Nonlinear Model Reduction Based on Manifold Learning with Application to the Burgers' Equation

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Christopher Joel Winstead

May 2017

Abstract

Real-time applications of control require the ability to accurately and efficiently model the observed physical phenomenon in order to formulate control decisions. Complex flow interactions may require the modelling of millions of states making the problem computationally intractable. Model order reduction aims to reduce this computational burden while still retaining accuracy as compared to the full order model. Nonlinear dimension reduction methods such as Local Linear Embedding, Diffusion Maps, and Laplacian Eigenmaps are implemented on a series of solution snapshots of the one dimensional Burgers' equation to generate a set of basis functions to be used in Galerkin projections.

The new basis functions are shown to compare favorably to their proper orthogonal decomposition counterparts across different time domains and with different levels of nonlinearity in the system.

Table of Contents

Chapter 1 Introduction 1
1.1 Motivation and Literature Review 1
1.2 Reduced Order Modelling and Turbulence 3
1.3 Organization of Thesis 3
Chapter 2 Problem Setup 5
2.1 Burgers' Equation 5
2.2 The Galerkin Projection 6
2.3 Evaluating the Model Reduction Methods 10
Chapter 3 Proper Orthogonal Decomposition 11
3.1 Background 11
3.2 The POD Method 11
3.3 Reduced Order Dynamics 16
3.4 POD and the Burgers' Equation 17
3.5 POD Limitations 21
Chapter 4 Locally Linear Embedding 24
4.1 Introduction 24
4.2 The LLE Method 25
4.3 Reduced Order Modelling 29
Chapter 5 Diffusion Maps 33
5.1 Introduction 33
5.2 The Diffusion Maps Method 34
5.3 Reduced Order Modelling 38
Chapter 6 Laplacian Eigenmaps 46
6.1 Introduction 46
6.2 The Laplacian Eigenmaps Method 47
6.3 Reduced Order Modelling 49
Chapter 7 Further Testing 54
7.1 Time Domain and Reynolds Number 54
7.2 Predictive Models 56
Chapter 8 Conclusions 59
References 61
Vita 67

List of Tables

Table 1: Information Captured by Successive Modes 18
Table 2: Relative Energy Captured and Compution Time for POD Reduced Dynamics 23
Table 3: Relative Captured Energy and Computation Time for LLE Modes 31
Table 4: Relative Captured Energy and Computation Time with Diffusion Map Reduced Dynamics 42
Table 5: Relative Captured Energy and Computation Time of Laplacian Eigenmap Reduced Dynamics 53
Table 6: Reduced Dynamics From Nonlinear Modes on [0,10] (s) (3 modes, Re = 300) 55
Table 7: Reduced Dynamics from Nonlinear Modes on [0,10] (s) (5 modes, Re = 300) 55
Table 8: Reduced Dynamics From Nonlinear Modes on [0,10] (s) (3 modes, Re = 700) 55
Table 9: Reduced Dynamics From Nonlinear Modes on [0,10] (s) (5 modes, Re = 700) 55
Table 10: Reduced Dynamics From Nonlinear Modes on [0,10] (s) 57
Table 11: Reduced Dynamics From Nonlinear Modes on [0,10] (s) 57
Table 12: Reduced Dynamics From Nonlinear Modes on [0,10] (s) 57
Table 13: Reduced Dynamics From Nonlinear Modes on [0,10] (s) 57

List of Figures

Figure 1: Point Cloud in 3D 12
Figure 2: Point Cloud Projected into 2D 12
Figure 3: Full Solution Manifold 18
Figure 4: Information Captured by Additional POD Modes 18
Figure 5: Error Between POD Reduced Dynamics and Original Snapshots 19
Figure 6: Reconstructed Snapshots at Various Times 20
Figure 7: Various POD Modes 20
Figure 8: 3D Swiss Roll Dataset 22
Figure 9: Swiss Roll Projected Into 2D via POD 22
Figure 10: Locally Linear Embedding Algorithm [32] 25
Figure 11: Error Between Reduced Model and Full with Different Numbers of Nearest Neighbors (knn = 2, 3, 5) 30
Figure 12: Reconstruction of LLE with 3 Modes and Two Nearest Neighbors (knn = 2) 30
Figure 13: Reconstructed LLE Models Utilizing More Modes 32
Figure 14: Modes Generated by LLE 32
Figure 15: Swiss Roll Projected into 2D via Diffusion Maps 33
Figure 16: Diffusion Map Error vs POD Error 39
Figure 17: Snapshots of the Reconstructed DMAP System 39
Figure 18: Reduced Dynamics Using Diffusion Maps (ϵ=15C) and Different Numbers of Modes 41
Figure 19: Diffusion Map Dynamics with Eliminated Redundant Modes 41
Figure 20: Reconstructed Diffusion Map Snapshots 43
Figure 21: Sampling of Diffusion Map Generated Modes 43
Figure 22: Error Between Diffusion Map Reconstructed Model and POD Reconstructed Mode 45
Figure 23: 2D Laplacian Eigenmap Projection of Swiss Roll Using 10 Nearest Neighbors 46
Figure 24: Error from Reduced Dynamics Using Laplacian Eigenmap Modes with Different Weights 50
Figure 25: Laplacian Eigenmaps Generated Dynamics with Different Numbers of Nearest Neighbors (knn) 50
Figure 26: Dynamics Generated with Different Numbers of Laplacian Eigenmap Generated Modes (knn defines number of nearest neighbors considered) 51
Figure 27: Laplacian Eigenmap Modes (knn = 2, ε = 10) 52
Figure 28: Laplacian Eigenmap Modes (knn = 15, ε = 10) 52

Chapter 1 Introduction

1.1 Motivation and Literature Review

In applications such as highly resolved fluid flow solutions, systems must describe thousands of states, millions in the case of modelling turbulent flow. This makes the computation of feedback control a computationally intractable problem, and consequently, the order of the system must be reduced [1] [2]. This strategy is particularly useful when the internal dynamics of the system can be captured by a small number of modes relative to the size of the original system [3] [4] [5]. Much of the previous efforts in this area have centered around Proper Orthogonal Decomposition (POD) [6] [7] [8].

In a "design-then-reduce" approach, full order control design follow full order model design, and then proceeds to control order reduction [6] [9] [10]. This method works with a more complete set of physical phenomenon and results in robust low order models and can provide insight into sensor design [11] [12] [13]. However, for applications such as fluid flow, the discretized system can describe millions of state variable. For example, if one were to use a linear quadratic regulator (LQR) control formulation [14], the computation for a system describing 10^6 states would require roughly 10^{12} Ricatti unknowns. Conventional computing power and computational algorithms are inadequate for the task. In these types of problems, POD is usually the means to perform model reduction of either the plant or the controller [1] [2] [15]. In [16], POD was applied to the functional gains of an infinite dimensional control law to obtain a reduced-order controller. In [17] [18] [19] reduced order models based on POD have been obtained for linearized compressible flow equations. Extending these methods to high Reynolds number nonlinear flow, where the nonlinear terms dominate, remains an open question

however. In [20], POD was used to obtain a reduced order model in the control of room temperature in buildings. The model was based on a 3D heat equation. POD was able to find a low dimensional model that preserved most of the energy [21] [22] [23].

The method of POD however, poses challenges for general domains. While optimal for a given data set in terms of energy, POD is not complete and thus may result in a loss of stability properties. Dynamic Mode Decomposition (DMD) is an alternative data-driven designed to match stability eigenmodes under suitable conditions. DMD trades the optimal resolution efficiency of POD against distillation of pure eigenfrequencies in short-time sampled data [24]. Further, for nonlinear PDE's, the snapshot sets may contain nonlinear structures not captured by POD, as POD uses (numerically) the Euclidean distance [21] [23] [25]. Snapshots far apart in terms of their geodesic may be considered close by POD due to their Euclidean distance [26].

Trajectory piecewise linear techniques (TPWL) linearize the nonlinear system around suitably selected states and approximate the system by a piecewise linearization obtained from combing the reduced linearized system via a weighing procedure [27]. However, this method is not robust due to the procedure to determine the appropriate linearization points and the training input [27]. Nonlinear model order reduction methods based on the concept of local reduced-order bases were proposed in [25] [28] [29] [30]. These methods lack robustness with respect to parameter variation however [28].

Nonlinear embedding techniques such as Locally Linear Embedding (LLE) [31] [32], Laplacian Eigenmaps [33] [34], and Diffusion Maps [35] [36] offer the ability to embed nonlinear solution manifolds into a significantly lower dimensional space than linear techniques. Harnessing this ability for simulation could potentially allow the modelling of nonlinear systems more efficiently and with less lost information than traditionally linear techniques.

1.2 Reduced Order Modelling and Turbulence

The history of reproducing vorticity and turbulence in dynamical models lies in the works of Hopf [37] and Landau [38]. Early work concerned the identification of attractors in complex flow through the use of Fourier spectrum analysis. They ascertained that a manifold exists to which the solution of the Navier-Stokes equation decays under particular initial conditions. This work progressed into the truncation of the Fourier modes in the attempt to find a lower dimensional "strange attractor" of the manifold [39]. Ultimately, methods such as invariant manifold techniques, bifurcation theory, and the unfolding of degenerate singularities have become meaningful in the analysis of "pre-turbulent" interactions.

In [40], Holmes extends these low dimensional systems to the modelling of open, fully developed turbulent flow. He does this by using Proper Orthogonal Decomposition (POD) in the identification of "modes" that carry the most significant amounts of information (in terms of kinetic energy) about the system. These modes are then used as the global basis functions of a Galerkin projection of the system. These leads to a reduced set of ODE's representing the original PDE, which are helpful in the analysis of the flow structure and identification of "coherent structures" in a complex system. In the effort of developing nonlinear methods for the analysis of turbulent structures, this thesis will follow the advice of Holmes, in restricting focus to functions of a single variable in time. The function in question for the purposes of this thesis is the one dimensional Burgers' Equation.

1.3 Organization of Thesis

The focus of this thesis is an investigation of the applicability of nonlinear embedding techniques as applied to the POD-Galerkin model reduction methods laid out in [40] on the 1D

Burgers' equation. Specifically, Locally Linear Embedding (LLE), Diffusion Maps (Dmaps), and Laplacian Eigenmaps (Emaps) will be used to generate the modes used as global basis functions in the Galerkin projection. The ability to recreate the dynamics of the Burgers' Equation using these nonlinear modes will be compared to the traditional POD-Galerkin linear model reduction technique.

Chapter 2 will review the problem setup of the 1D Burgers' Equation and the Galerkin projection as a means to discretize the PDE in space. Chapter 3 will focus on the optimality and justification of POD and its application to reduced order modelling. Chapter 4 will explore the method of LLE, and explore the utility of the basis of the LLE embedding with respect to the Galerkin projection. Similarly, Chapter 5 will explore the derivation of Diffusion Maps and the efficacy of using a basis based on the embedding vectors generated by the Diffusion Maps process in the Galerkin method. Chapter 6 will review the optimality and derivation of Laplacian Eigenmaps, and investigate their application towards generating basis functions to be used in the Galerkin method. Chapter 7 will further compare the different dimensionality reduction methods by extending the time domain of the simulation and increasing the nonlinearity of the system. Chapter 8 will conclude the thesis with final conclusions and potential avenues for further improvement.

Chapter 2 Problem Setup

2.1 Burgers' Equation

The Burgers' Equation in its 1D form is as follows [41],

$$\frac{\partial w(x,t)}{\partial t} = \frac{1}{Re}\frac{\partial^2 w(x,t)}{\partial x^2} + w(x,t)\frac{\partial w(x,t)}{\partial x} \tag{1}$$

In the above, Re is the analogue of the Reynolds number, a dimensionless scalar value denoting the influence of the nonlinearity. The Burgers' Equation is a popular testing equation for model reduction as the nonlinearity in the dynamics is similar to that exhibited by the more complicated Navier-Stokes equation. The conservative form of the equation is achieved by transforming the term $w\frac{dw}{dx}$ to $\frac{d(w^2)}{dx}$ so that equation (1) becomes,

$$\frac{\partial w}{\partial t} = \frac{1}{Re}\frac{\partial^2 w}{\partial x^2} + \frac{\partial\left(w^2\right)}{\partial x} \tag{2}$$

This equation is defined on the one dimensional domain $[a, b]$ with initial condition,

$$w(0,x) = w_0(x) \tag{3}$$

And boundary conditions,

$$w(a,0) = f(t) = u_a(t) \tag{4}$$

$$w(b,0) = g(t) = u_b(t) \tag{5}$$

In setting up the reduced order machinery, first an ensemble of solutions at different times, or "solution snapshots,"

$$\left\{\vec{w}_i\right\}_{i=1}^{m} \in R^n \tag{6}$$

must be gathered. That is, there exists m snapshots, each of dimension n. For the purposes of this thesis, these snapshots are generated by computing a solution manifold using the Galerkin method in MATLAB with 500 sinusoidal global basis functions. The solution manifold is then sampled to generate the ensemble.

While the Burger's equation can be solved explicitly, more complex systems are often modelled by equations that rely exclusively on numerical techniques for their solutions. As the model reduction methods in question are ultimately intended for these more complex systems, this thesis will rely on numerical methods for solving the Burgers' equation as well.

2.2 The Galerkin Projection

The aforementioned snapshots and the future model reduction machinery rely on using the Galerkin Method to discretize the original PDE into a system of ODE's. The procedure is as detailed in [2], beginning with the weak formulation

$$\frac{\partial w(x,t)}{\partial t}\phi_i(x) = \frac{1}{Re}\frac{\partial^2 w(x,t)}{\partial x^2}\phi_i(x) + \frac{1}{2}\frac{\partial\left(w(x,t)^2\right)}{\partial x}\phi_i(x) \tag{7}$$

In Equation (7), $\phi_i(x)$ are the global basis functions upon which the solution is solved

Further, the solution *w* is a linear combination of the basis functions such that,

$$w(x,t)=\sum_{i=1}^{N}\alpha_i(t)\phi_i(x) \tag{8}$$

The temporal coefficients of the solution are represented by $\alpha_i(t)$. Integrating Equation (7) on the domain Ω and using Equation (8) and the orthogonality of the basis functions $\phi_i(x)$, the left hand side becomes,

$$\int_{\Omega}\frac{\partial w(x,t)}{\partial t}\phi_j(x)\partial x=\int_{\Omega}\frac{\partial(\sum_{i=1}^{n}\alpha_i(t)\phi_i(x))}{\partial t}\phi_j(x)\partial x=\frac{\partial\alpha_i(t)}{\partial t} \tag{9}$$

Continuing the integration on the right hand side of Equation (7) term by term via integration by parts, the linear diffusion term becomes,

$$\int_{\Omega}\frac{1}{Re}\frac{\partial^2 w(x,t)}{\partial x^2}\phi_j(x)\partial x=\frac{1}{Re}\left[\frac{\partial w(x,t)}{\partial x}\phi_j(x)\Big|_{\Omega}-\int_{\Omega}\frac{\partial w(x,t)}{\partial x}\frac{\partial\phi_j(x)}{\partial x}\partial x\right]$$

$$=\frac{1}{Re}\left[\frac{\partial w(b,t)}{\partial x}\phi_j(b)-\frac{\partial w(a,t)}{\partial x}\phi_j(a)-\int_{\Omega}\frac{\partial w(x,t)}{\partial x}\frac{\partial\phi_j(x)}{\partial x}dx\right] \tag{10}$$

The derivatives on the boundary are computed manually as,

$$\frac{\partial w(a,t)}{\partial x}=\frac{w(a+h,t)-w(a,t)}{h} \tag{11}$$

$$\frac{\partial w(b,t)}{\partial x}=\frac{w(b,t)-w(b-h,t)}{h} \tag{12}$$

Where,

$$w(a+h,t)=\sum_{i=1}^{n}\alpha_i(t)\phi_i(a+h) \tag{13}$$

$$w(b-h,t)=\sum_{i=1}^{n}\alpha_i(t)\phi_i(b-h) \tag{14}$$

as is defined in Equation (8). In matrix form, the diffusion term becomes,

$$\int_{\Omega}\frac{\partial^2 w(x,t)}{\partial x^2}\phi_i(x)\partial x = A\alpha + Bu \tag{15}$$

Where u is the set of boundary conditions defined in Equations (4) and (5), and A and B are defined as below.

$$A_{j,i}=\frac{1}{Re}\left[\frac{\phi_i(b-h)\phi_j(b)}{h}-\frac{\phi_i(a+h)\phi_j(a)}{h}+\int_{\Omega}\frac{d\phi_i(x)}{dx}\frac{d\phi_j(x)}{dx}dx\right] \tag{16}$$

B is an N x 2 matrix such that,

$$B_{j,1}=\frac{\phi_j(a)}{hRe},\quad B_{j,2}=\frac{\phi_j(b)}{hRe} \tag{17}$$

The nonlinear term in Equation (7) becomes,

$$\frac{1}{2}\int_{\Omega}\frac{dw(x,t)^2}{dx}\phi_j(x)dx=\frac{1}{2}\left[w(x,t)^2\phi_j(x)\Big|_{\Omega}-\int_{\Omega}w(x,t)^2\frac{d\phi_j(x)}{dx}dx\right] \tag{18}$$

$$=\frac{1}{2}\left[w(b,t)^2\phi_j(b)-w(a,t)^2\phi_j(a)-\int_{\Omega}w(x,t)^2\frac{d\phi_j(x)}{dx}dx\right] \tag{19}$$

This term, in matrix form, becomes,

$$\frac{1}{2}\int_{\Omega}\frac{\partial w(x,t)^2}{\partial x}\phi_j(x)\partial x = N(a) + B'u^2 \tag{20}$$

Where,

$$N(a) = \frac{1}{2}\int_{\Omega}\sum_{i=1}^{N}\left(\alpha_i(t)\phi_i(x)\right)^2\frac{d\phi_j(x)}{dx}dx \tag{21}$$

B' is an N x 2 matrix where,

$$B'_{j,1} = \frac{\phi_j(a)}{2}, \quad B_{j,2} = -\frac{\phi_j(b)}{2} \tag{22}$$

This discretization ultimately results in the equation below, which, in vector form, represents a system of ODE's discretized from the original PDE. This can then be solved by a numerical solver such as ode45 in Matlab to find the temporal coefficients α.

$$\frac{d\alpha}{dt} = A\alpha + N(\alpha) + Bu + B'u^2 = f(\alpha) \tag{23}$$

In equation (23), u is the 2×1 matrix containing the boundary conditions, and u^2 is an element-wise squaring of u.

The ϕ_j are instrumental in the model reduction processes. In this thesis, the various dimensionality reduction algorithms will be used to generate the global basis functions ("modes") that will be used to solve the system numerically. In the model reduction process, unless otherwise stated, the snapshots will be fed into the reduction algorithms as *m* spatial vectors of dimension *n*. Thus, the global basis functions will consist of vector valued functions evaluated at each node in the spatial discretization.

2.3 Evaluating the Model Reduction Methods

An important aspect of validating the advantages of the different model reduction methods is how to compare their advantages against each other. This thesis will take two approaches. The first is a direct comparison between the snapshot set and the reconstructed reduced dynamics. This will be a simple calculation of the Euclidean norm of the error between the original dynamics and the reconstructed dynamics at each node in the snapshots. The second metric of comparison will be the "relative captured energy" taken from [34], defined as follows,

$$e = 1 - \frac{\sum_{i=1}^{m} \left\langle \delta w(t_i, \vec{x}), \delta w(t_i, \vec{x}) \right\rangle}{\sum_{i=1}^{m} \left\langle w(t_i, \vec{x}), w(t_i, \vec{x}) \right\rangle} \tag{24}$$

In Equation (24), $\delta w(t, \vec{x})$ represents the error between the original snapshots and the reconstructed snapshots, $\langle \cdot \rangle$ is the L_2 inner product, and m is the number of snapshots sampled from the manifold.

Chapter 3 Proper Orthogonal Decomposition

3.1 Background

Proper Orthogonal Decomposition (POD), also known as Principal Component Analysis, is a statistical technique in which a data set is projected along its principal components so as to maximize the variance of the data in the lower dimension [42]. Equivalently, POD minimizes the mean squared error between the original data and the projection [43]. A demonstration of this can be seen in Figure 1 and Figure 2, in which a three dimensional point cloud is projected into two dimensions corresponding to the two principal components preserving the most variance of the data. In Figure 2, the two principal components are the x and y axis.

3.2 The POD Method

The ultimate goal POD can be defined as finding a subspace approximating a set of data in an optimal least-squares sense [44]**.** This "data" consists of the solution snapshots described in Equation (6) where, $w = \{\vec{w}_i\}_{i=1}^{m}$ with $w_i \in R^n$. The derivation below is as described in [45].

Minimizing the Residual Error

As previously stated, POD can be looked at as minimizing the mean squared error of the projection. The mean squared error is the sum of the residual error between the original snapshot ensemble (under the assumption of centered data, i.e. zero mean) and their projection onto their nearest principal components. Let $\xi \in R^{n\times r}$ be the orthonormal projection matrix of principal components, with a projection of $\vec{w}_i$ defined by, $\vec{u}_i = \xi^T \vec{w}_i : \ \vec{u}_i \in R^{r\times 1}$, with $r < n$. The

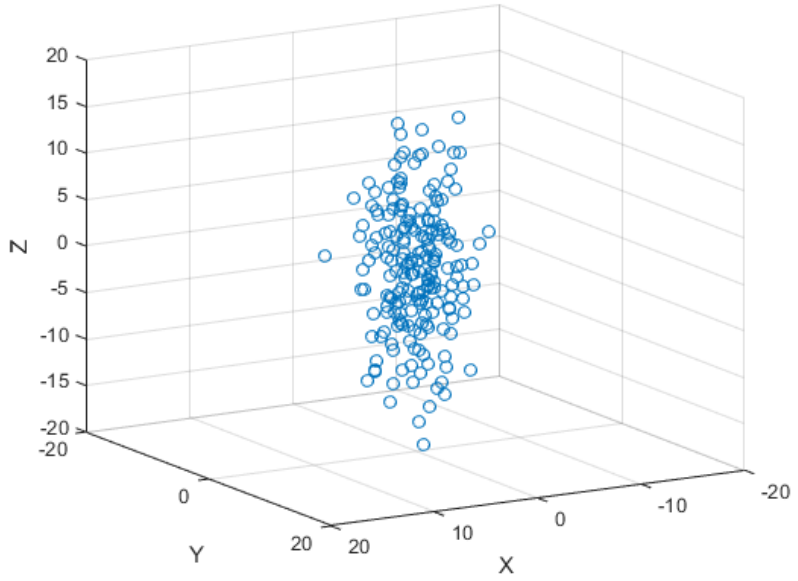


Figure 1: Point Cloud in 3D

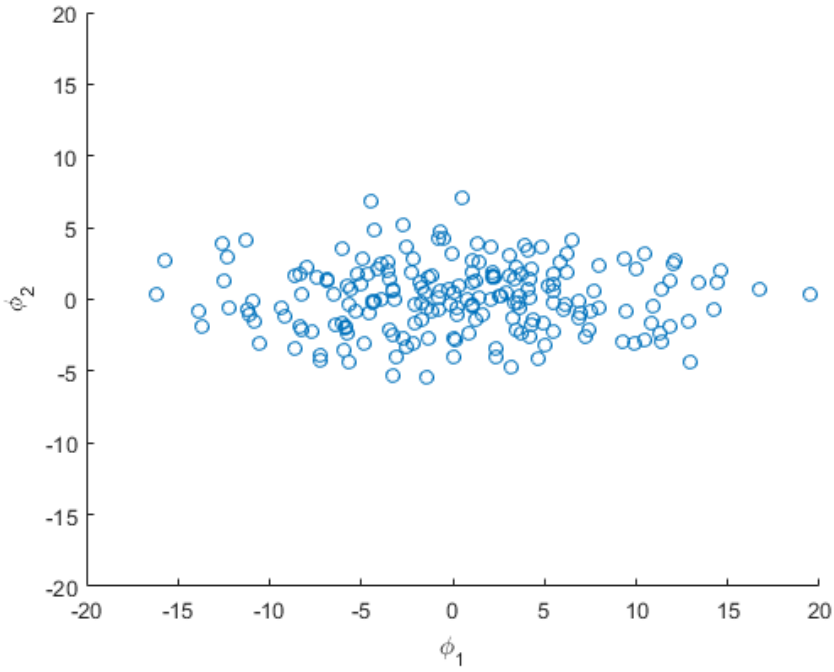


Figure 2: Point Cloud Projected into 2D

definition of residual error for any one vector of data (referred to as a solution snapshot in this thesis), is

$$\begin{aligned}\left\|\vec{w}_i-\xi\left(\xi^T\vec{w}_i\right)\right\|^2 &=\left(\vec{w}_i-\xi\left(\xi^T\vec{w}_i\right)\right)^T\left(\vec{w}_i-\xi\left(\xi^T\vec{w}_i\right)\right)\\ &=\vec{w}_i{}^T\vec{w}_i-2\vec{w}_i{}^T\xi\xi^T\vec{w}_i+\vec{w}_i{}^T\xi\xi^T\xi\xi^T\vec{w}_i\\ &=\vec{w}_i{}^T\vec{w}_i-\vec{w}_i{}^T\xi\xi^T\xi\xi^T\vec{w}_i\\ &=\left\|\vec{w}_i\right\|^2-\left\|\xi^T\vec{w}_i\right\|^2\end{aligned} \tag{25}$$

This follows from the orthonormality of ξ as $\xi^T\xi = I$. Summing the residuals leads to the mean square error,

$$MSE(\vec{\xi})=\frac{1}{m}\left(\sum_{i=1}^{m}\left\|\vec{w}_i\right\|^2-\sum_{i=1}^{m}\left\|\xi^T\vec{w}_i\right\|^2\right) \tag{26}$$

In terms of the minimization problem, minimizing the residual error requires maximizing the second term in Equation (26), $\frac{1}{m}\left(\sum_{i=1}^{m}\left\|\xi^T\vec{w}_i\right\|^2\right)$ subject to ξ. To this end, note that the mean of a square is equivalent to the square of the mean plus the variance, such that,

$$\frac{1}{m}\sum_{i=1}^{m}\left\|\xi^T\vec{w}_i\right\|^2=\left\|\frac{1}{m}\sum_{i=1}^{m}\xi^T\vec{w}_i\right\|^2+Var\left[\xi^T\vec{w}_i\right] \tag{27}$$

Further, note that the mean of the data vectors (assumed to be zero) are equal to the mean of the projections of those vectors [45]. That is,

$$\frac{1}{m}\sum_{i=1}^{m}\xi^T\vec{w}_i=\xi^T\left(\frac{1}{m}\sum_{i=1}^{m}\vec{w}_i\right) \tag{28}$$

Therefore, the minimization problem of POD is an equivalent problem to a maximization of the variance of the projection

Maximizing the Variance of the Projection

The variance of the data, or the average distance to the mean [45] (taken to be zero in this case), is defined as

$$\begin{aligned} Var\left[\xi^T \vec{w}_i\right] &= \frac{1}{m}\left(\sum_{i=1}^{m}\left\|\xi^T \vec{w}_i\right\|^2\right) \\ &= \frac{1}{m}\left(\sum_{i=1}^{m}\left\|\xi^T \vec{w}_i \vec{w}_i{}^T \xi\right\|\right) \\ &= \xi^T X \xi \end{aligned} \tag{29}$$

In equation (29), u_i are the columns of the projection matrix ξ, and X is equal to the covariance matrix of $\boldsymbol{w}$. To achieve the desired maximization, the Lagrange multiplier λ, with constraint, ${u_i}^T u_i = 1$, is introduced [46], and maximizing each ${u_i}^T X u_i$ becomes,

$$L(\xi, \lambda) = \xi^T X \xi - \lambda(\xi^T \xi - 1) \tag{30}$$

$$\Rightarrow \frac{dL}{d\lambda} = \xi^T \xi - 1 \tag{31}$$

$$\Rightarrow \frac{dL}{d\xi} = 2X\xi - 2\lambda\xi \tag{32}$$

Setting these derivatives to zero, Equations (31) and (32) become,

$$\xi^T \xi = 1 \tag{33}$$

$$X\xi = \lambda\xi \tag{34}$$

From this, it can be seen that the desired matrix ξ is the matrix of eigenvectors of the covariance matrix *X*. The first eigenvector of *X* represents the optimal direction to project the data (in the "energy" sense). The second eigenvector of *X* represents the optimal direction to project the snapshots, but constrained to a space orthogonal to the space represented by the first eigenvector and so on. The eigenvectors will hereafter be referred to as the POD modes. This method of choosing the correct optimization problem, and solving the eigenvalue problem, is fundamental to many linear and nonlinear model reduction techniques.

Approximation Accuracy

The accuracy of the low dimensional POD projection depends on the number of eigenvectors used to create the projection matrix ξ. This quantity is connected to the "relative information content" (RIC) [47]. RIC is a means of evaluating the contribution of POD modes towards the system dynamics. That is, larger valued eigenvalues of the covariance matrix and their corresponding eigenvectors correspond to modes containing more significant information about the system dynamics. Smaller valued eigenvalues and their corresponding modes are associated with smaller perturbations of the system. The RIC can be computed from,

$$RIC(r) = \frac{\sum_{j=1}^{r} \lambda_j}{\sum_{j=1}^{n} \lambda_j} \tag{35}$$

Where $RIC(r) = 1$ when $r = n$. Of note is that often there exists a large "break" in the value of eigenvalues for large scale systems. That is, $\lambda_1 > \lambda_2 > \cdots > \lambda_r \gg \lambda_{r+1} > \cdots > \lambda_n > 0$. This break often represents a viable candidate for the size of the reduced order dimension.

3.3 Reduced Order Dynamics

Returning to the Burgers' equation described in Section 2.1, the general dynamical system can be denoted,

$$\frac{\partial w}{\partial t} = F\left(w\right) \tag{36}$$

Desiring to find an approximate dynamical system that evolves on a reduced subspace ϕ (represented by the collection of r POD modes), the new system can be defined as,

$$\frac{\partial w}{\partial t} = F_{\phi}\left(w\right) \tag{37}$$

A direct projection to achieve this subspace can be achieved from,

$$F_{\phi}\left(w\right) = \phi F(w) \tag{38}$$

Applying this explicitly in the Galerkin method begins with generating the approximation of *w*,

$$w\left(x,t\right) = \sum_{i=1}^{k} \alpha_i\left(t\right)\phi_i\left(x\right) \tag{39}$$

The discretization and solution of the dynamics then follows as laid out in Section 2.2. The resulting system can be solved using a numerical solver such as ODE45 in MATLAB, starting from an initial point projected into the lower dimension, i.e. $u_0(x) = \xi w_0(x) \in R^r$.

In real systems, the number of degrees of freedom (dimension n), can grow to be much larger than the number of snapshots (dimension m). This can create undue expense in the solution of the eigenvalue problem described in equation (34). An alternative method to compute the POD modes is the "method of snapshots" [48], which reduces the size of the correlation

matrix, and thus the eigenvalue problem, to $m \times m$, rather than $n \times n$.

3.4 POD and the Burgers' Equation

Now that the method has been established, POD can be utilized to compute the dynamics of the Burgers' Equation in 1D. The initial solution snapshot ensemble was generated on the Fourier modes as described in Section 2.2 over the time domain $[0, 3](\text{seconds})$ and the spatial domain $[0, 1](\text{meters})$. The initial condition was set to,

$$w_0(x) = \sin(2\pi x) \tag{40}$$

with boundary conditions of,

$$u_a(t) = \sin(\pi x) \tag{41}$$

$$u_b(t) = -\sin(\pi x) \tag{42}$$

The solution was sampled at a rate of 0.6 s, to generate 50 snapshots in time, with a spatial discretization of 300 nodes. The Reynolds number was set to 300. The solution was computed using ODE45 and the Galerkin method in Matlab using 500 sinusoidal modes. The full manifold to be tested against can be seen in Figure 3. POD was then performed on the sampled snapshots to generate the POD modes used in the computation of the reduced dynamics. Table 1**:** Information Captured by Successive Modes and Figure 4 display the amount of information captured by adding modes to the reduced order model. With three modes, the amount of information captured by more modes begins to taper off, with 5 modes capturing ~99% of the information. Figure 4 further reinforces that the information gained by adding successive modes to the reduced model diminishes with additional modes. An asymptote appears with the

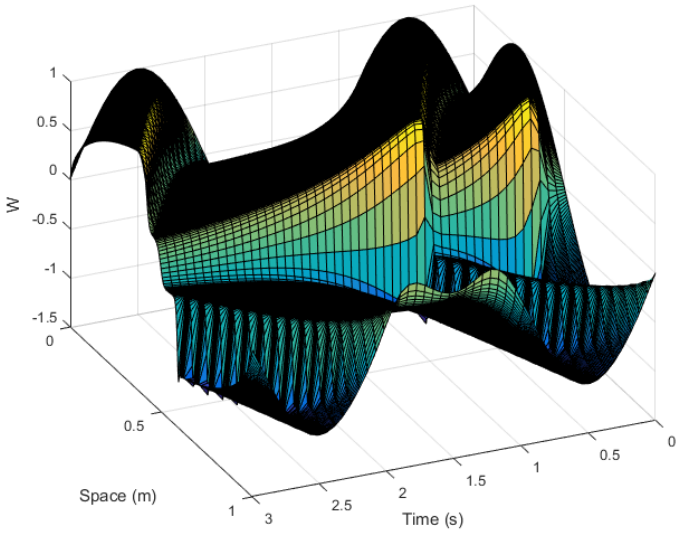


Figure 3: Full Solution Manifold

Table 1: Information Captured by Successive Modes

Number of Modes	2	3	5	10	20
Information Preserved	0.9321	0.9725	0.9848	0.9983	0.99992

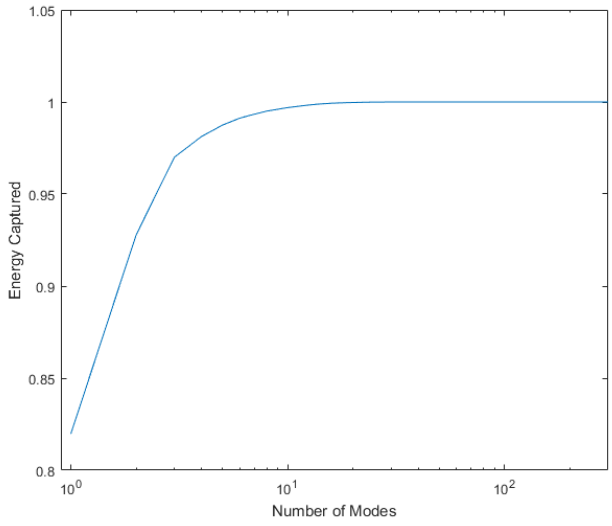


Figure 4: Information Captured by Additional POD Modes

introduction of the third mode, and very little additional information is captured with the introduction of more than five modes.

The error between the reconstructed dynamics and the original dynamics for various numbers of modes can be seen in Figure 5. As can be seen, there is a direct relationship between the number of modes used in the reduced dynamics, and the quality of the reconstruction. It may be beneficial to closer examine the ability of the reduced model to reconstruct the original dynamics at specific points in time. To that end, Figure 6 shows the reconstructed solution snapshots at various times using three modes. Figure 7 shows the various POD modes that are used in the reduced dynamics.

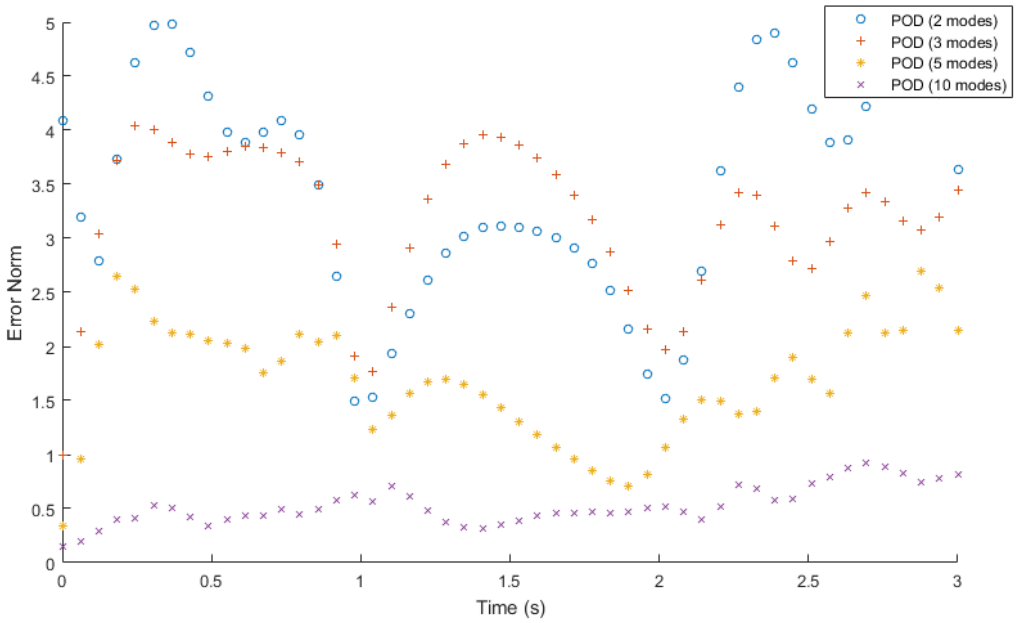


Figure 5: Error Between POD Reduced Dynamics and Original Snapshots

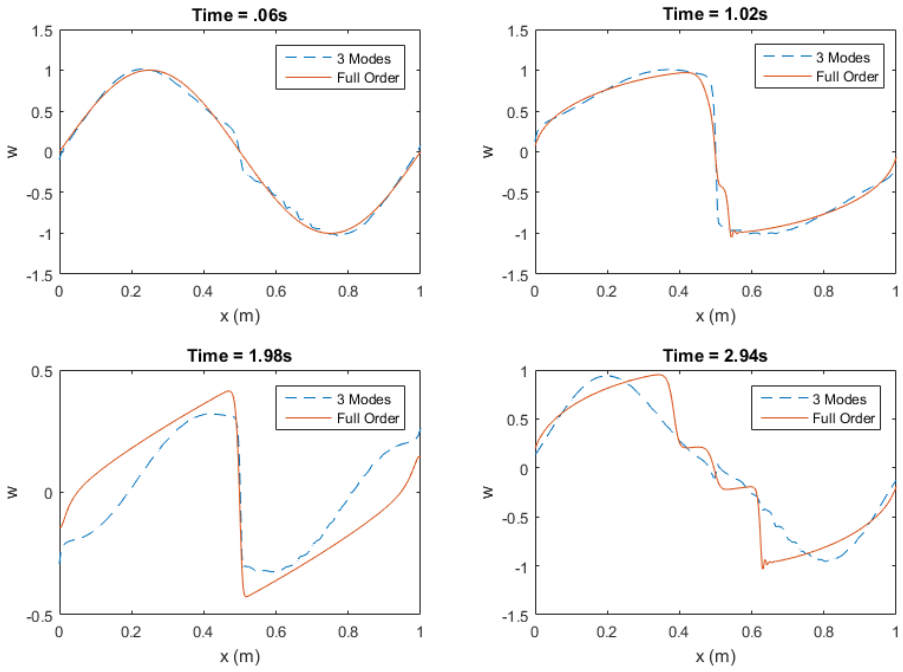


Figure 6: Reconstructed Snapshots at Various Times

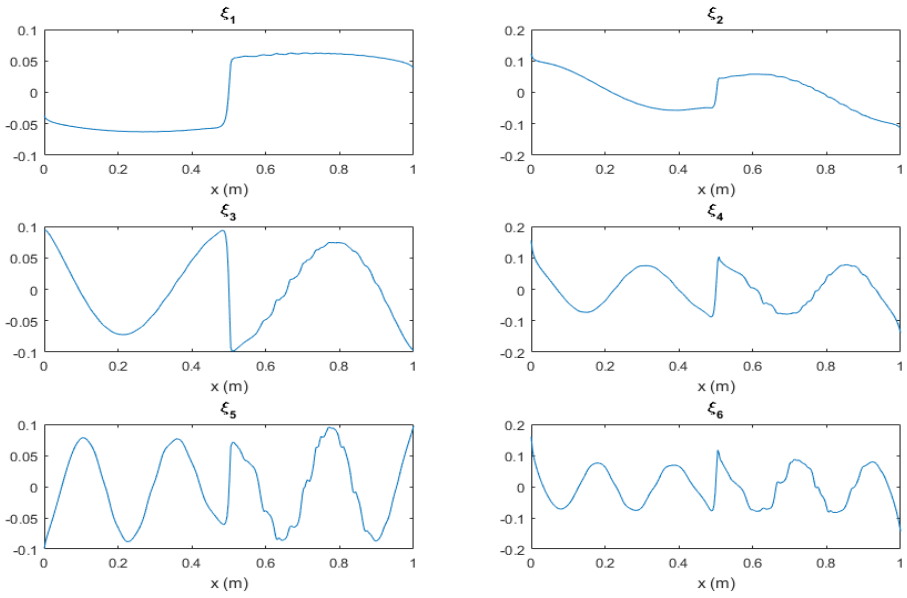


Figure 7: Various POD Modes

Table 2 shows the relative captured energy e and computation time (τ) for different numbers of modes ('N'). As would be expected, the addition of more modes leads to more relative energy captured at the expense of a generally longer computation time. This is line with the information metrics from Table 1 and Figure 4**:** Information Captured by Additional POD Modes. For the remainder of this thesis, the results from utilizing three and five modes will be compared to.

3.5 POD Limitations

While POD reconstructs linear systems very well, it struggles with nonlinear data. An illustrative example is the "Swiss Roll" in 3D data set shown in Figure 8. The Swiss Roll data set is a common training dataset used in the development of nonlinear embedding algorithms [32] [33] [49]. The data is comprised of a two dimensional flat submanifold in R^3. The manifold is generated by computing *n* points $x_i \in R^3$ according to $\boldsymbol{x} = (tcos(t), h, tsin(t))$, where $t = [\frac{3\pi}{2}, \frac{9\pi}{2}]$, and $h = [0, 30]$. Attempting to project the dataset onto the principal components identified by POD does little to capture the structure of the data along the spiral. Thus, the evolution of the colormap is lost in the attempt to preserve the variance of the data. This can be seen in Figure 9.

Further, it should be noted that POD attempts to minimize the error between the original

Table 2: Relative Energy Captured and Compution Time for POD Reduced Dynamics

Method	Full Solution	POD	POD	POD	POD
N	**500**	**2**	**3**	**5**	**10**
e	1	0.8681	0.891	0.9687	0.9969
τ (s)	35.9	0.0324	0.0234	0.0368	0.07

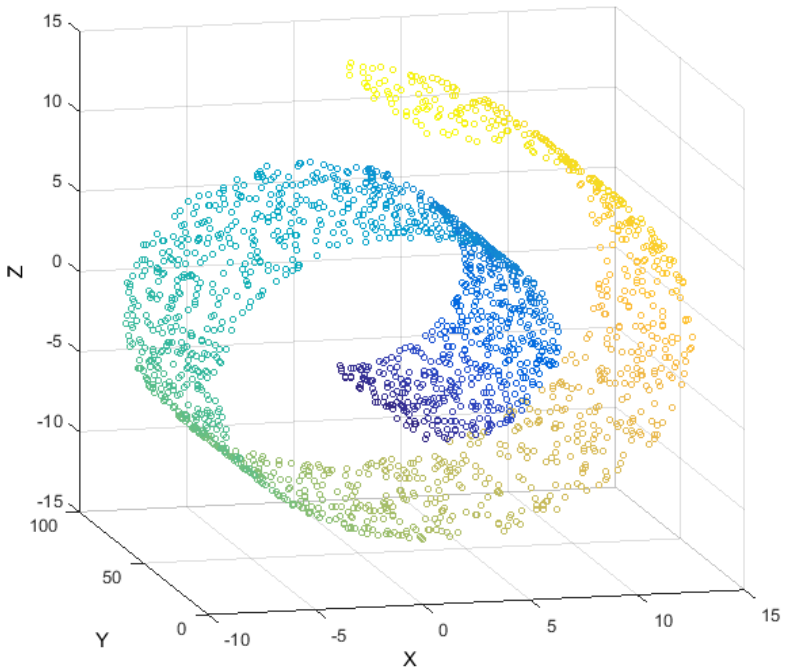


Figure 8: 3D Swiss Roll Dataset

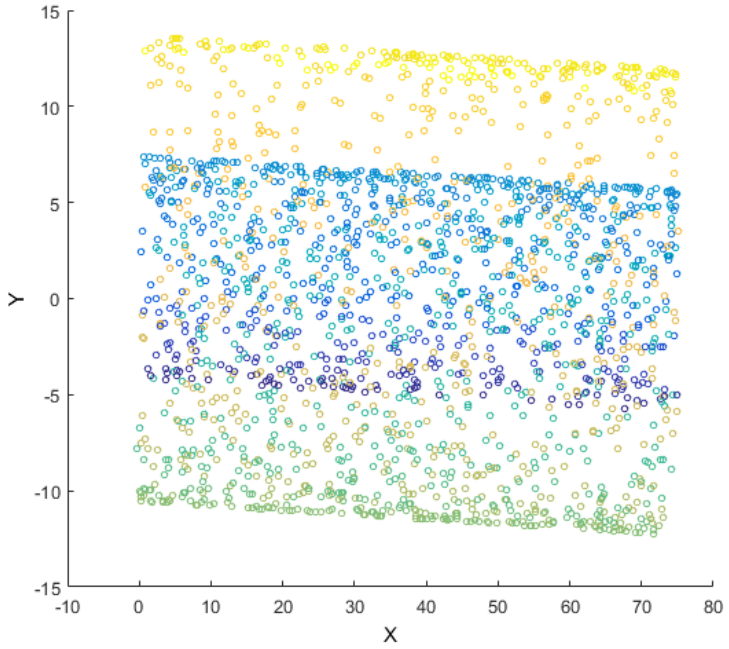


Figure 9: Swiss Roll Projected Into 2D via POD

snapshots and the approximation in a least-squares sense rather than approximating the specific dynamics of a system. Thus, important details of a system's behavior can be lost if the energy represented by changes are small relative to the overall energy of the system. Additionally, a decrease in the least-squares error does not necessarily have a direct relationship to accuracy relative to the system dynamics. In other words, the POD modes do not have a direct physical relevance to the system dynamics [50]. Thus, additional POD modes used in the low order representation do not necessarily lead to a richer representation of the system dynamics [44].

While POD modes are optimal in preserving the energy in a projection, the hope in using modes generated by the nonlinear reduction techniques in the next chapters is that some behavior arising in the simulation due to the nonlinearity is better handled by the nonlinear modes, as opposed to the linear POD modes.

Chapter 4 Locally Linear Embedding

4.1 Introduction

Locally Linear Embedding (LLE) is a nearest neighbors based nonlinear reduction technique that has found use in applications such as clustering and image processing. Similar to POD, LLE is an eigenvector decomposition technique. However, the nonlinear structure of the high-dimensional data is discovered by exploiting the local symmetry of linear reconstructions [31]. Otherwise stated, LLE seeks to map points nearby on-manifold to points nearby in a reduced dimension space. Returning to the Swiss-Roll example, the LLE projection is illustrated by reducing the 3D Swiss Roll Dataset to the 2D space using ten nearest neighbors.

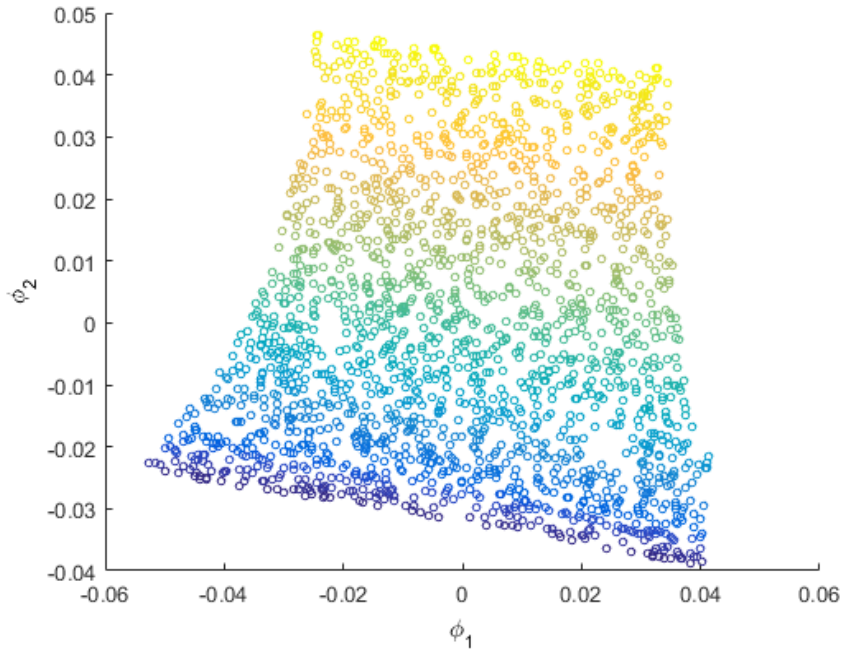


Figure 10: LLE Reduction of Swiss Roll (10 nearest neighbors)

4.2 The LLE Method

The ultimate goal of LLE is the preservation of the geometry of the underlying structure of the data by keeping points in close proximity on the manifold containing the data, in close proximity in a reduced order space. LLE accomplishes this by reconstructing each point from its "K" nearest neighbors using an adjacency weight matrix. The algorithm that accomplishes this is illustrated in Figure 10 [32]. The algorithm begins with identifying the neighbors that will influence the reconstruction of a point. The next stage in the algorithm is the construction of a weight matrix that will reconstruct the point in question. Finally, the low dimensional embedding is created using the weight matrix.

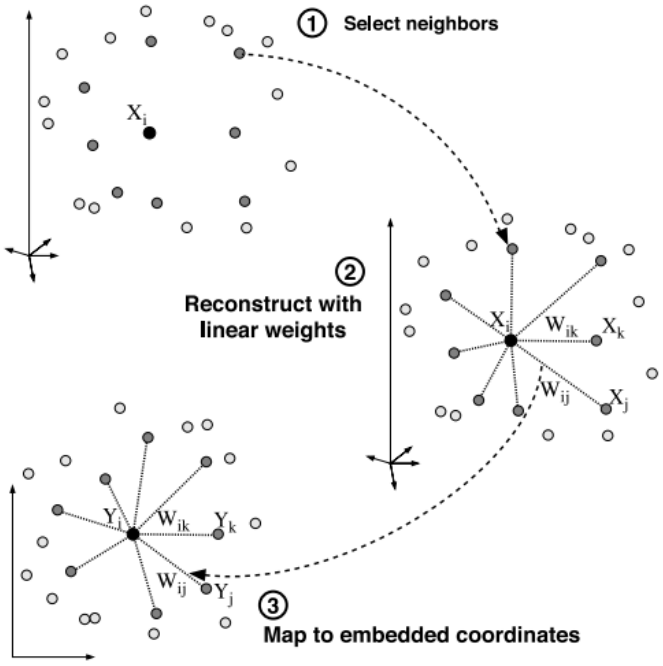


Figure 10: Locally Linear Embedding Algorithm *[32]*

The implementation of the above algorithm is as follows. In this derivation, to avoid confusion with the weight matrix, the snapshots set will be defined as, $X = \{\overrightarrow{X_\iota}\}_{i=1}^{m}$, $\overrightarrow{X_\iota} \in R^n$

Finding the Weights

The weight matrix W is found by introducing the following cost function for each point $\overrightarrow{X_\iota}$,

$$RSS_i(W) = \sum_i \left\| \overrightarrow{X_i} - \sum_j W_{ij} \overrightarrow{X_j} \right\|^2 \tag{43}$$

Where $\overrightarrow{X_\iota}$ represents the solution snapshots and W_{ij} is the weight matrix that quantifies the influence of a point's nearest neighbors in its reconstruction. These nearest neighbors are identified by Euclidean distance in the higher dimensional space. It is necessary however, that the distance by which a neighbor considered "near," does not break the manifold plane.

The weight matrix is computed by minimizing Equation (43) subject to two constraints. Specifically, equation (43) is minimized under the conditions that a point is reconstructed only by its neighbors, that is, $W_{ij} = 0$ if a neighbor $\overrightarrow{X_J}$ is not considered near, and that $\sum_j W_{ij} = 1$ (to maintain invariance under translation). Forcing $\sum_j W_{ij} = 1$ also has parallels in the setting of W as a stochastic transition matrix [51], a relationship that will be explored in the future investigation of diffusion maps. Continuing the minimization problem and utilizing the invariance under translation, define k as the number of nearest neighbors under consideration. Then, for $\overrightarrow{X_\iota}$ and all $\{\overrightarrow{X_J}\}_{j=1}^{k}$, the points are centered around the point to be reconstructed $\overrightarrow{X_\iota}$, such that the residual sum of squares is [33],

$$RSS_i = \left\| \sum_{j=1}^{k} W_{ij}(\vec{X}_j - \vec{X}_i) \right\|^2 = \left\| \sum_{j=1}^{k} W_{ij}\vec{z}_j \right\|^2 \tag{44}$$

Where $\vec{z_j} = \vec{X_j} - \vec{X_i}$. Taking the $\vec{z_j}$ to form a $k \times n$ matrix z and setting $\boldsymbol{W}_i$ as a $k \times 1$ matrix, the vector resulting from the summation in equation (44) is, $\boldsymbol{W}_i^T z$. Further,

$$RSS_i = \left\| \sum_{j=1}^{k} W_{ij}\vec{z}_j \right\|^2 = \left\| \boldsymbol{W}_i^T z \right\|^2 = \boldsymbol{W}_i^T z z^T \boldsymbol{W}_i \tag{45}$$

The symmetric $k \times k$ matrix zz^T is known as the Gram matrix (hereafter G_i) of the vectors $\vec{z_j}$ and contains all the inner products of the nearest neighbors of $\vec{X_i}$. Returning to the original minimization problem, minimizing

$$RSS_i = \boldsymbol{W}_i^T G_i \boldsymbol{W}_i \tag{46}$$

subject to its constraints, we use the vector **1** (a $k \times 1$) vector of ones (to transform the constraint $\sum_j W_{ij} = 1$ to $\mathbf{1}^T \boldsymbol{W}_i - 1 = 0$), and introduce the Lagrangian

$$L(\boldsymbol{W}_i, \lambda) = \boldsymbol{W}_i^T G_i \boldsymbol{W}_i - \lambda(\mathbf{1}^T \boldsymbol{W}_i - 1) \tag{47}$$

Taking the derivative results in,

$$\frac{dL}{d\boldsymbol{W}_i} = 2G_i \boldsymbol{W}_i - \lambda \mathbf{1} \qquad \Rightarrow \boldsymbol{W}_i = G_i^{-1} \frac{\lambda}{2} \mathbf{1} \tag{48}$$

Computing the Embedding

With the weight matrices generated by Equation (48) , the low dimensional embedding can be computed. The objective function for finding the optimal low dimensional embedding *Y*

is,

$$\phi(Y) = \min_{Y} \sum_{i} \left\| \vec{Y_i} - \sum_{j} W_{ij} \vec{Y_j} \right\|_2^2 \tag{49}$$

Similar to Equation (43), in Equation (49) the weight matrices are fixed, and the optimization concerns finding the optimal $\vec{Y_i}$. After performing some algebra, a form similar to that in the POD optimization problem in Section 3.2 is found.

$$\phi(Y) = \sum_{i=1}^{m} Y_i^2 - Y_i \left(\sum_{j=1}^{k} W_{ij} Y_j \right) - \left(\sum_{j=1}^{k} W_{ij} Y_j \right) Y_i + \left(\sum_{j=1}^{k} W_{ij} Y_j \right)^2 \tag{50}$$

$$= \boldsymbol{Y}^T \boldsymbol{Y} - \boldsymbol{Y}^T \left(\boldsymbol{W}\boldsymbol{Y} \right) - \left(\boldsymbol{W}\boldsymbol{Y} \right)^T \boldsymbol{Y} + \left(\boldsymbol{W}\boldsymbol{Y} \right)^T \left(\boldsymbol{W}\boldsymbol{Y} \right) \tag{51}$$

$$= \left((I - \boldsymbol{W})\boldsymbol{Y} \right)^T \left((I - \boldsymbol{W})\boldsymbol{Y} \right) = \boldsymbol{Y}^T (I - \boldsymbol{W})^T (I - \boldsymbol{W}) \boldsymbol{Y} \tag{52}$$

If the $m \times m$ matrix$(I - \boldsymbol{W})^T (I - \boldsymbol{W})$ is set as the matrix $\boldsymbol{M}$, then the minimization problem becomes,

$$\phi(\boldsymbol{Y}) = \min_{\boldsymbol{M}} \boldsymbol{Y}^T \boldsymbol{M} \boldsymbol{Y} \tag{53}$$

This is the same form as the maximization problem solved in Section 3.2, and the solution is found in the same manner, by introducing the Lagrangian under the constraint $\boldsymbol{Y}^T \boldsymbol{Y} = I$. However, as this is a minimization problem, as opposed to the maximization problem, the low dimensional vectors will be those associated with the smallest valued eigenvectors, as opposed to the large-valued eigenvectors.

Estimating the optimal size of the reduced embedding is not a trivial task unfortunately.

Eigendecompositions that utilize information associated with the bottom several eigenvalues do not exhibit the spectral gap properties associated with methods such as POD and diffusion maps [52]. Thus, finding the number of dimensions in the embedding can be helped by an estimate of the "intrinsic dimension" of the manifold space.

4.3 Reduced Order Modelling

Similar to as was performed using POD modes, the embedding vectors are used as the global basis functions in the Galerkin projection as described in Section 2.2. Once again, the system tested upon was on the interval $[0, 1](m)$ over the time domain $[0, 3](s)$. The boundary conditions are $u_a(x) = \sin(x)\,;\ \ u_b(x) = -\sin(x)$. The solution snapshot set consists of $\{W_i\}_{i=1}^{50} \in R^{300}$. The Reynolds number was set to 300. The LLE embedding was computed using different numbers of nearest neighbors on the snapshot set. In order to achieve the correct dimension of the basis functions, the data vectors were fed into the algorithm as 300 data vectors in dimension $m = 50$. Alternatively, the method of snapshots could have been utilized [40].

Figure 11 shows the error over time between the original snapshots and the reconstructed reduced models. The dynamics were generated using three modes for each implementation and compared to the POD reduction with three and five modes. From the results, The LLE reduced model using two nearest neighbors produces comparable results to the POD reduced model using an equivalent number of modes. However, the LLE generated modes were unable to produce the initial evolution of the system as well as the POD modes. Additionally, Figure 12 shows a reconstruction of the snapshots of the most accurate LLE dynamics at different times. The LLE modes were able to reconstruct the shock change in solution comparably well to the POD modes,

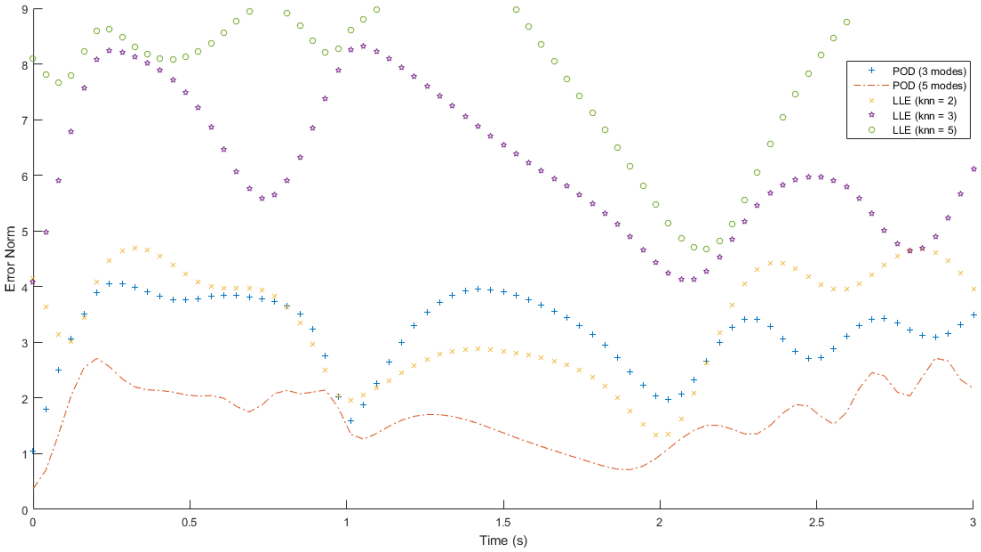


Figure 11: Error Between Reduced Model and Full with Different Numbers of Nearest Neighbors (knn = 2, 3, 5)

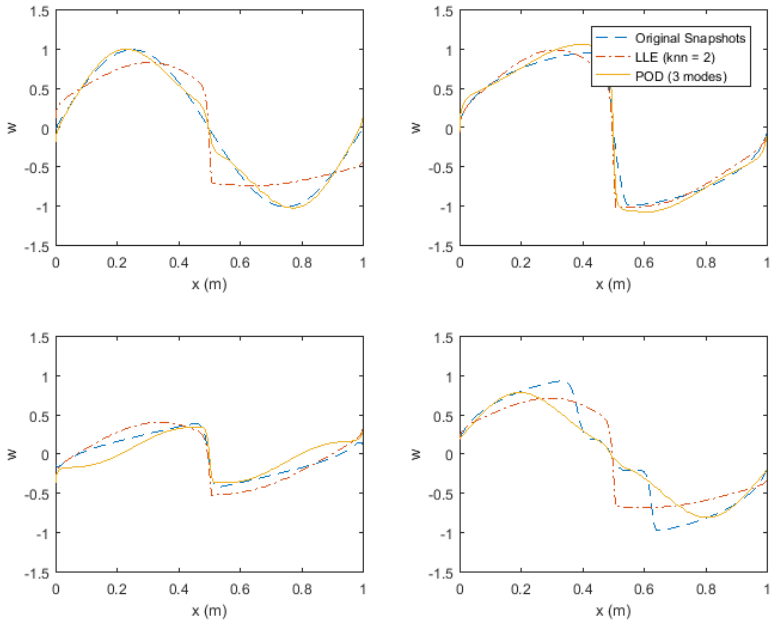


Figure 12: Reconstruction of LLE with 3 Modes and Two Nearest Neighbors (knn = 2)

but were unable to reconstruct the solution accurately at other points in the solution. Further, adding two additional modes to the POD model resulted in a clearly improved reconstruction. In an attempt to create a more accurate representation through LLE basis functions, different reduced models were created using still two nearest neighbors, but additional modes.

The results of these reduced models can be seen in Figure 13. As can be seen, it takes twice as many modes in the LLE representation to achieve similar accuracy to the POD reduced model with only five modes. Interestingly, across the time domain, introducing a fourth and fifth LLE mode results in worse performance on average. Possibly the fourth and fifth LLE modes are introducing dynamics that hold information on the initial and final condition of the dynamics, but that cause a conflict with previous modes. The modes used to generate these reduced models can be seen in Figure 14. The behavior captured by the LLE modes clearly differs from the structure captured by the modes in Figure 7.

Table 3 displays the relative captured energy and computation time of various LLE reconstructions as compared to the POD reduced dynamics. Table 3 displays the relative captured energy of the dynamics produced using LLE modes. While using the first three LLE modes with two nearest neighbors offers comparable performance, the first three POD modes are still superior. With the addition of two more POD mods, it takes up to ten LLE modes to compete. Additionally, LLE generally has a longer computation time.

Table 3: Relative Captured Energy and Computation Time for LLE Modes

Method	POD	POD	LLE (knn = 2)	LLE (knn = 5)	LLE (knn = 2)	LLE (knn = 5)
N	**3**	**5**	**3**	**3**	**5**	**10**
e	0.891	0.9687	0.8737	0.5392	0.5392	0.9661
τ	0.0234	0.0368	0.0317	0.0167	0.0277	0.0499

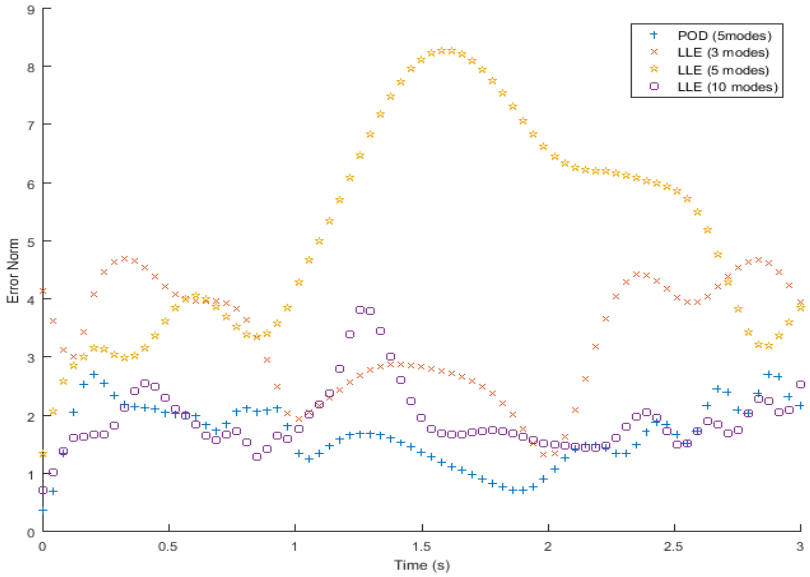


Figure 13: Reconstructed LLE Models Utilizing More Modes

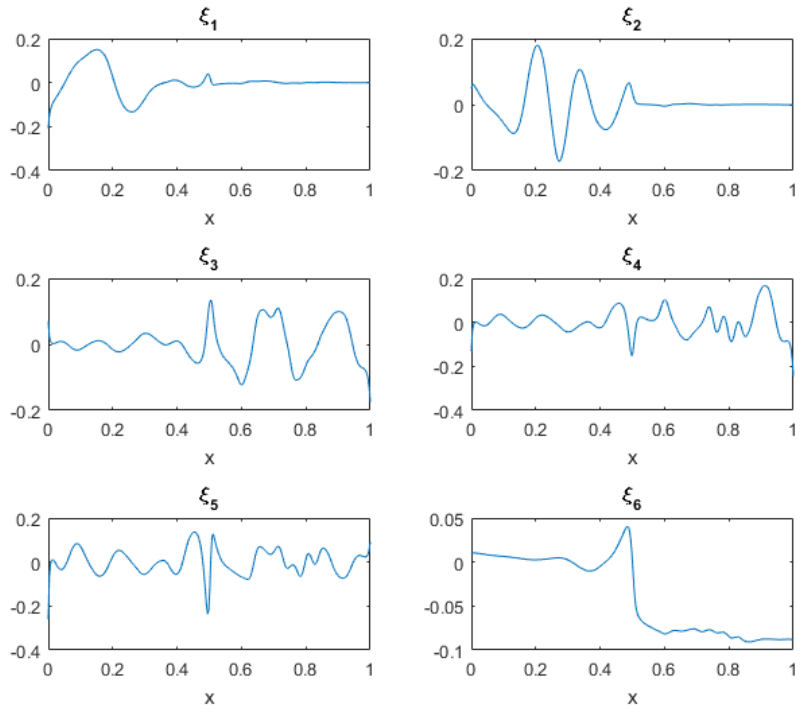


Figure 14: Modes Generated by LLE

Chapter 5 Diffusion Maps

5.1 Introduction

Diffusion maps is a dimensionality reduction technique developed by Coifman and Lafon that uses a Markov random walk to create a low dimensional embedding based on diffuse processes [35]. This allows for the representation of nonlinear manifolds in a low dimension, while still retaining a high degree of information about the underlying structure of the high dimensional representation [35] [49] [53]. An example of this can be seen below, in which the 3D Swiss Roll shown in Section 3.4 is projected into two dimensions using diffusion maps.

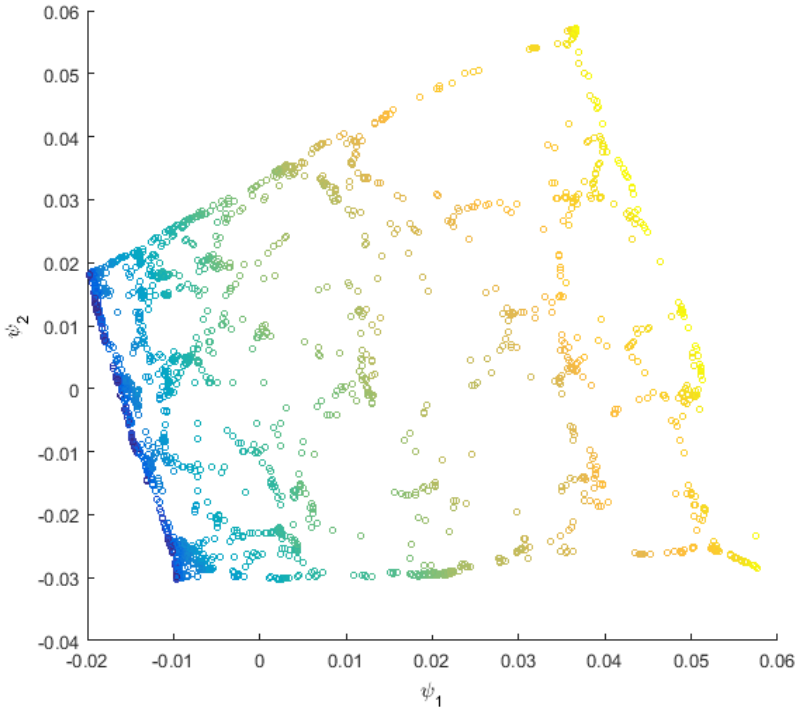


Figure 15: Swiss Roll Projected into 2D via Diffusion Maps

As with the LLE algorithm in Chapter 4, Diffusion Maps attempt to "unroll" the dataset.

5.2 The Diffusion Maps Method

Diffusion maps centers around an objective function utilizing the "diffusion distance" Quantifying the diffusion distance begins with creating a random walk as originally described in [35]. Given a dataset *X*, a kernel must be chosen such that k is symmetric and positivity preserving, i.e. $k(x,y) = k(y,x)$, and $k(x,y) \geq 0$. A popular kernel choice in the diffusion map algroithm is the Gaussian kernel [49] [53] [54] [36].

$$k\left(x,y\right) = exp\left[\left(\frac{d_{ij}}{\varepsilon}\right)^2\right] \tag{54}$$

In Equation (54), ε is a dimensionless scaling factor used to determine the connectivity of the data and d_{ij} is a chosen distance metric. In this thesis, the distance metric will be $\left\|\overrightarrow{X_i} - \overrightarrow{X_j}\right\|$, where $\|\cdot\|$ is the euclidean norm. The choice of the distance metric is application specific however, and other choices have been observed in literature such as the mahalanobis distance [55] [56].

Using the chosen kernel, a reversible Markov chain is created by constructing the normalized graph Laplacian [35] [57]. This begins with defining the degree $d(x_i)$ for each "node" of a dataset $X = \{X_i\}_{i=1}^{m} \in R^n$. These nodes are analogous to the snapshots previously mentioned. The degrees are defined as,

$$d\left(x_i\right) = \sum_{z \in X} k\left(x_i, z\right) \tag{55}$$

An $m \times m$ matrix P of transition probabilities is then constructed such that its entries are

defined as [35],

$$P_1\left(x_i, x_j\right) = \frac{k\left(x_i, x_j\right)}{d\left(x_i\right)} \tag{56}$$

P_1 can be defined as the probability of moving from x_i to x_j in one time-step. For a fully connected graph [58],

$$\lim_{t \to +\infty} P_t\left(x_i, x_j\right) = \pi\left(x_j\right) \tag{57}$$

where $\phi_0(y)$ is the stationary distribution and is computed as,

$$\pi\left(x_i\right) = \frac{d\left(x_i\right)}{\sum_{z \in X} d\left(z\right)} \tag{58}$$

Using the above, the diffusion distance can finally be defined as [35],

$$D_t^2\left(x_i, x_j\right) = \left| P_t\left(x_i\ ,\bullet\right) - P_t\left(x_j\ ,\bullet\right) \right|^2_{\frac{1}{\pi(x_i)}} = \sum_{y \in X} \frac{\left(P_t\left(x_i\ , y\right) - P_t\left(x_j, y\right)\right)^2}{\pi\left(y\right)} \tag{59}$$

In effect D_t is a, "functional weighted L^2 distance" [35]. The diffusion distance is based on a proximity in terms of graph connectivity for a diffuse process [57]. It follows that points in a low dimensional embedding based on this distance will be close if there are a large number of short paths between the two points in the original space. Otherwise stated, the points will be close if there is a high probability of moving from a point x to a point y and vice-versa [35]. Further, this distance holds an advantage over geodesic distances, also used in nonlinear model reduction, due to the fact that D_t involves summing over all paths connecting any two points.

Once the diffusion distance has been defined, the goal becomes to create a low-dimensional embedding such that snapshots that are in close proximity in the higher dimension in terms of their diffusion distance are in close proximity in a Euclidean sense in the lower dimensional space. The derivation of the embedding follows from the connection the diffusion distance has to the spectral theory of random walk [57]. The transition matrix P_t has a set of right and left eigenvectors and eigenvalues such that, $|\lambda_0| > |\lambda_1| > \cdots > |\lambda_{n-1}| > |\lambda_{n-1}| > |\lambda_n| = 0$ [35], and

$$\phi_j^T P_t = \lambda_j \phi_j^T \text{ and } \psi_j P_t = \lambda_j \psi_j \tag{60}$$

Further, it can be verified that, ψ_j is constant and $\lambda_0 = 1$. For convenience, the left eigenvectors of P_t are normalized with respect to $\frac{1}{\phi_0}$, and the right eigenvectors with respect to ϕ_0 such that,

$$\left\|\phi_l^{\,2}(x)\right\|_{\frac{1}{\pi(x)}} = \sum_x \frac{\phi_l^{\,2}(x)}{\pi(x)} = 1 \tag{61}$$

$$\left\|\psi_l^{\,2}(x)\right\|_{\frac{1}{\pi(x)}} = \sum_x \psi_l^{\,2}(x)\pi(x) = 1 \tag{62}$$

With a biorthogonal spectral decomposition of p_t from equation (56),

$$p_t(x,y) = \sum_{k\geq 0} \lambda_k^t \psi_k(x)\phi_k(y) \tag{63}$$

Inserting equation (63) into equation (59) yields,

$$D_t^2(x_i, x_j) = \sum_{y\in X} \frac{\left(P_t(x_i, y) - P_t(x_j, y)\right)^2}{\pi(y)}$$

$$= \sum_{y \in X} \frac{\left(\sum_{k=0}^{n-1} \lambda_k^t \psi_k(x_i) \phi_k(y) - \sum_{j=0}^{n-1} \lambda_k^t \psi_k(x_j) \phi_k(y) \right)^2}{\pi(y)}$$

$$= \sum_{y \in X} \frac{\left(\sum_{k=0}^{n-1} \lambda_k^{2t} \phi_k{}^2(y) \left(\psi_k(x_i) - \psi_k(x_j) \right)^2 \right)}{\pi(y)} \tag{64}$$

Using the identity from equation (61),

$$D_t^2(x_i, x_j) = \sum_{k=1}^{n-1} \lambda_k^{2t} \left(\psi_k(x_i) - \psi_k(x_j) \right)^2 \tag{65}$$

As the first left eigenvector ψ_0 is constant, it does not enter in the above sum. Further, as the eigenvalues decay, for a certain accuracy, only "*r*" eigenvalues and eigenvectors are potentially needed, such that D_t^2 becomes,

$$D_t^2(x_i, x_j) = \sum_{k=1}^{r} \lambda_k^{2t} \left(\psi_k(x_i) - \psi_k(x_j) \right)^2 \tag{66}$$

Equation (66) can be viewed as the Euclidean distance in R^k if the right eigenvectors are weighted with λ_j^t as coordinates on the data [57]. Otherwise stated, if the below diffusion mapping from the original space to the diffusion map space

$$\Psi_t : x \to \left\{ \lambda_1^t \psi_1(x) \;\; \lambda_2^t \psi_2(x) \;\; \ldots \;\; \lambda_r^t \psi_r(x) \right\}^T \tag{67}$$

is introduced, one can observe that,

$$D_t^2(x_i, x_j) = \sum_{k=1}^{r} \lambda_k^{2t} \left(\psi_k(x_i) - \psi_k(x_j) \right)^2 = \left\| \Psi_t(x_i) - \Psi_t(x_j) \right\|^2 \tag{68}$$

Equation (67) provides a lower dimensional embedding of the original points or snapshots in R^n into R^k.

5.3 Reduced Order Modelling

Now that the diffusion map machinery is in place, it can be implemented upon the snapshot data set to form the modes that will serve as the global basis functions, as was done for POD and LLE. Defining the same system conditions as have been used previously, a solutions set solution set $\{W_i\}_{i=1}^{50} \in R^{300}$ over the space domain $[0,1](m)$ and time domain $[0,3](s)$, with boundary conditions $u_a(x) = \sin(x)\,;\ u_b(x) = -\sin(x)$. Similar as was done with the LLE algorithm, the snapshots were fed into the algorithm as 300 data vectors with dimension 50. In the diffusion map embedding, the value of t in λ^{2t} was chosen as $t = 1$, a choice often made throughout the literature [36] [59].

A vital aspect of creating a reduced order model using diffusion maps is the choice of the correct constant ε in equation (54). The value of ε ultimately determines the proximity from which points on a manifold are considered similar. A suggestion in the literature for an appropriate weighting is some multiple of the minimum distance for the graph to remain fully connected [54] [59] [60]. In other words,

$$\varepsilon = nC = n\left(\max_{j}\min_{i\neq j} d_{ij}\right) \tag{69}$$

where d_{ij} is the distance in the Euclidean sense, as defined before, and n is some integer. This distance represents the maximum distance of any one point to its nearest neighbor.

In Figure 16, the reduced order model has been generated using three modes for different values of ε and compared against a POD model utilizing three modes. In Figure 17 a selection of reconstructed snapshots are shown. Similar to the performance of the modes generated from

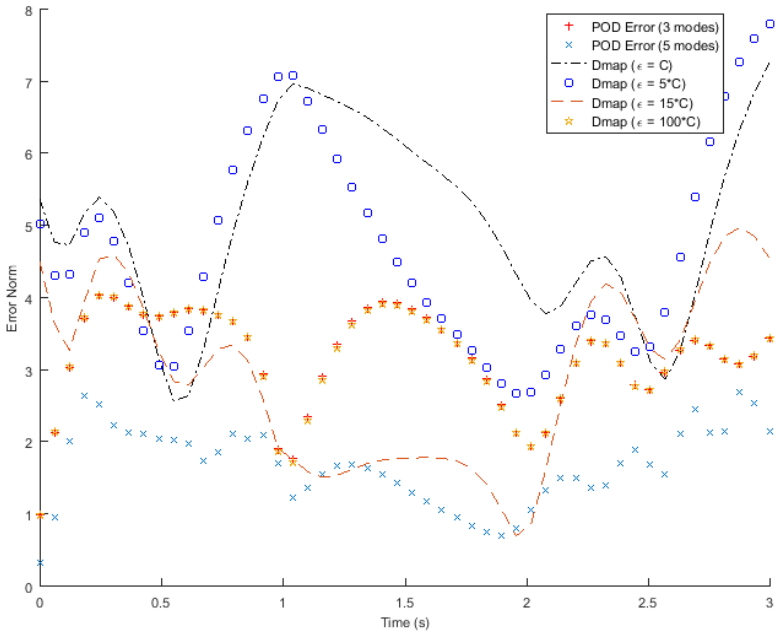


Figure 16: Diffusion Map Error vs POD Error

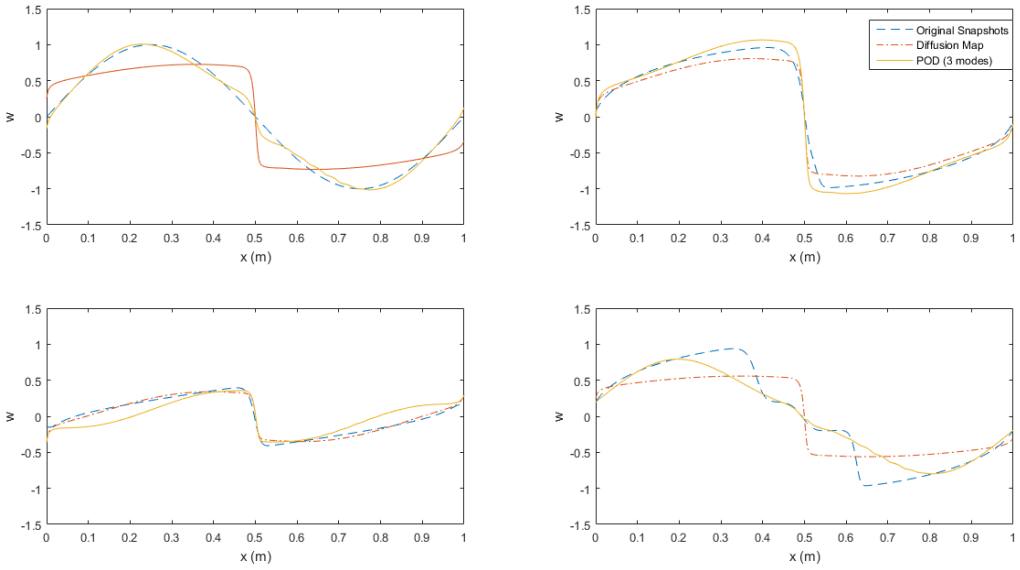


Figure 17: Snapshots of the Reconstructed DMAP System

LLE, the Diffusion Map modes struggle to recreate snapshots near the initial and final points in the simulation. Instead, the dynamics are attracted to the "shock" through the middle of the simulation. However, with a weighting to the similarity kernel of $\varepsilon = 15C$, the three diffusion map modes are able to approach the performance of three POD modes in terms of the error between the reconstruction and original. Note that as the distance weight grows, the performance of the diffusion map modes begins to approach the performance of the POD modes. This follows from the diffusion map process beginning to weight all points in the dataset similarly if the weight to the similarity measure is large enough.

Figure 18 shows the results of including a higher number of modes in the reduced dynamics. Of note, is that there is no significant difference between the reduced dynamics generated with two diffusion map modes and the dynamics generated with three modes. This suggests that redundant information is contained in the second and third modes, an idea supported in the literature concerning diffusion maps [23] [59] [60]. Using the knowledge that redundant modes exist, and further tuning the weight parameter, reduced dynamics utilizing the first, second, fourth, and sixth modes are generated and compared against POD modes utilizing the top three and five modes. The results can be seen in Figure 19.

From Figure 19, it can be seen that the Diffusion Map dynamics are able to outperform the POD dynamics in minimizing error with one less mode across the majority of the simulation. This performance comes with the caveat that time was invested in discovering the redundant modes. Suggestions for finding these redundant modes includes inspection to find linear dependence between new modes and unique modes [35] [59] [60]. However, this method becomes increasingly difficult as the number of modes continues to grow.

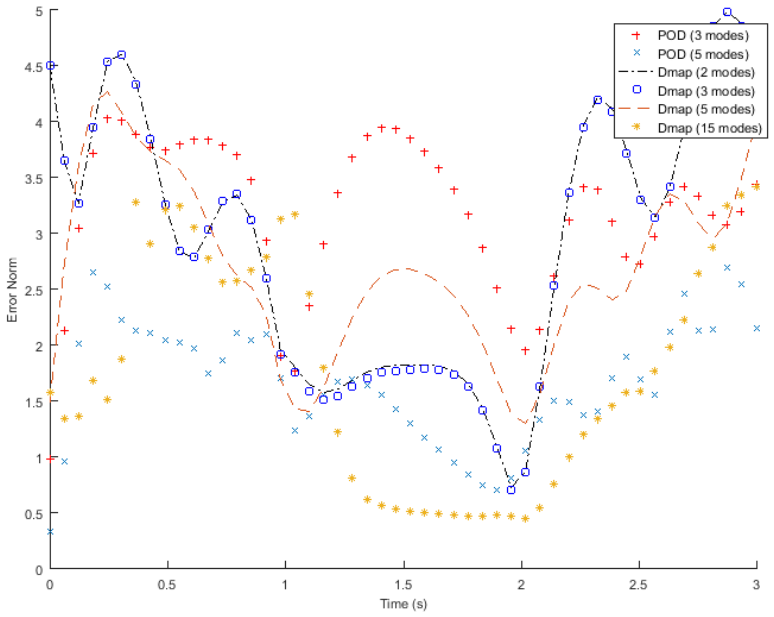


Figure 18: Reduced Dynamics Using Diffusion Maps (ϵ=15C) and Different Numbers of Modes

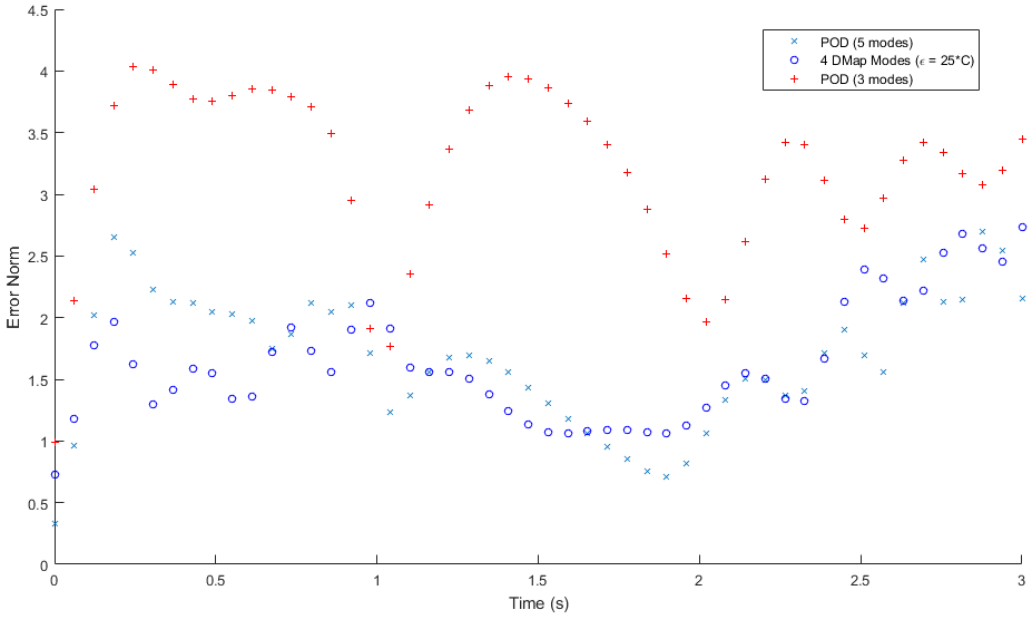


Figure 19: Diffusion Map Dynamics with Eliminated Redundant Modes

Figure 20 shows the reconstructed snapshots (generated using the four unique diffusion map modes) at various times, while Figure 21 shows a sampling of the diffusion map generated modes. Table 4 shows the relative captured energy of the reconstructed dynamics using the four unique diffusion map modes. With the elimination of redundant modes and the choice of the correct weighting parameter, the diffusion map modes are able to reconstruct the snapshots in the initial evolution of the dynamics, a task that the previous nonlinear modes struggled with. Further the relative captured energy of the dynamics generated using the Diffusion Map modes compare favorably with the POD dynamics.

Choice of Kernel

The scaling factor ε defined in equation (54) is a global scaling weight. However, the introduction of a different kernel may result in a more optimal weighting of the snapshots. To this end, the local similarity kernel in equation (70) is introduced as defined by Zelnik-Manor and Perona [61] and used to generate a new reduced model.

$$k(x,y) = \frac{-d(x,y)}{\sigma_x \sigma_y} \tag{70}$$

In equation (70), σ_x and σ_y are local scales defined by the density surrounding the points in question. Mirroring the demonstration by Zelnik-Manor and Perona, the local scales are

Table 4: Relative Captured Energy and Computation Time with Diffusion Map Reduced Dynamics

Method	POD	POD	Dmap (ε = 25*C)	Dmap (ε = 25*C)	Dmap (ε = 25*C)
N	**3**	**5**	**3**	**4**	**5**
e	0.891	0.9687	0.9007	0.9707	0.972
τ	0.0234	0.0368	0.0242	0.0319	0.0487

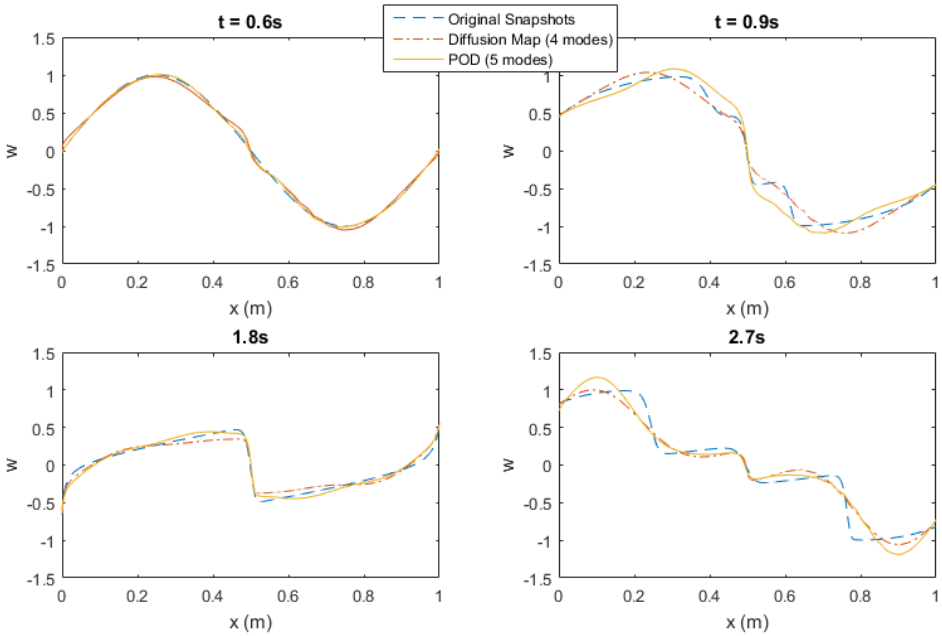


Figure 20: Reconstructed Diffusion Map Snapshots

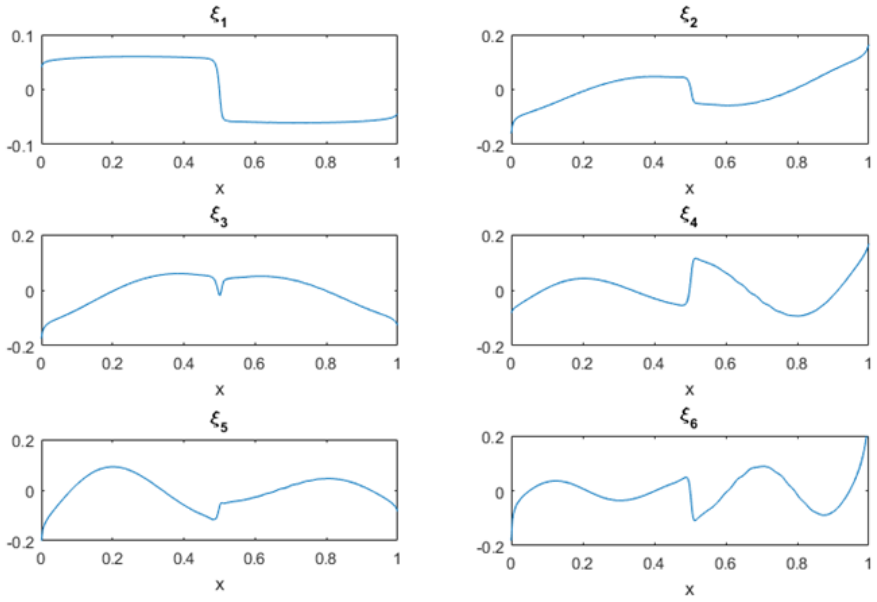


Figure 21: Sampling of Diffusion Map Generated Modes

defined by,

$$\sigma_x = d(x, x_k) \tag{71}$$

That is, σ_x is the distance between a point x it's k'th nearest neighbor. This kernel was used in the diffusion map implementation on the solution set generated from the full solution, in place of the global kernel. The local scaling was chosen based on the distance from the seventh nearest neighbor. Other choices of neighbors were investigated, but did not make significant differences in the quality of the basis functions generated. With the new kernel, the process of disregarding redundant eigenvectors was performed, and the error compared to POD dynamics, and the global kernel dynamics. The results are shown in Figure 22.

While the reduced dynamics generated with the new kernel using three modes showed comparable performance to the equivalent POD dynamics, modes generated using the local kernel were not able to outperform the global kernel. However, the choice of an improved kernel could result in superior performance than a general kernel. The choice of kernel would necessarily be application specific however.

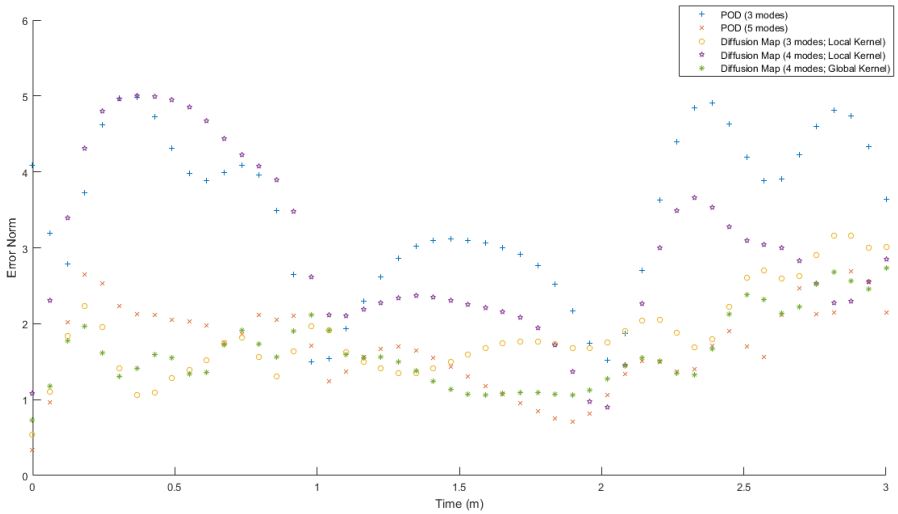


Figure 22: Error Between Diffusion Map Reconstructed Model and POD Reconstructed Mode

Chapter 6 Laplacian Eigenmaps

6.1 Introduction

Laplacian Eigenmaps is a nonlinear dimensionality reduction technique developed by Mikhail Belkin and Partha Niyogi [33]. Sharing similarities with both LLE and Diffusion Maps, Laplacian Eigenmaps utilizes both a neighborhood graph, as with LLE, to define the local reconstruction and the heat kernel to weight the neighbors and further quantify the influence in the reconstruction that each neighbor has. The heart of the method lies in the connection of the graph Laplacian, the Laplace-Beltrami operator, and the heat equation. In keeping with the convention of the rest of this thesis, Figure 23 demonstrates the results of the method by showing the projection of the 3D Swiss Roll to 2 dimension

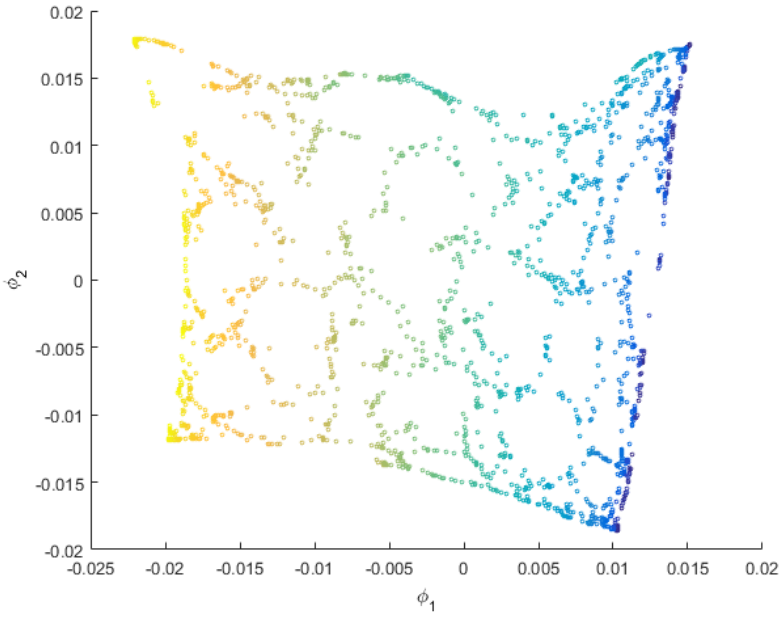


Figure 23: 2D Laplacian Eigenmap Projection of Swiss Roll Using 10 Nearest Neighbors

6.2 The Laplacian Eigenmaps Method

The algorithm for computing the Laplacian eigenmap embedding $Y = \{Y_i\}_{i=1}^{m} \in R^r$, from an initial snapshot set $X = \{X_i\}_{i=1}^{m} \in R^n$ begins with creating an adjacency graph similar to the construction in LLE. An edge is placed between adjacent points on the graph. Nearness of points can be determined by selecting all points within a maximum distance in a Euclidean sense or by selecting a set number of nearest neighbors.

Once the neighbors have been found, the weights of the neighbors are chosen. This can be done by utilizing the heat kernel so that for a weight matrix W [33],

$$W_{ij} = \exp\left(\frac{\left(\vec{X}_i - \vec{X}_j\right)}{\varepsilon}\right) \tag{72}$$

Where W_{ij} is the weight between a snapshot $\overrightarrow{X_i}$ and the snapshot $\overrightarrow{X_j}$. Meanwhile, ε is a scalar parameter that determines the connectedness of the graph.

Once the weights are computed, the eigenmapping can be found. Similar to the methods outlined in Chapters 3 - 5, a generalized eigenvalue problem of the form [33],

$$L\boldsymbol{f} = \lambda D\boldsymbol{f} \tag{73}$$

is solved, where D is a diagonal weight matrix with entries of column sums of W. $L = (D - W)$ is the positive semi-definite Laplacian matrix.

Optimality

Consider the mapping of a dataset $X = \left\{\vec{X}_i\right\}_{i=1}^{m} \in R^n$ onto a line $y = \{y_1, y_2, \ldots, y_m\}^T \in R$. The optimality of this mapping is defined by [62],

$$\min_{Y_i} \sum_{ij} \left(y_i - y_j\right)^2 W_{ij} \tag{74}$$

Where $W_{i,j}$ quantifies the weights between points in the higher dimensional space (based on the similarity measure in equation (75)). Further, as W_{ij} is symmetric and $D_{i,i} = \sum_{j=1}^{n} W_{j,i}$,

$$\begin{aligned}\min_{y_i} \sum_{ij} \left(y_i - y_j\right)^2 W_{ij} &= \min_{y_i} \sum_{ij} \left(y_i{}^2 - y_j{}^2 - 2y_i y_j\right)^2 W_{ij} \\ &= \min_{y_i} \left\{ \sum_i y_i{}^2 D_{ii} + \sum_j y_j{}^2 D_{jj} D_{jj} - \sum_{i,j} y_i y_j W_{ij} \right\} \\ &= \min_{\mathbf{y}} 2\mathbf{y}^T \boldsymbol{L}\mathbf{y}\end{aligned} \tag{75}$$

To this, the constraint $y^T Dy = 1$ is added. Extending this to an embedding into an r-dimensional Euclidean space $\{\vec{Y}_i\}_{i=1}^{m} \in R^r$, the objective function becomes,

$$\min_{\vec{Y}_i} \sum_{ij} \left\| \vec{Y}_i - \vec{Y}_j \right\|^2 W_{ij} = \min_{\mathbf{Y}} 2tr(\mathbf{Y}^T \boldsymbol{L}\mathbf{Y}) \tag{76}$$

With the constraint $\boldsymbol{Y}^T D\boldsymbol{Y} = I$. This optimization problem can be found by solving the generalized eigenvalue problem [63],

$$L\boldsymbol{Y} = \lambda D\boldsymbol{Y} \tag{77}$$

The Graph Laplacian

The previously mentioned graph Laplacian is analogous to the Laplace-Beltrami operator [33], a generalized operator on functions on manifolds in Euclidean space. The justification for the Laplacian Eigenmap method lies in the suitability of the eigenvectors of the Laplace-Beltrami operator for an embedding. Put another way, the eigenvectors are functions that maps nodes in the high dimensional space to the low dimensional space in such a way that points near in the high-dimensional space are assigned points near in the low-dimensional space. As an example, take two points $\vec{Z}$ and $\vec{X}$ mapped to $f(\vec{Z})$ and $f(\vec{X})$. Then, as shown by Belkin and Niyogi [33],

$$\left| f(\vec{Z}) - f(\vec{X}) \right| \leq \left\| \nabla f(\vec{X}) \right\| \left\| \vec{Z} - \vec{X} \right\| + o\left(\left\| \vec{Z} - \vec{X} \right\| \right) \tag{78}$$

Where the gradient ∇f is a vector in the tangent space such that given another vector $\vec{v}$, $df = \langle \nabla f(\vec{x}), \vec{v} \rangle$ [33]. Further, $\left\| \nabla f(\vec{X}) \right\|$ provides a measure for how far apart f maps nearby points [33]. Thus, it is desirable to minimize the gradient. To this end, the objective function becomes,

$$\min_{\vec{X}} \int \left\| \nabla f\left(\vec{X} \right) \right\|^2 \tag{79}$$

Further, Belkin and Niyogi go on to show that minimizing Equation (79) reduces to finding the eigenfunctions of the Laplace Beltrami operator.

6.3 Reduced Order Modelling

Utilizing Laplacian Eigenmap modes in conjunction with the Galerkin projection has basis in the literature. In [34], Pyta and Abel perform the Laplacian Eigenmap reduction on a lid-driven cavity problem. In creating the basis functions, they utilize the method of snapshots, performing the reduction on the snapshots in space rather than snapshots in time, to generate the temporal coefficients α of the Galerkin method. The coefficients are then transformed to the global basis functions using the transformation,

$$\vec{\xi} = \left(\alpha^T \alpha \right)^{-1} \alpha^T \vec{W} \tag{80}$$

With the method established, the global basis function were then computed for the model reduction. As with other methods, the spatial domain is set to $[0, 1](m)$ while the time domain is set to $[0, 3](s)$. The initial condition is $w_o(x) = \sin(2\pi x)$ with boundary conditions of, $u_a(t) = \sin(\pi x)$, $u_b(t) = -\sin(\pi x)$. The heat kernel weight and the number of nearest neighbors knn was varied, and the error between the Laplacian Eigenmap generated dynamics was compared against the POD dynamics. The results can be seen in Figure 24 and Figure 25.

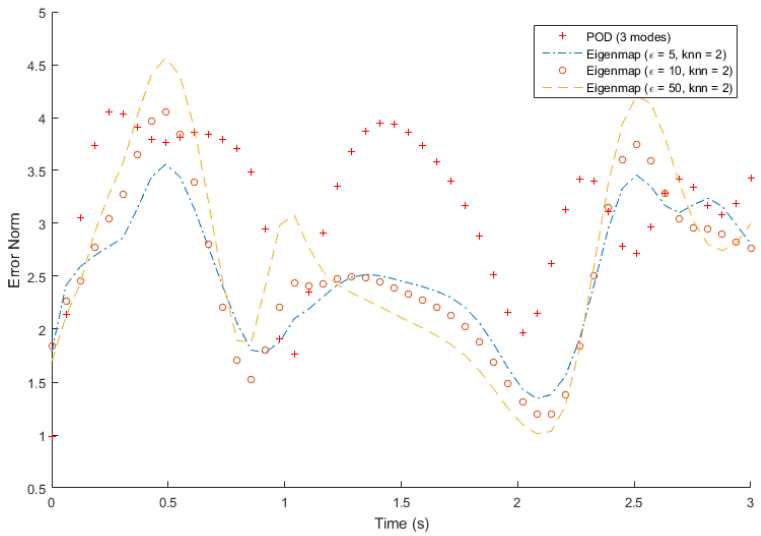


Figure 24: Error from Reduced Dynamics Using Laplacian Eigenmap Modes with Different Weights

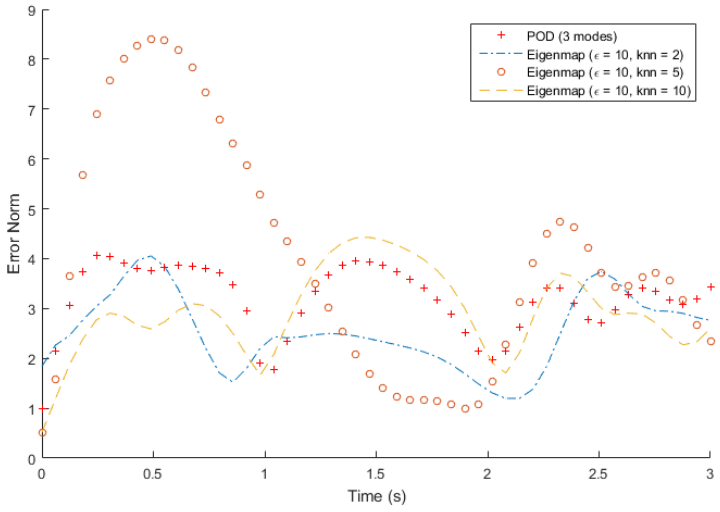


Figure 25: Laplacian Eigenmaps Generated Dynamics with Different Numbers of Nearest Neighbors (knn)

In the simulations, the Laplacian Eigenmap generated modes compared favorably against the POD generated modes, with different configurations of nearest neighbors and heat kernel weighting providing advantages against POD at different points in the simulation. The optimal configuration to use in the generation of the modes would most likely be application specific, depending on the dynamics to be reproduced. In Figure 26, the reduced dynamics were reproduced using different numbers of modes to test against POD produced dynamics using higher numbers of modes. No significant improvements could be seen in the Laplacian Eigenmap modes that offered an advantage over an equivalent number of POD modes. In Figure 27 and Figure 28, the top six modes generated by Laplacian Eigenmaps are seen for different weighting configurations.

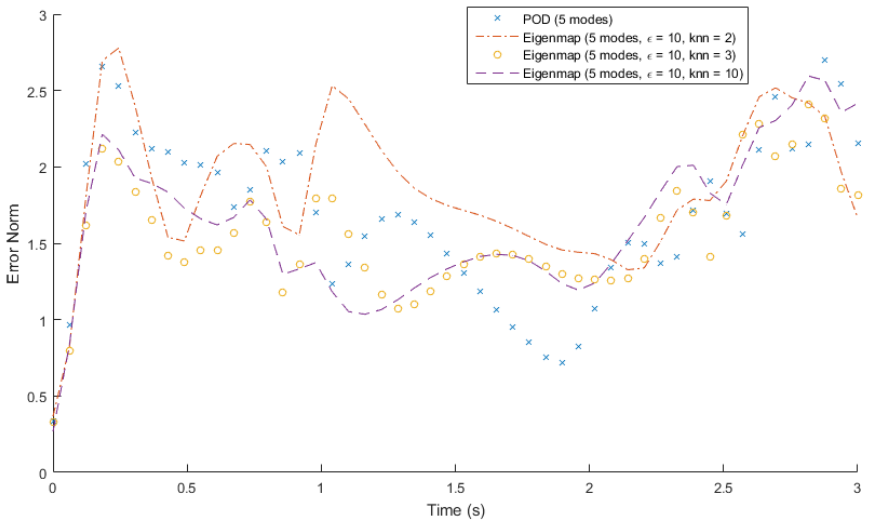


Figure 26: Dynamics Generated with Different Numbers of Laplacian Eigenmap Generated Modes (knn defines number of nearest neighbors considered)

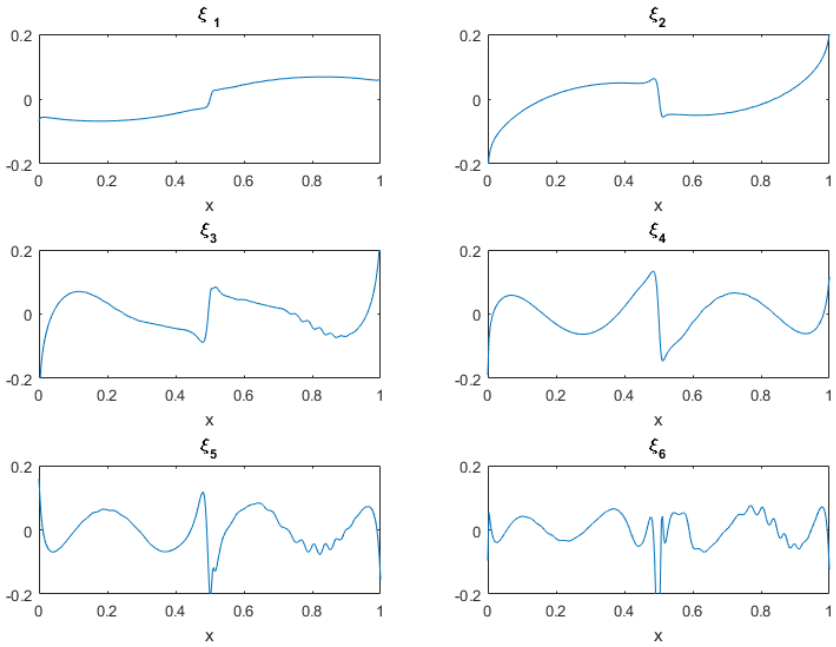


Figure 27: Laplacian Eigenmap Modes (knn = 2, ε = 10)

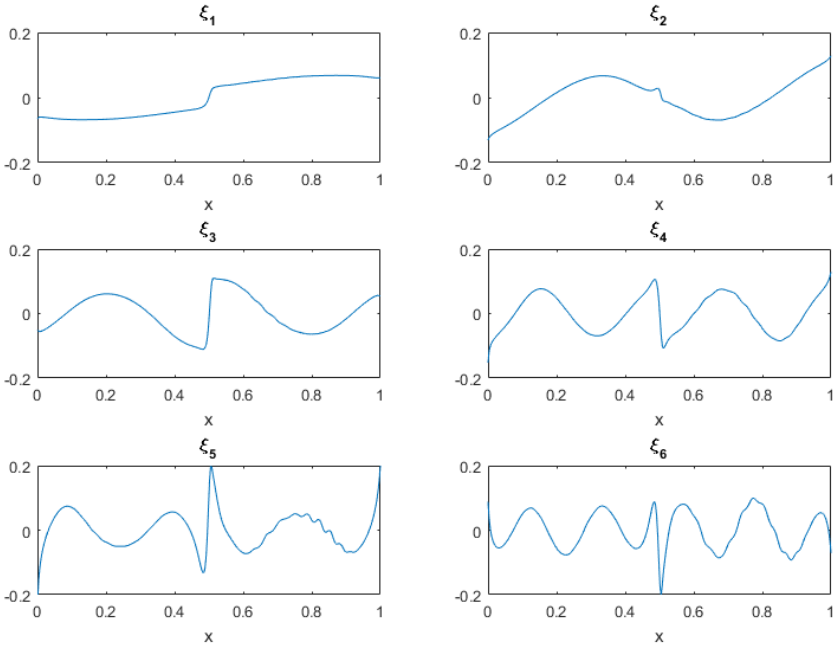


Figure 28: Laplacian Eigenmap Modes (knn = 15, ε = 10)

In Table 5, the relative captured energy of the reduced dynamics generated by the Laplacian Eigenmap modes can be seen. At three modes, the Laplacian Eigenmap generated modes outperform the equivalent number of POD modes. However, with the addition of the fourth and fifth POD modes, the POD dynamics are superior to the dynamics generated by the top five Laplacian Eigenmap modes.

Table 5: Relative Captured Energy and Computation Time of Laplacian Eigenmap Reduced Dynamics

Method	POD	POD	Emap ($\varepsilon = 10$, knn = 2)	Emap ($\varepsilon = 10$, knn = 10)	Emap ($\varepsilon = 10$, knn = 3)	Emap ($\varepsilon = 10$, knn = 10)
N	**3**	**5**	**3**	**3**	**5**	**5**
e	0.891	0.9687	0.9297	0.9068	0.9742	0.9712
τ	0.0234	0.0368	0.0379	0.0378	0.0487	0.0487

Chapter 7 Further Testing

7.1 Time Domain and Reynolds Number

All previous testing concerned simulations over the time domain $[0, 3]$ (s), a sampling rate of .06 seconds, and a Reynolds number of $Re = 300$. In this section, the performance of the nonlinear modes across a longer time domain and with different values of the Reynolds number is tested. The configurations of the reduction methods were tuned for better results.

Extending the Time Domain:

For a new set of simulations, the time domain was expanded to $[0, 10](s)$. The sampling rate was kept the same (0.06s), resulting in 167 solution snapshots, and the spatial discretization of 300 nodes was kept the same. The full order solution was still created using 500 sinusoidal modes. The boundary conditions of $u_a = \sin(\pi x)$, $u_a = -\sin(\pi x)$ and initial condition $w(x, 0) = \sin(2\pi x)$ were kept. The simulation was performed using both three and five modes using the best performing model reduction methods from previous testing. The results can be seen in Table 6 and Table 7. On the extended time domain, the POD modes outperform the nonlinear modes for all methods other than those generated by Diffusion Maps. However, the POD generated dynamics hold a slight advantage in computation time.

Increasing the Nonlinearity

The simulations were again performed on the domain $[0, 10](s)$, with equivalent initial and boundary conditions. However, the nonlinearity was increased by increasing the Reynolds number to $Re = 700$. This necessitated increasing the spatial discretization to 700 nodes. The results can be seen in Table 8 and Table 9. With the increase in the Reynolds number, the

Table 6: Reduced Dynamics From Nonlinear Modes on [0,10] (s) (3 modes, Re = 300)

Method	Full Solution	POD	LLE (knn = 2)	Dmap ($\varepsilon = 25*C$)	Emap (knn = 10, $\varepsilon = 10$)
N	-	**3**	**3**	**3**	**3**
e	1	0.8924	0.8603	0.9175	0.8747
τ (s)	115.4	0.0536	0.0531	0.0627	0.1153

Table 7: Reduced Dynamics from Nonlinear Modes on [0,10] (s) (5 modes, Re = 300)

Method	Full Solution	POD	LLE (knn = 2)	Dmap ($\varepsilon = 25*C$)	Emap (knn = 3, $\varepsilon = 10$)
N	-	**5**	**5**	**5**	**5**
e	1	0.9551	0.9213	0.9575	0.9584
τ (s)	115.4	0.0891	0.1114	0.1164	0.1387

Table 8: Reduced Dynamics From Nonlinear Modes on [0,10] (s) (3 modes, Re = 700)

Method	Full Solution	POD	LLE (knn = 2)	Dmap ($\varepsilon = 25*C$)	Emap (knn = 3, $\varepsilon = 5$)
N	**500**	**3**	**3**	**3**	**3**
e	1	0.8638	0.8287	0.8817	0.8743
τ (s)	304.77	0.0638	0.0518	0.0712	0.1029

Table 9: Reduced Dynamics From Nonlinear Modes on [0,10] (s) (5 modes, Re = 700)

Method	Full Solution	POD	LLE (knn = 2)	Dmap ($\varepsilon = 20*C$)	Emap (knn = 3, $\varepsilon = 5$)
N	**500**	**5**	**5**	**5**	**5**
e	1	0.9067	0.4573	0.9315	0.9473
τ (s)	304.77	0.1416	0.7279	0.1709	0.2357

nonlinear modes generated by the Diffusion Map and Laplacian Eigenmap model begin to show a clearer advantage over the POD modes, particularly with the inclusion of 5 modes. The nonlinear modes generally hold to their performance against the low Reynolds number dynamics, while the performance of the POD modes falls off with the growing nonlinearity. Computations using the POD modes continue to hold an advantage in computation time over the nonlinear modes however. The modes generated by the LLE process were generally inferior to the POD modes in all simulations run.

Increasing the Number of Snapshots

A final set of simulations were performed under equivalent conditions to those previously described, however, the solution manifold was sampled more densely. The manifold was sampled at a rate of .03 seconds, resulting in 300 solution snapshots. The results can be seen in Table 10 - Table 13. Similar advantages from the nonlinear modes can be seen. Particularly when the Reynolds number is large. The greater density of snapshots also results in improved performance by the Laplacian Eigenmap modes.

7.2 Predictive Models

A final measure of the ability of the nonlinear modes to capture fundamental aspects of the system dynamics is the ability to reproduce the dynamics of models generated under different initial conditions, rather than a comparison towards the specific solution set the snapshots are taken from. To examine this, different reduction methods were applied to a training set of data. This set of data was similar to the previous systems, with an initial condition of $w_o(x) = \sin(2\pi x)$ and boundary conditions $u_a(t) = \sin(\pi x)$, $u_b(t) = -\sin(\pi x)$. The Reynolds

Table 10: Reduced Dynamics From Nonlinear Modes on [0,10] (s)

(3 modes, Re = 300, 300 snapshots)

Method	Full Solution	POD	LLE (knn = 2)	Dmap (ε = 35*C)	Emap (knn = 3, ε = 0.5)
N	**500**	**3**	**3**	**3**	**3**
e	1	0.8599	0.8596	0.8892	0.9307
τ (s)	131.8	0.0612	0.0613	0.0804	0.1586

Table 11: Reduced Dynamics From Nonlinear Modes on [0,10] (s)

(5 modes, Re = 300, 300 snapshots)

Method	Full Solution	POD	LLE (knn = 2)	Dmap (ε = 25*C)	Emap (knn = 3, ε = 1)
N	**500**	**5**	**5**	**5**	**5**
e	1	0.955	0.597	0.9581	0.9744
τ (s)	131.8	0.1061	0.0859	0.1386	0.1322

Table 12: Reduced Dynamics From Nonlinear Modes on [0,10] (s)

(3 modes, Re = 700, 300 snapshots)

Method	Full Solution	POD	LLE (knn = 2)	Dmap (ε = 35*C)	Emap (knn = 3, ε = 15)
N	**500**	**3**	**3**	**3**	**3**
e	1	0.8638	0.8287	0.8817	0.8859
τ (s)	315.6	0.0624	0.0594	0.0931	0.1038

Table 13: Reduced Dynamics From Nonlinear Modes on [0,10] (s)

(5 modes, Re = 700, 300 snapshots)

Method	Full Solution	POD	LLE (knn = 2)	Dmap (ε = 35*C)	Emap (knn = 3, ε = 5)
N	**500**	**5**	**5**	**5**	**5**
e	1	0.8943	0.8605	0.9321	0.9473
τ (s)	315.6	0.1216	0.7974	0.1901	0.3548

number was set to 500, and 500 snapshots were sampled from the solution manifold. The basis functions were then generated from the training set of data. From the training set of data, the initial conditions were perturbed three times, and the dynamics were simulated using the Galerkin method based on the previously computed basis functions. The results were then averaged. This approach is similar to that applied to Laplacian Eigenmaps in the context of the Navier-Stokes equation in [34]. The results can be seen in Table. Under these conditions, the nonlinear modes generated by Diffusion Maps and Laplacian Eigenmaps significantly. Further, in the process of generating the reduced dynamics, as the initial conditions were perturbed farther from those in the training data set, the dynamics generated by the POD modes and Diffusion Map modes performed worse, while those generated by Laplacian Eigenmaps held a steady performance. This corroborates the results in [34]. The modes generated by LLE were not able to reconstruct the dynamics in a measure comparable to the other model reduction techniques.

Table 14: Relative Captured Energy of Models Produced Under Perturbed Initial Conditions

Method	Full Solution	POD	LLE (knn = 2)	Dmap ($\varepsilon = 50*C$)	Emap (knn = 10, $\varepsilon = 35$)
N	**500**	**5**	**5**	**5**	**5**
$\mathbf{e_1}$	1	0.6382	0.299	0.7237	0.8226
$\mathbf{e_2}$	1	0.6382	0.1597	0.7497	0.8449
$\mathbf{e_3}$	1	0.6769	0.252	0.7881	0.8159
$\mathbf{e_{avg}}$	1	0.6358	0.2369	0.7538	0.8278

Chapter 8 Conclusions

In this thesis, the one dimensional Burgers' equation was investigated in the context of reduced order models using the Galerkin Method. Global basis functions generated by Proper Orthogonal Decomposition were compared to basis functions generated by the nonlinear model reduction methods of Locally Linear Embedding, Diffusion Maps and Laplacian Eigenmaps. The Reduced order dynamics were compared over different time domains, different snapshot samplings, and different Reynolds numbers.

In the simulations, the nonlinear modes generated by Diffusion Maps and Laplacian Eigenmaps compared favorably in terms of accuracy against the traditional POD modes, particularly when the Reynolds number is larger. Modes generated by LLE however, were not compared to those generated from the other three methods. This possibly could be due to the binary nature of the adjacency graph used in LLE introducing an "averaging" effect in its attempts to capture the structure of the evolving data. In terms of computation time however, dynamics generated using POD modes were typically computed more quickly.

The benefits gained from the nonlinear modes occur only in the context of computing the dynamics via the Galerkin method. POD is optimal in the energy preservation sense in a direct linear projection. Further, the advantages of the nonlinear modes come with the caveat of necessary tuning of parameters. As the dynamics in question change, the weights and nearest neighbors used in the computation of the nonlinear modes will change. The is not a trivial requirement to accommodate between different systems. Further, utilizing Diffusion maps requires identification of the relevant modes before utilizing them as basis functions. In general,

these nonlinear embedding methods are more involved than the equal global weighting in POD and direct proportional ranking (in terms of information contained) of the POD modes.

Further applications of the nonlinear methods in the context of model reduction involve their application in simulating more complex dynamics such as the two dimensional Burgers' Equation, or the more complex Navier-Stokes equations. Further, the effect of the modes on the introduction of controls, both for linearized models and nonlinear optimal control such as in [64]. Additionally, improvements have been shown in POD by performing the method locally based on space-vector clustering [65]. Nonlinear model reduction techniques have shown promise in their application in clustering [66] [67], and thus could be incorporated in a clustering algorithm for the purposes of local reduction.

More research is needed in understanding the behavior embodied by the nonlinear modes and why they hold advantageous properties in the reduced order modelling of the 1D Burgers' equation. Clearly though, these model reduction schemes have the potential to offer great benefits in the analysis of fluid flow.

References

[1] R. C. Camphouse, "Boundary Feedback Control Using Proper Orthogonal Decmoposition Models," *Journal of Guidance, Control, and Dynamics,* vol. 28, no. 5, pp. 931-938, 2005.

[2] R. C. Camphouse and J. H. Myatt, "Reduced Order Modelling and Boundary Feedback Control of Nonlinear," *AIAA Guidance, Navigation, and Control Conference,* 15 - 18 August 2005.

[3] A. Antoulas, "Approximation of Large-Scale Dynamical Systems," in *Society for Industrial and Applied Mathematics*, Philadelphia, 2005.

[4] C. Kasnakoglu, A. Serrani and M. O. Efe, "Control input by actuation mode expansion for flow control problems," *International Journal of Control,* vol. 81, no. 9, pp. 1475-1492, 2008.

[5] S. Ahuja and C. W. Rowley, "Feedback control of unstable steady states of flow past a flat plate using reduced-order estimators," *Journal of Fluid Mechanics,* vol. 645, pp. 447-478, 2010.

[6] J. Atwell and B. King, "Computational Aspects of Reduced Order Feedback Controllers for Spatially Distributed Systems," *Proceedings of the 38th IEEE Conference on Control and Decision,* pp. 4301-4306, December 1999.

[7] J. Ausseur, J. Pinier and M. Glauser, "Flow Separation Control Using a Convection Based POD Approach," *AIAA Paper,* June 2006.

[8] H. Banks, R. del Rosario and R. Smith, "Reduced Order Model Feedback Control Design: Numerical Implementation in a Thin Shell Model," Center for Research in Scientific Computation, North Carolina State University, 1998.

[9] W. Stankiewicz, M. Morzynski, B. R. Noack and G. Tadmore, "Reduced order Galerkin models of flow around NACA-0012 airfoil"," *Mathematical Modeling and Analysis,* vol. 13, no. 1, pp. 113-122, 2008.

[10] K. A. Evans, "Reduced Order Controller for Distributed Parameter Systems, Ph.D Thesis," Department of Mathematics, Virginia Tech, 1999.

[11] J. A. Burns and B. B. King, "Optimal Sensor Location for Robust control of Distributed Parameter Systems," in *Proceedings of the 33rd Conference on Decision and Control*, Lake Buena Vista, FL, 1994.

[12] J. Borggaard, J. A. Burns and L. Zietsman, "On Using LQG performance Metrics for Sensor Placement Cntrol," in *Proceedings of the 2011 American Control Conference*, San Fransico, CA, 2011.

[13] J. Borggaard, M. Stoyanov and L. Zietsman, "Linear Feedback Control of a von Karman Street by a Cylinder Rotation," in *Proceedings of the 2010 American Control Conference*, Baltimore, MD, 2010.

[14] D. Kirk, in *Optimal Control Theory*, New York, Dover Publications, 2004, pp. 90-93.

[15] D. A. Lawrence, "Empirical Model Reduction for Active Closed-Loop Flow Control," AFRL, July, 2004.

[16] K. Hulsing, "Methods for Computing Functional Gains for LQR Control of PDEs, Ph.D Thesis," Department of Mathematics, Virginia Tech, 1999.

[17] I. Kalashnikova and S. Arunajatesan, "A Stable Galerkin Reduced Order Model (ROM) for Compressible Flow," in *10th World Congress for Computational Mechanics (WCCM)*, Sao Paulo, Brazil, 2012.

[18] M. F. Barone, I. Kalashnikova, D. Segalman and T. H., "Stable Galerkin Reduced Order Models for Linearized Compressible Flow," *Journal of Computational Physics*, vol. 288, pp. 1932-1946, 2009.

[19] I. Kalashnikova and M. Barone, "Stable and Efficient Galerkin Reduced Order Model for Non-Linear Fluid Flow," in *6th AIAA Theoretical Fluid Mechanics Conference*, Honolulu, HI, June 2011.

[20] S. Sahyoun, C. Nelson and S. K. T. Djouadi, "Control and Room Temperature Optimization of Energy Efficient Building," in *Proceedings of the IEEE Multi-conference on Systems and Control*, Dubrovnik, Croatia, October 2012.

[21] K. Kunisch and S. Volkwein, "Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics," *SIAM Journal of Numerical Analysis*, vol. 40, no. 2, pp. 492-515, 2002.

[22] F. Leibfritz and S. Volkwein, "Reduced order output feedback control design for PDE systems using proper orthogonal decomposition and nonlinear semidefinite programming," in *Linear Algebra and its Applications*, 2006, pp. 542-575.

[23] C. Gu and J. Roychowdhury, "Model reduction via projection onto nonlinear manifolds, with applications to analog circuits and biomechanical systems.," *Proceedings of the 2008 IEEE-ACM International Conference on Computer-Aided Design*, p. 8592, 2008.

[24] B. Noack, M. Schlegel, M. Morzynski and G. Tadmore, Galerkin Method for Nonlinear Dynamics in Reduced Order Modelling for flow Control, Wien-New York: Springer, 2011.

[25] M. Dihlmann, M. Drohmann and B. Haasdonk, "Model reduction of parameterized evolution problems using the reduced basis method with adaptive time partitioning," Stuttgart Research Centre for Simulation Technology, 2011.

[26] J. N. Tenenbaum, V. de Silva and J. Langford, "A Global Geometric Framework for Bonlinear Dimensionality Reduction," *Science*, vol. 290, pp. 2319-2323, 2000.

[27] M. Striebel and J. Rommes, *Model Order Reduction of Nonlinear Systems: Status, Open Issues, and Applications*, Chemnitz Scientific Computing Preprints.

[28] D. Amsallem, M. Zahr and C. Farhat, *Nonlinear Model Order Reduction Based on Local Reduced-Order Bases*, http://www.stanford.edu/amsallem/LocalROBpreprint.pdf..

[29] A. D., "Stabilization of projection-based reduced-order models," *International Journal for Numerical Methods in Engineering*, pp. 1-22, 2011.

[30] K. Carlberga, C. Fahratb, J. Cortiala and D. Amsallem, *The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows,* http : //arxiv.org/abs/1207.1349.

[31] L. K. Saul and S. T. Roweis, "An Introduction to Locally Linear Embedding," Available from http://wol.ra.phy.cam.ac.uk/pub/mackay/..

[32] S. Roweis and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, pp. 2323-2326, 200.

[33] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation,* vol. 15, pp. 1373-1396, 2003.

[34] L. Pyta and D. Abel, "Nonlinear Model Reduction of the Navier-Stokes-Equations," *American Control Conference,* pp. 5249-5254, July, 2016.

[35] R. Coifman and S. Lafon, "Diffusion Maps," *Applied and Computational Harmonic Analysis..*

[36] B. E. Sonday, A. Singer, C. W. Gear and a. I. G. Kevrekidis, *Manifold Learning Techniques and Model Reduction Applied to Dissipative PDEs,* arXiv:1011.5197 [physics.comp-ph], 2010.

[37] E. Hopf, "A mathematical example displaying the features of turbulence," *Comm. Pure Appl. Math.,* vol. 1, pp. 303-322, 1948.

[38] L. D. Landau and E. M. Lifschitz, Fluid Mechanics, Pergamon Press, 1987.

[39] S. E. Newhouse, D. Ruelle and F. Takens, "Occurence of stange axiom A attractors near quasiperiodic flows on Tm, m > 3," *Comm. Math. Phys.,* vol. 64, pp. 35-40, 1978.

[40] P. Holmes, Turbulence, coherent structures, dynamical systems and symmetry, New York: Cambridge University Press , 2012.

[41] S. Kutluay, A. R. Bahadir and A. Ozdes, "Numerical solution of one-dimensional Burgers equation: explicit and exact-explicit finite difference methods," *Journal of Computational and Applied Mathematics,* vol. 103, no. 2, pp. 251 - 261, 1999.

[42] U. Lorenzo-Seva, "How to Report the Percentage of Explained Common Variance in Exploratory Factor Analysis," Universitat Rovira i Virgili, Tarragona, 2003.

[43] Y. C. Liang, H. Lee, W. Lin, K. H. Lee, C. G. Wu and S. Lim, "Proper Orthogonal Decomposition and its Applications - Part I: Theory," *J. Sound Vib.,* vol. 252, no. 3, p. 527–544, 2002.

[44] R. Pinnau, "Model Reduction via Proper Orthogonal," in *Model Order Reduction: Theory, Research Aspects and Applications,* Springer, 2008, p. 96–109.

[45] C. R. Shalizi, Principal Components Analysis, Cambridge University Press, 2016.

[46] E. Kokiopoulou, J. Chen and Y. Saad, "Trace Optimization and Eigenproblems in Dimension Reduction Methods," University of Minnesota, 2010.

[47] K. Afanasiev and M. Hinze, "Adaptive Control of a Wake Flow using Proper Orthogonal Decomposition," in *Lecture Notes in Pure and Applied Mathematics 216*, Marcel Dekker, 2001, pp. 317-332.

[48] L. Sirovich, "Turbulence and the Dynamics of Coherent Structures," *Quarterly of Applied Mathematics,* vol. XLV, no. 3, pp. 561-590, 1987.

[49] B. Nadler, S. Lafon, R. Coifman and I. G. Kevrekidis, "Diffusion Maps - a Probabilistic Interpretation for Spectral Embedding and Clustering Algorithms," in *Principal Manifolds for Data Visualization and Dimension Reduction*.

[50] G. Kerschen and J.-C. Golinval, "The Method of Proper Orthogonal Decomposition for Dynamical Characterizationand Order Reduction of Mechanical Systems: An Overview," *Nonlinear Dynamics,* vol. 41, pp. 147-169, 2005.

[51] *Nonlinear Dimensionality Reduction I: Local Linear Embedding,* The Pennsylvania State University , 2009.

[52] L. Xiao, J. Sun and S. Boyd, "A duality view of spectral methods for dimensionality reduction," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 1041-1048.

[53] D. l. Porte, J. Herbst, B. Hereman and W. v. d. Walt, "An Introduction to Diffusion Maps," in *Proceedings, International*, 2008.

[54] E. Chiavazzo, C. W. Gear, B. E. Sonday and I. G. Kevrekidis, "A Manifold Learning Approach to Model Reduction of Combustion," Available from http://modelreduction.net/wp-content/uploads/2013/04/Chiavazzo.pdf.

[55] W. Lian, R. Talmon, H. Zaveri, L. Carin and R. Coifman, "Multivariate Time-Series Analysis and Diffusion Maps," *Signal Processing,* p. 13–28, 2015.

[56] C. J. Dsilva, R. Talmon, C. W. Gear and R. R. Coifman, *Data-Driven Reduction for Multiscale Stochastic Dynamical Systems,* arXiv preprint arXiv:1501.05195, 2015.

[57] A. Lee and S. Lafon, "Diffusion Maps and Coarse-Graining: A Unified Framework for Dimensionality Reduction Graph Partitioning and Data Set Parameterization," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 28, no. 9, pp. 1393-1403, 2006.

[58] F. Chung, *Lectures on Spectral Graph Theory*, University of Philidelphia: Lecture.

[59] E. Chiavazzo, C. Gear, C. Dsilva, N. Rabin and I. Kevrekidis, "Reduced Models in Chemical Kinetics via Nonlinear Data-Mining," *Processes*, pp. 112-140, 2014.

[60] C. W. Gear, I. G. Kevrekidis and B. E. Sonday, "Slow Manifold Integration on a Diffusion Map Parameterization," in *AIP Conference Proceedings*, 2011.

[61] L. Zelnik-Manor and P. Perona, "Self-Tuning Spectral Clustering," *Advances in Neural Information Processing Systems,* vol. 17, pp. 1601-1608, 2005.

[62] R. Sorcher and M. Hein, "Manifold Learning and Dimensionality Reduction with Diffusion Maps," Saarland University, 2008.

[63] E. Kokiopoulou, J. Chen and Y. Saad, "Trace Optimization and Eigenproblems in Dimension Reduction Methods," *Numerical Linear Algebra with Applicacations,* vol. 18, no. 3, pp. 565-602, 2011.

[64] S. Sahyoun and S. Djouadi, "Control of Nonlinear PDEs based on Space Vectors Clustering reduced order systems," in *The International Federation of Automatic Control*, Cape Town, South Africa, 2014.

[65] S. Sahyoun and S. Djouadi, "Nonlinear Model Reduction Using Space Vectors Clustering POD with Application to the Burger's Equation," in *American Control Conference*, Portland, 2014.

[66] B. Nadler, S. Coifman and I. Kevrekidis, "Diffusion maps, spectral clustering and eigenfunctions of Fokker-PlanckOperators," in *Advances in Neural Information Processing Systems 18*, MIT Press, 2006, pp. 955-962.

[67] T. G. Dietterich, S. Becker and Z. Ghahramani, "Laplcian eigenmaps nd spectral techniques for embeding and clustering," *Advances in Neural Information Processing Systems*, vol. 14, 2002.

[68] N. D. Monniga, B. Fornberga and F. G. Meyer, "Inverting Nonlinear Dimensionality Reduction with Scale-Free Radial Basis Function Interpolation," *Applied and Computational Harmonic Analysis,* vol. 37, no. 1, pp. 162-170, 2014.

[69] A. Ferguson, A. Panagiotopoulos, . I. Kevrekidis and P. Debenedetti, "Nonlinear dimensionality reduction in molecular simulation: The diffusion," *Chemical Physical Letters,* vol. 509, no. 1-3, pp. 1-11, 2011.

[70] H. Huang, H. Qin, S. Yoo and D. Yu, "A Robust Clustering Algorithm Based on Aggregated Heat Kernel Mapping," in *11th IEEE International Conference on Data Mining*, 2011.

[71] S. Chaturantabut and D. C. Sorenson, "Non-linear model reduction via discrete emperical interpolation," *SIAM Journal on Scientific Computing,* vol. 32, pp. 2737-2764, 2010.

[72] D. Granata and V. Carnevale, "Accurate Estimation of the Intrinsic Dimension Using Graph Distances: Unraveling the Geometric Complexity of Datasets," *Scientific Reports,* vol. 6 [Published Online] doi: 10.1038/srep31377, 2016.

Vita

Christopher Winstead was born in Memphis, Tennessee to parents Joel and Suzanne Winstead. Holding a passion for science through an early age, he graduated Houston High School in spring of 2010 and was accepted to The University of Tennessee for the following Fall. At the University of Tennessee, Christopher pursued a degree in Electrical Engineering, and along the way discovered a newfound passion for mathematics, and ultimately graduated with a Bachelor of Science in Electrical Engineering, with a minor in Mathematics in Spring 2014. Looking to continue his education, Christopher applied and was accepted to the Electrical Engineer Master's program at the University of Tennessee, supplementing his research with Teaching and Research Assistantships at the university and nearby Oak Ridge National Laboratory.