identify coherent structures, we measure the pairwise Euclidean distance between Lagrangian trajectories, and construct an undirected weighted graph describing the spatio-temporal evolution of fluid flows. We then identify coherent vortices as clusters of Lagrangian particles remaining close under the flow using two different algorithms. In the first algorithm, we used Shi & Malik [171] normalized cut to identify coherent vortices whose nodes on graph have large internal (external) (in-)coherence. We demonstrate the effectiveness of the corresponding Algorithm 3 to detect Lagrangian coherent vortices in periodic, quasiperiodic, and unsteady two-dimensional flows. This includes the determination of the a priori unknown number of present vortices in a given domain using the eigengap heuristic.

In Algorithm 4, we apply a recently developed graph sub-sampling technique [33, 129] to handle the memory bottleneck associated with large-scale graphs. We apply Algorithm 4 in our last example, the 3D steady ABC flow, where we succeeded to combine high sampling resolution with computational efficiency.

An advantage of our approach is that it requires a relatively low number of Lagrangian trajectories as input, making it suitable for the analysis of low-resolution trajectory data sets (see also [79, 194, 76] for similar approaches designed for low numbers of Lagrangian trajectories). Moreover, our method is taking advantage of trajectories’ intermediate positions, i.e., information that comes in most cases without additional computational cost, e.g., in time resolved trajectory data sets or numerical integration of velocity data sets/vector fields (see also [79]).

Moreover, we argue that in fluid-like flows coherence-related phenomena can only be conceived in the presence of an incoherent background, which prohibits the partitioning of the fluid domain into purely coherent sets or regions. Here, we introduced the definition of incoherent cluster and partitioned the fluid domain into coherent and incoherent clusters, an idea that appears to be missing in other similar approaches [135, 79, 166].
Finally, we chose spectral clustering as a tool of choice due to its solid mathematical foundation and its performance. However, other clustering algorithms such as density-based clustering approaches [62] that can incorporate the definition of noise or incoherent cluster may be used alternatively. Incorporating other clustering algorithms, and comparing their performance for the purpose of Lagrangian coherent vortex identification remains a viable future research direction. Moreover, further work is needed to connect graph properties with physical or mechanical quantities characterizing the fluid motion, beyond the heuristic and numerical arguments given in Sections 4.1 and 4.4.

**4.A Approximating Ncut**

In this section, we recall how the NCut problem can be solved for the case $k = 2$, which partitions the graph into two disjoint sets. We follow closely the arguments of [171, 185].

Our goal is to solve the optimization problem

$$\min_{A \in V} \text{NCut}(A, \bar{A}). \quad (4.6)$$

First, we rewrite the problem in a more convenient form. Given a subset $A \subset V$ we define the cluster indicator vector $f = (f_1, ..., f_n)^T \in \mathbb{R}^n$ with entries

$$f_i = \begin{cases} \sqrt{\frac{\text{vol}(\bar{A})}{\text{vol}(A)}}, & \text{if } v_i \in A, \\ -\sqrt{\frac{\text{vol}(A)}{\text{vol}(\bar{A})}}, & \text{if } v_i \in \bar{A}. \end{cases} \quad (4.7)$$

Now, Eq. (4.6) can be conveniently rewritten using the graph Laplacian $L$ as

$$\min_{g \in \mathbb{R}^n} g^T D^{-1/2} L D^{-1/2} g \quad \text{subject to } g \perp D^{1/2}, \quad \|g\|^2 = \text{vol}(V).$$

This is a Rayleigh quotient, and minimizing it is of complexity NP-hard, since we have constrained $f$ to take on only discrete values as described in (4.7). We relax the problem by allowing $f$ to take arbitrary real values ($l_2$-relaxation), to obtain:

$$\min_{f \in \mathbb{R}^n} f^T L f \quad \text{subject to } D f \perp 1, \quad f^T D f = \text{vol}(V).$$

After substitution of $g := D^{1/2} f$, the problem converts to

$$\min_{g \in \mathbb{R}^n} g^T D^{-1/2} L D^{-1/2} g \quad \text{subject to } g \perp D^{1/2}, \quad \|g\|^2 = \text{vol}(V),$$
to which the standard Rayleigh-Ritz theorem applies, such that its solution \( g \) is given by the second eigenvector of \( D^{-1/2}LD^{-1/2} \). Re-substituting \( f = D^{-1/2}g \), we see that \( f \) is the second generalized eigenvector of \( Lu = \lambda Du \).

Similarly, we can decompose the graph into \( k \) partitions by using cluster indicator vectors 
\[
    h_j = (h_{1,j}, \ldots, h_{n,j})^T
\]

\[
    h_{i,j} = \begin{cases} 
        \frac{1}{\sqrt{\text{vol}(A_j)}}, & \text{if } v_i \in A_j, \\
        0, & \text{otherwise},
    \end{cases} 
    \quad i = 1, \ldots, n, \ j = 1, \ldots, k. \tag{4.8}
\]

Then we set the matrix \( H \in \mathbb{R}^{n \times k} \) as the matrix containing those \( k \) cluster indicator vectors as columns. Observe that the columns in \( H \) are orthonormal to each other, that is \( H^T H = I \), and \( h_i^T L h_i = \text{cut}(A_i, \bar{A}_i)/\text{vol}(A_i) \). So we can write the problem of minimizing NCut as
\[
    \min_{A_1, \ldots, A_k} \text{Tr}(H^T L H) \quad \text{subject to } H^T D H = I, \ H \text{ as in (4.8)}
\]

Relaxing the discreteness condition and substituting \( T = D^{1/2}H \) we obtain the relaxed problem
\[
    \min_{T \in \mathbb{R}^{n \times k}} \text{Tr}(T^T D^{-1/2} LD^{-1/2} T) \quad \text{subject to } T^T T = I.
\]

Again, this is the standard trace minimization problem, which is solved by the matrix \( T \) composed of the first \( k \) eigenvectors of \( D^{-1/2}LD^{-1/2} \) as columns. Re-substituting \( H = D^{-1/2}T \), we see that the solution \( H \) consists of the first \( k \) generalized eigenvectors of \( Lu = \lambda Du \). This yields the normalized spectral clustering algorithm according to [171].

### 4.B Bipartite spectral graph partitioning

In this section, we briefly recall how spectral clustering is applied to bipartite graphs. This specification is also referred to as spectral co-clustering [55, 198], and is presented here in the sub-sampling terminology introduced in Section 4.2. It applies, however, verbatim to the bipartite transfer-operator graph.

Let \( Z \in \mathbb{R}^{q \times n} \) be a tight similarity matrix between the \( n \) graph nodes and the \( q \) supernodes. To explicitly capture the node-supernode relationship, we consider a bipartite graph \( G_B = (V_B, E_B, W_B) \) whose nodes can be divided into two disjoint sets \( A \) and \( B \) such that internal edges all have zero weights, i.e., \( w^B_{ij} = 0 \) if \( v^B_i, v^B_j \in A \) or \( v^B_i, v^B_j \in B \). The similarity
matrix of the whole bipartite graph $W_B$ then reads as

$$W_B = \begin{pmatrix} 0 & Z^\top \\ Z & 0 \end{pmatrix}$$  \hspace{1cm} (4.9)$$

To partition the bipartite graph, the optimization task can be formalized as a generalized eigenvalue problem with suitable relaxation, see Section 4.A,

$$L_B q = (D_B - W_B) q = \lambda D_B q$$  \hspace{1cm} (4.10)$$

where $D_B$ is the degree matrix of $W_B$.

Substituting (4.9) in (4.10), we get

$$\begin{pmatrix} 0 & Z^\top \\ Z & 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} D_1 - 1/2 & 0 \\ 0 & D_2 - 1/2 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix},$$  \hspace{1cm} (4.11)$$

where $D_1$ is an $n \times n$ diagonal matrix whose entries are column sums of $Z$ and $D_2$ is an $q \times q$ diagonal matrix whose entries are row sums of $Z$. Breaking the block matrix form into parts, Eq. (4.11) can be rewritten as:

$$Z^\top q_2 = (1 - \lambda)D_1 q_1,$$

$$Z q_1 = (1 - \lambda)D_2 q_2.$$  

Let $b = D_1^{1/2} q_1$ and $a = D_2^{1/2} q_2$, and after variable substitution, we have

$$D_1^{-1/2} Z^\top D_2^{-1/2} a = (1 - \lambda) b,$$

$$D_2^{-1/2} Z D_1^{-1/2} b = (1 - \lambda) a.$$  

These equations define the SVD of the normalized matrix $\tilde{Z} = D_2^{-1/2} Z D_1^{-1/2}$. Particularly, $a$ and $b$ are the left and right singular vectors and $1 - \lambda$ is the corresponding singular value [198].
Chapter 5

Level Set Formulation of Two-dimensional Lagrangian Vortex Detection Methods

Chapter Abstract

We propose here the use of variational level set methodology to capture Lagrangian vortex boundaries in 2D unsteady velocity fields. This method reformulates earlier approaches that seek material vortex boundaries as extremum solutions of variational problems. We demonstrate the performance of this technique for two different variational formulations built upon different notions of coherence. The first formulation uses an energy functional that penalizes the deviation of a closed material line from piecewise uniform stretching [102]. The second energy function is derived for a graph-based approach to vortex boundary detection, as described in Chapter 4. Our level-set formulation captures an a priori unknown number of vortices simultaneously at relatively low computational cost. We illustrate the approach by identifying vortices from different coherence principles in several examples.

5.1 Background

5.1.1 Implicit boundary representation

We begin by reviewing the standard level set method, as devised by Osher & Sethian [152]. Consider a closed moving interface as a curve $\Gamma(\tau)$ in $\mathbb{R}^2$, with $\tau$ denoting the time of evolution. Let $\omega(\tau)$ be the open region that $\Gamma(\tau)$ encloses in the domain $\Omega$ (see fig. 5.1). The main idea of the level set methodology is to embed $\Gamma(\tau)$ as the zero-level set of a higher-dimensional function $\phi(\cdot, \tau) : \Omega \to \mathbb{R}$, called the level set function, which is assumed Lipschitz continuous and satisfies
the following conditions:

\[
\begin{align*}
\phi(x, \tau) &> 0 \quad \text{for } x \in \omega(\tau), \\
\phi(x, \tau) &< 0 \quad \text{for } x \in \Omega - (\omega(\tau) \cup \Gamma(\tau)), \\
\phi(x, \tau) &= 0 \quad \text{for } x \in \Gamma(\tau).
\end{align*}
\]

Conversely, if we know \( \phi(x, \tau) \), we may locate the interface by finding the zero level set of \( \Gamma(\tau) = \{ x : \phi(x, \tau) = 0 \} \). Evolving the interface \( \Gamma(\tau) \) in \( \tau \) is equivalent to updating \( \phi(x, \tau) \).

A typical example of a level set function is given by the Signed Distance Function (SDF) measured from a curve. The SDF computed for \( \Gamma(\tau) \) gives the distance of a given point \( x \) from the interface \( \Gamma(\tau) \), with the sign determined by whether \( x \) is inside or outside \( \omega(\tau) \). The SDF has positive values inside \( \Gamma(\tau) \), decreases to zero as \( x \) approaches \( \Gamma(\tau) \), and takes negative values outside of \( \Gamma(\tau) \). Signed distance functions share all the properties of implicit functions, such as supporting Boolean operations (union, intersection, and difference), in addition to the identity \( |\nabla \phi| = 1 \) (cf. [151]).

### 5.1.2 Front evolution and level set theory

Given an interface \( \Gamma(\tau) \), our goal is to produce an equation for evolving \( \phi(x, \tau) \), as the embedding of \( \Gamma(\tau) \), through space and time such that the interface \( \Gamma(\tau) \) advances toward the vortex boundaries. The variational level set approach obtains the equations governing the evolution of \( \Gamma(\tau) \) by minimizing a certain energy functional \( E \) defined on the level set function \( \phi(x, \tau) \). The energy functional \( E \) can depend on the intrinsic geometric properties of the interface (e.g,
curvature) or on extrinsic quantities (e.g., velocity of the fluid flow). The spatio-temporal partial differential equation describing the evolution of the level set function is given by

\[
\frac{\partial \phi}{\partial \tau} = -\frac{\partial E}{\partial \phi}. \tag{5.1}
\]

Equation (5.1) is a gradient flow \[63\] that minimizes the functional \(E\) and simultaneously governs the evolution of the interface \(\Gamma(\tau)\). There are several advantages associated with this perspective:

1. Although \(\phi(x, \tau)\) remains a smooth function, the level surface \(\phi(x, \tau) = 0\) corresponding to the propagating interface may develop sharp corners, break apart, or merge. No elaborate numerical mechanism is required to handle such topological changes.

2. The level set function always remains a function on a fixed grid, which allows for efficient numerical schemes.

3. Intrinsic geometric properties of the interface \(\Gamma(\tau)\) are obtained directly from the level set function \(\phi\). For instance, the outward unit normal vector to \(\Gamma(\tau)\) is given by \(n = \nabla \phi |_{\nabla \phi}|\), and the mean curvature of each level set is \(\kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}\). Other geometric quantities, such as the arclength \(|\Gamma|\) and the enclosed area \(|\omega|\) of \(\omega\), can be expressed respectively as \[36\] \[151\]:

\[
|\Gamma| = \int_{\Omega} \delta(\phi(x, \tau)) |\nabla \phi(x, \tau)| \, dx, \quad |\omega| = \int_{\Omega} H(\phi(x, \tau)) \, dx, \tag{5.2}
\]

where \(H(\phi)\) is the Heaviside function, and \(\delta(\phi)\) is its derivative, the Dirac delta function.

We shall omit the dependence of \(\phi\) on the spatial variable \(x\) and the evolution time \(\tau\) for notational simplicity.

### 5.2 Variational level-set-based vortex detection

In the previous section, we discussed how to represent a curve implicitly and advect it with a gradient flow using the level set method. We have not yet discussed, however, how the energy function can be constructed to ensure that an arbitrary closed curve moves towards vortex boundaries. As we shall see below, such an energy functional should have local minima that mark the desired vortex boundaries. We work out the derivations of two energy functionals for detecting vortex boundaries. Our first derivation relies on the uniform stretching properties of Lagrangian vortex boundaries. Our second functional characterizes vortex boundaries based on the sustained proximity of Lagrangian particles in the spatio-temporal domain these boundaries enclose.
5.2.1 Stretching-based formulation

We start with an unsteady velocity field

\[ \dot{x} = v(x, t), \quad x \in U \subset \mathbb{R}^2, \quad t \in [t_0, t], \]  

(5.3)

which defines a two-dimensional flow over the finite time interval \([t_0, t]\) in the spatial domain \(U\). The flow map \(F_{t_0}^t(x_0) : x_0 \mapsto x_t\) of (5.3) then maps the initial condition \(x_0\) at time \(t_0\) to its evolved position \(x_t\) at time \(t\). The right Cauchy–Green (CG) strain tensor associated with (5.3) is defined by eq. (2.3). We shall suppress the dependence of CG on \(t_0\) and \(t\) for notational simplicity.

We seek a Lagrangian vortex boundary as an exceptional closed material line \(\Gamma\) around which \(O(\epsilon)\)-thick coherent belts show minimal variation in the length-averaged Lagrangian strain over the time interval \([t_0, t]\). This view is motivated by [102], where the authors seek a perfectly coherent boundary as a material line exhibiting no leading order variation in material strain across the \(O(\epsilon)\)-thick coherent belts. Solutions to this variational problem turn out to be closed material lines that are infinitesimally uniformly stretching, i.e., all their subsets stretch by the same amount between the times \(t_0\) and \(t\). Compared to [102], we do not explicitly enforce such uniform stretching, but require the vortex boundary to have as little nonuniformity in its stretching as possible. In section 5.3.2 we will apply both the original coherence principle [102] and its present relaxed version to identify the boundary of the Great Red Spot (GRS) in Jupiter’s atmosphere.

To express our stretching-based energy functional mathematically, we select a parametrization \(r(s)\) with \(s \in [0, \sigma]\) for the closed \(\Gamma\). We let \(l_{t_0}(s)\) denote the length of a tangent vector \(r'(s)\) at initial time \(t_0\), and \(l_t(s)\) denote the length of the corresponding tangent vector at final time \(t\). These two tangent lengths can be calculated as [182]:

\[ l_{t_0} = \sqrt{\langle r'(s), r'(s) \rangle}, \quad l_t = \sqrt{\langle r'(s), C(r(s)) r'(s) \rangle}. \]  

(5.4)

The quadratic variation of tangential strain along \(\Gamma\) is then given by

\[ E(\Gamma, c_0) = \int_{\Gamma} \left( \frac{l_t(s)}{l_{t_0}(s)} - c_0 \right)^2 ds, \]  

(5.5)

where \(c_0\) is an unknown constant to be determined. Expressing the interface \(\Gamma\) implicitly as the
zero level set of a function $\phi$, we obtain

$$E(\phi, c_0) = \int_{\Omega} \left( \frac{\sqrt{\langle \nabla \phi, \tilde{C} \nabla \phi \rangle}}{\sqrt{\langle \nabla \phi, \nabla \phi \rangle}} - c_0 \right)^2 \delta(\phi) \, dx$$

$$= \int_{\Omega} \left( \frac{1}{\sqrt{\langle \nabla \phi, \nabla \phi \rangle}} \sqrt{\langle \nabla \phi, \tilde{C} \nabla \phi \rangle} \delta(\phi) \right)^2 \, dx,$$

where $\tilde{C} = R_{\pi/2}^T C R_{\pi/2}$, with $R_{\pi/2}$ referring to a counter-clockwise rotation by $\pi/2$.

Equation (5.6) is a multivariable functional that can be minimized via the alternative optimization procedure as follows. First, we fix $\phi$ to optimize for $c_0$ and then fix $c_0$ for optimizing over $\phi$. When $\phi$ is fixed, we obtain the optimum

$$c_0 = \frac{1}{\sigma} \int_{\Omega} \frac{1}{|\nabla \phi|} \sqrt{\langle \nabla \phi, \tilde{C} \nabla \phi \rangle} \delta(\phi) \, dx \equiv \frac{1}{\sigma} \int_{\Gamma} \frac{l_t(s)}{l_0(s)} \, ds,$$

which is just the average relative stretching along the curve $\Gamma$. Keeping this $c_0$ fixed and formally optimizing the energy with respect to $\phi$, we obtain the Euler-Lagrange equation

$$\frac{\partial E}{\partial \phi} = \frac{\partial f}{\partial \phi} - \nabla \cdot \frac{\partial f}{\partial \nabla \phi} = 0,$$

$$\frac{\partial E}{\partial \phi} = -2\delta(\phi) \nabla \cdot \left[ \left( c_0 \frac{\sqrt{\langle \nabla \phi, \tilde{C} \nabla \phi \rangle}}{|\nabla \phi|} - \left( \frac{\nabla \phi}{|\nabla \phi|} \cdot \frac{\tilde{C} \nabla \phi}{|\nabla \phi|} \right) \right) \frac{\nabla \phi}{|\nabla \phi|^2} + \left( 1 - c_0 \frac{\sqrt{\langle \nabla \phi, \tilde{C} \nabla \phi \rangle}}{|\nabla \phi|} \right) \frac{\tilde{C} \nabla \phi}{|\nabla \phi|^2} \right],$$

with the Neumann boundary conditions imposed on the domain boundaries.

To find the minimum of $E$ with respect to $\phi$ numerically, we parameterize the descent direction by an artificial time $\tau \geq 0$, and solve the gradient descent eq. (5.1). The total energy (5.6) is then minimized by iterating the contour evolution (5.1) in alternation with the update (5.7) of the average stretching parameter.

### 5.2.2 Graph-based formulation

In this section, we describe an alternative approach to vortex identification that relies on spectral graph theory and a localized level set model. Within this framework, the contour moves based on the localized energies obtained directly from nearby particle trajectories. To
compute these local energies, we form small regions around each point along the evolving curve such that each region is split into a local interior and a local exterior by the curve (see fig. 5.2). We then obtain the level-set evolution equation by optimizing a functional that incorporates these local energies. Below we describe this approach in more detail using related concepts from [93, 121].

We start by defining a second spatial variable $y$ that also labels points in $\Omega$. We then define a mask function $B(x, y)$, that acts as an indicator function for points $x$ and $y$ within a distance $R$ [121]:

$$B(x, y) = \begin{cases} 
1, & \text{if } \|x - y\| < R \\
0, & \text{otherwise.}
\end{cases}$$

The function $B(x, y)$ is, therefore, equal to 1 when the point $y$ is within a ball of radius $R$ centered at $x$, and is equal to 0 otherwise.

The associated localized energy along an evolving curve $\Gamma$ is then given by

$$E(\phi) = \int_{\Omega_x} \delta(\phi(x)) \int_{\Omega_y} B(x, y) \cdot F(\phi(y)) \, dy \, dx,$$

where $F$ is a function designed to detect the presence of vortex boundaries within a $B(x, y)$ neighborhood of a point on the evolving curve $\Gamma$. We then optimize the energy functional (5.9) by taking its first variation with respect to $\phi$ as follows (see [121] for more details)

$$\frac{\partial E}{\partial \phi} = \delta(\phi(x)) \int_{\Omega_y} B(x, y) \cdot \nabla_{\phi(y)} F(\phi(y)) \, dy.$$

Here, we propose $F$ to be the normalized cut or Ncut [171] value obtained from bi-partitioning of a similarity graph $G$ built locally in a $B(x, y)$ neighborhood of each point on the evolving curve $\Gamma$ (see fig. 5.2). To construct the similarity graph $G$, we follow the procedure specified in [93].

In short, we define the similarity graph $G = (V, E, W)$ through the set of its nodes $V = \{v_1, ..., v_n\}$, the set of its edges $E \subseteq V \times V$ between nodes, and a similarity matrix $W \in \mathbb{R}^{n \times n}$ which associates weights $w_{ij}$ to the edge $e_{ij}$ between the nodes $v_i$ and $v_j$. In our context, we interpret the graph nodes $V$ as a set of Lagrangian particles released within $B(x, y)$, and the associated similarity weights $w_{ij}$ as the inverse of the average Euclidean distance between particle trajectories. We compute this average Euclidean distance using the dynamic distance metric [93].

The Ncut graph-clustering algorithm seeks to partition the nodes $V$ into a set $A$ and its complement $\bar{A}$, such that both of the following hold:
Figure 5.2: The optimality of the Normalized Cut value for three different scenarios. (a) localized graph is centered in the vicinity of a vortex boundary (orange). (b) localized graph is centered far away from the vortex boundary. (c) localized graph is centered inside the vortex. The evolving zero level set is illustrated in dark blue.

Within-cluster similarity Nodes in the same cluster are similar to each other, i.e., particles in a coherent structure have mutually short dynamical distances.

Between-cluster dissimilarity Nodes in a cluster are dissimilar from those in the complementary cluster, i.e., particles in a coherent structure are expected to have long dynamical distances from the rest of the particles.

We notice that Ncut directly implements both (dis)similarity conditions [185].

With this definition, we now argue that the value of Ncut is locally minimum when the localized graph is centered in the vicinity of a vortex boundary. To clarify this further, we discuss the optimality of Ncut value for three plausible scenarios: localized graph is centered in the vicinity of a vortex boundary, inside the mixing region and inside a vortex (see fig. 5.2). In the first scenario, the value of Ncut is small since the graph can be split into a cluster $A$ and its complement $	ilde{A}$ such that the edges between $A$ and $\tilde{A}$ have low weights and the edges within $A$ have high weights. In contrast, the Ncut value will be large inside the mixing region since the edges within $A$ will have low weights. We also expect that the value of Ncut will be large inside the vortex as well because all nodes are strongly connected. This means that the evolving level set function $\phi$ becomes trapped at vortex boundaries, given that the energy functional (5.9) is locally minimal.

5.3 Numerical results

We now summarize our algorithms for detecting coherent Lagrangian vortices using stretching- and graph-based formulations in the tables entitled Algorithm 5 and Algorithm 6 below.
Algorithm 5 Stretching-Based Level Set Method

1. Initialization:
   
   (a) Generate a sufficiently large closed curve and initialize the level set function \( \phi \) as a signed distance function \( \phi_0 \) measured from this curve.
   
   (b) Construct the active set \( L_0 \) and populate the neighbor layers \( L_i \) by determining the distance of a neighborhood point form the nearest active point (see Section 5.B).

2. Update the zero level set:
   
   (a) Compute the gradient flow using (5.8).
   
   (b) Evolve the active set with (5.1) to time \( \tau_{k+1} = \tau_k + \Delta \tau \) such that \( \Delta \tau \) satisfies the CFL condition (cf. Section 5.A).

3. Update the sparse band: Update the level set location and the corresponding neighboring layers \( L_i \).

4. Convergence: Check whether the iterations have converged. If yes, stop; otherwise go to step 2.

Algorithm 6 Graph-Based Level Set Method

1. Initialization:
   
   (a) Generate a sufficiently large closed curve and initialize the level set function \( \phi \) as an SDF.
   
   (b) Construct the active set \( L_0 \) and populate the neighbor layers \( L_i \).

2. Update the zero level set:
   
   (a) Construct localized graphs for the active set
   
   (b) Calculate the Ncut for each localized graph \( G \) such that the graph will be partitioned into a local interior and a local exterior by the curve.
   
   (c) Compute the gradient flow using (5.10).
   
   (d) Evolve the active set to time \( \tau_{k+1} = \tau_k + \Delta \tau \) such that \( \Delta \tau \) satisfies the CFL condition, and the total energy decreases.

3. Update the sparse band: Update the level set location and the corresponding neighboring layers \( L_i \).

4. Convergence: Check whether the iterations have converged. If yes, stop; otherwise go to step 2.
The computational cost of our implementation is primarily due to step (2), i.e. the construction of the Cauchy–Green strain tensor or the localized graph for the active set. This accounts for about $75 - 95\%$ of the total execution time, depending on the perimeter length of the zero level set and the resolution of the grid.

We demonstrate the implementation of Algorithms 5 and 6 on three examples to detect coherent Lagrangian vortices. In the first example, we consider a periodically forced pendulum for which we can explicitly confirm our results using an appropriately defined Poincaré map. Our second example, Jupiter’s unsteady wind-velocity field has a higher-level temporal complexity. In this example, we use a time-resolved velocity field reconstructed from an enhanced video footage of Jupiter, capturing Jupiter’s Great Red Spot (GRS) [92]. In the third example, we detect coherent Lagrangian vortices in a quasigeostrophic ocean surface flow derived from satellite-based sea-surface height observations [81].

To implement Algorithms 5 and 6 in the forthcoming examples, we use a variable-order Adams-Bashforth-Moulton solver (ODE113 in MATLAB) to advect fluid particles with the differential equation (5.3). The absolute and relative tolerances of the ODE solver are chosen as $10^{-6}$. In sections 5.3.2 and 5.3.3, we obtain the velocity field at any given point by interpolating the velocity data set using bilinear interpolation.

To evolve the level set function, we use an explicit time-marching scheme governed by the CFL condition (see Section 5.A). We choose the corresponding CFL number $\mu = 0.5$ and the regularization parameter $\varepsilon = 10^{-4}$, unless stated otherwise. Moreover, we initiate the level set evolution with a large enough closed curve that is expected to encircle all coherent Lagrangian vortices. We then evolve the level set function inward so as to capture the coherent vortices individually.

### 5.3.1 Periodically forced pendulum

Consider the periodically forced pendulum

$$
\dot{x}_1 = x_2
$$

$$
\dot{x}_2 = -\sin x_1 + \epsilon \cos t.
$$

For $\epsilon = 0$, the system is integrable, and has chains of alternating elliptic and hyperbolic fixed points, with periodic orbits encircling the elliptic fixed points, and heteroclinic orbits connecting adjacent hyperbolic fixed points. These orbits form invariant sets on the Poincaré map $\mathcal{P} := \mathcal{F}_{0}^{2\pi}$. For $0 < \epsilon \ll 1$, however, the closed invariant sets for $\mathcal{P}$ generally break up. We set the perturbation strength to $\epsilon = 0.4$ and reveal the surviving KAM regions by constructing the Poincaré map $\mathcal{P}$ for 800 iterations. A similar parameter setting was also studied in [93] using
a spectral clustering approach. Here, we would like to capture the surviving KAM regions as coherent structures using the level set method, as described in Algorithm 5.

To identify these coherent regions, we construct the level set function $\phi$ over a uniform grid of $300 \times 300$ points. The spatial domain ranges from $-2.6$ to $-0.3$ in the $x_1$ direction and from $-1.2$ to $1.2$ in the $x_2$ direction. We compute the Cauchy–Green strain tensor $C_{t_0}^t$, with $t_0 = 0$ and $t = 800 \times 2\pi$, over the active set as the level set function evolves. Hence, the Cauchy–Green strain tensor is just computed for those grid nodes that are visited by the zero level set over its evolution.

In fig. 5.3, we show the evolution of the zero level set toward KAM region boundaries. The complete evolution sequence of the zero level set is illustrated in the online supplemental movie M1. This example highlights how the level set method can be used for detecting multiple structures automatically.

Figure 5.4 shows the execution times for two major steps of Algorithm 5 as a function of increasing spatial resolution of the computational domain. The main computational bottleneck, as shown in the figure, is computing the Cauchy–Green strain tensor for the active set. For this reason, we utilized parallel computing techniques with 28 MATLAB workers, with each worker just computing the CG strain tensor for a few active points. At the same time, we used simple serial computation to update the zero level set and its corresponding sparse band.
Chapter 5. A Variational Level Set Approach to Vortex Detection

Section 5.3.2 Jupiter’s wind-velocity field

We use the level set method of Algorithm 5 to uncover unsteady mixing barriers in an unsteady velocity field extracted from a video footage of Jupiter’s atmosphere [92]. The video footage is acquired over NASA’s Cassini mission, covering 24 Jovian days that range from October 31 to November 9 in the year 2000. To reconstruct the velocity field, we apply the Advection Corrected Correlation Image Velocimetry (ACCIV) method [12] that yields a high-density, time-resolved representation of Jupiter’s wind field at the cloud deck (see [92] for more details).

For the level-set computation described in section 5.2, we calculate the Cauchy–Green strain tensor field $C_{tt}$, with $t_0 = 0$ and $t = 24$ days, over a uniform grid of 300 $\times$ 200 points. The spatial domain $U$ ranges from $-61.6^\circ$ W to $-31.6^\circ$ W in longitude and from $-8.9^\circ$ S to $-28.9^\circ$ S in latitude. Figure 5.5 shows the level set-based vortex boundary for the Great Red Spot (GRS), superimposed on the cylindrical map of Jupiter acquired by NASA’s Hubble Space Telescope [202]. The complete evolution sequence of the level set function is illustrated in the online supplemental movie M2.

Beyond executing Algorithm 5 to extract the boundary of the GRS using the level set framework, we also use this example to make a comparison with the geodesic LCS theory [102]. As mentioned in section 5.2.1, the latter theory seeks vortex boundaries as closed material-lines that remain perfectly non-filamenting over a finite time interval of interest. Such vortex boundaries turn out to be closed material curves in the flow that stretch uniformly by a constant factor. Figure 5.6a shows the result from the geodesic approach at the initial time [92], with the level set-based vortex boundary superimposed.

Figure 5.6a shows that both methods label the GRS as a vortex, but the geodesic method yields a tighter boundary compared to the level set approach. This is because the geodesic...
Chapter 5. A Variational Level Set Approach to Vortex Detection

Figure 5.5: Lagrangian vortex boundary of the GRS obtained with the level-set method shown at initial time $t = 0$. The initial zero level set is shown with blue dashed line. The new global map of Jupiter acquired by NASA’s Hubble Space Telescope on January 19, 2015 is used as background. The complete evolution sequence of the zero level set is illustrated in the online supplemental movie M2.

Figure 5.6: Geodesic vortex boundary (green) at time $t = 0$ for the Jupiter data set [92], with the level set-based vortex boundary (black) superimposed. (b) Advected position of the Lagrangian vortex boundaries at final time $t = 24$.

Figure 5.7: Relative stretching of the geodesic boundary in comparison with the relative stretching of the level set-based boundary. The relative stretching of a material line segment is defined as the ratio of its length at the final time $t = t_0 + T$ to its initial length at time $t = t_0$. 
method adopts a more stringent definition of coherence, which imposes the uniform stretching of the boundary. This observation is also consistent with the earlier comparison made between the geodesic LCS method and the more recent Lagrangian-Averaged Vorticity Deviation approach \[103]\.

In fig. 5.6b, we show the advected image of the extracted vortex boundaries at the final time, confirming the sustained coherence for both boundaries over 24 Jovian days periods. For the purposes of this comparison, we have used the numerical implementation of the geodesic eddy detection method described in Hadjighasem & Haller \[92]\. A MATLAB implementation of this algorithm is available under \[https://github.com/LCSETH]\.

In fig. 5.7, we show a comparison of relative stretching of the geodesic vortex boundary and the level set based vortex boundary. Figure 5.7 confirms the expectation that the geodesic boundary only exhibits uniform stretching, while the level set-based boundary can exhibit larger variation in the relative stretching. The small deviation from constant stretching in the computed geodesic boundary is only due to finite sampling of the curve, as well as to the interpolation error in the computation of Cauchy–Green strain tensor field.

While the geodesic LCS method yields a perfectly coherent boundary, the level set approach comes with a lower computational cost for the following reasons. First, the search for a maximal limit cycle in the vector field family induced by the value of relative stretching is absent in the level set approach. Second, the evolution of the level set function is governed by a vector field which does not rely on Cauchy-Green invariants. This in turn eliminates the need for the Cauchy-Green eigendecomposition, which must be carried out with high precision close to the tensor singularities. Third, the geodesic method requires integrating a vector field for which orientational discontinuities need to be resolved locally at each integration step. Such orientational discontinuities are not present in the level set approach.

### 5.3.3 An ocean surface data set

Next, we apply Algorithm 6 to a two-dimensional unsteady velocity data set derived from AVISO satellite-observed sea-surface heights (SSH) under the geostrophic approximation. In this approximation, the sea-surface height $\eta(\varphi, \theta, t)$ serves as a stream function for surface velocities in $(\varphi, \theta)$ longitude-latitude spherical coordinate system. The evolution of fluid particles is given by

\[
\frac{d\varphi}{dt} = \frac{g}{R_E^2 f(\theta) \cos \theta} \frac{\partial \eta}{\partial \varphi},
\]

\[
\frac{d\theta}{dt} = \frac{g}{R_E^2 f(\theta) \cos \theta} \frac{\partial \eta}{\partial \theta},
\]
where \( g \) is the constant of gravity, \( R_E \) is the mean radius of the Earth, and \( f(\theta) \equiv 2\Omega_E \sin \theta \) is the Coriolis parameter, with \( \Omega_E \) denoting the Earth's mean angular velocity.

Here, we illustrate the detection of coherent Lagrangian vortices with Algorithm 6 over a period of 90 days, ranging from \( t_0 = \text{November 11, 2006} \), to \( t = 9 \text{ February, 2007} \). We select the computational domain in the longitudinal range \([-4^\circ, 6^\circ]\) and the latitudinal range \([-34^\circ, -28^\circ]\), which falls inside the region of the Agulhas leakage in the Southern Ocean. The region in question with the same time interval is studied earlier in [103] using Lagrangian-Averaged Vorticity Deviation approach.

We select a uniform grid of \( 250 \times 150 \) points to represent the level set function \( \phi \). To evolve the level set function across the active set at each iteration, we first construct a localized graph, with 64 nodes distributed uniformly in a ball of radius \( R = 1/25^\circ \), for each active point. We then partition each localized graph into a local interior and local exterior, and find the subsequent Ncut value. The optimality of partitioning in return drives the zero level set toward the vortex boundaries (see section 5.2.2). In this computation, we set the regularization term as \( \varepsilon = 10^{-3} \).

Figure 5.8a shows the time \( t_0 \) position of the identified vortices, and fig. 5.8b shows their final position at time \( t \), confirming the coherence of the extracted vortices over the 90-day period. The complete evolution sequence of the zero level set toward vortex boundaries is illustrated in the online movie M3. The complete advection sequence over the time interval \([0, 90]\) days is illustrated in the online supplemental movie M4.
5.4 Conclusion

We have demonstrated the application of the variational level set methodology to coherent material vortex detection in fluid flows. To identify coherent structures, we minimize appropriate energy functionals defining the boundaries of coherent vortices. We carry out the minimization via a gradient-descent method, that drives the zero level set towards the desired boundaries.

We have illustrated the performance of the proposed technique on two different energy functionals, each using a different Lagrangian notion of coherence. Our first variational formulation seeks coherent vortices as closed material lines that are close to uniformly stretching. This notion of coherence derives from earlier work of Haller & Beron-Vera [102]. We show the effectiveness of the corresponding approach by detecting Lagrangian coherent vortices in periodic and unsteady two-dimensional flows.

In the second approach, we adopt the idea of normalized graph cut [171] to identify coherent structures based on the proximity of particles in the spatio-temporal domain. Here, we conceive coherent structures, in a fashion similar to [93, 79], as a set of Lagrangian particles that remain tightly grouped. We apply this second approach in our last example, the ocean surface data set, to identify Agulhas eddies in the Southern Ocean.

A drawback of the level set technique is the effort required for the construction of energy functionals whose local minima mark the vortex boundaries. A reward for this effort is a versatile numerical platform that can capture vortices in an automated fashion.

Future challenges include extending the current level set approach to three-dimensional problems and using parallel implementation for speeding up the related calculations.

5.A Numerical aspects

The numerical implementation of Equations (5.8) and (5.10) is simple, but requires some care to ensure sufficient accuracy and efficiency. In this section, we address these implementation aspects.

Stability and CFL condition To keep numerical stability and obtain accurate approximation results, the time step for solving (5.1) with explicit time-marching scheme must satisfy the Courant-Friedrichs-Lewy (CFL) condition [151], which states the front should not cross more than one grid cell at each time step:

$$\Delta \tau \left( \frac{\max |u|}{\Delta x} \right) = \mu, \quad 0 < \mu < 1.$$ 

Here, $u$ refers to the speed with which the zero level set propagates. A common near-
optimal choice for the **CFL number** is $\mu = 0.9$, and a common conservative choice is $\mu = 0.5$ (cf. [151]).

For stability concerns, *implicit* or *semi-implicit* methods may also improve the efficiency of level set methods. Compared to the time steps of explicit schemes limited by a CFL condition, the implicit or semi-implicit level set methods allow for larger time steps (see, for example, [196]). Consequently, the convergence of implicit or semi-implicit schemes is usually faster compared to the explicit methods.

**Reinitialization** In general, even if we initialize the level set function $\phi$ as a signed distance function, it is not guaranteed to remain a distance function at later times. As a consequence, the level set function $\phi$ develops steep or flat shapes during the evolution, making the results inaccurate. Classic level set methods often use the re-initialization remedy to avoid this problem, that is, periodically initialize the level set function as a signed distance function using either the fast marching method [167] or PDE-based approaches [179]. The re-initialization process, however, is complicated, expensive and has an unwanted side effect of shifting the zero level set away from its original location [125]. Moreover, this process is conducted in an ad-hoc manner because there is no rule as to when and how to reinitialize the level set function to a signed distance function. A better approach is to limit re-initialization [168] or use methods that do not require re-initialization at all (see [125] [199] for examples).

**Finite Difference Scheme** Equation (5.8) is a nonlinear Hamilton-Jacobi equation composed of both parabolic and hyperbolic terms. When implementing Eq. (5.8), one must give special attention to how parabolic terms, such as $|\nabla \phi|$, are calculated, as standard finite difference methods fail for non-linear hyperbolic PDEs. Thus, one needs the special machinery of *upwind finite differencing* or *upwinding*, where spatial derivatives are computed using one-sided differencing based on the direction of propagation. We make use of the state-of-the-art high order ENO [152] [153] and WENO [111] schemes in our implementations, whenever it is appropriate to do so.

**Level Set Regularization** In section 5.1 we assumed that the interface $\Gamma$ stays smooth over its evolution, but in applications, smoothness is often lost. A well-known example is the cosine curve evolving with unit speed $u(x, \tau) = 1$, where the propagating curve develops a sharp corner in finite time [168]. Once the corner develops, the normal direction becomes undefined and the differentiability of the interface is lost. Thus, it is important to ensure that the interface $\Gamma$ stays smooth and non-intersecting all along its evolution. This is commonly achieved by adding a regularization term $\varepsilon K$ to the evolution equation (see...
The curvature term $\varepsilon K$ regularizes the interface by accelerating the movement of those segments of the interface that remain behind the average speed of the interface and slowing down the segments marching faster than the average speed. The parameter $\varepsilon$ determines the strength of regularization. If $\varepsilon$ is large, the regularization term will smooth out interface irregularities such that the interface ultimately will become convex. If $\varepsilon$ is small, the front will maintain sharp curvatures and may have a concave geometry at the end of the evolution.

**Narrow Band** The classic level set approach evolves the level set function $\phi$ by solving an initial value problem for a partial differential equation in the entire computational domain. This is superfluous if only information near the zero level set is of interest. Instead, an efficient modification is to perform the computation in a neighborhood or narrow band of the zero level set, as introduced by Adalsteinsson and Sethian [1]. The idea of the narrow band approach was later extended to the Sparse Field Method (SFM), in which the narrow band is only one pixel wide and the level set function is re-initialized with a distance transform in each iteration [192]. We will discuss the Sparse Field Method further in section 5.B.

More details concerning the numerical schemes for level set methods can be found in [151].

### 5.B Sparse Field Method

In classical level set methods, the value of the level set function $\phi$ is updated in the full computational domain, which is computationally costly. Narrow band methods [1] [183] address this problem by only updating pixels near the evolving curve. To optimize and simplify the implementation of the narrow-band scheme, Whitaker [192] proposed the Sparse Field Method which takes the narrow-band strategy to the extreme. The basic idea of the SFM is to use lists of points that represent the zero level set as well as points adjacent to the zero level set (see fig. 5.9). By using these lists and carefully adding and removing points from the appropriate list, the level set function $\phi$ can efficiently be maintained. The fact that the SFM uses lists to keep track of the points near the zero level set means that the computational speed at each iteration depends only on the length of the curve $|\Gamma|$, and not on the size of the domain.

We call the minimal connected set of grid points that are closest to the level set as the *active set*, denoting it by $L_0$, and the individual elements in this set are the *active points*. We then define its neighborhood layers by $L_{\pm i}$ for $i = \pm 1, \ldots, \pm N$, where $i$ indicates the city block distance of a neighborhood point from the nearest active point (see fig. 5.9b). In this chapter, we use up to the second-order derivatives of $\phi$, so we need only five layers: $L_2, L_1, L_0, L_{-1},$ and $L_{-2}$. In addition to the lists, two arrays are used to save the information of the above lists. The first is
the $\phi$ array which has the same dimensions as the computational domain and should be stored at full floating point precision. The second array is a label map which is used to record the status of each point and takes integer values $\{-3, -2, -1, 0, 1, 2, 3\}$, as shown in fig. 5.9a.

The procedure of SFM can be divided into three main steps: initialization, curve evolution and updating the lists. The initialization process of the interface is fairly simple and starts by defining a level set function whose zero level set is explicitly stored at various grid points. This is done by assigning the corresponding points in $\phi$ to 0, and by adding them to the $L_0$ list. The other lists are then filled with points according to their distance from the nearest active point, and are updated accordingly. Next, points in $\phi$ which are members of the active set $L_0$ are updated by the level set evolution equation. These changes are then reflected in the neighboring layers, and finally, the lists are updated accordingly. How these steps are executed is described in details in [192, 120].
Chapter 6

A Comparison of Lagrangian Methods for Coherent Structure Detection

Chapter Abstract

Transport and mixing in geophysical flows are often influenced by presence of long-lived coherent structures that divide the fluid domain into regions with dynamically distinct properties. The problem of identifying coherent Lagrangian regimes in a given velocity field can be studies at two levels: First, through the development of mathematical approaches that directly define coherent structures in terms of physical properties of the flow that can be expressed objectively, regardless of the reference frame. Second, through the development of observable scalar fields in which coherent structures are expected to become visible. The coherent structures in these scalar fields may then be extracted heuristically by applying various image segmentation and classification techniques. Here, we review eleven approaches, consisting of both mathematical and diagnostic tools, and demonstrate the efficacy of each method via its performance on three examples, each brings different challenges that highlight the strengths and weaknesses of each method. For each surveyed method, we also discuss its computational cost and the level of autonomy in terms of particle advection and user interactions, respectively.

6.1 Diagnostics for Lagrangian coherence

We briefly review here the Lagrangian diagnostic scalar fields used in our comparisons. Based on geometric or physical arguments, these approaches propose a scalar quantity whose distribution is expected to reflect coherent features of the flow. Many Lagrangian diagnostics offer neither a strict definition of the coherent flow structures they seek, nor a precise mathematical connection between the topology of the advected scalar field and structures they are expected to reveal.
Instead, these diagnostics offer simply computable scalar fields for a quick visual assessment of flow regions with different Lagrangian behaviors.

A basic expectation of diagnostic scalar fields is that they should outperform generic passively advected scalar fields in their diagnostic abilities. By definition, Lagrangian coherent structures (LCS) create coherent trajectory patterns [99], and hence the footprint of LCSs invariably appears in any generic tracer distribution advected by trajectories. To this end, in our comparisons performed on given examples, we will also include ad hoc passive scalar fields as baselines for the efficacy of diagnostic and mathematical approaches to material coherence.

Another expectation for Lagrangian diagnostics stems from the fact that LCSs are composed of material trajectories. The assessment of whether or not a trajectory is part of an LCS is, therefore, inherently independent of the observer. Any self-consistent LCS method should, therefore, identify the same set of trajectories as LCSs under all Euclidean observer changes of the form 

\[ x = Q(t)y + b(t), \]

where \( y \in \mathbb{R}^2 \) is the coordinate in the new frame, \( Q(t) \in SO(2) \) represents time-dependent rotation, and \( b(t) \in \mathbb{R}^2 \) represents time-dependent translation.

Frame-invariance is particularly important in truly unsteady flows which have no distinguished frame of reference [133]. Within this class, geophysical fluid flows represent an additional challenge, being defined in a rotating frame. The detection or omission of a feature by a diagnostic should clearly not be an artifact of the co-rotation of the frame with the Earth. For each surveyed diagnostic below, we will discuss its objectivity or lack thereof.

### 6.1.1 Finite-time Lyapunov exponent (FTLE)

Haller [104, 96] proposed that the time \( t_0 \) positions of the strongest repelling LCSs over the time interval \([t_0, t_1]\) should form ridges of the Lyapunov exponent (FTLE) field

\[
\text{FTLE}_{t_0}^{t_1}(x_0) = \frac{1}{|t_1 - t_0|} \log \left| \nabla F_{t_0}^{t_1}(x_0) \right| = \frac{1}{|t_1 - t_0|} \log \sqrt{\lambda_{\text{max}}[C_{t_0}^{t_1}(x_0)]}. \tag{6.1}
\]

Similarly, time \( t_1 \) positions of the strongest attracting LCSs over \([t_0, t_1]\) are expected to be marked by ridges of the backward-time FTLE field \( \text{FTLE}_{t_1}^{t_0} \). Repelling and attracting LCSs are usually referred to as hyperbolic LCSs, as they generalize the notion of hyperbolic invariant manifolds to finite-time dynamics.

The FTLE field (6.1) measures the largest finite-time growth exponent experienced by infinitesimal perturbations to the initial condition \( x_0 \) over the time interval \([t_0, t_1]\). It is therefore a priori unclear if a given FTLE ridge indeed marks a repelling material surface, or just a surface of
There are both conceptual and mathematical issues with such an identification, and the evolving ridges so obtained may well be far from Lagrangian (cf. Haller [99]). The FTLE field is objective by the objectivity of the invariants of the Cauchy–Green strain tensor (Gurtin [89]).

The FTLE diagnostic is not geared towards detecting elliptic (vortex-type) LCSs in finite-time flow data. While the FTLE values are expected to be low near elliptic LCS, a sharp boundary for vortex-type structures does not generally emerge from this diagnostic, as seen in our examples below.

### 6.1.2 Finite-Size Lyapunov Exponent (FSLE)

An alternative assessment of perturbation growth in the flow is provided by the Finite-Size Lyapunov exponent (FSLE). To define this quantity, one first selects an initial separation \( \delta_0 > 0 \) and a separation factor \( r > 1 \) of interest. The separation time \( \tau(x_0; \delta_0, r) \) is then defined as the minimal time in which the distance between a trajectory starting from \( x_0 \) and some neighboring trajectory starting \( \delta_0 \)-close to \( x_0 \) first reaches \( r\delta_0 \). The FSLE associated with the location \( x_0 \) is then defined as (cf. [9, 14, 112]):

\[
\text{FSLE}(x_0; \delta_0, r) = \log \frac{r}{\tau(x_0; \delta_0, r)}
\]  

In contrast to the FTLE field, the FSLE field focuses on separation scales exceeding the threshold \( r \), and hence can be used for selective structure detection. A further conceptual advantage of the FSLE field is that its computation requires no a priori choice of a time scale.

By analogy with FTLE ridges, FSLE ridges have also been proposed as indicators of hyperbolic LCSs (see [112, 59, 23]). This analogy is mathematically justified for sharp enough FSLE ridges of nearly constant height [116]. A general correspondence between FSLE and FTLE ridges, however, does not exist. This is because FSLE\((x_0; \delta_0, r)\) lumps trajectory separation events occurring over different time intervals into the same scalar field, and hence has no general relationship to the single finite-time flow map \( F_{t_0}^t(x_0) \).

The FSLE field has generic jump discontinuities and a related sensitivity to the computational time step (see [116] for details). It is, however, an objective field, given that frame-invariant is purely a function of particle separation, which is a frame-invariant notion.

### 6.1.3 Mesochronic analysis

Mezić et al. [142] proposed the eigenvalue configuration of the deformation gradient \( \nabla F_{t_0}^t(x_0) \) as a diagnostic for qualitatively different regions of material mixing. Specifically, their *mesochronic*
classification considers regions where $\nabla F^t_{t_0}(x_0)$ have real eigenvalues hyperbolic, and regions where $\nabla F^t_{t_0}(x_0)$ has complex eigenvalues elliptic. Since $\nabla F^t_{t_0}(x_0)$ is an orientation preserving diffeomorphism, we necessarily have $\det \nabla F^t_{t_0}(x_0) > 0$, implying that real eigenvalues of $\nabla F^t_{t_0}(x_0)$ are either both negative or both positive. Mezic et al. [142] refers to these two cases as mesohyperbolic with or without a $180^\circ$ rotation, respectively.

Data collected in the aftermath of Deepwater Horizon Spill [142] shows that mixing zones are predominantly mesohyperbolic when the integration time is selected to be about 4 days. Longer studies of ocean data suggest that oceanic flows are predominantly mesoelliptic over longer time scales [22]. This is in line with the expectation that accumulated material rotation along general trajectories unavoidably creates nonzero imaginary parts for the eigenvalues of $\nabla F^t_{t_0}(x_0)$, even if the underlying trajectory starting from $x_0$ is of saddle-type. The mesochronic partition of the flow domain is not objective due to the frame-dependence of the deformation gradient $\nabla F^t_{t_0}(x_0)$ (see e.g., [128]). As a consequence, the elliptic-hyperbolic classification of trajectories obtained from this method will change under changes of the observer.

Mesochronic plots have a tendency to mark vortical regions with alternating, nested sequences of mesoelliptic and mesohyperbolic annuli [142]. Such sequences, however, also arise in non-coherent vortical regions that show significant stretching. Furthermore, due to the frame-dependence of this diagnostic, even the cores of coherent vortices (as confirmed by particle advection) may come out to be hyperbolic if the vortex rotates fast enough (see our examples). There is no published account available on identifying a vortex boundary from mecoschronic analysis, but outermost concentric circular hyperbolic-elliptic boundary curves have recently been suggested as boundaries [141].

6.1.4 Trajectory length

Mancho et al. [139] propose that abrupt variations (i.e., curves of high gradients) in the arclength function

$$M^t_{t_0}(x_0) = \int_{t_0}^{t_1} |v(x(s; t_0, x_0), s)| \, ds$$

of the trajectory $x(s; t_0, x_0)$, considered as a function of the initial condition $x_0$, indicate the time $t_0$ positions of boundaries of qualitatively different dynamics. The $M^t_{t_0}(x_0)$ function is arguably the quickest to compute out of all Lagrangian diagnostics considered here. It also naturally lends itself to applications to float data, given that the arclength of a trajectory can be computed without any reliance on a velocity field or on neighboring trajectories.

As any scalar field computed along trajectories, $M^t_{t_0}(x_0)$ is generally expected to show an imprint of Lagrangian coherent structures, as indeed found by Mancho et al. [139] in examples. The function $M^t_{t_0}(x_0)$ is, however, not objective or even Galilean invariant. For instance, in a
frame co-moving with any selected trajectory \( x(s; t_0, x_0) \), the trajectory itself has zero arclength. The level curve structure of \( M_{t_0}^{t_1}(x_0) \) is not objective either, because the integrand of its gradient field \( \nabla M_{t_0}^{t_1}(x_0) \) consists of factors that are frame-dependent.

### 6.1.5 Trajectory complexity

Rypina et al. [163] propose a partitioning of the flow domain into regions where trajectories exhibit different levels of complexity. They quantify trajectory complexity using the \textit{ergodicity defect}

\[
d(s; x_0, t_0) = \sum_{j=1}^{s^{-2}} \left( \frac{N_j(s)}{N} - s^2 \right)^2,
\]

where \( N_j(s) \) is the number of trajectory points that lie inside the \( j^{th} \) element of a square grid of sidelength \( s \), and \( N \) denotes the total number of sampled points along each trajectory. The most non-ergodic trajectory is a fixed point, for which we have \( d = 1 \). In contrast, for an ergodic trajectory, one should obtain \( \lim_{s \to 0} d(s; x_0, t_0) = 0 \). Rypina et al. [163] define the average ergodicity defect over different scales of \( s \) as

\[
\bar{d}(s; x_0, t_0) = \text{mean}_s(d(s; x_0, t_0)),
\]

where the mean is taken over a broad range of spatial scales \( s \) of interest. Locations of abrupt changes (large gradients) in the topology of the function \( \bar{d}(s; x_0, t_0) \) as a function of \( x_0 \) are expected to mark boundaries between qualitatively different flow regions. The quantity \( \bar{d}(s; x_0, t_0) \) is objective, because presence in, or absence from, a grid cell is invariant under rotations and translations, as long as the same rotations and translations are applied to both the trajectories and the grid cells. The approach is simple to implement and has proven itself effective on low-resolution data (Rypina et al. [163]).

### 6.1.6 Shape coherence

Ma and Bollt [135] seek coherent set boundaries as closed material lines at time \( t_0 \) that are nearly congruent with their advected images at time \( t_1 \). Such near-congruence is ensured by classic results if the curvature distributions along the original and advected curve are close enough.

Motivated by examples of steady linear flows, Ma and Bollt [135] propose finding shape-coherent curves as minimizers of the angle between the dominant eigenvectors of the forward-time and the backward-time Cauchy–Green strain tensors. Stated in our present context, the position of the boundary of a shape-coherent set at time \( \hat{t} = (t_0 + t_1)/2 \) is a closed curve along...
which the splitting angle function

$$\theta(\hat{x}_0) = \arcsin \left( |\xi^{fw}_{2}(\hat{x}_0) \times \xi^{bw}_{2}(\hat{x}_0)| \right), \quad \hat{x}_0 = F^\hat{t}_{t_0}(x_0),$$

vanishes. Here we used the definitions

$$C^{fw}_{t}(\hat{x}_0)\xi^{fw}_{2}(\hat{x}_0) = \lambda^{fw}_{2}(\hat{x}_0)\xi^{fw}_{2}(\hat{x}_0), \quad C^{bw}_{t}(\hat{x}_0)\xi^{bw}_{2}(\hat{x}_0) = \lambda^{bw}_{2}(\hat{x}_0)\xi^{bw}_{2}(\hat{x}_0), \quad |\xi^{fw}_{2}| = |\xi^{bw}_{2}| = 1.$$

Ma and Bollt \[135, 136\] argue that level curves of eq. (6.5) with \(|\theta(\hat{x}_0)| \ll 1\) should show significant shape coherence over a finite time interval. They support this expectation with examples of steady, linear velocity fields.

For finite-time flows with general time dependence, the smallness of \(|\theta(\hat{x}_0)| \ll 1\) along closed structure boundaries remains a heuristic assertion that we will test here on aperiodic examples. For a direct comparison with other methods, we will identify the set \(|\theta(\hat{x}_0)| \ll 1\) for initial conditions \(\hat{x}_0\) seeded at time \(\hat{t}\), then advect these initial conditions back under the flow map \(F^\hat{t}_{t_0}\) to time \(t_0\). The splitting angle diagnostic (6.5) is objective, given that it only depends on the angle between appropriate Cauchy–Green eigenvectors.

### 6.2 Mathematical approaches to Lagrangian coherence

Next, we briefly summarize approaches that locate coherent structures by providing precise solutions to mathematically formulated coherence principles. These approaches, however, are only precise relative to their starting coherence principle. One still needs to test whether those coherence principles capture observed coherent trajectory patterns consistently and effectively in various finite-time data sets. Indeed, a heuristic but well-motivated diagnostic tool may consistently outperform a rigorous mathematical approach that is based on an ineffective coherence principle.

As in the case of diagnostics, we consider frame-indifference (or objectivity) to be a fundamental requirement for the self-consistency mathematical approaches to Lagrangian coherence. All mathematical approaches considered below satisfy this requirement.

### 6.2.1 Geodesic theory of LCS

The geodesic theory of LCSs is a collection of global variational principles for material surfaces that form the centerpieces of coherent, time-evolving tracer patterns [99]. Out of these material surfaces, hyperbolic LCSs act as generalized stable and unstable manifolds, creating coherence by generality of sustained repulsion or attraction. Parabolic LCSs minimize Lagrangian shear.
and hence serve as generalized jet cores. Finally, elliptic LCSs extend the notion of Kolmogorov–Arnold–Moser (KAM) tori and serve as generalized coherent vortices in finite-time unsteady flows. Geodesic LCS theory is objective, as it builds on (material) notions of strain and shear that are expressible through the invariants of the Cauchy–Green strain tensor.

Below we summarize the main results for two-dimensional flows from Farazmand et al. [65] for hyperbolic and parabolic LCSs, and from Haller and Beron–Vera [102] for elliptic LCSs. A general review with further mathematical LCS results, as well as extensions to three-dimensional flows, can be found in [99].

**Stationary curves of the average shear: Hyperbolic and parabolic LCSs**

A shearless LCS is a material curve whose average Lagrangian shear shows no leading-order variation when compared to nearby $C^1$-close material lines. Specifically, the time $t_0$ position of a shearless LCS is a stationary curve for the material-line-averaged tangential shear functional. Farazmand et al. [65] show that such LCSs coincide with null-geodesics of the metric tensor

$$D^{t_1}_{t_0}(x_0) = \frac{1}{2} \left[ \Omega C^{t_1}_{t_0}(x_0) - \Omega C^{t_1}_{t_0}(x_0) \right], \quad (6.6)$$

with the matrix $\Omega$ defined (2.3). The tensor $D^{t_1}_{t_0}(x_0)$ is Lorentzian (i.e., indefinite) wherever $\lambda_1(x_0) \neq \lambda_2(x_0)$. All null-geodesics of $D^{t_1}_{t_0}(x_0)$ are found to be trajectories of one of the two line fields

$$x'_0 = \xi_j(x_0), \quad j = 1, 2. \quad (6.7)$$

We refer to trajectories of (6.7) with $j = 1$ as *shrink lines*, as they strictly shrink in arclength under the flow map $F^{t_1}_{t_0}$. Similarly, we call trajectories of (6.7) with $j = 2$ *stretch lines*, as they strictly stretch under $F^{t_1}_{t_0}$. For lack of a well-defined orientation for eigenvectors, eq. (6.7) is only a line field, not an ordinary differential equation. Nevertheless, the trajectories of (6.7) (i.e., curves tangent to the eigenvector field $\xi_j(r)$) are well-defined at all points where $\lambda_1(x_0) \neq \lambda_2(x_0)$.

**Repelling LCSs** are defined as special shrink lines that start from local maxima of $\lambda_2(x_0)$; **attracting LCSs**, by contrast, are special stretch lines that start from local minima of $\lambda_1(x_0)$. As a consequence of their definitions, repelling and attracting LCSs (or *hyperbolic LCSs*, for short) have a role similar to that of stable and unstable manifolds of strong saddle points in a classical dynamical system. Between any two of their points, hyperbolic LCSs are solutions of the stationary shear variational problem under fixed endpoint boundary conditions.

**Parabolic LCSs**, in contrast, are composed of structurally stable chains of alternating shrink–stretch line segments that connect tensorline singularities (i.e., points where $\lambda_1(x_0) = \lambda_2(x_0)$). Out of all such possible chains, one builds parabolic LCSs (generalized jet cores) by identifying
tensor lines that are closest to being neutrally stable (cf. Farazmand et al. [65] for further details). Parabolic LCSs are more robust under perturbations than hyperbolic LCSs, because they are solutions of the original stationary shear variational principle under variable endpoint boundary conditions.

**Stationary curves of the average strain: Elliptic LCSs**

An elliptic LCS is a closed material line across which the material-line-averaged Lagrangian stretching shows no leading-order variation when compared to closed, $C^1$-close material lines. Specifically, the time $t_0$ position of an elliptic LCS is a stationary curve for the material-line-averaged tangential strain functional. As shown by Haller and Beron–Vera [102], such stationary curves coincide with closed null-geodesics of the one-parameter family of Lorentzian metric tensors

$$ E_\lambda(x_0) = \frac{1}{2} \left[ C^t_{t_0}(x_0) - \lambda I \right], $$

where $\lambda > 0$ is a parameter. These closed null-geodesics turn out to be closed trajectories (limit cycles) of the line-field families

$$ x'_0 = \eta^+_{\lambda}(x_0) = \sqrt{\frac{\lambda_2(x_0) - \lambda^2}{\lambda_2(x_0) - \lambda_1(x_0)}} \xi_1(x_0) \pm \sqrt{\frac{\lambda^2 - \lambda_1(x_0)}{\lambda_2(x_0) - \lambda_1(x_0)}} \xi_2(x_0). \quad (6.8) $$

A simple calculation shows that all limit cycles of (6.8) are infinitesimally uniformly stretching. Specifically, any subset of such a limit cycle is stretched exactly by a factor of $\lambda$ over the time interval $[t_0, t]$ under the flow map $F^t_{t_0}$. As a result, elliptic LCSs exhibit no filamentation when advected under the flow map $F^t_{t_0}$.

Elliptic LCSs occur in nested families due to their structural stability with respect to changes in $\lambda$. The outermost member of such a nested limit cycle family serves as a *Lagrangian vortex boundary*.

For computing geodesic vortices in the forthcoming examples, we use the automated algorithm developed in Haller and Beron-Vera [102] and Karrasch et al. [117]. A MATLAB implementation of this method is provided in [https://github.com/LCSETH](https://github.com/LCSETH).

### 6.2.2 Rotationally coherent Lagrangian vortices

**Rotational coherence form Polar Rotation Angle (PRA)**

Farazmand & Haller [68] introduce the notion of polar LCSs as tubular material surfaces whose elements exhibit similar material rotation in average over a finite time interval $[t_0, t_1]$. To quantify the average rotation for each material element, Farazmand & Haller [68] use the classic...
polar decomposition to factor the flow gradient $\nabla F_{t_1}^{t_0}$ into a rigid-body rotation and a stretching component. They further characterize the rotational component of the flow gradient with a Polar Rotation Angle (PRA), whose value $\theta_{t_1}^{t_0}$ can be computed explicitly from the flow map gradient and invariants of the Cauchy–Green strain tensor. In two-dimensional flows, the PRA can be computed as follows

$$\cos \theta_{t_0}^{t_1} = \frac{\langle \xi_i, \nabla F_{t_0}^{t_1} \xi_i \rangle}{\sqrt{\lambda_i}}, \quad \text{for } i \in \{1, 2\} \tag{6.9}$$

$$\sin \theta_{t_0}^{t_1} = (-1)^j \frac{\langle \xi_i, \nabla F_{t_0}^{t_1} \xi_j \rangle}{\sqrt{\lambda_j}}, \quad \text{for } (i, j) \in \{(1, 2), (2, 1)\}$$

which give

$$\theta_{t_0}^{t_1} = \left[1 - \text{sign} \left(\sin \theta_{t_0}^{t_1}\right) \right] \pi + \text{sign} \left(\sin \theta_{t_0}^{t_1}\right) \cos^{-1} \left(\cos \theta_{t_0}^{t_1}\right), \tag{6.10}$$

Polar LCSs can then be obtained as material lines (or material surfaces) across which fluid elements exhibit similar average rotation by sharing the same PRA value. The initial positions of these material lines or surfaces are marked with tublar level sets of the PRA scalar field $\theta_{t_0}^{t_1}$. The polar LCSs obtained in this fashion are objective in two-dimensional flows. In three dimensions, however, the level surfaces of the PRA are not objective and hence can change under translation and/or rotation of the reference frame \[68\]. An additional limitation of this approach is its dynamical inconsistency: polar rotations do not commute over adjacent sub-intervals, that is the total deformation computed for the full time interval does not match with the sum of rotations computed over smaller sub-intervals \[103\].

**Rotational coherence form Lagrangian-Averaged Vorticity Deviation (LAVD)**

In a recent work, Haller \[100\] proposes the dynamic polar decomposition (DPD) as an alternative to the classic polar decomposition to resolve both the non-objectivity and the dynamic inconsistency issues mentioned above for the polar rotation. Specifically, the DPD factorize the deformation gradient $\nabla F_{t_0}^{t_1}$ uniquely into the product of two deformation gradients: The first one constitutes a purely rotational flow with zero rate of strain, while the second one represents a purely straining flow with zero vorticity. The dynamic polar decomposition of the deformation gradient is of the form

$$\nabla F_{t_0}^{t_1} = O_{t_0}^{t_1} M_{t_0}^{t_1} = N_{t_0}^{t_1} O_{t_0}^{t_1}, \tag{6.11}$$

where $O_{t_0}^{t_1}$ is the **dynamic rotation tensor** and $M_{t_0}^{t_1}$ and $N_{t_0}^{t_1}$ are the **left dynamic stretch tensor** and **right dynamic stretch tensor**, respectively. Compared to the classic polar decomposition where the rotational and stretching components are obtained from matrix manipulations, the dynamic rotation and stretch tensors should be obtained from solving linear differential equations. Specif-
ically, the dynamic rotation tensor $O_{t_0}^t$ is the deformation gradient of a purely rotational flow given as

$$O_{t_0}^t = \nabla_{a_0} a(t), \quad \dot{a} = W(x(t; x_0), t) a,$$

(6.12)

where the spin tensor $W(x, t)$ defined as $W(x, t) = \frac{1}{2} (\nabla v(x, t) - (\nabla v(x, t))^T)$. The dynamic rotation tensor $O_{t_0}^t$ can further be factorized into two deformation gradients:

$$O_{t_0}^t = \Phi_{t_0}^t \Theta_{t_0}^t,$$

(6.13)

where the mean rotation tensor $\Theta_{t_0}^t$ measures rigid-body-like rotations, and the relative rotation tensor $\Phi_{t_0}^t$ subsequently can be computed as the deformation gradient of the following relative rotation flow

$$\Phi_{t_0}^t = \nabla_{\alpha_0} \alpha(t), \quad \dot{\alpha} = [W(x(t; x_0), t) - \bar{W}(t)] \alpha,$$

(6.14)

where $\bar{W}(t)$ is the spatial average of the spin tensor. On the other hand, the mean rotation tensor $\Theta_{t_0}^t$ can be obtained as the deformation gradient of the following mean-rotation flow

$$\Theta_{t_0}^t = \nabla_{\beta_0} \beta(t), \quad \dot{\beta} = \Phi_{t_0}^t \bar{W}(t) \Phi_{t_0}^t \beta.$$

(6.15)

We notice that the mean rotation tensor $\Theta_{t_0}^t$ is dynamically consistent, implying that the intrinsic angle $\psi_{t_0}^t(x_0)$ swept by $\Phi_{t_0}^t$ over the full time interval $[t_0, t_1]$ can be obtained alternatively from summation of intrinsic angles computed from smaller and adjacent sub-intervals. The intrinsic rotation angle $\psi_{t_0}^t(x_0)$ can be sought as the PRA counterpart, with the difference that $\psi_{t_0}^t(x_0)$ is dynamically consistent and is objective for both two- and three-dimensional flows (see [100] for more details).

Using the results obtained in [100], Haller et al. [103] use the Lagrangian-Averaged Vorticity Deviation (LAVD), i.e., twice the value of the intrinsic rotation angle $\psi_{t_0}^t(x_0)$, to identify the polar LCSs. Specifically, the LAVD can be computed as the path integral of the vorticity deviation norm, defined as

$$\text{LAVD}_{t_0}^{t_1}(x_0) = \int_{t_0}^{t_1} |\omega(x(s; x_0), s) - \bar{\omega}(s)| \, ds,$$

(6.16)

where $\bar{\omega}$ is the spatial mean of the vorticity $\omega$. The LAVD field (6.16) can be viewed as the dual of PRA distribution (6.10), and hence the initial positions of polar LCSs can be found similarly as tubular level surfaces of the corresponding field. By a tubular level surface, we mean here a toroidal surface whose size and convexity measure satisfy the thresholds $l_{\min}$ and $d_{\min}$ respectively (see [103] for more details).

The LAVD method inherently does not rule out the possibility of transverse filamentation in
the so obtained vortex boundaries, but it guarantees that any developing filament would rotate at the same rate with the vortex body, without a global breakdown [103]. Another interesting feature of the LAVD method is its predictive capability for the inertial finite-size (inertial) particles. Specifically, centers of LAVD vortices (defined by local maxima of the LAVD field) can be proven to be the observed centers of attraction or repulsion for inertial particles in the limit of vanishing Rossby-number.

In this chapter, we consider the LAVD method as the successor of the PRA approach, and hence will only consider the former in our comparisons. To compute the LAVD vortices, we use a MATLAB implementation of the LAVD method provided in [https://github.com/LCSETH](https://github.com/LCSETH).

### 6.2.3 Coherent pairs of sets

The second mathematical approach considered here is inherently probabilistic, targeting the interiors of coherent regions, as opposed to material surfaces bounding such regions. Probabilistic approaches provide a global view of density evolution in the phase space, identifying maximally coherent or minimally non-dispersive regions over a finite time interval \([t_0, t_1]\). The regions so identified are known as almost-invariant sets for autonomous systems [75] or coherent sets for non-autonomous systems [73, 78].

**Probabilistic transfer operator method**

Given a set \(A_0\) at initial time \(t_0\), the set \(A_{t_1} = F_{t_0}^{t_1}(A_0)\) represents the advected image of \(A_0\) at time \(t_1\). The sets \(A_0\) (source) and \(A_t\) (target) are a coherent pair if tracer placed in the source fills the target with minimal leakage in the presence of additional diffusion. More precisely, the sets \(A_0\) and \(A_t\) are a coherent pair if

\[
\mu(A_0 \cap F_t^{t_0}(A_t))/\mu(A_0) \approx 1 \quad (6.17)
\]

where \(\mu\) is reference probability measure that describes the mass distribution of the quantity we wish to study over the interval \([t_0, t]\) (See [80, 74] for more information). The probabilistic transfer operator method seeks only coherent pairs that remain coherent under diffusive perturbations of the flow. This requirement excludes the choice of any arbitrary set \(A_0\) and its advected image \(A_t = F_{t_0}^{t_1}(A_0)\) as a coherent pair.

This method uses transfer operators to approximate structures which mix slowly with their neighborhood within the state space of a dynamical system. Originally this approach was applied to approximate so-called almost-invariant sets in autonomous dynamical systems [49, 50, 73]. In the past few years this approach has been extended and used for the approximation of transport phenomena in non-autonomous dynamical systems [80, 74].
Hierarchical coherent pairs

The transfer operator method described in [80] focuses on identifying two sets, $A_0$ and its complement $\tilde{A}_0$, that partition a given region of interest to two coherent sets. Ma and Bollt [134] propose an extension of this idea that enables the identification of multiple coherent pairs in a given domain. The extension is based on an iterative and hierarchical refinement of coherent pairs using a reference measure of probability $\mu$. Specifically, Ma and Bollt [134] refine the coherent pairs $A_0$ and $\tilde{A}_0$ identified earlier over several steps by applying transfer operator method restricted to these sets. This iterative refinement of coherent pairs can be stopped once $\mu$ shows no appreciable improvement compared to the earlier iterations. We refer to this method as hierarchical transfer operator method throughout the chapter.

6.2.4 Spectral clustering of trajectories

Recently, Hadjighasem et al. [93] propose a method that identifies coherent structures in a general flow by grouping Lagrangian particles into coherent and incoherent clusters. Specifically, they define a coherent structure as a distinguished set of Lagrangian particles which have mutually short dynamical distances relative to the distances to particles located outside of the coherent structure. Here, the dynamical distance between two Lagrangian particles refers to the distance between their corresponding trajectories in space-time over a finite time interval $[t_0, t_f]$ of interest, as it will be defined mathematically shortly. This notion of coherence which is adopted from spatio-temporal clustering algorithms [119] to coherence in fluid flows, emphasizes that coherence is a relative concept that can only be conceived in presence of incoherence [93].

To identify coherent structures, Hadjighasem et al. start with $n$ trajectories whose positions are available at $m$ discrete times $t_0 < t_1 < \ldots < t_k < \ldots < t_{m-1} = t_f$ in a $d$-dimensional spatial domain. They store this information in an $n \times m \times d$-dimensional numerical array, with elements $x^i_k := x^i(t_k) \in \mathbb{R}^d$. From this trajectory data, they define the dynamical distance $r_{ij}$ between Lagrangian particles $x^i$ and $x^j$ using (4.2.1). We notice that the dynamic distance $r_{ij}$ is an objective metric in any $d$-dimensional space [93].

Next, they construct a similarity graph $G = (V, E, W)$, which is specified by the set of its nodes $V = v_1, \ldots, v_N$, the set of edges $E \subseteq V \times V$ between nodes, and a symmetric similarity matrix $W \in \mathbb{R}^{n \times n}$ which assigns weights $w_{ij}$ to the edge $e_{ij}$. The similarity matrix's entries $w_{ij} \geq 0$ are called weights, and characterize the probability of nodes $v_i$ and $v_j$ to be in the same cluster. In the context of coherent structure detection, the graph nodes $V$ are Lagrangian
particles themselves, and the associated similarity weights are defined as

\[ w_{ij} = \frac{1}{r_{ij}} \quad \text{for } i \neq j. \] (6.18)

Here, we notice that transforming distances \( r_{ij} \) to similarities \( w_{ij} \) could be done via any other monotonically decreasing function, since eq. (4.2.1) defines a metric which preserves the relative differences between the graph nodes [186]. This is according to the general intuition that the arrangement of graph nodes from most dissimilar to least dissimilar should remain the same through the similarity conversion (6.18).

Having the similarity weights in hand, the **degree** of a node \( v_i \in V \) is defined as

\[ \text{deg}(v_i) := \sum_{j=1}^{n} w_{ij}. \]

The subsequent **degree matrix** \( D \) is then defined as a diagonal matrix with the degrees \( \text{deg}(v_i) \) on the diagonal. Given a subset of nodes \( A \in V \), the size of \( A \) is measured by

\[ \text{vol}(A) := \sum_{i \in A} \text{deg}(v_i). \]

where \( \text{vol}(A) \) summing over the weights of all edges attached to nodes in \( A \).

With the notation developed so far, the problem of coherent structure detection can be posed in terms of a **normalized graph cut** problem: Given a similarity graph \( G = (V, E, W) \), partition the graph nodes \( V \) into \( k \) sets \( A_1, A_2, \ldots, A_k \) such that the following conditions hold

**Within-cluster similarity:** Nodes in the same cluster are similar to each other, that is particles in a coherent structure have mutually short dynamical distances.

**Between-cluster dissimilarity:** Nodes in a cluster are dissimilar from those located in the complementary cluster, that is particles in a coherent structure have long dynamical distances to the rest of particles, particularity those located in the mixing region (i.e., noise cluster) that fills the space between the coherent structures.

The normalized cut that implement the above (dis)similarity conditions can be formulated mathematically as

\[ \text{NCut}(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A_i})}{\text{vol}(A_i)}, \quad \text{cut}(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A_i}), \] (6.19)

where \( \overline{A} \) denotes the complement of set \( A \) in \( V \). The minimization of the normalized cut exactly
is an \textit{NP-complete problem}. However, the solution of \textsc{ncut} problem can be approximated by solutions of a generalized eigenproblem associated with the graph Laplacian \( L = D - W \). \[ Lu = \lambda Du. \] (6.20)

In particular, the first \( k \) eigenvectors \( u_1, \ldots, u_k \) whose corresponding eigenvalues are close to zero minimize approximately the \textsc{ncut} objective (6.19). The value of \( k \), in this case, is equal to the number of eigenvalues preceding the largest gap in the eigenvalue sequence [25]. The first \( k \) generalized eigenvectors \( u \) then offer an alternative representation of the weighted graph data such that each leading eigenvector highlights a single coherent coherent structure in the computational domain. Finally, these \( k \) coherent structures besides the complementary incoherent region can be extracted from the eigenvectors \( u_1, \ldots, u_k \) using a simple K-means algorithm [132] or more sophisticated approaches such as PNCZ [197].

\subsection*{6.3 Method comparisons on three examples}

We now compare the performance of diagnostics and mathematical methods reviewed in Sections (6.1)-(6.2) on three specific examples. Our first example, the Bickley jet is an analytically defined velocity field with quasiperiodic time dependence [19]. With its infinite time interval of definition and recurrent time dependence, this example falls in the realm of a classical dynamical systems problem with uniquely defined, infinite-time invariant manifolds. The parameter setting we choose, however, is not near-integrable, and hence the survival of the stable and unstable manifolds and KAM tori of the unperturbed steady limit is a priori unknown.

Our second example is a finite velocity sample obtained from a direct numerical simulation of two-dimensional turbulence [67]. This flow captures most major aspects of a real-life coherence identification problem: the velocity field is a data set; several coherent regions exist, move around and even merge; and the time dependence of the vector field is aperiodic.

Our third example is a velocity field reconstructed from an enhanced video footage of Jupiter, capturing Jupiter’s Great Red Spot (GRS) [92]. This last example has only a single vortical structure, but the data set is short relative to rotation period of the GRS. This shortness relative to characteristic time scales in the data set is an additional challenge relative to our second example.

Table 6.1 compares the computational effort required by each method in terms of the number of particles advected. We select the constants \( n_x, n_y \) and \( N_s \) in a way that the total number of trajectories used in each method is the same for each example. Beyond comparing the results in a single composite plot for all methods in all three examples, we also illustrate different aspects of select approaches on each example.
Table 6.2 compares the degree of autonomy for each method in terms of the number of inputs it requires from the user. Here, we only list major user inputs, and ignore minor inputs such as the integration time, grid resolution and ODE solver tolerance conditions which are required invariably by all the methods. Moreover, we specify some inputs as optional since they are not strictly required for the implementation. Importantly, the number of user inputs required by each method should be viewed according to the functionality the method provides. For instance, the majority of diagnostic tools do not offer any procedure for extracting coherent structures, while other methods such as geodesic, LAVD and spectral clustering give the capability of automated coherent structure detection. Hence, the latter methods may naturally require more user inputs as opposed to simple diagnostic tools.

<table>
<thead>
<tr>
<th>Method</th>
<th># particles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trajectory length, Trajectory complexity, Spectral clustering, LAVD</td>
<td>( n_x \times n_y )</td>
</tr>
<tr>
<td>FTLE, Mesochronic, Geodesic, Shape coherence</td>
<td>( 4 \times n_x \times n_y )</td>
</tr>
<tr>
<td>FSLE</td>
<td>((4 + 1) \times n_x \times n_y)</td>
</tr>
<tr>
<td>Coherent pair</td>
<td>(n_x \times n_y \times N_s)</td>
</tr>
</tbody>
</table>

Table 6.1: Comparison of the minimum number of particles required by each method to construct a Lagrangian field with the resolution \( n_x \times n_y \). The number \( N_s \) is the number of sample points placed in each grid box for the transfer operator method.

6.3.1 Bickley jet

An idealized model for an eastward zonal jet in geophysical fluid dynamics is the Bickley jet \cite{48, 19}, comprising a steady background flow and a time-dependent perturbation. The time-dependent Hamiltonian (stream function) for this model is given by

\[
\psi(x, y, t) = \psi_0(y) + \psi_1(x, y, t),
\]

where

\[
\psi_0(y) = -UL \tanh \left( \frac{y}{L} \right),
\]

is the steady background flow and

\[
\psi_1(x, y, t) = UL \text{sech}^2 \left( \frac{y}{L} \right) \Re \left[ \sum_{n=1}^{3} f_n(t) \exp(i k_n x) \right],
\]

(6.23)
### Table 6.2: Comparison of the minimum number of user inputs required by each method to construct a Lagrangian field with the resolution $m \times n$ over the time interval $[t_0, t_1]$. Here, we ignore trivial user inputs such as the ODE solver tolerance conditions which is required by all methods for advecting particles. Moreover, some inputs are specified as optional since they are not strictly required for implementing a method.

<table>
<thead>
<tr>
<th>Method</th>
<th># inputs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTLE</td>
<td>0-1</td>
<td>• (optional) auxiliary grid space to increase the accuracy of finite differencing [66]</td>
</tr>
</tbody>
</table>
| FSLE              | 2        | • initial separation distance $\delta_0$  
• separation factor $r$  |
| Trajectory length | 0-1      | • (optional) number $N_t$ of sampled points along each trajectory                                                                          |
| Mesochronic       | 0-1      | • (optional) auxiliary grid space                                                                                                            |
| Shape coherent    | 0-1      | • (optional) auxiliary grid space                                                                                                            |
| Trajectory complexity | 2    | • number $N_t$ of sampled points along each trajectory  
• vector specifying a range of spatial scales $s$                                           |
| Coherent pairs    | 2-3      | • number of sample points $N_s$ for initial boxes $B_i$  
• threshold on a relative improvement of reference measure of probability $\mu$  
• (optional) size or number of boxes $C_j$ at final time. |
| Geodesic          | 6-7      | • (optional) auxiliary grid space  
• minimum distance threshold between admissible singularities (cf. [117])  
• radius of circular neighborhood around each singularity to determine its type (cf. [117])  
• minimum distance threshold between a wedge pair (cf. [117])  
• length for the Poincaré section  
• number of initial conditions on each Poincaré section for which $\eta^{\pm}(x_0)$ will be computed  
• range of stretching parameters $\lambda$ needs to be searched for identifying closed orbits |
| Spectral clustering | 1-2   | • (optional) number $N_t$ of sampled points along each trajectory  
• graph sparsification radius $\epsilon$                                                                 |
| LAVD              | 2-3      | • (optional) auxiliary grid space for computing vorticity along trajectories, assuming the direct measure of vorticity is not available  
• arclength threshold $l_{min}$ for discarding small-sized vortex boundaries  
• convexity deficiency threshold $d_{min}$ for relaxing the strict convexity requirement |
is the perturbation. The constants $U$ and $L$ are characteristic velocity and characteristic length scale, with values adopted from [19] as

$$U = 62.66 \text{ ms}^{-1}, \quad L = 1770 \text{ km}, \quad k_n = 2n/r_0, \quad (6.24)$$

where $r_0 = 6371 \text{ km}$ is the mean radius of the earth. For $f_n(t) = \epsilon_n \exp(-ik_n c_n t)$, the time-dependent part of the Hamiltonian consists of three Rossby waves with wave-numbers $k_n$ traveling at speeds $c_n$. The amplitude of each Rossby wave is determined by the parameters $\epsilon_n$. In line with [19], we take $f_n(t) = \epsilon_n \exp(-ik_n c_n t)$, with constant amplitudes $\epsilon_1 = 0.075$, $\epsilon_2 = 0.4$ and $\epsilon_3 = 0.3$. The time interval of interest is $t \in [0, 11] \text{ day}$.

We generate 0.5 million trajectories from a grid of initial conditions in the domain $[0, 20] \times [-3, 3]$. For FTLE, mesochronic analysis, shape coherence and geodesic LCS, this means computing the gradient of the flow map on a grid of $500 \times 250$ grid points and using 4 auxiliary points at each grid point $x_0$ to use finite-differencing to compute the gradient of the flow map. FSLE similarly requires 4 auxiliary points in addition to the main grid point $x_0$ to measure the minimal separation time $\tau$ between the auxiliary points and the main grid. In contrast, the arclength function, trajectory complexity, spectral clustering and LAVD methods are computed on a grid of $1000 \times 500$ grid to ensure that the same number of points are used in the comparison. We compute the transfer operator and its hierarchical version using a partition of $250 \times 125$ boxes, with 16 particles per box. We show the results for all methods in the Figure 6.1.
Chapter 6. A Comparison of Lagrangian Coherent Structure Detection Methods

Figure 6.1: Comparison of Lagrangian methods on the quasiperiodic Bickley jet example (forward-time calculation only).

(a) $M$ function
(b) FSLE ($r=5$)
(c) Complexity Method
(d) FTLE
(e) Shape coherent candidate regions
(f) Geodesic LCS
(g) Transfer operator
(h) Transfer operator with hierarchy ($n=5$)
(i) Arbitrary field: $\log(7x^2+3xy+10)$
(j) LA/D
(k) Spectral clustering
(l)
Chapter 6. A Comparison of Lagrangian Coherent Structure Detection Methods

Figure 6.2: Advected image of Lagrangian coherent structures at the final time $t_1 = 11$ day for three different methods: (a) Geodesic (b) Spectral clustering and (c) LAVD. See the on-line supplemental movies M1, M2 and M3 for the complete advection sequence of the coherent structures.

The majority of scalar fields in Figure 6.1 indicate the presence of six vortices except for the shape coherence, transfer operator, mesochronic analysis, and geodesic method. Below, we discuss these exceptions in more details.

Figure 6.1e shows candidate regions (red) where shape coherent sets may exist at the initial time $t_0 = 0$. In these regions, the splitting angle between the dominant eigenvectors of the forward-time and the backward-time CG is smaller than $5.7^\circ$. As mentioned earlier, the regions so identified are expected to encompass vortex boundaries that have significant shape coherence over the time interval $[t_0, t_1]$ of interest. In Figure 6.1e however, the candidate regions have spiral-shaped appearance which refutes this expectation.

Figure 6.1i shows the result for the geodesic LCS analysis, where elliptic (e.g., vortices), parabolic (e.g., jet core) and repelling hyperbolic LSCs are shown in green, blue and red respectively. These LCS are distinguished material lines that have a prolonged influence on nearby trajectories through out the time interval of interest. As for vortex detection in this example, the geodesic method misses three out of six vortices present in the computational domain. This is because of the strict definition of coherence that the geodesic method takes into account, which requires the uniform stretching for the vortex boundaries. In Figure 6.2a we examine the coherence of the detected geodesic vortices by advecting them to the final time $t_1 = 11$ day. The complete evolution sequence is illustrated in the online supplemental movie M1.

Figure 6.1d shows the mesochronic partitioning of the domain into three different mixing zones: pure strain (blue), mesoelliptic (green) and mesohyperbolic with strain (red). The vortex regions are then defined as alternating, nested sequences of circular mesoelliptic and mesohyperbolic annuli [141]. To identify vortex structures in this example, we plot the contours of
mesochronic map and seek alternating nested sequences of circular mesoelliptic and mesohyperbolic (see Figure 6.3). Looking at Figure 6.3 we observe saddle-type critical points in all the vortex regions, implying that the color patterns are not circular. Hence, the mesochronic procedure put forward in [141] does not detect any of the vortices in this example.

As for jet identification, we observe that all the methods signal the presence of the central jet flowing to the right, except the shape coherence, spectral clustering and LAVD approach. The majority of methods, however, do not offer a systematic approach for extracting the jet core or jet boundaries. The only exceptions are the geodesic and the transfer operator methods that give a clear boundary for the jet core (see Figures 6.1g to 6.1i).

On this example, we detail the computation of spectral clustering method to clarify some important numerical aspects about this method. To apply the spectral clustering method, we first construct a trajectory data set with $n = 5 \times 10^5$ particles, whose position are evenly sampled over 300 intermediate times. We then construct the pairwise dynamic distances $r_{ij}$ using eq. (4.2.1), and subsequently construct the similarity matrix $W$ as defined in (6.18). Here, we only retain the similarity weights $w_{ij}$ whose values exceed the cutoff value $\epsilon = 1$. All other $w_{ij}$ entries are set to zero and hence require no storage in sparse matrix data structure. In practice, this can be done concurrently with computing pairwise distances $r_{ij}$ using nearest neighbor search algorithms [11]. Next, we perform a partial eigendecomposition and find the first 20 generalized eigenvectors $u_k$ of the graph Laplacian $L$. Figure 6.4 shows the computed generalized eigenvalues of $L$ as a function of their indices. In Figure 6.4, we observe that the largest gap exist between the sixth and seventh eigenvalues, indicating the presence of six coherent structures in the domain. The information concerning these coherent structures can then be found in the generalized eigenvectors $u_1, \ldots, u_6$, which are also known as cluster indicators (see Figure 6.5). Finally, we extract these six coherent structures in addition to the complementary incoherent region from the cluster indicators using the K-means algorithm (see Figure 6.1). In Figure 6.2b, we confirm the coherence of extracted vortices by advecting them to the final time $t_1 = 11$ day. The complete evolution sequence is illustrated in the online supplemental movie M3.
Figure 6.4: Sorted generalized eigenvalues for the graph Laplacian $L = D - W$ for the quasiperiodic Bickley jet flow.

Figure 6.5: The first six generalized eigenvectors of the graph Laplacian $L$ are computed with the spectral clustering approach [90] for the Bickley jet flow. These eigenvectors serve as cluster indicators and highlight individual vortices at the initial time $t_0 = 0$. 
6.3.2 Two-dimensional turbulence

In this second example, we solve the forced Navier–Stokes equation

$$
\partial_t v + (\nabla v) v = -\nabla p + \nu \Delta v + f, \quad \nabla \cdot v = 0,
$$

for a two-dimensional velocity field $v(x,t)$ with $x = (x_1, x_2) \in U = [0, 2\pi] \times [0, 2\pi]$. We use a pseudo-spectral code with viscosity $\nu = 10^{-5}$ on a $512 \times 512$ grid, as described in [67]. A random-in-phase velocity field evolves in the absence of forcing ($f = 0$) until the flow is fully developed. At this point, a random-in-phase forcing is applied. For the purposes of the following Lagrangian analysis, we identify this instance with the initial time $t = 0$. The finite time interval of interest is then $t \in [0, 50]$.

Figure 6.6 shows the result of various Lagrangian methods applied to the resulting finite-time dynamical system $\dot{x} = v(x,t)$. For reference, an arbitrary scalar field $\sin(2x(t)) \sin(2y(t))$ is also plotted over the initial conditions $(x_0, y_0)$. The parameters are set-up such that the total number of particles advected is $1024 \times 1024$ for all methods. We use the auxiliary grid approach with the distance $\rho = 10^{-3}$ to construct the FTLE, FSLE, mesochronic and shape coherence diagnostic fields. The same auxiliary distance is used to compute the Cauchy–Green strain tensor as well as the vorticity for the geodesic and LAVD methods, respectively.
Chapter 6. A Comparison of Lagrangian Coherent Structure Detection Methods

Figure 6.6: Comparison of Lagrangian methods on the two-dimensional turbulence simulation example.
Most scalar fields in Figure 6.6 signal vortex-type structures except for the shape coherence and transfer operator methods. The hierarchical application of the transfer operator method, however, also signals vortex-like structures. In addition to the vortices, the hierarchical approach reveals many non-vortical structures as well that remain coherent under diffusive perturbations imposed numerically. These non-vortical structures, however, cannot be distinguished from the vortical structures using this approach. An additional shortcoming of this approach is its convergence issue: The hierarchical approach sets a threshold on the relative improvement of $\mu$, which needs to be measured and satisfied over consecutive refinements of coherent pairs. However, the reference measure of probability $\mu$ at each iteration depends on the initial numerical diffusion imposed by the box covering. As a consequence, identifying similar coherent sets under various box covering resolutions requires different threshold values. Figure 6.7 shows the hierarchical coherent sets obtained with a fixed termination threshold for three different box covering resolutions. Figure 6.7 indicates overall no convergence, except for some minor details.

Figure 6.6e shows candidate regions (red) where shape coherent sets may exist at the initial time $t_0 = 0$. In these regions, the angle between stable and unstable foliations is smaller than 5.7°, and vortex boundaries might be found. A quick inspection of fig. 6.6e, however, reveals that the majority of these candidate regions are spiraling, and hence vortex boundaries cannot be detected.

Figure 6.6i shows the geodesic Lagrangian vortex boundaries (green) as well as the repelling hyperbolic LCSs (red) at the initial time $t_0 = 0$. These hyperbolic LCSs are defined as locally the most repelling and attracting material lines that form the centerpieces of turbulent mixing. By contrast, the coherent Lagrangian vortices are defined as the outermost elliptic LCSs that stay invulnerable to straining and filamentation under advection. In Figure 6.8a we confirm the sustained coherence of the geodesic vortex boundaries by advecting them to the final time.
Figure 6.8: Advected image of Lagrangian coherent structures at the final time $t_1 = 50$ for three different methods: (a) Geodesic (b) Spectral clustering and (c) LAVD. See the on-line supplemental movies M4, M5 and M6 for the complete advection sequence of the coherent structures.

$t_1 = 50$. The complete evolution sequence is illustrated in the online supplemental movie M4.

Figure 6.6j shows the coherent sets detected by the spectral clustering method at the initial time. These coherent sets include the vortices captured by the Geodesic and LAVD methods, besides some other structures which are not captured by other methods. Figure 6.8b shows the advected image of these coherent sets at the final time $t_1 = 50$. The advected image confirms that particles composing these coherent sets remain compact over the finite time $[0, 50]$ of the flow evolution, which is consistent with the claim of spectral clustering method. The complete evolution sequence of the extracted coherent sets is illustrated in the online movie M5.

Figure 6.6k shows the Lagrangian vortex boundaries extracted using LAVD method at the initial time $t_0 = 0$. In this computation, we have set the minimum arc-length, $l_{\text{min}} = 0.3$ and convexity deficiency bound $d_{\text{max}} = 0.005$. In Figure 6.8c, we confirm the Lagrangian rotational coherence of these vortex boundaries by advecting them to the final time $t_1 = 50$. As expected, the vortex boundaries do not give in the general trend of exponential stretching and folding observed for generic material lines. Instead, they display only local (tangential) filamentation. The complete advection sequence over the time interval $[0, 50]$ is illustrated in the online supplemental movie M6.

Beyond showing the results of various methods, we also use this example to investigate weather contours of diagnostic tools such as the arclength function or mesochronic field can be used for the purpose of vortex detection. Specifically, we extract the contours of these two diagnostic methods for two selected vortex regions at initial time $t_0 = 0$, and advect them to the final time $t_1 = 50$. In addition, we make a comparison with the geodesic vortex boundaries obtained for the same regions.
Figure 6.9 shows the advection of the level-curves of the arclength function $M_{0}^{50}$ around two select vortices. The level-curves closer to the vortex core remain coherent for both vortices. A comparison with the geodesic vortex boundary, however, shows that the contours of $M_{0}^{50}$ underestimate the size of the upper vortex substantially. On the other hand, the mesochronic analysis does not signal vortex-type structures in any of the selected regions due to presence of the saddle-type critical points [141].
Figure 6.9: M-function contours (black curves) and the geodesic vortex boundary (red curves) at the initial time $t = 0$ (left) and at the final time $t = 50$ (right).

Figure 6.10: Contours of the mesochronic scalar (black curves) and the geodesic vortex boundary (red curves) at the initial time $t = 0$ (left) and at the final time $t = 50$ (right).
6.3.3 Wind field from Jupiter’s atmosphere

We use different LCS methods to uncover mixing barriers in the unsteady velocity field extracted from video footage of Jupiter’s atmosphere. The video footage is acquired by Cassini spacecraft and covers 24 Jovian days, ranging from October 31 to November 9 in year 2000. To reconstruct the velocity field, we apply the Advection Corrected Correlation Image Velocimetry (ACCIV) method [12] that yields a high-density, time-resolved velocity vectors that can be used for constructing high-resolution trajectory or flow map (cf. [92] for more details).

In this example, we use a total number of $1800 \times 1200$ particles for all the methods. The spatial domain $U$ in question ranges from $-61.6^\circ$ W to $-31.6^\circ$ W in longitude and from $-8.9^\circ$ S to $-28.9^\circ$ S in latitude. We perform the computation of gradient-based approaches such as FTLE, FSLE, Mesochronic, shape coherence and Geodesic LCS analysis using auxiliary grid approach to ensure high-precision and numerical stability in the finite-differencing. This implies that an embedded grid of $900 \times 600$ used to construct the corresponding scalar fields. In contrast, we use a uniform grid of $1800 \times 1200$ for the gradient-free methods. As for transfer operator-based approaches, we use a grid of $450 \times 300$ boxes, with 16 uniformly uniformly sampled points per grid box. Here, we use a variable-order Adams–Bashforth–Moulton solver (ODE113 in MATLAB), with relative and absolute tolerances of $10^{-6}$, for trajectory advection. we obtain the velocity field at any given point by interpolating the velocity data set using bilinear interpolation.

As seen in Figure 6.11, several methods signal vortex-like structures except for the transfer operator, shape-coherent, and mesochronic analysis. As in the case of our previous example, however, the hierarchical application of the transfer operator also signals vortex-like structures (see fig. 6.11h). The mesochronic analysis, again, does not signal vortex-type structure because of the saddle-type critical points present in the region filled with closed contours of alternating colors.

As for jet identification, most scalar fields in Figure 6.11 give some indications about the two existing jets passing of north and south of the GRS. However, in the absence of a clear procedure for extracting jet-type structures from most methods surveyed here, we could not go beyond visual comparison of the scalar fields. Perhaps, the only exceptions are geodesic and transfer operator with hierarchy methods which signal clear boundaries for jet cores.

In Figure 6.12, we advect the vortex boundaries yield by geodesic LCS analysis, spectral clustering and LAVD method to the final time $t_1 = 24$ in order to verify their coherence. See on-line supplemental movies M7, M8 and M9 for the advection sequence of the so obtained coherent structures.
Figure 6.11: The output of all methods at initial time $t_0$ for Jupiter's wind-velocity field [92].
Chapter 6. A Comparison of Lagrangian Coherent Structure Detection Methods

6.4 Summary and conclusions

In this chapter, we have reviewed eleven Lagrangian methods for identifying the coherent structures in the fluid domain. For each surveyed method, we discuss its objectivity or lack thereof, in conjunction with its strengths and weaknesses. For the latter, we discuss specifically two practical aspects of each method: First, the computational effort required by each method in terms of particle advection. Second, the degree of autonomy in terms of the number user inputs needed from the user. Finally, we demonstrate the performance of each method for three different examples, ranging from analytic velocity fields to time-dependent observational data set.

Generally speaking, Lagrangian methods have the advantage over the Eulerian methods by being insensitive to transient anomalies and outliers in the velocity or vorticity fields [87]. While all the Lagrangian methods benefit from this feature, they differ in terms of the degree of sensitivity. In our experience, the gradient-free approaches tend to be more robust with respect to the numerical noise, and hence may perform better over long integration times when numerical noise is more prevailing. This can be visually observed by directly comparing the output of a derivative-based method such as Mesochronic plot [142] with a derivative-free approach such as arclength function [139] (see Figures 6.9 and 6.10 for example).

Another important aspect of Lagrangian coherent structure detection methods is their computational cost, which can be viewed in terms of CPU-time and memory consumption. One major source of computational cost for Lagrangian methods is advecting Lagrangian particles. Lagrangian methods differ significantly in terms of the number of Lagrangian particles they require for the computation. For instance, gradient-based approaches (e.g., FTLE, Mesochronic, Geodesic, and shape coherence) require four times higher resolution grids compared to the
gradient-free approaches (e.g., arclength function, trajectory complexity, spectral clustering and LAVD), when they are implemented with the auxiliary grid technique. In our experience, the transfer operator-based approaches \[80, 134\] require, at least, as many Lagrangian particles as the gradient-based methods, to construct the transition matrix. While the demand for a large number of Lagrangian particles may sound as a minor computational issue, in fact, it turns to be a major conceptual problem when one deals with low resolution data sets. In this case, the spatial differentiation or delineating probability density may not be meaningful given the relatively coarse resolution of the data set and the interpolation error associated with the approximation of the nonlinear velocity field.

We also observed that an arbitrary scalar field performs well in comparison to several of the heuristic diagnostic tools surveyed in this chapter. This is not surprising knowing that Lagrangian coherent structures leave observable footprints in any transport-related indicator associated with the flow. This manifestation, for instance, can be observed in various physical processes in the ocean, ranging from larval transport \[105\] and algal blooms \[147\] to massive transport of salinity and temperature via coherent structures \[16\]. These imprints, however, do not reveal the time-evolving transport barriers as the root cause of flow coherence \[99\]. The theory of Lagrangian coherent structures seeks to determine this root cause by identifying distinguished surfaces that split the fluid domain into regions with dynamically distinct behaviors \[99\]. These distinguished surfaces will not generally not align with the contours of path-averaged scalars over a finite time interval. Hence, the reverse engineering approach to seek coherent structures through their footprints seems to be a non-constructive path to take.

As is well known, the physical properties of a fluid such as shear or strain do not depend on the frame of the observer \[54\]. The implication is that questions that are inherently about material (i.e., Lagrangian) properties of the fluid should be addressed similarly in terms of the objective physical quantities. The question whether a fluid parcel remains coherent or mixes with its surroundings is one such question that needs to be answered irrespective of the observer, as the question itself is expressed without reference to a coordinate frame. As a first step, one needs to formulate the question in terms of equations involving frame-independent (objective) variables. What then follows is a solution or a method that consists of objective variables and quantities. This fact, no matter how simple it sounds, has been ignored in several of the Lagrangian diagnostic tools developed in the past few years (see \[139, 142\], for example). A possible reason for this is the lack of a universal definition for coherence, which in turn has promoted individualistic definitions for coherence in fluid flows. Before selecting a unified definition for coherence, however, the fluid dynamics community should ideally reach a consensus that only those coherence definitions that are objective can be regarded as plausible definitions.
Chapter 7

Bibliography


[141] I. Mezić. personal communication. 2015.


CURRICULUM VITAE – ALIREZA HADJIGHASEM

CONTACT INFORMATION
Institute for Mechanical Systems
ETH Zürich, LEE, M 228
Leonhardstrasse 21
8092 Zürich
Email: alirezah@ethz.ch
Web: www.zfm.ethz.ch/~hadjighasem

RESEARCH INTERESTS
Nonlinear Dynamical Systems
Mathematical and Computational Fluid Mechanics
Machine Learning and Data Mining
Digital Image and Video Processing

ACADEMIC APPOINTMENTS
Postdoctoral Associate
Department of Mechanical Engineering
Massachusetts Institute of Technology (MIT)
Mentors: Prof. Pierre Lermusiaux and Prof. Thomas Peacock

EDUCATION
Ph.D.
Mechanical Engineering, ETH Zürich
Thesis: Extraction of dynamical coherent structures from time-varying data sets
Advisor: Prof. George Haller
Co-advisor: Prof. Markus Holzner

M.Eng. (45 credits)
Mechanical Engineering, McGill University
M.Eng project: Detecting invariant manifolds in aperiodically forced mechanical system
Advisor: Prof. George Haller
GPA: 3.94/4

B.Sc. (144 credits)
Aerospace Engineering, Khaje Nasir Toosi University of Technology
Thesis: Fault Tolerant Control Techniques
Advisor: Prof. Hassan Karimi Mazraeshahi
GPA: 3.4/4

Diploma in Mathematics and Physics Discipline
Rouzbeh Cultural Institute
GPA: 19.11/20

AWARDS AND HONORS
McGill University Lorne Trottier Graduate Fellowship (declined) 2013
McGill University Graduate Excellence Fellowship (declined) 2010
McGill Engineering Doctoral Award (MEDA) (declined) 2012
DFW award, McGill University 2011
M.Sc. admission award, IRAN 2010
Exemption from the entrance exam of universities
Ranked 4th in Aerospace Engineering Department, Khaje Nasir University 2010
Ranked 2nd in LTA (Lighter-Than-Air) vehicles competition, IRAN 2009
PUBLICATIONS

**Published:**


Hadjighasem, A., Karrasch, D., Teramoto, H., & Haller, G. (2016). *Spectral-clustering approach to Lagrangian vortex detection*. Physical Review E, 93(6), 063107. *Figure 16(c) from this paper is selected for PRE Kaleidoscope.*


This paper has also been appeared at popular science news services: SIAM Nuggets, ScienceDaily, phys.org and ETH news


**Preprint:**


**In Preparation:**


INVITED TALKS


CONFERENCE PRESENTATIONS

A Spectral Clustering Approach to Lagrangian Vortex Detection, 68th Annual Meeting of the APS Division of Fluid Dynamics, November 2015, Boston, USA.


Detecting coherent structures from video streams of Jupiter, MFO Workshop on Mixing, Transport and Coherent Structures, Oberwolfach, Germany.

Attractors, and Generalized KAM Tori in Aperiodically Forced Mechanical Systems, SIAM Conference on Applications of Dynamical Systems, Snowbird, Utah, USA.

TEACHING EXPERIENCE

ETH Zürich

Mechanics III, Teaching Assistant, Fall 2014

Mechanics III, Teaching Assistant, Spring 2013
RESEARCH EXPERIENCE

Research Assistant

Research assistant at McGill University after completion of the Master’s degree Aug. 2012-May 2013

Coursework projects:
- Mathematical modeling & computer simulation of pole vaulting 2011
  Advisor: Prof. Rama Bhat
- Aircraft Design (Including CATIA simulation) 2010
  Advisor: Prof. H. Sadraei
- Mathematical modeling of flow around wing-fuselage junction 2009
  Advisor: Prof. Masoud Mirzaei
- Shockwave drag reduction 2009
  Advisor: Prof. Reza Ebrahimi
- Mathematical modeling of regenerative cooling method in nozzles 2008
  Advisor: Prof. Hassan Karimi Mazraeshahi

WORK EXPERIENCE

Internship at Tehran Aerospace Exhibition Center June 2008-Sept. 2008

Field supervisor at Mehrz Bana Construction and Engineering Company in a part-time manner 2006-2010

PROGRAMMING SKILLS

MATLAB & Simulink
MPI parallel programming and distributed programming
C#, MATHCAD, CATIA, Visual NASTRAN

SOFTWARE

**LAVD:** A tool to detect rotationally coherent Lagrangian vortices automatically in 2D/3D unsteady flows. [GitHub Link]

**Elliptic LCS:** A tool to detect Geodesic vortex barriers in a fast and automated way in 2D unsteady flows. [GitHub Link]

REFERENCES

George Haller
Professor, Chair in Nonlinear Dynamics
Department of Mechanical and Process Engineering
Swiss Federal Institute of Technology - ETH Zürich
georgehaller@ethz.ch

Markus Holzner
Assistant Professor
Department of Environmental Fluid Mechanics
Swiss Federal Institute of Technology - ETH Zürich
holzner@ifu.baug.ethz.ch

Farhang Daneshmand
Associate Professor
Department of Bioresource Engineering
McGill University
farhang.daneshmand@mcgill.ca