# Model Free Techniques for Reduction of High-Dimensional Dynamics

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Tyrus Hunter Berry
Master of Science
Ohio State University, 2008
Bachelor of Science; Bachelor of Arts
University of Virginia, 2006

Director: Dr. Timothy Sauer, Professor
Department of Mathematical Sciences

Spring Semester 2013
George Mason University
Fairfax, VA

# Acknowledgments

I would like to thank my advisor, Dr. Timothy Sauer, for the personalized coursework, patient attention, unyielding dedication, and many key insights that have led to this dissertation. I thank all of my professors at George Mason University, including all the committee members, for putting so much effort into teaching excellent courses, many of which have influenced the direction and results of this dissertation. Specifically, I thank Dr. Sauer, Dr. Wanner, and Dr. Walnut for setting a high standard of mathematical precision to which I continue to aspire. I would also like to thank Dr. Walnut for encouraging and assisting my study of the Bible which has become an important source of inspiration for me. Most importantly I would like to thank my wife, Miruna Tecuci, for encouraging me when I most needed it and readily sharing the sacrifices and uncertainties of my academic pursuit.

# Table of Contents

# Abstract

MODEL FREE TECHNIQUES FOR REDUCTION OF HIGH-DIMENSIONAL DYNAMICS

Tyrus Hunter Berry, PhD

George Mason University, 2013

Dissertation Director: Dr. Timothy Sauer

There is a growing need in science and engineering to extract information about complex phenomena from large data sets. A rapidly developing approach to building a model from data is manifold learning, and analysis of such a model may allow isolation of the desired features of the data. By introducing an additional geometric structure, the techniques of differential geometry become available for analyzing the model. In this dissertation we extend previous methods of analyzing the geometry of data. Our key contribution is the theory of *local* kernels, which generalizes previous nonparametric techniques such as Laplacian eigenmaps and diffusion maps. We show that every geometry can be represented by a local kernel in the limit of large data. Moreover, using the discrete exterior calculus (DEC) we show that a local kernel can be used to introduce a discrete Hodge star operator on a data set. This shows that local kernels introduce a discrete geometry on a data set without the need for an explicit simplicial complex.

From the perspective of dimensionality reduction, the theory of local kernels represents a nonlinear generalization of principal component analysis (PCA), where a geometric prior replaces the linear generative model of PCA. From the perspective of signal processing and time-series analysis, the theory of local kernels represents a generalization of Fourier

analysis where the basis can be adapted to the geometry of the reconstructed state space, or even the Lyapunov geometry which is intrinsic to the dynamical system.

The wide variety of data types in science and engineering applications requires a framework for integration of prior information with the model-free analysis. We outline a framework based on geometry with several case studies demonstrating integration of different prior data structures. The geometric framework naturally allows integration of prior information about the structure of the data into the analysis. This is achieved by identifying features of the geometry which are extrinsic and making the geometric construction invariant to these features. We illustrate this approach by exploring the intrinsic geometry for various data structures commonly found in dynamical systems, including temporal, spatial, spatiotemporal data.

The most important theoretical contributions are the theory of local kernel functions developed in Chapter 2 and the geometry of time delay embeddings developed in Chapter 3. Chapter 4 outlines how the theory of Chapter 2 can be applied to data with an a priori spatial structure. Chapter 4 also connects the discrete geometry to emerging research on data-adapted wavelet bases via the symmetry group of the geometry. Finally, Chapter 5 begins the ambitious program of integrating nonparametric analysis with existing parametric models. The ensemble Kalman filter is developed as a tool for extracting the portion of the data explained by the model, which naturally leads to nonparametric model extension.

# Chapter 1: Introduction

The rapid expansion of model complexity and data availability in the applied sciences is quickly outpacing the classical approaches to mathematical modeling. Meeting this challenge requires developing methods which can provably find, reconstruct, and represent the hidden stochastic, geometric, and algebraic structure of data. Classical approaches gravitate towards two different extremes. One extreme is fully parametric techniques which require that an exact model is specified. On the other extreme are nonparametric methods, which require no explicit model. For dynamical systems, data assimilation techniques are fully parametric while state space reconstruction techniques are nonparametric. Often, the most successful analysis will find a middle ground between these two extremes, sometimes called semi-parametric modeling. Typically, integrating nonparametric techniques into an existing parametric analysis is difficult and requires an improved understanding of the nonparametric approach.

The difficulties with parametric modeling for complex problems are ubiquitous. The curse of dimensionality fundamentally limits the ability to estimate large numbers of parameters with statistical significance, and for complex problems it is difficult to find models with small numbers of parameters. Nonparametric techniques attempt to avoid over specifying the model, leaving enough freedom that the best implicit parameters can be adaptively selected and fit simultaneously from the data. However, this same freedom can allow nonparametric techniques to focus on the most prevalent features of the data, and these are usually features which can be well represented by a simple model. The goal of a nonparametric approach is to maximize the freedom in the implicit model space, without allowing the technique to rediscover the known features of the data.

To illustrate how nonparametric models may inadvertently rediscover structure in the data which is already known, consider a time series sampled from a dynamical system evolving on a manifold. We may wish to learn features of the state space (such as geometric or topological features of the manifold). However, there is another structure in the data. Since temporally successive sample points will be nearby in state space, there is a linear path through the state space given by the time ordering. If we are not careful, a nonparametric technique may identify the state space as a one-dimensional linear space, but in fact it has only rediscovered the time ordering. By failing to properly inform the nonparametric model of our a priori information, namely that the data was time ordered, we wasted time and data rediscovering this fact.

Since our focus is on applications to dynamical systems, the previous example is particularly important. However, dynamical data may also have an a priori spatial structure, or even an existing model. In each case, we need to develop a specialized technique which optimally combines the prior information with a nonparametric technique for analyzing the residual structure. While we will begin to develop these specialized techniques, this appears to be an inherently domain specific problem. The more urgent need is for a framework of nonparametric techniques that can work naturally with various types of priors.

In particular we will focus on priors which involve (or can be well approximated by) assumptions of continuity or smoothness. There is a rapidly developing literature focusing on nonparametric modeling where the central assumption is that the data lies near a topological or geometric manifold. In Chapter 2 we will develop these methods into a general theory of *local* kernel functions, which we show can represent any geometry on a compact manifold. We also show how to recover geometrically and topologically important information from an abstract data set. In later chapters we will apply these techniques and show how this information can be used to discover the hidden features of data. However, as we have stressed above, a successful nonparametric modeling technique should be able to ignore the known structure and focus on the unknown structure.

Coifman and Lafon first addressed this issue in [2] when they showed that it was possible to build a nonparametric model which was invariant to the sampling distribution. This was a significant technical achievement since previous techniques required the unrealistic assumption of uniform sampling of the data structure. However, more importantly, they introduced the notion of intrinsic vs. extrinsic features. Of course, in some cases we may want the sampling density to affect the geometry in which case this would be an intrinsic geometric feature. However, when the sampling distribution is merely an artifact of an unknown sampling processes, the approach of [2] could remove its influence. In general, one would like the ability to construct a geometry which is invariant to any particular set of extrinsic features, and while this may not be possible in general we will provide an additional tool in Chapter 2 for constructing such geometries. Moreover, this gives a natural framework for integrating the geometric approach to nonparametric modeling with various priors. Namely, the prior determines which features of the geometry are intrinsic and which are extrinsic.

In Chapters 3-5 we begin carrying out our program of integrating nonparametric modeling with temporal, spatial, and existing model priors. Since we are focused on dynamical systems, we begin with temporal structure in Chapter 3. Our key result in Chapter 3 is that classical time-delay reconstruction severely biases the geometry in favor of the most stable Lyapunov component of the dynamics. Moreover, for dynamical systems the sampling distribution can inform us about the invariant measure of the system. In this context, the sampling distribution is an intrinsic feature of the geometry, and a previous result of Coifman and Lafon allows us to construct a geometry which respects this measure. Based on our theory of local kernels, developed in Chapter 2, we outline an important new direction which may allow reconstruction of the full intrinsic geometry of a dynamical system.

As our first case study, Chapter 3 demonstrates the importance of finding the intrinsic geometry for a data set. In the context of time series, we provide a new interpretation of the algorithm of Coifman and Lafon as a low-pass filter adapted to the geometry of the data. We show that by leveraging the biased geometry of delays, the geometry adapted low-pass filter can be used to separate time scales. This interpretation expands on work started by Giannakis and Majda in [3, 4]. The major ideas of Chapter 3 are implemented in the diffusion mapped delay coordinates (DMDC) algorithm which is detailed in the appendix.

Numerical examples in Chapter 3 use the DMDC algorithm to separate time scales for high dimensional data sets with complicated dynamical systems.

In Chapter 4 we begin to address the more complicated issue of integrating prior spatial or spatiotemporal structure with a nonparametric model. Unlike temporal structure, which has a natural intrinsic geometry, it is unclear which geometric features are intrinsic to spatial data. What is clear is that a spatial structure introduces a notion of translation within data samples. We show how to use this prior notion of translation to impose spatial smoothness on the nonparametric model. This suggests that the geometry of a model with spatial structure should locally decompose into the product of the spatial geometry and the residual geometry (which represents the values of the data in the local spatial region). However, spatial data structures such as images, and spatiotemporal systems are both known to exhibit spatial discontinuity, which is poorly represented in the bases developed in the previous chapters. To address this issue we overview the developing theory of diffusion wavelets [5] and wavelets on graphs [1]. We suggest that the remaining weakness of these techniques is that the symmetry group of the discrete geometries are poorly understood and we suggest techniques to understand these symmetry groups using the theory developed in Chapter 2.

In Chapter 5 we address a timely issue in the inference of dynamical systems from data, where the prior information includes a sophisticated model, yet significant model error remains. Clearly, the model should not be discarded as irrelevant since a nonparametric technique might require impractical amounts of data in order to simply reproduce the model, much less improve on it. Thus, the nonparametric technique must be combined with a data assimilation technique, such as a generalized Kalman filter, which can extract the portion of the data which is explained by the model. The remaining portion of the data is represented as a residual time series called the innovations. We show that the innovation series is a natural candidate for nonparametric modeling. We first show that the innovations can be modeled as correlated white noise to automatically compensate for significant model error. This novel technique is important as it optimally tunes the existing parameters of the filter to minimize the residual, thus extracting the maximum amount of information which can be explained by the model. Finally, we suggest how a nonparametric technique such as DMDC could be used to automatically extend the existing model as a nonparametric model for the innovation sequence.

# Chapter 2: Nonparametric modeling and the implicit geometry of kernel functions

In this chapter we develop a geometric approach to nonparametric modeling, in which we attempt to explain observed data by assuming that the data lies near a smooth manifold. The key development of the chapter is the theory of *local* and *shift-local* kernels which generalize Gaussian kernels (also known as radial basis functions). In Section 2.4 we show that in the limit of large data local kernels are equivalent to smooth geometries. Furthermore, in Section 2.5 we demonstrate a novel connection between the theory of local kernels and the *discrete exterior calculus* [6] which describes the geometry of a finite data set. Together these results show that local kernels define a discrete geometry which converges to a smooth geometry in the limit of large data. Shift-local kernels are a further generalization of local kernels and we conjecture that they describe dynamically important elliptic operators and are connected to Finslerian geometries.

Crucially, the choice of a local kernel determines which features of a data set are represented in the geometry. The challenge is to design a kernel which preserves the desired features, which are called intrinsic, and is invariant to the remaining features, which are called extrinsic. This chapter introduces the intrinsic/extrinsic dichotomy and continues a program of Coifman and Lafon [2] to design kernels invariant to various features of data which has no a priori structure. The remaining chapters will begin to address this challenge for some ubiquitous data types such as time-ordered data or data with a spatial structure.

## 2.1 Overview

Nonparametric modeling, also known as model-free data analysis, employs priors which are not defined by a fixed set of parameters. Perhaps the simplest type of non-parametric model is a histogram for a real valued random variable. By fixing a bin width we have implicitly assumed that the density function we are estimating is sufficiently slowly varying that its values inside the bin are well approximated by a constant function. Of course, one could claim that this is a parametric model where the parameters are the means of the density function in each bin. The sense in which it is non-parametric is that we do not expect to use all of the infinite possible bins, and we do not know a priori which bins will be used. Thus we do not keep track of all of the infinitely many parameters, but adaptively select those most relevant to the data. Moreover, one can prove that if the density function is continuous then in the limit as the number of samples goes to infinity the histogram converges to the true density function. We refer to this as the *limit of large data*, and this is an important step in establishing the validity of a non-parametric modeling technique.

Suppose that we know that in addition to being continuous, the density function must be band limited. In this case, a histogram, which is discontinuous at each stage, would not seem to be making the best use of the data given our prior. Instead, we know from sampling theory how to interpolate a band limited function, and this technique makes much better use of our a priori information. Alternatively, we may know that the density function had a certain number of continuous derivatives or perhaps we know that it is continuously differentiable. We refer to this type of a priori information as a *smoothness prior*, and kernel density estimation is a well studied method to interpolate the data samples into a smooth function which converges to the correct density in the limit of large data.

So far we have been considering a one dimensional random variable; but now consider the case when each sample is multi-dimensional. In particular, consider the case where the dimensionality is very high. In these cases, even if parametric models are available they tend to be high-dimensional and the curse of dimensionality makes parameter estimation extremely difficult. Thus, it is natural to assume that the data lies near a low dimensional space which is embedded in the high dimensional observation space. This low dimensional space is often called the latent space. If the latent space is linear, then principal component analysis (PCA) can automatically identify the correct linear subspace and project the observed data onto the latent space. However, if the latent space is nonlinear then it could appear that all of the coordinates are relevant and PCA may not be sufficient.

A low dimensional latent space which is continuos and nonlinear is a topological manifold, and if the latent space is smooth then we can also associate a geometry to it. This gives a us a natural nonparametric prior for our data, which is that the observations in the high dimensional space lie near a low dimensional manifold. This chapter will explore the geometric prior, including how to represent and estimate geometric information and how to use this geometric information to reduce, decompose, and explore the observed data.

In Section 2.2, we explain the basic dimensionality reduction techniques which attempt to generalize PCA to nonlinear data sets. This sets the stage for the geometric interpretation of Kernel PCA by Belkin and Nyogi [7] and the development of the intrinsic/extrinsic dichotomy by Coifman and Lafon [2]. In Section 2.3 we summarize the important developments and techniques as found in [2, 5, 7–19] as well as presenting some new geometric interpretations of the constructions. In Section 2.4 we generalize diffusion maps to a large class of kernels called local kernels which are equivalent to Riemannian metrics in the limit of large data, we also introduce a new notion of intrinsic geometry which extends that of diffusion maps. In order to extend our understanding of the geometry to differential forms we demonstrate a connection between local kernels and a theory of discrete geometry called the *discrete exterior calculus* (DEC) [6,20] in Section 2.5. We show that using the using the DEC with local kernels we can recover analogues of many important geometric operators and even recover topological invariants such as Betti numbers and representatives of the cohomology classes. Finally, in Section 2.6 we further generalize local kernels to *shift-local* kernels and conjecture that these give discrete approximations of important elliptic operators. We also suggest connections to a generalization of Riemannian geometry known as Finslerian geometry.

## 2.2 Dimensionality reduction background

Dimensionality reduction techniques attempt to reduce the complexity of high-dimensional data sets, called the observed data, by imposing assumptions on the structure of the data. These assumptions should imply that a low dimensional set of variables, called the latent variables, can explain the important phenomena of the observed data. One common definition for "important phenomena" is the total variance of the data but what is important in the data is dependent on the application. The goal of dimensionality reduction is to recover the hidden latent variables using only the observed data.

In this section we give a brief overview of the basic dimensionality reduction techniques which will be relevant to understanding the rest of the chapter. All the techniques in this section are an attempt to generalize the incredibly successful linear technique known as principal component analysis (PCA) described in Section 2.2.1. Moreover, many of the technique developed in this chapter can be understood simplistically in terms of Kernel PCA which is developed in Section 2.2.2 and Metric MDS which is developed in Section 2.2.3. However, these techniques will take on dramatically new interpretations as we develop the geometry of discrete data sets.

### 2.2.1 Linear methods

Linear dimensionality reduction methods make the restrictive assumption that the observed data, $\{y(i)\}_{i=1,...,M} \subset \mathbb{R}^n$, comes from an unknown linear observation function

$$y(i) = Wx(i) \qquad W^T W = I$$

where $\{x(i)\}_{i=1,...,M} \subset \mathbb{R}^p$ are unknown latent variables drawn from p independent Gaussian distributions $x(i)_k = \mathcal{N}(0, \sigma_k)$ where we assume $\sigma_k$ are decreasing in $k = 1, ..., p$. Several closely related techniques for recovering the latent variables are Principal Component Analysis (PCA), Karhunen-Loeve Transform (KLT), and Multi-Dimensional Scaling (MDS) [21]. These techniques are optimal in the sense of root mean squared reconstruction error which measures the average distance between the recovered latent variables $\hat{x}_i$ and the actual latent variables $x_i$. In the rest of this section we will assume that the data sets $Y$ and $X$ are arranged in matrix form with the $i$-th data point, $y(i)$, arranged as the $i$-th column of the matrix $Y$.

The key to Principal Component Analysis (PCA) is the assumption that the latent variables, $x(i)$, are drawn from independent Gaussian distributions. Since the observed variables, $y(i)$, are simply weighted sums of the latent variables, the observed variables follow a multivariate Gaussian distribution. Moreover, the covariance matrix for the observed variables has the following representation

$$\mathrm{cov}(y(\cdot)) = \mathbb{E}[YY^T] = \mathbb{E}[WXX^TW^T] = W\mathrm{cov}(x(\cdot))W^T$$

and since the latent variables are independent $cov(x(\cdot))$ is a diagonal matrix and thus the above decomposition is actually the unique diagonalization of the covariance matrix. Thus, by diagonalizing the covariance matrix of the observed variables, we recover both the variances (given by the eigenvalues) and the transformation $W$ (given by the eigenvectors). Since we assume that $W^TW = I$ (meaning that $W$ is a change of variables matrix) we can recover the latent variables by computing $\hat{x}(i) := W^TWx(i) = W^Ty(i)$. Note that the reconstruction may not be exact, since the rank of $W$ could be less than the number of latent variables but this reconstruction is optimal in the sense of the mean squared error $\mathbb{E}[||\hat{x}(i) - x(i)||_2^2]$. Finally, note that if $W$ has rank $p$ and $n > p$ then there are redundancies in the observed variables and PCA will produce latent variables with lower dimensionality but that contain the same information as the redundant high-dimensional observations. This justifies calling PCA a linear dimensionality reduction technique.

Multi-Dimensional Scaling (MDS) works with the same model as PCA and actually produces a numerically identical reconstruction, however the method is fundamentally different and provides a natural extension to nonlinear techniques such as Isomap [21]. Instead of working with the covariance matrix of the observed data, MDS works with the Gram matrix, which is the matrix of inner products $S_{ij} = \langle y(i), y(j)\rangle$ so $S = Y^T Y$. It is immediate that

$$S = Y^T Y = X^T W^T W X = X^T X = U \Lambda U^T$$

where $S = U\Lambda U^T$ is the unique eigenvalue decomposition of the symmetric matrix $S$. Thus, setting $\tilde{X} = \Lambda^{1/2} U^T$ we recover the latent variables. To see that MDS and PCA give the same result, that is $\tilde{X} = \hat{X}$, let $Y = A\Sigma B$ be the singular value decomposition (SVD) of the matrix $Y$. Note that $S = Y^T Y = B\Sigma^T \Sigma B^T$ so $B = U$ and $\Sigma^T \Sigma = \Lambda$ and also $YY^T = A\Sigma\Sigma^T A^T$ so $V = A$. Now we can use the SVD again to see that $\hat{X} = V^T Y = V^T A \Sigma B = V^T V \Lambda^{1/2} U^T = \tilde{X}$.

Note that MDS finds the latent variables using only the Gram matrix $S$ whereas PCA requires the observed variables $Y$ to compute the latent variables. Moreover, MDS can work with the distance matrix $D_{ij} = ||y(i) - y(j)||_2$ by reconstructing $S$ from $D$ using a process called double centering (see Section 2.2.3 below). Thus, MDS can produce latent variables without any explicit observed vectors, all it requires is a matrix of distances. This proved useful for applications such as representing survey results. Instead of producing an ad hoc representation of surveys in Euclidean space, it was more natural to simply measure differences between survey responses and then apply MDS.

Finally, the Karhunen-Loeve Transform (KLT) is computationally identical to PCA and MDS, however, it reinterprets the data as coming from a Gaussian stochastic process. This is important to note as it is one of the few dimensionality reduction techniques targeted at dynamical systems.

The methods in this section are called linear because of they reduce the dimensionality by a linear projection and are only provably optimal for a linear generative model. However, even when the generative model is not known to linear, these techniques are pervasive, mainly because of a lack of stable, feasible, and demonstrably better alternatives. In Section 2.3 we will develop a certain type of extension of PCA to nonlinear generative models which will lead us to the definition of local kernels and the geometric approach to nonlinear dimensionality reduction.

### 2.2.2 Kernel based extensions of PCA

Kernel based extensions of principal component analysis form a special Gram matrix where the standard inner product is replaced by a new inner product that is defined by a kernel function. Assume that the data lies in a subset $U \subset \mathbb{R}^n$ and that there exists a nonlinear map $\phi : \mathbb{R}^n \to \mathbb{R}^N$ such that $\phi(U)$ is or lies near a linear subspace of $\mathbb{R}^N$. Then we can define a new inner product on the data using the function $\phi$ by setting

$$\langle x, y \rangle_\phi = \langle \phi(x), \phi(y) \rangle_{\mathbb{R}^N} .$$

Thus, instead of forming the Gram matrix $S$ we can instead form $(S_\phi)_{ij} = \phi(y_i)^T \phi(y_j)$ and simply apply MDS to this Gram matrix. Of course, one would like to know when such a $\phi$ exists and more importantly how to find it. As a matter of practical experience, it seems that when the new dimension $N$ is large enough, such a $\phi$ often exists, and the developments of this chapter may provide some insight as to why this is the case. However, this raises two practical problems, the first is finding an appropriate $\phi$ and the second is computing the large inner products in the new space. The kernel trick allows one to bypass both of these difficulties by using a kernel function.

A kernel function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ assigns a real value to pairs of data points. The kernel function is called symmetric if $K(x, y) = K(y, x)$ for all $x, y \in \mathbb{R}^n$, and is called positive semidefinite if for all square integrable functions $f$ we have

$$\int_{\mathbb{R}^n} \int_{\mathbb{R}^n} K(x, y) f(x) f(y) dx dy \geq 0.$$

Often we only need the kernel to be positive semidefinite on a subset $U \subset \mathbb{R}^n$, in which case we can restrict the integrals to $U$. A symmetric and positive semidefinite kernel is guaranteed to yield an inner product since the matrix $(S_K)_{ij} = K(x_i, x_j)$ is a symmetric and positive semidefinite matrix, which guarantees $S_K$ is the Gram matrix for some set of vectors.

A generalization of this fact is Mercer's theorem [21], which states that a symmetric and positive semidefinite kernel which is continuous and integrable can be written $K(x, y) = \sum_{i=1}^{\infty} \lambda_i \varphi_i(x)\varphi_i(y)$ where $\varphi_i$ is a basis of $L^2(\mathbb{R}^n)$ made up of eigenfunctions of $T_K\varphi_i(x) = \int_{\mathbb{R}^n} K(x, y)\varphi(y)dy = \lambda_i\varphi_i(x)$. When we apply MDS to the Gram matrix $S_K$, the eigenvectors we compute are a discrete approximation of the continious eigenfunctions $\varphi_i$. This interpretation will be expanded on significantly in Section 2.3.

The phrase *kernel trick* refers to defining the embedding function $\phi$ implicitly via an appropriate kernel function and makes Kernel PCA an efficient technique. However, we have replaced the difficult problem of finding an appropriate embedding $\phi$ with the seemingly equivalent problem of finding an appropriate kernel. As we will see starting in Section 2.3, certain types of kernels are natural choices because they can recover geometric operators in the limit of large data sets. Moreover, as we will show in Section 2.5, these kernels can define a notion of discrete geometry on the finite data set. These facts will allow us to use a large class of kernel functions called local kernels to truly explore the nonlinear features of a data set.

### 2.2.3 Distance based extensions of PCA: Isomap and path metric spaces

Metric Multi-Dimensional Scaling (MDS) gives a general technique for constructing a low dimensional embedding which minimizes distortion of a metric. Given a metric $d$ represented by a matrix of squared distances $D_{ij} = d(y_i, y_j)^2$ metric MDS reconstructs the Gram matrix $G$ then uses the first $k$ eigenvectors of the Gram matrix $G$ as an optimal embedding of the data set $x_i$ into $\mathbb{R}^k$. As shown in [21], the matrix of inner products can be approximately recovered from a matrix of squared distance by double centering the matrix of squared distances

$$G \approx D - \frac{1}{N} D \mathbf{1}_M - \frac{1}{M} \mathbf{1}_M^T D + \frac{1}{M^2} \mathbf{1}_M^T D \mathbf{1}_M$$

where $\mathbf{1}_M$ is the vector of all ones in $\mathbb{R}^M$. By applying MDS to the reconstructed Gram matrix $G$, one can show that Metric MDS gives an optimal low dimensional representation of the data set in the sense of minimizing the distortion of the metric $d$ [21].

Metric MDS seems to suggest a promising nonlinear extension of PCA to nonlinear subsets of $\mathbb{R}^n$. On any Riemannian (or even Finslerian) manifold, one can define a unique distance between points which is given by the infimum of the path length over all paths on the manifold. This distance is sometimes called the intrinsic metric, and a metric space with the intrinsic metric is called a path metric space. Thus, if our data lies on a manifold embedded in $\mathbb{R}^n$, the euclidean distance will not make this data set a path metric space. One may hope to 'linearize' the manifold by finding an embedding in a Euclidean space that makes the manifold with the euclidean distance a path metric space. Unfortunately, this is only possible for very special manifolds such as the developable surfaces we discuss below. However, later we will return to this idea by considering an embedding into an infinite dimensional linear space such that the ambient metric makes the embedded manifold a path metric space.

Isomap [22] is a Nonlinear Dimensionality Reduction technique based on a modification of Multi-Dimensional Scaling (MDS) [21]. Isomap requires that the high-dimensional input data lies near a low-dimensional developable manifold. The idea is to construct low-dimensional coordinates by applying MDS to inter-point distances which are computed on the manifold. To find these distances, Isomap starts with a k-nearest-neighbor graph consisting of Euclidean distances. Then all inter-point distances are computed as the length of the shortest path in the graph. Finally, MDS is applied to the matrix of inter-point distances.

The generative model for Isomap assumes that the data is uniformly sampled for a

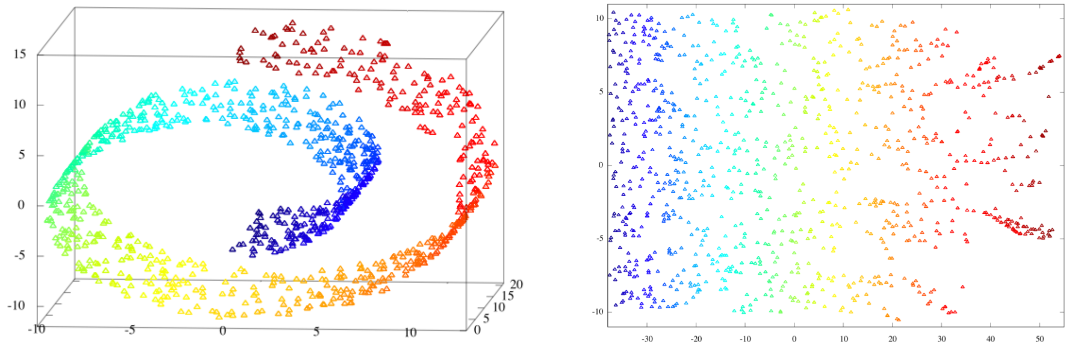


Figure 2.1: Isomap applied to a 2D manifold (left) produces the intrinsic coordinates (right)

special type of smooth manifold called a developable manifold. Developable manifolds can always be parameterized in the form $y = F(x)$ such that $\delta(y_i, y_j) = ||x_i - x_j||_2$ where $\delta$ is the geodesic distance on the manifold and $y$ are the observed data and $x$ are the latent data. Under this model, and as the sampling density tends to infinity, Isomap will recover the latent variables exactly [21]. Isomap has also been successfully applied to a broader class of manifolds called geodesically convex manifolds, however in these cases Isomap may distort the probability distribution of the latent variables [22].

## 2.3 The geometric approach to dimensionality reduction

For a finite set of points on a manifold embedded in a high-dimensional Euclidean space, a diffusion map is a nonlinear map to a lower-dimensional space. In rough analogy to the principal components from a singular value decomposition, the components of a diffusion map [2, 12] are eigenvectors of a transition matrix for a random walk on the data set. Under appropriate normalizations, the transition matrix is a discrete approximation to the Laplace-Beltrami operator [23], which is by definition the divergence of the gradient on the manifold inherited from the embedding. Thus the components of the diffusion map will be approximations to eigenfunctions of this operator, and we will see that the diffusion map minimizes an energy functional that measures the distortion of the manifold's geometry.

In this section we introduce the notion of a geometric prior for a data set which motivates the geometric interpretation of certain kernel based extensions of PCA. We then overview the results of the diffusion maps literature and the geometric interpretation of these results. This will lead to a novel generalization of diffusion maps to a large class of kernel functions called *local* kernels which are given a geometric interpretation in Section 2.4. While the results of Section 2.4 are only true in the limit of large data, in Section 2.5 we show that there is also a since in which local kernels define a discrete geometry for finite data sets.

### 2.3.1 The geometric prior

Consider a compact $n$-dimensional differentiable manifold $\mathcal{M}$ embedded in $\mathbb{R}^m$ by $\iota : \mathcal{M} \to \mathbb{R}^m$. For many applications we will not have any a priori knowledge of the underlying manifold or the embedding. Instead we simply have a finite collection of vectors from the embedding space $\mathbb{R}^m$, and we would like to assume that this data set lies on or near the image of the manifold under the embedding. This is a nonparametric model for our data, since we assume that the manifold and embedding exist but we do not assume any specific functional form for either. In this section we explore the assumptions needed to study a geometry on $\mathcal{M}$. For the rest of this chapter we use geometry as a short hand for Riemannian geometry which is completely determined by a smoothly varying inner product on the tangent spaces called the Riemannian metric. For convenience whenever we refer to the metric on a manifold this means the Riemannian metric.

Letting $x \in \mathcal{M}$ and $u, v \in T_x\mathcal{M}$, we can define the induced metric on $\mathcal{M}$ by

$$g_x^{\iota}(u,v) = \langle D\iota(x)u, D\iota(x)v \rangle_{\mathbb{R}^m} = (D\iota(x)u)^T (D\iota(x)v) \tag{2.1}$$

A simple approach to studying the geometry of an embedded manifold is to assume that the geometry is given by the induced metric. This trivially implies that $\iota$ is an isometry. Equivalently, one may assume that there is an existing metric $g$ and that the embedding $\iota$ is an isometry, which then implies that $g = g^{\iota}$. If nothing is known about the embedding $\iota$ then it would not be possible to study an existing metric $g$ since $\iota$ could have distorted the geometry $g$ and the problem is underspecified.

Suppose that $g_x(u, v)$ is an arbitrary metric on $\mathcal{M}$, then the embedding will distort $g_x$. To make this distortion explicit, note that for any $\hat{u}, \hat{v} \in T_{\iota(x)}(\iota(\mathcal{M}))$ we can define $D\iota(x)^\dagger = \left(P_{T_{\iota(x)}(\iota(\mathcal{M}))} D\iota(x)\right)^{-1}$ where $P_{T_{\iota(x)}(\iota(\mathcal{M}))}$ is the projection onto of $\mathbb{R}^m$ onto the $n$-dimensional tangent plane at $\iota(x)$. Now we can express the inner product in the embedding space $\mathbb{R}^m$ as

$$\langle \hat{u}, \hat{v} \rangle_{\mathbb{R}^n} = g_x(D\iota(x)^\dagger \hat{u}, D\iota(x)^\dagger \hat{v}). \tag{2.2}$$

By choosing a local coordinate system $(x^1, ..., x^n)$ we can write the metric at $x$ as a matrix by setting $g_{ij}(x) = g_x\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j}\right)$. We can also write $D\iota(x)$ as a matrix by setting $D\iota_{ij}(x) = \left\langle D\iota(x)\frac{\partial}{\partial x^i}, e_j \right\rangle$. Then for fixed $x$ we can rewrite equation (2.2) as

$$\langle \hat{u}, \hat{v} \rangle_{\mathbb{R}^m} = \hat{u}^T (D\iota^\dagger)^T g (D\iota^\dagger) \hat{v}.$$

We can then define the induced volume form by setting

$$d\text{vol} = \sqrt{\det g_{ij}(x)} dx_1 \wedge \cdots \wedge dx_n. \tag{2.3}$$

This volume form gives rise to a natural probability measure $d\mu$ by

$$d\mu = \frac{d\text{vol}}{\int_{\mathcal{M}} 1 d\text{vol}}. \tag{2.4}$$

**Example: Induced measure on the circle embedded in the plane.** Let $S^1$ be parameterized by $\theta \in [0, 2\pi)$, and let $u, v \in T_\theta S^1 \cong \mathbb{R}$. Then the embedding $\phi(\theta) = (\sqrt{2}\cos(\theta), \sin(\theta))^T \in \mathbb{R}^2$ induces the metric

$$g_\theta(u, v) = u^T (D\phi(\theta))^T (D\phi(\theta)) v = u^T (1 + \sin^2(\theta)) v.$$

Thus the induced measure for this embedding is $d\mu(\theta) = \frac{1}{\int_0^{2\pi}\sqrt{1+\sin^2(t)dt}}\sqrt{1+\sin^2(\theta)}$.

### 2.3.2 Diffusion maps

In this section we present the theory which motivates the diffusion maps algorithm as established in [2]. The key result is that a certain type of kernel function can be used to approximate the Laplace-Beltrami operator on a manifold embedded in euclidean space. If a data set in euclidean coordinates can be assumed to satisfy a geometric prior, then we can give a geometric interpretation of the kernel based extensions of PCA. We will explain the geometric interpretation in Section 2.3.3, and then in Section 2.4 we will present a novel generalization of these results.

Assume that $N$ data points are sampled from an $n$-dimensional manifold $\mathcal{M}$ which is embedded in $\mathbb{R}^m$. Consider a symmetric kernel $J_\epsilon(x,y) = h_\epsilon(|x-y|)$; in our applications, we use a Gaussian $h_\epsilon(z) = e^{-z^2/(2\epsilon)}$. The kernel is first used to interpolate the sampling density $p(x)$ of the data as

$$P(x) = \int_{\mathcal{M}} J_\epsilon(x,y)p(y)dy \approx \sum_{i=1}^{N} J_\epsilon(x,x_i) \tag{2.5}$$

where $\{x_i\}_{i=1}^N \in \mathcal{M} \subset \mathbb{R}^m$ are the discrete observations, which are assumed to be sampled from the true density $p(x)$. In the limit of large $N$ the discrete approximation becomes equality.

The idea is to use a kernel density estimate to approximate the heat kernel $e^{-t\Delta}$, where $\Delta$ is the Laplace-Beltrami operator on $\mathcal{M}$ inherited from $\mathbb{R}^m$. Here we use the convention that the eigenvalues of $\Delta$ are positive. Unfortunately, the sampling density will affect such a kernel estimate, but a subtle renormalization can recover the correct operator. The sampling bias parameter $\alpha$ will be introduced to control the influence that the sampling density will have on the geometry.

Define the discrete version of the kernel to be the $N \times N$ matrix $J$, where $J_{ij} = J_\epsilon(x_i, x_j)$.

The discrete version of the interpolated measure above is the $N \times N$ matrix $P = \mathrm{diag}(J1_N)$, where $1_N$ denotes the $N$-vector of ones. That is, $P$ is the diagonal matrix where $P_{ii}$ is the $i$th row sum of $J$.

An $N \times N$ matrix with nonnegative entries is *stochastic* if its rows add to 1. The eigenvalues of a stochastic matrix are nonnegative and the largest eigenvalue is 1. In our case, the matrix $P^{-1}J$ is stochastic. However, it is still biased by the sampling density.

The parameter $\alpha$ controls the influence of the sampling. For fixed $\epsilon > 0$ and $\alpha \geq 0$, define

$$\begin{aligned} K(x,y) &= \frac{J_\epsilon(x,y)}{P(x)^\alpha P(y)^\alpha} & K &= P^{-\alpha} J P^{-\alpha} \\ Q(x) &= \int_{\mathcal{M}} K(x,y)p(y)dy & Q &= \mathrm{diag}(K1_N) \\ F_{\epsilon,\alpha}(f)(x) &= \int_{\mathcal{M}} \frac{K(x,y)}{Q(x)} f(y)p(y)dy & T &= Q^{-1}K \end{aligned} \tag{2.6}$$

where we have, by abuse of notation, listed the discrete counterpart of each operator on the left as an $N \times N$ matrix on the right. Note that $T$ is a stochastic matrix.

The key result of diffusion maps is that the matrix $T$ can be used to approximate a differential operator $\mathcal{L}$ that captures the geometry of the data set. Namely, it is shown in [2] that

$$\mathcal{L}\varphi \equiv \lim_{\epsilon \to 0} \frac{I - F_{\epsilon,\alpha}}{\epsilon} \varphi = \Delta\varphi + 2\frac{\nabla(p^{1-\alpha})}{p^{1-\alpha}} \cdot \nabla\varphi.$$

Alternatively, if we write the sampling density as $p = e^{-U}$, then

$$\mathcal{L}\varphi = \Delta\varphi - 2(1-\alpha)\nabla U \cdot \nabla\varphi. \tag{2.7}$$

In the special case $\alpha = 1$, we recover the Laplace-Beltrami operator $\Delta$; in the case of general $\alpha$, we recover a backwards Fokker-Planck operator $\mathcal{L}$ for a diffusion process.

Now we can define the diffusion maps as in [2], which give a low-dimensional representation of the data. They are defined for each time scale $t$ by the projection of the data onto the eigenfunctions of the operator $F_{\epsilon,\alpha}^{t/\epsilon} \approx T^{t/\epsilon}$. Without loss of generality, on the scale of 1 time unit, we can define the eigenfunctions of $T^{1/\epsilon}$ by $T^{1/\epsilon}\psi_l = \lambda_l^2\psi_l$ where $1 = \lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_L > 0$. Here we have used the fact since $T$ is a stochastic matrix, it has eigenvalues between 0 and 1, and therefore so does $T^{t/\epsilon}$ for any $t, \epsilon > 0$. The *diffusion map* $\Psi_{\alpha,t} : \mathcal{M} \to \mathbb{R}^L$ at the data point $x_i$ is given by

$$\Psi_{\alpha,t}(x_i) = [\lambda_1^t\psi_1(x_i), ..., \lambda_L^t\psi_L(x_i)]^T \tag{2.8}$$

where $L$ is chosen large enough that $\lambda_{L+1}^t$ is sufficiently negligible. Note that $(\psi_l)_i = \psi_l(x_i) = \langle\psi_l, \delta_{x_i}\rangle$ is the $i$th coordinate of the $l$th eigenvector of the matrix $T$.

The eigenvalues and eigenvectors of the $N \times N$ nonsymmetric matrix $T$ need to be calculated. Fortunately, $T$ is closely related to a symmetric matrix $\hat{T}$ where calculations can be done more stably, and which furthermore can be readily described as a matrix of inner products, or a *Gramian matrix.* Define

$$\hat{T} = Q^{-1/2}KQ^{-1/2} = Q^{1/2}TQ^{-1/2}.$$

It is preferable to do the eigenvalue calculation with the symmetric matrix $\hat{T}$, since it has the same eigenvalues as $T$. If $\xi_l$ are the eigenvectors of $\hat{T}$, then $\psi_l = Q^{-1/2}\xi_l$ are the eigenvectors and $\lambda_l^2 = a_l^{1/\epsilon}$ are the eigenvalues of $T^{1/\epsilon}$, collectively satisfying $T^{1/\epsilon}\psi_l = \lambda_l^2\psi_l$.

If we let $U$ be the matrix whose columns are $\xi_l$, then $\hat{T} = U\Lambda U^T$ where $\Lambda_{ll} = \lambda_l^{2\epsilon}$. Then we let $V = Q^{-1/2}U$ be the matrix whose columns are $\psi_l$ and we let $R = Q^{1/2}U = QV$ be the matrix whose columns are $R_l = Q\psi_l$. This gives a biorthogonal decomposition of

$$T = Q^{-1/2}\hat{T}Q^{1/2} = Q^{-1/2}U\Lambda U^TQ^{1/2} = V\Lambda R^T = V\Lambda V^TQ.$$

Moreover, since $\hat{T}$ is symmetric and positive definite it is a Gramian matrix, and similarly $V\Lambda V^T$ is a Gramian matrix.

We now connect $T^{t/\epsilon}$ to the Gramian matrix

$$(T^{t/\epsilon}Q^{-1})_{ij} = \sum_{l=0}^{N-1} \lambda_l^{2t}\psi_l(x_i)\psi_l(x_j) \approx \sum_{l=0}^{L} \lambda_l^{2t}\psi_l(x_i)\psi_l(x_j) = \langle \Psi_{\alpha,t}(x_i), \Psi_{\alpha,t}(x_j)\rangle \,.$$

The distance which corresponds to the inner product above is the diffusion distance in the space of functions $L^2(\mathcal{M}, d\mu/p(x))$ such that

$$D_t(x_i,x_j)^2 = ||e^{-t\mathcal{L}}\delta_{x_i} - e^{-t\mathcal{L}}\delta_{x_j}||^2_{L^2(\mathcal{M},d\mu/p(x))} \approx ||\Psi_{\alpha,t}(x_i) - \Psi_{\alpha,t}(x_j)||_2^2 \tag{2.9}$$

which is shown to satisfy the requirements of a distance in [2]. This shows that the Euclidean distance between diffusion coordinates approximates the diffusion distance on the underlying manifold $\mathcal{M}$. In fact, the theory of multidimensional scaling shows that the diffusion coordinates give the best preservation of the diffusion distance of any $L$ dimensional representation of the data [2, 21].

As mentioned above, $\mathcal{L} = \Delta$ in the special case $\alpha = 1$ in (2.7), and it is shown in [2] that $F_{\epsilon,1}^{t/\epsilon}$ approximates the heat kernel in the sense that

$$\lim_{\epsilon\to 0} F_{\epsilon,1}^{t/\epsilon} = e^{-t\Delta}.$$

The approximation $T_{ij}^{t/\epsilon} \approx \langle \delta_{x_i}, e^{-t\Delta}\delta_{x_j}\rangle$ becomes equality in the limit as $\epsilon \to 0$ where the limit assumes that the number of samples $N$ increases to infinity simultaneously. So for $\alpha = 1$, the matrix $T^{t/\epsilon}$ is a discrete approximation to the heat kernel.

Finally, we give a new interpretation of the parameter $\alpha$. Intuitively, $\alpha$ will determine how much influence the sampling density will have on the operator $\mathcal{L}$. More formally, we now show that $\alpha$ corresponds to a conformal change of metric and that the operator $\mathcal{L}$

constructed by diffusion maps is related to the Laplacian with respect to the new metric. Note that the manifold $\mathcal{M}$ inherits a Riemannian metric $g$ from the ambient space $\mathbb{R}^m$ and a volume form $d\text{vol} = \sqrt{|\det(g)|}d\omega$ which is given by the sampling density $\sqrt{|\det(g)|} = p = e^{-U}$. Consider the conformal change of metric $\tilde{g} = e^{4(1-\alpha)U/(n-2)}g$ and note that the Laplacian with respect to this metric is given by

$$\Delta_{\tilde{g}}\varphi = e^{-4(1-\alpha)U/(n-2)}(\Delta_g\varphi - 2(1-\alpha)\nabla\varphi \cdot \nabla U) = e^{-4(1-\alpha)U/(n-2)}\mathcal{L}\varphi.$$

This shows that the operator $\mathcal{L}$ constructed by the diffusion map is always a scalar function multiple of a Laplacian $\Delta_{\tilde{g}}$ given by the conformal change of metric. Therefore the $\alpha$ parameter determines the degree to which the sampling density influences the geometry of the diffusion mapped data. By taking $\alpha = 1$ we can remove this influence entirely, and recover the Laplace-Beltrami operator $\mathcal{L} = \Delta_g$.

The results of Section 2.3.3 will apply for arbitrary $\alpha$, and we interpret the operator $F_{\epsilon,\alpha}^{t/\epsilon}$ as a generalization of the heat kernel with respect to the conformal change in metric described above. We will see in Sections 3.3.1 and 3.4 that for certain dynamical systems $\alpha = 1/2$ will allow us to adapt the diffusion map to the invariant measure.

### 2.3.3 Geometric interpretation of diffusion maps

In this section, we first show that the diffusion map gives intrinsic coordinates which are independent of the observation space. Next we show that the diffusion coordinates give a mapping (an embedding for $L$ sufficiently large) of our manifold into $\mathbb{R}^L$ that gives the minimal distortion of the geometry of the manifold. Then we interpret the parameter $t$ as the 'scale' of our approximation to the manifold, which allows us to reduce the dimensionality while maintaining the geometry as well as possible. Finally, we will show that the parameter $\alpha$ corresponds to a choice of measure which will allow us to match the invariant measure of a dynamical system.

In Section 2.3.2 we saw that given a finite data set sampled from a manifold embedded in $\mathbb{R}^m$, a diffusion map constructs a discrete approximation to a heat kernel on the manifold (when $\alpha \neq 1$ the heat kernel is with respect to a conformal change of metric). The heat kernel on a manifold is equivalent to the Riemannian metric, so we can preserve the geometry of an embedding by preserving the heat kernel. Since the diffusion coordinates of (2.8) approximate the dominant eigenfunctions of the heat kernel, they give the best low-dimensional approximation to the heat kernel. Moreover, since the the heat kernel is invariant under isometries, the diffusion coordinates will be the same for any isometric copies of the manifold. Thus, the diffusion coordinates are intrinsic and will not depend on the observation space.

Note that since the matrix $T^{t/\epsilon}Q^{-1}$ is a Gramian matrix, the diffusion coordinates (given by the eigenfunctions $\psi_l$) are the principal components of the associated inner product space [21]. This naturally gives two interpretations to the diffusion maps. First, in terms of Kernel Principal Component Analysis, we will see that the diffusion map maintains local distances. Second, in terms of multidimensional scaling, we will see that the diffusion map gives the minimum distortion of the nonlocal diffusion distances, which are a generalization of geodesic distances.

First, to see that the diffusion map is a Kernel Principal Component Analysis, note that the diffusion coordinates are given by the solutions of the generalized eigenvalue problem $A\psi = \lambda B\psi$ where $A_{ij} = K(x_i, x_j)$ and $B = Q$. As shown in [5], this implies that the diffusion coordinates give an embedding into $\mathbb{R}^L$ that minimizes the functional

$$E[\Phi] = \sum_{l=1}^{L} \int_{\mathbb{R}^N} \int_{\mathbb{R}^N} (\phi_l(x) - \phi_l(y))^2 \frac{K(x,y)}{Q(x)} p(x)p(y)dxdy$$

over maps $x \mapsto \Phi(x) = (\phi_1(x), ..., \phi_L(x))$ subject to the constraint $\langle \phi_i, \phi_j \rangle = \delta_{ij}$. Since $K_{\epsilon,\alpha}$ is a localizing kernel with exponential decay, this functional penalizes moving points apart that were close in the original space. In this sense, the diffusion coordinates seek to minimizes the local distortion of distance. This shows that the diffusion map preserves the local geometry of the embedded manifold.

Whereas the above interpretation shows that the diffusion map minimizes a stress functional, we can also interpret the diffusion map in terms of multidimensional scaling. In the previous section we saw that the matrix $T^{t/\epsilon}Q^{-1}$ contains inner products, and that there is a corresponding metric which is given by the diffusion distance from (2.9). The theory of multidimensional scaling shows that the diffusion coordinates minimize the distortion given by

$$\mathcal{P}[\Phi] = \sum_{i,j=1}^{N} \left(D_t(x_i, x_j) - ||\Phi(x_i) - \Phi(x_j)||\right)^2$$

with $\Phi$ constrained as above [21]. Thus the Euclidean distance in the diffusion coordinates is as close as possible to the diffusion distances on the manifold. Optimality in terms of the functionals $E$ and $\mathcal{P}$ shows that the diffusion map will maintain local distances, while changing long distances to approximate the intrinsic diffusion distance.

Next we provide a geometric interpretation of the parameter $t$. The diffusion distances were introduced in [2] and shown to be distances on the manifold for each value of the parameter $t$. Intuitively, the parameter $t$ controls the scale at which we approximate the manifold. For example, as $t \to \infty$, the diffusion distance between all points approaches zero, and for $t$ large our data set is approximated as a single point. Conversely, as $t \to 0$ the diffusion distance is related to the geodesic distance $d$ on the manifold by the equation

$$\begin{aligned} d(x,y) &= \lim_{t\to 0} -4t \log \left\langle e^{-t\Delta}\delta_x, \delta_y \right\rangle \\ &= \lim_{t\to 0} -4t \log \left( \frac{||e^{-t\Delta/2}\delta_x||^2 + ||e^{-t\Delta/2}\delta_y||^2 - D_{t/2}(x,y)^2}{2} \right). \end{aligned}$$

The geodesic distance is equivalent to the Riemannian metric, so as $t \to 0$ the diffusion distance captures all the details of the geometry. Thus, the parameter $t$ allows us to control the scale at which we try to represent the manifold: $t$ large gives a coarse scale representation, and $t$ small gives the fine scale.

Of course, to represent the fine scale will require a high dimensional diffusion map, and often the fine scale will be dominated by noise. Thus by choosing $t$ large we can approximate our manifold at a coarse scale that will allow a low-dimensional representation while still maintaining the geometry in the sense of the functionals $E$ and $\mathcal{P}$.

## 2.4 Generalization of diffusion maps to local kernels

In this section we define *local* kernels and show that, given the geometric prior, in the limit of large data each local kernel defines a geometry on the embedded manifold. Section 2.4.1 introduces the formal definition of a local kernel and develops the theory of local kernels as a natural generalization of the results of diffusion maps in [2]. In Section 2.4.3 we give an practical example of how local kernels can be used to regularize the geometry on an embedded manifold. For convenience and clarity we restrict our construction in this section to manifolds without boundary; we conjecture that the results could be extended to manifolds with boundary following the technique of [2].

### 2.4.1 The limiting geometry of local kernels

As we saw in Section 2.3.2, the diffusion maps construction of Coifman and Lafon [2] starts with a kernel which can be written as a scalar function of the Euclidean distance squared, namely $J_\epsilon(x,y) = h(||x-y||^2/\epsilon)$. Such a kernel is sometimes called a *radial kernel*. In this section we introduce the terms *local* and *isotropic* which describe kernels that have special properties. We show that all radial kernels with fast decay are local and isotropic, and that this strongly restricts the geometry which a radial kernel can describe. The key lemma in the theory of diffusion maps (in [2]) applies to radial kernels, and in Lemma 2.4.2 we generalize this result to all local kernels. This leads to our main result in Theorem 2.4.1 which shows that every local kernel defines a geometry on the embedded manifold. The geometry is determined by the limit of the Hessian matrix of the local kernel. The limit of large data is needed, as shown in [2] and summarized in Section 2.3.2, in order to approximate the integral operator $G_\epsilon$ (defined below) and coincides implicitly with the limit $\epsilon \to 0$.

The key result of diffusion maps is that $(I - F_{\epsilon,1})/\epsilon \to \Delta_{\mathcal{M}}$ as $\epsilon \to 0$ so that the extrinsic effect of the sampling density is removed and the intrinsic Laplace-Beltrami operator on the manifold is recovered. We will show that this results depends on two properties of the kernel $J_\epsilon(x,y)$. The first property is that the kernel is *local*.

**Definition 2.4.1** (Local Kernel)**.** A kernel $K : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is called *local* if it can be written in the form $K(x,y) = \overline{K}(x, y-x)$ where $\overline{K}(x,z)$ is smooth and nonzero in a neighborhood of $z = 0$ with the properties

1. (*fast decay*) For some $c, \sigma > 0$ and all $x, z \in \mathbb{R}^n$ we have $0 \le \overline{K}(x,z) \le ce^{-\sigma||z||^2}$.

2. (*centered*) For all $x \in \mathbb{R}^n$, $\overline{K}(x,z)$ has a local maxima at $z = 0$.

We will make extensive use of the scaling $K_\epsilon(x,y) = \overline{K}\left(x, \frac{y-x}{\sqrt{\epsilon}}\right)$ which intuitively describes the geometry at scales coarser than $\epsilon$ (see Section 2.5). We will use this scaling to define a geometry in the limit of $\epsilon \to 0$. The fast decay property will allow the kernel operator $G_\epsilon f(x) = \epsilon^{-n/2}\int_{\iota(\mathcal{M})} K_\epsilon(x,y)f(y)dy$ to be localized to a $\sqrt{\epsilon}$-neighborhood of $x$, which will be important in Lemma 2.4.2. In Section 2.6 we will discuss the possibility of non-centered kernels to describe more general geometries. The following proposition establishes notation that will be used below as well as some important homogeneity properties of a scaled local kernel.

**Proposition 2.4.1** (Homogeneity of Local Kernels)**.** *Let $K(x,y) = \overline{K}(x, y-x)$ be a local kernel and define the $\epsilon$-scaled kernel $K_\epsilon(x,y) = \overline{K}\left(x, \frac{y-x}{\sqrt{\epsilon}}\right)$. The diagonal $k = K_\epsilon(x,x)$ and the Hessian matrix $K_x = -\epsilon H_y \left. K_\epsilon(x,y)\right|_{y=x}$ are independent of $\epsilon$ and there exists a unique symmetric positive definite square root of $K_x$ which we denote $K_x^{1/2}$.*

*Proof.* For any $x \in \mathbb{R}^m$ we have $k = K_\epsilon(x,x) = \overline{K}(x,0)$ and $K_x = -\epsilon H_y K_\epsilon(x,x) = -H_z\overline{K}(x,z)\big|_{z=0}$ which reveals these expressions to be independent of $\epsilon$. Since $\overline{K}(x,z)$

has a maximum at $z = 0$ the Hessian matrix $H_z\overline{K}(x,0)$ is negative definite and thus $K_x = -H_z\overline{K}(x,0)$ is positive definite and so the symmetric positive definite square root $K_x^{1/2}$ exists and is unique. □

Throughout this section, by a slight abuse of terminology, we will refer to the positive definite matrix $K_x$ as the Hessian of the local kernel $K$.

The second property of $J_\epsilon$ in the diffusion maps construction is that the kernel is *isotropic.*

**Definition 2.4.2** (Isotropic Local Kernel)**.** A local kernel $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is called *isotropic* if there exists a constant $c \neq 0$ such that for all $x \in \mathbb{R}^n$ the Hessian matrix $K_x = cI_{n\times n}$.

In the diffusion maps construction, the base kernel $J$ is assumed to be of the form $J(x,y) = h(||x-y||^2)$ where $h$ is a smooth scalar function with fast decay and is decreasing at zero, we first show that any kernel of this form is local and isotropic.

**Proposition 2.4.2** (Radial kernels are local and isotropic)**.** *Assume a kernel $J$ can be written in the form $J(x,y) = h(||x-y||^2)$ where $h$ has fast decay, at least two continuous derivatives at zero, and is decreasing at zero, then $J$ is local and isotropic.*

*Proof.* Since $h$ has fast decay, we only need to show that $J$ is centered and isotropic. We first write $J(x,y) = \overline{J}(x, y-x)$ where $\overline{J}(x,z) = h(||z||^2)$. Note that $\frac{\partial}{\partial z_i}\overline{J}(x,z) = 2h'(||z||^2)(z_i)$ so $D_z\overline{J}(x,0) = 0$. Similarly we have

$$\frac{\partial}{\partial z_j}\frac{\partial}{\partial z_i}\overline{J}(x,z) = 4h''(||z||^2)z_i z_j + 2h'(||z||^2)\frac{\partial}{\partial z_j}z_i$$

so for $i \neq j$ we have $\frac{\partial}{\partial z_j}\frac{\partial}{\partial z_i}\overline{J}(x,0) = 0$ and for $i = j$ we have $\frac{\partial}{\partial z_j}\frac{\partial}{\partial z_i}\overline{J}(x,0) = 2h'(0) = c < 0$ so $K_x = H_z\overline{J}(x,0) = cI_{n\times n}$ which implies that $J$ is isotropic. Since $H_z\overline{J}(x,0)$ is negative definite and $D_z\overline{J}(x,0) = 0$ we conclude that $J$ is centered and therefore $J$ is local. Thus $J$ is local and isotropic. □

We highlight two examples of radial kernels which are local and isotropic which will be used later.

**Example 2.4.1** (Examples of radial kernels)**.** *The standard kernel for the diffusion maps construction uses $h(x) = e^{-x}$ as the base function so that $J(x,y) = e^{-||x-y||^2}$. In Lemma 2.4.2 below we use the local isotropic kernel defined by $h(x) = \max\{1-x, 0\}$.*

While $J_\epsilon$ is local and isotropic, the diffusion maps kernel $K_{\epsilon,\alpha}$ has a very special type of anisotropy which is determined by the $\alpha$ parameter. As we have seen, this anisotropy is allows the diffusion maps construction to access different geometries which are conformally equivalent to the geometry induced by the ambient space. Our goal is to allow any type of anisotropic kernel and find the operators which can be approximated in the limit of $\epsilon \to 0$ using local kernels. The following example is the prototype of a local kernel which can be used to define a geometry.

**Example 2.4.2** (Prototypical local kernels)**.** *Let $A(x)$ be a matrix valued function on the embedded manifold $\iota(\mathcal{M})$ such that each $A(x)$ is a symmetric positive definite $m$-by-$m$ matrix. Note that $K(x,y) = e^{-(x-y)^T A(x)(x-y)}$ is a local kernel with Hessian $K_x = A(x) + A(x)^T$.*

In order to understand the limiting behavior of local kernels, we first need to generalize the following lemma from [2] which allows the approximation of the integral operator $G_\epsilon$.

**Lemma 2.4.1** (Expansion of Radial Kernels, Coifman and Lafon [2])**.** *Let $f$ be a smooth real-valued function on $\iota(\mathcal{M}) \subset \mathbb{R}^m$ and let $h$ have fast decay, then we have*

$$G_\epsilon f(x) = \epsilon^{-n/2} \int_{\iota(\mathcal{M})} h\left(\frac{||x-y||^2}{\epsilon}\right) f(y)dy = m_0 f(x) + \epsilon m_2(\omega(x)f(x) + \Delta f(x)) + \mathcal{O}(\epsilon^2)$$

*where $m_0$ and $m_2$ are constants determined by $h$, and $\omega(x)$ depends on the induced geometry of $\iota(\mathcal{M})$. The operator $\Delta$ is the Laplace-Beltrami operator for $\iota(\mathcal{M})$ with the induced metric.*

The next lemma generalizes this result to local kernels. We introduce the standard notation div and $\nabla$ to refer to the intrinsic operators on the embedded manifold such that $\Delta = -\mathrm{div} \circ \nabla$ is the Laplace-Beltrami operator for $\iota(\mathcal{M})$ with the induced metric. Note that the Hessian matrix $K_x$ is $m$-by-$m$ and is a linear operator on the tangent space of the ambient space $T_x\mathbb{R}^m \cong \mathbb{R}^m$, whereas the tangent space $T_x\iota(\mathcal{M})$ is only $n$-dimensional. In order for $K_x$ to operate on a tangent vector $u \in T_x\iota(\mathcal{M})$ we need the $m$-by-$n$ matrix $\Xi_x$ of the inclusion map $T_x\iota(\mathcal{M}) \to T_x\mathbb{R}^m$, however for convienence we identify $K_x$ and $K_x^{1/2}$ with $\Xi_x^T K_x \Xi_x$ and $\Xi_x^T K_x^{1/2} \Xi_x$ respectively.

**Lemma 2.4.2** (Expansion of Local Kernels)**.** *Let $f$ be a smooth real-valued function on $\iota(\mathcal{M}) \subset \mathbb{R}^m$ and let $K(x,y)$ be a local kernel. For each $x \in \iota(\mathcal{M})$ let $K_\epsilon, K_x$ and $K_x^{1/2}$ be defined as in Proposition 2.4.1. We have the following expansion*

$$\begin{aligned} G_\epsilon f(x) &= \epsilon^{-n/2} \int_{\mathcal{M}} K_\epsilon(x,y) f(y) dy \\ &= m_0 f(x) + \epsilon m_2(\omega(x) f(x) - \Delta_{\hat{g}} f(x)) + \mathcal{O}(\epsilon^{3/2}) \end{aligned}$$

*where $\hat{g} = K_x^{1/2} g K_x^{1/2}$ and $g$ is the induced metric on $\iota(\mathcal{M})$. The constants $m_0$ and $m_2$ are determined by $K$, and $\omega(x)$ depends only on the induced metric $g$.*

*Proof.* Let $x \in \mathcal{M}$, since $K_\epsilon$ is a local kernel it is centered so we can expand $K_\epsilon$ in a Taylor series for the variable $y$ based at $x$ as

$$\begin{aligned} K_\epsilon(x,y) &= K_\epsilon(x,x) - (y-x)^T H_y K_\epsilon(x,x)(y-x) + \mathcal{O}(||y-x||^3) \\ &= k - (y-x)^T \frac{K_x}{\epsilon}(y-x) + \mathcal{O}(||y-x||^3) \end{aligned}$$

Since the kernel is local it has fast decay, thus we can restrict the integral in $G_\epsilon$ to a $\sqrt{\epsilon}$

neighborhood of $x$.

$$
\begin{aligned}
G_\epsilon f(x) &= \epsilon^{-n/2}\int_{N_{\sqrt{\epsilon}}(x)} K_\epsilon(x,y)f(y)dy + \mathcal{O}(\epsilon^{3/2}) \\
&= \epsilon^{-n/2}\int_{N_{\sqrt{\epsilon}}(x)} \left(k-(y-x)^T\frac{K_x}{\epsilon}(y-x)\right)f(y)dy + \mathcal{O}(\epsilon^{3/2})
\end{aligned}
$$

Now make the change of variables $\hat{y}-x = K_x^{1/2}(y-x)$ so that $d\hat{y} = \sqrt{\det(K_x)}dy$ and define the radial kernel $h(||\hat{y}-x||^2/\epsilon) = \max\{k-||\hat{y}-x||^2/\epsilon, 0\}$ then we can write

$$
\begin{aligned}
G_\epsilon f(x) &= \frac{\epsilon^{-n/2}}{\sqrt{\det(K_x)}}\int_{K_x^{1/2}N_{\sqrt{\epsilon}}(0)+x} h(||\hat{y}-x||^2/\epsilon)f(K_x^{-1/2}(\hat{y}-x)+x)d\hat{y} + \mathcal{O}(\epsilon^{3/2}) \\
&= \frac{\epsilon^{-n/2}}{\sqrt{\det(K_x)}}\int_{N_{\lambda\sqrt{\epsilon}}(x)} h(||\hat{y}-x||^2/\epsilon)\hat{f}(\hat{y})d\hat{y} + \mathcal{O}(\epsilon^{3/2}(1+\lambda^{-3})) \\
&= \frac{\epsilon^{-n/2}}{\sqrt{\det(K_x)}}\int_{\iota(\mathcal{M})} h(||\hat{y}-x||^2/\epsilon)\hat{f}(\hat{y})d\hat{y} + \mathcal{O}(\epsilon^{3/2})
\end{aligned}
$$

where $\lambda > 0$ is the infimum of the smallest singular values of the matrices $K_x^{1/2}$ on the compact manifold and $\hat{f}(\hat{y}) = f(K_x^{-1/2}(\hat{y}-x)+x)$. Note that $\hat{f}(x) = f(x)$ so by the previous lemma we have

$$
\begin{aligned}
G_\epsilon f(x) &= m_0 f(x) + \epsilon m_2\left(\omega(x)f(x) - \Delta_g f(K_x^{-1/2}(\hat{y}-x)+x)\Big|_{\hat{y}=x}\right) + \mathcal{O}(\epsilon^{3/2}) \\
&= m_0 f(x) + \epsilon m_2(\omega(x)f(x) - \Delta_{\hat{g}} f(x)) + \mathcal{O}(\epsilon^{3/2})
\end{aligned}
$$

To justify the final equation, note that for $\hat{y}$ in a small enough neighborhood of $x$, the map $\kappa(\hat{y}) = K_x^{-1/2}(\hat{y}-x)+x$ is a diffeomorphism such that $\kappa(x) = x$ and $D\kappa(x) = K_x^{-1/2}$. In this small neighborhood of $x$, the change of variables $y = \kappa(\hat{y})$ induces the pullback metric

$\hat{g}_x(u,v) = g_x(D\kappa(x)^{-1}u, D\kappa(x)^{-1}v) = g_x(K_x^{1/2}u, K_x^{1/2}v)$. Exactly at the point $\hat{y} = y = x$ Laplacian $\Delta_g$ of $\hat{f}(\hat{y}) = f \circ \kappa(\hat{y})$ is equivalent to the Laplacian $\Delta_{\hat{g}}$ of $f(y)$ via the change of variables $y = \kappa(\hat{y})$. Denoting the new metric by $\hat{g} = K_x^{1/2} g K_x^{1/2}$ completes the proof. □

Note that when $K_x$ is the identity map then $\hat{g} = g$ and we recover Lemma 2.4.1 extended to all local isotropic kernels. The following theorem is the main result of this chapter and shows that given the geometric prior, a local kernel can be used to approximate the Laplace-Beltrami operator with respect to the Riemannian metric $\hat{g} = K_x^{1/2} g K_x^{1/2}$ in the limit as $\epsilon \to 0$.

**Theorem 2.4.1.** *Let $K(x,y)$ be a Riemannian local kernel, let $\hat{g}$ be defined as in Lemma 2.4.2, and set*

$$L_\epsilon f = \frac{\mathrm{Id}(f) - (G_\epsilon 1)^{-1} G_\epsilon f}{\epsilon}.$$

*Then $\lim_{\epsilon \to 0} L_\epsilon = c\Delta_{\hat{g}}$ where $c$ is a constant depending on $K$.*

*Proof.* By the Lemma 2.4.2 we have $G_\epsilon 1 = m_0 + \epsilon m_2(\omega(x) + \Delta_{\hat{g}} 1) + \mathcal{O}(\epsilon^{3/2})$. Note that for constants $a, b, c, d$ with $c > 0$ we can expand the ratio

$$\frac{a + \epsilon b}{c + \epsilon d} = \frac{a}{c} + \epsilon \frac{cb - ad}{c^2} + \mathcal{O}(\epsilon^2).$$

Thus we can expand the ratio

$$(G_\epsilon 1)^{-1} G_\epsilon f = f(x) + \epsilon \frac{m_0 m_2(\omega(x) f(x) + \Delta_{\hat{g}} f(x)) - m_0 m_2 \omega(x) f(x)}{m_0^2} + \mathcal{O}(\epsilon^{3/2})$$

where $m_0, m_2$ are evaluated at $\epsilon = 0$. Thus we have

$$L_\epsilon f = \frac{-m_2}{m_0} \Delta_{\hat{g}} f(x) + \mathcal{O}(\sqrt{\epsilon})$$

which implies the desired limit $\lim_{\epsilon\to 0} L_\epsilon = c\Delta_{\hat{g}}$. □

The Laplacian $\Delta_{\hat{g}}$ is equivalent to the Riemannian metric $\hat{g}$ in the sense that either can be uniquely recovered from the other [23]. Since the previous theorem shows that a local kernel defines the Laplacian with respect to the metric $\hat{g}$, we conclude that every local kernel determines a geometry on the embedded manifold. Of course, many local kernels could define the same geometry, and Lemma 2.4.2 reveals that it is the Hessian at $K_x$ that determines the geometry. This establishes the central result of this chapter, which is that every local kernel defines a geometry in the limit of large data. The next theorem establishes the converse, that every Riemannian geometry on a manifold can be represented by a local kernel.

**Theorem 2.4.2.** *Let $\iota(\mathcal{M})$ be an embedded Riemannian manifold with $g$ the induced metric. Let $\hat{g}$ be another Riemannian metric on $\mathcal{M}$, then there exists a local kernel $K$ such that for $L_\epsilon$ defined as in Theorem 2.4.1 we have* $\lim_{\epsilon\to 0} L_\epsilon = c\Delta_{\hat{g}}$.

*Proof.* For each $x \in \iota(\mathcal{M})$ choose local coordinates $\{x^i\}_{i=1}^n$ and write $g_x$ and $\hat{g}_x$ as symmetric positive matrices (see Section 2.3.1). Then we can define the symmetric positive definite square roots $g_x^{1/2}$ and $\hat{g}_x^{1/2}$ and $S_x = \hat{g}_x^{1/2} g_x^{-1/2}$ so that

$$\hat{g}_x = \hat{g}_x^{1/2}\hat{g}_x^{1/2} = S_x g_x S_x$$

where we have used the fact that all the matrices are symmetric. Now let $A_x = S_x^2$ and extend $A_x$ to act on all of $T_x\mathbb{R}^m$ by setting $A_x$ to be the identity on $(T_x\iota(\mathcal{M}))^\perp$. Note that the extended $A_x$ is still positive definite and is smooth in the variable $x$ since $g_x$ and $\hat{g}_x$ are smooth on $T_x\iota(\mathcal{M})$ and $A_x$ is constant on $(T_x\iota(\mathcal{M}))^\perp$. Thus we can define the smooth local kernel $K(x,y) = \exp\{-(y-x)^T A_x (y-x)/2\}$ so that the Hessian is given by $K_x = A_x$ and $K_x^{1/2} = S_x$. By Theorem 2.4.1, the local kernel $K(x,y)$ yields $\lim_{\epsilon\to 0} L_\epsilon = \Delta_{\hat{g}}$ as desired. □

Together, Theorems 2.4.1 and 2.4.2 show that Riemannian metrics are in one-to-one correspondence with equivalence classes of local kernels that have the same Hessian matrix $K_x$ for all $x \in \iota(\mathcal{M})$. Of course, it is also possible to use diffusion maps the find the Laplacian with respect to any Riemannian metric. By Nash's theorem [24], every Riemannian manifold admits an isometric embedding into $\mathbb{R}^M$ for $M$ large enough. To recover a given metric $\hat{g}$ with diffusion maps we would have to find a global isometric embedding of our manifold into a Euclidean space and use this to re-embedd our data. With the re-embedded data set we could use diffusion maps to find $\Delta_{\hat{g}}$ since $\hat{g}$ is the induced metric of this embedding. Of course in practice finding such a global isometric embedding is impractical.

The theory of local kernels provides an alternative which is valuable in two respects. First, just as the kernel trick provides a more efficient technique of computing a Gram matrix (see Section 2.2.2), a local kernel allows one to easily change the metric using only local information without having to construct a globally consistent embedding. This is a significant advantage when trying to form data driven techniques to modify the metric as we will see in the following sections. Second, the theory of local kernels gives an important geometric interpretation to many existing techniques which use local kernels such as $K(x,y) = e^{-||y-x||^2_{A(x)}}$ where $A(x)$ defines a special distance measure on the embedded data. The theory of local kernels shows that these techniques are changing the geometry of the embedded data, an analogy which will be explored further in Section 2.5. As we will see in later sections, understanding the geometric content of kernel based methods provides novel avenues for analyzing the data.

In the next section we demonstrate the numerical application of a local kernel to modify the geometry of a data set, and in the following section we demonstrate a new technique for data driven geometric regularization using local kernels.

### 2.4.2 Numerically recovering the flat metric on a torus with a local kernel

In this section we show that a local kernel can recover the flat metric on a torus embedded in $\mathbb{R}^3$ with nonzero Riemannian curvature. Let $\theta, \phi \in [0, 2\pi]$ be the intrinsic coordinates of the torus, the flat metric is given simply by $g_{\theta,\phi} = Id$, this is the product metric induced by the structure $T^2 = S^1 \times S^1$. Now consider the embedding $\iota : T^2 \to \mathbb{R}^3$ given by

$$\iota((\theta,\phi)) = \begin{bmatrix} (2+\sin\theta)\cos\phi \\ (2+\sin\theta)\sin\phi \\ \cos\theta \end{bmatrix} \qquad D\iota((\theta,\phi)) = \begin{bmatrix} \cos\theta\cos\phi & -(2+\sin\theta)\sin\phi) \\ \cos\theta\sin\phi & (2+\sin\theta)\cos\phi \\ -\sin\theta & 0 \end{bmatrix}$$

which induces a curved metric on the torus. Our goal is to use a local kernel to undo the curvature induced by the embedding and recover the flat metric.

We generated 8100 points on a uniform grid in $[0,2\pi]^2$ to represent the intrinsic variables and then mapped these points into $\mathbb{R}^3$ via $\iota$ to generate the observed variables. We first applied the Diffusion Maps algorithm to the observed data set with $\alpha = 1$ (since the points are not uniformly distributed on the embedded manifold) in order to approximate the first four eigenvectors of the Laplacian with respect to the curved metric from the embedding space. In Figure 2.2 we show these eigenfunctions plotted against the intrinsic variables along with the diffusion map embedding defined by the first three eigenfunctions (see section 2.3.2. As shown in Section 2.3.2, the diffusion maps algorithm estimates the Laplacian with respect to the Riemannian metric induced by the embedding, up to a conformal transformation.

To show that a local kernel could recover the Laplacian with respect to the flat metric we defined the following local kernel

$$K(x,y) = \exp\left\{-(y-x)^T A(x)(y-x)\right\} \qquad A(x) = \left(D\iota(\iota^{-1}(x))^\dagger\right)^T D\iota(\iota^{-1}(x))^\dagger.$$

Thus we have $\overline{K}(x,z) = e^{-z^T A(x) z}$ which implies that $K_x = A(x)$ and $K_x^{1/2} = D\iota(\iota^{-1}(x))^\dagger$. Since the metric induced by the embedding is $g_x = (D\iota(x))^T D\iota(x)$, Theorem 2.4.1 implies

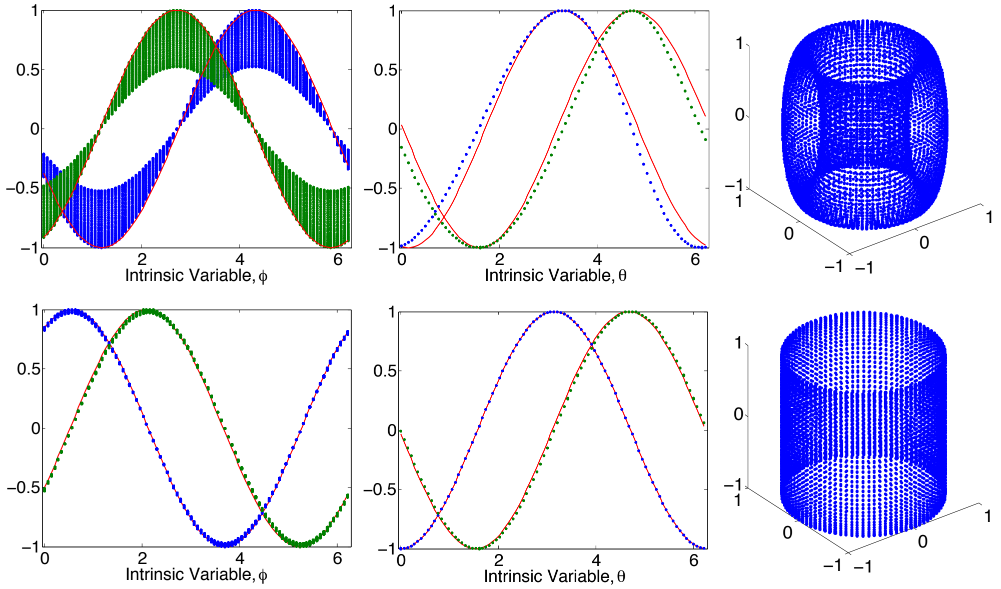


Figure 2.2: Top row, Left: First (blue) and second (green) eigenfunctions of the Laplacian with respect to the induced metric approximated by the diffusion maps construction, the red curves are sine functions with the same phase as the eigenfunctions; Middle: Eigenfunctions five (blue) and six (green), note that all the plots contain the same number of points and the vertical spread in this plot indicates the $\theta$ dependence; Right: The diffusion maps embedding of the torus using eigenfunctions one, two, and five. Bottom row, Same as above but using eigenfunctions from the local kernel construction described in the text, Left: Eigenfunctions one (blue) and two (green); Middle: Eigenfunctions three (blue) and four (green), Right: Embedding using eigenfunctions one, two, and three. Note that the surface shown in the bottom right plot is flat (zero Riemannian curvature) as expected but is not an embedding of the torus; this is because a smooth isometric embedding of the flat torus requires four dimensions.

that using the local kernel $K$ approximates the Laplacian with respect to the metric

$$\hat{g}_x = K_x^{1/2} g_x K_x^{1/2} = I$$

which is the flat metric on the torus. In Figure 2.2 we confirm this result numerically using the data set described above.

Of course, this kernel is not purely data driven since we have assumed knowledge of the embedding $\iota$. However, the point of this example is simply that a local kernel can achieve a desired change of metric without having to re-embed the data. Note that the first four eigenfunctions of Laplacian with respect to the local kernel $K(x,y)$, as shown in Figure 2.2, approximate $(\sin(\theta+\theta_0),\cos(\theta+\theta_0),\sin(\phi+\phi_0),\cos(\phi+\phi_0))$ up to a phase shifts $\theta_0$ and $\phi_0$. Since these coordinates give an isometric embedding of the flat torus into $\mathbb{R}^4$ we see that we have recovered the flat metric.

The example in this section illustrates the power of local kernels to modify the geometry of data, however, this example made use of the embedding function which is not typically known. In the next section we will examine a data driven approach to regularizing the geometry of data using local kernels and show how this is a natural extension of the regularization achieved by diffusion maps via the $\alpha$ parameter.

### 2.4.3 Data driven geometry regularization via local kernels

Diffusion maps correctly claims that in many applications the sampling distribution is an extrinsic factor which we do not wish to influence the geometry. However, as we have shown in Section 2.3.1, unless we know a priori that the embedding is isometric, the entire embedding geometry could be considered extrinsic. In this section we apply a data-driven anisotropic local kernel to regularize the geometry. Consider the example shown in Figure 2.3 where we have two data sets where one has simply be stretched in the $y$-coordinate. We can imagine a case where some unknown type of stretching has corrupted our data and we would like to find a construction which is independent of this type of effect. Since this stretching changes the embedding, the induced Riemannian metric is also changed, and this fact is reflected in the eigenfunctions estimated by diffusion maps as shown in Figure 2.3.

Motivated by this example, we introduce the following kernel. Let $J_\epsilon(x,y)=e^{-||x-y||/\epsilon}$ be the initial kernel. For each data point $x_i$, we consider the localized data set

$$z_j = J_\epsilon(x_i,x_j)(x_j-x_i)$$

centered at $x_i$ where $x_j$ ranges over the $\epsilon$ ball surrounding $x_i$, this guarantees the the regularized kernels will have fast decay. Letting $Z$ be the matrix with $z_j$ as its columns we can attempt to remove the bias towards any particular direction by finding the singular value decomposition $Z = U\Lambda V^T$ and then forming $\hat{Z} = UV^T$. We define the partially regularized kernel by $\hat{K}(x_i, x_j) = ||\hat{z}_j||$, where $\hat{z}_j$ is the $j$th column of $\hat{Z}$. We define the fully regularized kernel by letting $\tilde{Z} = U\Lambda^{-1}V^T$ and $\tilde{K}(x_i, x_j) = ||\tilde{z}_j||$, where $\tilde{z}_j$ is the $j$th column of $\tilde{Z}$. These kernels use a localized SVD to define a data driven anisotropy in an attempt to regularize the geometry and remove extrinsic features of the embedding. In Figure 2.3 we demonstrate the effects of these new kernels. Note that for these closely related data sets, the corresponding eigenfunctions for the diffusion maps construction look very different, while the eigenfunctions for the regularized kernels are more similar.

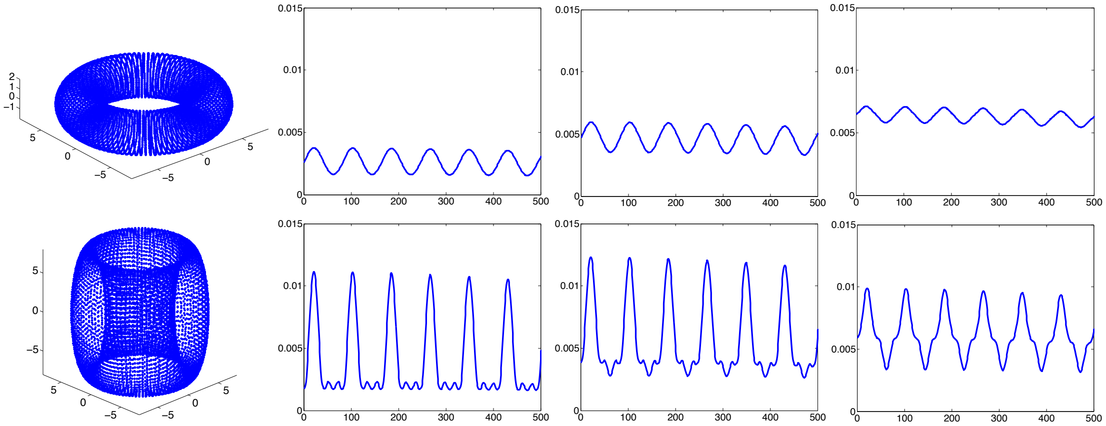


Figure 2.3: In the first column we show two data sets where the second data set is formed simply by multiplying the y-coordinates of the first data set by 3. In each row we plot the second eigenfunction of the Laplacian (associated to the first nonzero eigenvalue) for the corresponding data set by using the diffusion maps regularization (second column, $\alpha = 1$), using partial SVD regularization (third column) and full SVD regularization (fourth column).

An important remaining goal is to describe analytically the new geometry defined by the local SVD regularization. Moreover, the idea of regularizing the geometry is to remove all the geometric content of the embedding. This is closely related to a theoretical idea called geometrization, which seeks to define an intrinsic geometry by choosing an arbitrary initial Riemannian metric and then evolving the metric according to a PDE called a geometric evolution equation. One example is the Ricci flow which is given by a nonlinear PDE and intuitively expands areas of negative curvature and contracts areas of positive curvature to regularize the metric.

## 2.5 Local kernels and the discrete exterior calculus

In the previous section we saw that local kernels define a geometry in the limit of large data. In this section we show that local kernels can be used to define discrete geometries on finite data sets. This is achieved by translating the diffusion maps construction of the Laplacian into a formal language for describing a discrete geometry called the discrete exterior calculus, which was developed in [6, 20]. For simplicity we restrict the discussion to the kernel used by diffusion maps; we conjecture that the result generalizes to all local kernels.

In Section 2.5.1 we show the formal connection between the discrete exterior calculus and diffusion maps by showing that the corresponding approximations of the Laplacian differ only by multiplication by a constant. In Section 2.5.2 we complete the translation of diffusion maps into the discrete exterior calculus by introducing an abstract simplicial complex on the data set given by the Vietoris-Rips complex. Moreover, using the results of [25] we show that for sufficiently dense sampling, the discrete geometry should have the same homology as the true manifold. Finally, in Section 2.5.3 we use the discrete geometry to compute Betti numbers for a torus and a genus-2 surface. We also show how to numerically approximate closed representatives of the cohomology classes.

### 2.5.1 The discrete geometry of local kernels

In [6, 20] an alternative discrete construction of Laplacian operators is defined via a simplicial complex on a data set. In this section we unify the two constructions by interpreting the normalized transition matrix of Diffusion Maps as a weighted simplicial complex corresponding to a Riemannian Metric on the underlying manifold. This new interpretation will be give a new geometric insight into the Diffusion Maps construction and will be important in correctly interpreting the results of Diffusion Maps applied to complex dynamical data. We will define a simplicial complex on our data set by simply taking the complete graph of the data set and including every possible simplex. While this seems excessive, we will use a probabilistic metric to insure that almost all these simplices are given a negligible weight. This reveals the power of Diffusion Maps as a probabilistic approach to geometry, Diffusion Maps weights simplices continuously instead of making a binary decision for each simplex.

Let $\{x_i\}_{i=1}^S$ be our discrete observations, following [6] we let the basic $k$-simplices be given by ordered collections of $k+1$ data points. Thus there are $S^{k+1}$ $k$-simplices, and we represent $k$-chains as column vectors of length $S^{k+1}$ with each entry giving the weight of the corresponding simplex. We now construct the discrete boundary operator on 1-simplices following [6] as

$$\partial\{x_i x_j\} = \{x_j\} - \{x_i\}$$

which extends linearly to a sparse operator taking 1-chains to 0-chains. We represent $k$-forms as row vectors of length $S^{k+1}$ and define the discrete exterior derivative $d = \partial^t$ to be the transpose of the discrete boundary operator. Note that

$$d\{x_i\} = \sum_{j \neq i} \{x_j x_i\} - \{x_i x_j\}.$$

The operators $\partial$ and $d$ are purely topological in nature, however to construct the Laplacian we will need a Riemannian Metric. Again, following [6] we define the Riemannian Metric through the Hodge star and the equation

$$\left\langle \gamma^k, \beta^k \right\rangle = \gamma^T \star_k \beta$$

but instead of the Hodge star in [6] we set

$$\star_0\{x_i\} = \frac{1}{p_{\epsilon,\alpha}(x_i)}\{x_i\} \tag{2.10}$$

$$\star_1\{x_ix_j\} = K_{\epsilon,\alpha}(x_i,x_j)\{x_ix_j\} \tag{2.11}$$

using $p_{\epsilon,\alpha}$ and $K_{\epsilon,\alpha}$ from equation (2.6). This allows us to define the codifferential $\delta$ as the adjoint of the exterior derivative $d$ under the Riemannian Metric. Thus $\delta$ is defined through the formula

$$\begin{aligned} \delta\{x_ix_j\} &= -\star_0 d^T \star_1 \{x_ix_j\} = -\star_0 \partial \star_1 \{x_ix_j\} \qquad (2.12) \\ &= -K_{\epsilon,\alpha}(x_i,x_j)\left(\frac{\{x_j\}}{p_{\epsilon,\alpha}(x_j)} - \frac{\{x_i\}}{p_{\epsilon,\alpha}(x_i)}\right). \end{aligned}$$

So finally we define the discrete Laplacian on 0-forms by $\Delta_0 = (\delta + d)^2 = \delta d$ and a simple computation confirms that

$$\Delta_0 = 2(T - I) = -2\epsilon\overline{\Delta}$$

which differs from the Diffusion Maps discrete Laplacian by a constant multiple (including the well known sign difference). This confirms that for each $\alpha$ the operator given by $\lim_{\epsilon\to 0}\frac{I-F_{\epsilon,\alpha}}{\epsilon}$ is the Laplacian with respect the Riemannian metric defined by equation (2.10).

To make the connection to the Riemannian metric explicit, note that each tangent space is represented by discrete 1-chains. The Riemannian metric, which is locally an inner product, is related to the Hodge star, $\star_1$, simply by choosing local bases of 1-chains. For each vertex $\{x_l\}$ we can form $n$ 1-chains $\{\gamma_{ls}\}_{s=1}^n$ such that the Euclidean vectors $\overline{\gamma_{ls}} = \sum_k \gamma_{ls}(k)(x_k - x_l)$ are linearly independent. Thus $\{\gamma_{ls}\}$ define local coordinates on the tangent space at $\{x_l\}$ and the metric is given by

$$g_{ij}(x_l) = \langle \gamma_{li}, \gamma_{lj} \rangle = \gamma_{li}^T \star_1 \gamma_{lj} = \sum_k \gamma_{li}(k)\gamma_{lj}(k)K_{\epsilon,\alpha}(x_l, x_k).$$

The connection between Diffusion Maps and the geometric construction provides valuable insights into both methods. The geometric formulation reveals that the choice of $\alpha$ in Diffusion Maps is actually a choice of Riemannian metric, this will allow us to better understand the effect of time-delay coordinates on Diffusion Maps. In particular this reveals that when $\alpha = 1$ the Riemannian metric is chosen to be that inherited from the ambient space (since the input to Diffusion Maps is always given as an embedding in an ambient Euclidean space). Moreover, the results of Diffusion Maps in [2] show that a normalization by $\frac{1}{2\epsilon}$ will be required for the discrete Laplacian $\Delta_0$ to converge to the correct operator in the limit as $\epsilon \to 0$.

The key to diffusion maps is the construction of a discrete approximation to the Laplace-Beltrami operator $\Delta$ using the data set. However, the Laplace-Beltrami operator is simply the zeroth order Laplacian $\Delta = \Delta_0$, which operates on functions. Topological features in the de Rham cohomology can be inferred from the higher order Laplacians, $\Delta_k$ which operate on $k$-forms. In fact, the harmonic $k$-forms (solutions of $\Delta_k \psi = 0$) exactly pick out the various equivalence classes of $k$-forms. So, for example, on the torus there are exactly two solutions to $\Delta_1 \psi = 0$ and each solution corresponds to an intrinsic variable on the torus. This is a powerful generalization of the fact, pointed out in [2], that the harmonic forms found in diffusion maps correspond to the connected components of the data set (since the connected components are the equivalence classes of the zeroth order de Rham cohomology).

### 2.5.2 The Vietoris-Rips complex and Laplacians on $k$-forms

In the previous section we showed that the downside of the Discrete Exterior Calculus formalism was that it required a consistent simplicial complex in addition to the raw data points. Simplicial complexes are not available for many data sets; moreover they can be quite costly to construct and may bias the discrete geometry. Diffusion maps avoids this difficulty by focusing on the limiting continuous geometry however, this means that, a priori, diffusion maps may not give a consistent discrete geometry. Of course, the previous section showed that the equations for the discrete Laplacian (which completely determines the geometry) are equivalent. This might suggest that a diffusion map implicitly defines a simplicial complex. We will see that this is not the case, since the implicitly defined cells of a diffusion map do not necessarily satisfy the intersection requirements of a simplicial complex. Instead, we show that the diffusion maps construction is consistent with an abstract simplicial complex known as the Vietoris-Rips (VR) complex. In this section we will explore this implicit cell complex and show how to construct higher order Laplacians in this context.

We can understand the VR complex implicit to the diffusion maps construction by looking at the Hodge star operator defined in the previous section. Since the Hodge star on 1-forms assigns a non-zero value to every pair of data points, every single possible 1-simplex must be included in the complex. Each of these simplices are given a weight which decays exponentially in their length. Of course, this means that many of these simplices are given negligible weight by the Hodge star. Thus we can economize, and form a sparse Hodge star operator, by removing all simplices with length greater than $\epsilon$. In [5] it was shown that the sparse construction still converges to the Laplace-Beltrami operator in the limit of large data. We cannot expect that all of these simplices will satisfy the requirements of a simplicial complex (in particular they may intersect at points which are not in the data set; this will be very likely for high dimensional embeddings or for noisy data sets). However, by taking the collection of all simplices in a ball of radius $\epsilon$ we form the VR complex which is an abstract simplicial complex.

The connection to the VR complex is particularly interesting because of a recent paper [25]. The authors show that for a closed Riemannian manifold $\mathcal{M}$, and any metric space $\mathcal{N}$ sufficiently close to $\mathcal{M}$ (as measured by the Gromov-Hausdorff distance) the VR complex of

$\mathcal{N}$ is homotopy equivalent to $\mathcal{M}$. Our non-parametric model assumes that the data set $\mathcal{N}$ is sampled from an embedded manifold $\mathcal{M}$ which is compact and hence closed. The results of [25] imply that if our discrete samples are sufficiently dense and near the embedded manifold, then the VR complex which is implicit to the diffusion maps construction is homotopy equivalent to the manifold $\mathcal{M}$.

Now that we understand that the Vietoris-Rips complex is implicit to the diffusion maps construction of the Laplace-Beltrami operator, we can use this fact to perform an analogous construction of higher order Laplacian operators. In fact, constructing the higher order Laplacians is natural in the context of the Discrete Exterior Calculus. Following [6] the boundary $\partial^k$ and exterior derivative $d^k$ operators can be easily extended to $k$-forms. We now generalize the Hodge star construction of diffusion maps to $k$-simplices by setting

$$\star_k\{x_{i_1}x_{i_2}\cdots x_{i_{k+1}}\} \quad = \quad K_\epsilon(x_{i_1}, x_{i_2}, ..., x_{i_{k+1}})\{x_{i_1}x_{i_2}\cdots x_{i_{k+1}}\}. \tag{2.13}$$

Since each 1-simplex was weighted by a function exponentially decaying in length, we generalize to $k$-simplices by a function exponentially decaying in volume. Explicitly, the $k$-simplex formed by the vectors $\{x_{i_1}x_{i_2}\cdots x_{i_{k+1}}\}$ is part of a parallelotope defined by the columns of the matrix $X = \left[x_{i_2} - x_{i_1}, x_{i_3} - x_{i_1}, ..., x_{i_{k+1}} - x_{i_1}\right]$. The volume of the parallelotope is given by the Grammian determinant $\sqrt{\det(X^T X)}$. The volume of the $k$-simplex is given by

$$\mathrm{vol}(\{x_{i_1}x_{i_2}\cdots x_{i_{k+1}}\}) = \frac{\sqrt{\det(X^T X)}}{k!}$$

so we define the kernel function $K$ as

$$K_\epsilon(x_{i_1}, x_{i_2}, ..., x_{i_{k+1}}) = \exp\left(-\frac{1}{\epsilon}\mathrm{vol}(\{x_{i_1}x_{i_2}\cdots x_{i_{k+1}}\})^2\right) = \exp\left(-\frac{\det(X^T X)}{k!^2\epsilon}\right).$$

Now that we have defined the higher order Hodge star operators, we can construct the discrete codifferential $\delta$ on $k$-forms as $\delta^k = (-1)^k \star_{k-1}^{-1} \partial^k \star_k$ and the Laplacian on $k$-forms

$$\Delta_k = \delta^{k+1} d^k + d^{k-1} \delta^k$$

following the definitions of [6].

### 2.5.3 Computing the de Rham cohomology and representatives

The construction of a discrete Laplacian on $k$-forms is particularly interesting because it allows us to compute the de Rham cohomology $H_k(\mathcal{M}) \cong \text{Kernel}(\Delta_k)$. Moreover, each $k$-form in the kernel of $\Delta_k$ corresponds to a unique cohomology class. Of course, really we are finding the cohomology groups of the Vietoris-Rips complex of the metric space given by our discrete samples, however, as shown in [25] this will be the same as the homology of $\mathcal{M}$ for dense sampling sufficiently near $\mathcal{M}$. In the following examples we carry out the construction of the Laplacian on 1-forms for a torus and a genus-2 torus. We show that we get the correct Betti number and show how to approximate closed representatives of each cohomology class.

**Example 2.5.1. Eigenforms on a Torus** In this example we carried out the above construction for $\Delta_1$ using 8000 points generated on a torus embedded in $\mathbb{R}^3$ as shown in Figure 2.4. Since $\Delta_1$ is a discrete approximation to the Laplacian on 1-forms, the eigenvectors of this matrix give discrete 1-forms which assign a scalar value to each vector in the tangent

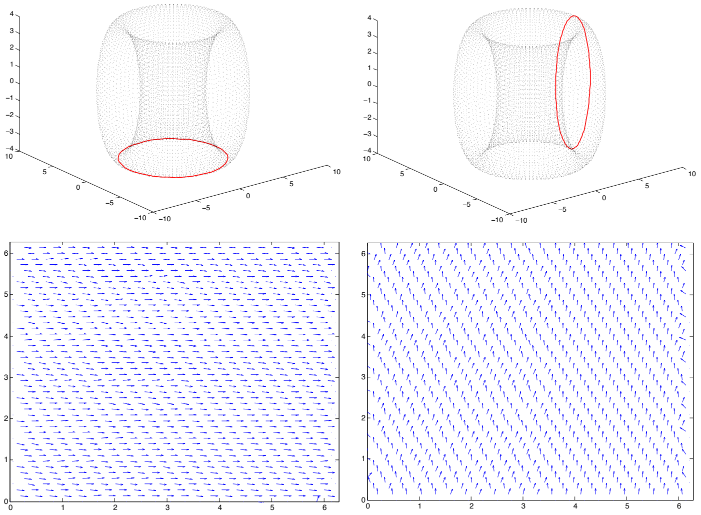


Figure 2.4: Representatives of the de Rham cohomology classes for a torus. Corresponding vector fields in the kernel of the Laplacian on 1-forms shown in the latent coordinates $[0, 2\pi]^2$.

bundle. We computed the 8 smallest eigenvalues to be

$$\begin{bmatrix} -7.75 \times 10^{-17} \\ -8.3 \times 10^{-18} \\ 2.260 \times 10^{-4} \\ 2.261 \times 10^{-4} \\ 4.081 \times 10^{-4} \\ 6.056 \times 10^{-4} \\ 6.190 \times 10^{-4} \\ 6.191 \times 10^{-4} \end{bmatrix}.$$

Since these eigenvalues are computed numerically with double precision, we recognize the first two eigenvalues as machine zero. Since the kernel of $\Delta_1$ is isomorphic to the de Rham cohomology $H_1(\mathcal{M})$ this computation correctly finds the first Betti number to be $b_1 = \dim(\text{kernel}(\Delta_1)) = 2$.

To get a closer look at the eigenforms, we convert a 1-form into a vector field by averaging

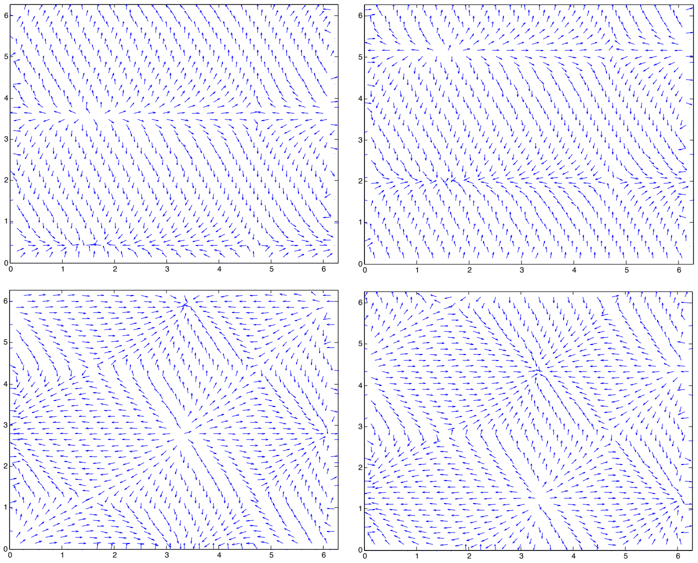


Figure 2.5: Eigenforms of the Laplacian on 1-forms for a torus shown in the latent coordinates $[0, 2\pi]^2$. The corresponding eigenvalues are $2.260 \times 10^{-4}$, $2.261 \times 10^{-4}$, $6.190 \times 10^{-4}$, and $6.191 \times 10^{-4}$

all of the 1-simplices (which are vectors in the ambient space) according to their weight assigned by the 1-form. In Figure 2.4 we show the two vector fields in the kernel of $\Delta_1$. For clarity we draw these vector fields on a square which represents the latent coordinates for the torus (this causes some distortion near the edges which are supposed to be identified). Finally, in order to generate closed representatives of the cohomology classes, we chose a random point on the torus and generated a trajectory by following the vector field given by a 1-form in the kernel of $\Delta_1$ (we discarded any initial transient to find a closed orbit). As shown in Figure 2.4, these closed representatives accurately depict the two cohomology classes. We used $\epsilon = 0.25$ to create Figure 2.4, however the shape of the vector fields and the Betti number of $b_1 = 2$ was robust to a large range of $\epsilon$ values from 0.02 up to 1.

Since eigenforms are quite difficult to compute analytically, in Figure 2.5 we plot the vector fields associated to the eigenforms associated to the eigenvalues listed in the caption.

**Example 2.5.2. Eigenforms on a Genus-2 Surface** With the above machinery in place we repeated the above computations for a dataset consisting of 8000 points sampled from

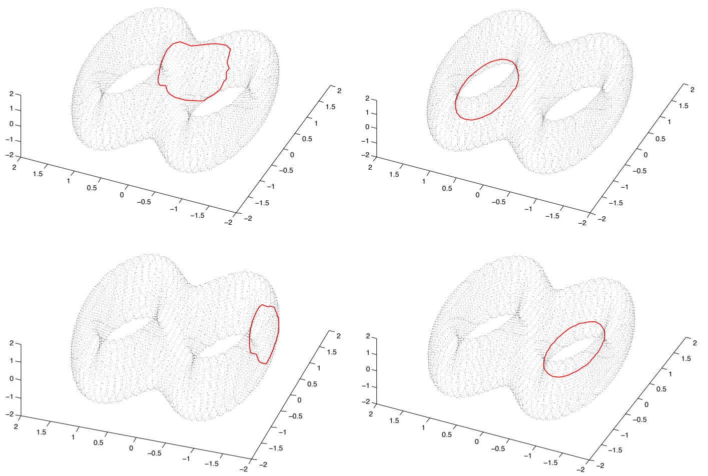


Figure 2.6: Representatives of the de Rham cohomology classes for a genus 2 surface.

a genus-2 surface. In this case the first 8 eigenvalues of $\Delta_1$ were

$$\begin{bmatrix} -2.18 \times 10^{-17} \\ -9.6 \times 10^{-18} \\ -1.5 \times 10^{-18} \\ 8.40 \times 10^{-17} \\ 1.812 \times 10^{-4} \\ 3.298 \times 10^{-4} \\ 6.101 \times 10^{-4} \\ 6.626 \times 10^{-4} \end{bmatrix}.$$

which correctly identifies the first Betti number of the genus-2 surface as $b_1 = 4$. In Figure 2.6 we draw closed representatives of the cohomology classes, each coming from one of the 1-forms in the kernel of $\Delta_1$.

## 2.6 Shift-local kernels and drift-diffusion processes

In the Section 2.4 we showed that local isometric kernels approximate the geometry induced by the embedding. By removing the isometric requirement, we showed that local kernels in general give rise to a Riemannian metric which is determined by the Hessian of the kernel. In this section we speculate as to a further generalization of local kernels called *shift-local* kernels and conjecture that these kernels define Finslerian geometries (see for example [26]).

Let $K(x,y) = \overline{K}(x, y-x) = \overline{K}(x,z)$ be a kernel given by $\overline{K}(x,z) = e^{-z^T A_x z + 2b_x^T z}$ so that the scaled kernel is given by

$$\overline{K}\left(x, \frac{z}{\sqrt{\epsilon}}\right) = \exp\left\{-\frac{z^T A_x z}{\epsilon} + \frac{2b_x^T z}{\sqrt{\epsilon}}\right\}.$$

Setting $z_s = \sqrt{\epsilon} A_x^{-1} b_x$ we can complete the square to find

$$\overline{K}\left(x, \frac{z}{\sqrt{\epsilon}}\right) = \exp\left\{-\frac{(z-\sqrt{\epsilon}A^{-1}b)^T A(z-\sqrt{\epsilon}A^{-1}b)}{\epsilon} + b^T A^{-1} b\right\} = e^{-\frac{(z-z_s)^T A(z-z_s)}{\epsilon}} e^{b^T A^{-1} b}$$

Since the kernel is no longer centered at the base point $x$ these kernels are no longer local. Moreover, the new maximum is at $z_s$ which has an $\epsilon$ dependence. Motivated by this example, we introduce a generalization of local kernels which we call shift-local kernels since they no longer satisfy the centered property of a local kernel. Instead, the center of the kernel will be shifted by a vector whose length is order $\mathcal{O}(\sqrt{\epsilon})$ so intuitively shift-local kernels are only centered in the limit as $\epsilon \to 0$.

**Definition 2.6.1** (Shift-local kernel)**.** A kernel $K : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is called *shift-local* if it can be written in the form $K(x,y) = \overline{K}(x, y-x)$ where $\overline{K}(x,z)$ is smooth and nonzero in a neighborhood of $z = 0$ with the properties

1. (*fast decay*) For some $c, \sigma > 0$ and all $x, z \in \mathbb{R}^n$ we have $0 \le \overline{K}(x,z) \le ce^{-\sigma ||z||^2}$.

2. (*Finslerian*) For $b_x = \sqrt{\epsilon} D_z \overline{K}(x,0)$ and $K_x = -\epsilon H_z \overline{K}(x,0)$ we have $b_x^T K_x^{-1} b_x < 1$.

We conjecture that every shift-local kernel defines a Finslerian geometry. Moreover, we hypothesize that shift-local kernels can approximate certain elliptic operators such as the

Fokker-Planck operator, which are important to the study of dynamical systems. In the next section we present a numerical experiment which motivates these conjectures.

### 2.6.1 Ulam's method and approximation of elliptic operators

The results in [2] show that for a data set generated by a stochastic differential equation of the form

$$dX_i = \nabla U \ dt + 1 \ dW_i \tag{2.14}$$

one can numerically construct a discrete operator which (in the limit of large data) converges to the evolution operator that solves a Fokker-Planck equation. In short, the diffusion maps approach uses a discrete set of observations, $\{X(t_k)\}_{k=1}^{N}$, to find a kernel density estimate of the operator $\mathcal{O}$ and the solution semigroup. Since the construction of the discrete operator is based on distances between the observed points and not their coordinates, the method is extremely robust to nuisance variables and redundant variables. With this construction, given any distribution which is localized near the observed data, one can push forward this density to find the future distribution. Note that while the dynamics are assumed to follow the gradient of a potential, $\nabla U$ is not known and is implicitly recovered by the construction. Of course, equation (2.14) is very restrictive. In this section we explore the conjecture of the previous section in an attempt to approximate a Fokker-Planck evolution. The most general setting in which the evolution is governed by the classical Fokker-Plank equation is

$$dX_i = f(X) \ dt + \sum_j h_{ij}(X) \ dW_j, \tag{2.15}$$

that is, where the gradient dynamics is generalized to having an arbitrary drift term. The properties of the solution semigroup in this case are well established, see for example [27], so the difficulty is finding a data-driven construction to recover the operator when $f$ and $h_{ij}$ are unknown.

We conjecture that approximating the forward operator of a stochastic dynamical system will require a shift-local kernel, and since the vector field $f(x)$ is not known, we must

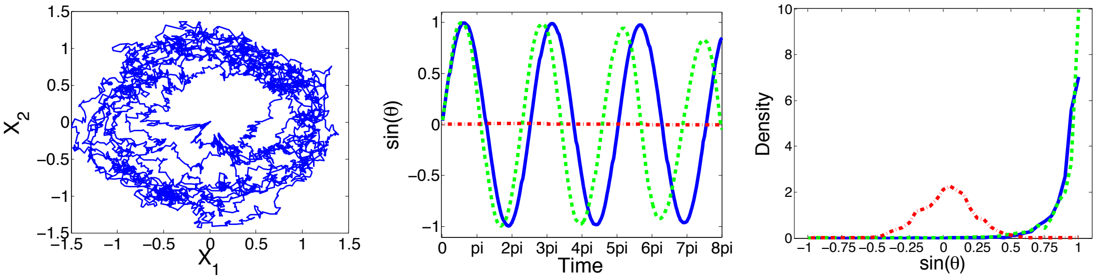


Figure 2.7: (a) 5000 samples from simulation of equation (2.16); this is the only input to both the diffusion map algorithm and the shift-local kernel. (b) Angle of the first moment of the Monte Carlo simulation (blue, solid), $K_\epsilon^*$ (green, dashed), and diffusion maps (red, dot-dashed). (c) Distribution of $\sin\theta$ at time $t = 2\pi/3$ (same coloring as middle).

take a data driven approach. In this section we approach this problem by using the time orientation of the data and incorporating the known time-scaling for the Brownian noise. Thus we introduce the shift local kernel $K_\epsilon^*(x,y) = h_\epsilon(|x - \phi_{\sqrt{\epsilon}}(y)|)$, where $\phi_{\sqrt{\epsilon}}$ evolves the initial condition forward in time by $\sqrt{\epsilon}$. This information is readily available in the data by simple taking $\phi_{\sqrt{\epsilon}}(x_i) = x_{i+N}$ where $N$ is the number of steps corresponding to an elapsed time of $\sqrt{\epsilon}$. We conjecture that this procedure recovers the correct Fokker-Planck solution semigroup and that the kernel $K_\epsilon^*$ is a discrete Finsler function for our manifold. Our next example gives numerical evidence which seems to support these conjectures.

**Example 2.6.1.** In this numerical example we show that an operator approximated with a shift-local kernel is able to track the first moment of a density which evolves according to (2.15), whereas the baseline diffusion maps method cannot. Define $X(t) \in \mathbb{R}^2$ in polar coordinates as the solution of

$$\begin{aligned} dr &= r(1-r)\; dt + \frac{1}{3}\; dW_1 \qquad (2.16) \\ d\theta &= 1\; dt + \frac{1}{3}\; dW_2 \end{aligned}$$

This simple system clearly does not follow the gradient of a potential. We first simulate this system to generate 5000 data points. This data is used by both diffusion maps operator

$K_{\epsilon,1} \cong e^{\epsilon\Delta}$ and the shift local kernel $K_{\epsilon}^{*}$ to predict the evolution of a distribution which is initially a delta function at $(r,\theta) = (1,0)$. Note that these data points are the only input to the two algorithms, and neither knows anything about the analytic form of the system. We can then compare the results to the true evolution, which is estimated by repeatedly simulating (2.16) with the same initial condition (giving a Monte Carlo estimate of the true density). In Figure 2 we compare the evolution of the first moment of the true density (estimated by Monte Carlo) to those of the two algorithms. The diffusion maps operator loses track of the first moment immediately, because it attempts to interpret the data as following the gradient of a potential, whereas $K_{\epsilon^*}$ gives a reasonable approximation of the first two moments.

# Chapter 3: Leveraging temporal structure

In this chapter we consider data which has an a priori temporal structure and demonstrate how this structure should inform a nonparametric analysis. In particular, we consider data generated by a dynamical system such that there is a natural notion of intrinsic geometry given by the Lyapunov metric. The key result is Theorem 3.2.1 which shows that time delay embeddings bias the geometry in favor of the most stable component of such a dynamical system. This bias can be beneficial or detrimental depending on which aspects of the dynamics are of interest, so we provide a preliminary method of reinforcing or ameliorating this effect via a simple weighting scheme for the delay embedding. A more far reaching program based on the theory of local kernels is outlined in the discussion.

We also demonstrate that the geometric approach to nonparametric modeling naturally generalizes linear techniques and improves on previous approaches to state space reconstruction. Moreover, we show that the temporal structure allows us to interpret the nonparametric analysis as a generalization of the Fourier basis which is adapted to the dynamics and leads to a kind of time-scale separation for certain systems. As this is first and most important case study of integrating prior structure with a nonparametric analysis, we give many examples which demonstrate the importance of the geometric perspective and the intrinsic/extrinsic dichotomy.

## 3.1 Overview

The method of time-delay embedding was first introduced by Takens, Ruelle and others for the purpose of reconstructing dynamical attractors from data [28–31]. With a series of generic delayed observations, it was shown that topological properties are preserved in reconstruction dimensions greater than $2d$, for a smooth attractor of dimension $d$ [28], and for an attractor of fractal dimension $d$ [32]. Considerable effort followed to develop methods of estimating attractor dimension, in order to carry out embeddings in a Euclidean space of minimal dimension.

More recently, the development of pervasive and cheap sensors has caused a shift in emphasis toward methods capable of handling large quantities of time-ordered data. For example, we could imagine a complex system with a relatively low-dimensional attractor, possibly chaotic, where the observations are represented by a high-dimensional multivariate time series. In such a case, it is of great interest to apply a data analysis technique to a video of an experiment, for instance, and to seek ways to selectively project that data onto various dynamical time scales of interest.

Dealing with multivariate recordings in high dimensions requires some form of dimensionality reduction. In 1985, Broomhead and King [33] combined the ideas of time-delay embedding and singular value decomposition (SVD) to project high-dimensional reconstructions to lower dimension for analysis. This idea is in the tradition of Karhunen-Loeve decompositions of dynamical systems. Local versions of the same idea were used later to develop algorithms for time series prediction [34] and noise reduction. One purpose of this chapter is to provide a theoretical explanation for the power of time delays when combined with dimensionality reduction, as demonstrated in the following simplistic scenario.

**Example 3.1.1.** Consider the task of analyzing a video which contains a single black pixel which moves periodically in a circle, surrounded by grayscale static. The input to our method is the video, a set of time-ordered frames, each a two-dimensional array of pixel values. The deterministic dynamics of the underlying dynamics can be represented by a single variable: the angular position of the black pixel. However, such a representation of our data would contain only a tiny fraction of the variance of the data set (which is dominated by noise) and the reconstruction error would be large. Directly applying a standard dimensionality reduction technique like the SVD, which is greedy for variance, would ignore the black dot and instead focus on learning the moments and correlations in the noise. On the other hand, if delay coordinates are used followed by a projection by

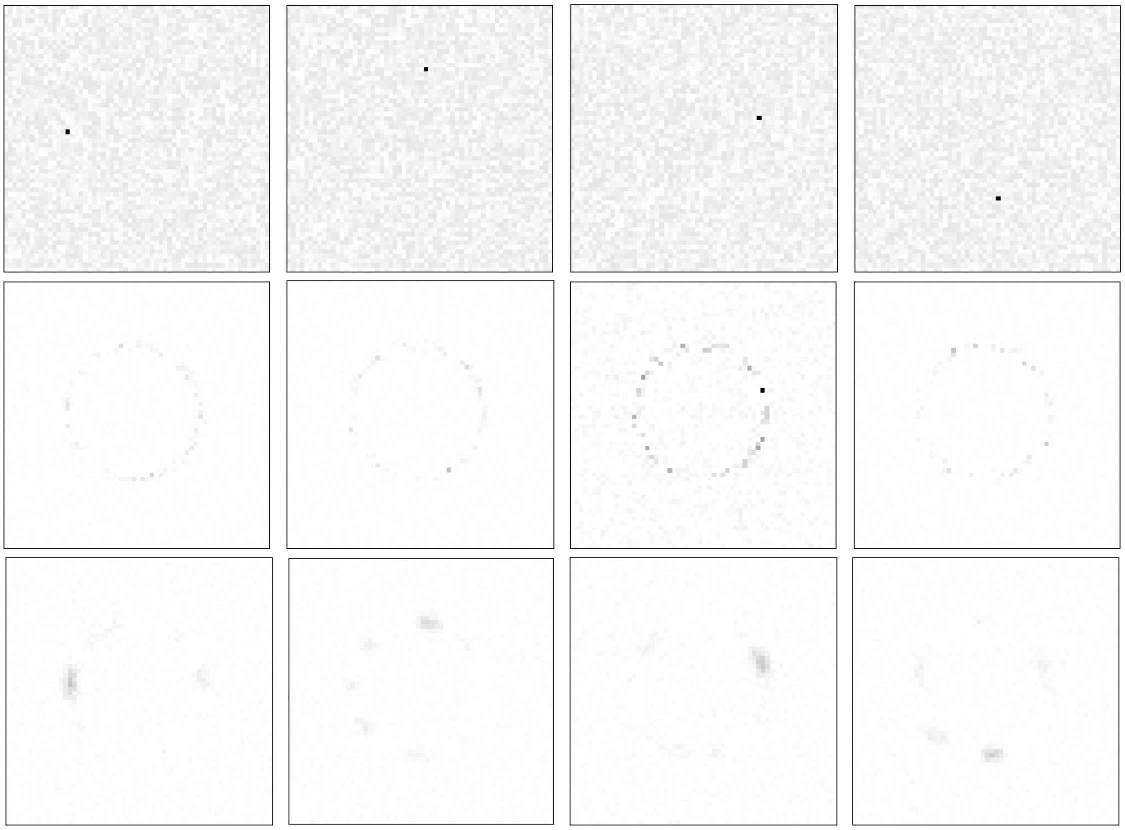
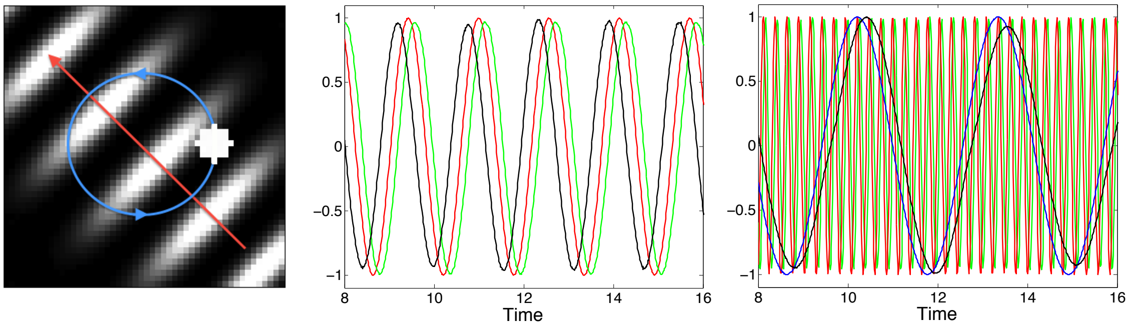


Figure 3.1: Top row: Four sample images from a video of the black pixel moving on a noisy circle; Middle row: Reconstruction of the four sample images using first 32 SVD modes; Bottom row: Reconstruction with SVD applied to time-delay embedding, using the first 32 SVD modes.

SVD into the dominant directions, the desired dynamics are isolated from the noise. In Figure 3.1, we show the results of applying the SVD to the images with and without delays. This example shows a key feature of delay coordinates: they project the data onto the most stable dynamical variables. We will describe this feature more quantitatively in Section 3.2.

Although useful in Example 3.1.1, a severe limitation of SVD is that it is a linear projection. Generic linear projections of complicated sets retain their topological characteristics, if the image dimension is large enough, but fine details of the geometry are in general lost. More refined dimensionality reduction techniques do less damage to the geometry. In particular, diffusion maps, developed by Coifman and his collaborators [2, 12–14], are able to partially retain geometric properties of attractor manifolds under projection from high dimensions. In a second example, we explore replacing the SVD projection by a diffusion map.

**Example 3.1.2.** In this example we analyze a video with two periodically moving artificial objects. The largest variance component of the video is made up of white stripes that move diagonally across the image as shown in Figure 3.2(a). A smaller variance component is a localized "defect" which moves periodically on a circle. As in Example 1.1, we consider the input "data" to be the unprocessed video itself. In Figure  3.2, we consider two versions of the video, where the stripes move slower (Figure 3.2(b)), respectively faster (Figure 3.2(c)) than the defect. The projection using SVD with time delays (green trace) picks out the higher variance component (the stripes) in both versions. On the other hand, pairing the delay-coordinate embedding with a diffusion map projection (black trace) projects to the slower mode in both versions: the stripes in Figure 3.2(b) and the defect in Figure 3.2(c).



(a) (b) (c)

Figure 3.2: (a) A frame from the video with arrows added to indicate the dynamics; the stripes follow the red arrow and the defect moves along the blue circle. (b) When the stripes move more slowly than the defect, both SVD (green trace) and DMDC (black trace) reconstruct the phase of the slow moving stripes (red). (c) When the stripes move more quickly, DMDC (black) finds the phase of the defect (blue) on its circular trajectory, however SVD (green) still finds the phase of the stripes because they have higher variance.

The goal of this chapter is to explain the mechanisms behind the above examples, namely (1) the contribution of delay coordinates to reconstructing intrinsic attractor geometry and (2) the use of diffusion maps to control the metric properties of the reconstructed attractor.

We will use this knowledge to build a flexible computational framework called Diffusion-Mapped Delay Coordinates (DMDC) that can be used to analyze high-dimensional data such as videos for low-dimensional dynamics. In particular, we are interested in separating time scales in the data, by projecting onto the stablest dynamical directions (using weighted delay coordinates) and further slicing the dynamics there into Fourier-like components (using diffusion maps).

Nonlinear dimensionality reduction has been successfully applied to very high dimensional observations such as images [21, 22]. However, these methods are typically designed for data that have no dynamics. Dynamical data, meaning data with a time ordering, is a unique problem with different goals than generic dimensionality reduction. In our view, the key to success of nonlinear dimensionality reduction techniques is the imposition of appropriate assumptions on the structure of the data. Whereas the goal of generic dimensionality reduction is typically to minimize the reconstruction error, or to maximize the variance of the low dimensional representation, such a representation *may not contain dynamically interesting information.*

To adapt nonlinear dimensionality reduction for use on data measured from dynamical systems, we find it important to separate the extrinsic features of the data, such as the observation and embedding space, from the intrinsic dynamical features, such as Lyapunov exponents and associated invariant manifolds. These intrinsic dynamical features are independent of the observation space and can be represented by an intrinsic geometry on the state space, given by the Lyapunov metric, which is a Riemannian metric that characterizes the Oseledets splitting [35]. In Section 3.2, we show that the classical technique of time-delay embeddings, with appropriate weights, effectively recovers the Lyapunov metric in the most stable Lyapunov direction.

Within the most stable direction there may still be multiple time scales. A periodic or quasi-periodic trajectory such as Example 1.2 may have a zero Lyapunov exponent along the trajectory that harbors a fast mode and a slower mode. For other examples, see Sections 3.4 and 3.5 below. We separate time scales within the most stable Lyapunov direction using the reconstructed eigenfunctions of the Laplace operator with respect to the Lyapunov metric. The Laplace operator is approximated by a diffusion map [2], as we discuss in Section 3.3.

Previous work on time-scale separation from dynamical data includes the work [14] on stochastic systems governed by Brownian motion in a potential field. Ideally, time-scale separation finds a hierarchy of slow variables which are independent of the higher frequency oscillations of the dynamics. For example, the evolution of a dynamical system is independent of the fast oscillations of observation noise, as in Example 3.1.1. Moreover, if the system contains a slow manifold, then the evolution on the slow manifold will be independent of the faster dynamics. Time-scale separation involves finding the variables that are governed by these slow dynamics and ordering the variables according to the relevant time-scale. Our goals of projecting onto slow dynamics have some overlap to those of [14, 17, 36, 37], although we assume no knowledge of the equations generating the data, or availability of surrogates such as microscopic models or legacy solvers. Furthermore, in analyzing video data of experiments, we are reduced to assuming that the observations are related to the true system state in an unknown way.

In Section 3.2 below, we show that for data which is not necessarily observed in its latent state space, the standard technique of time-delay embeddings will partially recover the Lyapunov metric. This improves significantly on the results of Takens, which show that time-delay coordinates reconstruct the topology of the latent state space. This will allow the diffusion map to be applied to generic observations of a dynamical system.

In Section 3.3 we explain how diffusion maps can be used for dimensionality reduction and for time-series analysis. As a method for dimensionality reduction, we show how a diffusion map can be interpreted as a kernel Principal Component Analysis, resulting in an optimal low-dimensional embedding of a data set with respect to a conformal transformation of the geometry. In fact, the diffusion map constructs an operator $\mathcal{L}$ which is the Laplacian operator with respect to the geometry introduced by the conformal transformation. Moreover, when the data has a time ordering, the diffusion map coordinates can also be interpreted as time-series give by the projection of the dynamics onto eigenfunctions of $\mathcal{L}$. We show that for a dynamical system with an invariant measure, the conformal transformation will adapt the operator $\mathcal{L}$ to this invariant measure.

In Section 3.4 we show how projecting the dynamics onto the eigenfunctions of the operator $\mathcal{L}$ can lead to time scale separation. If the dynamics are elliptic when restricted to the most stable Lyapunov direction, then the operator $\mathcal{L}$ can be written as a Laplacian plus lower order terms (with respect to a special Riemannian metric which depends on $\mathcal{L}$). In many cases the most stable Lyapunov direction will be one-dimensional, and the Laplacian will be unique up to a choice of measure. Then the eigenfunctions will simply be spatial sine and cosine functions which are rescaled to the invariant measure on the stable manifold. Projecting the dynamics onto this adapted Fourier basis on the manifold will regularize the lower order terms which is analogous to the smoothing of a low-pass filter. We interpret the projection onto each eigenfunction as a solution of a reduced dynamical system with a specific time scale which is related to the corresponding eigenvalue.

In Section 3.5 we implement the proposed new technique and apply it to the high dimensional observations of spatiotemporal dynamics. We apply DMDC to a computation model of a meandering spiral wave. We find that even for these complex dynamics, the method produces a time-scale separation, in addition to producing decorrelated latent variables and achieving significant noise reduction.

The appendix includes a self-contained description of the DMDC algorithm and implementation details.

## 3.2 Geometry of time-delay coordinates

The theory of time-delay coordinates as introduced by Takens [28] shows that by appending delayed values of a generic observation of a dynamical system, one achieves a diffeomorphic copy of the attractor in some Euclidean space. The fact that the embedding is given by a diffeomorphism of the attractor shows that the time-delay embedding is topology-preserving, although crucially it introduces a new geometry to the data set. In this section, we will see that the new geometry is related to the Lyapunov metric [38] restricted to the most stable Oseledets subspace. In the past, the increased dimension of the ambient Euclidean space was a limitation for time-delay embeddings, and significant effort was made to look for parsimonious time-delay embeddings using as few delays as possible. However, as we will see Section 3.3, diffusion maps are not adversely affected by the ambient dimension, which will allow us to take hundreds or even thousands of time delays and computationally study the effect of the delay embedding on the geometry of the attractor.

The context for this section will be the natural setting of the Multiplicative Ergodic Theorem (MET) on a smooth manifold. Let $\mathcal{M}$ be an $n$-dimensional smooth compact Riemannian manifold which is the attractor of a system denoted $\dot{x} = f(x)$, with invariant measure $\mu$ for the induced flow $F_t$. To accommodate discrete observations of the dynamics we will consider the flow $F_\tau$ for a fixed time step $\tau > 0$. According to Oseledets [35], there exist real numbers $\sigma_1 < \ldots < \sigma_k$, with $k \leq n$, such that for $\mu$-almost every $x$ there is a splitting $T_x\mathcal{M} = \bigoplus_{i=1}^k E_i(x)$, where $\dim E_i = d_i$, and where $d_1 + \ldots + d_k = n$. Each Oseledets space $E_i(x)$ is invariant under the dynamics, meaning any nonzero vector $u_i \in E_i(x)$ has image $DF_{-j\tau}(x)u_i \in E_i(F_{-j\tau}(x))$. Moreover, for any $u_{i,x} \in E_i(x)$,

$$\begin{aligned} \lim_{j\to\infty} \frac{1}{j} \ln ||DF_{j\tau}(x)u_i|| &= \sigma_i, \\ \lim_{j\to-\infty} \frac{1}{j} \ln ||DF_{j\tau}(x)u_i|| &= -\sigma_i. \end{aligned}$$

Assume a multivariate observation of dimension $r$, given by a smooth nonlinear $h \in C^\infty(\mathcal{M}, \mathbb{R}^r)$. For $\kappa, \tau > 0$ define the $\kappa$-weighted *delay coordinate map* $H : \mathcal{M} \to \mathbb{R}^{r(s+1)}$ by

$$H(x) = [h(x), e^{-\kappa}h(F_{-\tau}(x)), e^{-2\kappa}h(F_{-2\tau}(x)), \ldots, e^{-s\kappa}h(F_{-s\tau}(x))]^T. \tag{3.1}$$

Under weak technical assumptions [32], by choosing $r(s+1) \geq n$, the delay coordinate map is an immersion for a prevalent choice of $h$. Additionally, for $r(s+1) > 2n$, the delay coordinate map $H$ is an embedding of the manifold $\mathcal{M}$ into $\mathbb{R}^{r(s+1)}$. There is no reason to assume that this embedding preserves the metric on the manifold, and thus geometry of the embedded manifold $H(\mathcal{M})$ is an extrinsic factor from our point of view.

For $\epsilon > 0$, the *$\epsilon$-Lyapunov metric* $\langle u, v\rangle_\epsilon$ is defined by

$$\langle u_i, v_i\rangle_\epsilon = \sum_{j\in\mathbb{Z}} e^{-2(\sigma_i j+\epsilon|j|)}\langle DF_{j\tau}(x)u_i, DF_{j\tau}(x)v_i\rangle_{T_x\mathcal{M}}$$

for $u_i, v_i \in E_i(x)$. The Lyapunov metric is intrinsic to the dynamics because, when measured in this metric, the dynamics satisfy the uniform bounds

$$e^{-(\sigma_i+\epsilon)j}||u_i||_\epsilon \leq ||DF_{-j\tau}(x)u_i||_\epsilon \leq e^{-(\sigma_i-\epsilon)j}||u_i||_\epsilon.$$

Thus we will consider the Riemannian metric of interest on $\mathcal{M}$ to be the Lyapunov metric.

We now return to the extrinsic geometry and investigate the metric induced on the embedded manifold $H(\mathcal{M})$. Let $u = u_1 + \ldots + u_k, v = v_1 + \ldots + v_k \in T_x\mathcal{M}$, where $u_i, v_i \in E_i(x)$ and denote $\hat{u} = DH(u), \hat{v} = DH(v)$ in $T_{H(x)}H(\mathcal{M})$. Then the inner product $\langle\cdot,\cdot\rangle$ in the Euclidean reconstruction space $R^{r(s+1)}$ is

$$\begin{aligned}\langle\hat{u},\hat{v}\rangle &= \sum_{j=0}^{s} e^{-2j\kappa}\langle Dh(F_{-j\tau}(x))DF_{-j\tau}(x)u, Dh(F_{-j\tau}(x))DF_{-j\tau}(x)v\rangle_{\mathbb{R}^r}\\ &= \sum_{i=1}^{k}\sum_{j=0}^{s} e^{-2j\kappa}\langle Dh(F_{-j\tau}(x))DF_{-j\tau}(x)u_i, Dh(F_{-j\tau}(x))DF_{-j\tau}(x)v_i\rangle_{\mathbb{R}^r}.\end{aligned}$$

The main theorem of this chapter shows that with the right choice of $\kappa$, the metric $\langle\cdot,\cdot\rangle$ in the embedding space projects onto the most stable Oseledets subspace $E_1$.

**Theorem 3.2.1.** *Let $\mathcal{M}$ be a compact manifold, $u, v \in T_x\mathcal{M}$ and let $\hat{u} = DH(u)$ and $\hat{v} = DH(v)$ be the images under the time-delay embedding $H$ in* (3.1). *Let $u_i = \pi_i(u)$*

*be the projection onto the ith Oseledets space, and assume $u_1$ and $v_1$ are nonzero. Let $0 < \kappa < -\sigma_1$. Then for a prevalent choice of $h$ and for all $i \neq 1$,*

$$\lim_{s\to\infty} \frac{\langle \hat{u}_i, \hat{v}_i \rangle}{||\hat{u}||\ ||\hat{v}||} = 0 \qquad \textit{and therefore} \qquad \lim_{s\to\infty} \frac{\langle \hat{u}, \hat{v} \rangle - \langle \hat{u}_1, \hat{v}_1 \rangle}{||\hat{u}||\ ||\hat{v}||} = 0.$$

*Proof.* First note that

$$\begin{aligned} |\langle \hat{u}_i, \hat{v}_i \rangle| &= \left| \sum_{j=0}^{s} e^{-2j\kappa} \langle Dh(F_{-j\tau}(x))DF_{-j\tau}(x)u_i, Dh(F_{-j\tau}(x))DF_{-j\tau}(x)v_i \rangle_{\mathbb{R}^r} \right| \\ &\leq \sum_{j=0}^{s} e^{-2j\kappa} ||Dh(F_{-j\tau}(x))DF_{-j\tau}(x)u_i||_{\mathbb{R}^r} ||Dh(F_{-j\tau}(x))DF_{-j\tau}(x)v_i||_{\mathbb{R}^r} \\ &\leq h_{\max}^2 \sum_{j=0}^{s} e^{-2j\kappa} ||DF_{-j\tau}(x)u_i||_\epsilon ||DF_{-j\tau}(x)v_i||_\epsilon \\ &\leq h_{\max}^2 ||u_i||_\epsilon ||v_i||_\epsilon \sum_{j=0}^{s} e^{-2j(\sigma_i + \kappa - \epsilon)} \end{aligned}$$

for all $\epsilon > 0$, where $h_{\max}$ is the maximum of the matrix norm $||Dh||$ over the compact manifold $\mathcal{M}$. This bounds the growth rate in all Oseledets subspaces.

Next we need to show that the component of $\hat{u}$ in the most stable direction dominates as $s$ increases. Thus we will bound $||\hat{u}||$ from below and we will focus on the component $u_1$, which is assumed to be nonzero. Choose $k \geq 0$ such that $r(k+1) \geq n$; then the delay coordinate map is an immersion for a prevalent choice of $h$ [32]. From this we only need the fact that for all $x \in \mathcal{M}$ and all $j \geq 0$, the rank of the $r(k+1) \times n$ matrix

$$A_j(x) = \begin{bmatrix} e^{-\kappa j} Dh(F_{-j\tau}(x))DF_{-j\tau}(x) \\ \vdots \\ e^{-\kappa(k+j)} Dh(F_{-(k+j)\tau}(x))DF_{-(k+j)\tau}(x) \end{bmatrix}$$

is $n$, implying that the kernel of the matrix is zero. For any nonzero vector $u_1 \in E_1(x)$ the vector $A_j(x)u_1$ is nonzero, and for some $j \leq l \leq j+k$ the $r$-vector

$$e^{-\kappa l} Dh(F_{-l\tau}(x))DF_{-l\tau}(x)u_1 \neq 0.$$

Thus for each $x \in \mathcal{M}$ we have $\max_{j \leq l \leq j+k} ||Dh(F_{-l\tau}(x))e_1||_{\mathbb{R}^r} > 0$ for any unit vector $e_1 \in E_1(x)$. Since $\mathcal{M}$ is compact we can define

$$\begin{aligned} h_{\min} &\equiv \min_{x \in \mathcal{M}} \min_{j \geq 0} \max_{j \leq l \leq j+k} ||Dh(F_{-l\tau}(x))e_1||_{\mathbb{R}^r} \\ &= \min_{x \in \mathcal{M}, j \geq 0, y = F_{-j\tau}(x)} \max_{0 \leq l \leq k} ||Dh(F_{-l\tau}(y))e_1||_{\mathbb{R}^r} \\ &\geq \min_{x \in \mathcal{M}} \max_{0 \leq l \leq k} ||Dh(F_{-l\tau}(x))e_1||_{\mathbb{R}^r} > 0. \end{aligned}$$

This allows us to establish the lower bound

$$||A_j(x)u_1||_{\mathbb{R}^{r(k+1)}} \geq h_{\min} e^{-j(\kappa+\sigma_1+\epsilon)} ||u_1||_\epsilon$$

for all $x \in \mathcal{M}$. We can use this to obtain a lower bound on $||\hat{u}||$ by splitting the $s$ terms into $\lfloor s/k \rfloor$ (the greatest integer less than $s/k$) blocks of size $k$ so that

$$\begin{aligned} ||\hat{u}||^2 &\geq \sum_{j=0}^{s} e^{-2j\kappa} ||Dh(F_{-j\tau}(x))DF_{-j\tau}(x)u_1||^2_{\mathbb{R}^r} \\ &\geq \sum_{l=0}^{\lfloor s/k \rfloor} ||A_{lk}(x)u_1||^2 \\ &\geq h^2_{\min} ||u_1||^2_\epsilon e^{-2k\lfloor s/k \rfloor(\sigma_1+\kappa+\epsilon)} \\ &\geq h^2_{\min} ||u_1||^2_\epsilon e^{-2s(\sigma_1+\kappa+\epsilon)}. \end{aligned}$$

By combining the upper and lower bounds we have

$$\frac{|\langle \hat{u}_i, \hat{v}_i\rangle|}{||\hat{u}|| \cdot ||\hat{v}||} \leq \left(\frac{h_{\text{max}}^2}{h_{\text{min}}^2} \cdot \frac{||u_i||_\epsilon}{||u_1||_\epsilon} \cdot \frac{||v_i||_\epsilon}{||v_1||_\epsilon}\right) \frac{\left|1 - e^{-2(s+1)(\sigma_i+\kappa-\epsilon)}\right|}{e^{-2s(\sigma_1+\kappa+\epsilon)}} \to 0$$

as $s \to \infty$ for $i \neq 1$ since by hypothesis $\sigma_1 + \kappa < 0$ and $\sigma_1 < \sigma_i$. To find the second limit note that $\hat{u} = \hat{u}_1 \oplus \hat{u}_1^\perp$ and by writing $\hat{u}_1^\perp = \sum_{i\neq 1} \tilde{u}_i$ with $\tilde{u}_i \in E_i$ we can apply the first limit on each component. □

For large $s$ we have $\langle \hat{u}, \hat{v}\rangle \approx \langle \hat{u}_1, \hat{v}_1\rangle$, so the metric in the embedding space is negligible in all but the most stable Lyapunov direction. The proof fails when the hypothesis $0 < \kappa < -\sigma_1$ is not satisfied. When $\kappa \leq 0$, we cannot bound the matrix norm of $A(x)$, and in fact the construction does not yield a well defined metric. When $\kappa \geq -\sigma_1$, the norm converges to a finite value in each Oseledets component, destroying the projection onto the stable component.

The constants $h_{\text{min}}, h_{\text{max}}$ allow for local deviations from the long term behavior of the dynamical system, which is governed by the Lyapunov exponents. These constants are an extrinsic feature of the observed dynamics in the sense that they are accidental aspect of the observation function $h$. Of course, the dynamics would be uniform if we could reconstruct the intrinsic Lyapunov metric. Unfortunately, delays alone cannot reconstruct the Lyapunov metric, because we have shown that delay coordinates project the tangent spaces onto the most stable Oseledets subspace. The proof also illuminates the natural tradeoff in the choice of $\kappa$. For $\kappa$ near zero, the reconstruction projects strongly onto the stable component, but the dynamics may not be regular on this component. As we increase $\kappa$ the dynamics are increasingly regularized on the stable projection until we get to close to $-\sigma_1$, at which point the projection fails, according to Theorem 3.2.1. See Example 3.2.1 for an illustration of this tradeoff.

If the dynamics are reversible, the least stable Oseledets space in forward time is the most stable in reverse time. For a time series we can easily reverse time by inverting the time ordering of the data. Thus for $0 < \kappa < \max_i\{\sigma_i\}$ the algorithm can be applied to the reversed time series to project onto the least stable Oseledets space. This is equivalent to taking sequels instead of delays in the original time series. The bottom row in Figure 3.3 demonstrates the projection onto the least stable Oseledets space for the cat map.

This new interpretation of weighted time-delay coordinates reveals that they not only reconstruct the topology of the state space, but that they can also regularize the dynamics, while projecting onto the most stable Lyapunov direction. Of course, if the other Lyapunov directions correspond to dynamical noise this projection could be a useful feature. Moreover, even if the other Lyapunov directions correspond to interesting dynamics, these dynamics will be less stable, and therefore operate on a faster time-scale, than the most stable directions. Most importantly, the projection onto the most stable manifold will often achieve a significant dimensionality reduction.

**Example 3.2.1.** (Cat Map.) In this example we illustrate the effect on the geometry of the torus by weighted time-delay embedding via Arnold's cat map, a discrete time map on the torus $S^1 \times S^1$ given by

$$\begin{aligned} a_{j+1} &= 2a_j + b_j \bmod 1 \\ b_{j+1} &= a_j + b_j \bmod 1. \end{aligned}$$

We choose a point $(a_0, b_0) \in [0,1]^2$ randomly and apply the cat map for $N$ iterations, producing the points $\{(a_j, b_j)\}_{j=0}^{N}$.

To test Theorem 3.2.1, we embed the torus into $\mathbb{R}^3$ via

$$\begin{aligned} x_j &= (2 + \sin 2\pi a_j) \sin 2\pi b_j \\ y_j &= (2 + \sin 2\pi a_j) \cos 2\pi b_j \\ z_j &= \cos 2\pi a_j \end{aligned}$$

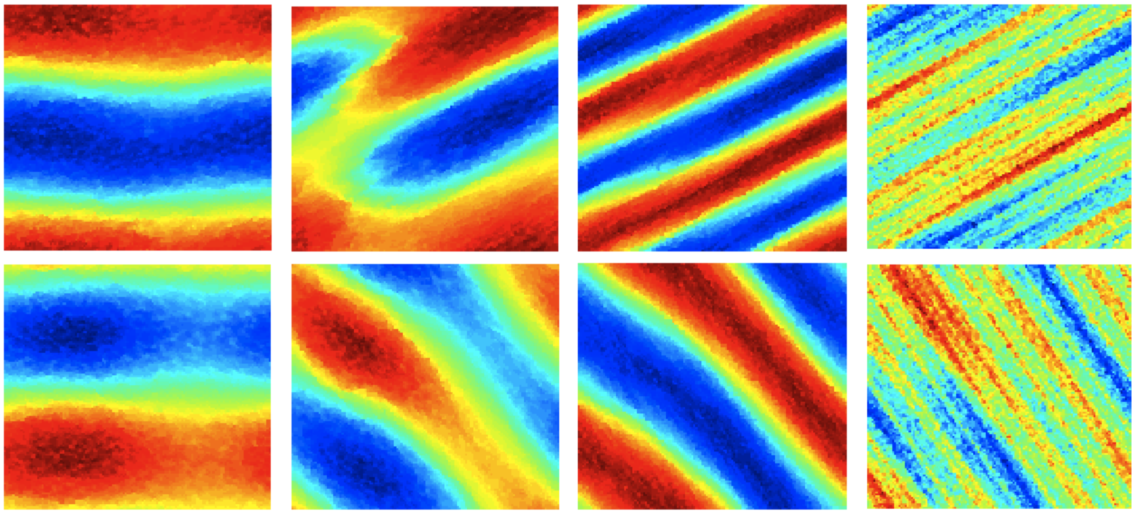


Figure 3.3: The top row shows the first nontrivial eigenfunctions of the Laplace-Beltrami operator in the $(a, b)$-plane, calculated from DMDC, for a time-delay embedding of the cat map with $s = 2048$ delays and $\kappa = 1.2, 0.8, 0.4, 0.01$ from left to right. When $\kappa = 1.2$, Theorem 3.2.1 does not apply, and the projection to the stable direction fails. When $0 < \kappa < -\sigma_1 \approx 0.962$, the projection successfully reconstructs the stable Oseledets direction in theory. As $\kappa \to 0$, the stable projection remains but regularization begins to fail. The bottom row shows DMDC applied to the time series in reverse order for the same values of $\kappa$.

and observe the dynamics through the weighted delay coordinates

$$w_j = [x_j, y_j, z_j, e^{-\kappa} x_{j-1}, e^{-\kappa} y_{j-1}, e^{-\kappa} z_{j-1}, ..., e^{-s\kappa} x_{j-s}, e^{-s\kappa} y_{j-s}, e^{-s\kappa} z_{j-s}].$$

Here we have chosen $r = 3$ in the application of Theorem 3.2.1.

The system is ergodic on the torus with one positive Lyapunov exponent and one negative. We can visualize the geometry of the time delay embedding by plotting the eigenfunctions of the Laplace-Beltrami operator on the embedded torus (see Figure 3.3). These eigenfunctions are computed using diffusion maps, described in Section 3.3; they are shown here to display the effect of varying $\kappa$.

When $\kappa > -\sigma_1 = -\log \frac{1}{2}(3 - \sqrt{5}) \approx 0.962$, we do not achieve the projection onto the stable Oseledets space and thus the Riemannian metric of the torus in $\mathbb{R}^3$ (our observation space) is recovered. This is shown in Figure 3.3 when $\kappa = 1.2$ by the eigenfunction oscillating vertically in the $(a, b)$-plane, which corresponds to the large circle on the torus in the $(x, y, z)$-space. As $\kappa$ decreases the metric becomes increasing localized in the stable Lyapunov direction. Note that for $\kappa < -\sigma_1$ the eigenfunctions become constant in the unstable direction $(1, (\sqrt{5} - 1)/2) \approx (1, 0.618)$, and oscillatory in the orthogonal stable direction. Thus the eigenfunctions of the Laplacian in the embedding space are only representing the location of the point on the stable manifold.

## 3.3 Diffusion maps for delay coordinates

In the previous section we found that an appropriately weighted time-delay embedding can reconstruct the intrinsic geometry of a smooth attractor from a generic observable. In cases where the observable is high-dimensional, such as a video, we will want to pair the delay embedding with a dimensionality reduction technique that preserves the geometry as faithfully as possible. In this section we show that a carefully chosen diffusion map will give, in a specific sense, the best preservation of the delay geometry. Moreover, we will show that the diffusion map has a natural dynamical interpretation when applied to time series.

We have seen that a time-delay embedding introduces a natural geometry on a dynamical system and we wish to preserve this geometry in the new (lower-dimensional) coordinates. Unfortunately, the sampling density in the embedding space also influences the geometry. The power of the diffusion maps technique is the ability to control the influence of the sampling density on the geometry in the new coordinates. We will see that for a certain normalization a diffusion map can match the invariant measure of a dynamical system.

In general there are many techniques for dimensionality reduction (see [21] for an overview). For the application to delay coordinates, a diffusion map is the natural choice because (1) it preserves geometry and (2) it has a dynamical interpretation. First, we will see that for any dimension $l$, a diffusion map into $\mathbb{R}^l$ yields the minimum distortion of the metric (and for $l > 2n$ the map will generically be an embedding). Secondly, when the data is a time series, each coordinate function of the diffusion map can be interpreted as a time series. In fact, each coordinate function will correspond to a generalized low-pass filter of the time series. In Section 3.4 we will show that this dual interpretation, as a geometry-preserving map and a low-pass filter, can be readily exploited for time-scale separation. In this section we review the main results of diffusion maps [2] and give a new interpretation of the construction as a conformal change of metric. We then interpret diffusion maps for generic observations of ergodic dynamical systems as a generalization of Fourier analysis.

### 3.3.1 Time series interpretation of diffusion maps

In Section 3.2 we saw how $\kappa$-weighted time-delay coordinates give an embedding of the invariant manfold into Euclidean space, where the Riemannian metric inherited from the ambient space is intrinsic to the dynamical system. Of course, while the Riemannian metric is intrinsic, the embedding itself is not and will depend on the details of the observation. Moreover, the dimension of the ambient space will be very large, preventing efficient analysis. For the delay coordinates to be useful we need to map them to a lower dimensional ambient space while maintaining the intrinsic geometry.

We now extend the interpretation of diffusion maps to the case when our data has a time ordering and show that setting the sampling bias parameter $\alpha = 1/2$ will match the invariant measure. Note that a diffusion map estimates the values of the eigenfunctions $\psi_l$ of a heat kernel at the discrete points $x_i$ which are the input to the diffusion maps algorithm. If the input points have a time ordering, we can give the eigenfunctions a time ordering by setting $\hat{\psi}_l(t_i) = \psi_l(x_i) = \psi_l(x(t_i))$. In Section 3.4 we will further investigate these time series. In this section we will show that if $y_i = y(t_i)$ is a trajectory of a dynamical system with an invariant measure, then the diffusion map interprets $y(t_i)$ as the trajectory of a simplified system which has the same invariant measure.

As a simple example, assume that the data $\{x_i\} \in \mathcal{M} \subset \mathbb{R}^m$ are generated by a path of the stochastic differential equation

$$dx = -\nabla U(x)\ dt + \sqrt{2}\ dB_t \tag{3.2}$$

where $U(x)$ is a smooth potential function on the manifold $\mathcal{M}$, $B_t$ is Brownian motion, and $x_i = x(i\tau)$ where $\tau$ is a sampling interval. The backward Fokker-Planck equation for the system given by (3.2) is

$$-\frac{\partial \varphi}{\partial t} = \Delta\varphi - \nabla\varphi \cdot \nabla U \tag{3.3}$$

and it was shown in [2] that

$$\mathcal{L}\varphi = \Delta\varphi - 2(1-\alpha)\nabla\varphi \cdot \nabla U,$$

so clearly when $\alpha = 1/2$, the diffusion map approximates the backward Fokker-Planck operator. The invariant measure of this system is given by the eigenfunction with eigenvalue zero of the corresponding forward Fokker-Planck operator

$$\mathcal{L}^*\varphi = \Delta\varphi + \nabla(\varphi\nabla U).$$

Thus, $e^{-U}$ is the invariant measure for (3.2) since $\mathcal{L}^*(e^{-U}) = 0$. Moreover, the invariant measure is also given by the eigenfunction $q_0(x_i) = (Q\psi_0)(x_i) \approx p(x_i) = e^{-U(x_i)}$.

Next we generalize to the case of dynamics that are more complex than those of (3.2). Assume that the data $\{y_i = y(t_i)\}$ are sampled from a trajectory of a dynamical system that has an invariant measure $\mu(y) > 0$ on a smooth manifold $\mathcal{M}$. Defining the potential function $U(x) = -\log\mu(x)$, equation (3.2) is a stochastic system for $x(t)$ that has the same invariant measure as $y(t)$. Note that $\mu = e^{-U}$ is the invariant measure for both $y(t)$ and $x(t)$, so in particular it is the sampling density for $y(t)$. A diffusion map with $\alpha = 1/2$ applied to the data set $\{y_i\}$ will construct the Fokker-Planck operator for (3.2) with $U = -\log\mu$. The eigenfunctions of this operator give a basis for square integrable functions on $\mathcal{M}$, and this basis is adapted to the invariant measure of our system. Thus, even when the trajectory is not governed by a stochastic differential equation of form (3.2), diffusion maps with $\alpha = 1/2$ will treat the data as if it were generated by (3.2) but with $U = -\log \mu$. The resulting eigenfunctions will be a generalization of the Fourier basis which is adapted to the correct invariant measure of $y(t)$ on $\mathcal{M}$.

## 3.4 Time-scale separation

The goal of time-scale separation is to decompose a dynamical system by a coordinate transformation such that the transformed variables are ordered by time scale and are approximately independent. In this chapter we are interested in working directly with multivariate time series where no equations are known. In this section we show how a basis of eigenvectors for an operator on the state space can be used to separate time scales. In particular we show that diffusion maps, combined with delay embeddings to reconstruct and simplify the state space, can separate time scales for a large class of interesting systems.

A diffusion map approximates eigenfunctions $\psi_l : \mathcal{M} \to \mathbb{R}$ of a heat kernel on a manifold; these are generalizations of sine and cosine. Note that the eigenfunctions of the Laplacian and the heat kernel are the same but the eigenvalues are different. We previously denoted the eigenvalues of the heat kernel by $1 = \lambda_0 \geq \lambda_1 \geq \cdots > 0$, so that $\gamma_l = -\log(\lambda_l)$ are the eigenvalues of the Laplacian operator. Moreover the eigenvalues $0 = \gamma_0 \leq \gamma_1 \leq \cdots$ order the eigenfunctions according to how oscillatory they are. Intuitively, given a continuous trajectory $y(t) : \mathbb{R} \to \mathcal{M}$ the time series formed by composing $\hat{\psi}_l(t) = \psi_l(y(t))$ will be more oscillatory for large $l$ and less oscillatory for small $l$. We will see that this time scale separation is most effective when the Laplacian operator is adapted to the dynamics of $y(t)$.

Consider a trajectory $y : \mathbb{R} \to \mathcal{M}$ which is reconstructed at discrete times $\{t_i\}_{i=1}^N$ by a delay embedding as $\{y(t_i)\}_{i=1}^N \subset \mathbb{R}^{r(s+1)}$. A diffusion map applied to the samples $\{y(t_i)\}$ will ignore the time ordering and treat these data points merely as a collection of samples from the manifold $\mathcal{M}$ embedded in $\mathbb{R}^{r(s+1)}$. If $\mu$ is the invariant measure for the evolution of $y(t)$, then we have seen in Section 3.3.1 that a diffusion map with $\alpha = 1/2$ will approximate the operator

$$\mathcal{L}(\varphi) = \Delta\varphi - \nabla\varphi \cdot \nabla(\log(\mu(x))).$$

Moreover, the diffusion map will approximate the eigenfunctions $\psi_l$ evaluated at the sample points $\{y(t_i)\}$. Thus we can consider the diffusion mapped modes to be time series given by $\hat{\psi}_l(t_i) = \psi_l(y(t_i)) = \langle \psi_l, \delta_{y(t_i)} \rangle$. The delta function $\varphi(x,t) = \delta_{y(t)}(x)$ along the trajectory of the dynamics allows us to apply harmonic analysis on the manifold to the trajectory. Moreover, the harmonic analysis is adapted to the dynamics because it matches the invariant distribution in the limit as $t \to \infty$.

Since diffusion maps can only approximate the operator $\mathcal{L}$, we now assume that we can write the full evolution as a non-autonomous perturbation of $\mathcal{L}$ so that

$$\frac{\partial\varphi}{\partial t} = -\mathcal{L}(\varphi) + \mathcal{F}(x,t). \tag{3.4}$$

We have seen that a diffusion map will produce a basis $\{\psi_l\} \subset L^2(\mathcal{M})$ consisting of eigenfunctions of $\mathcal{L}$ such that the eigenvalues are sorted as $0 = \gamma_0 \leq \gamma_1 \leq \gamma_2 \leq \cdots$. Therefore, the solution $\varphi(x,t)$ and the unknown function $\mathcal{F}$ have a decomposition in this basis so that (3.4) becomes

$$\frac{d}{dt}\langle \varphi(x,t), \psi_l(x) \rangle = -\gamma_l \langle \varphi(x,t), \psi_l(x) \rangle + \langle \mathcal{F}(x,t), \psi_l(x) \rangle$$

which is an ODE. Moreover, the time series $\hat{\psi}_l(t) = \langle \varphi(x,t), \psi_l(x) \rangle$ is precisely the output of our diffusion map. Setting $\hat{\mathcal{F}}(t) = \langle \mathcal{F}(x,t), \psi_l(x) \rangle$ we find that the $l$-th diffusion map mode satisfies

$$\frac{d}{dt}\hat{\psi}_l(t) = -\gamma_l \hat{\psi}_l(t) + \hat{\mathcal{F}}(t)$$

which has solution

$$\hat{\psi}_l(t) = ae^{-\gamma_l t} + b\int_0^t e^{-\gamma_l(t-s)}\hat{\mathcal{F}}(s)ds.$$

Thus, the eigenvalue $\gamma_l$ will determine the amount of history from the non-autonomous term $\hat{\mathcal{F}}$ is integrated into the mode $\hat{\psi}_l$. For $\hat{\mathcal{F}}$ sufficiently regular the time scale of $\hat{\psi}_l$ will be determined by $\gamma_l$. Of course, if $\hat{\mathcal{F}}$ is large then we may not be able to separate time scale with this method. However, since (3.2) and (3.4) have the same invariant measure, we expect at least the slowest time scales to be well approximated. Note that this method could be improved in the future by finding better methods of approximating $\hat{\mathcal{L}}$ which would make $\hat{\mathcal{F}}$ smaller and improve the time scale separation.

Previous work on time scale separation was motivated by singular perturbation theory, and focused on approximating the local dynamics. In particular, the local evolution of the slow variables was approximated by repeated simulations of the fast variables using legacy code [36], or stochastic simulators [17]. A closely related technique in [37] requires knowledge of the local fast and slow directions to decompose the tangent bundle to the state space. Since we have no knowledge of the system, the local factorizations in the observation space may not consistently identify the correct global fast and slow directions. Thus we have taken two methods of incorporating global information. First, the time delay embedding projects onto the most stable space and incorporates the long term behavior into the local geometry. Second, by projecting onto eigenfunctions of the Laplacian we are guaranteed a globally consistent coordinate transformation. While it is not yet clear whether this global technique will be able to achieve the higher order perturbation expansions of [36, 37], we have seen that for many interesting systems the global technique can at least project onto the slow manifold (see Example 3.4.1 also).

The ideas of this section are a generalization of the more formal approach taken in [12, 14]. It was shown in [14] that if a dynamical system had the simple form (3.2), then each spectral gap $\lambda_k >> \lambda_{k+1}$ produced a coarse time-scale version of the dynamics. This

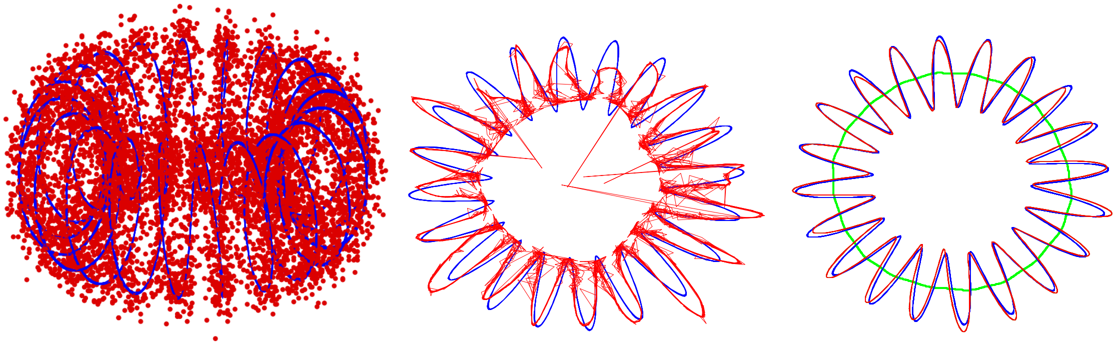


Figure 3.4: Reconstruction of dynamics from noisy observations using various values of $\kappa$. All plots show the $x, y$-plane dynamics in blue. (a) The noisy observations are shown in red (b) The reconstructed dynamics for $\kappa = 1$ is shown in red. (c) Same as (b), but $\kappa = 0.02$. The projection onto the slow manifold is shown in green.

means that in the coordinates $r(t) = \Psi_{1/2,s_k}(x(t))$ given by the diffusion map at scale $s_k = -1/\log(\lambda_{k+1})$ we have

$$\dot{r} = F(r, \omega)$$

for some $F$ which depends only on the reduced coordinates $r$ and on a stochastic noise vector $\omega$. While this is a strong result, exactly the type we are interested in, it applies in only a very narrow context. In fact, solutions of (3.2) simply follow the gradient of a potential and are driven purely by Brownian motion. This is a reasonable context for studying systems with large numbers of particles, although for a single trajectory (3.2) precludes even periodic dynamics. In contrast, many simple physical systems follow the gradient of a potential under the influence of a non-automous energy input as in (3.4).

**Example 3.4.1.** As an illustration, we analyze a simple dynamical system with two time scales. The dynamics lie on a one-dimensional subset of the torus embedded in $\mathbb{R}^3$ and are given by

$$\begin{aligned}\dot{x} &= -y + \frac{bzx}{\sqrt{x^2+y^2}} \\ \dot{y} &= x + \frac{bzy}{\sqrt{x^2+y^2}} \\ \dot{z} &= -b(\sqrt{x^2+y^2} - a)\end{aligned}$$

where we set $a = 6$, $b = 20$. We can write this system as a slow/fast system by changing to cylindrical coordinates where $r^2 = x^2 + y^2$ and $\tan\theta = y/x$. Setting $\epsilon = 1/b$ the system becomes

$$\begin{aligned}\dot{\theta} &= 1 \\ \epsilon\dot{r} &= z \\ \epsilon\dot{z} &= -(r-a)\end{aligned}$$

so setting $\epsilon = 0$ we find a slow system given by $z = 0$, $r = a$ and $\dot{\theta} = 1$. The slow system simply oscillates in the $x$,$y$-plane, and the fast direction is primarily in the $z, r$-plane.

To test our algorithm in an equation-free setting, we use only a time series of simulated data points, and do not incorporate any information from the above equations. The data will be given in the $x, y, z$ coordinates and to further illustrate the advantage of the delay geometry we introduce Gaussian observational noise (see Figure 3.4). The ODE with initial conditions $(x(0), y(0), z(0)) = (6, 0, 2)$ was simulated for $N = 8000$ equally spaced time steps from $t = 0$ to $t = 12\pi$. We form the delay coordinates $w_i = [x_i, y_i, z_i, ..., e^{-s\kappa}x_{i-s}, e^{-s\kappa}y_{i-s}, e^{-s\kappa}z_{i-s}]$ with $s = 500$ time delays with various values of $\kappa$ to study the time delay geometry.

Letting $\psi_l$ be the eigenvectors (as described in Section 2.3.2) we note that $(\psi_l)_{i-s} = \psi_l(w_i) = \langle \psi_l, \delta_{w_i} \rangle$ for $i = s+1, ..., N$. Thus each eigenvector can be interpreted spatially (as a function of the location $w_i$ in state space) and temporally (as a function of the time step $i$). In this way, we can interpret the eigenvectors as component time series, each of which represent the dynamics at a unique time scale. We can project the observed time series onto the span of the first $L+1$ eigenfunctions by setting

$$\hat{w}_i = \sum_{l=0}^{L} W_l \psi_l(w_i) \text{ where } W_l = \langle w, q_l \rangle = \sum_{i=s+1}^{N} w_i q_l(w_i) = \sum_{i=s+1}^{N} w_i (q_l)_{i-s}.$$

For $L = 2$ we achieve the projection onto the slow manifold as shown by the green curve in Figure 3.4(c). For $L = 60$ we achieve a low pass filter that removes high frequency oscillations as shown by the red plots in Figure 3.4(b)(c). Note that to determine the slow manifold and to successfully remove noise, having the correct geometry is crucial as shown by the failure of the reconstruction in Figure 3.4(b).

Furthermore, by taking the discrete Fourier spectrum of each of the time series $\psi_l(w_i)$, the time-scale separation is illustrated by the localized peaks in the Fourier spectrum of each eigenfunction and the ordering of the peaks as shown in Figure 3.5. Moreover, we can see that introducing the delays improves the time-scale separation. Finally, by reconstructing the dynamics using only the first few eigenvectors we can form the projection onto a slow time-scale. These projections can reveal slow manifolds as shown in Figure 3.4(c).

## 3.5 Time-scale separation for spatiotemporal dynamics

In this section we apply the proposed algorithm to video data. The first goal is to apply the algorithm to a meandering spiral wave, a reasonably complex model exhibiting spatiotemporal dynamics which are intrinsically low-dimensional. We will demonstrate how DMDC extracts meaningful low-dimensional dynamics and separates time scales.

We assume that observations take the form of still images which are discretized versions of $h(x)$, where $x$ is a dynamical state of an $n$-dimensional attractor $\mathcal{M}$ and $h : \mathcal{M} \to \mathbb{R}^r$, where $r$ represents the number of pixels of the still image. The delay coordinate map is

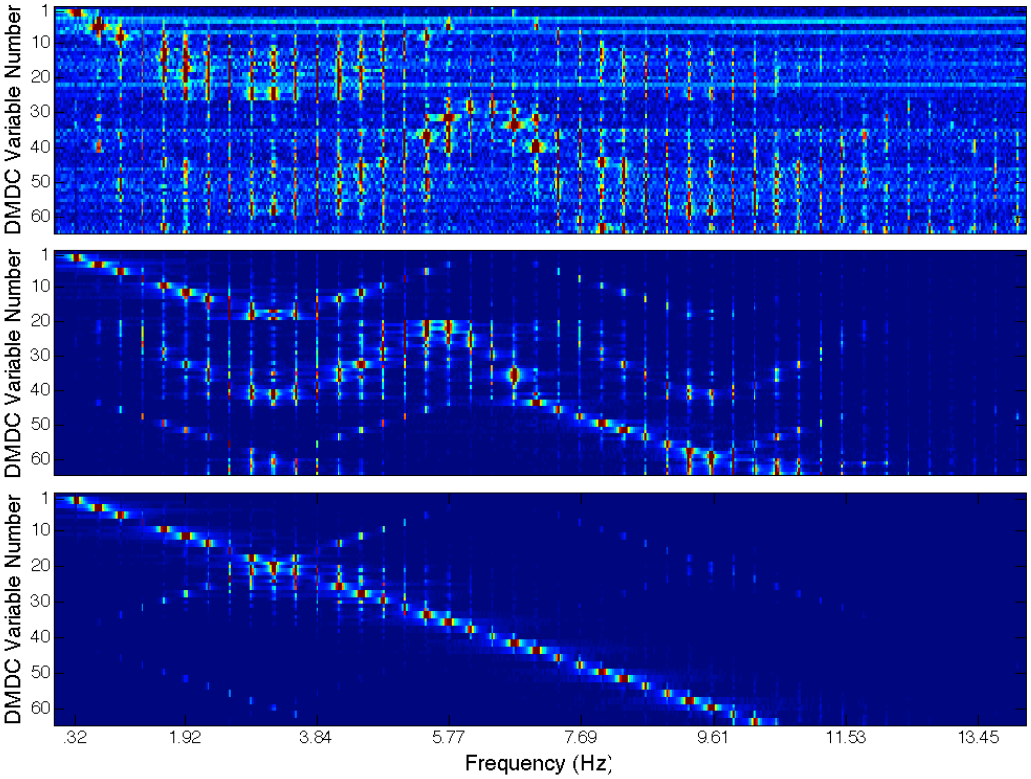


Figure 3.5: The Fourier spectrum of the first 64 eigenvectors of the Laplacian in the embedding geometry of Example 3.4.1 viewed as time series. The x-axis indicates the frequency of the oscillation, the y-axis indicates the eigenvectors and color represents norm of the Fourier coefficient for the relevant eigenvector at the indicated frequency. The top plot is for $\kappa = 10$, effectively no delays, the middle is for $\kappa = 0.1$, and the bottom is for $\kappa = 0.02$.

then $H : \mathcal{M} \to \mathbb{R}^{r(s+1)}$ from (3.1), where $s$ is the number of delays, which has the effect of concatenating $s + 1$ still images.

Note that both DMDC and SVD (on the delay embedding space) produce time series. For SVD the time series are linear projections of the time-delay embedding of the images (which are observations of the state), whereas for DMDC the time series are eigenfunctions of the Laplacian on the embedding. Let $x_i$ be the $i$-th image in the observed video. Note that as in Example 3.4.1 we can project the observed time series onto the span of the first

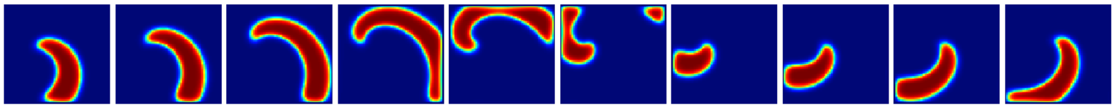


Figure 3.6: Images from spiral wave dynamics with $\rho = 0.01$, $a = 0.48$, $b = 0.01$ shown at time steps 0.3 apart (every sixth observed frame is shown).

$L + 1$ DMDC modes by setting

$$\hat{x}_i = \sum_{l=0}^{L} X_l \psi_l(x_i) \text{ where } X_l = \langle x, q_l \rangle = \sum_{i=s+1}^{N} x_i q_l(x_i) = \sum_{i=s+1}^{N} x_i (q_l)_{i-s}.$$

For video data the inner products $X_l = \langle x, q_l \rangle$ are averages of the images $x_i$ weighted by the $l$-th projection $q_l$. Thus, each $X_l$ is an image which shows the parts of the image which are varying synchronously with the $l$-th DMDC mode. Thus we call the image $X_l$ the $l$-th *component* of the decomposition. Note that SVD gives an orthonormal basis (instead of the biorthogonal system of DMDC) so for SVD the modes (analogous to $\{\psi_l\}$) and projections (analogous to $\{q_l\}$) are both given by the singular vectors of the centered delay coordinates.

### 3.5.1 Meandering spiral waves

We begin by applying DMDC to a model of meandering spiral waves. We find that resulting modes have a natural ordering which is determined by the time scale on which the mode is active. Then we compare results to the more conventional singular value decomposition in the delay-embedding space in order to illustrate the significant advantages of the DMDC modes. Moreover, we show that time-delay coordinates are crucial to finding the best decomposition, in that an application of a diffusion map without delays is not sufficient even for ideal observations from a computational model.

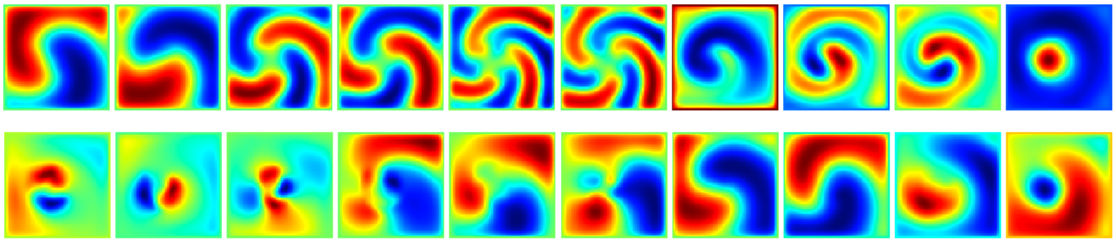


Figure 3.7: Top: First 10 modes of variation found from delay coordinates projected by SVD. Bottom: First 10 modes of variation found by DMDC. Images show the DMDC components (produced by averaging the spiral images weighted by a SVD or DMDC mode respectively). Note the first two DMDC modes are localized near the tip of the spiral and pick out the vertical and horizontal component of the slow-moving tip location.

Consider the reaction-diffusion system

$$\begin{aligned} u_t &= \Delta u + \frac{1}{\rho} u(1-u)\left(u - \frac{v+b}{a}\right) \\ v_t &= u - v \end{aligned}$$

where the parameters $\rho = 0.01, a = 0.48, b = 0.01$ determine the behavior of the meandering spiral waves. In [39, 40] bifurcations of the parameter space are explored and the system is shown to be related to a two-dimensional ODE, indicating the presence of a low-dimensional state space for the dynamics. Instead of working with the equations, our method will work directly with a sequence of still images separated by $\tau = 0.05$ (see Figure 3.6). The variables $u$ and $v$ are held to zero at the boundary and the initial condition is $u(x, y) = v(x, y) = \sin 7x$ where the domain is $(x, y) \in [0, 1]^2$. While the behavior is robust to adding small amounts of noise to the initial condition, totally random initial conditions often die out quickly and do not lead to self sustaining spirals.

In parallel with Example 3.1.2, we find that while SVD modes are organized according to variance, the DMDC modes are organized according to time-scale. For example, Figure 3.7 shows that the 7th and 8th DMDC modes correspond to the position of the main body of the spiral, which oscillates once per 60 frames. These modes are the first two modes found by SVD because they are the modes of largest variance in the video. In contrast, the first two DMDC modes represent the location of the tip of the spiral wave. These modes represent a slow precession in the spiral tip which oscillates every 240 frames. Thus the DMDC modes have identified an important slow mode of oscillation which has a low variance and thus would be very difficult to extract with SVD or an *ad hoc* technique.

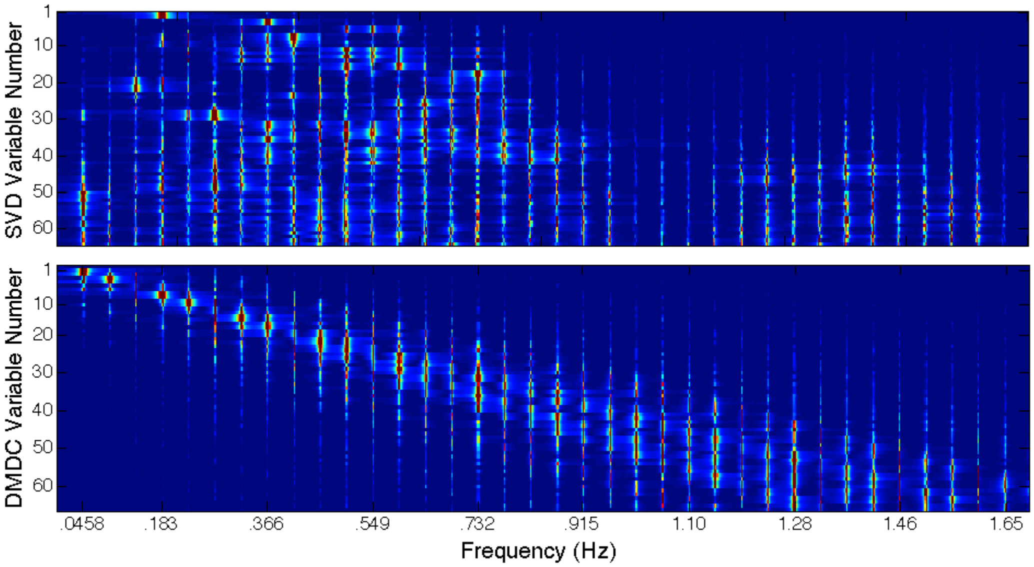


Figure 3.8: Top row: Fourier spectra of SVD modes reveals no time scale ordering, Bottom row: Fourier spectra of DMDC modes shows ordering according to Fourier spectral peak.

To underscore this point, Figure 3.8 shows that the time series corresponding to the DMDC modes are ordered according to the peaks in the Fourier spectra. The figure compares the Fourier spectra of the first 64 SVD modes with the first 64 DMDC modes. Note that the SVD spectra are spread out horizontally, indicating that SVD modes contain information about many different time scales, whereas the DMDC spectra are horizontally localized, indicating that the modes have a definite time-scale. Moreover, the DMDC spectra are ordered in the sense that higher order DMDC modes contain information at faster time scales. Thus, in analogy with the results of [14], projecting on the first $k$ DMDC modes would contain all the information about oscillations on a sufficiently slow time scale. For example, the first two DMDC modes contain all the information relevant to the slow precession of the spiral tip which has a period of approximately 240 frames. In contrast, the first two SVD modes capture the primary spiral oscillation which occurs approximately every 60 frames.

Finally, we note that DMDC can be used for noise reduction. In Figure 3.9 we show an example of additive Brownian noise. In order to make a fair comparison, notice in Figure 3.7 that the first two SVD modes represent the main spiral oscillation, and will be the best SVD modes in terms of noise reduction. Figure 3.7 shows that these modes also appear in the DMDC analysis, but occur later because of the time ordering. Thus these modes can be fairly compared for noise content, and in Figure 3.10 we show the dramatic noise reduction that DMDC achieves.

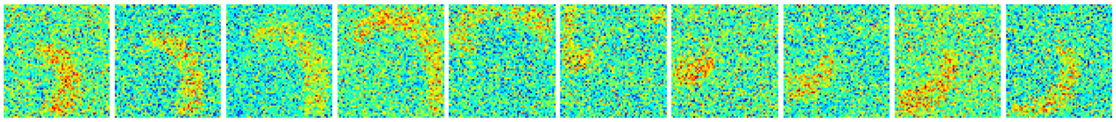


Figure 3.9: Same images from spiral wave dynamics as Figure 3.6, but with white noise added with standard deviation of 850% of the mean.

Intuitively, this noise reduction is achieved because DMDC is attempting to maximize the variance as measured on the manifold which the dynamics are near, whereas SVD tried to maximize the variance as measured in the ambient space in which the data is embedded, as shown in Section 2.3.3.

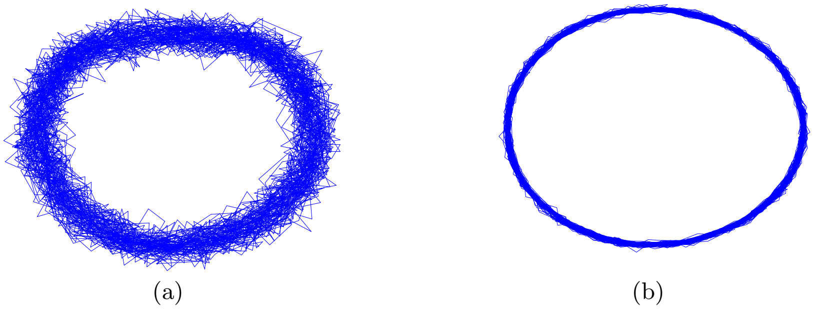


Figure 3.10: Embedding of noisy spiral images from Figure 3.9, using (a) first two SVD modes (b) equivalent two DMDC modes.

A further interpretation of the noise reduction results is that DMDC provides a data-adapted Fourier transform. If the invariant measure is uniform, or if we take $\alpha = 1$ in the diffusion map analysis to eliminate sampling effects, then the DMDC modes are given by the inner product with the eigenfunctions of the Laplace-Beltrami operator on the manifold $\Sigma$. This is a well known generalization of Fourier analysis to manifolds, and in this natural basis of generalized Fourier modes the observation noise will be concentrated in the high frequency modes. In the case where the invariant measure is nontrivial this analysis gives a possible further generalization of Fourier analysis adapted to measures on manifolds.

## 3.6 Discussion

The main theme of this chapter is that the methodology used to analyze complex data should depend on the end goal of the analysis. In the case under discussion, we are concerned with reconstructing the topology and geometry of dynamical data, without knowledge of the governing equations. Depending on the purpose, different parts of the reconstructed geometry can be considered intrinsic or extrinsic to the data, the intrinsic geometry being simply that part which relates to the features of the data which one wishes to study. Previous applications of diffusion maps [2] tend to consider the sampling density of a data set an ‘extrinsic’ detail of the geometry, whereas the angles defined by the geometry were considered ‘intrinsic’.

In this chapter we have applied the extrinsic/intrinsic dichotomy more broadly. Dynamical data, unlike generic data, has the additional structure of a time-ordering. There is a more fundamental notion of intrinsic geometry for dynamical systems which goes beyond the geometry of the observations. This is the geometry of the Lyapunov metric [38], which is independent of the embedding space of the dynamical system. Thus, for studying dynamical data, the natural extension of the ideas of diffusion maps is to try to isolate the intrinsic geometry of the Lyapunov metric, and remove the extrinsic factors of the observed embedding. This is the focus of Sections 3.2 and 3.3 of this chapter. Our use of diffusion maps in the algorithm focuses on its preservation of the local geometry on the reconstructed manifold. The choice of the sampling bias parameter $\alpha$ allows a flexible choice of measure, which is crucial to isolating the slow dynamical modes. The end goal of the DMDC algorithm is to construct a version of harmonic analysis for the observed system, that is specially adapted to the system by using the observed data to approximate its heat kernel.

While the idea of combining nonlinear dimensionality reduction with a time-delay reconstruction has been explored previously, most notably in [4], DMDC provides the theoretical connection between these techniques. Moreover, we have shown that careful adjustment of the time-delay embedding (via our weighting scheme) and the diffusion maps algorithm (via careful selection of parameters) is necessary for these techniques to work together optimally.

The theory of nonlinear dimensionality reduction is quickly advancing with an emphasis on understanding the geometry of data, and the theory developed here provides the basis for future techniques to leverage the intrinsic geometry of dynamical systems. For example, it is interesting to compare this technique to the approach of local SVD in the embedded state space, mentioned in the introduction. While a local SVD can preserve, or even finely adjust, the local geometry, it currently cannot generate a global low-dimensional representation. On the other hand, a diffusion map gives a global low-dimensional representation, but it inherits the local geometry given by the embedding and cannot currently leverage the geometric information of a local SVD. Combining the global approach of diffusion maps with the extra information available from a local SVD is a question that has only begun to be addressed (for example in [10]). This is a fundamental shortcoming of diffusion maps, and future developments should allow promising new extensions of DMDC.

Although the theory in Section 3.2 was written for deterministic dynamics, assuming the existence of a standard Oseledets splitting, the arguments can be adapted to hold in the case of stochastic dynamics, as in the more general splitting of [41]. Here Takens' theorem would be replaced by a stochastic version as in [42, 43]. Moreover, the basic idea of the theorem, that stable directions are magnified in backward time by the delay coordinates, may apply in more generality, such as within multidimensional Oseledets subspaces, leading to finer notions of time scale separation.

Incorporating the temporal structure of data into the theory of local and shift-local kernels is a promising direction for future research. In particular, since local kernels can define new geometries on the data, it may be possible to approximate the full Lyapunov geometry empirically using a well designed local kernel. Moreover, improving the approximation of the forward operator would dramatically improve the ability to separate time scales as discussed in Section 3.4. If shift-local kernels can be used to approximate Fokker-Planck operators we speculate that the associated eigenbases may provide natural time scale separations for complicated dynamical systems.

# Chapter 4: Leveraging spatial and spatiotemporal structure

The previous chapter focused on the temporal structure of data. However, many data sets will also have a spatial structure, which should inform our analysis. The most obvious examples are images (including 3-dimensional and hyper-spectral images), and it is immediately clear that treating the pixels as independent and unrelated observations, as PCA does for example, is a considerable underutilization of the data. Many *ad hoc* solutions are available, each specially tuned to various data types and applications, but the growing complexity of these problems calls for a more unified strategy. In this section we present some preliminary ideas and related computational examples which motivate the geometric approach to understanding the spatial structure of data.

We propose a two-step approach to utilizing the spatial structure of data. In the first step a user-supplied spatial structure, such as a pixel layout or a network structure, is used to develop a harmonic analysis or multi-scale wavelet analysis on the supplied spatial structure. We refer to this first step as the *spatial analysis.* Combined with the temporal analysis, the *in situ* spatial and temporal structure of the data set can be fully utilized. In the second step we modify the spatial structure to fit the actual content of the data, to form an *adapted* spatial analysis.

## 4.1 Learning spatial structure

Many large data sets have a known spatial structure, meaning that at any time each data coordinate has a fixed relationship to the other coordinates. In an image the pixels have a certain layout, and each pixel has a known relative distance to every other pixel. Weather, economic, and commercial data may be associated with various sensor, survey, and factory locations for example. In even greater generality, information networks, neural networks, and even social networks often have a priori notions of relative distances between the various coordinates which make up a data set. Of course, these *a priori* notions of distance may not always capture the perfect notion of similarity (we will address this in the next section), but they usually capture at least some information about the similarity of the various coordinates.

In each example described above, the spatial structure consists of a notion of distance between the various coordinates of a data set. If the data can be viewed as $n$-dimensional vectors, a spatial structure will be defined to be an $n \times n$ distance matrix $S$ whose $i, j$ entry represents the spatial distance between the $i$th and $j$th coordinates of the data vector $v \in \mathbb{R}^n$. Thus, if $v$ were a $1024 \times 1024$ pixel image represented as a $1024^2$-dimensional vector, then $S$ would be a $1024^2 \times 1024^2$ matrix containing the distances between each pair of pixels. For simplicity we assume here that the spatial structure is the same for each $v$ and that the full matrix $S$ is known, although these are not required.

The key to the proposed spatial analysis is to consider the $i$th coordinate of the vector $v$ to be a node $x_i$ on the underlying spatial structure, and then consider the data vector $v$ to be a function on the set of the spatial nodes $\{x_i\}$. As a concrete example, consider an image, where $\Omega$ is the plane of the image and the spatial nodes $x_i$ are the pixels. The image $v$ can be viewed as a function assigning a value to each pixel, or $v : \Omega \to \mathbb{R}$. We consider $v \in L^2(\Omega)$ to be a square integrable function, written in the standard basis of delta functions $v(x) = \sum_{i=1}^{N} v(x_i)\delta_{x_i}(x)$. In analogy to classical signal processing, we could smooth or localize the vector $v$ by writing it with respect to Fourier bases or wavelet bases on $\Omega$. In fact, we develop special versions of Fourier analysis and wavelet analysis on this spatial structure to allow us to represent $v$ in more suitable bases.

Note that the distance matrix $S$ which represents our spatial structure also defines a weighted graph between the nodes $\{x_i\}$. The notion of harmonic analysis on graphs for data analysis began with Belkin and Niyogi [7] who considered the eigenfunctions of the graph Laplacian to be generalizations of sine and cosine functions. These ideas were made formal by Coifman and Lafon in [2] who showed that a subtle renormalization of the graph

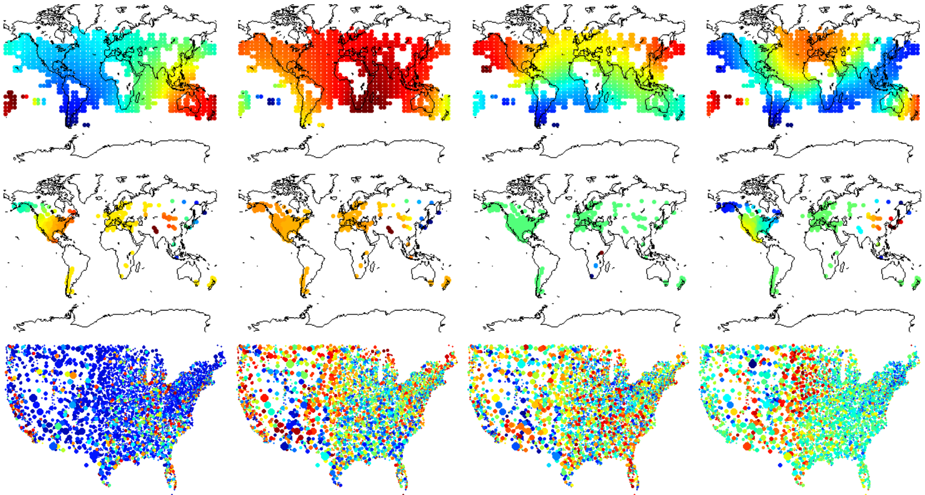


Figure 4.1: Examples of generalized harmonic modes of the proposed approach. The rows represent spatial structure of (top) temperature data from the NOAA database, where the left-right periodicity reveals the spherical geometry; (middle) tree-ring growth index data from the NCDC database (note in particular the harmonic mode in the third frame localized on three locations in Africa, an effect that would be absent in classical Fourier analysis); and (bottom) U.S. zip code demographic data where proximity is defined using demographic data from the 2010 Census and dot size indicates relative areas of the zip codes. These generalized harmonic modes provide an efficient basis for representing data defined on these spatial coordinates; intuitively they provide a custom low pass filter which allows the user to define proximity in a way that can isolate intrinsic properties of the data set.

Laplacian is required to approximate the Laplace-Beltrami operator when the graph nodes are sampled from a smooth manifold. In [2], Coifman and Lafon also introduced the notion of a multiscale distance on the graph, and in [5] Coifman and Maggioni used a sparse orthogonalization technique to build wavelets and a multi-scale analysis on a general graph. Finally, [1] developed an abstract construction of wavelets on graphs and a fast wavelet transform based on a Chebyshev polynomial expansion of the mother wavelet which is then applied to the graph Laplacian. In Figures 4.1 and 4.2 we construct examples of harmonics and wavelets on various type of spatial graphs that correspond to different data types.

In previous work, graph harmonic and wavelet analysis constructions have been applied directly to the full data set using the distances between the full data vectors. Such an analysis implicitly assumes that the coordinates have no special relationships, since the distance between two vectors collects all the coordinate differences simply by summation.

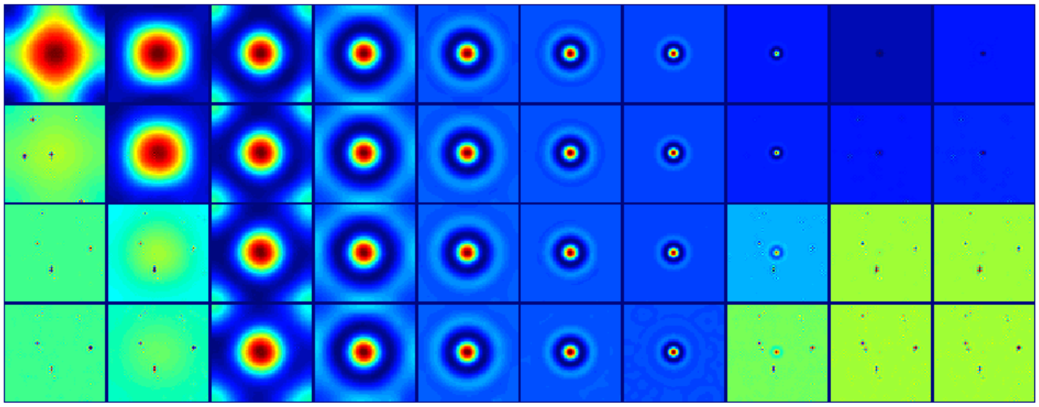


Figure 4.2: Example of spatial scale collapse in small world network. The spatial structure begins with an 64-by-64 pixel where each pixel is connected to the 256 nearest neighbors using the planar distance. To create a small world, we progressively introduce (top to bottom) 0,70,140, and 210 short connections (short cuts) between randomly chosen pixels. For each number of short connections, we show a graph wavelet centered in the image at 10 spatial scales (left to right) constructed using the technique of [1] with a second derivative of a Gaussian for the mother wavelet and using 100 Chebyshev polynomial terms. Note that there are $2^{24}$ possible connections in this graph, and initially $2^{20}$ are active, and yet adding just 210 connections leads to the collapse of at least five spatial scales.

While this simplifying assumption has led to good results in toy examples, in the examples below we will indicate how a spatial analysis can dramatically improve the understanding of data structure. For example, Figure 4.2 shows how a common spatial structure, known as a small world graph, leads to the collapse of spatial scales. A dataset represented on a collapsed spatial scale (such as the naive full representation on the basis of delta functions) would be very inefficient.

## 4.2 Dynamically-adapted spatiotemporal analysis

In Section 4.1 we showed how to leverage an existing spatial structure to improve the representation of data. However, often the spatial structure that is available is only partially defined and does not fully capture the correct notion of similarity in the data set. In this section we propose a radical restructuring of the data, by using the data itself, together with the given spatial structure (if available), to construct an adapted spatial analysis. We will present a preliminary technique applied to images, which is based on building a over-complete dictionary for the adapted spatial structure by sampling sub-images. A related idea was suggested in [5] and developed in [16]; however, it was not realized that the data samples have to be considered a over-complete dictionary and not a basis. Moreover, this technique is limited to two scales, a micro-scale at the level of sub-images, and a macro-scale that represents the full images. We propose to generalize this technique from images to arbitrary spatial structures using a wavelet basis to capture the initial notion of localization. This wavelet basis will also allow a full multi-scale generalization of the two-scale approach. Finally, we propose to investigate the theoretical foundations of such a micro/macro or multi-scale splitting of data, we believe that underlying these splittings are fundamental connections to abstract algebraic structures represented by symmetries in the data, as described next.

Consider a collection of related images, as from a video. Often there are similarities between sub-images across this set, even within each image. For example, in a video of an object moving in the plane of the screen, the content of the frames is substantially the same but with the sub-images rearranged. More generally, natural images tend to be dominated by sharp one-dimensional boundaries and smooth two-dimensional regions, a fact that has been exploited using sparse optimal basis reconstruction [44]. The weakness of an optimal basis construction, obtained through residual minimization, is that it does not recognize the geometry of the sub-image space, and thus it cannot generalize to slightly different elements. If a video contains a rotating object, then the optimal basis will require a different basis element for each angle in the rotation. Instead, by automatically finding the symmetries of the sub-image space, our proposed method will summarize all these angles into a single basis element plus a variable that gives the angle. Of course, since we are going to study the geometry of the entire subimage space, there is no need to restrict ourselves to a basis, and thus we will use an over-complete representation of the sub-image space.

For the sake of clarity we present a concrete example. Figure 4.3 shows frames from a video of a liquid crystal being driven periodically at 10 and 12 volts. At 10 V, the dynamics are dominated by rolls, which appear as continuous lines from the bottom left of the image

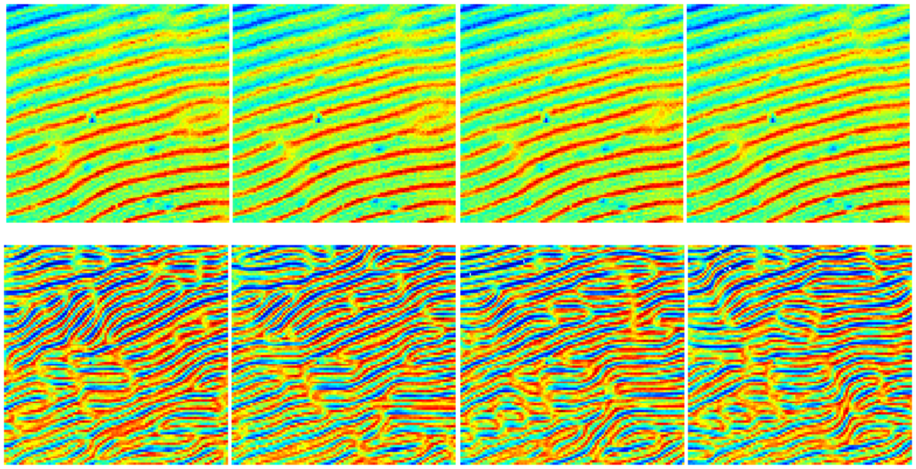


Figure 4.3: Frames from a recording of liquid crystal experiment at 10 V (top) and 12 V (bottom) produced by Dr. John Cressman's lab in the Physics Department at George Mason University.

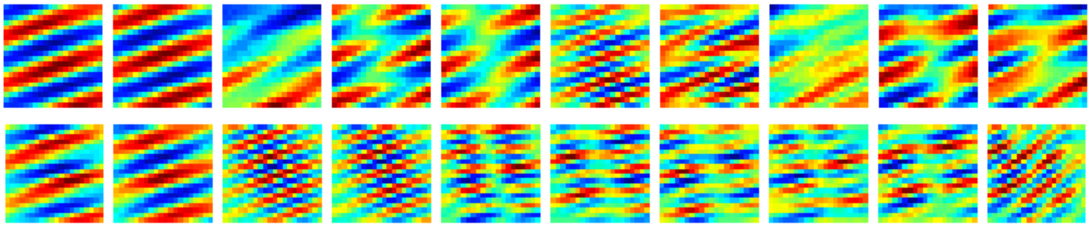


Figure 4.4: Nonlinear components from the space of sub-images for 10V (top) and 12V (bottom). Each sub-image is constructed using a principal eigenmode of the adapted spatial Fourier basis.

to the top right. These rolls move slowly. The other dynamical features of the video are the defects, which are breaks between rolls. The defects move in the plane of the image much faster than the rolls. When the voltage increases to 12 V, the defects become the dominant features and the dynamics become highly complex. These liquid crystal dynamics are an ideal illustration of how sub-images of a collection of images can have a relatively simple structure that is opaque to current data analysis.

To find the geometry of the sub-image space we randomly sample 4096, 20-by-20 pixel sub-images $\{x_j\}_{j=1}^{4096}$ from various locations and frames of a liquid crystal video. These 4096 sub-images are considered points sampled from a manifold $\Omega$ embedded in $\mathbb{R}^{400}$, and we use the technique of Section 4.1 to build an adapted Fourier basis on the manifold of sub-images. When we compare these sub-images we do not take their location into account; however the location is used when the sub-images are considered as an over-complete basis for the set of all full images. Each element of the adapted Fourier basis is a function $\psi_i : \Omega \to \mathbb{R}$ defined on this manifold, and thus we can represent $\psi_i$ as a sub-image by setting $X_i = \sum_j \psi_i(x_j)x_j$. Using this method, we constructed the ten lowest frequency eigenfunctions of the Fourier basis for the 10 V and 12 V liquid crystal videos; these modes are shown in Figure 4.4. The $X_i$ sub-images can be thought of as the nonlinear generalization of principal components for the micro-scale model. Note that these sub-images automatically pick out the characteristic features that make up the larger images, namely the rolls and the defects.

Now we can integrate the micro-scale model into the macro-scale analysis on the full image by realizing each full image $Y$ as both a vector represented in the over-complete dictionary $\{x_j\}_{j=1}^{4096}$ of all sub-images and simultaneously as a function on the sub-image space $\Omega$. For this technique to be valid the set of sub-images (considered as full images that are zero outside the sub-image location) must be large enough to satisfy the frame condition for the subspace of $L^2(\Omega)$ generated by the full images. In other words, the sub-images must sufficiently sample the manifold $\Omega$ so that each full image $Y$ can be reconstructed from the collection of inner products $\{\langle Y, x_j \rangle\}$ with all the localized sub-images. Thus we can consider each full image to be equivalent to this collection of inner products, however this data set can also be considered as discrete samples $f(x_j) = \langle Y, x_j \rangle$ of a function $f : \Omega \to \mathbb{R}$ defined on the space of all sub-images. Thus, when we write this function in a Fourier or wavelet basis on the space of sub-images, we have represented the full image in this basis. Now we can significantly improve our understanding of the full image space since the adapted geometry of the sub-images efficiently represents the features of the images.

We propose two generalizations of the technique presented in this section for general spatio-temporal analysis. First, in the micro/macro-scale dichotomy, the basis elements could be sub-videos, consisting of a sub-images taken across multiple frames. This would allow the spatial and temporal structures to interact at the micro scale. Second, we propose

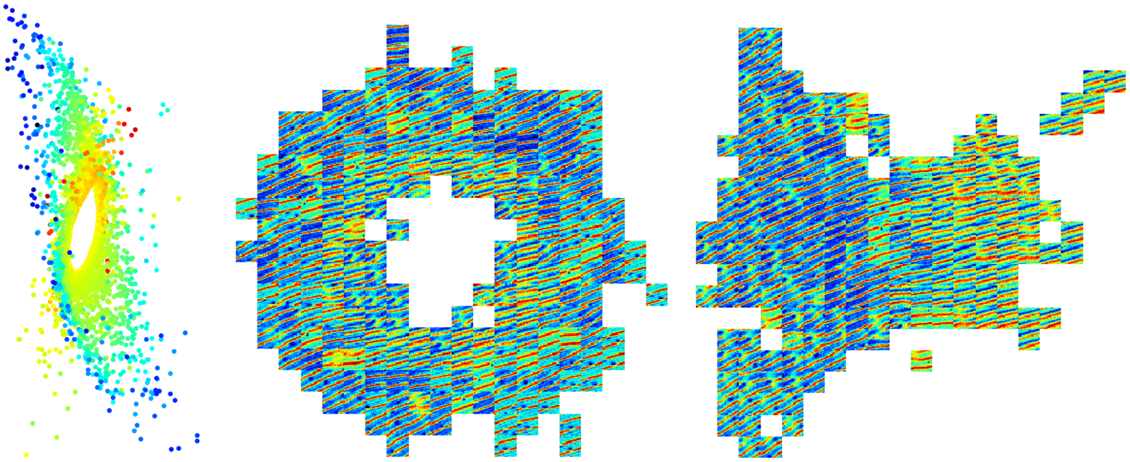


Figure 4.5: Left: The 4-D geometry of the sub-image space defined by the liquid crystal dynamics at 10V; the fourth dimension is represented as color. Middle: The first 2 dimensions represent the phase of the horizontal rolls in the sub-image as an annulus. Representative sub-images are plotted at their location in the 2-dimensional subspace spanned by the first two modes. Right: The third and fourth dimensions represent the darkness of the roll and the amount of defect in the sub-image respectively.

to expand the micro/macro-scale model into a full multi-scale model by using a spatial wavelet basis as constructed in Section 4.1. This would also allow us to more easily apply our method to arbitrary spatial structures such as weighted networks where the notion of sub-image may not have a natural generalization. In the multi-scale approach, we first build a full wavelet basis $\psi_{i,s}$ where $i$ ranges over all the spatial coordinates (for images it ranges over the pixels) and $s$ ranges over the dyadic scales. Note for a macro data point $Y$, at any fixed location $i$, the collection of inner products $Y_{i,s} = \langle Y, \psi_{i,s} \rangle$ represent the local information in $Y$ centered at the $i$th spatial node. The coefficients $\{Y_{i,s}\}_{s=1}^{S}$ are therefore a multiscale generalization of a sub-image to an arbitrary spatial structure. In analogy to our analysis of the micro-scale geometry, we propose to study the multiscale geometry using the data set $\{\hat{Y}_{i,s} = (Y_{i,s}, i, s) : 1 \leq i \leq N, 1 \leq s \leq S\}$ and distance $d(\hat{Y}_{i,s}, \hat{Y}_{j,t}) = |Y_{i,s} - Y_{j,t}| + \sigma_1 |i - j| + \sigma_2 |s - t|$. This generalizes the notion of distance between sub-images by comparing the wavelet coefficients at each location and scale; then $\sigma_1$ and $\sigma_2$ allows us to introduce spatial and scale heterogeneity respectively. In the simple case where $\sigma_1 = \sigma_2 = 0$ we are simply assuming that the geometry is uniform at all locations and across the scales. Alternatively, if we did not include the first term and set $\sigma_2 = 0$ we would recover the initial spatial distance. Thus $d$ is a data-adapted spatial distance, and the parameter $\sigma_1$ controls the influence of the *a priori* spatial structure. The distance $d$ gives us a new spatial structure on the data and we can use the methods of Section 4.1 to construct data-adapted versions of Fourier and wavelet bases.

## 4.3 Diffusion wavelets and symmetries

A successful data-adapted spatial analysis depends on a redundant structure in which similar patterns are repeated at various times and locations. Hopefully these repeated patterns have small changes that can be represented in a low-dimensional feature space. Understanding the symmetries in the underlying feature space may allow one to isolate a particular desired feature. We now overview the emerging field of wavelets on manifolds and graphs as another context where understanding the symmetries of nonlinear space is important.

Diffusion wavelets were introduced in [5] and seek to generalize the construction of wavelet bases to smooth manifolds where there is still a natural notion of translation given by the local coordinate maps. However, since there is no dilation operator on a manifold, Maggioni and Coifman substitute a diffusion operator. Since then, a related combinatorial construction has also been developed in [1]. For a smooth manifold $\Omega$, the prototype of a diffusion operator is the heat kernel of the laplacian on a manifold given by

$$T\phi = \int_{\Omega} K(t,x,y)\phi(y)dy = e^{t\triangle}\phi$$

which is always a smoothing operator. Thus, instead of dilations on a Euclidean domain, diffusion wavelets are adapted to a diffusion operator such as $T$. In [1,5] it is shown that one can define a multiresolution adapted to the operator $T$ in analogy to the classical wavelet construction. Moreover, algorithms for constructing scaling functions and wavelets from the operator $T$ are presented in [1, 5].

The key to constructing a wavelet analysis from the multiresolution is finding a basis which is made up of translates of a single function. Since there is no reason to expect the eigenfunctions of $T$ (the generalized Fourier basis) to have this property, another approach is required to construct such a basis. Both [5] and [1] consider a simple notion of translation which allows one to move anywhere on the manifold or the graph. However, translation on a manifold can be considerably more complex than on the Euclidean domains of classical wavelet analysis. In particular, curved manifolds can have nontrivial holonomy, which implies that parallel transport around a closed loop can change the coordinate system. A more detailed understanding wavelets on a nonlinear domain requires respecting the symmetry group which defines translation on that domain. For example, we may wish to consider wavelets which are invariant to only a certain subgroup of the full symmetry group. In the remainder of this section we will explore the challenges of understanding the symmetries of the discrete geometry and introduce some possibilities for addressing them.

In the geometric context, a symmetry is simply a map, called an *isometry*, from the manifold to itself which preserves the geometry. In the context of discrete geometry, [45, 46] introduced the idea of invariance under the diffusion geometry as a generalization of isometry. This was followed by a method for finding intrinsic symmetries in a data set using the eigenfunctions of the Laplace-Beltrami operator with isolated eigenvalues [47]. However, isolated eigenvalues correspond to simple reflection symmetries, and we are more interested in repeated eigenvalues which correspond to subgroups of orthogonal matrices. These complex group structures may represent significant inefficiencies in the data representation. Moreover, in the case of many interconnected symmetries, there may be efficient representations which correspond to group factorizations. Finally, by finding partial symmetries we may be able to produce new, unobserved data by generating the data points that would complete these symmetries. We propose to investigate how known symmetry groups can be represented in image spaces with the goal of finding a theoretical basis for the reduction achieved by the data-adapted methods proposed in this section.

**Example 4.3.1.** In this example we illustrate how the eigenfunctions of the Laplace-Beltrami operator can be used to search for symmetries. Using the same data from Figure 4.5 above, we applied the diffusion maps construction to build a discrete approximation to the Laplace-Beltrami operator. We found that many features of the subimage space were well represented by using just 8 eigenfunctions, so we will restrict this analysis to just those eigenfunctions. The first eigenfunction has eigenvalue zero and represents the sampling density on the manifold. The fourth eigenvalue was isolated and may have described a reflection symmetry, but here we are interested in continuous symmetries so we will focus on repeated eigenvalues. The second and third eigenvalues were very close numerically, and also the fifth through eighth were very close numerically. This implies that any subgroup of $SO(2) \oplus SO(4)$ (where $SO(n)$ is the special orthogonal group of dimension $n$ over $\mathbb{R}$) is a candidate for a symmetry.

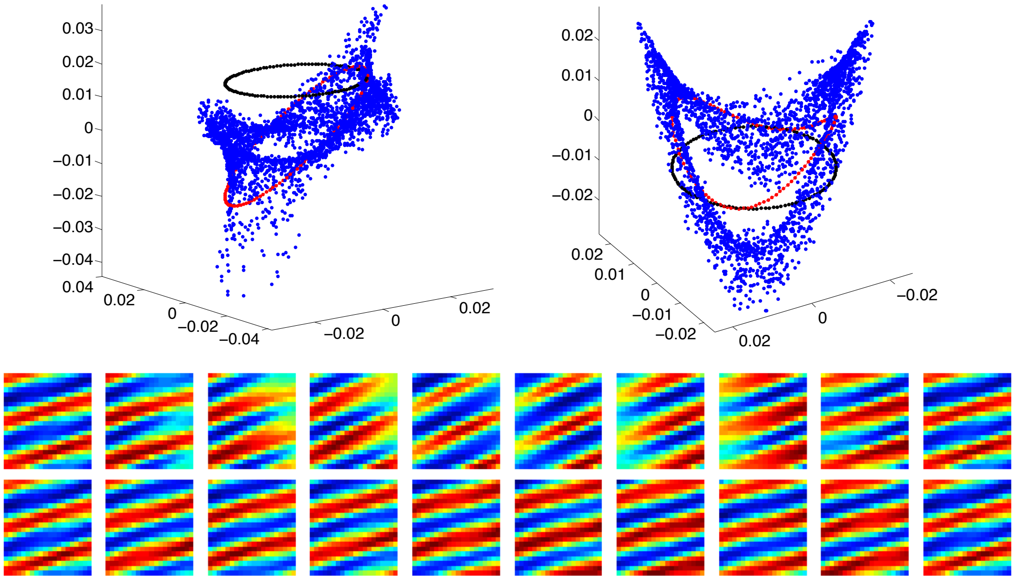


Figure 4.6: Comparison of a naive symmetry (black trajectory) using only modes 2 and 3 and an improved symmetry (red trajectory) using modes 2-3 and 5-8. Top left, modes 2,3 and 6 plotted for the data set (blue) a naive symmetry (black) and improved symmetry (red). Top right shows modes 2,3, and 7. Each row of images shows a reconstructed symmetry, the top row is the naive symmetry and the bottom row is the improved symmetry shown in the plots above. Note that the improved symmetry captures the phase of the stripes.

Naively, we first select a rotation $\theta \in SO(2)$ of $\pi/60$ radians. We selected a random subimage, and applied the group element $\theta \oplus Id_{SO(4)} \in SO(2) \oplus SO(4)$ to the generalized Fourier coefficients of the subimage. In Figure 4.6, we see that the naive rotation in the second and third coefficients did not stay on the manifold. This is not inconsistent, any symmetry must correspond to a subgroup of $SO(2) \oplus SO(4)$, however many such subgroups will not correspond to symmetries. Clearly, $\theta \oplus Id_{SO(4)}$ does not generate a symmetry on this manifold. Next, we constructed a more complex symmetry $\theta \oplus (\theta \oplus 2\theta)$ which stays much closer to the manifold, as shown in Figure 4.6, and is probably close to a real symmetry. In Figure 4.6 we also reconstruct the images that correspond to the paths of these two candidate symmetries. This reveals that the well constructed symmetry captures the phase of the stripes in the subimage, while the poorly constructed symmetry does not seem to capture a feature of the data set.

The above example demonstrates the challenges of finding symmetries. The construction of Section 2.5 gives a strong possibility for finding these symmetries. As we have seen, eigenforms of the Laplacian on 1-forms correspond to vector fields on the manifold. These vector fields generate deterministic dynamical systems whose solutions give one parameter translation groups on the manifold. The prospect of using the discrete geometry to find symmetries which represent data features illustrates how the geometric perspective of data analysis leads to new ways of understanding data.

# Chapter 5: Leveraging existing models

So far we have seen that time ordering, spatial structure, and meta-data provide a priori structure to a data set which should inform the analysis. Existing models also provide an a priori structure to the data which can be extremely sophisticated. In some sense, these models can be considered a compression or summary of many specialized experiments or related data sets which are not available and would be difficult to connect to the full analysis. However, even if all this data was available, existing models often implicitly incorporate physical constraints or first-principles modeling assumptions which amount to extremely complex priors. Nevertheless, modeling errors, or differences between the model and reality, are ubiquitous and difficult to quantify or compensate for.

As we have previously emphasized, non-parametric modeling is strongly limited by the curse-of-dimensionality, making it unrealistic and wasteful to expect a non-parametric technique to recover an existing model. The natural opportunity for non-parametric modeling is to improve an existing model by compensating for model error. However, compensating for model error first requires extracting the component of the data which is not already explained by the model.

In this chapter we show that filtering techniques, such as the Kalman filter and its many extensions, naturally extract the portion of the data which is not explained by an existing model. As we will see, a Kalman-type filter produces a time series of errors, called innovations, which are the difference between the model prediction and the observed data. However, the innovation is a complex mixture of model error and uncertainty in the state variables. This makes using the innovations to extend the model in a way which can be used by the filter is non-trivial. As a first step towards resolving this difficulty we present a novel technique for extending the model with a stochastic term. In the discussion section we will indicate how this is a natural first step towards non-parametric model extensions.

## 5.1 Overview

The Kalman filter is provably optimal for systems where the dynamics and observations are linear with Gaussian noise. If the covariance matrices $Q$ and $R$ of the model error and observation error, respectively, are known, then the standard equations provided originally by Kalman [48] give the maximum likelihood estimate of the current state. If inexact $Q$ and $R$ are used in Kalman's equations, the filter may still give reasonable state estimates, but it will be suboptimal.

In the case of linear dynamics, [49, 50] showed that the exact $Q$ and $R$ could be reconstructed by auxiliary equations added to the Kalman filter, in the case of Gaussian white noise model error. This advance opened the door for adaptive versions of the filter. More recently, approaches to nonlinear filtering such as the Extended Kalman Filter (EKF) and Ensemble Kalman Filter (EnKF) have been developed [51–55]. Proposals have been made [56–59] to extend the idea of adaptive filtering to nonlinear filters. However, none of these showed the capability of recovering an arbitrary $Q$ and $R$ simultaneously, even in the simplest nonlinear setting of Gaussian white noise model error and observation error. One goal of this chapter is to introduce an adaptive scheme in the Gaussian white noise setting that is a natural generalization of Mehra's approach, and that can be implemented on a nonlinear Kalman-type filter. We demonstrate the reconstruction of full, randomly-generated error covariance matrices $Q$ and $R$ in Example 5.4.1.

The technique we describe builds on the innovation correlation method of Mehra. Significant modifications are required to lift this technique to the nonlinear domain. In particular, we will explicitly address the various stationarity assumptions required by Mehra which are violated for a nonlinear model. As shown by Mehra and later [57] and [56], the innovation correlations are a blended mixture of observation noise, predictability error (due to state uncertainty), and model error. Disentangling them is nontrivial due to the nonstationarity of the local linearizations characteristic of strongly nonlinear systems. Many previous techniques have been based on averages of the innovation sequence, but our technique will not directly average the innovations.

Instead, at each filter step, the innovations are used, along with locally linearized dynamics and observations, to recover independent estimates of the matrices $Q$ and $R$. These estimates are then integrated sequentially using a moving average to update the matrices $Q_k^{\text{filt}}$ and $R_k^{\text{filt}}$ used by the filter at time step $k$. By treating each innovation separately, using the local linearizations relevant to the current state, we are able to recover the full matrices $Q$ and $R$ when the model error is Gaussian white noise, and to improve state estimates in more complicated settings.

Gaussian white noise model error is an unrealistic assumption for practical applications. Typically, more significant model errors, as well as partial observability, must be addressed. In this case, we interpret the covariance matrix $Q_k^{\text{filt}}$ (see equation (5.2)) as an additive covariance inflation term whose role is to prevent filter divergence and to improve the estimate of the underlying state.

Covariance inflation, originally suggested in [60], adjusts the forecast error covariance to compensate for systematic underestimation of the covariance estimates given by the Kalman filter. There is a rapidly growing list of approaches to covariance inflation strategies [58,61–66]. In [58,67,68] techniques are presented for finding inflation factors that implicitly assume a very simple structure for the noise covariance. There are also many techniques based on the covariance of the implied observation errors at each time step [61,69–72]. These were actually anticipated and rejected by Mehra, who called them covariance matching techniques in [50]. He showed that these techniques were not likely to converge unless the system noise was already known. Since untangling the observation noise and system noise is the key difficulty of adaptive filtering, we do not wish to assume prior knowledge of either.

At the same time, new theory is emerging that begins to give mathematical support to particular applications, including the recent article [73] that shows convergence of the EnKF in the sense of the existence of a shadowing trajectory under reasonable hyperbolicity assumptions. In their approach, there is no system noise, and perfect knowledge of the (deterministic) model and the observation noise covariance matrix are necessary to make the proof work. Furthermore, optimality is not addressed, which naturally leads to the question of the optimal choice of the covariance matrices used by the filter. We speculate that an appropriate adaptive filter may be the key to finding the optimal covariance matrices.

We begin the next section by demonstrating the importance of knowing the correct $Q$ and $R$ for the performance of a Kalman filter. Next we recall the adaptive filter developed in [49,50] that augments the Kalman filter to estimate $Q$ and $R$ in the case of linear dynamics. In Section 5.3 we describe an adaptive filter that can find $Q$ and $R$ in real time even for nonlinear dynamics and observations, building on the ideas of Mehra. In Section 5.4 we test the adaptive filter on a 40-dimensional model of [74] and show the dramatic improvements in filtering that are possible. In addition, we show that this adaptive filter can also compensate for significant model error in the Lorenz96 model. We also propose new ideas to extend our technique to the cases of rank deficient observations and non-additive noise, and discuss an implementation that augments a localized version of the ensemble Kalman filter.

## 5.2 Extensions of the Kalman filter

Kalman filtering [48] is a well-established part of the engineering canon for state and uncertainty quantification. In the linear case with Gaussian noise, Kalman's algorithm is optimal. Since our main interest is the case of nonlinear dynamics, we will use notation that simplifies exposition of two often-cited extensions, the Extended Kalman Filter (EKF) and the Ensemble Kalman Filter (EnKF). For simplicity we will work in the discrete setting, and assume a nonlinear model of the form

$$\begin{aligned} x_{k+1} &= f(x_k) + \omega_{k+1} \\ y_{k+1} &= h(x_{k+1}) + \nu_{k+1} \end{aligned} \tag{5.1}$$

where $\omega_{k+1}$ and $\nu_{k+1}$ are zero-mean Gaussian noise with covariance matrices $Q$ and $R$, respectively. The system is given by $f$, and $Q$ is the covariance of the one-step dynamical noise. The value of an observation is related to $x$ by the function $h$, with observational noise covariance $R$. To simplify notation, we assume the covariance matrices are fixed in time, although in later examples we allow $Q$ and $R$ to drift and show that our adaptive scheme will track slow changes.

We are given a multivariate time series of observations, with the goal of estimating the state $x$ as a function of time. The Kalman filter follows an estimated state $x_k^a$ and an estimated state covariance matrix $P_k^a$. Given the estimates from the previous step, the Kalman update first creates forecast estimates $x_{k+1}^f$ and $P_{k+1}^f$ using the model, and then updates the forecast estimate using the observation $y_{k+1}$. The goal of nonlinear Kalman filtering is to correctly interpret and implement the linear Kalman equations

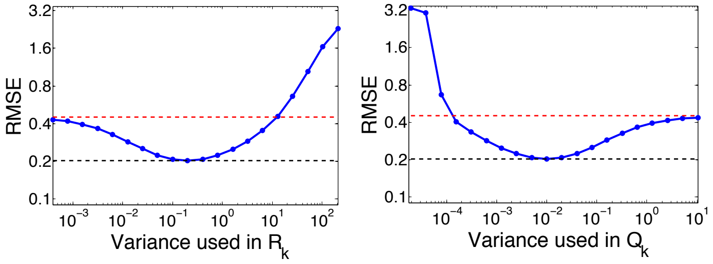
$$\begin{aligned}
x_{k+1}^f &= F_k x_k^a \\
y_{k+1}^f &= H_{k+1} x_{k+1}^f \\
P_{k+1}^f &= F_k P_k^a F_k^T + Q_k^{\text{filt}} \\
P_{k+1}^y &= H_{k+1} P_{k+1}^f H_{k+1}^T + R_k^{\text{filt}} \\
K_{k+1} &= P_{k+1}^f H_{k+1}^T (P_{k+1}^y)^{-1} \\
P_{k+1}^a &= (I - K_{k+1} H_{k+1}) P_{k+1}^f \\
x_{k+1}^a &= x_{k+1}^f + K_{k+1} \left( y_{k+1} - y_{k+1}^f \right).
\end{aligned} \tag{5.2}$$

where $P_{k+1}^y$ is the covariance of the observation, and $y_{k+1}^f$ is the forecast observation implied by the previous state estimate. The matrices $Q_k^{\text{filt}}$ and $R_k^{\text{filt}}$ should be chosen to equal $Q$ and $R$, respectively, in the linear case; how to choose them in general is the central issue of our investigation. The EKF extends the Kalman filter to systems of the form (5.1) by explicitly computing linearizations $F_k$ and $H_{k+1}$ from the dynamics. In the EnKF, these quantities are computed more indirectly, using ensembles; we give more complete details in Section 5.4.

Filter performance depends strongly on the covariances $Q_k^{\mathrm{filt}}$ and $R_k^{\mathrm{filt}}$ that are used in the algorithm. In fact, for linear problems the algorithm is provably optimal only when the true covariances $Q_k^{\mathrm{filt}} = Q$ and $R_k^{\mathrm{filt}} = R$ are used. For nonlinear problems, even for Gaussian white noise model error, using the exact $Q_k^{\mathrm{filt}} = Q$ can lead to filter instability (see Figure 5.5(c)-(d)). This is because the local linearizations $F_k$ and $H_{k+1}$ introduce an additional model error which may systematically underestimate $P_k^f$. More generally, the model error in applications is typically not Gaussian white noise, and $Q_k^{\mathrm{filt}}$ must be interpreted as an additive inflation which attempts to compensate for the covariance structure of the model error.

In Figure 5.1 we illustrate the effect of various suboptimal $Q_k^{\mathrm{filt}}$ and $R_k^{\mathrm{filt}}$ on a nonlinear problem by running an EnKF on a 40-site Lorenz96 model [74, 75]. (Full details of the model are given in Section 5.4.) In the limit of large $Q_k^{\mathrm{filt}}$ or small $R_k^{\mathrm{filt}}$ we find that the RMSE of filtered state estimates approach the RMSE of the unfiltered signal. Intuitively, when $Q_k^{\mathrm{filt}}$ is large and $R_k^{\mathrm{filt}}$ is small the filter has no confidence in the forecast relative to the observation and the filter simply returns the observation. For a filter to be nontrivial we must use a smaller $Q_k^{\mathrm{filt}}$ and a larger $R_k^{\mathrm{filt}}$. However, as shown in Figure 5.1, when the former is too small or the latter is too large, the RMSE of the filtered signals can actually be higher than the RMSE of the original noisy signals. In one extreme the filter becomes trivial, and in the other extreme it is possible for the filter to actually degrade the signal. Figure 5.1 illustrates that a key to filter performance lies in the choice of $Q_k^{\mathrm{filt}}$ and $R_k^{\mathrm{filt}}$.

In Section 5.3 we present a novel adaptive scheme that augments equation (5.2) to also update the matrices $Q_k^{\mathrm{filt}}$ and $R_k^{\mathrm{filt}}$ at each filter step based on the observations. Since the adaptive scheme is a natural generalization of the Kalman update, it can be used in many of the extensions of the Kalman filter for nonlinear problems.



(a) (b)

Figure 5.1: Results of EnKF on a Lorenz96 data set (see Section 5.4) with 400 different combinations of diagonal $Q_k^{\text{filt}}$ and $R_k^{\text{filt}}$ matrices. RMS error was computed by comparing the filter output to the time series without observation noise. The correct $Q$ and $R$ (used to generate the simulated data) were diagonal matrices with entries $Q_{ii} = 0.01$ and $R_{ii} = 0.2$, respectively. The RMS error of the signal prior to filtering was 0.44 (shown as red dotted line) the RMSE of the optimal filter using $Q_k^{\text{filt}} = Q$ and $R_k^{\text{filt}} = R$ was 0.20 (shown as black dotted line). In (a) we show the effect of varying $R_k^{\text{filt}}$ when $Q_k^{\text{filt}} = Q$ and in (b) the effect of varying $Q_k^{\text{filt}}$ when $R_k^{\text{filt}} = R$.

## 5.3 An adaptive nonlinear Kalman filter

For the case of linear dynamics with linear full-rank observation, the adaptive filtering problem was solved in two seminal papers [49, 50]. Mehra considered the innovations of the Kalman filter, which are defined as $\epsilon_k = y_k - y_k^f$ and represent the difference between the observation and the forecast. In his innovation correlation method, he showed that the cross-correlations of the innovations could be used to estimate the true matrices $Q$ and $R$. Intuitively, this is possible because cross-correlations of innovations will be influenced by the system and observation noise in different ways, which give multiple equations involving $Q$ and $R$. These differences arise because the perturbations of the state caused by the system noise persist and continue to influence the evolution of the system, whereas each observation contains an independent realization of the observation noise. When enough observations are collected, the resulting system can be solved for the true covariance matrices $Q$ and $R$.

### 5.3.1 Adaptive Kalman filter for linear dynamics.

In the case of linear dynamics with linear full-rank observations, the forecast covariance matrix $P^f$ and the Kalman gain $K$ have a constant steady state solution. Mehra shows that for an optimal filter we have the following relationship between the expectations of the cross-correlations of the innovations and the matrices involved in the Kalman filter:

$$\begin{aligned}\Gamma_0 &\equiv \mathbb{E}[\epsilon_k \epsilon_k^T] = HP^f H^T + R \\ \Gamma_1 &\equiv \mathbb{E}[\epsilon_{k+1} \epsilon_k^T] = HFP^f H^T - HFK\Gamma_0.\end{aligned}$$

Now consider the case when the Kalman filter is not optimal, meaning that the filter uses matrices $Q_k^{\mathrm{filt}}$ and $R_k^{\mathrm{filt}}$ that are not equal to the true values $Q$ and $R$. In this case we must consider the issue of filter convergence. We say that the filter converges if the limit of the state estimate covariance matrices exists and is given by

$$M \equiv \lim_{k\to\infty} P_k^f = \lim_{k\to\infty} \mathbb{E}[(x_k - x_k^f)(x_k - x_k^f)^T]$$

where $P_k^f$ is the estimate of the uncertainty in the current state produced by the Kalman filter at time step $k$. The filter is called non-optimal because the state estimate distribution will have higher variance than the optimal filter, as measured by the trace of $M$. As shown in [49, 50], for the non-optimal filter, $M$ is still the covariance matrix of the state estimate as long as the above limit exists. Moreover, $M$ satisfies an algebraic Riccati equation given by

$$M = F\left[(I - K_\infty H)M(I - K_\infty H)^T + K_\infty R K_\infty^T\right] F^T + Q \tag{5.3}$$

where $K_\infty$ is the limiting gain of the non-optimal filter defined by $Q^{\mathrm{filt}}$ and $R^{\mathrm{filt}}$. Note that the matrices $Q$ and $R$ in (5.3) are the true, unknown covariance matrices.

Motivated by the appearance of the true covariance matrices in the above equations, Mehra gave the following procedure for finding these matrices and hence the optimal filter. After running a non-optimal filter long enough that the expectations in $\Gamma_0$ and $\Gamma_1$ have converged, the true $Q$ and $R$ can be estimated by solving the equations

$$\begin{aligned} M &= (HF)^{-1}(\Gamma_1 + HFK_\infty \Gamma_0)H^{-T} \\ R &= \Gamma_0 - HMH^T \\ Q &= M - F\left[(I - K_\infty H)M(I - K_\infty H)^T + K_\infty R K_\infty^T\right]F^T. \end{aligned} \tag{5.4}$$

Clearly this method requires $H$ to be invertible, and when the observation had low rank Mehra could not recover $Q$ with his method. With a more complicated procedure he was still able to find the optimal Kalman gain matrix $K$ even when he could not recover $Q$. However, this procedure used the fact that the optimal gain is constant for a linear model.

### 5.3.2 Extension to nonlinear dynamics.

Our goal is to apply this fundamental idea to nonlinear problems. Unfortunately, while the technique of Mehra has the correct basic idea of examining the time correlations of the innovation sequence, there are many assumptions that fail in the nonlinear case. First, the innovation sequence is no longer stationary, and thus the expectations for $\Gamma_0$ and $\Gamma_1$ are no longer well-defined. Second, the matrices $H$ and $F$ (interpreted as local linearizations) are no longer fixed and the limiting values of $K$ and $P^f$ no longer exist, and therefore all of these matrices must be estimated at each filter step. Third, Mehra was able to avoid explicitly finding $Q$ in the case of a rank deficient observation by using the limiting $K$, which does not exist in the nonlinear case. For nonlinear dynamics the optimal Kalman gain is not fixed, making it more natural to estimate $Q$ directly. When the observations are rank-deficient, parameterization of the matrix $Q$ will be necessary.

Consequently, the principal issue with lifting Mehra's technique to a nonlinear model is that the local linear dynamics are changing with each time step. Even in the case of Gaussian white noise model error, the only quantities which may be assumed fixed over time in the nonlinear problem are the covariance matrices $Q$ and $R$. With this insight, we solve Mehra's equations (5.4) at each filter step and update $Q_k^{\text{filt}}$ and $R_k^{\text{filt}}$ at each step with a exponentially weighted moving average. Thus our iteration becomes

$$\begin{aligned}
P_k^e &= (H_{k+1}F_k)^{-1}(\epsilon_{k+1}\epsilon_k^T + H_{k+1}F_kK_k\epsilon_k\epsilon_k^T)H_k^{-T} \\
Q_k^e &= P_k^e - F_{k-1}P_{k-1}^aF_{k-1}^T \\
R_k^e &= \epsilon_k\epsilon_k^T - H_kP_k^fH_k^T \\
Q_{k+1}^{\text{filt}} &= Q_k^{\text{filt}} + \delta(Q_k^e - Q_k^{\text{filt}}) \\
R_{k+1}^{\text{filt}} &= R_k^{\text{filt}} + \delta(R_k^e - R_k^{\text{filt}}) \qquad (5.5)
\end{aligned}$$

where $\delta$ must be chosen small enough to smooth the moving average. Note that these equations naturally augment the Kalman update in that they use the observation to update the noise covariances. This iteration is straightforward to apply with the Extended Kalman Filter, because $H_k$ and $F_k$ are explicitly known. For the Ensemble Kalman Filter, these quantities must be estimated from the ensembles. This extra step is explained in detail in Section 5.4.

To explain the motivation behind our method, first consider the following idealized scenario. We assume that the unknown state vector $x_k$ is evolved forward by a known time varying linear transformation $x_{k+1} = F_kx_k + \omega_{k+1}$ and observed as $y_{k+1} = H_{k+1}x_{k+1} + \nu_{k+1}$. Furthermore, we assume that $\omega_{k+1}$ and $\nu_{k+1}$ are stationary white noise random variables which are independent of the state, time, and each other, and that $\mathbb{E}[\omega\omega^T] = Q$ and $\mathbb{E}[\nu\nu^T] = R$. In this scenario, we can express the innovation as

$$\epsilon_k = y_k - y_k^f = H_k(x_k - x_k^f) + \nu_k.$$

If we were able to average over all the possible realizations of the random variables $x_k, x_k^f$, and $\nu_k$ (conditioned on all previous observations) at the fixed time $k$ we would have

$$\mathbb{E}[\epsilon_k\epsilon_k^T] = \mathbb{E}[H_k(x_k - x_k^f)(x_k - x_k^f)^TH_k^T] + \mathbb{E}[\nu_k\nu_k^T] = H_kP_k^fH_k^T + R.$$

However, the most important realization is that only the last expectation in this equation can be replaced by a time average. This is because unlike $\nu_k$, which are independent, identically distributed random variables, the distribution $P_k^f$ is changing with time and thus each innovation is drawn from a different distribution. In our idealized scenario, the non-stationarity of $\epsilon_k$ arises from the fact that the dynamics are time varying, and more generally for nonlinear problems this same effect will occur due to inhomogeneity of the local linearizations.

Since we have no way to compute the expectation $\mathbb{E}[\epsilon_k \epsilon_k^T]$, we instead compute the matrix

$$\begin{aligned} R_k^e &= \epsilon_k \epsilon_k^T - H_k P_k^f H_k^T \\ &= \nu_k \nu_k^T + H_k(x_k - x_k^f)(x_k - x_k^f)^T H_k^T - H_k P_k^f H_k^T + \nu_k (x_k - x_k^f)^T H_k^T + H_k (x_k - x_k^f) \nu_k^T. \end{aligned}$$

Note that the last two terms have expected value of zero (since $\mathbb{E}[\nu_k] = 0$ for all $k$), and for each fixed $k$ the difference

$$H_k(x_k - x_k^f)(x_k - x_k^f)^T H_k^T - H_k P_k^f H_k^T$$

has expected value given by the zero matrix. Since these terms have expected value of zero for each $k$, we can now replace the average over realizations of $\nu_k$ with an average over $k$ since $\nu_k$ is assumed to be stationary. Thus we have

$$\lim_{K \to \infty} \frac{1}{K} \sum_{k=1}^{K} R_k^e = \mathbb{E}_k[R_k^e] = \mathbb{E}_k[\nu_k \nu_k^T] = \mathbb{E}_{\nu_k}[\nu_k \nu_k^T] = R$$

where $\mathbb{E}_k$ denotes an average over time and $\mathbb{E}_{\nu_k}$ denotes an average over possible realizations of the random variable $\nu_k$. This motivates our definition of $R_k^e$ as the empirical estimate of the matrix $R$ based on a single step of the filter. Our method is to first recover the stationary component and then average over time instead of averaging the innovations. Thus the equation for $R_k$ is simply a moving average of the estimates $R_k^e$. Of course, in real applications, the perturbation $\nu_k$ will not usually be stationary. However, our method is still advantageous since the matrix $\epsilon_k \epsilon_k^T$ is largely influenced by $H_k P_k^f H_k^T$ and thus by subtracting these matrices we expect to improve the stationarity of the sequence $R_k^e$.

A similar argument motivates our choice of $P_k^e$ and $Q_k^e$ as the empirical estimates of the forecast and model error covariances respectively. First, we continue the expansion of the $k$-th innovation as

$$\begin{aligned}
\epsilon_{k+1} = y_{k+1} - y_{k+1}^f &= H_{k+1}(x_{k+1} - x_{k+1}^f) + \nu_{k+1} \\
&= H_{k+1}(F_k x_k + \omega_{k+1} - F_k x_k^a) + \nu_{k+1} \\
&= H_{k+1}(F_k x_k - F_k(x_k^f + K_k \epsilon_k)) + H_{k+1}\omega_{k+1} + \nu_{k+1} \\
&= H_{k+1}F_k(x_k - x_k^f) - H_{k+1}F_k K_k \epsilon_k + H_{k+1}\omega_{k+1} + \nu_{k+1}.
\end{aligned}$$

By eliminating terms which have mean zero we find the following expression for the cross covariance

$$\epsilon_{k+1}\epsilon_k^T = H_{k+1}F_k(x_k - x_k^f)(x_k - x_k^f)^T H_k^T - H_{k+1}F_k K_k \epsilon_k \epsilon_k^T.$$

Note that the expected value of $(x_k - x_k^f)(x_k - x_k^f)^T$ is the forecast covariance matrix $P_k^f$ given by the filter. Solving the above equation for $(x_k - x_k^f)(x_k - x_k^f)^T$ gives an empirical estimate of the forecast covariance from the innovations. This motivates our definition of $P_k^e$, the empirical forecast covariance, which we find by solving

$$H_{k+1}F_k P_k^e H_k^T = \epsilon_{k+1}\epsilon_k^T + H_{k+1}F_k K_k \epsilon_k \epsilon_k^T.$$

It is tempting to use the empirical forecast to adjust the filter forecast $P_k^f$, however this is infeasible. The empirical forecast is extremely sensitive to the realization of the noise and since the forecast covariance is not stationary for nonlinear problems there is no way to average these empirical estimates. Instead, we isolate a stationary model error covariance which can be averaged over time by separating the predictability error from the forecast error.

In our idealized scenario, the forecast error $P_k^f = P_k^p + Q$ is the sum of the predictability error $P_k^p$ and the model error $Q$ which we wish to estimate. We can estimate the predictability error using the dynamics and the analysis covariance from the previous step of the Kalman filter as $P_k^p \approx F_{k-1}P_{k-1}^a F_{k-1}^T$. Finally, we estimate the model error covariance by

$$Q_k^e = P_k^e - F_{k-1}P_{k-1}^a F_{k-1}^T.$$

As with $R_k^e$, in our example of Gaussian white noise model error, this estimate of $Q_k^e$ is stationary and thus can be averaged over time in order to estimate the true model error covariance $Q$. While in real applications the actual form of the forecast error is more complicated, a significant component of the forecast error will often be given by the predictability error estimated by the filter. Thus it is natural to remove this known non-stationary term before attempting to estimate the stationary component of the model error.

If the true values of $Q$ and $R$ are known to be constant in time, then a cumulative average can be used, however this would take much longer to converge. The exponentially weighted moving average allows the estimates of $Q_k^{\text{filt}}$ and $R_k^{\text{filt}}$ to adjust to slow or sporadic changes in the underlying noise covariances. A large delta will allow fast changes in the estimates but results in $Q_k^{\text{filt}}$ and $R_k$ varying significantly. We define $\tau = 1/\delta$ to be the stationarity time scale of our adaptive filter. Any changes in the underlying values of $Q$ and $R$ are assumed to take place on a time scale sufficiently greater than $\tau$. In the next section, we will show (see Figure 5.2) that our iteration can find complicated covariance structures while achieving reductions in RMS error.

### 5.3.3 Compensating for rank-deficient observations.

For many problems, $H_k$ or $H_{k+1}F_k$ will not be invertible because the number of observations per step $m$ is less than the number of elements $n$ in the state. In this case, the above algorithm cannot hope to reconstruct the entire matrix $Q$. However, we can still estimate a simplified covariance structure by parameterizing $Q_k^e = \sum q_p Q_p$ as a linear combination of fixed matrices $Q_p$. This parameterization was first suggested by Bélanger in [76]. To impose this restriction we first set

$$C_k = \epsilon_{k+1}\epsilon_k^T + H_{k+1}F_k K_k \epsilon_k \epsilon_k^T - H_{k+1}F_k F_{k-1}P_{k-1}^a F_{k-1}^T H_k^T$$

and note that we need to solve $H_{k+1}F_k Q_k^e H_k^T = C_k$. Thus we simply need to find the vector $q$ with values $q_p$ that minimize the Frobenius norm

$$\left|\left| C_k - \sum_p q_p H_{k+1} F_k Q_p H_k^T \right|\right|_{\mathrm{F}} .$$

To solve this we simply vectorize all the matrices involved. Let $\mathrm{vec}(C_k)$ denote the vector made by concatenating the columns of $C_k$. We are looking for the least squares solution of

$$A_k q = \sum_p q_p \mathrm{vec}(H_{k+1}F_k Q_p H_k^T) \approx \mathrm{vec}(C_k)$$

where the $p$th column of $A_k$ is $\mathrm{vec}(H_{k+1}F_k Q_p H_k^T)$. We can then find the least squares solution $q = A_k^\dagger \mathrm{vec}(C_k)$ and form the estimated matrix $Q_k^e$.

In the applications section we will consider two particular parameterizations of $Q_k^e$. The first is simply a diagonal parameterization using $n$ matrices $(Q_p) = E_{pp}$, where $E_{ij}$ is the elementary matrix whose only nonzero entry is 1 in the $ij$ position. The second parameterization is a block constant structure which will allow us to solve for $Q_k^e$ in the case of a sparse observation. For the block constant structure we choose a number $b$ of blocks which divides $n$ and then we form $b^2$ matrices $\{Q_{(p,r)}\}_{p,r=1}^{b}$ where $(Q_{(p,r)}) = \sum_{l,m=1}^{n/b} E_{pn/b+l,rn/b+m}$. Thus each matrix $Q_{(p,r)}$ consists of a $n/b \times n/b$ submatrix which is all ones, and they sum to a matrix whose entries are all ones.

Note that it is very important to choose matrices $Q_p$ which complement the observation. For example, if the observations are sparse then the matrix $H_k$ will have rows which are all zero. The block constant parameterization naturally interpolates from nearby sites to fill the unobserved entries.

## 5.4 Application to the Lorenz model

In this section we demonstrate the adaptive EnKF by applying it to the Lorenz96 model [74, 75], a nonlinear system which is known to have chaotic attractors and thus provides a suitably challenging testbed. We will show that our method recovers the correct covariance matrices for the observation and system noise. Next we will address the role of the stationarity parameter $\tau = 1/\delta$ and demonstrate how this parameter allows our adaptive EnKF to track changing covariances. Finally, we demonstrate the ability of the adaptive EnKF to compensate for model error by automatically tuning the system noise.

The Lorenz96 model is an $N$-dimensional ODE given by

$$\frac{dx^i}{dt} = -x^{i-2}x^{i-1} + x^{i-1}x^{i+1} - x^i + F \tag{5.6}$$

where $x = [x^1(t), \ldots, x^N(t)]$ is a vector in $\mathbb{R}^N$ and the superscript on $x^i$ (considered modulo N) refers to the $i$-th vector coordinate. The system is chaotic for parameters such as $N = 40$ and $F = 8$, the parameters used in the forthcoming examples. To realize the Lorenz96 model as a system of the form (5.1), we consider $x = x_k$ to be the state in $\mathbb{R}^N$ at time step $k$. We define $f(x_k)$ to be the result of integrating the system (5.6) with initial condition $x_k$ for a single time step of length $\Delta t = 0.05$. For simplicity we took the observation function $h$ to be the identity function except in Example 5.4.3 below, where we examine a lower dimensional observation. When simulating the system (5.1) to generate data for our examples we generated the noise vector $\omega_{k+1}$ by multiplying a vector of $N$ independent standard normal random numbers by the symmetric positive definite square root of the system noise covariance matrix $Q$. Similarly, we generated $\nu_{k+1}$ using the square root of the observation noise covariance matrix $R$.

In the following examples we will demonstrate how the proposed adaptive EnKF is able to improve state estimation for the Lorenz96 model. In all of these examples we will use the root mean squared error (RMSE) between the estimated state and the true state as a measure of filter performance. The RMSE will always be caluculated over all $N = 40$ observation sites of the Lorenz96 system. In order to show how the error evolves as the filter runs, we will often plot RMSE against filter steps by averaging over a window of 1000 filter steps centered at the indicated filter step. Since we are using our algorithm to estimate the matrices $Q$ and $R$ we will also be interested in how close our estimates $Q_k^{\text{filt}}$ and $R_k^{\text{filt}}$ are. Since we are interested in recovering the entire matrices $Q$ and $R$ we use RMSE to measure the error $Q - Q_k^{\text{filt}}$ and $R - R_k^{\text{filt}}$. The RMSE in this case refers to the square root of the average of the squared errors over all the entries in the matrices.

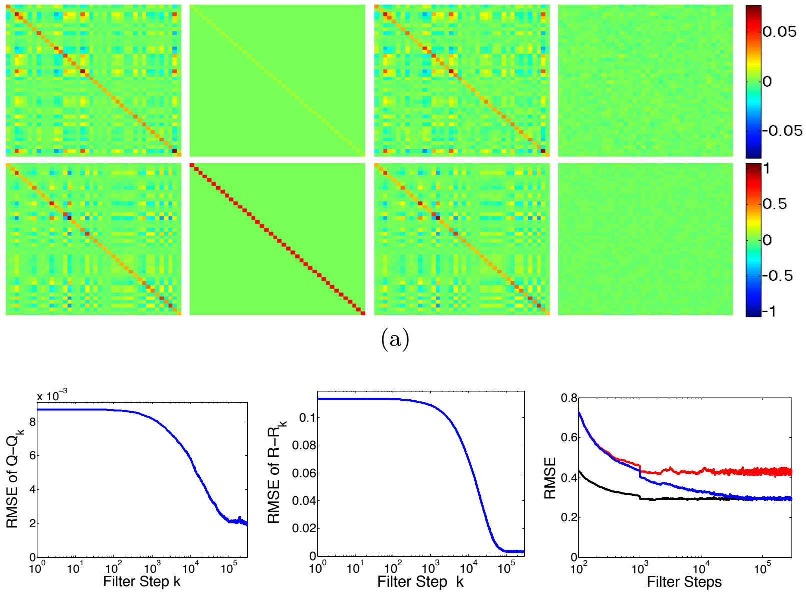
The EnKF uses an ensemble of state vectors to represent the current state information. In examples 5.4.1 through 5.4.4 below we will use the unscented version of the EnKF [77,78]. For simplicity, we do not propagate the mean ($\kappa = 0$ in [55]) or use any scaling. When implementing an EnKF, the ensemble can be integrated forward in time and observed using the nonlinear equations (5.1) without any explicit linearization. Our augmented Kalman update requires the local linearizations $F_k$ and $H_{k+1}$. While these are assumed available in the extended Kalman filter, for an EnKF these linear transformations must be found indirectly from the ensembles. Let $E_k^a$ be a matrix containing the ensemble perturbation vectors (the centered ensemble vectors) which represents the prior state estimate and let $E_{k+1}^f$ be the forecast perturbation ensemble which results from applying the nonlinear dynamics to the prior ensemble. We define $F_k$ to be the linear transformation that best approximates the transformation from the prior perturbation ensemble to the forecast perturbation ensemble. Similarly, to define $H_{k+1}$ we use the inflated forecast perturbation ensemble $E^x_{k+1}$ and the ensemble $E^y_{k+1}$ which results from applying the nonlinear observation to forecast ensemble. To summarize, in the context of an ensemble Kalman filter, we define $F_k$ and $H_{k+1}$ by

$$\begin{aligned} F_k &= E^f_{k+1}(E^a_k)^\dagger \\ H_{k+1} &= E^y_{k+1}(E^x_{k+1})^\dagger \end{aligned} \tag{5.7}$$

where $^\dagger$ indicates the pseudo-inverse. There are many methods of inflating the forecast perturbation ensemble from $E^f_{k+1}$. In the examples below we always use the additive inflation factor $Q^{\text{filt}}_k$ by finding the covariance matrix $P^f_{k+1}$ of the forecast ensemble, forming $P^x_{k+1} = P^f_{k+1} + Q^{\text{filt}}_k$, and then taking the positive definite square root of $P^x_{k+1}$ to form the inflated perturbation ensemble $E^x_{k+1}$.

**Example 5.4.1** (Finding the noise covariances)**.** In the first example, both the $Q$ and $R$ matrices were generated randomly (constrained to be symmetric and positive definite) and the discrete-time Lorenz96 model was simulated for twenty thousand time steps. We then initialized diagonal matrices $Q^{\text{filt}}_k$ and $R^{\text{filt}}_k$ as shown in Figure 5.2, and applied the standard EnKF to the simulated data. The windowed RMS error is shown in the red curve of Figure 5.2(d). Note that the error initially decreases but then quickly stabilizes as the forecast covariance given by the EnKF converges to its limiting behavior.

Next we applied the adaptive EnKF on the same simulated data using $Q^{\text{filt}}_k$ and $R^{\text{filt}}_k$ as our initial guess for the covariance matrices. In Figure 5.2(a)-(c) we compare the true $Q$ matrix to the initial diagonal $Q^{\text{filt}}_k$ and the final $Q^{\text{filt}}_k$ estimate produced by the adaptive EnKF. Here, the adaptive EnKF recovers the complex covariance structure of the system noise in a 40-dimensional system. Moreover, the resulting RMS error, shown by the blue curve in Figure 5.2(d), shows a considerable improvement. This example shows that for



(a)

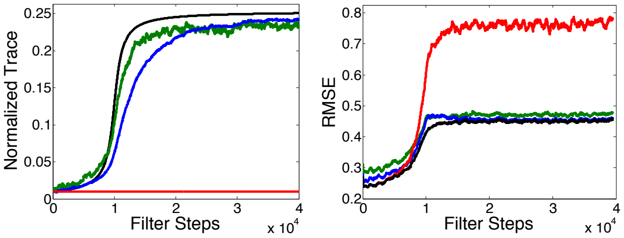
(b) (c) (d)

Figure 5.2: We show the long term performance of the adaptive EnKF by simulating Lorenz96 for 300000 steps and running the adaptive EnKF with stationarity $\tau = 20000$. (a) First row, left to right: true $Q$ matrix used in the Lorenz96 simulation, the initial guess for $Q_k^{\mathrm{filt}}$ provided to the adaptive filter, the final $Q_k^{\mathrm{filt}}$ estimated by the adaptive filter, and the final matrix difference $Q - Q_k^{\mathrm{filt}}$. The second row shows the corresponding matrices for $R$; (b) RMSE of $Q - Q_k^{\mathrm{filt}}$ as $Q_k^{\mathrm{filt}}$ is estimated by the filter; (c) RMSE of $R - R_k^{\mathrm{filt}}$ as $R_k^{\mathrm{filt}}$ is estimated by the filter; (d) comparison of windowed RMSE vs. number of filter steps for the conventional EnKF run with the true $Q$ and $R$ (black, lower trace), and the conventional EnKF run with the initial guess matrices (red, upper trace), and our adaptive EnKF initialized with the guess matrices (blue, middle trace).

Gaussian white noise model and observation errors, our adaptive scheme can be used with an EnKF to recover a randomly generated covariance structure even for a strongly nonlinear model.

Figure 5.2 shows that the improvement in state estimation builds gradually as the adaptive EnKF converges to the true values of the covariance matrices $Q$ and $R$. The speed of this convergence is determined by the parameter $\tau$, which also determines the accuracy of the final estimates of the covariance matrices. The role of the stationarity constant $\tau$ will be explored in the next example.

**Example 5.4.2** ( Tracking changing noise levels)**.** To demonstrate the role of $\tau = 1/\delta$ and to illustrate the automatic nature of the adaptive EnKF, we consider a Lorenz96 system where both $Q$ and $R$ are multiples of the identity matrix, with $R$ constant and $Q$ varying



(a) (b)

Figure 5.3: A Lorenz96 data set with slowly varying $Q$ is produced by defining the system noise covariance matrix $Q_k$ as a multiple of the identity matrix, with the multiple changing in time. (a) Trace of $Q_k$ (black) and $Q_k^{\text{filt}}$ for the EnKF (red) compared to the trace of $Q_k^{\text{filt}}$ (normalized by $N = 40$) for the Adaptive EnKF at stationarity levels $\tau = 500$ (green) and $\tau = 2000$ (blue). (b) Comparison of the RMSE in state estimation for the EnKF with fixed $Q_k^{\text{filt}}$ (red) and the Adaptive EnKF with stationarity $\tau = 500$ (green) and $\tau = 2000$ (blue). The black curve represents an oracle EnKF which is given the correct covariance matrix $Q$ at each point in time.

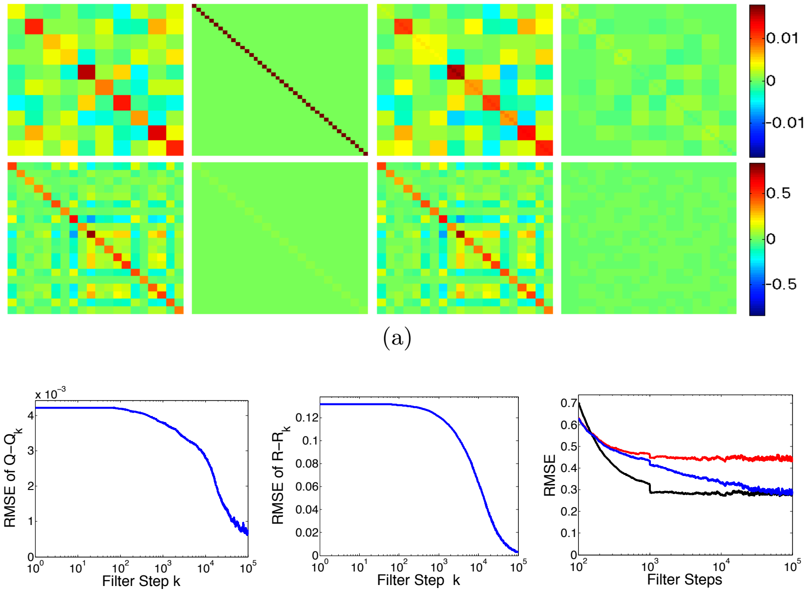
in time. The trace of $Q$ (normalized by the state dimension $N = 40$) is shown as the black curve in Figure 5.3(a) where it is compared to the initial $Q$ (shown in red) and the estimates of $Q$ produced by the adaptive EnKF with $\tau = 500$ (shown in green) and $\tau = 2000$ (shown in blue). Note that when $\tau$ is smaller the adaptive filter can move more quickly to track the changes of the system, however, when the system stabilizes the larger value of $\tau$ is more stable and gives a better approximation of the stable value of $Q$.

Next we examine the effect of $\tau$ on the RMS error of the state estimates. Figure 5.3(b) shows that leaving the value of $Q_k^{\text{filt}}$ equal to the initial value of $Q$ leads to a large increase in RMSE for the state estimate, while the adaptive EnKF can track the changes in $Q$. We can naturally compare our adaptive EnKF to an "oracle" EnKF which is given the exact value of $Q$ at each point in time; this is the best case scenario represented by the black curves in Figure 5.3. Again we see that a small $\tau$ results in a smaller peak deviation from the oracle, but the higher stationarity constant $\tau$ tracks the oracle better when the underlying $Q$ is not changing quickly. Thus the parameter $\tau$ trades adaptivity (lower $\tau$) for accuracy in the estimate (higher $\tau$).

**Example 5.4.3** (Sparse observation)**.** In this example we examine the effect of a low-dimensional observation on the adaptive EnKF. As explained in Section 5.3, Mehra uses the stationarity of the Kalman filter for linear problems to build a special adaptive filter for problems with rank deficient observations. However, our current version of the adaptive EnKF cannot find the full $Q$ matrix when the observation has lower dimensionality than the state vector (possible solutions to this open problem are considered in Section 5.5). In Section 5.3 we presented a special algorithm that parameterized the $Q$ matrix as a linear combination of fixed matrices reducing the required dimension of the observation..

To demonstrate this form of the adaptive EnKF, we use a sparse observation. We first observe 20 sites equally spaced among the 40 total sites (below we will consider observing only 10 sites). In this example the observation $y_k$ is 20-dimensional while the state vector $x_k$ is 40-dimensional, giving a rank-deficient observation. We note that because the observation is sparse the linearization, $H$, of the observation will have rows which are all zeros. Solving for $Q_k^{\mathrm{e}}$ requires inverting $H$ so we cannot use the diagonal parameterization of $Q_k^{\mathrm{filt}}$, and instead we use the block constant parameterization which allows the $Q_k^{\mathrm{filt}}$ values to be interpolated from nearby sites. In order to check that the adaptive EnKF can still find the correct covariance matrices we simulated 100000 steps of the discretized Lorenz96 system observing only 20 evenly spaced sites. We take the true $Q$ to have a block constant structure, and we take the true $R$ to be a randomly generated symmetric positive definite matrix as shown in Figure 5.4. Since our algorithm will impose a block constant structure on $Q_k^{\mathrm{filt}}$, we chose a block constant matrix for $Q$ so that we could confirm that the correct entry values were recovered.

To test the block constant parameterization of the adaptive EnKF we chose initial matrices $Q_k^{\mathrm{filt}}$ and $R_k^{\mathrm{filt}}$ which were diagonal with entries 0.02 and 0.05 respectively. We used a large stationarity of $\tau = 15000$ which requires longer to converge but gives a better final approximation of $Q$ and $R$, this is why the Lorenz96 system was run for 100000 steps. Such a large stationarity and long simulation was chosen to illustrate the long term behavior of the adaptive EnKF. The estimates of $Q_k^{\mathrm{filt}}$ were parameterized with a $10 \times 10$ block constant structure ($b = 10$ using the method of Section 5.3). In Figure 5.4 we compare the true $Q$



(a)

(b) (c) (d)

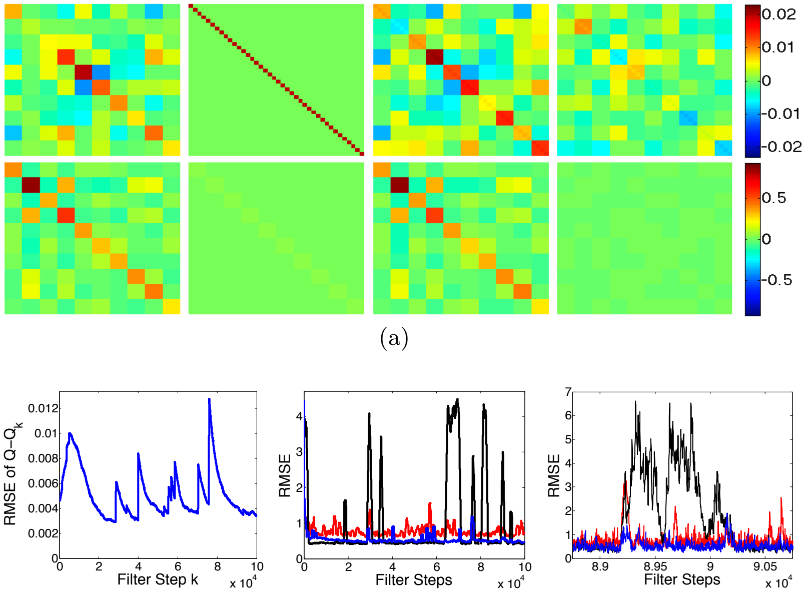
Figure 5.4: We apply the adaptive EnKF with a sparse observation by only observing every other site (20 total observed sites) of a Lorenz96 simulation with 100000 steps. The $Q_k^{\mathrm{filt}}$ matrix is assumed to be constant on 4x4 sub-matrices and the true $Q$ used in the simulation is given the same block structure. (a) First row, left to right: true $Q$ matrix used in the Lorenz96 simulation, the initial guess for $Q_k^{\mathrm{filt}}$ provided to the adaptive filter, the final $Q_k^{\mathrm{filt}}$ estimated by the adaptive filter, and the final matrix difference $Q - Q_k^{\mathrm{filt}}$. The second row shows the corresponding matrices for $R$; (b) RMSE of $Q - Q_k^{\mathrm{filt}}$ as $Q_k^{\mathrm{filt}}$ is estimated by the filter; (c) RMSE of $R - R_k^{\mathrm{filt}}$ as $R_k^{\mathrm{filt}}$ is estimated by the filter; (d) comparison of windowed RMSE vs. number of filter steps for the conventional EnKF run with the true $Q$ and $R$ (black, lower trace), and the conventional EnKF run with the initial guess matrices (red, upper trace), and our adaptive EnKF initialized with the guess matrices (blue, middle trace).

and $R$ matrices to the initial guesses and the final estimates of our adaptive EnKF. This example shows that, even in the case of a rank deficient observation, the adaptive EnKF can recover an arbitrary observation noise covariance matrix and a parameterized system noise covariance matrix.

Observations in real applications can be very sparse, so we now consider the case when only 10 evenly spaced sites are observed. Such a low dimensional observation makes state estimation very difficult as shown by the increased RMSE in Figure 5.5 compared to Figure 5.4. Even more interesting is that we now observe filter divergence, where the state estimate trajectory completely loses track of the true trajectory. Intuitively this is much more likely when fewer sites are observed, and the effect is shown in Figure 5.5(c) by sporadic periods of very high RMSE. Filter divergence occurs when the variance of $Q_k^{\text{filt}}$ is small, implying overconfidence in the state estimate and the resulting forecast. In Figure 5.5 this is clearly shown since filter divergence occurs only when the true matrix $Q$ is used in the filter, whereas both the initial guess and the final estimate produced by the adaptive EnKF are $i$nflated. The fact that the filter diverges when provided with the true $Q$ and $R$ matrices illustrates how this example pushes the boundaries of the EnKF. However, our adaptive EnKF produces an inflated version of the true $Q$ matrix as shown in Figure 5.5 which not only avoids filter divergence, but also significantly reduces RMSE relative to the initial diagonal guess matrices. This example illustrates that the breakdown of the assumptions of the EnKF (local linearizations, Gaussian distributions) can lead to assimilation error even when the nonlinear dynamics are known. In the presence of this error, our adaptive EnKF must be interpreted as an inflation scheme and we judge it by its performance in terms of RMSE rather than recovery of the underlying $Q$.

**Example 5.4.4** (Compensating for model error)**.** The goal of this example is to show that the covariance of the system noise is a type of additive inflation and thus is a natural place to compensate for model error. Intuitively, increasing $Q_k^{\text{filt}}$ increases the gain, effectively placing more confidence in the observation and less on our forecast estimate. Thus, when we are less confident in our model, we would naturally want to increase $Q_k^{\text{filt}}$. Following this line of thought, it seems natural that if $Q_k^{\text{filt}}$ is sub-optimally small, the model errors would manifest themselves in poor state estimates and hence large innovations. In this example, the adaptive EnKF automatically inflates $Q_k^{\text{filt}}$ based on the observed innovations, leading to significantly improved filter performance.

To illustrate this effect, we fixed the model used by the adaptive EnKF to be the Lorenz96 model from (5.6) with $N = 40$ and $F = 8$, and changed the system generating the data. The sample trajectory of the Lorenz96 system was created with 20000 time steps. For the first 10000 steps we set $F = 8$. We then chose $N = 40$ fixed random values of $F^i$, chosen from a distribution with mean 8 and standard deviation 4. These new values were



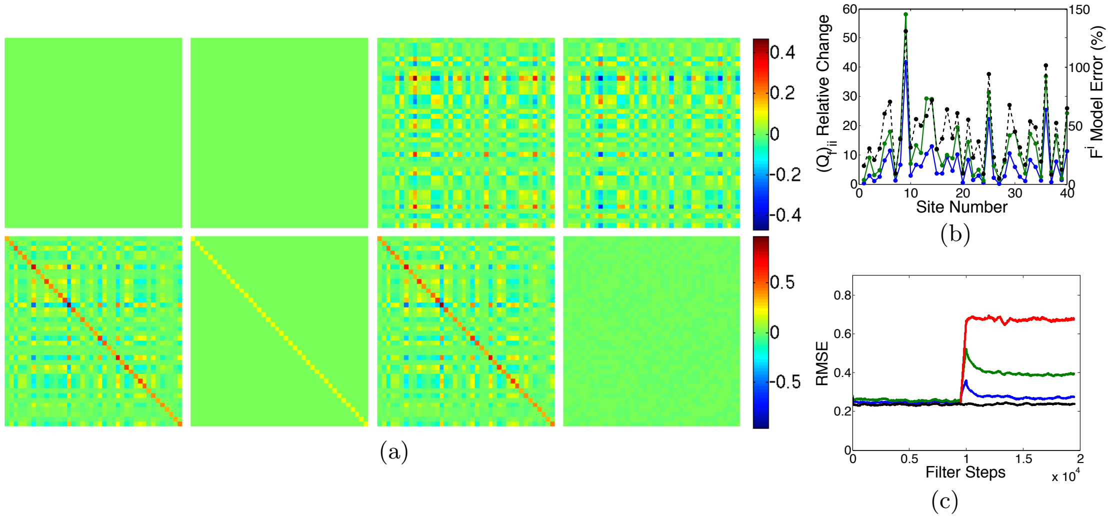
(a)

(b) (c) (d)

Figure 5.5: We illustrate the effect of extremely sparse observations by only observing every forth site (10 total observed sites) of the Lorenz96 simulation, the $Q_k^{\rm filt}$ matrix is assumed to be constant on 4x4 sub-matrices and the true $Q$ used in the simulation is given the same block structure. (a) First row, left to right: true $Q$ matrix used in the Lorenz96 simulation, the initial guess for $Q_k^{\rm filt}$ provided to the adaptive filter, the final $Q_k^{\rm filt}$ estimated by the adaptive filter, and the final matrix difference $Q - Q_k^{\rm filt}$. The second row shows the corresponding matrices for $R$; (b) RMSE of $Q - Q_k^{\rm filt}$ as the adaptive EnKF is run; (c) comparison of windowed RMSE vs. number of filter steps for the conventional EnKF run with the true $Q$ and $R$ (black, lower trace), and the conventional EnKF run with the initial guess matrices (red, upper trace), and our adaptive EnKF initialized with the guess matrices (blue, middle trace); (d) Enlarged view showing filter divergence, taken from (c). Note that the conventional EnKF occasionally diverges even when provided the true $Q$ and $R$ matrices. The $Q_k^{\rm filt}$ found by the adaptive filter is automatically inflated relative to the true $Q$ which improves filter stability as shown in (c) and (d).

used for the last 10000 steps of the simulation. Thus, when running our filter we would have the correct model for the first 10000 steps but a significantly incorrect model for the last 10000 steps.

We first ran a conventional EnKF on this data and found that the RMSE for the last 10000 steps was approximately 180% greater than an oracle EnKF which was given the new parameters $F^i$. Next we ran our adaptive EnKF and found that it automatically increased the variance of $Q_k^{\rm filt}$ in proportion to the amount of model error. In Figure 5.6 we show how the variance of $Q_k^{\rm filt}$ was increased by the adaptive filter; note that the increase in



(a)

(b)

(c)

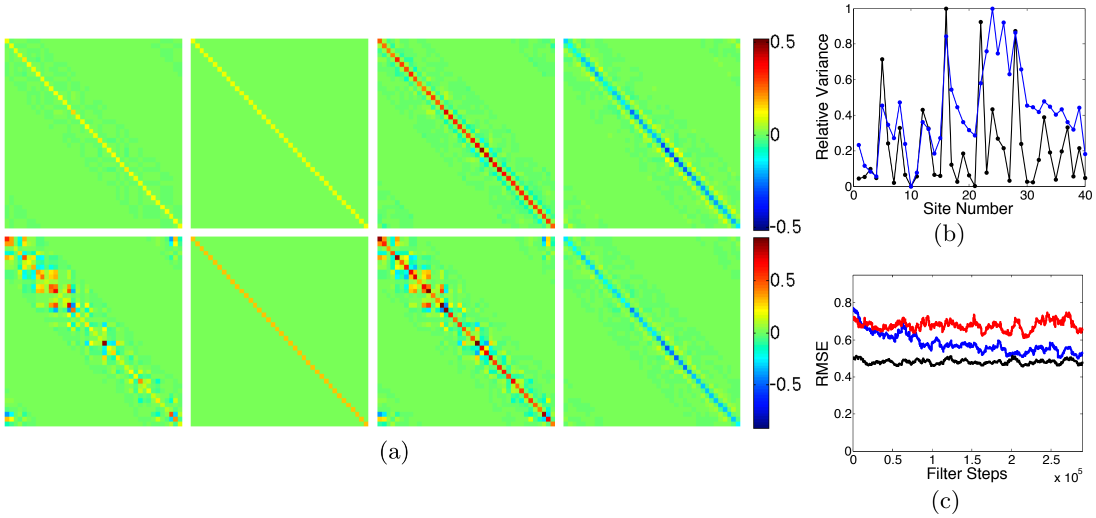
Figure 5.6: For the first 10000 filter steps the model is correct and then the underlying parameters are randomly perturbed at each site. The conventional EnKF is run with the initial true covariances $Q_k^{\text{filt}} = Q$ and $R_k^{\text{filt}} = R$; the adaptive EnKF starts with the same values but it automatically increases the system noise level ($Q_k^{\text{filt}}$) to compensate for the model error resulting in improved RMSE. (a) First row, left to right: true $Q$ matrix used in the Lorenz96 simulation, the initial guess for $Q_k^{\text{filt}}$ provided to the adaptive filter, the final $Q_k^{\text{filt}}$ estimated by the adaptive filter, and the final matrix difference $Q - Q_k^{\text{filt}}$. The second row shows the corresponding matrices for $R$; (b) Model error (black, dotted curve) measured as the percent change in the parameter $F^i$ at each site compared to the relative change in the corresponding diagonal entries of $Q_k^{\text{filt}}$ found with the adaptive EnKF (blue, solid curve), diagonal (green, solid curve). (c) Results of the adaptive EnKF (blue) compared to conventional EnKF (red) on a Lorenz96 data set in the presence of model error. The green curve is an adaptive EnKF where $Q_k^{\text{filt}}$ is forced to be diagonal and the black curve shows the RMSE of an oracle EnKF which is provided with the true underlying parameters $F^i$ for both halves of the simulation.

variance is highly correlated with the model error measured as the percentage change in the parameter $F^i$ at the corresponding site. Intuitively, a larger error in the $F^i$ parameter would lead to larger innovations thus inflating the corresponding variance in $Q_k^{\text{filt}}$. However, note that when $Q_k^{\text{filt}}$ was restricted to be diagonal the adaptive filter required greater increase (shown in Figure 5.6(b)) and gave significantly worse state estimates (as shown in Figure 5.6(c)). Thus, when $Q_k^{\text{filt}}$ was not restricted to be diagonal, the adaptive filter was able to find complex new covariance structure introduced by the model error. This shows that the adaptive filter is not simply turning up the noise arbitrarily but is precisely tuning the covariance of the stochastic process to compensate for the model error. This automatic tuning meant that the final RMSE of our adaptive filter increased less than 15% as a result of the model error.

**Example 5.4.5** (Adaptive version of the LETKF)**.** The goal of this example is to show that our algorithm can naturally integrate into a localization scheme. The unscented version of the EnKF used in the previous examples requires large ensemble sizes, which are impractical for many applications. Localization is a general technique, implemented in various forms, that assumes the state vector has a spatial structure, allowing the Kalman update to be applied locally rather than globally. Some localization schemes have been shown to allow reduced ensemble size.

The local ensemble transform Kalman filter (LETKF) of [79–81] is a localization technique that is particularly simple to describe. The local update avoids inverting large covariance matrices, and also implies that we only need enough ensemble members to track the local dynamics for each local region. For simplicity we used the algorithm including the ensemble transform given in equation 41 of [79]. We performed the local Kalman update at each site using a local region with 11 sites (l=5) and we used a global ensemble with 22 members, compared to 80 ensemble members used in the unscented EnKF.

Building an adaptive version of the LETKF is fairly straightforward. A conventional Kalman update is used to form the local analysis state and covariance estimates. We design an adaptive LETKF by simply applying our iteration, given by equation (5.5), after the local Kalman update performed in each local region. We will estimate 40 separate pairs of $11 \times 11$ matrices $Q_k^{i,\text{filt}}$ and $R_k^{i,\text{filt}}$. This implies that some entries in the full $Q_k^{\text{filt}}$ (those far from the diagonal) have no representatives in these local $Q_k^{i,\text{filt}}$. Moreover, the other entries in the full $Q_k^{\text{filt}}$ are represented by entries in several $Q_k^{i,\text{filt}}$ matrices. In Figure 5.7 we combine the final estimates of $Q_k^{i,\text{filt}}$ into a single $40 \times 40$ $Q_k^{\text{filt}}$ matrix by averaging all the representatives of each entry. It would also be possible to form this global $Q_k^{\text{filt}}$ matrix at each step, however, the LETKF often allows many choices for forming the global results from the local, and our objective here is only to show that our adaptive iteration can be integrated into a localized data assimilation method.



(a)

(b)

(c)

Figure 5.7: We compare the conventional and adaptive LETKF on a simulation of 300000 steps of Lorenz96 (a) First row, left to right: benchmark $Q + .1I_{40}$ matrix where $Q$ is was the matrix used in the Lorenz96 simulation, the initial guess for $Q_k^{\text{filt}}$ provided to the adaptive filter, the final $Q_k^{\text{filt}}$ estimated by the adaptive filter, and the final matrix difference $Q - Q_k^{\text{filt}}$. The second row shows the corresponding matrices for $R$ (leftmost is the true $R$); (b) the variances from the diagonal entries of the true $Q$ matrix (black, rescaled to range from zero to one) and the those from the final global estimate $Q_k^{\text{filt}}$ produced by the adaptive LETKF (blue, rescaled to range from zero to one) (c) Results of the adaptive LETKF (blue) compared to conventional LETKF with the diagonal covariance matrices (red) and the conventional LETKF with the benchmark covariances $Q_k^{\text{filt}} = Q + .1I_{40}$ and $R_k^{\text{filt}} = R$ (black).

Interestingly, the conventional LETKF requires significant inflation to prevent filter divergence. For example, the noise variances in Figure 5.7 are comparable to Example 5.4.1, however the LETKF often diverges when the diagonal entries of $Q_k^{\text{filt}}$ are less than 0.05. So for this example we use $Q_k^{\text{filt}} = Q + (0.1)I_{40}$ and $R_k^{\text{filt}} = R$ as the benchmark for all comparisons, since this represents a filter which was reasonably well-tuned when the true $Q$ and $R$ were both known. To represent that case when the true $Q$ and $R$ are unknown we choose diagonal matrices with the same average covariance as $Q+(0.1)I_{40}$ and $R$ respectively. For the conventional LETKF, we form the local $Q_k^{i,\text{filt}}$ and $R_k^{i,\text{filt}}$ by simply taking the 11 × 11 sub-matrices of $Q_k^{\text{filt}}$ and $R_k^{\text{filt}}$ which are given by the local indices. In Figure 5.7 we show that the adaptive LETKF significantly inflates both $Q_k^{\text{filt}}$ and $R_k^{\text{filt}}$, and it recovers much of the structure of both $Q$ and $R$, and improves the RMSE, compared to the conventional LETKF using the diagonal $Q_k^{\text{filt}}$ and $R_k^{\text{filt}}$.

While this example shows that our iteration can be used to make an adaptive LETKF, we found that the adaptive version required a much higher stationarity ($\tau = 50000$ in Figure 5.7). We speculate that this is because local regions can experience specific types of dynamics for extended periods which bias the short term estimates and by setting a large stationarity we remove these biases by averaging over a larger portion of state space. We also found that the LETKF could have large deviations from the true state estimate similar to filter divergence but on shorter time scales. These deviations would cause large changes in the local estimates $Q_k^{i,\text{filt}}$ which required us to impose a limit of on the amount an entry of $Q_k^{i,\text{filt}}$ could change in a single time step of 0.05. It is possible that a better tuned LETKF or an example with lower noise covariances would not require these ad hoc adjustments. Since large inflations are necessary simply to prevent divergence, this adaptive version is a first step towards the important goal of optimizing the inflation.

## 5.5 Discussion and outlook

A central feature of the Kalman filter is that the innovations derived from observations are only used to update the mean state estimate, and the covariance update does not incorporate the innovation. This is a remnant of the Kalman filter's linear heritage, in which observation, forecast and analysis are all assumed Gaussian. In any such scenario, when a Gaussian observation is assimilated into a Gaussian prior, the posterior is Gaussian and independent of observation. Equation (5.2) explicitly shows that the Kalman update of the covariance matrix is independent of the innovation.

Any scheme for adapting the Kalman filter to nonlinear dynamics must consider the possibility of non-Gaussian estimates, which in turn demands a reexamination of this independence. But the stochastic nature of the innovation implies that any attempt to use this information directly will be extremely sensitive to particular noise realizations. In this chapter, we have introduced an augmented Kalman update in which the innovation is allowed to affect the covariance estimate indirectly, through the filter's estimate $Q^{\text{filt}}$ of the system noise. We envision this augmented version to be applicable to any nonlinear generalization of the Kalman filter, such as the EKF and EnKF. Application to the EKF is straightforward, and we have shown in equation (5.7) how to implement the augmented equations into an ensemble Kalman update.

The resulting adaptive EnKF is an augmentation of the conventional EnKF, to allow the system and observation noise covariance matrices to be automatically estimated. This removes an important practical constraint on filtering for nonlinear problems, since often the true noise covariance matrices are not known *a priori*. Using incorrect covariance matrices will lead to sub-optimal filter performance, as shown in Section 5.2, and in extreme cases can even lead to filter divergence. The adaptive EnKF uses the innovation sequence produced by the conventional equations, augmented by our additional equations developed in Section 5.3, to estimate the noise covariances sequentially. Thus, it is easy to adopt for existing implementations, and as shown in Section 5.4 it has a significant performance advantage over the conventional version. Moreover, we have shown in Section 5.4 that the adaptive EnKF can adjust to non-stationary noise covariances and even compensate for significant modeling errors.

The adaptive EnKF introduced here raises several practical and theoretical questions. A practical limitation of our current implementation is the requirement that the rank of the linearized observation $H_k$ is at least the dimension of the state vector $x_k$. In Section 5.3 we provide a partial solution which constrains the $Q$ estimation to have a reduced form which must be specified in advance. However, in the context of the theory of embedology, developed in [32], it may be possible to modify our algorithm to recover the full $Q$ matrix from a low-rank observation. Embedology shows that for a generic observation, it is possible to reconstruct the state space using a time-delay embedding. Motivated by this theoretical result, we propose a way to integrate a time-delay reconstruction into the context of a Kalman filter and discuss the remaining challenges of such an approach.

In order to recover the full system noise covariance matrix in the case of rank deficient observation we propose an augmented observation formed by concatenating several iterations of the dynamics,

$$\hat{y}_k^f = \hat{h}(x_k^f) = (h(x_k^f), h \circ f(x_k^f), ..., h \circ f^M(x_k^f))$$

which will be compared to time-delayed observation vector $\hat{y}_k = (y_k, y_{k+1}, ..., y_{k+M})$. For a generic observation $h$, when the system is near an attractor of box counting dimension $N$, our augmented observation $\tilde{h}$ is generically invertible for $M > 2N$ [32]. We believe that this augmented observation will not only solve the rank deficient observation problem, but may also improve the stability of the Kalman filter by including more observed information in each Kalman update. However, a challenging consequence of this technique is that the augmented observation is influenced by both the dynamical noise and the observation noise realizations. Thus extracting the noise covariances from the resulting innovations may be non-trivial.

An important remaining theoretical question is to develop proofs that our estimated $Q$ and $R$ converge to the correct values. For $R$ this is a straightforward claim, however the interpretation of $Q$ for nonlinear dynamics is more complex. This is because the Kalman update equations assume that at each step the forecast error distribution is Gaussian. Even if the initial prior is assumed to be Gaussian, the nonlinear dynamics will generally change the distribution, making this assumption false. Until recently it was not even proved that the EnKF tracked the true trajectory of the system. It is likely that the choice of $Q_k^{\text{filt}}$ for an EnKF will need to account for both the true system noise $Q$ as well as the discrepancy between the true distribution and the Gaussian assumption. This will most likely require an additional increase in $Q_k^{\text{filt}}$ above the system noise covariance $Q$.

We believe that the eventual solution of this difficult problem will involve computational approximation of the Lyapunov exponents of the model as the filter proceeds. Such techniques may require $Q_k^{\text{filt}}$ to enter non-additively or even to vary over the state space. Improving filter performance for nonlinear problems may require reinterpretation of the classical meanings of $Q$ and $R$ in order to find the choice that leads to the most efficient shadowing. In combination with the results of [73], this would realize the long-sought nonlinear analogue of the optimal Kalman filter.

# Appendix A: The DMDC algorithm

When the spatial extent of the dynamics is restricted, the intrinsic dimensionality will be small and thus a diffusion map may be applied directly to the full images simply by taking the $L^2$ distance between delay embedded images in $\mathbb{R}^{r(s+1)}$ constructed in section 3.5. However, there are several computational considerations that can dramatically effect this analysis. By careful application of sparse data structures and extensive parallelization of the algorithm we have found that the diffusion map can be effectively applied to video data using consumer grade hardware.

As in section 3.5, for each image we form a high-dimensional state vector by concatenating as many of the previous images as possible in analogy to the classical time-delay embedding. Since this creates an extremely high dimensional vector, we need a preliminary reduction of the dimension to make the $k$-nearest neighbor algorithm feasible. However, we need to preserve the geometry of the embedding, so the initial reduction will be to an intermediate dimension on the order of $10^3$, much larger than the latent dimension (order of 10) but much smaller than the delay embedding dimension (order of $10^6$). The preliminary dimensionality reduction is simply a linear projection from the delay embedding space to the intermediate space. This projection is chosen by generating a matrix of Gaussian random entries which is then made orthonormal using a QR-decomposition (the $R$ matrix is discarded). At this point we are ready to apply the diffusion map to the data represented in the intermediate space.

The full diffusion map algorithm is impractical because it requires the full matrix of Euclidean distances between all pairs of state vectors and this requires too much memory. However, in [5] it was shown that for sufficiently high $k$, the sparse matrix containing the $k$-nearest neighbors of each state vector is sufficient. This is still a computationally intensive algorithm since the state vectors are too high-dimensional to apply kd-tree or ball-tree data structures. However, the $k$-nearest neighbors algorithm is highly parallelizable, and it was recently shown in [82] that inexpensive Graphics Processing Units (GPUs) could achieve a significant acceleration of the $k$-nearest neighbor algorithm. In our test we saw up to 200-times speedup using the GPU over single processor CPU applications. The remainder of the diffusion map algorithm is straightforward and is detailed in the DMDC algorithm summary below.

The parameter $\epsilon$ is the diameter of the neighborhood that is used to approximate the tangent plane to the manifold. The results of Section 3.3 involve the limit as $\epsilon \to 0$, however for the discrete approximations to converge we require the amount of data to go to infinity. In [2], the number of sample points is required to grow faster than $\epsilon^{n/4-1/2}$, where $n$ is the dimension of the underlying manifold. However, in most cases we have a fixed data set that we wish to analyze, so a limiting requirement on $\epsilon$ is not very useful.

There are many practical ways of choosing $\epsilon$. The theoretical approach in [15] may become useful if this algorithm can be adapted into an iterative form which continually updates as new data becomes available. Currently we assume that the data set is fixed and therefore we take a pragmatic approach to choosing $\epsilon$. Since we are assuming that the local structure can be well represented by the $k$-nearest neighbors, this implies that a transition probability to any state beyond the $k$-nearest neighbors should be near zero. On the other hand, if the transition probabilities decay too quickly the states will quickly become isolated and the matrix will not have numerically stable eigenvalues. To balance these effects we have had very good results with $\epsilon = \text{mean}_i\{d(i, I(i, k_{\text{min}}))\}$ where $I(i, k_{\text{min}})$ is the index of the $k_{\text{min}}$-th nearest neighbor to the $i$-th vector and $d(\cdot,\cdot)$ is the distance between the referenced data points. In general $k_{\text{min}}$ will be small, and we took $k_{\text{min}} = 6$ in this paper. The choice of $k_{\text{min}}$ too small will lead to $\epsilon$ being too small and the eigenvector problem will be ill posed. Choosing $k_{\text{min}}$ too large leads to $\epsilon$ being too large and the approximation of the heat kernel will be poor.

As described in section 3.3, the parameter $\alpha$ chooses the influence of the sampling density on the heat kernel constructed by diffusion maps. Intuitively $\alpha$ is a bias parameter, and for $\alpha = 1$ the heat kernel is unbiased by the sampling density. If the video represents a dynamical system that has an invariant measure, and if the sampling density can be assumed to be the same as the invariant measure, then setting $\alpha = 1/2$ will adapt the analysis to the invariant measure. In this case, the first DMDC mode, with eigenvalue $\lambda_0 = 1$ (and hence time scale $\infty$), will be an approximation of the invariant measure. In this paper we used $\alpha = 1/2$ for all the examples.

We now summarize all the steps of DMDC. Note that the algorithm works unchanged when $x_i \in \mathbb{R}^r$ is a time series in any data format where the index $i$ represents time. In this summary we assume that the $x_i$ are images in a video because many elements of DMDC have a natural interpretation in this case. Steps 1-3 build the weighted delay embedding explored in section 3.2. Steps 4-6 are an optional precompression for extremely high dimensional data (such as videos), without the precompression the $k$-nearest neighbor algorithm may be infeasible depending on the embedding dimension $r(s+1)$. Steps 7-19 are a computationally efficient version of the diffusion maps algorithm which was introduced in [2]. Finally, in steps 20-24 we use the Diffusion Mapped Delay Coordinates to separate time scales in the time series.

### Diffusion Mapped Delay Coordinates (DMDC) Algorithm

1. Let $x_i \in \mathbb{R}^r$ be the $i$-th frame of a video with $N$ frames of $r$ pixels each.
2. Choose bias parameter $\alpha \in \{1/2, 1\}$, weight $\kappa > 0$ and number of delays $s$.
3. For $i = s+1, ..., N$ form the state vector,
$$y_i = [x_i, e^{-\kappa} x_{i-1}, ..., e^{-s\kappa} x_{i-s}]^T \in \mathbb{R}^{r(s+1)}.$$
4. Generate an $m \times r(s+1)$ dimensional matrix $\pi$ of Gaussian random numbers.
5. Use a QR-decomposition to orthonormalize the rows of $\pi$ to form $\hat{\pi}$.
6. Form the compressed states $\hat{y}_i = \hat{\pi} y_i$.
7. For each $i$ find the $k$-nearest neighbors of $\hat{y}_i$ in $\mathbb{R}^m$, let their indices be $I(i,j)$ for $j = 1, ..., k$ ordered by increasing distance.
8. Form a sparse $(N-s) \times (N-s)$ matrix with $(N-s)k$ nonzero entries given by
$$d(i, I(i,j)) = ||\hat{y}_i - \hat{y}_{I(i,j)}||$$
9. Pick $k_{\min}$ as described above and set $\epsilon = \text{mean}_i\{d(i, I(i, k_{\min}))^2\}$.
10. Form the sparse matrix $\hat{d}(i, I(i,j)) = \exp\{-d(i, I(i,j))^2/\epsilon\}$.
11. Form the symmetric matrix $J = (\hat{d} + \hat{d}^T)/2$.
12. Form the diagonal normalization matrix $P_{ii} = \sum_j J_{ij}$.
13. Normalize to form Kernel matrix $K = P^{-\alpha} J P^{-\alpha}$.
14. Form the diagonal normalization matrix $Q_{ii} = \sum_j K_{ij}$.
15. Form the symmetric matrix $\hat{T} = Q^{-1/2} K Q^{-1/2}$.
16. Find the $L+1$ largest eigenvalues $a_l$ and associated eigenvectors $\xi_l$ of $\hat{T}$.
17. Compute the eigenvalues of $\hat{T}^{1/\epsilon}$ by $\lambda_l^2 = a_l^{1/\epsilon}$.
18. Compute the eigenvectors of the matrix $T = Q^{-1}K$ by $\psi_l = Q^{-1/2}\xi_l$.
19. Compute the diffusion map variables at time scale $t$ by
$$\Psi_{\alpha,t}(x_i) = [\lambda_1^t \psi_1(x_i), ..., \lambda_L^t \psi_L(x_i)]^T$$
20. The $l$-th DMDC mode $(\psi_l)_i = \psi_l(x_{i+s})$ is a time series at time scale $\frac{-1}{\log(\lambda_l)}$.
21. Form the $l$-th DMDC projection by $q_l = Q\psi_l$.
22. Form the $l$-th DMDC component by $X_l = \sum_{i=s+1}^{N} x_i (q_l)_{i-s}$, this is an image which shows which parts of the video are relevant to the $l$-th DMDC mode.
23. Choose any collection $L_0$ of indices of interesting DMDC components.
24. Construct a video by taking the $i$-th frame to be $\hat{x}_i = \sum_{l \in L_0} \lambda_l (\psi_l)_{i-s} X_l$ for $i = s+1, ..., N$. This projects the original video onto the modes of $L_0$.

# Bibliography

[1] D. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129 – 150, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1063520310000552

[2] R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comp. Harmonic Anal.*, vol. 21, pp. 5–30, 2006.

[3] D. Giannakis and A. J. Majda, "Time series reconstruction via machine learning: Revealing decadal variability and intermittency in the north pacific sector of a coupled climate model," in *Conference on Intelligent Data Understanding (CIDU)*, Mountain View, California, 2011.

[4] ——, "Nonlinear laplacian spectral analysis for time series with intermittency and low-frequency variability," *Proceedings of the National Academy of Sciences*, vol. 109, no. 7, pp. 2222–2227, 2012.

[5] R. Coifman and M. Maggioni, "Diffusion wavelets," *Appl. Comp. Harmonic Anal.*, vol. 21, no. 1, pp. 53 – 94, 2006.

[6] M. Desbrun, E. Kanso, and Y. Tong, "Discrete differential forms for computational modeling," pp. 39–54, 2006. [Online]. Available: http://doi.acm.org/10.1145/1185657.1185665

[7] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[8] M. Saerens, F. Fouss, L. Yen, and P. Dupont, "The principal components analysis of a graph, and its relationships to spectral clustering," in *Proceedings of the 15th European Conference on Machine Learning (ECML 2004). Lecture Notes in Artificial Intelligence.* Springer-Verlag, 2004, pp. 371–383.

[9] F. Fouss, A. Pirotte, and M. Saerens, "The application of new concepts of dissimilarities between nodes of a graph to collaborative filtering," 2004.

[10] A. Singer and H.-T. Wu, "Vector diffusion maps and the connection laplacian," *Communications on Pure and Applied Mathematics*, vol. 65, no. 8, pp. 1067–1144, 2012. [Online]. Available: http://dx.doi.org/10.1002/cpa.21395

[11] M. Saerens, F. Fouss, L. Yen, and P. Dupont, "The principal components analysis of a graph, and its relationships to spectral clustering," *Lecture Notes in Artificial Intelligence No. 3201, 15th European Conference on Machine Learning (ECML)*, pp. 371–383, 2004.

[12] R. Coifman, S. Lafon, B. Nadler, and I. Kevrekidis, "Diffusion maps, spectral clustering and reaction coordinates of dynamical systems," *Appl. Comp. Harmonic Anal.*, vol. 21, pp. 113–127, 2006.

[13] R. Coifman, R. Erban, A. Singer, and I. Kevrekidis, "Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps," *PNAS*, vol. 106, pp. 10 000 000–3, 2009.

[14] R. Coifman, S. Lafon, M. Maggioni, B. Nadler, and I. Kevrekidis, "Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems," *SIAM Journal for Multiscale Modeling & Simulation*, vol. 7, pp. 842–864, 2008.

[15] A. Singer, "From graph to manifold laplacian: The convergence rate," *Appl. Comp. Harmonic Anal.*, vol. 21, pp. 128–134, 2006.

[16] A. Szlam, M. Maggioni, and R. Coifman, "Regularization on graphs with function-adapted diffusion processes," *J. Mach. Learn. Res.*, vol. 9, pp. 1711–1739, 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1390681.1442788

[17] A. Singer, R. Erban, I. G. Kevrekidis, and R. Coifman, "Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps," *PNAS*, vol. 106, no. 38, pp. 16 090–16 095, 2009.

[18] M. Hein, J. yves Audibert, and U. V. Luxburg, "From graphs to manifolds - weak and strong pointwise consistency of graph laplacians," in *Proceedings of the 18th Conference on Learning Theory (COLT)*. Springer, 2005, pp. 470–485.

[19] D. Ting, L. Huang, and M. I. Jordan, "An analysis of the convergence of graph laplacians," 2010.

[20] M. Fisher, P. Schröder, M. Desbrun, and H. Hoppe, "Design of tangent vector fields," *ACM Transactions on Graphics*, vol. 27(3), 2007.

[21] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction.* Springer-Verlag New York, 2007.

[22] J. Tenenbaum, V. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2324, 2000.

[23] J. Jost, *Riemannian Geometry and Geometric Analysis.* Springer-Verlag Berlin, 2002.

[24] S. Rosenberg, *The Laplacian on a Riemannian manifold.* Cambridge University Press, 1997.

[25] J. Latschev, "Vietoris-rips complexes of metric spaces near a closed riemannian manifold," *Archiv der Mathematik*, vol. 77, no. 6, pp. 522–528, 2001.

[26] D. Bao, S. Chern, and Z. Shenk, *An Introduction to RiemannFinsler Geometry.* Springer-Verlag, 2000.

[27] D. Stroock, *An Introduction to the Analysis of Paths on a Riemannian Manifold.* American Mathematical Soc., 2005.

[28] F. Takens, “Detecting strange attractors in turbulence,” in *In: Dynamical Systems and Turbulence, Warwick, Eds. Rand, D. and Young, L.-S.*, ser. Lecture Notes in Mathematics, D. Rand and L.-S. Young, Eds. Springer Berlin / Heidelberg, 1981, vol. 898, pp. 366–381. [Online]. Available: http://dx.doi.org/10.1007/BFb0091924

[29] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, “Geometry from a time series,” *Phys. Rev. Lett.*, vol. 45, pp. 712–716, 1980. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevLett.45.712

[30] D. Aeyels, “Generic observability of differentiable systems,” *SIAM Journal on Control and Optimization*, vol. 19, no. 5, pp. 595–603, 1981. [Online]. Available: http://link.aip.org/link/?SJC/19/595/1

[31] J.-P. Eckmann and D. Ruelle, “Ergodic theory of chaos and strange attractors,” *Rev. Mod. Phys.*, vol. 57, pp. 617–656, Jul 1985. [Online]. Available: http://link.aps.org/doi/10.1103/RevModPhys.57.617

[32] T. Sauer, J. Yorke, and M. Casdagli, “Embedology,” *Journal of Statistical Physics*, vol. 65, no. 3, pp. 579–616, 1991.

[33] D. Broomhead and G. P. King, “Extracting qualitative dynamics from experimental data,” *Physica D: Nonlinear Phenomena*, vol. 20, no. 23, pp. 217 – 236, 1986. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016727898690031X

[34] T. Sauer, “Time series prediction by using delay coordinate embedding,” in *In: Time Series Prediction: Forecasting the future and understanding the past, Eds. Weigend, A.S. and Gershenfeld, N.A. Addison-Wesley*, A. S. Weigend and N. A. Gershenfeld, Eds. Harlow, UK: Addison Wesley, 1994, pp. 175–193.

[35] V. I. Oseledets, “A multiplicative ergodic theorem,” *Trans. Moscow Math. Soc.*, vol. 19, pp. 197–231, 1968.

[36] C. W. Gear, T. J. Kaper, I. G. Kevrekidis, and A. Zagaris, “Projecting to a slow manifold: Singularly perturbed systems and legacy codes,” *SIAM Journal on Applied Dynamical Systems*, vol. 4, pp. 711–732, 2005. [Online]. Available: http://link.aip.org/link/?SJA/4/711/1

[37] A. Zagaris, H. Kaper, and T. J. Kaper, “Two perspectives on reduction of ordinary differential equations,” *Math. Nach.*, vol. 278, no. 12-13, pp. 1629–1642, 2005. [Online]. Available: http://dx.doi.org/10.1002/mana.200410328

[38] Y. Pesin, “Families of invariant manifolds corresponding to nonzero characteristic exponents,” *Math. USSR Izvestia*, vol. 10, pp. 1261–1305, 1976.

[39] D. Barkley and I. Kevrekidis, “A dynamical systems approach to spiral-wave dynamics,” *Chaos*, vol. 4, pp. 453–460, 1994.

[40] D. Barkley, “Spiral meandering,” in *In: Chemical Waves and Patterns, Eds. Kapral, R. and Showalter, K.*, R. Kapral and K. Showalter, Eds. New York, NY: Springer-Verlag, 1995, pp. 163–190.

[41] L. Arnold, *Random Dynamical Systems.* Springer-Verlag New York, Inc., 1998.

[42] J. Stark, "Delay embeddings for forced systems. I. Deterministic forcing," *Journal of Nonlinear Science*, vol. 9, pp. 255–332, 1999. [Online]. Available: http://dx.doi.org/10.1007/s003329900072

[43] J. Stark, D. Broomhead, M. Davies, and J. Huke, "Delay embeddings for forced systems. II. Stochastic forcing," *Journal of Nonlinear Science*, vol. 13, pp. 519–577, 2003. [Online]. Available: http://dx.doi.org/10.1007/s00332-003-0534-4

[44] S. Mallat, "A Wavelet Tour of Signal Processing," *3rd ed., Academic Press*, 2008.

[45] D. Raviv, M. Bronstein, G. Sapiro, A. Bronstein, and R.Kimmel, "Diffusion symmetries of non-rigid shapes," in *In Proc. 3DPVT*, 2010.

[46] A. Bronstein, M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro, "A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching," *Int. J. Comput. Vision*, vol. 89, no. 2-3, pp. 266–286, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1007/s11263-009-0301-6

[47] M. Ovsjanikov, J. Sun, and L. Guibas, "Global intrinsic symmetries of shapes," in *Proceedings of the Symposium on Geometry Processing*, ser. SGP '08. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2008, pp. 1341–1348. [Online]. Available: http://dl.acm.org/citation.cfm?id=1731309.1731314

[48] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[49] R. Mehra, "On the identification of variances and adaptive kalman filtering," *IEEE Trans. Auto. Cont.*, vol. 15, pp. 175 – 184, 1970.

[50] ——, "Approaches to adaptive filtering," *IEEE Trans. Auto. Cont.*, vol. 17, pp. 693 – 698, 1972.

[51] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics," *J. Geophys. Res.*, vol. 99(C5), pp. 10 143–10 162, 1994.

[52] P. Houtekamer and H. L. Mitchell, "Data assimilation using an ensemble kalman filter technique," *Mon. Wea. Rev.*, vol. 126, pp. 796–811, 1998.

[53] G. Evensen, *Data Assimilation: The Ensemble Kalman Filter, 2nd ed.* Springer, Berlin, 2009.

[54] E. Kalnay, *Atmospheric modeling, data assimilation, and predictability.* Cambridge University Press, Cambridge, 2003.

[55] D. Simon, *Optimal State Estimation: Kalman, $H_\infty$, and Nonlinear Approaches.* John Wiley and Sons, Inc., 2006.

[56] D. Dee, "On-line estimation of error covariance parameters for atmospheric data assimilation," *Mon. Wea. Rev.*, vol. 123, pp. 1128–1145, 1995.

[57] R. Daley, “Estimating model-error covariances for application to atmospheric data assimilation,” *Mon. Wea. Rev.*, vol. 120, pp. 1735–174, 1992.

[58] H. Li, E. Kalnay, and T. Miyoshi, “Simultaneous estimation of covariance inflation and observation errors within an ensemble kalman filter,” *Quarterly Journal of the Royal Meteorological Society*, vol. 135, pp. 523–533, 2009.

[59] J. Anderson, “An adaptive covariance inflation error correction algorithm for ensemble filters,” *Tellus*, vol. 59A, pp. 210–224, 2007.

[60] J. L. Anderson and S. Anderson, “A monte-carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts,” *Mon. Wea. Rev.*, vol. 127, pp. 2741–2758, 1999.

[61] G. Desroziers, L. Berre, B. Chapnik, and P. Poli, “Diagnosis of observation, background and analysis-error statistics in observation space,” *Quarterly Journal of the Royal Meteorological Society*, vol. 131, p. 33853396, 2005.

[62] J. Anderson, “Spatially and temporally varying adaptive covariance inflation for ensemble filters,” *Tellus*, vol. 61A, pp. 72 – 83, 2008.

[63] E. Constatinescu, A. Sandu, T. Chai, and G. Carmichael, “Ensemble-based chemical data assimilation. i: General approach,” *Quar. J. Roy. Met. Soc.*, vol. 133, pp. 1229–1243, 2007.

[64] ——, “Ensemble-based chemical data assimilation. ii: Covariance localization,” *Quar. J. Roy. Met. Soc.*, vol. 133, pp. 1245–1256, 2007.

[65] X. Luo and I. Hoteit, “Robust ensemble filtering and its relation to covariance inflation in the ensemble kalman filter,” *Mon. Wea. Rev.*, vol. 139, p. 39383953, 2011.

[66] ——, “Ensemble kalman filtering with residual nudging,” *Preprint*, 2012.

[67] X. Wang and C. H. Bishop, “A comparison of breeding and ensemble transform kalman filter ensemble forecast schemes,” *J. Atmos. Sci.*, vol. 60, p. 11401158, 2003.

[68] T. Hamill, J. Whitaker, and C. Snyder, “Distance-dependent filtering of background error covariance estimates in an ensemble kalman filter,” *Mon. Wea. Rev.*, vol. 129, p. 27762790, 2001.

[69] H. Mitchell and P. Houtekamer, “An adaptive ensemble kalman filter,” *Mon. Wea. Rev.*, vol. 128, p. 416433, 2000.

[70] J. Anderson, “An ensemble adjustment kalman filter for data assimilation,” *Mon. Wea. Rev.*, vol. 129, p. 28842903, 2001.

[71] G. Evensen, “The ensemble kalman filter: Theoretical formulation and practical implementation,” *Ocean Dynamics*, vol. 53, pp. 343–367, 2003.

[72] J. Z. Jiang, “A novel adaptive unscented kalman filter for nonlinear estimation,” *46th IEEE Conference on Decision and Control*, pp. 4293–4298, 2007.

[73] D. Gonzalez-Tokman and B. Hunt, "Ensemble data assimilation for hyperbolic systems," *To appear, Physica D.*, 2013.

[74] E. Lorenz, "Predictability: A problem partly solved," *Seminar on Predictability*, vol. 1, ECMWF, p. 118, 1996.

[75] E. Lorenz and K. A. Emanuel, "Optimal sites for supplementary weather observations: Simulation with a small model," *J. Atmos. Sci.*, vol. 55, p. 399414, 1998.

[76] P. R. Bélanger, "Estimation of noise covariance matrices for a linear time-varying stochastic process," *Automatica*, vol. 10, pp. 267–275, 1974.

[77] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in lters and estimators," *IEEE Trans. Automat. Control.*, vol. 45, pp. 477–482, 2000.

[78] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, pp. 401–422, 2004.

[79] E. Ott and et al., "A local ensemble kalman filter for atmospheric data assimilation," *Tellus*, vol. 56A, pp. 415–428, 2004.

[80] B. Hunt, E. Kalnay, E. Kostelich, E. Ott, D. Patil, T. Sauer, I. Szunyogh, J. Yorke, and A. Zimin, "Four-dimensional ensemble kalman filtering," *Tellus*, vol. A56, pp. 273–277, 2004.

[81] B. Hunt, E. Kostelich, and I. Szunyogh, "Efficient data assimilation for spatiotemporal chaos: A local ensemble transform kalman filter," *Phys. D: Nonlinear Phenomena*, vol. 230, pp. 112–1263, 2007.

[82] V. Garcia, E. Debreuve, and M. Barlaud, "Fast k-nearest neighbor search using GPU," in *CVPR Workshop on Computer Vision on GPU*, Anchorage, Alaska, USA, 2008.