

CLARKSON UNIVERSITY

**Transport Analysis and Motion Estimation of Dynamical Systems of
Time-Series Data**

A dissertation

by

Naratip Santitissadeekorn

Department of Mathematics and Computer Science

Submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Applied Mathematics

Date

Accepted by the Graduate School

Date

Dean

CLARKSON UNIVERSITY

The undersigned have examined the dissertation entitled **Transport Analysis and Motion Estimation of Dynamical Systems of Time-Series Data** presented by **Naratip Santitissadeekorn** a candidate of the degree of **Doctor of Philosophy** and hereby certify that it is worthy of acceptance.

Date

Erik Bollt (Advisor)

Examining Committee

Joseph Skucfa

Date

Daneil Ben-Avraham

Date

Scott Fulton

Date

Brian Hallenbrook

Date

Transport Analysis and Motion Estimation of Dynamical Systems of Time-Series

Data

Copyright © 2008

by

Naratip Santitissadeekorn

Abstract

Transport Analysis and Motion Estimation of Dynamical Systems of Time-Series

Data

by

Naratip Santitissadeekorn

Doctor of Philosophy in Applied Mathematics

Clarkson University

This thesis addresses the global analysis of dynamical systems known only via a sequence of data, i.e., experimental time-series data, or video data (image sequences). A primary interest of this thesis is to reconstruct a velocity field from a 2-D image sequence. The problem of motion estimation has been intensively studied in the last two decades in many areas of computer vision to perform motion detection and tracking, object segmentation, time-to-collision estimation, and etc. However, an application of the motion estimation has not yet been demonstrated in the area of measurable dynamical systems, in particular, to approximate the Frobenius-Perron transfer operator from a sequential image data. Therefore, we offer a practical technique to reconstruct a dynamical model from image sequences from various areas of science. Also, we provide a mathematical formalism in a framework of the Frobenius-Perron operator to validate our technique. Furthermore, We explore and apply certain graph theory methods to identify almost invariant sets. These almost invariant set corresponds to the situation of basin-hopping in a multi-stable system under a small stochastic perturbation. For a general time-dependent vector field, we employ the concept of the finite-time Lyapunov exponent (FTLE) to define the time-varying Lagrangian coherent structure. We demonstrate some applications of the combination of the above tools to a study of transport and mixing for certain real-world mixing devices.

Contents

Contents	i
List of Figures	iv
List of Tables	xi
Acknowledgements	xii
1 Introduction	1
1.1 Motivation and Background	1
1.2 Purposes	5
1.3 Organization	7
2 Dynamical systems terminology and definitions	10
2.1 Linearization	10
2.2 Hyperbolicity	12
2.3 Hyperbolicity: Nonautonomous vector fields	16
3 Frobenius-Perron Operator and Infinitesimal Generator	22
3.1 Frobenius-Perron Operator	22
3.2 Infinitesimal Operators	25
3.3 Finite-rank approximation of the Frobenius-Perron operator	29
3.4 Almost invariant set	33
4 Graph Community Structure	36
4.1 Graph Theory and Reducible Matrices	36
4.2 Modularity Method	40

4.3	Markov Clustering	47
4.4	Examples	51
4.4.1	Duffing Oscillator	52
4.4.2	Gearbox Model	53
4.4.3	A Stochastic Model with many basins	56
4.5	On Computational Complexity	61
5	LCS in 2-D dynamical systems	65
5.1	Lyapunov exponents	65
5.2	Finite-time Lyapunov exponents and LCS	68
5.3	LCS analysis of a mixing device	71
5.3.1	Experimental setup	71
5.3.2	Transport and Mixing in the Mixers	74
5.3.3	Mathematical model of the mixers	80
5.4	Mixing notions	83
6	Inverse Problem of Motion Estimation	87
6.1	Inverse Problem of Motion Estimation	87
6.2	Euler-Lagrange Equation	91
6.3	Discontinuity-Preserving Regularization	96
6.4	PIV Approach	98
6.5	Coarse-to-Fine Approach	100
6.6	Experimental Evaluation	103
6.6.1	Example 1	104
6.6.2	Example 2	111
6.6.3	Example 3	113
6.6.4	Example 4	122
7	Conclusion	129
7.1	Contributions	130
7.2	Open questions and further research	131
	Bibliography	133
A	Ridge detection	139

B	Differentiation of image intensity	144
C	Mass conservation in transmittance imaging	146

List of Figures

1.1	(a) Potential wells of the Duffing oscillator. (b) Vector fields and the homoclinic orbits of the hyperbolic point o . The homoclinic orbit separates the motion of points starting below the dash line (e.g. points a or a') in (a) from those initially located above the line (e.g. points b or b').	3
2.1	An example of a homoclinic orbit of a hyperbolic fixed point p . Here q is the homoclinic point and the map f is assumed to be an orientation-preserving map, in which there must be at least one homoclinic point between f and $f(q)$	15
2.2	An example of a heteroclinic orbit connecting two hyperbolic fixed points p_1 and p_2 . Here q is the heteroclinic point and the map f is assumed to be an orientation-preserving map as in Figure 2.1	16
2.3	The time slice $\Sigma_\tau = E^s(\tau) \oplus E^u(\tau)$. A trajectory of an initial segment projected onto $E^s(\tau)$ by the associated linearized vector field (2.13) approaches the hyperbolic trajectory $\Gamma(t)$ at an exponential rate as $t \rightarrow \infty$. Similarly, the initial vector projected onto $E^u(\tau)$ approaches $\Gamma(t)$ at an exponential rate as $t \rightarrow -\infty$.	18
2.4	The forward-time evolution of a segment of $W^u(\gamma(t_1))$ from the time slice Σ_{t_1} to the time slice Σ_τ yields the unstable manifold of $W^u(\gamma(\tau))$. Similarly, the backward-time evolution of a small segment of $W^s(\gamma(t_2))$ from the time slice Σ_{t_2} to the time slice Σ_τ yields the unstable manifold $W^s(\gamma(\tau))$.	20
3.1	A simulation of the flow of a density function according to the continuity equation (3.32) with the velocity field given by Eq. (3.31). The time increment between each snapshot is $t = 0.08$.	30
4.1	The matrix in this figure is irreducible. Members of the original transition matrix before sorting are placed haphazardly and the structure is not obvious. After using the modularity method discussed in Section 4.2 the “almost” block diagonal form is revealed. In this case, the sorted matrix indicates the existence of three communities and the few off block diagonal elements correspond to intra-basin diffusion.	38

4.2	An undirected graph before and after sorting. The arrows point at the inter-community edges.	39
4.3	Plot of dendrogram for a graph network in Figure 4.2(a). As we can see the peak in the modularity (dashed line) occurs where the network is divided into three communities.	45
4.4	A graph network of 20 vertices.	49
4.5	The change in matrix P at different iterations of MCL with $r = 2.0$. (a) The original stochastic matrix P . From (b)-(d) we show the matrix P at steps 10,20 and 30 respectively. Notice that the matrix in (d) reveals three communities in the network, that is $\{1 - 5, 18, 20\}$, $\{6 - 14, 19\}$ and $\{15 - 17\}$	50
4.6	The effect of the scale of r is shown at three different scales. (a) For $r = 1.2$ two communities, $\{15 - 17\}$ and $\{1 - 20\} \setminus \{15 - 17\}$, are detected by MCL. (b) For $r = 1.5$ MCL discovers three communities (c) At $r = 2$ MCL detects four communities. Note that for $r > 2$ MCL gives the same result as the case of $r = 2$	50
4.7	A partition of an orbit generated from the Duffing model with $\delta = 0.095$ and normal noise, $\sigma = 0.01$. The colors, magenta and yellow, are according to the two communities found by the modularity method applied to the transition matrix, and we see that the colors of these corresponding triangles agree well with the a priori known basins of the deterministic system. The blue dots shown are the test orbit used in the process.	54
4.8	Galerkin matrices before and after sorting of the Duffing oscillator. (a) The Ulam-Galerkin matrix from the Duffing oscillator before sorting. No obvious basin structure or transport in this operator. (b) The Ulam-Galerkin matrix from the Duffing oscillator after sorting into into approximate reducible form. We see the basin structure of off-diagonal elements showing relative transport between basins.	55
4.9	Schematic view of the gearbox system [1]	56
4.10	Bifurcation diagram for the time of impact.	57
4.11	A time series of the τ_n for $\alpha = 1.985$ at noise level $\sigma = 0.0018$	57
4.12	A partition of an orbit generated from the gearbox model with normal noise, $\sigma = 0.0018$	58
4.13	Transition matrix before and after sorting of the gearbox model. Notice in the sorted matrix that two distinct basins become clear as those triangles indexed to associate to the diagonal blocks, and the regions causing hopping are clear as those few off diagonal elements. Each of the two basins, the smaller and larger blocks, are the pink and yellow colored regions in Figure 4.12.	59
4.14	The velocity field for the noiseless case of Eq. (4.35) is computed from the above streamfunction by the relation $(dx/dt, dy/dt) = \nabla \times \psi$	60
4.15	A test orbit of the time-T map of Eq. (4.35) for a normal noise with standard deviation $\sigma = 0.001$	60

4.16	Transition matrix before and after sorting of the model Eq. (4.35). Notice decreasing the number of basins as the noise increases.	62
4.17	The phase space partition of the duffing-like model with various values of σ using the graph modularity method. The values of the modularity measure as defined in Eq. (4.5) in each case are $Q = 0.7267, 0.7183, 0.6968$, and 0.631 for $\sigma = 0.001, 0.005, 0.01$ and 0.05 , respectively. Similarly, the basin structure is changing with σ	63
4.18	Two iterations of the Baker map with $\lambda_1 = \lambda_2$. (a) All stripes after $n = 2$. A square box of size ϵ_1 cannot resolve a structure finer than ϵ_1 . But we see that a smaller square box of size ϵ_2 is able to resolve a stripe of width up to ϵ_2 . (b) Here, however, the box of size ϵ_2 cannot resolve the stripe left in the unit box after $n = 3$	64
5.1	Experimental mixing apparatus.	71
5.2	Representative phase averaged velocity and vorticity fields from PIV data. Case shown is from the centered data.	73
5.3	The forward-time FTLE field of the mixer with the centered blade is computed at different time t with the integrating time $\tau = 64\Delta t$, where $\Delta t = 0.033$ seconds. From (a) to (f), the FTLE is computed at time $t = \Delta t$, $t = 10\Delta t$, $t = 20\Delta t$, $t = 30\Delta t$, $t = 40\Delta t$ and $t = 60\Delta t$, respectively. The initial particles are placed to straddle the LCS and they are advected to demonstrate that the resulted LCSs serve as the transport barrier.	75
5.4	The backward-time FTLE field of the mixer with the centered blade is computed at different time t with the integrating time $\tau = 64\Delta t$ as in Figure 5.3. From (a) to (c), the FTLE is computed at time $t = \Delta t$, $t = 60\Delta t$ and $t = \tau + 60\Delta t$, respectively. Here we observe that the initial particles colored by the magenta accumulated along the ridge of the FTLE field. Thus the ridge of the backward-time FTLE field may be regarded as an unstable manifold.	76
5.5	Snapshots of particle tracers for the mixer with the centered blade. Two sets are particles, A (green) and B (magenta) are initially separated to occupy half of the physical space as shown (a). The initial particle distribution is then allowed to change as dictated by the flow field for 1,2,3,4, and 5 periods, (b)-(f), respectively.	77
5.6	A heuristic rendering of a stable and unstable heteroclinic tangle made of the mixing system inferred from the observed time-dependent LCS structures. The periodic points P_1 and P_2 correspond to the blade tips. The forward and backward time FTLE fields in Figure 5.3 and 5.4 suggest that the transverse intersection by the stable and unstable manifolds of P_1 and P_2 generate a trapping region that bounds the trajectories of particles within this cell. Note that the mixing inside the trapping region is a result of the hyperbolic fixed point at the center of the mixer.	77
5.7	Computed the FTLE fields with different integrating times, $\tau = T$, $\tau = 2T$ and $\tau = 4T$, respectively, where T is the time for one period.	78

5.8	The forward-time FTLE field of the mixer with the offset blade is computed at different times t with the integrating time $\tau = 64\Delta t$. From (a) to (f), the FTLE is computed at time $t = \Delta t$, $t = 10\Delta t$, $t = 20\Delta t$, $t = 30\Delta t$, $t = 40\Delta t$ and $t = 50\Delta t$, respectively.	79
5.9	Snapshots of particle tracers for the mixer with the centered blade. Two sets are particles, A (green) and B (magenta) are initially separated to occupy half of the physical space as shown (a). The initial particle distribution is then allowed to change as dictated by the flow field for 1,2,3,4, and 5 periods, (b)-(f), respectively.	80
5.10	The vertical vorticity field, ω_z , of the a) experimental data and b) analytic model using Eq. (9) with $C=2.0$	82
5.11	The in-plane velocity field and representative streamlines for the a) experimental data and b) analytical model with $C=2.0$. Note the spatial dimensions have been normalized by the radius of the cylinder.	82
5.12	The forward-time FTLE field at time $t =$ of the model Eq. (5.15) in a rotating frame. The parameter C in the vorticity field Eq. (5.17) is varied to observed the change of the FTLE field. (a) $C = 0.5$, (b) $C = 1.0$, (c) $C = 1.5$, (d) $C = 2.0$, (e) $C = 2.5$ and (f) $C = 3.0$	83
5.13	The backward-time FTLE field at time $t =$ of the model Eq. (5.15) in a rotating frame. The parameter C in the vorticity field Eq. (5.17) is varied to observed the change of the FTLE field. (a) $C = 0.5$, (b) $C = 1.0$, (c) $C = 1.5$, (d) $C = 2.0$, (e) $C = 2.5$ and (f) $C = 3.0$	84
5.14	The plot of the ratio R in Eq. (5.24) at each period using different scales of N_R . (a) the mixer with a centered blade. (b) the mixer with an offset blade	86
6.1	The illustration of the assumption implied by Eq. (6.4)	89
6.2	The process of the backwards warping.	100
6.3	The original image is at the bottom of the pyramid (the highest level). Each lower level are the subsample version of the previous level. Starting from the lowest level the velocity is approximated and then upsampled to the next level. The second image is then warped toward the first image based on the upsampled version of the velocity filed. The incremental velocity field is then computed. The process is repeated until the highest level is reached. . . .	101
6.4	A sample image at different level of the pyramid. The original image is shown in (d).	102
6.5	(a) The velocity field of the dynamical system (6.38) defined on the unit box $[0, 1] \times [0, 1]$. It can be seen that there are four basins separated by two nullclines $x = 0.5$ and $y = 0.5$. (b) The initial distribution at time $t = 0$. . .	105
6.6	A sequence of some images captured from the numerical simulation of the flow continuity equation Eq. (3.28) with the velocity field given by the ode Eq. (6.38). The time increment between each snapshot is $t = 32$	106

6.7	A plot of the average angular error with respect to parameters α and β . The sensitivity of the parameter α and β is observed in this plot. The thick line represents the case of the smoothness constraint Eq. (6.9) where $\alpha = \beta$. Although it is not conclusive that the smoothness constraint performs better than the div-curl regularizer in this experiment, our experimental result demonstrates that the optimal value is obtained when using the smoothness regularizer with $\alpha = \beta = 0.01$	107
6.8	(a) Exact velocity field (in green arrows) and the optical flow field for $\alpha = 1, \beta = 0.01$ shown in black arrows. (b) Exact velocity field (in green arrows) and the optical flow field for $\alpha = \beta = 0.01$ shown in black arrows.	108
6.9	Comparison between the exact and approximated velocity fields from the image sequence using the smoothness parameter $\alpha = \beta = 0.01$. The transparent solid lines illustrate the exact velocity field and the colored plots represent the approximation.	109
6.10	The plot shows the eigenvalues of the centers and the saddle node in the phase space of the exact velocity field in cross markers. The square markers locate the eigenvalues of those points computed from the estimated velocity field with the smoothness constraint $\alpha = 0.01$	109
6.11	When solving Equation (6.1) the time duration has to be chosen in a way that the (triangular) grid is expanded far enough so that it lie across itself as less as possible. Then the solution of the majority of randomly chosen points will not remain inside the initial grid. This prevents us from generating a graph network with inapplicable self-connected edges.	110
6.12	The phase space in this figure are partitioned by triangular grids. The edges that connect the nodes corresponding to the triangles that lie across the basin boundary are the intergroup edges, which connect between each community in the graph network.	111
6.13	The Ulam-Galerkin matrix generated by the exact velocity fields expressed in Equation 6.38 and by the approximated optical flow field using 200000 randomly chosen points.	112
6.14	The Ulam-Galerkin matrix after sorting using the modularity method. . . .	112
6.15	A sequence of some images captured from the numerical simulation of the Complex Ginzburg-Landau equation. The arrows point at the sinks that can be found through a careful observation and as revealed by the Ulam-Galerkin matrix analysis as in Figure 6.16- 6.19	114
6.16	The velocity fields approximated from the div-curl regularization with various values of parameters α and β are plotted side by side with the corresponding magnitude of the velocity fields.	115

6.17	(a) For a small β at $\alpha = 1$, the magnitude of the velocity field is more emphasized at the region where occlusion occurs. In this region the velocity become discontinuous. (b) For a large β at $\alpha = 1$, we observe that the velocity field is isotropically distributed throughout the image region. Nonetheless, the discontinuity can still be observed at the region where occlusion take places.	116
6.18	The Ulam-Galerkin matrix before and after sorting. Notice that the “almost” block diagonal form is unveiled after sorting. The middle block corresponds to those of several small communities that we group together. The other blocks represent the two large communities.	116
6.19	The phase space are partitioned into three regions corresponding two basins of the sinks corresponding to two communities discovered by the modularity method and the region that an initial point does not converge to neither basin.	117
6.20	Two pairs of images with differences in the number of particles, the particle diameter and the standard deviation of particle diameters. The number of particles for the pair (a)-(b) and (c)-(d) are 4000 and 1000, respectively. . .	118
6.21	Exact velocity field of the two image pairs shown Figure 6.20.	119
6.22	(a)-(c) show the vector fields obtained from the variational, cross-correlation, adaptive cross-correlation approaches for the image pair with dense particles, respectively. (d)-(f) compare the resulted obtained from the variational, cross-correlation, adaptive cross-correlation approaches for the image pair with sparse particles, respectively.	120
6.23	Average angular error (AAE) and average relative displacement error (ADE) for the image pair with dense particles.	121
6.24	Average angular error (AAE) and average relative displacement error (ADE) for the image pair with dense particles.	121
6.25	Raw experimental images: the first two frames of the images recorded from an experiment. The entire image sequence is published by Wieneke in CVGPR website as a test data for the PIV method.	122
6.26	(a) correlation-based PIV method. (b) The velocity field obtained from the lowest resolution level with the smoothness parameter $\alpha = 10$. (c) Frobenius-Perron based variational technique without using the coarse-to-fine scheme. From (d)-(f) The velocity fields are computed by the variational minimization based on the Frobenius-Perron operator with the coarse-to-fine strategy. (d) $\alpha = 1$. (e) $\alpha = 4$. (f) $\alpha = 10$. Note that the each velocity field is amplified by a factor of 2 to make it more perceptible.	124
6.27	(a) The velocity vectors obtained at the highest pyramid level (lowest resolution). (b) The velocity vectors obtained from the next lower pyramid level (higher resolution). The location bounded by the red rectangle in (a) and (b) are enlarged and shown in (c) and (d) to observe the appearance of the vorticity motion when the motion increment computed at the current level is used to correct the velocity vector obtained from the previous level. . . .	125

6.28	Three images of three slightly different patterns. However, the evolution of the pattern is governed by the same velocity field. (a)-(c) The first images of the three image sequences. (d)-(e) The final images of the three image sequences.	126
6.29	The variation of the average angular errors between the velocity fields of the three image sequences with respect to the smoothness parameter α	127
6.30	(a) The separation rate of the vector field of the frame 1. (b) The FTLE field of the frame 1. When comparing (a) and (b), we see that the separation rate computed at a frozen time can be misleading to indicate the coherent structure of the dynamical system. (b)-(f) The evolution of the FTLE field from frames 1 to 5. Notice that there are some small closed-loop coherent structures emerging as time progresses.	128
A.1	The image data of a smoothed FTLE field. The yellow dot is the initial point of the iteration and the magenta dots are the points (x_n, y_n) in the iterative algorithm. Each blue dot is the point (p_n, q_n) in the iteration, which has a distance r from its corresponding point (x_n, y_n) pointed in the direction of eigenvector ν_2 . The line in black color is in the direction of the line search subproblem and it is oriented according to the eigenvector ν_1 perpendicular to ν_2	142
A.2	The final result of the ridge detection for the FTLE field shown in Figure A.1 The green line plotted on the contour plot of the FTLE field is the ridge detected by the above algorithm.	143

List of Tables

6.1	Average angular error (AAE) and average relative displacement error (ADE) in the case of dense particle images.	119
B.1	Matched pairs of pre-filter(p) and derivative kernels(d_i) of the i^{th} order of various sizes.	145

Acknowledgements

I would like to thanks my advisor Erik M. Bollt for his constant guidance, support, and motivation. I owe a great intellectual debt to him for his patient mentorship that have provided insightful concepts on which this thesis is based. His influence on my professional development will benefit me throughout my carrier life.

I am also thankful to the research group consisting of Erik M. Bollt, Joseph Skufca, Daniel Ben-Avraham, Jim Bagrow, Hernán D. Rozenfeld, Chen Yao, Abbas Alkahim, Sun Jie, and other members who have joined the group in the past year. Many talks and discussions during the group meeting have expanded my knowledge of the graph network theory and its applications from which this thesis has also benefited.

My collaboration with Erik and Dough Bohl for the FTLE analysis of mixing devices has been both productive and fun. Particularly, I thank Dough Bohl for providing velocity field data and his experimental insights into the simulation results obtained from the FTLE and mixing computations.

I must also thanks Jim Bagrow for providing me a computer resource to remotely run the C++ code for a graph partition. Also, I would like to thank Akshey Mehta for running the cross-correlation methods to several image sequences used in this thesis.

I would also like to thank all the committee members, Erik M. Bollt, D. Ben-Avraham, J. Skufca, B. Halenbrook, and S. Fulton, for all their times and efforts to review and comment this dissertation.

I was supported by a National Science Foundation grant DMS-0404778 during my graduate work, which provided much freedom.

This list would not be complete without a special thanks to all my friends and my family. In particular, the constant support and encouragement of my mother and brother, Prapai and Poramin Santitissadeekorn, has accompanied me over many years of studies. Most important of all, I am grateful to Misun Kim to her encouragement and a number of things that motivated me to complete this thesis.

Chapter 1

Introduction

1.1 Motivation and Background

Informally, dynamical systems are mathematical models that describe the evolution of a point in its ambient space. The definition of a dynamical system in the semigroup formalism is given by the definition 3.1. In many cases, a physical system can be appropriately modeled by a differential equation

$$\dot{x} = F(x, t), \tag{1.1}$$

where $F(x, t)$ is the vector field, $x \in \mathbb{R}^n$ “particle” positions, and $t \in \mathbb{R}$ the time. When the velocity field $F(x, t)$ does not depend on time, that is $F(x, t) \equiv F(x)$, the model of a system is called an *autonomous* system. For such a system, a number of methods have been developed to analyze its transport behaviors. In particular, separatrices can exist for autonomous systems, which can divide the phase space into regions of qualitatively different motion. These separated regions can be either basin of attractions, basin cells, or trapping regions. Although different in their definitions, these regions share the same property that for a 2-D space trajectories of particles inside these regions remain inside the regions for all time. Thus, a separatrix is also known as a *transport barrier* since no particles can transverse across them. These transport barriers are often formed by the stable and unstable manifolds of certain homoclinic or heteroclinic points of the system. Roughly speaking, what we mean

by “transport” in this thesis is the transitions of particles between the regions separated by a geometrical structure such as the separatrices. Our main interest is to characterize phase space regions with qualitatively different dynamics and the transitions between these regions instead of individual phase space trajectories.

To illustrate the aforementioned concept of the separatrix, let consider as an example the unforced and dissipation-free system of the Duffing oscillator, which describes the motion of a particle in a potential well as shown in Figure 1.1(a), whose its corresponding vector field and homoclinic orbit are shown in Figure 1.1(b). In the finite-volume regions bounded inside the homoclinic orbit, all trajectories are periodic evolving around the center c (or c') whereas outside the homoclinic orbit, the particles move periodically in the phase space about the hyperbolic point. The first case corresponds to the particle in the potential released from the level below the dot line in Figure 1.1(a) and the latter describes the motion of particles above the dot line that visits both wells alternately during that motion. Note that the homoclinic orbit itself is an *invariant manifold*, i.e., all trajectories of particles in the set of homoclinic orbit will always be in the set. Hence, the problem of extracting separatrices is related to that of detecting the invariant manifolds of the system. It will be explained later that when a system with an invariant manifold is perturbed by a random noise, it is still possible to statistically identify an “almost invariant” set.

An analysis of the transition between the qualitatively different regions gives rise to a notion of *mixing* or *chaotic mixing*. However, for a deterministic autonomous system, the notion of mixing is quite transparent since each point in a phase space has a unique trajectory passing through it. Also, many real-world applications are autonomous dynamical systems stochastically perturbed by noise instead of the deterministic one. For these cases, the analysis of mixing becomes more important.

Noise in deterministic dynamical systems can play an important role in a global change in the system. The stochastic dynamics, can be dramatically different from the unperturbed deterministic analogue, even under small noise inputs. For example, additive noise in a multi-stable system can cause basin hopping between two states [1, 2, 3, 4, 5, 6]. The questions of when noise induces chaos, phase space chaotic transport, and mean escape

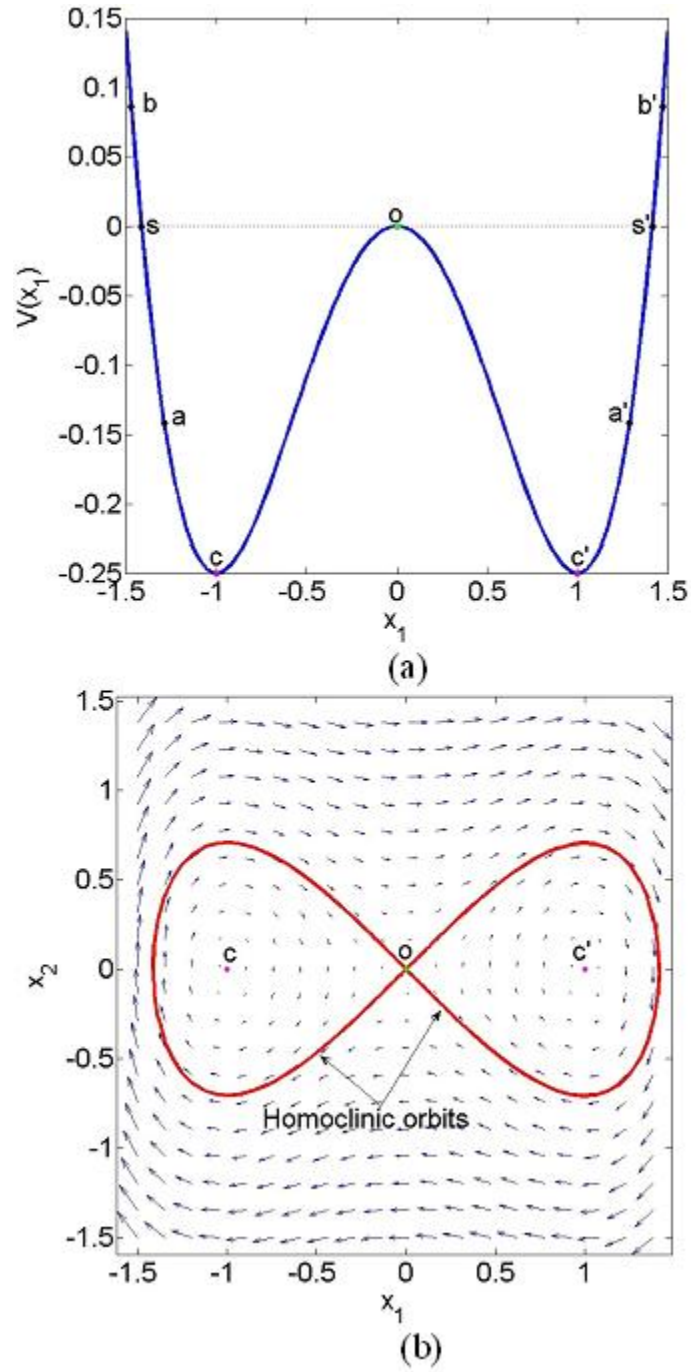


Figure 1.1. (a) Potential wells of the Duffing oscillator. (b) Vector fields and the homoclinic orbits of the hyperbolic point o . The homoclinic orbit separates the motion of points starting below the dash line (e.g. points a or a') in (a) from those initially located above the line (e.g. points b or b').

times from certain basins of attractions are pressing for a given stochastic dynamical system. These questions lead us to find a way to determine basin boundaries of stochastic systems. Boltt et al. [2] developed a technique to compute phase space transport without a priori knowledge of the basin boundaries based on the Ulam's method coupled with graph theory. In a different approach, recent efforts have focused to identify the number and location of *almost invariant* sets; those subsets of a state space where trajectories tend to stay for comparatively long period of time before they escape to other regions. Particularly, the *Set oriented* method was developed to numerically study the problem of identifying the almost invariant regions [7, 8, 9, 10]. In this language, we can say that a basin in a multi-stable system subject to stochastic noise can be considered as an almost invariant set.

Other special cases of Eq. (1.1) are those of periodic and quasi-periodic vector fields. These types of problem can be studied by using the Poincaré map. The transport and mixing is characterized by the *lobe* dynamics and its relationship to flux. In such an analysis, the phase space is divided into regions with boundaries consisting of segments of stable and unstable manifolds of hyperbolic periodic points and the crossing of the boundaries between these regions are allowed only via the *turnstile*s defined by the lobe dynamics. These lobes are the result of the homoclinic or heteroclinic (transverse) intersection of the stable and unstable manifolds of hyperbolic periodic points; regularly, it is called the homoclinic (or heteroclinic) tangle. Precise definitions of lobes, boundaries, and turnstiles are introduced by Mackay, Meiss, and Percival [11]. In addition, when the system are time-periodic perturbation of two-dimensional Hamiltonian systems, there exists a global perturbation method, Melnikov's method, that provides necessary conditions for an occurrence of the intersection of the stable and unstable manifold of such systems [12, 13]

Generally, dynamical systems in practice are aperiodic. Hence, recent research trends emphasize analysis of the phase space transport of a general time-dependent dynamical system Eq. (1.1). Several techniques and criteria have been developed to approximate time-varying geometrical structures analogous to those separatrices of autonomous systems, which prevent the transitions of particles between regions with different dynamics. The notion of stable and unstable manifolds for a general time-dependent system is that of an

exponential dichotomy, which is used to define the hyperbolic trajectory (see Wiggins [13]). With this notion in hand one can locate hyperbolic trajectories of certain dynamical systems by the persistence theorem. That is, if $F(x, t)$ in Eq. 1.1 is in the form of a deterministic time-dependent perturbation of an autonomous vector field;

$$F(x, t) = f(x) + \epsilon g(x, t), \quad (1.2)$$

the hyperbolic fixed point of the vector field $f(x)$ becomes a time varying hyperbolic trajectory with (local) stable and unstable manifolds “attached” along the trajectory. Nevertheless, such techniques require a knowledge of vector field for all time whereas most practical vector fields are known only in a finite-time interval. One approach to circumvent this problem is to use the so-called *Lagrangian coherent structure* (LCS) developed in a series of papers by Haller and coworkers [14, 15, 16]. Instead of directly detecting hyperbolic fixed points at frozen time slices, Haller and coworkers defined the finite-time hyperbolic structure, referred to as the LCS, via the local maximum of the *finite-time Lyapunov exponent* (FTLE) field. Other works by Shadden [17, 18] define the LCS as the second-derivative ridge of the FTLE field and proved that particle fluxes crossing the ridge are small and approach zero for a sufficiently large integrating time.

1.2 Purposes

As seen from the preceding section, it is important to detect the geometrical structures in a phase space of a general time-dependent vector field that minimally allow the transitions between regions separated by this structure such as separatrices and LCSs. In many applications of science such as meteorology, climatology, oceanography, fluid dynamic experiments, etc., camera and satellite platforms provide a large amount of information of continuous time spatial-temporal data in a form of a time-series data or an image sequence. In the latter case, if we would like to study a dynamic of a data observed in the form of video, the first task we have to do is to approximate the dynamical model from the image sequence since a sequence of images does not provide a direct access to the trajectory of a point in phase space. Instead, Successive images only describe an evolution in time of a

distribution of brightness of image pixels.

For this reason, a primary interest of this thesis is to reconstruct a velocity field from a two-dimensional image sequence that transforms the intensity pattern from one image to the next image in a sequence, which is called *optical flow* computation [19]. The problem of determining the optical flow has been intensively studied in the last two decades in many areas of computer vision to perform motion detection and tracking, object segmentation, time-to-collision estimation, etc. Recently, the analysis of the optical flow has been applied to many other science areas such as meteorology, oceanography, climatology, where image sequences from satellite platforms are the main source of their information [20, 21, 22, 23, 24, 25, 26, 27]. However, the application of this optical flow analysis has not yet been demonstrated in the area of the measurable dynamical system, in particular, to approximate the Frobenius-Perron transfer operator from a sequential image data.

In this thesis we assume that the desired velocity field between two consecutive image frames is autonomous throughout the image region and so the trajectory of a point in the image region is governed by the following equation:

$$\begin{aligned}\frac{dx}{dt} &= v_1(x, y) \\ \frac{dy}{dt} &= v_2(x, y),\end{aligned}\tag{1.3}$$

where $(x, y) \in \mathbb{M}$ for some compact $\mathbb{M} \subset \mathbb{R}^2$ and $v_1(x, y)$, $v_2(x, y)$ are two unknown velocity fields to be approximated from a sequence of images, which provides to us only a temporal variation of the brightness pattern. We describe this temporal variation in a framework of the Frobenius-Perron operator in chapter 3 and its implementation in chapter 6.

We summarize here the goals of this thesis:

1. Data Assimilation and Dynamics Recovering: First, we offer a practical technique to reconstruct a dynamical model from movies (image sequences) from various areas of science such as fluid experiments, satellite data, climatology, etc. Also, we provide a mathematical formalism to validate our technique to estimate vector fields from an image sequence.

2. Global Analysis of Autonomous Systems: We explore and apply certain graph theory methods to identify almost invariant sets, when they exist, in a phase space defined in the sense that the particles that initially belong to a particular set will stay, with a high probability, in that set for a comparatively long time before escaping to other regions. These almost invariant sets correspond to the situation of basin-hopping in a multi-stable system under a small stochastic perturbation. We perform several numerical experiments to investigate the validity of the method. However, the notion of the almost invariant set for a general time-dependent vector field is itself at odds. The graph method is therefore restricted only to autonomous systems both with and without stochastic perturbations.

3. Global Analysis of Non-Autonomous Systems: For a general time-dependent vector field, we employ the concept of the finite-time Lyapunov exponent (FTLE) to define the time-varying Lagrangian coherent structure, which prevents a flux from traversing across the structure.

4. Real-world Modeling: We demonstrate some applications of the combination of the above tools to a study of transport and mixing for certain real-world problems.

1.3 Organization

The organization of this dissertation can be summarized as follows:

Chapter 2: Some basic terminologies and backgrounds in dynamical systems are reviewed.

Particularly, we will review the notions of the hyperbolicity and hyperbolic trajectory. Although they will not be used in our analysis, they can be regarded as alternative techniques to understand transport and mixing in dynamical system. Therefore, we also review some research literature related to the global analysis of dynamical systems using these notions and the view point of measurable dynamical systems.

Chapter 3 This chapter begins by reviewing the background of the Frobenius-Perron operator and its infinitesimal generator. We then explore the relationship between the evolution of density by the Frobenius-Perron operator and the motion estimation from

image sequences. This serves as an introduction to Chapter 6 where the implementation of motion estimation in the framework of the Frobenius-Perron operators is presented. In addition, this chapter shows the construction of the Ulam-Galerkin matrix (or transition matrix) as a finite-rank approximation of the Frobenius-Perron operator. Furthermore, we review the notion of the almost invariant set and its relation to the transition matrix.

Chapter 4 In this chapter, we introduce an application of a novel graph theory method to the global analysis of autonomous systems (with/without stochastic perturbation) known only through time-series data. In particular, we show how to use the concept of graph modularity to discover a phase space partition that reveals some meaningful global structures of a given time-series data. Several examples are presented as benchmarks to demonstrate that global structures can still exist under stochastic perturbation if the amplitude of noise is sufficiently small. Most of the results in this chapter are from my contribution to the paper by Santitissadeekorn and Boltt [28].

Chapter 5 This chapter reviews the concept of the Lagrangian coherent structure (LCS) and its computation based on the finite-time Lyapunov exponent (FTLE). We then employ this concept to transport and mixing analysis of some real-world mixing devices. The results in this chapter are published under collaboration of Santitissadeekorn, Bohl, and Boltt [29].

Chapter 6 In this chapter, we present a numerical implementation of the idea introduced in Chapter 3. We review some previous work in optical flow, which is traditionally used in the image processing community. We then point out the reasons for using the smoothness constraint in our work, instead of the discontinuity preserving constraints. Then we review a coarse-to-fine strategy and use it to deal with the problem of large motions. Finally, we present several examples to benchmark and contrast our results obtained from our model with the correlation-based PIV, which is widely used in fluid mechanics. Most materials and examples in this Chapter are from my contribution to the paper by Santitissadeekorn and Boltt [30].

Chapter 7 This chapter gives a summary of my contributions and concludes with open questions to point to future directions of research in global analysis of dynamical system and motion estimation of image sequences.

Chapter 2

Dynamical systems terminology and definitions

For completeness, we present in this chapter some standard dynamical system terminologies and concepts, which are needed in our research described in the subsequent chapters. We have made every effort to draw these definitions from several standard and well-accepted texts, respecting to the usual notations, but translated to the notations we will use consistently in this thesis. Some general references for the following materials are Guckenheimer and Holmes [31], Robinson [32], and Wiggins [13].

2.1 Linearization

In this section, we summarize the notion of derivative as a linear map, which we will use to study and classify the local behavior “near” the fixed or periodic points of nonlinear differential equations. Loosely speaking, the Hartman-Grobman theorem, see [31], shows that the local behavior near the hyperbolic fixed point, which will be defined in the next section, is “similar” to that of the associated linearized system.

Consider maps $f : U \subset \mathbb{R}^k \rightarrow \mathbb{R}^n$, where U is an open subset of \mathbb{R}^m . The first partial

derivative of f at a point p can be expressed in a $n \times k$ matrix form

$$Df_p = \left[\frac{\partial f_i}{\partial x_j} \right]. \quad (2.1)$$

We consider this matrix as a linear map from \mathbb{R}^m to \mathbb{R}^n , that is $Df_p \in L(\mathbb{R}^m, \mathbb{R}^n)$, and recall that $L(\mathbb{R}^m, \mathbb{R}^n)$ is isomorphic to \mathbb{R}^{mn} . With this in mind we may define the (Fréchet) derivative in the following way.

Definition 2.1. A map $f : U \subset \mathbb{R}^m \rightarrow \mathbb{R}^n$ is said to be Fréchet differentiable at $x \in \mathbb{R}^m$ if and only if there exists $Df_x \in L(\mathbb{R}^m, \mathbb{R}^n)$, called the Fréchet derivative of f at x , for which

$$f(x+h) = f(x) + Df_x h + o(\|h\|) \quad \text{as } \|h\| \rightarrow 0. \quad (2.2)$$

The derivative is called *continuous* provided the map $Df : U \rightarrow L(\mathbb{R}^m, \mathbb{R}^n)$ is continuous with respect to the Euclidean norm on the domain and the operator norm on $L(\mathbb{R}^m, \mathbb{R}^n)$. If the first partial derivatives at all points in U exist and are continuous, the derivative is also continuous and the map f is called *continuously differentiable* or $f \in C^1$. Higher order derivative can be defined recursively, e.g.,

$$Df(x+k) = Df(x) + D^2 f_x k + o(\|k\|) \quad (2.3)$$

defines the second derivative, $D^2 f_x \in L(\mathbb{R}^m, L(\mathbb{R}^m, \mathbb{R}^n))$. We again note that $L(\mathbb{R}^m, \mathbb{R}^n)$ is isomorphic to bilinear maps in $L(\mathbb{R}^m \times \mathbb{R}^m, \mathbb{R}^n)$. The mapping $(h, k) \mapsto (D^2 f_x k)h$ is bounded and bilinear from $\mathbb{R}^m \times \mathbb{R}^m$ into \mathbb{R}^n . If $v, w \in \mathbb{R}^m$ are expressed in terms of the standard basis as $v = \sum_i v_i \mathbf{e}^i$ and $w = \sum_i w_i \mathbf{e}^i$, then we have

$$(D^2 f_x v)w = \sum_{i,j} \left(\frac{\partial^2 f}{\partial x_i \partial x_j} \right)_x v_i w_j \quad (2.4)$$

Since the mixed cross derivatives are equal, $D^2 f_x$ is a symmetric bilinear form. Generally, if all the derivatives of order $1 \leq j \leq r$ exist and are continuous, then f is said to be r -continuously differentiable, or $f \in C^r$. If $f : \mathbb{R}^m \rightarrow \mathbb{R}^m$ is C^r , Df_x is a linear isomorphism at each point $x \in \mathbb{R}^k$, and f is 1-to-1 and onto, then f is called a C^r -*diffeomorphism*.

Now, consider a dynamical system

$$\dot{x} = f(x, t) \quad (2.5)$$

and assume that $f : M \rightarrow M$ is at least C^2 in the domain M of x and C^1 in time, t . Then, it is possible to linearize a dynamical system about a point $\tilde{x} \in M$. Suppose an initial point \tilde{x} is advected to $x(t) = \tilde{x} + \delta x(t)$ after time t by a vector field. Then, the linearization of $\dot{x} = f(x, t)$ is

$$\frac{d\delta x}{dt} = Df_{\tilde{x}}\delta x + o(\|\delta x\|). \quad (2.6)$$

Note that $Df_{\tilde{x}}$ is generally time-dependent along the trajectory and it is also non-singular at each time t by the existence and uniqueness of the solution. If $\gamma(t)$ is a trajectory of the vector field, we write the associated linearized system of $\dot{x} = f(x, t)$ as

$$\dot{\xi} = Df(\gamma(t), t)\xi. \quad (2.7)$$

In the next section, we review some main concepts of the hyperbolicity of a map. The definitions and theorems still hold for a flow $\gamma(t)$ of an autonomous vector field with small differences (See Robinson [32]). In such a case, the coefficient function Df in (2.7) is time-independent. However, the notion of hyperbolicity of a nonautonomous system becomes more complicated and will be discussed later in this chapter.

2.2 Hyperbolicity

The intention of this section is to provide a sufficient background of hyperbolic dynamics. The importance of hyperbolicity in applied dynamical system is that it gives a complex and chaotic dynamics despite its mathematical simplicity. We will limit ourselves to the notion of uniform hyperbolicity but note that a thorough treatment of hyperbolic dynamics and non-uniform hyperbolicity and its relation to ergodic theory can be found in Pollicott and Yuri [33].

In this section, we are interested in the map $f : M \rightarrow M$ where M is a compact n -dimensional Riemannian C^∞ manifold. Here, the Riemannian manifold can be viewed as a differentiable topological space with an inner product on tangent spaces which is locally isomorphic to a Euclidean metric space.

Definition 2.2. (Anosov diffeomorphism [13]) A diffeomorphism $f : M \rightarrow M$ is *Anosov* if there exist constants $c > 0, 0 < \lambda < 1$ and a decomposition of continuous tangent space $T_x M = E_x^s \oplus E_x^u$ at each $x \in M$ satisfying the following properties:

(a) E_x^s and E_x^u are invariant with respect to the derivative Df_x :

$$\begin{aligned} Df_x E_x^s &= E_{f(x)}^s \\ Df_x E_x^u &= E_{f(x)}^u \end{aligned} \tag{2.8}$$

(b) The expansion rate under the forward and backward iteration are uniformly bounded:

$$\begin{aligned} \|Df_x^n v_s\| &= c\lambda^n \|v_s\| & \text{for } v_s \in E_x^s \\ \|Df_x^{-n} v_u\| &= c\lambda^n \|v_u\| & \text{for } v_u \in E_x^u \end{aligned} \tag{2.9}$$

Since the constants c and λ given in the above definition are the same for each point $x \in M$, Anosov systems are also called *uniformly hyperbolic* systems and these two terminologies will be used interchangeably in the course of this thesis. In the following, we call a closed set $A \subset M$ an *invariant* set if $f(A) \subseteq A$. If f is uniformly hyperbolic at each point $x \in A$, then A is called a *hyperbolic invariant* set. If $A = \{\bar{x}\}$, that is $f(\bar{x}) = \bar{x}$, then \bar{x} is called a *hyperbolic fixed point*. Hyperbolic periodic points can also be defined in a similar manner. The importance of hyperbolicity is that the dynamics in a small neighborhood of the hyperbolic points exhibits both expanding and contracting behavior and the dynamics are homeomorphic to the associated linearized system, which is stated more precisely by the Hartman-Grobmann theorem [13, 32, 31].

Definition 2.3. (local stable and unstable manifold [32]) Let $A \subset M$ be a hyperbolic invariant set, $x \in A$ and a neighborhood $N(x, \epsilon)$ for $\epsilon > 0$ sufficiently small. The local stable manifold, $W_\epsilon^s(x)$, and the local unstable manifold, $W_\epsilon^u(x)$, are given by

$$\begin{aligned} W_\epsilon^s(x) &= \{y \in N(x, \epsilon) \mid d(f^n(y), x) \rightarrow 0 \text{ as } n \rightarrow \infty\} \\ W_\epsilon^u(x) &= \{y \in N(x, \epsilon) \mid d(f^{-n}(y), x) \rightarrow 0 \text{ as } n \rightarrow \infty\} \end{aligned} \tag{2.10}$$

Note that the local unstable manifold can still be defined even if f is not invertible by abusing the notation $f^{-n}(x)$ as the preimage of x .

Theorem 2.1. (Stable manifold theorem [13]) Let $f : M \rightarrow M$, A be a hyperbolic invariant set with respect to f , and $x \in A$. Then, there exist local stable and unstable manifolds $W_\epsilon^s(x)$ and $W_\epsilon^u(x)$ that satisfy the following properties:

- (i) $W_\epsilon^s(x)$ and $W_\epsilon^u(x)$ are differentiable submanifolds and depend continuously on x .
- (ii) $TW_\epsilon^s(x) = E_x^s$ and $TW_\epsilon^u(x) = E_x^u$, i.e., $W_\epsilon^s(x)$ and $W_\epsilon^u(x)$ are of the same dimension and tangent to the subspace E_x^s and E_x^u , respectively.

The previous result can be extended to the entire domain M by the well-known Hadamard-Perron Manifold Theorem (See Wiggins [13]). Therefore, we can discuss the local stable and unstable manifold at each point in the domain of an Anosov diffeomorphism. Now, we can determine the global stable and unstable manifolds by taking union of backward and forward iterates of local stable and unstable manifolds.

Definition 2.4. (global stable and unstable manifolds [32]) The global stable and unstable manifolds at each $x \in M$, $W^s(x)$ and $W^u(x)$ respectively, are given by

$$\begin{aligned}
W^s(x) &\equiv \{y \in M \mid d(f^j(y), f^j(x)) \rightarrow 0 \text{ as } j \rightarrow \infty\} \\
W^s(x) &= \bigcup_{n \geq 0} f^{-n}(W_\epsilon^s(f^n(x))) \quad \text{and} \\
W^u(x) &\equiv \{y \in M \mid d(f^{-j}(y), f^{-j}(x)) \rightarrow 0 \text{ as } j \rightarrow \infty\} \\
W^u(x) &= \bigcup_{n \geq 0} f^n(W_\epsilon^u(f^{-n}(x))).
\end{aligned} \tag{2.11}$$

It is clear from the above definition that the stable and unstable manifolds are invariant since they are a union of trajectories. Also, it is worth mentioning that a stable (unstable) manifold cannot neither cross itself nor other stable (unstable) manifolds of another point; otherwise, the crossing point must be iterated to two different points by definition but this is not permitted.

The following definitions are some types of specific orbits that are often mentioned in the course of this thesis and indeed important to transport and mixing since they usually form the transport barrier between regions of qualitatively different dynamics.

Definition 2.5. (Homoclinic orbit [32]) Let p be a hyperbolic periodic point of period n

for a diffeomorphism f and $\mathcal{O}(p)$ be the orbit of p . Let

$$W^{s,u}(\mathcal{O}(p)) = \bigcup_{j=0}^{n-1} W^{s,u}(f^j(p)) \quad \text{and} \quad (2.12)$$

$$\hat{W}^{s,u}(\mathcal{O}(p)) = W^{s,u}(\mathcal{O}(p)) \setminus \mathcal{O}(p).$$

A point $q \in \hat{W}^s(\mathcal{O}(p)) \cap \hat{W}^u(\mathcal{O}(p))$, if it exists, is called a *homoclinic point* for p . The orbit of q is then called the *homoclinic orbit* for p , see Figure 2.1.

It follows from the above definition that a point in the homoclinic orbit will asymptotically approach the same hyperbolic periodic point p both in forward and backward time. The Smale-Birkhoff homoclinic theorem proves that the existence of a transverse homoclinic point induces a Smale horseshoe in the homoclinic tangle [13, 32, 34].

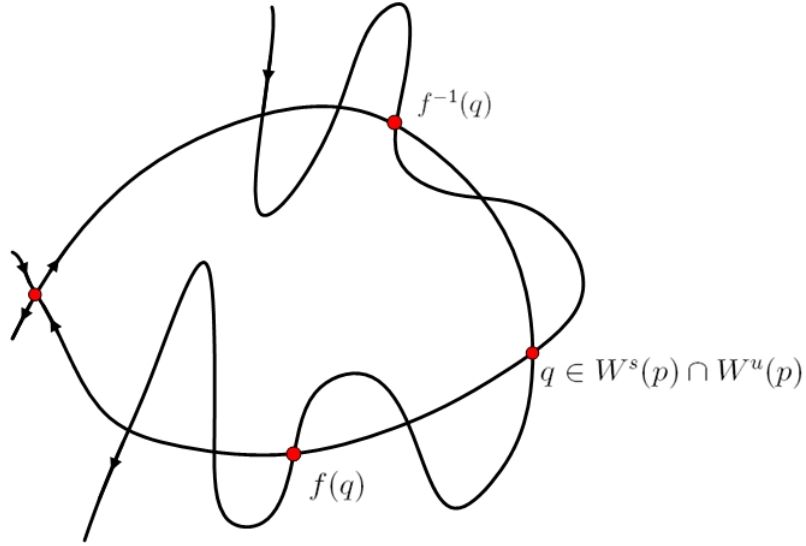


Figure 2.1. An example of a homoclinic orbit of a hyperbolic fixed point p . Here q is the homoclinic point and the map f is assumed to be an orientation-preserving map, in which there must be at least one homoclinic point between f and $f(q)$

Definition 2.6. (Heteroclinic orbit [32]) Let $\{p_i\}_{i=1 \dots n}$ be a collection of hyperbolic periodic orbits for a diffeomorphism f . A point $q \in \hat{W}^s(\mathcal{O}(p_i)) \cap \hat{W}^u(\mathcal{O}(p_j))$ for some $i \neq j$, if it exists, is called a *heteroclinic point*. Similarly, the orbit of q is called the *heteroclinic orbit*.

Therefore, a point in a heteroclinic orbit asymptotically approaches a hyperbolic peri-

odic orbit forward in time and a different periodic orbit backward in time, see Figure 2.2.

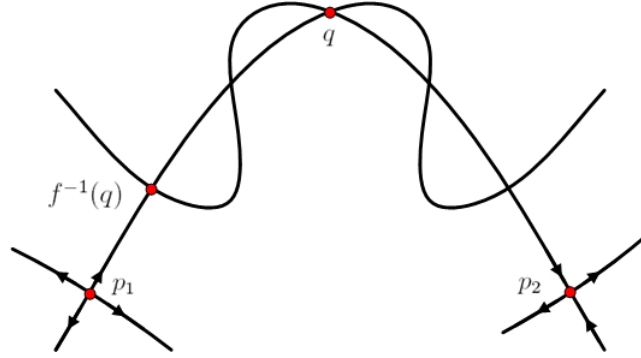


Figure 2.2. An example of a heteroclinic orbit connecting two hyperbolic fixed points p_1 and p_2 . Here q is the heteroclinic point and the map f is assumed to be an orientation-preserving map as in Figure 2.1

2.3 Hyperbolicity: Nonautonomous vector fields

In the preceding section, the linearization of a fixed point or an invariant set was used to determine the hyperbolicity property. However, for a general time-dependent system, we have to define the notion of the hyperbolicity for the time-dependent trajectory for which the linearization along the trajectory does not give rise to a constant matrix associated with the linearized vector field. The standard notion to characterize the hyperbolicity of a time-dependent trajectory is that of the exponential dichotomy described below. We also summarize below the concept of stable and unstable manifolds of the hyperbolic trajectory and their significance and relation to the Lagrangian coherent structure (LCS), which will be discussed in Chapter 5.

As in the case of an autonomous system, we begin by defining an appropriate decomposition into stable and unstable subspaces along a trajectory of a nonautonomous system. This is a general extension of the hyperbolicity of the autonomous vector field. Thus, we first consider the linearized vector field of the form (2.7) and keep in mind that the coefficient matrix is now time-dependent.

Definition 2.7. (Exponential Dichotomy [13]) Consider a time-dependent linear differential

equation

$$\dot{\xi} = A(t)\xi \quad \xi \in \mathbb{R}^n, \quad (2.13)$$

where $A(t) \in \mathbb{R}^{n \times n}$ is a time-dependent coefficient and continuous in time, $t \in \mathbb{R}$. Let $X(t) \in \mathbb{R}^{n \times n}$ be the solution matrix such that $\xi(t) = X(t)\xi(0)$ and $X(0) = I$. Then (2.13) is said to possess an *exponential dichotomy* if there exists a projection operator P , $P^2 = P$, and constants $K_1, K_2, \lambda_1, \lambda_2 > 0$ such that

$$\begin{aligned} \|X(t)PX^{-1}(t)\| &\leq K_1 \exp(-\lambda_1(t - \tau)), & t \geq \tau \\ \|X(t)(I - P)X^{-1}(t)\| &\leq K_2 \exp(\lambda_2(t - \tau)), & t \leq \tau \end{aligned} \quad (2.14)$$

Examples illustrating a comparison between the hyperbolicity of linear autonomous systems and the exponential dichotomy of nonautonomous system can be found in Wiggins [13].

A generalization of a hyperbolic fixed point in autonomous systems is that of the hyperbolic trajectory of a time-dependent vector field, which can be defined via the exponential dichotomy.

Definition 2.8. (Hyperbolic trajectory [13]) Let $\gamma(t)$ be a trajectory of the vector field $\dot{x} = f(x, t)$. Then $\gamma(t)$ is called a hyperbolic trajectory if the associated linearized system $\dot{\xi} = D_x(f(\gamma(t), t))\xi$ has an exponential dichotomy.

The geometry of a hyperbolic trajectory can be understood in the extended phase space:

$$\mathcal{E} \equiv \{(x, t) \in \mathbb{R}^n \times \mathbb{R} | x \in U\}. \quad (2.15)$$

The nonautonomous vector field can then be viewed as an autonomous one as follows

$$\begin{aligned} \dot{x} &= f(x, t) \\ \dot{t} &= 1. \end{aligned} \quad (2.16)$$

In the extended phase space \mathcal{E} , we denote the hyperbolic trajectory by $\Gamma(t) = (\gamma(t), t)$ and define a *time slice* of the extended phase space \mathcal{E} by

$$\Sigma_\tau \equiv \{(x, t) \in \mathcal{E} | t = \tau\}. \quad (2.17)$$

Then the condition (2.14) requires that there exists a projection onto a subspace of Σ_τ , called $E^s(\tau)$, so that a trajectory of an initial vector at time $t = 0$ projected onto $E^s(\tau)$ by the associated linearized vector field (2.13) will have to decay to zero at an exponential rate, determined by λ_1 as $t \rightarrow \infty$. Similarly, the complimentary projection $(I - P)$ onto a subspace of Σ_τ , called $E^u(\tau)$, exists so that the initial vector projected onto $E^u(\tau)$ will decay at exponential rate, determined by λ_2 as $t \rightarrow -\infty$. Moreover, $\Sigma_\tau = E^s(\tau) \oplus E^u(\tau)$. Therefore, the exponential dichotomy guarantees the existence of a stable (unstable) subspace for which the initial conditions on these spaces asymptotically approach a hyperbolic trajectory at an exponential rate at a forward (backward) time, see Figure 2.3. This is analogous to that of the autonomous vector field where the initial conditions on the stable (unstable) manifold converge to some critical points, e.g., fixed points or periodic orbits, forward (backward) in time. However, the difference is that the stable (unstable) manifolds for an autonomous system are time-independent whereas those of a nonautonomous case vary in time. In particular, for a 2-D vector field, the stable (unstable) manifolds in the autonomous case is an invariant curve, but the stable (unstable) manifold in the nonautonomous case become a time-varying curve or an invariant surface in the extended space E . With the

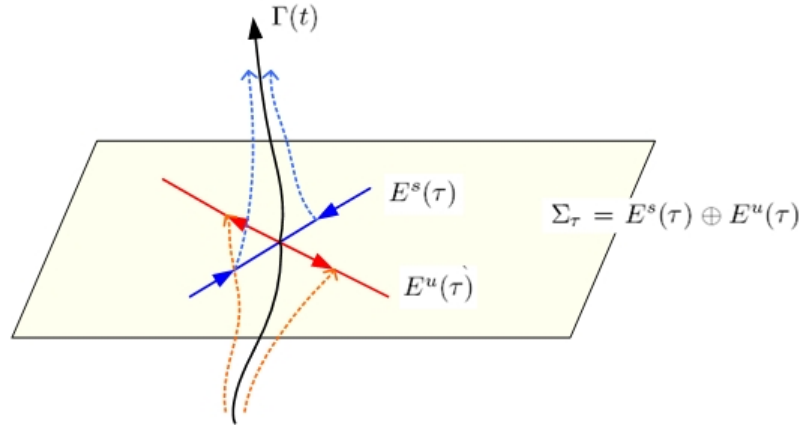


Figure 2.3. The time slice $\Sigma_\tau = E^s(\tau) \oplus E^u(\tau)$. A trajectory of an initial segment projected onto $E^s(\tau)$ by the associated linearized vector field (2.13) approaches the hyperbolic trajectory $\Gamma(t)$ at an exponential rate as $t \rightarrow \infty$. Similarly, the initial vector projected onto $E^u(\tau)$ approaches $\Gamma(t)$ at an exponential rate as $t \rightarrow -\infty$.

geometrical explanation of the exponential dichotomy in mind, we can now state the theorem that describes the existence of local stable and unstable manifolds of a hyperbolic trajectory.

Theorem 2.2. (Local stable and unstable manifolds [13]) Let $D_\rho(\tau)\Sigma_\tau$ denote the ball of radius ρ centered at $\gamma(\tau)$ and define the tubular neighborhood of $\Gamma(t)$ in \mathcal{E} by $\mathcal{N}_\rho(\Gamma(t)) \equiv \cup_{\tau \in \mathbb{R}} (D_\rho(\tau), \tau)$. There exists $(k+1)$ dimensional C^r manifold $W_{loc}^s(\Gamma(t)) \subset \mathcal{E}$, and an $(n-k+1)$ dimensional C^r manifold $W_{loc}^u(\Gamma(t)) \subset \mathcal{E}$ and ρ_0 sufficiently small such that for $\rho \in (0, \rho_0)$

(i) $W_{loc}^s(\Gamma(t))$, the local stable manifold of $\Gamma(t)$, is invariant under the forward time evolution generated by (2.16); $W_{loc}^u(\Gamma(t))$, the local unstable manifold of $\Gamma(t)$, is invariant under the backward time evolution generated by (2.16).

(ii) $W_{loc}^s(\Gamma(t))$ and $W_{loc}^u(\Gamma(t))$ intersect along $\Gamma(t)$, and the angle between the manifolds is bounded away from zero uniformly for all $t \in \mathbb{R}$.

(iii) Every trajectory on $W_{loc}^u(\Gamma(t))$ can be continued to the boundary of $\mathcal{N}_\rho(\Gamma(t))$ backward in time, and every trajectory on $W_{loc}^s(\Gamma(t))$ can be continued to the boundary of $\mathcal{N}_\rho(\Gamma(t))$ forward in time.

(iv) Trajectories starting on $W_{loc}^s(\Gamma(t))$ at time $t = \tau$ approach $\Gamma(t)$ at an exponential rate $e^{-\lambda'(t-\tau)}$ as $t \rightarrow \infty$ and trajectories starting on $W_{loc}^u(\Gamma(t))$ at time $t = \tau$ approach $\Gamma(t)$ at an exponential rate $e^{-\lambda'|t-\tau|}$ as $t \rightarrow \infty$, for some constant $\lambda' > 0$.

(v) Any trajectory in $\mathcal{N}_\rho(\Gamma(t))$ not on either $W_{loc}^s(\Gamma(t))$ or $W_{loc}^u(\Gamma(t))$ will leave $\mathcal{N}_\rho(\Gamma(t))$ in both forward and backward in time.

The above theorem suggests a way to determine global stable and unstable manifolds of a hyperbolic trajectory [35, 36, 37]. In short, it allows us to extend an initial segment of a local unstable manifold, $W^u(\gamma(t_1))$, of a hyperbolic trajectory $\gamma(t_1)$ on the time slice Σ_{t_1} for some $t_1 < \tau$ by evolving this segment forward in time. Similarly, evolving the initial segment of a stable manifold $W^s(\gamma(t_2))$ of a hyperbolic trajectory $\gamma(t_2)$ on the time slice Σ_{t_2} for some $t_2 > \tau$ backward in time yields a global stable manifold. Figure 2.4 illustrates this concept. One rigorous method to determine a meaningful hyperbolic trajectory, called *distinguished hyperbolic trajectory* (DHT), and its stable and unstable manifolds will not be discussed in this thesis, but it can be found in [35, 36, 37]. Since a dynamical system can possess infinitely many hyperbolic trajectories, e.g., all trajectories in the stable or unstable manifolds, the bounded hyperbolic trajectory with the special properties is used as the

reference hyperbolic trajectory for “growing” the stable and unstable manifolds [35, 36]. It is demanded in [38] that a DHT has to remain in a bounded region for all time and that there exists a neighborhood \mathcal{N} of the DHT such that all other hyperbolic trajectories within \mathcal{N} have to leave \mathcal{N} in a finite time, either by forward or backward time manner. If one attempts to grow the stable or unstable manifold from a hyperbolic trajectory different from the DHT, it is possible to observe a “drifting” phenomenon in the flow due to a slow expansion rate as demonstrated in a number of numerical examples in [37].

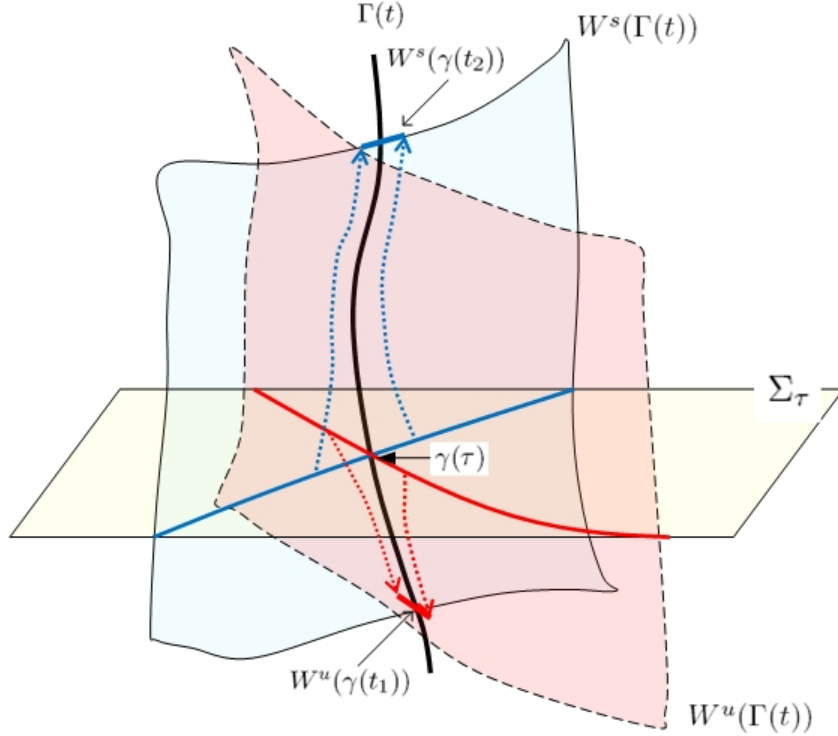


Figure 2.4. The forward-time evolution of a segment of $W^u(\gamma(t_1))$ from the time slice Σ_{t_1} to the time slice Σ_τ yields the unstable manifold of $W^u(\gamma(\tau))$. Similarly, the backward-time evolution of a small segment of $W^s(\gamma(t_2))$ from the time slice Σ_{t_2} to the time slice Σ_τ yields the stable manifold $W^s(\gamma(\tau))$.

A rigorous method to determine the DHT in general is still an active research area. In most cases, DHT has to be selected with a careful observation and knowledge of the system. In another approach, Haller and Poje [14] determined a finite-time uniformly hyperbolic trajectory based on the local maxima of the time duration in which an initial point remains hyperbolic. The contour plot of such hyperbolicity time for each grid point of initial conditions reveals the uniformly hyperbolic invariant set, which can be used as

a “seed” to gradually construct the corresponding global stable and unstable manifolds by a traditional technique, e.g., the straddling technique of You, Kostelich and Yorke [39]. Nevertheless, this technique demands the deformation of the uniformly hyperbolic trajectory to be at a slower speed than the speed of individual particles.

In a recent approach, following the work in a series of papers by Haller [14, 15, 16], the search of DHT or hyperbolic invariant sets can be circumvented by computing the finite-time Lyapunov exponent field [17, 18], which directly captures the region of a large expansion rate, see Chapter 5. These regions corresponds to the stable (unstable) manifold of the hyperbolic trajectory when computing in forward (backward) time manner. This technique is based on the fact that the stable (unstable) manifold of a hyperbolic trajectory is repelling (attracting) in the normal direction to the manifold and hence initial points straddling the stable (unstable) manifold will eventually be separated from (attracted to) each other at some exponential rate. However, this approach gives us merely a scalar field indicating the expansion rate of the grid point in a domain of a nonautonomous system whereas the preceding techniques yield the parameterized curve representing invariant manifolds. Nevertheless, the second-derivative ridge of the finite-time Lyapunov exponent field can be used as a curve that represents the global stable (or unstable) manifold of a nonautonomous dynamical system by using our proposed ridge detection technique (See Appendix A).

Chapter 3

Frobenius-Perron Operator and Infinitesimal Generator

In many areas of science, main sources of data are given in a form of image sequences or movies, which capture the underlying dynamics. To understand the transport behavior of these dynamical systems, it is mandatory to extract the velocity fields from image sequences. Thus, we apply the well-known optical flow technique to estimate the velocity fields from image sequences and we verify the validity of the optical flow model in the framework of the Frobenius-Perron operator. Therefore, this chapter is devoted to a review of some classical tools and techniques for studying the evolution of density under a continuous time process. Specifically, we concentrate on the concept of the Frobenius-Perron operator and its infinitesimal generator, which is a necessary prelude to the optical flow problem in a subsequent chapter.

3.1 Frobenius-Perron Operator

We define a continuous process in a topological Hausdorff space \mathbb{X} by a family of mappings

$$S_t : \mathbb{X} \rightarrow \mathbb{X}, \quad t \geq 0. \tag{3.1}$$

In particular, we focus on a continuous time process generated by an autonomous d -dimensional system of differential equations

$$\frac{dx}{dt} = F(x) \quad (3.2)$$

or

$$\frac{dx_i}{dt} = F_i(x), \quad i = 1, \dots, d, \quad (3.3)$$

where $x = (x_1, \dots, x_d)$ and $F : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is sufficiently smooth to guarantee the existence and uniqueness of solutions. Then, Eq. (3.2) defines a transformation $S_t(x^0) = x(t)$ where $x(t)$ is the solution of Eq. (3.2) at time t starting from the initial condition x^0 at time 0.

Definition 3.1. A *semidynamical system* $\{S_t\}_{t \geq 0}$ on \mathbb{X} is a family of transformations $S_t : \mathbb{X} \rightarrow \mathbb{X}$, $t \in \mathbb{R}$, satisfying

- (a) $S_0(x) = x$ for all $x \in \mathbb{X}$;
- (b) $S_t(S_{t'}(x)) = S_{t+t'}(x)$ for all $x \in \mathbb{X}$ with $t, t' \in \mathbb{R}^+$; and
- (c) The mapping $(x, t) \rightarrow S_t(x)$ from $\mathbb{X} \times \mathbb{R}^+$ into \mathbb{X} is continuous

Remark 3.1. The reason to call $\{S_t\}_{t \geq 0}$ in the above definition a semidynamical system instead of a dynamical system is its invertibility. By restricting $t \in \mathbb{R}^+$ a family of transformation satisfied the above conditions possesses an Abelian semigroup properties and hence it may be called a **semigroup of transformations** as well. Also, the condition (b) can be interpreted to mean that dynamics of a semigroup of transformation are the same on time interval $[0, t']$ and $[t', t]$.

Now we present an important tool for studying a propagation of densities, known as the Frobenius-Perron operator. Let $(\mathbb{X}, \mathbb{A}, \mu)$ be a measure space, where \mathbb{A} is the σ -algebra of Borel sets. Assume that each transformation of a semidynamical system $\{S_t\}_{t \geq 0}$ is a non-singular measurable transformation on $(\mathbb{X}, \mathbb{A}, \mu)$, that is,

$$\mu(S_t^{-1}(A)) = 0 \quad \text{for each } A \in \mathbb{A} \text{ such that } \mu(A) = 0. \quad (3.4)$$

The **Frobenius Perron operator**, $P_t : L^1(\mathbb{X}) \rightarrow L^1(\mathbb{X})$ with respect to the transformation

S_t is defined [40] by the condition

$$\int_{S_t^{-1}(A)} f(x)dx = \int_A P_t f(x)dx, \quad \text{for each } A \in \mathbb{A}. \quad (3.5)$$

In the sequel, we will consider only the action of P_t on the space $D(\mathbb{X}, \mathbb{A}, \mu)$ defined by $D(\mathbb{X}, \mathbb{A}, \mu) = \{f \in L^1(\mathbb{X}, \mathbb{A}, \mu) : f \geq 0 \text{ and } \|f\| = 1\}$. A function $f(x) \in D(\mathbb{X}, \mathbb{A}, \mu)$ is called a probability density function (PDF). Thus $P_t f(x)$ gives us a new probability density function, which is unique a.e., and depend on the transformation S_t and the probability density function $f(x)$. It is simple to show that P_t satisfies the following properties:

$$\begin{aligned} \text{(a)} \quad & P_t(\lambda_1 f_1 + \lambda_2 f_2) = \lambda_1 P_t f_1 + \lambda_2 P_t f_2, \quad \text{for all } f_1, f_2 \in L^1, \lambda_1, \lambda_2 \in \mathbb{R}; \\ \text{(b)} \quad & P_t f \geq 0 \quad \text{if } f \geq 0; \\ \text{(c)} \quad & \int_X f(x)dx = \int_X P_t f(x)dx, \quad \text{for all } f \in L^1. \end{aligned} \quad (3.6)$$

By using the above properties, one may prove that P_t satisfies properties (a) and (b) of the definition of a semidynamical system.

Consider in addition to the differential equation (3.2) an observable $K_t f$ defined by

$$K_t f(x) = f(S_t(x)), \quad (3.7)$$

where $f, K_t f \in L^\infty(\mathbb{X})$. Hence, the operator $K_t : L^\infty \rightarrow L^\infty$ as defined in Eq. (3.7), for every $t \geq 0$, can be interpreted as the operator that evolves an observable $f(S_t(x))$ of a semidynamical system $\{S_t\}_{t \geq 0}$ given by Eq. (3.2). The operator K_t is known as the **Koopman operator** associated with the transformation S_t . It is easy to check that $\{K_t\}_{t \geq 0}$ is a semigroup. An important mathematical relation between the Frobenius-Perron and Koopman operators is that they are adjoints, that is,

$$\langle P_t f, g \rangle = \langle f, K_t g \rangle \quad (3.8)$$

for all $f \in L^1(\mathbb{X}), g \in L^\infty(\mathbb{X})$, and $t \geq 0$. Note that although the Frobenius-Perron operator preserves the L1-norm $\|\cdot\|_{L^1}$, the Koopman operator satisfies the inequality

$$\|K_t f\|_{L^\infty} \leq \|f\|_{L^\infty}. \quad (3.9)$$

In a subsequent section we will derive the infinitesimal operators of the Frobenius-Perron and Koopman operators. Therefore, we will concentrate only a semigroup of contracting linear operators defined in the following.

Definition 3.2. For $L = L^p$, $1 \leq p \leq \infty$, a family $\{T_t\}_{t \geq 0}$ of operators, $T_t : L \rightarrow L$ is called a **semigroup of contracting operators** if T_t has the following properties:

- (a) $T_t(\lambda_1 f_1 + \lambda_2 f_2) = \lambda_1 T_t f_1 + \lambda_2 T_t f_2$;
- (b) $\|T_t f\|_L \leq \|f\|_L$;
- (c) $T_0 f = f$; and
- (d) $T_{t+t'} f = T_t(T_{t'} f)$,

for $f, f_1, f_2 \in L$ and $\lambda_1, \lambda_2 \in \mathbb{R}$.

Also, the semigroup is called a **continuous semigroup**, if it satisfies

$$\lim_{t \rightarrow t_0} \|T_t f - T_{t_0} f\|_L = 0 \quad \text{for } f \in L, t_0 \geq 0. \quad (3.10)$$

3.2 Infinitesimal Operators

For a continuous semigroup of contractions $\{T_t\}_{t \geq 0}$ we define $D(A)$ by the set of all $f(x) \in L^p(\mathbb{X})$, $1 \leq p \leq \infty$, such that the limit

$$Af = \lim_{t \rightarrow 0} \frac{T_t f - f}{t}, \quad (3.11)$$

exists in the sense of strong convergence, that is,

$$\lim_{t \rightarrow 0} \left\| Af - \frac{T_t f - f}{t} \right\|_{L^p} = 0. \quad (3.12)$$

The operator $A : D(A) \rightarrow L$, for $L \equiv L^p$, is called the **infinitesimal generator**. Let $I(t) \equiv I(x, t) = T_t f(x)$ for fixed $f(x) \in D(A)$. The function $I'(t) \equiv I'(t)(x) \in L$ is said to be the **strong derivative** of $I(t)$ if it satisfies the following condition:

$$\lim_{t \rightarrow 0} \left\| I'(t) - \frac{I(t) - f(x)}{t} \right\|_L = 0. \quad (3.13)$$

In this sense, $I'(t)$ describes the derivative of the ensemble of points with respect to time t . The following theorem demonstrates an important relation between the infinitesimal generator and the strong derivative

Theorem 3.1. (See [40]) Let $\{T_t\}_{t \geq 0}$ be a continuous semigroup of contractions and $A : D(A) \rightarrow L$ the corresponding infinitesimal generator. Then for each fixed $f \in D(A)$ and $t \geq 0$, the function $I(t) = T_t f$ has the properties:

- (a) $I(t) \in D(A)$;
- (b) $I'(t)$ exists; and
- (c) $I(t)$ satisfies

$$I'(t) = AI(t) \quad (3.14)$$

with the initial condition $I(0) = f(x)$.

Example 3.1. Consider the family of operators $\{T_t\}_{t \geq 0}$ defined by

$$T_t f = f(x - ct) \quad \text{for } x \in \mathbb{R}, t \geq 0. \quad (3.15)$$

Under this operation a function $f(x)$ is translated in a positive direction of x by the length of ct . By using the “change of variable” formula we can see that the L^p norm is preserved for $1 \leq p \leq \infty$. The conditions (a),(c), and (d) of the definition 3.10 straightforwardly follow from Eq. (3.15). Thus, $\{T_t\}$ is a semigroup of contracting operators. It is slightly more complicated to show that $\{T_t\}$ is also continuous, see [40]. Now assume that f is bounded and at least $C^1(\mathbb{R})$, then by the mean value theorem we have

$$\frac{f(x - ct) - f(x)}{t} = -cf'(x - \theta ct), \quad (3.16)$$

where $|\theta| \leq 1$. This implies that

$$Af = \lim_{t \rightarrow 0} \frac{T_t f - f}{t} = -cf' \quad (3.17)$$

and the limit is strong in L^p , $1 \leq p \leq \infty$ if f has compact support. Therefore, it follows from Eq. (3.14) that at each point in the (x, t) –plane $u(t, x)$ satisfies the partial differential equation

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad \text{with } u(0, x) = f(x), \quad (3.18)$$

where $u(t, x)$ is in $D(A)$ for each fixed $t \geq 0$.

Remark 3.2. This example offers some insight of a relation between semigroup of continuous operators, strong derivative, and partial differential equations. It is well-known that the solution of Eq. (3.18) at time t is $T_t f$ as defined in Eq. (3.15).

Now consider a calculation of the infinitesimal generator of the semigroup of the Frobenius-Perron operators $\{P_t\}_{t \geq 0}$ as defined in Eq. (3.5) and the evolution of time-dependent density function $I(x, t)$ under an action of the Frobenius-Perron operator. This will be done indirectly through the adjoint property of the Frobenius-Perron and Koopman operators.

It follows directly from the definition of the Koopman operator Eq. (3.7) that the infinitesimal of the Koopman operator denoted by A_K is

$$A_K g(x) = \lim_{t \rightarrow 0} \frac{g(S_t(x^0)) - g(x^0)}{t} = \lim_{t \rightarrow 0} \frac{g(x(t)) - g(x^0)}{t}. \quad (3.19)$$

If g is continuously differentiable with compact support, we can apply the mean value theorem to obtain

$$A_K g(x) = \lim_{t \rightarrow 0} \sum_{i=1}^d \frac{\partial g}{\partial x_i}(x(\theta t)) x'_i(\theta t) = \sum_{i=1}^d \frac{\partial g}{\partial x_i} F_i(x), \quad (3.20)$$

where $0 < \theta < 1$. Combining equations (3.14) and (3.20) we conclude that the function

$$I(x, t) = K_t f(x) \quad (3.21)$$

satisfies the first-order partial differential equation

$$\frac{\partial I}{\partial t} - \sum_{i=1}^d \frac{\partial I}{\partial x_i} F_i(x) = 0. \quad (3.22)$$

This leads to a derivation of the infinitesimal generator for the semigroup of Frobenius-Perron operators generated by the family $\{S_t\}_{t \geq 0}$ defined in Eq. (3.1).

Let $f \in D(A_{FP})$ and $g \in D(A_K)$, where A_{FP} and A_K denote the infinitesimal operators of the semigroups of the Frobenius-Perron and Koopman operators, respectively. Using the adjoint property of the two operators it can be shown that

$$\langle (P_t f - f)/t, g \rangle = \langle f, (K_t g - g)/t \rangle. \quad (3.23)$$

Taking the limit as $t \rightarrow 0$ we obtain

$$\langle A_{FP} f, g \rangle = \langle f, A_K g \rangle. \quad (3.24)$$

Provided that g and f are continuously differentiable, and g has compact support we can show that [40]

$$\langle A_{FP}f, g \rangle = \langle - \sum_{i=1}^d \frac{\partial f F_i}{\partial x_i}, g \rangle. \quad (3.25)$$

Hence, we can conclude that

$$A_{FP}f = - \sum_{i=1}^d \frac{\partial f F_i}{\partial x_i}. \quad (3.26)$$

Again, we can use Eqs. (3.14) and (3.26) to conclude that the function

$$I(x, t) = P_t f(x) \quad (3.27)$$

satisfies the partial differential equation (continuity equation)

$$\frac{\partial I}{\partial t} + \sum_{i=1}^d \frac{\partial I F_i}{\partial x_i} = 0, \quad (3.28)$$

or symbolically,

$$\frac{\partial I}{\partial t} + F \cdot \nabla I + I \nabla \cdot F = 0. \quad (3.29)$$

Note that this equation is actually the same as the well-known continuity equation in fluid mechanics and many fields, but now it is a statement of conservation of density function of ensembles of trajectories. In the case when F is a divergence-free vector field, i.e. $\nabla \cdot F = 0$, Eq. (3.28) corresponds to incompressible fluids such as water and it can be simplified to

$$\frac{dI}{dt} = \frac{\partial I}{\partial t} + F \cdot \nabla I = 0. \quad (3.30)$$

A comparison of Eqs. (3.28) and (3.30) to the classical optical flow problem will be discussed in a subsequent chapter.

Example 3.2. Now consider a Duffing's oscillator in the domain $[0, 1] \times [0, 1]$ given by the following differential equation:

$$\begin{aligned} \frac{dx}{dt} &= 4y - 2 \\ \frac{dy}{dt} &= 4x - 2 - 8(2x - 1)^3. \end{aligned} \quad (3.31)$$

According to Eq. (3.28), given the initial density $u(x, 0) = f(x)$ the flow of the density ($u(x, t) = P_t f(x)$) under the Duffing's oscillator is given by

$$\frac{\partial u}{\partial t} + (4y - 2) \frac{\partial u}{\partial x} + (4x - 2 - 8(2x - 1)^3) \frac{\partial u}{\partial y} = 0. \quad (3.32)$$

A numerical simulation of the density propagation by Eq. (3.32) along with the vector field of Eq. (3.31) is shown in Figure 3.1 with an initial density $u(x, 0)$ illustrated in Figure 3.1(a). Notice that the density is stretched and folded due to the hyperbolic structure of the system.

3.3 Finite-rank approximation of the Frobenius-Perron operator

Now we would like to extend the well-established technique of the deterministic Frobenius-Perron operator in the preceding section to study the phase space transport of a stochastically perturbed dynamical system. The central theme of this section is to present the background of the Frobenius-Perron operator perturbed by any noise distribution with compact support and its finite-rank approximation, known as the Ulam-Galerkin matrix or the transition matrix. The results will allow us to develop a tool to characterize the mass transport between one phase space region to another.

Let F be a (non-singular) discrete time dynamical system acting on initial conditions $z \in M$ (say, $M \subset \mathbb{R}^n$). That is,

$$F : M \rightarrow M, \quad x \mapsto F(x). \quad (3.33)$$

The Frobenius-Perron operator $P_F : L^1(M) \rightarrow L^1(M)$ associated with the map F is defined by [40],

$$P_F[\rho(x)] = \int_M \delta(x - F(y)) \rho(y) dy, \quad (3.34)$$

where $\rho(x)$ is a probability density function (PDF) defined in $L^1(M)$. Thus $P_F[\rho(x)]$ gives us a new probability density function, which is unique a.e., and depends on F and $\rho(x)$. Alternatively, the Frobenius-Perron can also be defined by

$$P_F[\rho(x)] = \sum_{y \in F^{-1}(x)} \frac{\rho(y)}{|\det D_y F|}. \quad (3.35)$$

Note that for all measurable sets $A \subset M$, we have a continuity equation [40]

$$\int_{F^{-1}(A)} \rho(x) dx = \int_A P_F(\rho(x)) dx. \quad (3.36)$$

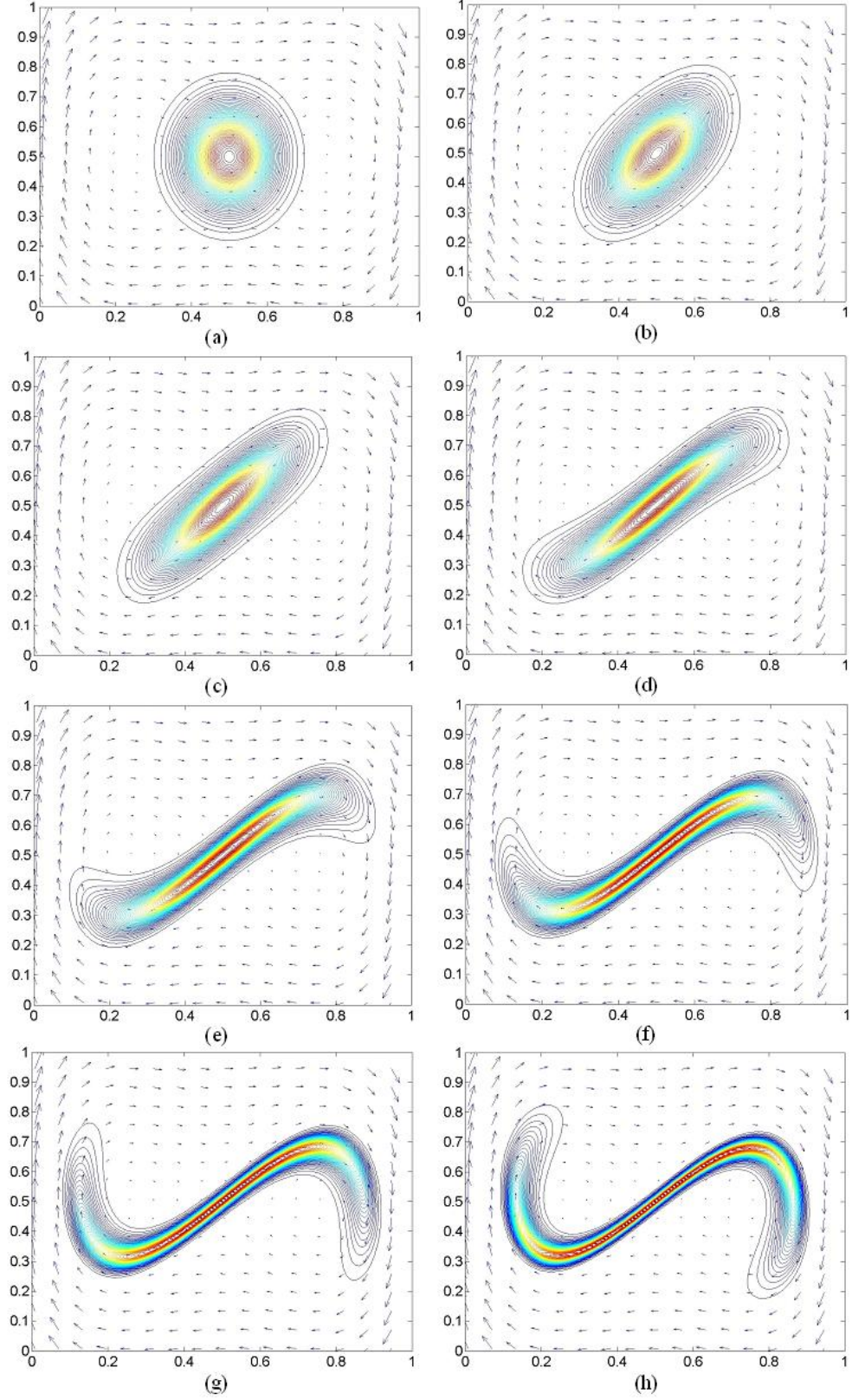


Figure 3.1. A simulation of the flow of a density function according to the continuity equation (3.32) with the velocity field given by Eq. (3.31). The time increment between each snapshot is $t = 0.08$.

Next, we consider a stochastic process defined by

$$x_{n+1} = \nu_n S(x_n) + T(x_n) \quad (3.37)$$

where ν_n are independent random vectors each having the same density g and a relatively small magnitude compared to the deterministic part S and T . Note that if we set $T \equiv 0$, we would have a process with a multiplicative perturbation, whereas when $S \equiv 1$, we have a process with an additive stochastic perturbation. Suppose the density of x_n given by ρ_n . We desire to show a relation of ρ_n and ρ_{n+1} of the above stochastic process analogous to Eq. (3.34) using a similar approach employed in [40]. We assume that $S(x_n)$, $T(x_n)$, and ν_n are independent. We let $h : M \rightarrow M$ be an arbitrary, bounded, measurable function, and recall that the expectation of $h(x_{n+1})$ is given by

$$E[h(x_{n+1})] = \int_M h(x) \rho_{n+1}(x) dx. \quad (3.38)$$

Then, using Eq. (3.37) we also obtain

$$E[h(x_{n+1})] = \int_M \int_M h(zS(y) + T(y)) \rho_n(y) g(z) dy dz. \quad (3.39)$$

By a change of variable, it follows that

$$E[h(x_{n+1})] = \int_M \int_M h(x) \rho_n(y) g((x - T(y))S^{-1}(y)) |J| dx dy, \quad (3.40)$$

where $|J|$ is the absolute value of the Jacobian of the transformation $x = zS(y) + T(y)$. Since h was an arbitrary, bounded, measurable function, we can equate Eqs. 3.38 and 3.40 to conclude that

$$\rho_{n+1}(x) = \int_M \rho_n(y) g((x - T(y))S^{-1}(y)) |J| dy. \quad (3.41)$$

The Frobenius-Perron operator for this general form of a stochastic system with both parametric and additive terms can then be expressed similarly to the deterministic case as

$$P_{F_\nu}(\rho(x)) = \int_M \rho(y) g((x - T(y))S^{-1}(y)) |J| dy. \quad (3.42)$$

Notice that the original deterministic kernel δ now becomes a stochastic kernel $g|J|$. Specifically, for the case of the multiplicative perturbation, where $T(x) \equiv 0$, we have

$$P_{F_\nu}(\rho(x)) = \int_M \rho(y) g(xS^{-1}(y)) S^{-1}(y) dy. \quad (3.43)$$

Similarly, the stochastic Frobenius-Perron operator for the additive perturbation, where $S(x) \equiv 1$, is

$$P_{F_\nu}(\rho(x)) = \int_M \rho(y)g(x - T(y))dy. \quad (3.44)$$

We use the Ulam-Galerkin method (a particular case of the Galerkin method [41]) to approximate the (stochastic) Frobenius-Perron operator (3.42). The approximation by this approach is based on the projection of the infinite dimensional linear space $L^1(M)$ with basis functions $\{\phi_i(x)_{i=1}^\infty\} \subset L^1(M)$ onto a finite-dimensional linear subspace with a subset of the basis functions,

$$\Delta_N = \text{span}\{\phi_i(x)\}_{i=1}^N. \quad (3.45)$$

For the Galerkin method, the projection

$$\Pi : L^1(M) \rightarrow \Delta_N, \quad (3.46)$$

maps an operator from the infinite-dimensional space to an operator of finite rank $N \times N$ matrix by using the inner product

$$A_{i,j} = \langle P_{F_\nu}(\phi_i), \phi_j \rangle = \int_M P_{F_\nu}(\phi_i(x))\phi_j(x)dx. \quad (3.47)$$

The quality of this approximation is discussed in Bollt, et al. [2] and similar references can also be found in [42, 43, 44, 45, 46, 47]. For the Ulam method the basis functions are a family of characteristic functions

$$\phi_i(x) = \chi_{B_i}(x) = 1 \text{ for } x \in B_i \text{ and zero otherwise.} \quad (3.48)$$

In the deterministic case, using the inner product Eq. (3.47) the matrix approximation of the Frobenius-Perron operator has the form of

$$A_{i,j} = \frac{m(B_i \cap F^{-1}(B_j))}{m(B_i)}. \quad (3.49)$$

where m denotes the normalized Lebesgue measure on M and $\{B_i\}_{i=1}^N$ is a finite family of connected sets with nonempty and disjoint interiors that covers M , that is $M = \cup_{i=1}^N B_i = M$, and indexed in terms of nested refinements [48]. These $A_{i,j}$ can be interpreted as the ratio of the fraction of the box B_i that will be mapped inside the box B_j after one iteration of a map to the measure of B_i .

Note that one may consider this matrix approximation of the Frobenius-Perron operator as a finite Markov chain, where the partition set $\{B_i\}_{i=1}^N$ represents a set of “states” and A_{ij} characterizes transitions between states. It is well-known that the matrix \mathbf{A} in (3.49) is stochastic and has a left eigenvector with the eigenvalue one. Simply put, this eigenvector characterizes the equilibrium distribution of the Frobenius-Perron operator. In fact, it was proved in [49] that if the partition $\{B_i\}_{i=1}^N$ is a Markov partition, then the (unique) left eigenvector of the matrix \mathbf{A} defines a good approximation of the equilibrium distribution.

A key observation is that the kernel form of the operator in Eq.(3.42) allows us to generally approximate the action of the operator with test orbits as follows. If we only have a test orbit $\{x_j\}_{j=1}^N$, which is actually the main interest of this work, the Lebesgue measure can be approximated by

$$A_{i,j} = \frac{\#(\{x_k | x_k \in B_i \text{ and } F(x_k) \in B_j\})}{\#(\{x_k \in B_i\})}. \quad (3.50)$$

3.4 Almost invariant set

As explained earlier in Chapter 1, one of the main goals of this research is to quantify almost-invariant sets. Now, we would like to formally define a notion of the almost-invariant set based on the Markov model (3.49).

Definition 3.3. A Borel measurable set $A \subset M$ is called *almost invariant* [7] if

$$\frac{\mu(A \cap F^{-1}(A))}{\mu(A)} \approx 1, \quad (3.51)$$

where μ is a probability measure.

A common choice of a probability measure in the above definition is the so-called *natural measure*, which will be denoted by μ from here on unless otherwise stated. Roughly speaking, the natural measure $\mu(B)$ of a measurable set B is high if the amount of time the orbit of Lebesgue almost-all points $x \in M$ spending in B is large, and $\mu(B)=0$ if no orbits enter B or revisit B after a certain sequence of iterations. Hence, we assume that there

exists a unique probability measure μ of a map F that satisfies

$$\mu(B) = \lim_{k \rightarrow \infty} \frac{\#\{F^i x \in B : 0 \leq i \leq k-1\}}{k}. \quad (3.52)$$

When the above measure is used to define the almost invariant set, as expressed in (3.55), we have that if A is an almost invariant set then the probability of a point in A to remain in A after one iteration is close to 1. Generally, the definition of the natural measure is based on the Birkhoff ergodic theorem and given as follows.

Definition 3.4. An ergodic invariant probability measure is called a *natural measure* if

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=0}^N f(F^i x) \rightarrow \int_M f d\mu \quad (3.53)$$

for all integrable $f : M \rightarrow \mathbb{R}$ almost everywhere.

Remark 3.3. Recall that a measure is called invariant under F if $\mu(F^{-1}(A)) = \mu(A)$ for all measurable sets A and an invariant measure is ergodic if $\mu(A) = 0$ or $\mu(A) = 1$ for all measurable sets A such that $F(A) = A$. This notion of ergodicity emphasizes indecomposability of a space to be studied. In short, if a space comprises of more than one invariant set of positive measure, then one could study them separately. Several variant definitions of ergodicity and its relation to the notion of topological transitivity can be found in [50].

Unfortunately, for many dynamical systems this asymptotic notion of the natural measure in Eq. (3.53) can be difficult to compute. First, it is possible that a dynamical system has a very long transient state. Thus, we need to compute a long sequence of iterations to observe eventual behavior of such a system. Therefore, one may not be able to observe the equilibrium distribution unless N in the above definition is sufficiently large. Furthermore, in an extreme case, the round-off problem can prevent us from discovering the equilibrium distribution; for example, a sequence of iterations of the tent map will send any initial point to 0 in finite time. A way to circumvent these problems is to find the invariant measure based on the left eigenvector with the eigenvalue one of the matrix \mathbf{A} defined in (3.49) as explained above. More precisely, the invariant measure of the transition matrix (3.50) is a good approximation of the natural measure of the Frobenius-Perron operator P_F . Note

that a deterministic chaotic dynamical system can have more than one invariant measure, but it always has only one natural measure. In general, there is no guarantee that the invariant density we discovered is the natural measure. However, it was shown [51, 52] that the supports of the approximate invariant measures contain the support of the natural measure, and so we at least capture all the regions with the positive natural measure.

Thus, let p be the left eigenvector of \mathbf{A} with eigenvalue one. After normalizing p so that $\sum_{i=1}^n p_i = 1$, one may approximate the natural measure μ_N for $\{B_i\}_{i=1}^N$ by $\mu_N(B_i) = p_i$. Then the approximate invariant measure of a measurable set B can be defined by

$$\mu_N(B) := \sum_{i=1}^N \mu_N(B_i) \frac{m(B_i \cap B)}{m(B_i)}. \quad (3.54)$$

Note that the quality of this approximation depends upon the number of partition sets N through refinement. The results concerning the convergence of this approximation can be found in [7]. By combining Eqs. (3.49) and (3.54), one can alternatively define almost invariance based on the estimated invariant density p and the transition matrix A , as in the following lemma [53].

Lemma 3.1. Given a box partition set $\{B_i\}_{i=1}^N$ with $\mu_N(B_i) = p_i$, a Borel measurable set $A = \cup_{i \in I} B_i$ for some set of box indices I is called *almost invariant* if

$$\frac{\sum_{i,j \in I} p_i A_{ij}}{\sum_{i \in I} p_i} \approx 1. \quad (3.55)$$

Proof. It is easy to see from (3.54) that $\mu_N(B_i) = p_i$ and hence we have

$$\mu_N(A) = \sum_{i \in I} p_i. \quad (3.56)$$

Now observe that for each fixed $i \in I$, we have

$$\begin{aligned} \mu_N(B_i \cap F^{-1}B_j) &= \sum_{j \in I} \frac{m(B_i \cap F^{-1}B_j)}{m(B_i)} p_i \\ &= \sum_{j \in I} A_{ij} p_i. \end{aligned} \quad (3.57)$$

The desired result follows after summing the above equation over all i . \square

Chapter 4

Graph Community Structure

In the preceding chapter we presented a matrix approximation of the Frobenius-Perron operator, which can be considered as a Markov model (a transition matrix) of the underlying dynamical system. In applying this Markov model, in this chapter we now have occasion to quantify the almost invariant sets of a dynamical system. We will relate the notion of the almost invariant set to that of the graph community structure, which is widely used in the study of social networks. In particular, we will optimize the modularity measure of a graph network to discover the community structure and relate that to the dynamical system. In a short section, we will show a series of examples to demonstrate how this technique can be used in the autonomous case.

4.1 Graph Theory and Reducible Matrices

In this section we briefly review some basic concepts of reducible matrices and corresponding relationships to the graph induced by the matrices due to discrete approximations of a Frobenius-Perron operator. In particular, we introduce the question of how to discover a permutation of the transition matrix (3.50) that reveals the basin structure of a dynamical system given to us in the form of a test orbit $\{x_j\}_{j=1}^N$.

A graph G_A associated to a matrix \mathbf{A} consists of a set of vertices V and a set of edges

$E \subset V^2$. The entry A_{ij} is nonzero when there exists an edge $(i, j) \in E$ that connects the vertices i and j . If a disjoint collection $\{S_i\}_{i=1}^k$ of subsets $S_i \subset V$ consists of vertices such that there is a higher density of edges within each S_i than between them, then we say roughly that $\{S_i\}_{i=1}^k$ forms a *community structure* for G_A [54, 55].

Let $I = \{1 \dots N\}$ be an index set. For our specific application to the transition matrix \mathbf{A} generated by the Ulam-Galerkin method, we define the set of vertices, $V = \{v_i\}_{i \in I}$ to label the original boxes $\{B_i\}_{i \in I}$ used to generate the matrix \mathbf{A} , and define the edges to be the set of ordered pairs of integers $E = \{(i, j) : i, j \in I\}$ which label the vertices as their starting and ending points.

Definition 4.1. A graph is said to be *reducible* if there exists a subset $I_o \subset I$ such that there are no edges (i, j) for $i \in I_o$ and $j \in I \setminus I_o$, otherwise it is said to be *irreducible* [56, 57].

This condition implies that the graph is irreducible if and only if there exists only one connected component of a graph, which is G_A itself. In term of the transition matrix \mathbf{A} , G_A is reducible if and only if there exist a subset $I_o \subset I$ such that $A_{ij} = 0$ whenever $i \in I_o$ and $j \in I \setminus I_o$. Furthermore, \mathbf{A} is said to be a reducible matrix if and only if there exists some permutation matrix P such that the result of the similarity transformation,

$$\mathbf{R} = \mathbf{P}^{-1} \mathbf{A} \mathbf{P} \quad (4.1)$$

is block upper triangular

$$R = \begin{bmatrix} R_{1,1} & R_{1,2} \\ 0 & R_{2,2} \end{bmatrix}. \quad (4.2)$$

This means that G_A has a decomposition $V = V_1 \cup V_2$, such that V_1 connects with V_1 and V_2 , but V_2 connects only with itself. When $R_{1,2} = 0$, \mathbf{A} is said to be “completely reducible” [57]. A key observation is that in the case that G_A is generated from a bi-stable dynamical system the transition matrix A is completely reducible, where $R_{1,1}$ and $R_{1,2}$ corresponds to the two basins of attractions of the system. Also, in a general multi-stable dynamical systems the transition matrix \mathbf{A} has a similarity transformation into a block (upper) triangular form.

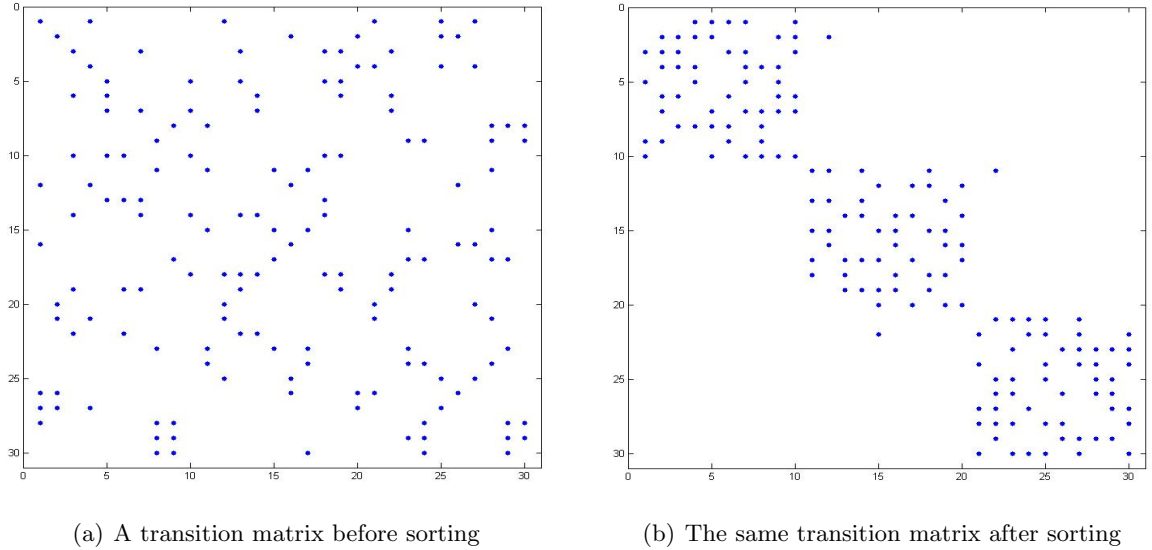


Figure 4.1. The matrix in this figure is irreducible. Members of the original transition matrix before sorting are placed haphazardly and the structure is not obvious. After using the modularity method discussed in Section 4.2 the “almost” block diagonal form is revealed. In this case, the sorted matrix indicates the existence of three communities and the few off block diagonal elements correspond to intra-basin diffusion.

A key point to this work is that most (randomly or arbitrarily realized) indexing of the configuration makes it difficult to observe even simple structures like community structure or reducibility of the graph and corresponding transition matrix. In particular, a very reasonable partitioning of the phase space, and a reasonable indexing of the boxes or triangles is not likely to reveal dynamic organization like reducibility or a community structure. In Figures 4.1(a) and 4.1(b) we show an example of a transition matrix for a 30-vertex random community-structured graph with three communities of which members are placed haphazardly, but after a proper permutation this matrix is transformed into an “almost” block diagonal form. Figures 4.2(a) and 4.2(b) illustrate the associated graph of the matrix in the above example. Before sorting the vertices into three separate communities the graph looks like a random graph that has no community structure. However, after sorting we can see a community structure of the graph.

There are several techniques to find an appropriate permutation if the matrix is reducible. In the language of graph theory, we would like to discover all connected components of a graph. The betweenness methods in [55] and the local method in [58] are examples of

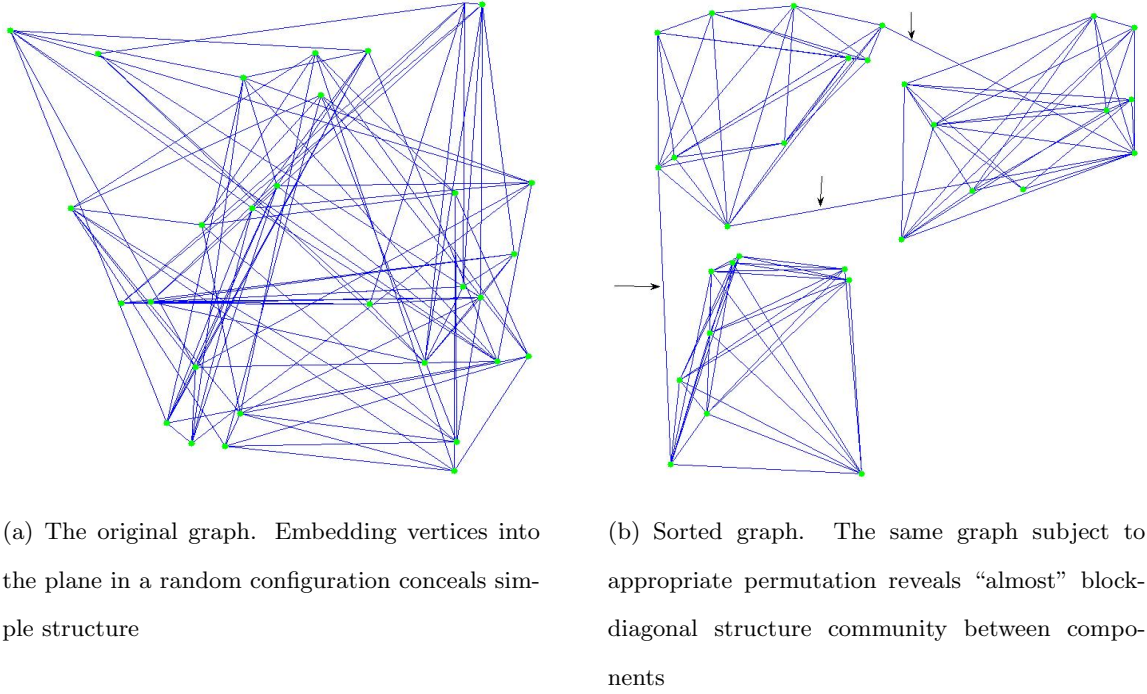


Figure 4.2. An undirected graph before and after sorting. The arrows point at the inter-community edges.

numerous methods as reviewed in [59, 55] that can be successfully used to uncover various kinds of community structures.

In the more general case in which noise is added to a multi-stable dynamical system, we do not expect a reducible transition matrix because noise can cause basin hopping, which in turn adds the connections (edges) between each original connected components representing each basin in the noiseless system. The larger the magnitude of the noise, the larger the number of the inter-community edges. Therefore, there will be no similarity transformation of the transition matrix into a block diagonal form for a noisy system. However, G_A still has a community structure if noise is small enough. Note that in a case that a magnitude of noise is large a number of inter-community edges may become so large that a community structure is destroyed. In other words, one might be able to discover only one community, which represents the entire phase space of the time-series data, or one may obtain a large number of very small communities that have no meaningful community structure. Bollt, et al. [2] developed a technique to discover an appropriate permutation based on a convergence property of elements in B_i . However, such a technique could not be applied to the case

that we merely have a test orbit $\{x_i\}_{i=1}^N$ of a real world dynamical systems, i.e., the data collected from experiments.

The question of how to discover a proper community structure of a noisy multi-stable dynamical system without a priori knowledge of a mathematical model of the system is the main interest of this chapter. The edge betweenness method [54], based on computing shortest paths, is a good robust candidate algorithm for detecting communities, but its run time is of the order $O(m^2n)$ for a graph with m edges and n vertices, or $O(n^3)$ for a sparse graph ($m \sim n$). Such expensive demands on computational resources make it infeasible for a large graph. Furthermore, a d -dimensional dynamical system with an h^d -element box (for box of order length h on a side) or triangle covering of the invariant set makes therefore an $O(h^{3d})$ algorithm, which is quite expensive for a fine grid, $h \ll 1$, in high dimension- d . In the next section we will describe a method based on a modularity measure of a graph partition developed by Newmann [60]. This method is far less expensive than the betweenness method and gives useful results in many applications. In the last section of this chapter we demonstrate examples of how to apply this method to our specific applications in dynamical systems.

4.2 Modularity Method

As discussed in the previous section our main purpose is to discover community structure in a network derived from an orbit of a dynamical system through a grid partitioning of the phase space; such structure indicates the dynamical basin structure. Thus, our point of view is to map the problem of phase space partition of the dynamical system (approximately) into a problem of partitioning a digraph. Many methods have been proposed in recent years [60, 58, 55, 54] to appropriately partition a digraph. Simply put, the community structure is a partition of the digraph into subgraphs such that there are relatively more connections within each defined component than between components, a sort of self clustering.

First we discuss the concept of the modularity as a measure of quality of a proposed community structure found by any graph partition algorithm. The modularity is a cost

function associated with a partitioning of a given graph G

$$Q : \mathcal{P}_G \rightarrow \mathbb{R}, \quad (4.3)$$

where \mathcal{P}_G is the set of all subpartitions P , of a given graph G . Given a graph G and a (test) partition, $P \in \mathcal{P}_G$, $P = \cup_k P_k$, each P is a set of subsets P_k , and P_k is a collection of vertices of G . P includes all of the vertices of G . We will refer to P_k as “community”- k . The modularity of the partition P is meant to reflect the quality of the split into self clustered elements P_k . Newman [60] proposed that a good division of a graph network into communities could be justified only if the number of the within-communities edges is significantly larger than the expected number of such edges when edges are randomly placed into a network, see also Eq. (4.18). Let us denote by c_i the community that the vertex i belongs to and A_{ij} be an element of the adjacency matrix of the network:

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \quad (4.4)$$

The modularity Q can be expressed by

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - R_{ij}) \delta(c_i, c_j), \quad (4.5)$$

where m is the total number of edges, R_{ij} is the so-called “null” model to be discussed below and the delta-function $\delta(c_i, c_j) = 1$ if $c_i = c_j$ and 0 otherwise. Hence, the large value of Q reflects a strong community structure of a test partition and a good division of a network into communities can then be achieved by an optimization of Q over all possible candidates.

Now observe that in Eq. (4.5) the first term of the summation is the fraction of within-communities edges, i.e., those edges that connect vertices belonging to the same community. The second term is the null model R_{ij} that is used to compare against our network and describes the expected number of the within-community edges when edges are placed in certain equivalent randomized network regardless to the community structure hidden in the actual network. Thus R_{ij} is the expected number of edges between vertices i and j . One may assume a different null model according to each application in which a particular density of edges may become more important to define a community than the others. In [60] Newman used the following constraints to define the null model.

Let k_i denote the degree of vertex i , which is defined to be the number of edges incident upon it:

$$k_i = \sum_j A_{ij} \quad (4.6)$$

The first constraint is that the expected degree of each vertex within a null model is equal to that of the corresponding vertex in the actual network. Hence, for each vertex i within the null model, we have

$$\sum_j R_{ij} = k_i. \quad (4.7)$$

Consequently, summing over all the vertex in the null model yields

$$\sum_{ij} R_{ij} = \sum_{ij} A_{ij} = 2m. \quad (4.8)$$

The above expression implies that $Q = 0$ when there is solely one community in the entire network; one may derive this directly from the definition of Q by setting $Q = 0$, and that the expected number of edges within the null model is equal to the number of edges in the actual network.

The second constraint requires that the probability that a vertex i is attached by an end of a randomly chosen edge depends only on the degree k_i of the vertex i regardless of the probability of the other vertex attached by the other end of the same edge. Therefore, we assume that P_{ij} has a form of

$$R_{ij} = f(k_i)f(k_j), \quad (4.9)$$

where f is a (discrete) probability function. Note that the symmetry of R_{ij} is automatically obtained from the above constraint. Substituting Eq. (4.9) to Eq. (4.7) yields

$$\sum_j R_{ij} = f(k_i) \sum_j f(k_j) = k_i \quad \text{for all } i. \quad (4.10)$$

Hence, we must have that $f(k_i) = Ck_i$ for some constant C . It then follows from Eq. (4.8) that

$$2m = \sum_{ij} R_{ij} = C^2 \sum_{ij} k_i k_j = (2mC)^2, \quad (4.11)$$

or

$$C = \frac{1}{\sqrt{2m}}. \quad (4.12)$$

Therefore, the expected number of edges between vertices i and j is

$$R_{ij} = \frac{k_i k_j}{2m}. \quad (4.13)$$

Using the above probability we can rewrite the modularity Q in Eq. (4.5) as

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j). \quad (4.14)$$

Since $\delta(c_i, c_j) = \sum_k \delta(c_i, k) \delta(c_j, k)$, the above expression can be rewritten as

$$\begin{aligned} Q &= \frac{1}{2m} \sum_{ij} (A_{ij} - R_{ij}) \sum_k \delta(c_i, k) \delta(c_j, k) \\ &= \sum_k \left(\frac{1}{2m} \sum_{ij} A_{ij} \delta(c_i, k) \delta(c_j, k) - \frac{1}{2m} \sum_i k_i \delta(c_i, k) \frac{1}{2m} \sum_j k_j \delta(c_j, k) \right). \end{aligned} \quad (4.15)$$

Now we define e_k to be the fraction of edges that connect vertices within the community P_k ,

$$e_k = \frac{1}{2m} \sum_{ij} A_{ij} \delta(c_i, k) \delta(c_j, k), \quad (4.16)$$

and a_k to be the fraction of ends of edges that are attached to vertices in community P_k ,

$$a_k = \frac{1}{2m} \sum_i k_i \delta(c_i, k). \quad (4.17)$$

Therefore, an alternative expression of Eq. (4.14) is

$$Q = \sum_k (e_k - a_k^2). \quad (4.18)$$

Example 4.1. Consider a graph network consisting of K identical communities without any inter-communities edges. Let m be the total number of edges within the graph network. If we choose a test partition to be a collection of these K communities, it is simple to see that $e_k = a_k = \frac{1}{K}$. So, we have $Q = 1 - \frac{1}{K}$.

Example 4.2. Consider again the graph network in the preceding example but allow an edge to connect between community i and $i + 1$. Thus we now have $2m + K - 1$ edges in the network. Again, we choose the test partition to be the K communities. It follows that

$$\begin{aligned} Q &\leq K \left(\frac{1}{K} - \left(\frac{\frac{2m}{K} + 1}{2m + K - 1} \right)^2 \right) \\ &= 1 - \frac{1}{K} \left(\frac{\frac{2m}{K} + 1}{\frac{2m}{K} + 1 - \frac{1}{K}} \right)^2 \\ &< 1 - \frac{1}{K}. \end{aligned} \quad (4.19)$$

We see that Q is decreased when edges connected between communities are added to the network.

Now we generalize the notion of modularity to a weighted, directed network. An equivalent definition to Eq. (4.14) of the modularity for this network is given by

$$Q = \frac{1}{m} \sum_{ij} \left(P_{ij} - \frac{k_i^{out} k_j^{in}}{m} \right) \delta(c_i, c_j), \quad (4.20)$$

where m now becomes the total weight, P_{ij} is the weight of the edge between the vertices i and j and k_i^{out} and k_j^{in} are the out- and in- degree of the vertices, respectively. More precisely, we have $m = \sum_{ij} P_{ij}$, $k_i^{out} = \sum_j P_{ij}$ and $k_j^{in} = \sum_i P_{ij}$. Note that the factor of 2 in the denominator is eliminated since the edges are not counted twice as in the case of Eq. (4.5) for the undirected network. If the weight of the matrix is given according to the matrix approximation of a Frobenius-Perron operator as described in Eq. (3.50), then the total weight m is equal to the number of vertices in the network and $k_i^{out} = \sum_j P_{ij} = 1$.

Recall that a good partition of a graph network into communities could be found by optimizing the modularity function Q . The true optimization is, however, very costly to implement in practice for very large networks. Clauset, Newman, and Moore [61] proposed an approximate optimization algorithm based on greedy optimization. Suppose that we have a graph with n vertices and m edges. This algorithm starts with each vertex being the only member of one of n communities. At each step, the change in $Q(P)$ is computed after joining a pair of communities together and then choosing the pair that gives the greatest increase or smallest decrease in $Q(P)$. The algorithm stops after $n - 1$ such joins in which a single community is left. This algorithm therefore runs in time $O((m + n)n)$, or $O(n^2)$ on a sparse graph ($n \sim m$). Note that using more sophisticated data structure as introduced in [61] can reduce a run time to $O(n \log^2 n)$. Furthermore, the algorithm builds a dendrogram which represents nested hierarchy of all possible community partition of the network. Then we select the partition corresponding to local peaks of the objective function $Q(P)$ for our satisfactory community division. Figure 4.3 shows an example of the dendrogram of the graph network shown in Figure 4.2(a). We can see that the peak in the modularity corresponds to the correct identification of the community structure and the

modularity is dropped from the peak when two of the three communities are joined together at the final step. We note that the quality of the partition can be determined by the value of Q . A typical value of Q for a good division in many real-world network is around $0.2 - 0.3$. If the partition obtained from an optimization scheme shows a considerably smaller value of Q , say $Q < .1$, we assume that there is no significant partitions of the given network into communities.

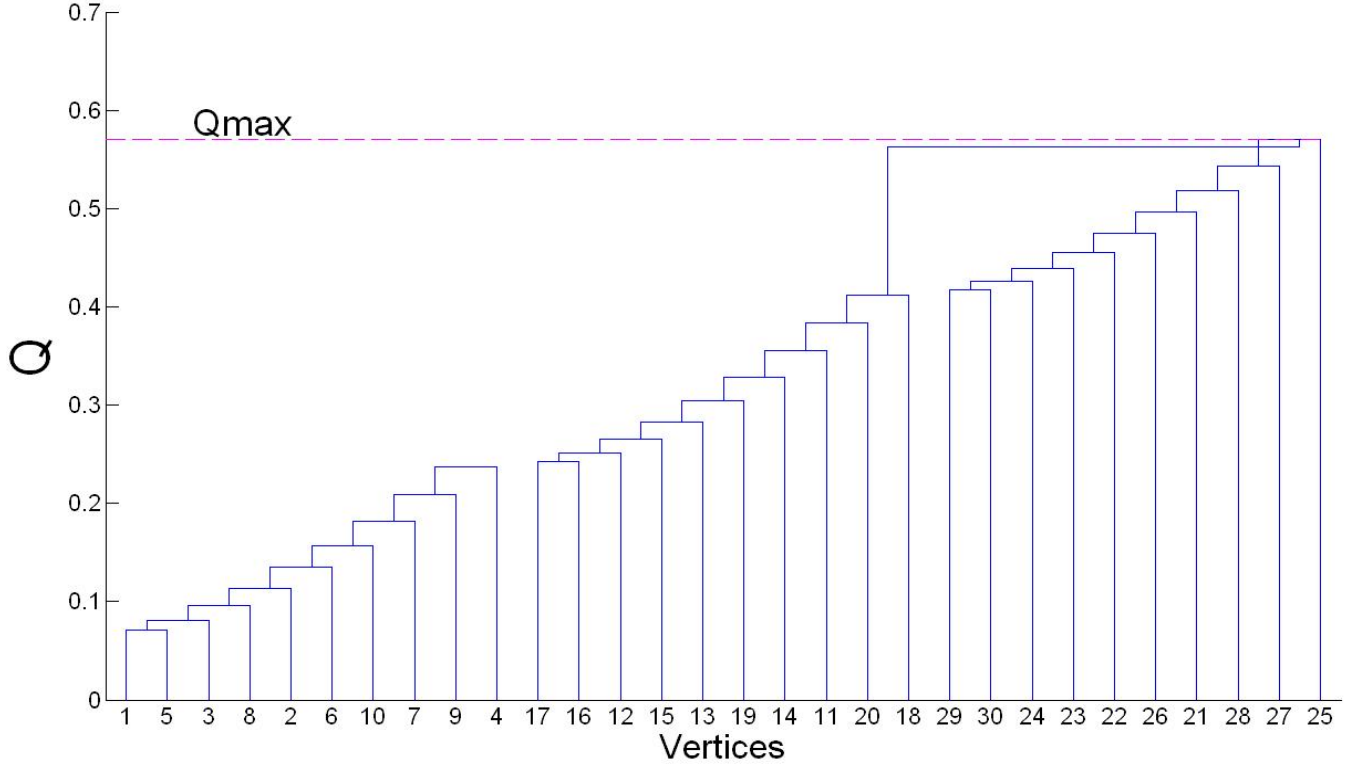


Figure 4.3. Plot of dendrogram for a graph network in Figure 4.2(a). As we can see the peak in the modularity (dashed line) occurs where the network is divided into three communities.

Other approaches to optimize Q are spectral method. An idea of optimization schemes based on spectral methods has been traditionally used to determine a division of a network so that the minimum cut is achieved. To those types of problems the eigenvalue spectrum of the Laplacian matrix is utilized to divide the network. A review of this method as applied to the minimum cut problem may be found in [62]. For our purpose, we first consider the partition of a network into two communities. Let a vector \mathbf{s} of n elements, where n is the

number of vertices, be defined as

$$s_i = \begin{cases} 1 & \text{if } i \text{ is in community 1} \\ -1 & \text{if } i \text{ is in community 2} \end{cases} \quad (4.21)$$

Then we can rewrite Eq. (4.20) as

$$Q = \frac{1}{2m} \mathbf{s}^T \mathbf{B} \mathbf{s}, \quad (4.22)$$

where \mathbf{B} is called the *modularity* matrix defined by

$$B_{ij} = P_{ij} - \frac{k_i^{\text{out}} k_j^{\text{in}}}{m}. \quad (4.23)$$

Therefore, we would like to determine the vector \mathbf{s} that maximizes Q under the constraint that $\mathbf{s}^T \mathbf{s} = n$. In general, the modularity matrix B for a directed network is not symmetric and so the spectral method cannot be directly applied. Instead, we consider the modularity Q of the following form:

$$Q = \frac{1}{4m} \mathbf{s}^T (\mathbf{B} + \mathbf{B}^T) \mathbf{s}. \quad (4.24)$$

Note that the matrix $\mathbf{B} + \mathbf{B}^T$ is not the same as the modularity matrix of the same network but ignores the directions. The idea of the spectral scheme is to write \mathbf{s} as a linear combination of the eigenvectors \mathbf{v}_i of $\mathbf{B} + \mathbf{B}^T$. Then we can express Q in term of the eigenvectors \mathbf{v}_i and their corresponding eigenvalues λ_i as

$$Q = \sum_i \lambda_i (\mathbf{v}_i^T \mathbf{s})^2. \quad (4.25)$$

Then we can choose \mathbf{s} to be as close to parallel with the leading eigenvector, say \mathbf{v}_1 , of the largest eigenvalue to maximize Q , i.e., maximizing the quantity

$$|\mathbf{v}_1^T \mathbf{s}| = \left| \sum_i s_i v_1^{(i)} \right| \leq \left| \sum_i v_1^{(i)} \right|. \quad (4.26)$$

Under the constraint $s_i = \pm 1$, this can be achieved by choosing the sign of s_i to be the same as that of $v_1^{(i)}$.

We have only determined the optimal partition of the network into two communities. If the network comprises of more than two communities, we have to further divide these two communities. Thus we repeat the bisection process to each community of the network

using the spectral partitioning as described above and continue dividing the result further until the modularity of the network cannot be increased. In the course of this procedure, we have to compute the change in modularity denoted by ΔQ to decide whether or not the community k can be subdivided further. We define a vector \mathbf{s} as before except that only vertices within the community g are taken into account and note that ΔQ has to be computed based on the entire network. Then we have

$$\begin{aligned}
\Delta Q &= \frac{1}{2m} \left[\sum_{i,j \in c_g} (B_{ij} + B_{ji}) \frac{s_i s_j + 1}{2} - \sum_{i,j \in c_g} (B_{ij} + B_{ji}) \right] \\
&= \frac{1}{4m} \sum_{i,j \in c_g} \left[(B_{ij} + B_{ji}) - \delta_{ij} \sum_{k \in c_g} (B_{ik} + B_{ki}) \right] s_i s_j \\
&= \frac{1}{4m} \mathbf{s}^T (\mathbf{B}^{(g)} + \mathbf{B}^{(g)T}) \mathbf{s},
\end{aligned} \tag{4.27}$$

where we set

$$B_{ij}^{(k)} = B_{ij} - \delta_{ij} \sum_{k \in g} B_{ik}. \tag{4.28}$$

To obtain the first line in the above derivation we subtract the modularity Q after subdivision from Q before subdivision and notice that the term concerning vertices outside the community g can be canceled out. Then we use the fact that $s_i^2 = 1$ to derive the second line. Now, we can apply the spectral bisection scheme to determine the partition of the community g that maximizes ΔQ of the entire network. If ΔQ happens to be negative or smaller than some threshold value of ΔQ , it means that the community k should not be subdivided further.

4.3 Markov Clustering

Another interesting method to detect communities structure in a network is the Markov clustering (MCL) introduced by Stijn Van Dongen [63]. We will see that this method strongly relates with the notion of the almost invariant set and random walks on the network. The basic idea of the MCL is that when a random walk on a network falls into a dense cluster or community, it will not likely to leave the cluster until many of the vertices have been visited. Therefore, this observation of a random walk has a strong relation to the almost invariant set where a random walk has a high probability to stay inside the community for

a long period of time before escaping to another community. To apply the idea of a random walk on a weighted graph, we first have to normalize the matrix associated to a weighted graph to obtain a stochastic matrix of which the weight sum of all outgoing edges for each vertex equals to one. Therefore, the random walk in this section can be viewed as a finite Markov chain. Note that in the case of undirected graphs, we can readily see that the stochastic matrix obtained from the normalization of the outgoing edges will be diagonally similar to a symmetric matrix; the finite Markov chain is time-reversible. Note also that we will see below that the stochastic matrix will be nonlinearly “inflated” and so most results from the finite Markov chain theory are not valid for the analysis of the MCL. So, suppose that we have a stochastic matrix P such as the transition matrix defined in (3.49). The MCL algorithm is the following.

Algorithm: MCL

Choose the parameter $r > 1$ and the number of iterations n

begin iterations

Multiplication step: $P = P^2$

Inflation step: $P_{ij} = P_{ij}^r$

Normalize each column so that P becomes stochastic

$n:=n+1$

end iterations

In the multiplication step, we compute the probability P_{ij}^2 that a random walk starting from vertex i will go to vertex j in two steps. In the inflation step, the random walk with

a high probability is favored by taking the r^{th} power of the matrix P componentwise. The matrix then has to be normalized for the next iteration. Indeed, we will refer to the combination of inflation and normalization steps as the inflation step. Therefore, the most probable path of the random walk starting from i will become more probable, whereas the path connected vertices from different communities will be less probable in the next iteration of the random walk. This implies that a random walk will eventually end up at some most probable path. In most cases, the matrix P will converge to the equilibrium state of the MCL process in which P is invariant under multiplication and inflation. As a consequence, if two random walks based on the *MCL* algorithm starting at two different vertices eventually end up at the same vertex, they are regarded as the members of the same community. Note that the inflation step is necessary to detect a cluster or community structure since without the inflation the random walks will end up leaving the community and converge to the stationary state as the number of iteration increases.

Due to the multiplication step, the computational time of the algorithm is $O(n^3)$, where n is the number of vertices. Nevertheless, most real-world networks realizing a dynamical system are sparse and so the method can still be useful. In fact, Gfeller [64] has demonstrated an application of MCL for the community detection of a large network with $N \sim 10^4$, which is done in a reasonable time. Note that as far as our knowledge is concerned, the convergence of this method has not yet been developed.

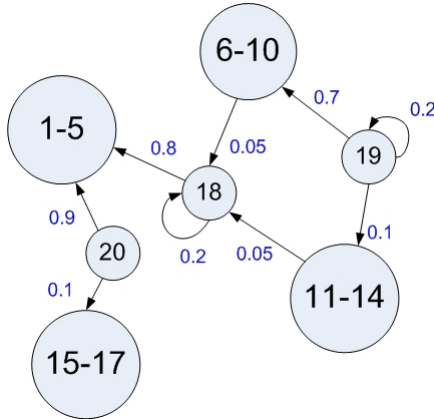


Figure 4.4. A graph network of 20 vertices.

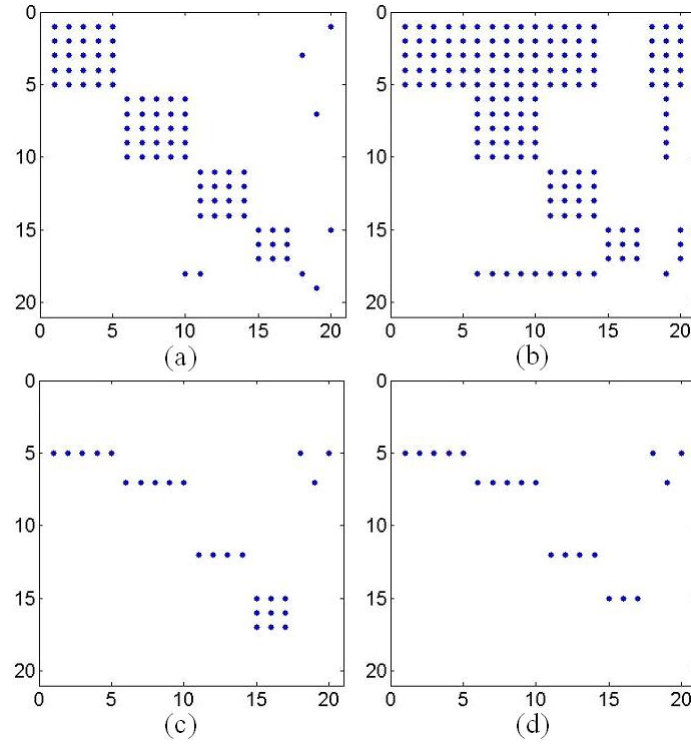


Figure 4.5. The change in matrix P at different iterations of MCL with $r = 2.0$. (a) The original stochastic matrix P . From (b)-(d) we show the matrix P at steps 10, 20 and 30 respectively. Notice that the matrix in (d) reveals three communities in the network, that is $\{1 - 5, 18, 20\}$, $\{6 - 14, 19\}$ and $\{15 - 17\}$.

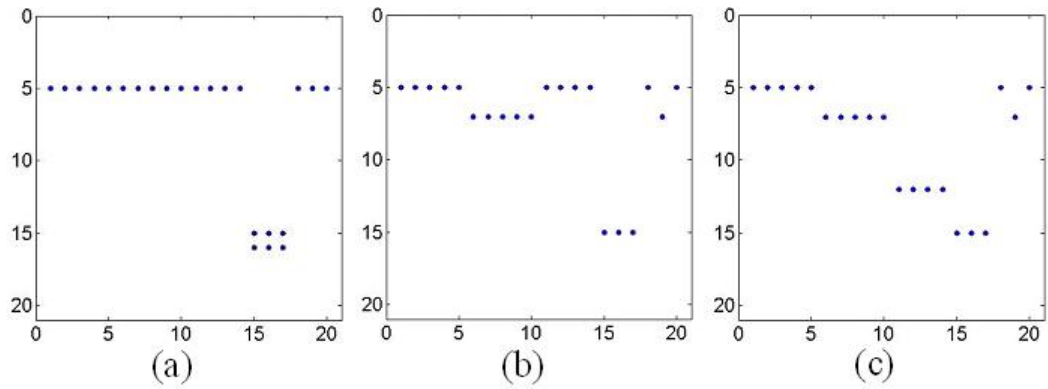


Figure 4.6. The effect of the scale of r is shown at three different scales. (a) For $r = 1.2$ two communities, $\{15 - 17\}$ and $\{1 - 20\} \setminus \{15 - 17\}$, are detected by MCL. (b) For $r = 1.5$ MCL discovers three communities (c) At $r = 2$ MCL detects four communities. Note that for $r > 2$ MCL gives the same result as the case of $r = 2$.

Now we consider as a toy example a weighted, directed network of 20 vertices shown in Figure 4.4. In this network, there are four complete subgraphs; $\{1 - 5\}$, $\{6 - 10\}$, $\{11 - 14\}$ and $\{15 - 17\}$, and they are connected via the vertices $\{18 - 20\}$. Heuristically, each of these four subgraphs should structure a community and the vertices 18 and 20 should also belong to the community consisting of $\{1 - 5\}$. Also, the vertex 19 should combine with $\{6 - 10\}$ to form a community. The results of MCL with $r = 2.0$ at different iterative steps are demonstrated in Figure 4.5. The rows with nonzero entries reveal the community structure of the network and all the nonzero entries in Figure 4.5(d) are equal to one. Figure 4.6 shows how the result changes as the parameter r is varied. We can see that the number and size of communities depend on the parameter r . Fewer but larger communities are detected for a small value of r whereas for a larger values of r MCL detects a larger number of communities but smaller in size. Note that the result shown in Figure 4.6(a) is obtained after 500 iterations and the nonzero entries related to the community $\{15 - 17\}$ are all equal to 0.5 instead of 1.

Finally, we remark there are some interesting cases in which a periodic limit occurs in the MCL [63] and that the idea of the random walk used in the MCL algorithm has a strong similarity to the simulated annealing method as explained in [64].

4.4 Examples

In this section we first for benchmarking purpose demonstrate an application of the modularity method to the double-well Duffing oscillator, where the solution is the expected double wells. Then we apply the method to a gearbox model from mechanical engineering [65, 1] and then finally to a many basin system which is known only through data. We will observe changing basin structure with the standard deviation of the noise.

For each model, we simulate a long test orbit. Recall that we are interested in the case that we only have a given orbit, but lack a mathematical model of the map that generates the test orbit. Then as explained above, we develop a digraph transition model of the Frobenius-Perron operator, discretized on a fine triangulation grid of the phase space.

Then our goal becomes the determination of an appropriate partition of the phase space using the modularity method explained in the previous section.

4.4.1 Duffing Oscillator

Our first example is chosen for sake of simplicity to benchmark and clarify our method. Consider the the double-well Duffing oscillator given by.

$$\begin{aligned}\dot{x} &= y \\ \dot{y} &= x - x^3 - \delta y + F_0 \cos(2\pi\omega t),\end{aligned}\tag{4.29}$$

where $\delta = 0.25$, $F_0 = 0.25$, and $\omega = 1$. To obtain a stochastic version of this model noise is added to the above equations periodically at the same phase having mean zero and standard deviation σ . Thus this model may be viewed as a stochastic map,

$$F : \mathbb{R}^2 \rightarrow \mathbb{R}^2, \quad (x, y)(t+1) = F[(x, y)(t)] + \nu(t),\tag{4.30}$$

where $\nu(t)$ is a discrete noise term. This additive noise induces chaos-like oscillations since it causes noise-induced basin hopping between two basins of the noise-free system. In order to study where in the phase space these basin hoppings are most likely to occur we should first determine the basin boundaries.

For the particular experiment shown below, we randomly choose an initial condition and iterate it 50,000 times using the time- T map, $T = 1$, of the Duffing model with normal noise of zero mean and standard deviation $\sigma = 0.01$, which is added after each iteration of the map. Note that in the noiseless case, for the particular chosen parameter values of the deterministic system, if the initial point belongs to one of the wells it will converge to the bottom of the well ($x = -1$ or $x = 1$). However, when noise is added, hopping between the two basins may occur sporadically, depending on the nature of the noise. Of course, given just the data of a sampled orbit, it is not immediately obvious where are the basins and where the jumping between the basins occurs. Of course, we only know this information since we have chosen a particularly simple benchmark example. We will now demonstrate that our algorithms defined in the previous sections can automatically discover what we already know for this problem.

With the test orbit in hand we choose a bounded phase space that covers every point in the orbit and divide it into small triangular regions (3200 triangles are used in this example). After (arbitrarily) indexing each triangle we then generate a graph whose edges and vertices correspond to the transition of points in the orbit and the index of these triangles, respectively. Then we can apply the modularity method to find a partition of this graph.

The results of the digraph partitioning are shown in Figure 4.7, and the correspondingly indexed triangles are embedded back in their native phase space positions, and colored according to the partition results. We can see from 4.7 that there are two basins colored by magenta and yellow in accordance with left and right wells, as expected. Furthermore, hopping between the two basins occurs many times. The triangles are automatically colored according to the partition by the modularity method, which agrees very well with a priori known behavior described in the above paragraph. We see that the two basins are each automatically colored as their own connected components in the phase space. Figure 4.8(a) and 4.8(b) shows the Galerkin matrices before and after the sorting by using the partition due to the modularity method. Of particular interest when inspecting the transitions matrices in their almost canonical reducible form is the elements in the off-diagonal blocks. These off block-diagonal elements indicate the hopping between two the basins, which is approximately in agreement with (4.2). Accordingly, the corresponding triangles are the more active transport regions in the phase space.

4.4.2 Gearbox Model

Our second example is a model for rattling in a single-stage gearbox system with a backlash, comprised of two wheels with a sinusoidal driving (See Figure 4.9) [1, 65] and this development is featured in our work [28]. The relative displacement between the gears due to the backlash is

$$s = \frac{AR_e}{\nu} \sin \omega t - \frac{R}{\nu} \varphi, \quad (4.31)$$

where A and ω are angular amplitude and frequency of the driving gear, respectively, R and R_e are the radii of two gears with a backlash ν , and φ is the angular displacement

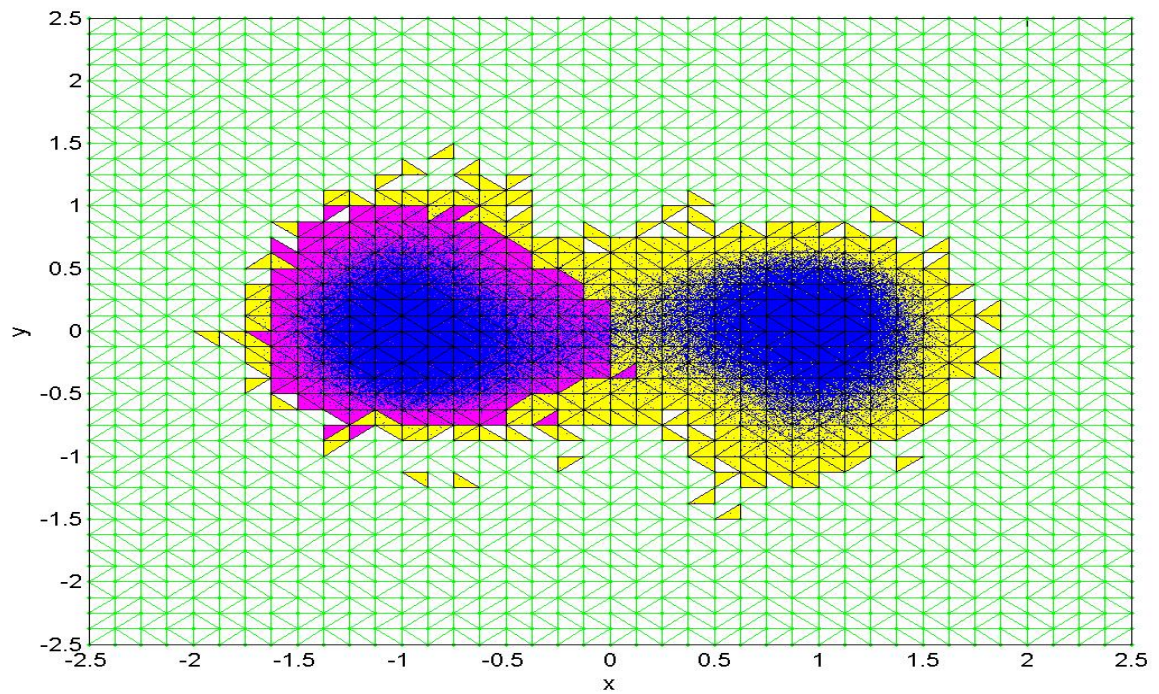


Figure 4.7. A partition of an orbit generated from the Duffing model with $\delta = 0.095$ and normal noise, $\sigma = 0.01$. The colors, magenta and yellow, are according to the two communities found by the modularity method applied to the transition matrix, and we see that the colors of these corresponding triangles agree well with the a priori known basins of the deterministic system. The blue dots shown are the test orbit used in the process.

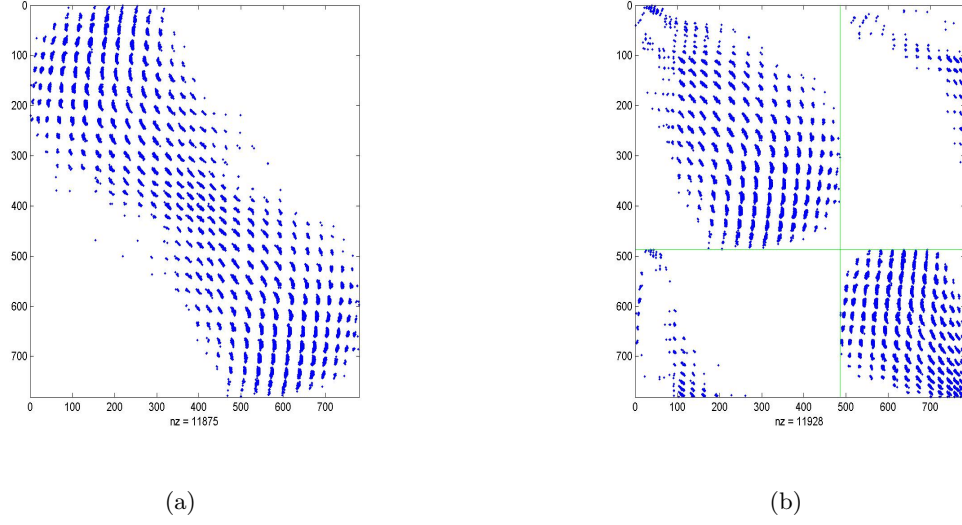


Figure 4.8. Galerkin matrices before and after sorting of the Duffing oscillator. (a) The Ulam-Galerkin matrix from the Duffing oscillator before sorting. No obvious basin structure or transport in this operator. (b) The Ulam-Galerkin matrix from the Duffing oscillator after sorting into approximate reducible form. We see the basin structure of off-diagonal elements showing relative transport between basins.

of the second gear. The s variable varies from -1 to 0 and the impact occurs when $s = 0$ or -1 . Let s_n, \dot{s}_n , and τ_n represent the displacement, velocity, and time (modulo 2π) just before the n th impact. Then the dynamical variable for $(n+1)$ th impact can be derived as a function of the n th impact [6];

$$s_{n+1} = s_n + \alpha(\sin \tau_{n+1} - \sin \tau_n) + \frac{\gamma}{\beta}(\tau_{n+1} - \tau_n) - \frac{1}{\beta}(1 - \exp[-\beta(\tau_{n+1} - \tau_n)])(r\dot{s}_n + \alpha \cos \tau_n + \frac{\gamma}{\beta}), \quad (4.32)$$

$$\dot{s}_{n+1} = \alpha \cos \tau_{n+1} + \frac{\gamma}{\beta} - \exp[-\beta(\tau_{n+1} - \tau_n)](r\dot{s}_n + \alpha \cos \tau_n + \frac{\gamma}{\beta}). \quad (4.33)$$

Pictures showing the physical meaning of these variables in more detail can be found in [65, 1].

We fix values, $s_0 = -0.5$, $\beta = 0.1$, $\gamma = 0.1$, and $r = 0.95$. The bifurcation diagram shown in Figure 4.10 shows that two attractors co-exist near $\alpha = 2$ and these attractors appear as the successive time of five impacts. Following Souza *et. al.* [1], we add noise to the control parameter α . That is,

$$\alpha_n = \alpha + \eta(n), \quad (4.34)$$

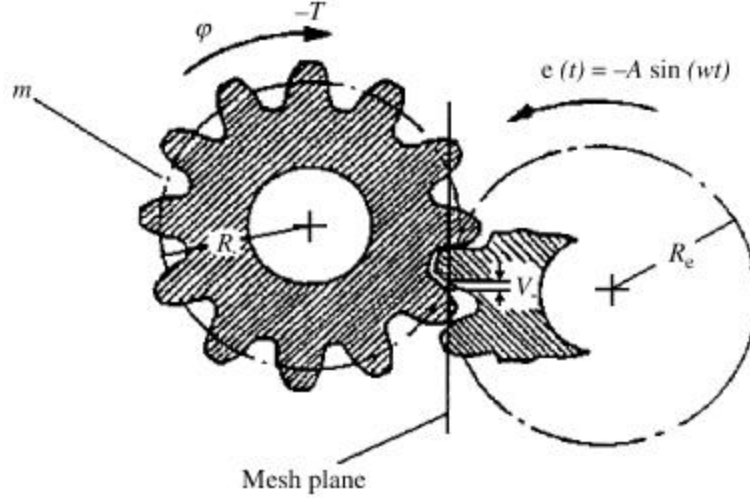


Figure 4.9. Schematic view of the gearbox system [1]

where $\eta(n)$ is a discrete noise term, with a normal distribution with zero mean and with standard deviation σ . Note that in this example we have the case of the stochastic Frobenius-Perron operator Eq. (3.42), in which the perturbation is neither additive nor multiplicative but a combination of both. We simulate a time series using 60,000 iterations for $\alpha = 1.985$ and set the noise standard deviation at $\sigma = 0.0018$. A time series for this case is shown in Figure 4.11, where the basin hopping can be noticed.

Following the same procedures as described in the preceding example we can generate a graph associated to the orbit of the gearbox model. The result from the modularity method for this case is shown in Figure 4.12, where the large squares and dots indicate each attractor for the noiseless case. The transition matrices before and after sorting according to the partition from the modularity method are shown in Figure 4.13(a) and 4.13(b).

4.4.3 A Stochastic Model with many basins

Consider the following Hamiltonian system with a stochastic white noise:

$$\begin{aligned} \frac{dx}{dt} &= \frac{\partial \psi}{\partial y} + \eta_x(t) \\ \frac{dy}{dt} &= -\frac{\partial \psi}{\partial x} + \eta_y(t) \end{aligned} \tag{4.35}$$

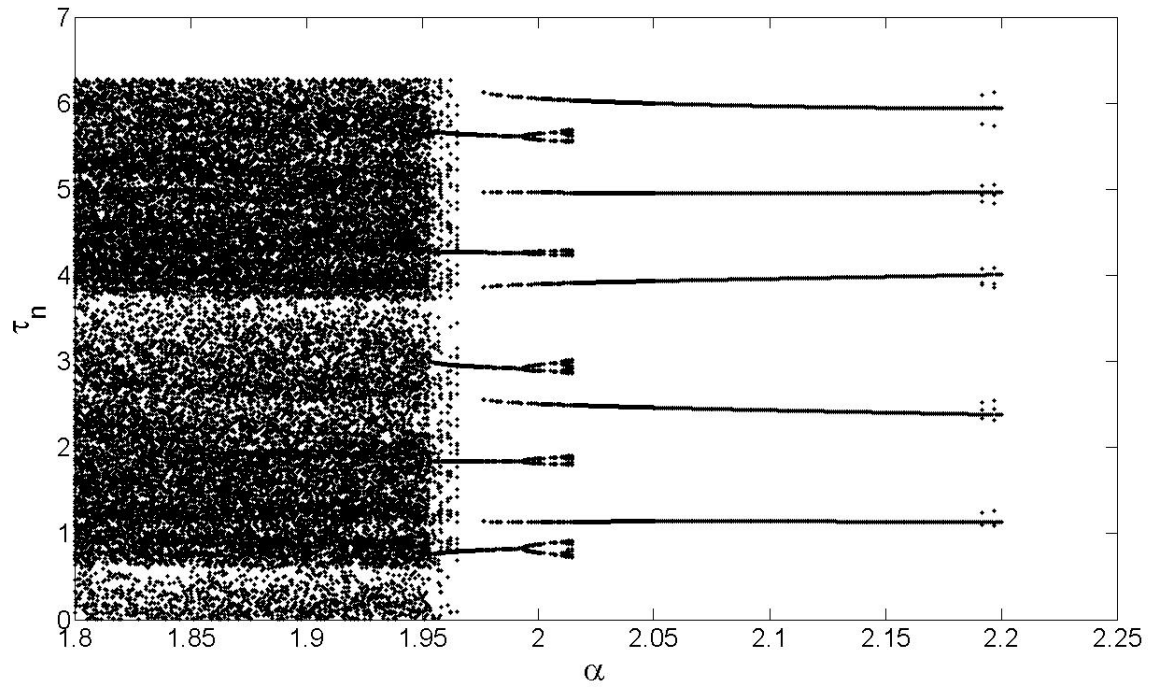


Figure 4.10. Bifurcation diagram for the time of impact.

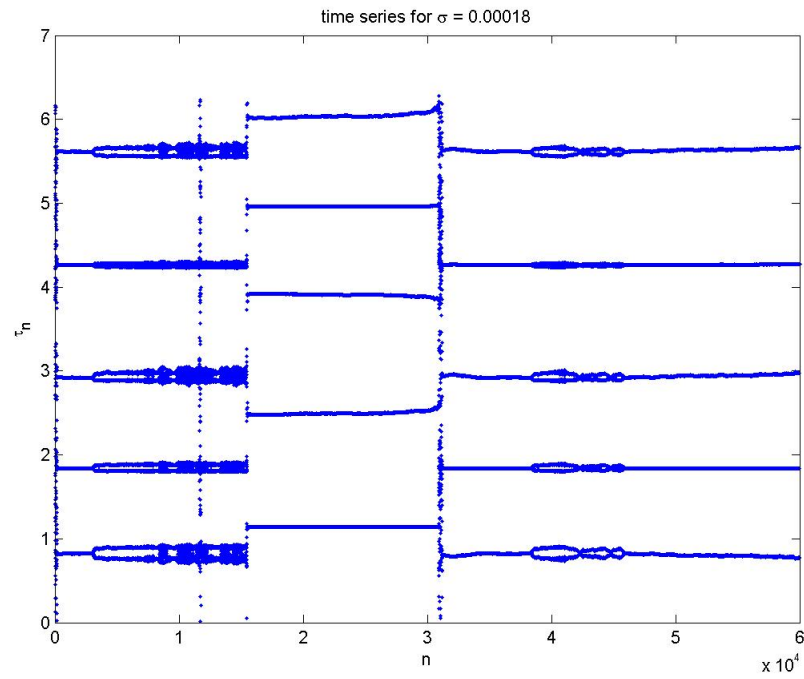


Figure 4.11. A time series of the τ_n for $\alpha = 1.985$ at noise level $\sigma = 0.0018$.

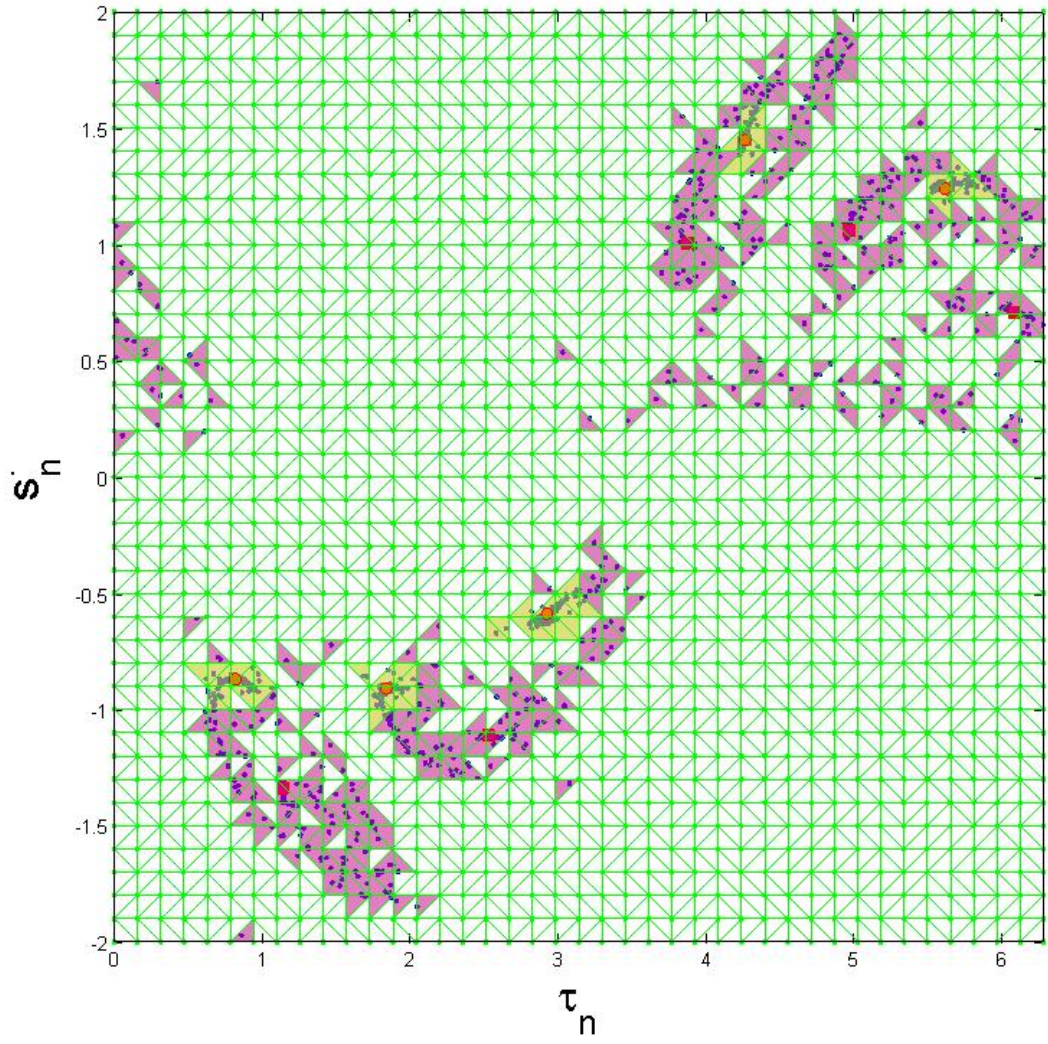


Figure 4.12. A partition of an orbit generated from the gearbox model with normal noise, $\sigma = 0.0018$.

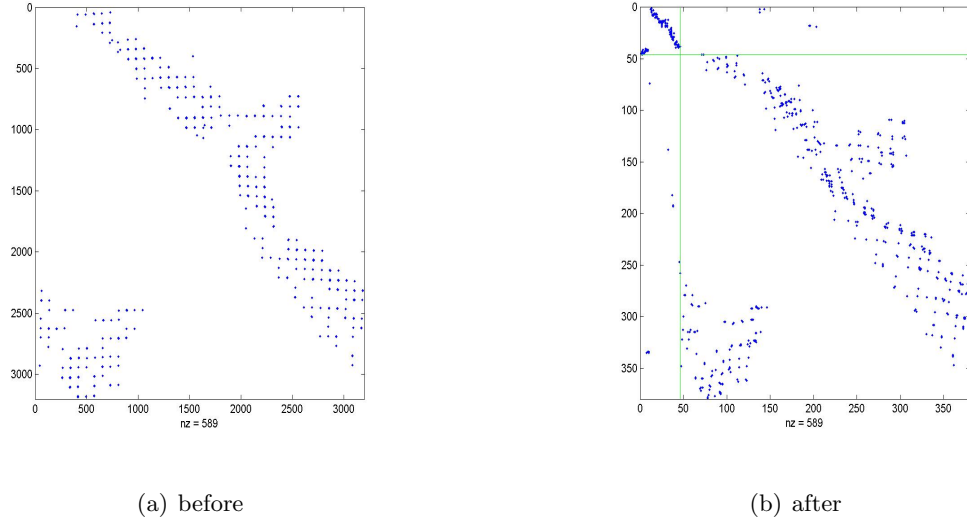


Figure 4.13. Transition matrix before and after sorting of the gearbox model. Notice in the sorted matrix that two distinct basins become clear as those triangles indexed to associate to the diagonal blocks, and the regions causing hopping are clear as those few off diagonal elements. Each of the two basins, the smaller and larger blocks, are the pink and yellow colored regions in Figure 4.12.

where the stream function ψ is given numerically as shown in Figure 4.14 and η_x and η_y are the normal noises with zero mean. We then generate a time-series data based on the time-T map of Eq. (4.35) for a normal noise with zero mean and standard deviation $\sigma = 0.001$ as shown in Figure 4.15. One may notice bursting in the time series, which suggests transitions between the almost invariant sets.

To investigate how the basin structure changes as the standard variation of noise is varied, we compare the result obtained from the modularity method for four cases of $\sigma = 0.001, 0.005, 0.01$, and 0.05 as shown in Figure 4.16 and 4.17. In all four cases we start with an initial point $(x, y) = (-0.2, 0.6)$. Note that the regions with no color do not contain any point in a test orbit and so they are not used in a process of generating a graph. One may observe from Figure 4.16 that the bandwidth of the Ulam-Galerkin matrix before sorting becomes larger as σ increases. Moreover, the modularity measure as defined in Eq. (4.5) becomes lower when noise becomes larger ($Q = 0.7267, 0.7183, 0.6968$, and 0.6361 for $\sigma = 0.001, 0.005, 0.01$, and 0.05 , respectively). Recall that the higher the value of the

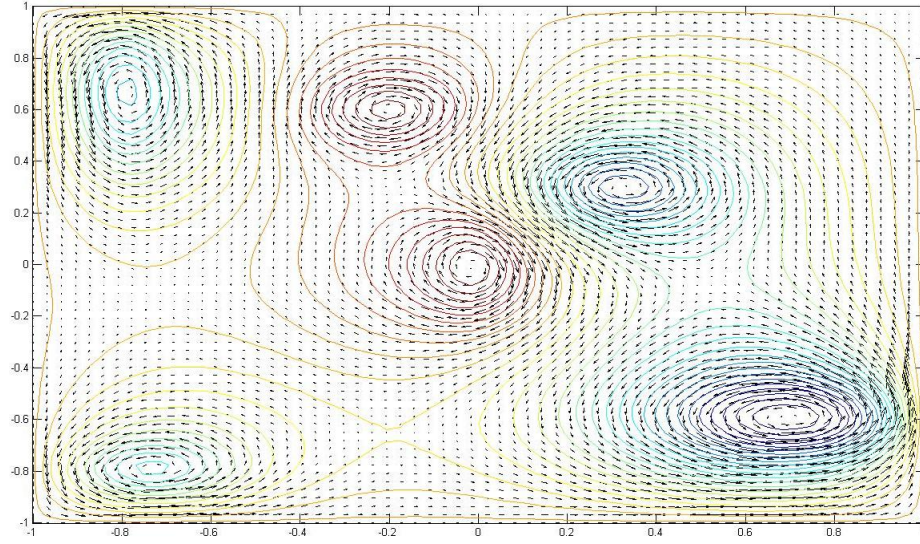


Figure 4.14. The velocity field for the noiseless case of Eq. (4.35) is computed from the above streamfunction by the relation $(dx/dt, dy/dt) = \nabla \times \psi$.

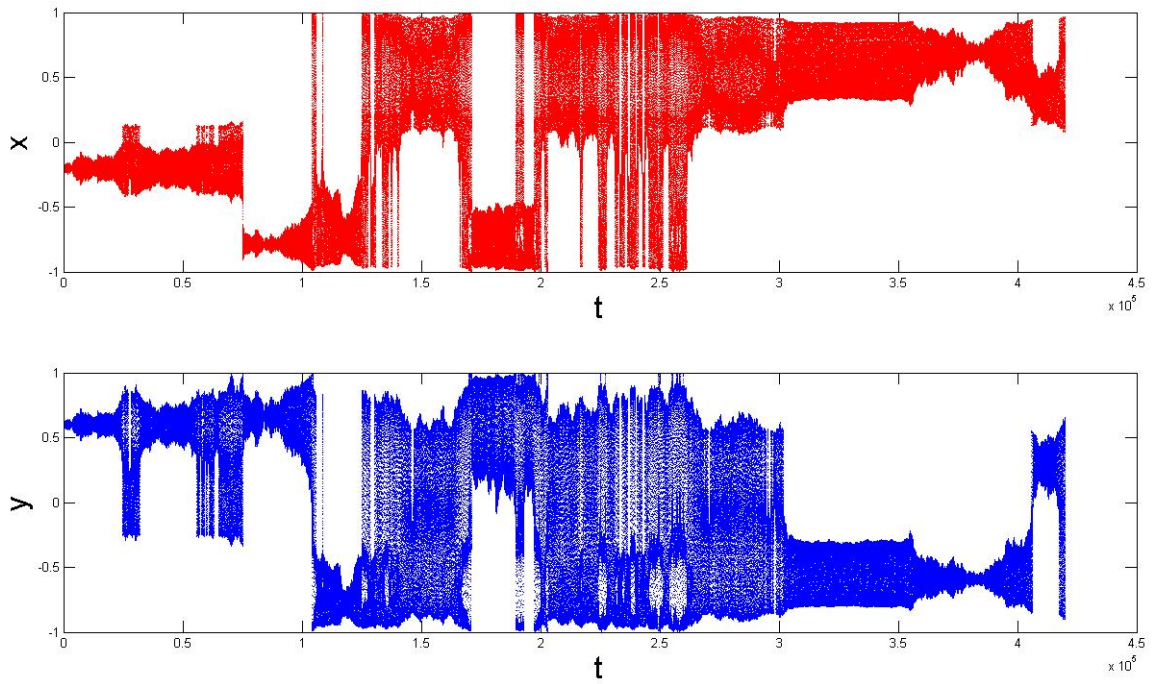


Figure 4.15. A test orbit of the time-T map of Eq. (4.35) for a normal noise with standard deviation $\sigma = 0.001$.

modularity measure, the stronger the community structure (the lower value of the ratio of the inter-community edges and the within-community edges).

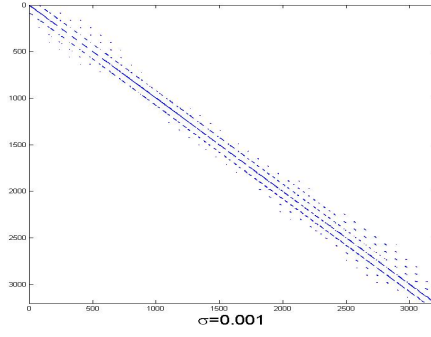
4.5 On Computational Complexity

As featured in our work [30], we briefly discuss the computational complexity of the methods described in this chapter. Although increasing the number of box coverings, and hence the number of vertices in the graph, can make a graph partition that reveal more refined structure of basin, there exists a necessary number of box coverings to discover a basin structure. This minimum requirement can be related to the Lyapunov exponent of a given dynamical system from which the data comes. This means that refining a covering of a phase space beyond the necessary requirement will not reveal more structure of basins, but any over-refinement still requires more computer storage and more computer work. The following analysis assumes the zero noise, and so this would seem to imply it may be relevant at best for only small noise profiles. We remind however that there does exist a concept of stretching and Lyapunov exponent for a stochastic system [66], and so we offer the following as a heuristic model which is suggestive of the necessary refinement of the grid to gain a given knowledge of the system's evolution. We find that these considerations give a useful starting point when the noise profile is not overwhelming.

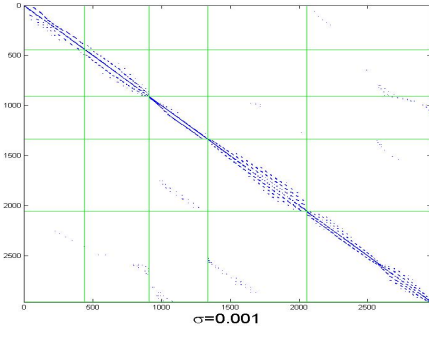
We use the Baker's map as an example to describe the trade-off between the number of box coverings and the Lyapunov exponent. Recall that a Baker's map on the unit square $[0, 1] \times [0, 1]$ maybe defined by [67],

$$\begin{aligned} x_{n+1} &= \begin{cases} \lambda_1 x_n & \text{if } y_n < \alpha, \\ (1 - \lambda_2) + \lambda_2 x_n & \text{if } y_n > \alpha, \end{cases} \\ y_{n+1} &= \begin{cases} y_n / \alpha & \text{if } y_n < \alpha, \\ (y_n - \alpha) / (1 - \alpha) & \text{if } y_n > \alpha, \end{cases} \end{aligned} \tag{4.36}$$

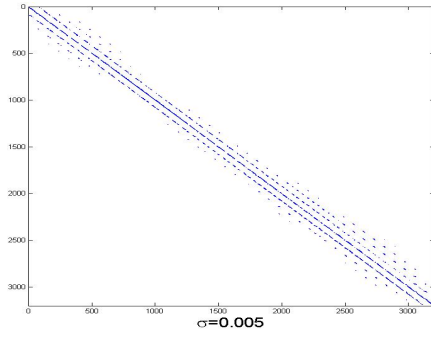
where $\lambda_1 + \lambda_2 < 1$ and $0 < \alpha < 1$. The forward-time invariant set of this map is a Cantor set of parallel vertical lines. Consider learning these structures with box covering methods,



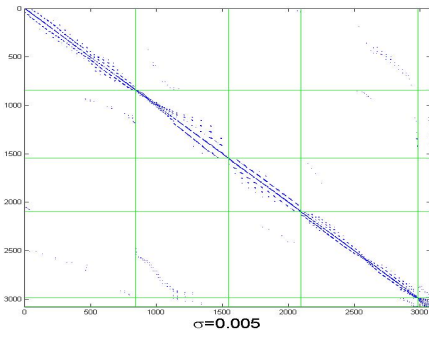
(a)



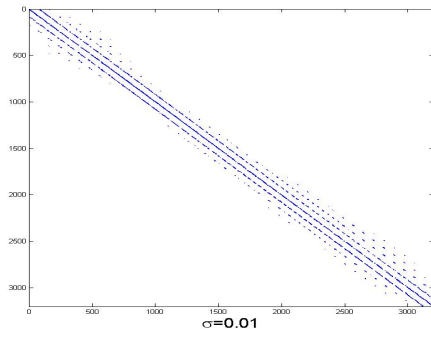
(b)



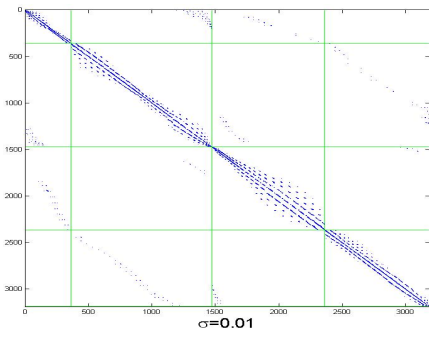
(c)



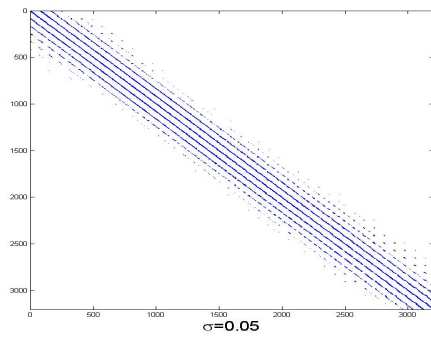
(d)



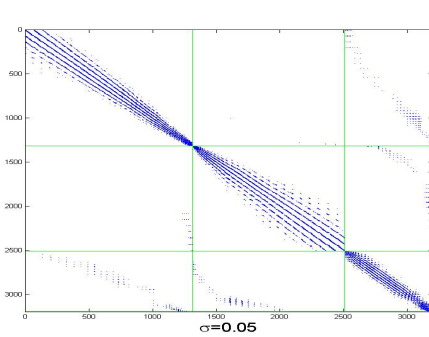
(e)



(f)



(g)



(h)

Figure 4.16. Transition matrix before and after sorting of the model Eq. (4.35). Notice decreasing the number of basins as the noise increases.

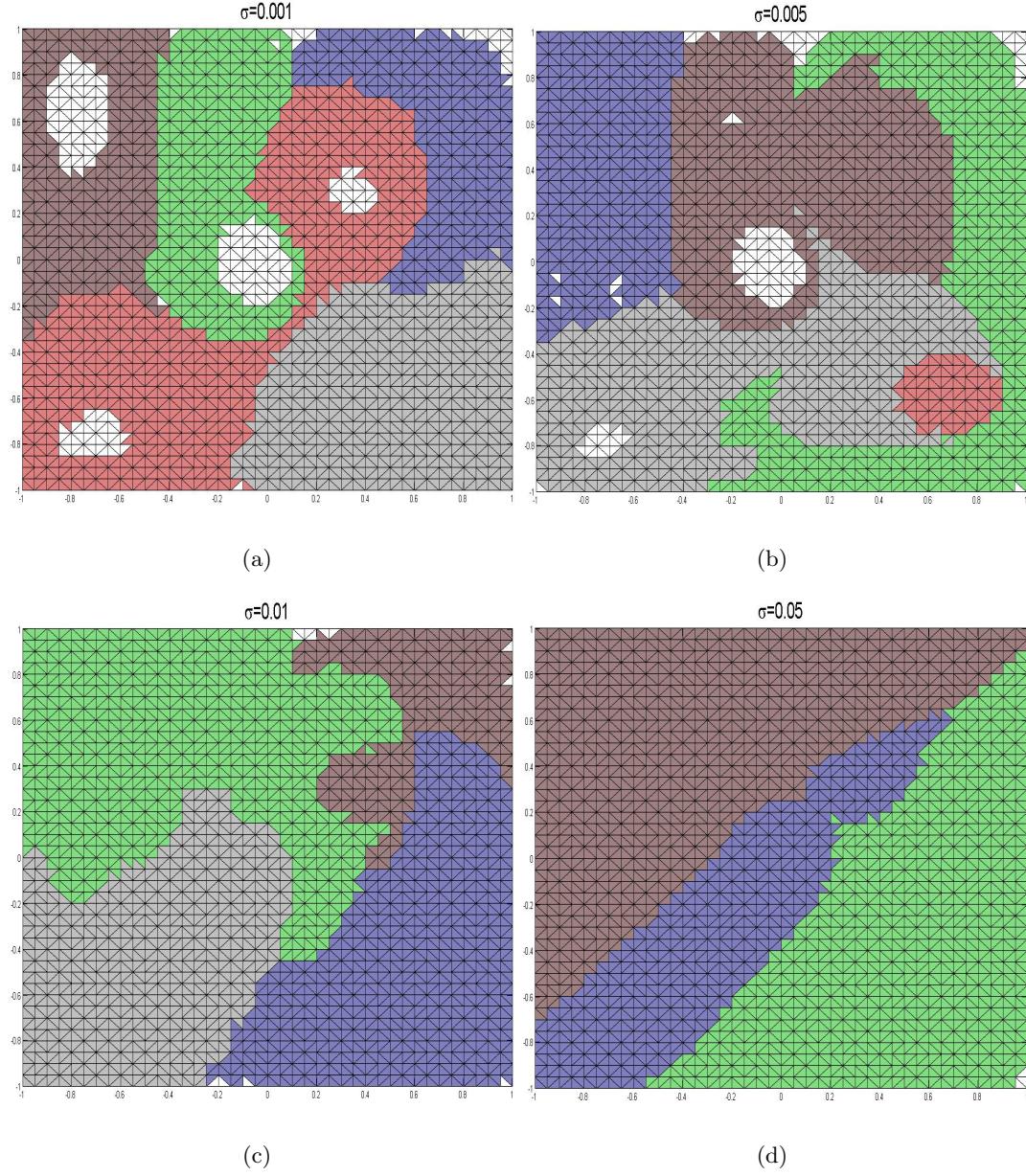


Figure 4.17. The phase space partition of the duffing-like model with various values of σ using the graph modularity method. The values of the modularity measure as defined in Eq. (4.5) in each case are $Q = 0.7267, 0.7183, 0.6968$, and 0.631 for $\sigma = 0.001, 0.005, 0.01$ and 0.05 , respectively. Similarly, the basin structure is changing with σ .

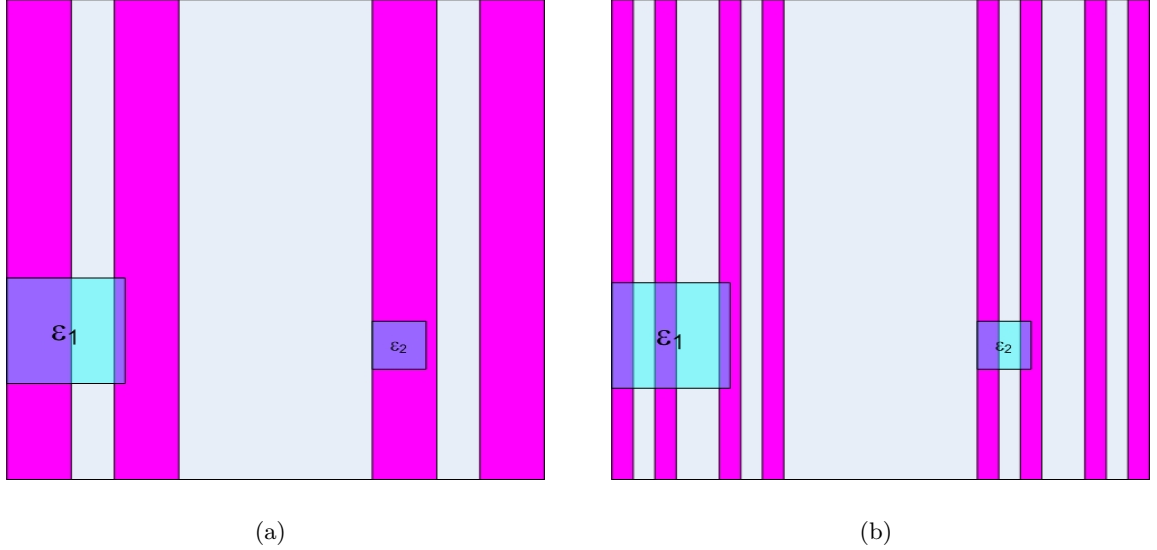


Figure 4.18. Two iterations of the Baker map with $\lambda_1 = \lambda_2$. (a) All stripes after $n = 2$. A square box of size ϵ_1 cannot resolve a structure finer than ϵ_1 . But we see that a smaller square box of size ϵ_2 is able to resolve a stripe of width up to ϵ_2 . (b) Here, however, the box of size ϵ_2 cannot resolve the stripe left in the unit box after $n = 3$.

and what can be discovered by such finite computations. Suppose that we want to cover all parallel stripes left in the unit box *after n iterations* by square boxes of size ϵ so that two stripes are not contained in the same box, see Figure 4.18. The size of the square box required to do this is

$$\epsilon(n) = \min \{\lambda_1^n, \lambda_2^n\}. \quad (4.37)$$

Therefore, the required number of boxes to reveal the fractal structure of these parallel stripes is

$$N(\epsilon) \geq \epsilon(n)^{-2}, \quad (4.38)$$

and $d = 2$ is the dimension of the embedding phase space. Using more boxes than the minimum requirement (4.38) would only increase required storage and work but does not give us a better knowledge of the stripes left in the unit box. Thus we see that there is a trade-off between Lyapunov exponents, λ_i , time, n , box size, ϵ , the size of the memory requirements and size of the resulting digraph, $N(\epsilon)$, and the dimension of the underlying space, d .

Chapter 5

LCS in 2-D dynamical systems

This chapter is dedicated to the analysis and applications of the finite-time Lyapunov exponents, which are used to determine their associated Lagrangian coherent structure (LCS). It was demonstrated by Haller [15, 16] that the finite-time Lyapunov exponent can be used to capture the material lines advected as a set by the flow. Thus, these material lines approximate the hyperbolic invariant manifolds of the system and form a building block to understand the global geometry of transport and mixing of the flow.

5.1 Lyapunov exponents

Before we discuss the notion of the finite-time Lyapunov exponent in the subsequent section, we review here the traditional asymptotic notion of the Lyapunov exponent, which provides a long-term rate of expansion or contraction of tangent vectors. We first review the definitions for a diffeomorphism and then we will turn to those for the flow later in this section.

Definition 5.1. Let $f : \mathbb{M} \rightarrow \mathbb{M}$ be a C^2 diffeomorphism (that is, a twice differentiable function with continuous second derivative) on a smooth manifold \mathbb{M} with an invariant measure μ . Let $\|\cdot\|$ be the norm on the collection of tangent vectors at $x \in \mathbb{M}$, denoted by $T_x\mathbb{M}$. For each $x \in \mathbb{M}$ and $v \in T_x\mathbb{M}$, the Lyapunov exponent $\chi(x, v)$ at point x in direction

of v is given by

$$\chi(x, v) = \lim_{k \rightarrow \infty} \frac{1}{k} \log \|Df_x^k v\| \quad (5.1)$$

whenever the limit exists.

Note that for an arbitrary choice of x the limit may not exist, but the supremum limit always exists since the norm $\|Df_x^k v\|$ is bounded for a C^2 diffeomorphism. Since the largest growth is often of interest, it would be possible to use the limsup to define the Lyapunov exponent as:

$$\chi(x, v) = \limsup_{k \rightarrow \infty} \frac{1}{k} \log \|Df_x^k v\|. \quad (5.2)$$

However, we will assume that the limit exists and restrict to the definition given in Eq. (5.1), which is often used in most applied dynamical system literature. Nevertheless, the Multiplicative Ergodic Theorem of Oseledec gives conditions under which the limit in Eq. (5.1) exists for all tangent vectors $v \in T_x M$ at almost all points $x \in \mathbb{M}$.

Theorem 5.1. (Multiplicative Ergodic Theorem, See Robinson [32]) Let $f : \mathbb{M} \rightarrow \mathbb{M}$ be a C^2 diffeomorphism on a compact manifold \mathbb{M} of dimension m with an invariant measure μ defined on a Borel σ -algebra \mathcal{B} . Then there is an invariant set $B_f \in \mathcal{B}$ of full measure, $\mu(B_f) = 1$, such that the following properties are satisfied:

(a) For each $x \in B_f$, there exists $l(x)$ distinct values $\chi_1(x) < \cdots < \chi_{l(x)}(x)$ such that the tangent space at x , $T_x M$, can be decomposed by the bundle subspaces $T_x M = V_x^1 \oplus \cdots \oplus V_x^{l(x)}$ with $\sum_{j=1}^{l(x)} \dim(V_x^j) = m$. Then, for $v \in (V_x^j \oplus \cdots \oplus V_x^{l(x)}) \setminus (V_x^{j-1} \oplus \cdots \oplus V_x^{l(x)})$, the limit defining $\chi(x, v)$ exists and $\chi_j(x) = \chi(x, v)$ for all such v , that is,

$$\chi_j(x) = \lim_{k \rightarrow \infty} \frac{1}{k} \log \|Df_x^k v\|. \quad (5.3)$$

In addition, the bundle subspaces are invariant in the sense that $Df_x V_x^j = V_x^j$ for all $1 \leq j \leq l(x)$.

(b) If $x \in B_f$, the exponents satisfy $-\infty \leq \chi_1(x) < \cdots < \chi_{l(x)}(x)$. For $1 \leq j \leq m$, the function $\chi_j(\cdot)$ is measurable for each $x \in B_f$ and invariant, $\chi_j \circ f = \chi_j$.

Note that the real numbers $\chi_j(x)$ in the above theorem are the Lyapunov exponents at x , which are the same for all tangent vectors under the above conditions. As a consequence of

the condition (a), there are at most m distinct values of $\chi(x, v)$ for each point $x \in \mathbb{M} \subset \mathbb{R}^m$ and if one arbitrarily choose the initial tangent vector so that it is a mixture of all Lyapunov basis components (i.e., $v_x^{l(x)}$), we will have $\chi(x, v) = \chi_{l(x)}(x)$. Also, the second property in the theorem gives that the lyapunov exponents at x are constant along an orbit, which can also be seen directly from the definition of the Lyapunov exponent (5.1). Therefore, if the diffeomorphism f is ergodic for an invariant measure μ , the Lyapunov exponents $\chi_j(x)$ for each x are the same for μ -almost everywhere. Hence, there is no need to specify the initial point x for the Lyapunov exponent of an ergodic diffeomorphism that satisfies the conditions in the Multiplicative Ergodic Theorem.

As an example of an ergodic measure, consider the natural measure on a basin of attraction. Therefore, one may refer to the Lyapunov exponent of an attractor without specifying an initial condition. In addition, the attractor is normally defined to be *chaotic* if its Lyapunov exponent is positive since it implies sensitivity of initial conditions for the attractor. Nonetheless, the positive Lyapunov exponent does not always imply the sensitivity of initial conditions for an arbitrary invariant set. For example, the separatrix, homoclinic loop of the Hamiltonian system with $H(x_1, x_2) = 0.5(x_2^2 - x_1^2) + ax^3$ for some $a > 0$ has one negative Lyapunov exponent in the tangent direction to the separatrix and one positive Lyapunov exponent in the transverse direction. However, the separatrix does not have sensitive dependence on initial conditions despite of the existence of a positive Lyapunov exponent. Despite this contradictory example, one has to keep in mind that the homoclinic loop is non-generic, and it has a Lebesgue measure of zero. In general, the Lyapunov exponent is a useful tool to measure chaos.

Another alternative definition of the Lyapunov exponent directly relates to the change in length of the tangent vectors and it will be used to define the finite-time Lyapunov exponent in the following section. Remember that Df_x is the linearization of the map f at point x and since $Df \in L(M, M)$ we may consider Df_x as a square matrix on the manifold. Therefore, the length of the tangent vector v after one iteration by the linear map Df_x^k is

$$\|Df_x^k v\| = (v^T \Lambda_k(x) v)^{1/2}, \quad (5.4)$$

where $\Lambda_k(x) = (Df_x^k)^T Df_x^k$ is symmetric and positive definite. $\Lambda_k(x)$ measures how much

the lengths of v are changed by Df_x^k . It can be shown that for almost all x , the limit

$$\Lambda_x = \lim_{k \rightarrow \infty} [\Lambda_k(x)]^{1/2k}, \quad (5.5)$$

exists. Then, the Lyapunov exponent $\chi_j(x)$ is

$$\chi_j(x) = \lambda_j(\Lambda_x), \quad (5.6)$$

where $\lambda_j(\Lambda_x)$ denotes the j -th smallest eigenvalue of the matrix Λ_x with the multiplicity of $\dim(V_x^j)$ as defined in the Multiplicative Ergodic Theorem.

5.2 Finite-time Lyapunov exponents and LCS

We have discussed in the preceding section the asymptotic notions of the Lyapunov exponent. In the past decade, a number of research studies have focused on the notion of the Lyapunov exponent when the vector field is known only in a finite-time interval. For examples, the finite-time vector fields, which are also discrete in space, might be numerically constructed from a numerical solution of the Navier-Stokes equation, a particle image velocimetry (PIV) data or motion estimation from a satellite image sequence (See Chapter 6). Therefore, several asymptotic notions such as Lyapunov exponent, hyperbolic trajectory and stable/unstable manifold, as described in Chapter 2 become questionable. A number of concepts and methods in hundreds of papers have been developed for this purpose in the last two decades [36, 14, 15, 17, 18, 10, 37, 38]. However, in the context of the finite-time, discrete velocity fields, an approximation of the (finite-time) hyperbolic or distinguished hyperbolic trajectories becomes difficult and demanding in terms of computational effort.

A different concept is to directly resort to the behavior of the flow of the invariant manifolds. In particular, the finite-time stable (unstable) manifold of a hyperbolic trajectory behaves like a repelling (attracting) material surface in an extended phase space [15, 68]. Therefore, we may directly detect the finite-time stable (unstable) invariant structure by computing the expansion rate in a forward (backward) time. Normally, these invariant structures have one dimension less than the dimension of the phase space and they form the boundary between regions of different dynamical or statistical properties. Note that

in fluid dynamics, coherent structures can be viewed as sets of fluid regions with different statistical properties. Hence, these structures are also called the *Lagrangian coherent structure*.

Since we will focus on only the case of 2-D velocity fields and hence we consider a nonautonomous velocity field of the form

$$\begin{aligned}\dot{x}(t, t_0, x_0) &= v(x(t_0, t, x_0), t) \\ x(t_0, t_0, x_0) &= x_0\end{aligned}\tag{5.7}$$

where $x \in \mathbb{M} \subset \mathbb{R}^2 \times \mathbb{R}$ and $v(x, t) : \mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}^2$ is C^2 in space \mathbb{M} and C^1 in time. Here, we write the vector field in the extended phase space by redefining time as a new dependent variable. For brevity, we shorten the notation $x(t_0, t_0, x_0)$ by using $x(t_0)$ instead for the rest of this chapter. In this extended phase space, we can assume the existence and uniqueness of the solution of (5.7) and hence the solution of (5.7) satisfies the so-called *cocycle property*:

$$x(t_2, t_1, x_0) = x(t_2, t_1, x(t_1, t_0, x_0)), \quad \forall x_0 \in \mathbb{R}^n, \text{ and } t_0 \leq t_1 \leq t_2.\tag{5.8}$$

Then, we can integrate the velocity field to obtain the semigroup of transformation

$$\phi_\tau : x(t_0, x_0) \mapsto x(t_0 + \tau, t_0, x_0).\tag{5.9}$$

We have seen from (5.4) that the Lyapunov exponent measures the expansion rate of a tangent vector at a point x . Now, we would like to describe its finite time version for the semigroup of transformation (5.9) associated to the nonautonomous system (5.7). In analogy to (5.4) we can define a symmetric matrix

$$\Lambda_\tau \equiv \Lambda_\tau(x(t_0)) = (D\phi_\tau(x(t_0)))^T D\phi_\tau(x(t_0))\tag{5.10}$$

as a finite-time version of (5.4), which measures the change in length of the small tangent vector at the initial condition $x(t_0)$ in a finite time interval $[t_0, t_0 + \tau]$. For an nonautonomous velocity field, this matrix depends upon the initial point $x(t_0)$, the initial time t_0 , and the integrating of the flow map τ . Note that in the language of fluid dynamics, the matrix

Λ_τ is called the Cauchy-Green deformation tensor. Therefore, we can define the finite-time Lyapunov exponent (FTLE) of the initial condition $x(t_0)$ as

$$\sigma_\tau(\mathbf{x}(t)) = \frac{1}{|2\tau|} \ln \sqrt{\lambda_{\max}(\tau)}, \quad (5.11)$$

where $\lambda_{\max}(\tau)$ is the largest eigenvalue of Λ_τ (the largest singular value of $D\phi_\tau(x(t_0))$).

When computing in forward time ($\tau > 0$), the ridge of the FTLE field reveals the so-called repelling Lagrangian coherent structure (LCS), which has a similar feature to the stable manifold of a hyperbolic trajectory, see Shadden [17, 18]. Similarly, the second-derivative ridge of the FTLE computed with negative integrating time ($\tau < 0$) locates the attracting Lagrangian coherent structure. It was analytically and experimentally verified by Shadden [17, 18] that the LCS defined as the second-derivative ridge of the FTLE field is nearly Lagrangian in a sense that the particle flux across the LCS becomes negligible as the integrating time τ increases provided that the “true” LCS associated with the vector field is hyperbolic for all time. However, the LCS obtained from a finite-time data set is likely to be hyperbolic only for a finite time and so if the integrating time τ increases beyond the hyperbolic time of the trajectory, which is not known a priori, the resulting LCS may not exhibit the Lagrangian property. Here, the second-derivative ridge is defined as a parameterized curve $c : (a, b) \rightarrow \mathbb{M}$ that satisfies two conditions. First, the derivative of the curve $c'(s)$ has to be parallel to the vector $\nabla\sigma_\tau(c(s))$. This is to force the tangent line of the curve to be oriented in the direction of the largest variation of the FTLE field. Secondly, the direction of the steepest descent is that of the normal vector to the curve $c(s)$, which can be expressed by

$$n^T H(\sigma_\tau) n = \min_{\|v\|=1} v^T H(\sigma_\tau) v < 0, \quad (5.12)$$

where n is the unit normal vector to the ridge curve $c(s)$ and $H(\sigma_\tau)$ is the Hessian matrix of the FTLE field:

$$H(\sigma_\tau) = \begin{bmatrix} \frac{\partial^2 \sigma_\tau}{\partial x^2} & \frac{\partial^2 \sigma_\tau}{\partial x \partial y} \\ \frac{\partial^2 \sigma_\tau}{\partial y \partial x} & \frac{\partial^2 \sigma_\tau}{\partial y^2} \end{bmatrix}. \quad (5.13)$$

5.3 LCS analysis of a mixing device

5.3.1 Experimental setup

Details of the experimental methods can be found in Bohl [69]. Figure 1 shows a schematic of the experimental apparatus. A clear acrylic cylindrical flat bottomed container was constructed with an inside radius of $r_w=6.93$ cm. A flat stainless steel blade was placed in the cylinder such that the long axis of the blade was parallel to the z-axis of the container. The blade had a width of $r_b = 2.99$ cm ($0.43r_w$), a thickness of 0.14 cm ($0.02r_w$) and could be placed at various locations (i.e. offsets) with respect to the wall. The blade was rotated about its long axis by a DC motor to drive the fluid motion. The data presented in this paper used glycerin as the working fluid. The Reynolds number was $Re = 9$, with the Reynolds number defined as $Re = 2fr_w^2/\nu$ where f is the rotational frequency of the mixing blade. The geometry was chosen as a simplified first order model of a batch mixer to provide simple, well defined boundary conditions that could be used to aid model development and provide validation data for the developed computational tools.

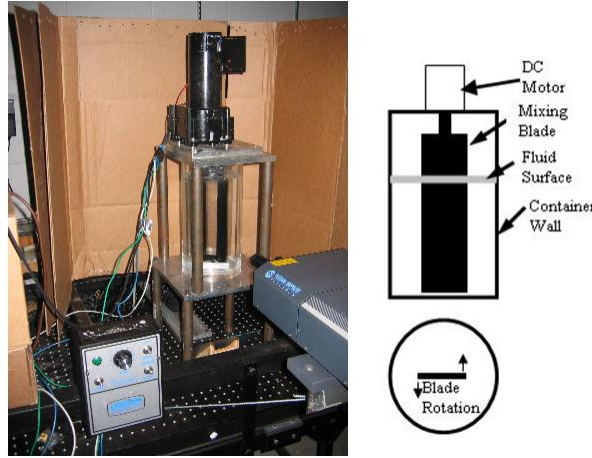


Figure 5.1. Experimental mixing apparatus.

The 2-D velocity fields were measured using Particle Image Velocimetry (PIV). Briefly, PIV relies on seed particles in a fluid that scatter light from a thin laser sheet created in the plane of interest. These particles may exist within the fluid naturally or be added to the fluid before measurements are made. The particles in the flow are imaged twice

with a short, known delay between images. The displacement of the particles, or groups of particles, is determined via pattern matching techniques and a local velocity vector is calculated by dividing the displacement by the time between images. The velocity can be determined at discrete points over the entire image and the 2-D velocity field in the plane of the laser sheet can be assembled. Other kinematic quantities, such as vorticity, strain and stress, can then be calculated using the planar velocity field. A detailed discussion of PIV can be found in Adrian [70].

In the current work, the working fluid was seeded with 10 micron silver coated glass spheres. Light was provided by a 4 Watt Spectra Physics Argon Ion laser. The mixing blades were painted with flat black paint to reduce the reflection of the laser sheet in the measurement region. The laser light was pulsed using a NM Laser Products fast mechanical shutter. This shutter had a minimum closed-open-closed cycle time of 1 msec. This was short enough that the imaged particles did not experience blurring during the exposure time. Images were capture using a Cooke Corporation Sensicam-QE CCD camera (8 bit, 1376×1040 pixels). The flow was viewed up through the clear bottom of the container to provide data in a plane perpendicular to the long axis of the mixing blade (i.e. the $r - \theta$ plane).

The displacement of groups of particles was determined using the direct correlation technique described in Gendrich and Koochesfahani [71]. Each FOV was approximately $8.7 \text{ cm} \times 12 \text{ cm}$ in size (0.0087 cm/pixel). Delay times between the images were chosen to give maximum displacements of nominally 10 pixels and the source windows used in the correlation technique were 45×45 pixels with a 50% overlap. The error in the velocity measurements is limited by the ability to measure the displacement of the particles between images. The correlation technique used to process the data in this work is well documented and has a 95% uncertainly level of 0.1 pixel which corresponds to an uncertainty in the instantaneous velocity measurements of 0.08 cm/sec .

The flow field under investigation was periodic in nature, with the periodicity linked to the motion of the mixing blade, which allowed the data to be phase averaged with respect to the motion of the blade. The blade phase, was defined such that $1/2$ rotation was given

for $\phi = 0$ to 1 (i.e. ϕ ranged from 0 to 2 for a complete blade rotation). This allowed data from multiple experiments and multiple Fields of View (FOV) to be combined to form a single data set. Phase averaging and the combining of multiple data sets allowed multiple individual measurements to be averaged for a single measurement location which reduces the estimated error in the presented data by a factor of 2 to 0.04 cm/sec. Local vorticity was calculated using a second order finite difference method using the spatially nearest velocity measurements. Vorticity error values were estimated to be 0.38 and 0.16 sec^{-1} for the high and low frequency cases respectively. All data presented in this work are from PIV measurements taken at the mid-height of the fluid column. Data not shown here indicated that the flow field was independent of the z-location over the middle half of the mixing blade.

Representative phase averaged velocity and corresponding vorticity fields from the PIV data are shown in Figure 5.2 for the centered blade case. Of note in this figure is the strong vortex that forms near the tip of the blade and travels with the blade as it sweeps through the fluid. A second region of vorticity forms along the blade surface and along the container bounding wall. A region of reverse flow is observed between the tip vortex and the bounding wall. Detailed discussion of this flow field can be found in Bohl [69].

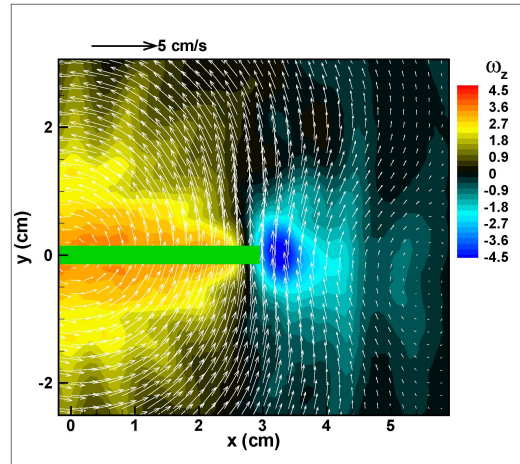


Figure 5.2. Representative phase averaged velocity and vorticity fields from PIV data. Case shown is from the centered data.

5.3.2 Transport and Mixing in the Mixers

In this section we demonstrate an analysis of transport and mixing for our rotating flat plate mixer by approximating the Lagrangian coherent structure (LCS) based on the FTLE. The PIV output provides discrete measurements of the velocity field over a plane at finite times which can be used to compute the FTLE based on Eq. (5.11). However, an interpolation is needed to integrate points where the velocity field is not given in the PIV data. Furthermore, if the time step for the integration is smaller than the duration of two consecutive experimental time steps, we have to assume that the variation of velocity fields between two consecutive velocity data is small so that an interpolation in the time domain is also suitable. Conditions for the existence of finite-time hyperbolic trajectory can be found in Haller and Poje [14].

Figure 5.3 shows snapshots of the forward-time FTLE of the mixer with the blade positioned at the center. We set the integrating time to half of a full rotation, $\tau = 2.1342$ seconds. Note that one period of the vector field here is half of a full revolution of the blade since there is no distinction between the two tips of the blade. Numerically, at each time t , we integrate samples of grid points by a fourth-order Runge-Kutta method to find their positions at time $t + \tau$. The cubic method is used for an interpolation during the integration. Then, we compute the spatial gradient of the flow by a central-difference scheme at each initial grid points. The FTLE at time t for each initial point can be evaluated according to Eq. (5.11). The SVD is used to compute the eigenvalues of Eq. (5.10).

The LCS of the flow can be defined as ridges of the FTLE field as shown in Figures 5.3 and 5.4. We observe the Lagrangian property of the LCS by tracing the trajectory of particles straddling the LCS. One can see that these particles do not transverse the LCS. In order to understand the motion differences between the two regions separated by the LCS, we use the particle tracers shown in Figure 5.5 to visualize the geometry of the flow. This further suggests that the repelling LCS divide the space mainly into two regions; the region surrounded by the repelling LCSs experiences a very low degree of mixing whereas a higher degree of mixing occurs in the “outside” region. Furthermore, we compute the

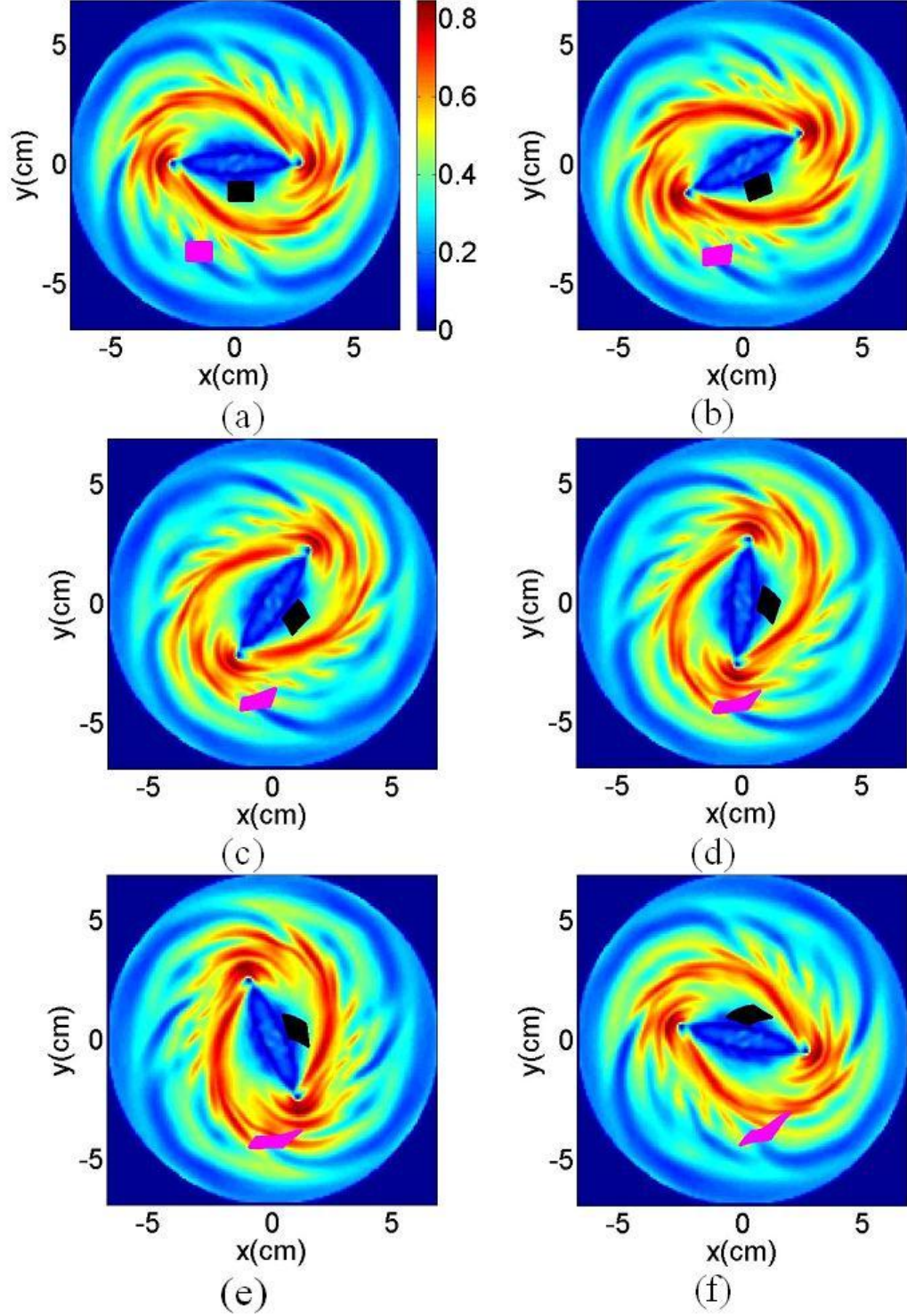


Figure 5.3. The forward-time FTLE field of the mixer with the centered blade is computed at different time t with the integrating time $\tau = 64\Delta t$, where $\Delta t = 0.033$ seconds. From (a) to (f), the FTLE is computed at time $t = \Delta t$, $t = 10\Delta t$, $t = 20\Delta t$, $t = 30\Delta t$, $t = 40\Delta t$ and $t = 60\Delta t$, respectively. The initial particles are placed to straddle the LCS and they are advected to demonstrate that the resulted LCSs serve as the transport barrier.

backward-time FTLE of the flow of the same mixer. Recall that the backward-time FTLE reveals to us the attracting LCSs, which is a finite-time version of an unstable manifold of a steady flow. Therefore, the particles tend to accumulate along the attracting LCSs as observed in Figure 5.4. With the aid of these forward and backward time FTLE, we see that the stable and unstable of periodic points forms a trapping region (Figure 5.6) that traps fluid particles inside itself and hence mixing has a potential to occur only between the particles in the trapping region. Notice that the boundary of the trapping region moves along with the rotation of the blade tips and also that the center point of the mixer is a hyperbolic fixed point that is responsible for mixing inside the trapping region.

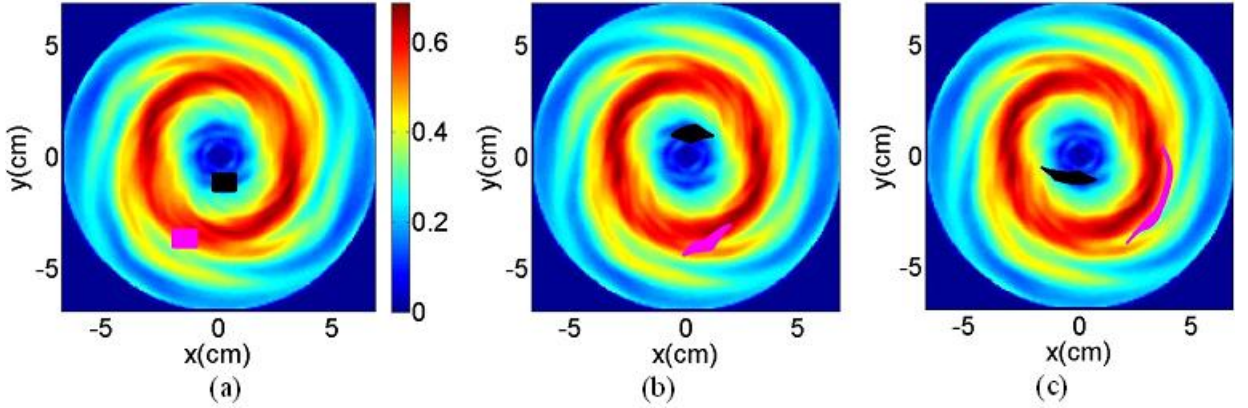


Figure 5.4. The backward-time FTLE field of the mixer with the centered blade is computed at different time t with the integrating time $\tau = 64\Delta t$ as in Figure 5.3. From (a) to (c), the FTLE is computed at time $t = \Delta t$, $t = 60\Delta t$ and $t = \tau + 60\Delta t$, respectively. Here we observe that the initial particles colored by the magenta accumulated along the ridge of the FTLE field. Thus the ridge of the backward-time FTLE field may be regarded as an unstable manifold.

We point out that the velocity field of our mixer may be considered as a periodically perturbed Hamiltonian system. For such a system, it is well-known that the stable and unstable manifolds have an infinite length in general [13]. Therefore, we should expect the resulting LCS to be revealed more as the integrating time increases. However, this requires higher resolutions of sample points and hence a longer computational time, see Figure 5.7. One can see stretching and folding of “stripes” in the mixer, which suggests a need of a larger number of grid points to compute the FTLE as the integrating time is increased.

We observe that FTLE fields for the mixer with the offset blade, Figure 5.8, reflects

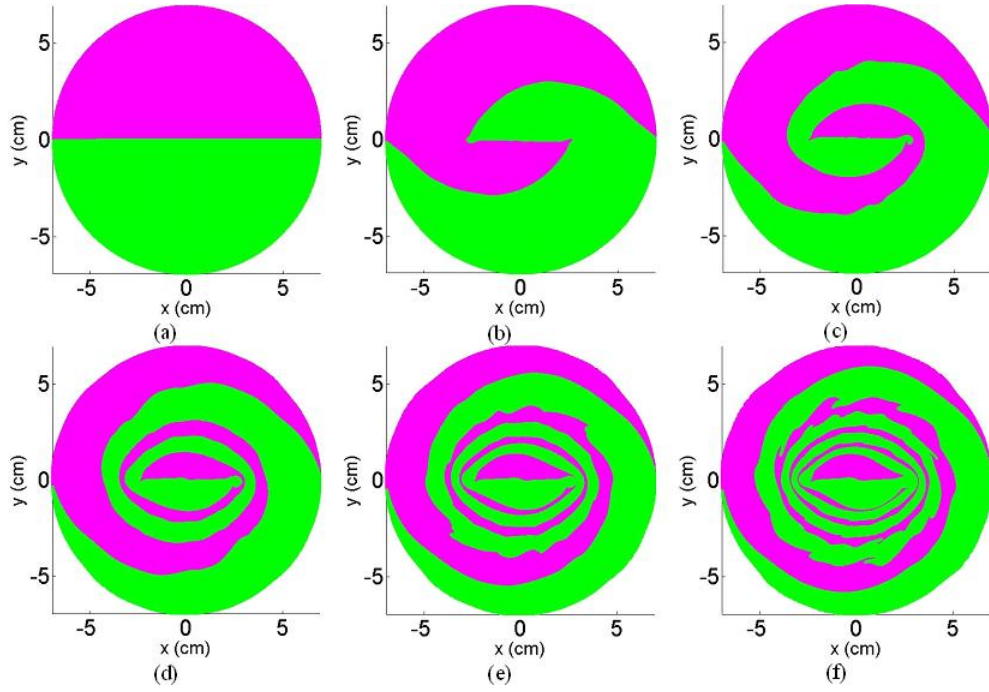


Figure 5.5. Snapshots of particle tracers for the mixer with the centered blade. Two sets are particles, A (green) and B (magenta) are initially separated to occupy half of the physical space as shown (a). The initial particle distribution is then allowed to change as dictated by the flow field for 1,2,3,4, and 5 periods, (b)-(f), respectively.

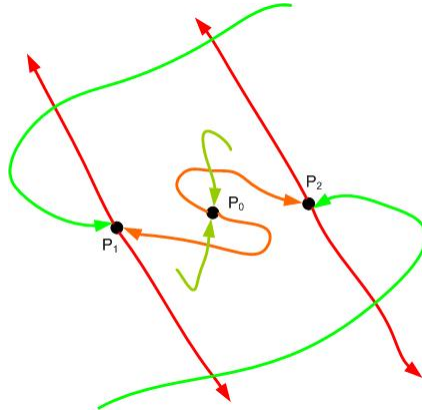


Figure 5.6. A heuristic rendering of a stable and unstable heteroclinic tangle made of the mixing system inferred from the observed time-dependent LCS structures. The periodic points P_1 and P_2 correspond to the blade tips. The forward and backward time FTLE fields in Figure 5.3 and 5.4 suggest that the transverse intersection by the stable and unstable manifolds of P_1 and P_2 generate a trapping region that bounds the trajectories of particles within this cell. Note that the mixing inside the trapping region is a result of the hyperbolic fixed point at the center of the mixer.

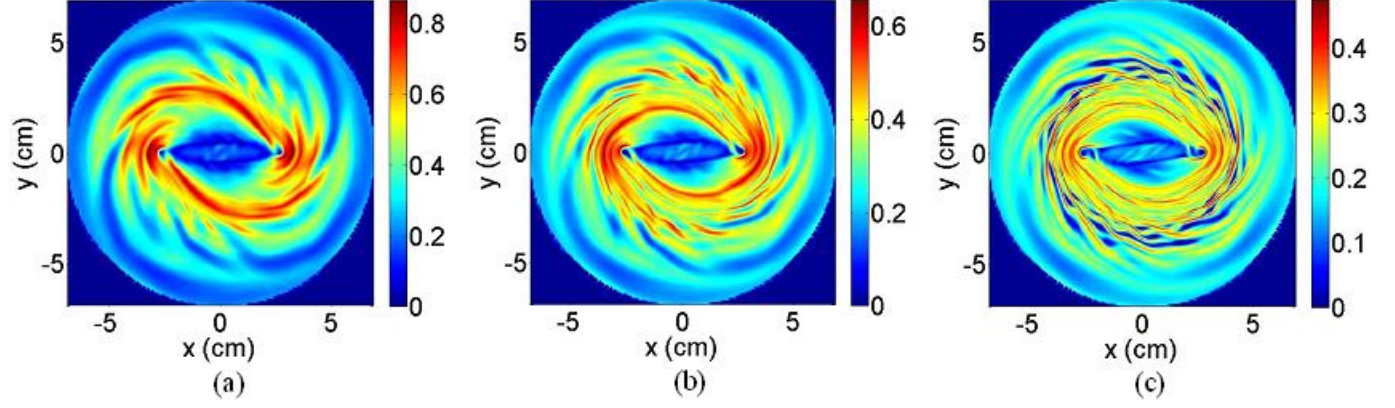


Figure 5.7. Computed the FTLE fields with different integrating times, $\tau = T$, $\tau = 2T$ and $\tau = 4T$, respectively, where T is the time for one period.

a high degree of stretch near the wall where the tips of the blade rotate. Again, the LCS occurs on the boundary that separates strong and weak mixing as observed from the tracer plot in Figure. However, the mixed region in this case is smaller than the that of the centered-blade mixer and it is the mixed region that is almost enclosed by the LCS instead of the region with a weak mixing as in the case of the centered blade. Also, we notice a vanishing of the LCS as one of the blade tips rotates away from the wall and its reappearance when the other tip rotates toward the wall. Again, we observe the numerical displacement of the tracers to further our understanding of the transport. It is shown in Figure 5.9 that the LCSs in Figure 5.8 separate the region of a relatively higher degree mixing within the sweep of the blade from that of a lower degree of mixing. This is different from the case of the centered-blade mixer in which the region bounded inside by the LCSs experience a lower degree of mixing than the other region. Furthermore, the particle striation in this case occurs in a more intricate fashion with a faster mixing rate than the case of the centered-blade mixer. However, in this case the amount of particles that are mixed together is clearly less than that of the centered-blade mixer. We verify these mixing behaviors again using the definition of mixing in Section 5.4.

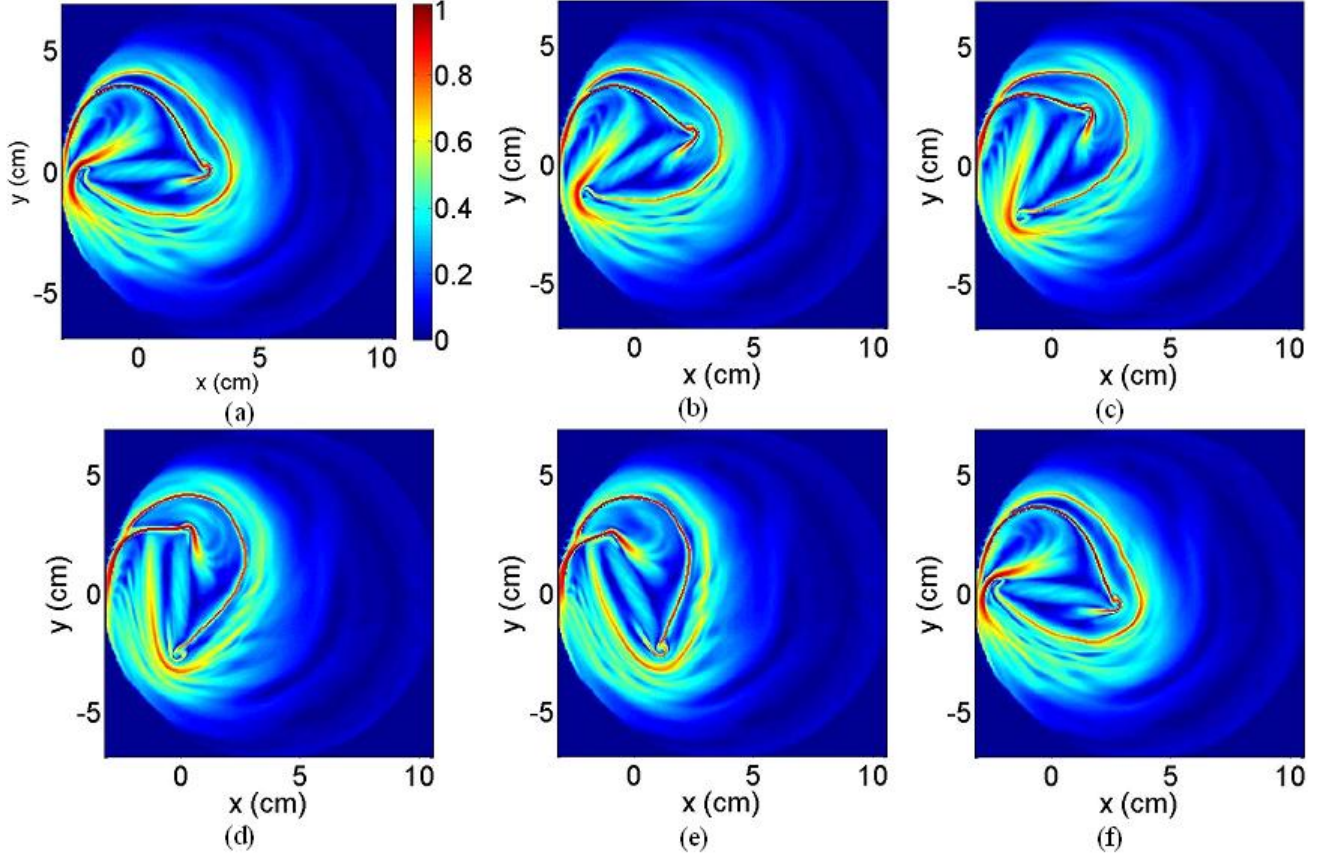


Figure 5.8. The forward-time FTLE field of the mixer with the offset blade is computed at different times t with the integrating time $\tau = 64\Delta t$. From (a) to (f), the FTLE is computed at time $t = \Delta t$, $t = 10\Delta t$, $t = 20\Delta t$, $t = 30\Delta t$, $t = 40\Delta t$ and $t = 50\Delta t$, respectively.

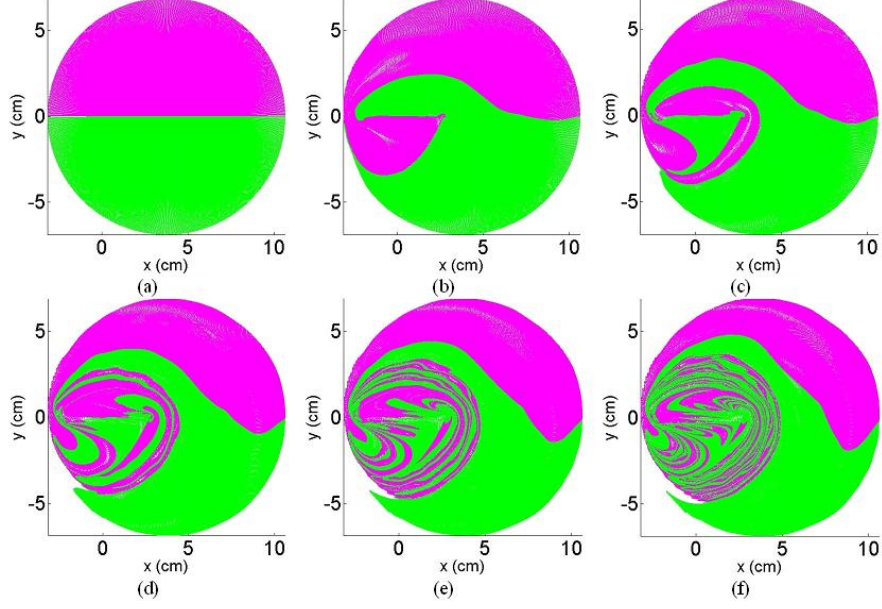


Figure 5.9. Snapshots of particle tracers for the mixer with the centered blade. Two sets are particles, A (green) and B (magenta) are initially separated to occupy half of the physical space as shown (a). The initial particle distribution is then allowed to change as dictated by the flow field for 1,2,3,4, and 5 periods, (b)-(f), respectively.

5.3.3 Mathematical model of the mixers

This section is aimed to define a mathematical model of our mixing devices and use it to investigate certain salient features of the FTLE fields of the mixing devices. We suppose that at time $t = 0$ the velocity field is constructed from a vorticity field under assumption of two-dimensional flow that the vorticity field is given by a vertical vorticity, $\omega = \omega_z$. Hence, we solve the Poisson's equation of the stream-function:

$$\Delta H = -\omega_z \quad \text{on } \Omega, \quad u = 0 \text{ on } \Omega. \quad (5.14)$$

Here Ω is the circular boundary of the mixer (radius=7.0) and H is the Hamiltonian of the velocity field, that is,

$$\begin{aligned} \frac{dx}{dt} &= -\frac{\partial H(x, y)}{\partial y} \\ \frac{dy}{dt} &= \frac{\partial H(x, y)}{\partial x} \end{aligned} \quad (5.15)$$

We then recast the above time-independent system to a time-dependent system in a rotating frame using the change of coordinate

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{bmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix}. \quad (5.16)$$

Now that we develop a mathematical model for the mixer, we have to determine the vorticity ω_z that is well-suited to replicate the important feature of the FTLE field of our mixing devices; that is the pseudo-barrier observed in the preceding section. We assume that the vorticity field ω_z is given by

$$\omega_z = G_0 - C(G_1 + G_2), \quad (5.17)$$

where

$$G_0 = \exp(-\sigma_0(\sigma_{0,x}x^2 + \sigma_{0,y}y^2))^{2m} \quad \text{and} \quad (5.18)$$

and

$$G_{1,2} = \exp(-\sigma_{1,2}(\sigma_{1x,2x}(x - s_{1,2})^2 + \sigma_{1y,2y}y^2))^2. \quad (5.19)$$

Here C determines the relative magnitude of the vortices at the tips of the blade represented by the bivariate Gaussian function G_1 and G_2 . The super-Gaussian function is used to replicate the vorticity due to the blade. Note the G_0 becomes a bivariate Gaussian function when $m = 1$. We vary the parameter C to observe changes in the FTLE and fix the other parameters to $m = 2, \sigma_0 = 100, \sigma_{0,x} = 0.4, \sigma_{0,y} = 2.5, \sigma_1 = \sigma_2 = 400, s_1 = 0.5, s_2 = -0.5, \sigma_{1x} = \sigma_{2x} = 1.0$ and $\sigma_{1y} = \sigma_{2y} = 0.5$. Our specific choice of the form of Eqs. (5.18) and (5.19) and the values of the parameters is justified in Figures 5.10 and 5.11, remembering that our goal here is to develop a model that contains the main topological features of the true system, which is simplifying but not necessarily predictive. Figures 5.10 and 5.11 show a comparison between the vorticity and velocity field of the given data and those numerically computed by Eq. (5.15).

Figure 5.12 illustrates the change in FTLE field at time $t = 0$ as C varied. First, We notice that the FTLE fields of our model has a similar LCS, or a transport pseudo-barrier, to that of the experimental data presented in the preceding section. Note that the FTLE

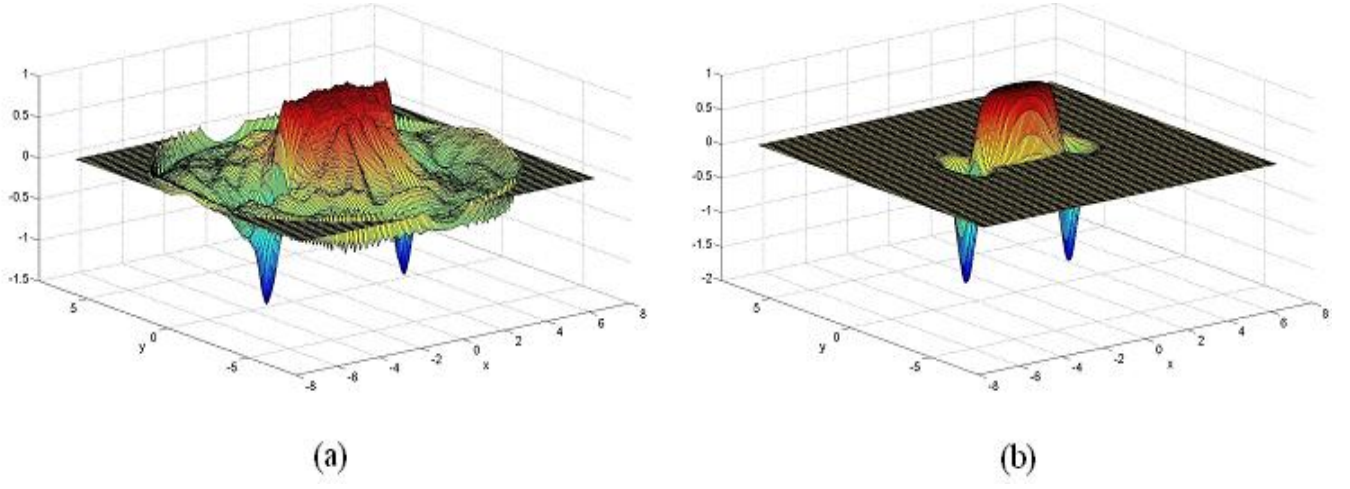


Figure 5.10. The vertical vorticity field, ω_z , of the a) experimental data and b) analytic model using Eq. (9) with $C=2.0$.

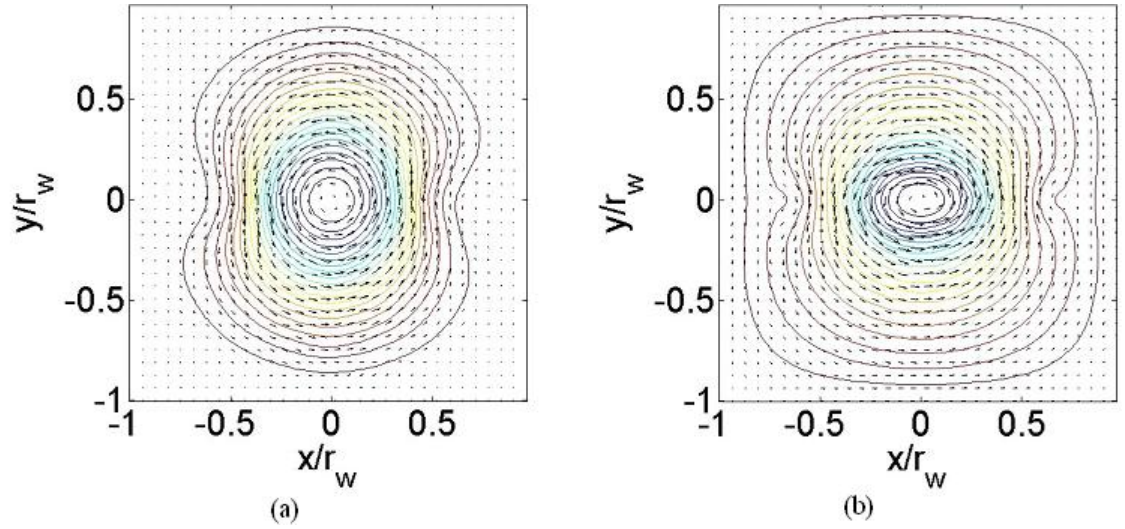


Figure 5.11. The in-plane velocity field and representative streamlines for the a) experimental data and b) analytical model with $C=2.0$. Note the spatial dimensions have been normalized by the radius of the cylinder.

fields of our numerical model also rotate in the same fashion as observed in the simulation of the real data and so we show here only the FTLE field at time $t = 0$. It is clear that the magnitude of the vortices near the tips of the blade are responsible for the strong barrier, which is the region with large FTLE. Now, we observe the backward-time FTLE field of the model illustrated in Figure 5.13. Again, as the magnitude of the vortices increased, the region with a large backward-time FTLE becomes more evident. Recall that the backward-time LCS signifies a generalization of an unstable manifold.

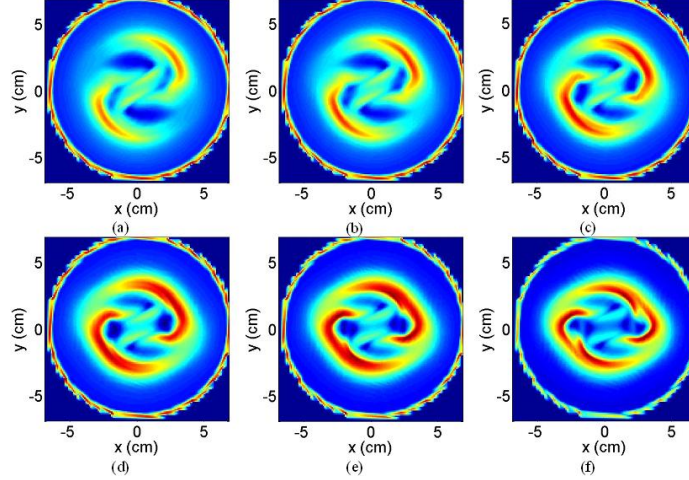


Figure 5.12. The forward-time FTLE field at time $t = 0$ of the model Eq. (5.15) in a rotating frame. The parameter C in the vorticity field Eq. (5.17) is varied to observe the change of the FTLE field. (a) $C = 0.5$, (b) $C = 1.0$, (c) $C = 1.5$, (d) $C = 2.0$, (e) $C = 2.5$ and (f) $C = 3.0$

5.4 Mixing notions

This section deals with some mixing notions used to measure the degree of mixing in terms of homogeneity of two initial sets (or fluid) A and B after some time. For our mixing devices, the mixing notion we will use is preferable to the notions of mixing defined in ergodic theory. Indeed, there are a variety of mixing notions used in the ergodic theory

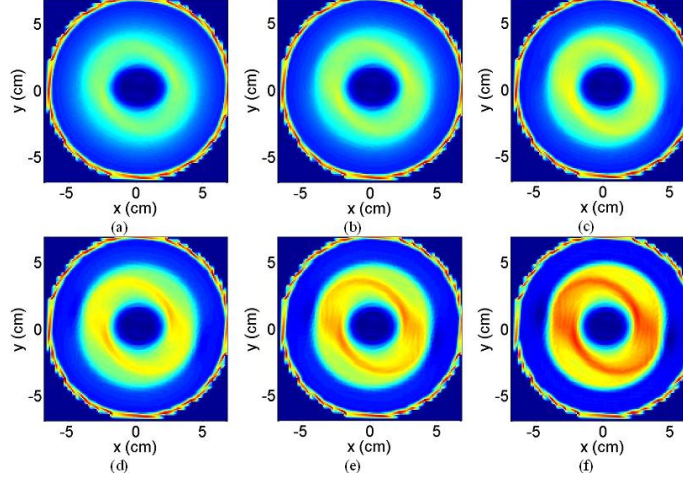


Figure 5.13. The backward-time FTLE field at time $t = 0$ of the model Eq. (5.15) in a rotating frame. The parameter C in the vorticity field Eq. (5.17) is varied to observe the change of the FTLE field. (a) $C = 0.5$, (b) $C = 1.0$, (c) $C = 1.5$, (d) $C = 2.0$, (e) $C = 2.5$ and (f) $C = 3.0$

literature and they describe different views of degrees of mixing. Nevertheless, the standard notion of mixing in the ergodic theory is given below.

Definition 5.2. A measure-preserving transformation $f : \mathbb{M} \rightarrow \mathbb{M}$ is said to be *mixing* if for any two measurable sets A and B in \mathbb{M} we have

$$\lim_{n \rightarrow \infty} \mu(f^{-n}(A) \cap B) = \mu(A)\mu(B). \quad (5.20)$$

In fact, the above definition of mixing is originally called *strong mixing*, but we will refer to it simply as *mixing* since we will not use the notion of *weak mixing* (see [50]) in this thesis. To aid the interpretation, we rewrite the above mixing definition, provided that $\mu(B) > 0$, as

$$\lim_{n \rightarrow \infty} \frac{\mu(f^{-n}(A) \cap B)}{\mu(B)} = \mu(A). \quad (5.21)$$

This can be interpreted to mean that under some iterations by T the proportion of A found in B will approach the proportion of A in \mathbb{M} . Note that this notion of mixing should be compared with the definition of the transition matrix (3.49). As a consequence, if each B_i in the finite collection $\{B_i\}$ used to generate the transition has the same measure, the limit distribution (invariant distribution) of the transition matrix of the transformation f will be a uniform distribution. In a probabilistic manner (when μ is the probability measure), this

means that if f is mixing, $f^{-n}(A)$ will eventually become independent from B for any two subsets A and B in \mathbb{M} . Another useful and intuitive notion of mixing is called *topological mixing*, which is defined in a language of topological dynamics.

Definition 5.3. (See Sturman [50]) A (topological) dynamical system $f : \mathbb{X} \rightarrow \mathbb{X}$ is *topologically mixing* if for any two open sets $U, V \in \mathbb{X}$ there exists an integer $N > 0$ such that $f^n(U) \cap V \neq \emptyset$ for all $n \leq N$.

Interestingly, it can be proved that the measure-theoretic notion of mixing in (5.2) implies topological mixing, but not the converse [50, 72]. However, although the notion of mixing in ergodic theory provides a useful concept, it involves infinite time limits, which is not suitable for quantifying mixing in a real-world application. Nonetheless, the *decay of correlations of a scalar function* have been used to quantify mixing in various areas of applications and it is defined as.

$$C_n(\phi, \psi) = \left| \int \phi(\psi \circ f^{-n}) d\mu - \int \phi d\mu \int \psi d\mu \right|. \quad (5.22)$$

Here ϕ and ψ are scalar functions, which can be regarded as observables, e.g., the fluid concentration. If the transformation f is mixing, $C_n(\phi, \psi)$ will decay to zero and the rate of decaying is usually used to quantify the speed of mixing. This concept of mixing is, indeed, motivated by the measure-theoretic notion of mixing by observing that the integral expression of the mixing condition (5.2) can be written as

$$\int \chi_B(\chi_A \circ f^n) d\mu - \int \chi_A d\mu \int \chi_B d\mu \rightarrow 0 \quad \text{as } n \rightarrow \infty. \quad (5.23)$$

Now, we can see that the notion of mixing in (5.2) describes transport from the initial set A into B and vice versa since the sets A and B are initialized as the reference sets. However, we would like to observe the mixing in terms of homogeneity instead of how much transport occurs between two fixed sets. Therefore, for our study of mixing, we will use the following mixing ratio. First, we consider a measure of regions with mixing of the centered-blade mixer. We initially divide the phase space into two distinct regions A and B as seen in Figure 5.5(a). We will then proceed in a fashion closely motivated by the definition of the mixing (5.21), which demands relative measures become asymptotically

distributed. We identify the particles initially located in A species S_a , and similarly S_b for those particles in B . We then partition the space M into disjoint regions $R_i, i = 1, \dots, N_R$, such that $M \subseteq \bigcup_{i=1}^{N_R} R_i$. Let $S = \{R_i : R_i \text{ contains both species } S_a \text{ and } S_b\}$. Finally, we define the ratio of the regions including both S_a and S_b by

$$R = \frac{\sum_{R_i \in S} \mu(R_i)}{\sum_i \mu(R_i)}, \quad (5.24)$$

where the measure μ can be a Lebesgue(volume) measure. Clearly, the quality of mixing depends on the initial topology of sets A and B . For the sets A and B initialized as in Figure 5.5(a), the ratio R at each period is given in Figure 5.14 for different scales of N_R . These results agree with the behavior observed from the LCSs and tracers of the mixers in section 5.3.2. Although the mixing rate of the offset-blade mixer is faster than that of the centered-blade, which can be observed by comparing the slope of the ratio R between Figures 5.14(a) and (b) at the early periods, the ratio R become saturated since only the small region surrounded by the LCS, see Figure 5.8 and 5.9, experiences a high degree of mixing. Since the homogeneity depends on the scale of the partition, the mixing ratio becomes smaller as the partition scale decreases. In fact, the mixing ratio as defined above will asymptotically approach to zero as the size of the partition goes to zero.

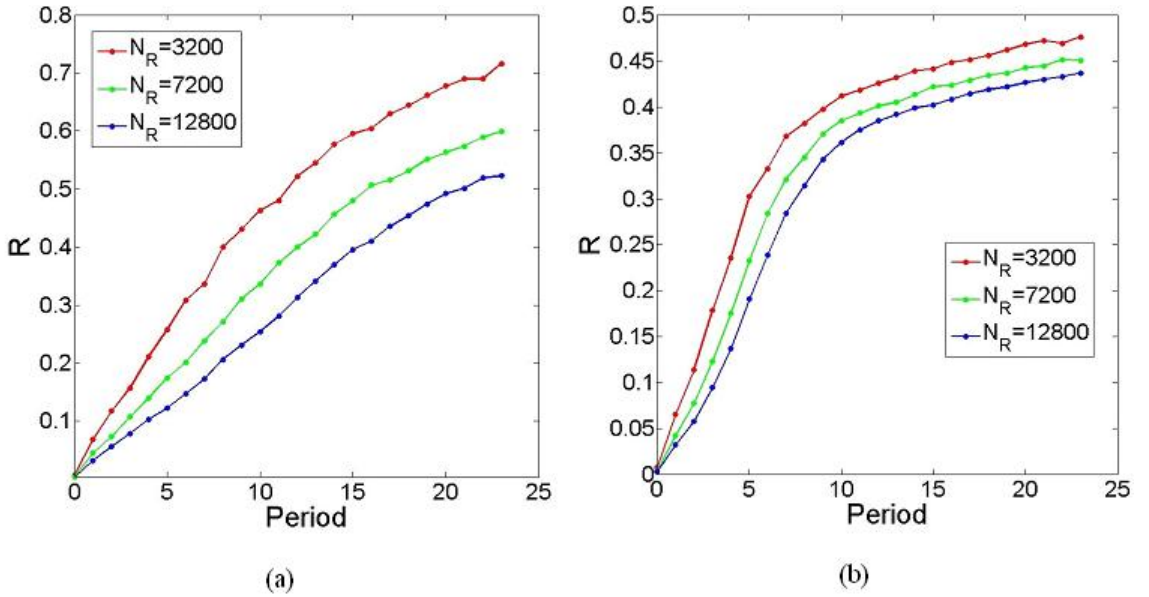


Figure 5.14. The plot of the ratio R in Eq. (5.24) at each period using different scales of N_R . (a) the mixer with a centered blade. (b) the mixer with an offset blade

Chapter 6

Inverse Problem of Motion Estimation

We present in this chapter an approach to approximate the Frobenius-Perron transfer operator from a sequence of time-ordered images, that is, a movie data set. Unlike time-series data, successive images do not provide a direct access to a trajectory of a point in phase space; more precisely, a pixel in an image plane. Therefore, we need to reconstruct the velocity field from image sequences based on the infinitesimal generator of the Frobenius-Perron operator. Moreover, we relate this problem to the well-known optical flow problem from the computer vision community.

6.1 Inverse Problem of Motion Estimation

This section describes an approach to extract the velocity field that transforms an intensity pattern of one image into the next image in a sequence. This problem is referred to as the *optical flow* problem in image processing. Based on the framework of the Frobenius-Perron operator reviewed in Chapter 3, this problem is inherently an inverse problem and it is also ill-posed. There are numerous methods to the optical flow problem which can be developed to emphasize different aspects of the expected solution. We will review some of

these below in order to better understand how to apply optical flow methods to measurable dynamics.

We assume that the desired velocity field is autonomous throughout the image region and so the trajectory of a point in the image region is governed by the following equation:

$$\begin{aligned}\frac{dx_1}{dt} &= u(x_1, x_2) \\ \frac{dx_2}{dt} &= v(x_1, x_2),\end{aligned}\tag{6.1}$$

where $u(x_1, x_2)$ and $v(x_1, x_2)$ are two unknown velocity fields to be approximated from a sequence of images, which provides to us only a temporal variation of the brightness pattern. Let $I(x_1, x_2, t)$ represent a gray-scale intensity function of an image, a function of brightness of a pixel at point $\mathbf{x} = (x_1, x_2)$ and (discrete) time t . Since the aim of this work is to approximate the Frobenius-Perron operator from a sequence of images, the optical flow constraint for this matter, which will be called the *Frobenius-Perron constraint* (FPC), is the continuity equation (3.28) derived in Chapter 3. Therefore, the temporal variation of the image intensity function $I(x_1, x_2, t)$ is given by

$$\frac{\partial I}{\partial t} + \text{div}(I\mathbf{v}) = 0,\tag{6.2}$$

Recall that this equation is derived from the infinitesimal generator of the Frobenius-Perron operator [40], and so it describes the temporal variation of the distribution of brightness under $[u(x_1, x_2), v(x_1, x_2)]$. In the language of fluid mechanics, this equation also describes the temporal variation of the brightness within an infinitesimal volume as evolved by the flux of the brightness through the boundary surface of the volume. As a special case of this constraint when $\text{div}(\mathbf{v})$ is zero throughout the image plane, we arrive at the classical *brightness constancy constraint* [19], which can be expressed by

$$I(\mathbf{x} + \mathbf{d}(\mathbf{x}), t + \delta t) - I(x, t) = 0,\tag{6.3}$$

where $\mathbf{d}(\mathbf{x})$ is the displacement from time t to $t + \delta t$ and δt is the temporal sampling rate.

This constraint can be regarded as the integrated form of the following PDE:

$$\frac{dI}{dt} = \frac{\partial I}{\partial x_1}u + \frac{\partial I}{\partial x_2}v + \frac{\partial I}{\partial t} = 0.\tag{6.4}$$

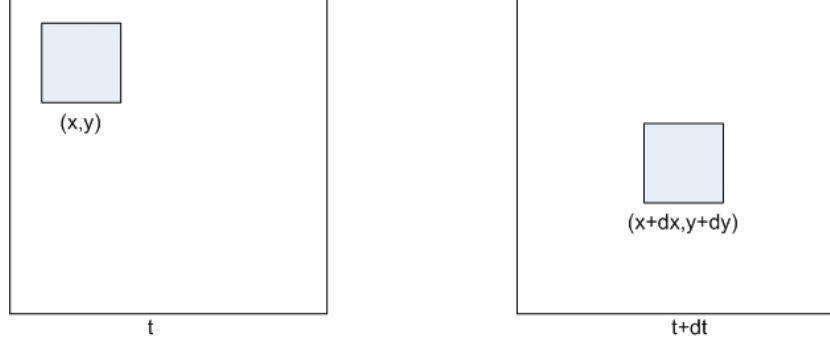


Figure 6.1. The illustration of the assumption implied by Eq. (6.4)

The above PDE assumes that the intensity pattern of local-time varying image regions are constant under a motion in a short time duration, which follows from the first-order Taylor series expansion about $I(x_1, x_2, t)$:

$$I(x + \delta x_1, x_2 + \delta x_2, t + \delta t) = I(x_1, x_2, t) + \frac{\partial I}{\partial x_1} \delta x_1 + \frac{\partial I}{\partial x_2} \delta x_2 + \frac{\partial I}{\partial t} \delta t + O^2, \quad (6.5)$$

where O^2 denotes the second order $\delta x_1, \delta x_2$, and δt . This implies that the PDE (6.4) describes the rigid motion in which the brightness at each pixel is kept constant along its trajectory, if the displacement is sufficiently small. The illustration of this assumption is shown in Figure 6.1. However, this assumption is usually violated by most natural images obtained from meteorology, oceanography, climatology, fluid experiment, etc. For such image data, it is common to have a gain or loss of the brightness of a given image pixel due to the non-zero diverging motion. Hence, the constraint equation (6.2) would be better suited to capture the physics underlying these natural images.

In contrast to the brightness conservation (6.3), the integrated form of the FPC (6.2) is:

$$I(x + \mathbf{d}(\mathbf{x}), t + \delta t) - I(x, t) \exp(-\text{div} \mathbf{d}(\mathbf{x})) = 0. \quad (6.6)$$

This can be seen by rewriting Eq. (6.6) as

$$I(x + \mathbf{d}(\mathbf{x}), t + \delta t) = I(\mathbf{x}, t)(1 - \text{div} \mathbf{d}(\mathbf{x}) + O^2). \quad (6.7)$$

Thus, if we assume that the velocity is constant along the displacement $\mathbf{d}(\mathbf{x})$ from time t to $t + \delta t$, then we have $\mathbf{d}(\mathbf{x}) = \mathbf{v}(\mathbf{x}, t)/\delta t$, which combining with (6.7) implies the FPC (6.2)

as desired. When the divergence is zero, this constraint reduces exactly to the brightness constancy constraint (6.3). However, with a non-zero divergence, the constraint (6.7) allows the brightness to be scaled by a factor $\exp(-\text{div}\mathbf{d}(\mathbf{x}))$. This model dictates the diverging nature of compressible fluids that causes the appearance/disappearance of fluid particles. Another key theoretical importance of using the constraint (6.2) is that in the case of $2D$ transmittance imagery of a $3D$ fluid flow, the conservation of mass satisfied by $3D$ flow also hold in its corresponding $2D$ flow captured by the image sequence as well, see Fitzpatrick [73] and Wildes [27]. A sufficient condition for this to be true is the null normal flow condition along the image boundary, see Appendix C.

The challenge is that the optical flow computation based only either on the constraint equation (6.2) or (6.4) is an ill-posed problem because on each location and each time, we have to solve a single scalar equation for two scalar unknowns u and v . In the case of the brightness constancy assumption, this is called the *aperture problem*, where only the normal component of the velocity field, given in the direction of the gradient $\nabla\mathbf{v}$ can be solved from the constraint equation (6.4), but not the tangential component. Numerous methods have been proposed to overcome this ill-posed problem, which can be categorized into two general classifications. The *local* methods such as the Lucas-Kanade method [74] and the structure of tensor method [75] employ the optimization of some local energy-like expression, whereas the *global* approaches attempt to minimize a global energy functional [19]. Survey and comparison of various methods was demonstrated by Barron [76], and Galvin [77].

We resort to the Tikhonov regularization technique [78], which belongs to the class of global methods, to cope with the ill-posed problem. The idea is to approximate the solution of the constraint equation by solving a minimization problem of the form

$$\inf_{\mathbf{v}} \int_{\Omega} (F(\mathbf{v}) + S(\mathbf{v})) d\Omega, \quad (6.8)$$

where $F(\mathbf{v})$ is the data fidelity term based either on the constraint equation (6.2) or (6.4) and $S(\mathbf{v})$ is an additional regularization term to stabilize the solution and to relax the assumption of invariance of the flow used in the data fidelity term. Thus it allows materials to leave the image domain although it may not satisfy the flow invariant assumption. The classical regularization term proposed by Horn and Schunck [19] is the so-called *smoothness*

constraint:

$$S(\mathbf{v}) = \alpha \left(\|\nabla u(x_1, x_2)\|^2 + \|\nabla v(x_1, x_2)\|^2 \right), \quad (6.9)$$

where α is a constant and the norm is the standard L^2 norm. This constraint claims that neighboring pixels of a point in a sequence of images are likely to move in a similar way, i.e. the motion vectors are spatially varying in a smooth way. Thus, it encourages the isotropic smoothness of the recovered optical flow without taking into account the discontinuities at the edges where the gradient of intensity is large. It also have an advantage of ameliorating the effects of noise commonly occurring in imperfect data. Another alternative to the smoothness regularizer is the so-called div-curl regularizer [79]

$$S(\mathbf{v}) = \alpha \|\operatorname{div} \mathbf{v}\|^2 + \beta \|\operatorname{curl} \mathbf{v}\|^2, \quad (6.10)$$

which reduces to the smoothness constraint when $\alpha = \beta$.

6.2 Euler-Lagrange Equation

For the sake of readers' convenience, we summarize some essential theorems from the classical calculus of variation used to study the problem presented in Eq. (6.8). Particularly, we review the theorems, without proof, for the existence of the solution and the conditions in which the solution of the associated Euler-Lagrange equation is also the solution of the minimization problem; if these conditions are not satisfied, the solution of the Euler-Lagrange equation might be merely the stationary point of the minimization problem. There are a number of excellent books for this subject and we refer to some of them, e.g. Dacorogna [80, 81], Gelfand and Fomin [82], Attouch, Buttazzo and Michaille [83] or Aubert [84]. In this section, we will use the following notations concerning the space of the continuous and differentiable functions and the Sobolev space:

- Let $\Omega \subset \mathbb{R}^n$ be an open set. $C^k(\overline{\Omega})$ is the set of continuous functions $f : \Omega \rightarrow \mathbb{R}$ of which partial derivatives up to the order k exists. If $f : \Omega \rightarrow \mathbb{R}^N$, we write $C^k(\overline{\Omega}; \mathbb{R}^N)$
- $C^k(\overline{\Omega})$ is the set of continuous functions in $C^k(\Omega)$ which can be extended continuously to $\overline{\Omega}$.

- $W^{1,p}(\Omega)$ is the set of functions $f : \Omega \rightarrow \mathbb{R}$, for $f \in L^p(\Omega)$, of whose the weak partial derivatives $f_{x_i} \in L^p(\Omega)$ for every $i = 1, \dots, n$.
- If $1 \leq p < \infty$, We denote by $W_0^{1,p}(\Omega)$ the closure of $C_0^\infty(\Omega)$ functions in $W^{1,p}(\Omega)$. This is equivalent to say that, if Ω is bounded, $f \in W_0^{1,p}(\Omega)$ is such that $f \in W^{1,p}(\Omega)$ and $f = 0$ on $\partial\Omega$.
- By abuse of language, we write $f \in f_0 + W_0^{1,p}(\Omega)$ when $f, f_0 \in W^{1,p}(\Omega)$ and $f - f_0 \in W_0^{1,p}(\Omega)$.

First, we consider the problem of the following form:

$$\inf_f \int_{\Omega} (h(x, f(x), \nabla f(x))) d\Omega \quad (6.11)$$

where

- $\Omega \subset \mathbb{R}^n$ is a bounded open set;
- $h : \overline{\Omega} \times \mathbb{R} \times \mathbb{R}^n \longrightarrow \mathbb{R}, h = h(x, f, \xi)$;
- $f \in f_0 + W_0^{1,p}(\Omega)$.

The problem of this form is different from our minimization stated in Eq. (6.8) in that we have, instead, $S, F : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^2$, vectorial cases. However, the theorems concerning the scalar case as in (6.11) are simpler to understand and so we present them first and defer the theorems relating to the vectorial case to the end of this section.

Theorem 6.1. (See Dacorogna [81]) Let $\Omega \subset \mathbb{R}^n$ be a bounded open set with Lipschitz boundary. Let $h^0(\overline{\Omega} \times \mathbb{R} \times \mathbb{R}^n)$, $h = h(x, f, \xi)$, satisfy

- Convexity: $\xi \rightarrow h(x, f, \xi)$ is convex for every $(x, f) \in \overline{\Omega} \times \mathbb{R}$;
- Coercivity: there exist $p > q \geq 1$ and $\alpha_1 > 0, \alpha_2, \alpha_3 \in \mathbb{R}$ such that

$$h(x, f, \xi) \geq \alpha_1 |\xi|^p + \alpha_2 |f|^q + \alpha_3, \quad \forall (x, f, \xi) \in \overline{\Omega} \times \mathbb{R} \times \mathbb{R}^n.$$

Let

$$m = \inf_f \left\{ J(f) = \int_{\Omega} (h(x, f(x), \nabla f(x))) d\Omega : f \in f_0 + W_0^{1,p}(\Omega) \right\}$$

where $f_0 \in W^{1,p}(\Omega)$ with $J(f_0) < \infty$. Then there exists $\bar{f} \in f_0 + W_0^{1,p}(\Omega)$ a minimizer of which the infimum m is obtained.

As shown in [80, 81, 84], the hypotheses of the above theorem are nearly necessary, that is when any of them, except the continuity of h , are weakened, a counterexample to the existence of the minima can be found. Note that in the above theorem the coercivity is assumed to guarantee the boundness of the minimizing sequences and the convexity implies the lower semicontinuity of the minimizing sequence, which is sufficient to pass to the limit, if it exists $J(\bar{f}) < \infty$, on the sequence. In the following theorem, we assume the existence of the minimizer $\bar{f} \in W^{1,p}$ and state the weak form of the Euler-Lagrange equation. With an additional condition to the minimizer, i.e., \bar{f}^2 , we will be able to write the standard optimality condition, i.e., the Euler-Lagrange equation.

Theorem 6.2. (See Dacorogna [81]) Let $\Omega \subset \mathbb{R}^n$ be a bounded open set with Lipschitz boundary. Let $p \geq 1$ and $h \in C^1(\bar{\Omega} \times \mathbb{R} \times \mathbb{R}^n)$, $h=h(x,f,\xi)$, satisfy

- there exists $\beta \geq 0$ so that for every $(x, f, \xi) \in \bar{\Omega} \times \mathbb{R} \times \mathbb{R}^n$

$$|h_f(x, f, \xi)|, |h_{\xi}(x, f, \xi)| \leq \beta(1 + |f|^{p-1} + |\xi|^{p-1})$$

where $h_{\xi} = (h_{\xi_1}, \dots, h_{\xi_n})$.

Let $\bar{f} \in f_0 + W_0^{1,p}(\Omega)$ be a minimizer of

$$m = \inf_f \left\{ J(f) = \int_{\Omega} (h(x, f(x), \nabla f(x))) d\Omega : f \in f_0 + W_0^{1,p}(\Omega) \right\} \quad (6.12)$$

where $f_0 \in W^{1,p}(\Omega)$, then \bar{f} satisfies the weak form of the *Euler-Lagrange* equation

$$\int_{\Omega} [h_f(x, \bar{f}, \nabla \bar{f}) \varphi + \langle h_{\xi}(x, \bar{f}, \nabla \bar{f}), \nabla \varphi \rangle] dx = 0, \quad \forall \varphi \in W_0^{1,p}(\Omega). \quad (6.13)$$

Furthermore, if $h \in (C^2(\bar{\Omega}) \times \mathbb{R} \times \mathbb{R}^n)$ and $\bar{f} \in C^2(\bar{\Omega})$ then \bar{f} satisfies the *Euler-Lagrange* equation

$$\sum_{i=1}^n \frac{\partial}{\partial x_i} [h_{\xi_i}(x, \bar{f}, \nabla \bar{f})] = h_f(x, \bar{f}, \nabla \bar{f}), \quad \forall x \in \bar{\Omega}. \quad (6.14)$$

Conversely, if $(f, \xi) \rightarrow h(x, f, \xi)$ is convex for every $x \in \overline{\Omega}$ and if \bar{f} is a solution of (6.13) or (6.14) then it is a minimizer in which m is obtained.

Next, we consider the case where the unknown function $f(x)$ is a vector, that is $f : \Omega \in \mathbb{R}^n \rightarrow \mathbb{R}^N$ with $n, N > 1$. In dealing with a gray-scale image we will consider only the case of the two-dimensional vector field, i.e. $n = N = 2$. We summarize the existence of the minimizer only for this case. However, the following theorem still holds for the higher dimensional case.

Theorem 6.3. (See Darcorogna [81]) Let $n = N = 2$ and $\Omega \subset \mathbb{R}^2$ be a bounded open set with Lipschitz boundary. Let $h : \overline{\Omega} \times \mathbb{R}^2 \times \mathbb{R}^{2 \times 2} \rightarrow \mathbb{R}$, $h = h(x, f, \xi)$, and $H : \overline{\Omega} \times \mathbb{R}^2 \times \mathbb{R}^{2 \times 2} \times \mathbb{R} \rightarrow \mathbb{R}$, $H = H(x, f, \xi, \delta)$, be continuous and satisfying

$$h(x, f, \xi) = H(x, f, \xi, \det \xi), \quad \forall (x, f, \xi) \in \overline{\Omega} \times \mathbb{R}^2 \times \mathbb{R}^{2 \times 2}$$

where $\det \xi$ denotes the determinant of the 2×2 matrix ξ . Let

$$m = \inf_f \left\{ J(f) = \int_{\Omega} (h(x, f(x), \nabla f(x)) d\Omega : f \in f_0 + W_0^{1,p}(\Omega) \right\} \quad (6.15)$$

Assume also that

- Quasi-convexity: $(\xi, \delta) \rightarrow H(x, f, \xi, \delta)$ is convex for every $(x, f) \in \overline{\Omega} \times \mathbb{R}^2$;
- Coercivity: there exists $p > \max[q, 2]$ and $\alpha_1 > 0, \alpha_2, \alpha_3 \in \mathbb{R}$ such that

$$H(x, f, \xi, \delta) \geq \alpha_1 |\xi|^p + \alpha_2 |f|^q + \alpha_3, \quad \forall (x, f, \xi, \delta) \in \overline{\Omega} \times \mathbb{R}^2 \times \mathbb{R}^{2 \times 2} \times \mathbb{R}.$$

Let $f_0 \in W_0^{1,p}(\Omega)$ be such that $J(f_0) < \infty$, then the problem (6.15) has at least one solution.

We note that the quasi-convexity condition in the above theorem is a weaker definition of convexity. If $\xi \rightarrow h(x, f, \xi)$ is convex then one can choose $H(x, f, \xi, \delta) = h(x, f, \xi)$, and the convexity and quasi-convexity is equivalent in this case. However, if $\xi \rightarrow h(x, f, \xi)$ is non-convex (w.r.t. ξ), it is still possible to choose $H(x, f, \xi, \delta)$ as a convex function of (ξ, δ) .

Similar to the theorem 6.2, the associated Euler-Lagrange of the problem (6.15) becomes a pair of partial differential equations:

$$\sum_{i=1}^n \frac{\partial}{\partial x_i} [h_{\xi_{i,j}}(x, f, \bar{f})] = h_{f_j}(x, \bar{f}, \nabla \bar{f}), \quad \forall x \in \overline{\Omega}, j = 1, 2. \quad (6.16)$$

The above partial differential equation is the optimality condition of the minimization problem. In the subsequent sections, we will present some techniques to solve the Euler-Lagrange equation associated to the problem of velocity field estimation of certain types of real-world images. We end this section by mentioning that in most computer-vision problems the discontinuities of velocity fields and in the images are important to the analysis. In such cases, the mathematical formalism based on the Sobolev space, as presented in this section, becomes inappropriate since it cannot take into account the gradient of a discontinuous function f . Instead, one has to regard the gradient of f as a measure and the analysis based on the space of the functions of bounded variation $BV(\Omega)$ is widely used for this purpose. The mathematical analysis, however, becomes somewhat involved and will not be included in this thesis. We refer the interested readers to [84, 85, 86]. Nonetheless, we would like to remind that in this thesis the dynamical systems are assumed to be at least C_0^1 as mentioned in the preceding section. Therefore, it is appropriate here to use the classical results stated in this section.

Now consider the variational problem (6.8) with the data fidelity term given by either Eqs. (6.2) or (6.4) and the smoothness constraint as described by Eq. (6.9). First, consider the case of the brightness constancy hypothesis of which the (Horn-Schunck) energy functional is given by

$$E(u, v) = \int_{\Omega} (I_{x_1}u + I_{x_2}v + I_t)^2 d\Omega + \int_{\Omega} \alpha (\|\nabla u(x_1, x_2)\|^2 + \|\nabla v(x_1, x_2)\|^2) d\Omega \quad (6.17)$$

Thus we can apply a classical variational principle, see Theorem 6.3, to this problem, i.e, if $(u(x_1, x_2), v(x_1, x_2))$ is a minimizer of the functional (6.17), then it necessarily satisfies the associated Euler-Lagrange equations:

$$\begin{aligned} \Delta u &= \alpha I_{x_1} (I_{x_1}u + I_{x_2}v + I_t) \\ \Delta v &= \alpha I_{x_2} (I_{x_1}u + I_{x_2}v + I_t), \end{aligned} \quad (6.18)$$

where $\Delta := \partial_{xx} + \partial_{yy}$ denotes the Laplace operator. At the image boundary we enforce the Neumann boundary condition (a null normal flow at the boundary) to the above equations; that is

$$\mathbf{n} \cdot \mathbf{v} = 0 \quad \text{on } \partial I, \quad (6.19)$$

where \mathbf{n} specifies the normals to the image boundary, denoted by ∂I . This boundary condition allows the flow to run across the image boundary. However, if the flow is bounded within the image, one may consider using the zero boundary condition, instead. The above equations can be regarded as the steady state ($\theta \rightarrow \infty$) of the diffusion-reaction equation

$$\begin{aligned}\partial_\theta u &= \Delta u - \alpha I_{x_1}(I_{x_1}u + I_{x_2}v + I_t) \\ \partial_\theta v &= \Delta v - \alpha I_{x_2}(I_{x_1}u + I_{x_2}v + I_t),\end{aligned}\tag{6.20}$$

where θ denotes an artificial evolution parameter in this case. Recall that $\Delta u = \text{div}(g\nabla u)$ with $g \equiv 1$, i.e., the diffusion term of the above equation can be regarded as the linear diffusion with the diffusivity $g \equiv 1$. Hence, the diffusion process of the above diffusion-reaction affect the solution in a similar fashion to the convolution by the Gaussian kernel. As a consequence, it “blurs” across the flow discontinuity. We will resume this discussion again in the next section, where the discontinuity-preserving optical flow is discussed.

Now we consider the Euler-Lagrange equation of the optimization problem (6.8) with the data fidelity term given by the FPC (6.2):

$$E(u, v) = \int_{\Omega} \left(\frac{\partial I}{\partial t} + \text{div}(I\mathbf{v}) \right)^2 d\Omega + \alpha \int_{\Omega} (\|\nabla u(x_1, x_2)\|^2 + \|\nabla v(x_1, x_2)\|^2) d\Omega \tag{6.21}$$

In this case, the pair of the associated Euler-Lagrange equations is

$$\begin{aligned}\Delta u &= -\alpha I(I_{tx_1} + I_{x_1x_1}u + I_{x_1x_2}v + 2I_{x_1}u_{x_1} + I_{x_2}v_{x_1} + I_{x_1}v_{x_2} + Iv_{x_1x_2} + Iu_{x_1x_1}) \\ \Delta v &= -\alpha I(I_{tx_2} + I_{x_1x_2}u + I_{x_2x_2}v + I_{x_2}u_{x_1} + I_{x_1}v_{x_2} + 2I_{x_2}v_{x_2} + Iu_{x_1x_2} + Iv_{x_2x_2}).\end{aligned}\tag{6.22}$$

The implementation of PDEs (6.22) and (6.18) via finite differences is given in Appendix B.

6.3 Discontinuity-Preserving Regularization

In this section, we overview a variational technique for a discontinuity-preserving motion detection. In many applications, such as vehicle tracking and motion detection, the discontinuities of the motion at the boundary of moving objects are very important for an analysis of image sequences. However, as mentioned earlier in this chapter, the regularization using the smoothness constraint (See Eq. (6.17)) does not preserve the discontinuity of

the optical flow; instead, it isotropically smooths the discontinuity of the flow. Numerous methods were proposed to resolve this problem [87, 88, 89, 90, 91, 85]. Although different in details, these methods shared the same idea; that is changing the smoothness regularization term into

$$\int_{\Omega} \left[\phi(\|\nabla u\|) + \phi(\|\nabla v\|) \right] d\Omega, \quad (6.23)$$

where the function ϕ is tailored so that it encourages isotropic smoothing where the magnitude of the velocity field is small but attenuates the diffusion by smoothing kernel across the edge presented by a strong gradient. We give an example of the energy functional for this purpose used in [84, 85].

Given an image sequence $I(\mathbf{x}, t)$ we want to compute the velocity field $\mathbf{v} = (u(\mathbf{x}), v(\mathbf{x}))$ that minimizes the energy functional:

$$E(\mathbf{v}) = \int_{\Omega} |I_{x_1}u + I_{x_2}v + I_t| d\Omega + \alpha \int_{\Omega} \phi(\|\nabla u\|) + \phi(\|\nabla v\|) d\Omega + \beta \int_{\Omega} c(\nabla I) |\mathbf{v}|^2 d\Omega, \quad (6.24)$$

where α and β are positive constants. The first term of the energy functional (6.24) is the L^1 norm of the OFC (6.4). The second term is the smoothing regularizer. The last term in the energy functional (6.24) forces the flow field to be zero in homogeneous regions where there is no texture. To guarantee the well-posedness of the problem, the function ϕ in this term must satisfy the following hypotheses:

- ϕ is strictly convex, nondecreasing function from \mathbb{R}^+ to \mathbb{R}^+ , with $\phi(0) = 0$
- $\lim_{s \rightarrow \infty} \phi(s) = \infty$
- There exist two constants $c > 0$ and $b \geq 0$ such that $cs - b \leq \phi(s) \leq cs + b$, $\forall s \geq 0$.

To clearly see how to choose the function ϕ , we observe that the Euler-Lagrange equation associated with the second term of the energy functional (6.24) gives the diffusion operator of the form

$$\operatorname{div} \left(\frac{\phi'(\|\nabla s\|)}{\|\nabla s\|} \nabla s \right). \quad (6.25)$$

Here the function s represents either u or v in the second term of the functional (6.24). As done in Aubert [84, 85], we can decompose the action of this operator into the tangent and

normal directions to the isophote line, that is the normal direction at point \mathbf{x} is defined by $N(\mathbf{x}) = \frac{\nabla s}{\|\nabla s\|}$ and the tangent direction is in the orthogonal direction to $N(\mathbf{x})$. Then we can rewrite Eq. (6.25) as

$$\text{div}\left(\frac{\phi'(\|\nabla s\|)}{\|\nabla s\|}\nabla s\right) = \left(\frac{\phi'(\|\nabla s\|)}{\|\nabla s\|}s_{TT} + \phi''\|\nabla s\|s_{NN}\right), \quad (6.26)$$

where s_{TT} and s_{NN} denote the second derivatives of s in the tangential and normal direction, respectively. Therefore, if we would like to promote isotropic smoothing in the region of a low variation, we may require that

$$\phi'(0) = 0, \lim_{s \rightarrow 0} \frac{\phi'(s)}{s} = \lim_{s \rightarrow 0} \phi''(s) = \phi''(0) > 0. \quad (6.27)$$

On the other hand, in a neighborhood of a region with a strong gradient, which represents the edge, we encourage smoothing only in the tangential direction and attempt to nullify the smoothing in the normal direction. We can achieve this by imposing the following conditions:

$$\lim_{s \rightarrow \infty} \phi''(s) = \lim_{s \rightarrow \infty} \frac{\phi'(s)}{s} = 0 \text{ and } \lim_{s \rightarrow \infty} \frac{\phi''(s)}{\phi'(s)/s} = 0. \quad (6.28)$$

6.4 PIV Approach

A different approach for the velocity field estimation by image technique is the particle image velocimetry (PIV) technique. PIV is an important research area and widely used in fluid mechanics to quantitatively extract relevant instantaneous velocity field from image pairs. In this technique, illuminated particles (or passive tracers) are sowed on the plane under investigation. The image is then visualized via a laser sheet pulse associated with a camera. Then the cross-correlation technique of local interrogation window is used to estimate the motion from the observed particles. Specifically, the $m \times n$ pairs of subregions of two consecutive images, $I_1(\mathbf{x}), I_2(\mathbf{x})$, are interrogated by the spatial cross-correlation window $W(\mathbf{x})$ of size $m \times n$ as

$$C(u(\mathbf{x}), v(\mathbf{x})) = \frac{1}{\|W(\mathbf{x})\|} \sum_{\mathbf{y} \in W(\mathbf{x})} I_1(\mathbf{y}, t) \times I_2(\mathbf{y} + \mathbf{d}(\mathbf{x})). \quad (6.29)$$

Here, $W(\mathbf{x})$ is the interrogation window centered at the point \mathbf{x} on image plane and $\mathbf{v}(\mathbf{x})$ is the velocity vector at \mathbf{x} . The peak of the cross-correlation $C(u(\mathbf{x}), v(\mathbf{x}))$ is then used as the estimation of the velocity field. Roughly speaking, the cross correlation function is used to determine the displacement (or velocity) vector that results in the best “match” of the image pair within the interrogation region. Therefore, as opposed to the global nature of variational methods, the PIV approach is considered as a local method. In practice, the computation of the above cross-correlation is performed by the Fourier method based on FFT or the spectral method, which allows the real-time computation. However, as a consequence of using the Fourier method, the aliasing problem may arise if the Nyquist criteria is not satisfied, that is, if the particles in the interrogation region move a distance larger than half the size of the interrogation region ($d > \min\{m, n\}$). Therefore, a larger interrogation region or a higher sampling rate of image sequence may be necessary to avoid the aliasing problem. Although there have been numerous researches leading to improvement of the PIV based on the cross-correlation method, all those techniques suffer from some common limitations (see Corpetti, Loureño and Krothpalli [? 21]) due to the local nature of the cross-correlation method.

First, the finite size of the interrogation areas restricts a spatial scale where the motion variation is assumed negligible. Also, if there is a loss of particles in some interrogation windows, the correlation becomes inaccurate due to the loss of pairing. In addition, choosing a size of interrogation areas can be problematic since it must not be too small to detect correlation peak reliably. However, a large interrogation size reduces the resolution of the estimated velocity vector since the cross correlation results in only one velocity field in each interrogation areas, and it also presents a bias toward the subregion with higher seeds due to more possibilities of pairing. Nevertheless, this problem can be partially ameliorated by using iterative deformable grids.

The above problems will be experimentally demonstrated in Section 6.6 and compared with the variational method. Due to the global nature of variational methods, all of the aforementioned problems of the correlation-based PIV approach can be (at least partially) overcome.

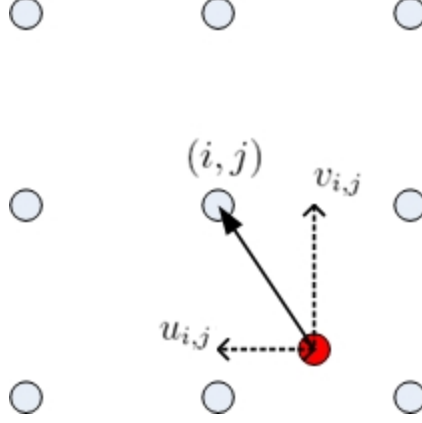


Figure 6.2. The process of the backwards warping.

6.5 Coarse-to-Fine Approach

When large displacements occur between two consecutive images, the motion estimation based on the energy functional (6.17) or (6.34) may not be suitable due to its differential nature. To overcome this problem we resort to the so-called “multiresolution” or “coarse-to-fine” strategies, which were introduced by Black [92] and has been widely used in several tasks of image processing to remedy the aliasing problems.

The basic idea is to use the pyramid representation of the image sequence. Starting from a coarse scale (low spatial frequency resolution) at the top of the pyramid we compute a coarse displacement field and use it to “backwards warp” the original second image toward the first image. By backwards warping, we mean that the velocity field at each pixel is treated as the incoming velocity field to each pixel in the second image, and so the tail of velocity field tells us where the pixel image comes from (see Figure 6.2). This process can be expressed by

$$I(x_1, x_2, t) \leftarrow I(x_1 - u\Delta t, x_2 - v\Delta t, t + \Delta t). \quad (6.30)$$

Then, we can use a bilinear (or bi-cubic spline) interpolation to construct the backwards warped image. The “forwards warping” scheme can also be performed in a similar fashion. However, we will use the backwards warp as our choice for this work.

At the next finer level we compute the “correction” field between the first image at

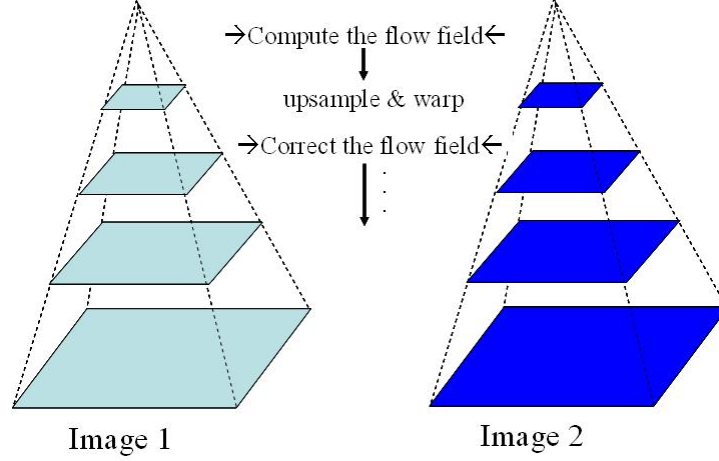


Figure 6.3. The original image is at the bottom of the pyramid (the highest level). Each lower level are the subsample version of the previous level. Starting from the lowest level the velocity is approximated and then upsampled to the next level. The second image is then warped toward the first image based on the upsampled version of the velocity field. The incremental velocity field is then computed. The process is repeated until the highest level is reached.

this level and the warped version of the second image based on the previously computed velocity field. Therefore, we only compute small displacement fields at this level. Then we use the correction field to update the velocity field computed from the previous level. This correction process, sometimes called “motion increment”, is repeated for each level until the finest level (original images) is reached.

Let us now discuss the pyramid representation of image and the computation of the velocity field based on this pyramid representation. We denote the size of a generic image, I , by $n_x \times n_y$. Let L be the number of level of the pyramid representation. For $i = 0, 1, \dots, L-1$ we denote by n_x^i and n_y^i the number of pixel columns and rows of the image at level i , I^i , respectively. Here we use the 0^{th} level image to represent the original image, i.e. $I^0 = I$, $n_x^0 = n_x$ and $n_y^0 = n_y$. Then the image I^1 in the next lower resolution level is subsampled by a bilinear interpolation. The new image pixel is located in the middle of four neighbor pixels as illustrated in Figure. This is called *dyadic* pyramid structure.

Consider the model problem (6.17). The coarse-to-fine procedure starts from the lowest resolution level where we compute the velocity field between image I_1^{L-1} and I_2^{L-1} based on the Euler-Lagrange equation (6.18). The resulting velocity field are then upsampled to

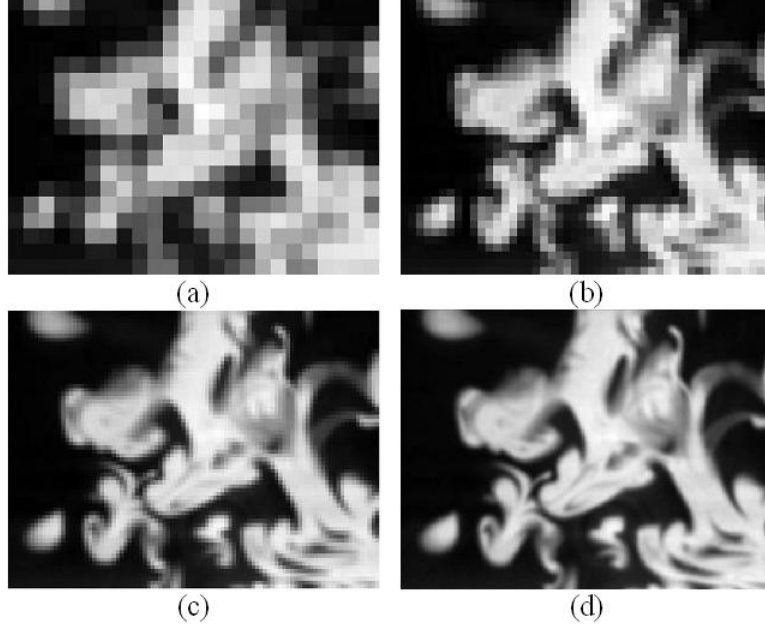


Figure 6.4. A sample image at different level of the pyramid. The original image is shown in (d).

the next higher resolution level using the bilinear interpolation. Then the image I_2^{L-2} is warped toward the first image based the upsampled velocity field, i.e.,

$$W_2^{L-2}(x_1, x_2, t) = I_2^{L-2}(x_1 - u^{L-1}\delta t, x_2 - v^{L-1}\delta t, t + \delta t), \quad (6.31)$$

where (u^{L-1}, v^{L-1}) is the upsampled version of the velocity fields computed from the previous pyramid level. This warped image allows us to compute the small increment of velocity field, denoted by $\delta \mathbf{v}^{L-2} \equiv (\delta u^{L-2}, \delta v^{L-2})$, between the images I_1^{L-2} and W_2^{L-2} . Therefore, the complete velocity at this level is obtained by

$$\mathbf{v}^{L-2}(x_1, x_2) = \mathbf{v}^{L-1}(x_1, x_2) + \delta \mathbf{v}^{L-2}, \quad (6.32)$$

where δu^{L-2} and δv^{L-2} are computed by minimizing the energy functional:

$$E(\delta u^{L-2}, \delta v^{L-2}) = \int_{\Omega} (I_{x_1} \delta u^{L-2} + I_{x_2} \delta v^{L-2} + I_t)^2 d\Omega + \alpha \int_{\Omega} (\|\nabla u^{L-1} + \delta u^{L-2}\|^2 + \|\nabla v^{L-1} + \delta v^{L-2}\|^2) d\Omega. \quad (6.33)$$

Notice that we minimize $u^{L-1} + \delta u^{L-2}$ instead of δu^{L-2} in the smoothness regularization since the smoothness assumption is required for the complete velocity field. This motion increment process is then repeated for each level of the pyramid until the level of original resolution is reached.

Similarly, when applying the coarse-to-fine procedure to the energy functional (6.34), the complete velocity field at each level of the pyramid can be computed by minimizing the energy functional:

$$E(\delta u^{L-k}, \delta v^{L-k}) = \int_{\Omega} \left(\frac{\partial I}{\partial t} + \text{div}(I \mathbf{v}^{L-k}) \right)^2 d\Omega + \alpha \int_{\Omega} (\|\nabla u^{L-k-1} + \delta u^{L-k}\|^2 + \|\nabla v^{L-k-1} + \delta v^{L-k}\|^2) d\Omega, \quad (6.34)$$

where $k = 2, \dots, L$.

6.6 Experimental Evaluation

First, we consider cases of computer-generated image sequences, which are obtained from solving some PDEs, in the first two examples of this section. For these examples, image pixels in consecutive images do not present a very strong loss or gain of luminance due to a large divergence of velocity field, which is true in a video sequence with a high frame rate, and so the classical brightness constancy equation is our preferred choice. Next we consider some real-world image sequences of fluid imagery. It should be noted that for these sequences of fluid imagery, fluid may exhibit a high temporal deformation and large displacement between two consecutive frames, and the brightness constancy cannot be nearly satisfied. Also, the motion between two consecutive frames could be large and the coarse-to-fine approach may be necessary. Hence, for these cases, the development of the minimization technique embedded by Eq. (3.28) with the coarse-to-fine strategy becomes more attractive [20, 21, 27].

In the first two examples, we will solve the minimization problem Eq. (6.8) using the brightness constancy constraint Eq. (6.4) for the data fidelity term and the div-curl regularizer Eq. (6.10) for the regularization term:

$$\inf_{\mathbf{v}} \int_{\Omega} (I_x u + I_y v + I_t) d\Omega + \int_{\Omega} \alpha \|\nabla \text{div} \mathbf{v}\|^2 + \beta \|\nabla \text{curl} \mathbf{v}\|^2 d\Omega, \quad (6.35)$$

The associate Euler-Lagrange equation for minimization of the problem Eq. (6.35) is the

following pair of PDEs:

$$\begin{aligned}\alpha u_{xx} + \beta u_{yy} + (\alpha - \beta)v_{xy} &= I_x(I_x u + I_y v + I_t) \\ \beta v_{xx} + \alpha v_{yy} + (\alpha - \beta)u_{xy} &= I_y(I_x u + I_y v + I_t)\end{aligned}\tag{6.36}$$

with homogeneous Neumann boundary conditions:

$$\begin{aligned}\frac{\partial u}{\partial N} &= 0 & \text{on } \partial\Omega \\ \frac{\partial v}{\partial N} &= 0 & \text{on } \partial\Omega.\end{aligned}\tag{6.37}$$

Herein, N is an outward vector normal to the image boundary $\partial\Omega$. We will see in the first example (our benchmarking example) that the velocity field obtained from the smoothness condition ($\alpha = \beta$) is more accurate than general cases where $\alpha \neq \beta$.

6.6.1 Example 1

Consider the following differential equation as the first example:

$$\begin{aligned}\frac{dx}{dt} &= -\sin(2\pi x)\cos(2\pi y) \\ \frac{dy}{dt} &= \cos(2\pi x)\sin(2\pi y).\end{aligned}\tag{6.38}$$

Now we may use the flow continuity equation (3.28) to numerically translate an initial distribution in time to generate a synthetic movie data set $I(x, y, t)$ representing a time evolving spatial density function. Figure 6.5 illustrates the velocity field of this system and a given initial distribution function and Figure 6.6 shows a sequence of some images that are captured from our numerical simulation of Eq. (3.28). We compute this simulation by using the Fourier spectral method [93, 94] for the differentiation in space and the Jameson-Schmidt-Turkel Runge-Kutta time integrator [95].

Based on this sequence of images we benchmark our approach by comparing the approximated optical flow field extracted from the images by the regularization method described in the previous sections and the exact velocity field in Eq. (6.38). First, let us investigate the sensitivity of the method with respect to variation of the parameters α and β defined in Eq. (6.10). We quantitatively evaluate our result via the average angular error (AAE):

$$e_{ang} = \arccos\left(\frac{u_{ex}u_e + v_{ex}v_e + 1}{\sqrt{(u_{ex}^2 + v_{ex}^2 + 1)(u_e^2 + v_e^2 + 1)}}\right),\tag{6.39}$$

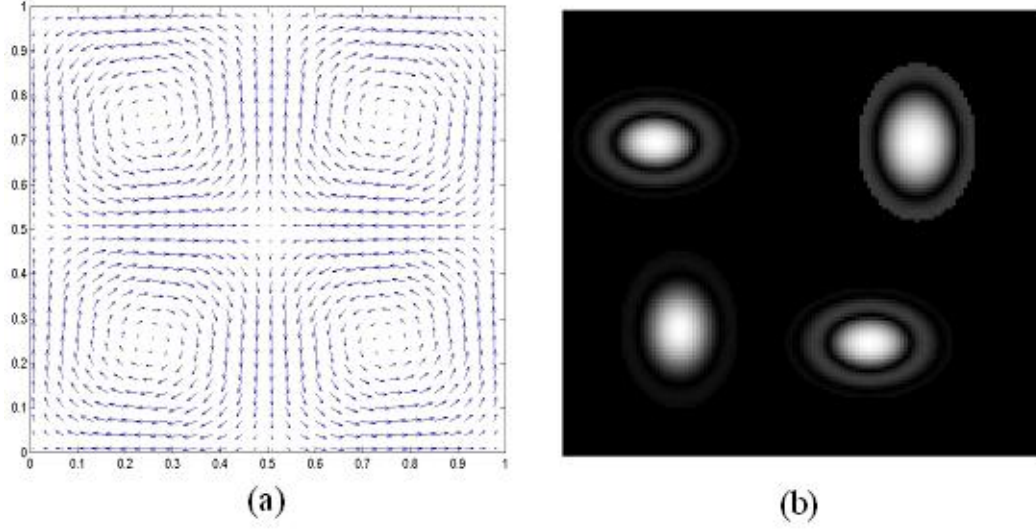


Figure 6.5. (a) The velocity field of the dynamical system (6.38) defined on the unit box $[0, 1] \times [0, 1]$. It can be seen that there are four basins separated by two nullclines $x = 0.5$ and $y = 0.5$. (b) The initial distribution at time $t = 0$.

where (u_{ex}, v_{ex}) denotes the exact velocity field and (u_e, v_e) is the estimated velocity. Note that we do not evaluate the error via a norm because in general we would lack information about the physical domain of the image plane and we can only recover the velocity field whose dynamics is in some representative of the true dynamical system, perhaps in the sense of “almost conjugacy” [96]. Observe that for the image sequence in this example the smoothness regularizer performs better than the div-curl regularizer at the experimentally optimal value $\beta = 0.01$, see Figure 6.7. Nevertheless, the div-curl regularizer produces better results for other values of β . To evaluate the results qualitatively we lay the exact velocity field on the estimated velocity fields for $\alpha = \beta = 0.01$ and $\alpha = 1, \beta = 0.01$ as illustrated in Figure 6.6.1. We notice that the div-curl regularizer does not isotropically smooth the velocity field; hence we obtain only a velocity field where the motion take places meaning that there is a nontrivial support of the dataset $I(x, y, t)$. However, the smoothness regularizer produces the velocity field throughout the whole image region, notwithstanding an absence of motion in some regions, see also Figure 6.5. Moreover, we observe that the approximated velocity field from the smoothness constraint maintains qualitative qualities of the exact velocity fields. Therefore, we want to investigate the topological structure of the result. Note that the phase portrait of the exact velocity field has four centers with

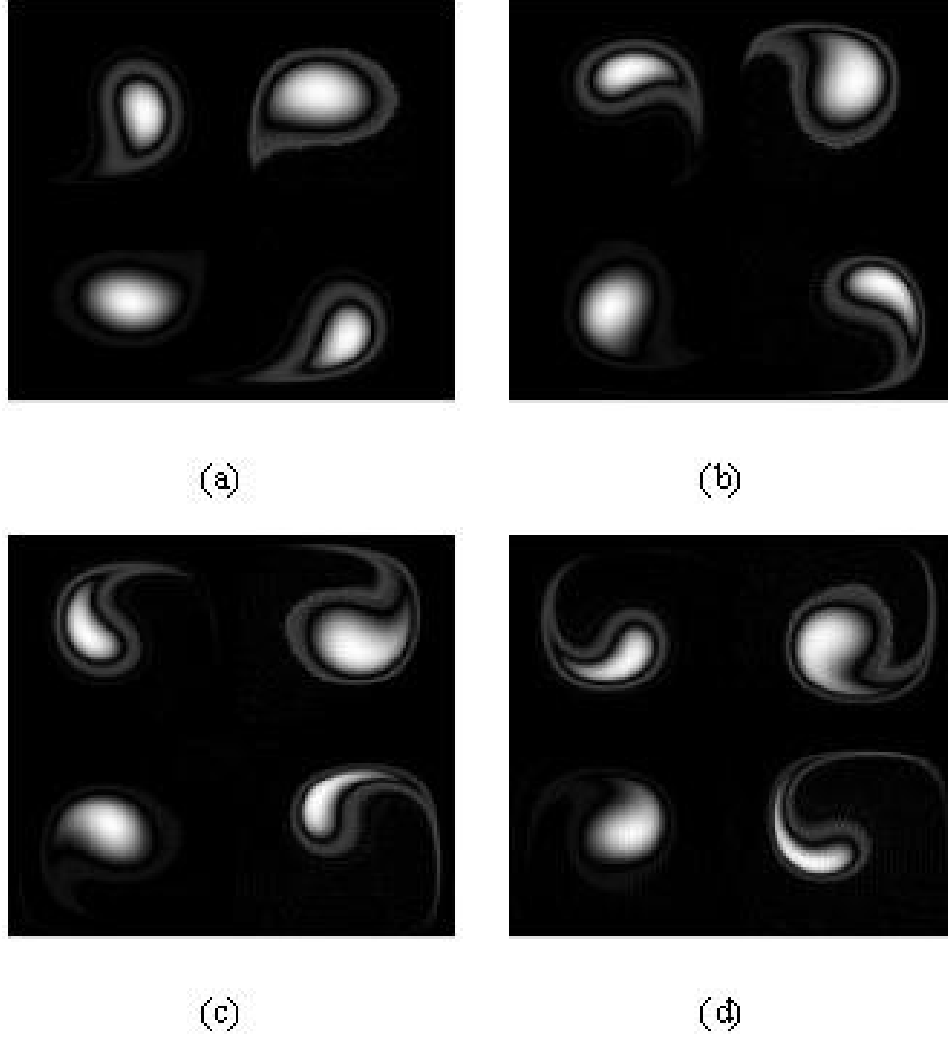


Figure 6.6. A sequence of some images captured from the numerical simulation of the flow continuity equation Eq. (3.28) with the velocity field given by the ode Eq. (6.38). The time increment between each snapshot is $t = 32$.

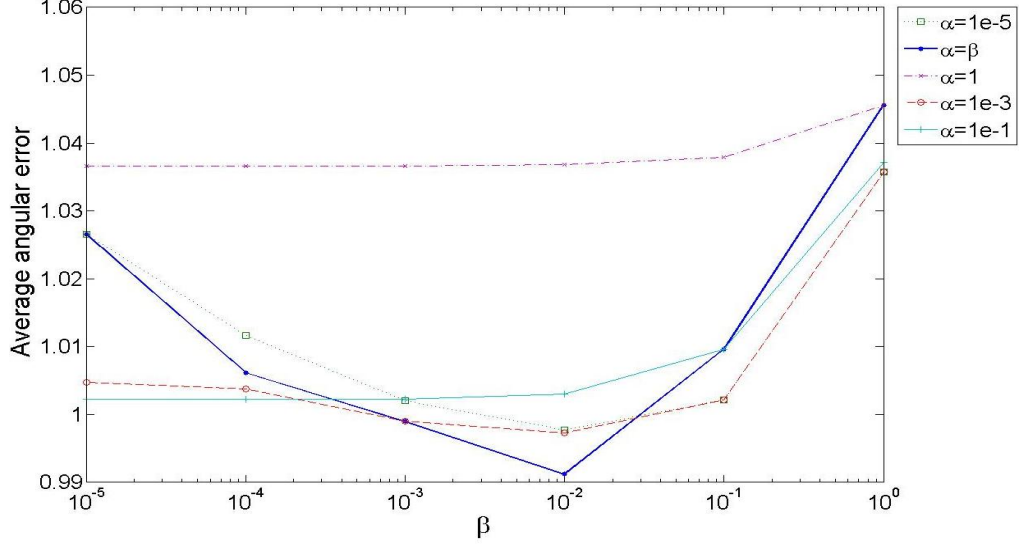
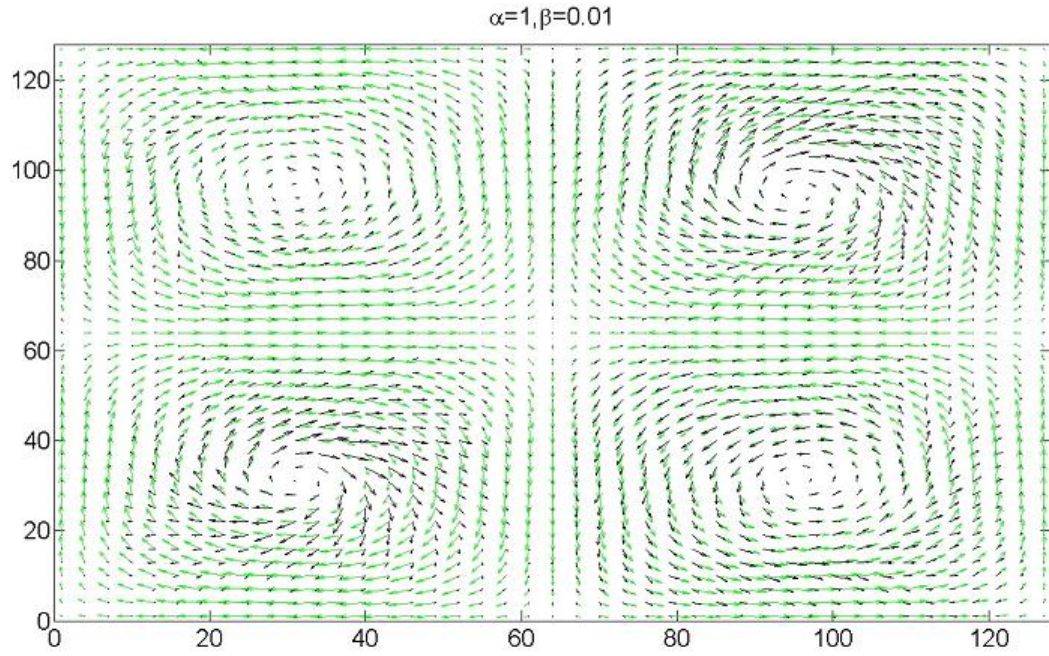


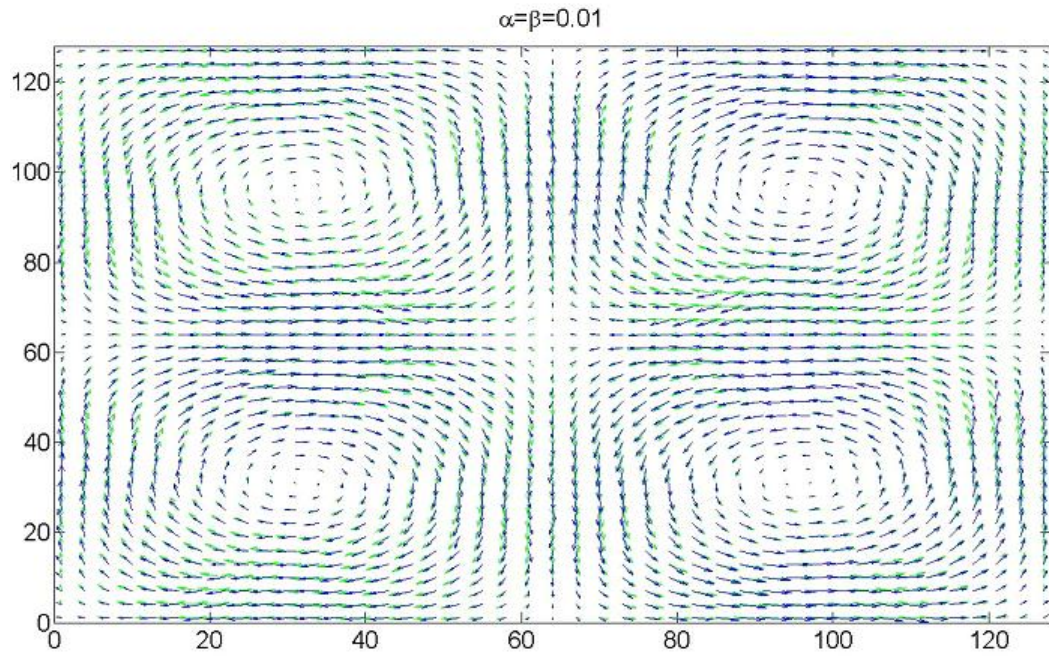
Figure 6.7. A plot of the average angular error with respect to parameters α and β . The sensitivity of the parameter α and β is observed in this plot. The thick line represents the case of the smoothness constraint Eq. (6.9) where $\alpha = \beta$. Although it is not conclusive that the smoothness constraint performs better than the div-curl regularizer in this experiment, our experimental result demonstrates that the optimal value is obtained when using the smoothness regularizer with $\alpha = \beta = 0.01$

eigenvalues $\pm 2\pi i$ at points $\{(0.25, 0.25), (0.25, 0.75), (0.75, 0.25), (0.75, 0.75)\}$ and a saddle point with eigenvalue $\pm 2\pi$ at the central point $(0.5, 0.5)$ of the phase space. We compute the eigenvalues of the points in the image region corresponding to those points for the velocity field approximated from the smoothness constraint with $\alpha = 0.01$ and plot the result in Figure 6.10. The result shows that we recover the saddle node at the central point of the image region, but we obtain four spiral sinks at those points instead of center nodes; however, the real part of their eigenvalues are much smaller than the imaginary parts, which indicates that they are suggestive of a type of center node.

Now let us discuss how to apply the optical flow result to approximate the Ulam-Galerkin approximation of the Frobenius-Perron transfer operator. We use the approximated velocity field to advance randomly distributed points in time to generate a finite-rank approximation of the Frobenius-Perron operator as described in Section 3.3. Figure 6.13 shows the Ulam-Galerkin matrix generated by the exact velocity fields in Eq. (6.38) and by the approximated optical flow field. As seen in Figure 6.5 there exist four basins and hence we may expect the Ulam-Galerkin matrix to be diagonalizable to four diagonal blocks. To reveal this



(a)



(b)

Figure 6.8. (a) Exact velocity field (in green arrows) and the optical flow field for $\alpha = 1, \beta = 0.01$ shown in black arrows. (b) Exact velocity field (in green arrows) and the optical flow field for $\alpha = \beta = 0.01$ shown in black arrows.

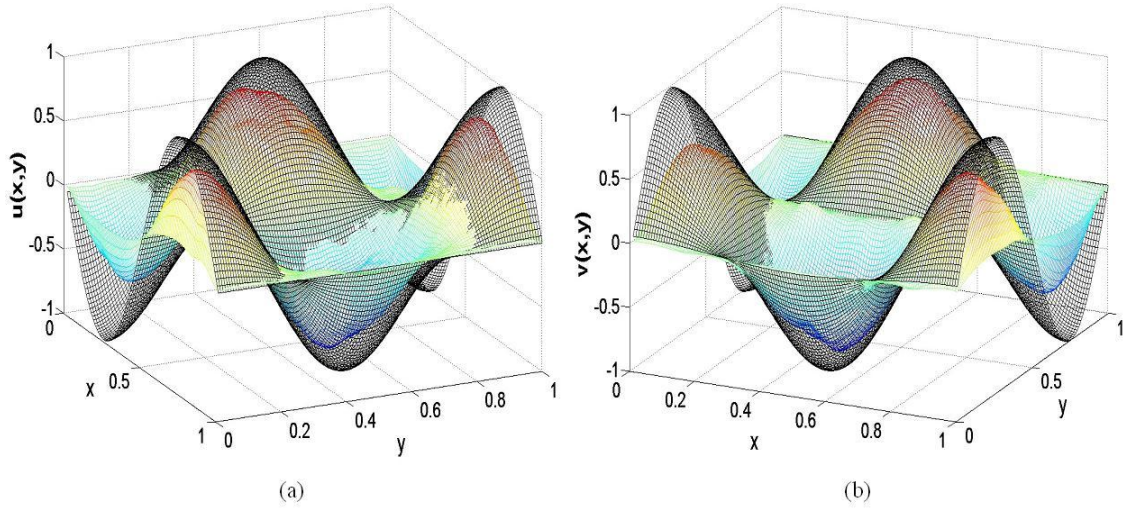


Figure 6.9. Comparison between the exact and approximated velocity fields from the image sequence using the smoothness parameter $\alpha = \beta = 0.01$. The transparent solid lines illustrate the exact velocity field and the colored plots represent the approximation.

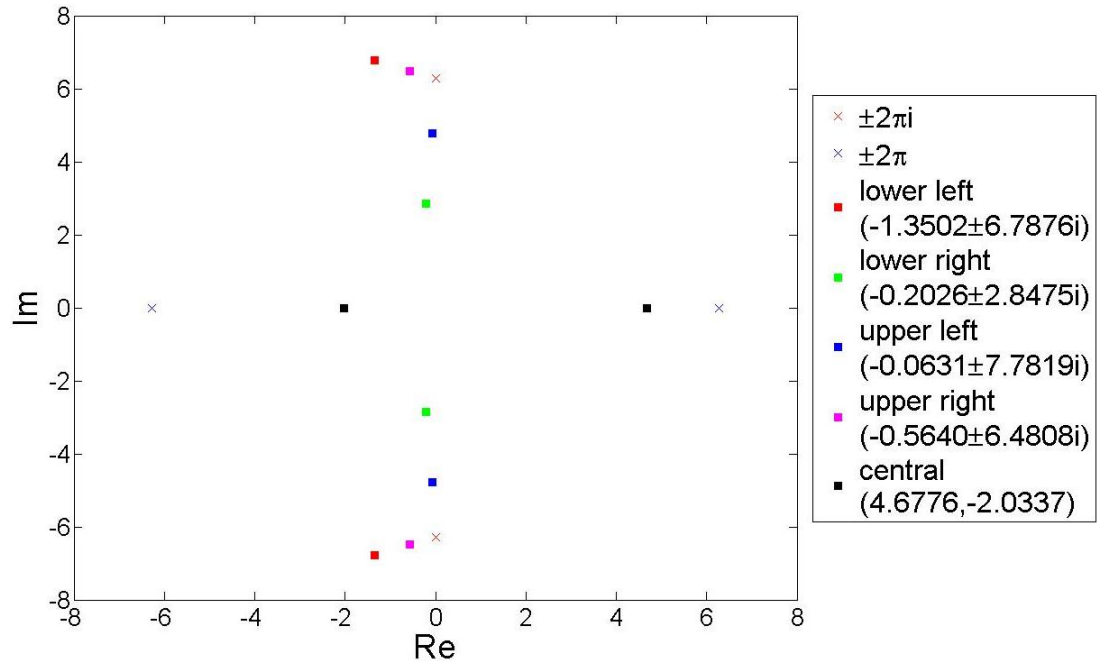


Figure 6.10. The plot shows the eigenvalues of the centers and the saddle node in the phase space of the exact velocity field in cross markers. The square markers locate the eigenvalues of those points computed from the estimated velocity field with the smoothness constraint $\alpha = 0.01$.

correct block-diagonal form based on the modularity method we first partition the phase space into small triangles and index them to generate a graph network that represents the transport between these triangles. However, to obtain a meaningful graph network we need to determine the time duration of the trajectory of the initial points. If the time duration is too short, some initial points may remain in the same grid in which they initially reside, see Figure 6.11. This circumstance induces some self-connected edges in the graph network that do not provide useful information for a graph partition method. Nevertheless, a self-connected edge generated by a grid that contains a fix point cannot be avoided. A useful criteria to choose the time duration can be formulated using the well-known Gronwall's inequality [97].

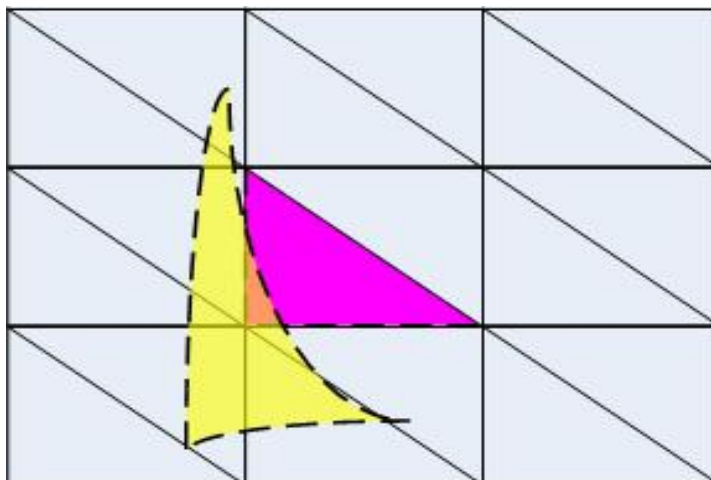


Figure 6.11. When solving Equation (6.1) the time duration has to be chosen in a way that the (triangular) grid is expanded far enough so that it lie across itself as less as possible. Then the solution of the majority of randomly chosen points will not remain inside the initial grid. This prevents us from generating a graph network with inapplicable self-connected edges.

Now our purpose is to discover community structure in the graph network to identify the existing basin structure in the phase space. Therefore, our point of view is to map the problem of phase space partition of the dynamical system captured through a sequence of image into a problem of partitioning a graph network representing the action of the transfer operator. Note that although points in the phase space may not travel across from one basin to another, it is possible to have edges in the graph that connect between each community;

this represents a basin in the phase space, due to the grids that lie across the basin boundary as seen in Figure 6.12. However, this is only a small boundary effect. Likewise, there can be a small amount of spurious basin leakage due to finite grid effects.

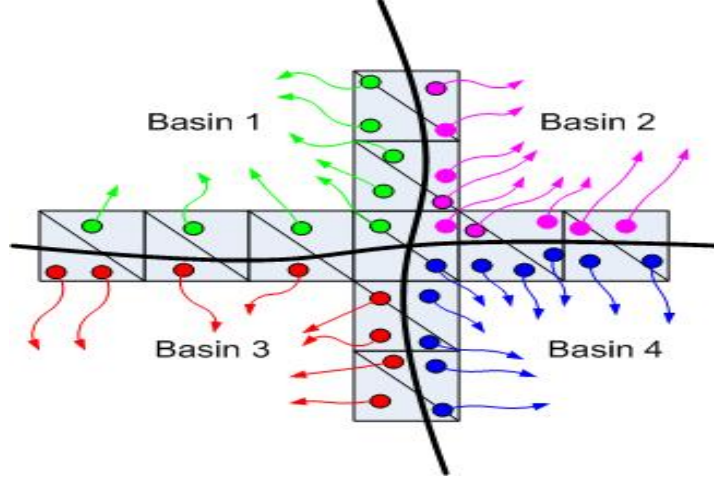


Figure 6.12. The phase space in this figure are partitioned by triangular grids. The edges that connect the nodes corresponding to the triangles that lie across the basin boundary are the intergroup edges, which connect between each community in the graph network.

After generating a graph network corresponding to the dynamical system, we can apply the modularity method to discover the community structure of the graph network and use this structure to reorder the Ulam-Galerkin matrix. The results of the Ulam-Galerkin after reordering are shown in Figures 6.14. They reveal to us the four basins and the sorted Ulam-Galerkin matrix generated by the exact velocity is now in the correct block-diagonal form, whereas the sorted matrix generated by the optical flow is in the “almost” block-diagonal form due to the imperfection of the approximated velocity field that causes the transport between the basins.

6.6.2 Example 2

Our next example is a simulation result obtained from a numerical solution of the complex Ginzburg-Landau equation (CGLE). Figure 6.15 illustrates the initial distribution and a time series of brightness patterns. By a careful observation one may expect that there exists two sinks in the phase space located by the arrows in Figure 6.15. Since the ana-

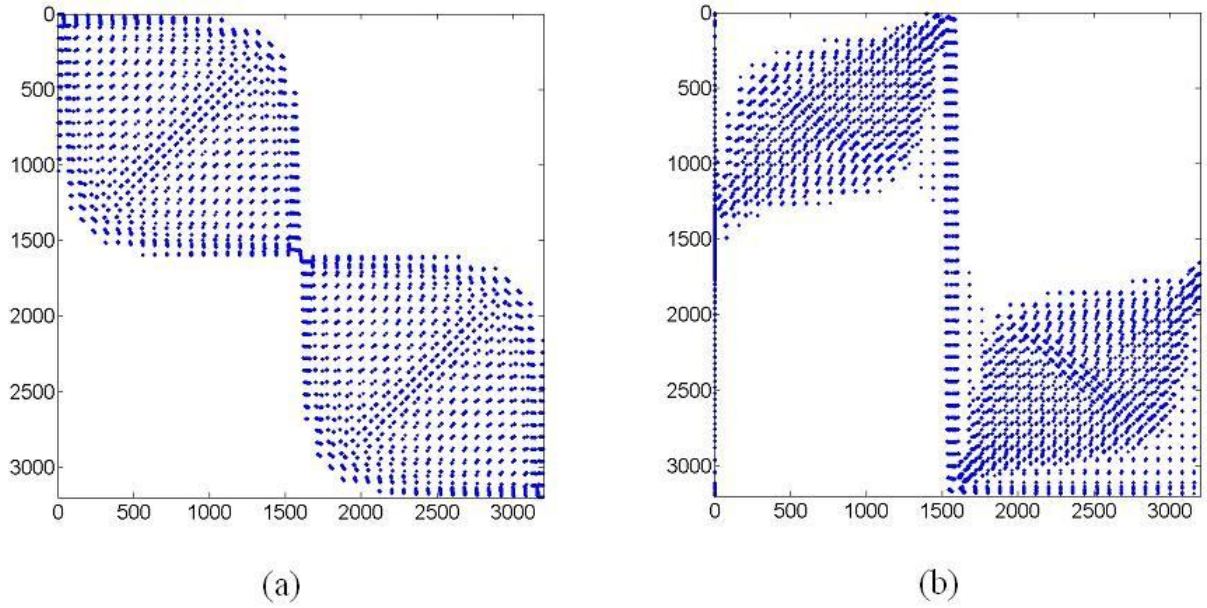


Figure 6.13. The Ulam-Galerkin matrix generated by the exact velocity fields expressed in Equation 6.38 and by the approximated optical flow field using 200000 randomly chosen points.

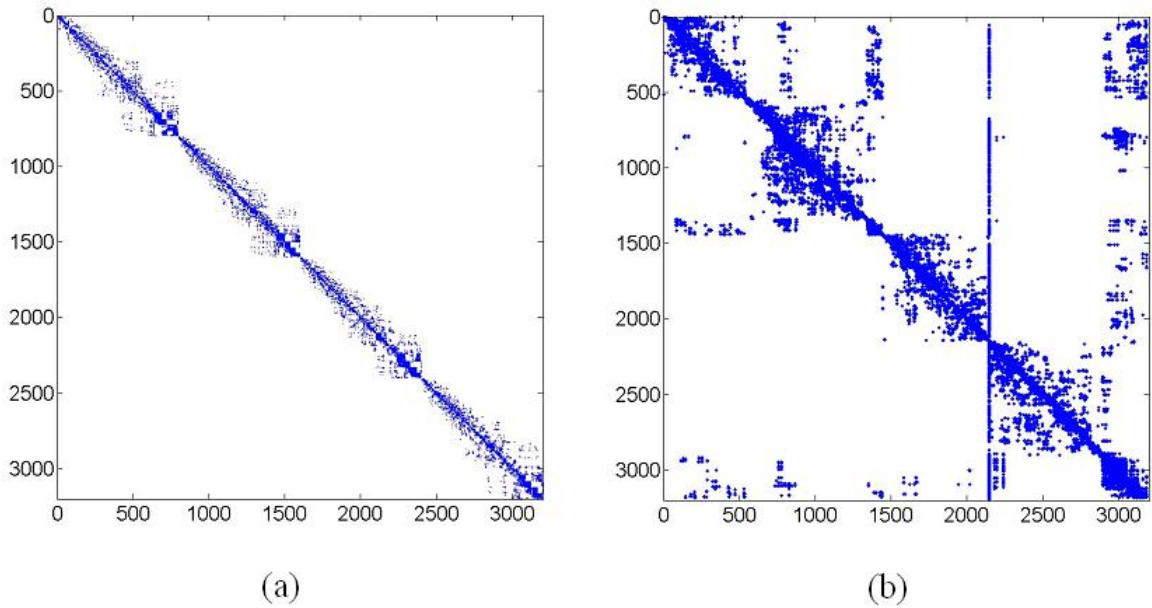


Figure 6.14. The Ulam-Galerkin matrix after sorting using the modularity method.

lytic solution for this example is not priory known, we ascertain our result only qualitatively.

The estimated velocity fields with different values of parameters β but fixed $\alpha = 1$ are shown in Figure 6.16. We observe that when β is small, the velocity field at the region where occlusion occurs is strongly emphasized compared to other regions in the image whereas the magnitude of the velocity field is distributed more isotropically throughout the image region while increasing the values of β . To have a better view of this phenomenon, we plot in Figure 6.17 the contour line of the magnitude of the velocity onto the image of the initial distribution. We also notice that the velocity field becomes discontinuous where the occlusions take place. Nonetheless, we still observe the discontinuity of velocity field in those regions in the case of large β . Note that similar results can also be obtained when β is fixed and α is varied.

Next, we compute the matrix approximation of the Frobenius-Perron operator shown in Figure 6.18. Then we apply the modularity method to the graph generated from the semigroup of transformation of the ode (6.1) associated with the estimated velocity field. We obtain two large communities and several trivially small communities from the result. Thus we group all those small communities together. Subsequently, we use the new ordering based on the community structure to reordered the Ulam-Galerkin matrix to reveal the “almost” block diagonal form, see Figure 6.18, and we also paint the phase space according the community structure discovered by the modularity method as shown in Figure 6.19, where the two large communities are painted in different colors and those small communities are not painted. Notice that there are three blocks in sorted matrix, where the middle block corresponds to those of several small communities that we group them together and the other two blocks represent the two large communities.

6.6.3 Example 3

In this section, two pairs of successive PIV images published by the Visualization Society of Japan (VSJ) are used for the experimental evaluation of our variational method in

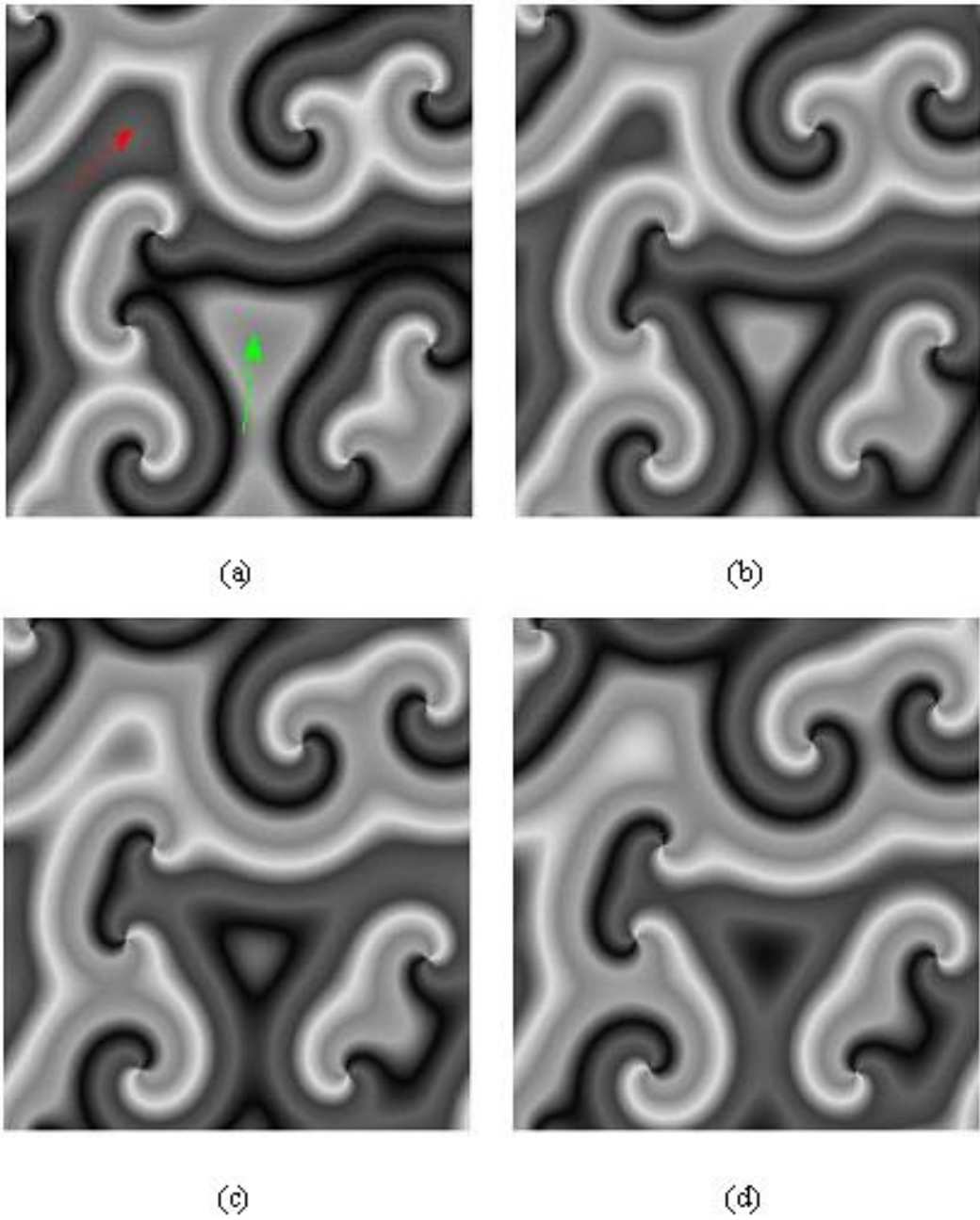


Figure 6.15. A sequence of some images captured from the numerical simulation of the Complex Ginzburg-Landau equation. The arrows point at the sinks that can be found through a careful observation and as revealed by the Ulam-Galerkin matrix analysis as in Figure 6.16- 6.19

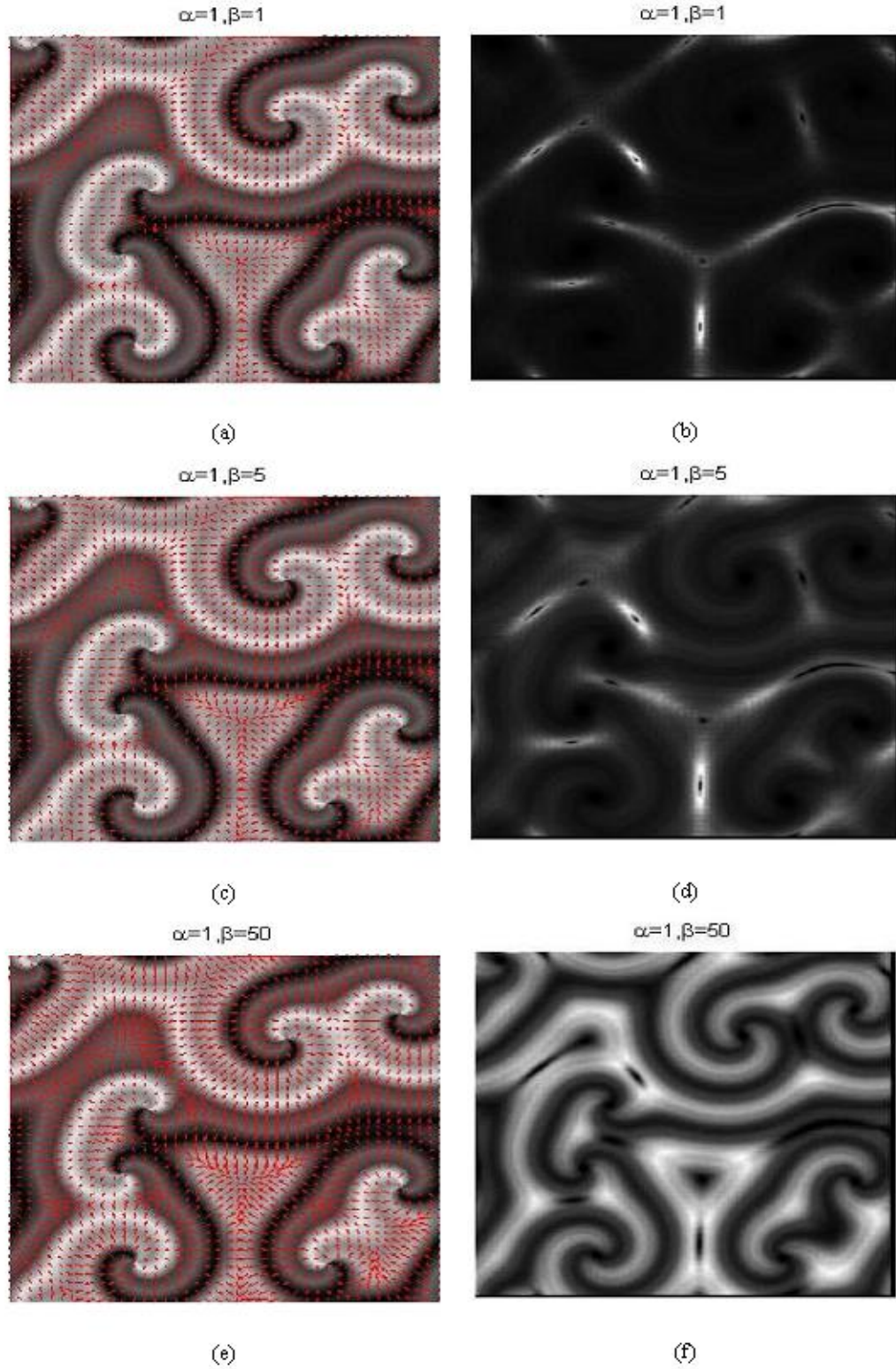


Figure 6.16. The velocity fields approximated from the div-curl regularization with various values of parameters α and β are plotted side by side with the corresponding magnitude of the velocity fields.

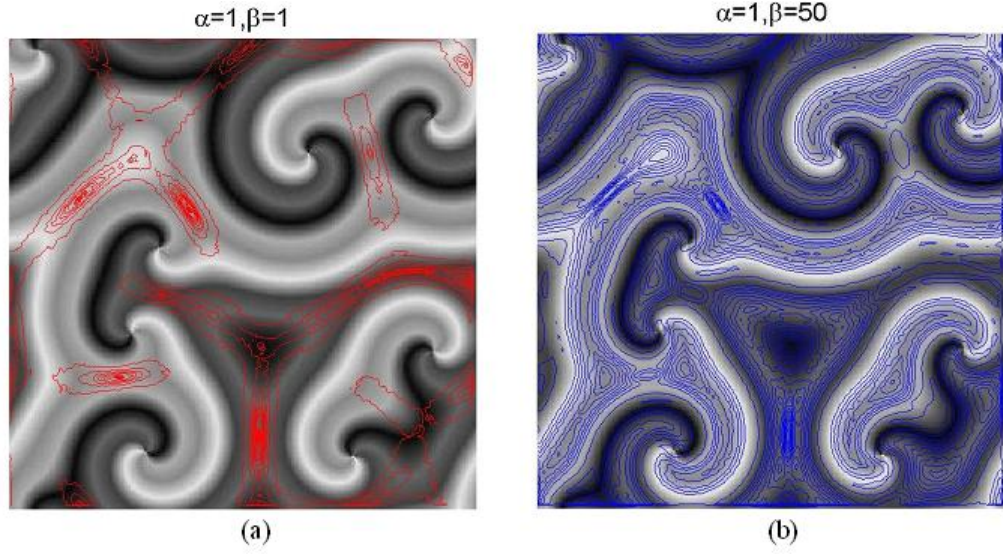


Figure 6.17. (a) For a small β at $\alpha = 1$, the magnitude of the velocity field is more emphasized at the region where occlusion occurs. In this region the velocity become discontinuous. (b) For a large β at $\alpha = 1$, we observe that the velocity field is isotropically distributed throughout the image region. Nonetheless, the discontinuity can still be observed at the region where occlusion take places.

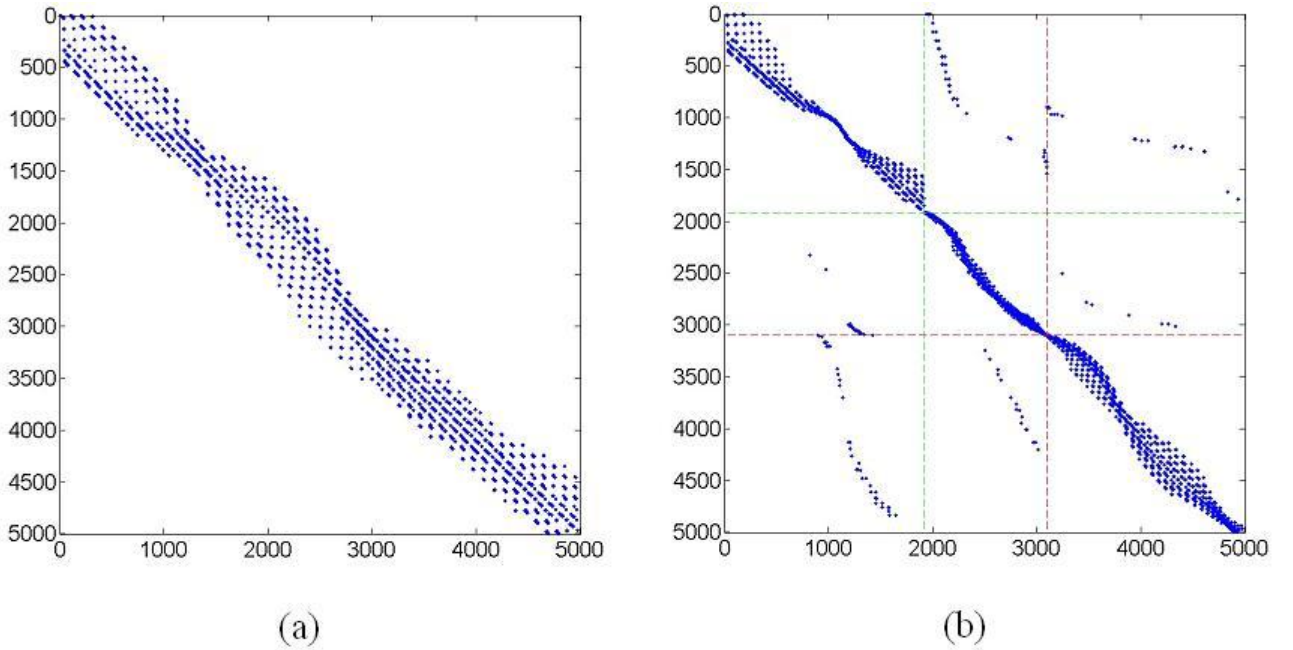


Figure 6.18. The Ulam-Galerkin matrix before and after sorting. Notice that the “almost” block diagonal form is unveiled after sorting. The middle block corresponds to those of several small communities that we group together. The other blocks represent the two large communities.

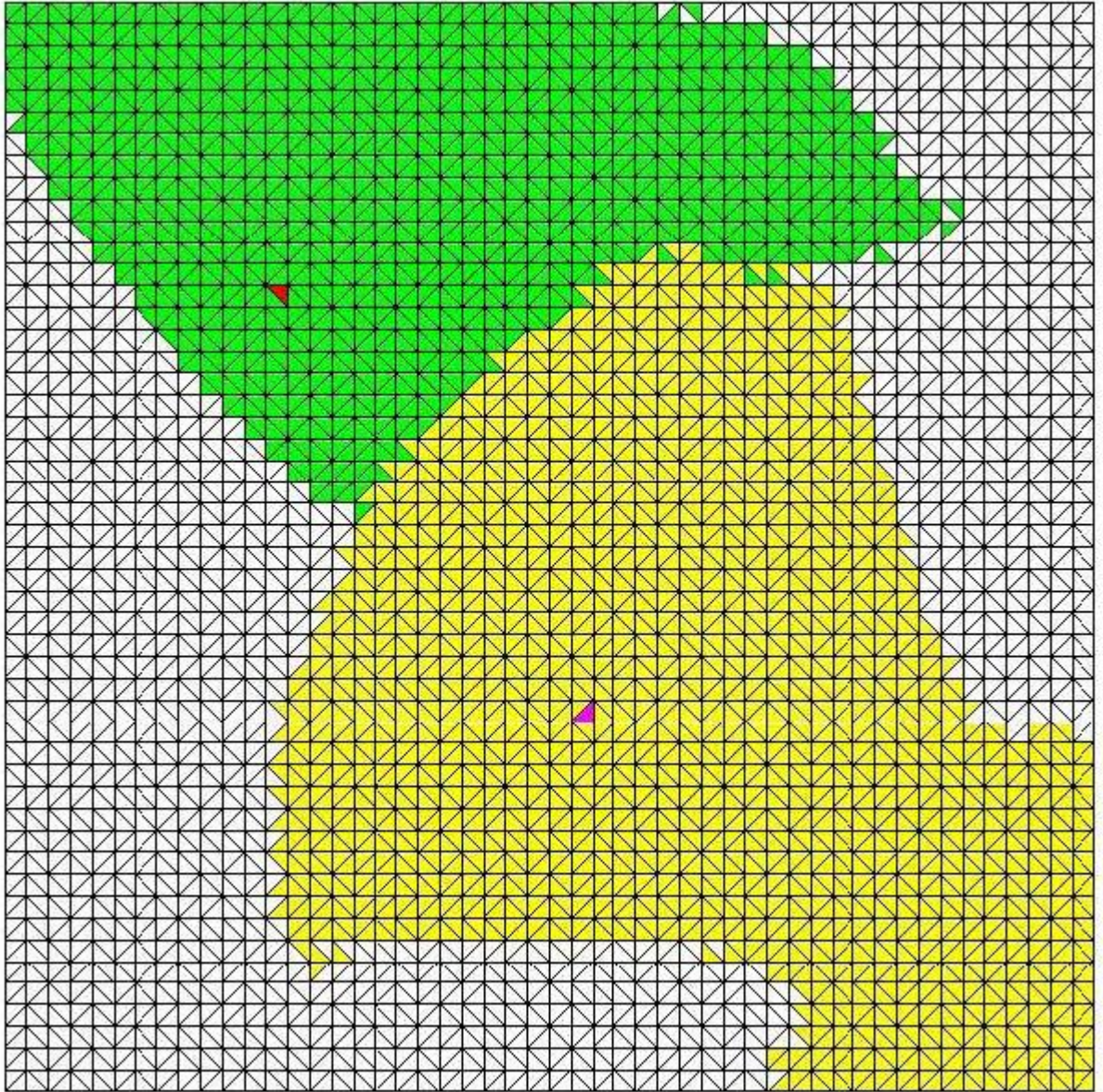


Figure 6.19. The phase space are partitioned into three regions corresponding two basins of the sinks corresponding to two communities discovered by the modularity method and the region that an initial point does not converge to neither basin.

comparison with the cross-correlation and adaptive cross-correlation methods. Note that both methods are computed using the FFT. Figure 6.20 shows these two pairs of images, which are computer-generated by the same velocity field and have a size of 256×256 pixels. These two image pairs are different in the number of particles, the particle diameters and the standard deviation of particle diameters. The exact velocity field is shown in Figure 6.21.

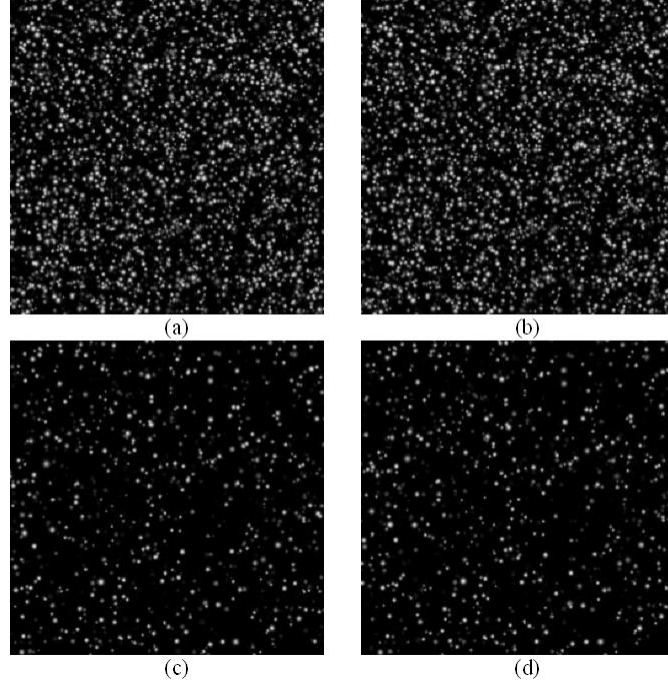


Figure 6.20. Two pairs of images with differences in the number of particles, the particle diameter and the standard deviation of particle diameters. The number of particles for the pair (a)-(b) and (c)-(d) are 4000 and 1000, respectively.

Figure 6.22 compares the results of our variational techniques and the cross-correlation and adaptive cross-correlation methods. Note that both methods based the cross correlation yield sparse velocity field as previously explained in Section 6.4, whereas our variational approach gives dense velocity field at every pixel locations. However, for purposes of comparison and visualization, the velocity fields resulted from the variational approach are shown only at every other 8 pixels and those obtained from the cross-correlation techniques are interpolated to 32×32 pixels. Qualitatively, the results from the three methods for the case of the image pair with dense particles (Figure 6.22(a)-(c)) are similar. In the case of the image pair with sparse particles, visually comparing the results of the three ap-

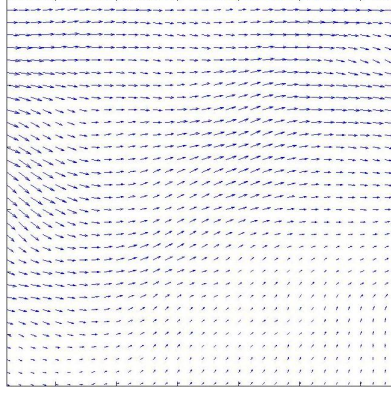


Figure 6.21. Exact velocity field of the two image pairs shown Figure 6.20.

Table 6.1. Average angular error (AAE) and average relative displacement error (ADE) in the case of dense particle images.

Error	Adaptive cross correlation	Cross correlation	Variational
AAE: Dense	0.0024	0.0038	0.0029
AAE: Sparse	0.0078	0.0445	0.0076
ADE: Dense	0.0017	0.0027	0.0020
ADE: Sparse	0.0030	0.0164	0.0050

proaches shows that the variational and adaptive cross-correlation approaches yield much better results than the cross-correlation technique without adaptive interrogation windows. Furthermore, both cross-correlation and adaptive cross-correlation methods fail to provide the vector fields near the boundary because of their use of a local interrogation window. Table 6.6.3 summarizes the average angular error (AAE) and average relative displacement error (ADE), which is computed using the L_2 error. Comparing the three methods, we see that the adaptive cross-correlation method provides slightly better results than the variational method. Also, both methods yields much better results than the cross-correlation method. Note that the results of the variational technique reported here are computed by using 4 pyramid levels and setting the smoothness parameter λ to be the optimal one for $10\lambda \leq 15$ (See Figures 6.23 and 6.24).

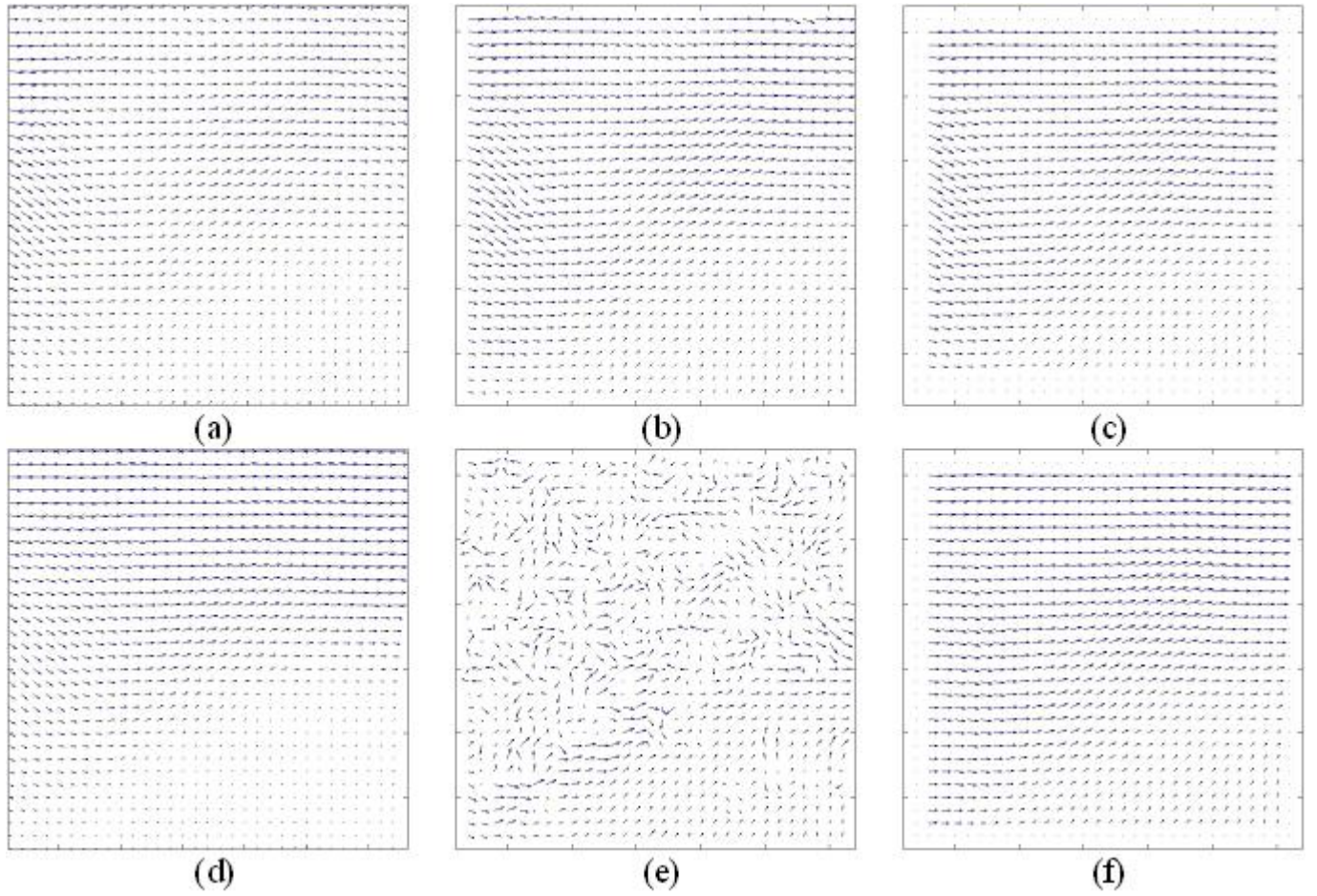


Figure 6.22. (a)-(c) show the vector fields obtained from the variational, cross-correlation, adaptive cross-correlation approaches for the image pair with dense particles, respectively. (d)-(f) compare the resulted obtained from the variational, cross-correlation, adaptive cross-correlation approaches for the image pair with sparse particles, respectively.

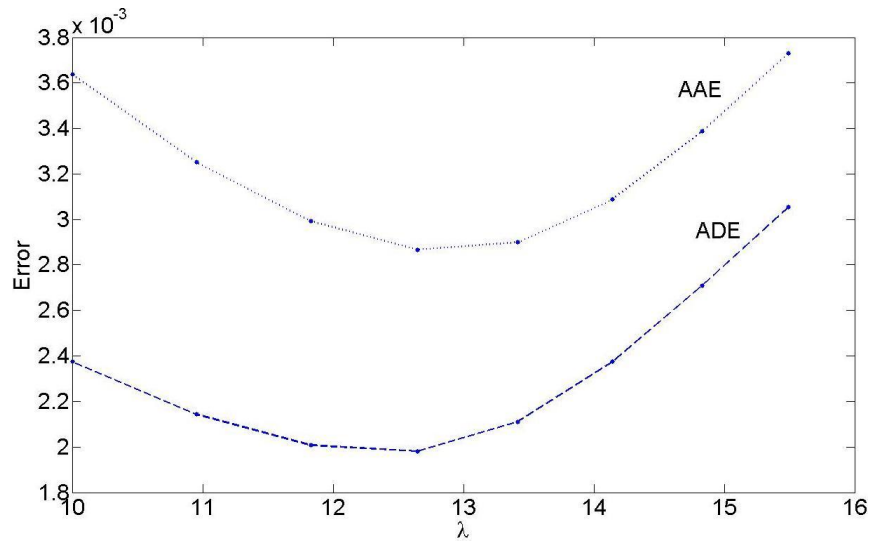


Figure 6.23. Average angular error (AAE) and average relative displacement error (ADE) for the image pair with dense particles.

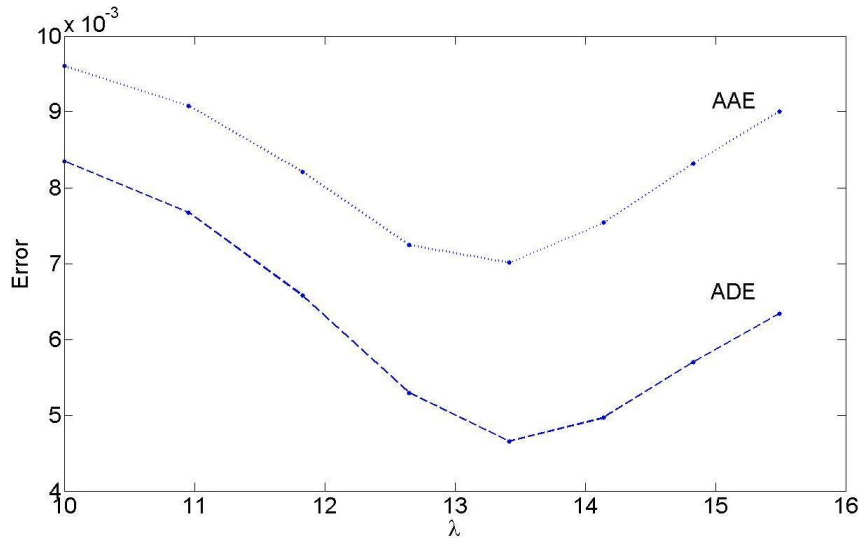


Figure 6.24. Average angular error (AAE) and average relative displacement error (ADE) for the image pair with dense particles.

6.6.4 Example 4

The data set to be evaluated in this section are published in the Computer Vision, Graphics, and Pattern Recognition (CVGPR) website. The sequence of 20 PIV images are recorded in water with another fluid with a certain fraction of carbonoxy snarf. Three sets of fluids with three different ratio of carbonoxy snarf show slightly different intensity pattern, but they all correspond to the same flow field. It is suggested by CVGPR group that the difference between the three flow fields gives some measure of the computation errors. Note also that the actual flow is three-dimensional.

First of all, we compare our results with the velocity fields obtained from the direct correlation-based method conventionally used for the PIV images. Note that the cross-correlation and adaptive cross-correlation methods in the preceding section fail to provide reasonable results for comparison. We first consider the first two images of the sequence as shown in Figure 6.25. A comparison between the velocity fields obtained from a PIV method

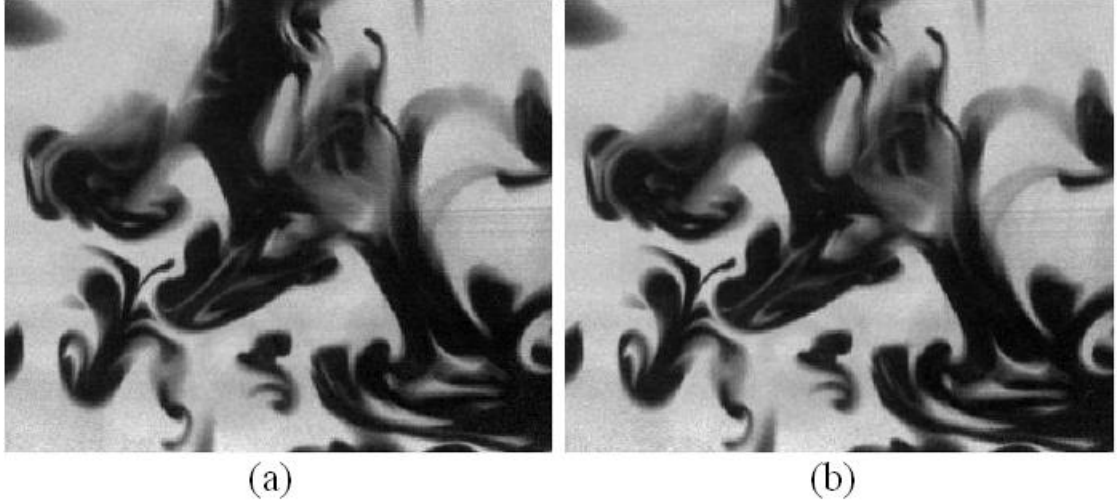


Figure 6.25. Raw experimental images: the first two frames of the images recorded from an experiment. The entire image sequence is published by Wieneke in CVGPR website as a test data for the PIV method.

and those computed from our variational technique (6.34) with various values of smoothness parameter is shown in Figure 6.26(d)-(f). As we can see, the variation approach based on the Frobenius-Perron operator provides a velocity field that is more applicable to a dynamical system analysis than the velocity field obtained from the cross-correlation-based PIV. We

also observe the impact of the smoothness parameter to the discontinuity of the velocity fields. We note that we use the coarse-to-fine strategy to compute the velocity field in this example with 4 pyramid levels. Figure 6.26(b) shows the velocity field obtained at the top of the pyramid level (lowest resolution level), which is to be compared with Figure 6.26(f) to see the final result at the lowest level (at the original resolution of the image). Particularly, we see from Figure 6.27 that there is no noticeable vorticity near the left image boundary of the velocity vector at the highest pyramid level (lowest resolution) whereas the vorticity become apparent at this location for the next lower pyramid level (higher resolution). In addition, we also show the velocity field computed from the variation technique with $\alpha = 10$ but without using the coarse-to-fine strategy in Figure 6.26(c). It is evident that without the coarse-to-fine strategy the method fail to provide a useful velocity field due to the large motion between two consecutive frames.

Now we consider three different image sequences of which the first frame of each sequence has slightly different patterns as shown in Figure 6.28. These three images are recorded from the experiments as described above, but their difference in the pattern is due to the differences ratio of carbonoxy snarf. In other word, Figure 6.28 shows three different initial conditions of the evolution described the Frobenius-Perron operator. However, all these initial conditions are propagated by the same velocity field. Therefore, the difference of their velocity field can be used a measure of the accuracy of the method.

We evaluate our results based on the angular errors (6.39) between the velocity fields obtained from the three image sequences. We use the velocity fields computed from the first and second images of each sequence. The result is shown in Figure 6.29 and it is evident that the difference of the velocity fields becomes asymptotically smaller as the smoothness parameter α increases.

In addition, we observe the Lagrangian coherent structure of the vector field computed from the above image sequence. Note that unlike the first two examples, the vector field is neither an autonomous system nor a stochastically perturbed autonomous system. Therefore the analysis of coherent structure is based on approximating the Lagrangian coherent structure. The results are shown in Figure 6.30. The FTLE field is computed for frames

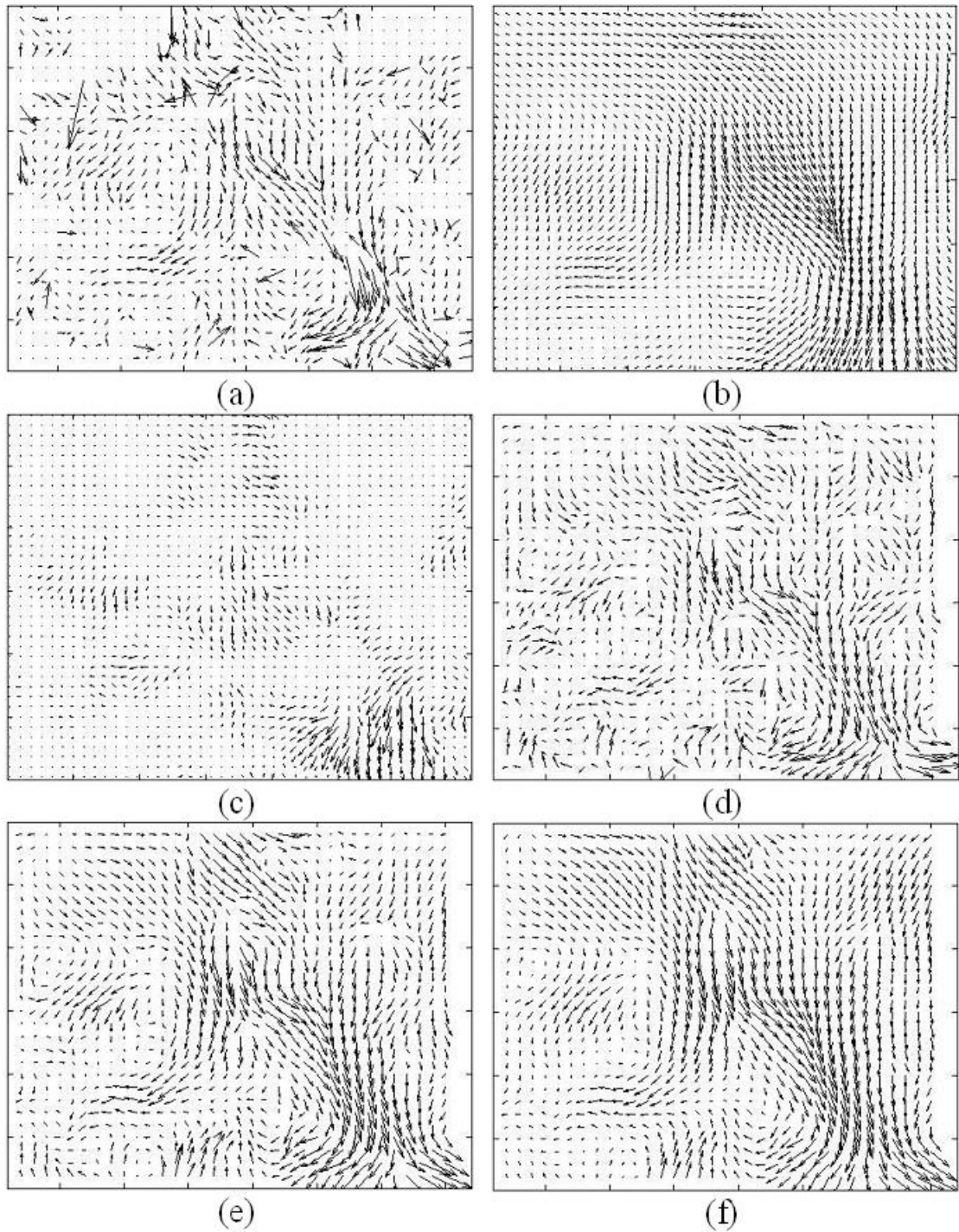


Figure 6.26. (a) correlation-based PIV method. (b) The velocity field obtained from the lowest resolution level with the smoothness parameter $\alpha = 10$. (c) Frobenius-Perron based variational technique without using the coarse-to-fine scheme. From (d)-(f) The velocity fields are computed by the variational minimization based on the Frobenius-Perron operator with the coarse-to-fine strategy. (d) $\alpha = 1$. (e) $\alpha = 4$. (f) $\alpha = 10$. Note that the each velocity field is amplified by a factor of 2 to make it more perceptible.

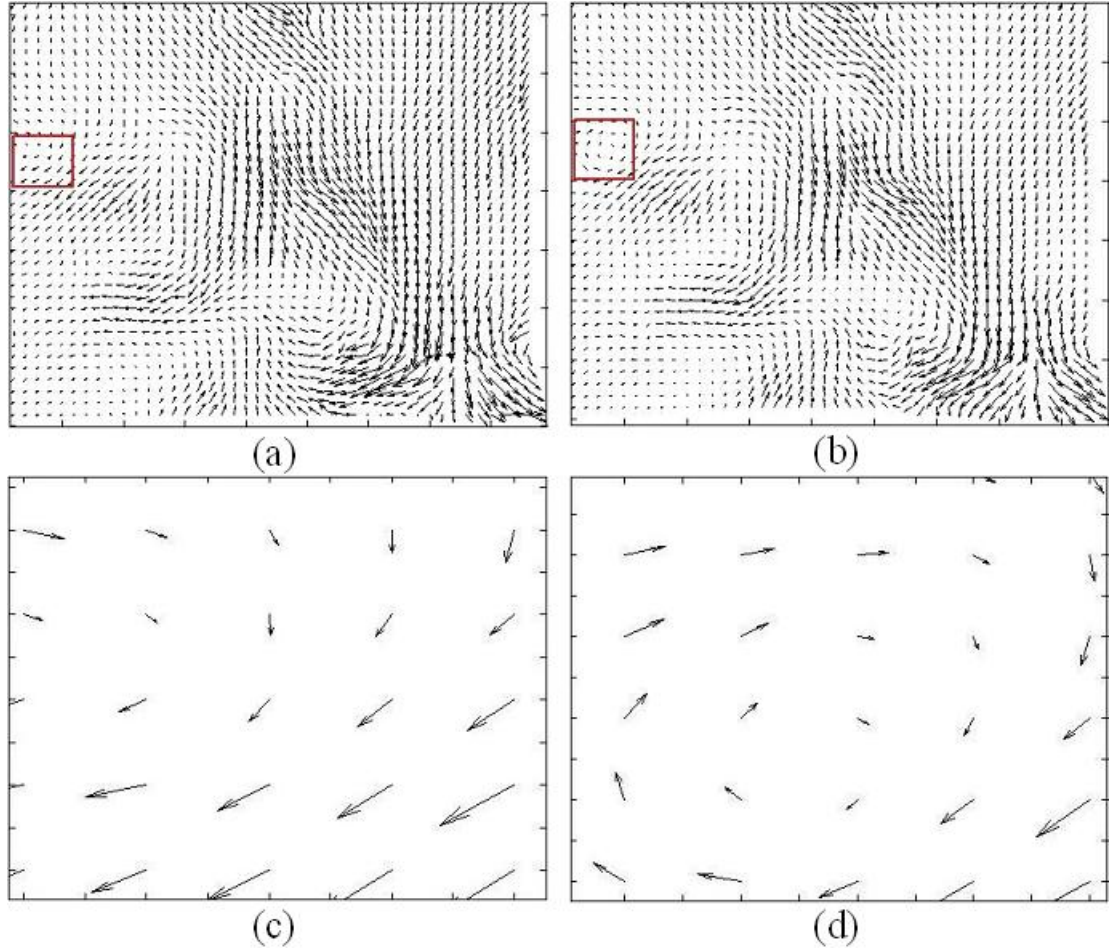


Figure 6.27. (a) The velocity vectors obtained at the highest pyramid level (lowest resolution). (b) The velocity vectors obtained from the next lower pyramid level (higher resolution). The location bounded by the red rectangle in (a) and (b) are enlarged and shown in (c) and (d) to observe the appearance of the vorticity motion when the motion increment computed at the current level is used to correct the velocity vector obtained from the previous level.

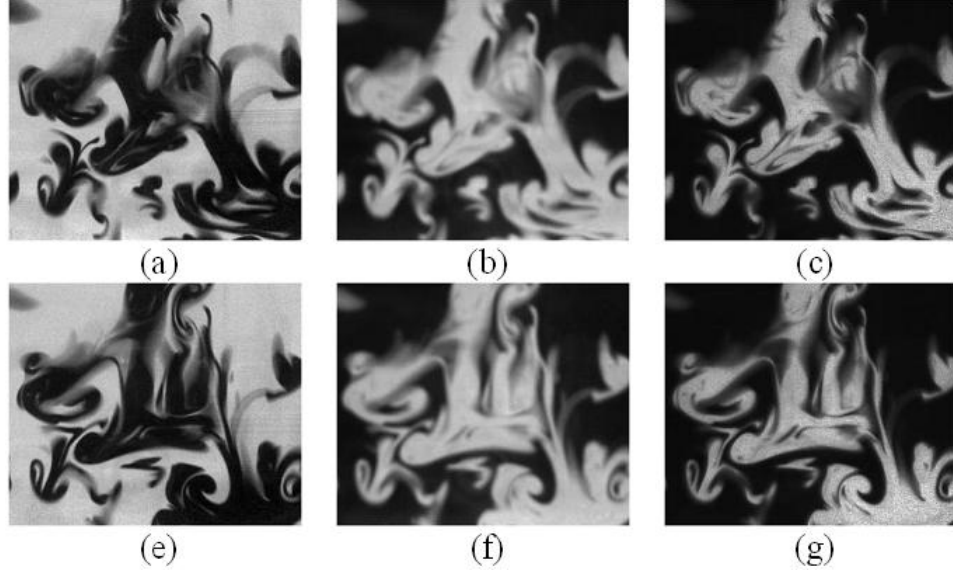


Figure 6.28. Three images of three slightly different patterns. However, the evolution of the pattern is governed by the same velocity field. (a)-(c) The first images of the three image sequences. (d)-(e) The final images of the three image sequences.

1 to 5 in Figure 6.30(b)-(f). We show in Figure 6.30(a) the separation rate at the frozen time of the frame 1, that is, we compute the FTLE field under the assumption that the vector field at frame 1 is autonomous. It is evident that the Eulerian structure in Figure 6.30(a) is different from the Lagrangian structure in Figure 6.30(b). Since the vector field is non-autonomous the passive particle could transverse across the Eulerian structure, but not the Lagrangian structure. Note that the integration time for these results are 15 frames and we assume that the time interval between two consecutive frames is 0.25 unit. Here, We choose the time interval based on our experiment to be the time interval, in which the coherent structures become relatively apparent and there is no new structure emerging to be observed when increasing the time interval. Also, we assume that the physical space is the unit box $[0, 1] \times [0, 1]$.

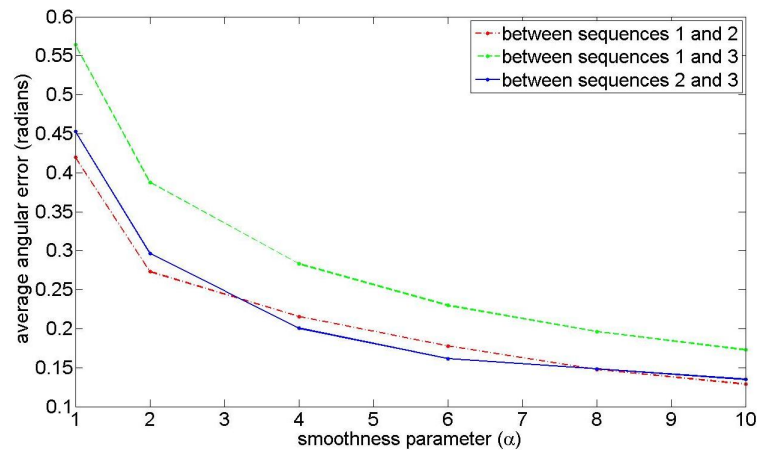


Figure 6.29. The variation of the average angular errors between the velocity fields of the three image sequences with respect to the smoothness parameter α .

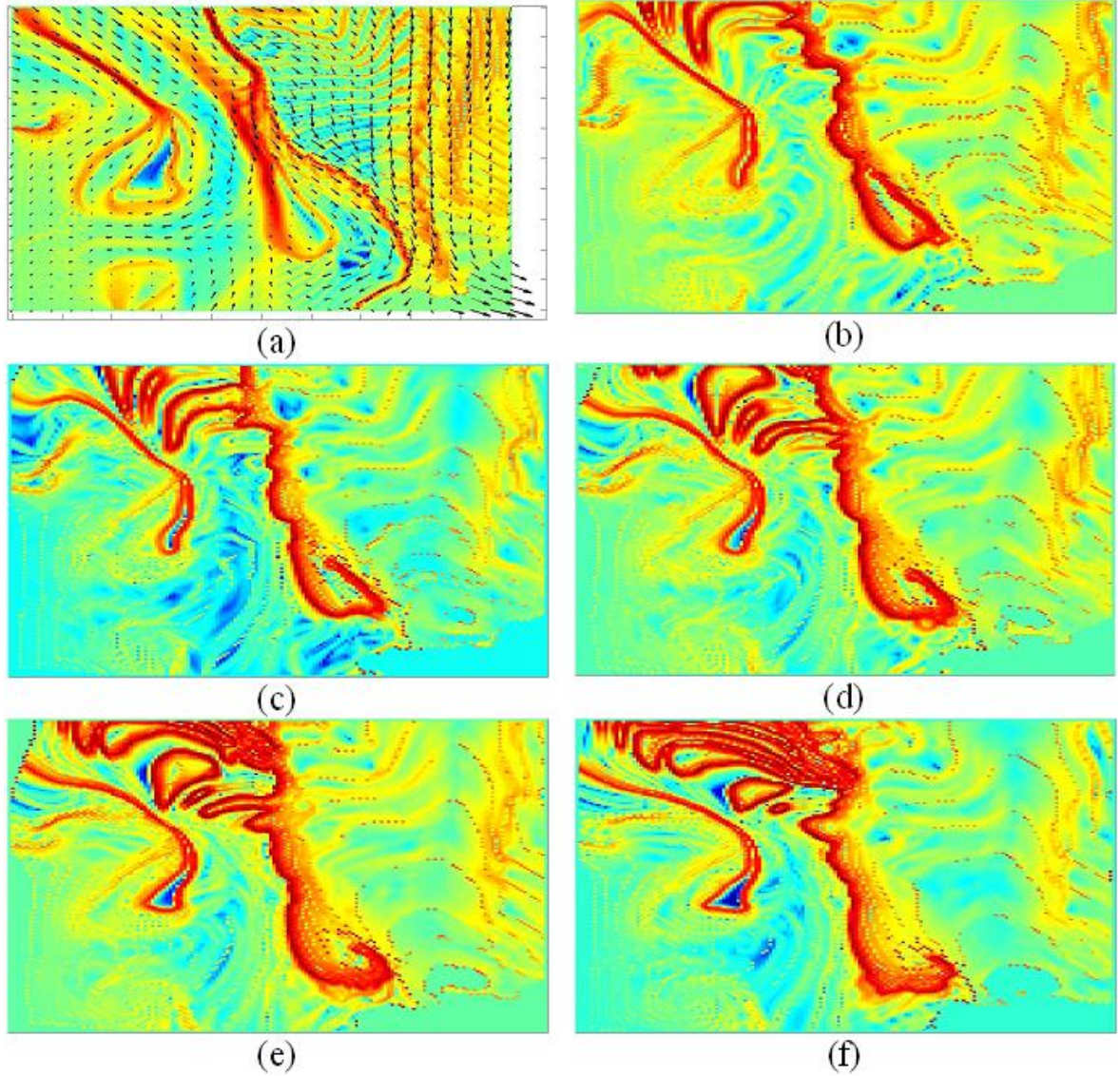


Figure 6.30. (a) The separation rate of the vector field of the frame 1. (b) The FTLE field of the frame 1. When comparing (a) and (b), we see that the separation rate computed at a frozen time can be misleading to indicate the coherent structure of the dynamical system. (b)-(f) The evolution of the FTLE field from frames 1 to 5. Notice that there are some small closed-loop coherent structures emerging as time progresses.

Chapter 7

Conclusion

This thesis can be divided into two main themes. The first is the problem of approximating the almost invariant set of a dynamical system known only through an experimental time-series data. Therefore, we only have a discrete-time realization of a dynamical system with/without stochastic perturbation, and so our data analysis is done in the framework of the transition matrix, which describes the transition probability of a particle in a given phase space. In this framework, we employ the modularity method to discover a phase space partition into almost invariant sets.

The second theme of this thesis is the problem of approximating a vector field from an image sequence, in which we assume that the dynamical system underlying the motion in the image sequence is a smooth velocity field, i.e., the velocity field is assumed to be at least in (C^1) . We introduced the idea of using the infinitesimal operator of the Frobenius-Perron operator to develop a model that describes a motion of image sequence. The problems are interesting inverse problems in their own right. We have chosen the modern theory based on total variation in image processing methods. With this model in hand, we employ a variational technique to approximate the velocity field. However, a problem of a large motion between two consecutive frames may arise when the sampling rate of an image sequence is small. We utilize the well-known coarse-to fine strategy to resolve this problem and demonstrate its applicability using various benchmark and real-world examples.

7.1 Contributions

The first part of this thesis has dealt with an application of clustering techniques used to extract a meaningful structure organization in a graph network. The clustering problems have been studied for many years and applied to many areas of engineering and sciences, but not widely used in the dynamical system paradigm to extract the almost invariant sets, such as basins of attraction. Unfortunately, definitions of clustering have never been unified, which has resulted in a large number of definitions and heuristic algorithms to extract a meaningful clustering of a graph network. Rather than developing a personal technique to extract the almost invariant sets, we have explored a relation between almost invariant sets in the dynamical system and community structures in graph theory, which is defined based on the optimization of the modularity measure. We have used the notion of the transition matrix (or Ulam-Galerkin matrix) to approximate a dynamical system of a time-series data and treated it as a Markov process. Then we introduce an application of the graph modularity method to partition a phase space, which is visited by a given time-series data, into almost invariant sets. We benchmark and verify the validity of our methods using a variety of examples. However, our partition method based on the graph modularity measure is applicable only to autonomous vector fields both with and without stochastic perturbations. This has led us to explore a generalization of this technique for general non-autonomous cases.

In the course of developing a technique to generalize the idea of almost invariant sets to a non-autonomous system, we have found that Haller and coworkers [14, 15, 16, 68] have already developed a technique of using the Finite-time Lyapunov exponent to approximate the Lagrangian coherent structure. This method was developed as an empirical generalization of the invariant manifold of an autonomous system and it can be used to understand the invariant sets of a non-autonomous system. Therefore, we have adopted this method to analyze the real-world experiment of a mixing device to understand the mixing process and efficiency.

In addition, we have introduced the Frobenius-Perron operator viewpoint to approx-

imate the dynamical system underlying image sequences. The image intensity has been regarded as a density function and so an image sequence has been viewed as a temporal evolution of a density function by the Frobenius-Perron operator. We have pointed out that the idea of the brightness constancy assumption widely used in the optical flow problem is a special case of our model based on the infinitesimal operator of the Frobenius-Perron operator. We have demonstrated the validity of this concept by both benchmark and real-world examples. Also, we have compared our result with a traditional PIV method based on the cross-correlation technique and experimentally demonstrated that our method provided superior results. Hopefully, the new idea of motion estimation based on the Frobenius-Perron operator framework may provide some needed insight to advance a better understanding and algorithms suitable for users in the dynamical system community.

7.2 Open questions and further research

The problem of dynamical systems estimation for image sequences in a domain of fluid flows is still an active research area although a number of approaches have been proposed in the past decade. To advance understanding of the flow obtained from image sequences, there is a need for a dedicated approach to reveal complex structures of the flow such as vorticity, divergence, and fixed points. The smoothness regularizer used in our method can fail to accurately estimate small vorticity/divergence regions. The augmented formulation of the div-curl regularizer proposed in [21] is used to cope with this problem. However, it requires prior estimates of vorticity and divergence by a parametric description of these two quantities in a neighborhood of fixed points. Recently, a low dimensional estimation of vorticity and divergence from image sequences has been proposed to extract vorticity and divergence maps from image sequences [98]. Instead of directly estimating the velocity field, the Helmholtz decomposition is used to define vorticity and divergence maps equipped with parameters that describe the locations, influenced neighbors, and strength of these two quantities. Then these parameters are minimized to comply with a constraint equation of brightness variation. Although this method may locate small vorticity/divergence region

better than the variational method, the quality of the overall velocity is not generally better; it may result in a poorer quality of the overall velocity field, indeed. Therefore, it is very attractive to develop a technique in the same vein as the aforementioned research to successfully apply it to various domains of image sequences with complex flows.

Furthermore, as we have seen from several examples in this thesis, the quality of the velocity vector obtained from the regularization method strongly depends on the smoothness parameter. The issue of how to determine a priori a suitable value of the smoothness parameter has not been much investigated theoretically. All existing variational methods using some regularization constraints require that we have to choose the parameter based on our observation of the estimated velocity field. Any attempt to obtain the “optimal” value of the smoothness parameter based on some criteria would be worthwhile.

Bibliography

- [1] Silivio L.T. de Souza, Ibereẽ L. Caldas, Ricardo L. Viana, Antõnio M. Batista, and Tomasz Kapitaniak. Noise-induced basin hopping in a gearbox model. *Chaos Solitons and Fractals*, 26:1523–1531, 2005.
- [2] E. Bollt, L. Billings, and I. Schwartz. A manifold independent approach to understanding transport in stochastic dynamical systems. *Physica D*, 173:153–177, 2002.
- [3] L. Billings, E. Bollt, and I. B. Schwartz. Phase-space transport of stochastic chaos in population dynamics of virus spread. *Physical Review Letters*, 88(234101), 2002.
- [4] S. Kraut and U. Feudel. Multistability, noise and attractor-hopping: the crucial role of chaotic saddles. *Physica D*, 181:222–234, 2003.
- [5] S. Kraut and U. Feudel. Enhancement of noise-induced escape through the existence of a chaotic saddle. *Phys. Rev. E*, 67:1–4, 2003.
- [6] S. L. T. de Souza and I. L. Caldas. Basin of attraction and transient chaos in a gear-rattling model. *J. Vibrat. Contr.*, 7:849–862, 2004.
- [7] M. Dellnitz and O. Junge. On the approximation of complicated dynamical behavior. *SIAM Journal on Numerical Analysis*, 36(2):491–515, 1999.
- [8] R. Preis and M. Dellnitz. *Congestion and Almost Invariant Sets in Dynamical Systems*. Proceedings of Symbolic and Numerical Scientific Computation (SNSC’01). Springer, 2003.
- [9] R. Preis, M. Dellnitz, M. Hessel, Ch. Schtte, and E. Meerbach. Dominant paths between almost invariant sets of dynamical systems. *Preprint 154 of the DFG Schwerpunktprogramm 1095*, 2004.
- [10] K. Padberg, R. Preis, and M. Dellnitz. Integrating multilevel graph partitioning with hierarchical set oriented methods for the analysis of dynamical systems. *Preprint 152 of the DFG Schwerpunktprogramm 1095*, 2004.
- [11] R.S. Mackey, J.D. Meiss, and I.C. Percival. Transport in Hamiltonian systems. *Physica D*.
- [12] E. Simiu. *Chaotic transistion in deterministic and stochastic dynamcail systems*. Princeton University Press, 2002.
- [13] S. Wiggins. *Introduction to Applied nonlinear dynamical systems and chaos*. Springer, 2 edition, 2003.

- [14] G. Haller and A. C. Poje. Finite-time transport in aperiodic flows. *Physica D*, 119:352–380, 1998.
- [15] G. Haller. Finding finite-time invariant manifolds in two-dimensional velocity fields. *CHAOS*, 10(1):99–108, 2000.
- [16] G. Haller. Lagrangian coherent structures from approximate velocity data. *SPhys. Fluids A*, 14:1851–1861, 2002.
- [17] S.C. Shadden, F. Lekien, and J.E. Marsden. Definition and properties of lagrangian coherent structures from finite-time lyapunov exponents in two-dimensional aperiodic flows. *Physica D*, 212:271–304, 2005.
- [18] S.C. Shadden, J.O. Dabiriand, and J.E. Marsden. Lagrangian analysis of fluid transport in empirical vortex ring flows. *Phys. of Fluids*, 18(4):271–304, 2006.
- [19] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17:185–203, 1981.
- [20] T. Corpetti, E. Měmin, and P. Pérez. Dense estimation of fluid flows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):365–380, 2002.
- [21] T. Corpetti, D. Heitz, G. Arroyo, E. Memin, and A. Sant-Cruz. Fluid experimental flow estimation based on an optical-flow scheme. *Experiments in Fluids*, 40:80–97, 2006.
- [22] D. Bereziat and J.-P. Berroir. Motion estimation on meteorological infrared data using a total brightness invariance hypothesis. *Env. Mod. Soft.*, 15.
- [23] J. J. Simpson and J. I. Gobat. Robust velocity estimates, stream functions, and simulated lagrangian drifters from sequential spacecraft data. *IEEE Transactions on Geoscience and Remote Sensing*, 32(3).
- [24] P. Heas, E. Memin, and N. Papadakis. Dense estimation of layer motions in the atmosphere. In *Int. Conf. Pattern Recognition (ICPR’ 06)*, Hong-Kong, China, 2006.
- [25] S. Gutierrez and D.G. Long. Optical flow and scale-space theory applied to sea-ice motion estimation in antarctica. *Geoscience and Remote Sensing Symposium, 2003. IGARSS ’03. Proceedings.*, 4:2805– 2807, 2003.
- [26] A. Suvichakorn and A. Tatnall. The application of cloud texture and motion derived from geostationary satellite images in rain estimation - a study on mid-latitude depressions. In *Geoscience and Remote Sensing Symposium, 2005. IGARSS apos;05. Proceedings*, volume 3.
- [27] R. P. Wildes, M. J. Ambabile, A. M. Lanzillotto, and T.S. Leu. Recovering estimate of fluid flow from image sequence data. *Computer vision and Image understanding*, 80:246–266, 2000.
- [28] N. Santitissadeekorn and E. M. Bollt. Identifying stochastic basin hopping and mechanism by partitioning with graph modularity. *Physica D*, (231):95–107, 2007.

- [29] N. Santitissadeekorn, D. Bohl, and E. Bollt. Analysis and modeling of an experimental device by finite-time lyapunov exponent method. *International Journal of Bifurcations and Chaos*, 2008.
- [30] N. Santitissadeekorn and E. Bollt. The infinitesimal operator for the semigroup of the frobenius-perron operator from image sequence data: Vector fields and computational measurable dynamics from movies. *Chaos*, 17(023126), 2007.
- [31] J. Guckenheimer and P. Holmes. *Nonlinear Oscillators, Dynamical Systems, and Bifurcations of Vector Fields*, volume 42 of *Applied Mathematical Sciences*. Springer, 1983.
- [32] C. Robinson. *Dynamical Systems: Stability, Symbolic Dynamics, and Chaos*. CRC, 2 edition, 1999.
- [33] M. Pollicott and M. Yuri. *Dynamical Systems and Ergodic Theory*, volume 40 of *London Mathematical Society student texts*. Cambridge University Press, 1998.
- [34] S. Wiggins. *Chaotic Transport in Dynamical Systems*. Springer, 1992.
- [35] A.M. Mancho, D. Small, S. Wiggins, and K. Ide. Computation of stable and unstable manifolds of hyperbolic trajectories in two-dimensional, aperiodically time-dependent vectors fields. *Physica D*, 182:188–222, 2003.
- [36] A.M. Mancho, D. Small, and S. Wiggins. Computation of hyperbolic trajectories and their stable and unstable manifolds for oceanographic flows represented as data sets. *Nonlinear Proc. Geoph.*, 11:17–33, 2004.
- [37] A.M. Mancho, D. Small, and S. Wiggins. A tutorial on dynamical systems concepts applied to lagrangian transport in oceanic flows defined as finite time data sets: theoretical and computational issues. *Physics Reports*, 437(3-4):55–124, 2006.
- [38] K. Ide, D. Small, and S. Wiggins. Distinguished hyperbolic trajectories in time-dependent fluid flows: analytical and computational approach for velocity fields defined as data sets. *Nonlinear Proc. Geoph.*, (9):237263, 2002.
- [39] Z.P. You, E.J. Kostelich, and J.A. Yorke. Calculating stable and unstable manifolds. *Int. J. Bifurcation and Chaos*, (1):605–624, 1991.
- [40] A. Lasota and M. C.Mackey. *Chaos, fractals, and noise, Stochastic Aspects of Dynamics*. Springer, New York, 2 edition, 1994.
- [41] T.Y. Li. Finite approximation for the Frobenius-Perron operator. a solution to Ulam’s conjecture. *J. Approx. Theory*, 17(2):177–186, 1976.
- [42] G. Froyland. *Estimating Physical Invariant Measures and Space Averages of Dynamical Systems Indicators*. PhD thesis, University of Western Australia, 1996.
- [43] G. Froyland. Finite approximation of Sinai-Bowen-Ruelle measure for Anosov systems in two dimension. *Random Comput. Dynam.*, 3(4):251–263, 1995.
- [44] C. Chui, Q. Du, and T. Li. Error estimates of the Markov finite approximation of the Frobenius Perron operator. *Nonlinear Analysis*, 19(4):291–308, 1992.

- [45] J. Ding and A. Zhou. The projection method for computing multidimensional absolutely continuous invariant measures. *J. of Statistical Physics*, 77(3/4):899–908, 1994.
- [46] F. Hunt. Approximating the invariant measures of randomly perturbed dissipative maps. *J. Math. Anal. Appl.*, 198(2):534–551, 1996.
- [47] A. Boyarsky and Y. S. Lou. Approximating measure invariant under higher-dimensional chaotic transformation. *J. Approx. Theory*, 65(2):231–244, 1991.
- [48] S. M. Ulam. *Problems in Modern Mathematics*. Science Editions. Wiley, New York, 1970.
- [49] G. Froyland. Computer-assisted bounds for the rate of decay of correlations. *Comm. Math. Phys.*, 189(1):235–237, 1997.
- [50] R. Sturman, J. M. Ottino, and S. Wiggins. *The mathematical foundations of mixing, The linked twist map as a paradigm in applications: micro to macro, fluids to solids*. Cambridge, 2006.
- [51] G. Froyland. Extracting dynamical behaviour via markov models. In Alistair Mees, editor, *Nonlinear Dynamics and Statistics*, pages 283–324. Birkhauser, 2001.
- [52] W. M. Miller. Stability and approximation of invariant measures for a class of nonexpanding transformations. *Nonlinear Analysis*, 23(8):1013–1015, 1994.
- [53] G. Froyland. Statistically optimal almost-invariant sets. *Physica D*, 200:205–219, 2005.
- [54] M. E. J. Newman and M. Girvan. *Statistical Mechanics of Complex Networks*. Springer, Berlin, 2004.
- [55] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(026113), 2004.
- [56] Bélla Bollobás. *Modern Graph Theory*. Springer, NewYork, 1998.
- [57] N. G. Van Kampen. *Stochastic Processes in Physics and Chemistry*. North-Holland, 4 edition, 2003.
- [58] J.P. Bagrow and E. M. Bollt. A local method for detecting communities. *Phys. Rev. E*, 72(046108), 2005.
- [59] L. Dannon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Preprint cond-mat/0505245*, 2005.
- [60] M.E.J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69(066133), 2004.
- [61] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70(066111), 2004.
- [62] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74(036104), 2006.
- [63] S. M. van Dongen. *Graph clustering by flow simulation*. PhD thesis, Universiteit Utrecht.

- [64] D. Gfeller. *Simplifying complex networks: from a clustering to a coarse graining strategy*. PhD thesis, Ecole polytechnique fédérale de Lausanne.
- [65] F. Pfeiffer and A. Kunert. Rattling models from deterministic to stochastic precesses. *Nonlinear Dyn.*, 1:63–74, 1991.
- [66] L. Arnold. *Random Dynamical Systems*. Springer, New York, 2 edition, 2003.
- [67] E. Ott. *Chaos in Dynamical Systems*. Cambridge University Press, 1 edition, 1993.
- [68] G. Haller and G. Yuan. Lagrangian coherent structures and mixing in two-dimensional turbulence. *Physica D*, 147:352–370, 2000.
- [69] D. Bohl. Experimental investigation of the fluid motion in a cylinder driven by a flat plate impeller. *Journal of Fluids Engineering*, 129(1):137–146, 2007.
- [70] R.J. Adrian. Twenty years of particle image velocimetry. *Experiments in Fluids*, 39(2):159–169, 2005.
- [71] C.P. Gendrich and M.M. Koochesfahani. A spatial correlation technique for estimating velocity fields using molecular tagging velocimetry (mtv). *Experiments in Fluids*, 22(1):67–77, 1996.
- [72] K. Petersen. *Ergodic Theory*. Cambridge University Press, 1983.
- [73] J. M. Fitzpatrick. The existence of geometrical density-image transformations corresponding to object motion. *Computer vision and image understanding*, 44:155–174, 1988.
- [74] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981.
- [75] J. Bign, G.H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991.
- [76] J. L. Barron, D. J. Fleet, and D. J. Beauchemin. Performance of optical flow techniques. *Int. J. Comput. Vis.*, 12(1):43–77, 1994.
- [77] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: An evaluation of eight optical flow algorithms. In *In Proc. the Ninth British Machine Vision Conference (BMVC '98)*, volume 1, pages 195–204, 1998.
- [78] A.N.Tikhonov and V.Y.Arsenin. *Solutions of ill-posed problem*. Winston, Washington, DC, 1977.
- [79] D. Suter. Motion estimation and vector splines. pages 939–942, 1994.
- [80] B. Dacorogna. *Direct methods in the calculus of variations*. Springer, Berlin, 1989.
- [81] B. Dacorogna. *Introduction to the calculus of variation*. Imperial College Press, 2004.
- [82] I. M. Gelfand and S. V. Fomin. *Calculus of Variation*. Prentice-Hall, Englewood, 1963.

- [83] H. Attouch, G. Buttazzo, and G. Michaille. *Variational Analysis in Sobolev and BV Space*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2006.
- [84] G. Aubert and P. Kornprobst. *Mathematical problems in image processing: partial differential equations and the calculus of variations*. Springer, New York, 2000.
- [85] G. Aubert, R. Deriche, and P. Kornprobst. Computing optical flow via variational techniques. *SIAM J. Appl. Math.*, 60(1):156–182, 1999.
- [86] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of bounded variation and free discontinuity problems*. Claredon Press, Oxford, 2000.
- [87] I. Cohen. Nonlinear variational method for optical flow computation. In *In Proc. the 8th Scandinavian Conference on Image Analysis*, 1993.
- [88] P. Nesi. Variational approach to optical flow estimation managing discontinuities. *Image and Vision Computing*, 11:419–439, 1993.
- [89] M. J. Black. Recursive non-linear estimation of discontinuous flow fields. pages 138–145, Stockholm, Sweden, 1994.
- [90] R. Deriche, P. Kornprobst, and G. Aubert. Optical-flow estimation while preserving its discontinuities: A variational approach. In *ACCV*, pages 71–80, 1995.
- [91] J. Weickert. On discontinuity-preserving optic flow. pages 115–122, 1998.
- [92] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 1996.
- [93] B. Fronberg. *A practical guide to pseudospectral methods*. Cambridge University Press, 1996.
- [94] L. N. Trefethen. *Spectral methods in MATLAB*. SIAM, Philadelphia, PA, 2000.
- [95] A. Jameson. Analysis and design of numerical schemes for gas dynamics 1: Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *International Journal of Computational Fluid Dynamics*, 4:171–218, 1995.
- [96] Joseph D. Skufca and Erik M. Bollt. Mostly conjugate: Relating dynamical systems — beyond homeomorphism. *submitted to SIAM Journal on Applied Dynamical Systems*, 2006.
- [97] M. W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems, and Linear Algebra*. Academic Press, Orlando, 1974.
- [98] A. Cuzol, P. Hellier, and E. Mémin. A low dimensional fluid motion estimator. *International Journal of Computer Vision*, 75(3):329 – 349, 2007.
- [99] E. P. Simoncelli. Design of multi-dimensional derivative filters. *IEEE Int. Conf. Image Processing*, 1:790–793, 1994.

Appendix A

Ridge detection

Recall that the Lagrangian coherent structure (LCS) at time t is defined as the second-derivative ridge of the Finite-time Lyapunov exponent (FTLE) field at time t as defined by Shadden [17]. Therefore, there is a need for us to develop an algorithm to detect the ridge. The main steps of our method of ridge detection are as follows. First, we estimate the local structure of the FTLE field based on the gradient structure tensor as shown below. Then, the ridge orientation at each point of the FTLE field is computed based on the smallest eigenvector of the structure tensor since its orientation corresponds to the direction of the smallest variation where as the largest eigenvector orients to the steepest descent/ascent direction. Thus, in order to trace the ridge, we may “walk” along the direction of the ridge orientation starting from an initial point, in which the global maximum point of the FTLE field is our preferred candidate. Remark that our method of the ridge detection is similar to those of line and edge detections and edge-preserving image restoration, which are fundamental tasks in image processing.

Given a FTLE field at time t , $\sigma_t(x, y)$, we define the 2×2 structure tensor S at each point (x, y) by

$$S \equiv \nabla \sigma_t \nabla \sigma_t^T. \quad (\text{A.1})$$

Then, we can explicitly compute the eigenvalues ($\lambda_1 \geq \lambda_2$) of the matrix S by

$$\lambda_{1,2} = (s_{11} + s_{22} \pm \sqrt{(s_{11} - s_{22})^2 + 4s_{12}^2})/2 \quad (\text{A.2})$$

where $s_{11} = \partial_x \sigma_t^2$, $s_{22} = \partial_y \sigma_t^2$, and $s_{12} = s_{21} = \partial_x \sigma_t \partial_y \sigma_t$. Here, $\partial_x \sigma_t$ and $\partial_y \sigma_t$ are the Gaussian derivative in the x and y direction, respectively. The eigenvector associated with the smallest eigenvalue λ_2 is give by

$$\nu_2 = [2s_{12}, s_{22} - s_{11} \pm \sqrt{(s_{11} - s_{22})^2 + 4s_{12}^2}]^T. \quad (\text{A.3})$$

Note that eigenvector ν_1 corresponding to λ_1 is perpendicular to ν_2 . The eigenvectors ν_1 and ν_2 corresponds to the directions of maximum and minimum variations, respectively. The ridge orientation is then defined by

$$\theta = \arctan(2s_{12}/(s_{22} - s_{11} \pm \sqrt{(s_{11} - s_{22})^2 + 4s_{12}^2})). \quad (\text{A.4})$$

Remark that it is also possible to compute the ridge orientation based on the Cauchy-Green tensor. That is one can correspondingly replace each entry of S by that of the Cauchy-Green tensor matrix except that the ridge orientation is now the direction of the dominant

eigenvector instead of the smaller one. This will also give a similar result as observed in our experiment since points on the LCS have a large stretching rate due to the dominant eigenvalue. Hence, walking along the direction of the dominant eigenvector of the Cauchy-Green tensor also trace the ridge line. The algorithm for the ridge detection is summarized below.

Algorithm: Ridge detection

```

set the “walking” distance  $r$  for each iteration

set the initial point  $(x_0, y_0)$ 

 $S := \nabla \sigma_t \nabla \sigma_t^T$     % compute structure tensor

 $\theta = \arctan(2s_{12}/(s_{22} - s_{11} \pm \sqrt{(s_{11} - s_{22})^2 + 4s_{12}^2}))$ 

begin iterations

 $p_n = x_n + r \cos(\theta(x_n, y_n))$ 

 $q_n = y_n + r \sin(\theta(x_n, y_n))$ 

 $(x_{n+1}, y_{n+1}) = \operatorname{argmax}_\tau I(p_n + \tau \cos(\theta + \frac{\pi}{2}), q_n + \tau \cos(\theta + \frac{\pi}{2}))$     % line search

n:=n+1

end iterations

```

One of the key components in the above algorithm is the line search. This subproblem ensures that the points (x_n, y_n) tracing the ridge line is locally the highest point transverse to the ridge in the FTLE field by searching for the local maximum point in the perpendicular direction to the ridge orientation. This line search subproblem is a minimization problem in a one-dimensional space and we use the minimization routine in MATLAB for this subproblem. We use various stopping criterions as follows. The iteration is stopped when the ridge line is elongated across the boundary or when the ridge length meets a given upper limit. The iteration is also stopped if the $|\lambda_1 - \lambda_2| < \delta$ at a point (x_n, y_n) in the iteration for some $\delta > 0$. The later condition characterizes a homogeneous region, where the difference between λ_1 and λ_2 becomes small. The illustration of the iterative algorithm is shown in Figure A.1 for the “volcano” image obtained from the computation of the FTLE field for the center-blade mixer experiment. Here, the yellow dot is the global maximum point and also the initial point of the iteration. Points (x_n, y_n) and (p_n, q_n) in the algorithm are shown in magenta and blue, respectively. The line in black color aligns along the direction for the line search procedure and it is oriented in the direction of the

eigenvector ν_1 , in which the point (x_{n+1}, y_{n+1}) in the algorithm is the local maximum in the search direction. The final result is shown in Figure A.2.

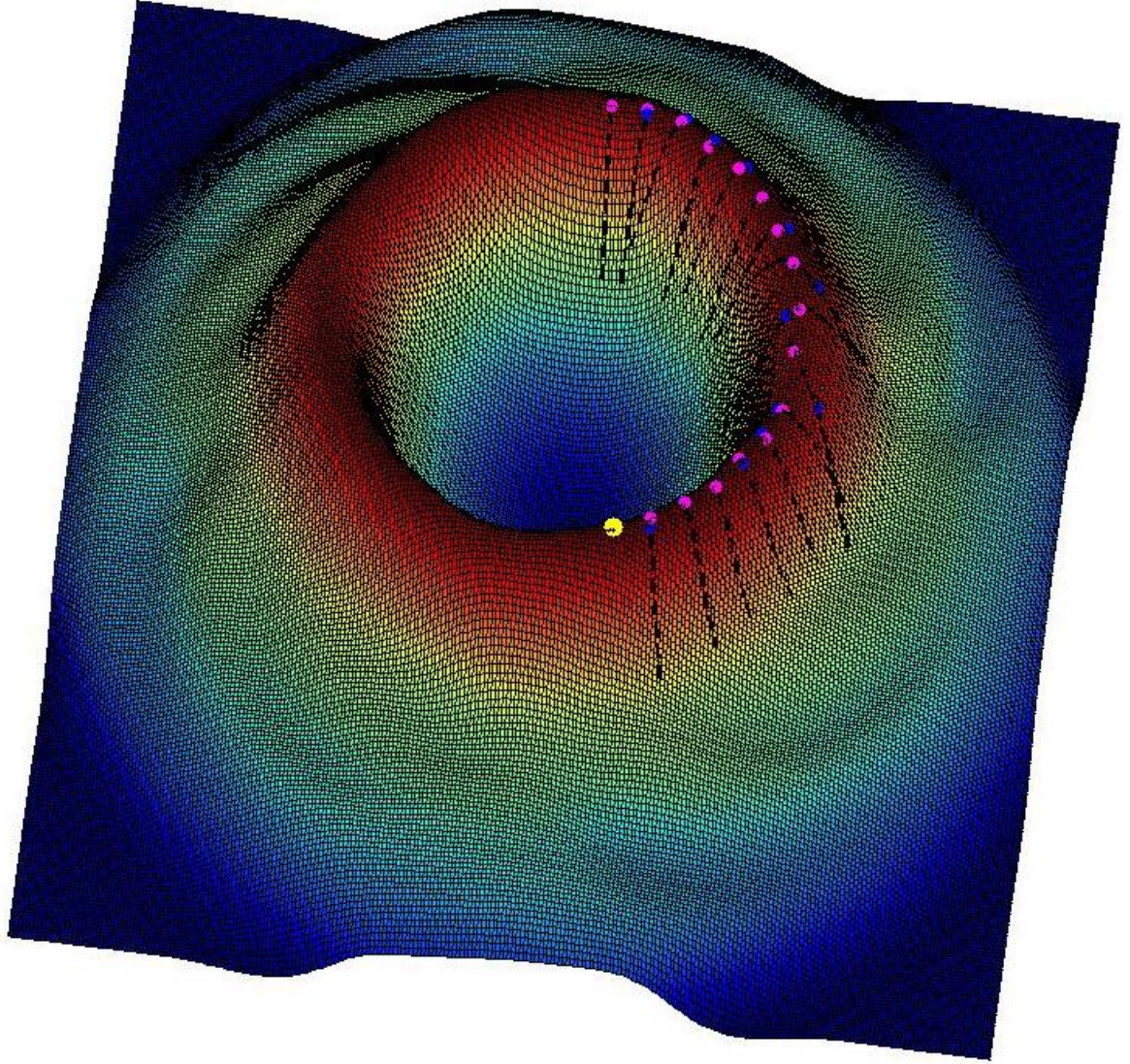


Figure A.1. The image data of a smoothed FTLE field. The yellow dot is the initial point of the iteration and the magenta dots are the points (x_n, y_n) in the iterative algorithm. Each blue dot is the point (p_n, q_n) in the iteration, which has a distance r from its corresponding point (x_n, y_n) pointed in the direction of eigenvector ν_2 . The line in black color is in the direction of the line search subproblem and it is oriented according to the eigenvector ν_1 perpendicular to ν_2

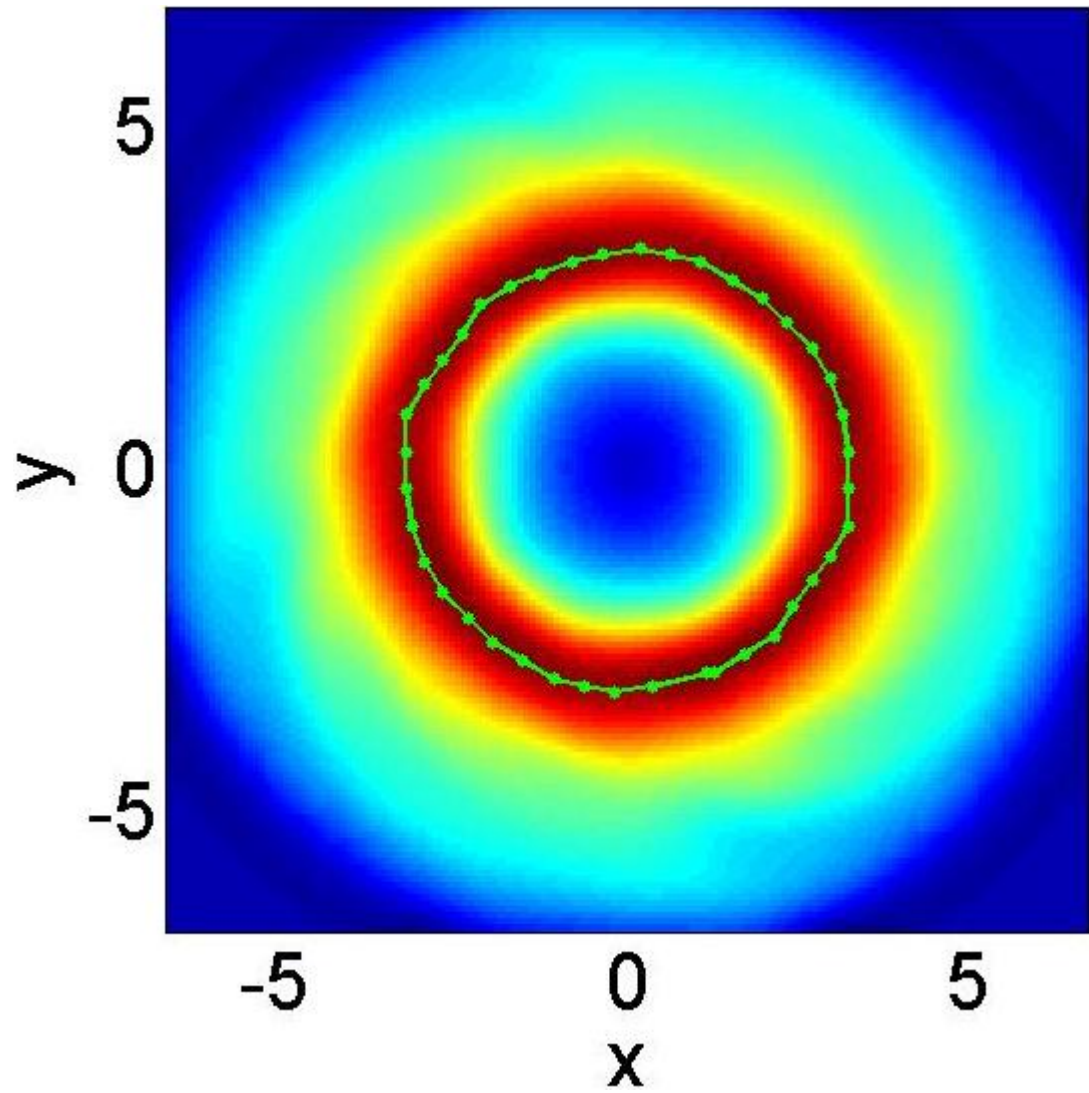


Figure A.2. The final result of the ridge detection for the FTLE field shown in Figure A.1. The green line plotted on the contour plot of the FTLE field is the ridge detected by the above algorithm.

Appendix B

Differentiation of image intensity

One important step of many image processing algorithms is the computation of image intensity derivatives. In the optical flow problem described in Chapter 6, we need to compute I_{x_1} , I_{x_2} , I_t for a give image intensity function $I(x_1, x_2, t)$ and Δu and Δv for the velocity fields $u(x_1, x_2)$ and $v(x_1, x_2)$. We use the Simoncelli's matched filter method [99] that was demonstrated to have superior accuracy to the traditional central and backward differences and preserve the rotationally invariant property of the 2-D derivative and Laplacian operators. Simoncelli formulated a set of matched pair of derivative filters and lower pass pre-filters as a filter design problem. Below are pairs of matched pre-filters and derivative kernels of various sizes. To obtain I_{x_1} we convolve the n-tap smoothing kernel in the x_2 dimension and then convolves that result with the differentiation kernel to obtain the final result, I_x . This procedure can be summarized in the following equation:

$$I_{x_1}(x_1, x_2, t) \approx d1 * p^T * (I(x_1, x_2, t)). \quad (\text{B.1})$$

Note that the pre-smoothing filter makes the derivative filter become less sensitive to noise. The implementation of the above procedure in MATLAB can be simple. For example, if the 3-tap filter pairs is used, the derivative kernel in the x_1 dimension can be computed by the following code:

```
pp=[0.2298 0.5402 0.2298];
pp=[zeros(1,3); pp; zeros(1,3)];
dd=[0.4252 0.0000 -0.4252];
dd=[zeros(1,3); dd; zeros(1,3)];
dKernel=imfilter(pp',dd,'replicate');
```

Note that the option “replicate” is used to impose the Neumann boundary condition. If the zero boundary condition is preferred, the option “zero” can be used, instead. The computation of I_{x_2} can be done using the transpose of the x_1 -derivative kernel. To obtain I_t we use the forward difference; that is $I_t = I(x, y, t + 1) - I(x, y, t)$, since we assume that the velocity field between two consecutive images is autonomous.

Let us now discuss an algorithm to compute the solution of the PDE (6.18). A numerical solution of the Euler-Lagrange equation (6.18) via a finite different method, resorting the

Table B.1. Matched pairs of pre-filter(p) and derivative kernels(d_i) of the i^{th} order of various sizes.

3-tap	p: 0.2298 0.5402 0.2298
	d1: 0.4252 0.0000 -0.4252
	d2: 0.3557 -0.7114 0.3557
5-tap	p: 0.0376 0.2491 0.4263 0.2491 0.0376
	d1: 0.1096 0.2766 0.0000 -0.2766 -0.1096
	d2: 0.2190 -0.0007 -0.4366 -0.0007 0.2190
7-tap	p: 0.0047 0.0693 0.2454 0.3611 0.2454 0.0693 0.0047
	d1: 0.0187 0.1253 0.1930 0.0000 -0.1930 -0.1253 -0.0187
	d2: 0.0543 0.1370 -0.0534 -0.2758 -0.0534 0.1370 0.0543

Simoncelli's derivative kernels, takes the following form:

$$\begin{aligned}
L * u &= \alpha D_{x_1} * I(D_{x_1} * Iu + D_{x_2} * Iv + I(x_1, x_2, t + 1) - I(x_1, x_2, t)) \\
L * u &= \alpha D_{x_1} * I(D_{x_1} * Iu + D_{x_2} * Iv + I(x_1, x_2, t + 1) - I(x_1, x_2, t)),
\end{aligned} \tag{B.2}$$

where “ $*$ ” denotes the 2-D convolution, L is the Laplacian kernel matrix and D_{x_1} and $D_{x_2}(= D_{x_1}^T)$ are the derivative kernel matrices in the direction of x_1 and x_2 , respectively. For convenience, we will denote the finite-difference approximations of derivative, $D_{x_1} * I$ and $D_{x_2} * I$, by I_{x_1} and I_{x_2} , respectively. Then, an iterative method, such as Gauss-Seidel or SOR, can be used to estimate a new set of solutions (u^{n+1}, v^{n+1}) at the n^{th} step by

$$\begin{aligned}
u^{n+1} &= \bar{u}^n (I_x \bar{u}^n + I_y \bar{v}^n + I_t) / (\alpha + I_x^2 + I_y^2) \\
v^{n+1} &= \bar{v}^n (I_x \bar{u}^n + I_y \bar{v}^n + I_t) / (\alpha + I_x^2 + I_y^2),
\end{aligned} \tag{B.3}$$

where $\bar{u} \equiv \tilde{L} * u$ and \tilde{L} is the Laplacian kernel matrix L with the nullified center.

Appendix C

Mass conservation in transmittance imaging

Although the mass conservation is common in 3D fluid flow, it is questionable whether or not this condition also holds for a corresponding 2D motion captured by a corresponding image sequence. Fitzpatrick [73] demonstrated that for transmittance imagery such a case is true, subject to certain constraint on normal flow at the image boundary. In particular, it required that the flow at opposite boundary points along the path of integration be equal or that the flow has a compact support within the boundary of integration. The latter requirement also complies with the assumption of the compact support of the initial density evolved by the Koopman operator, see Eq. (3.20).

Consider a 3D fluid flow of the fluid density $\rho(\mathbf{R}, t)$ subject to a velocity field $\mathbf{V}(\mathbf{R}, t) = (U(\mathbf{R}, t), V(\mathbf{R}, t), W(\mathbf{R}, t))$, where \mathbf{R} is an abbreviated notation for the coordinate vector (x_1, x_2, x_3) . For a transmittance imagery, the image is given as

$$I(x_1, x_2, t) = \int_{s_1(x_1, x_2)}^{s_2(x_1, x_2)} \rho(\mathbf{R}, t) dx_3, \quad (\text{C.1})$$

where $s_1(x, y)$ and $s_2(x, y)$ are the bounding surface of the specimen being imaged. Under an assumption of the mass conservation the fluid density $\rho(\mathbf{R}, t)$ and the velocity field $\mathbf{V}(\mathbf{R}, t)$ are related according to

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho \mathbf{V}) = 0. \quad (\text{C.2})$$

Integrating the continuity equation (C.2) along the x_3 coordinate yields

$$\int_{s_1(x_1, x_2)}^{s_2(x_1, x_2)} \text{div}(\rho \mathbf{V}) dx_3 + \int_{s_1(x_1, x_2)}^{s_2(x_1, x_2)} \frac{\partial \rho}{\partial t} dx_3 = 0. \quad (\text{C.3})$$

Then we can rewrite the first term of the above equation as

$$\int_{s_1(x_1, x_2)}^{s_2(x_1, x_2)} \text{div}(\rho \mathbf{V}_{x_1, x_2}) dx_3 + \int_{s_1(x_1, x_2)}^{s_2(x_1, x_2)} \frac{\partial \rho}{\partial W} dx_3, \quad (\text{C.4})$$

where $\mathbf{V}_{x_1, x_2} = (U, V, 0)$. Applying the Leibniz integral rule to the first term of the above equation yields

$$\int_{s_1(x_1, x_2)}^{s_2(x_1, x_2)} \text{div}(\rho \mathbf{V}) dx_3 = \text{div} \int_{s_1(x_1, x_2)}^{s_2(x_1, x_2)} \rho \mathbf{V}_{x_1, x_2} dx_3 + \rho \mathbf{n} \cdot \mathbf{V} \Big|_{s_1}^{s_2}, \quad (\text{C.5})$$

where $n = \left(-\frac{\partial x_3}{\partial x_1}, -\frac{\partial x_3}{\partial x_2}, 1\right)$ is the outward normal vector to the surface s_1 and s_2 . A new $2D$ velocity field $\mathbf{v} = (u(x_1, x_2), v(x_1, x_2))$ can be defined as the density-weight average of the original $3D$ velocity field; that is

$$\mathbf{v} \equiv \frac{\int_{s_1(x_1, x_2)}^{s_2(x_1, x_2)} \rho \mathbf{V}_{x_1, x_2} dx_3}{\int_{s_1(x_1, x_2)}^{s_2(x_1, x_2)} \rho dx_3}. \quad (\text{C.6})$$

Combing Eqs. (C.1), (C.2), and (C.6) gives

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho \mathbf{v}) = \rho \mathbf{n} \cdot \mathbf{V} \Big|_{s_1}^{s_2}. \quad (\text{C.7})$$

It is clear from Eq. (C.7) that we obtain the two-dimensional continuity equation when the normal flow vanish at the boundaries or the normal flows at opposite boundary points along the path of integration are equal.